On the use of stochastic systems for sensing and security

by

Lachlan J. Gunn

Bachelor of Electrical and Electronic Engineering (Hons)
Bachelor of Mathematical and Computer Sciences (Pure)
The University of Adelaide, 2012

Thesis submitted for the degree of

Doctor of Philosophy
in
Electrical and Electronic Engineering

The University of Adelaide

2017

Contents

Abstract

No measurement system is perfect, and two varieties of error compete to frustrate their designers and operators. Random errors produce measurement-to-measurement to variation, while systematic errors result in consistently-incorrect results.

The interplay between these two phenomena has been the subject of research for many years, particularly within the area of *stochastic resonance*, which focusses upon cases where the signal-to-noise ratio of a nonlinear system can increase with the addition of noise to its input signal. While it has been demonstrated many times that noise can overcome systematic deficiencies in a measurement system, there remain open questions on how to take advantage of this in practical systems, what information can be extracted, and whether such 'randomised' systems are useful in other settings.

In this thesis, we consider this general theme in the context of two main settings: the adversarial, and the nonadversarial. In both cases, there is a significant advantage to be gained from the use of techniques that are adapted to the problem domain, in contrast to previous ad-hoc approaches that have failed to take advantage of the structures of the problems at hand.

The first part of this thesis considers the elimination of static nonlinearity from noisy measurements. We start with the phenomenon of 'classical' stochastic resonance, showing how input noise can be used to linearise the response of a nonlinear system. This phenomenon has been observed in the past, however we demonstrate that the use of nonlinear signal processing allows the linearisation to take place with far smaller levels of noise. We then investigate several approaches to the implementation of this technique, with the aim of supporting real-time operation in embedded systems and VLSI.

The remainder of the thesis concerns the use of randomness in measurements made as part of adversarial systems. This can be split into two situations: that where the operation of a system requires that measurement be difficult, and that where measurement must be straightforward. We first discuss the Kish key distribution system, a proposed classical alternative to quantum key distribution. This system claims to derive its security from the second law of thermodynamics, however these claims have been the subject of controversy. We examine the claims in detail, and show that the use of random signals does not render implausible the measurement of the system state.

Finally, we describe a number of approaches to the topical problems of key distribution and identity verification. We show how various forms of multi-path probing can be treated as a form of random sampling; much like in the first section, this randomness allows for the characterisation of systematic errors, in this case the consistent changes introduced by an attacker. We then compute bounds on the probability that an attacker achieves a deception against a user taking part in this sampling process.

The first approach that we consider uses an anonymising system such as Tor or a mix-net; if all users make anonymous requests to a service in lock-step, then a malicious service cannot guarantee a self-consistent set of responses to anyone without providing the malicious response to all users. This allows the development of a statistically guaranteed consensus, and thus permits auditors to assure themselves that they have examined the same data as has been provided to other users. This provides an attractive alternative to blockchain technology, avoiding the complexity of the proof-of-work and proof-of-stake-based systems that dominate the landscape today.

We have developed a second approach that allows the random-sampling approach to be used with the existing public-key infrastructure. By demonstrating that the entities chosen to carry out the verification of an identity holder are selected at random from a substantial number of independent entities, relying parties can be confident that small numbers of compromised verifiers cannot unilaterally issue certificates for identities that they do not hold. This provides a basis for the development of highly robust distributed certificate issuance systems that do not share the current 'weakest-link' nature of the existing public-key infrastructure.

Ultimately, these systems all hold in common the use of randomness in their measurement conditions in order to characterise systematic effects. While this phenomenon has been acknowledged, its potential to characterise real systems has until now not been realised. We demonstrate that randomness, whether natural and unavoidable or artificially introduced, can ironically render far more predictable the behaviour of many systems, and in more realistic situations than have been seen in the literature to date.

Statement of Originality

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I give consent to this copy of my thesis when deposited in the University Library, being made available for loan and photocopying, subject to the provisions of the Copyright Act 1968.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

I acknowledge the support I have received for my research through the provision of an Australian Government Research Training Program Scholarship, and an Endeavour Research Fellowship.

2017-09-05

Signed

Date

Acknowledgments

This thesis is the culmination of several years of work, but would not exist without the tremendous support of a great many people and organisations.

Above all, the greatest of thanks go to my supervisors, Derek Abbott and Andrew Allison, to whom I owe much of my academic development; it was they who bore the weighty responsibility of transforming a newly-graduated engineer into an academic, and without their endless toil and advice, I would not have reached this point. *Primary Supervision*

During my PhD, I was fortunate enough to spend six months at the University of Angers, under François Chapeau-Blondeau. His excellent supervision helped give me new perspective on my work, making my time in Angers some of the most productive of my Ph.D. *Supervision in Angers*

Thanks are also due to the Australian Government for the award of an *Australian Postgraduate Award* in 2013 and an *Endeavour Research Fellowship* in 2015, as well as to the School of Electrical and Electronic Engineering of the University of Adelaide for providing travel funding. *Funding Sources*

I would also like to thank the Australian Mathematical Sciences Institute, who organised and supported my attendance at the *2014 AMSI Summer School*. As well as the courses presented at this event, the interesting discussions that I had with both fellow students and giants of the field, such as Daniel J. Bernstein and Tanja Lange, were most illuminating, and it is to this opportunity—giving me the chance to see the points of view of proponents of both conventional and alternative approaches to cryptography—more than anything else that I owe the perspective on cybersecurity that I have gained throughout the course of my research.

I owe also a great deal to those with whom I have collaborated over the last few years, both in Australia and abroad. In particular, I offer my thanks to Waddah Al-Ashwal, M. Ali Babar, François Chapeau-Blondeau, James Chappell, Bruce Davis, Alex Dinovitser, Samuel Drake, Fabing Duan, John Hartnett, Azhar Iqbal, Mark MacDonnell, Olaf Maennel, Heiki Pikker, Cameron Seidel, Tony Vladusich, and Liyan Xu. *Coauthors*

Talks and Visits

I have been graciously hosted by and given talks to number of research groups. While space restricts me from listing all those with whom I have had so many most stimulating discussions, I would like to give credit to my hosts: Nigel Stocks, University of Warwick; Peter McClintock and Aneta Stefanovska, University of Lancaster; Jason Ralph, University of Liverpool; Raúl Toral, University of the Balearic Islands; Juan Parrondo, Complutense University of Madrid; François Chapeau-Blondeau, University of Angers; Luigi Fortuna, University of Catania; and Zoltan Gingl and Robert Mingesz, University of Szeged.

Technical Discussions

Beyond this, I owe much to those with whom I have had many technical discussions; this was especially so with Laszlo Kish, who introduced the system that occupies most of Chapter 4, and with whom I have had many stimulating discussions that have contributed greatly to the shape that my work has taken. Robert Bogner, Tobias Eggendorfer, and Matthew Sorell all deserve mention for useful discussions regarding various aspects of my research.

Technical Staff

Returning to home, there are many others at the University of Adelaide who have contributed to the research underlying this thesis in one way or another. The technical staff of the School of Electrical and Electronic Engineering are particularly notable; Danny Di Giacomo, Alban O'Brien, Ian Linke, Brandon Pullen, Pavel Simcik, and Aubrey Slater have aided me immeasurably in realising my constructions.

Other Staff

Much vital support was provided by the administrative staff of the School of Electrical and Electronic Engineering. None more so than Rose-Marie Descalzi, whose responsibility for travel and seminars resulted in a near-constant barrage of talk proposals and overly-complicated itineraries. But thanks are equally due to the others who keep the machine of the department running, and in particular to David Bowler, Franca Guest, Stephen Guest, Deb Koch, Mark Innes, Jodie Schluter, and Pavel Simcik.

Personal Thanks

In addition to these professional credits, thanks are also due to the family and friends who have supported me throughout these last few years. Without them, I would not have reached the beginning of my Ph.D, let alone the end.

Conventions

This thesis is typeset using the LuaTeX and LaTeX2e software. Harvard style is used for referencing and citation. Australian English spelling is adopted, as defined by the Macquarie English Dictionary (Delbridge 2001).

Acronyms are given in small caps—e.g. ADC, PDF—where this does not create ambiguity. Exceptions generally involve plurals, e.g. ADC/ADCs.

The main text is typeset in *Minion Pro*, with the mathematics set in *TeX Gyre Pagella*. Figure captions and other sans-serif text are set in *Charlotte Sans*.

Publications

Papers marked ▶ are directly relevant to this thesis.

JOURNAL PAPERS

1. ▶ L. J. Gunn, A. Allison, and D. Abbott (2013). Identification of static distortion by noise measurement. *Electronics Letters* 49(21), pp. 1321–1323. DOI: 10.1049/el.2013.2547
2. L. J. Gunn, P. G. Catlow, W. A. Al-Ashwal, J. G. Hartnett, A. Allison, and D. Abbott. Simplified three-cornered-hat technique for frequency stability measurements. *IEEE Transactions on Instrumentation and Measurement* 63(4), pp. 889–895. DOI: 10.1109/tim.2013.2285796
3. J. M. Chappell, S. P. Drake, C. L. Seidel, L. J. Gunn, A. Iqbal, A. Allison, and D. Abbott (2014). Geometric algebra for electrical and electronic engineers. *Proceedings of the IEEE* 102(9), pp. 1340–1363. DOI: 10.1109/jproc.2014.2339299
4. ▶ L. J. Gunn, A. Allison, and D. Abbott (2014a). A directional wave measurement attack against the Kish key distribution system. *Scientific Reports* 4. Art. 6461. DOI: 10.1038/srep06461
5. A. Dinovitser, L. J. Gunn, and D. Abbott (2015). Towards quantitative atmospheric water vapor profiling with differential absorption lidar. *Optics Express* 23(17), pp. 22907–22921. DOI: 10.1364/OE.23.022907
6. L. Xu, T. Vladusich, F. Duan, L. J. Gunn, D. Abbott, and M. D. McDonnell (2015). Decoding suprathreshold stochastic resonance with optimal weights. *Physics Letters A* 379(38), pp. 2277–2283. DOI: 10.1016/j.physleta.2015.05.032
7. ▶ L. J. Gunn, A. Allison, and D. Abbott (2015b). A new transient attack on the Kish key distribution system. *IEEE Access* 3, pp. 1640–1648. DOI: 10.1109/access.2015.2480422
8. ▶ L. J. Gunn, F. Chapeau-Blondeau, M. D. McDonnell, B. R. Davis, A. Allison, and D. Abbott (2016d). Too good to be true: when overwhelming evidence fails to convince. *Proceedings of the Royal Society A* 472(2187). DOI: 10.1098/rspa.2015.0748

CONFERENCE PAPERS

1. ► L. J. Gunn, J. M. Chappell, A. Allison, and D. Abbott (2014c). Physical-layer encryption on the public internet: a stochastic approach to the Kish-Sethuraman cipher. *International Journal of Modern Physics: Conference Series* 33. Presented at HotPI-2013. DOI: 10.1142/S2010194514603615
2. ► L. J. Gunn, A. Allison, and D. Abbott (2014b). Allison mixtures: where random digits obey thermodynamic principles. *International Journal of Modern Physics* 33. Presented at Hot Topics in Physical Informatics 2013. DOI: 10.1142/S2010194514603603
3. J. M. Chappell, L. J. Gunn, and D. Abbott (2013). The double-padlock problem: is secure classical information transmission possible without key exchange? Presented at Hot Topics in Physical Informatics. DOI: 10.1142/S201019451460355X
4. ► L. J. Gunn, A. Allison, and D. Abbott (2015a). "Real-time compensation of static distortion by measurement of differential noise gain". *Proc. IEEE Workshop on Signal Processing Systems*. Belfast, United Kingdom. DOI: 10.1109/SiPS.2014.6986079
5. ► L. J. Gunn, F. Chapeau-Blondeau, A. Allison, and D. Abbott (2016c). Towards an information-theoretic model of the allison mixture stochastic process. *Journal of Statistical Mechanics: Theory and Experiment* 2016(5). DOI: 10.1088/1742-5468/2016/05/054041
6. ► L. J. Gunn, A. Allison, and D. Abbott (2017). "Safety in numbers: anonymization makes keyservers trustworthy". *10th Workshop on Hot Topics in Privacy Enhancing Technologies*. Minneapolis, USA[1]

[1] See Gunn et al. (2016a) for the full paper.

List of Figures

List of Tables

Chapter 1

Introduction

The goal of many designers is to develop measurement systems that are devoid of uncertainty, that will consistently produce identical measurements when faced with an identical quantity of interest. Though in reality this may be but a dream, it can be approximated by various means, such as by using filters, which remove much of the variation. In this thesis, we investigate the extent to which this often-discarded probabilistic data can be useful in practice. In this chapter, we examine this idea in general, the history of stochastic measurement systems, and provide a brief outline of the remainder of the thesis.

1.1 INTRODUCTION

The great paradigm shift of the 20th century was the transition from a deterministic to a probabilistic view of the world. Some systems are fundamentally stochastic—quantum physics is the canonical example of this—whereas in other cases, probabilistic models are used to account for unknown information in deterministic systems.

Whatever the reasons for a probabilistic system model, often no attempt is made to take take advantage of its stochastic nature. In many cases, the system designer will attempt to minimise probabilistic disturbances without considering the information that is discarded in doing so. In this thesis we consider a number of probabilistic systems that use—or attempt to use—noise to provide functionality beyond the deterministic systems that they approximate.

The systems that we consider vary substantially in character—in some the randomness is neither measured nor even intentionally present; in others, the randomness is vital to the operation of the system. This is particularly so in adversarial systems, many of which depend upon the unpredictable nature of a stochastic element of the system in order to prevent an adversary from violating the security requirements of its legitimate users.

This does not mean that stochastic systems are a panacea; it is easy to mistake stochasticity for complete unpredictability, however in practical systems independence assumptions are easily broken when an incomplete model is used. This is particularly important in the adversarial setting, where an adversary can exploit unmodelled behaviour whose effects are not sufficiently large as to be discovered by testing.

1.2 THESIS OUTLINE

We begin in this chapter by motivating the use of stochastic systems and reviewing the system identification and information-theoretic security literature in order to demonstrate the importance of stochastic systems for information transfer, making clear the as-yet unexploited gap in the literature that we fill in this thesis.

In Chapter 3 we demonstrate an approach by which a noisy sensor can be used to measure and compensate its nonlinearity, providing an output-only system identification capability that is not available in the existing literature.

In Chapter 4 we discuss the use of noise-measurement systems as a method of physical-layer key establishment. We begin by examining the use of internet round-trip-times as a source of entropy, before introducing the Kish key distribution system. We describe two

attacks against this system, and rebut the arguments used by some in the literature to claim that it is information-theoretically secure.

In Chapter 5 we examine several more conventional topics in computer security from a stochastic measurement point of view. We begin by showing how an anonymity system can be used to provide anti-equivocation functionality, providing a formal security reduction to that of the underlying anonymity system. We then consider distributed Public-Key Infrastructure (PKI). Current approaches to PKI suffer from a 'weakest-link' property: the security of high-quality certificate issuance processes is irrelevant because that an attacker can compromise the weakest authority in the system. We show how this can be overcome using a jury-like system where certificate authorities are selected with the aid of a randomness beacon.

1.3 MOTIVATION

Failure modes can be broadly divided into two categories: *hard failures*, and *soft failures*. Hard failures are those where the system is no longer able to function, whereas a soft failure results in a non-catastrophic degradation of performance.

Fortunately, the continued operation of a system that experiences a soft failure provides an opportunity for mitigation. Systems in such a failure state often leave tell-tale traces in their output that can be used for system identification; this allows us to compensate for the changes in system behaviour, or to fail gracefully if the change in behaviour is unacceptable.

This approach is particularly compelling when it is impractical or impossible to measure internal features of the system; indeed, many failure modes do not have predictable effects and can only be characterised by their effect on the system response. This can be the case for many reasons—device variations and ageing in electronics or mechanical systems are not easily measured in production, and in an adversarial setting attackers are necessarily unpredictable in their behaviour.

Such identification is a difficult task at the best of times, and in this thesis we search for techniques that will remain applicable in the most general case. It is the quest for generality that leads us to the noise properties of the system as a source of information on its behaviour—noise is almost unavoidable, particularly in analog systems, but more importantly it is *unpredictable*. This is significant in the adversarial setting, as it prevents an attacker from easily tailoring their inputs to the state of the system in question.

1.4 BACKGROUND

In this section we briefly discuss previous work in this area in order to provide a context for this thesis; the existing literature is discussed in greater detail in the respective chapters.

1.4.1 SYSTEM IDENTIFICATION

One of the first things learnt by an aspiring signal-processing engineer is the modelling of linear time-invariant systems; it is well-known that the output $y(t)$ of such a system, given an input $x(t)$, is given by

$$y(t) = h(t) \otimes x(t),$$

where $h(t)$ is the impulse response of the system and $\otimes$ the convolution operator. Less straightforward a task is to determine the function $h(t)$ given some knowledge of the input $x(t)$ and corresponding output $y(t)$. This is the goal of *system identification:* to generate an accurate model of a system based on its behaviour.

1.4.1.1 Static nonlinearity

In this thesis we do not focus on dynamical systems, but static nonlinear systems; these in important in devices such as analog-to-digital converters whose structure subjects them to unpredictable nonlinearity across their entire range; when this changes with time, current techniques require regular offline calibration.

The simplest way to identify such a nonlinearity is to apply a known signal at the input, and then tabulate inputs and outputs ("IEEE Standard for Digitizing Waveform Recorders" 2008, §4.7.1); this requires high-precision test equipment, and cannot be carried out without disconnecting the device under test from its system.

More advanced versions of this technique reduce the accuracy demanded of the test setup, but are ultimately offline techniques that cannot be used on a system in operation. Histogram methods (Doernberg et al. 1984) use an input signal whose exact value at any given time is not known, but whose statistics are. A triangular wave has a uniform distribution of values, and the number of times that a given output occurs is in proportion to the size of the input range to which it corresponds. This requires an extremely precise test signal, as any variation from a perfect triangle will result give the appearance of nonlinearity where it does not exist.

A sinusoidal signal can be used to the same effect, and can be generated with much greater precision (Blair 1994). Martins and Cruz Serra (1999) suggested the use of Gaussian noise as a test signal, presaging our own work in Chapter 3.

An alternative approach, suggested by Alegria et al. (2001), makes lighter demands on the test signal generator; their approach is still a histogram method, however their triangular signal is small, and superimposed upon a slowly shifting Direct Current (DC) offset. With the triangle covering only a few output codes, its specifications can be far less demanding. This idea is closely related to our work in Chapter 3.

From here we will turn our attention to an adversarial system of the cryptographic variety, namely noise-based physical key establishment systems; these are also based on electronic noise, but used in a linear way that is intended to reduce the information available to an eavesdropper. However, before doing so, we will briefly discuss more conventional cryptographic methods in order to place this work in context.

1.4.2 CRYPTOGRAPHY

The study of cryptography has taken place, with varying degrees of rigour, for thousands of years (Kahn 1966). However, the field as it is known today began, to a great degree, with the work of Shannon (1945), who proved that a stream cipher (Schneier 1996, p. 189) provides perfect secrecy—that is, the ciphertext provides no information on the plaintext—so long as its keystream remains confidential, is uniformly distributed, and its symbols independent.

This provides secrecy, but it was not until some years later that information-theoretically secure authentication came into being. The Carter-Wegman authenticator (Wegman and Carter 1981) uses a randomly-selected hash function to produce an authentication tag; the class of hash functions is chosen in such a way that knowing the tag corresponding to one input does not provide substantial help in forging the tag of a second input. Variations of this authenticator are widely used today, notably by the Galois/Counter Mode (GCM) block cipher mode (Dworkin 2007) and the *Poly1305* authenticator (Bernstein 2005).

Carter-Wegman-type authenticators are a good example of the type of system that we examine in the latter part of this thesis; the random nature of the system means that an attacker cannot predict the inputs necessary to force a particular output. A problem, however, is that the keys for these systems may only be used once, and when used for encryption must have the same length as the message in question. The result is that these systems have only been used in the highest-security applications; for everyday use we must look elsewhere.

1.4.2.1 *Abandoning purity for practicality*

The greatest cryptographic advances of the late 20th century were not built on an information theoretic footing, but a computational one. In the late 1970s, Diffie and Hellman (1976) and Rivest et al. (1978) introduced a pair of cryptosystems that allow secure key establishment without pre-shared secrets. They function as follows:

1. Bob generates a key pair $(\mathsf{pk}, \mathsf{sk})$ composed of a *public key* and a *secret key.*
2. He publishes the public key.
3. Alice takes Bob's public key, and uses it to encrypt a message, sending the ciphertext to Bob.
4. Bob takes the ciphertext, and decrypts it using the secret key.

This ability to use different keys for encryption and decryption made cryptography accessible to the public, as individuals no longer need to individually establish a key with every entity with whom they wished to communicate; it is necessary merely to assure one's self of the public key of the user with which one desires to communicate.

This is not the end of the problem, however—though public-key cryptography is now well-entrenched, doubts about the real difficulty of solving the underlying problems have existed since the early days of public-key cryptography (Boak 1981), and indeed new factoring algorithms (Pomerance 1996) have forced system designers to dramatically increase key sizes, though as yet no polynomial-time classical factoring algorithm is known.

Algorithmic advances are not the only threat to these systems; all mainstream public-key cryptography algorithms—in particular, Rivest-Shamir-Adleman cryptosystem (RSA), Diffie-Hellman, and Elliptic-Curve Cryptography (ECC) —are based on a mathematical structure that leaves them vulnerable (Hallgren and Vollmer 2009) to the quantum algorithm by Shor (1994). Other cryptosystems have been proposed that do not appear vulnerable to Shor's algorithm (Bernstein 2009), but with the exception of hash-based signatures (Lamport 1979) they have not yet been subjected to the same level of scrutiny as more conventional cryptosystems.

1.4.2.2 *Physical approaches to key establishment*

The risk that these hitherto-intractable problems will suddenly become efficiently soluble has led researchers to investigate systems whose security can be reduced to *physical* laws, rather than computational limits or pure probability.

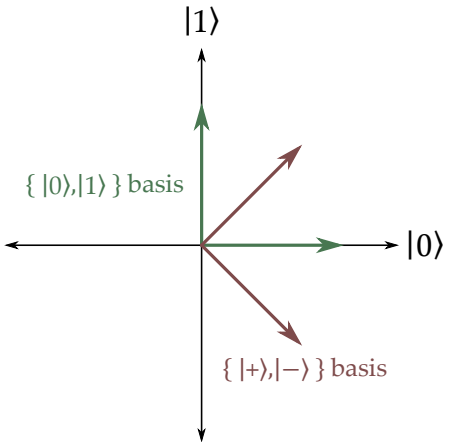


Figure 1.1: Measurement bases for BB84. One of the four basis vectors is selected at random, and sent to the recipient, Bob, in the form of a single polarised photon. If the photon is measured with the wrong measurement basis, the measurement will be wrong 50% of the time.

The most well-known of these methods is Quantum Key Distribution (QKD); introduced by Bennett and Brassard (1984); it derives its security from the laws of quantum mechanics. This first protocol is known as BB84, and its function is based on the idea that we can build a quantum system that will normally behave deterministically, but whose measurements become stochastic in the presence of an eavesdropper.

In the most common protocol, by BB84, a photon is sent from Alice to Bob in the quantum state determined by one of four polarisations, as shown in Figure 1.1.

Because each pair of basis vectors is orthogonal, the state can be measured with perfect fidelity (Nielsen and Chuang 2000), yielding a binary value, if it is known from which basis the state is chosen. If not, the result will be wrong up to 50% of the time, depending on which measurement basis was chosen. In BB84, Alice does not reveal which of the two orthogonal measurement bases was chosen until after Bob has received and measured the photon; if Bob chose the wrong measurement basis, then he discards the bit. This means that an eavesdropper who attempts to measure and copy the photon will select the wrong basis 50% of the time, resulting in a 25% error rate when the photon is decoded by Bob. Such an eavesdropper will be detected and their information rendered useless by privacy amplification (Bennett et al. 1988).

The no-cloning theorem ("A single quantum cannot be cloned" 1982) prevents an eavesdropper from copying a quantum state whilst in transit; in the simplest form of attack, this forces the eavesdropper to commit to a measurement basis before she knows what it is. A

truly random selection of the measurement basis thus results in the eavesdropper being correct with probability no better than chance.

This type of system is theoretically very secure, however practical systems have succumbed to hacking-type attacks (Lydersen et al. 2010; Dixon et al. 2017). The expensive and exotic nature of QKD has led to a desire for simpler physical key distribution methods (Kish 2006c).

The time-varying nature of wireless channels has led to key establishment systems that use channel characteristics as a source of randomness for key establishment (Mathur et al. 2008; Zhang et al. 2016b; Zhang et al. 2016a); physical separation of the eavesdropper and recipient result in differing channel characteristics. The unpredictability of these differences is used to provide the 'information gap' that can be used to generate a secret key (Wyner 1975).

Another system, the Kish Key Distribution (KKD) system—also known as the Kirchoff-Law-Johnson-Noise (KLJN) system—by Kish (2006c), has been proposed as an alternative to QKD for wired systems. The KKD family of systems uses simulated thermal noise in an attempt to admit a security reduction to the second law of thermodynamics (Kish and Granqvist 2014b). Its security remains controversial; despite the lofty claims made about it, a number of attacks have been presented over the years (Scheuer and Yariv 2006; Hao 2006; Bennett and Riedel 2013; Gunn et al. 2014a; Gunn et al. 2015b) that breach its security claims to some extent, at least for non-ideal implementations. However the only previous experimental work, by Mingesz et al. (2008), suggests that none of the attacks proposed at the time are effective in practice. We discuss these attacks in more detail in Sections 4.5 and 4.7.3.

The strongest theoretical argument against the security of the KKD system was made by Bennett and Riedel (2013); they argue on information theoretic grounds that guided electromagnetic waves cannot be used to distill a secret key, on the grounds that an eavesdropper with a directional coupler has access to the same transmitted information as the legitimate users.

This was rebutted by Kish et al. (2013), with the authors arguing that a short cable cannot support a 'true' wave, rendering the argument of Bennett and Riedel (2013) moot. It is the unresolved disagreement between these two views that leads us to develop the wave-based attacks that we present in Chapter 4; the difference in efficacy between the experimental attacks considered by Mingesz et al. (2008) and the theoretical attack described by Bennett and Riedel (2013) is too great to be ignored, and is the topic of a large part of this thesis.

1.4.2.3 Identity management

Key establishment is but one part of secure communication; equally important is authentication, since it is not enough just to have a secret key—the key must be shared with the right people.

Prior to the advent of public-key cryptography, the only way to ensure that a secret key belonged to the intended recipient was for it to be delivered by a trusted party; in a military context this was not a problem, as keys[2] can be produced and assigned centrally (Boak 1973, p. 42) to their users, who do not need to establish the provenance of keys themselves.

Unfortunately, the problem is not so easily solved in the private sector, where communication is not dictated by a chain of command. The solution came with the efficient digital signatures provided by the RSA algorithm. Rivest et al. (1978) described a central directory of public keys that uses a public-key signature to authenticate its responses; this is the predecessor to the certificates used today by X.509 (Cooper et al. RFC 5280, 2008) and Pretty Good Privacy (PGP) (Callas et al. RFC 4880, 2007). A trusted entity signs a certificate stating the identity of the owner of a public key, allowing it to be relied upon by others.

The need for a trusted central issuer of certificates is a major problem—any issuer can issue a certificate for anyone, unless their authority has been technically constrained. Of the many PKI failures seen to date, the majority have occurred as a result of error or compromise of an organisation with the power to issue certificates, rather than cryptographic failures (CAcert 2017).

There have been a number of attempts at reducing this level of centralisation; the PGP Web of Trust has users certify each other, allowing users to find a path through the certification graph to identify someone as a friend-of-a-friend-of-a-friend. Unfortunately, even with years of development this has in practice proven to be too difficult for users (Whitten and Tygar 1999; Sheng et al. 2006; Ruoti et al. 2015).

Another approach is to decentralise the certificate issuance process itself, whilst retaining the hierarchical PKI. Such an approach has been described by Syta et al. (2015a), who have developed a system that allows the efficient production of group signatures. Unfortunately, their approach requires major changes on the part of certificate authorities, and is therefore is unlikely to see significant adoption in the short term.

Trust-on-first-use is a commonly-used authentication paradigm, referred to by less generous authors as the *leap of faith* approach (Gilad and Herzberg 2013). In this mode, all

[2]Note that in the terminology of the NSA, a key is referred to as a *cryptovariable*, or simply a *variable*.

identity verification is initially left to the user; the software confines itself to ensuring that the key remains consistent over time. In practice, this means that the key is accepted the first time it is used, with the user being warned each time it changes. This is of little help against a pervasive adversary who can effect a Man-in-the-Middle (MITM) attack against all communications by a user, but to avoid detection such an attack must be maintained indefinitely. Despite this, its ease of use means that end-to-end secure systems using trust-on-first-use, such as Whatsapp (*WhatsApp encryption overview: technical white paper* 2016), have received greater acceptance than any of the more secure systems that came before.

In order to reduce the risk posed by an unconditional trust-on-first-use approach, a number of authors have proposed systems based on *multi-path probing*. These are based on the principle that a MITM attack will generally be local in scope; by connecting to a service from several network vantage points, local MITM attacks are evaded, providing greater certainty in the legitimacy of the key that is used for the first connection.

The first of these systems is known as *Perspectives* (Wendlandt et al. 2008), and uses notary servers to certify that a particular Secure Shell (SSH) server has presented the same key to a number of hosts over a long period of time. A similar system was proposed by Melara et al. (2015) for their proposed cryptographically-auditable directory service.

Attempting to avoid the need for dedicated notary servers, the *Doublecheck* system by Alicherry and Keromytis (2009) makes parallel connections to websites via the Tor anonymisation network (Dingledine et al. 2004), comparing the certificates presented in order to detect local attacks. Anonymity systems provide a convenient way to access the internet from different physical locations, and have been used by Alicherry and Keromytis (2009), Engert (2013), and Gilad and Herzberg (2013) for their multi-path probing systems.

No discussion of modern identity management techniques would be complete without mention of blockchain methods. Namecoin (Kalodner et al. 2015) is a fork of Bitcoin (Nakamoto 2008) that includes a key-value which is used as a name store; the Namecoin currency is then used to impose a price on name registration and renewal. A Namecoin name might map to a regular domain name, an Internet Protocol (IP) address, or a Tor hidden service; in all of these cases, certificate fingerprints can be added in order to ensure that one is communicating with the owner of the name record.

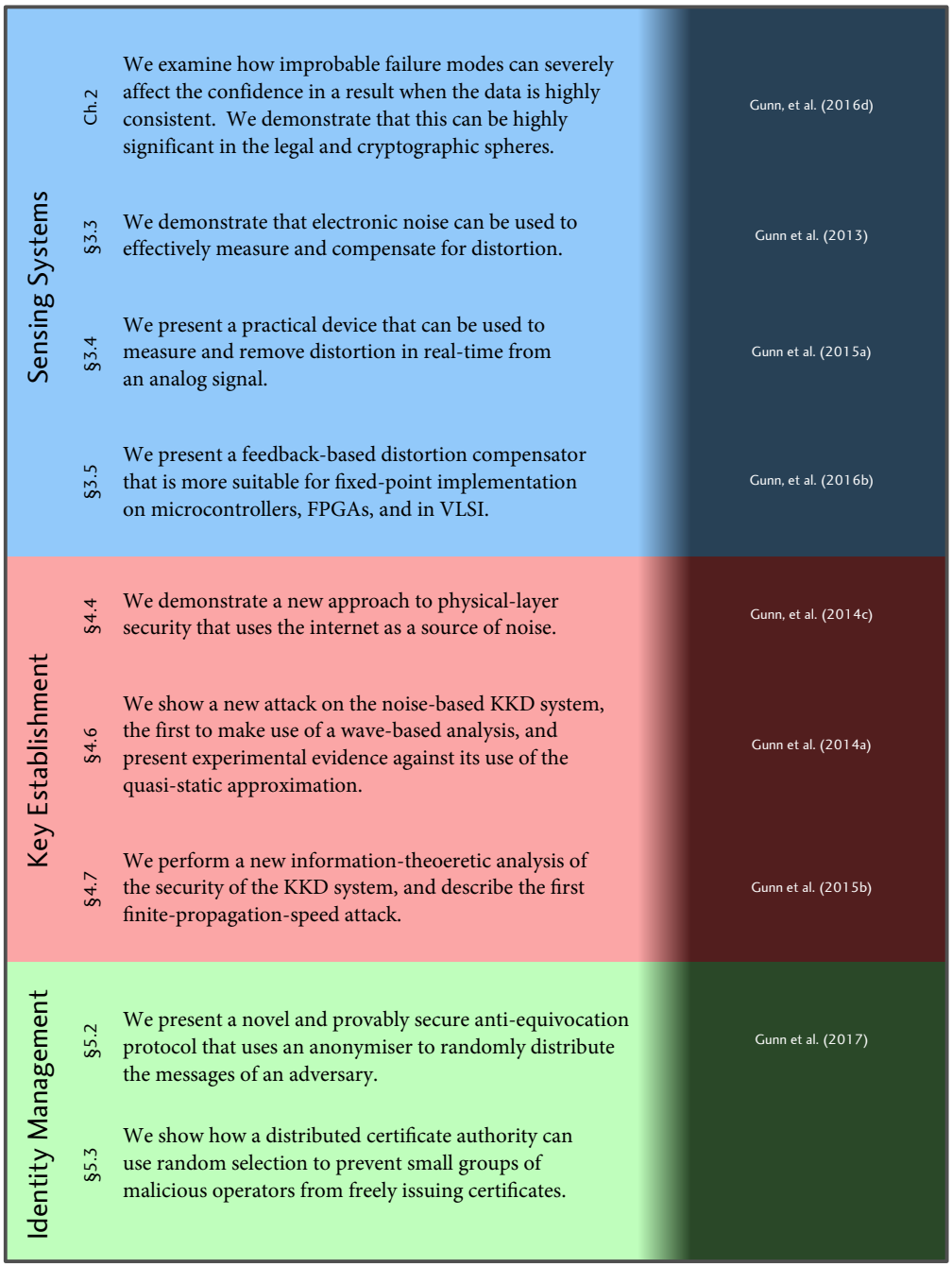
As with Bitcoin, the security of Namecoin is based on a proof-of-work; the network periodically publishes a new database state, with a valid publication—a *block*—requiring a large computational effort to produce. Clients use the longest chain of blocks known to them, considering the most recent few blocks to be only tentative. Its security derives from the fact

that an attacker who does not control the majority of the computational power of the network will not be able to out-pace the honest nodes who will quickly produce a longer chain and thus render his effort pointless. This prevents a malicious participant from rewriting history and thereby taking control of other users' names[3].

[3]In both Bitcoin and Namecoin, property is assigned to the holder of an elliptic-curve private key. Transfers and updates require a signed request from the owner, and these signatures can be followed back to the original name registration or coin issuance. Rewriting of history allows transfers to be repudiated or names to be re-registered in the key of an attacker.

1.5 ORIGINAL CONTRIBUTIONS

We summarise our original contributions here, with more detailed discussions appearing at the end of each chapter.



We have made some further contributions (Gunn et al. 2014b; Gunn et al. 2016c) in the area of stochastic modelling, however these are not strongly connected to the remainder of this thesis. These are described in Appendix C.

Chapter 2

Decisions, Failures, and Stochastic Systems

If something sounds too good to be true, it probably is. The assumption of independence is often made in good faith, however rarely is consideration given to whether this is still the case when a systemic failure has occurred. Taking this into account can, paradoxically, cause certainty in a hypothesis to decrease as the evidence for it becomes increasingly strong. We perform a Bayesian analysis of this effect with several examples, including cryptographic primality testing and an ancient rule of legal procedure that appears to take advantage of it.

We find that even with very low failure rates, high confidence is very difficult to achieve, and in particular we find that certain analyses of cryptographically-important operations are highly optimistic, overestimating their failure rate by as much as a factor of 2^{80}.

2.1 INTRODUCTION

As the case drags on, witness after witness is called; soon, thirteen have testified to having seen the defendant commit the crime. Witnesses may be notoriously unreliable, but the sheer magnitude of the testimony is overwhelming. After all, anyone can make a misidentification, but intuition tells us that with each additional witness, the chance of them all being incorrect will approach zero. However, this is not necessarily the case, a fact that has been recognised intuitively in ancient times. Under ancient Jewish law (Epstein 1961), one could not be convicted of a capital crime unanimously—it was held that the absence of even one dissenting opinion among the judges indicated that there must remain some form of undiscovered exculpatory evidence.

Such approaches are greatly at odds with standard practice in signal processing, where measurements are often taken to be independent. When this is so, each new measurement tends to lend support to the outcome with which it most concords. An important question, then, is to distinguish between the two types of decision problem; those where additional measurements truly lend support, and those for which increasingly consistent evidence either fails to add or actively reduces confidence; this is commonly seen when humans analyse data collected by a stochastic measurement system: nothing is more damning than an unexpected lack of randomness.

The key ingredient to this unexpected reversal of confidence is the presence of a hidden failure state that changes the measurement response. This change may be quite rare *a priori*—in the applications that we discuss, it ranges from 10^{-1} to 10^{-19}—but when several samples are aggregated, the *a posteriori* probability of the failure state can increase substantially, and even come to dominate the *a posteriori* estimate of the measurement response. By changing the information-fusion rule in a measurement-dependent way, unintuitive effects such as non-monotonicity can occur.

As we will show, this phenomenon has a substantial effect on the level of confidence achievable by such systems, and must be considered during design of experiments; the applications of this analysis range from cryptography to criminology, and we will provide examples of how rare failure modes can have a surprising effect on the achievable level of confidence.

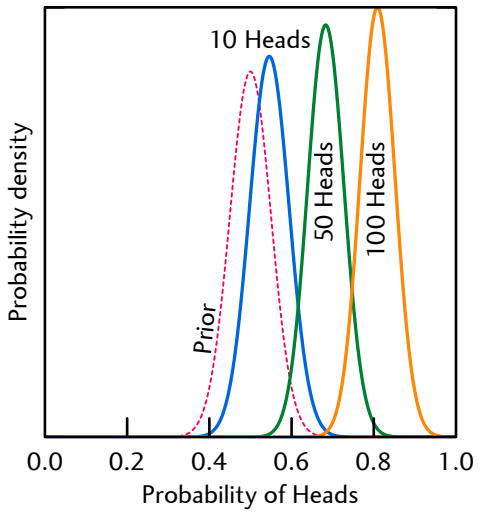


Figure 2.1: Prior and posterior distributions of the heads-probability of a biased coin. We suppose a prior distribution of $H \sim \mathcal{N}(0.5, 0.05)$. Because the tails of the Gaussian distribution are not heavy, the posterior distribution moves only very slowly away from the unbiased value of 0.5.

2.2 ANALYSIS OF A BIASED COIN

We begin with a simple and well-known problem related to what we have discussed, namely the question of whether or not a coin is biased. We follow the Bayesian approach given by Sivia and Skilling (2006).

They use Bayes' law in its proportional form,

$$P[H|\{\text{data}\}] \propto P[\{\text{data}\}|H]P[H],$$

where H is the probability that a coin-toss will yield heads. Various prior distributions $P[H]$ can be chosen, a matter that we will discuss momentarily.

As the coin tosses are independent, the data can be boiled down to a binomial random variable $X \sim \text{Bin}(p, n)$, where n is the number of coin tosses made. Substituting the binomial probability mass function into (2.2), we find

$$P[H|X] \propto H^X(1-H)^{n-X}P[H].$$

As the number of samples N increases, this becomes increasingly peaked around the value $H = X/n$, limited by $P[H]$. As the number of samples increases, the $H^X(1-H)^{n-X}$ part of the expression eventually comes to dominate the shape of the posterior distribution $P[H|X]$, and we have no choice but to believe that the coin genuinely does have a bias of X/n. In the examples to be discussed, we assume that bias is very unlikely; in the coin example, this

corresponds to a prior distribution $P[H]$ that is strongly clustered around $H = 0.5$; in this case, a very large number of samples will be necessary in order to conclusively reject the hypothesis that the coin is unbiased or nearly so, as visible in Figure 2.1. However, eventually this will occur, and the posterior distribution will change; when this occurs, the system has visibly failed—a casino using the coin will decide that the coin is biased and pull out, having realised they are not really playing the game that they had planned.

2.3 A HYPOTHETICAL ROMAN POT

Now let us proceed to our main topic, sensing, beginning with a simple scenario, the identification of the origin of a clay pot that has been dug from British soil. Its design identifies it as being from the Roman era, and all that remains is to determine whether it was made in Roman-occupied Britain or whether it was brought from Italy by travelling merchants. Suppose that we are fortunate and that a test is available to distinguish between the clay from the two regions; clay from one area—let us suppose that it is Britain—contains a trace element which can be detected by laboratory tests with an error rate $p_e = 0.3$. This is far too unreliable to make archaeological conclusions, and so we run the test several times. After k tests have been made on the pot, the number of errors will be binomially-distributed $E \sim \text{Bin}(k, p_e)$. If the two origins, Britain and Italy, are *a priori* equally likely, then the most probable origin is the one suggested by the greatest number of samples.

Now imagine that several manufacturers of pottery deliberately introduced large quantities of this element during their production process, and that therefore it will be detected with 90% probability in their pots, which make up $p_c = 1\%$ of those found; of these, half are of British origin. We call p_c the *contamination rate*. Note that these values are arbitrary; we introduce them now, rather than later, in order to provide the reader with a mental image whilst we derive a formal model of the procedure. This is the hidden failure state to which we alluded in the introduction. Then, after the pot tests positive several times, we will become increasingly certain that it was manufactured in Britain. However, as more and more test results are returned from the laboratory, all positive, it will become more and more likely that the pot was manufactured with this unusual process, eventually causing the probability of British origin, given the evidence, to fall to 50%. This is the essential paradox of the system with hidden failure states—overwhelming evidence can itself be evidence of uncertainty, and thus be less convincing than more ambiguous data.

2.3.1 FORMAL MODEL

Let us now proceed to formalise the problem above. Suppose we have two hypotheses, H_0 and H_1, and a series of measurements $\mathbf{X} = (X_1, X_2, \dots, X_n)$. We define a variable $F \in \mathbb{N}$ that determines the underlying measurement distribution, $p_{X|F,H_i}(x)$. We may then use Bayes' law to find

$$P[H_i|\mathbf{X}] = \frac{P[\mathbf{X}|H_i]P[H_i]}{P[\mathbf{X}]}, \tag{2.1}$$

which can be expanded by condition with respect to F, yielding

$$= \frac{\sum_f P[\mathbf{X}|H_i,f]P[H_i,F=f]}{\sum_{f,H_k} P[\mathbf{X}|H_k,f]P[H_k,F=f]}. \tag{2.2}$$

In our examples there are a number of simplifying conditions—there are only two hypotheses and two measurement distributions, reducing (2.2) to

$$P[H_i|\mathbf{X}] = \frac{1}{1 + \dfrac{\sum_{f=0}^{1} P[\mathbf{X}|H_{1-i},F=f]\ P[H_{1-i},F=f]}{\sum_{f=0}^{1} P[\mathbf{X}|H_i,\quad F=f]\ P[H_i,\quad F=f]}}. \tag{2.3}$$

Computation of these *a posteriori* probabilities thus requires knowledge of two distributions: the measurement distributions $P[\mathbf{X}|H_k,F]$, and the state probabilities $P[H_i,F]$. Having tabulated these, we may substitute them into (2.3), yielding the *a posteriori* probability for each hypothesis. In this thesis, the measurement distributions $P[\mathbf{X}|H_i,F=f]$ are all binomial, however this is not the case in general.

2.3.2 ANALYSIS OF THE POT ORIGIN DISTRIBUTION

In the case of the pot, the hypotheses and measurement distributions—the origin and contamination, respectively—are shown in Table 2.1.

Each measurement is Bernoulli-distributed, and the number of positive results is therefore described by a Binomial distribution, with the probability mass function

$$P[X=x] = \binom{n}{x} p^x (1-p)^{N-x}$$

$P[F, H_i]$

Contaminated	Origin: Italy H_0	Origin: Britain H_1
Y, $F=0$	0.005, $\frac{1}{2} p_c$	0.005, $\frac{1}{2} p_c$
N, $F=1$	0.495, $\frac{1}{2}(1-p_c)$	0.495, $\frac{1}{2}(1-p_c)$

Each square represents an outcome

$P[\text{Positive result} \mid F, H_i]$

Contaminated	Origin: Italy H_0	Origin: Britain H_1
Y, $F=0$	0.9	0.9
N, $F=1$	0.3, p_e	0.7, $1-p_e$

Each square represents a distribution

Table 2.1: The model parameters for the case of the pot for use in (2.3) with a contamination rate $p_c = 10^{-2}$. The *a priori* distribution of the origin is identically 50% for both Britain and Italy, whether or not the pot's manufacturing process has contaminated the results. As a result, the two columns of $P[F, H_i]$ are identical. The columns of the measurement distribution, shown right, differ from one another, thereby giving the test discriminatory power. When the pot has been contaminated, the probability of a positive result is identical for both samples, rendering the test ineffective.

after N trials, the probability p being taken from the measurement distribution section of Table 2.1.

Substituting these probability masses into (2.2), we see in Figure 2.2 that as more and more tests return positive results, we become increasingly certain of its British heritage, but an unreasonably large number of positive results will indicate contamination and so yield a reduced level of certainty.

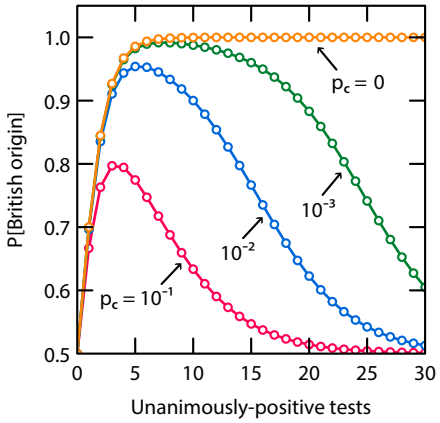


Figure 2.2: Probability that the pot is of British origin given n numbers of tests, all coming back positive, for a variety of contamination rates p_c and a 30% error rate. In the case of the pot above, with $p_c = 10^{-2}$, we see a peak at $n = 5$, after which the level of certainty falls back to 0.5 as it becomes more likely that the pot originates at a contaminating factory. When $p_c = 0$, this is the standard Bayesian analysis where failure states are not considered. We see therefore that even small contamination rates can have a large effect on the global behaviour of the testing methodology.

2.4 THE RELIABILITY OF IDENTITY PARADES

We initially described the scenario of a court case, in which witness after witness testifies to having seen the defendant commit the crime of which he is accused. But in-court identifications are considered unreliable, and in reality if identity is in dispute then the identification is made early in the investigation under controlled conditions (Devlin et al. 1976). At some point, whether before or after being charged, the suspect has most likely been shown to each witness amongst a number of others, known as fillers, who are not under suspicion. Each witness is asked to identify the true perpetrator, if present, amongst the group.

This process, known as an *identity parade* or *line-up*, is an experiment intended to determine whether the suspect is in fact the same person as the perpetrator. It may be performed only once, or repeated many times with many witnesses. As human memory is inherently uncertain, the process will include random error; if the experiment is not properly carried out then there may also be systematic error, and this is the problem that concerns us in this thesis.

Having seen how a unanimity of evidence can create uncertainty in the case of the unidentified pot, we now apply the same analysis to the case of an identity parade. If the perpetrator is not present—that is to say, if the suspect is innocent—then in an unbiased

parade the witness should be unable to choose the suspect with a probability greater than chance. Ideally, they would decline to make a selection, however this does not always occur in practice (Foster et al. 1994; Devlin et al. 1976), and forms part of the random error of the procedure. If the parade is biased—whether intentionally or unintentionally—for example because (i) the suspect is somehow conspicuous (Wogalter and Marwitz 1992), (ii) the staff running the parade direct the witness towards him, (iii) by chance he happens to resemble the perpetrator more closely than the fillers, or (iv) because the witness holds a bias, for example because they have previously seen the suspect (Devlin et al. 1976), then an innocent suspect may be selected with a probability greater than chance. This is the hidden failure state that underlies this example; we assume in our analysis that this is completely binary—either the parade is completely unbiased or it is highly biased against the suspect.

In recent decades, a number of experiments (Malpass and Devine 1981; Foster et al. 1994) have been carried out in order to establish the reliability of this process. Test subjects are shown the commission of a simulated crime, whether in person or on video, and asked to locate the perpetrator amongst a number of people. In some cases the perpetrator will be present, and in others not. The former allows estimation of the false-negative rate of the process—the rate that the witness fails to identify the perpetrator when present—and the latter the false-positive rate—the rate at which an innocent suspect will be mistakenly identified. Let us denote p_{fn} the false-negative rate; this is equal to the proportion of subjects who failed to correctly identify the perpetrator when he was present, and was found by Foster et al. (1994) to be 48%.

Estimating the false positive rate is complicated by the fact that only one suspect is present in the lineup—when the suspect is innocent, an eyewitness who incorrectly identifies a filler as being the perpetrator has correctly rejected the innocent suspect as being the perpetrator, despite their error. For the purposes of our analysis, we assume that the witness selects at random in this case, and therefore divide the 80% perpetrator-absent selection rate of Foster et al. (1994) by the number of participants $L = 6$, yielding a false-positive rate of $p_{\text{fp}} = 0.133$.

Let us now suppose that there is a small probability p_c that the line-up is conducted incorrectly—for example, volunteers have been chosen who fail to adequately match the description of the perpetrator—leading to identification of the suspect 90% of the time, irrespective of his guilt. The probability of the suspect being identified for each of the cases is shown in Table 2.2.

$P[F, H_i]$

Biased parade	Suspect is… Innocent H_0	Suspect is… Guilty H_1
Y, F = 0	0.005, ½ p_c	0.005, ½ p_c
N, F = 1	0.495, ½ $(1-p_c)$	0.495, ½ $(1-p_c)$

Each square represents an outcome

$P[\text{Identification} \mid F, H_i]$

Biased parade	Suspect is… Innocent H_0	Suspect is… Guilty H_1
Y, F = 0	0.9	0.9
N, F = 1	0.13, p_{fp}	0.52, $1-p_{fn}$

Each square represents a distribution

Table 2.2: The model parameters for the hypothetical identity parade. In a similar fashion to the first example, we assume *a priori* a 50% probability of guilt. In this case, the measurement distributions are substantially assymmetric with respect to innocence and guilt, unlike Table 2.1.

If we assume a 50% prior probability of guilt, and independent witnesses, the problem is now identical to that of identifying the pot. The probability of guilt, given the unanimous parade results, is shown in Figure 2.3 as a function of the number of unanimous witnesses.

We see that after a certain number of unanimously positive identifications the probability of guilt diminishes. Even with only one in ten-thousand line-ups exhibiting this bias towards the suspect, the peak probability of guilt is reached with only five unanimous witnesses, completely counter to common sense—in fact, with this rate of failure, ten identifications in agreement provide less evidence of guilt than three. We see also that even with a 50% prior probability of guilt, a 1% failure rate renders it impossible to achieve 95% certainty if the witnesses are unanimous.

This tendency to be biased towards a particular member of the lineup when an error occurs has been noted by Devlin et al. (1976, paragraph 4.31) prior to the more rigorous research stimulated by the advent of DNA testing, leading us to suspect that our sub-1% contamination rates are probably overly optimistic.

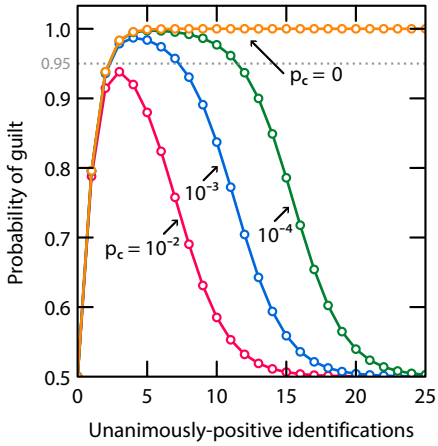


Figure 2.3: Probability of guilt given varying numbers of unanimous line-up identifications, assuming a 50% prior probability of guilt and identification accuracies given by Foster et al. (1994). Of note is that for the case that we have plotted here where the witnesses are unanimous, with a failure rate $p_c = 0.01$ it is impossible to reach 95% certainty in the guilt of the suspect, no matter how many witnesses have been found.

2.5 ANCIENT JUDICIAL PROCEDURE

The acknowledgement of this phenomenon is not entirely new; indeed, the adage "too good to be true" dates to the sixteenth century (*Oxford English Dictionary* 2015, *good*, P5.b). However, its influence on judicial procedure was visible in Jewish law even in the classical era; until the Romans ultimately removed the right of the Sanhedrin to confer death sentences, a defendant unanimously condemned by the judges would be acquitted (Epstein 1961, Sanhedrin 18b), the Talmud stating "If the Sanhedrin unanimously find guilty, he is acquitted. Why? — Because we have learned by tradition that sentence must be postponed till the morrow in hope of finding new points in favour of the defence".

The value of this rule becomes apparent when we consider that the Sanhedrin was composed, for ordinary capital offenses, of 23 members (Epstein 1961, Sanhedrin 2a). In our line-up model, this many unanimous witnesses would indicate a probability of guilt scarcely better than chance, suggesting that the inclusion of this rule should have a substantial effect.

We show the model parameters for the Sanhedrin decision in Table 2.3, which we use to compute the probability of guilt in Figure 2.4 for various numbers of judges condemning the defendant. We see that the probability of guilt falls as judges approach unanimity, however excluding unanimous decisions substantially reduces the probability of false conviction.

$P[F, H_i]$

Biased proceedings	Suspect is… Innocent H_0	Suspect is… Guilty H_1
Y, $F=0$	0.005, $\frac{1}{2} p_c$	0.005, $\frac{1}{2} p_c$
N, $F=1$	0.495, $\frac{1}{2}(1-p_c)$	0.495, $\frac{1}{2}(1-p_c)$

Each square represents an outcome

$P[\text{Identification} \mid F, H_i]$

Biased proceedings	Suspect is… Innocent H_0	Suspect is… Guilty H_1
Y, $F=0$	0.95	0.95
N, $F=1$	0.14, p_{fp}	0.75, $1-p_{fn}$

Each square represents a distribution

Table 2.3: The model parameters for the Sanhedrin trial. Again, we assume an *a priori* 50% probability of guilt. However, the measurement distributions are the results of Spencer (2007, model (2)) for juries; in contrast to the case of the identity parade, the false negative rate is far lower. Despite the trial being conducted by judges, we choose to use the jury results, as the judges tendancy towards conviction is not reflected in the highly risk-averse rabbinic legal tradition.

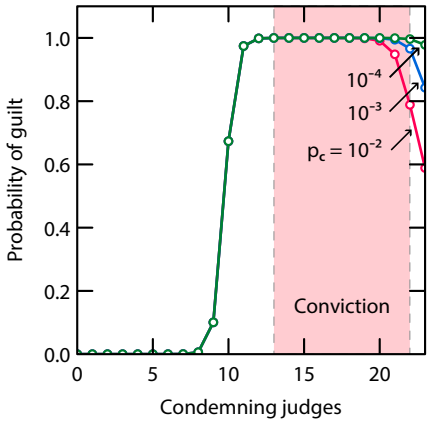


Figure 2.4: Probability of guilt as a function of judges in agreement out of 23—the number used by the Sanhedrin for most capital crimes—for various contamination rates p_c. We assume as before that half of defendants are guilty, and use the estimated false-positive and false-negative rates of juries from Spencer (2007, model (2)), 0.14 and 0.25 respectively. We arbitrarily assume that a 'contaminated' trial will result in the a positive vote 95% of the time. The panel of judges numbers 23, with conviction requiring a majority of two and at least one dissenting opinion (Epstein 1961, Sanhedrin), thus requiring 13 to 22 votes inclusive in order to secure a conviction, as shown in the graph.

It is worth stressing that the exact shapes of the curves in Figure 2.4 are unlikely to be entirely correct; communication between the judges will prevent their verdicts from being entirely independent, and false-positive and false-negative rates will be very much dependent upon the evidentiary standard required to bring charges, the strength of the contamination when it does occur, and the accepted burden of proof of the day. However, it is nonetheless of qualitative interest that with reasonable parameters, this ancient law can be shown to have a sound statistical basis.

2.6 THE RELIABILITY OF CRYPTOGRAPHIC SYSTEMS

We now consider a different example, drawn from cryptography. An important operation in many protocols is the generation and verification of prime numbers; the security of some protocols depends upon the primality of a number that may be chosen by an adversary; in this case, one may test whether it is a prime, whether by brute-force or by using another test such as the Rabin-Miller (Ferguson et al. 2010, p. 176) test. As the latter is probabilistic, we repeat it until we have achieved the desired level of security—in Ferguson et al. (2010), a probability 2^{-128} of accepting a composite as prime is considered acceptable. However, a naïve implementation cannot achieve this level of security, as we will demonstrate. Should the test fail to detect a composite number, our victim may unknowingly send confidential information that is not protected nearly as well as the Diffie-Hellman—for example—security parameters lead them to believe.

The reason is that the results of this test may be misleading is that, despite it being proven that each iteration of the Rabin-Miller test will reject a composite number with probability at least 0.75, a real computer may fail at any time. The chance of this occurring is small, however it turns out that the probability of a stray cosmic ray flipping a bit in the machine code, causing the test to accept composite numbers, is substantially greater than 2^{-128}.

2.6.1 CODE CHANGES CAUSED BY MEMORY ERRORS

Data provided by Google (Schroeder et al. 2009) suggests that a given memory module has approximately an 8% probability of suffering an error in any given year, independent of capacity. Assuming a 4 GB module, this results in approximately a $\lambda = 10^{-19}$ probability that any given bit will be flipped in any given second. We will make the assumption that, in the machine code for the primality-testing routine, there exists at least one bit that, if flipped, will cause all composite numbers—or some class of composite numbers known to the adversary—to be accepted as prime. As an example of how this could happen, consider

```
int trialdivision(long to_test)
{
   long i;
   long threshold;

   if(to_test % 2 == 0)
   {
      return 1;
   }

   threshold = (long)sqrt(to_test);

   for(i = 3; i <= threshold; i += 2)
   {
      if(to_test % i == 0)
      {
         return 1;
      }
   }

   return 0;
}
```

Figure 2.5: A function that tests for primality by attempting to factorise its input by brute force.

the function shown in Figure 2.5 that implements a brute-force factoring test. Assuming that the input is odd, the function will reach one of two return statements, returning zero or one. The C compiler GCC compiles these two return statements to

```
45 0053 B8010000      movl    $1, %eax
45      00
46 0058 EB14          jmp .L3
```

and

```
56 0069 B8000000      movl    $0, %eax
56      00
57                .L3:
```

respectively. That is to say, it stores the return value as an immediate into the `EAX` register and then jumps to the cleanup section of the function, labelled `.L3`. The store instructions on lines 45 and 56 have machine-code values `B801000000` and `B8000000000` for return values of one and zero respectively. These differ by only one bit, and therefore can be transformed into one another by a single bit-error. If the first instruction is turned into the second, this will cause the function to return zero for any odd input, thus always indicating that the input is prime.

P[F, H_i]

Always positive result	Number is… Prime H_0	Composite H_1
Y F=0	0.5×10^{-13} ½ p_c	0.5×10^{-13} ½ p_c
N F=1	≈0.001 ½ $(1-p_c)$	≈0.999 ½ $(1-p_c)$

Each square represents an outcome

P[Acceptance | F, H_i]

Always positive result	Number is… Prime H_0	Composite H_1
Y F=0	1.0	1.0
N F=1	1.0	0.25

Each square represents a distribution

Table 2.4: Model parameters for the Rabin-Miller test on random 2000-bit numbers. However, we have no choice but to assume the lower bound on the composite-number rejection rate, and so this model is inappropriate. Furthermore, in an adversarial setting the attacker may intentionally choose a difficult-to-detect composite number, rendering the prior distribution optimistic.

2.6.2 THE EFFECT OF MEMORY ERRORS ON CONFIDENCE

At cryptographically-interesting sizes—on the order of 2^{2000}—roughly one in a thousand numbers is prime (Ferguson et al. 2010, p. 173). We might calculate the model parameters as before—for interest's sake, we have done so in Table 2.4—and calculate the confidence in a number's primality after a given number of tests. However, this is not particularly useful, for two reasons: first, the rejection probability of 75% is a lower bound, and for randomly chosen numbers is a substantial underestimate; second, we do not always choose numbers at random, but rather may need to test those provided by an adversary. In this case, we must assume that they have tried to deceive us by providing a composite number, and would instead like to know the probability that they will be successful. The Bayesian estimator in this case would provide only a tautology of the type: 'given the data and the fact that the number is composite, the number is composite'.

Let us suppose that the machine containing the code is rebooted every month, and the Rabin-Miller code remains in memory for the duration of this period; then, neglecting other potential errors that could affect the test, at the time of the reboot the probability that the bit has flipped is now $p_f = 2.6 \times 10^{-13} = 60 \times 60 \times 24 \times 30$; this event we denote A_F. Let k be the number of iterations performed; the probability of accepting a composite number is at most 4^{-k}, and we assume that the adversary has chosen a composite number such that this is the true probability of acceptance. We denote the event that the prime is accepted by the correctly-operating algorithm A_R.

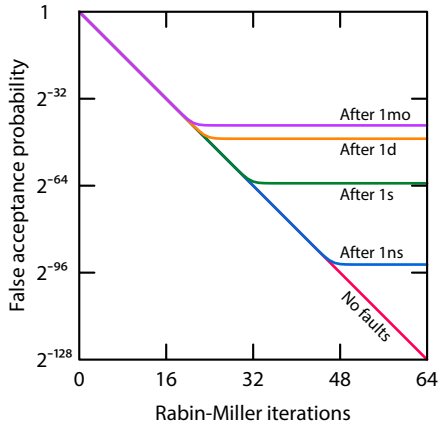


Figure 2.6: The acceptance rate as a function of time in memory and the number of Rabin-Miller iterations under the single-error fault model described in this thesis. An acceptance rate of 2^{-128} is normally chosen, however without error correction this cannot be achieved. The false-acceptance rate after k iterations is given by $p_{\text{fa}}[k] = 4^{-k}(1-p_f)+p_f$, where p_f is the probability that a fault has occurred that causes a false acceptance 100% of the time. We estimate p_f to be equal to $10^{-19}T$, where T is the length of time in seconds that the code has been in memory.

When hardware errors are taken into account, the probability of accepting a composite number is no longer 4^{-k}, but

$$p_{\text{fa}} = P[A_F \cup A_R] \tag{2.4}$$
$$= P[A_F] + P[A_R] - P[A_F, A_R]. \tag{2.5}$$

Since A_F and A_R are independent,

$$= P[A_F] + P[A_R] - P[A_F]P[A_R] \tag{2.6}$$
$$= 4^{-k}(1-p_f) + p_f \tag{2.7}$$
$$\geq p_f. \tag{2.8}$$

No matter how many iterations k of the algorithm are performed, this is substantially greater than the 2^{-128} security level that is predicted by probabilistic analysis of the algorithm alone, thus demonstrating that algorithmic analyses that do not take into account the reliability of the underlying hardware can be highly optimistic. The false acceptance rate as a function of the number of test iterations and time in memory is shown in Figure 2.6.

A real cryptographic system will include many such checks in order to make sure that an attacker has not chosen weak values for various parameters, and a failure of any of these may result in the system being broken, so our calculations are somewhat optimistic.

Error-correcting-code equipped (ECC) memory will substantially reduce the risk of this type of fault, and for regularly-accessed regions of code—multiple times per second—will approach the 2^{-128} level. A single parity bit, as used in at least some CPU level-one instruction caches (Advanced Micro Devices 2007), requires two bit-flips to induce an error. Suppose the parity is checked every R seconds, then the probability of an undetected bit-flip in any given second is

$$\lambda' = \frac{(\lambda R)^2}{R} = \lambda^2 R. \tag{2.9}$$

For code that is accessed even moderately often, this will come much closer to 2^{-128}.

The stronger error-correction codes used by the higher-level caches and main memory will detect virtually all such errors—with two-bit detection capability, the rate of undetected bit-flips will be at most

$$\lambda' = \lambda^3 R^2. \tag{2.10}$$

Returning to our concrete parameters, if $R = 100$ ms then one-bit-error-detecting ECC results in a false-acceptance rate of 2^{-108} after one month, much closer to the 2^{-128} level of security promised by analysis of the algorithm alone. With two-bit-error detection, even with a check rate of only once per 100 ms, the rate of memory errors is essentially zero, increasing the false-acceptance rate by a factor of only 10^{-14} above the 2^{-128} level that would be achieved in a perfect operating environment.

2.7 DISCUSSION

This counter-intuitive phenomenon, that increasingly consistent measurements can result in a reduction in certainty, is interesting in that it is commonly known and applied heuristically; trivial examples such as the estimation of coin bias (Sivia and Skilling 2006, section 2.1) have been well-analysed, but these unusual failure states are rarely, if ever, considered when a statistical approach to decision-making is applied to an entire system. Real systems that attempt to counter failure modes producing consistent data tend to focus upon the detection of particular failures rather than the mere fact of consistency. Sometimes there is little choice—a casino that consistently ejected gamblers on a winning streak would soon find itself without a clientele—however we have demonstrated that in many cases the level of consistency needed to cast doubt on the validity of the data is surprisingly low.

If this is so, then we must reconsider the use of thresholding as a decision mechanism when there is the potential for such failure modes to exist, particularly when the consequences

of an incorrect decision are large. When the decision rule takes the form of a probability threshold, it is necessary to deduce an upper threshold as well, such as was shown in Figure 2.4, in order to avoid capturing the region indicative of a systemic failure.

That this phenomenon was accounted for in ancient Jewish legal practice indicates a surprising level of intuitive statistical sophistication in this ancient law code; though predating by millennia the statistical tools needed to perform a rigorous analysis, our simple model of the judicial panel indicates that the requirement of a dissenting opinion acts to prevent conviction in cases that are substantially weaker than intuition suggests.

Applied to cryptographic systems, we see that even the probability that one particular bit in the system's machine code will be flipped due to a memory error over the course of a month, rendering the system insecure, is approximately 2^{80} times larger than the risk predicted by algorithmic analysis. This demonstrates the importance of strong error correction in modern cryptographic systems that strive for a failure rate on the order of 2^{-128}, a level of certainty that appears to be otherwise unachievable without active mitigation of the effect.

The use of naturally-occuring memory errors for Domain Name System (DNS) hijacking (Dinaberg 2011) has previously been demonstrated, and the ability of a user to disturb protected addresses by writing to adjacent cells (Kim et al. 2014) has been demonstrated, however litle consideration has been given to the possibility that this type of fault might occur simply by chance, implying that security analyses that assume reliable hardware are substantially flawed when applied to consumer systems lacking error-corrected memory.

2.8 CONCLUSION

In this chapter, we have discussed the importance of model verification in decision making. We have discussed several situations, ranging from the clearly-artificial to the realistic, showing that plausible hidden failure states can reduce confidence in a result to a level far below that which one would conclude via a naïve statistical analysis.

We have analysed the behaviour of systems that are subject to systematic failure, and demonstrated that with relatively low failure rates, large sample sizes are not required in order that unanimous results start to become indicative of systematic failure.

Our results suggest that in many cases it is highly advantageous to include some kind of 'model plausibility' check in a decision-making process, mimicking the human tendency to be suspicious of results that are so consistent that they are simply too good to be true.

2.8.1 ORIGINAL CONTRIBUTIONS

- We have investigated the effect of unidentified bias upon identity parades, and shown that even with only a 1% rate of failure, confidence begins to decrease after only three unanimous identifications, failing to reach even 95%.
- We have also applied our analysis of the phenomenon to cryptographic systems, investigating the effect by which confidence in the security of a parameter fails to increase with further testing due to potential failures of the underlying hardware. Even with a failure rate of only 10^{-13} per month, this effect dominates the analysis and is thus a significant determining factor in the overall level of security, increasing the probability that a maliciously-chosen parameter will be accepted by a factor of more than 2^{80}.

Having demonstrated that stochastic models of apparently-reliable systems are unavoidable if we are to adequately interpret their data, we now show how one can do so in practice. In the next chapter, we will apply stochastic methods to the interpretation of data from nonlinear sensors, demonstrating that stochastic signal processing can be used to mitigate nonlinearity.

Chapter 3

Nonlinear Sensing

Sensors, circuits, and devices under measurement often exhibit substantial nonlinearity. Though it is relatively simple to compensate for static nonlinearities, determination of the distorting function is often considered to be a rather Sisyphean task—the exact distorting function can vary with temperature, from device to device, with age, or for no apparent reason whatsoever. We develop here a method of online system identification that allows static distortion to be characterised with the aid of noise produced by the system itself.

3.1 INTRODUCTION

While random errors may be more obtrusive than those of the systematic kind, the nature of the latter is far more insidious. Random errors can be easily characterised by observation of the system with a constant input signal, but systematic errors are apparent only when one already has at least partial knowledge of the correct measurement result. Mere repetition is replaced by the more onerous task of calibration.

Our goal is to perform this characterisation with only minimal prior information, allowing accurate measurements to be taken even by unpredictable nonlinear systems.

3.1.1 WHY SENSE IN A NONLINEAR REGIME?

A reasonable question to ask is: *why should we accept nonlinear measurement systems in the first place?* Given that nonlinear signal processing imposes intellectual burden on the designer and computational burden on the system itself, such designs come with a substantial cost.

In fact, such nonlinearity is sometimes unavoidable. Sensors often have a nonlinear response; metal detectors, for example, can be driven into a nonlinear regime by very large targets, thus impeding classification (Kim et al. 2015). Others are inherently nonlinear in their operation (Sutin and Donskoy 1998).

In addition, we might not choose to sense in a nonlinear regime at all, but later be called upon to extract meaningful results from data that has been obtained by a system operated outside its operational limits. When it is no longer viable to increase the dynamic range of a system due to cost or power constraints, nonlinear postprocessing may prove to be the only way to meet its design goals.

3.2 CHARACTERISATION OF NONLINEARITY IN METROLOGY

The linearity of electronic systems has always been an important concern; even in the small-signal regime, much thought goes into avoiding distortion (Self 2010). In audio systems (Self 2010), the metric of choice tends to be Total Harmonic Distortion (THD). This includes both static and dynamic distortion, but its measurement is not particularly useful from the point of view of *removing* the distortion.

More useful from this perspective is to measure a static nonlinear transfer function of the system; that is, the function $f(\cdot)$ that relates the input x to the output y by $y = f(x)$.

This is not a complete model of the system, as it does not take into account the time-varying nature of the input, nor for the possibility of system state, such as capacitance. However, this is sufficient for many systems, and this is the type of system that we consider in this chapter. Systems with more complicated dynamics can be described in terms of memory-carrying models such as Volterra series (Volterra 1887, p. 105), however this is beyond the scope that we consider.

System identification (Ljung 1987) is a well-established field, but while there exist techniques (Wellstead 1981) for output-only identification of linear systems, the characterisation of nonlinear systems requires some knowledge of the input as in Bai (2002) and Voss et al. (2003).

3.2.1 DIRECT RESPONSE MEASUREMENT

The simplest way to measure the input-output function of a device is to attach a variable DC source to its input, and a digitiser to its output. The source is swept across the input range of the device, and the corresponding input and output values tabulated.

Despite its simplicity, this approach has several drawbacks when compared with the more complex schemes that we discuss in the following sections:

- The system cannot be used for anything else while the characterisation takes place. This is a problem when the response changes over time, as it requires that the system be taken down for maintenance on a regular basis.
- The approach requires high-precision equipment. The accuracy of the source and digitiser must be substantially greater than that of the device under test ("IEEE Standard for Digitizing Waveform Recorders" 2008, §4.7.1).

The second point bears further explanation; the measurement results are in fact of the combined nonlinearity of the source, the device under test, and the digitiser. If the source and digitiser are noticeably nonlinear—at least where they are not part of the device under test—then the results are no longer indicative of the performance of the device under test.

3.2.2 HISTOGRAM MEASUREMENT

Another approach is to apply a signal with a known *distribution*; even when a voltage cannot be produced with sufficient accuracy, a known distribution—a triangle or sinusoid, for example (Doernberg et al. 1984)—is often far easier to synthesise. The linearity of triangular waves means that they can be produced with a capacitor and a constant-current source.

Even greater accuracy is achievable by high-precision sinusoidal generators (Doernberg et al. 1984; Blair 1994). Others have proposed the use of a Gaussian test distribution (Martins and Cruz Serra 1999). If a highly-Gaussian noise generator is available, then this provides another choice of test signal.

A hybrid approach was proposed by Alegria et al. (2001), in which a generator of small triangular waves is swept over the range of the device under test. This substantially reduces the linearity requirements on the source, as well as the number of samples required; this approach is far more efficient than either direct measurement or full-range histogram approaches when testing Analog-to-Digital Converters (ADCs) or other devices that provide a direct digital output.

3.3 LINEARISATION BY NOISE MEASUREMENT

We combine these two approaches—all electronic systems generate some noise of their own, normally added to the signal of interest, and can be used as a test signal. This is similar to the situation discussed by Alegria et al. (2001), and we demonstrate that the internal noise in an electronic circuit can indeed provide sufficient information to characterise its static nonlinearity without knowledge of the input.

3.3.1 METHOD

We assume that the input noise of the system dominates all other noise sources, having constant variance σ_i^2. The signal at the input we denote $Z(t) = x(t) + N(t)$, the sum of a deterministic band-limited signal and a small amount σ_i^2 of white noise[4]. This noise need not be Gaussian, as we do not use any of its higher-order statistics. The system then produces a distorted output $Y(t) = f(Z(t)) = f(x(t) + N(t))$. For our purposes, we assume that the transfer function $f(z)$ is strictly monotonically increasing.

Let us suppose that the noise $N(t)$ is small; we can therefore linearise $f(z)$ about $x(t)$, yielding the estimate

$$Y(t) \approx f(x(t)) + N(t)f'(x(t)), \tag{3.1}$$

which allows us to write

$$f'(x(t)) \approx \frac{\sqrt{\mathrm{Var}(Y(t))}}{\sigma_i}. \tag{3.2}$$

[4]Nothing about this technique precludes other additive noise types, such as $1/f$ noise, but the optimal estimators are likely to differ.

By our assumption that $x(t)$ is band-limited, one may estimate $f(x(t))$ by low-pass filtering the distorted output $Y(t)$. These two calculations provide an estimate of $f'(x(t))$ for each value of $f(x(t))$, thereby admitting numerical computation of

$$x(t) = \int \frac{dx(t)}{df(x(t))} df(x(t)) \tag{3.3}$$

$$\approx \int \frac{\sigma_i}{\sqrt{\mathrm{Var}\,(Y(t))}} df(x(t)). \tag{3.4}$$

This leaves free σ_i and a constant of integration, which determine the gain and offset respectively. While these can be determined using calibration points, a robust linear regression (Seber and Lee 2003) between the distorted and compensated measurements provides sensible choices without modification of the system.

Note that the integral (3.4) is amenable to recursive estimation. However, initial simulations demonstrate an unacceptable level of drift, so we instead choose to wait until a relatively large number of samples are available before computing (3.4) in its entirety.

A practical implementation can use curve-fitting (Rivera et al. 2007; Rivera et al. 2008), either on the integrated values or the derivatives directly, to produce a more efficient representation of the transfer function with greater resistance to noise.

3.3.2 Estimation of the Derivative

In order to estimate the gain $f'(z)$, one must determine the local standard deviation at z. However, this poses a dilemma; a small averaging time will produce a relatively noisy estimate, but a large averaging time will have greater bias due to the presence of signal. High-pass filtering can remove much of this unwanted signal, but not all. We empirically find that the best results are achieved in most cases with between 50 and 100 samples. This parameter can be increased as the sample rate rises and so reduces the averaging time. However, if quantisation noise is significant in the signal being measured, the averaging time must be larger. Averaging times significantly shorter than the duration of the quantisation steps will produce impulses in the estimated derivative of the transfer function where the corresponding window contained a step.

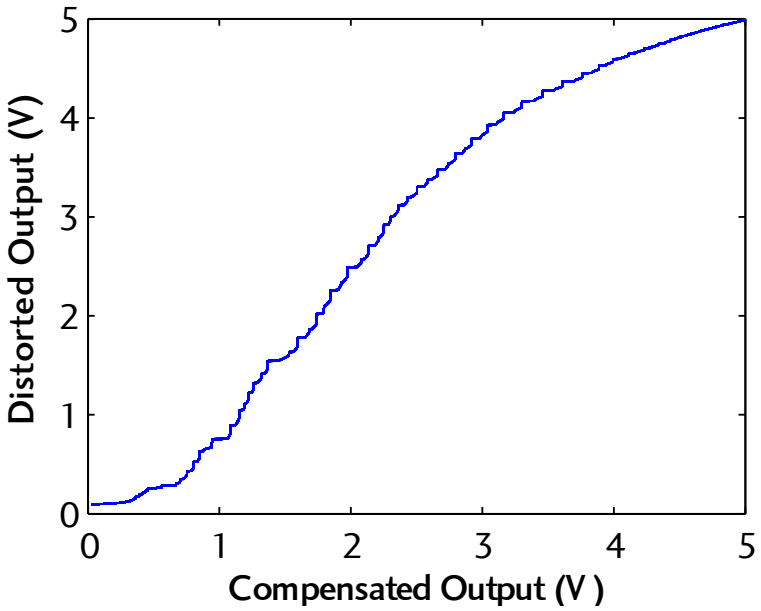


Figure 3.1: The experimentally-measured voltage transfer function $f(z)$ of the tested amplifier, as estimated using its noise variance. Saturation causes the gain to fall substantially near the supply rails at 0 V and 5 V. *After Gunn et al. (2013).*

3.3.3 HARMONIC DISTORTION

We have developed an implementation[5] of the above method using Labview and a National Instruments USB-6341 16-bit 500 kS/s Data Aquisition Unit (DAQ).

The DAQ generates a 10 Hz sinusoidal voltage, which is applied to a common-emitter amplifier without feedback—a single BC547 transistor with a 100 Ω pull-up resistor to $V_{cc} = 5$ V. The amplified signal is then digitised and processed in real-time, estimating the transfer function in Figure 3.1 using a combination of noise from the transistor and electronic and quantisation noise from the DAQ.

The effect of compensation on harmonic distortion is shown in Figure 3.2. The THD remains low even when the amplifier is driven well into saturation, extending the useful dynamic range of the amplifier by an order of magnitude.

3.3.4 STATIC ERROR

We claimed earlier that time-domain compensation is necessary for systems that operate near DC, where filtering cannot be used to suppress harmonics. In these situations, static

[5]Source code is provided on the accompanying media, or at https://github.com/LachlanGunn/stochastic-instrumentation-tools .

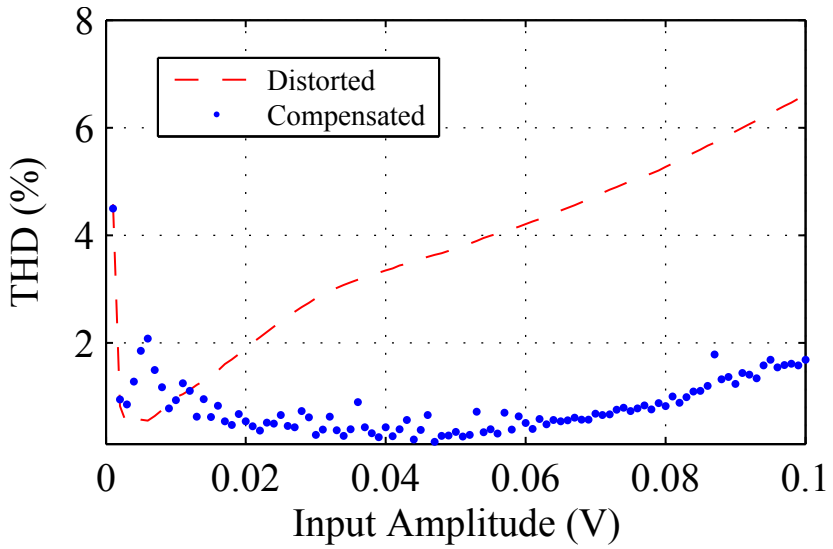


Figure 3.2: The Total Harmonic Distortion(THD) vs. signal amplitude for the tested amplifier. A 10 Hz sinusoid of each amplitude is applied to the transistor base and the THD of the digitised output measured using Labview. The large increase near zero amplitude is due to quantisation of the test signal. Each point shown is the median of three runs. *After Gunn et al. (2013).*

error ("IEEE Standard for Digitizing Waveform Recorders" 2008) provides a more useful measure of performance than THD. We apply a voltage ramp to the amplifier and then compensate the measured response. The deviation from an ideal ramp is shown in Figures 3.3 and 3.4.

Compensation allows recovery of the ramp with much improved linearity. While quantisation noise assists the reconstruction, its removal does not prevent the algorithm from functioning, as shown in Figure 3.4, demonstrating that electronic noise provides for a significant enhancement of linearity. We stress again that no preliminary calibration is required, and that this enhancement is achieved entirely in postprocessing.

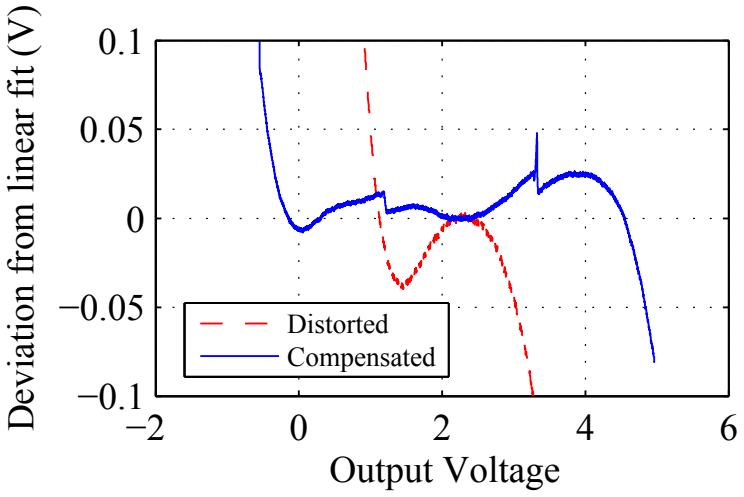


Figure 3.3: The experimentally-measured static error of the tested amplifier. A ramp between 0.7 V and 0.8 V is applied over 1 s and a linear fit to the central region subtracted to estimate nonlinearity. Compensation made the system linear over almost the entire output range despite heavy distortion of the signal. The discontinuities are caused by impulsive noise, which is removed in Figure 3.4 along with quantisation noise from the DAQ. *After Gunn et al. (2013).*

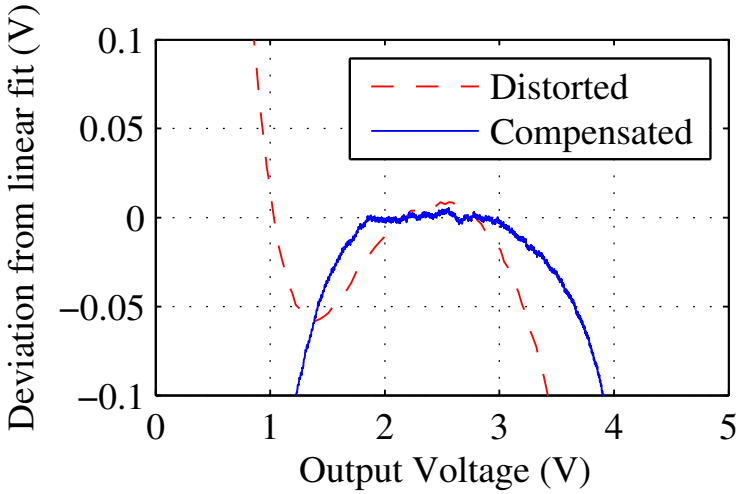


Figure 3.4: The experimentally-measured static error of the tested amplifier with quantisation noise excluded. A ramp between 0.7 V and 0.8 V is applied over 1 s and a linear fit to the central region subtracted to estimate nonlinearity. The input data is identical to that used in Figure 3.3, but a second implementation is used that ignores regions containing quantisation steps. This process also removes the quantisation noise visible in Figure 3.3. *After Gunn et al. (2013).*

3.4 OPTIMISATION FOR REAL-TIME USE

While tabulation and integration of the estimated $f'(x)$ values are conceptually simple processes, they are expensive in terms of both computation and memory, and cannot react to changes in the system without even more expensive pruning of the table.

Rather than attempting to directly integrate (3.2), as in (3.4), we instead construct an approximation $\hat{g}(f(x(t)))$ of $1/\bar{\sigma}_o(f(x(t)))$ that is amenable to recursive estimation.

We begin with three criteria for our approximation: first, it must be possible to efficiently compute arbitrary indefinite integrals without resort to numerical integration; second, it must be continuous in order that its integral is everywhere differentiable and so does not contain sharp corners that produce large amounts of harmonic content; third, it must be able to model a constant function exactly in order that a linear system can be represented. These criteria are satisfied by continuous piecewise linear functions, which we construct as the sum of radial basis functions (Buhmann 2000). We use this representation because it reduces the number of coefficients that are relevant to any one point.

We have selected basis functions of the form

$$r(x) = \begin{cases} 1 - |x|/\Delta & \text{if } x \in [-1, +1] \\ 0 & \text{otherwise,} \end{cases} \tag{3.5}$$

uniformly spaced Δ apart as shown in Figure 3.5. The number of basis functions that are used and the value of Δ are chosen according to the desired domain of approximation and the level of detail that is to be represented. The basis function widths are chosen to be 2Δ so that exactly two basis functions will cover each point, except at the centres of each basis function $r_k(x)$. The two basis functions will have opposing slopes, allowing a constant function to be trivially represented by equally-weighting adjacent basis functions.

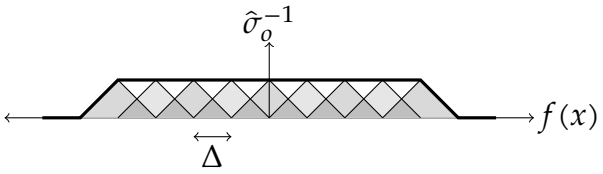


Figure 3.5: Basis functions for differential gain approximation, as described in (3.5). Triangular basis functions are chosen so that equally-weighted basis functions will sum to a constant function over the range of interest. *After Gunn et al. (2015a).*

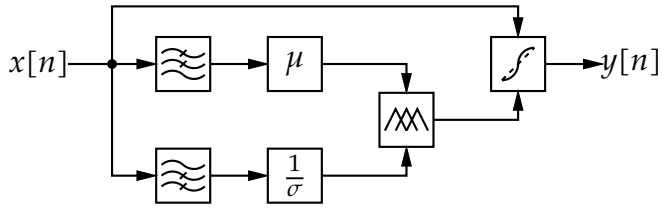


Figure 3.6: Adaptive compensator block diagram. Filters are used to separate signal from noise, and the mean and inverse standard deviation respectively are calculated from small blocks of samples. These are used to update the basis coefficients of the differential gain approximation, which is periodically integrated to update the distortion compensation function. *After Gunn et al. (2015a).*

The scaling coefficients c_n of each basis function are computed as a weighted average of the measured inverse-standard-deviations,

$$c_k[n] = \frac{\sum_{i=0}^{n} r_k(f(x(t_i)))\ \hat{\sigma}_o^{-1}(t_i)}{\sum_{i=0}^{n} r_k(f(x(t_i)))}, \tag{3.6}$$

with the weights proportional to the unweighted basis function. In order to accommodate variation over time of the distorting transformation f, we allow the weights to decay with time, resulting in the following estimation scheme:

$$a_k \longleftarrow \gamma a_k + (1-\gamma) r_k(f(x(t_i)))\hat{\sigma}_o^{-1}(t_i) \tag{3.7}$$

$$w_k \longleftarrow \gamma w_k + (1-\gamma) r_k(f(x(t_i))), \tag{3.8}$$

with the coefficients c_n computed as

$$c_k = a_k / w_k. \tag{3.9}$$

In practice we update only the coefficients of the two basis functions containing the current measurement within their supports.

Having constructed an approximation $\hat{g}(f(x(t)))$ of $1/\sigma_o(f(x(\tau)))$, we must now evaluate the integral from (3.4). The form of $r(x)$ allows this to be performed analytically:

$$\hat{x}(t) = \int_{-\infty}^{f(x(t))} \hat{g}(u)\ du \tag{3.10}$$

$$= \int_{-\infty}^{f(x(t))} \sum_{k=-\infty}^{\infty} c_k r(u - k\Delta)\ du. \tag{3.11}$$

Letting $n < f(x(t))/\Delta < n+1$, we use the fact that $\int_{-\infty}^{\infty} r(u)\ du = \Delta$ to simplify this to

$$= \sum_{k=-\infty}^{n-1} c_k \Delta + \int_{-\infty}^{f(x(t))} c_n r(u - n\Delta) + c_{n+1} r(u - (n+1)\Delta)\ du. \tag{3.12}$$

Noting that in this region the two terms are linear functions with slopes $-c_n/\Delta$ and c_{n+1}/Δ respectively, we make the substitution $v = u - n\Delta$ to produce

$$= \sum_{k=-\infty}^{n-1} c_k\Delta + \frac{1}{2}c_n\Delta + \int_0^{f(x(t))-n\Delta} \Delta(c_{n+1} - c_n)u + c_n\, du \tag{3.13}$$

$$= \sum_{k=-\infty}^{n-1} c_k\Delta + \frac{1}{2}c_n\Delta + (f(x(t)) - n\Delta)\, c_n + (f(x(t)) - n\Delta)^2 \frac{c_{n+1} - c_n}{2\Delta}. \tag{3.14}$$

This piecewise quadratic function may be evaluated far more efficiently than (3.4), paving the way for real-time implementation.

3.4.1 IMPLEMENTATION

We have implemented the technique described above on an STM32F407 microcontroller; the demonstration system is shown in Figure 3.7. Source code is available[6].

The system operates at a sampling rate of 1.55 MHz, using the on-board ADCs and DACs of the microcontroller. A block size of four samples is used, with a block computed and processed at 128-sample intervals. The computed $\hat{g}(\cdot)$ has 256 basis functions and is integrated every 1024 blocks to produce the corresponding piecewise polynomials.

Samples from the ADC are low-pass filtered to produce an estimate of the signal; this is subtracted from the original sample to produce an estimate of the signal's noise component. The compensating transformation is applied to the original (pre-filter) samples, and the results scaled and offset to match the Digital-to-Analog Converter (DAC) output range.

3.4.1.1 Triangular waves

In an attempt to describe the function of the device qualitatively, we have applied a distorted triangular wave to the device, with results shown in Figure 3.8. The amplifier distorts the wave to the point that one can no longer recognise its true form, however after compensation the wave is almost indistinguishable from its ideal form.

[6] Available on the accompanying media, or at https://github.com/LachlanGunn/stochastic-instrumentation-tools.

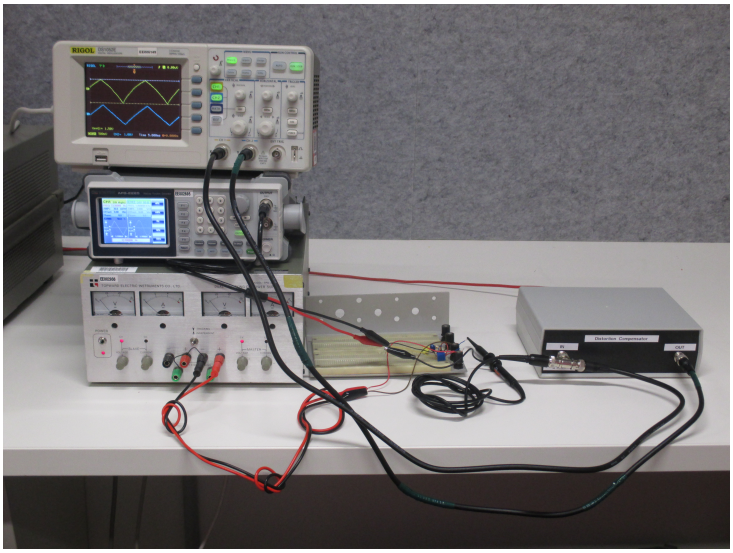


Figure 3.7: The distortion compensator unit in operation. A signal generator (left middle) produces a ramp signal, which is distorted by the transistor amplifier (centre), and processed by the compensator (right). The test circuit is powered by a linear laboratory power supply (left bottom), and waveforms measured with an oscilloscope (left top). Examples of the measured waveforms are provided in Figures 3.8 and 3.11. *After Gunn et al. (2015a).*

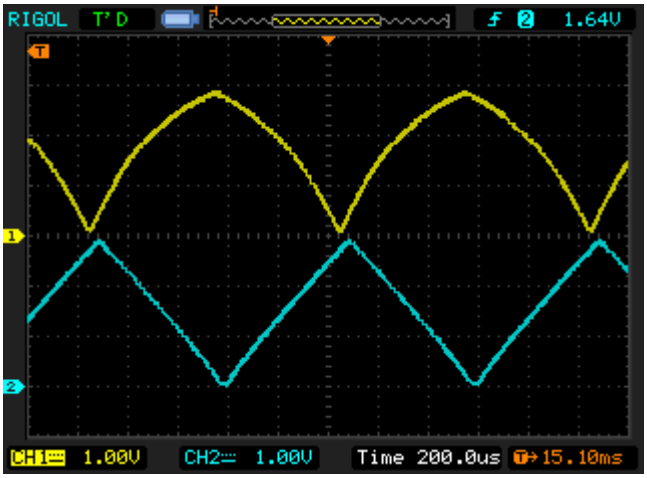


Figure 3.8: The compensator applied to a distorted triangle wave. The top signal is produced by applying a 50 mV, 1 kHz triangle wave to the base of a bipolar transistor configured as a common-emitter amplifier with rails of 0 V and 3 V. This is provided as input to the compensator, which produces the far less distorted signal underneath. *After Gunn et al. (2015a).*

The use of triangular waves allows simple histogram measurements ("IEEE Standard for Digitizing Waveform Recorders" 2008) to determine integral and differential nonlinearity (INL and DNL respectively). We measured histograms using an HP35665A Dynamic Signal Analyser, yielding the results in Figure 3.9. The improvement in differential nonlinearity is of particular note, remaining relatively flat over the entire range in comparison with the distorted signal with its enormous variation.

3.4.1.2 Total harmonic distortion measurements

The THD provides an alternative measure of distortion. We apply a sinusoidal signal to the distorting amplifier, and simultaneously measure the spectrum of the output before and after compensation. The THD of a signal with respect to the fundamental frequency f_0 is defined as

$$\mathrm{THD} = \sqrt{\frac{\sum_{n=1}^{\infty} |X(nf_0)|^2}{|X(f_0)|^2}}, \tag{3.15}$$

where $X(f)$ is the Fourier transform of the distorted signal.

We measure the first five harmonics and compute the THD before and after compensation, shown in Figure 3.10.

We see that the technique described above provides a substantial improvement in performance, especially at low signal levels where a reduction of 17 dB has been achieved. However, an improvement of more than 10 dB is possible even in the case of large input signals such as shown in Figure 3.11.

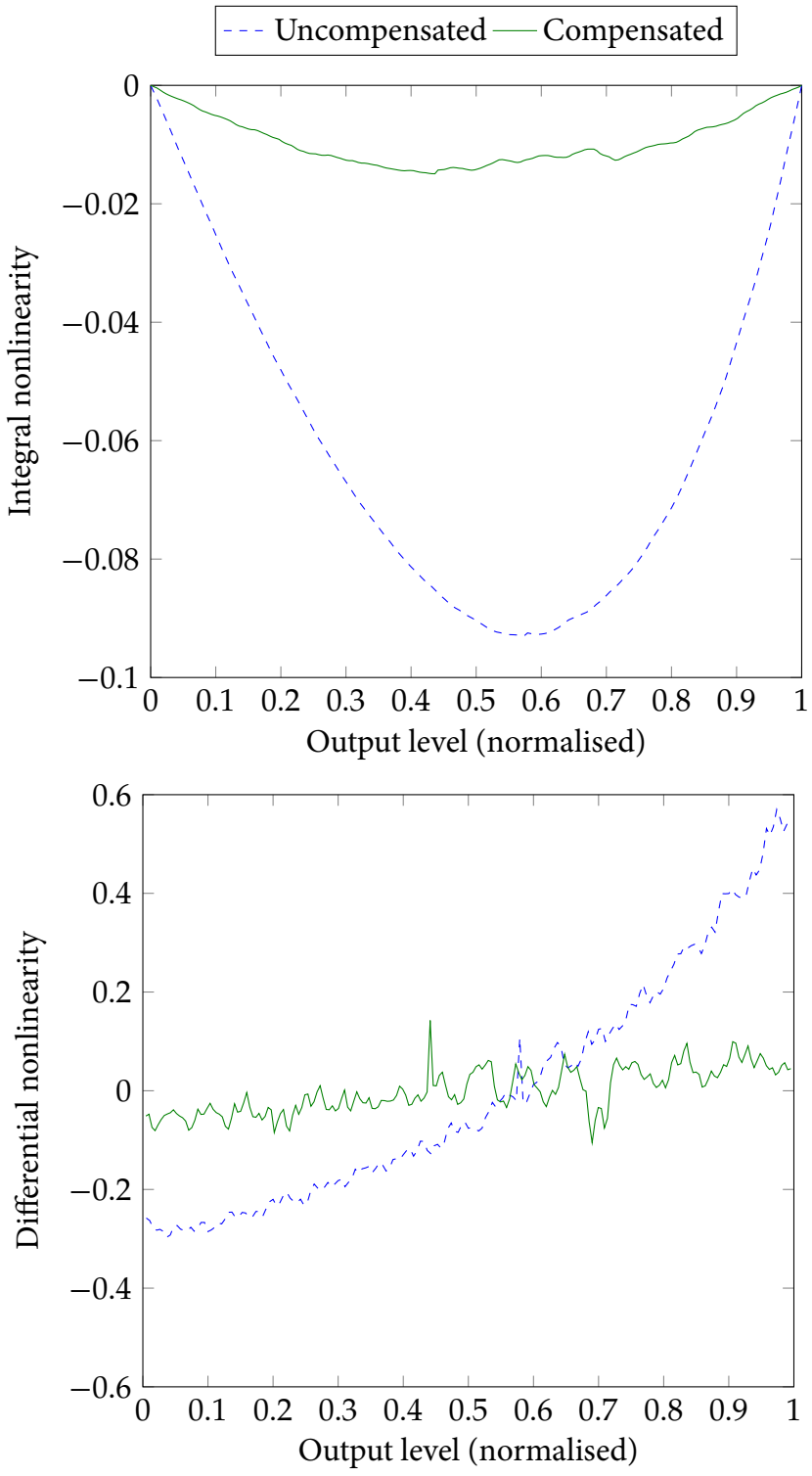


Figure 3.9: Measured integral and differential nonlinearity of the amplifier before and after compensation. By both measures, the compensation process has substantially improved the linearity of the amplifier. Linearity is calculated across the central 90% of the measurement range in order to remove measurement artifacts. *After Gunn et al. (2015a).*

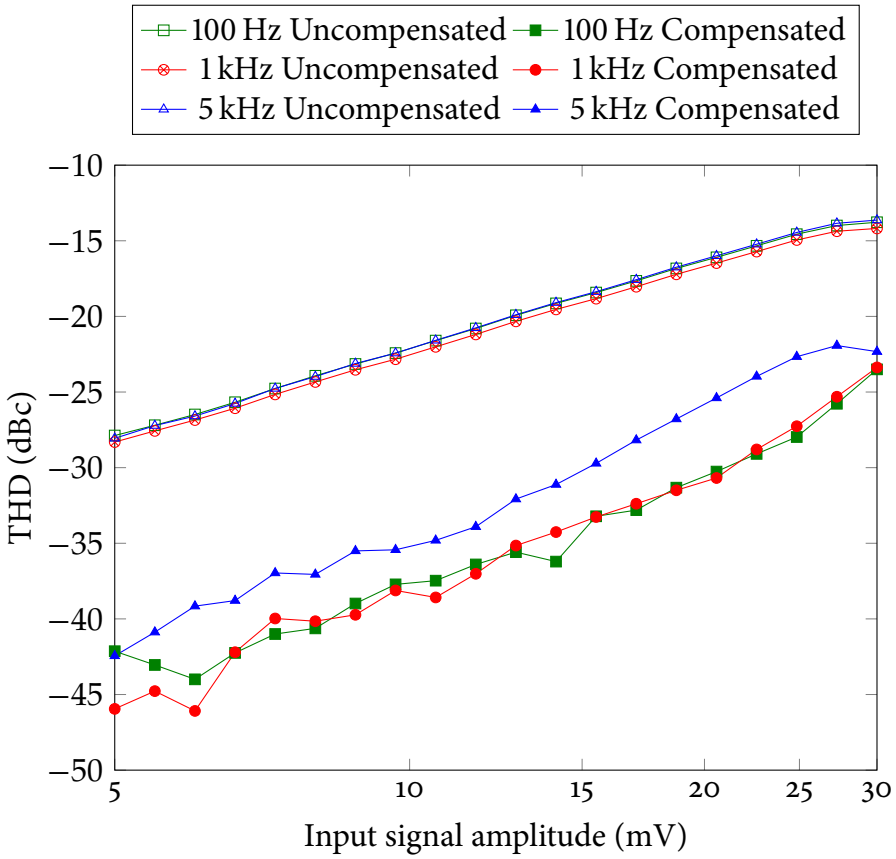


Figure 3.10: Measured THD of a common-emitter amplifier as a function of input voltage at 100 Hz, 1 kHz and 5 kHz, before and after compensation. At low signal levels (below around 10 mV input) an improvement of about 15 dB is achieved at 1 kHz and below. *After Gunn et al. (2015a).*

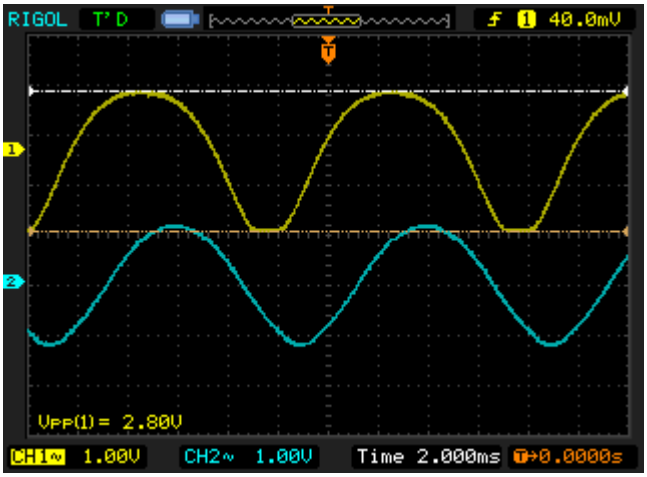


Figure 3.11: The compensator applied to a distorted sinusoid. Note that detail is extracted even from the base of the signal where the amplifier is heavily saturated. *After Gunn et al. (2015a).*

3.5 Adaptation for resource-constrained environments

While the use of curve-fitting to obtain a more easily integrated version of $(f^{-1})'(x)$ is relatively efficient, it suffers from some drawbacks when used on resource-constrained platforms. Estimation of the derivative of the inverse distorting function $(f^{-1})'(x)$ requires the computation of $(\sigma^2)^{-\frac{1}{2}}$; this is a relatively expensive operation which must be carried out for every noise estimate, and limits the rate at which the differential gain may be measured. Division and square-root operations are complicated and expensive in silicon, both in terms of transistor count and number of operations. Programmable logic often provides multipliers, but not dividers.

A further problem with this approach is that it is necessary to explicitly perform the integration—while this can be calculated analytically in terms of basis function coefficients, it is a slow process that cannot be carried out all at once. This raises the question of whether it is possible to perform the transfer function update operation directly in the integrated domain, thus avoiding the operation altogether.

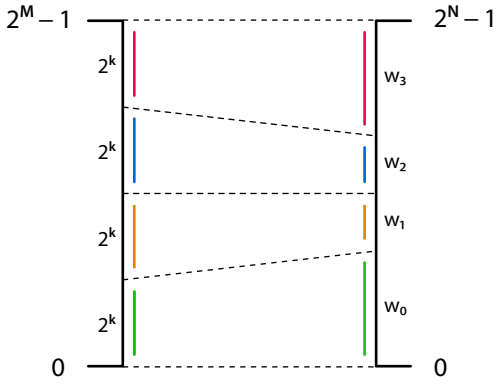
Until now we have measured the distortion present between the introduction of the dominant noise source and the digitisation of the signal. This approach follows directly from the mathematical analysis of the system, however we can, at the cost of this directness, simplify the system so as to avoid expensive computational operations. In this section, we consider a new approach that measures the distortion between the noise source and the output of the compensator; if the compensator is perfect then there will be no net distortion, but otherwise there will be some areas of greater-than-average differential gain. This can be corrected by adjusting the compensator to reduce the gain in this area, a substantially easier task than producing and inverting an explicit model of the nonlinear system in question.

Let us suppose that we have a discrete-time quantized signal $y[n]$, which takes values from $\{0, \ldots, 2^{M-1}\}$, and we wish to compensate this to yield a time series $x[n] \in \{0, \ldots, 2^{N-1}\}$. This latter time-series is found by compensating $y[n]$ with a nonlinear function $f_n(x[n])$ that varies with time. We choose $f_n(\cdot)$ to be piecewise linear with segments of size 2^k at the input. We may write this function as

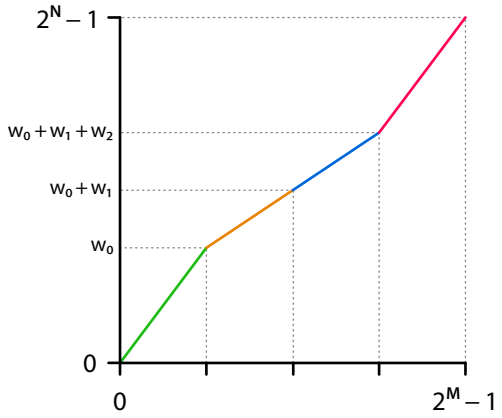
$$f_n(z) = \left\lfloor \frac{w_m[n]}{2^k}(z - m2^k) \right\rfloor + \sum_{i=0}^{m-1} w_i[n], \tag{3.16}$$

where

$$m = \left\lfloor \frac{z}{2^k} \right\rfloor \tag{3.17}$$



a) Partitioning of input and output spaces.



b) The resulting compensation function.

Figure 3.12: The construction of the compensating function described in Equation 3.16. The input and output ranges are partitioned into 2^k segments—the input uniformly, and the output with arbitrary widths, shown in (a). Corresponding partitions are mapped linearly onto one another, resulting in the piecewise linear function shown in (b). By using powers of two for the segment widths, time-consuming division operations are avoided. *After Gunn et al. (2016b).*

is the segment in which z lies, and the segment widths $w_i[n]$ in the output satisfy

$$\sum_{i=0}^{2^{M-k}-1} w_i[n] = 2^N; \tag{3.18}$$

that is to say, they cover the entire output range. The construction of this function is shown in Figure 3.12.

Previously we have measured the time-varying noise power at the input and then attempted to calculate the weights that would result in a stationary output process. In order to remove the need to calculate the necessary weights, we instead measure the noise power at the output, which allows us to use a simpler adjustment rule.

```
procedure UpdateSegmentWidths( x[n], σ²noise[n] )

    if σ²noise[n] < σ²avg then
        Δ ⟵ +1
    else if σ²noise[n] < σ²avg then
        Δ ⟵ −1
    else
        Δ ⟵ 0
    end if

    UpdateNoiseAverage( x[n], σ²noise[n] )

    i ⟵ FindSegment( x[n] )
    if wi[n] + Δ ∉ {1, …, 2^N−1} then
        return
    end if

    j ⟵ FindSegment( Random({1, …, 2^N−1}) )
    if wj[n] + Δ ∉ {1, …, 2N − 1} then
        return
    end if

    wi[n+1] ⟵ wi[n] + Δ
    wj[n+1] ⟵ wj[n] − Δ

end procedure
```

Figure 3.13: The width-update algorithm. *After Gunn et al. (2016b)*.

If the distorting function is exactly equal to the inverse of the compensating function $g(z)$, by our initial assumption the noise at the output will have a constant variance. If the variance is greater than average, this implies that the differential gain is also greater than average, and therefore we must reduce the gain of the compensating function in this region. Conversely, if the variance is less than average, we increase the gain of the compensating function.

We use a simple rule to determine the weight updates—if the noise is greater than average, the corresponding segment width $w_i[n]$ will be reduced by one, and if it is greater then $w_i[n]$ will be increased by one. However, after doing so the output-range constraint no longer satisfies (3.18); if $w_i[n]$ has been reduced, its output bin must be allocated somewhere else, and if it has been increased, its output bin must be taken from somewhere. Our key innovation is, rather than performing a time-consuming global rescaling, to simply give or take a random output bin—a random number is selected from $\{0, \ldots, 2^N\}$, and the width of the corresponding segment is increased or decreased by one. This is shown in greater detail in Figure 3.13. A proof of convergence for this algorithm will be the subject of a later work.

It is also worth noting that the noise need not be explicitly measured at the output of the compensator; if measured at the input, it may be converted to an output-equivalent noise by scaling its power by w_i^2, where the signal falls within bin i. Doing so allows the use of hardware filters to separate the noise from the low-frequency signal, thereby substantially reducing the computational burden on the processor.

We compute the average noise power $y[n]$ using a simple Infinite Impulse Response (IIR) filter of the form

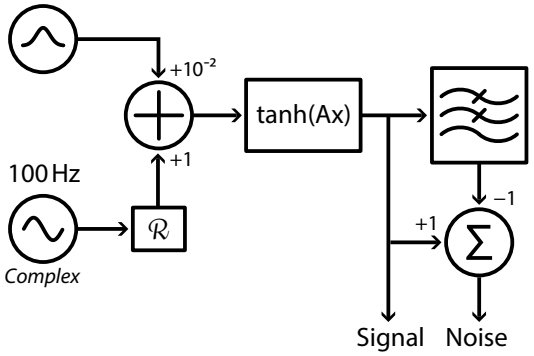
$$y[n] = (1 - 2^{-10})y[n-1] + 2^{-10}x[n],$$

chosen for ease of implementation. It remains to be seen whether more sophisticated techniques, yielding a more representative noise average, will provide a substantial advance in performance. One approach is be to take the median output noise variance, which would cause the random-bin-reallocation process to give and take bins 50% of the time, however it is not yet known whether the removal of this source of bias is important.

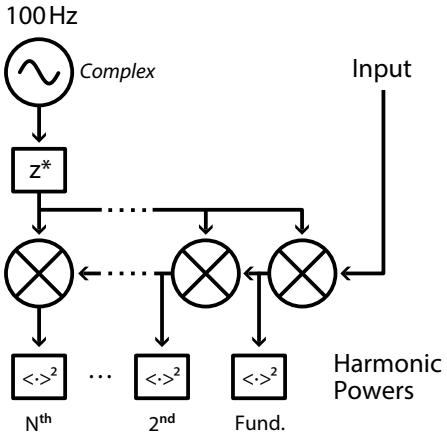
3.5.1 RESULTS

We simulate our system based on Equation 3.16 and UpdateSegmentWidths. The method used to generate the test signals and measure the THD ("IEEE Standard for Digitizing Waveform Recorders" 2008) of the output is shown in Figure 3.14. Ten million samples—ten seconds worth—were generated and processed, with the system being allowed five seconds to settle, after which the THD was computed from the remaining data points, up to the tenth harmonic. The inputs to the algorithm were quantised to fourteen bits, and the outputs quantised to twelve. We used 2^7 segments, thus yielding the parameters $M = 14$, $N = 12$, and $k = 7$ in the exposition above. Noise power was measured at the input and scaled by the square of the weights, yielding the output noise.

The THD of the output of the technique is shown in Figure 3.15. We see a substantial reduction in THD over a wide range of input distortion levels, however the achievable lower limit is determined by the block size; it is anticipated that refinements to the method will allow a reduction in the required block size whilst maintaining a low THD. The use of large block sizes appears to result in some 'peaking' at high levels of distortion, however at this level of distortion, the algorithm ceases to function, resulting in a net *increase* in distortion, and so would not be used, rendering the point moot. The results therefore indicate the use of larger block sizes. As this test is performed with a highly oversampled signal—by a factor of 5000—this is not a problem here. However, with lower oversampling ratios this design parameter may limit the achievable performance.



a) Test signal preparation.



b) THD measurement.

Figure 3.14: Test setup for the anti-distortion system; the sampling rate is $1\,\mathrm{MHz}$. The Gaussian noise was generated as a repeating sequence of 2^{16} random variates, formed by the sum of five scaled calls to the POSIX `rand` function. The unusual high-pass filtering approach is inherited from the arrangements in Gunn et al. (2015a), where we required both high-pass and low-pass outputs. In addition to what is shown in (a), we split the input into blocks of variable size—ranging from 2 to 32 in our tests—and compute mean-square value of the noise using the unbiased divide-by-$(N-1)$ formula. The harmonic powers are calculated by repeated complex downconversion; after k downconversions, the DC component will be that originally at kf, and can be found by coherent averaging. We measured up to the sixth harmonic. *After Gunn et al. (2016b).*

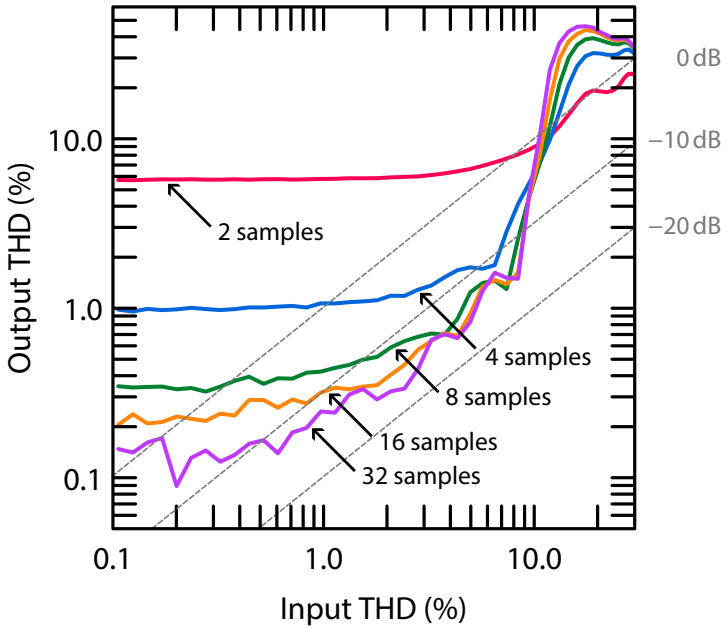


Figure 3.15: Input vs. output THD for the proposed method. A 100 Hz sinusoid had noise added and was distorted by the function $g(x) = \tanh(1.13x)$, with the level of distortion varied by changing the amplitude of the input signal, rescaling the output to match the output range to that corresponding to an input signal of amplitude one. The ultimate floor of the output THD is dependent upon the number of samples used—the block size—for the noise estimate. Each line represents a different block size. *After Gunn et al. (2016b).*

3.6 CONCLUSION

We have presented a novel technique for nonlinear system identification that uses the noise of the system in a fundamental way. This is a substantial advance on previous nonlinear system identification techniques, which require some kind of compromise with respect to the completely output-only measurements are most desirable. This allows our technique to be used retrospectively on datasets that were unexpectedly corrupted by nonlinearity.

3.6.1 ORIGINAL CONTRIBUTIONS

In this chapter, we have described a number of contributions to the state of the art:

- We have examined the output waveform of a transistor amplifier, and demonstrated that it contains noise components that can be used to characterise its response (Gunn et al. 2013).
- We have shown that this method of measurement and compensation is practical using current embedded systems, and demonstrated a real-time system that will automatically remove nonlinear distortion from its input (Gunn et al. 2015a).

- We have demonstrated by simulation that this compensation can be performed in a simplified manner that avoids expensive arithmetic operations such as divisions and square-roots, thereby rendering the approach more practical for integration with an ADC (Gunn et al. 2016b).

A natural extension of this topic is to look beyond simple sensing to an adversarial setting; when noise is used in a security system, the adversary may use more effective measurement techniques than anticipated by the designer of the system. We will investigate this type of system in the next chapter.

Chapter 4

Noise-based Communication

We have shown how noise can be used to enhance one's measurement abilities, but despite the practicality of the system previously described, it does not indicate any fundamental limits on what is achievable. An interesting application of noise-based measurement is the Kish key distribution (KKD) system, in which two parties agree on a random bit-string by simultaneously probing a pair of resistors with noise signals.

The KKD system promises the unconditionally provable security of quantum key distribution without the expense and complexity of quantum optics, allowing unconditional security from chip to chip or city to city. We present several attacks against this system, including experimental validation where practical. We consider also the information-theoretic implications of the KKD security claims, demonstrating that they are unrealistically strong.

4.1 KEY ESTABLISHMENT

One of the defining features of modern cryptography is the ability to produce a shared secret key by communicating over a public medium. Pioneered by Diffie and Hellman (1976), with a computational approach that is widely used today, less than ten years later Bennett and Brassard introduced a second, quite different, approach (Bennett and Brassard 1984) that derives its security from the laws of quantum mechanics.

An area of concern for most cryptosystems based on the first approach of computational security is the possibility that a large quantum computer will be built. The existence of a quantum algorithm (Shor 1994) allowing polynomial-time solution of the RSA and discrete logarithm problems suggests that if practical quantum computers are ever built then it will become possible to break all key-establishment systems in widespread use today.

Conversely, physical approaches such as quantum key distribution (Bennett and Brassard 1984) provide information-theoretic security guarantees and so are immune to computational advances. It is with this type of guarantee that we primarily concern ourselves here.

4.2 SECURITY DEFINITIONS

Before discussing particular key establishment systems, we first introduce several security definitions.

Definition 4.1 (Unconditional security)**.** *A system is unconditionally secure if its security properties hold against an adversary with unlimited computational power.*

This definition corresponds to *unconditional security* in Diffie and Hellman (1976) and Menezes et al. (1996/2001, §1.13.3(i)), and is a statistical property.

Most current systems use a weaker form of security, known as *computational security*. These forego the information-theoretic guarantees of unconditionally-secure systems in return for practicality of implementation.

Definition 4.2 (Computational security)**.** *A system is computationally secure if its security properties hold against any adversary that is unable to solve some computational problem $P(1^\lambda)$, where λ is the security parameter for the system.*

What this definition means in terms of security is encoded into the adversary definition; the adversary is most commonly limited by computation time and the lack of a quantum computer, but other limitations, such as memory usage, may also be considered.

This definition corresponds to *provable security* in Menezes et al. (1996/2001, §1.13.3(iii)). Computational security is only meaningful where it is proven relative to a problem that is hard to solve in some sense. This generally means that there is no known polynomial-time algorithm $\mathcal{A}$ to solve it with the information available to the adversary.

Proofs of computational security generally proceed as follows:

1. Suppose we can violate the security claims of the system with security parameter k in time τ.
2. For some polynomial $\mathsf{p}(k) \in \mathsf{poly}(k)$, this solves the underlying problem in time $\tau\mathsf{p}(k)$.
3. The system is therefore at most $1/\mathsf{p}(k)$ times easier to defeat than the underlying problem.

That is, we show that violating the system security claims allows the solution of a hard computational problem; the system is thus as secure as the problem is hard. Some common computational problems used as the basis for this type of security are the Diffie-Hellman problem (Diffie and Hellman 1976), the RSA problem, and the discrete logarithm (Odlyzko 2000) problem.

Definition 4.1 is often called *information-theoretic security*, because it concerns itself with inferences that an attacker can make based on the available information, irrespective of computational concerns; this is often most usefully described by information theory. When the security property of concern relates to secrecy—as it generally does in the case of key-establishment protocols that we discuss in this chapter—it is often useful to use an information-theoretic measure of security, known as the *secrecy rate* (Wyner 1975; Leung-Yan-Cheong and Hellman 1978; Maurer 1993).

Definition 4.3 (Secrecy rate)**.** *The secrecy rate C_s of a communications channel is the maximum rate at which information can traverse the channel whilst revealing an arbitrarily small amount of information to the eavesdropper.*

In this chapter, we use the secrecy rate in order to measure the ability of a key-establishment system to produce a key with unconditional security, thereby allowing a measure of the efficiency of the method—a low secrecy rate means that the protocol must be run many times in order to obtain a key sufficiently large for practical purposes.

4.3 CLASSICAL KEY ESTABLISHMENT PROTOCOLS

Though most of this chapter is concerned with physical apparatus for key establishment, most schemes in use today are purely computational, their security guarantees based upon computational hardness assumptions. These cryptosystems began to appear in the 1970s, and are used in essentially all cryptographic systems in use today.

The problem that they attempt to solve is as follows. Suppose Alice and Bob communicate via a public but authenticated channel. They execute a protocol that sends messages $m_1, m_2, \ldots, m_N$ across the channel. Then, Alice and Bob seek to compute a shared key k based on their private knowledge and the messages $\{m_i\}$ such that no polynomial-time adversary $\mathcal{A}$ can compute k from $\{m_i\}$ with non-negligible probability.

4.3.1 DIFFIE-HELLMAN

The first of the modern key-establishment protocols was the Diffie-Hellman protocol (Diffie and Hellman 1976). The computational problem underlying the Diffie-Hellman protocol is related to the discrete-logarithm problem, that of determining an integer k such that $g^k = y$ for some g and y in a finite cyclic group. This group can take a number of forms, but it is generally either a prime-order multiplicative group or an elliptic curve with the 'usual' group operation (Schneier 1996; Cohen et al. 2005).

Suppose that Alice and Bob have publicly agreed on a finite cyclic group $(G, \cdot)$ of order p and a generator $g \in G$ of the group. The simplified protocol operates as in Figure 4.1.

An eavesdropper cannot easily compute g^{ab} from g^a and g^b, whereas Alice can calculate $(g^b)^a$ and and Bob $(g^a)^b$ because they know a and b respectively. The most efficient known way to do so (Cohen et al. 2005) is to compute the discrete logarithm of g^a or g^b and then do as Alice and Bob do.

Significantly, one may publish a long-term g^a value, allowing anyone to generate a shared secret key with them on demand. However, if both a and b are chosen at random for each communication and destroyed afterwards, then the long-term key need only be used for authentication, and its compromise does not reveal prior messages. This property is known as *forward secrecy* (Günther 1989).

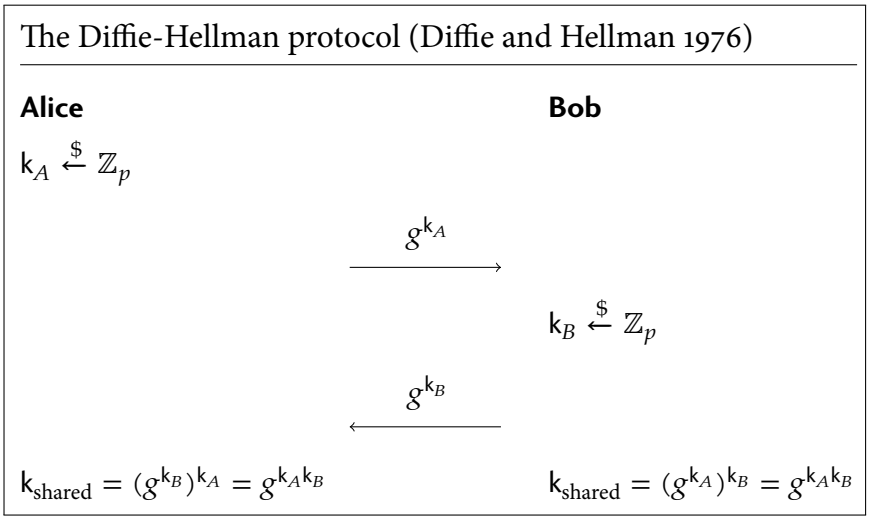


Figure 4.1: The basic Diffie-Hellman key-estabishment protocol. In this diagram, $\xleftarrow{\$}$ denotes the selection of a random element from the set to the right, $(G, \cdot)$ is a finite cyclic group of order p, and g a generator for the group.

4.3.2 RSA PUBLIC-KEY ENCRYPTION

Since its appearance in 1977, the RSA cryptosystem (Rivest et al. 1978) has set the benchmark by which all other public-key encryption and signature systems are judged. As with the Diffie-Hellman cryptosystem, we briefly describe the equations defining RSA and refer the reader to Ferguson et al. (2010, §12.5) for an explanation of the reasons why naïve implementations are insecure.

A public-key encryption cryptosystem is defined by three parts:

1. A probabilistic key generation algorithm KGen, yielding a public/secret key pair $(\mathsf{pk}, \mathsf{sk})$.
2. An encryption algorithm $\mathsf{Enc}_{\mathsf{pk}}(\cdot)$ mapping messages to ciphertexts.
3. A decryption algorithm $\mathsf{Dec}_{\mathsf{sk}}(\cdot)$ that recovers a message from a ciphertext.

In the case of the RSA cryptosystem, the encryption and decryption operations are similar to each other (Rivest et al. 1978):

$$\mathsf{Enc}_{\mathsf{pk}}(m) \equiv m^e \pmod{n} \tag{4.1}$$

$$\mathsf{Dec}_{\mathsf{sk}}(m) \equiv m^d \pmod{n}, \tag{4.2}$$

where parameters e, d, and n form the public and secret keys pk and sk:

$$\mathsf{pk} = (e, n) \tag{4.3}$$

$$\mathsf{sk} = (d, n), \tag{4.4}$$

where $n = pq$, with p and q large primes, and

$$d \equiv e^{-1} \pmod{(p-1)(q-1)}.$$

This can be shown (Rivest et al. 1978) to yield

$$\mathsf{Dec}_{(d,n)}\left(\mathsf{Enc}_{(e,n)}(m)\right) = m. \tag{4.5}$$

Normally e is a small fixed value, 65537 being a popular choice (Boneh 1999).

With this cryptosystem, a key can be generated and encrypted with the public key of the recipient, who may then decrypt it with the corresponding private key.

Simple RSA key establishment

Alice		**Bob**
		$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$
	$\xleftarrow{\quad \mathsf{pk} \quad}$	
$\mathsf{k} \xleftarrow{\$} \mathbb{Z}_n$		
	$\xrightarrow{\quad \mathsf{Enc}_{\mathsf{pk}}(\mathsf{k}) \quad}$	
		$\mathsf{k} \leftarrow \mathsf{Dec}_{\mathsf{sk}}(\mathsf{k})$

Like with Diffie-Hellman, this does not provide forward secrecy unless the asymmetric keys are ephemeral; a compromised private key can be used to decrypt all prior conversations. While one might theoretically generate ephemeral RSA keys, this is inefficient in practice—the RSA KGen procedure requires us to perform a potentially large number of primality tests (Ferguson et al. 2010, §12.4.5), making it far less efficient than Diffie-Hellman.

4.3.3 Shamir's three-pass protocol

Another approach, related to the Diffie-Hellman protocol, was proposed but never published by Shamir (Schneier 1996, p. 516), based on the idea of *commutative encryption*.

The idea was independently rediscovered by Kish and Sethuraman (2004), resulting in the development of a rather interesting mechanical analogy (Chappell et al. 2013): Alice wishes to send a message to Bob, and so places it into a box, which she locks with a padlock. She sends it to Bob, who attaches his own padlock and returns it to her. Alice removes

her padlock and sends the box again to Bob, who is finally able to access the contents with his own key. The simplicity of the scheme's physical implementation makes the hunt for a cryptographic analogue all the more enticing.

The formal definition of Shamir's protocol is as follows (Schneier 1996, p. 516):

Definition 4.4 (Shamir's three-pass protocol)**.** *Let* Enc_k *and* Dec_k *be the encryption and decryption functions for a symmetric cipher such that for all messages* m*, and keys* a, b*,*

$$\mathsf{Dec}_a(\mathsf{Enc}_b(m)) = \mathsf{Enc}_b(\mathsf{Dec}_a(m)).$$

Then, Alice and Bob select random keys k_A *and* k_B *and send the following messages:*

Shamir's three-pass protocol

Alice		**Bob**
$\mathsf{k}_A \leftarrow \mathsf{KGen}()$		$\mathsf{k}_B \leftarrow \mathsf{KGen}()$
$C_1 = \mathsf{Enc}_{\mathsf{k}_A}(m)$		
	$\xrightarrow{C_1}$	
		$C_2 = \mathsf{Enc}_{\mathsf{k}_B}(C_1)$
	$\xleftarrow{C_2}$	
$C_3 = \mathsf{Dec}_{\mathsf{k}_A}(C_2)$		
	$\xrightarrow{C_3}$	
		$m = \mathsf{Dec}_{\mathsf{k}_B}(C_3)$

It is straightforward to show the correctness of this scheme:

$$\begin{aligned} \mathsf{Dec}_{\mathsf{k}_B}(C_3) &= \mathsf{Dec}_{\mathsf{k}_B}(\mathsf{Dec}_{\mathsf{k}_A}(\mathsf{Enc}_{\mathsf{k}_B}(\mathsf{Enc}_{\mathsf{k}_A}(m)))) && (4.6)\\ &= \mathsf{Dec}_{\mathsf{k}_B}(\mathsf{Enc}_{\mathsf{k}_B}(\mathsf{Dec}_{\mathsf{k}_A}(\mathsf{Enc}_{\mathsf{k}_A}(m)))) && (4.7)\\ &= m. && (4.8)\end{aligned}$$

The approach taken by Shamir (Schneier 1996, p. 517) was to use finite-field multiplication as the encryption primitive. Kish and Sethuraman hypothesised that there might be some way to produce an information-theoretically secure three-pass scheme, however we show in Section 4.4.2 that this is not possible.

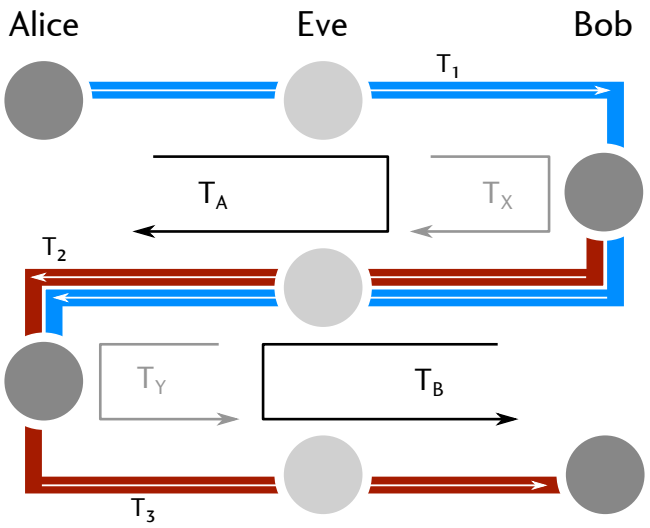


Figure 4.2: Consecutive round-trip measurements, where Bob's response to Alice forms a request for another measurement. Grey circles represent timestamping events. Alice measures the duration T_A of the blue round-trip, while Bob measures the duration T_B of the red round-trip. The transit time of the intermediate transmission contributes to the round-trip-time measurements of both Alice and Bob, providing a source of mutual information. An eavesdropper (Eve) measures the partial round-trip times T_X and T_Y. For convenience, during the analysis to follow we will subtract from each transit time the mean transit time T_μ.

4.4 A PHYSICAL IMPLEMENTATION OF THE SHAMIR THREE-PASS PROTOCOL

Clearly it is impractical to physically transport a lockbox from place to place, as discussed by Chappell et al. (2013). But whereas Chappell et al. (2013) discuss a class of linear-algebreic operations, we instead consider a physical analogue to the lockbox that performs a Diffie-Hellman-like handshake. In order to be practical, it must be possible to execute the protocol using only internet-based communications; this rules out systems that use wireless fading (Mathur et al. 2008) as a source of random key material.

One source of randomness on the internet is the transit time between two internet-connected terminals. If Alice and Bob rally packets back and forth via the internet, the time of each transit is a quantity common to the measurements of both, but measurable only with the addition of noise from the return trip (see Figure 4.2). An eavesdropper will suffer the same problem, however her noise will differ from that of Alice and Bob. This difference prevents her from taking advantage of the error correction performed by Alice and Bob during the information reconciliation (Liu et al. 2003) (IR) phase of processing, which discards bits likely to be incorrect, much like in the Bennett and Brassard (1984) protocol.

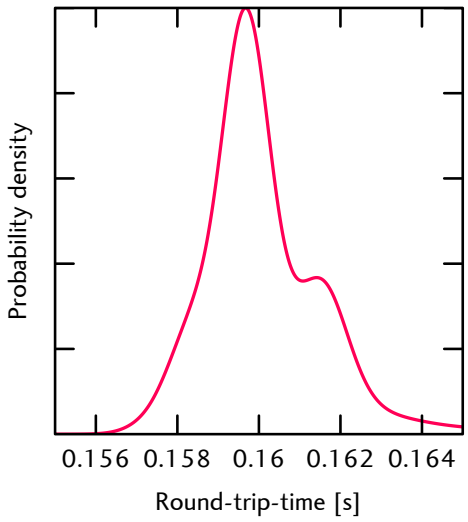


Figure 4.3: Distribution of round-trip-time measurements. Packets are sent from Los Angeles to Frankfurt and back, with the total round-trip-time shown. Note how the distribution is highly asymmetric.

We propose to extract random bits from the round-trip times by finding their median and declaring those times greater than the median to be a one, and those less to be a zero. With only one bit per round-trip, we avoid the problem that errors are more likely to fall into adjacent quantisation bins and so create correlations between bits.

While the distribution of round-trip times is actually quite skewed (Bolot 1993), as shown in Figure 4.3, we attempt to illustrate the technique theoretically by assuming transit times to be normally distributed and computing an upper limit on the key rate.

4.4.1 THE MUTUAL INFORMATION RATE BETWEEN ENDPOINTS

Denoting the mean transit time from Alice to Bob T_μ, let us write the three packet transit times from Figure 4.2 as $T_\mu + T_1$—Alice to Bob—$T_\mu + T_2$—Bob back to Alice—and $T_\mu + T_3$—Alice to Bob—respectively. Then, the deviations from the mean of the measured round-trips are $T_A = T_1 + T_2$, $T_B = T_2 + T_3$. Suppose $T_i \sim \mathcal{N}(0,1)$ for $i = 1$ to 3. Then, as the distribution

(and so the channel) is symmetric, we may calculate the bit-error rate as

$$p_e = 1 - \Pr[T_B < 0 | T_A < 0] \tag{4.9}$$

$$= 1 - \Pr[T_2 + T_3 < 0 \mid T_1 + T_2 < 0] \tag{4.10}$$

$$= 1 - \int_{-\infty}^{+\infty} 2\phi(t_2) \Pr[T_3 < -t_2 \cap T_1 < -t_2] \; dt_2 \tag{4.11}$$

$$= 1 - \int_{-\infty}^{+\infty} 2\phi(t_2)\Phi^2(-t_2) \; dt_2 \tag{4.12}$$

$$= 1 - \int_{0}^{1} 2u^2 \; du \tag{4.13}$$

$$= \frac{1}{3}, \tag{4.14}$$

where $\phi(t)$ and $\Phi(t)$ are the probability density and cumulative probability functions respectively of the normal distribution function. It should be noted that the derivation above holds for any zero-median symmetric distribution rather than just for the normal distribution.

This Bit-Error Rate (BER) of ⅓ corresponds to a channel capacity of 0.08 bits/measurement, suggesting that the achievable key rate with this technique may be too low for direct use as a one-time pad.

4.4.2 LIMITATIONS

Despite the allure of an information-theoretically secure key establishment method without specialised hardware, this method is not unconditionally secure and is necessarily dependent on the eavesdropper's inability to timestamp packets with perfect accuracy. This limits its use where an eavesdropper can timestamp packets on the link directly. To illustrate this point, we make use of the upper bound on the secrecy rate by Maurer (1993),

$$S(X;Y||Z) \leq I(X;Y);$$

here we denote the states of the two endpoints X and Y, and denote that of the eavesdropper Z. This states that the rate of secure communication is limited to the mutual information rate of the two endpoints. This implies that no protocol, no matter how clever, can provide secrecy using only independent random number generators at each end.

This is a fundamental limitation affecting three-pass systems such as those as described by Schneier (1996, p. 516), Kish and Sethuraman (2004), and Chappell et al. (2013); unless

the communications channel takes some active role, we can write (4.4.2) in terms of the random-number-generators of the two endpoints. These are uncorrelated, and thus the secrecy rate is zero.

The security of the system thus fails in the presence of a passive eavesdropper; we therefore do not consider the case of an active attacker, as this is a strict subset of the class of passive attackers.

In order to demonstrate the relevance of this inequality, imagine that Eve can timestamp Alice's and Bob's transmissions without error. Then, Eve has the same information as both legitimate parties, making secrecy impossible.

Now imagine that we have placed a router between Eve and the two endpoints. This introduces some randomness, but the two parties could achieve the same effect by simply adding a random delay to their transmissions; that is to say, it is as though they used random and independent keys. As discussed, this cannot form the basis for a secure system. Therefore, if Eve can measure without noise, information-theoretic security is not possible.

However, there are many cases in which this is not true. If an eavesdropper merely has copies of all traffic forwarded to them (such as by port mirroring), then routing delays and packet reordering provide the necessary source of noise (Zhang and Moore 2007). If the eavesdropper uses only a standard PC, uncertainty in the timing routines of its operating system provide an additional source of noise. These factors allow the system to provide security, especially against unsophisticated eavesdroppers using only commodity network hardware without hardware timestamping facilities.

4.4.3 Experimental round-trip measurements

In order to demonstrate this technique, we have constructed a test system to determine the performance of the method in the presence of an eavesdropper. The test system rallied User Datagram Protocol (UDP) packets back and forth along a chain of hosts—see Figure 4.4)—with the time of each arrival being timestamped. For our experiment, we timestamped the packet at four points:

- at the source, in Adelaide,
- on the same network, in Adelaide,
- in Los Angeles, and
- in Frankfurt.

Source code for the measurement system is shown in Section B.2.2. While the timescales were not synchronised, this information is sufficient to determine the various round-trip times.

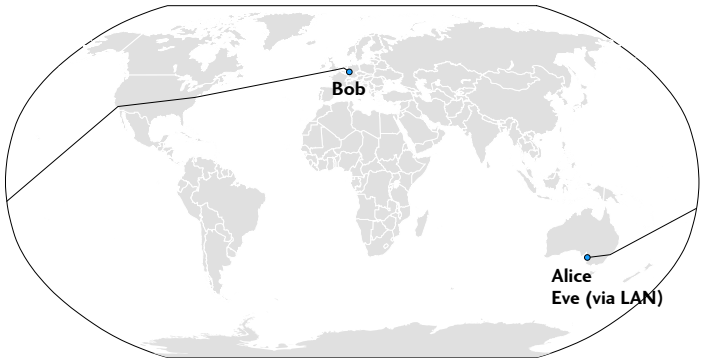


Figure 4.4: The constructed communications link. Alice is located in Adelaide, Australia, in the same room as Eve, to whom she is connected via the local network. The packet is then forwarded to through a relay in Los Angeles, and finally to Bob in Frankfurt. Alice sends a packet to Eve, who forwards it to Los Angeles, and finally to Bob in Frankfurt. The packet is then returned. At each step the arrival of the packet is timestamped, and the packet finally returned to Alice contains a time of arrival for Bob, and two for each intermediate node. The head of the chain in Adelaide, representing Alice, sends a packet which is timestamped at each of the three other hosts. The demonstration system described later does not transmit timestamps over the network, allowing each node to determine only its own round-trip time.

The effect of Information Reconciliation (IR) is shown in Figure 4.5. While the BER of the eavesdropper falls at first, it soon reaches a minimum value of around 2%. This demonstrates that a nonzero secrecy rate is achievable.

4.4.4 DEMONSTRATION SYSTEM

We have implemented the described protocol, which has been successfully operated over the internet. Source for the core module is given in Section B.2.3. Round-trip times are measured using UDP packets, whose times of transmission and receipt are determined using operating system routines. If a timeout occurs, due to a dropped packet for instance, the trip is marked as such and dropped during the reconciliation process. Information reconciliation is performed using the bit-pair iteration protocol (Maurer 1993).

Parameters for the information reconciliation and privacy amplification are determined automatically. A lower bound on eavesdropper BER is given as a parameter, and so their channel capacity is computed and thus the amount of information that they hold. From this,

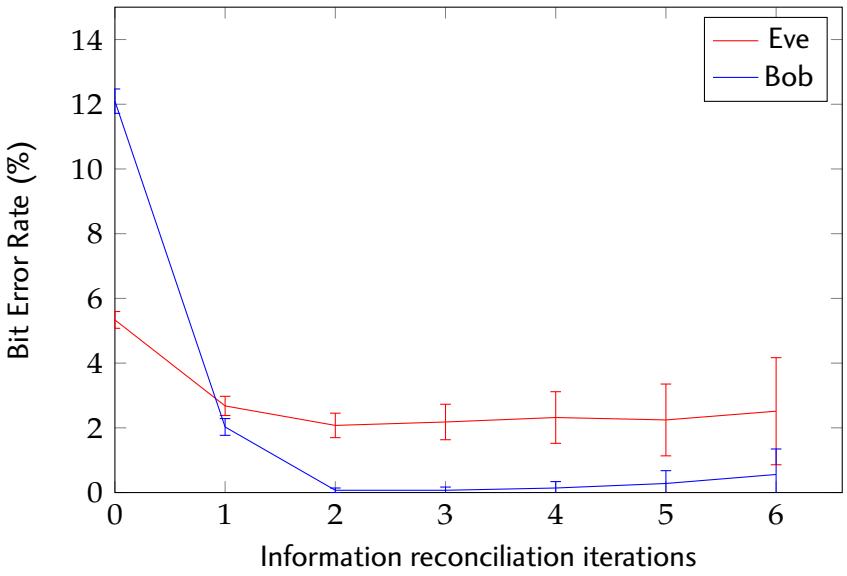


Figure 4.5: The effect of the bit-pair iteration protocol for Information Reconciliation upon the BER between Alice/Bob and Alice/Eve. This measurement used approximately 30,000 round-trips AU–US–EU, with the eavesdropper chosen to be the node in the same room as the sender. While the BER of the eavesdropper is improved slightly, it is not reduced to zero, which is evidence that there is sufficient measurement noise to allow secure communication. Error bars are shown for a 2σ confidence level.

a hashing function is chosen—the sum of some number of bits modulo two—that will discard sufficient information to eliminate the eavesdropper's knowledge of the secret key. As this process will increase the BER of the legitimate parties also, the target BER for the information reconciliation is reduced to compensate.

The BER of the channel is estimated using the error rate of the parity bits. A 2σ Agresti-Coull confidence interval (Agresti and Coull 1998) is constructed and back-propagated through the binomial probability mass function (Larsen and Marx 2012), yielding a confidence interval for the BER of the underlying channel. Then, the BER of the parity-checked output of each iteration can be predicted recursively in order to determine an interval containing the required number of IR iterations.

We succeed in generating keys at a rate of 13 bits/minute over the link shown in Figure 4.4, the lower bound on the eavesdropper BER set at 10^{-2}, based on the results shown in Figure 4.5. The 400 ms round-trip time makes the test relatively pessimistic by terrestrial standards, and greater key rates are potentially achievable across shorter distances.

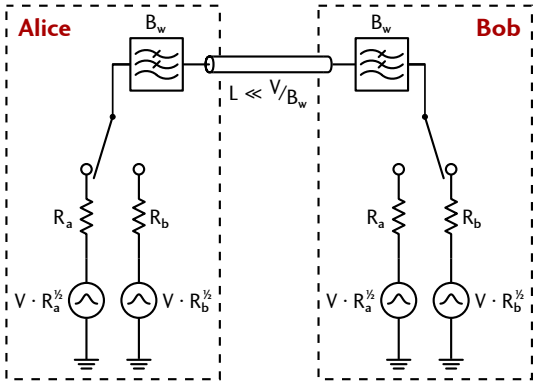


Figure 4.6: The KKD system under analysis. During each period, the two users set their switches randomly, resulting in a voltage on the line whose magnitude depends on the switch settings. If Alice and Bob select different resistor values, then in an ideal system the voltage and current on the transmission line provide no information on the switch settings.

4.5 THE KISH KEY DISTRIBUTION SYSTEM

The KKD system provides an alternative approach to the generation of shared secrets. Unlike the technique that we have described, it is electronic in nature, using noise on a transmission line. Two users inject noise into the line in such a way that they can each determine the noise power injected by the other user, but an eavesdropper can only measure the total power.

The idealized KKD system (Kish 2006c) has been proposed as a classical alternative to QKD (Bennett and Brassard 1984). Eschewing expensive and environmentally-sensitive optics, practical KKD can be implemented economically in a wider variety of systems than QKD. Such information-theoretic systems have been of great interest since the development of Shor's algorithm (Shor 1994), which, if successfully implemented on a significant scale, will potentially break most key-distribution schemes in use today.

The KKD system is claimed (Kish 2006c) to derive unconditional security from the second law of thermodynamics—the idea being that net power cannot flow from one resistor to the other under equilibrium.

An idealised KKD system is shown in Figure 4.6. Alice and Bob each apply a noise signal to a line through a series resistor. The voltage on the line is unchanged if the terminals of Alice and Bob are swapped; if the mean-square voltages applied by Alice and Bob are proportional to R_a and R_b respectively, then each branch of the circuit emulates a very hot—and thus very noisy—resistor. In this case, the second law of thermodynamics ensures that no net power flows through the line, and in the ideal case an eavesdropper, Eve, cannot determine which end has which resistance (Kish 2006c; Gingl and Mingesz 2014). Suppose $R_a < R_b$. If Alice

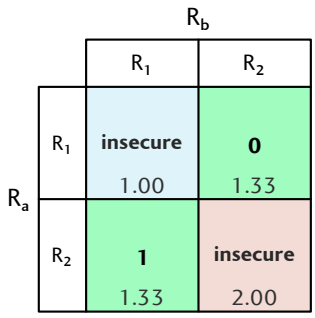


Figure 4.7: The four possible resistor states. Each time the protocol is run, the two switches are set at random, placing the system into one of the four states shown; at the bottom of each square is the mean-square line voltage for $R_a = 1\,\Omega$, $R_b = 2\,\Omega$, and $4kTB = 2$; this is only for illustrative purposes, and in practice the resistors will be of the order of several kilo-ohms. Two of the states are indistinguishable by an eavesdropper measuring only $\langle V^2 \rangle$, while Alice and Bob, who know their own selected resistor values, and so which row and column respectively the true state is in, can distinguish all four states. When running the protocol, Alice and Bob simply agree to drop any insecure bits from the generated random key.

and Bob randomly choose their resistances—resulting in corresponding noise amplitudes—to be either R_a or R_b, three possibilities avail themselves, shown in Figure 4.7: both choose R_a, yielding a small voltage on the line, both choose R_b, yielding a large voltage on the line, or one chooses R_a and the other chooses R_b, resulting in an intermediate voltage. In this third case, Alice knows the value of her own resistor, and so can deduce Bob's resistor via noise spectral analysis, and vice-versa. However, an eavesdropper lacks this knowledge, and so in the ideal case Alice and Bob have secretly shared one bit of information. This then forms the basis for Alice and Bob secretly sharing random numbers that can be exploited as secure cryptographic keys.

Several attacks against the KKD system exist, however none thus far have been shown experimentally to substantially reduce the security of the system (Mingesz et al. 2008).

The first attacks, proposed by Scheuer and Yariv (2006), rely upon imperfections in the line connecting the two terminals; the first exploits transients generated by the resistor-switching operation, while the second exploits the line's finite resistance. The former is foiled by the addition of low-pass filters to the terminals (Kish 2006d), while the latter was shown to leak less than 1 % of bits (Kish 2006d; Mingesz et al. 2008) in a practical system.

An attack by Hao (2006) instead focuses upon imperfections of the terminals; inaccuracies in the noise temperatures of Alice and Bob create an information leak. However, it was demonstrated (Kish 2006b; Mingesz et al. 2008) that noise can be digitally generated with a

sufficiently accurate effective noise temperature to prevent this attack from being useful in practice.

A theoretical argument has been made by Bennett and Riedel (2013) that no purely classical electromagnetic system can be unconditionally secure due to the structure of Maxwell's equations. It is argued that the upper bound on secrecy rate by Maurer (1993) must be zero because of the locally-causal nature of classical electromagnetics, and so an eavesdropper can perfectly reconstruct the key with the aid of a directional coupler. Kish et al. (2013) responded that a nonzero secrecy rate is unnecessary in practice, provided it can be achieved in the ideal limit.

It has been claimed (Kish and Horvath 2009) that transmission line theory does not apply to the the KKD system when operated at frequencies below $f_c = v/(2L)$, where L is the transmission line length and v the signal propagation velocity, because wave modes do not propagate below this cutoff frequency. We demonstrate that this is not the case by constructing a directional wave measurement device that is then used for a successful finite-resistance attack against the system. The position that frequencies below f_c do actually propagate is also supported by the fact that, at low frequencies, a coaxial cable is known to only support Transverse Electric/Magnetic (TEM) modes—these modes are known to have no low frequency cutoff (Jackson 1999). An exception occurs when the two ends of the line are held at equal potential; only standing waves possessing a frequency that is an integer multiple of $v/(2L)$ can fulfill these boundary conditions (Griffiths 2005). However, the KKD system differs in allowing arbitrary potentials to appear at the ends of the line, and so supports propagating waves.

4.6 Attacking KKD with wave measurement

A directional coupler separates forward- and reverse-travelling waves on a transmission line (Pozar 1998). We have constructed a similar device using differential measurements across a delay line (Gunn et al. 2014a), shown in Figure 4.8.

Consider the d'Alembert solution (Jackson 1999) to the wave equation in a medium with propagation velocity v,

$$v(t,x) = v_+\left(t - \frac{x}{v}\right) + v_-\left(t + \frac{x}{v}\right). \tag{4.15}$$

The forward-travelling component $v_+(\tau)$ differs from the reverse-travelling component $v_-(\tau)$ in the sign of its spatial argument. We use this to our advantage by computing the

linear combinations

$$\frac{\partial v}{\partial t} - \nu \frac{\partial v}{\partial x} = 2\frac{dv_+}{dt} \tag{4.16}$$

$$\frac{\partial v}{\partial t} + \nu \frac{\partial v}{\partial x} = 2\frac{dv_-}{dt}, \tag{4.17}$$

yielding the forward- and reverse-travelling waves as we desire. All that remains, then, is to determine $\partial v/\partial t$ and $\partial v/\partial x$.

The time derivative $\partial v/\partial t$ may be determined digitally from sampled values of $v(t)$. The spatial derivative is approximated as being proportional to the voltage across a short delay line, shown in Figure 4.8.

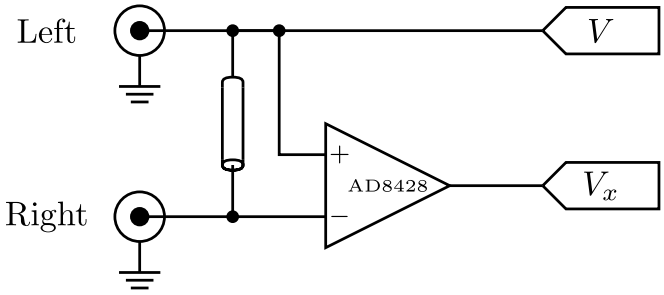


Figure 4.8: The analog frontend of the directional wave measurement device. Buffering, offset, gain control, and clamping are not shown. An instrumentation amplifier is used to measure the voltage across a 1.5 m length of coaxial cable, providing an estimate of $\partial v/\partial x$. After offset and gain adjustments, the signals are simultaneously sampled by the 12-bit ADCs of an STM32F407 microcontroller.

To demonstrate the practicality of this approach, we calculate the order of magnitude of the measured signal. Suppose we have a sinusoidal wave of frequency f and amplitude 1 V in a line of length L. The voltage in the line is given by

$$v(t,x) = \sin\left(2\pi f\left(t - \frac{x}{\nu}\right)\right), \tag{4.18}$$

and thus

$$\frac{\partial v}{\partial x} = -\frac{2\pi f}{\nu}\cos\left(2\pi f\left(t - \frac{x}{\nu}\right)\right). \tag{4.19}$$

Where the line is short compared to the wavelength in question, we can approximate the voltage across it as

$$V_x(t) \approx L\frac{\partial v}{\partial x} \tag{4.20}$$

$$= -\frac{2\pi f L}{\nu}\cos\left(2\pi f\left(t - \frac{x}{\nu}\right)\right), \tag{4.21}$$

resulting in a maximum voltage of $2\pi fL/v$. Supposing $f = 1\,\text{kHz}$, $L = 1\,\text{m}$, and $v = 1.5 \times 10^8\,\text{m}\,\text{s}^{-1}$, this is 42 µV—not a large voltage by any means, but well within the measurement capability of the system that we describe later, with an equivalent input noise of 2.8 µV RMS over the full 3.5 MHz bandwidth. Reflections at the ends of the line may reduce this, as the sign of $\partial v/\partial x$ switches with each reflection. If the terminations $\Gamma = (R - Z_0)/(R + Z_0)$ at each end are the same, and once again assuming a very short cable relative to the wavelength of the signal,

$$V'_x = V_x \sum_{n=0}^{\infty} (-\Gamma)^n \tag{4.22}$$

$$= \frac{V_x}{1 + \Gamma}. \tag{4.23}$$

The measured voltage will therefore be roughly half the originally-calculated value.

4.6.1 Experimental apparatus

We describe our implementation in more detail in Appendix A. The measurement of V_x is performed using a high-gain instrumentation amplifier. Common-mode rejection is an important factor here, with the differential signal calculated above being approximately 90 dB smaller than the common-mode one. We selected the AD8428 instrumentation amplifier, which has a 140 dB common-mode rejection ratio and a gain of 2000. The input-equivalent noise is specified as $1.5\,\text{nV}/\sqrt{\text{Hz}}$; over the full 3.5 MHz bandwidth, this results in 2.8 µV RMS of noise. As the voltage above is proportional to frequency, this noise floor limits the fidelity with which we can capture low-frequency signals. As we are interested in frequencies of only a few kilohertz, we therefore use the filtering facilities of the AD8428 to limit its bandwidth. The Signal-to-Noise Ratio (SNR), given a bandwidth B, is given by

$$\text{SNR} = \frac{V_{\text{in}}^2}{V_{\text{noise}}^2} \tag{4.24}$$

$$= \frac{4\pi^2 L^2 f^2}{v^2 B \cdot (1.5 \times 10^{-9})^2} V_{+\text{RMS}}\,, \tag{4.25}$$

By limiting the bandwidth to 12 kHz, we achieve more respectable input-equivalent noise of 164 nV RMS. This provides an SNR of 8 dB at 10 Hz, and 48 dB at 1 kHz.

We digitise the signal using the on-board ADCs of the STM32F407VG microcontroller that we used. This required that the signals be placed in the range 0 V–3.3 V. This is performed with simple op-amp circuitry—a difference amplifier based on the NE5532P op-amp (Self 2010) is used, with a buffered offset voltage on the low-impedance inverting imput, and

the voltage being measured applied to the high-impedance non-inverting input. A further inverting amplifier is included to allow tuning of the gain, however we found a gain of -1 V/V to be sufficient and therefore fixed the gain in order to avoid the distortion created by the potentimeter. A diode circuit is then used to clamp the output to the ADC voltage rails.

After digitisation, we high-pass filter the signals V and V_x in order to remove any DC offsets or mains interference. The signals are then combined to produce the left- and right-travelling waves. The time-derivative $\partial v/\partial t$ can be approximated by a difference operator, however in order to accommodate for the unknown propagation velocity and delay line length, common-mode leakage into V_x, and losses in the delay line, we instead use a first-order least-mean-squares (LMS) adaptive filter (Haykin 2002) for initial calibration. A signal source is applied to one port and the other is terminated; this produces a right-travelling wave on the line, but none travelling to the left. The left-travelling output V_- is used as an error signal for the LMS filter, suppressing any contribution from the right-travelling wave. The real part of the reflection coefficient, seen looking out of the right port, is computed

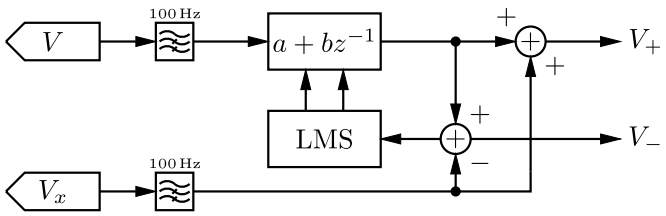


Figure 4.9: The digital signal processing of the directional wave measurement device, implemented on an STM32F407 microcontroller. A least-mean-squares filter is used at startup to determine the necessary filter coefficients; a signal is applied to one port while the other is connected to a terminator, and the filter coefficients adjusted to force $V_- = 0$. Filter updates are disabled once the apparent reflection coefficient becomes sufficiently small.

by a cross-correlation between left- and right-travelling waves. When this falls below 0.01, calibration is declared complete and filter updates cease. After calibration, we validate the system by configuring it as a reflectometer. Open and shorted measurements are made, yielding reflection coefficients of $+1$ and -1 respectively. The reflection coefficients of several resistors are also measured, again yielding the expected values.

We have used this device to implement the attack described above, using resistances $R_l = 1\,\mathrm{k\Omega}$, $R_h = 10\,\mathrm{k\Omega}$, and a coaxial transmission line of characteristic impedance $Z_0 = 50\,\Omega$. The voltage sources are produced by an arbitrary waveform generator, producing independent normally-distributed voltages over a frequency range of 500 Hz–5500 Hz. The bandwidth $B = 5\,\mathrm{kHz}$ results in an approximate correlation time of $B^{-1} = 200\,\mu\mathrm{s}$ (Kish

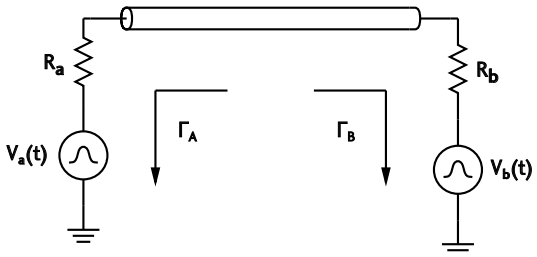


Figure 4.10: An s-parameter model of the KKD system. We consider the system in steady-state, with the switches in a constant position. The reflection coefficients Γ_A and Γ_B are determined by the resistor values R_a and R_b respectively.

2006a). Each configuration is set and the covariance matrices from (4.34) are measured during the setup phase. Resistor configurations are randomly selected for each test as would be the case in an operational system—though we used a pseudo-random number generator rather than a truly-random number generator—and the log-likelihood ratios are computed for the measured values of v_+ and v_-. Their differences are thresholded to compute (4.38).

4.6.2 CIRCUIT ANALYSIS

We begin our attack by analysing the system in Figure 4.10 to determine the forward- and reverse-travelling waves through the transmission line. Let us denote the equivalent noise voltages of Alice and Bob $V_a(t)$ and $V_b(t)$ respectively, and the waves injected onto the line $V'_a(t)$ and $V'_b(t)$. These are related by

$$V'_a(t) = \frac{1}{2}(1 - \Gamma_A)V_a(t) \tag{4.26}$$

$$V'_b(t) = \frac{1}{2}(1 - \Gamma_B)V_b(t). \tag{4.27}$$

Noting that the mean-squared thermal noise voltage is $\langle V^2 \rangle = 4kTBR$, we find that

$$\left\langle {V'_a}^2 \right\rangle = kTBZ_0(1 - \Gamma_A^2) \tag{4.28}$$

$$\left\langle {V'_b}^2 \right\rangle = kTBZ_0(1 - \Gamma_B^2). \tag{4.29}$$

As the transmission line in the KKD system is short—and so the forward- and reverse-travelling waves are equal throughout the line except for a loss factor α—we may write the left- and right-travelling waves at Bob's and Alice's ends of the line respectively as

$$V_+(t) = V'_a(t) + \alpha\Gamma_A V_-(t) \tag{4.30}$$

$$V_-(t) = V'_b(t) + \alpha\Gamma_B V_+(t) \tag{4.31}$$

and so

$$V_+(t) = \frac{V'_a(t) + \alpha\Gamma_A V'_b(t)}{1 - \alpha^2\Gamma_A\Gamma_B} \tag{4.32}$$

$$V_-(t) = \frac{V'_b(t) + \alpha\Gamma_B V'_a(t)}{1 - \alpha^2\Gamma_A\Gamma_B}. \tag{4.33}$$

We may write this in matrix form $\mathbf{v}_d(t) = A\mathbf{v}_i(t)$ and so find the covariance matrix $C = AC_iA^t$ of the directional components:

$$C = \frac{kTBZ_0}{\left(1 - \alpha^2\Gamma_A\Gamma_B\right)^2} \begin{bmatrix} 1 - \alpha^2\Gamma_A^2\Gamma_B^2 + (\alpha^2 - 1)\Gamma_A^2 & \alpha\Gamma_A(1 - \Gamma_B^2) + \alpha\Gamma_B(1 - \Gamma_A^2) \\ \alpha\Gamma_A(1 - \Gamma_B^2) + \alpha\Gamma_B(1 - \Gamma_A^2) & 1 - \alpha^2\Gamma_A^2\Gamma_B^2 + (\alpha^2 - 1)\Gamma_B^2 \end{bmatrix}. \tag{4.34}$$

When the line is lossless and so $\alpha = 1$, (4.34) is invariant under permutation of Γ_A and Γ_B, and so the covariance matrix provides no information on the choice of resistors. However, when $\alpha < 1$ this property fails to hold, allowing the choices of Γ_A and Γ_B to be determined from the distribution of (v_+, v_-); this allows an eavesdropper to attack the system by performing a statistical test between the two possible covariance matrices. Note that we need not measure the generator voltages themselves—which an eavesdropper cannot directly access—but merely the waves travelling in each direction.

4.6.3 STATISTICAL PROCESSING

We have derived a statistical representation of the noise that travels along the transmission line; while we might measure the power travelling in each direction in order to determine the resistor configuration, the distributions to be distinguished are very similar, resulting in a relatively large BER as was shown by Kish (2006d). However, comparison of the variances of v_+ and v_- is suboptimal. We derive an improved test using Bayesian methods and demonstrate that the two cases can be far more easily distinguished than with a direct difference-of-mean-squares test given by Scheuer and Yariv (2006).

Knowing the covariance matrices of $v_+(t)$ and $v_-(t)$ for each hypothesis, we may use Bayes' theorem (Larsen and Marx 2012) to determine the probability of each configuration. Let $C = 0$ and $C = 1$ refer to the events that $(R_a, R_b) = (R_h, R_l)$ and vice-versa, respectively.

Then,

$$\Pr[C=0|\mathbf{v}_+ \cap \mathbf{v}_-] = \frac{\Pr[\mathbf{v}_+ \cap \mathbf{v}_-|C=0]\Pr[C=0]}{\Pr[\mathbf{v}_+ \cap \mathbf{v}_-]} \tag{4.35}$$

$$= \frac{\frac{1}{2}p_0(\mathbf{v}_+,\mathbf{v}_-)}{\frac{1}{2}p_0(\mathbf{v}_+,\mathbf{v}_-)+\frac{1}{2}p_1(\mathbf{v}_+,\mathbf{v}_-)} \tag{4.36}$$

$$= \frac{1}{1+\frac{p_1(\mathbf{v}_+,\mathbf{v}_-)}{p_0(\mathbf{v}_+,\mathbf{v}_-)}}, \tag{4.37}$$

where $p_0(\cdot,\cdot)$ and $p_1(\cdot,\cdot)$ are the *multivariate* Gaussian PDFs for the measurements from each respective configuration.

The most probable state, then, is given by the maximum-likelihood estimator (Larsen and Marx 2012)

$$\hat{C} = \begin{cases} 0 & \text{if } p_0(\mathbf{v}_+,\mathbf{v}_-) > p_1(\mathbf{v}_+,\mathbf{v}_-) \\ 1 & \text{if } p_0(\mathbf{v}_+,\mathbf{v}_-) < p_1(\mathbf{v}_+,\mathbf{v}_-). \end{cases} \tag{4.38}$$

The comparison is more conveniently made in terms of the log-likelihood, which for the n-variate zero-mean Gaussian distribution with covariance matrix Σ is given by Cover and Thomas (2006)

$$\log p_\Sigma(\mathbf{x}) = \log\left[\frac{1}{(2\pi)^{\frac{n}{2}}|K|^{\frac{1}{2}}}e^{-\frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}}\right] \tag{4.39}$$

$$= -\frac{1}{2}\log|\Sigma| - \frac{n}{2}\log(2\pi) - \frac{1}{2}\mathbf{x}^T\Sigma^{-1}\mathbf{x}. \tag{4.40}$$

Noting that Σ is positive-definite, we may therefore write it in terms of its Cholesky decomposition $\Sigma = KK^T$, and so

$$= -\frac{1}{2}\log|\Sigma| - \frac{n}{2}\log(2\pi) - \frac{1}{2}\left\|K^{-1}\mathbf{x}\right\|^2. \tag{4.41}$$

Only the final term depends upon the data, and there only through the total power of a group of signals $K^{-1}\mathbf{x}$ formed by linear combinations of the measured waves.

It should be noted that this estimator differs substantially from that proposed by Scheuer and Yariv (2006), which makes a simple comparison of variances. The measured variables in our case are collected simultaneously and so exhibit the heavy correlations of (4.34). With these correlations, the likelihood-ratio test provides far better performance than the difference in the variances of the marginal distributions would suggest. However, if the voltage and current measurements are considered separately, as did Kish (2006d) and Mingesz et

al. (2008), then only the marginal distributions of each measurement are computed, these correlations vanish and so the estimator described in Eqns. 4.38 and 4.41 has substantially less power. The distribution of test statistics is shown in Figure 4.11 for a loss of 0.1 dB. The presence of correlation causes the distributions of test statistics to differ substantially, where otherwise they would be almost indistinguishable.

The results of simulation for various values of loss are shown in Figure 4.12. A pair of white noise processes are generated, Fourier-transformed, and the undesirable frequency components removed. They are combined according to (4.33) to produce the voltage waves, and the maximum-likelihood estimator is used to determine the resistor configurations. This demonstrates that our estimator can differentiate the two distributions without the unreasonably large sample sizes that were previously thought necessary by Kish (2006d).

4.6.4 EXPERIMENTAL RESULTS

Having demonstrated our attack in simulation, we proceed to experimental validation of the model. The estimation of $\partial v/\partial x$ is key to the operation of the device, however the circuit synthesis is dependent upon a wave-based analysis of the system. We therefore measure experimentally the frequency response of the electronically-estimated $\partial v/\partial x$, shown in Figure 4.14, with a wave travelling in a single direction in order to verify that our analysis is appropriate. Our apparatus is shown in Figure 4.13.

We expect to see a magnitude response linear in frequency and a constant $+90°$ phase response. This agrees with the experimental results shown in Figure 4.14, validating our analysis, and demonstrates that the signal through a short transmission line indeed propagates as a wave, in contradiction to the theoretical claims of Kish and Horvath (2009).

We measure the voltage components in each direction and compute the log-likelihoods (4.41). Their differences are thresholded to compute (4.38); the bit error rates for various averaging times and line parameters are shown in Figure 4.15. Even modest losses, below 0.1 dB, allow more than 99.9% of bits to be determined correctly in less than 20 correlation times, showing that the technique simulated in Figure 4.12 can be applied in practice.

4.6.5 PROPOSED COUNTERMEASURES AND ALTERNATIVE EXPLANATIONS

Several countermeasures to and alternative explanations of this attack have been proposed in response to a preprint of this chapter; we take a moment to discuss these papers point-by-point.

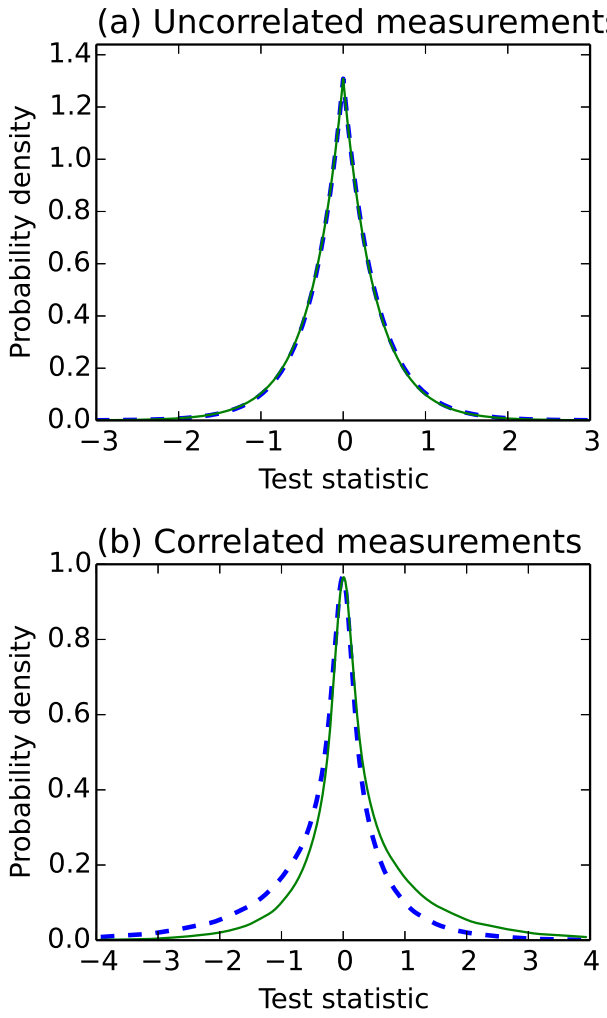


Figure 4.11: Log likelihood-ratio test statistics for each permutation of resistors in (4.34), as in (4.41) with scaling-factors omitted. The dashed lines correspond to the case where $(R_a, R_b) = (R_l, R_h)$, and the solid lines to $(R_a, R_b) = (R_h, R_l)$. Parameters are $R_l = 1\,\mathrm{k}\Omega, R_h = 10\,\mathrm{k}\Omega, Z_0 = 50\,\Omega$, and $\alpha = -0.1\,\mathrm{dB}$. In (a) the covariances are set to zero, and so (4.38) reduces to a simple power comparison. The distributions are almost indistinguishable. In (b), the measurement variables are drawn from a correlated bivariate distribution having the same marginal variances, and are far more distinguishable. In either case, as losses increase and so the variances of the measurements and transformed measurements respectively differ more greatly, the two distributions, which mirror each other about zero, become increasingly asymmetric and so far more distinguishable.

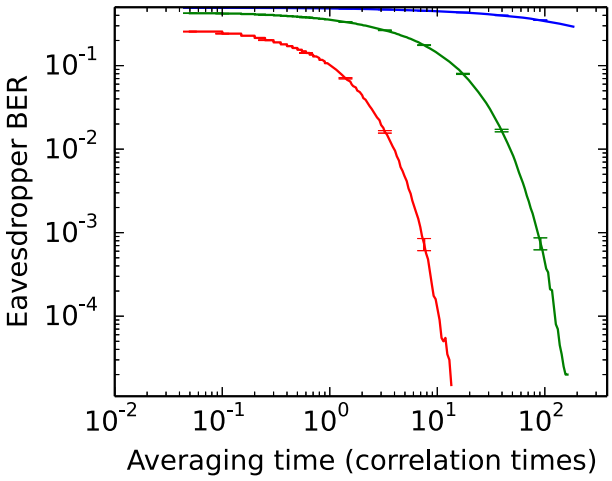


Figure 4.12: Simulated eavesdropper bit-error-rate as a function of averaging time, for line attenuations of 0.01, 0.1, and 1.0 decibels respectively from top to bottom. The idealised system has no attenuation at all, resulting in a bit-error-rate of 0.5. The link parameters are $R_L = 1\,\mathrm{k}\Omega, R_H = 1\,\mathrm{k}\Omega, Z_0 = 50\,\Omega$. Note that the averaging time is expressed in multiples of 200 µs. This is the correlation time (i.e. reciprocal of the system bandwidth) so that the results are bandwidth independent. Transmission lines with greater loss are more susceptible to attack, with substantial attenuations providing little protection due to the shunt currents that they produce. The error rates are estimated from a sample size of 10^5, with 2σ error bars shown.

4.6.5.1 Arguments against the transmission-line model of the KKD system

Kish et al. (2013) and Chen et al. (2014a) argue on several grounds that the wave-based model that we have used is inaccurate. It is first claimed that the wave equation on a finite domain does not admit sinusoidal solutions other than of frequencies $f_k = k\nu/2L$, where ν is the propagation velocity and L the length of the transmission line. However, this quantisation effect is induced by boundary conditions of the form $v(0) = v(L)$; in the KKD system, resistive terminations allow arbitrary potentials to appear at the two ends of the line and so this does not occur. We also note that these spatial frequencies do not directly correspond to temporal frequencies in the injected signals, but are instead indicative of the spatial spectrum of the periodic extension of the voltage distribution along the line.

It is next claimed by Chen et al. (2014a) that the signals within the KKD system cannot be waves because their energy does not exchange between electric and magnetic fields. However this will always be the case. Consider an infinitely long coaxial cable driven by a sinusoidal source $V_0(t)$. It is claimed by Chen et al. (2014a) that the relationship between the instantaneous voltages and currents in a small initial segment of the line will cause the energy

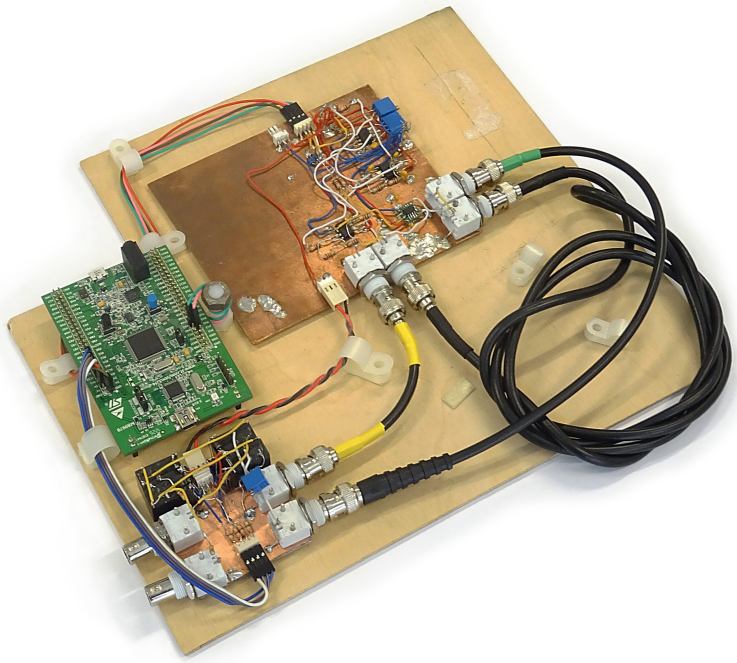


Figure 4.13: The constructed directional coupler. Noise signals enter the KKD board at the bottom-left, which contains the two resistors along with a set of relays that determines which will be connected to which end of the delay line, shown at right. The upper board contains all of the analog circuitry for the attack. The measurements are then provided to the microcontroller for further processing.

to be evenly split between electric and magnetic fields. As we are considering an infinitely long coaxial cable, the voltages and currents contain no reflected components, and so will be given by

$$V(x,t) = V_0(t - x/v) \tag{4.42}$$

$$I(x,t) = V_0(t - x/v)\frac{1}{Z_0}. \tag{4.43}$$

The distribution of energy between electric and magnetic fields therefore does not change as the signal propagates along the transmission line. The voltages and currents are known (Pozar 1998) to satisfy the wave equation, and yet they do not exchange energy in the manner suggested by Chen et al. (2014a).

It is further claimed that a lack of discretisation of frequencies disagrees with the calculations of Planck and would invalidate Planck's Law. However, it is incorrectly claimed by Chen et al. (2014a) that Planck's Law is derived for radiation inside a black-sided box; in fact, the box analysed by Planck and Masius (1914) is perfectly conductive. It is these perfectly conductive edges that induce the quantisation of the spatial frequencies discussed by Planck

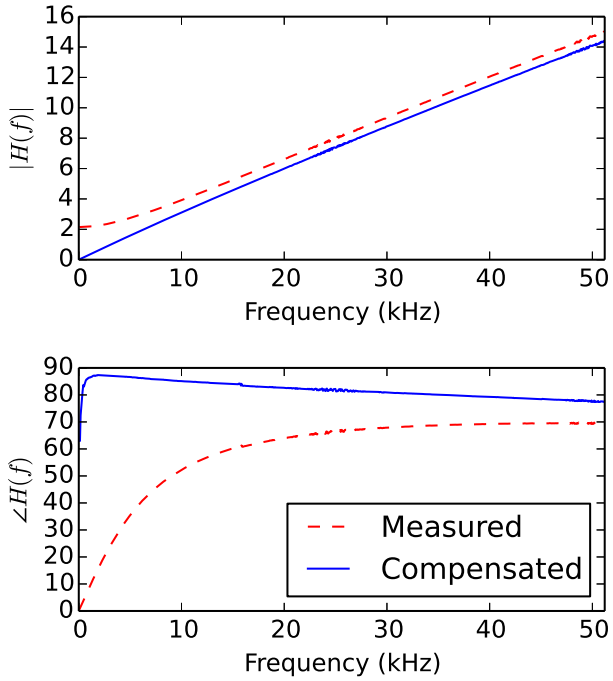


Figure 4.14: Measured frequency response of the $\partial v/\partial x$ estimation circuit in Figure 4.8. The derivative increases linearly with frequency, as would be expected from the d'Alembert solution to the wave equation. The response $H(0)$ at DC is subtracted in order to remove the effect of wire resistance, yielding the 'compensated' curves above. After this correction we see $\angle H(f)$ approximating the expected $+90°$ constant phase response, slightly drooping due to the limited frequency response of the system.

and Masius (1914). In simple terms, recall that Planck's formulation solves the ultraviolet catastrophe by introducing an *upper* frequency cut-off via quantisation. An attempt by Chen et al. (2014a) to use this analogy to argue for a *lower* frequency cut-off in a coax line is therefore not valid and appears to have the situation inverted.

Another argument has been made by Chen et al. (2014a) against the presence of waves, using the equipartition theorem. It is claimed that the equipartition theorem requires each wave mode of the transmission line to possess an energy of $\frac{1}{2}kT$, and that for a line in thermal equilibrium with the generators, the power on the line is insufficient to excite even a single wave mode. However, the non-idealized KKD system is not a thermodynamically closed system, but uses artificial noise sources and has resistive terminations. These terminations dissipate power into the environment, and the noise sources must be supplied with external power; the KKD system therefore is not in thermal equilibrium and the equipartition theorem does not apply.

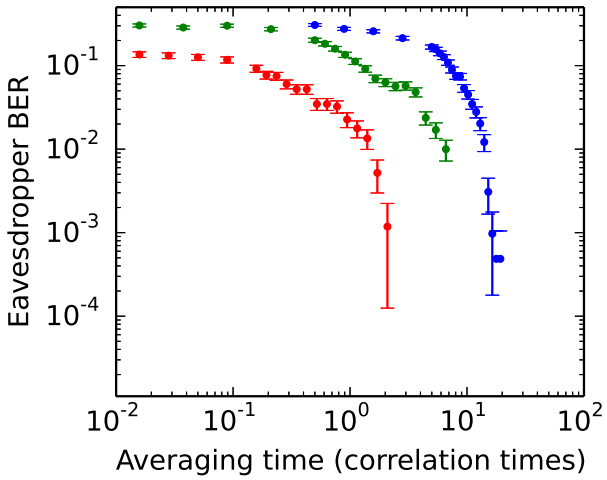


Figure 4.15: Measured eavesdropper bit-error-rate as a function of averaging time and line attenuation. The line is approximately 2 m in length and has a loss of less than 0.1 dB. From top to bottom, 0 dB, 0.1 dB, and 1 dB of additional attenuation provided by inserting an in-line T-attenuator at one end of the line. The shunt resistance of the T-attenuator violates the assumption of zero shunt current, meaning that the no-attenuation case—0 dB; top—is of primary interest here, as pointed out by Kish et al. (2015).

It is also claimed by Chen et al. (2014a), based on a lumped-model analysis, that the phase velocity of the propagating signal is dependent upon the line terminations, invalidating the use of the d'Alembert solution to the wave equation. However, this analysis conflates phase and propagation velocities, and similar results—can be derived from a wave-based analysis. We note also that, contrary to the claims of Chen et al. (2014a), for *guided* modes, superluminal phase velocities do not violate special relativity as they do not imply superluminal wave signal propagation (Sommerfeld 1952; Brillouin 1960, p. 139).

Contrary to the implication of Chen et al. (2014a), there is no definitive definition of a wave in the literature. Even attempting to define a wave as a solution of the wave equation is overly restrictive, as waves in dispersive media do not strictly satisfy the standard wave equation (Brillouin 1960). Thus physics texts such as Truesdell and Noll (2004) define a wave in the broadest possible terms as a transfer of energy from one state to another with a finite velocity. A wave does not even need to be periodic—for example, it can be overdamped or even chirped. It appears that, in each argument, Chen et al. (2014a) preselect their own *ad hoc* definition of what a wave is in order to arrive at a non-standard viewpoint.

4.6.5.2 *Experimental critique*

It was suggested by Chen et al. (2014b) that mains interference or DC offsets might be responsible for our results, as they would produce an apparent DC offset during each measurement. Note that DC offsets are removed by high-pass filtering after digitisation, as shown in Figure 4.9, and 50 Hz interference is suppressed as well. The delay line is shielded by the coaxial braid, and is wound in a non-inductive bifilar configuration (Kazimierczuk 2013) in order to further reduce mains pickup. The magnitude of the 50 Hz interference measured on the V_x channel—see Figure 4.8—is 15 mV RMS after amplification, and remains constant whether or not a complete circuit exists through the two resistors to ground, thus suggesting this effect to be insignificant on that channel. Interference picked up by the V channel—the quantity considered by Chen et al. (2014b)—increases with the establishment of a current loop, but at 40 μV RMS this is more than 85 dB below the generator signal, and so insignificant in the short time over which we average.

It is further suggested by Chen et al. (2014a) that our apparatus might have non-Gaussian signals present, and that this known vulnerability might be responsible for our results. However, our method uses only second-order statistics, and so does not depend upon the distributions of the signals, but merely their variances and correlations, which can be trivially computed as above.

4.6.5.3 *Proposed countermeasures*

A countermeasure to finite-resistance attacks has been proposed by Kish and Granqvist (2014a). They propose to boost the noise temperature of one source in order to compensate for the extra resistance of the cable.

While their analysis considers only lumped models, our analysis shows that this type of countermeasure is effective against our attack, requiring the temperatures to be varied according to

$$\frac{T_a}{T_b} = \frac{(1-\Gamma_b^2)(1-\alpha\Gamma_a^2)}{(1-\Gamma_a^2)(1-\alpha\Gamma_b^2)} \tag{4.44}$$

under our model. This allows our attack in its current form to be defeated if α can be accurately measured by the two parties. However, it remains for future work to determine if this can be implemented in a secure manner, as the measurement protocol for α remains unspecified.

4.6.6 DISCUSSION

The technique above exploits imperfections in the KKD implementation; while it might be theoretically possible to counter this attack by reduction of losses as proposed by Kish (2006d), the reduction of losses substantially below 0.1 dB ensures that this will be infeasible for all but the shortest or slowest of links.

This raises the question of why our attack should succeed where existing finite-resistance attacks have failed. The attack by Scheuer and Yariv (2006) considered only the variances of the measured variables. Our attack exploits the large correlation between waves in each direction; the estimator used above partially removes this common signal, increasing the ability to distinguish between the two cases statistically.

We have demonstrated an attack against the KKD key distribution system that exploits losses within the connecting transmission line. The attack has been shown experimentally to correctly determine more than 99.9 % of bits transmitted over a 2 m transmission line within 20 correlation times. As this attack requires that losses be reduced to a fraction of a decibel in order to maintain a meaningful level of security, modifications to the system, such as proposed by Kish and Granqvist (2014a), will be necessary in order to produce a secure link of any significant length and bitrate.

4.7 ATTACKING KKD WITH PROPAGATION SENSING

The fundamental objection of Bennett and Riedel (2013) is that an eavesdropper can measure the propagation of waves along the medium connecting the two endpoints, thereby collecting all of the information being transmitted between them. This has motivated us to develop an attack that makes fundamental use of the finite propagation speed within the transmission line.

We have developed a novel attack (Gunn et al. 2015b) on the KKD system that is built on finite propagation speeds in the transmission line, thereby demonstrating that an information leak related to signal propagation delay *does* exist.

4.7.1 QUANTIFICATION OF ATTACK EFFECTIVENESS

In order to quantify the effectiveness of the attack, we must choose a suitable figure of merit. Previous work has either failed to provide a measure or used bit-error-rates either directly or with the assumption of a binary symmetric channel (Hao 2006; Scheuer and Yariv 2006;

Gunn et al. 2014a); this latter approach, while providing a rough indication of the information available to Eve, does not provide a directly meaningful quantity. Another work (Kish and Granqvist 2014b), claiming to prove the unconditional security of the system, considers only asymptotic behaviour. We adopt a more general approach, taking account of the asymmetry of the channel and computing bounds on the secrecy rate for each given attack. This is particularly important for the attack that we introduce in Section 4.7.3, as its error rates are highly asymmetric.

4.7.1.1 Attack construction

As all the signals in the KKD system are zero-mean Gaussian, we describe the available measurement variables of the system using a multivariate Gaussian model, the covariance matrix conditioned upon the state of the two resistors, which may be swapped. We denote these two covariance matrices C_1 and C_2, the indices denoting whether Alice has chosen R_1 or R_2 respectively. The measurements in state i thus have a probability density function

$$f_i(\mathbf{x}) = (2\pi)^{-\frac{n}{2}} |C_i|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}^t C_i^{-1}\mathbf{x}\right], \tag{4.45}$$

where n is the number of measurement variables in the model. However, in many cases Bob and Eve make different measurements and thus see different covariance matrices $C_{i,b}$ and $C_{i,e}$, each containing a subset of the elements of C_i. We showed in Gunn et al. (2014a) that the Bayesian estimate for state S is given by the maximum-likelihood estimator,

$$\mathbf{x}^t \left(C_q^{-1} - C_p^{-1}\right) \mathbf{x} \underset{q}{\overset{p}{\lessgtr}} \log_e \frac{|C_p|}{|C_q|}, \tag{4.46}$$

for two arbitrary states p and q with corresponding covariance matrices C_p and C_q respectively.

However, a rigorous treatment of the system requires that we consider also the insecure states. In this case, we actually desire not the exact state of the system, but the resistance that was chosen by the sending party, since this is what will be used to determine the key bit. That is to say, if Alice is sending a message, the (R_1, R_1) state must be interpreted as a zero, since it lies within the $R_a = R_1$ row of Figure 4.7. Conversely, if Bob is the sender, a mistakenly-accepted (R_1, R_1) state will result in a one being used for the encryption, it falling within the same column as the true state.

Thus, while Alice and Bob—who need only distinguish between two states—can use the simple estimator above, Eve's maximum-likelihood estimator for the key bit used by Alice is

$$\begin{aligned}
&|C_{00}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}^t C_{00}^{-1}\mathbf{x}\right]\psi_{00}(\mathbf{x}) \\
+&|C_{10}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}^t C_{10}^{-1}\mathbf{x}\right]\psi_{10}(\mathbf{x}) \\
&\underset{R_2}{\overset{R_1}{\lessgtr}} \\
&|C_{11}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}^t C_{11}^{-1}\mathbf{x}\right]\psi_{11}(\mathbf{x}) \\
+&|C_{01}|^{-\frac{1}{2}} \exp\left[-\frac{1}{2}\mathbf{x}^t C_{01}^{-1}\mathbf{x}\right]\psi_{01}(\mathbf{x}),
\end{aligned} \tag{4.47}$$

where

$$\begin{aligned}
\psi_{ab}(\mathbf{x}) = \\
&u\left(\mathbf{x}\left(C_{ab}^{-1} - C_{a(1-b)}^{-1}\right)\mathbf{x} - \log_e \frac{|C_{ab}|}{|C_{a(1-b)}|}\right) \\
\times\, &u\left(\mathbf{x}\left(C_{ab}^{-1} - C_{(1-a)b}^{-1}\right)\mathbf{x} - \log_e \frac{|C_{ab}|}{|C_{a(1-b)}|}\right)
\end{aligned} \tag{4.48}$$

is the indicator function for the set of measurements $\mathbf{x}$ that results in the bit being kept. That is to say, if a bit is kept, the likelihood is zero for any state that results in it being dropped. With this estimator, we may now simulate the system as a whole, allowing us to estimate the secrecy rate of the system. If Bob is the sender, the terms involving C_{00} and C_{11} are be swapped.

4.7.1.2 Computation of secrecy bounds

In order to provide concrete numbers, we consider the secrecy rate (Wyner 1975; Maurer 1993) of the binary system formed by the application of this estimator to the variables $\mathbf{x}_b$ and $\mathbf{x}_e$ measured by Bob and Eve respectively. For this system, it can range from zero—where no security is available—to one—where a secret bit can be generated for every bit emitted by the system. This allows the security of an information-theoretic system to be evaluated independently of the available coding techniques, and in a fashion more directly applicable to the performance of the system. This is the first time that this has been evaluated for the KKD system, and so for reference we will apply the same technique to a number of previous attacks. In order to find this rate, the asymmetric error probabilities are computed by simulation, allowing mutual information and conditional mutual information to be estimated and thus

the evaluation of the bounds by Maurer (1993):

$$S(X;Y|Z) \leq \min\{I(X;Y), I(X;Y|Z)\} \tag{4.49}$$

$$S(X;Y|Z) \geq \max\{I(X;Y) - I(X;Z), \\ I(X;Y) - I(Y;Z)\}, \tag{4.50}$$

where X represents the variables available to Alice, Y those available to Bob, Z those available to Eve, and $I(X;Y|Z)$ the conditional mutual information of X and Y given Z. These results apply to arbitrary variables X, Y, and Z, not merely the wiretap channel. We denote $S(X;Y|Z)$ the secrecy rate of the channel with respect to an eavesdropper knowing Z. The error probabilities are calculated by generating, for each resistor configuration, an ensemble of random vectors of normal variables with the necessary covariance matrix and applying the state estimator being tested. Doing this simultaneously for Alice, Bob, and Eve provides us with the full joint probability distribution of X, Y, and Z, allowing computation of the secrecy rate bounds above from standard mutual information formulae. Increasing bit durations are modelled by adding additional independent steady-state samples; this enlarges the covariance matrix correspondingly.

It is important to note that in our analysis we use the binary variables X, Y, and Z produced by the bit estimation process. This is therefore not a canonical measure of security, but that of a hypothetical test setup. A bound on secrecy rate with respect to the raw measurements—as opposed to estimated resistor states—requires the consideration of more complex probability measures and is beyond the scope of this thesis, and therefore only the upper bound on secrecy rate is directly meaningful, as it remains a possibility that a more sophisticated eavesdropper might use the raw analog measurements to glean further information from the system, for example by propagating reliability estimates through the decoding stages.

4.7.1.3 System parameters

In order to provide a fair comparison, all of the attacks discussed will be considered with respect to the same system, described in Figure 4.16.

Resistors . We have chosen resistor values of 1 kΩ and 10 kΩ, following Gunn et al. (2014a) and Mingesz et al. (2008). This choice will affect the security of the system against all types of attacks—resistors further apart in value allow the use of shorter bit periods and

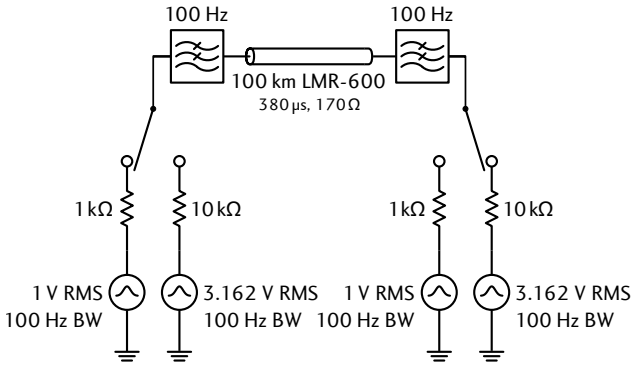


Figure 4.16: The KKD system under analysis, with component values included. We model a 100 km link constructed of low-loss LMR-600 (Times Microwave 2014) coaxial cable. This has a propagation velocity of $0.87c$, and thus a 380 µs electrical length and half-wavelength frequency of 1300 Hz.

so make the task of the eavesdropper more difficult when carrying out steady-state attacks, however this makes certain transient attacks more efficient; we will introduce such an attack in Section 4.7.3.

Transmission line . The line is chosen to be 100 km long. This falls into the middle of the range proposed by Mingesz et al. (2008), from chip-scale at the low end to 2000 km at the high end. This length is selected in order to achieve a cable resistance in line with the 200 Ω value considered by Mingesz et al. (2008). The cable itself is low-loss LMR-600 (Times Microwave 2014), with a propagation velocity of $0.87c$ and a core resistance of $1.7\ \Omega\,\mathrm{km}^{-1}$.

System bandwidth . The propagation time of the line is 380 µs, and therefore has a half-wavelength frequency of 1300 Hz. We follow the recommendation of Kish (2006c) and limit the bandwidth to somewhat less than a tenth of this; we therefore use noise sources and line filters with a bandwidth of 100 Hz. The sources and filters are assumed to be perfect; that is, their frequency spectra and frequency responses respectively are rectangular.

Noise sources . The noise sources themselves are assumed to be exactly Gaussian, with a linear ramp profile as used in Mingesz et al. (2008). The ramp lasts for 8% of the bit duration at both the beginning and the end of the cycle. The magnitudes of the voltages are chosen so that the 1 kΩ resistor has in series a noise voltage of 1 V RMS. This corresponds to a noise temperature of 1.8×10^{17} K.

4.7.2 NONIDEALITIES IN THE LUMPED MODEL

We begin by analysing the simple lumped model shown in Figure 4.16, modelling the transmission line as a resistor R_L. Let us denote the voltage sources of Alice and Bob $V_a(t)$ and $V_b(t)$ respectively, the voltage at Alice's end of the line $V_x(t)$, and the current through the line $I(t)$. Here, $V_x(t)$ and $I(t)$ are the measurement variables of the system, and are given by

$$\mathbf{x}(t) = \begin{bmatrix} V_x(t) \\ I(t) \end{bmatrix} \tag{4.51}$$

$$= \frac{1}{R_a + R_b + R_L} \begin{bmatrix} R_b + R_L & R_a \\ 1 & -1 \end{bmatrix} \begin{bmatrix} V_a(t) \\ V_b(t) \end{bmatrix} \tag{4.52}$$

$$= A\mathbf{V}(t). \tag{4.53}$$

From this we may compute the measurement covariance matrices $C_x = AC_vA^t$, where C_v is the covariance matrix of the noise sources of Alice and Bob and given by

$$C_v = 4kT_{\text{eff}}B \begin{bmatrix} R_a & 0 \\ 0 & R_b \end{bmatrix}, \tag{4.54}$$

where T_{eff} is the noise equivalent temperature of the system.

4.7.2.1 *Resistance errors*

The first nonideality that we consider is due to errors in the resistor values. These can be caused by manufacturing variations, but also by the resistance of the line itself—this can be interpreted as a known constant added to the resistors. With high-precision resistors available at low cost with tolerances less than 0.1%, it is the latter form of error that dominates, and so we focus our analysis there. By simulating the system in Equation 4.53, we compute the secrecy rate of such a system, shown in Figure 4.17.

A characteristic shape is visible—at first the secrecy rate increases with bit duration, before peaking and slowly falling away. With very few samples, the error rate between Alice and Bob is so high as to render communication almost impossible; the secrecy rate is therefore very low in this regime. As the number of samples increases, Alice and Bob, whose state classification problem is very simple, quickly reduce their error rate. However, Eve's error rate falls in a similar way, albeit more slowly due to her relative lack of per-sample information, and eventually the additional information that Alice and Bob can squeeze from each sample falls below that which Eve can extract, resulting in a peak such as in Figures 4.17 and 4.18. The secrecy rate slowly approaches zero as the number of samples increase and the four states therefore become increasingly difficult to confuse.

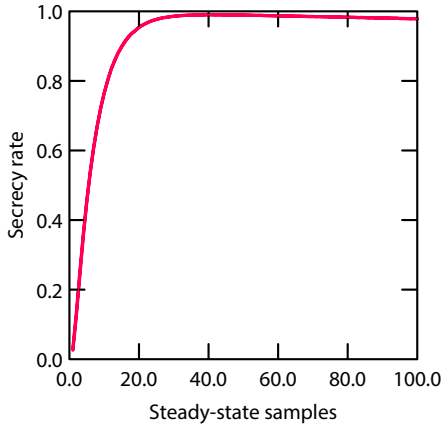


Figure 4.17: Secrecy rate as a function of steady-state averaging time in terms of equivalent independent samples, with 10^6 simulated bits per point. Upper and lower bounds are shown, though are not visible without magnification. Alice, Bob, and Eve make use of both voltage and current measurements. Note that the secrecy rate steadily increases as Alice and Bob reduce their error rate, eventually peaking as it approaches zero and so can no longer improve relative to Eve's performance.

We hasten to add that a countermeasure (Kish and Granqvist 2014a) is available that eliminates information leakage due to the line resistance. Briefly, when the system is in one of the two secure states, the line can be viewed as being part of the small resistor; by adjusting their mean-square voltages from $4kT_{\text{eff}}BR_1$ to $4kT_{\text{eff}}B(R_1 + R_L)$ the system becomes secure once more.

4.7.2.2 *Temperature errors*

A related phenomenon is temperature error in the two terminals; if the voltages are not correctly calibrated, the apparent temperatures of Alice's and Bob's resistors will differ, resulting in a net flow of power through the line (Hao 2006). This power flow manifests itself as a correlation between voltage and current, the sign depending upon its direction.

The effect is shown in Figure 4.18; we see that the effect is relatively small even with a pessimistic voltage error of 2%. In practice, one can regularly calibrate the noise temperatures, reducing the leak to a completely negligible level as seen experimentally by Mingesz et al. (2008).

4.7.3 TRANSIENT ATTACKS

Previously, we considered (Gunn et al. 2014a) the use of directional measurements of the wave components travelling in each direction along the line; this is frustrated by the band-limited

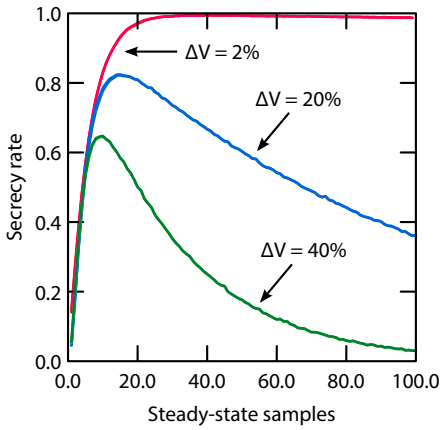


Figure 4.18: Secrecy rate of the ideal KKD system with various voltage mismatches and zero line resistance, with 10^5 simulated bits per point. We see that, even with a large mismatch of 2%, the effect on secrecy rate is slight; with calibration it will be almost negligible. Gross mismatch is necessary in order to substantially reduce the peak secrecy rate.

nature of the signals and the large reflection coefficients of typical endpoint designs. It was found that this model, while effective in the case of a resistive line, was unable to differentiate the zero- and one-states in the absence of propagation delays, though the general applicability of the attack that we proposed (Gunn et al. 2014a) remains somewhat contentious (Chen et al. 2014b; Chen et al. 2014a; Kish et al. 2015). This result has motivated us to consider the effect of propagation delays, with the goal of reconciling the non-constructive information-theoretic claims of Bennett and Riedel (2013)—which state that this type of system is inherently insecure—with the far less dire results found by analysis in the quasi-static limit.

4.7.4 PROPAGATION DELAYS AND TEMPERATURE MISMATCH

Irrespective of the veracity of the claims of Chen et al. (2014a) and Kish et al. (2013), the signals injected onto the line by the endpoints must propagate at some finite sub-c speed; there must therefore be, even with perfect synchronisation, some finite period during which each point along the line experiences only a signal due to the closest endpoint.

We demonstrate this phenomenon in Figure 4.19. At time $t = 0$, the noise temperature of each endpoint begins to rise. However, this rise is invisible to the majority of the line—the increasing potential of the fluctuation is retarded, to use the terminology of Chen et al. (2014a). In the middle of the line, the signals are retarded by equal amounts, and thus the apparent temperature profiles remain constant. Away from the centre of the line, however, the retardation times differ, resulting in an apparent temperature mismatch. This temperature

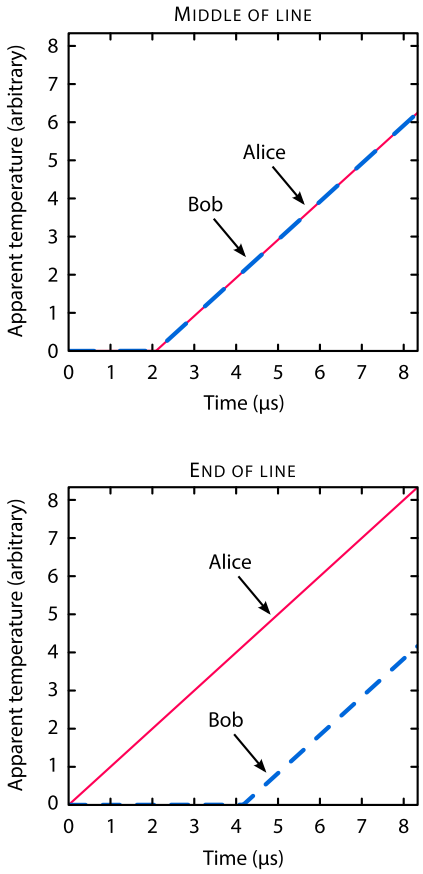


Figure 4.19: The effect of propagation on apparent noise temperatures with a linear profile. Parameters chosen are $L = 1\,\mathrm{km}$, $\nu = 2 \times 10^8\,\mathrm{m\,s^{-1}}$, $t_r = 1\,\mathrm{ms}$. In the top graph, the apparent temperatures are shown from the perspective of a point equidistant between the two endpoints. As the signals from both endpoints are equally retarded, the apparent temperatures are equal. The bottom graph shows the apparent temperatures at one end of the line; one endpoint suffers no retardation, while the other experiences that by the full length of the line. This results in a temperature imbalance during the ramp-up time.

mismatch allows a Hao-type attack (Hao 2006) to be performed without relying on errors of calibration. We note that the temperatures involved here are of the sources and not of the transients themselves.

Let L be the length and ν the speed of propagation of the line. We first consider a linear temperature profile, ramping from 0 to 1 in time t_r. The temperature of each source is at time t given by

$$T(t) = r(t/t_r) - r(t/t_r - 1), \tag{4.55}$$

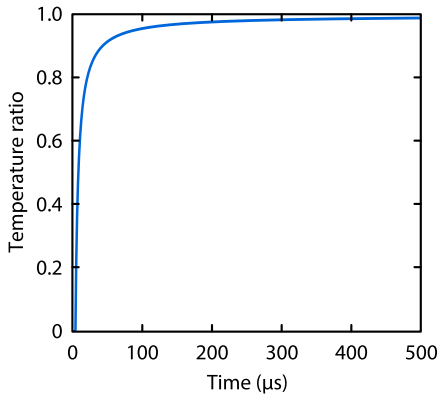


Figure 4.20: The ratio of apparent temperatures at one end, using a linear temperature profile. Parameters are identical to those described in Figure 4.19. With careful examination, the ratio is seen to remain zero some time after $t = 0$.

where r denotes the unit ramp function. At a distance x from Alice, the apparent temperatures are time-retarded and so given by

$$T_a(t) = T(t - x/v) \tag{4.56}$$

$$T_b(t) = T(t - (L - x)/v), \tag{4.57}$$

resulting in an apparent temperature ratio of

$$\frac{T_a(t)}{T_b(t)} = \frac{r\left(\frac{t-x/v}{t_r}\right) - r\left(\frac{t-x/v}{t_r} - 1\right)}{r\left(\frac{t-(L-x)/v}{t_r}\right) - r\left(\frac{t-(L-x)/v}{t_r} - 1\right)}. \tag{4.58}$$

Supposing without loss of generality that $x > L/2$,

$$\frac{T_a(t)}{T_b(t)} = \begin{cases} 0, & (L-x)/v < t \leq x/v \\ \dfrac{t - x/v}{t - (L-x)/v}, & x/v < t \leq t_r + (L-x)/v \\ \dfrac{t - x/v}{t_r}, & t_r + (L-x)/v < t \leq t_r + x/v \\ 1, & t > t_r + x/v, \end{cases}$$

shown in Figure 4.20.

4.7.5 LEAK ANALYSIS

The non-ergodic nature of the modulated noise process prevents the Hao attack from being used directly over the entire transient time; a full characterisation of the resulting information leak is therefore beyond the scope of this thesis. Instead, we restrict ourselves to the time period $x/v < t \leq T + (L - x)/v$, during which the signal produced by only one endpoint

Table 4.1: Values of $1 - \Gamma^2$ for various choices of resistor. A characteristic impedance of $50\,\Omega$ is assumed.

R	$1 - \Gamma^2$
$1\,\mathrm{k}\Omega$	0.1814
$10\,\mathrm{k}\Omega$	0.0198
$100\,\mathrm{k}\Omega$	0.0020

is apparent. Because the correlation time of the system is required (Kish 2006c) to be substantially longer than the propagation time of the line, these measurements will contain little information beyond that of a single sample. The sample is distributed

$$X \sim \mathcal{N}(0, k\sigma_{\text{in}}^2), \tag{4.59}$$

where k is some constant that depends upon the choice of filter and temperature profile. We know (Gunn et al. 2014a) that

$$\sigma_{\text{in}}^2 = \frac{1}{2} kTBZ_0 \left(1 - \Gamma^2\right), \tag{4.60}$$

where Γ is the reflection coefficient of the chosen resistance, and therefore this single sample provides us with enough information to estimate the choice of resistor. Values of $1 - \Gamma^2$ for various choices of resistor are shown in Table 4.1.

It is therefore clear that substantially different resistors will result in substantially different variances, resulting in an information leak.

We plot the error rates for the estimation of a single resistor in Figure 4.21, which in the single-variable case can be calculated using (4.46) as

$$E_{1\to 2} = \quad F_{\chi^2}\left(\frac{\log\gamma}{\gamma - 1}\right) \tag{4.61}$$

$$E_{2\to 1} = 1 - F_{\chi^2}\left(\frac{\log\gamma}{1 - \gamma^{-1}}\right) \tag{4.62}$$

where

$$\gamma = \frac{1 - \Gamma_1^2}{1 - \Gamma_2^2}. \tag{4.63}$$

We see that the error rate of the eavesdropper falls towards zero as the difference in resistance increases. With currently-favoured component values—on the order of $R_1 = 1\,\mathrm{k}\Omega$ and $R_2 = 10\,\mathrm{k}\Omega$—the error rate of the eavesdropper is approximately 25%. She can further reduce her error rate by making a similar measurement at the other end of the line, however doing

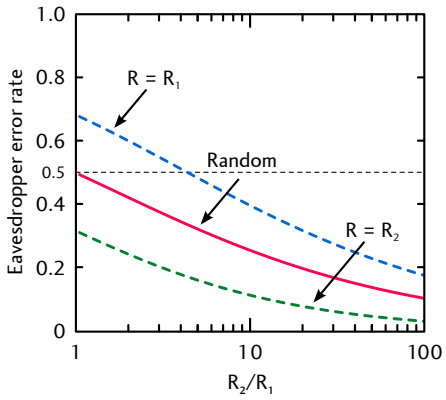


Figure 4.21: Resistor-estimation error rates for an eavesdropper using the attack discussed with $R_1 = 1\text{k}\Omega$. We show error rates for R_1 always chosen, R_2 always chosen, and for the resistor being chosen at random. Interestingly, the error rates are not symmetric with respect to the resistor actually chosen. The overall bit-error-rate approaches 0.5 as $R_2 \to R_1$; large gains in security are therefore possible if R_2 and R_1 are chosen to be similar. Note that these error rates are for estimation of a single resistor; by performing the attack at two points on the line, an eavesdropper may further reduce her bit error rate.

so requires a multivariate estimator and so resists the calculation of error rates analytically by straightforward means.

We now proceed to calculate the secrecy rate. As the signals emitted by Alice and Bob are independent, the measurement covariance matrix is diagonal, with the two entries given by (4.60). The effect of the attack upon the secrecy rate of the system is shown in Figure 4.22; the maximum secrecy rate is reduced by approximately one-third, and therefore this attack, if realised, has a significant effect upon security.

4.7.6 Countermeasures to the transient attack

As noted previously, the error rate of the eavesdropper is substantial, even with current designs. It is therefore feasible to simply increase the level of privacy amplification. However, this comes at the cost of key rate, and it is therefore desirable to tackle the problem more directly.

It is proposed by Kish (2013) that the resistor values themselves vary during the equilibriation period, allowing the line to reach an approximate thermal equilibrium, however all implementations to date (Mingesz et al. 2008; Gunn et al. 2014a) have used fixed resistors. A time-varying resistance can be used to thwart our proposed attack by filling the line with noise before allowing the resistors to differ, thereby removing the period in which each

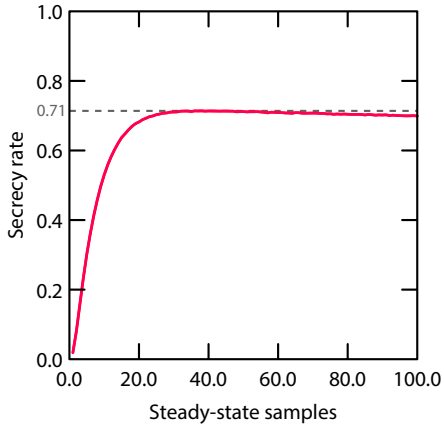


Figure 4.22: The secrecy rate of the system in the presence of an eavesdropper performing the described transient attack against both endpoints. We see that the plot is qualitatively similar to that of Figure 4.17, but reaching a maximum of only 0.71. Defending against this attack therefore requires substantial changes to the design parameters of the privacy amplification subsystem. We note that the maximum secrecy rate depends upon the resistor values; the resemblance to $\sqrt{2}$ is therefore coincidental.

resistor's final value can be probed separately. A similar effect can be achieved by modifying the temperature profile according to the choice of resistor such that the injected signal $\frac{1}{2}kTBZ_0(1-\Gamma^2)$ is initially identical, irrespective of the resistance chosen. Combining these two approaches has the potential to further complicate further attempts at attack, at the very least in a practical sense. Eavesdropping on such systems will require a more general attack than that which we have proposed, taking advantage of the smaller imbalance that persists throughout the lengthy period of equilibriation.

4.8 Remarks on the proposed KKD proof of security

A notable feature of the literature around the KKD system is the abundance of papers claiming it to be unconditionally secure. The most recent of these is by Kish and Granqvist (2014b), however the explicit claim that KKD is unconditionally secure dates back at least to Kish (2006d); indeed, the original paper describes it as "totally secure" (Kish 2006c).

We focus our attention here on the paper by Kish and Granqvist (2014b), as this is the only work in the literature that specifically aims to prove the security of KKD in the presence of nonidealities. Unfortunately, its proof is erroneous; the assumptions made are overly severe and thus it fails to provide a meaningful demonstration of security. Indeed, the actual proof of security is made without any reference to the physical system, and thus is equivalent to stating that any classical system, which can provide secure communication in an idealised

setting, will provide unconditional security when subject to the nonidealities of the real world.

Briefly, the argument is as follows. Let $Q = (x_1, x_2, ...)$ be a parametrisation of the system in such a way that they are all equal to zero for the mathematically-ideal system (Kish 2006c), which can be demonstrated to be unconditionally secure. Now suppose that the two legitimate parties have access to the same measurements as Eve, and define a score δ representing the clarity that these measurements provide; they reject all bits with δ greater than some threshold ω, thus reducing the effect of outliers.

We define a function $p_\delta(Q)$ representing the probability that an eavesdropper will correctly determine the bit. Perfect secrecy is achieved if $p_\delta(Q) = 0.5$, and unconditional security if—but not only if—$p_\delta(Q) < 1 - p_e$, where p_e is the error rate of the two legitimate parties. We note that the definition of unconditional security that we have used in this thesis, based on the secrecy rate, is a quantitative version of to the more common one by Diffie and Hellman (1976), and differs from that used by Kish and Granqvist (2014b), where it is erroneously defined to be the closest practical approximation to perfect secrecy. As Q was defined such that it was equal to zero in the idealised scenario, which achieves perfect secrecy, $p_\delta(\mathbf{0}) = 0.5$.

The paper then claims that as linear and stable nonlinear systems have variables described by continuous functions, the function $p_\delta(Q)$ must be continuous in $\{x_i\}$, and that therefore $p_\delta(Q)$ can be made arbitrarily close to 0.5 by setting the $\{x_i\}$ arbitrarily small, thereby demonstrating the system to be unconditionally secure.

However, these assumptions range from being overly strict to completely unjustified, as explained in the following.

4.8.1 PARAMETRISATION OF THE DESIGN

The assumption that one can completely parametrise the system in advance is a very strict condition and not at all practical. There will inevitably be unmodelled effects due to environmental conditions, tampering by the eavesdropper, or simply unintentional omission by the designer of the system. Any claim that a system is secure based on such an assumption must by accompanied by incontrovertible proof that the parameters have been completely enumerated, in order that they can not only be controlled in practice, but in order to ensure that the other assumptions are indeed valid—neither of which have been attempted by Kish and Granqvist (2014b).

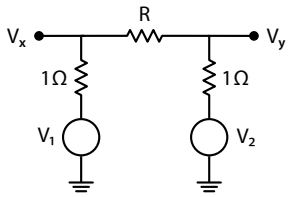


Figure 4.23: A resistive circuit containing two secret DC voltage sources V_1 and V_2, each with a series resistance of $1\,\Omega$. An eavesdropper can measure the voltage at two points on the line, yielding voltages V_x and V_y, which determine V_1 and V_2 if and only if $R \neq 0$.

It is further assumed that these parameters can all be varied towards their ideal values, something that is not true in practice. One example provided by Kish and Granqvist (2014b) of such a parameter is the cable length, however this is manifestly invariant—it cannot be made arbitrarily small, as it must be sufficiently long to connect the two endpoints. This immediately disqualifies the proof from application to any practical system, and also to the attack presented in Section 4.7.3, which relies upon the nonzero length of the transmission line.

4.8.2 Continuity argument

In addition, it is also claimed that linear systems and stable nonlinear systems produce variables that are continuous-valued, and in particular that this applies to the probability of error. However this is emphatically not true, which we demonstrate using the DC resistive circuit in Figure 4.23 as a counterexample.

Figure 4.23 shows a KKD-like system that operates at DC. The two voltage sources are given a randomly-determined voltage—we assume some continuous distribution such as the Gaussian distribution in which all elements of the support are chosen with probability zero[7]—and the eavesdropper is restricted to measuring the voltage at two points V_x and V_y. These voltages are given by

$$V_x = V_1 - \frac{V_1 - V_2}{1 + 1 + R} \tag{4.64}$$

$$V_y = V_2 - \frac{V_2 - V_1}{1 + 1 + R} \tag{4.65}$$

[7] That is to say, though some value must be chosen, any individual value is chosen almost never—that is to say, with probability zero.

or, more conveniently, in matrix form:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \frac{1}{R+2} \begin{bmatrix} R+1 & 1 \\ 1 & R+1 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix}. \tag{4.66}$$

This system is exactly soluble provided $R > 0$. However, when $R = 0$ the matrix is no longer invertible, and therefore Eve's estimate will be wrong almost surely, resulting in a probability of error equal to

$$p_e = \begin{cases} 1, & R = 0, \\ 0, & R > 0 \end{cases}. \tag{4.67}$$

This function is not continuous, and thus the statement is disproven by contradiction.

Though this circuit is purely theoretical—superconductors aside, an ideal short circuit does not exist in reality—it serves as an interesting counterexample to the claims of Kish and Granqvist (2014b), where the continuity argument is claimed to apply to any continuous system. Though it might be argued that this circuit is artificial, and that a real system will have imperfections and thus not behave in such a manner, the same can be said of the idealised KKD system. Given that this supposed theorem has been neither proven nor even clearly stated, we therefore content ourselves with the presentation of a counterexample to the argument as stated.

Knowing that this type of behaviour is possible, it is therefore necessary to demonstrate that this continuity property exists on a case by case basis after having found a set of design parameters that can be made to approach their ideal while still representing a viable system. This is not carried out by Kish and Granqvist (2014b), which simply assumes it to be so, rendering its argument invalid.

This is not to say that no classical system can obtain a level of security; merely that this proof technique does not provide sufficient justification for the claim of unconditional security.

4.9 CONCLUSION

The Kish key distribution system has been proposed as an alternative to quantum key distribution, with claims of equivalent or greater security. We have demonstrated that, despite these claims and supposed proofs of security, the system is vulnerable to a number of attacks. Various countermeasures have been proposed to overcome particular attacks but, despite this,

the KKD community has yet to produce a convincing argument that the proof of insecurity by Bennett and Riedel (2013) is invalid or inapplicable to KKD. In the absence of such a breakthrough, there is current little reason to believe that Bennett and Riedel (2013) are incorrect in their analysis, and that therefore KKD does not live up to its security claims, and will fall to a sufficiently powerful adversary as we showed would happen to our own system from Section 4.4.

The vulnerability of the KKD system makes clear the difficulty inherent in developing security systems that are dependent upon physical properties. Physical systems have a far greater capacity for unmodelled behaviour than purely mathematical ones, an unfortunate reality not only for KKD, but QKD as well (Lydersen et al. 2010). Though it remains to be seen whether real physical cryptographic systems can be modelled with the a precision approaching purely mathematical ones, given their sheer complexity and the need to cross many domains, from optics to solid-state physics to electronics to software, the outlook is not hopeful.

4.9.1 ORIGINAL CONTRIBUTIONS

In this chapter, we have described a number of contributions to the state of the art:

- We have examined the use of routing delays as a source of randomness for information-theoretically secure key establishment over the internet, and demonstrated its impossibility on information-theoretic grounds (Gunn et al. 2014c).
- We have experimentally demonstrated the directional measurement of waves in a very short transmission line, experimentally refuting the claims of Kish et al. (2013). We constructed a KKD system, and used this directional coupler to attack the system, in the process demonstrating a state estimator for the system that dramatically outperforms the naïve estimator that has thus far been used to determine the security implications of hardware nonidealities (Gunn et al. 2014a).
- We have shown that the transient behaviour of the KKD system allows a relatively simple attack that requires only two single-time-point measurements (Gunn et al. 2015b).
- We have rebutted the proof of security for the KKD system presented by Kish and Granqvist (2014b), showing that several of its arguments are erroneous and providing a counter-example (Gunn et al. 2015b).

Despite our negative view of the security of the KKD system, we do not reject the value of stochastic systems in information security. As we shall see in Chapter 5, more conventional systems can benefit from randomness in their operation. We show several methods by which

this can be achieved, demonstrating that by avoiding the need for physical modelling, we can produce novel systems with security proofs far more convincing than those that are relative to the accuracy of a physical model.

Chapter 5

Trustworthy Randomness for Identity Management

In the previous chapter we have shown how noise is not a security panacea: information-theoretic limits on the ability to distill a secret key from noise prevent information-theoretic security from being achieved using only endpoint-generated noise.

This does not mean that randomness is not useful for security—clearly it is necessary for keys to be random, and random nonces are often vital. We turn our attention to systems that derive security from the unpredictability of random values, and propose several schemes allowing decentralised identity management with statistical guarantees of the probability that an attacker can successfully deceive.

Digital identity management has proven to be a difficult problem, the electronic nature of the problem being both a help and a hindrance. The strength of modern electronic signature algorithms is such that forgery is essentially impossible; the security of credentials is therefore entirely determined by the issuance and key-management practices.

The problem with most existing approaches is the dependence upon trusted third parties (Ferguson et al. 2010, §19.3). These can issue—and have issued[8]—misleading certificates as a result of system compromises, validation errors, or outright corruption.

We propose the inclusion of a random element within the verification process, randomising the measurement conditions so that systematic failures are detected via repeated attempts. There are a number of approaches, several of which we have evaluated and prototyped, and these are the topic of the remainder of this thesis.

5.1 PUBLIC-KEY DISTRIBUTION: THE *STATUS QUO*

The most widely-accepted systems that are configured by end-users, such as SSH (Ylonen and Lonvick RFC 4251, 2006) and WhatsApp (*WhatsApp encryption overview: technical white paper* 2016), tend to use a trust-on-first-use (Wendlandt et al. 2008) model, in which initial communication takes place with either no or only manual authentication, after which the user is alerted to key changes. However, this does not prove the identity of the user unless the two parties use some out-of-band authentication method.

5.1.1 THE PUBLIC-KEY INFRASTRUCTURE

The need for out-of-band verification can be overcome by standards such as X.509 (Cooper et al. RFC 5280, 2008), which use signed certificates to verify identities: X.509 is widely adopted by email clients, but the need to acquire certificates from a commercial certificate authority has prevented it from seeing any significant use. This and related standards are used to create what is known as the public-key infrastructure (PKI).

The PKI is hierarchical in nature; a user-agent, such as a browser, will include a list of trusted certificate authorities. The server sends back a certificate identifying the owner of its public key, then a certificate certifying the issuer of the first certificate, and so on; the client accepts the certificate chain if it includes a certificate from one of its trusted authorities.

[8] See `http://wiki.cacert.org/Risk/History` for a list of known incidents.

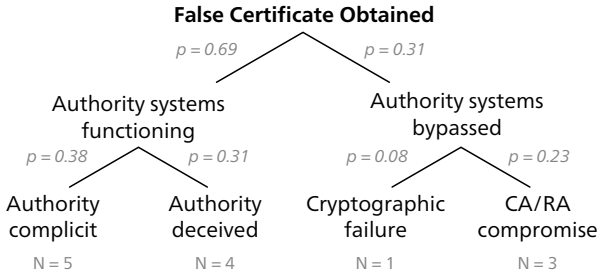


Figure 5.1: Failure modes for a certificate or registration authority. The cases are split according to whether or not the authority's systems were bypassed in producing the certificate, whether, if not, the operators knowingly issued a false certificate, and whether, if the authority systems were bypassed, it was because the attacker gained control of the issuance systems or because of a cryptographic failure. The observed probabilities of various failure modes are estimated by categorising thirteen known incidents that resulted in attackers fraudulently obtaining a certificate. We denote the observed probability of an event p, and the number of occurrences of an event N.

This is simple and easily-understood, however most certificate authorities can issue certificates for anyone in the world, an ability that has been abused on a number occasions both by attackers and the operators of the authorities.

5.1.2 PKI FAILURE MODES

We consider our proposals in the context of the existing PKI system; here there are a moderately large number of registration and certificate authorities, which are in general competently run. For the purposes of our analysis, we have placed each of the failures from the list of known PKI failures reported by CAcert (2017) into one four categories: intentional misissuance, deception, authority compromise, and cryptographic failure; see Figure 5.1. This list covers the period from 1995 until the present day, with the most recent event still ongoing. Here we define a PKI failure to be an event that results in an attacker obtaining a certificate for a key that is not under the control of the stated subject of the certificate. Our classifications are shown in the below, but we note here that our definition of a PKI failure excludes cases where attackers have obtained the private key attached to a legitimately issued certificate, either by theft or cryptanalysis.

We take each of the events recorded in CAcert (2017) and determine whether a false certificate was obtained by someone other than the entity referenced in the certificate. This therefore excludes the theft of private keys of leaf certificates.

Duplicates were then removed; exactly what is considered a duplicate is a somewhat arbitrary choice, however this only affected one vulnerability—the MD5 chosen-prefix attack, in which we considered the academic attack by Stevens et al. (2009) and the similar approach used by the Flame malware to be the same vulnerability.

We then determined whether the false certificate was obtained through the issuance systems of the certificate and registration authority, or whether they had been bypassed. If the issuance system was used, we then determined whether the misissuance was intentional or the result of deception. If the issuance system was bypassed, we then determined whether or not the private key of the certificate authority was used; the former case is by definition a cryptographic failure, and the latter a compromise of the authority.

The overall results are shown in Figure 5.1. The individual events are listed here, along with the identifier given to the event by CAcert (2017) and a brief description of the nature of the failure.

1. `Microsoft`: *2001, deception.* Unknown persons masqueraded as Microsoft employees in order to obtain certificates in the name of Microsoft.
2. `Flame`: *2007, cryptanalysis.* Intelligence agencies produced a false code-signing certificate by means of an MD5 chosen-prefix attack.
3. `RA-breach`: *2008, deception.* A registration authority was found not to perform ownership validation of certificate requests.
4. `ichsunx2-RA`: *2011, compromise.* Several registration authorities were compromised, resulting in the issue of several certificates.
5. `digiNotar`: *2011, compromise.* A certificate authority was thoroughly compromised, resulting in the issue of hundreds of false certificates without any audit trail.
6. `CA-MITMs`: *2012, intentional.* A certificate authority admitted to providing an intermediate certificate to a company for the purpose of performing man-in-the-middle attacks against its employees.
7. `accidental subroots`: *2012, intentional.* A certificate authority accidentally issued intermediate certificates to several customers that were then used to perform man-in-the-middle attacks.
8. `Signed Trojans`: *2013, deception.* Malware authors successfully obtained code-signing certificates in the names of nonexistent companies.
9. `ANSSI`: *2013, intentional.* A certificate authority issued an intermediate certificate to a government department, who then used it to perform man-in-the-middle attacks against internal traffic.

10. `India`: *2014, compromise.* Attackers compromised a certificate authority, issuing several false certificates.
11. `CNNIC`: *2015, intentional.* A certificate authority issued an intermediate certificate to a company that then used it perform man-in-the-middle attacks against its employees.
12. `Accidental Issuance`: *2015, intentional.* A certificate authority issued a number of certificates for domains not under its control during automated testing.
13. `Subdomains`: *2015, deception.* The website of a certificate authority performed no or insufficient validation of domain ownership.

We see that in most cases—69%, $N = 9$—the systems of the certificate authority functioned as designed. On just over half of these occasions, and in 38% ($N = 5$) of failures overall, these fraudulent certificates were knowingly issued by the authority in question. Another 31% ($N = 4$) of the time, the authority is deceived because they failed to accurately validate the identity of the attacker requesting the certificate.

Less commonly—31% ($N = 4$) of the time—the authority's controls are bypassed entirely. This may be due to a cryptographic failure, which occurred 8% ($N = 1$) of the time in our sample, but more commonly the controls are bypassed by a compromise of either the certificate or registration authority, this occurring 23% ($N = 3$) of the time.

These figures will be used to inform our attack model, and to estimate the effectiveness of our proposed technique in practice.

5.1.3 THE WEB OF TRUST

The PGP (Callas et al. RFC 4880, 2007) messaging format aims to provide message-level security to the masses, but has been hampered by the difficulty of its key management, which depends upon personal contact to establish trust relationships.

In contrast to the purely hierarchical structure used by the PKI, the web of trust makes certification structure completely free-form. Users use their public key to issue certifications of the identity of the holder of another public key. The user thus-certified attaches these certifications to their own public key, which is normally sent to a public keyserver.

A user can be certified in a number of ways; most common by far is to certify a user's identity—that they are who they say they are. However, certifications can also attest to a person's *trustworthiness and soundness of judgement.* This means that we can issue a certificate stating that we trust this person to issue only correct certifications, or even that we trust this

person to certify a person as a trustworthy verifier. This is rarely used in practice, and thus users are left to determine themselves whose verifications are credible.

Unfortunately, despite the highly distributed and democratic nature of the web of trust, it is not widely used; the need for active user involvement is ultimately too great a barrier to adoption.

5.1.4 Identity-based cryptography

Identity-based cryptography provides another approach, in which trust in a key needs only be established at the organisational level rather than between individual users, but this allows access to private keys by service providers; the ubiquity of adversaries with coercive powers and an interest in mass surveillance means that this is entirely inadequate from a privacy standpoint. While the risk might be mitigated with the aid of threshold cryptosystems or other distributed approaches, if the desire to decrypt a user's communications exists at an organisational level then threshold decryption and secret sharing provides little protection to users.

Our desire is to move in the other direction; rather than increasing the amount of centralisation involved, we wish to *decrease* it, allowing users to take responsibility for their own security to the greatest extent possible. We show in the next section that other internet infrastructure can be used for the purpose of secure key distribution in a manner that is far more easily audited than is the case with the current PKI.

5.2 Anonymous Auditing

Suppose you want to call someone, but do not know their phone number. How do you find it? The obvious way is to look them up in a phone book, but the phone company might have placed a different address under their name. If they are particularly security conscious, then you might presume that, when they received their phone book, they checked to see whether their number is correctly listed. But what if it were not modified in every phone book, but some contain the real number and some contain a false one?

The authentication of database entries and user attributes is an important problem in information security; one of the most prominent applications is in key-distribution for end-to-end secure messaging. Some systems use centralized key-distribution services, placing trust in the operators of their servers. Others use decentralized methods, but existing methods come with their own limitations; the public-key infrastructure allows most certificate authorities to

impersonate anyone, and mainstream blockchain systems waste power calculating proofs of work. The result is that even when a database can be realistically distributed, the designers of many systems choose not to do so.

This task is greatly simplified if we can decentralize a system in a generic way, adding standard components that can be reused for many systems. We show in Sections 5.2.3 and 5.2.4 that an anonymisation system serves this purpose, in many cases without modification or even the cooperation of the central server.

Our contribution in this paper is to show that a variation of the multi-path probing (Wendlandt et al. 2008) approach used by *DetecTor* (Engert 2013) is provably secure. Users simultaneously make identical requests to a central service via an anonymiser. If they receive consistent responses, then they can assure themselves that the server provides identical responses to identical requests; we show in Section 5.2.4.4 that a server can successfully equivocate across N users for M rounds with probability at most N^{1-M}.

This approach has a number of advantages over other anti-equivocation techniques in the literature:

- **No bootstrapping problem.** By using an existing anonymity system to audit quite general services, new systems can obtain the benefits of distributed auditing without an existing community to provide operator-diverse monitoring systems.
- **Scalability.** Users do not need to communicate with each other, except to signal that the service has misbehaved. As a result, the communication overhead is only $O(\log \epsilon)$ for a given security level ϵ.
- **Computational efficiency.** Because we do not use a proof-of-work system, no computational power is wasted on what is generally pointless busywork whose only purpose is to make participation costly.
  This is relevant to our first point: a new proof-of-work system is not secure until it has enough miners to out-compute any potential attacker. This creates a chicken-and-egg problem, in that the system is not secure until it is widely adopted, which will not happen if it is insecure.
- **No server-side cooperation needed.** This approach does not require any changes on the server-side; as a result, it is quite practical for motivated users to audit existing services without the need for effort or cooperation from their operators.

The first of these points is particularly important, as many pieces of software begin as a small-scale project by individuals or groups without third-party commitment. Our protocol

provides a distributed auditing capability that has until now been completely unavailable to such projects.

5.2.1 MOTIVATION

Our principal motivation for the development of this auditing method is to allow the use of centralized key-distribution servers in a secure manner. Key distribution is a difficult problem to solve, and as it stands there are few solutions that do not centralize trusted operations to a significant degree, requiring manual verification on the part of users in order to eliminate the risks posed by malicious infrastructure operators.

The need for manual effort is problematic in multiple ways; the first is that most users will simply not bother, but even amongst those users who do make the effort, they will not necessarily wait for the verification to take place before communicating. This leaves a window of vulnerability before an attack is detected, which in the case of manual in-person verification may be very long indeed.

Our desire, then, is to allow users to take responsibility for the security of their own identity to the greatest extent possible, but in a way that does not require a significant degree of manual effort.

5.2.2 RELATED WORK

The problem of obtaining agreement on a value amongst several—possibly malicious—users is an old one, known as the Byzantine Generals problem, and was first analyzed by Lamport *et al.* in 1982 (Lamport et al. 1982). Several officers plan for an attack, in which they must act simultaneously in order to be successful. This is complicated by the knowledge that some of the officers may be traitors—including the general in command of all of them—and may therefore send different messages to different units in an effort to induce a doomed attack.

Consensus protocols have seen increasing prominence in recent years with the rise of cryptocurrencies such as Bitcoin (Nakamoto 2008), whose security against double-spending depends upon public scrutiny of the submitted transactions. If consensus on the transaction ledger is broken—that is to say, if different users see different values—different transaction records can be sent to different users, allowing double-spending to occur.

Traditional consensus protocols. These (Lynch 1996) are effective, but typically do not scale well to large numbers of participants (Vukolić 2016). Significantly, they require communication

channels between many of the nodes taking part in the consensus protocol, with resistance to traitors being limited by the connectivity of the network graph. This is inconvenient in practice, as it requires individual clients to discover and communicate with large numbers of independent nodes, and requiring a large community in the first place in order to bootstrap the network, since additional nodes controlled by the same operator make the system *less* secure as it increases the number of traitorous nodes if the the operator is malicious.

Proof-of-work protocols. The Bitcoin protocol (Nakamoto 2008) prevents the consensus from being split by requiring a proof-of-work in order for the transaction to be published, hash-linked to the previous state of the ledger. This functions somewhat analogously to a voting system, with the state of the ledger being collectively determined by whichever group has the most computational power.

This type of protocol has the advantage over classical protocols (Cachin et al. 2000) of not requiring large amounts of communication amongst the users in question. For example, the algorithm presented by Cachin et al. (2000) requires a message count that is $O(N^2)$ in the number of users.

A disadvantage of the proof-of-work approach is the need for popularity—the security of proof-of-work-based systems comes from the expense of performing enough computation to compete with the rest of the network, meaning that smaller projects will initially be completely controlled by their founders, and even after the appearance of independent miners, they will be vulnerable for some time to the sudden appearance of an adversary with large amounts of computing power. Furthermore, the computation of these proofs-of-work requires large amounts of power, making the scheme rather inefficient.

Collective signing. An alternative method has recently been proposed by Syta et al. (2015b) and Kogias et al. (2016) that uses collective signing. This allows consensus to be efficiently demonstrated by collectively-generated digital signatures. If the consensus group is known in advance, then this allows us to ensure that the entire group has accepted the same piece of data.

Knowledge of the group members is a potential problem; in Kogias et al. (2016), where the collective signing approach was applied to Bitcoin, group membership is given to those who have recently mined a block, taking advantage of the proof-of-work system to prevent Sybil attacks, in which a single entity pretends to be many users, thus outnumbering the legitimate parties to the protocol (Douceur 2002).

Without a proof-of-work system, some other transparent way of determining who will be invited to take part in the collective signature process is necessary. Nonetheless, such an approach will prove effective, if the necessary infrastructure comes into being.

The primary disadvantage of this approach is the need for dedicated verification infrastructure; this creates a bootstrapping problem when new types of verification are needed. Nonetheless, this may be overcome for verification tasks such as domain verification that have wide commercial appeal.

Multi-path probing. Our proposed technique is a special case of multi-path probing. Multi-path probing involves accessing a service from several points of view in order to detect local variations in responses such as caused a man-in-the-middle attack located far from the service in question.

The first such system was *Perspectives* (Wendlandt et al. 2008). This system uses a number of *notary servers*, which scan publicly-accessible web services for keys. By doing so regularly, they obtain a record of a service's public-key history, and thus allowing users to convince themselves that the server has not changed its public key recently. By accessing multiple such notaries, they can see the key as it appears from several network perspectives. This reduces the risk posed by a malicious notary. Unfortunately, much of this functionality depends upon knowledge of the protocol in use, so that the public key can be extracted. This means that new services cannot be audited with Perspectives until they have developed a following sufficient to justify their support by a large number of notary servers.

A simpler approach, and the direct inspiration for our scheme, is implemented by *DoubleCheck* (Alicherry and Keromytis 2009). When connecting to a server, DoubleCheck makes a second connection via Tor, which it uses to acquire a copy of its certificate. This certificate is compared with that obtained via the direct connection, and the user is warned if they do not match. The same approach is used by *DetecTor* (Engert 2013), which is notable for suggesting that operators use it to verify the state of *their own* servers.

The CONIKS directory system (Melara et al. 2015), includes a Perspectives-like scheme in its architecture, going so far as to include bounds on the probability of successful equivocation by a given number of malicious auditors. Their analysis is related to a special case of our own, but crucially assumes the existence of independent auditors who store and distribute their signed tree roots. This allows clients to see the database from multiple viewpoints, but creates a bootstrapping problem.

Our contribution is to demonstrate that it is possible to design an anti-equivocation system like that proposed for CONIKS without dedicated auditing systems. We describe a DetecTor-like system whose consensus is provably secure, relative to the sender anonymity of the anonymisation system in use.

5.2.3 VERIFICATION PROTOCOL

Suppose that Bob wishes to acquire a piece of information from an untrusted anonymously-accessed service, and Alice the auditor can detect whether a given response from the server is valid. The protocol that we propose is as follows:

1. At a predetermined time, Alice and Bob both request a copy of the message from the service.
2. The service responds to their requests with the message provided.
3. Steps one and two are repeated M times.
4. If Bob does not receive M identical responses, he publicly signals an error.
5. Alice checks that the messages that she has received are identical and valid, and publicly signals an error if not.

We show this in Figure 5.2. Clients who see evidence of equivocation know that the service is untrustworthy, and can report its misbehavior. If the responses are signed, these clients can prove to third parties that the server has equivocated, providing a substantial deterrent to misbehavior on the part of the service.

We note that there is some flexibility in what, exactly, we consider to be our message. Our adversary model in Section 5.2.4 supposes that the channel is deterministic, and so we may apply any form of deterministic processing to the response, yielding our message. This happens implicitly at the IP and Transport Layer Security (TLS) layers, but at higher levels may involve the stripping of timestamps or more complex transformations.

Any anonymising system can be used for this protocol, but in general synchronized system such as a mix-net will be more effective, as these provide little-to-no room for timing attacks. In practice, low-latency anonymisation networks such as Tor are far more available than mix-nets; we discuss the methods used to close the timing side-channel attacks in Section 5.2.7.

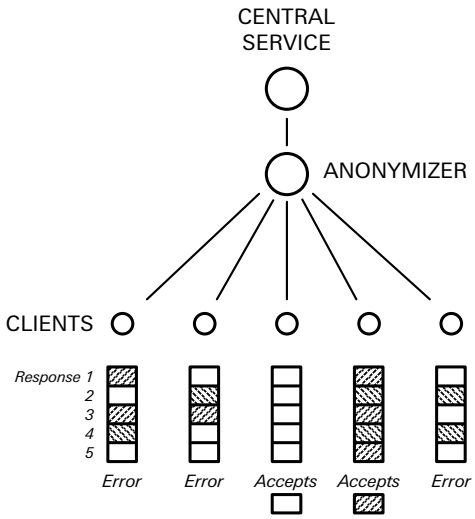


Figure 5.2: Interpretation of the results obtained from the protocol. Clients that have not received consistent responses from the server reject the response from the server, which they know to be faulty. Clients that have consistently received the same response accept it as unequivocated. In this figure, the server has equivocated, with the third and fourth clients being unaware of the fact and the others detecting the misbehavior of the service.

5.2.4 SECURITY ANALYSIS

There are many anonymising systems in use, the most popular by far being Tor (Dingledine et al. 2004). One of the goals of Tor is to prevent users from being deanonymised over the long term (Dingledine et al. 2004). This is a reasonable target, given that one of Tor's stated purposes is the protection of dissidents and journalistic sources from state-level adversaries. Compromizing a single request over the course of many years might well result in catastrophic consequences for the user; even if that single request does not contain any compromising information, it may tie them to a pseudonym—e.g. a social media account whose activities are known. As an example of this, the head of the hacking group *LulzSec* was arrested after connecting to an online chat server on just a single occasion without using Tor (Leyden 2012).

Our requirements are different—whereas a dissident, leaker, or criminal desires to minimize the probability that they will ever be deanonymised, our desire is to minimize the probability that an individual request is deanonymised, since the security of the design that we will describe shortly is determined by the number of requests that can be made without being connected to one another. We will discuss this distinction in greater detail in Section 5.2.4.2, but it is important to highlight that what we describe is only one of many possible definitions of anonymity that has been chosen to meet our needs.

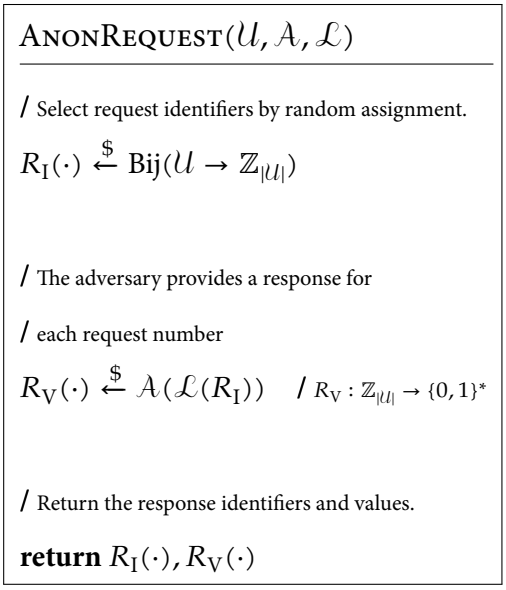


Figure 5.3: A model of an anonymously-accessed service, where $\mathcal{A}$ is the potentially-malicious service, and $\mathcal{L}$ is a leakage function that captures the information leaked to the adversary. In the case of Tor, for example, $\mathcal{L}$ is the user-to-request mapping R_I with its domain restricted to users whose entry guards are surveilled by the attacker. The service accepts a set of users, and selects a random mapping from users to request identifiers. The adversary is given system-dependent partial information on the source of each request, and invited to provide a response to each request.

5.2.4.1 *Definitions*

We begin by defining some notation. We consider a set of users $\mathcal{U} = \{U_1, \ldots, U_N\}$ who take part in the protocol above. This is our anonymity set.

We write the set of injections from A into B as $\text{Inj}(A \to B)$, and bijections from A to B as $\text{Bij}(A \to B)$.

These users connect to a service via an anonymiser, all making identical requests. We model this process in Figure 5.3. The anonymiser makes a request to the adversary on behalf of the clients, providing partial information on which response will go to which user.

5.2.4.2 *Adversary model*

We define our security relative to the security of an anonymity system, and in particular to the notion of sender anonymity as defined by Pfitzmann and Köhntopp (2000), and loosely follow the formalisation given by Backes et al. (2013), but extended to N simultaneous users and M request-response rounds. We select this definition because it provides the most direct route to our statistical quantities of interest; this type of definition can be related to

indistinguishability-based definitions such as those by Backes et al. (2013) in a straightforward manner.

Definition 5.1 (Sender-anonymous service)**.** *Suppose a set of users* $\mathcal{U} = \{U_1, \ldots, U_N\}$ *each make a series of* M *identical and synchronous requests to a service via an anonymiser, receiving a response, as in Figure 5.3.*

Then, consider the experiment in Figure 5.4 for any adversary $\mathcal{A}$*, with the leakage function* $\mathcal{L}$ *being a system parameter. We call the combination of anonymisation system and service* ϵ*-sender-anonymous,* $\epsilon \geq 0$*, if for all adversaries* $\mathcal{A}$*,*

$$\Pr\left[\mathsf{EXP\text{-}SA}_{\mathcal{L},\mathcal{U},M}(\mathcal{A}) = 1\right] \leq \frac{1}{N!}(1+\epsilon)\,. \tag{5.1}$$

This definition assumes that all users operate in lockstep, masking their identities by making identical requests with covert channels sufficiently masked that the probability of successfully linking consecutive requests is no better than chance. We use a multiplicative parameter $1 + \epsilon$ rather than an additive one because this simplifies the analysis to follow; the same results hold with an additive parameter $\epsilon_+ = N!\,\epsilon$.

The most straightforward way to achieve this is the mix-net (Chaum 1981), where a chain of hosts, called *mixes*, re-encrypt and shuffle fixed-sized messages, guaranteeing anonymity so long as at least one member of the chain is honest. The anonymity set here is the set $\{U_i\}$ of users who take part in the protocol.

From our perspective, this means that the adversary is unable respond in such a way as to target a particular user with a particular response. Whether the adversary has compromised the service or is performing a man-in-the-middle attack is immaterial; all we require is that they cannot deanonymise the requests in time to send messages tailored to a particular user.

In some systems this is proven with respect to particular computational hardness assumptions (Young and Yung 2014), whereas other systems such as Tor are *ad-hoc* (Camenisch and Lysyanskaya 2005) and will fall to a global passive adversary. Our approach is implicitly conditional upon whichever assumptions are made by the underlying anonymisation system; should a provably-secure alternative to Tor become equally widespread, it will serve just as well.

There exists the possibility that an attacker might use a denial-of-service to prevent an individual user from accessing the server, if the attacker is able to identify the link that

$\mathsf{EXP\text{-}SA}_{\mathcal{L},\mathcal{U},M}(\mathcal{A})$.

/ Prime the adversary with $M-1$ anonymous requests.

for $i = 1 \ldots (M-1)$

$\quad \text{State} \xleftarrow{\$} \text{State} \parallel \textsc{AnonRequest}(\mathcal{U}, \mathcal{A}_{\text{State}}, \mathcal{L})$

endfor

/ Perform the final request.

$R_{\text{I}}(\cdot), R_{\text{V}}(\cdot) \xleftarrow{\$} \textsc{AnonRequest}(\mathcal{U}, \mathcal{A}_{\text{State}}, \mathcal{L})$

/ Let the adversary identify a response identifier for each user.

$\hat{R}(\cdot) \xleftarrow{\$} \mathcal{A}(\mathcal{U}, \mathcal{L}, \text{State})$

/ The adversary wins if they sent their responses to the

/ users that they thought.

if $\hat{R}(\cdot) = R_{\text{I}}(\cdot)$

$\quad$ **return** 1

else

$\quad$ **return** 0

endif

Figure 5.4: Security experiment for sender-anonymity. An anonymity system, defined by its leakage function $\mathcal{L}$, is used to make requests to an adversary who aims provide particular messages to particular users. The adversary is asked to determine the users to whom each of its responses were sent; it wins if it correctly identifies all of the recipients.

they use to connect to the anonymising service. Defeating this type of attack is outside the scope of this thesis, however we note that it will always be recognized as a fault by the user in question and reported as such.

While this definition makes clear the capabilities of the adversary, it is not ideal for calculation. We will thus make extensive use of the following theorem, which transforms the previous non-constructive adversary definition into a distribution that we can use for further calculations:

Theorem 5.1. *Consider the protocol from Section 5.2.3 with N users, where the anonymising service is a synchronous ϵ-sender-anonymous service, as in Definition 5.1. Then, for any adversary $\mathcal{A}$ with arbitrary knowledge of the recipients of the previous messages, the recipients*

of responses $1, \ldots, N$ *are approximately uniformly distributed over* $\text{Bij}(\mathbb{Z}_N \to \mathcal{U})$*, with each of the* $N!$ *mappings from responses to recipients occuring with probability at most* $(N!)^{-1}(1+\epsilon)$.

Proof. Consider an arbitrary round of requests in the described protocol. We note that the security experiment in Figure 5.4 mirrors steps one to three of the protocol, with the function $R_{\text{I}}(\cdot)$ representing the response destinations for the round under consideration. Thus, by our assumption of ϵ-sender-anonymity, the adversary can predict all of the response destinations with probability at most $(N!)^{-1}(1+\epsilon)$.

We now proceed by contradiction. Suppose the adversary can act in such a way that some response-user mapping $R : \mathbb{Z}_N \to \mathcal{U}$ occurs with probability greater than $(N!)^{-1}(1+\epsilon)$. Then, in Figure 5.4, the adversary can select this mapping as their prediction $\hat{R} : \mathbb{Z}_N \to \mathcal{U}$ of the message destinations. By supposition, this is correct with a probability greater than $(N!)^{-1}(1+\epsilon)$, in contradiction of Definition 5.1, yielding the desired contradiction. ■

We note that when $\epsilon = 0$, this implies that the recipients of each message are perfectly uniformly distributed. In some cases we might be able to justify an individual user as being more secure in the sense of having a smaller ϵ, or we might hypothesise that the adversary behave stochastically or suboptimally; here we consider only the worst-case scenario, as other users cannot guarantee that any of these ϵ-reducing scenarios has actually taken place.

It is this mixing property that we use to provide security. Any response sent by the service will be received with equal probability by all of its users, and thus it is impossible to reliably provide auditors with a different set of records without defeating the anonymiser.

We also posit the existence of some global channel that allows a user to warn others that a fault has occurred, and that the adversary cannot block. We argue that this is a legitimate assumption, since failures can be provided to third-party reporting services or, if all else fails, manually sent via email to a public mailing list. If the server signs and time-stamps its responses, its misbehavior is non-repudiable, thus preventing false-positives from being used to flood the channel.

5.2.4.3 Probability of discord between pairs of users

In our analysis, we consider two separate scenarios. First, that where one user wishes to verify the details of another without trusting that others clients will inform them of inconsistencies in the responses. This is the case with many legacy systems, for example data from PGP keyservers or arbitrary websites, as it is reasonable to assume that one might be the only person attempting to verify the details of any particular user at any given moment.

In the second scenario, the service acts like a traditional broadcaster—many users attempt to access the same data, for example a Merkle tree root for Certificate Transparency, CONIKS, or Bitcoin. In this case we may assume that a certain number of users are active in the protocol and able to publicly report failures—for widely-distributed software, it is implausible that there would not be at least a few hundred or thousand active users at any given time—allowing misbehaviour to be detected with a yet-higher probability.

We start by considering the first case, where a given user is isolated from the other users as in Figure 5.2. Suppose N users each make M identical requests to the sender-anonymous service. It responds with K copies of one message x, and $N - K$ copies of another message x'. These destinations of these messages will be uniformly distributed over the set of users, as shown in Theorem 5.1.

We begin by justifying our use of only two messages, x and x'.

Lemma 5.1. *When the described protocol is run with more than two users, the maximum probability of successful equivocation occurs when only two values are transmitted.*

Proof. Suppose the service can transmit values $\{x, x', x'', \ldots\}$. Then, for any choice of responses, if $x'', x''', \ldots$ are replaced by x', every sequence of responses that do not trigger a failure by any set of users will still be accepted by those users. Thus the maximum probability of successful equivocation is achieved by a service transmitting only the true value x and a single false value x'. ■

This lemma is useful when bounding the probability of acceptance, as it permits us to consider only two possible responses. The goal of the attacker, then, is that some users receive one response value every time, and and others receive the other response value each time. Should Definition 5.1 hold, this is exceedingly unlikely, as we show in Lemma 5.3.

The analysis is eased substantially if we consider a perfect anonymisation system—that is, with $\epsilon = 0$—for which the process of responding to the anonymous requests with one of two responses is readily modelled by the process of pulling coloured balls from an urn. In this analogy, the response x is represented by a white ball, and x' by a black one; the balls are drawn from the urn without replacement, yielding the probability of a particular set of responses over the entire set of users.

We begin by showing how a probability bound calculated with respect to an 'ideal' 0-sender-anonymous service can be loosened in order to apply to a more realistic ϵ-sender-anonymous service.

Lemma 5.2 (Imperfect anonymiser correction.)**.** *Let E be some event taken over the sample space of $N!^M$ message→recipient mappings for an ϵ-sender-anonymous service with N users over M rounds, and $\Pr_\epsilon[E]$ be the probability that E occurs given some such service.*

Then,

$$\Pr_\epsilon[E] \leq \Pr_0[E]\,(1+\epsilon)^M, \tag{5.2}$$

where $\Pr_0[E]$ is the probability that E occurs given a uniform distribution of mappings.

Proof. Let us first consider an individual outcome

$$r \in \Omega = \mathrm{Bij}(\mathbb{Z}_N \to \mathcal{U})^M \tag{5.3}$$

$$= r_1 \times r_2 \times \cdots \times r_M, \tag{5.4}$$

where the $r_i \in \mathrm{Bij}(\mathbb{Z}_N \to \mathcal{U})$ are the response destinations for round i. The event $\{r\}$ in which the outcome r occurs may then be written

$$\{r\} = R_1 \cap \cdots \cap R_M \tag{5.5}$$

where

$$R_i = \mathrm{Bij}(\mathbb{Z}_N \to \mathcal{U})^{i-1} \times \{r_i\} \times \mathrm{Bij}(\mathbb{Z}_N \to \mathcal{U})^{M-i-1}. \tag{5.6}$$

With a uniform probability measure $\Pr_0[\cdot]$, r occurs with probability $N!^{-M}$.

With our adversary-degraded probability measure $\Pr_\epsilon[\cdot]$, r will occur with probability

$$\Pr_\epsilon[\{r\}] = \Pr_\epsilon[R_1 \cap \cdots \cap R_M] \tag{5.7}$$

$$= \prod_{i=1}^{M} \Pr_\epsilon[R_i \mid R_1 \cap \cdots \cap R_{i-1}], \tag{5.8}$$

which Theorem 5.1 bounds by

$$\leq \prod_{i=1}^{M} \frac{1}{N!}(1+\epsilon) \tag{5.9}$$

$$= (1+\epsilon)^M \Pr_0[\{r\}]. \tag{5.10}$$

As the probability space is finite, we may write any event E as a disjoint finite union

$$E = \bigcup_{e \in E} \{e\}, \tag{5.11}$$

and thus

$$\Pr_\epsilon[E] = \Pr_\epsilon\left[\bigcup_{e\in E}\{e\}\right] \tag{5.12}$$

$$= \sum_{e\in E}\Pr_\epsilon[\{e\}] \tag{5.13}$$

$$\leq (1+\epsilon)^M \sum_{e\in E}\Pr_0[\{e\}] \tag{5.14}$$

$$= (1+\epsilon)^M \Pr_0[E], \tag{5.15}$$

the statement that we set out to prove. ■

With this lemma in hand, we may bound the probability of any failure event as though the anonymity of the users is perfect, applying a multiplicative factor $(1+\epsilon)^M$ after the fact in order to account for the imperfect nature of the anonymisation system. This is convenient for our calculations, because the probabilities of our events of interest can then be easily determined by straightforward coloured-balls-in-an-urn calculations.

Lemma 5.3. *Suppose N users take part in the described protocol exactly once, including a particular pair of users Alice and Bob. The service provides K copies of the message x, and $N-K$ copies of a message x'. Then, Bob will receive the value x' and Alice the value x with probability*

$$p_i \leq \frac{K(N-K)}{N(N-1)}(1+\epsilon). \tag{5.16}$$

Proof. Where the service satisfies ϵ-sender-anonymity with $\epsilon = 0$, the recipients of the responses are uniformly distributed and thus this problem is equivalent to that as that of pulling balls from an urn without replacement—without replacement because the connection-oriented nature of the protocol means that each connection can receive only a single response, requiring a bijection from messages—not message values, which are represented by the colour of the balls—to requests. Given an urn containing K white balls and $N-K$ black balls, acceptance is equivalent to drawing first a white ball—probability K/N—and then a black ball—probability $(N-K)/(N-1)$—resulting in a joint probability of

$$p_{(b,w)} = \frac{K}{N}\frac{N-K}{N-1}. \tag{5.17}$$

Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

Theorem 5.2 (Probability of specific failure modes)**.** *Suppose N users take part in the described protocol with a sender-anonymous service, including Alice and Bob. Then, after M iterations Alice will accept the value x and Bob the value x' with probability at most*

$$p_{\text{decep}} \leq \frac{\left[\frac{N}{2}\right]^M \left(N - \left[\frac{N}{2}\right]\right)^M}{N^M (N-1)^M} (1+\epsilon)^M. \tag{5.18}$$

Proof. We first consider the case where $\epsilon = 0$. In order for Bob to accept a false value without detection by Alice, the service must succeed all M times in sending x to Alice and some other value x' to Bob. Lemma 5.1 indicates that the maximum probability of success occurs when only a single false value is emitted, so we assume that all responses are either x or x'.

Suppose that in round i, the service responds K_i times with x and $N - K_i$ times with x'.

The probability that Alice receives x and Bob x' is given by Lemma 5.3 as

$$p_i \leq \frac{K_i(N-K_i)}{N(N-1)}, \tag{5.19}$$

and the probability of Bob accepting the false value without Alice noticing is therefore

$$p \leq \frac{\prod_{i=1}^{M} K_i(N-K_i)}{N^M(N-1)^M}. \tag{5.20}$$

This is maximized by setting $K_i = [N/2]$; when N is odd, K_i can be rounded in either direction by the attacker—rounding up is more likely to result in the true value being accepted, whereas rounding down increases the likelihood of rejection. The maximum probability of a successful attack is therefore

$$p_{\text{decep}} \leq \frac{\left[\frac{N}{2}\right]^M \left(N - \left[\frac{N}{2}\right]\right)^M}{N^M (N-1)^M}. \tag{5.21}$$

Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

As Alice and Bob are unaware of the number of other users accessing the service, they must assume the worst-case value; this occurs when, $N = 2$ yielding $p_{\text{decep}} \leq 2^{-M}$. As the number of users increases, the bound will approach 4^{-M}; we note again that this is the probability of false acceptance for a single pair of users, and so does not take into account the possibility that other users will detect the substitution and report it publicly.

Theorem 5.2 provides an important quantity that is directly applicable to the security of a directory service: the maximum probability that the service can deceive a user looking

up a piece of information without being noticed by a single auditor. We can view this from another point of view, namely the probability of breaking the consensus between pairs of nodes. The essential difference is that Theorem 5.2 does not consider a broken consensus to be a failure if Alice accepts the value x, even if Bob receives a copy of x' and so reports misbehavior, despite the service having successfully broken the consensus.

Theorem 5.3 (Probability of pairwise discord)**.** *Suppose N users take part in the described protocol, including Alice and Bob. Then, after M rounds Alice and Bob will accept distinct values with probability at most*

$$p_{\text{split}} \leq 2\frac{\left[\frac{N}{2}\right]^M \left(N - \left[\frac{N}{2}\right]\right)^M}{N^M (N-1)^M}(1+\epsilon)^M. \tag{5.22}$$

Proof. We first consider the case where $\epsilon = 0$, proceeding as follows: first, we calculate the probability that Alice and Bob will receive different values in the initial round, then we apply Theorem 5.2 to calculate the maximum probability that they will both receive these initial values for the remainder of the protocol.

Suppose that in the first round of the protocol, the server responds with K_1 copies of the value z and $N - K_1$ copies of the value z'. Then, the probability that Alice and Bob will receive different values is the probability of receiving z and z' respectively, or z' and z. As these events are disjoint, this probability is equal to

$$p_1 = 2\frac{K_1(N-K_1)}{N(N-1)} \tag{5.23}$$

by Lemma 5.3.

Let us denote the value received by Alice x and that received by Bob x'. Then, by Theorem 5.2, the probability of that the remaining $M-1$ rounds will result in Alice receiving only the value x and Bob x' is at most

$$\frac{\left[\frac{N}{2}\right]^{M-1} \left(N - \left[\frac{N}{2}\right]\right)^{M-1}}{N^{M-1}(N-1)^{M-1}}, \tag{5.24}$$

yielding an overall consensus-breaking probability of

$$\begin{aligned} p_{\text{split}}[K_1] \leq\ & 2\,\frac{K_1(N-K_1)}{N(N-1)} \\ & \times \frac{\left[\frac{N}{2}\right]^{M-1} \left(N - \left[\frac{N}{2}\right]\right)^{M-1}}{N^{M-1}(N-1)^{M-1}}. \end{aligned} \tag{5.25}$$

This is maximized by setting $K_1 = [N/2]$ and thus

$$p_{\text{split}} \leq 2\frac{\left[\frac{N}{2}\right]^M \left(N - \left[\frac{N}{2}\right]\right)^M}{N^M (N-1)^M}. \tag{5.26}$$

Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

The probability of undetectably breaking the consensus between any pair of nodes thus falls exponentially with time, never being greater than 2^{1-M}.

5.2.4.4 *Probability of undetected consensus-breaking*

We now take a more global view, and calculate the probability that the service can equivocate without being detected by any of its users. Where trustworthy reporting infrastructure exists to allow the publication of equivocation reports to all of the users of the service, this is the applicable probability of failure. Furthermore, from the point of view of the attacker or malicious service operator, this is the probability that their attack will be detected, and thus the most important consideration from a deterrence point of view. The service may attempt to equivocate as before, but the peril in doing so is greatly increased by the need to provide consistent responses to all users.

Theorem 5.4 (Detection of consensus splits)**.** *Suppose N users take part in the described protocol, and the attacker provides the response x' to K users and $x \neq x'$ to $N - K$ users. The probability that the attacker will succeed in providing consistent responses to all N users over M rounds is*

$$p_c \leq \binom{N}{K}^{1-M} (1+\epsilon)^M. \tag{5.27}$$

Proof. We first consider the case where $\epsilon = 0$. By Theorem 5.1, responses to an anonymous service are randomly assigned to users. There exist $\binom{N}{K}$ ways to assign the K false responses amongst the N users, and the attacker must do so identically to the first round for each of the $M - 1$ subsequent or they will be detected.

Note that K must be the same for each round, otherwise at least one user will recognize the deception.

This results in a non-detection probability of

$$p_c \leq \left[\binom{N}{K}^{-1}\right]^{M-1}. \tag{5.28}$$

Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

This probability is maximized by setting K as far from $N/2$ as possible. That is to say, a well-behaved (or consistently misbehaving) service will respond consistently with probability 1, and the maximum probability of breaking the consensus between the users is N^{1-M}, achieved by providing identical responses to all but a single user each round.

In addition to consistency checking, an attacker must contend with users who have the ability to check the validity of their responses directly. Should one of these auditors receive the false value x' directly, they can immediately raise the alarm.

Corollary 5.1. *If an auditor having knowledge of the true value x takes part in the protocol, then the probability of successfully deceiving K out of N users—N including the auditor—without detection by anyone is*

$$p_s[K] = \frac{N-K}{N}\binom{N}{K}^{1-M}(1+\epsilon)^M. \tag{5.29}$$

Proof. We first consider the case where $\epsilon = 0$. We add an additional success criterion to Theorem 5.4. As well as responding consistently to each user, the service must respond to the auditor with the value x in the first round. This occurs with probability $(N-K)/N$, and we multiply by the result stated in Theorem 5.4 to obtain the result above. Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

Theorem 5.5. *The maximum probability of deceiving without detection any member of a group of N users, amongst them an auditor, who follow the protocol above for M rounds is*

$$p_{\text{decep}} \leq \frac{N-1}{N^M}(1+\epsilon)^M. \tag{5.30}$$

Proof. We first consider the case where $\epsilon = 0$. The value of p_{decep} above is that given by Corollary 5.1 with $K = 1$. We are only interested in the case where $K > 0$, since otherwise none of the group have been deceived, and with $K < N$, since then the auditor will detect the false message with probability one. We write the bound from Corollary 5.1

$$p_s[K] = \frac{N-K}{N}\binom{N}{K}^{1-M} \tag{5.31}$$

$$= \frac{(N-K)\,(K!\,(N-K)!\,)^{M-1}}{N!^{M-1}\,N}, \tag{5.32}$$

and hypothesizing that the maximum occurs when $K = 1$, we calculate

$$\frac{p_s[K]}{p_s[1]} = \frac{(N-K)\,(K!\,(N-K)!\,)^{M-1}}{(N-1)\,(N-1)!^{M-1}} \tag{5.33}$$

$$= \frac{N-K}{N-1}\left(\frac{N}{\binom{N}{K}}\right)^{M-1}. \tag{5.34}$$

Since $0 < K < N$, both of these terms are at most one, and thus $p_s[K]$ attains its maximum at $K = 1$, yielding the formula above, for $\epsilon = 0$. Application of Lemma 5.2 yields the original expression for arbitrary security parameters $0 \leq \epsilon$. ■

This demonstrates the difficulty of surreptitiously breaking the consensus between users shielded by an anonymiser. As before the probability of consensus-breaking falls rapidly with protocol iterations, but this time the probability of deception approaches zero—admittedly only polynomially—as the number of users increases.

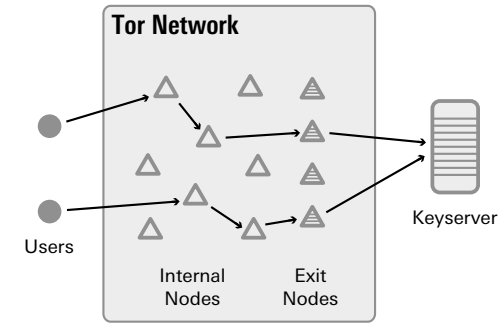
5.2.5 Anonymisation Methods

The question of how to perform the anonymisation is not as straightforward as it might first appear. The simplest way is to use a mix-net, as this naturally provides the lock-step behaviour that we have assumed in our analysis. However, this infrastructure is not widely available, and so we briefly turn our attention to more widely-deployed systems that might prove equally useful.

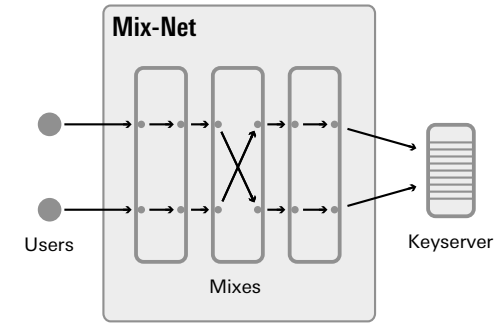
We use Tor in our prototype on account of its wide availability; in addition to its large deployed capacity and mature software, its diversity of relay operators renders systemic failure less likely than with a smaller-scale system intended specifically for our protocol.

Rather than transmitting batches of messages in lock-step, as a mix-net does, Tor immediately forwards its received messages—termed *onions* for their layers of encryption—to the following relay or, if they are the last in the routing chain, to their destination. This reduces the latency of the system, making it usable for interactive tasks. The difference between the structures of these two systems is shown in Figure 5.5.

Despite this, while Tor may render difficult the task of determining which sites a user visits, or conversely which users are visiting a site, our requirement of anonymity at the level of individual requests is more difficult. The first and most obvious point is that Tor channels are reused for ten minutes at a time, and therefore client software must demand a new channel for every request in order to prevent them from being linked by IP address.



a) Accessing a keyserver via Tor.



b) Accessing a keyserver via a mix-net.

Figure 5.5: Connecting to a public keyserver via Tor and via a mix-net. The user randomly selects several relays, then adds a layer of encryption for each relay. After receiving a message, the relays strip their layer of encryption, revealing the address of the next relay. Eventually, the message reaches an *exit node*, which passes it to the open internet. Anyone can contribute nodes to the network—including adversaries—however as the routing path is selected by the user, an attacker cannot gain access to the encrypted messages with probability better than chance. Mix-nets are composed of a chain of *mixes*, which take batches of messages, remove a layer of encryption, shuffle the messages, then pass them to a new mix. If at least one mix in the chain is honest, then an attacker cannot connect messages to their senders with probability better than chance.

Even so, users must be exceedingly careful if they are to avoid giving information away via fingerprinting of their client software.

Another risk is that information will be leaked via timing attacks; if the requests are made at a fixed time, then the order in which the server receives the requests may allow it to link the requests by the clock error of each user. The time of each request must therefore be randomized, as must the times at which channels are set up. An important topic for future work is therefore to develop an asynchronous alternative to the protocol that we have described.

Client software poses a risk as well—if the service being audited uses TLS, it might attempt to fingerprint a user by its available cipher suites, or by the time needed for negotiation to take place.

Tor's use of a long-term guard relay substantially degrades short-term linkability, despite its utility in maintaining anonymity over the long term. Guard relays are stable relays that are selected by the client and then used as the first hop over a period of weeks to months before being changed (Elahi et al. 2012). If the clients do not use a long-term guard relay, then they become vulnerable to predecessor attacks (Wright et al. 2004), in which a malicious relay simply waits until it is selected as the first hop by the client, which it can recognize with traffic analysis.

Our concern, is that an attacker will be chosen as a guard with non-negligible probability, effectively guaranteeing that that client will be deceived, and reducing N by one in the previous analysis. To avoid this, the first hop must change with every request, requiring reconfiguration of Tor.

A possibly more secure approach would be to use some kind of protocol that responds to a single fixed datagram packet, however as Tor does not support UDP, this approach would require the use of some other anonymising network. Nonetheless, with careful design it will be possible to reduce the information leakage to a level that sufficiently masks the source of each request.

Another approach is to use auditing servers, as suggested in the CONIKS architecture (Melara et al. 2015). This is similar to Tor in many respects, with clients selecting the server from which their traffic will appear to come. The use of dedicated auditing servers has some advantages in that they can sign responses from the server, allowing a degree of undeniability on the part of the service being audited, at least to the extent that the auditing servers are trusted. In addition, the auditing server can cache responses from the service, reducing its load and forcing it to commit to its equivocated response for all subsequent requests made to that auditor. This comes at the cost of new server infrastructure with multiple independent operators, or equivalent changes to existing anonymising systems; this prevents the technique from being immediately useful, however the security gain achievable by such 'intelligent' systems is a worthy avenue for future work.

5.2.5.1 Assignment of trust

Given that this protocol requires a trustworthy anonymity system, one might reasonably ask what has been gained from a trust point of view. We avoid the need for a Byzantine consensus

protocol, but depend upon systems like Tor whose basic security properties are themselves dependent upon a consensus protocol.

While such infrastructure might be used directly to audit the service in question, as discussed in previous works, this creates a bootstrapping problem. The value of a consensus protocol derives from the fact that users believe that most of participants are honest. The operators of the Tor directory operators are trusted by the community, and the directories that they produce are small enough to be well-scrutinized, and the consequences of misbehaviour are large. The result is that Tor is—to most users—more trustworthy than any new auditing mechanism will be.

The protocol that we describe in this paper does not allow us to completely sidestep the need to trust an infrastructure provider, but rather allows trust to be restricted to third parties that have no particular interest in the system in question. This solves the current problem of unavailability of trustworthy participants to emerging systems.

5.2.5.2 *Failure of the anonymity system*

Onion routing sacrifices some of the security of mix-networks for low latency (Dingledine et al. 2004). Despite the vulnerability to traffic analysis that results, low latency allows the system to be used for web browsing and other real-time applications, and has driven Tor's wide adoption. Our anonymity system model results in a loose security reduction, the value ϵ increasing rapidly with the number of users.

We consider two quite similar cases: a mix-net with a single honest mix whose outputs are surveilled, and the Tor network. In both of these cases, each request is deanonymised with a fixed probability and independently of the other requests.

We calculate ϵ as follows: let D be the number of users that have been deanonymised; if each user is deanonymised with probability p_d, then this is binomially distributed, with distribution $\mathrm{Bin}(N, p_d)$. The attacker has no knowledge of which requests belong to the other users, and must therefore guess them; this is successful with probability $(N - D)!^{-1}$. Combining these, we compute the probability of correctly identifying the source of every

request:

$$\Pr\left[\mathsf{EXP\text{-}SA}_{\mathcal{L}_{\text{Tor}},\mathcal{U},1}(\mathcal{A}) = 1\right] \tag{5.35}$$

$$= \frac{1}{N!}(1+\epsilon) \tag{5.36}$$

$$= \sum_{d=0}^{N} \Pr[D = d] \frac{1}{(N-k)!} \tag{5.37}$$

$$= \sum_{k=0}^{N} \binom{N}{k} p_d^k (1-p_d)^{N-k} \frac{1}{(N-k)!} \tag{5.38}$$

and thus

$$\epsilon = -1 + \sum_{k=0}^{N} p_d^k (1-p_d)^{N-k} \frac{N!^2}{(N-k)!^2\, k!}. \tag{5.39}$$

In the case of a mix-net, deanonymisation occurs when a layer of encryption is broken for a message; this allows the content at the input and output of the honest mix to be linked. Arbitrarily setting $p_d = 2^{-80}$ and $N = 10$, we obtain $\epsilon \approx 2^{-79}$; this is essentially negligible. Even with 10^4 users, ϵ increases to a still-negligible 2^{-53}.

With Tor, we examine the least favourable case, that where the service in question is malicious. This means that the attacker has knowledge of the anonymised request times and control over the responses. Traffic analysis allows them to deanonymise a request whenever one of their relays is selected by the client for the first hop, thus revealing the client IP address. Suppose that there are N_i entry relays, of which C_i are compromised or surveilled by the attacker. Then, if relays are chosen uniformly, the attacker can deanonymise a channel with probability $p_d = C_i/N_i$. In reality, modern versions of Tor do not select relays with uniform probability, but weighted by bandwidth (Elahi et al. 2012); this can be accounted for by defining C_i and N_i to be bandwidths rather than node counts, however Winter et al. (2016) do not provide this information for the suspicious relays that they detected.

If we consider a reasonably large cabal of 100 malicious relays out of 7000, for 2 users, (5.39) yields the small but non-negligible $\epsilon = 0.003$. This quickly increases to $\epsilon = 2.13$ for only 10 users, a substantial loosening of the security bound.

While we are constrained by space from re-deriving all the results above with respect to the properties of Tor, we will derive an equivalent to Theorem 5.2 for a low-latency onion router, specifically Tor. This demonstrates that control of a moderately-sized cabal of Tor relays does not greatly reduce the security of our initial prototype relative to what we have proven above.

Theorem 5.6. *Suppose N users, including Alice and Bob, access a service via an onion router, each of them being deanonymised independently with probability p_d during each round. Each user executes the protocol described in Section 5.2.3 for M rounds. Then, the probability that Alice will consistently receive the response x and Bob x' is bounded as*

$$p_{\text{decep}} \leq \left[1 - \frac{1}{2}(1 - p_d)^2\right]^M. \tag{5.40}$$

Proof. For each round, three cases are possible: neither Alice nor Bob are deanonymised, occurring with probability $(1 - p_d)^2$, or one is deanonymised, this time with probability $2p_d(1 - p_d)$, or both are deanonymised, this occurring with probability p_d^2. In the last case, the attacker's success is trivial for that round. The same is true if only one of the pair are deanonymised—we suppose without loss of generality that it is Alice—because the server can respond to Alice with x, and to everyone else with x'.

If neither Alice nor Bob have been deanonymised, Theorem 5.3 applies, with the number of users $N_a \geq 2$ being that remaining in the anonymity set. The probability of deception is therefore

$$\begin{aligned} p_r &= \frac{K(N_a - K)}{N_a(N_a - 1)} && (5.41)\\ &\leq \frac{\frac{N_a}{2}\left(N_a - \frac{N_a}{2}\right)}{N_a(N_a - 1)} && (5.42)\\ &= \frac{N_a^2}{4N_a(N_a - 1)} && (5.43)\\ &\leq \frac{1}{2}. && (5.44) \end{aligned}$$

This occurs with probability $1 - (1 - p_d)^2$, and thus the maximum probability that the attacker succeeds during a given round is

$$\begin{aligned} &1 - (1 - p_d)^2 + (1 - p_d)^2 p_r && (5.45)\\ &= 1 - (1 - p_d)^2(1 - p_r) \\ &\leq 1 - \frac{1}{2}(1 - p_d)^2. && (5.46) \end{aligned}$$

Success in each round is independent, so this occurs M times with probability

$$p_{\text{decep}} \leq \left[1 - \frac{1}{2}(1 - p_d)^2\right]^M. \tag{5.47}$$

■

Using Theorem 5.6, this yields a deception probability

$$p_{\text{decep}} \leq \left(1 - \frac{1}{2}\left(1 - \frac{C_i}{N_i}\right)^2\right)^M . \tag{5.48}$$

From Winter et al. (2016, Table 2), we see that most malicious relay groups which escape detection for any length of time have less than 100 members. The Tor network, by comparison, has approximately 7000 relays (The Tor Project n.d.) at the time of writing. The effect is to loosen the bound on attacker success from $p_{\text{decep}} \leq 0.5^M$ to $p_{\text{decep}} \leq 0.514^M$. This shows that Tor achieves the original $N = 2$ security bound with an arbitrarily large number of users, and so remains useful despite its poor ϵ-values in our more general analysis.

5.2.6 DISCUSSION

We have presented a protocol that uses an anonymising service to create an auditable broadcast service. This capability is extremely valuable, and can be used in several ways. We have demonstrated how anonymising networks can be used by individuals in order to distribute their own public keys, but with more and more systems being designed to allow the verification of database entries via a Merkle tree (Laurie 2014; *Keybase* 2016; Melara et al. 2015), we must analyze this type of system as well. In this case, many users can be assumed to access the same service simultaneously, and therefore the results from Section 5.2.4.4 apply. If more than a handful of users take part then detection is near-certain, even with very few rounds.

The requirement that the holder of an identity takes part is an onerous one, but one that could be met should such a technique become ubiquitous, for example if it is performed automatically by default installations of PGP implementations by all major vendors. Even if this were not the case, the approach still serves to reassure the holder of an identity that other users can communicate securely with them if they choose to take this approach.

The need for multiple rounds makes this approach relatively expensive in terms of communication. This, in addition to the time needed for failure reporting, rules it out in most interactive applications. With systems like CONIKS this is not a problem, as data that is a few minutes out of date will not cause any great harm, since the data being broadcast allows any user to be looked up. When verifying individual keys using the existing PGP keyserver network, the process must be performed separately for each key. This results in an delay before first communication can take place, but subsequent verification can be performed in the background to ensure that the previously-verified key is up to date.

	Single Records	Merkle Tree Root
Number of users assumed	2	$N \gg 2$
Items validated per user	L	L
Number of requests per user	ML	$M + L$
Probability of undetected failure	2^{-M}	$(N-1)N^{-M}$
Legacy system support	Yes	No

Table 5.1: Costs and security of the proposed protocol for literal-data and Merkle Tree systems.

5.2.6.1 *Implementation analysis*

Our discussion thus far has been quite general, and we briefly discuss what can be achieved in practice.

The relevant parameters for the system when used to audit keyservers and Merkle Trees is shown in Table 5.1.

The first scenario that we consider is the verification of entries on a PGP keyserver. In this case users access keys directly, verifying them on an individual basis. It is necessary to make a trade-off here between the time needed to achieve a reasonable level of verification, and the load placed upon the keyserver.

The requests in this case take the form of search queries for the email address in question. Both the users and the identity holder must agree on the form of these search queries and how the key is to be selected from the results. In our implementation, the search query is an email address, and the result is taken to be the first valid key listed in the response.

We suppose here that a round will take place every $T = 5$ minutes; thus after time t, $M = \lfloor t/T \rfloor$ rounds will have taken place. Therefore, in order to achieve a maximum failure probability p_{decep}, we require a verification time

$$t = -T \log_2 p_{\text{decep}}.$$

If we arbitrarily determine a success probability of 2^{-20} to be reasonable—it seems implausible that we could do substantially better by any other means, including in-person verification of identity documents—then verification requires 100 minutes, with server load being inversely proportional to the verification time. This is somewhat inconvenient, but far less so than

obtaining a personal certification, which in all likelihood will require several hours of time in order to coordinate, travel, and perform the verification.

Next we consider the Certificate Transparency system. This requires that a user periodically obtain a Merkle-tree root, with newer roots attesting to previous values as well. We model our system on Chrome's software-update system, supposing that the root will be downloaded at the same time. Chrome checks for software updates every five hours (Google n.d.); if it were to randomize the time of checking during each five-hour interval, then this matches the situation that we have analyzed, with the obvious exception being that Chrome does not currently perform any anonymisation.

We make a conservative estimate of Chrome having 100 million active users, though in reality it is most likely several times higher. This time we have $T = 5$ hours, and from Table 5.1 we will require

$$t = -T\frac{\log \frac{p_{\text{decep}}}{N-1}}{\log N}.$$

In this case, it is straightforward to obtain a probability of deception of at most 2^{-20}—after the second request, the probability that anyone will be deceived without the misbehaviour being detected by at least one browser instance or site owner is 2^{-26}, or approximately 1 in 100 million. These waiting times are shown in Figure 5.6.

We reinforce here that this probability is the maximum probability that the service may succeed in deceiving *any* user. Thus the average number of users deceived is approximately p_{decep}—it is possible, albeit unlikely, that more than one user will be deceived—and *not* Np_{decep}.

We see, then, that our results are useful in practice and can provide meaningful security against malicious services.

5.2.7 IMPLEMENTATION

We have developed an implementation of this system, which we have dubbed *Keywatch*[9]. It takes the form of a terminal program that continuously searches for a number of email addresses on an HKP keyserver Shaw 2003 via Tor. We chose to use Tor rather than a mix-network because of its wide public availability.

Requests are made via *libcurl*, using Tor's authentication isolation feature The Tor Project 2015, *IsolateSocksAuth* feature to force the creation of new channels. Connections are made

[9]Source code is available on the media provided.

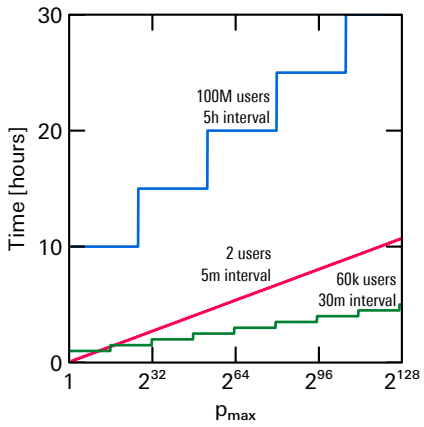


Figure 5.6: Waiting-time necessary to achieve various levels of security. We show the hypothesized Certificate Transparency system modelled on the Chrome auto-update mechanism (top), our proposed keyserver-auditing system (middle), and our conception of how a keyserver built on something like CONIKS might look (bottom). We see that very small probabilities of equivocation are achieved within only a few hours, such that deanonymisation and endpoint compromise quickly become far more likely than chance success by a malicious service.

using plain HTTP, reducing the potential for fingerprinting by the client's TLS configuration and round-trip times. The client is relied on to have a sufficiently accurate clock, which is used to determine the time window for each round of the protocol. The windows are 10 s in duration, and defined to start at integer numbers of periods since 2000-01-01 0000 GMT. This duration is short and only suitable for testing; before leaving the prototype stage, it will be lengthened to several minutes.

Since the clocks of the clients are not necessarily well-synchronized, the request times allow fingerprinting of the clients. In order to avoid this, the time of each request within each window is chosen at random according to $X_n T/2^{64}$, where T is the window duration and X_n is a random number between 0 and $2^{64} - 1$ found by filling a 64-bit unsigned integer with bytes from the operating-system cryptographic random-number generator.

After the index is downloaded, the fingerprint associated with the email address is taken to be that of the first valid—that is to say, unrevoked and unexpired—key to which the email address is associated. Once such a fingerprint has been received, it is retained in memory and compared with the first valid fingerprint from each subsequent request. Should they differ, the key provided by the offending request will be printed to the terminal.

5.2.7.1 *Effects on the Tor network*

Our somewhat unusual use of Tor raises the important question of whether the use of Tor in our system poses a risk to other users of the network, or conversely whether it might improve the anonymity provided by Tor. Our need to disable entry guards disabled renders clients using our protocol rather distinctive, but it is not clear whether this is problematic.

A greater risk from a usability perspective is that misconfigured applications might use our unusually-configured version of Tor for traditional applications, leaving users vulnerable to predecessor attacks. This might be avoided through application-filtering by a local firewall, but safest of all is to use a modified Tor client that enforces some kind of client authentication.

A potential positive effect of this protocol is the enlargement of the anonymity set of Tor users, though this must be balanced against the ease with which an eavesdropper can differentiate between Tor users using our protocol and those using Tor in a more traditional manner. Because the protocol is not highly latency-sensitive, a hypothetical onion router that allows clients to request some delay before the packet is retransmitted might reduce the risk of traffic confirmation attacks to the point that the use of an entry guard can be used, thereby making the use of our protocol far less obvious.

We have shown how an anonymising service such as Tor can be used to perform multi-path probing, and so create a public broadcast channel that permits clients to bound the probability that the broadcasting service can break consensus with the other clients without detection. Failed attempts to provide different messages to different parties can be proven by the detecting party with the aid of digital signatures.

This is an example of how we can improve the security of a database by forcing random selection at the time of access; we now turn our attention to more more traditional certificate infrastructure, and a method by which we can use random selection to prevent misissuance of certificates in the first instance.

5.3 DISTRIBUTED CERTIFICATE ISSUANCE

Suppose you are a bank teller, and one day a prospective customer asks to open an account. To prove their identity, they show their passport and a utility bill showing their address. There is a problem, however: the passport is difficult to forge, so you can be reasonably sure that they are who they say they are. But the proof of their address rests upon the validity of the utility bill, and who can say that none of the dozens of different utility companies, can be convinced or hoodwinked into sending an invoice with an incorrect address?

It is the ability of the prospective customer to choose their proof of address that introduces the risk of error—most companies will not send an invoice with a false address, however a malicious actor will not ask one of these companies, but rather the company at which he has an accomplice in the billing department.

Our approach is to introduce an element of randomness into the identity verification process, by demanding that verifiers be verifiably chosen by lot. This type of procedure is not without precedent, though it has in many cases fallen by the wayside in the modern age. In the ancient city-state of Athens, certain public posts were allocated by lot since at least the 7th century BC (Headlam 1891), a practice known as *sortition*.

The view at the time was that though this meant the administration of the state was neither continuous nor particularly competent, selection by lot prevented power from accumulating outside the public assembly (Headlam 1891); if the value of the city-state as a sovereign unit was its heterogeneity (Aristotle 1944), then a concentration of power diluted its utility.

The currently-most-popular approach, PKI, is particularly vulnerable because of its multitude of failure points—once a false certificate has been created by any of the hundreds of certificate authorities or their delegates, it will be accepted by every relying party until either it is either revoked by the issuer, or trust in the issuer is revoked entirely. In one well-known case (FOX-IT 2012), attackers produced hundreds of certificates, bypassing the auditing systems and remaining undetected until several months later when Google Chrome introduced certificate pinning (Adkins 2011).

Analogously to the Athenian public service, certificate authorities may protect their keys with layer upon layer of physical security ("Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates, version 1.3.3" 2016), but this naturally results in trust decisions being delegated to them; despite their security, their objectives might not be aligned with those of the relying party, whether through attacks on (FOX-IT 2012) or misbehaviour by (Syta et al. 2015a) the certificate authority.

Less reliable verification requires confirmation by multiple parties, ensuring that the agenda in verifying a certificate is not that of a privileged few, but rather but the collective will of the verifiers, analogous to those Athenians who attended the monthly meetings of the assembly and so collectively controlled the administration of their city-state.

A similar approach has been proposed by Schneier (1996, p. 77) for the generation of timestamps; while it has received some discussion in the literature (Bonnecaze et al. 2006), it

does not appear in the relevant standards (Adams et al. RFC 3161, 2001; "Trusted Time Stamp Management and Security" 2005) and is not widely used.

5.3.0.1 Distributed identity management

Whereas a bank that is desperate for the business of a prospective customer lacking sufficient documentation of their identity might send someone to their house to check their address, for online services this is generally not a possibility. Verification of domain names is possible but inconvenient for individuals, and less reliable due to the risk that one's own email account is compromised. This raises the question of whether we might produce our own documentation: can we run the equivalent of a passport office in a distributed manner?

One approach, used by the Perspectives (Wendlandt et al. 2008) system, has public *notary servers* request public keys from all the systems that they can find, allowing the public to query their database to see whether the key that they see matches that seen by the notary, and whether it has remained the same over time or been changed recently. This provides some evidence that a man-in-the-middle attack has not taken place, but the notaries are self-selecting and so may be malicious.

Another system, DoubleCheck (Alicherry and Keromytis 2009), uses the Tor (Dingledine et al. 2004) network to provide another viewpoint when looking up web addresses—the site is loaded via the local network, and the certificate compared with that obtained via Tor. This protects against man-in-the-middle attacks that take place near to the client, however it is of little use where the attacker can interfere with the communications of both the user and the Tor exit node.

The Cothority system (Syta et al. 2015a; Syta et al. 2015b) uses a group signature protocol in order to produce a certificate that is guaranteed to have been seen by a pool external parties, even if they lack the ability to perform the verification themselves. This is the most similar system to our own, however it has the disadvantage of requiring a strong consensus amongst relatively reliable nodes in order to produce a valid certificate.

We have presented a protocol (Gunn et al. 2016a) that uses the anonymisation service Tor to audit services allowing participants to detect whether the untrusted directory server is giving out false information. But most of these approaches require the active participation of one or both parties, making it difficult for them to displace existing systems that allow offline verification.

In this paper we will start by focussing our discussion on the mapping of public keys to email address, however we might almost as easily consider physical addresses, phone

numbers, email addresses, or legal identities. Domain names, however, have the advantage of being verifiable by anyone without great effort or geographical limitation.

There already exists a network of directory servers mapping names and email addresses to public keys, in the form of the keyserver network for the Pretty Good Privacy (Callas et al. RFC 4880, 2007) encryption system. Volunteers place their own servers into a public pool, with updates being received from and passed on to the other servers. This allows anyone to contribute resources to the pool, and reducing cost and providing redundancy, but the servers cannot trust one another, limiting the functionality that they may provide.

A server might be modified to verify the email address of users submitting keys, but others have no reason to believe that this verification actually took place since an attacker could have placed machines under their own control into the pool. The same can be said of signatures attached to the key: without some evidence that those signing it have genuinely verified the user-ids—this being difficult to establish without a personal relationship due to the signers being self-selected—signatures are of no use at all.

This is in contrast with another service, PGP Global Directory [10]. This service requires that users register their email address before they can add a key, and is entirely centralized, allowing it to easily remove old, unused keys from the database. Nonetheless, users only have the word of the server that it has performed the validation correctly, and it forms a single point of failure from the security and robustness points of view.

This raises the question of whether the email verification functionality of the centralized approach can be achieved with a decentralized architecture. We show in this section that the answer is yes.

5.3.1 PRELIMINARIES

We propose a cryptographic solution to the problem. After the procedure is carried out, users will receive a certificate that allows others to verify, with the aid of some public side-information, that the holder of a public key can receive emails sent to a given address.

5.3.1.1 Verification of elections

Our approach is closely connected to the problem of electoral verification. If machines are used to count votes made on paper, how can we be sure that the machines are not biased towards one candidate or another? Counting all of the ballot papers manually defeats the

[10] `https://keyserver.pgp.com`

purpose of the machines, but we might instead check the ballot papers from randomly-selected machines (Clark and Hengartner 2010). Therefore it is important that the selection of the machines to audit be truly unpredictable, and in such a way that the general public can convince themselves of the fact.

One approach is to use what is known as a *randomness beacon* (Rabin 1983): a publicly available source of randomness—stock prices, for example (Carback et al. 2010)—that is available to anyone seeking to verify the procedure after the fact, but that is unpredictable beforehand. This verifiable unpredictability renders random audits inescapable even by a fraudster colluding with the election organizers—deviation from truly random sampling is detectable by the public, meaning the audit of a misbehaving machine must itself be subverted in order to avoid detection.

We propose the use of a randomness beacon to select verifiers; just as its unpredictability allows untrusted parties to manage the electoral audit process, it also permits untrusted entities to coordinate their own verification.

5.3.2 RANDOM VERIFICATION

Our proposed scheme is as follows: suppose there exist verifiers $\mathcal{V} = \{\mathcal{V}_1, \dots, \mathcal{V}_N\}$. Each of these has signing keys $\{K_1, \dots, K_N\}$, which are publicly agreed upon by all relying parties, that is to say all those who wish to verify the certificate connecting key-pair (k, K) to identity $I \in \mathcal{I}$. Suppose also that there is a randomness beacon whose values $X_1, \dots, X_M \in \chi$ are unpredictable to all parties.

This motivates the following definition of an *identity certificate*.

Definition 5.2. *Let $\nu_{\mathcal{V}} : \chi \times \mathcal{I} \longrightarrow \mathcal{P}(\mathcal{V})$ be a function agreed upon by all parties such that for all t and I, $\nu_{\mathcal{V}}(X_t, I)$ is uniformly distributed over the set of r-element subsets of $\mathcal{V}$. Then, an identity certificate binding the key-pair (k, K) to an identity I is the tuple*

$$C = \left(t, k, I, \left\{(i, \text{SIGN}_{K_i}[(t, k, I)]) \mid \mathcal{V}_i \in \nu_{\mathcal{V}}(X_t, I)\right\}\right). \tag{5.49}$$

The certificate is obtained with the protocol shown in Figure 5.7.

The utility of this certificate comes from the difficulty of obtaining the binding signatures—a verifier must not produce a signature $\text{SIGN}_{K_i}[(t, k, I)]$ unless they are convinced that the entity described by I has authorised the binding $k \to I$ at time t, and must not produce more than a single certificate for a given identity for the same time t. Since parties requesting verification have some control over the messages being signed, this requires that the signatures be

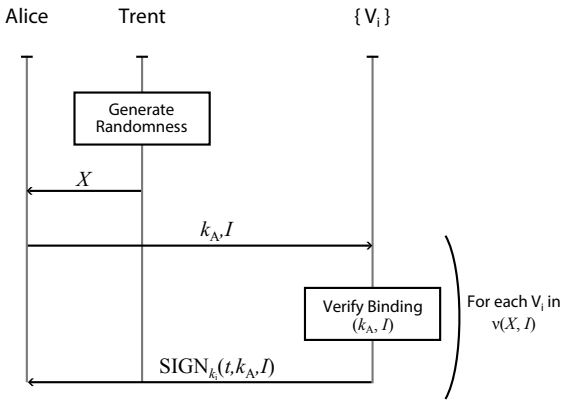


Figure 5.7: The protocol used to obtain a certificate. All communications channels are assumed to be authenticated. After receiving the output of the randomness beacon—named Trent here—Alice, the user in question, sends verification requests to each of the verifiers designated by the beacon, Trent. The verifiers check her identity before signing and returning the request. Note that the need for the beacon to be trustworthy is not a great hindrance, as it might generate its values based on visible public phemonena such as stock prices.

existentially-unforgeable under adaptive chosen-message attacks (Katz 2010, p. 17). If at least $N - r + 1$ verifiers are honest and the verification process perfect, then it is impossible for an unauthorised user to obtain the r signatures necessary to produce a valid identity certificate.

We now concern ourself with the case where more than $N - r + 1$ verifiers collude with the attacker. Provided there are not too many colluding verifiers, we shall see that it is still relatively difficult to obtain a fraudulent certificate.

Theorem 5.7. *Suppose c of the N verifiers are colluding; we call this set $\mathbb{C}$. Then, the probability that k of the r randomly-chosen verifiers chosen at time t collude is*

$$P\left[\left|\nu_{\mathbb{V}}(X_t, I) \cap \mathbb{C}\right| = k\right] = \frac{\binom{c}{k}\binom{N-c}{r-k}}{\binom{N}{r}} \tag{5.50}$$

Proof. Elements of $\nu_{\mathbb{V}}(X_t, I)$ are drawn without replacement from $\mathbb{V}$. The number of accomplices drawn is therefore hypergeometrically-distributed, and its probability mass function is as shown. ■

Corollary 5.2. *The probability that all of the verifiers will be in the colluding set is*

$$P[\nu_{\mathbb{V}}(X_t, I) \subset \mathbb{C}] = \frac{c!}{N!}\frac{(N-r)!}{(c-r)!}. \tag{5.51}$$

Proof. In order that $\nu_{\mathbb{V}}(X_t, I) \subset \mathbb{C}$, we must draw r elements of C from $\mathbb{V}$ without replacement. The number of colluding verifiers drawn is thus described by a hypergeometric distribution,

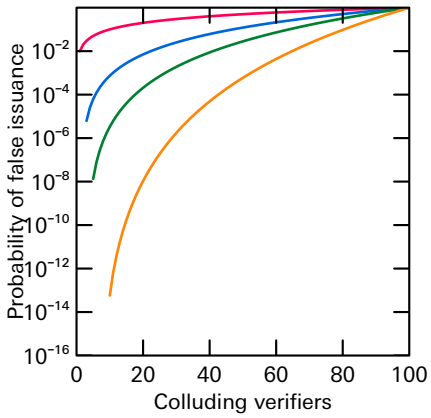


Figure 5.8: Probability of drawing only colluding verifiers as a function of the number of accomplices in a pool of 100. We calculate the probability of drawing—from top to bottom—1, 3, 5, and 10 accomplices using $\nu_{ij}(X_t, I)$, and calculate the mean number of draws that are necessary before this occurs.

having the probability mass function given in Theorem 5.7. In this case, $k = r$, and thus Equation 5.50 reduces to the probability shown. ■

This probability of false issue is shown in Figure 5.8 for a range of required-verifiers parameters r.

However, it must be noted that this per-attempt probability is not the probability that an attacker will succeed in generating a false certificate, since they might simply wait until the beacon produces a value X_t that results in the verifiers under their control being chosen. If they compromise the signing keys of several long-lived verifiers, then they may search historical values of X_t for draws that yield only verifiers under their control.

5.3.2.1 Imperfect verifiers

Our analysis above focusses upon the probability that sortition will fail to keep the verification of an attacker's identity from being entrusted to their accomplices. We now consider the possibility that honest verifiers will make a mistake.

Theorem 5.8. *Suppose that a verifier will incorrectly verify a false identity as true with probability p_e. Then, the probability of producing a false certificate at time t is*

$$p_E = \sum_{k=0}^{r} \frac{\binom{c}{k}\binom{N-c}{r-k}}{\binom{N}{r}} p_e^{r-k}. \tag{5.52}$$

Proof. We know the probability of drawing k colluding verifiers from Theorem 5.7.

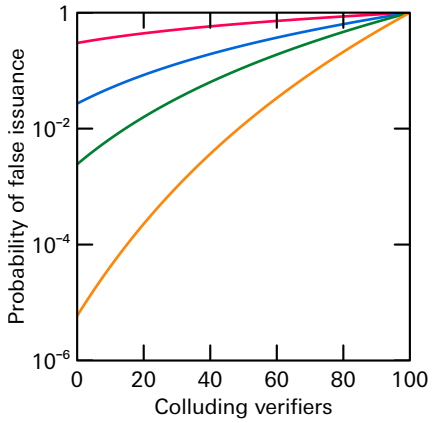


Figure 5.9: Probability of producing a false certificate as a function of the number of accomplices in a pool of 100. We calculate the probability of that 1, 3, 5, and 10 verifiers will agree to produce a certificate, assuming a 30% probability that an honest verifier will falsely accept the attacker's claim.

If k verifiers collude and $r-k$ are honest, then the probability that all r verifiers will emit a signature is p_e^{r-k}, since this will happen only if all of the honest verifiers do so by chance.

The probability that all of the chosen verifiers will provide a signature is thus found by

$$P[r \text{ signatures}] = \sum_{k=0}^{r} P[Y = r - K | K = k] P[K = k], \tag{5.53}$$

where Y Bin$(p_e, r-k)$ is the is the number of false verifications by the honest parties, and K is the hypergeometrically-distributed number of accomplices drawn from the pool of verifiers. ■

The number of attempts necessary after this is taken into account are shown in Figure 5.9.

5.3.3 SUCCESS OVER TIME IN GAINING FALSE CERTIFICATES

In Theorems 5.2 and 5.8 we calculate the probability that an attacker will obtain a false certificate as the result of a single draw. We now estimate the probability that an attacker will succeed in obtaining a certificate for an address over time.

The inclusion of the draw number t in the signed certificate is vital, above and beyond the normal need for expiration, because a machine that has signed a false certificate has, for all intents and purposes, become an accomplice of the attacker for the period of certificate validity.

Suppose an attacker requests verification from all N verifiers; each time they do this, they gain an average of $p_e N$ signatures. The worst-case scenario is that these errors are

independent from round to round, and thus after doing so l times they will gain certifications from an average of $(1-(1-p_e)^l)N$ verifiers. If these are undated, then the attacker will rapidly gain certifications from a large proportion of the verifiers unless prevented via rate-limiting.

By including t in the certificate, the attacker is limited to a single round of verification requests per draw—that is, $l \leq 1$ in the formula above—and they must make fresh requests to the honest verifiers in $v_{lj}(X_t, I)$ for every attempt. This means that after each draw, the probability that a valid signature will be available from a particular verifier from $1-(1-p_e)^l$ to *at most* p_e, as assumed in Theorem 5.8, irrespective of previous verification results.

This independence means that on average p_E^{-1} attempts are necessary in order to gain a certificate; whether this is acceptable depends upon on the reliability of the verification process and the number of verifications required.

5.3.4 AVOIDANCE OF REPEATED REQUESTS

Having calculated the probability that an attempt at obtaining a false certificate will succeed, we now turn our attention to the problem of reducing the number of requests that can be made. The true holder of an identity will not require a thousand attempts at verification in order to obtain a certificate, making it desirable to be able to detect large numbers of attempts by a single entity.

This task is rendered difficult by the fact that attempts at verification do not necessarily require any requests to the honest verifiers—an attacker might simply wait for their accomplices to be chosen by chance.

Trawling of historical data for draws that will result in a favourable result can be prevented by forcing certificates to expire after a time period decided by the relying party; in effect, the relying parties only accept a certain number of recent draws. This reduces the window of vulnerability opened by the draw of a colluding group.

5.3.4.1 Reduce the number of factors affecting the choice of verifier

If the choice of verifiers depends upon the identity I being checked—this might seem reasonable as a form of load-balancing—then an attacker can try many names or email addresses at once, checking the millions of possible $v_{lj}(X_t, I)$ and thereby succeeding far more quickly in gaining *some* certificate than if they tried to produce one for a particular identity.

Decision 5.1. *Should the choice of verifiers depend upon the identity being verified?*

Pro: Verification load is spread evenly amongst the verifiers, and a group of accomplices selected by the system can only produce false certificates for a small number of identities.

Con: An attacker can more easily gain a certificate for a random address by computing $v_U(X_t, I)$ for a large number of identities I.

Though perhaps obvious, it is worth stating explicitly the importance of there being as few inputs as possible to the verifier selection that an attacker might vary independently of the identity being verified; otherwise, the number of attempts possible per beacon output increases proportionally to the size of the parameter space.

5.3.4.2 Public verification requests

The only historical data that we have used thus far for verification are the lists of verifiers, along with those timestamps and public keys necessary to verify that they are correct. This approach reduces the amount of online checking required, but from a security perspective requires that the relying parties presume attackers to be ever-present.

An alternative approach is to require the publication of a verification request prior to the availability of the beacon output X_t. By placing these into a cryptographically secure ledger such as those used by *Keybase* (2016) and Laurie (2014), the number of presumed requests can be reduced, allowing a tighter bound on the probability of false-certificate generation.

Certificates with a corresponding verification request can be far more convincing, because all attempts to produce such a certificate are publicly visible: the security level is based not on the probability that only the accomplices have been selected at some arbitrary point in the past, but on the probability that only they have been selected at one of a finite number of times. Doing so allows the draws to occur far more often, since the number of samples available from the beacon no longer directly determines the probability of failure.

Decision 5.2. *Should public verification requests be required in order to accept a certificate?*

Pro: The probability of generating a false certificate as a function of the number of colluding verifiers is exactly known.

Con: We must maintain an unchangeable public ledger, which must be consulted every time a user wishes to verify a certificate.

This decision is a difficult one—such an approach greatly improves the security of the system, but at the same time increases the necessary infrastructure and requires that the list of verification requests be publicly available, bringing with it privacy and anti-spam concerns.

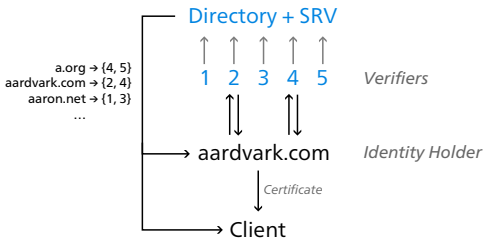


Figure 5.10: Obtaining a certificate for *aardvark.com*. Each verifier periodically registers itself with the directory, particpipating in a commitment protocol to generate a shared random value (SRV) (Goulet and Kadianakis 2015). The resulting directories are distributed globally. A site wishing to obtain a certificate uses the published directory to select the necessary verifiers, then requests verification from each of them. The client, having received a certificate, uses the same directory to select the verifiers responsible for that particular domain at that particular time and verifies that the certificate contains a valid signature from each of them.

5.3.4.3 *Rate-limiting*

A simpler approach is for the verifiers to perform rate-limiting; clearly this provides no benefit if the verifiers are in collusion with the attacker, but it will reduce the probability that an attacker will obtain a certification from an honest verifier by making repeated verification requests. Verifiers might simply refuse to emit certifications after a small number of failures have occurred, or they may add an additional indicator to the certificate indicating that additional scrutiny is required in the form of additional certifications.

The analysis of this situation is complex, demanding analysis of the strategy of the attacker, something beyond the scope of this work. Briefly, however, if each verifier allows only a single failed request per identity per unit time, then for each failed attempt at verification one of the verifiers in the pool will refuse to verify the attacker for some blocking period. This reduces the probability that verification will be possible in subsequent draws.

The attacker can fail at most $N - c$ verification attempts during this period before being blocked by all honest verifiers, forcing them to be far more selective about which draws will lead them to attempt a verification, thereby reducing the probability that they will successfully gain a certificate during a given period.

5.3.5 IMPLEMENTATION

We have developed a proof-of-concept implementation to demonstrate this technique. The system architecture is shown in Figure 5.10.

Central to the system is the directory, which is responsible for determining both $\nu_{ij}(X_i, I)$ and the set of verifiers. Both the identity holder and the relying party need copies of the directory—or at some of its contents—in order to obtain and validate a certificate.

It is possible to do this in a partially—as in the case of the Tor directory (Dingledine et al. 2004) authorities—or fully decentralised way, however we initially use a single server.

Note that a rogue directory server is less of a threat than a rogue certificate authority because of the shared nature of its data. In our implementation the directories are hash-linked, thereby preventing the undetectable substitution of a directory publication except by an adversary sufficiently pervasive that they can prevent a user from ever seeing a legitimate directory in the future.

The second component of our Certificate Authority (CA) infrastructure is the verifier. There are many of these, and they serve two functions: first, they perform verifications and issue certificates. In our case, this means obtaining a copy of the site's TLS certificate and calculating its fingerprint, which is then signed along with a timestamp.

The other role of the verifiers is to generate the random values X_i. We use a similar process to that specified by Goulet and Kadianakis (2015); when registering themselves with the directory server, each verifier commits to a 256-bit random value by providing its SHA-256 hash. At a specified cutoff time, the directory stops accepting new registrations and publishes the submissions. After enough time has passed for interested parties to download a copy, the verifiers reveal their random values, which are published alongside the verifier descriptors in the directory. From these, a random value is computed as the SHA-256 hash of the revealed values, sorted lexicographically by their associated public key.

In order to obtain a certificate, one must select at random the appropriate verifiers to query. We select these by treating an AES-CTR (Ferguson et al. 2010) bitstream as a series of 64-bit integers x, which are taken to refer to verifier number $x \mod N$. Values such that $x + N - (x \mod N) \geq 2^{64}$ are discarded in order to eliminate bias when N does not divide 2^{64}, and similarly when $x \mod N$ refers to a verifier which has already been selected.

The cipher key is the 256-bit random value from the directory, while the initialisation vector is the first 128 bits of the SHA-256 hash of the identity being verified.

The source code is available on the attached medium.

5.4 CONCLUSION

We have demonstrated several methods by which identity management systems can improve their security by functioning in a fundamentally random way. From a security perspective, this makes sense—a system that functions in a stochastic manner is unpredictable to an attacker, preventing it from subverting a distributed system by compromising only a small number of nodes.

5.4.1 ORIGINAL CONTRIBUTIONS

In this chapter, we have described a number of contributions to the state of the art:

- We have described and implemented a method by which multi-path probing can be performed in a way that is amenable to security reduction.
- We have shown that this method admits a security reduction from the probability of successful equivocation by the server to the probability that the server can deanonymise each user that accessing it.
- We have proposed a method for the construction of digital certificates without a central authority for identity verification.
- We show that under this system, attackers must compromise a large numbers of systems in order to achieve the issuance of a certificate with high probability.

Chapter 6

Conclusions and Future Directions

This thesis has examined a wide range of systems from a stochastic measurement perspective, and described a number of advances in the state of the art. We have examined the use of noise to measure nonlinearity in electronic systems, noise-based key establishment systems, and stochastic systems for identity verification. In this chapter, we reiterate our original contributions, and describe opportunities for future work.

6.1 Conclusions and contributions

In this thesis, we demonstrated that the use of stochastic systems and stochastic processing at a high level can significantly enhance the capabilities of a wide range of systems. We summarise our conclusions below, along with a recapitulation of the contributions listed at the end of each chapter.

In Chapter 2, we showed the importance of model validation for stochastic systems, and in particular how high levels of consistency from a system whose behaviour is known to behave probabilistically can negate the conclusions indicated by a naïve interpretation of the data.

- We have investigated the effect of unidentified bias upon identity parades, and shown that even with only a 1% rate of failure, confidence begins to decrease after only three unanimous identifications, failing to reach even 95%.
- We have also applied our analysis of the phenomenon to cryptographic systems, investigating the effect by which confidence in the security of a parameter fails to increase with further testing due to potential failures of the underlying hardware. Even with a minuscule failure rate of 10^{-13} per month, this effect dominates the analysis and is thus a significant determining factor in the overall level of security, increasing the probability that a maliciously-chosen parameter will be accepted by a factor of more than 2^{80}.

In Chapter 3, we demonstrated how knowledge of the noise properties of a signal can be used to measure and correct for nonlinearity in the system that processes it. We showed how this can be used in practice, and demonstrated experimentally a real-time black-box distortion compensator.

- We have examined the output waveform of a transistor amplifier, and demonstrated that it contains noise components that can be used to characterise its response (Gunn et al. 2013).
- We have shown that this method of measurement and compensation is practical using current embedded systems, and demonstrated a real-time system that will automatically remove nonlinear distortion from its input (Gunn et al. 2015a).
- We have demonstrated by simulation that this compensation can be performed in a simplified manner that avoids expensive arithmetic operations such as divisions and square-roots, thereby rendering the approach more practical for integration with an ADC (Gunn et al. 2016b).

In Chapter 4, we discussed several key establishment systems that are based on naturally and artificially noisy channels. We demonstrated several attacks against the Kish key distribution system, and refuted a number of physical arguments that have been used in purported proofs of its security.

- We have examined the use of routing delays as a source of randomness for information-theoretically secure key establishment over the internet, and demonstrated its impossibility on information-theoretic grounds (Gunn et al. 2014c).
- We have experimentally demonstrated the directional measurement of waves in a very short transmission line, experimentally refuting the claims of Kish et al. (2013). We constructed a KKD system, and used this directional coupler to attack the system, in the process demonstrating a state estimator for the system that dramatically outperforms the naïve estimator that has thus far been used to determine the security implications of hardware nonidealities (Gunn et al. 2014a).
- We have shown that the transient behaviour of the KKD system allows a relatively simple attack that requires only two single-time-point measurements (Gunn et al. 2015b).
- We have rebutted the proof of security for the KKD system presented by Kish and Granqvist (2014b), showing that several of its arguments are erroneous and providing a counter-example (Gunn et al. 2015b).

In Chapter 5, we proposed several systems for identity management based on random selection; by introducing random behaviour into these systems at a high level, we prevent an attacker from being able to make effective use of a small number of compromised participants.

- We have described and implemented a method by which multi-path probing can be performed in a way that is amenable to security reduction.
- We have shown that this method admits a security reduction from the probability of successful equivocation by the server to the probability that the server can deanonymise each user that accessing it.
- We have proposed a method for the construction of digital certificates without a central authority for identity verification.
- We show that under this system, attackers must compromise a large numbers of systems in order to achieve the issuance of a certificate with high probability.

6.2 FUTURE WORK

In this thesis we have demonstrated the application of stochastic techniques to a number of applications; however, our investigations are by no means exhaustive. During the course of

this thesis, we have encountered a number of potential research topics that fall outside the scope of or time available for this thesis. We discuss a number of these in the sections to follow.

6.2.1 Failure modes of stochastic systems

The counterintuitive results of Chapter 2 are highly instructive, but our practical results were limited to just a few domains. There are many areas with the potential for similar examination that we have not investigated.

An important limitation of our identity parade calculations derives from our not knowing the probability of bias, and our simple model of its effects. A better model of identity-parade bias will provide a much clearer picture of the practical effect of the phenomenon; unfortunately, the required data is not available to us at this time.

Another interesting application relates to the application of this analysis to airport security or drug screening; the high cost of a positive result provides an incentive for those subjects to avoid the test, making positive results indicate to some degree that the true status of the subject is actually false—a person who knows that they will test positive will evade the test. Our results therefore have practical application in the design of such tests.

6.2.2 Nonlinear sensing

The success of our approach—presented in Chapter 3—to the compensation of nonlinearity indicates that further research in this area is likely to be highly fruitful. Of particular interest is the problem of non-static distortion. Often a signal will be significantly distorted by filtering before it reaches the signal processing stage; this is not adequately described by our model.

A potential solution is to consider multi-dimensional binning; whereas we have considered nonlinearity as a function purely of $x(t)$, it may in fact depend upon $x'(t), x''(t), \ldots$, and so on. As we consider higher-order models, this will require many more samples, but this may be the price that must be paid for such a general technique.

In Section 3.5, we used a feedback method to reduce the computational burden of the algorithm. The use of feedback introduces feedback concerns; the discrete and stochastic nature of the system in question makes convergence rather difficult to prove, and as yet we have been unable to prove its convergence in general, despite empirical evidence in favour of this hypothesis.

6.2.3 NOISE-BASED COMMUNICATIONS

Despite our work to reconcile the non-constructive proof by Bennett and Riedel (2013) of the insecurity of general KKD-like systems with the positive security claims that have since been made by proponents of the KKD system, we are yet to find a concrete attack that cannot be countered by minor changes to the system. While the attack that we propose in Section 4.7.3 serves as a counter-example to many of the arguments in favour of security, its simplicity allows it to be overcome by relatively minor changes to the system.

Given that the information-theoretic arguments against KKD by Bennett and Riedel (2013) have not led to community consensus, debate regarding the system is unlikely to be settled until one of two things happens:

1. An attack is devised and demonstrated experimentally that will allow one to say 'nothing resembling the KKD system can be made secure', or
2. a proof of security is constructed that demonstrates the security of KKD at the lowest possible level, without resort to the simplifying assumptions that have been used up until now, and which have been exploited by our attacks and those of others.

The non-obviousness of the attacks against these systems raises the question of whether there might be some electromagnetic system that can be used for information-theoretic key establishment. Some have concluded in the negative (Bennett and Riedel 2013), whereas others take a more positive view (Kish and Granqvist 2014b). The use of wireless fading for key establishment (Mathur et al. 2008) has found some acceptance, however it depends upon the non-directed nature of the transmissions in question. A more rigorous treatment of the directed case will help to clarify this, whether by proving that such a system can be secure, or by providing more general attacks that will allow greater assurance in the validity of their modelling assumptions than a non-constructive proof of insecurity.

6.2.4 STOCHASTIC APPROACHES TO IDENTITY MANAGEMENT

Despite the practicality of the protocols that we have described, further work is necessary to render them suitable for widespread use.

The auditing protocol that we describe in Section 5.2 requires synchronisation between the clients; this need not be perfect, but nonetheless provides a potential avenue of attack. Another approach is to use Poissonian request timings, with the average number of requests in a given interval being proportional to its length. This approach has the potential to be fruitful, with the uncorrelated request times simplifying the analysis. However, because the

request times are random, the number of requests in a given interval is a random variable. This complicates the situation, and further research is necessary to turn this idea into a practical protocol.

A significant disadvantage of the distributed certificate authority that we describe in Section 5.3 relates to key-management; the large numbers of verifiers require a complex trust-establishment mechanism in the form of a directory. An alternative presents itself in the form of *collective signing* protocols, as described by Syta et al. (2015a). These are used to produce group signatures, which can only be obtained if all members of the group consent. Our technique is highly applicable to such a structure—it is clearly not viable for every authority to verify every website, as noted by Syta et al. (2015a), but as we have shown, the random selection of verifiers prevents misbehaviour except by the largest of cabals. As such, this seems a natural direction for future work in this area.

Appendix A

KKD Attack Apparatus

The attack on the KKD system that we discussed in Section 4.6 required the development of a test communicator, along with the hardware necessary for the actual attack. We detail this here, including a number of design decisions, as well as a description of its operation and software interface.

This appendix describes the directional wave measurement device described introduced in Section 4.6 (Gunn et al. 2014a), and includes a brief description of the theory of operation as well as instructions for testing, calibration, and operation.

A.1 THE HARDWARE PLATFORM

The system is split into three main segments:

- The analogue segment, powered from $\pm$10 V is constructed with AD8428 instrumentation amplifier and NE5532 operational amplifiers.
- The digital segment, powered from USB, runs on an STM32F4DISCOVERY development board.
- The PC segment is written in Python with Numpy, and runs on Linux.

The STM32F4DISCOVERY board was chosen because of its hardware floating-point unit, DMA controllers, high-speed and -resolution ADCs and DACs, and open-source GCC-based toolchain.

Note that the use of Linux for the host is necessary, as Windows will not not create a serial port for the device without a driver file. We used Virtualbox for this purpose.

A.2 THEORY OF OPERATION

A directional coupler separates forward- and reverse-travelling waves on a transmission line (Pozar 1998). We have constructed a similar device using differential measurements across a delay line.

Consider the d'Alembert solution (Jackson 1999) to the wave equation in a medium with propagation velocity ν,

$$v(t,x) = v_+\left(t - \frac{x}{\nu}\right) + v_-\left(t + \frac{x}{\nu}\right). \tag{A.1}$$

The forward-travelling component $v_+(\tau)$ differs from the reverse-travelling component $v_-(\tau)$ in the sign of its spatial argument. We use this to our advantage by computing the linear combinations

$$\frac{\partial v}{\partial t} - \nu\frac{\partial v}{\partial x} = 2\frac{dv_+}{dt} \tag{A.2}$$

$$\frac{\partial v}{\partial t} + \nu\frac{\partial v}{\partial x} = 2\frac{dv_-}{dt}, \tag{A.3}$$

yielding the forward- and reverse-travelling waves as we desire. We must therefore determine $\partial v/\partial t$ and $\partial v/\partial x$, a task that is accomplished with a combination of analogue circuitry and digital signal processing.

We have detailed the statistics of the system in Chapter 4, and will not repeat them here. We restrict ourselves to stating without proof that for a set of measurements $\mathbf{v}$ drawn with equal probability from one of two zero-mean multivariate normal distributions having covariance matrices AA^t and BB^t, the maximum-a-posterior estimator for the choice of distribution given measurements $\mathbf{v}_i$ is

$$\hat{C} = \begin{cases} 0 & \text{if } \sum_{i=1}^{N} \left\| A^{-1}\mathbf{v}_i \right\|^2 > \sum_{i=1}^{N} \left\| B^{-1}\mathbf{v}_i \right\|^2, \\ 1 & \text{otherwise.} \end{cases}$$

Therefore we compute these averaged norms and compare them to one another in order to estimate the bit chosen by Alice and Bob.

A.3 DESIGN

The analogue hardware for this system has been constructed in dead-bug style rather than on a PCB. This method of construction was chosen for ease of modification during development. For ease of development we also used an off-the-shelf microcontroller development board, the STM32F4DISCOVERY. This uses an STM32F407VG microcontroller, an ARM Cortex-M4 with hardware floating point and 12-bit high-speed ADCs and DACs. The microcontroller operates at 168 MHz and 3 V.

A.3.1 ANALOGUE FRONTEND

The analogue frontend to the directional coupler is responsible for measurement of the voltage across a 1.5 m delay line. It presents a moderately high impedance (300 kΩ typical according to the NE5532 datasheet) to the system, and is entirely DC coupled.

The voltage across the delay line is small but still measurable. For a sinusoid of frequency f across a delay line of length τ s, terminated at the far end, the difference in voltage will be approximately

$$\Delta V \approx \tau \frac{d}{dt} \left[\sin(2\pi f t) \right] \tag{A.4}$$

$$= 2\pi f \tau \cos(2\pi f t). \tag{A.5}$$

Letting $\tau = 10\,\text{ns}$ and $f = 1\,\text{kHz}$, this results in a peak differential voltage of $\Delta V = 62\,\mu\text{V}$.

The AD8428 instrumentation amplifier, used to measure the voltage across the delay line, has a fixed gain of 2000 V/V.

While the AD8428 has an equivalent input noise of only $1.5\,\text{nV}/\sqrt{\text{Hz}}$, its large bandwidth of 3.5 MHz demands filtering in order to keep noise to a reasonable level. We use a 2.2 nF capacitor across the filter terminals of the AD8428 to set the bandwidth at 12 kHz. Note that this capacitor must be removed in order to replicate the frequency response shown by Figure 4.14, as it will cancel the differentiating action of the delay line at high frequencies.

In addition to the instrumentation amplifier, the voltage between the line and ground is measured. To this end, one end of the delay line is connected directly to the ADC interface circuitry.

A.3.2 ADC INTERFACE

Additional circuitry is used to provide an interface between the frontend and ADCs. This provides buffering, offset control, additional gain, and clipping.

A DC offset is provided by an NE5532-based subtractor. The offset voltage is set by means of a potentiometer—as it experiences only a DC voltage, nonlinearity is a non-issue and thus any potentiometer will be suitable.

This is buffered by a non-inverting amplifier, the buffered voltage provided to the subtractor. An inverting amplifier with a gain of 1 V/V is used to allow tuning of output gain—we found that a unity-gain amplifier sufficed, and so fixed its gain at one in order to avoid the nonlinearity introduced by a potentiometer.

Finally, a limiting circuit constructed from Schottky diodes is included to limit the resulting signal to the microcontroller rails. The high-side diode is powered from the microcontroller power supply in order to set the upper voltage limit, and the low-side diode connected to ground.

A.3.3 DSP FRAMEWORK

The STM32F407 includes a pair of DMA controllers, of which we take advantage. A buffer, *adc_buffer*, is allocated for the storage of ADC data, and another, *dac_buffer*, for outgoing data to the DACs. In this system, the DACs are used only for debugging and do not serve any functional purpose in normal operation.

The DSP framework is implemented in *dsp.h* and *dsp.c*. The main thread of execution calls *dsp_process* from the main loop. This will check whether the appropriate interrupts have been received from the DMA controller, and if so call *dsp_block_ready* with pointers to the appropriate halves of the buffers.

A.3.4 DSP

The main signal processing code is located in *main.c*, within the function *dsp_block_ready*. It consists of three main parts:

1. Input filtering
2. Calibration
3. Power measurement

Initially a high-pass filter is applied to the incoming samples of both channels. This is a second-order Butterworth filter with a cutoff of 100 Hz. This was chosen to fall between 50 Hz, the local mains frequency, and 500 Hz, the lower frequency of the noise generator.

Next the directional wave components are determined. Rather than simple differentiation, we attempt to automatically correct for line resistance and the frequency response of the instrumentation amplifier by use of a first-order least-mean-squares filter. This filter is applied to the V channel, the voltage between the line and ground. The left- and right-travelling waves are given as

$$V_{\text{left}} = \text{LMS}\,[V] + V_x \tag{A.6}$$

$$V_{\text{right}} = \text{LMS}\,[V] - V_x \tag{A.7}$$

V_{left} is provided to the least-mean-squares algorithm as an error signal. Calibration—described later—consists of applying a right-travelling wave to the coupler and training the filter to produce an output of zero on the left channel.

The training process commences at powerup, after a short interval to allow the input filters to settle. It can also be manually initiated with either the user button or with a command via USB.

The reflection coefficient is estimated via an exponentially decaying average as $\Gamma = E[V_{\text{left}} V_{\text{right}}]/E[V_{\text{right}}^2]$. When $|\Gamma|$ falls below 0.01, filter updates cease and the filter is allowed to settle for a fixed time interval. If, at the end of this interval, $|\Gamma|$ remains less than 0.01, calibration is declared complete. Otherwise, filter updates are re-enabled and the process begins anew.

Power measurement in the system is quite flexible, in order to allow capture of the widest possible variety of statistics. Arbitrary linear combinations of the left- and right-travelling waves are computed—the coefficients are set via USB—and a measurement process is triggered via USB. Four of these combinations can be measured, allowing capture of the entire covariance matrix in a single run. Channels A and B are written to the two DAC channels of the microcontroller, allowing direct observation with an oscilloscope.

After the averaging time has elapsed, the mean-square values of each channel are written over USB as binary floating-point data.

A.3.5 COMMUNICATIONS

The firmware provided will cause the STM32F4DISCOVERY board to appear as a USB-to-serial device to Linux hosts. The device will not be recognised by Windows without a driver file. The host computer may issue SCPI commands—described later—to control the device and perform measurements.

The SCPI parser is implemented in *scpiparser.c*, and it implements a substantial portion of SCPI-99 functionality, including support for units.

The commands themselves are defined in *scpi_cmd.c*. These commands set a variety of flags and parameters to be used by the DSP code.

A.3.6 NOISE GENERATION

The noise used for testing is generated by an Instek AFG-2225 arbitrary waveform generator. Python code—*/Source/Host/noisegen.py*—is used to generate white noise, which is band-limited by zeroing the the appropriate FFT bins. We choose to retain the noise components at 3 ± 2.5 kHz.

Independent noise signals are generated for Alice and Bob respectively, and are saved to a USB drive and loaded to the appropriate channels of the signal generator.

Relays, controlled by the microcontroller board, are used to set the resistor configurations.

A.4 OPERATION

In general, operation of the system consists of the following steps:

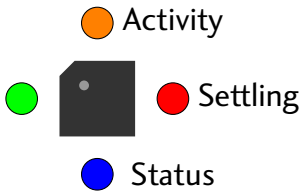


Figure A.1: Status LEDs on STM32F4DISCOVERY board.

1. Power on analog circuitry.
2. Apply test signal to P2, and terminate P5.
3. Power on microcontroller board.
4. Wait for calibration to complete.
5. Connect KKD system.
6. Connect to PC and begin initiating commands.

After step four, the directional coupler will be fully operational and can be commanded by the PC.

The status of the experiment is shown by the LEDs on the board as in Figure A.1.

The blue LED toggles with each block of samples processed.

When calibration is in progress, the orange LED indicates that adaptation is in occuring, while the red LED indicates that adaptation has ceased and that the system is measuring the quality of the calibration.

After calibration has completed, only the blue LED remains lit, except when a power-measurement operation is in progress, during which the orange LED will be lit.

A.4.1 KKD EXPERIMENT OPERATION

A.4.1.1 Interactive operation

Once the system is operating correctly (see Section A.4.2 below), one may interactively operate the experimental apparatus. The driver library is written in Python, and can be used from the IPython shell.

Begin by applying a 250 mV RMS signal to P2 and terminating P5, before powering on the microcontroller board and connecting the data port to the PC. Allow the board to complete calibration.

Take note of the TTY device associated with the microcontroller board. This can be found with the aid of *dmesg*. Then start an IPython shell and connect to the board, substituting the appropriate TTY for */dev/ttyACM0*:

```
import kljn
k = kljn.KLJN('/dev/ttyACM0', '')
```

If this succeeds, *k* will be an object that will henceforth be used to access the microcontroller.

The covariance matrix of the waves can be measured with the *est_C* method:

```
C = k.est_C(0.1)
C
matrix([[ 0.00529442, -0.00389766],
        [-0.00389766,  0.9986527 ]])
```

We can use this to calculate the reflection coefficient looking out of P5, given by $E[V_+V_-]/E[V_+^2]$,

```
C[0,1] / C[1,1]
-0.003902914864202764
```

which is near zero, as expected.

Now connect the KKD apparatus to the coupler, and set the signal voltages to the correct levels:

1. Connect P9 to P2
2. Connect P10 to P5
3. Connect 0.316 V RMS noise signal to P6.
4. Connect 1 V RMS noise signal to P7.

Measure the covariance matrix as above. The waves in each direction will be far closer to being identical, and very highly correlated:

```
C = k.est_C(0.1)
C
matrix([[ 1.4934057 ,  1.48970968],
        [ 1.48970968,  1.48783588]])
```

Now calibrate the coupler for the KKD system using the *set_A* method:

```
k.set_A(0.1)
```

Now we may begin collecting data:

```
k.set_output(0)
likelihoods_0 = [k.likelihood_test(10e-3)
                 for i in range(100)]

k.set_output(1)
likelihoods_1 = [k.likelihood_test(10e-3)
                 for i in range(100)]

plot(likelihoods_0, 'b.')
plot(likelihoods_1, 'r.')
legend(['0', '1'])
show()
```

The test statistics for the cases of zero and one should be distinguishable, as shown in Figure A.2.

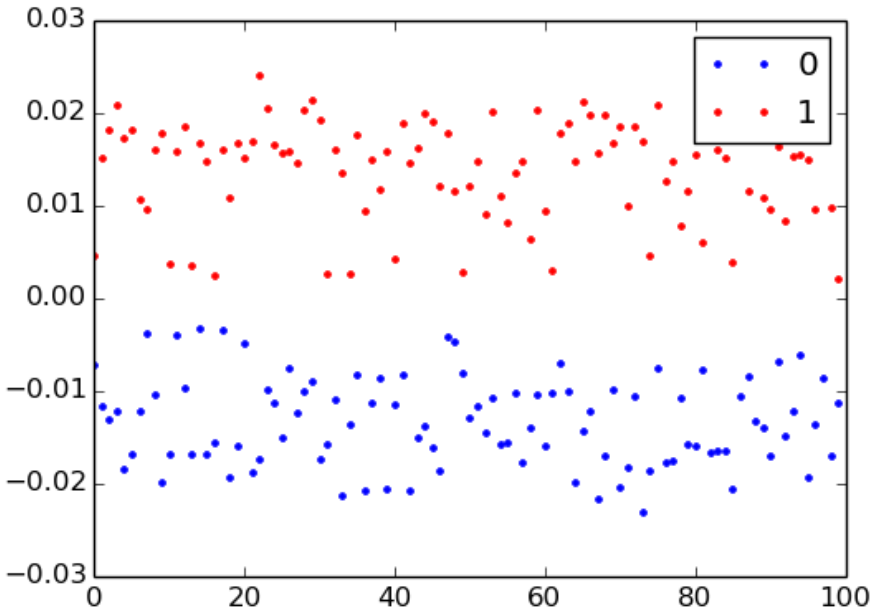


Figure A.2: Measured log-likelihood ratios from KKD test system, as in (4.41), vs. sample number. The red dots represent a single test with the system configured with a key bit of '1', and the blue dots with a key bit of zero. The clear separation demonstrates the distinguishability of the two cases.

A.4.1.2 Batch data collection

The results shown in Chapter 4 were not collected in the interactive fashion shown above; we have produced a script to allow collection of a large number of bits with a variety of averaging times. This script, *kljn-stats.py*, will produce space-delimited output on STDOUT, with each measurement containing:

1. Time of measurement (seconds since start of experiment)
2. Resistor state (zero or one)
3. Averaging time (seconds)
4. Log-likelihood-ratio.

These are processed by the script *process-likelihoods-manyavg.py*, which will read several data files produced by *kljn-stats.py* and plot the eavesdropper's BER vs. averaging time curve for each.

Paths are hardcoded in each of these scripts, and must be modified before use.

Note also that a separate calibration step is not necessary if our KKD apparatus is used, as it provides the required calibration signal.

A.4.2 TESTING

We briefly describe here a sequence of operations that may be used to test for correct operation of the system.

A.4.2.1 Analogue segment

This test will ensure that the full signal path of the analogue segment is operational.

Prerequisites

1. Analogue segment powered up.
   Note: the 3 V *rail is normally powered by the microcontroller board. If this is absent, an alternative source of power must be used.*
2. 1 V (peak) 100 Hz sinusoid applied to P2.
3. 50 Ω termination connected to P5.

Procedure

Coupler frontend

1. Ensure that a 1 V (peak) 100 Hz voltage is present on the line itself.
2. Inspect the voltages at the ends of the delay line. They should be of near-identical magnitude and phase.
3. Inspect the voltage at the output of U1. It should be inverted with respect to the voltage on the line. The magnitude is dependent upon the line resistance.

ADC frontends

1. Inspect the voltage at the output of the U2B of each channel. It should be identical to the input.
2. Inspect the voltage at the output of the U2A of each channel. Its DC level should be a negated copy of the output of the respective U2B.
3. Adjust each R23 in order to place the DC level of each U1A channel at approximately −1.5 V.
4. Inspect the voltage at the output of the U2A of each channel. It should be in phase with the respective V_{in}.
5. Inspect the voltage at the output of the U3B of each channel. It should be inverted with respect to the output of U2A and of equal magnitude. The DC level should be approximately 1.5 V.
6. Inspect the voltage V_{ADC} of each channel. It should be in phase with the output of U3B, possibly with clipping.
7. Increase the magnitude of the test signal in order to verify that clipping thresholds are 0 V and 3 V.

A.4.2.2 Digital and host segments

This test will ensure that the coupler is capable of communicating to the computer and measuring reflections from a variety of terminations.

Prerequisites

1. Analogue segment powered up and functioning.
2. 0.25 V RMS noise signal applied to P2.
3. 50 Ω termination connected to P5.

Procedure

1. Power on microcontroller board, and wait for calibration to complete—only a single LED should be lit after calibration.
2. Connect data port of microcontroller to PC. A new serial port should appear (use *dmesg* to verify this and determine its name).
3. Use *screen /dev/ttyACMX* or similar, where */dev/ttyACMX* is the serial port presented by the microcontroller, to open a serial terminal on this port.

4. Send the command "*IDN?\n" to the microcontroller. It should respond with an identification string.
5. Send the command ":MEASURE:REFLECTION?\n" to the microcontroller. It should respond with four bytes of binary data.
6. Run the Python script *reflection.py* with argument */dev/ttyACMX*. It should print a reflection coefficient near to zero.
7. Remove the termination from P5.
8. Run the Python script *reflection.py*. It should print a reflection coefficient near to +1.
9. Short P5.
10. Run the Python script *reflection.py*. It should print a reflection coefficient near to −1.

A.5 SCPI COMMAND REFERENCE

We describe here the commands supported by the coupler. These use the SCPI protocol, and numerical responses will generally be given in binary format as single-precision floating point. Arguments, however, are to be given as ASCII text. Units are supported, allowing, for example, the specification of time in units of seconds, milliseconds, microseconds, etc.

A.5.1 GENERAL COMMANDS

*A.5.1.1 *IDN?*

Usage:

```
*IDN?
```

Standard SCPI identify command. Responds with a one-line device identification string.

A.5.1.2 :CALibrate

Usage:

```
:CALIBRATE
:CALIBRATE?
```

Controls coupler calibration process. *:CALIBRATE* will reinitiate calibration, while the *:CALIBRATE?* query will respond with a "1" if calibration has been completed, or "0" if it is still in progress.

A.5.2 :MEASURE COMMANDS

A.5.2.1 :MEASure:REFLection?

Usage:

```
:MEASURE:REFLECTION?
```

Measures the reflection coefficient. The coupler continuously computes

$$\Gamma = \frac{E[V_+ V_-]}{E[V_+^2]},$$

and this command will return the current estimate as a single-precision binary floating-point value.

A.5.2.2 :MEASure:POWER?

Usage:

```
:MEASURE:POWER? <duration>
```

Measures the power on each channel, averaged over the specified duration. This command computes

$$P_i = \frac{1}{f_s T} \sum_{k=0}^{f_s T} V_i^2,$$

the mean-squared voltage on each of the four channels, which are returned as four single-precision floating-point values in binary format.

A.5.3 :SENSe COMMANDS

A.5.3.1 :SENSe:WEight<N><D>

Usage:

```
:SENSE:WEIGHT1L <weight>
:SENSE:WEIGHT1R <weight>
:SENSE:WEIGHT2L <weight>
:SENSE:WEIGHT2R <weight>
:SENSE:WEIGHT3L <weight>
:SENSE:WEIGHT3R <weight>
:SENSE:WEIGHT4L <weight>
:SENSE:WEIGHT4R <weight>
```

Sets weighting coefficients for the left- and right-travelling components for each channel. The voltage of each channel is given by

$$V_i = w_{i,\mathrm{L}} V_+ + w_{i,\mathrm{R}} V_-,$$

with channels one and two being written to the two DAC outputs.

A.5.4 :OUTPUT COMMANDS

A.5.4.1 :OUTPut:ZERO

Usage:

```
:OUTPUT:ZERO
```

Configures the resistors to represent a one.

A.5.4.2 :OUTPut:ONE

Usage:

```
:OUTPUT:ZERO
```

Configures the resistors to represent a zero.

A.5.4.3 :OUTPut:DIRectional

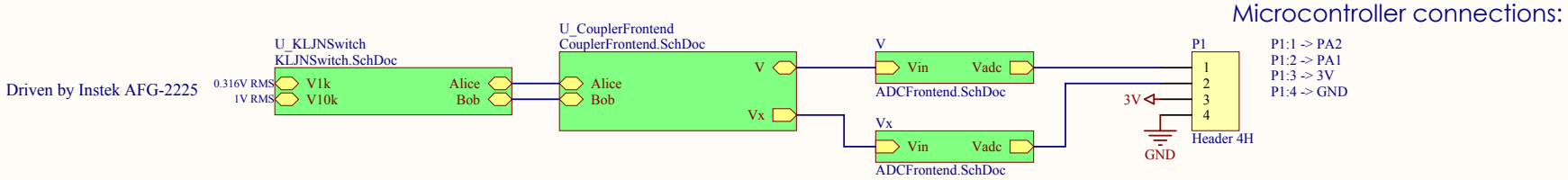
Usage:

```
:OUTPUT:DIRECTIONAL
```

Connects Alice's voltage source directly to P9, and terminates P10. This produces a wave travelling from left to right with no left-travelling component.

A.6 SCHEMATICS

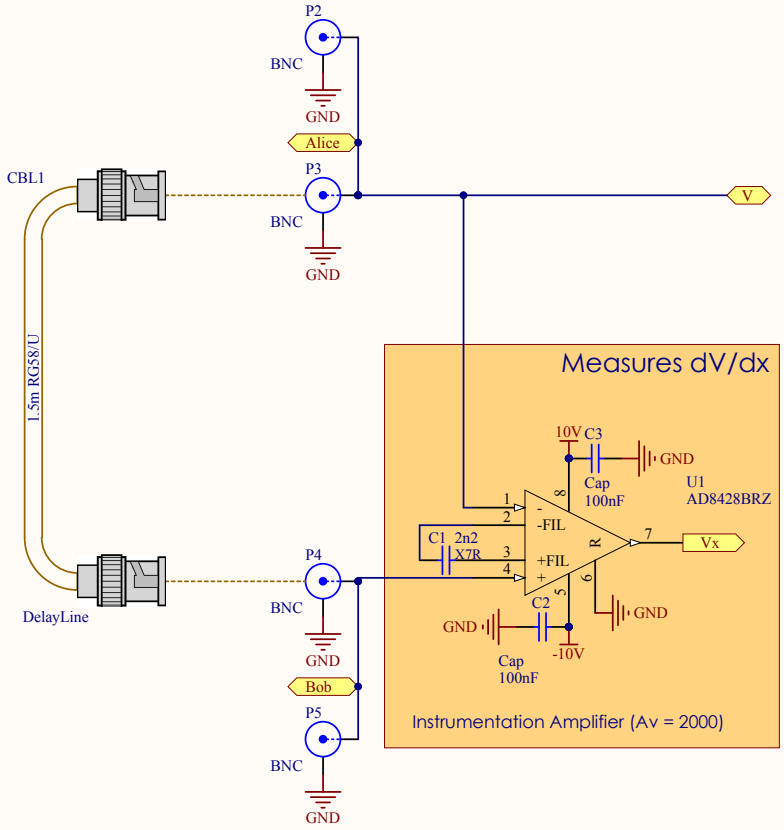
We provide here schematics for the system. Included in the package is the Altium project used to produce these diagrams.



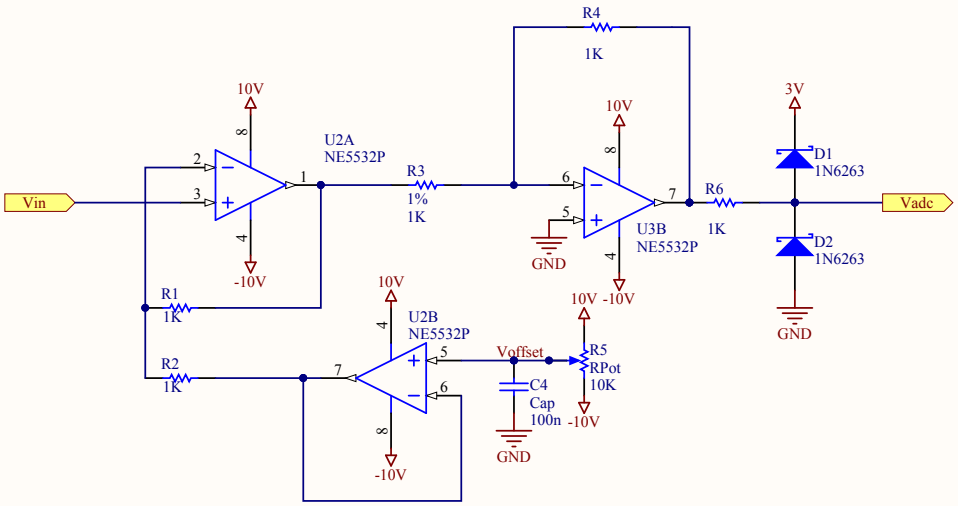
Title: KLJN Directional Coupler Attack		
Size: A4	Number:	Revision: 2
Date: 2/10/2014		Sheet 1 of 5
File: U:\Research\..\KLJNRoot.SchDoc		Drawn By: Lachlan Gunn

Winding directions

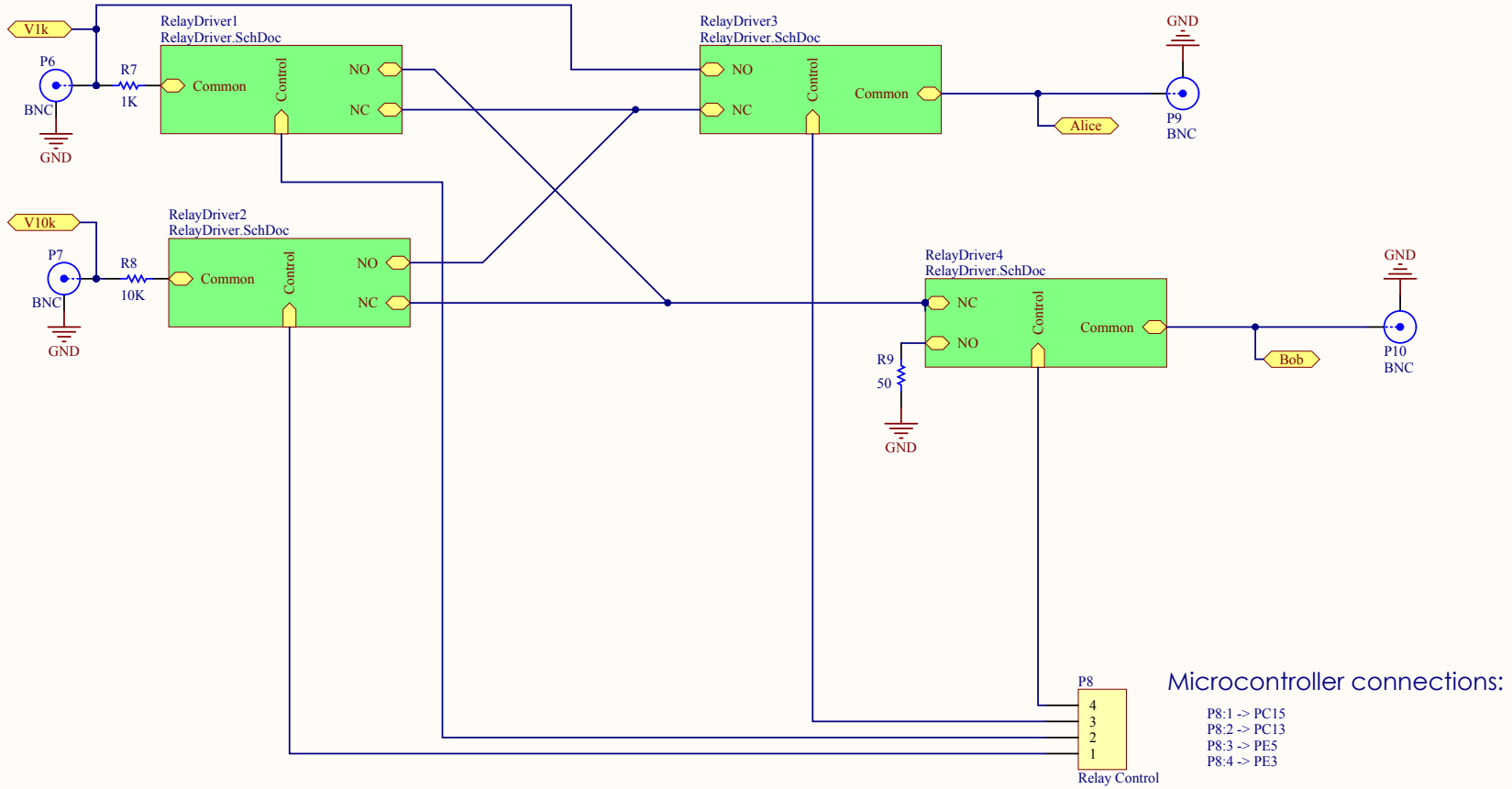
1. Fold in half, bringing connectors together.
2. Hold connectors and rotate other end to produce a twisted pair-like structure.
3. Wind resulting twisted pair into a loop.
4. Put one connector through the loop formed by twisting.
5. Adjust to minimise mains pickup at Vx.



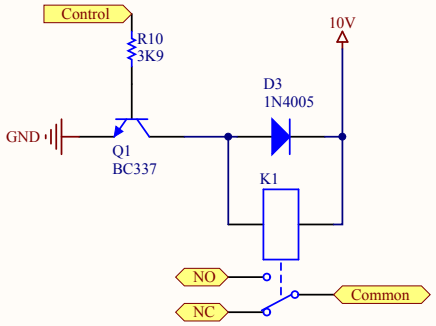
Title	KLJN Coupler Frontend		
Size	A4	Number	Revision 3
Date:	2/10/2014	Sheet 2 of 5	
File:	U:\Research\..\CouplerFrontend.SchDoc	Drawn By:	Lachlan Gunn



Title	ADC Frontend	
Size: A4	Number:	Revision: 3
Date: 2/10/2014		Sheet 3 of 5
File: U:\Research\..\ADCFrontend.SchDoc		Drawn By: Lachlan Gunn



Title: KLJN Switching Unit		
Size: A4	Number:	Revision: 1
Date: 2/10/2014		Sheet 4 of 5
File: U:\Research\..\KLJNSwitch.SchDoc		Drawn By: Lachlan Gunn



Title: Relay driver		
Size: A4	Number:	Revision: 1
Date: 2/10/2014		Sheet 5 of 5
File: U:\Research\..\RelayDriver.SchDoc		Drawn By: Lachlan Gunn

A.7 SOFTWARE BUILD ENVIRONMENT

We have provided full source code for both microcontroller and host. The host software is written in Python and does not require compilation. The microcontroller, however, is programmed in C and so requires the use of a cross-compilation toolchain. We have used the *arm-gcc-embedded* toolchain, provided by ARM themselves. This is available for a variety of platforms, however we have used Windows 7 for development, with the *make* utility provided by MSYS.

In addition, to program the microcontroller, the *ST-LINK Utility* is necessary. This is available only for Windows, though the cross-platform OpenOCD program supports the STM32F4DISCOVERY board.

The source code for the directional coupler is located within the directory */Source/Microcontroller/coupler*. It will be necessary to modify the variables *BINPATH* and *LIBPATH* to to match the installation directory of the toolchain. If on Windows, *STLINK* can be modified to allow programming of the microcontroller from the makefile.

Appendix B

Source Code

In undertaking this research, we have developed several pieces of software that in many cases formed a substantial part of the work involved in our treatment of a topic. Here we provide selected portions of this source code, the entirety of which is available electronically.

B.1 NONLINEAR SENSING

B.1.1 FLOATING-POINT IMPLEMENTATION

B.1.1.1 undistorter.c

```
#include <math.h>

#include "undistorter.h"

#include "filter.h"
#include "smoother.h"
#include "noise_demux.h"

#ifdef USE_CMSIS
    #include "arm_math.h"
#endif

#ifdef _MSC_VER
    int isnan(float x)
    {
        return ( x != x );
    }
#endif

void undistorter_init(
    struct undistorter_ctx* ctx,
    float cutoff,
    float fs, int stats_every,
    int recompensate_every, float min, float max,
    float time_constant)
{

    smoother_init(&(ctx->smoother), min, max,
        time_constant*fs/BLOCK_SIZE);
    smoother_create_integral(&(ctx->integral),
        &(ctx->smoother));

    noise_demux_init(&(ctx->demux), BLOCK_SIZE,
        cutoff, fs, stats_every, 0.0f);

    ctx->signal_min = min;
    ctx->signal_max = max;

    ctx->offset = 0.0f;
    ctx->scale = -1.0f;

    ctx->blocks_since_recompensation = recompensate_every;
    ctx->recompensate_every = recompensate_every;

}

void undistorter_process_sample(
    struct undistorter_ctx* ctx, float x)
```

```
{
    int new_stats;
    float new_signal, new_noise;

    new_stats = noise_demux_process(&(ctx->demux), x,
        &new_signal, &new_noise);

    if(new_stats)
    {
        undistorter_process_sample_noise(
            ctx, new_signal, new_noise);
    }
}

void undistorter_process_sample_noise(
    struct undistorter_ctx* ctx,
    float signal, float noise)
{
    float std_noise_reciprocol;

#ifdef USE_CMSIS
    arm_sqrt_f32( 1.0f/noise, &std_noise_reciprocol );
#else
    std_noise_reciprocol  = sqrtf( 1.0f/noise );
#endif

    if(noise > 0 &&
        !isnan(std_noise_reciprocol) && !isnan(noise))
    {
        undistorter_process_sample_gain(
            ctx, signal, std_noise_reciprocol);
    }
}

void undistorter_process_sample_gain(
    struct undistorter_ctx* ctx,
    float signal, float gain)
{
    smoother_process_point(&(ctx->smoother), signal, gain);

    if( ctx->blocks_since_recompensation
        >= ctx->recompensate_every)
    {
        float integral_min, integral_max;
        float new_scale;
        smoother_create_integral(
            &(ctx->integral), &(ctx->smoother));

        integral_min = smoother_evaluate_integral(
            &(ctx->integral), ctx->signal_min);
        integral_max = smoother_evaluate_integral(
            &(ctx->integral), ctx->signal_max);
```

```
        new_scale = 0.9f*(
            ctx->signal_max - ctx->signal_min)
            / (integral_max-integral_min);
        if(new_scale < ctx->scale || ctx->scale < 0.0f)
        {
            ctx->scale  = new_scale;
            ctx->offset =
                ctx->signal_min - integral_min
                + 0.05f*(integral_max - integral_min);
        }

        ctx->blocks_since_recompensation = 0;
    }

    ctx->blocks_since_recompensation++;
}

float undistorter_compensate_sample(
    struct undistorter_ctx* ctx, float x)
{
    undistorter_process_sample(ctx, x);
    return smoother_evaluate_integral(&ctx->integral, x);
}
```

B.1.1.2 smoother.c

```
#include <stdlib.h>
#include <math.h>

#include "smoother.h"

/**
 * Convert a location in the input domain into an array index.
 *
 * @param ctx   The context for the smoother in question.
 * @param x     The point whose index is to be located.
 *
 * @return The smallest index whose region of interest
 *          contains the point.
 */
static float smoother_find_index(
    struct smoother_ctx* ctx, float x)
{
    return (x - ctx->min_value)
        / (ctx->max_value - ctx->min_value)
        * (ctx->N-1.0f);
}

/**
 * Convert a location in the input domain into an array index.
 *
 * @param ctx_int   The context for the integral in question.
 * @param x         The point whose index is to be located.
```

```
 *
 * @return The smallest index whose region of interest
 *         contains the point.
 */
static float smoother_find_index_integral(
    struct smoother_integrated_ctx* ctx, float x)
{
    return (x - ctx->min_value)
            / (ctx->max_value - ctx->min_value)
            * (ctx->N-1.0f);
}

void smoother_init(struct smoother_ctx* ctx,
    float min, float max, float time_constant)
{
    int i;

    ctx->N = SMOOTHER_POINTS;
    ctx->min_value = min;
    ctx->max_value = max;

    ctx->decay_constant = expf(-1.0f/time_constant);

    for(i = 0; i < SMOOTHER_POINTS; i++)
    {
        ctx->values[i] = 0.0f;
        ctx->weights[i] = 0.0f;
    }
}

void smoother_process_point(
    struct smoother_ctx* ctx, float x, float y)
{
    float idx_f;

    int idx_integer;
    float idx_fractional;
    float weight_0;
    float weight_1;

    /* First find where in the signal range the point lies. */
    idx_f = smoother_find_index(ctx, x);

    /* Check for out-of-range. */
    if(idx_f < 0 || idx_f >= ctx->N - 1)
    {
        return;
    }

    /* Split the index into integer and fractional parts. */
    idx_integer = (int)idx_f;
    idx_fractional = idx_f - (float)idx_integer;
```

```
    /*
     * The integer part is the index of the reference
     * before the point, and the fractional part tells
     * how far into the interval it is. This is rather
     * convenient, since the fractional part thus
     * provides the weight.
     */

    weight_0 = (1 - idx_fractional)*(1-ctx->decay_constant);
    weight_1 = (    idx_fractional)*(1-ctx->decay_constant);

    /* Add to the weighted average. */
    ctx->weights[idx_integer] *= ctx->decay_constant;
    ctx->values[idx_integer] *= ctx->decay_constant;
    ctx->weights[idx_integer] += weight_0;
    ctx->values[idx_integer] += weight_0*y;

    ctx->weights[idx_integer+1] *= ctx->decay_constant;
    ctx->values[idx_integer+1] *= ctx->decay_constant;
    ctx->weights[idx_integer+1] += weight_1;
    ctx->values[idx_integer+1] += weight_1*y;
}

float smoother_evaluate(struct smoother_ctx* ctx, float x)
{
    float idx_f;

    int idx_integer;
    float idx_fractional;

    float weight_0;
    float weight_1;

    float value_0;
    float value_1;

    /* First find where in the signal range the point lies. */
    idx_f = smoother_find_index(ctx, x);

    /* Check for out-of-range. */
    if(idx_f < 0)
    {
        idx_f = 0.0f;
    }
    else if(idx_f > ctx->N)
    {
        idx_f = ctx->N;
    }

    idx_integer = (int)idx_f;
    idx_fractional = idx_f - (float)idx_integer;

    /*
```

```
     * The integer part is the index of the reference before
     * the point, and the fractional part tells how far into
     * the interval it is. This is rather convenient, since
     * the fractional part thus provides the weights.
     */
    weight_0 = 1 - idx_fractional;
    weight_1 =     idx_fractional;

    value_0  = ctx->values[idx_integer  ]
        / ctx->weights[idx_integer  ];
    value_1  = ctx->values[idx_integer+1]
        / ctx->weights[idx_integer+1];

    return weight_0*value_0 + weight_1*value_1;
}

void smoother_create_integral(
    struct smoother_integrated_ctx* ctx_int,
    struct smoother_ctx* ctx)
{
    int i;
    float max_0, max_1, width;

    ctx_int->N = ctx->N;
    ctx_int->min_value = ctx->min_value;
    ctx_int->max_value = ctx->max_value;

    /*
     * In this function we just set up the values from which
     * to interpolate. This means integrating up all of
     * triangles, which is rather straightforward---they each
     * have area 0.5*w*h.
     */
    ctx_int->interp_c0[0] = 0.0f;

    for(i = 1; i < ctx->N; i++)
    {
        width = (ctx->max_value - ctx->min_value)/(ctx->N - 1);

        if(ctx->weights[i-1] == 0)
        {
            max_0 = 0.0f;
        }
        else
        {
            max_0 = ctx->values[i-1]/ctx->weights[i-1];
        }

        if(ctx->weights[i] == 0)
        {
            max_1 = 0.0f;
        }
        else
```

```
        {
            max_1 = ctx->values[i  ]/ctx->weights[i  ];
        }

        /* Zero is a special case. */
        if(i > 0)
        {
            ctx_int->interp_c0[i] = ctx_int->interp_c0[i-1] +
                0.5f*width*(max_0 + max_1);
        }

        /*
         * Integrate
         *   max_1*((x-x_0)/width) + max_0*(1 - (x-x_0)/width)
         * to get these coefficients, substituting
         * x = width*u+x_0, so as to rescale from 0 to 1.
         */
        ctx_int->interp_c1[i-1] = max_0*width;
        ctx_int->interp_c2[i-1] = 0.5f*(max_1-max_0)*width;
    }
}

float smoother_evaluate_integral(
    struct smoother_integrated_ctx* ctx_int, float x)
{
    float idx_f;

    int idx_integer;
    float idx_fractional;

    float integral_value;

    /* First find where in the signal range the point lies. */
    idx_f = smoother_find_index_integral(ctx_int, x);


    /* Check for out-of-range. */
    if(idx_f < 0.0f)
    {
        idx_integer = 0;
        idx_fractional = 0.0f;
    }
    else if(idx_f >= ctx_int->N)
    {
        idx_integer = ctx_int->N-1;
        idx_fractional = 1.0f;
    }
    else
    {
        idx_integer = (int)idx_f;
        idx_fractional = idx_f - (float)idx_integer;
    }
```

```
    /* Now evaluate the integral.  First the "whole" part. */
    integral_value = ctx_int->interp_c0[idx_integer];

    /*
     * We have computed coefficients for the quadratics
     * between each reference point in terms of the fractional
     * parts of the indices.  We evaluate the polynomial
     * and add it on.
     */

    /* The linear term of the fractional part. */
    integral_value +=
        ctx_int->interp_c1[idx_integer]*idx_fractional;

    /* Finally the quadratic. */
    integral_value +=
        ctx_int->interp_c2[idx_integer]
        *idx_fractional*idx_fractional;

    return integral_value;
}

void smoother_create_integral_lookup(
    struct smoother_integrated_ctx* ctx_int,
    float* table, int N, int start, int count)
{
    int i;
    float this_value;
    float increment;

    increment  = (ctx_int->max_value - ctx_int->min_value)/N;
    this_value = ctx_int->min_value + increment*start;

    for(i = 0; i < count; i++)
    {
        table[i+start] = smoother_evaluate_integral(
            ctx_int, this_value);

        this_value += increment;
    }
}
```

B.2 NOISE-BASED COMMUNICATIONS

B.2.1 DIRECTIONAL COUPLER

```
#include <stdio.h>
#include <stdlib.h>
#include <stdint.h>
#include <stdbool.h>
#include <math.h>

#include "stm32f4xx_conf.h"
```

```
#include "stm32f4xx.h"
#include "main.h"

#include "usbd_cdc_core.h"
#include "usbd_usr.h"
#include "usbd_desc.h"
#include "usbd_cdc_vcp.h"
#include "stm32f4_discovery.h"

#include "dsp.h"
#include "tracking.h"
#include "scpiparser.h"

// Private variables
volatile uint32_t time_var1, time_var2;
__ALIGN_BEGIN USB_OTG_CORE_HANDLE  USB_OTG_dev __ALIGN_END;

uint8_t tracking = 3;
float a_1 =  -21.0f;
float a_2 = 21.0f;

int32_t a_1_int = 0;
int32_t a_2_int = 0;

int32_t w_1_L = 1024;
int32_t w_1_R = 0;

int32_t w_2_L = 0;
int32_t w_2_R = 1024;

int32_t w_3_L = 1024;
int32_t w_3_R = 1024;

int32_t w_4_L = 1024;
int32_t w_4_R = 1024;

float mean_V = 2048;
float mean_directional = 2048;

uint32_t mean_V_int;
uint32_t mean_directional_int;

float mean_LR = 1000.0f;
float mean_RR = 1000.0f;

float sum_A_squared;
float sum_B_squared;
float sum_C_squared;
float sum_D_squared;
int32_t power_samples_remaining = 0;
int32_t power_block_size;

float power_scaling_factor = 1.0f;
```

```
//Configure pins and clocks
void hw_init()
{
   GPIO_InitTypeDef  GPIO_InitStructure;

    // ---------- SysTick timer --------- //
    if (SysTick_Config(SystemCoreClock / 1000)) {
        while (true)    // Capture error
         ;
    }

    // ---------- GPIO --------- //
    // GPIOD Periph clock enable,
    // Need to enable GPIOA because that's where the UART pins are.
    // (Some of the USB is also on that port, and usb modules turn
    // it on later... but anyway, UART started working correctly
    // when I turned clock on first)
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOA, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOC, ENABLE);
    // LEDs are on GPIOD
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOD, ENABLE);
    RCC_AHB1PeriphClockCmd(RCC_AHB1Periph_GPIOE, ENABLE);

    // Relay outputs
    // Relays
    GPIO_StructInit(&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin  = GPIO_PIN_RELAY1;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIO_BANK_RELAY1, &GPIO_InitStructure);

    GPIO_StructInit(&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin  = GPIO_PIN_RELAY2;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIO_BANK_RELAY1, &GPIO_InitStructure);

    GPIO_StructInit(&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin  = GPIO_PIN_RELAY3;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIO_BANK_RELAY3, &GPIO_InitStructure);

    GPIO_StructInit(&GPIO_InitStructure);
    GPIO_InitStructure.GPIO_Pin = GPIO_PIN_RELAY4;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
```

```
    GPIO_Init(GPIO_BANK_RELAY4, &GPIO_InitStructure);


    // Configure PD12, PD13, PD14 and PD15 in output
    // pushpull mode
    GPIO_InitStructure.GPIO_Pin =
          LED_GREEN_PIN | LED_ORANGE_PIN
        | LED_RED_PIN   | LED_BLUE_PIN;
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_OUT;
    GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_100MHz;
    GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_NOPULL;
    GPIO_Init(GPIOD, &GPIO_InitStructure);

   // ---------------- USART ----------------- //
   // USART1+6=APB2, 2-5=APB1
   RCC_APB1PeriphClockCmd(DISCOVERY_COM_CLK, ENABLE);


   /* Configure USART Tx+Rx as alternate function  */
   GPIO_InitStructure.GPIO_Pin =
        DISCOVERY_COM_TX_PIN| DISCOVERY_COM_RX_PIN;
   GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF;
   GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
   GPIO_InitStructure.GPIO_OType = GPIO_OType_PP;
   GPIO_InitStructure.GPIO_PuPd = GPIO_PuPd_UP;
   // Both signals are on GPIOA
   GPIO_Init(GPIOA, &GPIO_InitStructure);

   /* Connect PXx to USARTx_Tx* + Rx*/
   GPIO_PinAFConfig(DISCOVERY_COM_TX_GPIO_PORT,
        DISCOVERY_COM_TX_SOURCE, DISCOVERY_COM_TX_AF);
   GPIO_PinAFConfig(DISCOVERY_COM_RX_GPIO_PORT,
        DISCOVERY_COM_RX_SOURCE, DISCOVERY_COM_RX_AF);

   // ---------------- USB ----------------- //
    USBD_Init(&USB_OTG_dev,
              USB_OTG_FS_CORE_ID,
              &USR_desc,
              &USBD_CDC_cb,
              &USR_cb);
}

/*
 * Called from systick handler
 */
void timing_handler()
{
    if (time_var1)
        time_var1--;

    time_var2++;
}
```

```
/*
 * Delay a number of systick cycles (1ms)
 */
void Delay(volatile uint32_t nCount)
{
    time_var1 = nCount;
    while (time_var1)
      ;
}

void dsp_block_ready(uint16_t* adc_in, uint16_t* dac_out)
{
    static uint16_t i = 0;
    static float last_coefficient = 0.0f;
    static float last_power = 0.0f;

    static float x_0_uf = 0.0f;
    static float x_0 = 0.0f;
    static float x_1_uf = 0.0f;
    static float x_1 = 0.0f;
    static float x_0_c = 0.0f;
    static float x_1_c = 0.0f;

    static float last_x_0_uf = 0;
    static float last_x_1_uf = 0;
    static float last_x_0 = 0.0f;
    static float last_x_1 = 0.0f;
    static float last_x_0_c = 0.0f;
    static float last_x_1_c = 0.0f;

    static float last_last_x_0_uf = 0;
    static float last_last_x_1_uf = 0;
    static float last_last_x_0 = 0.0f;
    static float last_last_x_1 = 0.0f;

    static uint32_t lms_settling_time;

    GPIO_ToggleBits(GPIOD, GPIO_Pin_15);

    uint32_t j;

    if(tracking == 3)
    {
        mean_V_int = (int32_t)mean_V;
        mean_directional_int = (int32_t)mean_directional;
    }

    for(j = 0; j < DSP_BLOCK_LENGTH/2; j++)
    {

        last_last_x_0_uf = last_x_0_uf;
        last_last_x_1_uf = last_x_1_uf;
```

```
last_x_0_uf = x_0_uf;
last_x_1_uf = x_1_uf;

x_0 = ((int32_t)adc_in[j*2]   - mean_V)/2;
x_1 = ((int32_t)adc_in[j*2+1] - mean_directional)/2;

x_0_uf = x_0;
x_1_uf = x_1;

x_0 =  0.999280180922122f*(         x_0_uf
                          - 2*last_x_0_uf
                        + last_last_x_0_uf)
          + 1.998559843704671f*last_x_0
     - 0.998560879983815f*last_last_x_0;

x_1 =  0.999280180922122f*(     x_1_uf
                       - 2*last_x_1_uf
                     + last_last_x_1_uf)

              + 1.998559843704671f*last_x_1
        - 0.998560879983815f*last_last_x_1;

x_0_c = x_0;
x_1_c = x_1;

last_last_x_0 = last_x_0;
last_last_x_1 = last_x_1;

last_x_0 = x_0;
last_x_1 = x_1;
last_x_0_c = x_0_c;
last_x_1_c = x_1_c;

if(tracking)
{

    float filter_output = a_1*x_0_c + a_2*last_x_0_c;

    float Vleft = -(x_1_c - filter_output);
    float Vright = (x_1_c + filter_output);

    mean_LR =     mean_LR*0.99999f
        + ( Vleft*Vright)*0.00001f;
    mean_RR =     mean_RR*0.99999f
        + (Vright*Vright)*0.00001f;

    dac_out[j*2] = adc_in[j*2];
    dac_out[j*2+1] = mean_V_int;

    if(tracking == 2)
    {
        float error = ((float)1e-9)*Vleft;
```

```
            a_1 -= (x_0*error);
            a_2 -= (last_x_0*error);

            float gamma = mean_LR/mean_RR;
            if( gamma < 0.01f && gamma > -0.01f)
            {
                tracking = 1;
                lms_settling_time = 32*DSP_BLOCK_LENGTH;
            }
        }
        else if(tracking == 1)
        {
            if(0 == --lms_settling_time)
            {
                float gamma = mean_LR/mean_RR;
                if( gamma < 0.01f && gamma > -0.01f)
                {
                    tracking = 0;

                    a_1_int = a_1*1024;
                    a_2_int = a_2*1024;

                    GPIO_SetBits(GPIO_BANK_RELAY3,
                                 GPIO_PIN_RELAY3);
                    GPIO_SetBits(GPIO_BANK_RELAY4,
                                 GPIO_PIN_RELAY4);

                    power_scaling_factor = 1.0f/mean_RR;
                }
                else
                {
                    tracking = 2;
                }
            }
        }

        dac_out[j*2] =   2048 + ((int32_t)(Vleft*w_1_L ))/1024
                              + ((int32_t)(Vright*w_1_R))/1024;
        dac_out[j*2+1] = 2048 + ((int32_t)(Vleft*w_2_L ))/1024
                              + ((int32_t)(Vright*w_2_R))/1024;
    }
    else
    {
        float filter_output = a_1*x_0_c + a_2*last_x_0_c;

        float Vleft = -(x_1_c - filter_output);
        float Vright = (x_1_c + filter_output);

        mean_LR = mean_LR*0.99999f + (Vleft*Vright)*0.00001f;
        mean_RR = mean_RR*0.99999f + (Vright*Vright)*0.00001f;

        float Va = ((int32_t)(Vleft*w_1_L ))/1024
                 + ((int32_t)(Vright*w_1_R))/1024;
```

```
float Vb = ((int32_t)(Vleft*w_2_L ))/1024
         + ((int32_t)(Vright*w_2_R))/1024;

float Vc = ((int32_t)(Vleft*w_3_L ))/1024
         + ((int32_t)(Vright*w_3_R))/1024;

float Vd = ((int32_t)(Vleft*w_4_L ))/1024
         + ((int32_t)(Vright*w_4_R))/1024;

/* Handle the :MEASURE:POWER command. */
if(power_samples_remaining > 0)
{
    sum_A_squared += Va*Va;
    sum_B_squared += Vb*Vb;
    sum_C_squared += Vc*Vc;
    sum_D_squared += Vd*Vd;

    if(0 >= --power_samples_remaining)
    {
        float power_A =
            ((float)sum_A_squared
                *power_scaling_factor
            )/power_block_size;

        float power_B =
            ((float)sum_B_squared
                *power_scaling_factor
            )/power_block_size;

        float power_C =
            ((float)sum_C_squared
                *power_scaling_factor
            )/power_block_size;

        float power_D =
            ((float)sum_D_squared
                *power_scaling_factor
            )/power_block_size;

        VCP_DataTx(
            (uint8_t*)&power_A,sizeof(power_A));
        VCP_DataTx(
            (uint8_t*)&power_B,sizeof(power_B));
        VCP_DataTx(
            (uint8_t*)&power_C,sizeof(power_C));
        VCP_DataTx(
            (uint8_t*)&power_D,sizeof(power_D));
        VCP_DataTx("\n", 1);

        GPIO_ResetBits(GPIOD, GPIO_Pin_12);
    }
}
```

```
            else
            {
                GPIO_ResetBits(GPIOD, GPIO_Pin_12);
            }

            dac_out[j*2]    = 2048 + Va;
            dac_out[j*2+1]  = 2048 + Vb;
        }
    }
}

int main(void)
{
    hw_init();

    dsp_setup();
    scpi_setup();

    GPIO_SetBits(GPIO_BANK_RELAY1, GPIO_PIN_RELAY1);
    GPIO_SetBits(GPIO_BANK_RELAY2, GPIO_PIN_RELAY2);
    GPIO_ResetBits(GPIO_BANK_RELAY3, GPIO_PIN_RELAY3);
    GPIO_ResetBits(GPIO_BANK_RELAY4, GPIO_PIN_RELAY4);

    uint8_t settling_time = 100;

    while(1)
    {
        if(GPIO_ReadInputDataBit(GPIOA, GPIO_Pin_0))
        {
            tracking = 2;
            mean_LR = 1000.0f;
            mean_RR = 1000.0f;
        }
        GPIO_WriteBit(GPIOD, GPIO_Pin_13, tracking == 2);
        GPIO_WriteBit(GPIOD, GPIO_Pin_14, tracking == 1);

        if(dsp_perform() && settling_time)
        {
            if(0 == --settling_time)
            {
                tracking = 2;
            }
        }
    }

   return 0;
}

//Something the runtime startup is trying to call this
void _init()
{
}
```

B.2.2 Round-trip-time measurement engine

The following sections contain the source code for our round-trip-time measurement setup. The system comprises three parts:

- The *receiver*. This program is responsible for coordinating the measurements, and will receive the timestamps.
- The *repeater*. This program forwards any packets to a predetermined destination, taking note of the time of sending. When it receives a response from the next server in the chain, it adds its own timestamps to the response before passing it back to the previous server in the chain.
- The *final server*. This program waits for packets from a *repeater* or *receiver*, responding with the current time.

Between them, these programs allow a packet to be bounced between a number of servers, and its progress to be tracked along the way.

B.2.2.1 Receiver

```
#!/usr/bin/env python

import socket
import time
import sys
import struct

UDP_IP = '0.0.0.0'
UDP_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
sock.settimeout(5)

#fh = open('E:/Dropbox/Dropbox/ping-uni.txt', 'w')
fh = sys.stdout

for i in range(1000):
    sock.sendto('', ('999.999.999.999', 5005))
    send_time = time.clock()

    try:
        data, addr = sock.recvfrom(16)
        receive_time = time.clock()
        eavesdropper_time = struct.unpack('!d', data)[0]

        print >>fh, '%.5f %.5f %.5f' % (send_time, eavesdropper_time, receive_time)
        sys.stderr.write('.'),
    except:
```

```
        sys.stderr.write('x'),

print ''
fh.close()
```

B.2.2.2 Repeater

```
#!/usr/bin/env python

import socket
import time
import sys
import struct

UDP_IP = '0.0.0.0'
UDP_PORT = 5005

UDP_REPEAT_ID = '999.999.999.999'

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))
sock.settimeout(5)

fh = sys.stdout

while True:
    try:
        data, orig_addr = sock.recvfrom(1024)
        sock.sendto('', (UDP_REPEAT_ID, UDP_PORT))
        send_time = time.clock()

        data, addr = sock.recvfrom(1024)
        receive_time = time.clock()

        send_time_packed = struct.pack('!d', send_time)
        receive_time_packed = struct.pack('!d', receive_time)

        sock.sendto(send_time_packed + data + receive_time_packed, orig_addr)

    except:
        pass
```

B.2.2.3 Final server

```
#!/usr/bin/env python

import socket
import time

import socket
import time

import struct
```

```
UDP_IP = '0.0.0.0'
UDP_PORT = 5005

sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind((UDP_IP, UDP_PORT))

sock.sendto('', ('999.999.999.999', 5005))

while True:
        data, addr = sock.recvfrom(1024)
        sock.sendto(struct.pack('!d', time.time()), addr)
```

B.2.3 Round-trip-time key establishment system

We have implemented a key establishment system based on round-trip times. The system rallies packets back and forth between two machines, measuring their round-trip times and producing a key with a target bit-error-rate.

B.2.3.1 physicallayersecurity.py

```
#!/usr/bin/env python

import socket
import math
import pickle
import numpy
import sys
import time
import os

if os.name == 'nt':
    clockfunc = time.clock
else:
    clockfunc = time.time

def coroutine(func):
    def start(*args,**kwargs):
        cr = func(*args,**kwargs)
        cr.next()
        return cr
    return start

def required_bit_pair_iterations(ber_in, ber_limit):
    """Determine the number of bit-pair iterations are needed for some BER."""

    if ber_in == 0.5 or ber_limit == 0:
        return None

    iterations = 0
    while ber_in > ber_limit:
```

```
        iterations += 1
        ber_in = ber_in**2 / (2*ber_in**2 - 2*ber_in + 1)

    return iterations

@coroutine
def trade_parities_bob(port=5005):
    """A server used for 'data swapping'.

    Usage:

        >> bob = trade_parities_bob(port=5005)
        >> print bob.send('Bob Text')
        Alice Text

    The trade_parities_bob and trade_parities_alice provide a simple network
    protocol---Bob will wait for Alice to send a pickled Python object, and
    when she does so, he will send back one in return.

    The two functions are implemented as coroutines, and will return generator
    objects.  Upon creation, trade_parities_bob will wait for a connection
    from trade_parities_alice on the specified port.

    """
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.setsockopt(socket.SOL_SOCKET, socket.SO_REUSEADDR, 1)
    sock.bind(('0.0.0.0', port))
    sock.listen(1)

    conn, client_addr = sock.accept()

    alice_parities = []

    f = conn.makefile('r+b')

    try:
        while True:
            bob_parities   = (yield alice_parities)

            alice_parities = pickle.load(f)
            pickle.dump(bob_parities, f)
            f.flush()
    except:
        conn.close()
        sock.close()

@coroutine
def trade_parities_alice(bob_addr):
    """A client for 'data swapping'.

    Usage:

        >> alice = trade_parities_bob(('127.0.0.1', 5005))
```

```
        >> print alice.send('Alice Text')
        Bob Text

    The trade_parities_bob and trade_parities_alice provide a simple network
    protocol———Bob will wait for Alice to send a pickled Python object, and
    when she does so, he will send back one in return.

    The two functions are implemented as coroutines, and will return generator
    objects.  Upon creation, trade_parities_alice will attempt to connect to
    an instance of trade_parities_bob at the specified address.

    """
    sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    sock.connect(bob_addr)
    f = sock.makefile('r+b')

    bob_parities = []
    try:
        while True:
            alice_parities = (yield bob_parities)

            pickle.dump(alice_parities, f)
            f.flush()
            bob_parities  = pickle.load(f)
    except:
        sock.close()

def bit_pair_iteration(bits, trade_parities):
    """Perform a round of error correction using bit—pair iteration.

    Usage:

        trade_parities = trade_parities_bob()
        bits = [1,0,1,1,1,1,0,1]
        corrected_bits = bit_pair_iteration(bits, trade_parities)

    The bit_pair_iteration function performs a single iteration of
    error—correction.  The data will be split into pairs, which each of which
    will have a parity bit computed and sent to the other party.

    Each party receives the other's parity bits and returns the first bits
    of each pair that passes the parity check.  The second bit is discarded
    in order that the parity check bits do not provide any information to an
    eavesdropper.

    """
    bits = bits[0:2*(len(bits)/2)]

    our_parities = (bits[0::2] + bits[1::2]) % 2
    their_parities = trade_parities.send(our_parities)

    accepted_bits = ((our_parities + their_parities) % 2)
```

```
    n = len(accepted_bits) + 1.96**2
    parity_ber = float(sum(accepted_bits))/n
    if parity_ber > 0.5:
        parity_ber = 1-parity_ber

    original_ber = (1.0-numpy.sqrt(1-2*parity_ber))/2.0

    bits = bits[::2][0==accepted_bits]

    return bits, original_ber

def information_reconciliation(bits, max_ber, eavesdropper_ber, trade_parities):
    r"""Perform the bit-pair iteration algorithm on an array of bits.

    Usage:

        trade_parities = trade_parities_bob()
        bits = [1,0,1,1,1,1,0,1]
        corrected_bits = information_reconciliation( \
                            bits, 1e-3, 5e-2, trade_parities)

    The information_reconciliation function uses the bit-pair iteration
    to perform error correction over some transport medium without
    revealing any information to an attacker.  The efficiency (ratio of output
    bits to input bits) of the algorithm is dependent upon the BER of the
    underlying channel (lower gives greater efficiency), the desired output BER
    (higher provides greater efficiency), and the lower bound on the
    eavesdropper's BER (higher provides greater efficiency).

    """
    i = 0

    eavesdropper_channel_capacity = \
        1 + eavesdropper_ber*math.log(eavesdropper_ber,2) \
          + (1-eavesdropper_ber)*math.log(1-eavesdropper_ber,2)

    ber_limit = 1-math.pow(1-max_ber, (1-eavesdropper_channel_capacity))

    while True:
        i = i + 1
        bits, ber = bit_pair_iteration(bits, trade_parities)
        est_ber = (ber**2 / (2*ber**2 - 2*ber + 1))
        if est_ber < ber_limit:
            break

    bits_of_output = math.floor((1-eavesdropper_channel_capacity)*len(bits))
    block_size = int(math.ceil(float(len(bits)/bits_of_output)))

    print >>sys.stderr, 'Block size:', block_size

    return [sum(bits[i:i+block_size])%2 for i in range(0,len(bits),block_size) ]

@coroutine
```

```
def information_reconciliation_continuous(
        max_ber, eavesdropper_ber,
        trade_parities, target, layer=None):
    r"""Perform bit-pair iteration and privacy amplification continuously.

    Usage:

        >> output_bits = []
        >> def handle_output_bits():
        >>     while True:
        >>         bits = yield
        >>         for bit in bits:
        >>             output_bits.append(bit)
        >>
        >> trade_parities = trade_parities_bob(port=5005)
        >> output_handler = handle_output_bits()
        >> output_handler.next()
        >> ir = information_reconciliation_continuous( \
        >>      1e-3, 0.05, trade_parities, output_handler)
        >>
        >> while more_bits_are_coming:
        >>     ir.send(get_more_bits())

    This function is a coroutine that will accept an array of bits which are
    then reconciled using bit-pair iteration.  Privacy amplification is then
    used in order to reduce the information gained by eavesdroppers.

    Output is provided to a coroutine passed in upon initialisation in the form
    of an array of bits.  New output data need not be provided every time, and
    indeed will not be produced at first while estimation of the input BER
    occurs.

    The number of iterations and the amount of privacy amplification are
    determined automatically from the estimated channel BER and the lower bound
    for the eavesdropper BER provided.
    """
    eavesdropper_channel_capacity = \
        1 + eavesdropper_ber*math.log(eavesdropper_ber,2) \
          + (1-eavesdropper_ber)*math.log(1-eavesdropper_ber,2)

    ber_limit = 1-math.pow(1-max_ber, (1-eavesdropper_channel_capacity))
    block_size = int(math.ceil(1.0/(1-eavesdropper_channel_capacity)))

    bits = []
    parity_checks = []

    total_errors = 0
    total_bits = 0

    true_target = None
    perform_amplification = False

    while True:
```

```
        new_bits = (yield)

        # Deal with dropped packets——simply remove them.
        if layer == None:
            our_nulls = [ bit == None for bit in new_bits]
            their_nulls = trade_parities.send(our_nulls)
            for i in range(len(new_bits)):
                if not ( their_nulls[i] or our_nulls[i] ):
                    bits = numpy.concatenate([bits, [new_bits[i]]])
        else:
            bits = numpy.concatenate([bits, new_bits])

        bits = numpy.array(bits)

        if len(parity_checks)*2 + 1 < len(bits):
            start_idx = len(parity_checks)*2
            bits_left = bits[start_idx::2]
            bits_right = bits[start_idx+1::2]

            if len(bits_left) > len(bits_right):
                bits_left = bits_left[:-1]

            our_parity = (bits_left + bits_right) % 2
            their_parity = trade_parities.send(our_parity)

            this_parity_check = (our_parity + their_parity) % 2
            errors = (numpy.array(this_parity_check) != 0).sum()

            parity_checks = numpy.concatenate([parity_checks, this_parity_check])

            if layer == None:
                total_bits += len(this_parity_check)
                total_errors += errors

                p_ber = (total_errors+2)/(total_bits+4.0)
                p_ber_ci = 2.0*math.sqrt(p_ber*(1-p_ber)/(total_bits+9.0))

                if p_ber + p_ber_ci < 0.5:
                    input_ber_max = 0.5*(1-math.sqrt(1-2*(p_ber + p_ber_ci)))
                else:
                    input_ber_max = 0.5

                if p_ber - p_ber_ci < 0.5:
                    input_ber_min = 0.5*(1-math.sqrt(1-2*(p_ber - p_ber_ci)))
                else:
                    input_ber_min = 0.0

                if input_ber_max > 0.5:
                    input_ber_max = 0.5
                if input_ber_min < 0:
                    input_ber_min = 0.0

                required_layers_max = required_bit_pair_iterations(
```

```
                                input_ber_max, ber_limit)
            required_layers_min = required_bit_pair_iterations(
                                input_ber_min, ber_limit)


        # Whether we emit or continue depends upon the BER.
        if (layer == None and true_target == None and
                    required_layers_min != None and
                    required_layers_max != None):
            print >>sys.stderr, ('      Channel BER confidence interval: (%.2e,%.2e)'
                           % (input_ber_min, input_ber_max))
            print >>sys.stderr, ('Information reconciliation iterations: %d'
                           % (required_layers_max))
            print >>sys.stderr, (' Block size for privacy amplification: %d'
                           % (block_size))
            if required_layers_max == 0 and required_layers_min == 0:
                true_target = target
                perform_amplification = True
            elif abs(float(required_layers_max) -
                        float(required_layers_min)) <= 1:
                true_target = information_reconciliation_continuous(
                    max_ber, eavesdropper_ber, trade_parities,
                    target, required_layers_max-2)

        elif layer != None and true_target == None:
            if layer == 0:
                true_target = target
                perform_amplification = True
            else:
                true_target = information_reconciliation_continuous(
                    max_ber, eavesdropper_ber, trade_parities, target,layer-1)

        if true_target != None:
            outgoing = bits[1::2][parity_checks == 0]

            if not perform_amplification:
                bits = bits[2*len(parity_checks):]
                parity_checks = []
                true_target.send(outgoing)
            else:
                output_bits = []
                while len(outgoing) > block_size:
                    output_bits.append(outgoing[:block_size].sum() % 2)
                    parity_checks = parity_checks[block_size:]
                    bits = bits[2*block_size:]
                    outgoing = outgoing[block_size:]
                if len(output_bits) > 0:
                    true_target.send(output_bits)

@coroutine
def measure_rtt_server(bind_address):
    """A server to measure round-trip times across the internet.
```

```
    Usage:
        >> server = measure_rtt_server(('0.0.0.0', 5005))
        >> round_trip_time = server.next()

    This function produces a generator whose values are the round-trip times
    between this computer and another.  This being the server, it will wait for
    the client to transmit a packet before responding in kind.  The time until
    the client's response is then measured and returned.
    """
    sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
    sock.bind(bind_address)

    sock.settimeout(None)
    rtt = None

    try:
        while True:
            yield rtt

            data, addr = sock.recvfrom(16)
            start_time = clockfunc()
            sock.sendto(data, addr)

            sock.settimeout(2)
            try:
                data2, addr2 = sock.recvfrom(16)
            except socket.timeout:
                rtt = None
                continue

            end_time = clockfunc()

            if data2 == data:
                rtt = end_time - start_time
            else:
                rtt = None

    except StopIteration:
        sock.close()

@coroutine
def measure_rtt_client(bob_address, bind_address):
    """A client to measure round-trip times across the internet.

    Usage:
        >> client = measure_rtt_client(('frankfurt.twopif.net', 5005))
        >> round_trip_time = client.next()

    This function produces a generator whose values are the round-trip times
    between this computer and another.  This being the client, it will transmit
    a packet, and then upon receiving a response it will transmit another packet
    to the server before returning the measured round-trip time.
    """
```

```
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(bind_address)

sock.settimeout(2)
rtt = None

try:
    while True:
        yield rtt

        sock.sendto('',  bob_address)
        start_time = clockfunc()

        try:
            data, addr = sock.recvfrom(16)
        except socket.timeout:
            rtt = None
            continue

        end_time = clockfunc()
        sock.sendto(data, addr)

        rtt = end_time - start_time

except StopIteration:
    sock.close()
```

Appendix C

The Allison Mixture

Modelling of stochastic systems is vital, as we are generally more interested in fundamental properties of the system at hand than individual measurements. In the early stages of this research, we undertook this study of the Allison mixture, a mixture process that originally proposed as a model for the linguistic properties of long texts. The Allison mixture displays the counter-intuitive property of displaying self-dependence despite its values being drawn from processes without any self-dependence. We investigate this phenomenon and the conditions under which it occurs.

C.1 LINEAR STATISTICS OF THE ALLISON MIXTURE

The independence of the samples of a stochastic process appears to be a feature that cannot easily be destroyed. However, it has been suggested by Allison et al. (2007) that a random mixture of two random sequences of digits can give rise to a resultant sequence with a nonzero autocovariance. Epstein (2009) has called this process the *Allison mixture*.

This phenomenon is not dissimilar to *Parrondo's paradox* (Harmer and Abbott 1999; Abbott 2010), in which games of chance that are individually biased against the player can be mixed in order to achieve a gain overall.

Parrondo's paradox and thermodynamics are closely related; indeed, the former has its genesis in the theory of Brownian ratchets (Abbott 2010). The flashing ratchet (Adjari and Prost 1993; Hänggi and Marchesoni 2005) transports particles using Brownian motion by repeated alternation of a potential. The potential has the appearance of a sawtooth, allowing net drift in one direction, but when switched on and off the particles drift in the opposite direction.

A related phenomenon in finance is *volatility pumping* (Luenberger 1998) whereby one regularly rebalances a portfolio to maintain a 50:50 split between, say, a volatile stock and a mediocre low-performing stock. It is surprising that it results in a theoretical exponential growth in capital (Abbott 2010). Volatility pumping is the result of an asymmetry that rectifies fluctuations in the market, which means it is acting as a type of Brownian ratchet. The action of maintaining the 50:50 portfolio split guarantees that we are always buying low and selling high, which is a form of ratcheting asymmetry (Abbott 2010).

The original motivation for the development of the Allison mixture is from the statistical modelling of language (Allison et al. 2007). Here, the word repetition intervals are modelled as a Poisson process whose rate parameter occasionally switches between a higher and a lower value according to whether the word is a topic of discussion.

C.1.1 THE ALLISON MIXTURE

The Allison mixture is a random process formed by the sampling of a pair of base processes, and is defined as follows:

Definition C.1. The Allison mixture

Let U_t and V_t be a pair of white, stationary, and independent random processes, and S_t be a binary random process whose values form a two-state Markov chain, as in Figure C.1, without

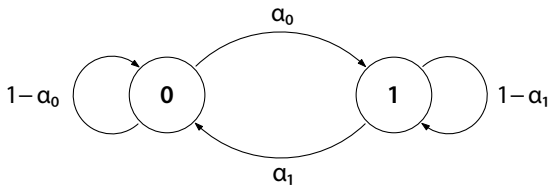


Figure C.1: The Markov chain defining the sampling process S_t of the Allison mixture. It is parametrised by the probabilities α_0 and α_1 of leaving states 0 and 1 respectively. When in state one, its value is equal to that of the first process U, and when in state two to that of the second V.

absorbing states. The Allison mixture is a random process X_t such that

$$X_t = U_t S_t + V_t(1 - S_t). \tag{C.1}$$

That is, $X_t = U_t$ when the Markov chain is in state one, and $X_t = V_t$ when in state two.

The behaviour of the Allison mixture is determined by its sampling process, a Markov chain such as in Figure C.1. The parameters α_1 and α_2 determine the rate at which the mixture will switch between processes. The requirement that the Markov chain have no absorbing states excludes the cases where $\alpha_1 = 0$ or $\alpha_2 = 0$, in which states one and two respectively are absorbing.

Theorem C.1. The autocovariance function of the Allison mixture

The Allison mixture X_t has autocovariance function

$$R_{XX}(\tau) = (E[U] - E[V])^2 R_{SS}(\tau). \tag{C.2}$$

Proof. The autocovariance function of X is defined by Papoulis (1991, p. 289) as

$$R_{XX}(\tau) = E[X_t X_{t+\tau}] - E[X_t]E[X_{t+\tau}]. \tag{C.3}$$

Our first step in evaluating this function is to consider the product

$$\begin{aligned} X_t X_{t+\tau} = {} & U_t U_{t+\tau} S_t S_{t+\tau} \\ & + V_t V_{t+k}(1 - S_t)(1 - S_{t+\tau}) \\ & + U_t V_{t+k} S_t (1 - S_{t+\tau}) \\ & + V_t U_{t+k}(1 - S_t) S_{t+\tau}, \end{aligned} \tag{C.4}$$

and its expectation, which by the independence, whiteness, and stationarity of U and V is equal to

$$\begin{aligned} E[X_t X_{t+\tau}] &= E[U_t]^2 E[S_t S_{t+\tau}] + E[V_t]^2 \left(1 - 2E[S_t] + E[S_t S_{t+\tau}]\right) && \text{(C.5)} \\ &\quad + 2E[U_t]E[V_t]\left(E[S_t] - E[S_t S_{t+\tau}]\right). \\ &= E[U_t]^2 (R_{SS}(\tau) + E[S_t]^2) && \text{(C.6)} \\ &\quad + E[V_t]^2 (R_{SS}(\tau) + (1 - E[S_t])^2) \\ &\quad + 2E[U_t]E[V_t]\left(E[S_t](1 - E[S_t]) - R_{SS}(\tau)\right). \end{aligned}$$

Similarly, we may write

$$\begin{aligned} E[X_t] = E[X_{t+\tau}] &= E[U_t S_t + V_t(1 - S_t)] && \text{(C.7)} \\ &= E[U_t]E[S_t] + E[V_t](1 - E[S_t]), && \text{(C.8)} \end{aligned}$$

and so

$$\begin{aligned} R_{XX}(\tau) &= E[X_t X_{t+\tau}] - E[X_t]E[X_{t+\tau}] && \text{(C.9)} \\ &= E[U_t]^2 (R_{SS}(\tau) + E[S_t]^2) && \text{(C.10)} \\ &\quad + E[V_t]^2 (R_{SS}(\tau) + (1 - E[S_t])^2) \\ &\quad + 2E[U_t]E[V_t]\left(E[S_t](1 - E[S_t]) - R_{SS}(\tau)\right). \\ &\quad - \left(E[U_t]E[S_t] + E[V_t](1 - E[S_t])\right)^2 \\ &= E[U_t]^2 R_{SS}(\tau) + E[V_t]^2 R_{SS}(\tau) + 2E[U_t]E[V_t]R_{SS}(\tau) && \text{(C.11)} \\ &= (E[U] - E[V])^2 R_{SS}(\tau). && \text{(C.12)} \end{aligned}$$

That is to say, the autocovariance of the sampling process is scaled by the squared difference of means.

■

We now focus our attention on the Allison mixture as defined above; we determine the single-step autocovariance of the sampling process before applying the scaling factor given by (C.2).

Theorem C.2. The lag-one autocovariance of the Allison mixture

The Allison mixture X_t associated with a fully mixed sampling process S_t as in Figure C.1 has a lag-one autocovariance of

$$R_{XX}(1) = (E[U] - E[V])^2 \frac{\alpha_1 \alpha_2}{(\alpha_1 + \alpha_2)^2}(1 - \alpha_1 - \alpha_2). \qquad \text{(C.13)}$$

Proof. We begin by analysing the Markov process S_t with structure given by Figure C.1. As it is fully mixed, the distribution of the states is equal to the stationary distribution (Goodman 2006) π_i of the chain. These may be calculated as

$$\pi_1 = \frac{\alpha_2}{\alpha_1 + \alpha_2} \tag{C.14}$$

$$\pi_2 = \frac{\alpha_1}{\alpha_1 + \alpha_2}. \tag{C.15}$$

We may then write

$$E[S_t] = \pi_2 \tag{C.16}$$

$$E[S_t S_{t+1}] = \sum_{s_t, s_{t+1}} S_t S_{t+1} P[S_t = s_t \cap S_{t+1} = s_{t+1}] \tag{C.17}$$

$$= P[S_t = 1 \cap S_{t+1} = 1] \tag{C.18}$$

$$= P[S_{t+1} = 1 | S_t = 1]\pi_2 \tag{C.19}$$

$$= (1 - \alpha_2)\pi_2 \tag{C.20}$$

leading us to its lag-one autocovariance

$$R_{SS}(1) = E[S_t S_{t+1}] - E[S_t]E[S_{t+1}] \tag{C.21}$$

$$= \pi_2(1 - \alpha_2 - \pi_2) \tag{C.22}$$

$$= \pi_1\pi_2(1 - \alpha_1 - \alpha_2) \tag{C.23}$$

$$= \frac{\alpha_1\alpha_2}{(\alpha_1 + \alpha_2)^2}(1 - \alpha_1 - \alpha_2). \tag{C.24}$$

We substitute this into (C.2), resulting in autocovariance

$$R_{XX}(1) = (E[U] - E[V])^2 \frac{\alpha_1\alpha_2}{(\alpha_1 + \alpha_2)^2}(1 - \alpha_1 - \alpha_2) \tag{C.25}$$

as originally stated. ■

For simplicity, if we put $\mu_1 = E[U]$ and $\mu_2 = E[V]$, we can write,

$$R_{XX}(1) = (\mu_1 - \mu_2)^2 \frac{\alpha_1\alpha_2}{(\alpha_1 + \alpha_2)^2}(1 - \alpha_1 - \alpha_2) \tag{C.26}$$

From this result we may trivially extract the necessary conditions for correlation.

Corollary C.1. *Consecutive samples of the Allison mixture are correlated if and only if all of the following are true:*

$$\alpha_1 \neq 0 \tag{C.27}$$

$$\alpha_2 \neq 0 \tag{C.28}$$

$$\alpha_1 + \alpha_2 \neq 1 \tag{C.29}$$

$$\mu_1 \neq \mu_2. \tag{C.30}$$

This result might at first seem counterintuitive, as the processes from which the elements are drawn exhibit no time dependence. This phenomenon can be explained by imagining two processes U_t and V_t with very different means. Then, from the value of X_t one may determine S_t with reasonable certainty. If the switching probability of the current state is small, then one would expect X_{t+1} to be drawn from the same process, and so be similar in value to X_t. This is the source of the correlation between subsequent values.

C.1.2 NUMERICAL RESULTS

The relationship between the probabilities α_1 and α_2 and the autocorrelation coefficient $\rho = R_{XX}(1)/\operatorname{Var}(X)$ is shown in Figure C.2.

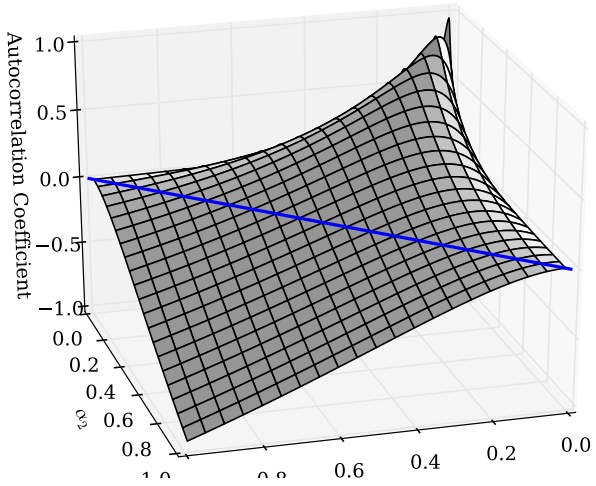
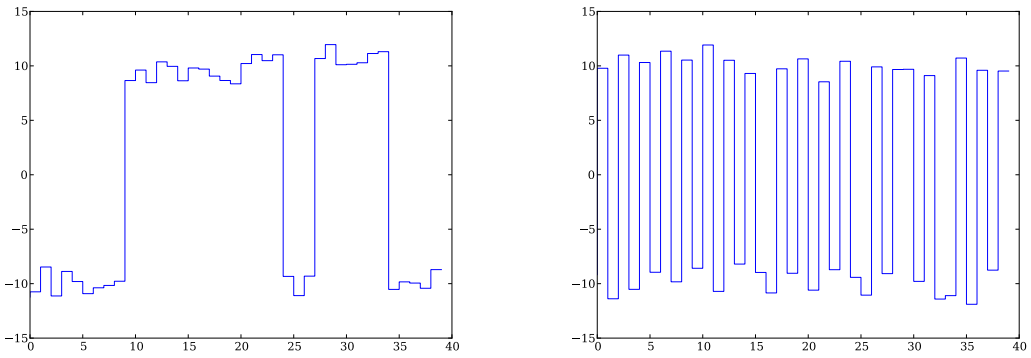


Figure C.2: The autocorrelation coefficient of the Allison mixture of $N(-10, 1)$ and $N(10, 1)$ for various values of α_1 and α_2. The thick line shows the case $\alpha_1 + \alpha_2 = 1$ in which the autocorrelation coefficient is zero.

We show the results of simulation in Figure C.3. When the parameters α are small, the process switches states only rarely and has a large positive autocovariance. When the switching probabilities are large, the process will switch almost every time, causing it to flit back and forth between input processes and so have a large negative autocovariance.



(a) $\alpha_1 = \alpha_2 = 0.1$ (b) $\alpha_1 = \alpha_2 = 0.9$

Figure C.3: The Allison mixture of $N(-10, 1)$ and $N(10, 1)$ with varying parameters α_i. In (a) the low probability of switching causes the process to stay with its current input for long periods of time. The autocorrelation coefficient is large and positive. Conversely, in (a) the probability of switching is high, causing the sampling operation to flit between the two processes almost every cycle. The autocovariance is large and negative.

C.2 Information-Theoretic Analysis of the Allison Mixture

This unintuitive feature of the Allison mixture results in the appearance of autocorrelation, despite all of its values being drawn from uncorrelated processes. However, as we have shown, this correlation vanishes if the parent processes are of equal mean, suggesting the use of autoinformation (Darbellay and Wuertz 2000; Chapeau-Blondeau 2007) as an alternative to correlation; this provides a canonical measure of the strength of the memory of the Allison mixture. We apply this measure to the binary-valued Allison mixture, producing analytic expressions for the k-step autoinformation of its sampling process.

C.2.1 The Allison Mixture

The Allison mixture draws its samples from one of two distributions, the choice determined by the state of a Markov chain (Goodman 2006). The marginal distribution of this process is a mixture of the two source distributions, the mixing constant determined by the stationary distribution of the Markov chain.

We use a spectral decomposition of the transition matrix P in order to compute the k-step probability matrix P^k and so the k-step transition probabilities $\alpha_{0,k}$ and $\alpha_{1,k}$.

Theorem C.3. *The sampling process S_t has k-step transition probabilities*

$$\alpha_{0,k} = \pi_1 \left[1 - (1 - \alpha_0 - \alpha_1)^k\right] \tag{C.31}$$

$$\alpha_{1,k} = \pi_0 \left[1 - (1 - \alpha_0 - \alpha_1)^k\right]. \tag{C.32}$$

Proof. S_t has transition matrix

$$\mathbb{P} = \begin{bmatrix} 1 - \alpha_0 & \alpha_1 \\ \alpha_0 & 1 - \alpha_1 \end{bmatrix} \tag{C.33}$$

with spectral decomposition

$$\mathbb{P} = \frac{1}{\alpha_0 + \alpha_1} \begin{bmatrix} \alpha_1 & 1 \\ \alpha_0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 - \alpha_0 - \alpha_1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \alpha_0 & -\alpha_1 \end{bmatrix}. \tag{C.34}$$

We therefore find that

$$\mathbb{P}^k = \frac{1}{\alpha_0 + \alpha_1} \begin{bmatrix} \alpha_1 & 1 \\ \alpha_0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & (1 - \alpha_0 - \alpha_1)^k \end{bmatrix} \begin{bmatrix} 1 & 1 \\ \alpha_0 & -\alpha_1 \end{bmatrix} \tag{C.35}$$

$$= \frac{1}{\alpha_0 + \alpha_1} \begin{bmatrix} \alpha_1 + \alpha_0(1 - \alpha_0 - \alpha_1)^k & \alpha_1 \left(1 - (1 - \alpha_0 - \alpha_1)\right) \\ \alpha_0 \left(1 - (1 - \alpha_0 - \alpha_1)\right) & \alpha_0 + \alpha_1(1 - \alpha_0 - \alpha_1)^k \end{bmatrix} \tag{C.36}$$

$$= \begin{bmatrix} \pi_0 + \pi_1(1 - \alpha_0 - \alpha_1)^k & \pi_0 \left[1 - (1 - \alpha_0 - \alpha_1)^k\right] \\ \pi_1 \left[1 - (1 - \alpha_0 - \alpha_1)^k\right] & \pi_1 + \pi_0(1 - \alpha_0 - \alpha_1)^k \end{bmatrix}, \tag{C.37}$$

and read off the stated transition probabilities from the minor diagonal. ■

Knowing this, we may now calculate the autocorrelation function of the process at arbitrary time-lags.

Theorem C.4. *The Allison mixture X_t has k-step autocovariance*

$$R_{XX}[k] = R_{XX}[1] \left(1 - \alpha_0 - \alpha_1\right)^{k-1}. \tag{C.38}$$

Proof. We begin by noting that as a decimated Markov chain—that is to say, one where all but every k-th step is discarded—is still a Markov chain, and that therefore a decimated Allison mixture is also an Allison mixture. We may therefore calculate arbitrary two-point statistics by simply substituting the k-step transition probabilities from Theorem C.3 for α_0 and α_1.

Let $\gamma = (1 - \alpha_0 - \alpha_1)$. Then, performing the substitution as described,

$$R_{XX}[k] = (\mu_0 - \mu_1)^2 \frac{\alpha_{0,k}\alpha_{1,k}}{(\alpha_{0,k} + \alpha_{1,k})^2}(1 - \alpha_{0,k} - \alpha_{1,k}) \tag{C.39}$$

$$= (\mu_0 - \mu_1)^2 \, \pi_0\pi_1\gamma^k \tag{C.40}$$

$$= (\mu_0 - \mu_1)^2 \frac{\alpha_0\alpha_1}{(\alpha_0 + \alpha_1)^2}(1 - \alpha_0 - \alpha_1)^k \tag{C.41}$$

$$= R_{XX}[1]\,(1 - \alpha_0 - \alpha_1)^{k-1} \tag{C.42}$$

as originally stated. ■

C.2.2 AUTOINFORMATION OF THE ALLISON MIXTURE SAMPLING PROCESS

The autoinformation function is an alternative to the autocovariance function as a measure of dependence—though not causality, which is to be the subject of a future paper—and is defined as follows:

Definition C.2 (Autoinformation function (Chapeau-Blondeau 2007))**.** *The autoinformation function of a stochastic process S_t is the mutual information (Cover and Thomas 2006)*

$$I_{SS}[t,k] = I(S_t, S_{t-k}) \tag{C.43}$$

$$= H(S_t) + H(S_{t-k}) - H(S_t, S_{t-k}). \tag{C.44}$$

If S_t is stationary, then we may omit t as a parameter, leaving us with

$$I_{SS}[k] = I(S_t, S_{t-k}) \tag{C.45}$$

$$= 2H(S_t) - H(S_t, S_{t-k}). \tag{C.46}$$

The autoinformation improves on the autocovariance function as a measure of dependence by providing a condition both sufficient and necessary—whereas a lack of correlation does not necessarily indicate independence, two variables will have zero mutual information if and only if they are statistically independent; this is vital when the processes U_t and V_t of the system being modelled have identical means but differing variances or skew, such as would occur when sampling particle velocities in statistical mechanics.

Substituting the stationary and transition probabilities into the entropy, we find the single-step autoinformation, stated without further detail in the following lemma.

Lemma C.1. *Let S_t be a binary-valued random process with transition probabilities and a stationary distribution equal to that of the Markov chain in Definition C.1. Then, in the fully-mixed regime—that is to say, when the state probability distribution is equal to the stationary*

distribution of the Markov chain—the single-step autoinformation $I_{SS}[1]$ is given by

$$I_{SS}[1] = \frac{\alpha_1(1-\alpha_0)\log_2 \frac{1-\alpha_0}{\alpha_1}}{\alpha_0+\alpha_1} + \frac{\alpha_0(1-\alpha_1)\log_2 \frac{1-\alpha_1}{\alpha_0}}{\alpha_0+\alpha_1} + \log_2(\alpha_0+\alpha_1), \tag{C.47}$$

where both α_0 and α_1 are nonzero, zero if exactly one of α_0 and α_1 is equal to zero, and undefined if both are equal to zero.

Thus the autoinformation $I_{SS}[1]$ is equal to zero when either $\alpha_0 = 0$, $\alpha_1 = 0$, or $\alpha_0 + \alpha_1 = 1$, and so these three previously-described conditions for decorrelation of the sampling process imply zero mutual information and therefore genuine independence.

Importantly, we have not assumed the Markov property of S_t, instead directly demanding that the formulae for the stationary probabilities hold. This weakening is intended to allow us later to generalise to the Allison mixture proper.

The mutual information $I_{SS}[1]$ as a function of (α_0, α_1) is shown in Figure C.4. As one would expect, we see a peak near $(\alpha_0, \alpha_1) = (0, 0)$, where consecutive states are highly dependent. Similarly, we see a large autoinformation $I_{SS}[1]$ near $(1, 1)$, where the strong anticorrelation makes consecutive states highly predictable. Between these two extremes lies a valley, its nadir falling along the line $\alpha_0 + \alpha_1 = 1$; along this line, consecutive states of the sampling process are completely independent.

Importantly, these results can be generalised to allow calculation of the autoinformation at arbitrary time-lags, shown in Theorem C.5 by substituting the k-step probabilities of the Allison mixture sampling process.

Theorem C.5. *The k-step autoinformation of a fully mixed two-state Markov chain with exit probabilities α_0 and α_1, as in Figure C.1, is given by Lemma C.1 under the substitution*

$$\alpha_0 \longrightarrow \pi_1\left[1-(1-\alpha_0-\alpha_1)^k\right] \tag{C.48}$$

$$\alpha_1 \longrightarrow \pi_0\left[1-(1-\alpha_0-\alpha_1)^k\right]. \tag{C.49}$$

Proof. By substituting the k-step transition probabilities, calculated in Equations C.31 and C.32, in place of the single-step probabilities α_0 and α_1, Equation C.47 yields the k-step autoinformation $I_{SS}[k]$ rather than the single-step autoinformation $I_{SS}[1]$. ■

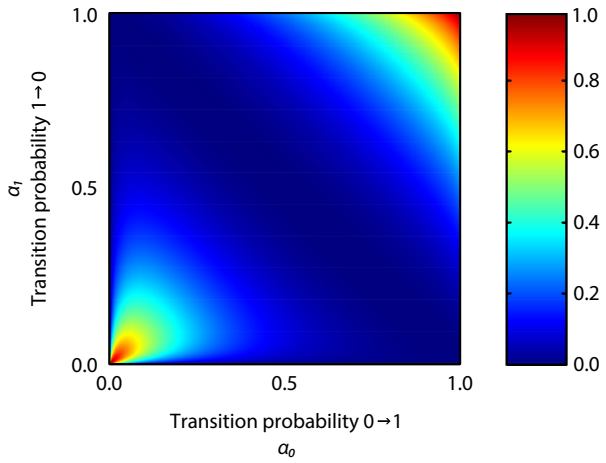


Figure C.4: Single-step autoinformation $I_{SS}[1]$ of the Allison mixture sampling process S_t as a function of the transition probabilities α_0 and α_1, calculated according to Equation C.47. Note the lines of zero autoinformation along $\alpha_0 = 0$, $\alpha_1 = 0$, and $\alpha_0 + \alpha_1 = 1$.

We show the autoinformation $I_{SS}[k]$ in Figure C.5 as a function of the lag k; it can be seen to decay at a roughly exponential rate.

C.2.3 OPEN QUESTIONS

The theorems that we have presented allow computation of the autoinformation function of the Allison mixture sampling process S_t, and can be readily extended to binary-valued Allison mixtures, that is to say those for which X_t takes only two values; the input processes U_t and V_t might each take a single distinct value, or perhaps a common pair of values. However, many physical systems are described by continuous-valued processes, and their autoinformation cannot be calculated by Lemma C.1. It remains to be seen whether the autoinformation $I_{XX}[k]$ of the Allison mixture X_t can be computed by transformation of the sampling process autoinformation $I_{SS}[k]$ in a similar fashion to that of the autocovariance function, potentially yielding a more practically-manipulated alternative to the rather cumbersome formulae that can be derived by manual calculation of mixture transition probabilities to be substituted into Equation C.47. There exists also the possibility that simpifying approximations will be possible in the large-k regime to allow further comparison of its properties with those of the autocorrelation $R_{XX}[k]$.

Furthermore, the information-theoretic approach that we have presented provides the starting point for an investigation of the transfer entropy (Schreiber 2000) between the sampling process and the Allison mixture; previous works on transfer entropy have focussed

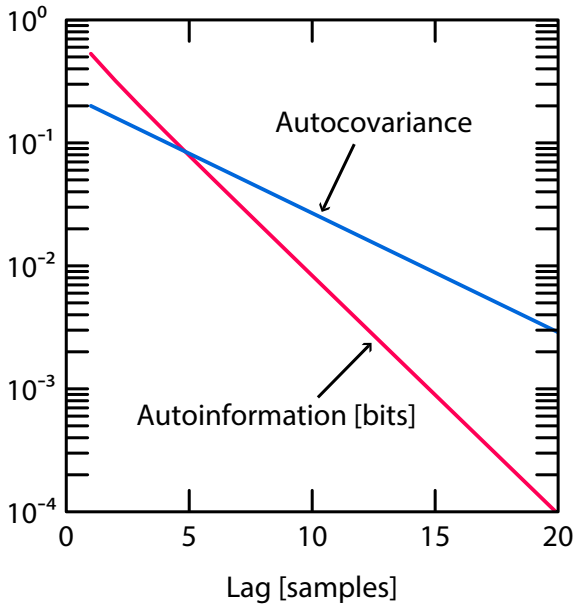


Figure C.5: The exponentially-decaying autoinformation $I_{SS}[k]$ and autocovariance $R_{SS}[k]$ of an Allison mixture sampling process with $\alpha_0 = 0.1, \alpha_1 = 0.1$. The slope of the autoinformation line is approximately double that of the autocorrelation line; the results of Chapeau-Blondeau (2007) hint that this may be exactly so asymptotically.

on complex systems, leaving room for the analysis of simpler and analytically tractable models in order to better probe its properties.

Bibliography

A single quantum cannot be cloned (1982). *Nature* 299, pp. 802–803. DOI: 10.1038/299802a0.

Abbott, D. (2010). Asymmetry and disorder: A decade of Parrondo's paradox. *Fluctuation and Noise Letters* 9(1), pp. 129–156. DOI: 10.1142/S0219477510000010.

Adams, C., P. Cain, D. Pinkas, and R. Zuccherato (RFC 3161, 2001). *Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)*.

Adjari, A. and J. Prost (1993). Drift induced by a periodic potential of low symmetry: pulsed dielectrophoresis. *Comptes rendus de l'Academie des Sciences, Série II* 315, pp. 1635–1693.

Adkins, H. (2011). An update on attempted man-in-the-middle attacks. *Google Security Blog*. Accessed 2016-04-07. URL: https://security.googleblog.com/2011/08/update-on-attempted-man-in-middle.html.

Advanced Micro Devices (2007). *AMD Opteron processor product datasheet*. http://support.amd.com/TechDocs/23932.pdf. accessed 2015-10-12.

Agresti, A. and B. A. Coull (1998). Approximate is better than 'exact' for interval estimation of binomial proportions. *The American Statistician* 52(2), pp. 119–126. DOI: 10.1080/00031305.1998.10480550.

Alegria, F., P. Arpaia, A. da Cruz Serra, and P. Daponte (2001). "ADC histogram test by triangular small-waves". *Proc. 18th IEEE Instrumentation and Measurement Technology Conference*. Vol. 3, pp. 1690–1695. DOI: 10.1109/IMTC.2001.929490.

Alicherry, M. and A. D. Keromytis (2009). "Doublecheck: multi-path verification against man-in-the-middle attacks". *IEEE Symposium on Computers and Communications*. Sousse, Tunisia, pp. 557–563. DOI: 10.1109/ISCC.2009.5202224.

Allison, A. G., C. E. M. Pearce, and D. Abbott (2007). "Finding keywords amongst noise: automatic text classification without parsing". *Proc. SPIE Noise and Stochastics in Complex Systems and Finance*. Florence, Italy. DOI: 10.1117/12.724655.

Aristotle (1944). *Politics*. section II.I.5, translated by H. Rackham. Heinemann, London.

Backes, M., A. Kate, P. Manoharan, S. Meiser, and E. Mohammadi (2013). "AnoA: A framework for analyzing anonymous communication protocols". *2013 IEEE 26th Computer Security Foundations Symposium*. DOI: 10.1109/csf.2013.18.

Bai, E.-W. (2002). A blind approach to the Hammerstein-Wiener model identification. *Automatica* 38(6), pp. 967–979. DOI: 10.1016/s0005-1098(01)00292-8.

Baseline Requirements Certificate Policy for the Issuance and Management of Publicly-Trusted Certificates, version 1.3.3 (2016). *CA/Browser Forum*.

Bennett, C. H. and G. Brassard (1984). "Quantum cryptography: public key distribution and coin tossing". *Proc. IEEE Int. Conf. Computers, Systems, and Signal Processing*. Bangalore, India, pp. 175–179.

Bennett, C. H. and C. J. Riedel (2013). On the security of key distribution based on Johnson-Nyquist noise. arXiv:1303.7435v1.

Bennett, C. H., G. Brassard, and J.-M. Robert (1988). Privacy amplification by public discussion. *SIAM Journal on Computing* 17(2), pp. 210–229. DOI: 10.1137/0217014.

Bernstein, D. J. (2005). "The Poly1305-AES Message-Authentication Code". *Proceedings of Fast Software Encryption*. Paris, France.

— (2009). "Introduction to post-quantum cryptography". *Post-Quantum Cryptography*. Springer Berlin Heidelberg, pp. 1–14. DOI: 10.1007/978-3-540-88702-7_1.

Blair, J. (1994). Histogram measurement of ADC nonlinearities using sine waves. *IEEE Transactions on Instrumentation and Measurement* 43(3), pp. 373–383. DOI: 10.1109/19.293454.

Boak, D. G. (1973). *A History of US Communications Security*. Volume 1. National Security Agency.

— (1981). *A History of US Communications Security*. Volume 2. National Security Agency.

Bolot, J.-C. (1993). "End-to-end packet delay and loss behavior in the internet". *Proceedings of the ACM SIGCOMM '93 Conference on Communications Architectures, Protocols and Applications*. San Francisco, CA.

Boneh, D. (1999). Twenty years of attacks on the RSA cryptosystem. *Notices of the AMS* 46, pp. 203–213.

Bonnecaze, A., P. Liardet, A. Gabillon, and K. Blibech (2006). Secure time-stamping schemes: a distributed point of view. *Annales des Télécommunications* 61(5), pp. 662–681.

Brillouin, L. (1960). *Wave Propagation and Group Velocity*. Academic Press.

Buhmann, M. D. (2000). Radial basis functions. *Acta Numerica* 9, pp. 1–38. DOI: null.

CAcert (2017). *History of risks & threat events to CAs and PKI*. URL: http://wiki.cacert.org/Risk/History.

Cachin, C., K. Kursawe, and V. Shoup (2000). "Random oracles in Constantipole: practical asynchronous Byzantine agreement using cryptography". *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing*. Portland, Oregon, USA, pp. 123–132. DOI: 10.1145/343477.343531.

Callas, J., L. Donnerhacke, H. Finney, D. Shaw, and R. Thayer (RFC 4880, 2007). *OpenPGP Message Format*.

Camenisch, J. and A. Lysyanskaya (2005). "A formal treatment of onion routing". *Proceedings of CRYPTO*. Santa Barbara, CA, pp. 169–187.

Carback, R. et al. (2010). "Scantegrity II municipal election at Takoma Park: the first E2E binding governmental election with ballot privacy". *Proceedings of the 19th USENIX Security Symposium*. Washington DC.

Chapeau-Blondeau, F. (2007). Autocorrelation versus entropy-based autoinformation for measuring dependence in random signal. *Physica A* 380, pp. 1–18. DOI: 10.1016/j.physa.2007.02.077.

Chappell, J. M., S. P. Drake, C. L. Seidel, L. J. Gunn, A. Iqbal, A. Allison, and D. Abbott (2014). Geometric algebra for electrical and electronic engineers. *Proceedings of the IEEE* 102(9), pp. 1340–1363. DOI: 10.1109/jproc.2014.2339299.

Chappell, J. M., L. J. Gunn, and D. Abbott (2013). The double-padlock problem: is secure classical information transmission possible without key exchange? Presented at Hot Topics in Physical Informatics. DOI: 10.1142/S201019451460355X.

Chaum, D. L. (1981). Untraceable electronic mail, return addresses, and digital pseudonyms. *Communications of the ACM* 24(2), pp. 84–90. DOI: 10.1145/358549.358563.

Chen, H.-P., L. B. Kish, C.-G. Granqvist, and G. Schmera (2014a). Do electromagnetic waves exist in a short cable at low frequencies? What does physics say? *Fluctuation and Noise Letters* 13(02). Art. 1450016. DOI: 10.1142/s0219477514500163.

— (2014b). On the 'cracking' scheme in the paper 'A directional coupler attack against the Kish key distribution system' by Gunn, Allison, and Abbott. *Metrology and Measurement Systems* 21(3), pp. 389–400.

Clark, J. and U. Hengartner (2010). "On the use of financial data as a random beacon". *Proceedings of the 2010 International Conference on Electronic voting Technology/Workshop on Trustworthy Elections*. Washington DC.

Cohen, H., G. Frey, R. Avanzi, C. Doche, T. Lange, K. Nguyen, and F. Vercauteren (2005). *Handbook of Elliptic and Hyperelliptic Curve Cryptography*. Chapman and Hall/CRC, Boca Raten, USA.

Cooper, D., S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk (RFC 5280, 2008). *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*.

Cover, T. M. and J. A. Thomas (2006). *Elements of Information Theory*. 2nd. Wiley, Hoboken, New Jersey.

Darbellay, G. A. and D. Wuertz (2000). The entropy as a tool for analysing statistical dependences in financial time series. *Physica A* 287, pp. 429–439. DOI: https://doi.org/10.1016/S0378-4371(00)00382-4.

Delbridge, A., ed. (2001). *Macquarie Dictionary : Australia's National Dictionary*. 3rd ed. Macmillan.

Devlin, P., C. Freeman, J. Hutchinson, and P. Knights (1976). *Report to the Secretary of State for the Home Department of the Departmental Committee on Evidence of Identification in Criminal Cases*. HMSO, London.

Diffie, W. and M. E. Hellman (1976). New directions in cryptography. *IEEE Transactions on Information Theory* 22(6), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

Dinaberg, A. (2011). "Bitsquatting: DNS hijacking without exploitation". *Proceedings of Black-Hat Security*. Las Vegas, USA.

Dingledine, R., N. Mathewson, and P. Syverson (2004). "Tor: the second generation onion router". *Proceedings of the 13th USENIX Security Symposium*. San Diego, CA.

Dinovitser, A., L. J. Gunn, and D. Abbott (2015). Towards quantitative atmospheric water vapor profiling with differential absorption lidar. *Optics Express* 23(17), pp. 22907–22921. DOI: 10.1364/OE.23.022907.

Dixon, A. R. et al. (2017). Quantum key distribution with hacking countermeasures and long term field trial. *Scientific Reports* 7(1). DOI: 10.1038/s41598-017-01884-0.

Doernberg, J., H. S. Lee, and D. A. Hodges (1984). Full-speed testing of A/D converters. *IEEE Journal of Solid-State Circuits* 19(6), pp. 820–827. DOI: 10.1109/JSSC.1984.1052232.

Douceur, J. R. (2002). "The Sybil attack". *Proceedings of the International Workshop on Peer-to-Peer Systems*. Cambridge, USA, pp. 251–260. DOI: 10.1007/3-540-45748-8_24.

Dworkin, M. (2007). *Special Publication 800-38D: Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*. Tech. rep. Gaithersburg, MD, United States.

Elahi, T., K. Bauer, M. AlSabah, R. Dingledine, and I. Goldberg (2012). Changing of the guards: a framework for understanding and improving entry guard selection in tor, pp. 43–54. DOI: 10.1145/2381966.2381973.

Engert, K. (2013). *DetecTor.io*. accessed 2017-02-20. URL: http://detector.io/.

Epstein, I., ed. (1961). *The Babylonian Talmud*. Soncino Press, London.

Epstein, R. A. (2009). *The Theory of Gambling and Statistical Logic*. 2nd ed. Academic Press.

Ferguson, N., B. Schneier, and T. Kohno (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley, Indianapolis, USA.

Foster, R. A., T. M. Libkuman, J. W. Schooler, and E. F. Loftus (1994). Consequentiality and eyewitness person identification. *Applied Cognitive Psychology* 8(2), pp. 107–121. DOI: 10.1002/acp.2350080203.

FOX-IT (2012). *Report of the Investigation into the DigiNotar Certificate Authority Breach*.

Gilad, Y. and A. Herzberg (2013). "Plug-and-play IP security". Ed. by J. Crampton, S. Jajodia, and K. Mayes. Egham, UK, pp. 255–272. DOI: 10.1007/978-3-642-40203-6_15.

Gingl, Z. and R. Mingesz (2014). Noise properties in the ideal Kirchhoff-Law-Johnson-Noise secure communication system. *PLoS ONE* 9(4), e96109. DOI: 10.1371/journal.pone.0096109.

Goodman, R. (2006). *Introduction to Stochastic Models*. Dover, New York.

Google. *Chrome for Work and Education Help: Manage Chrome updates on Windows*. https://-support.google.com/chrome/a/answer/3204698?hl=en.

Goulet, D. and G. Kadianakis (2015). *Tor specification proposal 250, random number generation during Tor voting*. https://gitweb.torproject.org/torspec.git/tree/proposals/250-commit-reveal-consensus.txt.

Griffiths, D. J. (2005). *Introduction to Quantum Mechanics*. 2nd. Prentice Hall.

Gunn, L. J., A. Allison, and D. Abbott (2013). Identification of static distortion by noise measurement. *Electronics Letters* 49(21), pp. 1321–1323. DOI: 10.1049/el.2013.2547.

— (2014a). A directional wave measurement attack against the Kish key distribution system. *Scientific Reports* 4. Art. 6461. DOI: 10.1038/srep06461.

— (2014b). Allison mixtures: where random digits obey thermodynamic principles. *International Journal of Modern Physics* 33. Presented at Hot Topics in Physical Informatics 2013. DOI: 10.1142/S2010194514603603.

— (2015a). "Real-time compensation of static distortion by measurement of differential noise gain". *Proc. IEEE Workshop on Signal Processing Systems*. Belfast, United Kingdom. DOI: 10.1109/SiPS.2014.6986079.

— (2015b). A new transient attack on the Kish key distribution system. *IEEE Access* 3, pp. 1640–1648. DOI: 10.1109/access.2015.2480422.

— (2016a). Verifying public keys without trust: how anonymity can guarantee data integrity. ArXiV:1602.03316v1.

— (2017). "Safety in numbers: anonymization makes keyservers trustworthy". *10th Workshop on Hot Topics in Privacy Enhancing Technologies*. Minneapolis, USA.

Gunn, L. J., P. G. Catlow, W. A. Al-Ashwal, J. G. Hartnett, A. Allison, and D. Abbott. Simplified three-cornered-hat technique for frequency stability measurements. *IEEE Transactions*

on Instrumentation and Measurement 63(4), pp. 889–895. DOI: 10.1109/tim.2013.2285796.

Gunn, L. J., F. Chapeau-Blondeau, A. Allison, and D. Abbott (2016b). "An output-only nonlinear system identification technique suited to integer arithmetic". *Proceedings of the 2016 IEEE Conference on Norbert Wiener in the 21st Century*. Melbourne, Australia. DOI: 10.1109/NORBERT.2016.7547454.

— (2016c). Towards an information-theoretic model of the allison mixture stochastic process. *Journal of Statistical Mechanics: Theory and Experiment* 2016(5). DOI: 10.1088/1742-5468/2016/05/054041.

Gunn, L. J., F. Chapeau-Blondeau, M. D. McDonnell, B. R. Davis, A. Allison, and D. Abbott (2016d). Too good to be true: when overwhelming evidence fails to convince. *Proceedings of the Royal Society A* 472(2187). DOI: 10.1098/rspa.2015.0748.

Gunn, L. J., J. M. Chappell, A. Allison, and D. Abbott (2014c). Physical-layer encryption on the public internet: a stochastic approach to the Kish-Sethuraman cipher. *International Journal of Modern Physics: Conference Series* 33. Presented at HotPI-2013. DOI: 10.1142/S2010194514603615.

Günther, C. G. (1989). "An identity-based key-exchange protocol". *Proceedings of EUROCRYPT '89*. Houthalen, Belgium, pp. 29–37. DOI: 10.1007/3-540-46885-4_5.

Hallgren, S. and U. Vollmer (2009). *Quantum computing*. DOI: 10.1007/978-3-540-88702-7_1.

Hänggi, P. and F. Marchesoni (2005). Introduction: 100 years of Brownian motion. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 15(2). DOI: 10.1063/1.1895505.

Hao, F. (2006). Kish's key exchange scheme is insecure. *IEE Proceedings—Information Security* 153(1) (4), pp. 141–142. DOI: 10.1049/ip-ifs:20060068.

Harmer, G. P. and D. Abbott (1999). Game theory: losing strategies can win by Parrondo's paradox. *Nature* 402, p. 864. DOI: 10.1038/47220.

Haykin, S. (2002). *Adaptive Filter Theory*. 4th. Prentice Hall.

Headlam, J. (1891). *Election by Lot at Athens*. Cambridge University Press.

IEEE Standard for Digitizing Waveform Recorders (2008). *IEEE Std 1057-2007 (Revision of IEEE 1057-1994)*. DOI: 10.1109/IEEESTD.2008.4494996.

Jackson, J. D. (1999). *Classical Electrodynamics*. 3rd. Wiley.

Kahn, D. (1966). *The Codebreakers*. London: Weidenfeld and Nicolson.

Kalodner, H., M. Carlsten, P. Ellenbogen, J. Bonneau, and A. Narayanan (2015). "An empirical study of Namecoin and lessons for decentralized namespace design". *Proceedings of Workshop on the Economics of Information Security*. Delft, The Netherlands.

Katz, J. (2010). *Digital Signatures*. Springer, New York. DOI: 10.1007/978-0-387-27712-7.

Kazimierczuk, M. K. (2013). *High-Frequency Magnetic Components*. Wiley.

Keybase (2016). https://keybase.io. accessed 2016-04-07.

Kim, B., S.-e. Lee, K. Kim, J. W. Yoon, and S.-H. Han (2015). Pulse-induction metal detector with time-domain bucking circuit for landmine detection. *Electronics Letters* 51(2), pp. 159–161. DOI: 10.1049/el.2014.3895.

Kim, Y., R. Daly, J. Kim, C. Fallin, J. H. Lee, D. Lee, C. Wilkerson, K. Lai, and O. Mutlu (2014). "Flipping bits in memory without accessing them: an experimental study of DRAM disturbance errors". *Proc. IEEE 41st Annual International Symposium on Computer Architecuture*. Minneapolis, MN, USA, pp. 361–372. DOI: 10.1145/2678373.2665726.

Kish, L. B. (2006a). Protection against the man-in-the-middle-attack for the Kirchhoff-loop-Johnson(-like)-noise cipher and expansion by voltage-based security. *Fluctuation and Noise Letters* 6(01), pp. L57–L63. DOI: 10.1142/S0219477506003148.

— (2006b). Response to Feng Hao's paper "Kish's key exchange scheme is insecure". *Fluctuation and Noise Letters* 6(04), pp. C37–C41. DOI: 10.1142/S021947750600363X.

— (2006c). Totally secure classical communication utilizing Johnson (-like) noise and Kirchoff's law. *Physics Letters A* 352(3), pp. 178–182. DOI: 10.1016/j.physleta.2005.11.062.

— (2006d). Response to Scheuer-Yariv: 'A classical key-distribution system based on Johnson (like) noise—how secure?' *Physics Letters A* 359(6), pp. 741–744. DOI: 10.1016/j.physleta.2006.07.037.

— (2013). Enhanced secure key exchange systems based on the Johnson-Noise scheme. *Metrology and Measurement Systems* 20(2), pp. 191–204. DOI: 10.2478/mms-2013-0017.

Kish, L. B., D. Abbott, and C.-G. Granqvist (2013). Critical analysis of the Bennett-Riedel attack on secure cryptographic key distributions via the Kirchhoff-law-Johnson-noise scheme. *PLOSONE* 8(12). Art. e81810. DOI: 10.1371/journal.pone.0081810.

Kish, L. B., Z. Gingl, R. Mingesz, G. Vadai, J. Smulko, and C.-G. Granqvist (2015). Analysis of an attenuator artifact in an experimental attack by Gunn-Allison-Abbott against the Kirchhoff-Law-Johnson-Noise (KLJN) secure key exchange system. *Fluctuation and Noise Letters* 14(1). Art. 1550011. DOI: 10.1142/S021947751550011X.

Kish, L. B. and C.-G. Granqvist (2014a). Elimination of a second-law-attack, and all cable-resistance-based attacks, in the Kirchhoff-law-Johnson-noise (KLJN) secure key exchange system. *Entropy*. DOI: 10.3390/e16105223.

Kish, L. B. and C.-G. Granqvist (2014b). On the security of the Kirchhoff-law–Johnson-noise (KLJN) communicator. *Quantum Information Processing* 13(10), pp. 2213–2219. DOI: 10.1007/s11128-014-0729-7.

Kish, L. B. and T. Horvath (2009). Notes on recent approaches concerning the Kirchhoff-law–Johnson-noise-based secure key exchange. *Physics Letters A* 373(32), pp. 2858–2868. DOI: 10.1016/j.physleta.2009.05.077.

Kish, L. B. and S. Sethuraman (2004). Non-breakable data encryption with classical information. *Fluctuations and Noise Letters* 4(2), pp. C1–C5. DOI: 10.1142/s0219477504001987.

Kogias, E. K., P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford (2016). "Enhancing bitcoin security and performance with strong consistency via collective signing". *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX, pp. 279–296.

Lamport, L. (1979). *Constructing Digital Signatures from a One Way Function*. Tech. rep. SRI International. URL: https://www.microsoft.com/en-us/research/publication/constructing-digital-signatures-one-way-function/.

Lamport, L., R. Shostak, and M. Pease (1982). The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems* 4(3), pp. 382–401. DOI: 10.1145/357172.357176.

Larsen, R. J. and M. L. Marx (2012). *An Introduction to Mathematical Statistics and Its Applications*. Pearson.

Laurie, B. (2014). Certificate transparency. *Queue* 12(8), 10:10–10:19. DOI: 10.1145/2668152.2668154.

Leung-Yan-Cheong, S. and M. E. Hellman (1978). The gaussian wire-tap channel. *IEEE Transactions on Information Theory* 24(4), pp. 451–456. DOI: 10.1109/tit.1978.1055917.

Leyden, J. (2012). The one tiny slip that put LulzSec chief Sabu in the FBI's pocket. *The Register*.

Liu, S., H. C. A. Van Tilborg, and M. Van Dijk (2003). A practical protocol for advantage distillation and information reconciliation. *Designs, Codes and Cryptography* 30(1), pp. 39–62. DOI: 10.1023/A:1024755209150.

Ljung, L. (1987). *System Identification: Theory for the User*. Prentice Hall.

Luenberger, D. S. (1998). *Investment Science*. Oxford University Press.

Lydersen, L., C. Wiechers, C. Wittmann, D. Elser, J. Skaar, and V. Makarov (2010). Hacking commercial quantum cryptography systems by tailored bright illumination. *Nature Photonics* 4(10), pp. 686–689. DOI: 10.1038/nphoton.2010.214.

Lynch, N. A. (1996). *Distributed Algorithms*. Morgan Kauffman, San Francisco.

Malpass, R. S. and P. G. Devine (1981). Eyewitness identification: lineup instructions and the absence of the offender. *Journal of Applied Psychology* 66(4), pp. 482–489. DOI: 10.1037/0021-9010.66.4.482.

Martins, R. C. and A. M. da Cruz Serra (1999). Automated ADC characterization using the histogram test stimulated by Gaussian noise. *IEEE Transactions on Instrumentation and Measurement* 48(2), pp. 471–474. DOI: 10.1109/19.769631.

Mathur, S., W. Trappe, N. Mandayam, C. Ye, and A. Reznik (2008). "Radio-telepathy". *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking – MobiCom '08*. DOI: 10.1145/1409944.1409960.

Maurer, U. M. (1993). Secret key agreement by public discussion from common information. *IEEE Transactions on Information Theory* 39(3), pp. 733–742. DOI: 10.1109/18.256484.

Melara, M. S., A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman (2015). "CONIKS: bringing key transparency to end users". *Proceedings of the 24th USENIX Security Symposium*. Washington, D.C., pp. 383–398.

Menezes, A. J., P. C. van Oorschot, and S. A. Vanstone (1996/2001). *Handbook of Applied Cryptography*. CRC Press, Boca Raton, USA.

Mingesz, R., Z. Gingl, and L. B. Kish (2008). Johnson(-like)-noise-Kirchhoff-loop based secure classical communicator characteristics, for ranges of two to two thousand kilometers, via model-line. *Physics Letters A* 372(7), pp. 978–984. DOI: 10.1016/j.physleta.2007.07.086.

Nakamoto, S. (2008). Bitcoin: a peer-to-peer electronic cash system. Accessed 2016-11-08. URL: https://bitcoin.org/bitcoin.pdf.

Nielsen, M. A. and I. L. Chuang (2000). *Quantum Computation and Quantum Information*. Cambridge University Press. DOI: 10.2277/0521635039.

Odlyzko, A. (2000). Discrete logarithms: the past and the future. *Designs, Codes, and Cryptography* 19(2). DOI: 10.1007/978-1-4757-6856-5_3.

Oxford English Dictionary (2015). accessed 2015-10-06. Oxford University Press.

Papoulis, A. (1991). *Probability, Random Variables, and Stochastic Processes*. 3rd ed. McGraw-Hill.

Pfitzmann, A. and M. Köhntopp (2000). "Anonymity, unobservability, and pseudonymity—a proposal for terminology". *Proceedings of International Workshop on Design Issues in Anonymity and Unobservability*. Berkeley, CA. DOI: 10.1007/3-540-44702-4_1.

Planck, M. and M. Masius (1914). *The Theory of Heat Radiation*. 2nd. P. Blackiston's Son and Co.

Pomerance, C. (1996). A tale of two sieves. *Notices of the AMS* 43, pp. 1473–1485.

Pozar, D. M. (1998). *Microwave Engineering*. Wiley.

Rabin, M. O. (1983). Transaction protection by beacons. *Journal of Computer and System Sciences* 27(2), pp. 256–267. DOI: 10.1016/0022-0000(83)90042-9.

Rivera, J., M. Carrillo, M. Chacón, G. Herrera, and G. Bojorquez (2007). Self-calibration and optimal response in intelligent sensors design based on artificial neural networks. *Sensors* 7(8), pp. 1509–1529. DOI: 10.3390/s7081509.

Rivera, J., G. Herrera, M. Chacón, P. Acosta, and M. Carrillo (2008). Improved progressive polynomial algorithm for self-adjustment and optimal response in intelligent sensors. *Sensors* 8(11), pp. 7410–7427. DOI: 10.3390/s8117410.

Rivest, R. L., A. Shamir, and L. Adleman (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM* 21(2), pp. 120–126. DOI: 10.1145/359340.359342.

Ruoti, S., J. Andersen, D. Zappala, and K. Seamons (2015). Why Johnny still, still can't encrypt: evaluating the usability of a modern PGP client. ArXiV:1510.08555. URL: https://arxiv.org/abs/1510.08555.

Scheuer, J. and A. Yariv (2006). A classical key-distribution system based on Johnson (like) noise—how secure? *Physics Letters A* 359(6), pp. 737–740. DOI: 10.1016/j.physleta.2006.07.013.

Schneier, B. (1996). *Applied Cryptography*. 2nd ed. Wiley, Indianapolis. DOI: 10.1002/9781119183471.

Schreiber, T. (2000). Measuring information transfer. *Physical Review Letters* 85, pp. 461–464. DOI: 10.1103/PhysRevLett.85.461.

Schroeder, B., E. Pinheiro, and W.-D. Weber (2009). "DRAM errors in the wild: A large-scale field study". *Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems*. Seattle, WA, USA, pp. 193–204. DOI: 10.1145/1555349.1555372.

Seber, G. A. F. and A. J. Lee (2003). *Linear Regression Analysis*. 2nd ed. Wiley.

Self, D. (2010). *Small Signal Audio Design*. Elsevier, Oxford, United Kindom.

Shannon, C. E. (1945). *A Mathematical Theory of Cryptography*.

Shaw, D. (2003). *The OpenPGP HTTP Keyserver Protocol (HKP)*. Internet Draft.

Sheng, S., L. Broderick, C. A. Koranda, and J. J. Hyland (2006). "Why Johnny still can't encrypt: evaluating the usability of email encryption software". *Proceedings of the Symposium on Usable Privacy and Security*. Pittsburgh, USA, pp. 3–4.

Shor, P. (1994). "Algorithms for quantum computation: discrete logarithms and factoring". *Proc. IEEE 35th Annual Symposium on Foundations Computer Science*. Santa Fe, NM, pp. 124–134. DOI: 10.1109/SFCS.1994.365700.

Sivia, D. S. and J. Skilling (2006). *Data Analysis: A Bayesian Tutorial*. Oxford University Press.

Sommerfeld, A. (1952). *Electrodynamics*. New York: Academic Press.

Spencer, B. D. (2007). Estimating the accuracy of jury verdicts. *Journal of Empirical Legal Studies* 4(2), pp. 305–329. DOI: 10.1111/j.1740-1461.2007.00090.x.

Stevens, M., A. Sotirov, J. Appelbaum, A. Lenstra, D. Molnar, D. A. Osvik, and B. de Weger (2009). "Short chosen-prefix collisions for MD5 and the creation of a rogue CA certificate". *Proceedings of CRYPTO*. Santa Barbara, CA, pp. 55–69. DOI: 10.1007/978-3-642-03356-8_4.

Sutin, A. M. and D. M. Donskoy (1998). "Vibro-acoustic modulation nondestructive evaluation technique". *Nondestructive Evaluation of Aging Aircraft, Airports, and Aerospace Hardware II*. San Antonio, USA. DOI: 10.1117/12.305057.

Syta, E., I. Tamas, D. Visher, D. I. Wolinsky, and B. Ford (2015a). "Certificate Cothority: towards trustworthy collective CAs". *Hot Topics in Privacy Enhancing Technologies*. Philadelphia, PA.

Syta, E., I. Tamas, D. Visher, D. I. Wolinsky, P. Jovanovic, L. Gasser, N. Gailly, I. Khoffi, and B. Ford (2015b). Keeping authorities "honest or bust" with decentralized witness cosigning. ArXiV:1503.08768.

The Tor Project. *Tormetrics—relays with Exit, Fast, Guard, Stable, and HSDir flags*. https://-metrics.torproject.org/relayflags.html. accessed 2016-01-02.

— (2015). *'torrc' manual page, v0.2.4.27*.

Times Microwave (2014). *LMR-600 datasheet*. accessed 2015-05-10.

Truesdell, C. and W. Noll (2004). *The Non-Linear Field Theories of Mechanics*. Springer.

Trusted Time Stamp Management and Security (2005). *ANSI X9.95-2005*.

Volterra, V. (1887). Sopra le funzioni che dipendono da altre funzioni. *Atti della Reale Accademia dei Lincei, Serie Quarta* 3(2), pp. 97–105.

Voss, H. U., H. Rust, W. Horbelt, and J. Timmer (2003). A combined approach for the identification of continuous non-linear systems. *International Journal of Adaptive Control and Signal Processing* 17(5), pp. 335–352. DOI: 10.1002/acs.750.

Vukolić, M. (2016). "The quest for scalable blockchain fabric: proof-of-work vs. BFT replication". *Open Problems in Network Security*. Zurich, Switzerland, pp. 112–125. ISBN: 978-3-319-39028-4. DOI: 10.1007/978-3-319-39028-4_9.

Wegman, M. N. and J. L. Carter (1981). New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences* 22(3), pp. 265–279. DOI: 10.1016/0022-0000(81)90033-7.

Wellstead, P. E. (1981). Non-parametric methods of system identification. *Automatica* 17(1), pp. 55–69. DOI: 10.1016/0005-1098(81)90084-4.

Wendlandt, D., D. Andersen, and A. Perrig (2008). "Perspectives: improving SSH-style host authentication with multi-path probing". *Proceedings of the USENIX Annual Technical Conference*. Boston, MA.

WhatsApp encryption overview: technical white paper (2016). https://www.whatsapp.com/-security/WhatsApp-Security-Whitepaper.pdf. accessed 2016-11-08.

Whitten, A. and J. D. Tygar (1999). "Why Johnny can't encrypt: a usability evaluation of PGP 5.0". *Proceedings of the 8th Conference on USENIX Security Symposium - Volume 8*. SSYM'99. Washington, D.C. URL: http://dl.acm.org/citation.cfm?id=1251421.1251435.

Winter, P., R. Ensafi, K. Loesing, and N. Feamster (2016). Identifying and characterizing Sybils in the Tor network. ArXiV:1602.07787v1.

Wogalter, M. S. and D. B. Marwitz (1992). Suggestiveness in photospread lineups: similarity induces distinctiveness. *Applied Cognitive Psychology* 6(5), pp. 443–452. DOI: 10.1002/acp.2350060508.

Wright, M. K., M. Adler, B. N. Levine, and C. Shields (2004). The predecessor attack: an analysis of a threat to anonymous communications systems. *ACM Transactions on Information and System Security* 7(4), pp. 489–522. ISSN: 1094-9224. DOI: 10.1145/1042031.1042032.

Wyner, A. D. (1975). The wire-tap channel. *The Bell System Technical Journal* 54(8), pp. 1355–1387.

Xu, L., T. Vladusich, F. Duan, L. J. Gunn, D. Abbott, and M. D. McDonnell (2015). Decoding suprathreshold stochastic resonance with optimal weights. *Physics Letters A* 379(38), pp. 2277–2283. DOI: 10.1016/j.physleta.2015.05.032.

Ylonen, T. and C. Lonvick (RFC 4251, 2006). *The Secure Shell (SSH) Protocol Architecture*.

Young, A. L. and M. Yung (2014). "The Drunk Motorcyclist Protocol for anonymous communication". *Proceedings of IEEE Conference on Communications and Network Security*. San Francisco, CA, pp. 157–165. DOI: 10.1109/CNS.2014.6997482.

Zhang, J. and A. Moore (2007). "Traffic trace artifacts due to monitoring via port mirroring". *2007 Workshop on End-to-End Monitoring Techniques and Services*. DOI: 10.1109/e2emon.2007.375317.

Zhang, J., T. Q. Q. Duong, A. Marshall, and R. Woods (2016a). Key generation from wireless channels: a review. *IEEE Access* 4, pp. 614–626. DOI: 10.1109/access.2016.2521718.

Zhang, J., R. Woods, T. Q. Q. Duong, A. Marshall, and Y. Ding (2016b). "Experimental study on channel reciprocity in wireless key generation". *2016 IEEE 17th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. Edinburgh, UK. DOI: 10.1109/spawc.2016.7536825.

Acronyms

ADC Analog-to-Digital Converter. 34, 41, 52, 71, 148

BER Bit-Error Rate. 62, 64, 65, 73, 162

CA Certificate Authority. 145
CONIKS Continuous Identity and Key Management System. xxiii, 110, 111, 117, 126, 130, 133

DAC Digital-to-Analog Converter. 41
DAQ Data Aquisition Unit. 36
DC Direct Current. 5, 33, 36
DNS Domain Name System. 29

ECC Elliptic-Curve Cryptography. 6

GCM Galois/Counter Mode. 5

IIR Infinite Impulse Response. 49
IP Internet Protocol. 10, 111, 124, 128
IR Information Reconciliation. 64, 65

KKD Kish Key Distribution. 8, 66–68, 72, 77–79, 82–84, 86, 89, 94, 98, 149, 151, 159, 160, 162

MITM Man-in-the-Middle. 10

PGP Pretty Good Privacy. 9, 105, 116, 130, 131
PKI Public-Key Infrastructure. 3, 9, 102, 103, 105, 106, 135

QKD Quantum Key Distribution. 7, 8, 66, 98

RSA Rivest-Shamir-Adleman cryptosystem. 6, 9, 54, 55, 57, 58

SNR Signal-to-Noise Ratio. 70
SSH Secure Shell. 10, 102

TEM Transverse Electric/Magnetic. 68
THD Total Harmonic Distortion. 32, 36, 37, 43, 45, 49, 51
TLS Transport Layer Security. 111, 126, 145

UDP User Datagram Protocol. 63, 64, 126

Index

Biography

Lachlan J. Gunn received his B.Eng. (Hons) and B.Ma. and Comp. Sc. (Pure) degrees from the University of Adelaide, Australia in 2012, receiving the 2012 J. Mazumdar Prize in Engineering and Mathematics, and four DSTO Scholarships in Radar Technology in the 2009–2012 period.

In 2013 he was granted an Australian Postgraduate Award (APA), and is currently undertaking a Ph.D. under Derek Abbott and Andrew Allison. In 2014 he was awarded an Endeavour Research Fellowship by the Australian Government in order to undertake research into stochastic phenomena at the University of Angers.

He has served as reviewer for a number of journals and conferences, including *IEEE Access*, *PLOS ONE*, *Fluctuation and Noise Letters*, *Scientific Reports*, and *IEEE SPT-IOT*.

His research interests include usable security, information-theoretic security, and the use of stochastic signal processing for characterisation of nonlinear systems.

Lachlan James Gunn
lachlan.gunn@adelaide.edu.au
lachlan@twopif.net

ORCID: 0000-0003-1767-7897
PGP: F3E3 8891 8560 5B82 933D 6180 D288 91D2 136D 33B0

Academic Genealogy