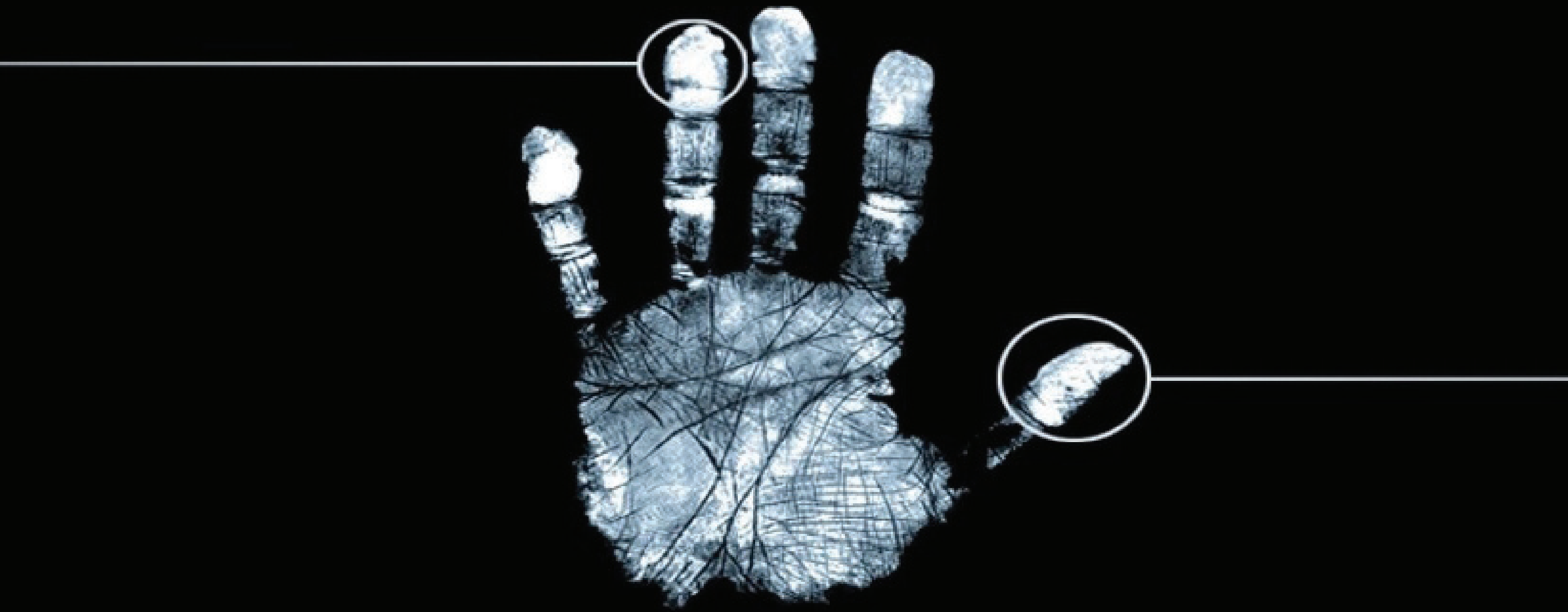
ISSN 1991-1696

Africa Research Journal

Research Journal of the South African Institute of Electrical Engineers

Incorporating the SAIEE Transactions



June 2018
Volume 109 No. 2

www.saiee.org.za

SAIEE AFRICA RESEARCH JOURNAL

(SAIEE FOUNDED JUNE 1909 INCORPORATED DECEMBER 1909)
AN OFFICIAL JOURNAL OF THE INSTITUTE
ISSN 1991-1696

Secretary and Head Office
Mrs Gerda Geyer
South African Institute for Electrical Engineers (SAIEE)
PO Box 751253, Gardenview, 2047, South Africa
Tel: +27-11-487-3003
E-mail: researchjournal@saiee.org.za

Published by
South African Institute of Electrical Engineers (Pty) Ltd, PO Box 751253, Gardenview, 2047
Tel. +27-11-487-3003
E-mail: researchjournal@saiee.org.za
Website: www.saiee.org.za | www.saiee.org.za/arj

Additional reviewers are approached as necessary
ARTICLES SUBMITTED TO THE SAIEE AFRICA RESEARCH JOURNAL ARE FULLY PEER REVIEWED PRIOR TO ACCEPTANCE FOR PUBLICATION
This journal is accredited by the Department of Higher Education and Training, South Africa.

VOL 109 No 2
June 2018

SAIEE Africa Research Journal

ISSN 1991-1696
SAIEE
Africa Research Journal
Research Journal of the South African Institute of Electrical Engineers
Incorporating the SAIEE Transactions
www.saiee.org.za
June 2018
Volume 109 No. 2

GUEST EDITORIAL

INFORMATION SECURITY SOUTH AFRICA (ISSA) 2017

This special issue of the SAIEE Africa Research Journal is devoted to selected papers from the Information Security South Africa (ISSA) 2017 Conference which was held in Johannesburg, South Africa from 16 to17 August 2017. The aim of the annual ISSA conference is to afford information security practitioners and researchers, from all over the globe, an opportunity to share their knowledge and research results with their peers. The 2017 Conference focused on a wide spectrum of aspects in the information security domain including the functional, business, managerial, human, theoretical and technological aspects of modern-day information security.

With the assistance of the original reviewers, eight conference papers that had received good overall reviews were identified. I attended the presentation of each of these papers and based on the reviewer reports and the presentations, six of these papers were selected for possible publication in this special issue. The authors of these six selected papers were invited to rework their papers by expanding and/or further formalising the research conducted. Five of these papers were submitted and subsequently reviewed again by a minimum of three reputable international subject specialists. These reviews were received to make a confident decision as to the inclusion of these papers in the special issue.

After the review process was completed, including attending to the reviewers' suggestions, only four papers were selected to be published in this special issue. These four papers cover various aspects of information security which include: Developing an electromagnetic noise generator to protect a Raspberry Pi from side channel analysis; guidelines for ethical nudging in password authentication; forensic attribution implications for NoSQL databases; and a SEADM model for social engineering attack detection. Therefore, this special issue includes four rather diverse papers in the discipline of information security, providing a true reflection of the multidisciplinary nature of this field of study.

I would like to thank the expert reviewers who diligently reviewed these papers. These reviews certainly contributed to the quality of this special issue.

To conclude, I would like to express my appreciation to IEEE Xplore who originally published the ISSA conference papers, and for granting permission for these reworked papers to be published in this special issue.

Prof. Stephen V. Flowerday
Guest Editor
Rhodes University

DEVELOPING AN ELECTROMAGNETIC NOISE GENERATOR TO PROTECT A RASPBERRY PI FROM SIDE CHANNEL ANALYSIS

I. Frieslaar and B. Irwin*

* Rhodes University, Department of Computer Science, Grahamstown, South Africa E-mail: ebie099@gmail.com and b.irwin@ru.ac.za

Abstract: This research investigates the Electromagnetic (EM) side channel leakage of a Raspberry Pi 2 B+. An evaluation is performed on the EM leakage as the device executes the AES-128 cryptographic algorithm contained in the libcrypto++ library in a threaded environment. Four multi-threaded implementations are evaluated. These implementations are Portable Operating System Interface Threads, C++11 threads, Threading Building Blocks, and OpenMP threads. It is demonstrated that the various thread techniques have distinct variations in frequency and shape as EM emanations are leaked from the Raspberry Pi. It is demonstrated that the AES-128 cryptographic implementation within the libcrypto++ library on a Raspberry Pi is vulnerable to Side Channel Analysis (SCA) attacks. The cryptographic process was seen visibly within the EM spectrum and the data for this process was extracted where digital filtering techniques was applied to the signal. The resultant data was utilised in the Differential Electromagnetic Analysis (DEMA) attack and the results revealed 16 sub-keys that are required to recover the full AES-128 secret key. Based on this discovery, this research introduced a multi-threading approach with the utilisation of Secure Hash Algorithm (SHA) to serve as a software based countermeasure to mitigate SCA attacks. The proposed countermeasure known as the FRIES noise generator executed as a Daemon and generated EM noise that was able to hide the cryptographic implementations and prevent the DEMA attack and other statistical analysis.

Key words: AES-128, Electromagnetic, Noise Generator, Raspberry Pi, Side Channel Analysis, SHA, Software Countermeasure.

1. INTRODUCTION

The volume of data that is produced daily is growing exponentially. Much of this is potentially sensitive, and the challenging aspect relating to this is the security of this information both at the time of use and longer term. The growth of the Internet of Things (IoT) in parallel has further contributed to the amount of data produced. These IoT devices are not only situated in the homes, but in businesses, manufacturing and as part of smart cities. Examples of these interconnected systems are smart grids, where the distribution of electricity [1], natural gas [2] and water are controlled by these IoT devices in real time. In addition, our homes have many sensors that are increasingly being utilised to monitor regular domestic conditions [3] and, recently, companies have introduced digital living assistants that have access to an individuals smartphone and other interconnected devices to assist the user. Although these devices contribute to the ease of daily activities, they pose a potential security threat.

The vulnerabilities of such IoT devices have been exploited. An example is the 2016 incident where IoT units were used to carry out a Distributed Denial of Service (DDoS) attacks on DynDNS which resulted in major sites and services being inaccessible to billions of people worldwide [4]. These IoT devices are however, not the only devices at risks. Industrial Programmable Logic Controllers (PLCs) are among other connected devices at risk [5, 6]. Interconnected PLCs, which allow the automation of electromechanical processes, such as those used to control machinery on factory assembly lines, amusement rides, or centrifuges for separating nuclear material (such as in the case of Stuxnet [7]), becomes a target and are potentially vulnerable if not correctly secured.

To ensure information is secured, cryptographic algorithms such as the Advance Encryption Standard (AES) have been adopted and are widely used [8]. These algorithms conceal vital information from potential eavesdroppers and are in theory mathematically secured. However, it has been demonstrated extensively that the implementation of AES is vulnerable to Side Channel Analysis (SCA) attacks [9, 10]. SCA exploits the power consumption or electromagnetic (EM) emissions to retrieve secret information by determining a correlation between the intermediate values and the power/electromagnetic consumption [11], which enables an adversary to recover encryption keys.

This research aims to investigate the susceptibility of the AES-128 cryptographic algorithm implementation on a Raspberry Pi, as EM radiation is leaked from the device during operation. The device under test is a Raspberry Pi 2 B+ model. This

Based on: "Investigating the Electromagnetic Leakage from a Raspberry Pi", by I. Frieslaar and B. Irwin which appeared in the Proceedings of Information Security South African (ISSA) 2017, Johannesburg, 16 & 17 August 2017. © 2017 IEEE

device was selected, as it these boards are readily available, widely used and on occasions used as PLCs [12]. This research implements and investigates the effect of multi-threads to purposely leak out obfuscated information at critical points alongside the AES-128 algorithm. Furthermore, there are still many unanswered questions as to the EM leakage from a Raspberry Pi and this study would contribute to the research field by answering the following questions:

1. How exposed is the libcrypto++ [13] implementation of AES on a Raspberry Pi to side channel attacks ?
2. Would various multi-thread libraries behave differently in terms of EM emanations ?
3. Can multi-threading be utilized as a software based countermeasure to SCA attacks ?
4. What effect would a multi-threaded software based countermeasure have on the EM emanations ?

In addition to the above questions, this research will investigate the implementation of an EM noise generator to obfuscate SCA analysis and mitigate the recovery of secret information from EM emissions.

The implementation of an EM noise generator is a software based countermeasure against Side Channel Analysis (SCA) that utilised multi-threading. A software solution was selected as it can be easily modified and rolled out to various devices, even with the possibility of remote updates. A hardware solution is tedious, as it requires users to return their devices or purchase a more expensive device, or on-site modifications. Hardware countermeasures can easily become obsolete whereas a software approach, can continuously be updated and easily maintained which can provide assurance to the end user that no additional cost would occur. This reduces the cost to manufactures as redesigning hardware is expensive.

The remainder of this paper is organized as follows: Section 2. discusses the impact and dangers of stolen encryption keys; Section 3. details the research carried out in the field of SCA attacks against high frequency devices; Sections 4. and 5. details potential techniques and approaches that are not commonly utilised as a software based countermeasure, but which can potentially serve as a countermeasure; Section 6. discusses how this research aims to capture, transform and utilise EM emissions from a Raspberry Pi to obtain useful information; Section 7. elaborates on the initial experiment and results with regard to the EM leakage from the various multi-threading libraries and a proposed software countermeasure; Section 8. demonstrates the recovery of the AES-128 secret key. This is followed by Section 9. that introduces a novel EM noise generator to prevent the recovery of the secret key. Attacks are performed against the novel EM noise generator in Section 10. Finally the paper is concluded with a discussion in Section 11.

2. IMPACT OF STOLEN ENCRYPTION KEYS

The impact of obtaining symmetric cryptographic or private keys could lead to a major catastrophe for industrial control hardware and federal systems. Once keys are obtained, decrypting information becomes a trivial task. Having open access to these systems would allow attackers to take control of a nuclear silo base or a strategic military asset capable of mass destruction. This was evident when the Iranian nuclear program had been sabotaged by stolen private keys in the Stuxnet attack [7]. The 2014 cyberattack on a German steel-mill [14] where critical infrastructure was destroyed is another example of a comprised system. Once the adversaries gained access to the system, they proceeded by taking control of the production management software of the steel mill and caused serious damage to the infrastructure.

The use of cyberweapons to gain an advantage on the battlefield or against foreign entities is nothing new [15, 16]. However, using these weapons on civilian targets such as power grids, traffic lights, hospitals and train stations is troubling and a compromise would lead to financial and socio-economic disaster. The possibilities of a compromised system are endless. A few scenarios would be that the attacker takes control over an insulin pump, where it is possible to distribute incorrect insulin doses, and produce a slow and painful death to an individual. Additionally, attackers have the ability to compromise and take control of a motor vehicle [17] by either disrupting or intercept electromagnetic signals to allow for malicious actions against the electronic control unit of the vehicle, which can lead to serious fatalities [18].

Devices such as satellite TV and other set-top decoders are vulnerable as the attacker would be able to intercept and decrypt the signal using the stolen encryption keys [19]. The decrypted signal can then be used and distributed to others without a subscription fee, thus resulting in industry losing revenue. Furthermore, cellphone calls could also be exposed and leaked to various third parties if the encryption keys were stolen.

In 2015 the hack into Gemalto [20], in which billions of mobile SIM cards cryptographic keys were stolen, demonstrated both the importance and impact of stolen keys. As a result of these stolen encryption keys, intelligence agencies and third parties are able to monitor mobile communications without applying for approval from telecommunication companies or foreign governments. Possessing these keys bypasses the need to obtain a warrant or a wiretap, while leaving zero evidence on the networks that communications were intercepted. Irrespective of the civil liberties, the main focus is that mobile communications have been compromised due to stolen keys.

Consumer entertainment services such as pay-per-view and video streaming would be susceptible as the attacker could pretend to be the streamer and make millions of dollars. Since the arrival of Digital Economy the use of IoT, smartphones and other devices has expanded specifically to include tasks such as banking, social networking, e-commerce and Bitcoin (and other cryptocurrency) transactions on a regular basis. Consequently it is therefore fundamentally important to ensure that all devices are protected.

The deployment and development of cyberweapons have increased drastically in the last decade. This has lead to cyber threats and attacks becoming more common, sophisticated and damaging. Yearly there are discussions on the security downsides and risks of an Internet–connected world [21, 22]. Recently Symantec [23] published a report detailing that the power grids around the world have been infiltrated by adversaries to firstly steal critical information such as technical diagrams, reports, passwords, cryptographic keys. and secondly causing mass destruction by destroying infrastructure via code. It is becoming extremely important and critical that all infrastructure be protected. Therefore, it is imperative to secure all avenues that could possibly lead to the recovery of cryptographic keys.

3. ELECTROMAGNETIC ATTACKS

Hitherto the research community has focused on SCA attacks based on smartcards, RFID tags, FPGAs and microcontrollers, particularly by analysing electromagnetic (EM) emissions [11, 24, 25]. In more recent years the research community began to target the vulnerabilities of high powered devices such as laptops and smartphones to SCA attacks [26–28]. The focus of this research is on the Raspberry Pi 2B+, which is a higher frequency device as mentioned in Section 1., that has been selected as it is readily available, widely used and on occasions used in the role of a PLCs [12].

EM emanations are captured from devices and subsequently used to retrieve secret information. EM measurements do not require the attacker to have direct contact with the device and are able to extract information without the user having knowledge of the attack. EM Attacks are therefore less intrusive than the conventional attacks via power analysis [11]. The remainder of this section will focus only on the attacks against high powered devices, such as smartphones and systems typically utilizing ARM family processors.

Aboulkassimi et al. [29] performed an EM attack on symmetric ciphers by capturing signals at the device clock rate on a Java based cellphone. However, they placed a MicroSD extension cable to the MicroSD card to extract EM information. Furthermore, Goller and Sigl [30] implemented attacks on a public key cryptography algorithm against an Android smartphone executing RSA. Additionally, the smartphone's shielding plate was removed.

Nakano et al. [31] attacked an Android smartphone using low frequency attacks. The smartphone ran at 832 MHz. Their attacks focused on the RSA and Elliptic curve cryptography (ECC) encryption implementations contained in the Java Cryptography Extension (JCE). Additionally, they removed the battery and metal covers in order to recover EM data from the smartphone. However, this was not mentioned in their work. Upon further analysis, the graphic provided in their paper, demonstrates visibly that modification had been made to the rear of the smartphone. Furthermore, no mention is made of the device specifications.

Belgarric et al. [27] and Genkin et al. [28] concurrently demonstrated successful attacks on the Elliptic Curve Digital Signature Algorithm (ECDSA) implementation of Android's BouncyCastle library. A minor difference found in the two studies was that the work of Belgarric et al. was intrusive as they placed the magnetic probe inside the smartphone where as Genkin et al. was non-intrusive and placed the magnetic probe in close proximity of the device. Genkin et al. demonstrated a successful secret key recovery of the ECDSA signing keys from OpenSSL executing on an iOS devices and partial keys from an Android device. Furthermore, they were able to recover secret keys from Corebitcoin off an iOS device.

Balasch et al. [32] performed a successful Differential Power Analysis (DPA) attack against the bitsliced AES encryption algorithm executing on a BeagleBone Black ARM development board. An ARM Cortex-A8 processor running at 1 GHz was used. A similar attack against the device was performed by Galea et al. [33]. The results obtained demonstrate the vulnerabilities of symmetric key encryption executing on high powered devices to SCA attacks. It is noted that in both studies the EM probe was physically glued onto the area of leakage and focused on specialised hardware with hardware countermeasures proposed.

As discussed in this section, high powered devices are vulnerable to SCA attacks. This is a fairly new research avenue which is evident by the minimal related work in the field. In addition, the research in [34] recently demonstrated the ability to recover partial (12 out of the 16 subkeys) AES-128 cryptographic keys from a Raspberry Pi. As it is a new field the research [27–34] has followed attacks introduced by targeting smartcards, RFID tags, FPGAs and microcontrollers. However, each attack against these higher frequency devices had different approaches with regards to recovery EM information and applying digital filtering techniques to recover sensitive information.

Many cryptographic algorithms and high frequency devices have not been put under scrutiny against SCA attacks. This distinctly ties into the questions posed in Section 1. This novel research will therefore be the first of its kind to investigate the EM effects of a symmetric encryption algorithm as it is executed on a Raspberry Pi in a multi-threaded environment and utilising multi-threads as a countermeasure on high frequency devices.

4. MULTI–THREADING AS A SOFTWARE COUNTERMEASURE

This section will discuss multi-threading techniques and approaches that this research has identified, but which are commonly utilised to increase the run-time execution of a program as a potential solution for a software based countermeasures to mitigate SCA attacks. This research has selected a multi-threading approach as it has been demonstrated in previous work that a multi-threading solution outperforms known software based countermeasures [25, 35, 36].

In terms of EM emissions and data leakage, it is unknown what effects each multi-threading approach could potentially have. Therefore, this research considers it a requirement to investigate the effects that each multi-threading approach would potentially have on the EM field. The experiments and results can be found in Section 7.

Multi-threading has been widely utilised to speed up run time operations of an executable. However, this research explores the use of multi-threading as a possible software based countermeasure. Four multi-threading Application Program Interface (API)s will be used on the Raspberry Pi. The four implementations are Portable Operating System Interface (POSIX) Threads or more formally referred to as pThreads,[*] C++11 multi-threads,[**] Thread Building Blocks (TBB),[***] and Open Multi-Processing (OpenMP).[†]

Multi-threading is the process of executing multiple threads or instructions concurrently. These threads are managed independently by a task scheduler [37]. Normally, one process can consists of multiple threads, as a thread could be a component of a process. The multiple threads execute their instructions in sequential order and share hardware resources such as memory, caches and registers [38]. Although the threads share resources, they are able to execute independently. Therefore, the threaded approach provides many developers with a good platform to create concurrent execution.

Two procedures are generally used in multi-threading to increase the performance of the system via improved utilisation of CPU cores, decreasing the runtime execution of a program [39]. These procedures are known as thread–level and instruction–level parallelism. They are normally used in combination with each other in hardware architecture that consists of multiple CPUs.

The main concern with implementing multi-threading on a single core processor is that the instructions are executed concurrently instead of simultaneously. However, it is still possible to achieve a performance gain using the multi-thread approach on a single core.

Developers utilise various libraries to develop their code in a multi-threaded framework. The most common framework in the UNIX environment is the pThreads [40]. Multi threading libraries provide developers access to creating applications with multi-threaded support. These libraries aslo provide developers with the option to create a synchronized environment between the threads by using mutexes, condition variables, semaphores, monitors and other synchronization primitives [41].

4.1 POSIX

pThreads, is a C API multi-thread library that has standardized functions above the Operating System (OS) infrastructure. It allows the user to spawn a new concurrent process flow and it is extremely effective on multi-processor or multi-core systems, where the process flow can be scheduled to run on another processor. This interface has been specified by the IEEE POSIX 1003.1c standard [42].

4.2 C++ Threads

The C++11 thread library is more of a memory model approach that supports multi-threading [43]. C++11 threads are fully incorporated into C++ as a language, so that there is no need to allocate arguments in a form of a struct. In addition, C++11 multi-threaded library allows for numerous arguments to be passed to a function that has a thread.

4.3 Threading Building Blocks

The Intel TBB [44] is a portable and open-source C++ template library for parallel data processing on shared memory architectures. It implements task flows in conjunction with the new C++ lambdas. The TBB is a high level implementation, thus it is implemented as a library and not a language extension. TBB allows

[*] pThreads – https://goo.gl/EmNDc6

[**] C++11 – https://goo.gl/wdev41

[***] TBB – https://goo.gl/xpNLFE

[†] OpenMP – www.openmp.org

the developer to create portable code that can execute on different OS architectures. Furthermore, TBB has been built upon pThreads as the underlying threading API.

4.4 OpenMp

OpenMP is a directive based extension to C/C++ and Fortran as it supports data and task parallelism on shared memory architectures [45]. As opposed to TBB, OpenMP is a language extension that requires an OpenMP-enabled compiler. OpenMP has support from various compilers except Clang. In addition, OpenMP is build on pThreads.

5. HASH FUNCTIONS

This research has selected cryptographic hash functions as a candidate for the software based countermeasure to mitigate SCA. It will be empirically demonstrated in Sections 7. – 10. that the hash function outperforms the other candidates thus, this section will discusses the basic concepts of a hash functions and the intended hash algorithms to be utilized within this research as a software based countermeasure. This section will discusses the basic concepts of a hash functions and the intended hash algorithms to be evaluated within this research as a software based countermeasure.

A cryptographic hash function takes a string of various length as input and converts it to a fixed output of bytes. The hash function is regarded to be extremely flexible as it can be utilised in many schemes such as encryption, authentication and even as a digital signature [46]. Hash function are generally utilised in cohesion with encryption algorithms to increase the security of a system.

A cryptographically secure hash function is regarded as a one-way function as the output of the hash cannot be reversed to determine the original input message [47]. In addition, hash functions should be resilient to collision attacks, where two different messages cannot have the same hash output.

This research focusess on the Secure Hash Algorithm (SHA) family, more specifically the SHA–1 and SHA–2 algorithms. The SHA family of cryptographic functions are published by the National Institute of Standards and Technology (NIST) [48]. SHA–3 also forms part of the published family. The specific algorithm used in this is known as Keccak.[††] However, this research will not consider SHA–3 as it differs substantially from SHA–1 and SHA–2 in its internal operation.

SHA converts plain text into secure information by utilising a cryptographic hash function. A cryptographic hash function is a mathematical algorithm consisting of bitwise operations, modular additions, and compression functions which transforms the input message to a bit string of a fixed size [49].

Each iteration of the SHA–1 and SHA–2 algorithms was designed to increase the security of the algorithm and prevent attacks that the previous version was vulnerable to. The core difference between these algorithms of interest are depicted in Table 1. The algorithms of interest in this research are the SHA–1, SHA–224, SHA–256, and the SHA–512 hash algorithms. The SHA algorithms ranging from SHA–224 to SHA–512 are part of the SHA–2 family.

Table 1: Basic information regarding the hash functions.

Algorithm	Output Size (bits)	Block Size	Max Input Size
SHA-1	160	512	$2^{64}-1$
SHA-224	224	512	$2^{64}-1$
SHA-256	356		
SHA-384	384	1024	$2^{128}-1$
SHA-512	512		

Until now, this research has been the first to design and develop a software based countermeasure to mitigate SCA by utilising the SHA family hashes to generate EM noise. Therefore, this research will be the first of its kind to embark on the process of determining if the SHA is a suitable candidate to serve as a noise generator. In addition, the SHA family can be implemented on various platforms and devices.

6. METHODOLOGY

This section elaborates on the equipment utilised in this research to capture the EM emanations and the process of collecting and analysing the EM data into meaningful information.

6.1 Equipment

This research utilised two Raspberry Pi's. The first device served as the victim, while the secondary device served as the adversary. The Lubuntu 14.04 [‡] Operating System, with the Linux 3.18.0-20-rpi2 kernel[‡‡] was utilised. It is to be noted that no services in the stock Operating System were disabled. In addition, to limit the CPU from using internal step-up controls to adjust power and CPU frequency, the victim's maximum CPU frequency was configured to 600 MHz. This research followed the approach by [32] and [28] to limit the CPU frequency. It is noted has built in time delays and interrupts as a countermeasure. Furthermore, no adjustments were made to the secondary device. The FUNcube dongle was inserted into a USB port and GNURadio was used to interface with the device. Figure 1 illustrates the experimental setup.

[††] Keccak – https://keccak.team/index.html

[‡] Lubuntu 14.04 – https://goo.gl/sUutNh

[‡‡] Linux 3.18.0-20-rpi2 kernel – https://goo.gl/5azpV5

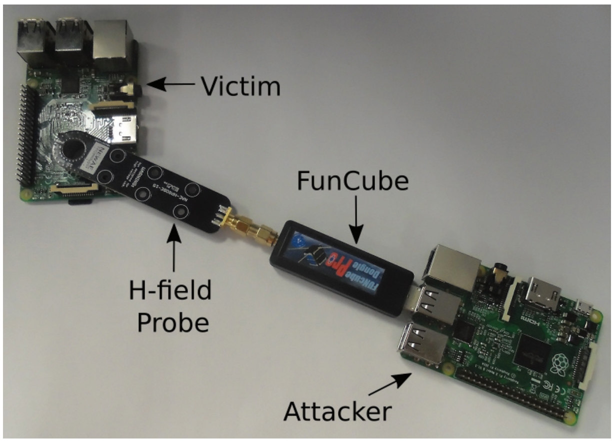


Figure 1: The experimental setup.

In order to interface with the SDR and perform digital signal processing, GNURadio 3.7.9[§]and Baudline 1.0.8[¶] were utilised.

GNU Radio is a free software development toolkit that provides signal processing blocks to implement software defined radios and signal processing systems. In addition, Baudline is a time-frequency browser designed for scientific visualization of the spectral domain. Baudline can be utilised to the extract signals at specific points in time.

6.2 Data Analysis

There are three stages at which the EM data can be analysed [31]. Stage one is to compute the Fast Fourier Transform (FFT) over the baseband waveform. This process assists in establishing a frequency signature for various operations, as different operations would produce a specific pattern. Figure 2 illustrates the raw signal after it had been processed through a FFT within GNURadio. The signal is obtained by monitoring the desired 600 MHz frequency in real time via the SDR.

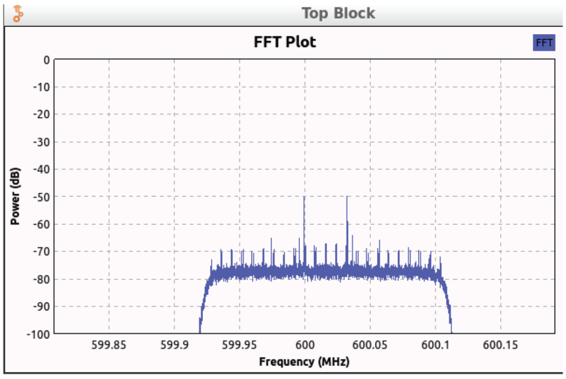


Figure 2: The signal as it is displayed by the FFT.

Stage two consists of selecting the region of interest at the point of EM leakage, i.e: when a specific operation executes and can be seen in the radio spectrum. This is performed by visual analysis and can be seen in the amplitude domain in Figure 3 by a rectangle encapsulating the peak signal. This informs the adversary of the location of execution in time and the type of signature that is produced.

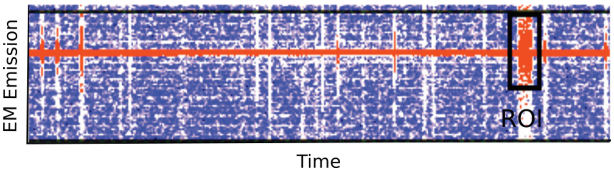
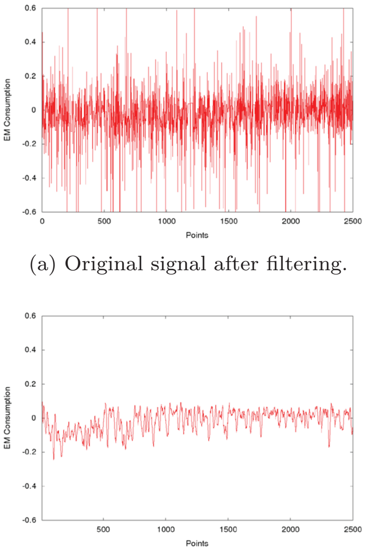


Figure 3: The region of interest.

Finally, the third stage is to apply digital filtering and noise removal techniques (de-noising) to filter unwanted information in the signal. Figure 4a illustrates the signal after digital filtering has been applied and Figure 4b depicts the signal after a de-noising method has been applied to the signal.



(a) Original signal after filtering.

(b) Signal after digital processing

Figure 4: The original (a) and (b) the signal after processing.

While the victim executes various test algorithms, the adversary uses GNURadio to capture the EM emissions from the device at 384 kHz. The FUNcube dongle has a range of features with regard to the input bandwidth, ranging from 44.1 kHz – 384 kHz. Sampling at a higher frequency allows for more data to be obtained. Therefore, each recording is captured at 384kHz. Digital filtering was applied to keep all signal information between 0 – 50 kHz, while the rest of the signal frequency was discarded. The resultant signal was sent to Baudline where the region of interest was extracted and sent back to GNUradio, where

[§] GNURadio – https://www.gnuradio.org/
[¶] Baudline – http://www.baudline.com/

Quadratic demodulating [50] was applied. This was followed by an additional low pass filter.

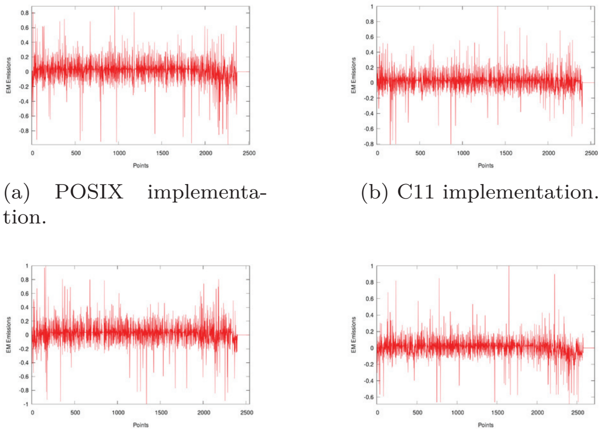
Since the research is dealing with digital signals and it is known that information is carried within the carrier signal [50], demodulation is applied to recover the information within the signal, specifically, quadratic demodulating.

7. EXPERIMENTS, RESULTS, AND ANALYSIS

This section discusses the software countermeasure investigated to mitigate the SCA attacks against the Raspberry Pi. The proposed approach uses multi-threads to purposely leak out obfuscated information while the AES-128 algorithm executes. These APIs are pThreads; C++11 multi-threads, henceforth referred to as C11 threads; TBB and OpenMP threads, as discussed in Section 4. In addition, Section 7.1 discusses the findings with regard to the EM leakage as the AES-128 encryption was executed, followed by Section 7.2 elaborating on the experiments as noise was introduced.

7.1 EM Leakage

The first set of experiments consisted of the AES-128 algorithm encapsulated into the four multi-threaded APIs. One thread executed the cryptographic algorithm on the four different platforms. No additional threads were utilised. While the programs were running, the EM emanations were captured alongside the time per execution.



(a) POSIX implementation.

(b) C11 implementation.

(c) OpenMP implementation.

(d) TBB implementation.

Figure 5: The EM leakage of the four cryptographic threaded implementations.

Figure 5 illustrates the EM leakage of the cryptographic threaded implementations as one thread was used for the various multi-threading APIs. It is observed that all four outputs have a similar EM signature. A better visual description can be seen in Figure 6 as the four implementations are overlaid onto one figure. The results are as expected, as only one thread was executing the cryptographic algorithm.

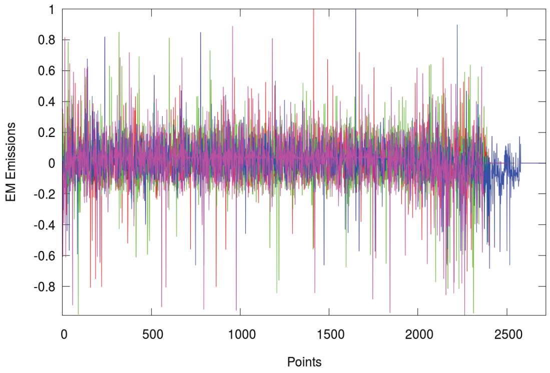


Figure 6: A combination of the EM leakage of the four cryptographic threaded implementations as one thread was used

It is seen in Figures 5d and 6 that the TBB implementation ends at a later stage. Even though the other EM signatures have are similar EM profile to that of the TBB implementation. The TBB has a larger output, i.e. TBB ends above 2500 data points where as the other implementations ends before 2500 data points. This is associated with the fact that TBB takes longer to execute the code. Additional data will be provided later on in this section to demonstrate the findings.

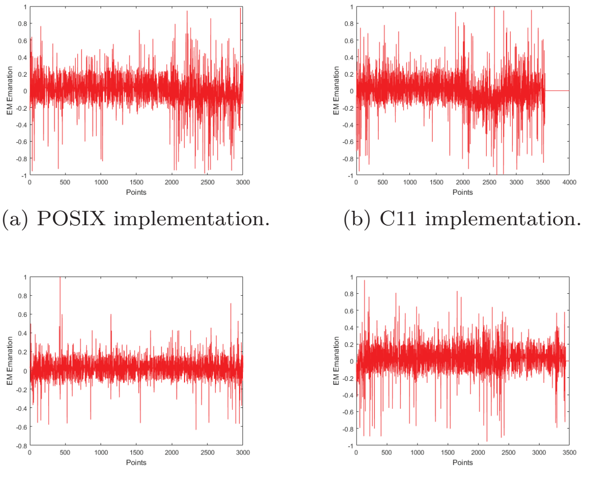
The second set of experiments consisted of the AES-128 algorithm encapsulated into the four threaded APIs with a secondary thread calculating the first 1000 Prime numbers and is depicted in Figure 7. The two threads executed in parallel with no synchronization and no thread had priority over the other. The EM emissions of the experiment is depicted in Figure 7.

The first notable finding is that the data points have increased which is due to the additional thread. In addition, the TBB emissions in Figure 7d have increased by 1000 data points. Each implementation has a noticeably different EM signature.

The third set of experiments consisted of the AES-128 algorithm encapsulated into the four threaded APIs with a secondary thread calculating the first 1000 Fibonacci numbers and is displayed in Figure 8.

Observing Figures 8a–8d, it is shown that each implementation of the four thread techniques have different power patterns. The pThread implantation in Figure 8a illustrates that it takes the fewest number of points as compared to the rest of Figures 8b – 8d.

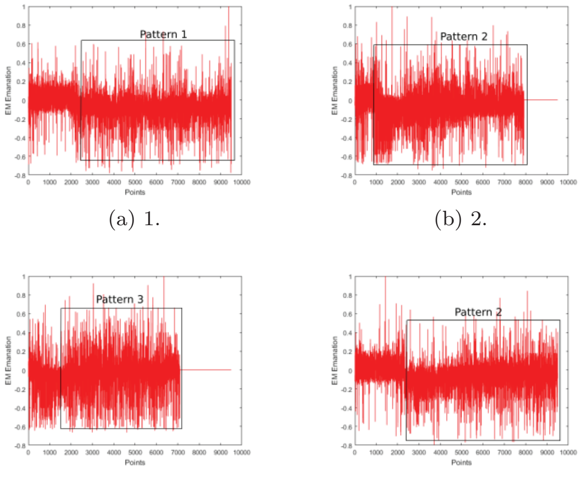
Additional analysis was performed. However, for this occasion only one multi-threaded API was



(a) POSIX implementation. (b) C11 implementation.

(c) OpenMP implementation. (d) TBB implementation.

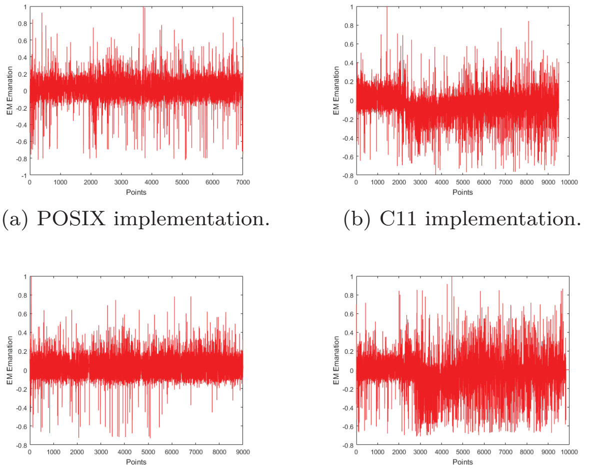
Figure 7: The EM leakage with an additional thread calculating prime numbers.



(a) 1. (b) 2.

(c) 3. (d) 4.

Figure 9: Various EM patterns.



(a) POSIX implementation. (b) C11 implementation.

(c) OpenMP implementation. (d) TBB implementation.

Figure 8: The EM leakage with an additional Fibonacci thread.

selected. Multiple instances of calculating the Fibonacci sequence analogised the AES-128 execution was performed and the EM data was captured. Figures 9a – 9d depict that there are three distinct patterns and only Figure 9b and 9d have similar patterns. Figures 9a and 9c have a different pattern and end at a different location in time.

Although, the same sequence of operations was performed, different EM patterns were being recorded. The research noted that this was due to the Raspberry Pi's CPU and power management setup [51]. More information relating to this will be discussed later in this section.

The data presented in the results continue by illustrating the average recorded execution times for the four thread techniques using the various programs in Table 2.

Table 2: The average run time using the four different multi-thread techniques.

	Threaded-Crypto	Crypto+ Prime Threaded	Crypto+ Fibonacci Threaded
C11	2,779	16,075	152,025
OpenMP	2,449	15,586	129,32
pThreads	0,593	13,826	117,252
TBB	6,377	31,602	151,316

Time in milliseconds.

The first column represents the four multi-thread implementations, followed by the execution time of cryptographic algorithm using one thread; the cryptographic algorithm with prime numbers executing on an additional thread; and finally, the cryptographic algorithm alongside the Fibonacci threaded implementation. The time is displayed in milliseconds.

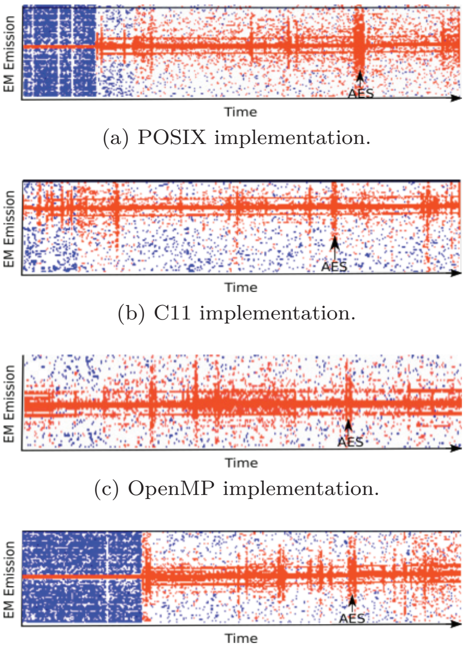
The data in Table 2 displays that the pThreads implementation was the fastest in all scenarios. This is further evident when comparing the EM emissions of the Fibonacci implementations in Figure 8 as it can be seen that the pThreads implementation has fewer data points over time. The TBB implementation is by far the slowest. Additionally, when computing the Fibonacci sequence the C11 is slower than the TBB.

Based on the data over all the graphs, in certain scenarios there are specific patterns. The CPU cores were monitored and the analysis indicated that the built in power management uses different cores on each execution of a program. The multi-core processors distribute resources to other existing cores in order to complete the task effectively. During the cryptographic operation, the process was moved from one core to another due to built in system interruptions and power management. It can also be executed in parallel by several cores for performance.

It is noted that all programs were compiled with the GCC compiler and the -O0 command. With -O, the compiler attempts to reduce code size and execution time, without performing any optimizations that reduce compilation time. The research in [52] details work done regarding to EM leakage as different compiler optimizations are utilised.

7.2 Generating Noise

The upcoming experiments will focus on the implementation of a noise generator program to execute in the background on a different thread as a daemon while the cryptographic algorithm executes. This daemon generated noise will be referred to as FRIES noise. The FRIES noise consisted of executing various arithmetic instructions in an infinite loop. These operations consisted of calculating the Fibonacci sequence, prime numbers, and other arithmetic instructions [36]. The Daemon approach has been selected as it is this research desire to develop a countermeasure that does not modify the existing cryptographic algorithm, thus it can be implemented with other application such as banking and various other financial applications.



(a) POSIX implementation.

(b) C11 implementation.

(c) OpenMP implementation.

(d) TBB implementation.

Figure 10: The EM leakage with a noise generator.

Figure 10 illustrates the time domain as the four multi-threaded implementations of the cryptographic algorithm are executed while the noise program is running in the background.

The annotated arrow in the Figures 10a–10d illustrates at which point the cryptographic algorithm took place. This indicates it is still possible to detect the location of execution of the cryptographic algorithm in the second stage as discussed in Section 6. although, on 32 occasions out of 50 runs, the cryptographic execution was not detected.

Further analysis was carried out and it was determined as the prime numbers was executing at the same time of the cryptographic thread, the cryptographic execution was hidden. A secondary set of experiments was carried out. This entailed calculating N number of prime numbers randomly in an infinite loop while the cryptographic code executed. Figure 11 indicates the sequence as the prime numbers were randomly calculated alongside the cryptographic algorithm.

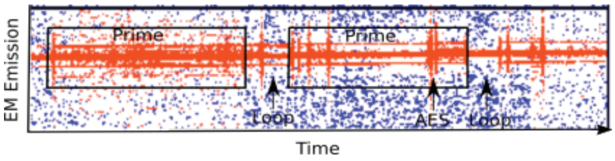


Figure 11: EM leakage as FRIES was executing.

The first rectangle represents the EM emanation as the prime numbers were calculated. In addition, the AES algorithm was executed within that time frame and it cannot be explicitly observed in the plot. However, in the second rectangle the AES execution becomes visible during prime calculation. The data and code had been analysed in order to determine why in certain cases the cryptographic algorithm is visible. The analysis indicated that when a low number of primes is calculated there is a window in the noise, which results in the where the cryptographic algorithm and for-loops can be seen visibly as depicted by the annotations in Figure 11.

Having established a range to execute the prime numbers while the cryptographic algorithm is executing, the third experiment involved calculating prime numbers between 100 and 1000 indefinitely. Figure 12 displays the EM emanations after demodulation has been applied. The waveform in the figure is repeated over time and it becomes extremely difficult to pin-point the exact location of the execution of the AES-128 algorithm.

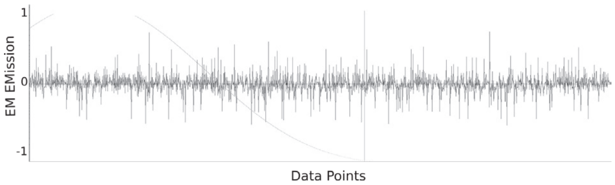
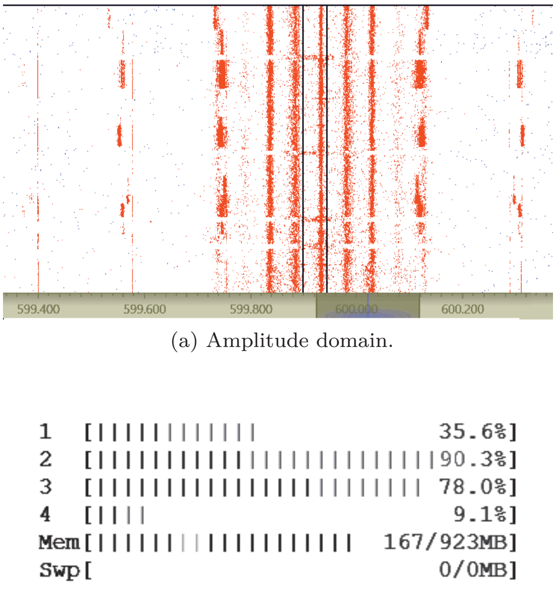


Figure 12: The EM leakage depicting a repeatable waveform.

Additional data has been identified by monitoring the EM emanations. As more power is used by the Raspberry Pi, more leakages appear across the spectrum. This is due to voltage harmonics. It is further evident in Figure 13a that as more power and/or cores are used by the CPU more data is available across the spectrum as illustrated in Figure 13b.



(a) Amplitude domain.

```
1  [|||||||||||||                     35.6%]
2  [||||||||||||||||||||||||||||||||||90.3%]
3  [||||||||||||||||||||||||||||||||| 78.0%]
4  [|||||                              9.1%]
Mem[||||||||||||||||||||||||||   167/923MB]
Swp[                                 0/0MB]
```

(b) Htop displaying core usage.

Figure 13: The leakage (a) across the spectrum and (b) the core usage from the device.

8. RECOVERING SECRET INFORMATION

Research from Frieslaar and Irwin [34] has recently demonstrated the capabilities of recovering partial AES-128 cryptographic subkeys from a Raspberry Pi. The research was able to retrieve 12 of the 16 subkeys. However, this research aims to build on the previous research by recovering the full encryption key, i.e: all 16 subkeys.

The same capturing procedure will be utilised as mentioned in [34]. Therefore, in order to acquire the EM data from the Raspberry Pi, the same software configuration was used and the procedure is as follows:

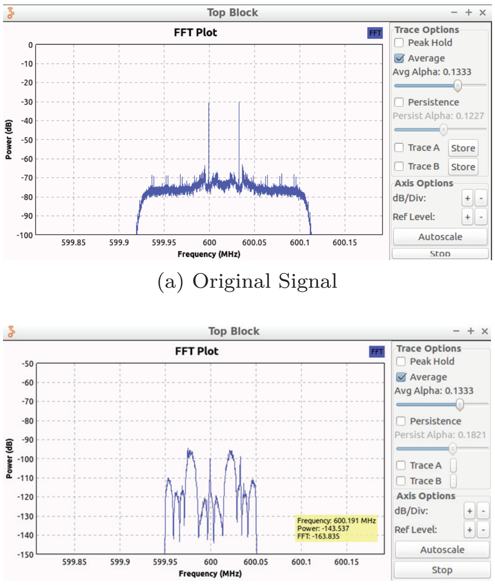
1. While the victim's device executes the cryptographic algorithm, GNURadio is utilised to capture raw signals.
2. The raw signal is sent to a Fast Fourier Transform (FFT) process.
3. Digital filtering techniques such as low and high pass filters are applied to the signal.
4. The region of interest is extracted from the filtered signal with Baudline.
5. Quadratic demodulation is applied to the region of interest.

The resultant signal will be passed to a prepossessing procedure where various alignment techniques will be applied to the signal. This research follows the process mentioned by [34] to align the data. The process is as follows:

1. The signal is segmented into three partitions.
2. Elastic alignment [53] is applied to each partition.
3. The aligned segmented partitions are rejoined.
4. Resyncing techniques, such as peak detection and sum of absolute difference are applied.
5. The Savitzky-Golay [54] filter is used to increase the signal to noise ratio.

Once the data has been aligned, the resultant data will be sent to the Correlation Power Analysis (CPA) [55] attack. The CPA attack is utilised to recover the secret key used by the AES-128 algorithm with the EM emissions as input data.

The original research had kept all signals ranging from 0–50 kHz within the captured signal. This research proposes the removal of certain frequencies. The frequencies between 0–15 kHz will be removed and the resultant signal will be sent to the attack procedure as input for the CPA attack. Figure 14 illustrates a comparison between the original signal and the signal with the 0 – 15 kHz frequency removed.



☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1227
☐ Trace A Store
☐ Trace B Store

(a) Original Signal

☐ Peak Hold
☑ Average
Avg Alpha: 0.1333
☐ Persistence
Persist Alpha: 0.1821
☐ Trace A
☐ Trace B

Frequency: 600.191 MHz
Power: -143.537
FFT: -163.835

(b) Omitted 0–15 kHz.

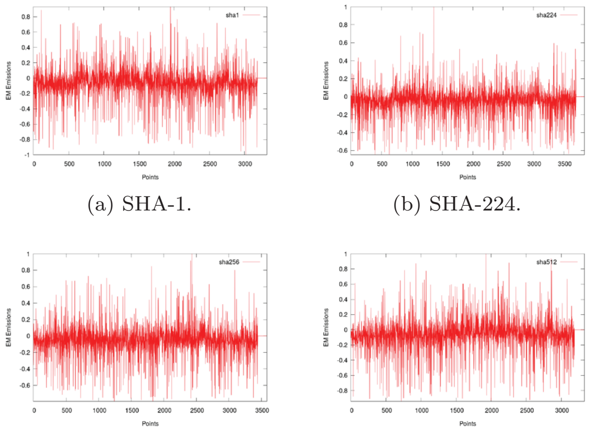
Figure 14: The FFT with the (a) original and (b) modified signal.

Following the removal of the frequencies between 0 –15 kHz, this research was successfully able to recover the entire secret key used for the AES-128 cryptographic algorithm. This is a significant improvement over the previous work where only 12 subkeys were recovered. In addition, only 50 traces were required as input data. The results of the CPA attack is depicted in Table 3.

The results indicate that utilising 10 traces, five subkeys were recovered, eventually as 50 traces were utilised the entire subkey was recovered. It is further observed that subkeys 2, 4, 6, 8, and 12 were recovered in all the subsets and only subkeys 4, 10, 15 and 16 were recovered when 50 traces were used.

9. IMPROVING THE COUNTERMEASURE

The prime number sequence within the FRIES noise generator was replaced with the SHA from the libcrypto++ library. Firstly, an EM signature profile was created by capturing the various implementations off the SHA as they executed on a Raspberry Pi. These implementations are SHA-1, SHA-224, SHA-256, and SHA512 functions as discussed in Section 5. The capturing of the EM emissions remained the same as discussed in Section 6.



(a) SHA-1.

(b) SHA-224.

(c) SHA-256.

(d) SHA-512.

Figure 15: The EM leakage for the various SHA functions.

Figure 15 depicts the EM leakage from the Raspberry Pi as the various SHA functions were executed within libcrypto++. The EM signature patterns changed as different hash function were utilised, more specifically, that of the SHA-512 function as seen in Figure 15d.

The SHA functions are based on strict cryptographic principles as opposed to the calculation of prime numbers as discussed in Section 5. Therefore, the FRIES noise generator was modified to execute the SHA-512 hash function in an infinite loop and run in the background as a Daemon. The new process of the FRIES noise generator is depicted in Figure 16.

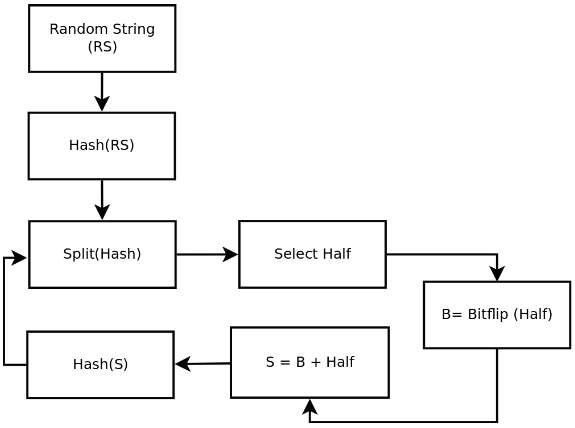


Figure 16: Genearting random hashes for the FRIES noise generator.

The process commences by generating a random alphanumeric string as input for the SHA-512 hash function. The resultant hash value was segmented into two and within each loop a decision was determined randomly whether to take either the upper or lower half of the segmented hash value. A bit-flip was applied to the segmented hash value and appended to the previous half that was not selected in the previous step. The resultant string was utilised as input for the SHA-512 hash function. This process was repeated indefinitely, thus increasing the entropy and generating random secure hashes.

While the FRIES noise generator was running in the background, the AES-128 algorithm was executed over time. After the noise generator had calculated 40 hashes, the CPU of the Raspberry Pi had reached 100% usage on all four cores. This resulted in the device executing the FRIES noise generator in a slow down state. While the device was in this state the AES-128 algorithm could be clearly observed as indicated in Figure 17 by the annotations.

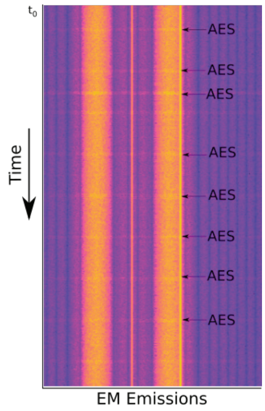


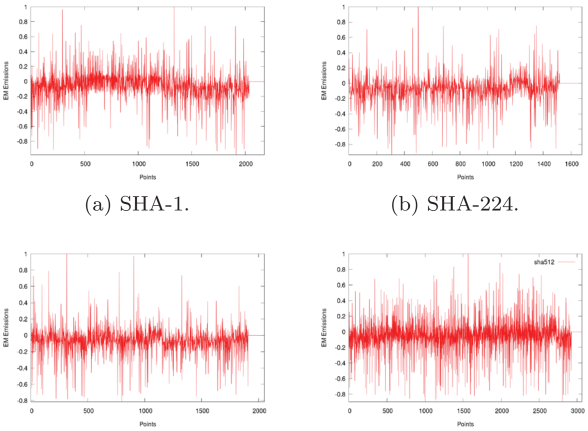
Figure 17: The runtime of the AES-126 algorithm while the FRIES noise was executing.

Instead of utilising the libcrypto++ library, the OpenSSL library that includes the SHA were

Table 3: The subkeys recovered as various traces were utilised.

	Subkey																
Traces	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Total
10	-	-	Y	-	Y	-	Y	-	Y	-	-	-	Y	-	-	-	5
20	Y	Y	Y	-	Y	-	Y	-	Y	-	-	-	Y	-	-	-	7
30	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	-	-	-	10
40	Y	Y	Y	-	Y	Y	Y	Y	Y	-	Y	Y	Y	Y	-	-	12
50	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	16

investigated with regards to the EM leakage and the potential candidate as a countermeasure. As the various hash functions of the OpenSSL library executed the EM leakage was captured, with the EM signatures depicted in Figure 18.



(a) SHA-1.

(b) SHA-224.

(c) SHA-256.

(d) SHA-512.

Figure 18: The EM leakage for the various SHA functions from the OpenSSL library.

Based on Figure 18, the data points i.e: length of the EM signature increased in length, as seen in Figure 18a (1500 points) to Figure 18d (2600 points). This can be related to the output size of each hash function as seen in Table 1. Furthermore, the block size of the SHA-512 function is 1024 bits where as the rest of the hash function has a block size of 512.

The SHA implementations within the OpenSSL library were incorporated into the FRIES noise generator. A stress test was firstly performed against the Raspberry Pi as the FRIES noise generator was executed to run in the background indefinitely. The results revealed that the CPU usage was 100% on only two cores as opposed to all four cores when the libcrypto++ implementation was in place. In addition, there was no lag from the device. This process demonstrated the SHA functions within the OpenSSL library was well-suited to be implemented as a noise generator as the device had not suffered from the additional overhead and the AES-128 cryptographic algorithm remained hidden.

As the FRIES noise generator was executing in the background the AES-128 program was executed with random intervals on multiple occasions over the monitored time period. The EM signature of this experiment is depicted in Figure 19.

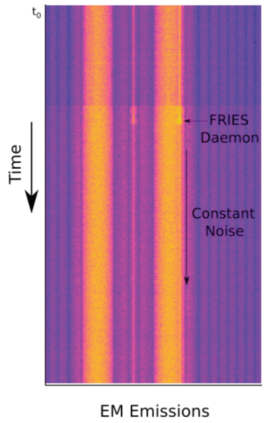


Figure 19: EM emissions depicting the startup point for the FRIES noise generator.

The annotated Figure 19 depicts an initial EM spike which is the start-up process of the FRIES noise generator. After the initialisation process has been completed, the FRIES noise generator produced a constant EM leakage footprint. Within this constant leakage, the execution of the AES-128 algorithm could not be identified visually.

A secondary experiment was conducted by alternating between the FRIES noise generator being on and off. While the FRIES noise generator was being toggled between on and off states, the AES-128 cryptographic program was executed. Within this period the EM emissions were captured and are depicted in Figure 20.

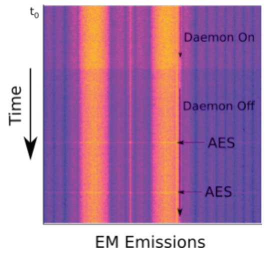


Figure 20: The EM frequency has the FRIES noise is switched on and off.

While the FRIES noise generator was executing in the background, there were no additional scan lines as seen in Figure 20. In addition, it is demonstrated that while the Daemon was in an off state the AES-128 cryptographic execution could be seen visually as annotated by the "AES" in Figure 20 and the arrow pointing to the spike in the EM spectrum.

10. ATTACKING THE IMPROVED COUNTERMEASURE

Although, the results from the previous section demonstrated that the encryption process could not be identified visually, the data was further analysed. The data demonstrated that out of 80 samples, there were ten traces that gave off slight harmonic noise. These traces were extracted and analysed as depicted in Figure 21.

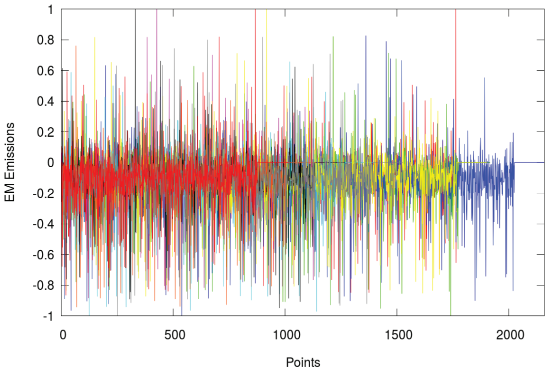


Figure 21: The EM emissions acquired from harmonic leakage.

It can be seen that each trace has a different EM signature. The data was sent to the alignment process as discussed in Section 8. The resultant aligned data was utilised as input for the CPA attack and the results indicated no secret information was revealed.

Statistical analysis was performed on the captured EM data. The baseline AES-128 cryptographic signal was used as a template for a cross correlation test to determine whether the AES-128 EM signature was within the signal while the noise generator was enabled.

The Cross-correlation between AES-128 cryptographic template and the signal produced by the noise generator is illustrated in Figure 22. The x-axis is a representation of time at the point of AES-128 cryptographic signature being similar to that of the EM signature from the noise generator. The figure reveals that cross-correlation test could not determine a correlation between the AES-128 EM signature and the EM signature produced by the generator as there are no profound spikes in the figure.

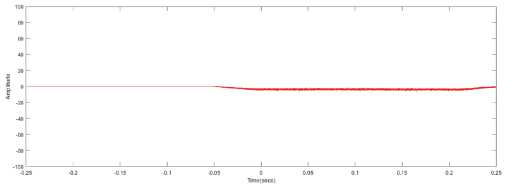


Figure 22: Cross-correlation results.

It is noted that there is no one evaluation process with regards to evaluating a software countermeasure, the countermeasures are generally evaluated by the number of subkeys recovered. This research felt that further evaluation was required, thus the Chi-square test [56] was applied. This approach is utilised to evaluate hardware countermeasures, especially with regard to hardware noise generators [57]. The purpose of this test is to evaluate how likely the EM emissions generated as the AES-128 algorithm executed can be detected within the noise. If it can still be detected that means the AES-128 cryptographic keys could be recovered. The EM signature of the iterations of the FRIES noise generator was compared to the original EM signature of the AES-128 cryptographic within a multi-thread. The results are depicted in Table 4.

Firstly the EM signature was compared to itself and the resulted indicated that there is a 99% probability that the signal is within the same signal as depicted by AES Plain. The first iterations of the proposed countermeasure as only multi-threads were utilised depicted that there was a 99% probability that the cryptographic signature is within that signal. The prime number generator, when the signal was visible, depicts the same results as that of the multi-threading.

Interestingly, as the AES-128 cryptographic thread was not visible, the statistical test indicated that there was a 50% probability that the AES-128 cryptographic signal was within the noise of the prime numbers. Furthermore, the final iteration of the FRIES noise generator with the SHA functions from the OpenSSL library revealed that there was only a 1% probability that the AES-128 cryptographic EM signature was located within the noise generated.

11. CONCLUSION

The EM leakage of the Raspberry Pi as the device executed the AES-128 algorithm of the Crypto++ library in a threaded environment was investigated. The multi-threaded libraries used were the pThreads; C11 threads; TBB; and OpenMP. The research illustrates that different patterns in the EM emanation occurs. This was determined by monitoring the CPU usage and core intensity. The Raspberry Pi's power management system used different cores on each execution of the code, for example: only three cores were used in the first execution; all the threads would be used in the second execution. The four different thread techniques were demonstrated to have variations in frequency and shape of EM emanations leaked. The pThread thread implementation was found to exhibit the fastest runtime execution.

Table 4: Results of the chi-square test.

	Probability								
Countermeasure	0.99	0.95	0.9	0.75	0.5	0.25	0.1	0.05	0.01
FRIES	–	–	–	–	–	–	–	–	X
Prime Numbers	–	–	–	–	X	–	–	–	–
Prime Numbers Visable	–	X	–	–	–	–	–	–	–
Multi-Threading	–	X	–	–	–	–	–	–	–
AES Plain	X	–	–	–	–	–	–	–	–

A Daemon noise generator known as the FRIES noise generator was introduced in Section 7. The results demonstrated that the FRIES noise had different effects on the radio spectrum. It was still possible to visually detect the location of the execution of the cryptographic algorithm. However, the visual detection could not been seen on some occasions. It was identified that the calculation of prime numbers would hide the cryptographic algorithm. The analysis revealed that as high prime numbers were calculated there was a window where the cryptographic algorithm could not be seen visibly.

This research improved on previous work [34] by removing the low frequencies from 0–15 kHz within the captured signal i.e: only frequencies between 15 – 50 KHz were kept. This data was utilised as input data for the CPA attack. Following this approach, this research was successfully able to recover the entire secret key used for the AES-128 cryptographic algorithm.

The prime number sequence within the FRIES noise generator was replaced by the libcrytpo++ implementation of the Secure Hash Algorithm (SHA) in Section 9. The results indicated that the CPU was utilising a 100% of resources and the device started to lag and slow down. In this slow down state the AES-128 encryption program could be visibly seen and extracted.

The FRIES noise generator was re-coded to utilise the OpenSSL libraries. The stress test demonstrated that the CPU usage was 100% and all cores were being utilised. However, there was no lag from the device. Furthermore, more hashes were being calculated per second as apposed to libcrypto++ implementation. While the hash function was introducing EM noise, the cryptographic implementation of AES-128 algorithm could not be visibly seen.

Statistical analysis was performed on the FRIES noise generator in Section 10., more importantly the cross-correlation between the FRIES noise and a non-protected AES-128 cryptographic implementation and the Chi-square test between the FRIES noise and a non protected AES-128 cryptographic implementation. The results revealed that there was no correlation between the two sets of EM signatures. The OpenSSL library is therefore deemed to be well suited as a base in order to obfuscate EM SCA attacks.

This research has successfully answered the questions posed in Section 1. The AES-128 cryptographic implementation within the libcrypto++ library on a Raspberry Pi is vulnerable to SCA attacks. The cryptographic process was seen visibly within the EM spectrum. The EM data for this process were extracted and additional digital filtering techniques was applied to the signal, more specifically the removal of the 0-15 kHz frequency. The resultant data was utilised in the CPA attack and the results revealed that the entire AES-128 key was recovered. This research introduced a multi-threaded approach with the utilisation of SHA to serve as a software countermeasure to mitigate SCA attacks. The proposed countermeasure executed as a Daemon and generated EM noise that was able to hide the cryptographic implementations and prevent a CPA attack and other statistical attacks.

12. FUTURE WORK

This research lays the foundation for future research towards building software based Electromagnetic noise generators not only for cryptographic processes, but for other programs and applications with sensitive information. Applying cryptography on dummy executions. The datasets utilised have been made available [58].

ACKNOWLEDGEMENT

This work was undertaken as part of the Distributed Multimedia CoE at Rhodes University, with financial support from the Information Security Competency Area within Modelling and Digital Science at the CSIR, Telkom SA, Tellabs/ CORIANT, Easttel, Bright Ideas 39, THRIP and NRF SA (UID 90243). The authors acknowledge that opinions, findings and conclusions or recommendations expressed here are those of the author(s) and that none of the above mentioned sponsors accept liability whatsoever in this regard.

REFERENCES

[1] M. Yun and B. Yuxin, “Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid,” in 2010

International Conference on Advances in Energy Engineering, June 2010, pp. 69–72.

[2] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, "Fog computing and its role in the internet of things," in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ser. MCC '12. New York, NY, USA: ACM, 2012, pp. 13–16.

[3] S. D. T. Kelly, N. K. Suryadevara, and S. C. Mukhopadhyay, "Towards the implementation of IoT for environmental condition monitoring in homes," IEEE Sensors Journal, vol. 13, no. 10, pp. 3846–3853, Oct 2013.

[4] R. Button, "Dyn DynDNS DDoS attack," 2016, accessed: 2017-05-01. [Online]. Available: http://www.red-button.net/blog/dyn-dyndns-ddos-attack/

[5] D. Wei, Y. Lu, M. Jafari, P. Skare, and K. Rohde, "An integrated security system of protecting smart grid against cyber attacks," in 2010 Innovative Smart Grid Technologies (ISGT), Jan 2010, pp. 1–7.

[6] S. Amin, X. Litrico, S. Sastry, and A. M. Bayen, "Cyber security of water SCADA systems; Part I: Analysis and experimentation of stealthy deception attacks," IEEE Transactions on Control Systems Technology, vol. 21, no. 5, pp. 1963–1970, Sept 2013.

[7] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," IEEE Security and Privacy, vol. 9, no. 3, pp. 49–51, May 2011.

[8] U.S. Depatment of Commerce, "Advanced encryption standard (AES)," National Institute of Standards and Technology (NIST), Tech. Rep., 2001.

[9] P. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in Advances in Cryptology — CRYPTO' 99: 19th Annual International Cryptology Conference Santa Barbara, California, USA, August 15–19, 1999 Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, pp. 388–397.

[10] I. Frieslaar and B. Irwin, "Towards a software approach to mitigate correlation power analysis," in Proceedings of the 13th International Joint Conference on e-Business and Telecommunications - Volume 4: SECRYPT, (ICETE 2016), INSTICC. SciTePress, 2016, pp. 403–410.

[11] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in Proceedings of the Third International Workshop on Cryptographic Hardware and Embedded Systems, ser. CHES '01. London, UK, UK: Springer-Verlag, 2001, pp. 251–261.

[12] P. B. Rao and S. Uma, "Raspberry Pi home automation with wireless sensors using smart phone," International Journal of Computer Science and Mobile Computing, vol. 4, no. 5, pp. 797–803, 2015.

[13] Crypto++, "Crypto++ library." [Online]. Available: https://www.cryptopp.com/

[14] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," Industrial Control Systems, 2014. [Online]. Available: https://goo.gl/M7qkAN

[15] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," Survival, vol. 53, no. 1, pp. 23–40, 2011.

[16] R. A. Clarke and R. K. Knake, Cyber war: The Next Threat to National Security and What to Do About It, 1st ed. HarperCollins, 2011.

[17] T. Ring, "Connected cars - the next targe tfor hackers," Network Security, vol. 2015, no. 11, pp. 11–16, Nov. 2015.

[18] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," IEEE Transactions on Intelligent Transportation Systems, vol. PP, no. 99, pp. 1–18, 2017.

[19] B. M. Macq and J.-J. Quisquater, "Cryptology for digital TV broadcasting," Proceedings of the IEEE, vol. 83, no. 6, pp. 944–957, Jun 1995.

[20] The Intercept, "The Great Sim Hack," 2015, accessed: 2017-02-01. [Online]. Available: https://theintercept.com/2015/02/19/great-sim-heist/

[21] M. J. Covington and R. Carskadden, "Threat implications of the internet of things," in 2013 5th International Conference on Cyber Conflict (CYCON 2013), June 2013, pp. 1–12.

[22] M. O'Neill, "The internet of things: do more devices mean more risks?" Computer Fraud & Security, vol. 2014, no. 1, pp. 16 – 17, 2014.

[23] Symantec, "Dragonfly: Western energy sector targeted by sophisticated attack group," 2017, accessed: 2017-10-01. [Online]. Available: https://www.symantec.com/connect/blogs/dragonfly-western-energy-sector-targeted-sophisticated-attack-group

[24] T. Plos, M. Hutter, and M. Feldhofer, "Evaluation of side-channel preprocessing techniques on cryptographic-enabled HF and UHF RFID-tag prototypes," in Workshop on RFID Security – RFIDSEC 2008, 2008, pp. 114–127.

[25] I. Frieslaar and B. Irwin, "Investigating multi–thread utilization as a software defence mechanism against side channel attacks," in Proceedings of the 8th International Conference on Signal Processing Systems, ser. ICSPS 2016. New York, NY, USA: ACM, 2016, pp. 189–193.

[26] D. Genkin, I. Pipman, and E. Tromer, "Get your hands off my laptop: Physical side-channel key-extraction attacks on PCs," in Proceedings of the 16th International Workshop on Cryptographic Hardware and Embedded Systems — CHES 2014 - Volume 8731. New York, NY, USA: Springer-Verlag New York, Inc., 2014, pp. 242–260.

[27] P. Belgarric, P.-A. Fouque, G. Macario-Rat, and M. Tibouchi, "Side-channel analysis of Weierstrass and Koblitz curve ECDSA on Android smartphones," in Proceedings of the RSA Conference on Topics in Cryptology – CT-RSA 2016 – Volume 9610. New York, NY, USA: Springer-Verlag New York, Inc., 2016, pp. 236–252.

[28] D. Genkin, L. Pachmanov, I. Pipman, E. Tromer, and Y. Yarom, "ECDSA key extraction from mobile devices via nonintrusive physical side channels," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1626–1638.

[29] D. Aboulkassimi, M. Agoyan, L. Freund, J. Fournier, B. Robisson, and A. Tria, "Electromagnetic analysis (EMA) of software AES on Java mobile phones," in 2011 IEEE International Workshop on Information Forensics and Security, Nov 2011, pp. 1–6.

[30] N. Golyandina and A. Zhigljavsky, Singular Spectrum Analysis for time series, 2013th ed. Springer Science & Business Media, 2013.

[31] Y. Nakano, Y. Souissi, R. Nguyen, L. Sauvage, J.-L. Danger, S. Guilley, S. Kiyomoto, and Y. Miyake, "A pre-processing composition for secret key recovery on Android smartphone," in Proceedings of the 8th IFIP WG 11.2 International Workshop on Information Security Theory and Practice. Securing the Internet of Things – Volume 8501. New York, NY, USA: Springer-Verlag New York, Inc., 2014, pp. 76–91.

[32] J. Balasch, B. Gierlichs, O. Reparaz, and I. Verbauwhede, "DPA, bitslicing and masking at 1 GHz," in Cryptographic Hardware and Embedded Systems - CHES 2015 - 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings, 2015, pp. 599–619.

[33] J. Longo, E. De Mulder, D. Page, and M. Tunstall, "SoC it to EM: Electromagnetic side-channel attacks on a complex System-on-Chip," in Cryptographic Hardware and Embedded Systems – CHES 2015: 17th International Workshop, Saint-Malo, France, September 13-16, 2015, Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 620–640.

[34] I. Frieslaar and B. Irwin, "Recovering AES-128 encryption keys from a Raspberry Pi," in Southern Africa Telecommunication Networks and Applications Conference (SATNAC), September 2017, pp. 228–235.

[35] ——, "A multi-threading approach to secure verifypin," in 2016 2nd International Conference on Frontiers of Signal Processing (ICFSP), Oct 2016, pp. 32–37.

[36] ——, "Evaluating the multi-threading countermeasure," International Journal of Computer Science and Information Security, vol. 14, no. 12, pp. 379–387, 2016.

[37] J. L. Hennessy and D. A. Patterson, Computer Architecture: A Quantitative Approach, 5th ed. Morgan Kaufmann, 2011.

[38] D. A. Patterson and J. L. Hennessy, Computer organization and design: the hardware/software interface, 2nd ed. Morgan Kaufmann Publishers Inc., 1998.

[39] B. R. Rau and J. A. Fisher, Instruction-level parallelism, 1st ed. John Wiley and Sons Ltd., 2001.

[40] W. Kunikowski, E. Czerwiński, P. Olejnik, and J. Awrejcewicz, "An overview of ATmega AVR microcontrollers used in scientific research and industrial applications," Pomiary Automatyka Robotyka, vol. 19, no. 215, pp. 15–20, 1 2015.

[41] A. S. Tanenbaum, A. S. Woodhull, A. S. Tanenbaum, and A. S. Tanenbaum, Operating systems: design and implementation, 3rd ed. Prentice-Hall Englewood Cliffs, NJ, 1987.

[42] B. Barney, "POSIX threads programming," 2017, accessed: 2017-03-01. [Online]. Available: https://computing.llnl.gov/tutorials/pthreads/

[43] M. A. Ellis and B. Stroustrup, The Annotated C++ Reference Manual, 1st ed. Addison-Wesley Longman Publishing, 1990.

[44] J. Reinders, Intel threading building blocks: outfitting C++ for multi-core processor parallelism, 1st ed. O'Reilly Media, 2007.

[45] L. Dagum and R. Menon, "OpenMP: an industry standard API for shared-memory programming," IEEE computational science and engineering, vol. 5, no. 1, pp. 46–55, 1998.

[46] J. L. Carter and M. N. Wegman, “Universal classes of hash functions,” in Proceedings of the Ninth Annual ACM Symposium on Theory of Computing, ser. STOC ’77. New York, NY, USA: ACM, 1977, pp. 106–112.

[47] M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” in Proceedings of the 16th Annual International Cryptology Conference on Advances in Cryptology, ser. CRYPTO ’96. London, UK, UK: Springer-Verlag, 1996, pp. 1–15.

[48] D. Eastlake, 3rd and P. Jones, “US Secure Hash Algorithm 1 (SHA1),” National Security Agency, United States, Tech. Rep., 2001. [Online]. Available: https://tools.ietf.org/html/rfc3174

[49] B. Preneel, “Cryptographic hash functions,” European Transactions on Telecommunications, vol. 5, no. 4, pp. 431–448, 1994.

[50] B. Sklar, Digital communications, 2nd ed. Prentice Hall, 2001.

[51] I. Frieslaar and B. Irwin, “Investigating the electromagnetic leakage from a Raspberry Pi,” in 2017 Information Security South Africa (ISSA), August 2017.

[52] ——, “Investigating the effects various compilers have on the electromagnetic signature of a cryptographic executable,” in Proceedings of the South African Institute of Computer Scientists and Information Technologists (SAICSIT 2017), ser. SACSIT 2017. New York, NY, USA: ACM, 2017.

[53] J. G. J. van Woudenberg, M. F. Witteman, and B. Bakker, “Improving differential power analysis by elastic alignment,” in Proceedings of the 11th International Conference on Topics in Cryptology: CT-RSA 2011, ser. CT-RSA’11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 104–119.

[54] R. W. Schafer, “What is a savitzky-golay filter? [lecture notes],” IEEE Signal Processing Magazine, vol. 28, no. 4, pp. 111–117, July 2011.

[55] E. Brier, C. Clavier, and F. Olivier, “Correlation power analysis with a leakage model,” in Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, MA, USA, August 11-13, 2004. Proceedings. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 16–29.

[56] A. Satorra and P. M. Bentler, “A scaled difference chi-square test statistic for moment structure analysis,” Psychometrika, vol. 66, no. 4, pp. 507–514, Dec 2001.

[57] A. Gornik, A. Moradi, J. Oehm, and C. Paar, “A hardware-based countermeasure to reduce side-channel leakage: Design, implementation, and evaluation,” IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 34, no. 8, pp. 1308–1319, Aug 2015.

[58] I. Frieslaar and B. Irwin, “Electromagnetic data from a Raspberry Pi 2 – dataset,” Aug 2017, open Science Framework. Accessed: 2017-08-01. [Online]. Available: https://osf.io/mte5q

GUIDELINES FOR ETHICAL NUDGING IN PASSWORD AUTHENTICATION

Karen Renaud[*] **and Verena Zimmermann**[†]

[*] *Division of Cyber Security, Abertay University, Scotland.* k.renaud@abertay.ac.uk

[†] *Technische Universität Darmstadt, Germany.* zimmermann@psychologie.tu-darmstadt.de

Abstract: Nudging has been adopted by many disciplines in the last decade in order to achieve behavioural change. Information security is no exception. A number of attempts have been made to nudge end-users towards stronger passwords. Here we report on our deployment of an *enriched nudge* displayed to participants on the system enrolment page, when a password has to be chosen. The enriched nudge was successful in that participants chose significantly longer and stronger passwords. One thing that struck us as we designed and tested this nudge was that we were unable to find any nudge-specific ethical guidelines to inform our experimentation in this context. This led us to reflect on the ethical implications of nudge testing, specifically in the password authentication context. We mined the nudge literature and derived a number of core principles of ethical nudging. We tailored these to the password authentication context, and then show how they can be applied by assessing the ethics of our own nudge. We conclude with a set of preliminary guidelines derived from our study to inform other researchers planning to deploy nudge-related techniques in this context.

Key words: nudge, ethics, autonomy

1. INTRODUCTION

The password is intended to be a secret shared exclusively between the password owner and the system the password controls access to. Passwords can leak for a number of reasons. They could be guessed, for example, especially if weak passwords are chosen, as users tend to do [1]. Leaked passwords permit unauthorised access to sensitive personal or organisational data in a way that is sometimes very hard to detect.

It is standard practice for organisations to offset this risk by expiring passwords on a regular basis [2, 3]. The rationale is that this curtails ongoing use of leaked passwords, and also reveals dormant accounts, thereby improving overall system security.

End users seldom contemplate password replacement with enthusiasm. They might well react to password expiry by choosing weak passwords. This, then, appears to justify password expiry, leaving organisations with no choice but to continue enforcing regular password expiry. The end result is deterioration of system security, with ever weaker passwords [4, 5]. The eventual result is weaker, not stronger, password defences.

Some have suggested a middle ground, where the expiry requirement is directly proportional to the strength of the password chosen by the user [6, 7, 8]. Under this scheme, weak passwords would expire much more quickly than strong passwords. This scheme rewards end users for strong passwords by allowing them to use the password for longer, thereby amortising the effort they put into formulating and remembering it. It also satisfies the organisation's need for measures that protect their systems.

We trialled this scheme, using an *enriched nudge* to ensure that the end users were aware of the variable expiry period, and to ensure that they were aware of being "nudged" towards stronger passwords. We wanted them to be aware of the extended expiry time scheme. Our trial proved efficacious: participants chose significantly longer and stronger passwords.

In carrying out the research we were somewhat perturbed by the fact that we were not able to find nudge-specific ethical guidelines to inform our experimentation in this context. By "ethics", we mean "*Moral principles that govern a person's behaviour or the conducting of an activity*" (OED). In this context, that refers to the way researchers should conduct nudge-related research.

We followed the generic BPS guidelines [9] to obtain ethical approval, as required by our institution, but we were left somewhat dissatisfied that we did not have more nuanced guidelines to apply. In this paper, we report on the deployment of our nudge, and we then consider how we could inform subsequent nudge trials by deriving a set of ethical guidelines specifically for this context.

The following section discusses the password expiry issue, presenting the *raison d'être* for the practice, and the consequences thereof. Then the enriched nudge, and the experiment carried out to assess its impact on end users, is presented (Section 4.) and the results reported (Section 5.). In Section 6. we report on a review we did of the ethical nudge literature, and conclude with a set of guidelines to inform and guide researchers in the password authentication discpline. Section 7. makes some recommendations about future directions for research. Section 8. concludes.

Based on: "Enriched Nudges Lead to Stronger Password Replacements ... but Implement Mindfully", by K. Renaud and V. Zimmermann which appeared in the Proceedings of Information Security South African (ISSA) 2017, Johannesburg, 16 & 17 August 2017. © 2017 IEEE

2. PASSWORD EXPIRY

It is common for organisations to require their employees to change passwords regularly,[*] and indeed this is often considered to be “good practice” [10, 11]. The reason for mandating password expiry is the belief that it improves security [12, 13].

Yet leading academics [4, 5, 14, 15], journalists [16], and standards bodies, such as NIST [17], are urging system administrators to rethink their traditional password expiry practices because it effectively weakens passwords in the long run and is not the cure-all many believe it to be.

2.1 The Burden

Passwords, in and of themselves, can impose an unacceptable burden on computer users [18]. Password expiry exacerbates this. Password expiration has two immediate consequences in terms of human behaviour, given the fact that users cannot amortise the effort involved in memorising a strong password over a long period of time.

The **first** consequence is that people are likely to drift towards ever weaker passwords with each successive change, perhaps incrementing a digit at the end of the password or appending the month and year, merely to offset the expense of memorising a new password each month [4, 5].

The **second** consequence is that they are more likely to record the password either on paper or digitally [19,20,21], because they dread forgetting it [22, 23].

Both of these consequences weaken the mechanism [24], the opposite of what password expiry is intended to achieve. It also undeniably undermines the user experience and often prevents users from accessing their accounts.

An expired password requires immediate action, perhaps when users have other urgent goals to satisfy. If they change the password in haste, merely to gain access to their account when there is some urgency, they are likely to forget the hastily-formulated password. They then have to go through the pain of password replacement, and for some systems this process is more arduous than for others. This can result in a loss of productivity and also financial cost related to help desk resources.

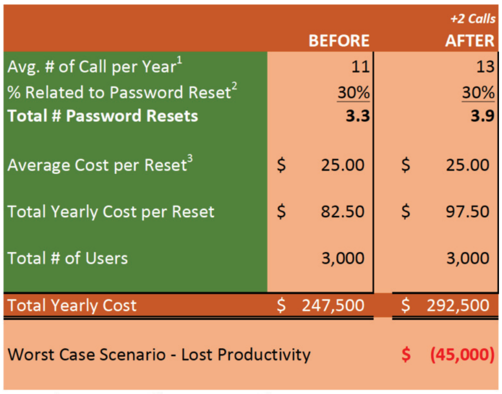
It is unsurprising that this leads to the use of weak passwords. People wish to avoid the pain and inconvenience of a replacement and act to prevent such an occurrence.

2.2 The Expense

Password expiration is expensive for end-users, but also for organisations. Password forgetting is likely to lead to an extra number of help desk calls.

If each person calls only two times more a year, for whatever reason, consider what the cost would be. Figure 1 shows a compilation of data collected from Gartner, Forrester, and the META Group by Osper [25] using this tool: `http://www.mandylionlabs.com/PRCCalc/PRCCalc.htm`. The table represents a possible cost scenario if a company were to experience just two additional help desk calls per person and year, on average.

The assumption here is that the extra calls can be related to anything, but that 30% relate to passwords, regardless of the number of calls. The numbers make it clear that any number of extra help desk calls have very real financial implications.



		+2 Calls
	BEFORE	AFTER
Avg. # of Call per Year[1]	11	13
% Related to Password Reset[2]	30%	30%
Total # Password Resets	**3.3**	**3.9**
Average Cost per Reset[3]	$ 25.00	$ 25.00
Total Yearly Cost per Reset	$ 82.50	$ 97.50
Total # of Users	3,000	3,000
Total Yearly Cost	$ 247,500	$ 292,500
Worst Case Scenario - Lost Productivity		$ (45,000)

Sources: [1] Gartner Research [2] Forrester Research [3] META Group

Figure 1: Cost scenario of just two additional password related helpdesk calls a year, on average [25]

This expenditure and undermined user experience might be warranted if expiry does indeed reduce the probability and duration of unauthorised access. This is only the case where the person who obtains the password plans to carry out long-term forays into the account. The reality is that most password thieves will carry out their nefarious activities as soon as they gain access to the password [4]. Password expiry, in these cases, is the equivalent of shutting the stable door after the horse has bolted. There are other ways of providing superior protection, and these are considered next.

2.3 The Threat

The specific threat that password expiry addresses is password leakage: it limits the period during which the leaked password can be used. Passwords can be leaked either deliberately or inadvertently (Table 1). Password expiry is not intended to address the former. If someone wishes to share a password he/she will simply share

[*] `https://www.symantec.com/connect/articles/simplest-security-guide-better-password-practices`
`http://hitachi-id.com/documents/password-management-best-practices.php`
`https://technet.microsoft.com/en-us/library/ff741764.aspx`

the newly-chosen password when the previous password expires.

Category	Action
(1) Deliberate	Owner shares password [26]
(2) Password Choice	Weak Password Guessed [27] Encrypted Weak Password stolen and easily decrypted [28]
	Strong Password Observed During Entry [29] Record of Strong Password Discovered [19]
(3) Technical Failure	Theft during transmission or storage or due to a website bug [30, 31]
(4) Deception	Phishing [32] Vishing [33]

Table 1: Password Leakage Categories

Leakage is only a major issue if the credential owner is unaware of the leakage, as and when it happens. If he or she becomes aware of it, he/she can act quickly to change the password and thereby curtail access and prevent further damage.

If the password owner does not discover the leak, the thief can keep using the password as long as there are no detectable side effects. So, for example, if a Phisher bot gets hold of a password it will probably be used pretty much immediately. The legitimate credential owner is likely to observe the side effects of such usage, and will act to curtail access.

Sometimes credentials can be used merely to snoop, which might be the case if an email password has been stolen by an ex-partner, for example. In some cases credentials are used to leapfrog into the organisation's systems and infrastructures. In this case, the side effects will not necessarily be evident in the password owner's account. If there are no immediately-observable side effects, the thief can keep using the leaked password unhindered.

2.4 Alternatives

Although there is no empirical evidence that password expiry does indeed result in more secure systems [14], it remains "good practice" in industry. It might limit the damage that can be carried out by undiscovered password leakage. What it definitely does is make those responsible for security feel that they are "doing something" to counteract insecurities caused by poor password practice. That being so, it is unlikely that organisations will discontinue this practice without compelling evidence that doing so will not weaken the security of their systems.

There are other measures that could serve a similar purpose to password expiry. Some of these alternatives perform better than expiry without the accompanying human cognitive cost. For example, the following techniques are examples of what could be used:

- **Notifications**: Some systems display an informative "last login" message whenever a person accesses their account to reveal illicit activity. This does serve to reveal password leakage if two pre-conditions are met: (1) the person notices the display and realises that he or she was not responsible for the access, (2) the legitimate user accesses the system frequently. Neither of these is, unfortunately, a given. Indeed, it is the infrequently-used account credentials that are most prized by hackers [34].

- **Alerts**: Other systems send an email to the legitimate user whenever anyone accesses the account. If the owner did not access the account, he or she is alerted and can act to terminate the hacker's access. This works unless the person accesses the system frequently, in which case it could become merely an annoyance and the emails would be ignored.

- **Multi-Password**: Some systems require an additional password whenever an action could have consequences. The hacker would then need to steal both passwords in order to carry out his/her nefarious activities, which makes things a bit harder but requires the user to memorize two passwords which increases effort and the possibility of forgetting a password.

- **Multi-Factor**: Some systems use two-factor authentication, requiring a token or biometric in addition to the password. This bolsters the mechanism and renders a single leakage less damaging. Still, tokens are costly and need to be carried around to be available upon log-in. Using biometrics as a second authentication mechanism is promising, but often also requires costly devices or sensors and has implications on privacy. Furthermore, several circumstances (e.g. age, injuries, hand cream, contact lenses) can prevent users from authenticating with biometrics [35]. The most popular fall-back mechanism then still is the password that is associated with the same problems as the multi-password approach.

- **Multi-Channel**: Some systems utilise a separate channel to authorise side-effect actions. So, for example, a hacker could steal a password and be able to log into someone's bank account to see their details. If he or she attempts to transfer money out of the account, a message is sent to the legitimate owner via another channel, perhaps a registered mobile phone, to authorise the transfer. In this way the hacker's purpose is revealed. This mechanism is often used for accounts people really care about, such as their bank accounts. While increasing the security without having to remember further passwords, this mechanism requires the user to have access to both channels at the same time which isn't always a given, e.g. when paying with a credit card in a foreign country without having access to a mobile network so that they can receive SMS messages.

- **One-Time Passwords**: Some organisations issue their users with a bespoke device that generates one-time passwords. These expire immediately and leakage is rendered a non-threat. However, the inconvenience of carrying the device around, and the expense thereof, probably limits its applicability.
- **Expire Intelligently:** Schneider [36] suggests expiring passwords only when anomalous behaviour is detected on a user's account.

Table 2 summarises this discussion.

Alternative	Advantage	Disadvantage
Notifications	Cheap	Habituation [37]; Only effective for frequently-used accounts
Alerts	Cheap	Habituation unless alerts only signal side-effect actions [38]; Habituation to receiving alerts could be exploited by Phishers
Multi-Password	Two Passwords have to be leaked; Improved Security	Expense: increases cognitive effort and doubles the number of password resets
Multi-Factor	Improved Security [39]	Expense; Not Scalable [40] Requires hardware on client that limits applicability; Biometrics have privacy implications
Multi-Channel	Improved Security [41]	Expense; Delays on some channels (eg. SMS texts) [42]
One-Time Password Device	High Security [43]	Expense and Inconvenience
Expire Intelligently	Reduces Expiry Burden	Still experimental (needs to be tested empirically)

Table 2: Password Expiry Alternatives

2.5 *Status Quo Rationale*

Many companies may consider these alternatives too complex or expensive, especially since they themselves would have to carry the implementation cost. Enforcing password expiry pushes the cost onto the end user and seems cheaper, at first glance. The fact that users migrate towards weaker passwords and thereby compromise the security of their accounts might be a consequence they feel is a reasonable trade-off. They might argue that it is the weak password holder, him or herself, who has to face the consequences. They are probably not considering the possibility that access to a hacked account can be used to leapfrog onto others, or to other organisational systems, using zero-day exploits.

2.6 *Proposed Experiment*

Our proposal was to implement a strength-dependent password expiry scheme, to determine whether this would encourage stronger password choice. In order to ensure that users were aware of this scheme, we deployed an *enriched nudge*.

Before we introduce the scheme we first review other uses of nudging reported in the research literature.

3. RELATED WORK

Nudging has enjoyed a great deal of media hype over the last few years. The UK [44] and USA [45] Governments, among others, have established units to investigate their use in the public sector. Public health is following suit [46].

The nudging technique [47] manipulates the *choice architecture* (the user interface, in this study's context) to induce people to take the wiser course of action. Nudging is a behavioural economics technique. Other fields also report techniques to change behaviours inexpensively [48, 49].

Not everyone considers nudging a worthwhile endeavour [50], but they have been successful in a range of contexts [51, 52, 53]. For example, small changes in the text of letters sent to citizens made a difference to tax payment rates [53].

What about the information security context? A security-related nudge study [54] successfully nudged users towards a more secure WiFi by using colour and menu order. This finding was confirmed by [55]. Privacy researchers have deployed nudges with some success [56, 57, 58] making people more aware of privacy invasions. People acted upon their new awareness, a strong result.

Password authentication nudge studies have not yet been as successful in delivering change [59, 60, 61, 6]. The password strength meter is also a nudge providing strength feedback, either post-entry or dynamically [62, 63, 64, 60]. Only Vance *et al.* [64] and Ur *et al.* [65] reported a positive result with these meters. The study by Ur *et al.* was an initial scoping study, using Mechanical Turk. As a next step other researchers have tested nudges in the wild and reported that the meters did not improve password strength, unless users perceived the account to be important. If people *do not* attribute value, then it is understandable that the password meter makes no difference to their choice.

Because password choice is such an important issue in the field of information security it was considered worthwhile to carry out a study to trial a previously-untested nudge in order to identify something that would indeed prove efficacious.

The study described here is part of a long-term project into the deployment of behavioural science techniques in password authentication contexts. The aim was to test the impact of authentication nudges *in the wild*, thus in a natural and realistic setting. Prior to the study reported here all the nudges we trialled were unsuccessful [66].

4. ENRICHED NUDGE

The aim of the investigation was to develop an intervention that was powerful enough to induce people to create stronger passwords. We implemented a strength-dependent password expiry scheme. To ensure that end users understood the scheme, we trialled an *enriched nudge* that comprised a three-pronged approach:

1. the first a user interface tweak (*a nudge*),
2. the second, the mainstay of economic theory: utility (*an incentive*),
3. the third (*a reminder*) at every system login to make users aware of the password expiration date.

4.1 The Theory

The *nudge* was the user interface element that communicated the scheme, the incentive was the utility, and the reminder, after each successful authentication, that ensured they were warned that their passwords were about to expire.

The idea of offering an *incentive* to encourage actions is based on the concept of utility. The fundamental idea behind neo-classical economics is that people maximise “utility” when they make choices [67]. They weigh up the benefits and costs of each choice option and choose the option that is “best” for them personally. Such an internal utility calculation is possible, and rational, if the information about the choices is complete. If the information is imperfect, on the other hand, Kelman [68] explains that fully rational choice becomes impossible. Hence this manipulation removes all uncertainty: participants were told exactly what the consequences of each choice was. It was unambiguously displayed as they typed in their password.

The idea of a *reminder* is based on the fact that people forget and are easily distracted, especially when a number of other information sources demand their attention [69, 70]. The use of a prominent notification was used to offset this tendency by displaying information about the remaining lifetime of the password every time participants logged into the system.

In effect, the idea was that participants choosing stronger passwords would have to change their passwords less frequently than those who chose weaker passwords (*the incentive*). The *nudge* would make this prominent as and when they formulated a password. The *incentive* would give them a reason (concrete utility) to choose a stronger password. The *reminder* ensured that they were reminded, frequently, about an upcoming expiry date.

4.2 The Implementation

The study was conducted with the help of a self-developed university web application that students could use to look up timetables, coursework information and coursework grades. The application was only accessible from within the campus network and with a valid student ID to prevent outsiders from attempting to use the system. Log in was possible with an alphanumeric password. No password policy or other password requirements were enforced.

Nudge: An image of a long dachshund (Figure 2) was displayed above the password entry field, both for initial creation and password replacement. The length of the dog, and the reputation of this particular breed for strength, would, it was hoped, communicate a subtle message to the participants: *go long and strong*. Even if they were unfamiliar with the breed they could hardly miss the presence of the nudge. This was calculated to draw attention to the speech bubble emerging from the dog’s mouth, telling them that the *stronger* the password, the *longer* they would be able to keep it.

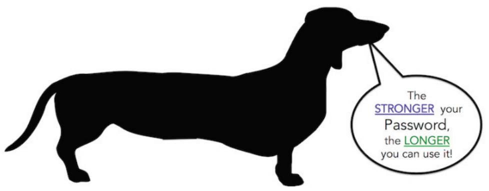
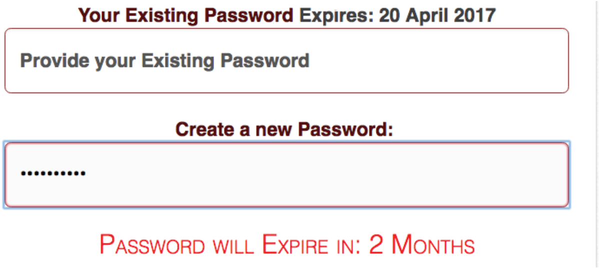


Figure 2: The Nudge

Incentive: The participants were offered extended expiration periods in relation to their password’s strength similar to the suggestion made by [6,7,8]. A survey carried out by Tam *et al.* [71], where participants responded positively to this concept, informed the decision to trial it in the wild. The utility of the password was updated and displayed, below the password field, as they typed in their password. (Figure 3). This communicates a direct benefit related to stronger password choice.



Your Existing Password Expires: 20 April 2017

Provide your Existing Password

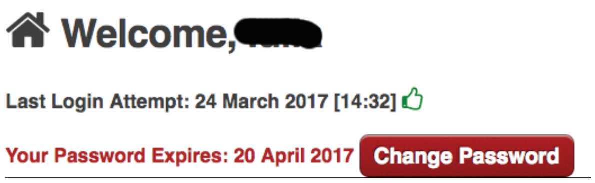
Create a new Password:

••••••••••

Password will Expire in: 2 Months

Figure 3: The Incentive

Reminder: A notification made users aware, every time they logged in, when their password would expire. A handy button was provided to facilitate convenient password changes (Figure 4).



Welcome, [redacted]

Last Login Attempt: 24 March 2017 [14:32]

Your Password Expires: 20 April 2017

Figure 4: The Reminder

Password strength was calculated using the client-based, free and open source JavaScript `zxcvbn.js` [72], a strength estimator that uses pattern matching and minimum entropy calculation. Among other values it delivers a strength score between 0 and 4 that was used in this study and that indicates whether the number of guesses required to break the password is less than 10^2 (score 0), 10^4 (score 1), 10^6 (score 2), 10^8 (score 3), or above (score 4). For example, the password "password" gets a rating of 0, where a password like "BouncyTigger92!" is issued a rating of 4. The script detects 10,000 common passwords, prevalent English words and surnames, as well as common patterns such as dates, repeats (e.g. "aaa"), sequences (e.g. "abcd"), and QWERTY patterns. Calculating strength on the client side ensured no transmission of unhashed passwords to the server. Moreover, the script is used in industry, with the popular Dropbox cloud service being a prominent user [72].

Password length was measured by the number of characters in the password.

4.3 Procedure

When choosing a password for the university web application the participating students were presented *the nudge* and *the incentive* on the registration page of the university web application as described above. Upon every access, the participants were notified of the expiry date by *the reminder*. Data was collected between September 2016 and March 2017.

Password recovery was relatively simple: the participants could request a one-time code that was emailed to their registered email address. This allowed them to define a new password and gain access to the system as painlessly as possible.

4.4 Participants

A total of 918 students, the majority of them being Computer Science students, registered to use the system that logged password change events. They logged in 10,317 times during the trial period. Of the 918, 672 opted into the study and the password strength and password length of their passwords were included in the analysis. Unfortunately, due to the requirements of the ethics committee we were not able to collect any further demographics of the students to protect their anonymity.

5. RESULTS

5.1 The Outcome

This particular enriched nudge delivered a positive result: When participants changed their passwords, either before the actual expiration date, or upon request after the expiration date, they chose significantly stronger and longer passwords than the previous one. The analysis is described in more detail below.

5.2 Password Changes

A total of 680 password changes occurred. Of these, 64% were forced changes due to expiration, 36% voluntarily changed their passwords. This could happen because the *reminder* warned them that the password was about to expire. It could also happen because the user decided spontaneously to change their password. The former is more likely. Over the previous year, when users used the same system without password expiration, only eleven voluntary password changes occurred over the six-month period. This confirms Inglesant and Sasse's finding that very few people voluntarily engage in password changing activities [19].

Most password changes happened in March 2017 (26.47%) when many coursework grades are published, and in January 2017 (24.12%) after the Christmas holidays. Fewest changes happened at the beginning of the term and in October (2.2%) and November (11.47%) 2016.

The password length before a change ranged from 1 to 24 characters with a mean of $\mu = 9.44$ and $\sigma = 2.47$ and a median of $\tilde{x} = 9$. After the change, password length ranged from 1 to 30 characters with a mean of $\mu = 10.42$ and σ

= 3.53 and a median of $\tilde{x}$ = 10. A visual inspection of the data revealed deviations from a normal distribution, thus a Wilcoxon signed rank test was conducted. The test was conducted on a significance level of α = .05. The increase in length was statistically significant with W(680) = 87827.00, p < .001. The resulting effect size of r = .20 can be interpreted as a small effect. Figure 5 shows the distribution of the password length before and after the change.

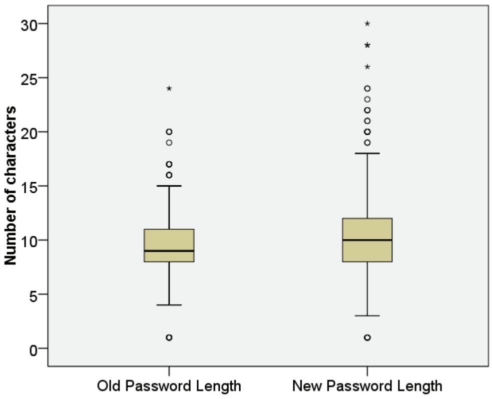


Figure 5: Visualisation of the password length before and after the change (Explanation: o indicates outliers, * indicates extreme values).

We recorded the strength of passwords calculated with the score metric of the strength estimator `zxcvbn.js` both before and after a change. As the scale of the password strength was ordinal, non-parametric Wilcoxon signed rank tests were used to evaluate the differences in password strength. Overall, the strength of changed passwords increased from a median of $\tilde{x}$ = 1 to a median of $\tilde{x}$ = 2. Figure 6 shows the password strength distribution before and after the change. The increase was statistically significant with W(680) = 22340.50, p < .001, r = .27. In more detail, the effect was significant for both, voluntary changes before expiration (W(245) = 9751.00, p < .001, r = .27) and forced changes after expiration (W(434) = 29759.00, p < .001, r = .27) with a similar effect size. The tests were conducted applying the Bonferroni-Hochberg-procedure for multiple comparisons correction. The effect size of r=.27 can be interpreted as a small effect that is close to the threshold of a medium effect (r = .3).

5.3 Forgotten Passwords

It was possible to log the number of forgotten password events for 672 of the participants. Of these, 282 forgot their password at least once, that is 41.96%. They forgot passwords between one and twelve times with a mean of μ = 2.01 (σ = 1.52) and a median of $\tilde{x}$ = 1. Looking at the whole group of participants, on average, 0.84 passwords were forgotten per person. Compared to a previous nudge study using the same system, in which 737 users participated, the number of people forgetting passwords increased. In the previous study, only 219 of the 737 participants forgot their passwords at least once, that is 29.72%. In that study, on average, 0.49 passwords were forgotten per user.

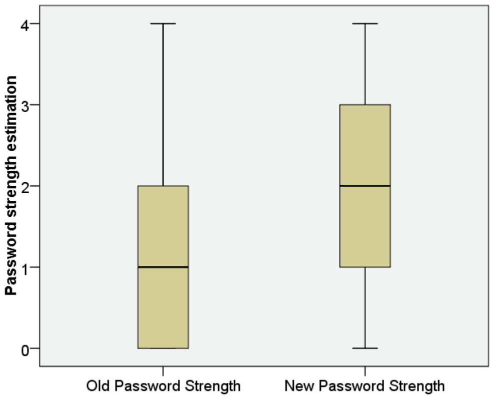


Figure 6: Visualisation of the password strength before and after the change.

5.4 Limitations

It is possible that the participants simply reused other strong passwords they already knew [73]. They could also have written down the password, a fairly common response to being confronted with a complex password that people know they are likely to forget. None of these coping mechanisms are easily measured and so we cannot be certain of the extent of their deployment.

The sample in this study consisted of university students that were mainly enrolled in technical courses such as Computer Science. Thus, the sample might be somewhat skewed towards being young and more technically adept than the average user.

The web application allowed us to collect realistic password data of actual users in contrast to more or less artificial data that could be collected with common survey platforms. This, on the one hand, increases the external validity of the study but, on the other hand, prevented us from collecting further demographic information or reasons for people's password choices. This could be valuable data to be collected in future studies.

6. ETHICAL DISCUSSION

This study showed that computer users can indeed be nudged to create stronger passwords. However, the stronger passwords also led to an increase of about 10% in the number of people who forgot passwords, replicating the findings of other studies [74]. Having nudged participants towards stronger passwords, and having observed the compromised user experience that resulted from our nudge trial, we owed it to our participants to reconsider Sunstein's suggestion that we contemplate whether nudging is indeed warranted in this context [75].

6.1 Requirements from the Nudge Literature

We ground this discussion by considering the definition of nudging as provided by Thaler and Sunstein [76]: *"Nudges are ways of influencing choice without limiting the choice set or making alternatives appreciably more costly in terms of time, trouble, social sanctions, and so forth"*. Two requirements emerge from this definition: (1) the existing available choices must be retained, and (2) the possible choices should be more or less equivalent, in terms of "cost" to the decision maker. Although not included in their definition, Thaler and Sunstein also explain that nudges should be used "for good". Indeed, Hansen and Jespersen [77] report that Thaler signs each copy of their book, *Nudge*, with the words "Nudge for good". Hence a third requirement is revealed: (3) the need for nudgers to ensure that the choice they are nudging people towards is actually beneficial *as judged by the nudgees themselves*.

What about how nudges actually exercise their influence? Nudges, according to Nys and Engelen [78], exploit predictable cognitive biases and heuristics in order to influence people towards wiser actions. This is also emphasised by Saghai [79], who explains that nudges trigger people's automatic cognitive processes. This targeting of automatic processes is reminiscent of the dual processing model proposed by Kahneman and Tversky [80]. They explain that processing happens at System 1 and System 2 levels. System 1 is the automatic and quick way of processing, and System 2 the reflective and more time-consuming kind of thinking. Humans prefer to operate on the System 1 level, and only engage System 2 when they have to because it is cognitively expensive.

System 1 nudges are processed automatically and they, and their impact, might not be transparent to nudgees. Such nudge transparency is certainly something that concerns many researchers [78, 81, 82].

An example of a System 1 nudge is to use smaller plates in canteens to reduce calorie intake [83]. People do not notice the change, and respond by eating less, but are probably unaware of their response to the smaller plate.

Nys and Engelen [78] also argue for transparency of nudges as a pre-requisite for their ethical deployment. They believe that people should be aware of the presence of the nudge. Indeed, Thaler and Sunstein [76] explain that adherence to Rawls' Publicity principle [84], i.e. full disclosure of the presence of the nudge and willingness to defend its "goodness", make it ethically sound [85].

It could reasonably be considered devious to nudge people in a way that they are not fully aware of. On the other hand, if we restrict ourselves to System 2 nudges, we lose a large arsenal of tools that can be used to improve user behaviours. It might be better to require experimenters to justify their intention to deploy System 1 nudges rather than forbidding them altogether.

Hence the fourth requirement is: (4) nudges should be transparent to nudgees, *unless* the experimenter or nudger is able to make a compelling argument for its opacity.

Finally, Thaler and Sunstein [76] also argue that nudge designers should be able to specify which particular behavioural bias their nudge is mitigating against. This requires people to have thought about the design of the nudge before-hand, so that its impact will be *predictable*. This is our fifth requirement.

Based on the nudge literature, to qualify as an ethical nudge, a mechanism needs to meet the following requirements:

N1. The original set of options should be retained: none should be removed [76, 78].

N2. The choices offered to nudgees should be more or less equivalent in terms of cost (effort, time etc.) [77, 76].

N3. It should nudge "for good", *as judged by the nudgee* [76, 86].

N4. The nudge mechanism should be transparent [85, 78], unless the experimenter or nudger is able to make a compelling argument for its opacity.

The rationale for this is that nudgers thereby respect the autonomy of the decision maker. Autonomy means that people retain the ability to construct their own goals and values, and are free to decide, plan and act in order to satisfy their goals and in accordance with their values [87]. If nudges are not seen and perceived by the nudgee, they can not reflect on their actions and decide to act in accordance with their own goals and values.

On the other hand, humans sometimes act emotionally on the spur of the moment [88, 89, 90]. A well-designed System 1 nudge could ameliorate this tendency, thus helping the person to act as they would if they were not in a 'hot state' [91].

N5. The nudge designer must be able to explain which particular behavioural bias the nudge mitigates against; to be willing to justify their choice architecture manipulation, and predict its effects.

6.2 Judging the Enriched Nudge

Does our enriched nudge satisfy these requirements? The first requirement is satisfied because participants were technically free to choose weak passwords if they wanted to. Our enriched nudge was unmistakable and obvious to participants. It did not attempt to influence people without their knowledge. It thus also satisfies the fourth condition, of transparency.

The fifth requirement justification was that we were attempting to offset the least effort principle [92] (taking the easiest course of action without thinking about the long-term consequences). We did this by making the consequences of a weak password salient, and ensuring that they were frequently reminded of the consequence or benefit of their choices every time they logged in.

The second and third requirements warrant closer scrutiny.

(Requirement 2) Option Equivalence

First let us consider the second requirement, that available options be equivalent in terms of cost. As a proviso, we need to acknowledge that password strength is a continuum, ranging from extremely weak, such as '*123*' to very strong, such as '*If what's done is done, t'were well it were done Quickly!!!*'.

To simplify this discussion we shall refer to 'weakest' passwords as passwords on the left end of the continuum and 'stronger' passwords denoting the better passwords, those whose strength tends towards the right of the weakest passwords on the continuum. We acknowledge the fact that this is an over-simplification of the issue but feel that it allows us to present a helpful delineation of password costs for the purposes of considering the cost of a password.

It seems as if the option we were nudging people towards was the more expensive one: stronger passwords. To tease apart this initial assumption, let us consider three main password costs: *(1) time and effort to memorise* [93], *(2) time and effort to enter the password* [94] and *(3) replacement cost* [19].

At first glance, memorising a weak password appears to be less costly than memorising a strong password i.e. $m^{weakest} < m^{stronger}$ where m = *memory effort*. The situation is not that straightforward, unfortunately. Certainly the weakest password "*password*" is far easier to remember than the stronger "*h6@g2D*". On the other hand, long (and therefore stronger) passwords such as passphrases are more memorable than short complex passwords [95]. A meaningful passphrase, such as "*Blue water ocean?*", is far more memorable than a complex nonsense password such as "*h6@g2D*". There is thus no linear correlation between the strength of a stronger password and the cost of memorisation.

Next, consider the entry cost. Stronger passwords, both long and complex, take longer to enter than the weakest ones. The *long* because there are more keys to type and the *complex* because they require the use of different parts of the keyboard [29]. Hence $e^{stronger} > e^{weaker}$ where e = *password entry effort*. On the other hand, it is likely that typing speeds improve the more often a person enters a particular password. If someone chooses to reuse a well-known stronger password from another system, the entry cost would not necessarily be significantly higher despite being stronger than the weakest password. Moreover, entry is only costly if the password is entered manually. If a browser add-on password manager is used, or the browser remembers the password, this cost is either non-existent or trivial. Once again, there is no linear correlation between weak and stronger passwords in this respect.

On a superficial level, if we discount the complexity of quantifying m and e correctly, we offered participants to a choice between Option 1 [weak password cost=$\sum_{i=1}^{n} c$] and Option 2 [stronger password cost = $0|c$] where c = *replacement effort*. c is composed of $\{p,n\}$, *p=cost of engaging with system's process to recover forgotten password* and *n=cognitive cost of coming up with a new password*. If the person deliberately changes the password, then p=0, but if they are locked out, they have engage with a password reset process, making $p > 0$.

Yet m, e and c are not orthogonal. Consider the following options that could be chosen by someone being asked to come up with a password in a real-life setting. Figure 7 presents the different options.

Option A: Reuse a stronger password from another system. The person chooses the reduced replacement effort, but benefits from the password requiring no effort to memorise ($m = 0$) and reduced entry effort because they have practised entering it: $e^{reused\ stronger} < e^{new\ stronger}$.

Option B: Choose a weak password. In this case m and e are minimised, but not 0. This is preferred to the costs related to a stronger password, and this preference is not affected by the consequent replacement cost of $\sum_{i=1}^{n} c$.

Option C: Choose a stronger password and memorise it. This is the opposite of option B. Here the person chooses to minimise the replacement cost while accepting that m and e are more costly than they would be for a weaker password.

Option D: Choose a stronger password and record it, perhaps by making a written record. Here $m = 0$ and $e^{stronger} > e^{weaker}$ but replacement cost is minimised.

Option E: Choose a stronger password and allow the browser to store it. Here $m = 0$ and $e = 0$, which explains why this option is so popular.

Option F: Use a browser-installed password manager to generate a stronger password, store it and populate the password entry field when required. The cost to the user is the same as Option E, but far more secure.

Option G: Use a password manager on a Smartphone to manage stronger passwords. This removes the need to memorise or record it, but the password still has to be entered manually. Here $m = 0$ and $e^{stronger} > e^{weaker}$ and replacement cost is minimised.

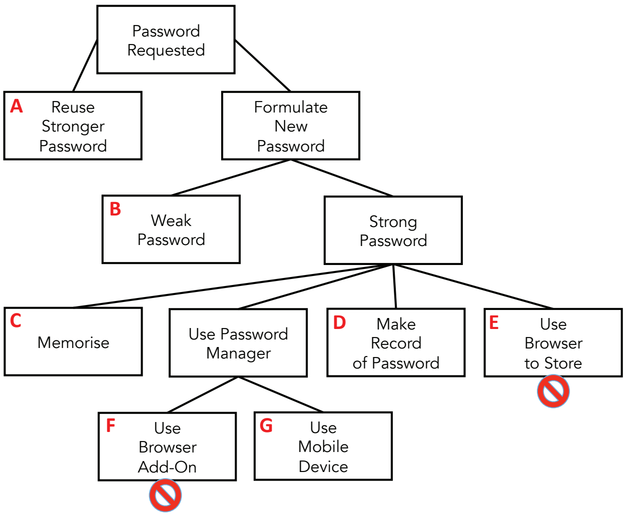


Figure 7: Password Choice Options & Costs

In our experiment, participants were not permitted to install browser add-ons nor to let the browser remember their passwords on lab machines. This means options E and F were not available. Students are permitted to connect their own laptops to eduroam in the lab, but the student system is only accessible from the University network, not eduroam, so that they could not use their own laptop browser to access the system. Hence the password choice options open to our participants were A, B, C, D and G.

It should also be mentioned, at this stage, that c (the replacement cost) was minimised. Participants simply requested a one-time code be sent to their email address, and then used that code to effect a password reset. It would be interesting to run another longitudinal experiment to ascertain whether a more arduous process would encourage even greater adoption of stronger passwords that are more durable.

An empirical investigation into the exact nature of the trade-offs people make between these costs should be the subject of a future study. Without such an investigation, we cannot argue that we either satisfied or violated the second ethical requirement.

(Requirement 3) Nudge for Good

The third requirement, to nudge "for good", is equally hard to judge. On the one hand, it is undeniable that weak passwords compromise system security. A serious hacking attack seldom affects only one computer user. Hackers will generally use one compromised account as a stepping stone to other accounts or other software systems on the same infrastructure. Because one person's poor password choice thus potentially has an impact on others we might well feel justified in nudging individuals towards stronger passwords.

On the other hand, the fact that we deployed nudges implicitly suggests that we thought the participants were either unaware of the need for strong passwords or not willing or able to transfer their intention of creating strong passwords into behaviour. The potential reasons for that are numerous and might even be reasonable from the user's point of view given his/her decision context and knowledge.

For example, users might underestimate the risk of an attack, shy away from the cost of stronger passwords, perceive the data as not having enough value to warrant a stronger password, or have a perception of a strong password that does not match actual password security.

Still, whatever the reason for choosing weaker passwords, the underlying assumption is that computer users are unable to choose a password to match the actual risk associated with the value of the asset and the vulnerabilities represented by a particular password. Deploying nudges might therefore be considered to demonstrate a lack of respect for the participants; a paternalistic intervention that does not respect human autonomy [96].

Hall and Carter [97] admit that nudges infringe autonomy but argue that it is justified to offset the nudges used by others because nudges are intended "for good" [77]. Moreover, Gordijn and Ten Have [98] claim that autonomy has not proved the "cure-all" for all ethical issues in society. Brooks [99] also acknowledges the autonomy issue but argues that nudges are inescapable as no choice architecture can be completely neutral. Thus, from his point of view the question is not whether to nudge, but how to do it in an ethical way. He argues for better mechanisms for obtaining informed consent and for nudge transparency. Finally, Norman [100] argues that it is not autonomy, *per se*, that is important, but rather that people can indeed retain some measure of control, and that they understand and have knowledge of what they are able to control.

With respect to our experiment, we did indeed obtain informed consent and participants could opt out of the study by checking a box during enrolment and all our interventions were fully visible. Our participants thus retained full control.

Still, this third requirement creates the most difficulty and brings us back to the initial question: "*were we justified in deploying the nudge at all?*" Would our participants have considered the choice we nudged them towards as being *for their own benefit*? If not, should we not allow end users to decide on how strong to make their password, without interference? Were we nudging "for good", which Thaler and Sunstein consider core to the ethical nudge [101]?

The real difficulty we have with nudging in this domain is that nudges are required to align with the end user's judgement of personal benefit, or Thaler's "goodness". The push to stronger passwords is generally good for the organisation, for the current software system, for the other users. Is it good for the individual, and should we be concerned about that? Conly [102] argues that when our actions can have a negative impact on others we can have no expectation of autonomy. As detailed previously, poor password choice can indeed have an impact on others.

This is a complex issue, with researchers arguing both for and against the humans having the right to individual autonomy [94, 103] and whether nudging violates this or not [96, 76, 102].

We therefore reserve judgement with respect to whether our enriched nudge justifiably infringed end-user autonomy, or not.

6.3 Authentication Requirements

Whereas nudging in other contexts has focused on simplifying and easing behaviours [53], that is not always an option in authentication, especially when it comes to password authentication.

Authentication is a context that operates under different constraints from other contexts. For example, in many contexts the aim is to simplify the process the human needs to engage with [104]. Some of the most common techniques in achieving this is to maximise feedback on actions [105], and to ensure that people can recover from errors [106, 107].

Now consider the authentication context. The system is interacting with someone who is required to prove his or her identity. An error could be a signal of an intrusion attempt, and the system cannot help the person to recover from his or her error. Feedback is minimal in this context, too, for the same lack of trust in the person interacting with the system.

How about nudging in authentication? The option we are nudging people towards, in the context of stronger passwords, must be warranted. We are trying to influence users towards expending more effort, and to do so every time they use the system. Hence:

A1. The nudge designer *must* be able to argue that nudges that encourage such efforts are indeed justified.

A2. Nudge proposers should have to make an argument for deploying nudging at all, purely based on the value of the resource being protected by the password, *to the nudgee him or herself*. User effort, with respect to passwords, is not free as many developers seem to believe.

A3. Those deploying nudges have to monitor the impact of the nudge by checking for the number of users being locked out, the impact of the nudge on password strength and the satisfaction of the users. If there are unanticipated and negative side effects the nudge can be disengaged.

We also need to remove one of the previously-listed principles. It is not possible for the user's options to be equivalent with a password authentication nudge. The pure nudge requirement for retention is not feasible and N2 will therefore be dropped.

6.4 Ethical Password Authentication Guidelines

The information security field does indeed consider efforts to persuade people to choose stronger passwords worthwhile. We do not suggest that researchers have deployed any unethical nudges in the information security context, but there is undoubtedly scope for unethical use of nudges in this context. This is especially true if nudgers do not have applicable guidelines to inform their deployment.

If we concede that nudges, as a technique, are indeed acceptable in password authentication, we should ensure that such nudges, in an organisational and real-time setting, exhibit a number of characteristics.

We believe it would be helpful to encode the principles we have derived from the literature as a list, to inform and assist researchers working in this area.

> The first over-arching directive is that organisations should prioritise the use of technical measures to scaffold and bolster the security of the system as much as possible before focusing their efforts on changing end-user behaviour.

Having exhausted technical measures and having judged that end users need to use stronger passwords, it should be ensured that password nudges are:

N1: retentive: End users must still be able to vary password strength to resist the influence of the nudge. It might be necessary to mandate a particular minimum password strength, and then conceivably use the nudge to encourage passwords that exceed the minimum. However, the weaker options should still be available so that the nudge respects their autonomy and agency.

N3: respectful : Choice architecture manipulations should respect user autonomy. Users should never feel that they have no agency when interacting with an authentication system. Otherwise the system risks triggering a reactance response. If the nudger can make a coherent argument for deploying an opaque System 1 nudge, and is able to satisfy the ethics review board that this is essential, then it is crucial for nudgees to be apprised of the manipulation once the experiment is over.

They should be told that: (1) they were nudged, (2) why this was done, (3) what the nudge was intended to achieve, and (4) what the impact of the nudge was over all participants.

N4: transparent : Nudgees ought to be fully aware of the nudge and the influence it is attempting to exert. If this would negate the influence of the nudge, it is necessary for the nudge designer to argue convincingly that this is the case, and for the Ethics Review Board to be persuaded of the need for opacity. For example, if the experimenter is concerned about the impact of demand characteristics [108], that could be a reason to make the nudge opaque.

The display of a password strength meter meets the transparency requirement, as does the enriched nudge we tested. One could imagine someone using a scary background on the web page subliminally to induce a fear of hacking and thereby attempting to nudge people towards stronger passwords. Such a nudge would not be transparent and therefore questionable as far as ethics is concerned.

N5: defensible : It must be trivial for nudgees to contact those deploying the nudge should they have any questions or concerns about it. This is in line with Rawl's Publicity principle [84] and requires nudgers to be able to justify the behavioural biases they are attempting to ameliorate with the nudge.

A1: justified : Strong passwords have a cost associated with them and user cost is not free. Nudgers should be aware of the fact that users may rationally respond to demands for greater strength by deploying less secure practices, such as allowing their browser to remember the password. This weakens the security of the system, while putting a greater burden on the end users. Nudgers should, if possible, ensure that the nudgees are apprised of the motivations for the nudge so that they understand why they are being asked to put extra effort into authenticating.

A2: sufficient : Nudging should only be deployed when the asset being protected requires stronger passwords than the *status quo* average password, based on the previous usage of a system. Designers should put some thought into applying a rule such as "*require passwords to be as strong as needed, as matched to the value of the asset, but no stronger*".

A3: monitored : A number of monitoring set points should be defined and adhered to after roll-out to carry out close examination so as to detect unexpected and undesirable side effects. Amelioration or abandonment should be considered seriously if the user experience is being compromised unacceptably.

So, for example, the number of forgotten passwords should be monitored and compared to the usual number, to determine whether the user experience is being impoverished. Other data should also be scrutinised, such as the number of logins, the password strength profile and any complaints from the users.

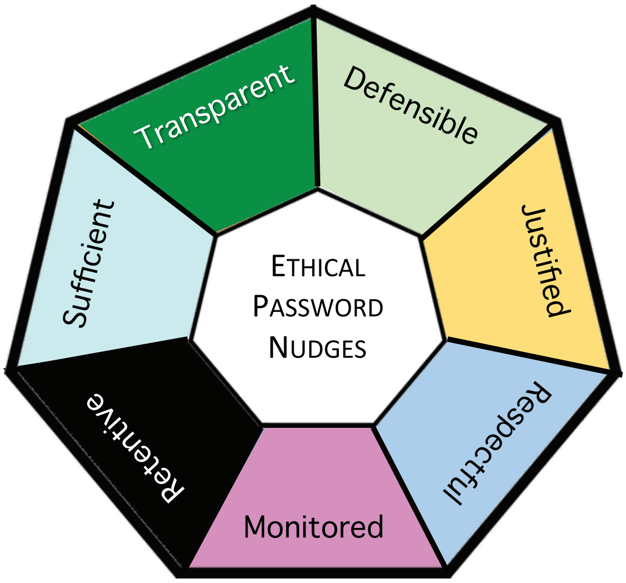


Figure 8: Ethical Password Nudge Characteristics

6.5 Summary

We started off this section by contemplating whether nudging is indeed warranted in this context, as advised by Sunstein [75]. Our investigation required us to consider the ethics of nudging. As we perused the literature five distinct requirements of ethical nudges emerged. We extended this with three authentication-specific ethical nudge principles, dropping one of the previous five principles. In total we arrived at seven ethical principles for nudging in information security.

We assessed the ethics of our enriched nudge using these as metrics and uncovered difficulties related to two areas. The first is the requirement for nudge options to be equivalent, and the second that the nudge be intended for the "good" of the nudgee.

Ajudging the latter seems to come down to the fact that it is considered necessary by the information security community to justify the deployment of these kinds of techniques when individuals' unwise actions can have serious and undesirable side effects on others. Weak password choice can lead to compromises that impact large numbers of people. Is it acceptable to violate autonomy for the greater good? Conly [102] would say yes, White [96] would disagree.

We conclude the paper with a list of ethical password nudge requirements, intended for those who decide that password nudging is indeed ethical and warranted. We do not yet make a strong argument for, or against, nudging in this context, in particular because of the fact that they could be considered to infringe autonomy.

7. FUTURE WORK

There is scope for further work in a number of directions.

Autonomy

We plan to carry out a more extensive investigation into the meaning of autonomy in this context and the meaning of potential violation that nudges can commit in the information security context.

Change Costs

The change costs, in our experiment, were as low as we could reasonably make them. It would be interesting to run the experiment again with a more expensive replacement process in order to see what impact that would have on password strength.

Password Costs

An investigation into the interplay between the different aspects making up actual password cost would be very insightful and is something worth pursuing.

Generalising the Ethical Guidelines

It would obviously be helpful if we were able to produce a more general set of guidelines to inform other research areas within Information Security and Privacy, and we plan to pursue this goal next.

8. CONCLUSION

In this paper we report on an investigation into the efficacy of an *enriched nudge*, comprising a nudge, an incentive and a reminder, in terms of influencing people towards choosing stronger passwords.

The focus of this paper was on learning lessons from our experiences, and from the nudge literature, in order to derive nudge-specific ethical guidelines. Our purpose was to provide guidance to other researchers experimenting with nudges in authentication, and to ethics review boards having to assess and approve research proposals. We thus conclude this paper with a set of ethical guidelines for nudging in password authentication and we demonstrate how these can be applied.

9. ACKNOWLEDGMENTS

We obtained ethical approval from the College of Science and Engineering at the University of Glasgow to carry out nudge-related research on the website (Approval #300140006). We wish to thank the support staff in the School of Computing Science for their advice and assistance during the course of this research.

This work was supported by the German Federal Ministry of Education and Research (BMBF) as well as by the Hessen State Ministry for Higher Education, Research and the Arts (HMWK) within CRISP (Center for Research in Security and Privacy).

REFERENCES

[1] D. Florencio and C. Herley, "A large-scale study of web password habits," in *Proceedings of the 16th international conference on World Wide Web*. ACM, 2007, pp. 657–666.

[2] E. H. Spafford, "Opus: Preventing weak password choices," *Computers & Security*, vol. 11, no. 3, pp. 273–278, 1992.

[3] ——, "Preventing weak password choices," Computer Science Technical Reports, Tech. Rep. Paper 875, 1991, http://docs.lib.purdue.edu/cstech/875.

[4] S. Chiasson and P. C. Van Oorschot, "Quantifying the security advantage of password expiration policies," *Designs, Codes and Cryptography*, vol. 77, no. 2-3, pp. 401–408, 2015.

[5] Y. Zhang, F. Monrose, and M. K. Reiter, "The security of modern password expiration: An algorithmic framework and empirical analysis," in *Proceedings of the 17th ACM conference on Computer and Communications Security*. ACM, 2010, pp. 176–186.

[6] T. Seitz, E. von Zezschwitz, S. Meitner, and H. Hussmann, "Influencing Self-Selected Passwords Through Suggestions and the Decoy Effect," in *EuroUSEC*. Darmstadt: Internet Society, 2016.

[7] G. R. Walters, "Variable expiration of passwords," USA Patent US 7 200 754 B2, US20040177272, https://www.google.com/patents/US7200754.

[8] R. Childress, I. Goldberg, M. Lechtman, and Y. Medini, "User policy manageable strength-based password aging," USA Patent, Feb. 5, 2013. [Online]. Available: https://www.google.com/patents/US8370925

[9] The British Psychological Society, "Code of human research ethics," 2014, http://www.bps.org.uk/publications/policy-and-guidelines/research-guidelines-policy-documents/research-guidelines-poli.

[10] D. A. Curry, *Unix system security: a guide for users and system administrators*. Addison-Wesley Longman Publishing Co., Inc., 1992.

[11] SANS Institute, "Password protection policy," https://www.sans.org/security-resources/policies/general/pdf/password-protection-policy.

[12] W. Cheswick, "Rethinking passwords," *Queue*, vol. 10, no. 12, pp. 50:50–50:56, Dec. 2012.

[13] C. Herley and P. Van Oorschot, "A research agenda acknowledging the persistence of passwords," *IEEE Security & Privacy*, vol. 10, no. 1, pp. 28–36, 2012.

[14] M. Bishop, "Best practices and worst assumptions," in *Proceedings of the 2005 Colloquium on Information Systems Security Education (CISSE) pp*, 2005, pp. 18–25.

[15] L. Cranor, "Time to rethink mandatory password changes," 2016, https://www.ftc.gov/news-events/blogs/techftc/2016/03/time-rethink-mandatory-password-changes.

[16] K. Hickey, "Mandatory password changes – not as secure as you think," 2016, https://gcn.com/articles/2016/06/07/mandatory-password-changes.aspx.

[17] P. A. Grassi, J. L. Fenton, E. M. Newton, R. A. Perlner, A. R. Regenscheid, W. E. Burr, J. P. Richer, N. B. Lefkovitz, J. M. Danker, Y.-Y. Choong, K. K. Greene, and M. F. Theofanos, "NIST Special Publication 800-63B. Digital Identity Guidelines Authentication and Lifecycle Management," 2017, https://pages.nist.gov/800-63-3/.

[18] M. A. Sasse, "'Technology Should Be Smarter Than This!': A Vision for Overcoming the Great Authentication Fatigue," in *Workshop on Secure Data Management*. Springer, 2013, pp. 33–36.

[19] P. G. Inglesant and M. A. Sasse, "The true cost of unusable password policies: password use in the wild," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2010, pp. 383–392.

[20] W. C. Summers and E. Bosworth, "Password policy: the good, the bad, and the ugly," in *Proceedings of the Winter International Symposium on Information and Communication Technologies*. Trinity College Dublin, 2004, pp. 1–6.

[21] K. Renaud, "Blaming noncompliance is too convenient: What really causes information breaches?" *IEEE Security & Privacy*, vol. 10, no. 3, pp. 57–63, 2012.

[22] C. L. Huntley, "A developmental view of system security," *Computer*, vol. 39, no. 1, pp. 113–114, 2006.

[23] C. Herley, "So long, and no thanks for the externalities: the rational rejection of security advice by users," in *Proceedings of the 2009 workshop on New Security Paradigms Workshop*. Colorado: ACM, 2009, pp. 133–144.

[24] J. Bonneau, C. Herley, P. C. van Oorschot, and F. Stajano, "Passwords and the evolution of imperfect authentication," *Communications of the ACM*, vol. 58, no. 7, pp. 78–87, 2015.

[25] J. Osper, "Password expiration policy best practices," 2016, 4 May https://www.portalguard.com/blog/2016/05/04/password-expiration-policy-best-practices/.

[26] K. Renaud, R. Blignaut, and I. Venter, "Smartphone owners need security advice. how can we ensure they get it?" in *International Conference on Information Resources Management (CONF-IRM)*, 2016.

[27] L. Gong, M. A. Lomas, R. M. Needham, and J. H. Saltzer, "Protecting poorly chosen secrets from guessing attacks," *IEEE Journal on Selected Areas in Communications*, vol. 11, no. 5, pp. 648–656, 1993.

[28] J. Bonneau, "The science of guessing: analyzing an anonymized corpus of 70 million passwords," in *Security and Privacy (SP), 2012 IEEE Symposium on*. IEEE, 2012, pp. 538–552.

[29] F. Tari, A. Ozok, and S. H. Holden, "A comparison of perceived and real shoulder-surfing risks between alphanumeric and graphical passwords," in *Proceedings of the Second Symposium on Usable Privacy and Security*. ACM, 2006, pp. 56–66.

[30] L. Falk, A. Prakash, and K. Borders, "Analyzing websites for user-visible security design flaws," in *Proceedings of the 4th symposium on Usable Privacy and Security*. ACM, 2008, pp. 117–126.

[31] C. Yue and H. Wang, "Characterizing insecure javascript practices on the web," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 961–970.

[32] K. Shah, "Phishing: An evolving threat," *International Journal of Students' Research in Technology & Management*, vol. 3, no. 1, pp. 216–222, 2015.

[33] E. O. Yeboah-Boateng and P. M. Amanor, “Phishing, SMiShing & Vishing: an assessment of threats against mobile devices,” *Journal of Emerging Trends in Computing and Information Sciences*, vol. 5, no. 4, pp. 297–307, 2014.

[34] C. Stoll, *The Cuckoo’s Egg: Tracking a Spy through the Maze of Computer Espionage*. Gallery Books, 2005.

[35] N. C. Sickler and S. J. Elliott, “An evaluation of fingerprint image quality across an elderly population vis-a-vis an 18-25 year old population,” in *39th Annual 2005 International Carnahan Conference on Security Technology, 2005. CCST’05.* IEEE, 2005, pp. 68–73.

[36] J. P. Schneider, “Managing password expiry,” USA Patent US 8 959 618 B2, Feb 17, 2015, https://www.google.com/patents/US8959618.

[37] C. Bravo-Lillo, S. Komanduri, L. F. Cranor, R. W. Reeder, M. Sleeper, J. Downs, and S. Schechter, “Your attention please: designing security-decision uis to make genuine risks harder to ignore,” in *Proceedings of the Ninth Symposium on Usable Privacy and Security*. ACM, 2013, pp. 6–24.

[38] S. Breznitz, *Cry Wolf: The Psychology of False Alarms*. Psychology Press, 2013.

[39] D. Glynos, P. Kotzanikolaou, and C. Douligeris, “Preventing impersonation attacks in manet with multi-factor authentication,” in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium on*. IEEE, 2005, pp. 59–64.

[40] W. Harwood, “Locking up passwords–for good,” *Network Security*, vol. 2016, no. 4, pp. 10–13, 2016.

[41] C.-Y. Huang, S.-P. Ma, and K.-T. Chen, “Using one-time passwords to prevent password phishing attacks,” *Journal of Network and Computer Applications*, vol. 34, no. 4, pp. 1292–1301, 2011.

[42] M. Al Fairuz and K. Renaud, “Multi-channel, multi-level authentication for more secure ebanking.” in *Information Security South Africa*, Johannesburg, South Africa, 2010.

[43] C. Herley, P. C. Van Oorschot, and A. S. Patrick, “Passwords: If were so smart, why are we still using them?” in *International Conference on Financial Cryptography and Data Security*. Springer, 2009, pp. 230–237.

[44] The Behavioural Insights Team, “Who we are,” 2014, http://www.behaviouralinsights.co.uk/about-us/ Accessed 19 Sept, 2016.

[45] J. Holden, “Memorandum to the Heads of Executive Departments and Agencies. Implementation Guidance for Executive Order 13707: Using Behavioral Science Insights to Better Serve the American People,” Washington, DC, 2015, sept 15. Executive Office of the President. Office of Science and Technology Policy https://www.whitehouse.gov/the-press-office/2015/09/15/executive-order-using-behavioral-science-insights-\better-serve-american Accessed 19 September 2016.

[46] M. Verweij and M. v. d. Hoven, “Nudges in public health: paternalism is paramount,” *The American Journal of Bioethics*, vol. 12, no. 2, pp. 16–17, 2012.

[47] R. H. Thaler, C. R. Sunstein, and T. C. Leonard, “Nudge: Improving decisions about health, wealth, and happiness,” *Constitutional Political Economy*, vol. 19, no. 4, pp. 356–360, 2008.

[48] M. Bateson, L. Callow, J. R. Holmes, M. L. R. Roche, and D. Nettle, “Do images of ‘watching eyes’ induce behaviour that is more pro-social or more normative? A field experiment on littering,” *PloS one*, vol. 8, no. 12, p. e82055, 2013.

[49] A. Dijksterhuis, J. A. Bargh, and J. Miedema, “Of men and mackerels: Attention, subjective experience, and automatic social behavior,” in *The message within: The role of subjective experience in social cognition and behavior*, H. Bless and J. Forgas, Eds. New York: Psychology Press, 2000, ch. 3, pp. 37–51.

[50] G. Rayner and T. Lang, “Is nudge an effective public health strategy to tackle obesity? No,” *British Medical Journal*, vol. 342, 2011.

[51] A. Oliver, “Is nudge an effective public health strategy to tackle obesity? Yes,” *British Medical Journal*, vol. 342, 2011.

[52] J. K. Turland, “Aiding information security decisions with human factors using quantitative and qualitative techniques,” Ph.D. dissertation, Newcastle University, 2016.

[53] D. Halpern, *Inside the Nudge Unit: How small changes can make a big difference*. London: WH Allen, 2015.

[54] D. Jeske, L. Coventry, P. Briggs, and A. van Moorsel, “Nudging whom how: It proficiency, impulse control and secure behaviour,” in *Personalizing Behavior Change Technologies CHI Workshop*. Toronto: ACM, 27 April 2014.

[55] I. Yevseyeva, C. Morisset, and A. van Moorsel, “Modeling and analysis of influence power for information security decisions,” *Performance Evaluation*, vol. 98, pp. 36–51, 2016.

[56] E. K. Choe, J. Jung, B. Lee, and K. Fisher, "Nudging people away from privacy-invasive mobile apps through visual framing," in *IFIP Conference on Human-Computer Interaction*. Springer, 2013, pp. 74–91.

[57] R. Balebako, P. G. Leon, H. Almuhimedi, P. G. Kelley, J. Mugan, A. Acquisti, L. F. Cranor, and N. Sadeh, "Nudging users towards privacy on mobile devices," in *Proc. CHI 2011 Workshop on Persuasion, Nudge, Influence and Coercion*. ACM, 2011.

[58] H. Almuhimedi, F. Schaub, N. Sadeh, I. Adjerid, A. Acquisti, J. Gluck, L. F. Cranor, and Y. Agarwal, "Your location has been shared 5,398 times!: A field study on mobile app privacy nudging," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15. New York, NY, USA: ACM, 2015, pp. 787–796.

[59] M. Ciampa, "A comparison of password feedback mechanisms and their impact on password entropy," *Information Management & Computer Security*, vol. 21, no. 5, pp. 344–359, 2013.

[60] S. Egelman, A. Sotirakopoulos, I. Muslukhov, K. Beznosov, and C. Herley, "Does my password go up to eleven?: The impact of password meters on password selection," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Paris: ACM, 2013, pp. 2379–2388.

[61] B. M. Josiam and J. P. Hobson, "Consumer choice in context: the decoy effect in travel and tourism," *Journal of Travel Research*, vol. 34, no. 1, pp. 45–50, 1995.

[62] X. de Carné de Carnavalet, "A large-scale evaluation of high-impact password strength meters," Ph.D. dissertation, Concordia University, 2014.

[63] A. Sotirakopoulos, "Influencing user password choice through peer pressure," Ph.D. dissertation, The University Of British Columbia (Vancouver), 2011.

[64] A. Vance, D. Eargle, K. Ouimet, and D. Straub, "Enhancing password security through interactive fear appeals: A web-based field experiment," in *2013 46th Hawaii International Conference on System Sciences (HICSS)*. Hawai'i: IEEE, 2013, pp. 2988–2997.

[65] B. Ur, P. G. Kelley, S. Komanduri, J. Lee, M. Maass, M. L. Mazurek, T. Passaro, R. Shay, T. Vidas, L. Bauer *et al.*, "How does your password measure up? The effect of strength meters on password creation," in *Presented as part of the 21st USENIX Security Symposium (USENIX Security 12)*. Bellevue: USENIX, 2012, pp. 65–80.

[66] K. Renaud, V. Zimmermann, J. Maguire, and S. Draper, "Lessons learned from evaluating eight password nudges in the wild," in *LASER Workshop. Arlington. 18-19 October*, 2017.

[67] W. S. Jevons, *The theory of Political Economy*. Macmillan and Company, 1879.

[68] M. Kelman, "Choice and utility," *Wisconsin Law Review*, p. 769, 1979.

[69] S. Misra and D. Stokols, "Psychological and health outcomes of perceived information overload," *Environment and Behavior*, vol. 44, no. 6, pp. 737–759, 2012.

[70] G. Pijpers, *Information overload: A system for better managing everyday data*. John Wiley & Sons, 2010.

[71] L. Tam, M. Glassman, and M. Vandenwauver, "The psychology of password management: a tradeoff between security and convenience," *Behaviour & Information Technology*, vol. 29, no. 3, pp. 233–244, 2010.

[72] D. L. Wheeler, "zxcvbn: Low-budget password strength estimation," in *USENIX Conference 2016*. Vancouver: USENIX, August 2016, Dropbox Inc.

[73] R. Wash, E. Rader, R. Berman, and Z. Wellmer, "Understanding password choices: How frequently entered passwords are re-used across websites," in *Symposium on Usable Privacy and Security (SOUPS)*, 2016.

[74] R. Shay, S. Komanduri, P. G. Kelley, P. G. Leon, M. L. Mazurek, L. Bauer, N. Christin, and L. F. Cranor, "Encountering stronger password requirements: user attitudes and behaviors," in *Proceedings of the Sixth Symposium on Usable Privacy and Security*. ACM, 2010, p. 2.

[75] C. R. Sunstein, "Nudges that fail," *Behavioural Public Policy*, vol. 1, no. 1, pp. 4–25, 2017.

[76] R. H. Thaler and C. R. Sunstein, *Nudge: Improving decisions about health, wealth, and happiness*. Yale University Press, 2008.

[77] P. G. Hansen and A. M. Jespersen, "Nudge and the manipulation of choice: A framework for the responsible use of the nudge approach to behaviour change in public policy," *European Journal of Risk Regulation*, vol. 4, no. 1, pp. 3–28, 2013.

[78] T. R. Nys and B. Engelen, "Judging nudging: Answering the manipulation objection," *Political Studies*, vol. 65, no. 1, pp. 199–214, 2017.

[79] Y. Saghai, "Salvaging the concept of nudge," *Journal of Medical Ethics*, vol. 39, no. 8, pp. 487–493, 2013.

[80] D. Kahneman, *Thinking, Fast and Slow*. Farrar, Straus and Giroux, 2011.

[81] T. Haugh, "The ethics of intracorporate behavioral ethics," *California Law Review Online*, April 2017.

[82] C. R. Sunstein, "Fifty shades of manipulation," 2015, https://dash.harvard.edu/bitstream/handle/1/16149947/manipulation2_18.pdf?sequence=1.

[83] B. Wansink, "Environmental factors that increase the food intake and consumption volume of unknowing consumers," *Annual Review of Nutrition*, vol. 24, pp. 455–479, 2004.

[84] J. Rawls, *A Theory of Justice*. Harvard university press, 2009.

[85] C. R. Sunstein and R. H. Thaler, "Libertarian paternalism is not an oxymoron," *The University of Chicago Law Review*, pp. 1159–1202, 2003.

[86] C. R. Sunstein, "Nudges Do Not Undermine Human Agency," *Journal of Consumer Policy*, vol. 38, no. 3, pp. 207–210, 2015.

[87] B. Friedman and H. Nissenbaum, "Software agents and user autonomy," in *Proceedings of the first international conference on Autonomous agents*. ACM, 1997, pp. 466–469.

[88] J. Benhabib, A. Bisin, and A. Schotter, "Present-bias, quasi-hyperbolic discounting, and fixed costs," *Games and Economic Behavior*, vol. 69, no. 2, pp. 205–223, 2010.

[89] T. Sharot, A. M. Riccardi, C. M. Raio, and E. A. Phelps, "Neural mechanisms mediating optimism bias," *Nature*, vol. 450, no. 7166, pp. 102–105, 2007.

[90] E. Castano, V. Yzerbyt, M.-P. Paladino, and S. Sacchi, "I belong, therefore, I exist: Ingroup identification, ingroup entitativity, and ingroup bias," *Personality and Social Psychology Bulletin*, vol. 28, no. 2, pp. 135–143, 2002.

[91] P. G. Hansen, "The definition of nudge and libertarian paternalism: Does the hand fit the glove?" *European Journal of Risk Regulation*, no. 1, pp. 1–20, 2015.

[92] G. K. Zipf, *Human behavior and the principle of least effort: An introduction to human ecology*. Ravenio Books, 2016.

[93] E. Stobert and R. Biddle, "The password life cycle: user behaviour in managing passwords," in *Proc. SOUPS*, 2014.

[94] K. K. Greene, M. A. Gallagher, B. C. Stanton, and P. Y. Lee, "I cant type that! p@ w0rd entry on mobile devices," in *International Conference on Human Aspects of Information Security, Privacy, and Trust*. Springer, 2014, pp. 160–171.

[95] M. Keith, B. Shao, and P. Steinbart, "A behavioral analysis of passphrase design and effectiveness," *Journal of the Association for Information Systems*, vol. 10, no. 2, p. 2, 2009.

[96] M. White, *The manipulation of choice: Ethics and libertarian paternalism*. Springer, 2013.

[97] L. Zhang and W. C. McDowell, "Am i really at risk? determinants of online users' intentions to use strong passwords," *Journal of Internet Commerce*, vol. 8, no. 3-4, pp. 180–197, 2009.

[98] B. Gordijn and H. Ten Have, "Autonomy, free will and embodiment," *Medicine, Health Care and Philosophy*, vol. 13, no. 4, p. 301302, 2010.

[99] T. Brooks, "Should we nudge informed consent?" *The American Journal of Bioethics*, vol. 13, no. 6, pp. 22–23, 2013.

[100] D. A. Norman, "How might people interact with agents," *Communications of the ACM*, vol. 37, no. 7, pp. 68–71, 1994.

[101] C. R. Sunstein, "The ethics of nudging," *Yale Journal on Regulation*, vol. 32, p. 413, 2015.

[102] S. Conly, "Against autonomy: justifying coercive paternalism," *Journal of Medical Ethics*, vol. 40, no. 5, pp. 349–349, 2014.

[103] B. O'Neill, "A message to the illiberal nudge industry: push off," 2000, spiked, 1 November.

[104] J. Nielsen, *Designing web usability: The practice of simplicity*. New Riders Publishing, 1999.

[105] K. Renaud and R. Cooper, "Feedback in human-computer interaction-characteristics and recommendations," *South African Computer Journal*, vol. 2000, no. 26, pp. 105–114, 2000.

[106] K. P. O'Hara and S. J. Payne, "Planning and the user interface: The effects of lockout time and error recovery cost," *International Journal of Human-Computer Studies*, vol. 50, no. 1, pp. 41–59, 1999.

[107] G. Buchanan, S. Farrant, M. Jones, H. Thimbleby, G. Marsden, and M. Pazzani, "Improving mobile internet usability," in *Proceedings of the 10th international conference on World Wide Web*. ACM, 2001, pp. 673–680.

[108] A. L. Nichols and J. K. Maner, "The good-subject effect: Investigating participant demand characteristics," *The Journal of General Psychology*, vol. 135, no. 2, pp. 151–166, 2008.

NOSQL DATABASES: FORENSIC ATTRIBUTION IMPLICATIONS

W. K. Hauger[*] **and M. S. Olivier**[†]

[*] *ICSA Research Group, Department of Computer Science, Corner of University Road and Lynnwood Road, University of Pretoria, Pretoria 0002, South Africa E-mail: whauger@gmail.com*

[†] *Department of Computer Science, Corner of University Road and Lynnwood Road, University of Pretoria, Pretoria 0002, South Africa E-mail: molivier@cs.up.ac.za*

Abstract: NoSQL databases have gained a lot of popularity over the last few years. They are now used in many new system implementations that work with vast amounts of data. Such data will typically also include sensitive information that needs to be secured. NoSQL databases are also underlying a number of cloud implementations which are increasingly being used to store sensitive information by various organisations. This has made NoSQL databases a new target for hackers and other state sponsored actors. Forensic examinations of compromised systems will need to be conducted to determine what exactly transpired and who was responsible. This paper examines specifically if NoSQL databases have security features that leave relevant traces so that accurate forensic attribution can be conducted. The seeming lack of default security measures such as access control and logging has prompted this examination. A survey into the top ranked NoSQL databases was conducted to establish what authentication and authorisation features are available. Additionally the provided logging mechanisms were also examined since access control without any auditing would not aid forensic attribution tremendously. Some of the surveyed NoSQL databases do not provide adequate access control mechanisms and logging features that leave relevant traces to allow forensic attribution to be done using those. The other surveyed NoSQL databases did provide adequate mechanisms and logging traces for forensic attribution, but they are not enabled or configured by default. This means that in many cases they might not be available, leading to insufficient information to perform accurate forensic attribution even on those databases.

Key words: database forensics, forensic attribution, NoSQL, survey.

1. INTRODUCTION

In recent years NoSQL databases have gained popularity both with developers who build new systems, and within organisations who want to optimise and improve their businesses [1]. Both of those parties are trying to adapt their information systems to meet today's data demands.

Certain NoSQL databases have even moved up from being niche products to leaders in Gartners Magic Quadrant for Operational Database Management Systems [2]. Gartner considers databases in the leaders quadrant to be of operational quality. According to Gartner, leaders generally represent the lowest risk for customers in the areas of performance, scalability, reliability and support.

The forerunners of today's NoSQL databases were started by big web companies such as Google, Amazon and Facebook to help them build and support their businesses [3]. After they made these new databases public and open source, other big web companies such as Twitter, Instagram and Apple started to use them as well [3]. This has lead to the development of a number of NoSQL databases based on the ideas and models of the original databases.

The use of NoSQL databases has started to filter down to ordinary organisations who are now also starting to use NoSQL databases for various purposes in their business processes. The consequence of this is that more and more data is being placed in NoSQL databases. This includes private and sensitive information which has to be kept secure and confidential.

Additionally one big area of use for NoSQL is Big Data. As the name implies, Big Data deals with vast amounts of data that needs to be stored, analysed and retrieved. Copious amounts of this data are normally unstructured and make NoSQL databases such an attractive proposition. However, this also means that unauthorised access to such NoSQL databases has the potential to expose very large amounts of information.

Together with the rise in popularity, the increased storage of data has made these NoSQL databases attractive targets for hackers, state actors, extortionists etc. Data security has thus started to become an important aspect of NoSQL databases. Some research has already been conducted into the security features provided by these NoSQL database and found them to be lacking [4, 5].

In contrast to the previous research, this paper looks at NoSQL database security from a forensic perspective. The focus is in particular on forensic attribution. Performing forensic attribution in digital systems is difficult and inherently limited [6]. These limitations include attribution delay, failed attribution, and mis-attribution.

This is because on the one end the actions that need to be attributed occurred in the digital world, but on the other end the actors that are ultimately responsible are located in the physical world. Therefore various researchers have

Based on: "Forensic attribution in NoSQL databases", by W. Hauger and M. Olivier which appeared in the Proceedings of Information Security South African (ISSA) 2017, Johannesburg, 16 & 17 August 2017. © 2017 IEEE

proposed different levels, categories or steps of attribution that can be performed. These different levels, categories or steps not only have increasing degrees of difficulty and complexity, but they also extend different distances from the actions in the digital world to the responsible actors in the physical world. A more detailed discussion follows in section 3.

Always performing the full spectrum of attribution from the actions to the actors might not even be necessary. Clark and Landau argue that the occasions when attribution to the level of an individual person is useful are actually very limited [7]. They note that, although criminal retribution requires identifying a specific person and assigning blame, "the evidence that is finally brought into court is unlikely to be [a] 'forensic quality' computer-based identity, but rather other sorts of physical evidence found during the investigation" [7].

Keeping the previous statement in mind, a valuable first step would be to tie the actions in question to a program or process on a machine inside the digital realm [8]. However, even tying actions to processes can be difficult without enough information sets that can be correlated to form a consistent chain of events [8]. Relational databases provide one such set of information in the form of traces in various log files and in system tables [9]. This information can then be used in conjunction with other information sets from outside the database to help perform attribution of the actions that occurred inside the database.

These traces can be left by measures such as access control and logging/auditing which are normally part of the security model of all relational databases. Consequently, this paper scrutinises the security features that are available in NoSQL databases and how useful their traces can be for forensic attribution. A survey of these specific security measures in NoSQL databases was conducted to determine what information they can provide to aid with forensic attribution in the case of a forensic examination.

There are more than two hundred NoSQL database implementations available [10] and to examine the security measures of all those databases would be prohibitive. Many of those databases are still experimental or being used in low volumes. Thus only a few NoSQL database management systems (DBMSs) were chosen of which the access control and logging features were studied. The choice was based on popularity and to be representative of the main types of data models used by NoSQL databases. Section 2 provides more details about the main NoSQL data types.

The NoSQL DBMSs examined were MongoDB, Cassandra, Redis and Neo4j. These selected NoSQL databases are among the most popular based on the number of web pages on the Internet according to DB-Engines ranking method [11]. They are being adopted in various markets in the industry and their prominence in those markets means that they would be encountered fairly often by the general digital forensic investigator.

To study the security features, the official documentation of the latest version of the selected NoSQL DBMS as found published on the manufacturer's website was used [12–15]. At the time of the examination the latest versions available were as follows: MongoDB 3.4, Cassandra 3.10, Redis 3.2 and Neo4j 3.1.3.

Even though each one of the selected NoSQL databases support scaling and data distribution via multi-node configurations, these databases were only considered as single node installations. Thus a discussion on distributed log files and the added complexities falls outside the scope of this study.

The remainder of this paper is structured as follows: Section 2 first gives a general introduction to NoSQL databases and their characteristics. Then it reviews each of the chosen NoSQL databases. Section 3 provides an overview of the field of forensic attribution. Section 4 then surveys the selected NoSQL databases regarding authentication, authorisation and logging. Section 5 analyses the results of the survey. Section 6 discusses the implications on digital forensics and specifically forensic attribution. Section 7 concludes this paper and contemplates future research.

2. NOSQL DATABASES

This section first provides a general introduction to NoSQL databases. Then the NoSQL databases chosen for the survey are examined in more detail.

2.1 NoSQL databases and types

The NoSQL movement was the development of new types of databases that were not relational and did not use the structured query language (SQL) as the data access language. These new database types were being created to address new demands in data forms and size that seemingly could no longer be met by existing relational databases and SQL.

NoSQL databases have more flexible data structures that can be completely schemaless. This allows for easier storage of unstructured and heterogeneous data. They also provide easier horizontal scalability to cater for big sets of data and data that grows unpredictably.

The "NoSQL" moniker became popular when Eric Evans chose it as the name for an event that Johan Oskarsson organised to discuss open source distributed databases [16]. Eric felt that the whole point of the event was to seek out alternatives that one needed to solve a problem that relational databases were a bad fit for. The event was the beginning of a movement that grouped together all database projects that were not relational.

Some people have objected to the NoSQL term for these new databases [17, 18], because it sounded like a definition based on what these databases were not doing rather than what they were. In recent years it has been suggested that the NoSQL term be changed from meaning “No SQL” to “Not Only SQL”. This is to express that NoSQL no longer meant anti-SQL and anti-relational, but rather expressed the notion that other database types besides relational ones existed that could help address the new data types and storage demands of today’s information society.

Today, a number of distinct types of NoSQL databases have established themselves. To make this research inclusive, databases from all the main types should be included in the study. The availability and usability of features such as access control and logging could also be influenced by how the different databases function internally. It might not be simply a matter of implementation to provide these features, but feasibility and practicality might also play a role.

It is thus worthwhile to take a closer look at the details of each of those NoSQL database types. The main types of NoSQL databases are the following four types: Document databases or stores, Key-value pair databases or stores, Column family store databases or wide column stores and Graph databases or stores [19]. A short summary of each type now follows.

Document Stores:

Document databases, also known as document stores or document-oriented databases, use a document-oriented model to store data. They store a record and its associated data within a single data structure called a document. Each document contains a number of attributes and associated values. Documents can be retrieved based on attribute values using various application programming interfaces (APIs) or query languages provided by the DBMS [19].

Document stores are characterized by their schema-free organization of data. That means that records do not need to have a uniform structure, i.e. different records may have different attributes. The types of the values of individual attributes can be different for each record. Records can also have a nested structure, while attributes can have more than one value such as an array.

Document stores typically use standard formats such as JavaScript Object Notation (JSON) or Extensible Markup Language (XML) to store the records [19]. This then allows the records to be processed directly in applications. Individual documents are stored and retrieved by means of an key. Furthermore, document stores rely on indexes to facilitate access to documents based on their attributes [20].

Wide Column Stores:

Wide column stores, also called extensible record stores, store data in records with an ability to hold very large numbers of dynamic columns. A column is the basic unit of storage and consists of a name and a value [19]. Any number of columns can be combined into a super column, which gives a name to a sorted set of columns. Columns are stored in rows, and when a row contains columns only, it is known as a column family. When a row contains super columns, it is known as a super column family [20].

As in document stores, column family databases do not require a predefined fixed schema. Different rows can have different sets of columns and super columns. Since the column names as well as the record keys are not fixed, and a record can have millions of columns, wide column stores can be seen as two-dimensional key-value stores [19].

Key-Value Stores:

Key-value stores are probably the simplest form of databases. They can only store pairs of keys and values, as well as retrieve values when the key or identifier is known. These systems can hold structured or unstructured data.

A namespace is a collection of identifiers. Keys must be unique within a namespace. A namespace could correspond to an entire database, which means all keys in the database must be unique. Some key-value stores provide for different namespaces within a database. This is done by setting up data structures for separate collections of identifiers within a database [19].

These simple systems are normally not adequate for complex applications. On the other hand, it is exactly this simplicity, that makes such systems attractive in certain circumstances. For example resource-efficient key-value stores are often applied in embedded systems or as high performance in-process databases.

One of the earliest such embedded key-value databases is Berkeley DB which was first released in 1991. It was developed at the University of California, Berkeley to replace certain patented components in their Unix release BSD 4.3. In 1992 BSD 4.4 was released which included Berkeley DB 1.85 [21].

Graph Stores:

Graph stores also known as graph databases are DBMSs with Create, Read, Update, and Delete (CRUD) methods that expose a graph data model. A graph database represents data in structures called nodes and relationships. A node is an object that has an identifier and a set of attributes. A relationship is a link between two nodes that contain attributes about that relation [19].

Some graph databases use native graph storage, which is designed to store and manage graphs directly. Other graph databases serialize the graph data into relational or object-oriented databases, or use other types of NoSQL stores [20]. In addition to having a certain approach to storing and processing graph data, a graph database will also use a specific graph data model. There are several different graph data models commonly used which include property graphs, hypergraphs, and triples.

Graph databases dont depend so much on indexes because the graph itself provides a natural index. In a graph database using native graph storage, the relationships attached to a node provide a direct connection to other related nodes. Graph queries use this characteristic to traverse through the graph [20]. Such operations can be carried out very efficiently, typically traversing millions of nodes per second. In contrast, joining data through a global index can be many orders of magnitude slower.

2.2 Surveyed NoSQL databases

Currently, the top five ranked NoSQL databases according to DB-Engines DBMS ranking are in order: MongoDB (document store), Cassandra (wide column store), Redis (key-value store), HBase (wide column store) and Neo4j (graph store) [11]. The top five thus represent all four NoSQL database types introduced above. To eliminate any possible bias of the survey due to multiple databases of the same type, the second wide column store HBase was excluded. More details about each of the other four databases now follow.

MongoDB:

MongoDB is an open-source document database that provides high performance, high availability, a rich query language and automatic scaling. It is published under a combination of the GNU Affero General Public License (AGPL) and the Apache License. The name MongoDB is derived from ”humongous database”, which alludes to the huge size a MongoDB database can have.

The software company 10gen began developing MongoDB in 2007 as a component of a planned platform as a service product. In 2009, the company shifted to an open source development model, with the company offering commercial support and other services. In 2013, 10gen embraced the database it had created and changed its name to MongoDB Inc. [22].

A record in MongoDB is a document, which is a data structure composed of a number of field and value pairs. MongoDB documents are similar to JSON objects. The values of fields may include other documents, arrays, and arrays of documents. MongoDB lists the advantages of using documents as follows: Firstly documents (and by extension objects) correspond to native data types in many programming languages. Secondly, embedded documents and arrays reduce the need for expensive joins. And finally dynamic schemas support fluent polymorphism [23].

MongoDB provides high performance data persistence and access through the use of the following features: Firstly it supports embedded data models which reduce I/O activity on the database system. Secondly it’s indexes can include keys from embedded documents and arrays thereby supporting faster queries. MongoDB uses B-trees for both data and index persistence. MongoDB has a rich query language that supports create, read, update and write (CRUD) operations as well as data aggregation, text search and geo-spatial queries [23].

MongoDB uses a replication facility called a “replica set” to provide automatic failover and data redundancy. A replica set is a group of MongoDB servers that maintain the same data set. Furthermore, MongoDB provides horizontal scalability by using sharding, which distributes data across a cluster of machines. It also supports the creation of zones of data based on a shard key. In a balanced cluster, MongoDB will direct reads and writes covered by a zone only to those shards inside the zone [23].

Prominent users of MongoDB include Metlife, Expedia, Ebay, SAP, SAGE, KPMG and Forbes [24].

Cassandra:

Cassandra is a free and open-source distributed wide column store DBMS. It is an Apache project published under the Apache 2.0 license. Cassandra is designed to handle large amounts of data across many commodity servers thereby providing high availability with no single point of failure. Cassandra offers support for clusters either in a single datacenter or spanning across multiple datacenters, with asynchronous masterless replication.

Avinash Lakshman and Prashant Malik initially developed Cassandra at Facebook to power the Facebook inbox search feature. They named their database after the Trojan mythological prophet Cassandra, who was given the power of prophecy by Apollo in order to seduce her. When she refused him favours, he cursed her prophecies to be never believed. The name thus alludes to a cursed oracle.

Facebook released Cassandra as an open-source project on Google code in July 2008 [25]. In March 2009 it became an Apache Incubator project and graduated to a top-level Apache project in February 2010. Cassandra is written in Java and thus available on any platform that can provide a Java virtual machine (JVM).

Cassandra’s column family (also called table) resembles a table in a relational database. Column families contain rows and columns. Each row is uniquely identified by a row key. Each row has multiple columns, each of which has a name, value, and a timestamp. Unlike a table in a relational database, different rows in the same column family do not have to share the same set of columns, and a column can be added to one or multiple rows at any time without blocking updates or queries.

Cassandra Query Language (CQL) is the primary interface into the Cassandra DBMS [26]. Using CQL is similar to using SQL. CQL and SQL share the same abstract idea of a table constructed of columns and rows. The main difference from SQL is that Cassandra does not support joins or sub-queries. Instead, Cassandra emphasizes denormalization through CQL features like collections and clustering specified at the schema level.

Cassandra uses a combination of memory tables (Memtables) and sorted string tables (SSTables) for persistence. Memtables are in-memory structures where Cassandra buffers all of its writes. When the Memtables are full, they are flushed onto disk by sequentially writing to the SSTables in append mode. Once written, the SSTables become immutable. Using this approach make it possible for Cassandra to avoid having to read before writing. Reading data involves combining the immutable sequentially-written SSTables to retrieve the correct query result [26].

In Cassandra, data is automatically replicated to multiple homogeneous nodes for fault-tolerance. A replication strategy determines the nodes where the replicas are placed. Cassandra employs a peer-to-peer distributed system across the nodes whereby the data is distributed among all nodes in the cluster [26]. Failed nodes in a cluster can be replaced with no downtime.

Prominent users of Cassandra include CERN, Netflix, Reddit and eBay [27].

Redis:

Redis is an open source key-value store that is published under the Berkeley Software Distribution (BSD) license. The in-memory data structure store can be used as a database, a cache or a message broker. The name Redis stands for REmote DIctionary Server. Redis was developed by Salvatore Sanfilippo who released the first version in May 2009. He was hired by VMware in March 2010 to work full time on Redis [28]. In 2015, Salvatore Sanfilippo joined Redis Labs which now sponsors development.

Redis maps keys to different types of values. It not only supports simple data structures such as strings but also abstract data structures such as hashes, lists, sets and sorted sets. Geo-spatial data is now supported through the implementation of the geohash technique.

The type of a value determines what operations (called commands) are available for the value itself. Redis supports high-level, atomic, server-side operations like appending to a string, incrementing the value in a hash, pushing an element to a list, computing set intersection, union and difference and sorting of lists, sets and sorted sets [29].

Redis is written in ANSI C and works in most POSIX systems like Linux, various BSD operating systems and OS X without external dependencies. It works with an in-memory dataset which can be optionally be persisted by either by dumping the dataset to disk every once in a while, or by appending each command to a log file.

Redis has built-in replication using a master-slave configuration which can be performed either via the dump file or directly from process to process. Redis also supports memory eviction methods such as Least Recently Used (LRU) which allows it to be used as a fixed size cache. Additionally Redis supports the publish/subscribe messaging paradigm, which allows it to be used a messaging platform.

Redis has a built-in Lua interpreter which can be used to write complex functions that run in the Redis server itself. Lua is a lightweight multi-paradigm programming language. Add-on Redis products provide additional features such as high availability via Redis Sentinel and automatic partitioning with Redis Cluster.

Prominent users of Redis include Twitter, Pinterest and Flickr [30].

Neo4j:

Neo4j is an open-source graph database management system that is an ACID-compliant transactional database with native graph storage and processing [31]. It is published under dual licence of the GNU Affero General Public License (AGPL) and the GNU Public License (GPLv3).

Neo4j is developed by Neo Technology, Inc. which released the first version in February 2010. It is implemented in Java and accessible from software written in other languages using the Cypher Query Language (CQL) through a transactional HTTP endpoint, or through the binary “bolt” protocol (Not related to the Cassandra Query Language).

The main features and capabilities of CQL are as follows: Firstly it works by matching patterns of nodes and relationships in the graph, to extract information or modify the data. Secondly it has the concept of variables which denote named, bound elements and parameters. Thirdly it can create, update, and remove nodes, relationships, labels, and properties. Lastly it is used to manage indexes and constraints [32].

In Neo4j, everything is stored in the form of either an edge, a node, or an attribute. Each node and edge can have any number of attributes. Both the nodes and edges can be labelled. Labels can be used to narrow searches. Neo4j uses on-disk linked lists for persistence.

Joining data together in Neo4j is performed as navigation from one node to another which provides linear performance degradation compared to relational databases, where the performance degradation is exponential for an increasing number of relationships [31].

Prominent users of Neo4j include Walmart, Monsanto and Ebay [33].

3. ATTRIBUTION

Having completed the introduction of NoSQL databases and the presentation of the selected NoSQL databases, focus is now placed on attribution. First an overview of the field is given to provide context for the survey. Then forensic attribution processes and techniques are discussed and their implications on database forensics explored.

3.1 General attribution

The Oxford English dictionary defines the term *attribution* as follows: "The action of regarding something as being caused by a person or thing". Attribution is performed in a number of diverse application areas. These include clinical psychology attribution, nuclear attribution, authorship attribution and cyber attack attribution.

In clinical psychology, attribution refers to the process by which individuals explain the causes of behaviour and events [34]. In nuclear forensics, nuclear attribution is the process of tracing the source of nuclear material from a radiological incident whether accidental (e.g. nuclear waste spill) or intentional (e.g. nuclear explosion) [35]. Authorship attribution refers to the process of inferring characteristics of the author from the characteristics of documents written by that author [36].

Cyber attack attribution has been defined and researched by various authors. Wheeler and Larson in their paper for the U.S. Department of Defense (DoD) defined it as "determining the identity or location of an attacker or an attackers intermediary" [6]. They define the resulting identity as a persons name, an account, an alias, or similar information associated with a person. A location is interpreted as a physical (geographic) location, or a virtual location such as an IP address or MAC address.

Boebert breaks the attribution question down into two attribution problems: technical attribution and human attribution [37]. According to the author, technical attribution consists of analysing malicious functionality and packets, and using the results of the analysis to locate the node which initiated, or is controlling, the attack. Human attribution on the other hand, consists of taking the results of technical attribution and combining it with other information to identify the person or organization responsible for the attack.

Clark and Landau in their paper "Untangling attribution" contend that there are many types of attribution and different types are useful in different contexts [7]. For example, attribution on the Internet could mean the identification of the owner of the machine (e.g. the company or organisation), the physical location of the machine (e.g. city or country) or the individual who is actually responsible for the actions.

Clark and Landau also define three classes of attacks: bot-net based attacks (e.g. DDoS), identity theft and data ex-filtration and espionage. Based on these classes different attribution types and techniques are more suitable or applicable than others. The timing of when attribution is performed also plays an important role.

For example during a DDoS attack, mitigation might be the most immediate concern. Attribution is then needed to identify the machines launching the attack so that they can be blocked. However, after the DDos attack is over, focus may shift towards retribution as deterrence. Attribution is then needed to identify the actors responsible so that they can be prosecuted [7].

3.2 Forensic attribution

When attribution is done as part of an investigation using scientific methods, the term forensic attribution is used. Forensic attribution is performed during the digital evidence interpretation step in a forensic examination [38]. This step is part of the investigative processes as defined in the ISO/IEC 27043 standard that describes incident investigation principles and processes [38].

Forensic attribution processes:

As already touched on in the introduction, a number of researchers have proposed different forensic attribution processes. These diverse processes define different levels, categories or steps of attribution that can be performed.

Cohen for example sees end-to-end attribution made up of four different levels of attribution [39]. The first two levels are performed in the digital world. Level 1 attempts to identify the closest computer involved while level 2 tries to pinpoint the source computer that initiated the actions. However, the next two levels of attribution are performed in the physical world. Level 3 attempts to identify the individual that caused source computer to act as it did, while at level 4 the organisation behind the individual is being sought.

Shamsi et al propose three steps of identification for attribution [40]. The first step deals with the identification of the cyber weapon used to launch the actions. They use the definition of *cyber weapon* given by Rid and McBurney. They, in turn, define cyber weapon as "computer code that is used, or designed to be used, with the aim of threatening or causing physical, functional, or mental harm to structures, systems, or living beings" [41].

Thus step 1 is executed in the digital realm. Step 2 deals with the identification of the country or city of the actor, while step 3 addresses the identification of the actor whether it is a person or an organisation. The last two steps thus take place in the physical world. Similarly, Clark and Landau define three categories into which attribution can fall [7]. These categories are the machine, the person, and the aggregate identity, such as a state actor. Again, the first category of attribution is executed in the digital realm, while the other two happen in the physical world.

Table 1 summarises these different attribution processes.

Table 1: Attribution Processes Summary

Author(s)	**Digital Realm**	**Physical Realm**
Cohen [39]	• Identify closest computer • Identify initiating computer	• Identify individual behind initiating computer • Identify organisation behind individual
Shamsi et al. [40]	• Identify cyber weapon	• Identify country/city • Identify person/organisation
Clark and Landau [7]	• Identify computer	• Identify individual • Identify organisation

Even though the authors use different terminology to describe the parts of their processes (steps/levels/categories), the parts and their goals seem to be very similar. For the sake of clarity this paper is going to overlook the deeper meaning of the authors' chosen terminology and from here on simply refer to the different parts as steps.

In the digital realm the steps from all authors have the same goal: to identify the computing device(s) responsible for the actions that are being examined. Various digital forensic attribution techniques can be used to achieve this goal. The more difficult steps are performed in the physical realm. They attempt to identify the individuals or actors responsible for the actions that are being examined. As already indicated in the introduction, this kind of attribution may not be always be needed. The paper will therefore concentrate on attribution steps that are performed in the digital realm.

Forensic attribution techniques:

There are a large number of digital attribution techniques with each technique having certain strengths and weaknesses. A taxonomy of these attribution techniques is provided by Wheeler and Larson [6]. According to them no single technique can replace all others and a combination of techniques can help compensate for their respective weaknesses.

One of the techniques that the forensic examiner can employ is to make inferences based on authentication and authorisation information [8]. This information enables the forensic examiner to create a basis for attribution. The authentication information provides the actors of a digital system and the authorisation information the actions that these actors can perform [42].

Another technique is to order and connect the different traces found in digital systems to build a chain of events [8]. The sequences of these events describe how the system arrived at the current state. By determining the actions that lead to the events and the actors that performed the actions, the person or program responsible can possibly be identified [42].

Performing forensic attribution in relational databases was investigated by Olivier [9]. He showed that database forensics can use the same techniques as general digital forensics to perform attribution. The traces in a relational database are available in various log files and also stored inside system tables. Furthermore, the authentication of database users and the authorisation of their operations has been standardised and is built into many relational databases [43].

4. NOSQL SECURITY SURVEY

A survey of the documentation for the chosen four NOSQL databases was conducted. Comparable information was obtained on the availability of the following features: authentication, authorisation and logging. The official documentation was used as far as possible, but in some cases document holes were supplemented with additional sources. In some cases the information was readily available, while in other cases it was necessary to delve into the database software files. This section presents the results of this survey.

It should be noted, that only the features available in the official free and community editions of the selected NoSQL databases were analysed. Some of the NoSQL databases also have paid-for enterprise editions available that provide additional features (See for example Neo4j Editions [44]). These additional features include enhancements and additions to authentication, authorisation and logging.

The results for the three features are presented in the same order for all the selected databases. This is to enable direct comparison between the different NoSQL databases and allow commonalities and/or differences to be established for later discussion.

4.1 MongoDB

Unless otherwise indicated, this section uses the official MongoDB documentation [12] to paraphrase the relevant information to indicate the availability of the surveyed features.

MongoDB supports a number of authentication mechanisms that clients can use to verify their identity. These include SCRAM-SHA-1 and x.509 client certificates. SCRAM-SHA-1 is an authentication mechanism from the Salted Challenge Response Authentication (SCRAM) family that uses the SHA-1 hash function. It is a mechanism for authenticating users with passwords and defined in the IETF standard RFC 5802 [45].

MongoDB employs role-based access control to govern access to a MongoDB system. A user is conferred one or more roles that determine the user's access to database resources and operations. Outside of role assignments, the user has no access to the system. MongoDB does not enable access control by default, but it can be enabled via the configuration file or a start-up parameter.

Since MongoDB does not have a built-in default user, an appropriate administration user must be created before authentication is enabled. Alternatively, MongoDB provides an exception where it allows an unauthenticated connection on the local loopback interface to the admin database. Once an appropriate administration user has been created via this connection, no further actions can be performed and this connection needs to be terminated to establish a new authenticated one.

MongoDB provides a number of built-in roles that can be used to control access to a MongoDB system. Each of these roles have specific privileges assigned to them. The roles are divided into different categories such as database user, database administrator, superuser etc. However, if the specific privileges of the built-in roles are not sufficient, one can create new roles with the desired privileges in a particular database.

A role grants privileges to perform sets of actions on defined resources. A given role applies to the database on which it is defined. Access can be granted on a whole cluster, a specific database in the cluster or to individual collections inside a database. Privileged actions that are available to roles are grouped together as follows: query and write actions, database management actions, deployment actions, replication actions, sharding actions and server administration actions.

MongoDB database instances can report on all their server activity and operations. Per default, these messages are written to standard output, but they can be directed to a log file via the configuration file or a start-up parameter. MongoDB's default log verbosity level includes just informational messages. This can be changed to include debug messages by setting the verbosity to a higher level.

Additionally, MongoDB allows logging verbosity to be controlled at a finer grain by providing verbosity settings on a component level. These components include items such as access control, commands, queries etc. Unless explicitly set, each component has the verbosity level of its parent. MongoDB verbosity levels range from the informational default of 0 to the most verbose debug level of 5.

When logging to a file is enabled, MongoDBs standard log rotation approach archives the current log file and starts a new one. This normally occurs when the MongoDB instance is restarted. While the MongoDB instance is running, this can also be triggered by either issuing the "logRotate" command inside the database or by sending the SIGUSR1 signal from the OS to the MongoDB process id.

4.2 Cassandra

Unless otherwise indicated, this section uses the official Cassandra documentation [13] to paraphrase the relevant information to indicate the availability of the surveyed features.

Cassandra provides pluggable authentication that can be configured via settings in the configuration file. The default Cassandra configuration uses the AllowAllAuthenticator which performs no authentication checks and therefore requires no credentials. It is used to disable authentication completely. Cassandra also includes the PasswordAuthenticator, which stores encrypted credentials in a system table. This is used to enable simple username/password authentication.

Cassandra uses a role-based access control framework, but provides no fixed or pre-defined roles. Cassandra roles do have a login property and a superuser property. The default Cassandra user has these properties set so it can be used to setup further users and roles once authentication has been enabled. Users and roles are the exact same concept, but to preserve backward compatibility they are both still used. User statements are simply synonyms of the corresponding role statements.

Cassandra also provides pluggable authorisation that can be configured in the same configuration file as authentication. By default, Cassandra is configured with the AllowAllAuthorizer which performs no checking and so effectively grants all permissions to all roles. This is used if the AllowAllAuthenticator is the configured authenticator. Cassandra also includes the CassandraAuthorizer, which implements full permissions management functionality and stores its data in Cassandra system tables.

Permissions on various resources are granted to the roles. The permissions available depend on the type of resource. Cassandra provides the following resource types: data resources such as keyspaces and tables, function resources, database roles and Java managed beans (MBeans). The resource types are structured as hierarchies and permissions can be granted at any level of these hierarchies and they flow downwards.

Cassandra provides all of the following permissions: CREATE, ALTER, DROP, SELECT, MODIFY, AUTHORIZE, DESCRIBE, EXECUTE. A matrix determines which permissions can be applied to which resources. One can grant individual permissions to resources or use the GRANT ALL syntax to grant all applicable permissions to a resource.

Cassandra uses the Java logging framework Logback to create various log files about everything that occurs in the system. Java logging classifies messages in levels [46], where a lower level of messages will include all the higher level ones as well. For example, the INFO level will include message from the higher ERROR level, while the lower DEBUG level will include the higher level INFO and ERROR messages. By default the following two log files are created: the system log file which contains all the INFO level messages produced in the system and the debug log file which contains all the DEBUG level messages. The debug log file additionally contains caller information as well.

Another log file available in Cassandra is the commit log. To enhance performance, Cassandra keeps column updates in memory and periodically flushes those changes to disk. To prevent data losses when the system goes down before flushing, these updates are also written to the commit log. When Cassandra starts up again, it reads the commit log back from the last known good point in time and re-applies the changes in the commit log so it can get into the same state as when it went down. Although the commit log only contains the most recent changes that have not been flushed to disk yet, there is a configuration option that will archive the contents of the commit log.

4.3 Redis

Unless otherwise indicated, this section uses the official Redis documentation [14] to paraphrase the relevant information to indicate the availability of the surveyed features.

Redis is an in-memory database that is designed to be run inside trusted environments and accessed by trusted clients. Untrusted access is expected to be mediated by an intermediary layer that implements access control, validates user input and determines what operations may be performed against the Redis database instance.

Although Redis does not implement access control, it does provide a tiny layer of authentication that can be enabled by editing the configuration file and setting a password. When the authentication layer is enabled, Redis will refuse any queries by unauthenticated clients. A client then must authenticate itself by sending the AUTH command followed by the password. The AUTH command, like every other Redis command, is sent unencrypted.

The purpose of the authentication layer is to serve as a protection layer against the accidental exposure of a Redis database instance to external untrusted environments. To force the setting of a password, a Redis instance in default configuration will only start in protected mode. In protected mode the Redis instance only accepts clients on the local loopback interface while throwing errors on all other available interfaces. Once the password has been set, the other configured interfaces will accept client connections.

Redis has no form of authorisation. Once a client is authenticated, any command can be called including the FLUSHALL command which will delete the whole data set. As mitigation, Redis allows commands to be renamed into unguessable names, so that normal clients can be limited to a specified set of commands. Systems that provide and manage Redis instances would then still be able to execute the renamed commands.

Redis does have some form of logging, although it is advised that it be only used for debugging purposes. The Redis “Slow Log” is a system to log queries that exceeded a specified execution time. However, by setting the execution time threshold to zero all commands including queries will be logged. Keeping with its in-memory nature, Redis keeps the slow log in memory. To prevent over usage of memory for logging purposes, by default only the last 1024 slow log entries will be kept. To retrieve the slow log entries, the SLOWLOG GET command needs to be used [47].

Another form of command logging happens when append-only file (AOF) persistence is enabled. When enabled, every time the Redis database instance receives a command that changes the dataset (e.g. SET) it will append it to the AOF. The purpose of the AOF is to rebuilt the state after the database was shutdown without a snapshot of the current state. To prevent the file from growing uncontrollably, Redis can from time to time rewrite the actual stored commands with the shortest sequence of commands needed to rebuild the current dataset in memory.

4.4 Neo4j

Unless otherwise indicated, this section uses the official Neo4j documentation [15] to paraphrase the relevant information to indicate the availability of the surveyed features.

Neo4j provides a basic authentication layer that is enabled by default. It has a built-in default user for whom the password can be set during installation. Should the password not be changed during installation, Neo4j will prompt for a password change on first connection. Additional users can be added to the database by the default user once authenticated.

Neo4j has no form of authorisation. This implies that once a client is authenticated, any operation can be performed on the database. Additionally Neo4j only accepts client connections on the local loopback interface in default configuration. External interfaces for remote connectivity need to be configured explicitly.

Neo4j does provides some logging. Traffic on the HTTP/HTTPS connector is logged to a file called http.log. However, this traffic logging is not enabled by default.

The enterprise version of Neo4j however, does provide a role-based access control framework that furnishes built-in roles as well as the ability to add custom roles. It also provides additional logging capabilities to audit security events and queries executed. These capabilities need to be configured first, since they are not enabled by default.

5. DISCUSSION

In this section the results from the security feature survey in the previous section are discussed using a few summary tables. Section V then discusses the implications of these results on forensic attribution.

Table 2: NoSQL Security Features

Database	Authentication	Authorisation	Logging
MongoDB	Yes	Yes	Yes
Cassandra	Yes	Yes	Yes
Redis	Yes	No	Yes
Neo4j	Yes	No	Yes

Table 2 summarises the results from the survey of access control and logging of the selected NoSQL databases. The first result this summary shows, is that all of the surveyed NoSQL databases do support authentication. However the second result is that two of the NoSQL databases do not provide authorisation. This divides the surveyed NoSQL databases into two groups: The first group of databases control both who can access them and what operations the authenticated users can perform. The second group of databases only control who can access them, but not what the authenticated users can do.

Specifically, Redis only provides a thin authentication layer that does not have different users, but rather restricts client access via a simple password. Since it has no differentiated user access, Redis also does not provide any authorisation. Neo4j also does not provide any role based authorisation, even though differentiated user authentication is supported. This implies that in both those databases all clients have the same full control over all database operations once they have been authenticated.

The third result that the summary in Table 2 shows, is that all of the surveyed NoSQL databases do provide some form of logging. It should be noted that this survey looked at all the log files that were being generated by the chosen NoSQL databases, not only audit logs. Some of the log files that were surveyed, are only created when special features in the database are enabled, while other log files are created by the storage mechanism that the particular database uses. This means that rather than being general log files, these files are specialised log files that contain only specific type of messages.

Some NoSQL databases like MongoDB and Redis include the ability to log queries that took particularly long to complete. In the case of Redis, the threshold used to determine when to log slow queries can be changed to zero, which will make Redis log every query executed. Thus the normal Redis slow log can be turned into a query audit log.

Table 3: Features Enabled by Default

Database	Access Control	Logging
MongoDB	No	No
Cassandra	No	Yes
Redis	No	Yes
Neo4j	Yes	No

Table 3 summarises the default state of the security features that are available for the surveyed NoSQL databases. This summary shows that only one of the surveyed NoSQL databases comes with access control enabled by default. The implication of this result is that the installations of all those other NoSQL databases will be accessible to anyone without explicit configuration changes.

A small security consolation is that some of these NoSQL databases will per default only accept client connections on the local loopback interface. This means that no remote access is possible and only clients on the same machine as the database can connect.

In the case of MongoDB, this default “local loopback only” state is created with the value of the network configuration option, which can easily be changed to the network interface of the machine. This single change will then open up the MongoDB database to remote clients without access control. In the case of Redis, this “local loopback only” state is enforced by a separate configuration option. However, by changing it and the network configuration option, the Redis database can be opened up to remote clients without authentication.

Table 3 also shows that logging is not enabled by default on some databases. So even though for example MongoDB has great logging capabilities that can audit database access and operations, none of that is available by default. Only after careful configuration of the various settings will the same information be available as found in many relational databases.

In the case of Neo4j the fact that logging is not enabled by default is not a great loss. This is because only HTTP traffic logging is available in the community edition of Neo4j. The logging capabilities for security events and queries is only available in the paid-for enterprise edition.

6. FORENSIC IMPLICATIONS

This sections considers the implications of the results from the previous section on forensic examinations and particularly forensic attribution. The availability of access control and logging/auditing in the surveyed NoSQL databases is considered separately.

6.1 Access Control

The traces or artefacts from access control in a database can help the forensic examiner as follows: firstly, the authentication traces can provide a list of users that connected around the time the operations being examined were performed. Secondly, the authorisation matrix can narrow down this list based on who was authorised to perform the operations in question. The first group of NoSQL databases that was identified in the previous section can aid forensic attribution in this way.

The second group of NoSQL databases that the survey identified, only have authentication available, but no authorisation. The implication is that in those databases all clients have the same full control over all database actions once they have been authenticated. This means it will not be possible to narrow down the list of users based on the operations they are authorised to perform, since theoretically all of the users had the authority to perform the operations being examined.

One of the databases in the second group also has no concept of a database user and just provides simple password based access. From a security standpoint this simple authentication layer provides an improvement over having no authentication, but from an forensic attribution standpoint, it adds almost no additional value. The forensic examiner can only deduce that the responsible person was in possession of the correct password, provided the security model of the database is sound and no unauthenticated access is possible.

The survey also showed that none of the selected NoSQL databases have authentication enabled by default. The forensic examiner is thus dependent on the database administrator to have enabled authentication for possible access control traces. But without these access control traces being persisted into a log file or some other data file, the mere presence of access control in the database is not sufficient to aid forensic attribution.

6.2 Logging

The different log files that were encountered during the survey of the selected NoSQL databases can be divided into three groups: audit logs, system logs and storage logs.

Audit Logs: Audit logs maintain a record of various activities in the database for later review or possible debugging in case of errors. Two pieces of information normally found in the audit logs that the forensic examiner can use for forensic attribution are the access records and the operation records.

The access records show who connected to the database and when, while the operation records show what operations or queries were performed when and by whom. However, without authentication enabled or available there will be no access records and the operations will not have any user associated with them.

None of the surveyed NoSQL databases provided specific audit logs in the free and community versions. This means that to perform forensic attribution in those database versions, the forensic examiner will have to look at the other groups of log files.

System Logs: System or operational logs are created by the databases during the normal running of the system and can contain many different informational and error messages. How valuable these system log files are to the forensic examiner depends on their content.

The survey showed that some of the NoSQL databases include the ability to configure what messages and operations are written to the system log. This includes to a certain extent access and operation records. Thus the normal system log file can be turned into an audit log as well.

Thus if the database administrator has enabled logging and configured the system log appropriately, the forensic examiner can use them to aid forensic attribution. Unfortunately this makes the availability of system logs not something the forensic investigator can depend on when performing forensic attribution.

Storage Logs: Storage logs that are available on some of the surveyed NoSQL databases are created by their persistence mechanisms. These storage logs contain the information of all the operations that modified the data. Storage logs may or may not be archived after the information they contain has been transferred to the data files. This depends on the configuration of the database storage and available space.

The storage logs perform two functions in the NoSQL databases that use them. Firstly they speed up the write speed of the database by first writing the change operations to a small linear file before applying them to the bigger complex data files. Secondly they maintain a record of changes in case the database goes down before the change operations have been fully applied to the data files.

After a failure, the database can re-apply the operations from the storage log file to the data files to get them to the current state. In the same way the forensic examiner can use the storage logs to roll back the state of the database to an earlier point in time. This process is called reconstruction and can help identify changes that were made and information that was removed [48].

In order to save space, Cassandra uses a technique called compaction. Compaction is the process where the DBMS goes through the storage log and replaces individual operations that made changes to the same data with a single operation that has the same outcome [49]. The problem for the forensic examiner is that he no longer can see the individual operations that were performed possibly by different users.

It ultimately depends on the scenario the forensic investigator is dealing with, as to whether these storage logs will aid forensic attribution or not. In the case where data was accessed or taken, there will be no changes to the storage log file. However, in the case where data was modified or removed there will be entries in the storage log file that could contain clues as to who was responsible.

7. CONCLUSION

The increase in the usage of NoSQL databases for data storage, analysis and retrieval in recent years has meant that more and more confidential and sensitive data is being stored in them. This has made NoSQL databases a new target for hackers and other unauthorised entities. This in turn would prompt more forensic examinations to determine which data was compromised and who was responsible. Since these NoSQL databases seem to lack adequate security features it was necessary to determine how this would impact forensic attribution.

Forensic attribution in digital systems is difficult to perform because the actions that need to be attributed occurred in the digital world. However, the initiators of these actions are located in the physical world. To be able to attribute actions to a program or process, various sources of traces are needed. These traces are then correlated to form a chain of events that can help pinpoint the initiating program or process.

A survey of four top ranked NoSQL databases was performed to determine what security measures are available, that could aid the forensic investigator to perform forensic attribution. The survey specifically looked at the areas of authentication, authorisation and logging.

Even though the surveyed NoSQL databases MongoDB and Cassandra have the same security features available as in widely used relational databases, they are not enabled and configured appropriately in default configuration mode. When performing a forensic examination, the forensic examiner is thus completely reliant on the configuration that the database administrator performed on the particular database.

Furthermore, the surveyed NoSQL databases Redis and Neo4j did not provide security features that left relevant traces. In those databases the forensic examiner is thus forced to only use traces from outside the database to help perform attribution of the actions that occurred inside the database. The lack of these traces can negatively impact the accuracy of the attribution result.

The biggest concern however, was that the surveyed NoSQL database Neo4j only provided relevant security measures in the paid for enterprise edition. This makes data security seem like an optional extra, even though many countries have laws regarding information privacy and data security. This would make Neo4j completely unsuitable for many applications in those countries. That is unless the organisations wanting to use Neo4j are prepared to pay for it.

The next step would be to determine the prevalence of free and open source editions compared to paid-for editions among deployed NoSQL databases. This would provide an idea of how many NoSQL databases would not have the necessary information at the database level to aid forensic attribution. In those cases the forensic examiner would need to rely on external systems and the network to perform forensic attribution. These same NoSQL databases would also present easier targets to hackers and other malicious actors because of their lack of available security features.

In order to make the available trace information that could aid forensic examinations more concrete, a study focussed on the detail content of the various identified log files would be required. Such a study would also have to consider multiple versions of the same NoSQL DBMS software. This is because the studied NoSQL databases are still rapidly developing and the information being logged can change between versions.

An aspect that has not been addressed in this paper is the protection provided to the log files that were identified. What methods are used to prevent tampering and what mechanisms are built into the DBMSs to detect compromised log files? Some future work would be required to answer these questions.

REFERENCES

[1] DB-Engines. (2017, Apr.) DB-Engines Ranking - Trend Popularity. [Online]. Available: https://db-engines.com/en/ranking_trend

[2] D. Feinberg, M. Adrian, N. Heudecker, A.M. Ronthal, and T. Palanca. (2015, Oct.) Magic Quadrant for Operational Database Management Systems. Gartner Inc. [Online]. Available: https://www.gartner.com/doc/3147919/magic-quadrant-operational-database-management

[3] C. Strauch. (2011) NoSQL Databases. Stuttgart Media University. [Online]. Available: http://www.christof-strauch.de/nosqldbs.pdf

[4] L. Okman, N. Gal-Oz, Y. Gonen, E. Gudes, and J. Abramov, "Security Issues in NoSQL Databases," in *Proceedings of the 10th International Conference on Trust, Security and Privacy in Computing and Communications*, Changsha, China, Nov. 16–18, 2011, pp. 541–547.

[5] S. Srinivas and A. Nair, "Security maturity in NoSQL databases," in *Proceedings of the 2015 International Conference on Advances in Computing, Communications and Informatics*, Kochi, India, Aug. 10–13, 2015, pp. 739–744.

[6] D.A. Wheeler and G.N. Larsen, "Techniques for Cyber Attack Attribution," Institute for Defense Analyses, VA, Oct. 2003.

[7] D.D. Clark and S. Landau, “Untangling Attribution,” in *Proceedings of a Workshop on Deterring CyberAttacks: Informing Strategies and Developing Options for U.S. Policy*, Washington, DC, Jun. 10–11, 2010, pp. 25–40.

[8] F. Cohen, *Digital Forensic Evidence Examination*, 4th ed. Livermore, CA: Fred Cohen & Associates, 2009.

[9] M.S. Olivier, “On metadata context in Database Forensics,” *Digital Investigation*, vol. 5, pp. 115–123, Mar. 2009.

[10] S. Edlich. (2017, Apr.) NOSQL Databases. [Online]. Available: http://nosql-database.org/

[11] DB-Engines. (2017, Apr.) DB-Engines Ranking. [Online]. Available: https://db-engines.com/en/ranking

[12] MongoDB Inc. (2017, Apr.) The MongoDB 3.4 Manual. [Online]. Available: https://docs.mongodb.com/manual/

[13] Apache Cassandra. (2017, Apr.) Apache Cassandra Documentation v3.2. [Online]. Available: https://cassandra.apache.org/doc/latest/

[14] Redis Labs. (2017, Apr.) Documentation. [Online]. Available: https://redis.io/documentation

[15] Neo4j. (2017, Apr.) The Neo4j Operations Manual v3.1. [Online]. Available: https://neo4j.com/docs/operations-manual/current/

[16] E. Evans. (2009, Oct.) NoSQL: What’s in a name? [Online]. Available: http://blog.sym-link.com/2009/10/30/nosql_whats_in_a_name.html

[17] J. Ellis. (2009, Nov.) The NoSQL Ecosystem. [Online]. Available: https://blog.rackspace.com/nosql-ecosystem

[18] E. Eifrem. (2009, Oct.) Emil Eifrem on Twitter. [Online]. Available: https://twitter.com/emileifrem/statuses/5200345765

[19] D. Sullivan, *NoSQL for Mere Mortals*, 1st ed. Hoboken, NJ: Addison-Wesley Professional, 2014.

[20] I. Robinson, J. Webber, and E. Eifrem, *Graph Databases*, 2nd ed. Sebastopol, CA: O’Reilly Media, Inc., 2015.

[21] M.A. Olson, K. Bostic, and M. Seltzer, “Berkeley DB,” in *Proceedings of the 1999 USENIX Annual Technical Conference*, Monterey, CA, Jun. 6–11, 1999.

[22] D. Harris. (2013, Aug.) 10gen embraces what it created, becomes MongoDB Inc. [Online]. Available: https://gigaom.com/2013/08/27/10gen-embraces-what-it-created-becomes-mongodb-inc/

[23] MongoDB Inc. (2017, Apr.) Introduction to MongoDB. [Online]. Available: https://docs.mongodb.com/manual/introduction/

[24] MongoDB Inc. (2017, Apr.) Our Customers — MongoDB. [Online]. Available: https://www.mongodb.com/who-uses-mongodb

[25] J. Hamilton. (2008, Jul.) Facebook Releases Cassandra as Open Source. [Online]. Available: http://perspectives.mvdirona.com/2008/07/facebook-releases-cassandra-as-open-source/

[26] DataStax. (2017, Apr.) Apache Cassandra 3.0. [Online]. Available: http://docs.datastax.com/en/cassandra/3.0/

[27] Apache Cassandra. (2017, Apr.) Apache Cassandra. [Online]. Available: https://cassandra.apache.org/

[28] S. Sanfilippo. (2010, Mar.) VMware: the new Redis home. [Online]. Available: http://oldblog.antirez.com/post/vmware-the-new-redis-home.html

[29] Redis Labs. (2017, Apr.) Introduction to Redis. [Online]. Available: https://redis.io/topics/introduction

[30] Redis Labs. (2017, Apr.) Who’s using Redis? [Online]. Available: https://redis.io/topics/whos-using-redis

[31] Neo4j. (2017, Apr.) Chapter 1. Introduction. [Online]. Available: https://neo4j.com/docs/operations-manual/current/introduction/

[32] Neo4j. (2017, Apr.) Neo4j Cypher Refcard 3.1. [Online]. Available: https://neo4j.com/docs/cypher-refcard/current/

[33] Neo4j. (2017, Apr.) Neo4j Customers. [Online]. Available: https://neo4j.com/customers/

[34] H.H. Kelley, “The processes of causal attribution,” *American Psychologist*, vol. 28, no. 2, pp. 107–128, Feb. 1973.

[35] M. Wallenius, K. Mayer, and I. Ray, “Nuclear forensic investigations: Two case studies,” *Forensic Science International*, vol. 156, no. 1, pp. 55–62, Jan. 2006.

[36] P. Juola, “Authorship Attribution,” *Foundations and Trends in Information Retrieval*, vol. 1, no. 3, pp. 233–334, Mar. 2008.

[37] W.E. Boebert, “A Survey of Challenges in Attribution,” in *Proceedings of a Workshop on Deterring CyberAttacks: Informing Strategies and Developing Options for U.S. Policy*, Washington, DC, Jun. 10–11, 2010, pp. 41–52.

[38] ISO, "Information technology – Security techniques – Incident investigation principles and processes," International Organization for Standardization, Geneva, Switzerland, ISO/IEC 27043:2015, Mar. 2015.

[39] F.B. Cohen, "Attribution of messages to sources in digital forensics cases," in *Proceedings of the 43rd Hawaii International Conference on System Sciences*, Honolulu, HI, Jan. 5–8, 2010, pp. 4459–4468.

[40] J.A. Shamsi, S. Zeadally, F. Sheikh, and A. Flowers, "Attribution in cyberspace: techniques and legal implications," *Security and Communication Networks*, vol. 9, no. 15, pp. 2886–2900, Oct. 2016.

[41] T. Rid and P. McBurney, "Cyber-Weapons," *The RUSI Journal*, vol. 157, no. 1, pp. 6–13, Feb. 2012.

[42] W.K. Hauger and M.S. Olivier, "Determining trigger involvement during Forensic Attribution in Databases," in *Advances in Digital Forensics XI*, G. Peterson and S. Shenoi, Eds. Heidelberg, Germany: Springer, 2015.

[43] ISO, "Information technology – Database languages – SQL – Part 2: Foundation (SQL/Foundation)," International Organization for Standardization, Geneva, Switzerland, ISO/IEC 9075-2:2012, Oct. 2012.

[44] Neo4j. (2017, Apr.) Compare Neo4j Editions. [Online]. Available: https://neo4j.com/editions/

[45] C. Newman, A. Menon-Sen, A. Melnikov, and N. Williams, "Salted Challenge Response Authentication Mechanism (SCRAM) SASL and GSS-API Mechanisms," Internet Requests for Comments, RFC Editor, RFC 5802, July 2010. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5802.txt

[46] Oracle. (2017, Apr.) Java Logging Package. [Online]. Available: https://docs.oracle.com/javase/6/docs/api/java/util/logging/package-summary.html

[47] K. Seguin, *The Little Redis Book*. Self-Published, 2012.

[48] O.M. Adedayo and M.S. Olivier, "Ideal log setting for database forensics reconstruction," *Digital Investigation*, vol. 12, pp. 27–40, Mar. 2015.

[49] J. Ellis. (2011, Oct.) Leveled Compaction in Apache Cassandra. [Online]. Available: http://www.datastax.com/dev/blog/leveled-compaction-in-apache-cassandra

FINITE STATE MACHINE FOR THE SOCIAL ENGINEERING ATTACK DETECTION MODEL: SEADM

Francois Mouton[*] , Alastair Nottingham[†] , Louise Leenen[‡] and H.S Venter[§]

[*] *Defence Peace Safety & Security, Council for Scientific and Industrial Research, Pretoria, South Africa E-mail: moutonf@gmail.com*

[†] *E-mail: anottingham@gmail.com*

[‡] *E-mail: lleenen@csir.co.za*

[§] *Department of Computer Science, University of Pretoria, Pretoria, South Africa E-mail: hventer@cs.up.ac.za*

Abstract: Information security is a fast-growing discipline, and relies on continued improvement of security measures to protect sensitive information. Human operators are one of the weakest links in the security chain as they are highly susceptible to manipulation. A social engineering attack targets this weakness by using various manipulation techniques to elicit individuals to perform sensitive requests. The field of social engineering is still in its infancy with respect to formal definitions, attack frameworks, and examples of attacks and detection models. In order to formally address social engineering in a broad context, this paper proposes the underlying abstract finite state machine of the Social Engineering Attack Detection Model (SEADM). The model has been shown to successfully thwart social engineering attacks utilising either bidirectional communication, unidirectional communication or indirect communication. Proposing and exploring the underlying finite state machine of the model allows one to have a clearer overview of the mental processing performed within the model. While the current model provides a general procedural template for implementing detection mechanisms for social engineering attacks, the finite state machine provides a more abstract and extensible model that highlights the inter-connections between task categories associated with different scenarios. The finite state machine is intended to help facilitate the incorporation of organisation specific extensions by grouping similar activities into distinct categories, subdivided into one or more states. The finite state machine is then verified by applying it to representative social engineering attack scenarios from all three streams of possible communication. This verifies that all the capabilities of the SEADM are kept in tact, whilst being improved, by the proposed finite state machine.

Key words: Bidirectional Communication, Finite State Machine, Indirect Communication, Social Engineering, Social Engineering Attack Examples, Social Engineering Attack Detection Model, Social Engineering Attack Framework, Unidirectional Communication.

1. INTRODUCTION

Protection of sensitive information is of vital importance to organisations and governments, and the development of measures to counter illegal access to such information is an area that continues to receive increasing attention. Organisations and governments have a vested interest in securing sensitive information and the trust of clients or citizens. Technology on its own is not a sufficient safeguard against information theft; staff members — often the weak link in an information security system — can be influenced or manipulated to divulge sensitive information that allows unauthorised individuals to gain access to protected systems.

The ‘art’ of influencing people to divulge sensitive information is known as social engineering, and the process of doing so is known as a social engineering attack (SEA). There are various definitions of social engineering, and a number of different models of social engineering attacks exist [1–5]. The authors of this paper considered different definitions of social engineering and social engineering attack taxonomies in a previous paper, *Towards an Ontological Model Defining the Social Engineering Domain* [6], and formulated a definition for both social engineering and a social engineering attack. In addition, the authors proposed an ontological model for a social engineering attack and defined social engineering as “the science of using social interaction as a means to persuade an individual or an organisation to comply with a specific request from an attacker where either the social interaction, the persuasion, or the request involves a computer-related entity” [6].

As clearly stated by various authors [7–10], the human component is one of the most vulnerable elements within security systems. Unfortunately it is the tendency towards cooperation and helpfulness in human nature that make people vulnerable to the techniques used by social engineers, as social engineering attacks exploit various psychological vulnerabilities to manipulate the individual to disclose the requested information [7, 10]. It is also the case that more and more individuals are exposed to electronic computing devices as the costs of these devices are decreasing drastically. Electronic computing devices have become significantly more affordable during the past few years and due to this nearly everyone has access to these devices. This provides the social engineer with more

Based on: “Underlying Finite State Machine for the Social Engineering Attack Detection Model”, by F. Mouton, A. Nottingham, L. Leenen and H.S. Venter which appeared in the Proceedings of Information Security South African (ISSA) 2017, Johannesburg, 16 & 17 August 2017. © 2017 IEEE

victims to target using skillfully crafted social engineering attacks.

The authors have previously performed research within the field of social engineering that focused on formalising and expanding upon concepts in the field. This research included proposing a social engineering attack framework, providing social engineering attack examples, considering ethical questions relating to social engineering, and both developing and revising the Social Engineering Attack Detection Model (SEADM) [11–14]. The previous iteration of the SEADM focused on covering all three different types of communication mediums for social engineering attacks [14]. Whilst using the SEADM to determine whether it is effective to detect social engineering attacks, it was noted that each set of questions focuses on a specific context. Also, the current iteration of the SEADM indicates that additional questions can be added to the model to address different implementation environments, but does not explicitly state how this should be done.

This paper focuses on addressing this problem by formalising the latest iteration of the SEADM into an abstracted deterministic finite state automata. The authors are not aware of similar approaches by other researchers. In its original form, SEADM was constructed as a non-deterministic flow chart that relied on general, qualitative sub-procedures to provide a model for detecting social engineering attacks. While effective as a procedure for reducing risk, the model made no provision and provided no guidance on how additional actions relevant in specific contexts and domains could be included, and at what points in the model these inclusions should be placed. Due to the inclusion of cycles in the SEADM model, the process was also non-deterministic, which added additional and unnecessary complexity in implementing the process.

This research aims to improve the extensibility of the SEADM, and to reduce its implementation complexity by restructuring the process to be cycle-free and deterministic. The extensibility of the model is addressed by replacing the qualitative sub-procedures with generalised states that better define the role of each sub-process, while treating questions posed in each state as general examples that may be expanded or removed, and not as a definitive collection of necessary queries. Organising the model as a finite set of generalised states provides a more concise representation of the process that encapsulates the broad set of questions into distinct units of related, deterministic, context specific states. This adjustment is intended to improve extensibilty, while simultaneously reducing the complexity of implementing the model as an organisational process or in software.

The remainder of the paper is structured as follows. Section II provides background information on the previous social engineering model and discusses the authors' previous work. Section III proposes the underlying deterministic finite state machine of the SEADM. Section IV provides a discussion on how each of the states were derived from the SEADM. Section V introduces the reader to social engineering attack templates, and evaluates the improved version of the SEADM against these templates. Section VI concludes the paper.

2. PREVIOUS ITERATIONS OF THE SOCIAL ENGINEERING ATTACK DETECTION MODEL

Many models and taxonomies have been proposed for social engineering attacks [1–6, 11]. The authors' ontological model depicts that a social engineering attack "employs either direct communication or indirect communication, and has a social engineer, a target, a medium, a goal, one or more compliance principles and one or more techniques" [6]. The ontological model clearly splits social engineering into three categories, namely bidirectional communication, unidirectional communication and indirect communication.

The initial SEADM was designed to cater specifically for social engineering attacks utilising bidirectional communication such as a call centre environment [13, 15]. This research was the first attempt to develop a detection model for social engineering attacks, as at the time of publishing the article there was still only limited research available in this field. Most of the research in this domain still centres around the training of users [7, 16, 17]. During the revision of the SEADM, the steps have been generalised to cater for all three communication categories, namely bidirectional communication, unidirectional communication and indirect communication. It has also been shown that the model is effective in detecting social engineering attacks by testing the model against known social engineering attack examples [14]. The previous iteration of the SEADM is depicted in Figure 1.

The following section discusses the representation of the SEADM as an abstracted finite state machine and provides a short discussion on each of the states.

3. UNDERLYING FINITE STATE MACHINE OF THE SEADM

A finite state machine (also known as finite state automaton) is an imaginary machine that embodies the idea of a sequential circuit. It has a finite set of states with a start state and accepting states, and a set of state transitions [18]. Finite state machines are commonly employed in the design and implementation of modern software and electronics, and range from simple and highly abstract models of computation or processing to complex and concrete executable mechanisms and physical circuitry.

Finite state machines can be deterministic or non-deterministic. A deterministic machine has exactly one path for every input-state pair. In a non-deterministic machine there may be multiple valid transitions for every input-state pair, and the chosen transition is not defined; any transition can be followed. A deterministic finite

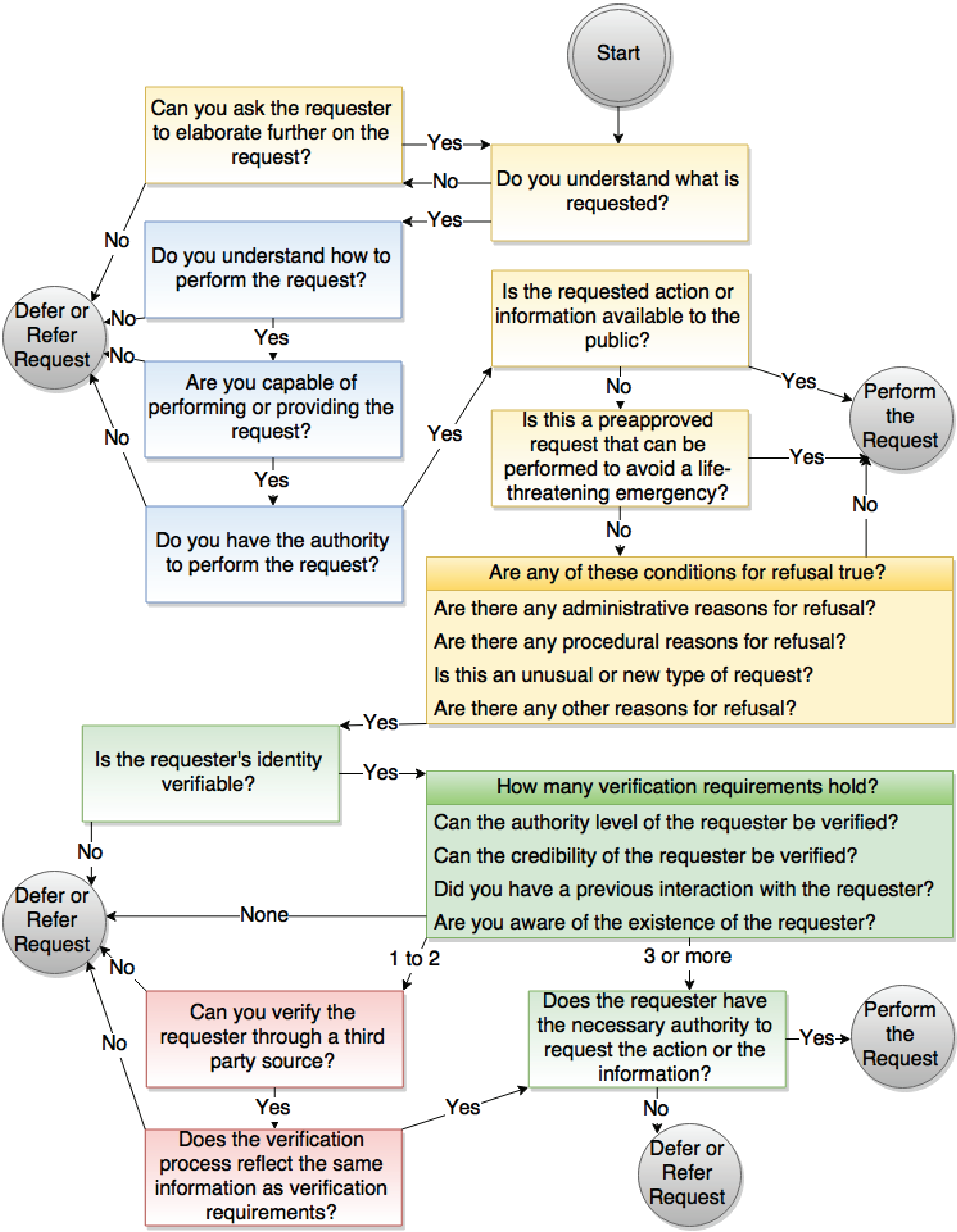

Start
Do you understand what is requested?
Yes
No
Can you ask the requester to elaborate further on the request?
Yes
Do you understand how to perform the request?
No
Defer or Refer Request
No
Yes
Are you capable of performing or providing the request?
No
No
Yes
Do you have the authority to perform the request?
Yes
Is the requested action or information available to the public?
Yes
Perform the Request
No
Is this a preapproved request that can be performed to avoid a life-threatening emergency?
Yes
No
No
Are any of these conditions for refusal true?
Are there any administrative reasons for refusal?
Are there any procedural reasons for refusal?
Is this an unusual or new type of request?
Are there any other reasons for refusal?
Yes
Is the requester's identity verifiable?
Yes
How many verification requirements hold?
Can the authority level of the requester be verified?
Can the credibility of the requester be verified?
Did you have a previous interaction with the requester?
Are you aware of the existence of the requester?
No
Defer or Refer Request
None
1 to 2
3 or more
No
Can you verify the requester through a third party source?
No
Yes
Does the verification process reflect the same information as verification requirements?
Yes
Does the requester have the necessary authority to request the action or the information?
Yes
Perform the Request
No
Defer or Refer Request

Figure 1: Social Engineering Attack Detection Model

state machine is a state machine that is guaranteed to complete for all inputs in a finite amount of time, while a non-deterministic finite state machine may execute indefinitely or fail to progress toward completion for certain input sets. A finite state machine is provably deterministic if and only if it is both free of cycles (that is, no state is ever revisited after being processed once) and defines a transition to a new state for each potential input in every state (that is, any valid input into a state results in a transition to a new state). These two properties together preclude the possibility of the state machine entering an infinite loop, either within a single state or between a collection of states, thereby ensuring that processing will always complete within a finite number of steps.

The finite state automaton described in this paper is an abstract or general model, and is intended to provide a structured but flexible deterministic high-level overview of the steps taken to mitigate a social engineering attack, improving upon the original SEADM flow-chart approach. While the SEADM flow-chart provides a static procedural template for implementing detection mechanisms for social engineering attacks, the finite state diagram provides a more accessible, abstract and extensible model that highlights the inter-connections between task categories associated with different scenarios. The abstract state-based model is intended to help facilitate the incorporation of domain, system or organisation specific extensions by grouping related activities into distinct categories, subdivided into one or more generalised nodes. Should a specific task, necessary in a particular domain, systems or organisational context not be included in the flow-chart, the state diagram may be used to infer the correct location within the model to incorporate the task. It further facilitates additional analysis on state transitions that are difficult to extract from the more verbose flow-chart.

The current iteration of the SEADM, as depicted in Figure 1, utilised four different state categories: the request, receiver, requester and third party. The request states, indicated in yellow, assesses information about the request itself. The receiver state, indicated in blue, considers the person handling the request and whether this person (the receiver) understands and is allowed to perform the request. The requester states, indicated in green, considers the requester and whether any information about the requester can be verified. The third party state, indicated in red, considers the involvement of a third party to support external verification of information supplied by the requester.

The same four categories and colour schemas are maintained in the state machine as they depict the primary topic that each specific state deals with, allowing one to better understand the attack detection model. The state machine is depicted in Figure 2. Each state has an associated letter which explains which condition needs to be met before the transition can be performed. As an example, a state can have an alphabet of Y and $\neg Y$. The symbol $\neg$ indicates negation, so $\neg Y$ is Not Y, or more accurately, the opposite of Y.

The states in Figure 2 are explained as follows:

- S_1 deals with the receiver's understanding the request. The request is either 'understood' (U) or 'not understood' ($\neg U$) by the receiver.
- S_2 deals with requesting more information from the requester by the receiver in an effort to properly understand the request. There is either 'sufficient information' (I) or 'insufficient information' ($\neg I$) for the receiver to understand and thus perform the request.
- S_3 deals with the capability of the receiver to perform the request. The receiver is either 'capable' (C) or 'incapable' ($\neg C$) of performing the request.
- S_4 deals with further verification requirements that may need to be met. The request either has further 'verification requirements' (R) or has 'no verification requirements' ($\neg R$). Additional verification requirements may be necessary in sensitive or secure contexts.
- S_5 deals with whether the receiver can verify and trust the identity of the requester, and how many of the verification steps hold. The verification steps hold to either a 'high amount' (V_H) where all or nearly all of the supplied information can be verified, a 'medium amount' (V_M) where the majority of information can be verified, or 'low amount' (V_L) where the majority of supplied information cannot be verified. These levels can be calibrated to be more or less restrictive, depending on the operating environment, and govern how the receiver should proceed.
- S_6 deals with trusted third party verification of information supplied by the requester. The requester is either 'verified' (T) by a third party or is 'not verified' ($\neg T$).
- S_7 deals with the authority of the requester. The requester either has 'sufficient authority' (A) or 'insufficient authority' ($\neg A$) for the particular request.
- S_F is an end state. This state indicates the failure state. The request should not be performed by the receiver, and should either be denied outright, or referred to a receiver with more knowledge or authority.
- S_S is an end state. This state indicates the success state. The request should be performed by the receiver.

The states are elaborated further in Section 4. A description of the full state machine in mathematical notation follows. The finite state machine is a 5-tuple consisting of the finite set of input alphabet characters Σ, the finite set of states Q, the start state S_0, a set of accepting states F, and a set of state transitions δ that contains 3-tuples representing state

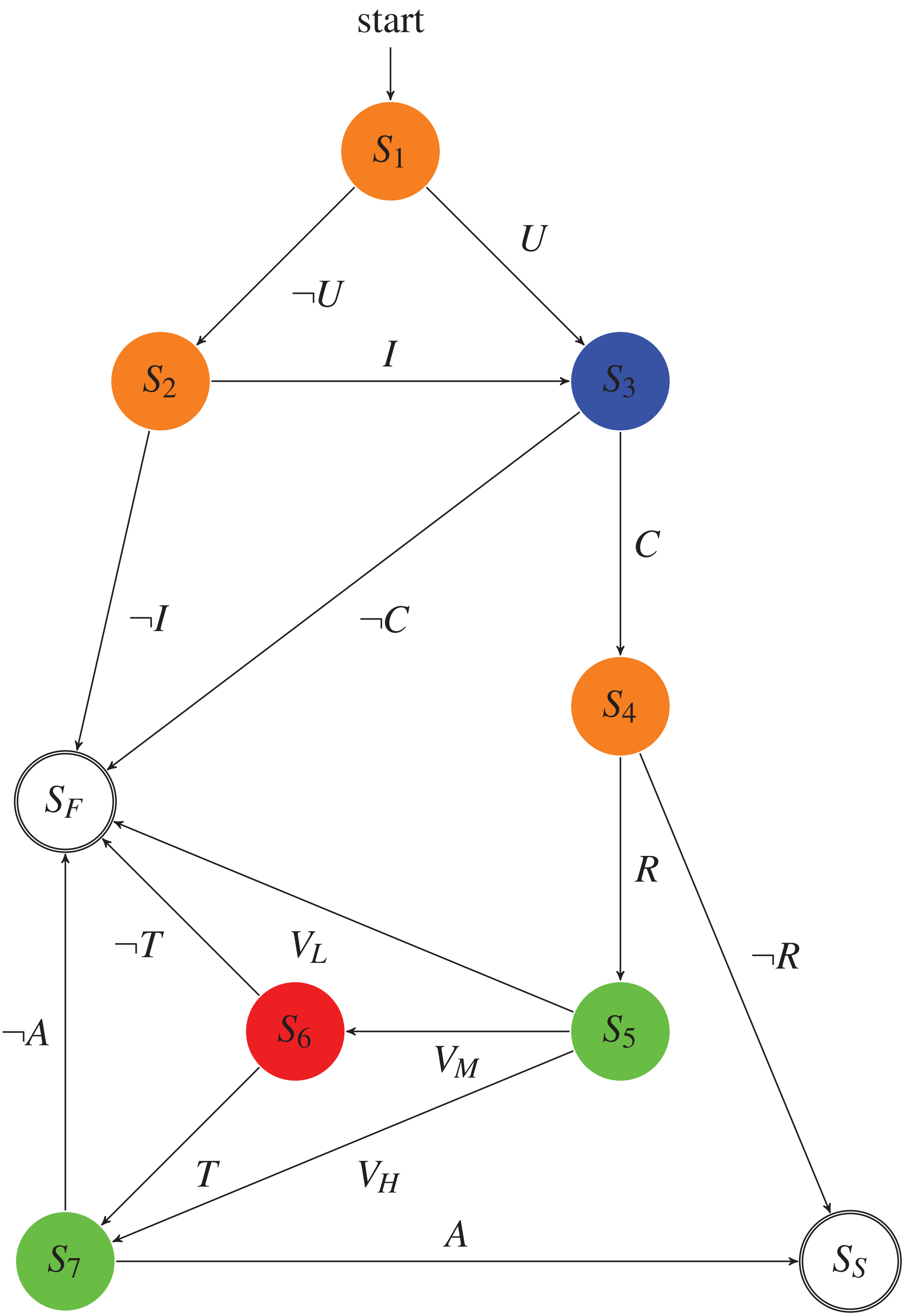


Figure 2: Underlying Finite State Machine of the SEADM

transitions. A 3-tuple in δ consists of a current state, a current input and the next state.

$$
\begin{array}{rcl}
\Sigma & = & \{U,\neg U,I,\neg I,C,\neg C,R,\neg R,V_L,V_M,V_H,T,\neg T,A,\neg A\} \\
Q & = & \{S_1,S_2,S_3,S_4,S_5,S_6,S_7,S_S,S_F\} \\
S_0 & = & S_1 \\
\delta & = & \{ \\
 & & (S_1,U,S_3),(S_1,\neg U,S_2) \\
 & & (S_2,I,S_3),(S_2,\neg I,S_F) \\
 & & (S_3,C,S_4),(S_3,\neg C,S_F) \\
 & & (S_4,R,S_5),(S_4,\neg R,S_S) \\
 & & (S_5,V_L,S_F),(S_5,V_M,S_6),(S_5,V_H,S_7) \\
 & & (S_6,T,S_7),(S_6,\neg T,S_F) \\
 & & (S_7,A,S_S),(S_7,\neg A,S_F) \\
 & & \} \\
F & = & \{S_S,S_F\}
\end{array}
$$

Using both Figure 2 and the provided mathematical model it is straightforward to infer a state transition table. Table 1 depicts all the possible state transitions given a specific input for each state. For all input states, the output is either a terminal node or a node with a higher state index. This illustrates that the state machine is deterministic, eliminating cycles present in the original SEADM flowchart.

Table 1: State Transition Table for the SEADM

State Input	S_1	S_2	S_3	S_4	S_5	S_6	S_7
U	S_3	—	—	—	—	—	—
$\neg U$	S_2	—	—	—	—	—	—
I	—	S_3	—	—	—	—	—
$\neg I$	—	S_F	—	—	—	—	—
C	—	—	S_4	—	—	—	—
$\neg C$	—	—	S_F	—	—	—	—
R	—	—	—	S_5	—	—	—
$\neg R$	—	—	—	S_S	—	—	—
V_L	—	—	—	—	S_F	—	—
V_M	—	—	—	—	S_6	—	—
V_H	—	—	—	—	S_7	—	—
T	—	—	—	—	—	S_7	—
$\neg T$	—	—	—	—	—	S_F	—
A	—	—	—	—	—	—	S_S
$\neg A$	—	—	—	—	—	—	S_F

To further show that the state machine model is deterministic, resulting in a valid outcome of either success or failure for all given alphabet sequences, a transition table indicating all possible input alphabet sequences (paths) and their corresponding results are shown in Table 2. Each row in the table represents a path. Σ_i indicates the input character is the i-th character in the path. The symbol ∀ indicates no transition occurred in the i-th position of the path in the case of shorter paths. This table shows that for all possible paths, the state machine returns either success or failure in a finite number of steps, and is thus deterministic.

Table 2: State Transition Table for all Input Alphabets

	Input Alphabet							Output	
No	Σ_1	Σ_2	Σ_3	Σ_4	Σ_5	Σ_6	Σ_7	S_S	S_F
1	U	∀	C	R	V_H	∀	A	✓	—
2	U	∀	C	R	V_M	T	A	✓	—
3	U	∀	C	R	V_M	T	$\neg A$	—	✓
4	U	∀	C	$\neg R$	∀	∀	∀	✓	—
5	U	∀	C	R	V_H	∀	$\neg A$	—	✓
6	U	∀	C	R	V_M	$\neg T$	∀	—	✓
7	U	∀	C	R	V_L	∀	∀	—	✓
8	U	∀	$\neg C$	∀	∀	∀	∀	—	✓
9	$\neg U$	$\neg I$	∀	∀	∀	∀	∀	—	✓
10	$\neg U$	I	C	R	V_H	∀	A	✓	—
11	$\neg U$	I	C	R	V_M	T	A	✓	—
12	$\neg U$	I	C	R	V_M	T	$\neg A$	—	✓
13	$\neg U$	∀	C	$\neg R$	∀	∀	∀	✓	—
14	$\neg U$	I	C	R	V_H	∀	$\neg A$	—	✓
15	$\neg U$	I	C	R	V_M	$\neg T$	∀	—	✓
16	$\neg U$	I	C	R	V_L	∀	∀	—	✓
17	$\neg U$	I	$\neg C$	∀	∀	∀	∀	—	✓

Having considered the high-level state-based model for the SEADM, the following section elaborates on the purpose of each state and exactly how it was derived from the SEADM.

4. DISCUSSION OF EACH STATE

This sections explains how each of the states has been designed and integrated from the SEADM model. Throughout this discussion the alphabet of the states are provided. During the discussion on the states, it is also shown how each state relates to the SEADM. Each state has been generalised to such an extent that it can contain any number of questions required to achieve a specific transition result. This provides a rough guide for flexibility and extensibility, depending on the particular context the model is applied to.

4.1 State S_1: Understanding the Request

This state considers whether the receiver of the request fully understands the request in its entirety. This means that the requester should have provided all the information required to enable the receiver to perform the request in full. The question that was provided in the SEADM was "Do you understand what is requested?"

In the SEADM there was only one question asked here. This has been created as the start state as it is still important to fully understand what is requested before the request can be processed any further. In this state the alphabet is as follows:

- U represents that the request is understood in its entirety and that the receiver has all the necessary information in order to be able to perform the request.
- $\neg U$ represents that the request is not fully understood and that the receiver requires more information about the request.

This state can only transition in one of two ways and is depicted as follows:

- (S_1, U, S_3) if the request is fully understood.
- $(S_1, \neg U, S_2)$ if more information is required.

4.2 State S_2: Requesting information to fully understand the request

In the SEADM this state was previously grouped together with the previous question. This resulted in a loop, as the receiver could always ask the requester to elaborate further. At some point the requester would no longer be able to elaborate on the request as there would be no more additional information available, and the loop would terminate. At what point the loop would terminate was unclear, and could not be formalised into a deterministic state machine without additional complexity.

Representing this task as a distinct state more clearly defines this, as it deals directly with the information that is required to complete the request and not whether one can request more information. This state considers whether the requester can provide information to such an extent that the receiver is able to fully understand the request. A state transition occurs when either sufficient information is provided, or it is determined that the requester cannot provide sufficient information. Previously the question that was asked was as follows, “Can you ask the requester to elaborate further on the request?” In this state all questions should be aligned with whether the requested information, in the case that additional information is provided, allows the receiver to understand the request in full or not. In this state the alphabet is as follows:

- I represents that the requester can and has provided enough information for the request to be understood in its entirety by the receiver.
- $\neg I$ represents that the receiver is unable to understand the request in full. This may be because the requester could not provide more information, the requester could not be reached (in the case of remote communication), or that the information provided by the requester was insufficient or incomplete.

This state can only transition in one of two ways and is depicted as follows:

- (S_2, I, S_3) if the requester can provide sufficient information to understand the request.
- $(S_2, \neg I, S_F)$ if the requester is unable to provide sufficient information to understand the request, or cannot be reached.

4.3 State S_3: Does the receiver meet the requirements to perform the request?

State S_3 is used to determine whether the receiver meets all the requirements to perform the request. This state is associated with three questions in the SEADM. The questions are as follows:

- “Do you understand how to perform the request?”
- “Are you capable of performing or providing the request?”
- “Do you have the authority to perform the request?”

The goal of this state is to ensure that the individual who deals with the specific request has the necessary skill level and has the required authority to perform the request. Each question in this state deals directly with the role of the receiver and determines whether the request has been issued to the correct receiver. In this state the alphabet is as follows:

- C represents that the receiver has met all the requirements to be capable of performing the requested action or to provide the requested information.
- $\neg C$ represents that the receiver does not meet the requirements to be capable of dealing with the request.

This state can only transition in one of two ways and is depicted as follows:

- (S_3, C, S_4) if the receiver is able to service the request.
- $(S_3, \neg C, S_F)$ if the receiver is unable or incapable of servicing the request.

4.4 State S_4: Does the request have any further requirements that need to be met before the request can be serviced?

State S_4 deals with the the request itself and whether there are any special conditions or requirements, such as policies and procedures that need to be followed, associated with the request. Examples of special conditions include whether the request relates to information already in the public domain and is accessible to all, or whether the request is a life threatening emergency. If the request is already in the public domain, for instance, there are no further requirements that need to be met. In the event of a life threatening emergency, the outcome depends on whether there are set policies in place to deal with such contingencies. For instance, if there is a policy in place that allows medical personnel to rather err on the side of caution in order to save an individuals life, there will be no further requirements that need to be met and the request may take place. The previous model dealt with these examples using the following questions:

- “Is the requested action or information available to the public?”
- “Is this a pre-approved request that can be performed to avoid a life-threatening emergency?”

In addition, this state also deals with any further requirements that need to be met. Examples of such requirements are any policies or procedures that are in place that require further verification of the identity or authority of the requester. This state can also cater for unusual requests. An unusual request is a request that is new to the receiver and/or is a request that the receiver does not usually deal with on a regular basis. By following the rest of the model, the receiver ensures that adequate information about the requester is obtained before the request is performed. It also allows the receiver time to think about the request and whether the request should be performed for the receiver. The previous model dealt with these examples using the following questions:

- “Are there any administrative reasons for refusal?”
- “Are there any procedural reasons for refusal?”
- “Is this an unusual or new type of request?”
- “Are there any other reasons for refusal?”

The goal of this state is to ensure that any request which is already public information should be immediately performed and that any request that requires verification should result in further interrogation of the requester. In this state the alphabet is as follows:

- R represents that the request has further verification requirements that need to be met in order to be able to perform the requested action or to provide the requested information.
- $\neg R$ represents that the request has no further verification requirements and that the requested action can be performed or that the requested information can be provided.

This state can only transition in one of two ways and is depicted as follows:

- (S_4, R, S_5) if further verification is required.
- $(S_4, \neg R, S_S)$ if no further verification is required.

4.5 State S_5: *To what extent is the requester’s identity verifiable?*

State S_5 aims to address the question of the extent of verifiability of the requester’s identity. The identity of the requester is determined to ensure that the requester has sufficient privileges to request the specific action or information. It is not always possible to verify the identity of the requester in full. The type of communication medium that is utilised by the requester usually will dictate to what extent the identity of the requester can be verified. Typically, if the requester makes the request in person one is able to verify significantly more information about the requester than would be possible over an e-mail or postal mail. It may also be the case that the request is performed over unidirectional communication and that the receiver is unable to receive any further communication from the requester.

The previous model provided a fourth question as to whether the requester’s identity is verifiable at all. The state machine has combined not being verifiable and having a low level of verification as the same transition as both states lead to S_F. The questions that were previously used to perform the verification requirements are as follows:

- “Can the authority level of the requester be verified?”
- “Can the credibility of the requester be verified?”
- “Did you have a previous interaction with the requester?”
- “Are you aware of the existence of the requester?”

All of the questions catered for a single point of verification. It was also noted in the previous model that the level of verification required to transition to different states should be based on what type of environment the model is applied to. The state machine makes this more generalised by having three possible transitions where there is a low, medium or high level of verification. The threshold for low, medium and high must still be determined based on the environment or context, but the state diagram is more flexible when more questions are added. In this state the alphabet is as follows:

- V_L represents that there is a low level of verification as only a few of the verification elements could be verified.
- V_M represents that there is a medium level of verification as some of the verification elements could be verified, but not all of them.
- V_H represents that there is a high level of verification as all or most of the verification elements could be verified. Note that edge V_H should not be followed if any information provided in a request is inconsistent with, or differs substantially from, established or generally available information.

This state can only transition in one of three ways and is depicted as follows:

- (S_5, V_L, S_F) if only a few verification elements hold.
- (S_5, V_M, S_6) if a moderate amount of verification elements hold.
- (S_5, V_H, S_7) if all or most verification elements hold.

4.6 *State S_6: Can you verify the requester's identity from a third party source?*

State S_6 is only entered when there is a medium level of verification requirements that have been met. If the requester could not be fully verified directly, the third party source is utilised to determine whether the information provided by the requester was indeed truthful. The previous model did not elaborate much on the third party verification and only had two questions associated with it, as follows:

- "Can you verify the requester through a third party source?"
- "Does the verification process reflect the same information as the verification requirements?"

The supplied questions did not mention specifically which verification requirements needed to be verified. Utilising the extensibility and generality of the state machine, one could build intelligence into the model to only ask such questions when the verification requirements were obtained directly from the requester. In this state the alphabet is as follows:

- T represents that the verification requirements, as obtained from the requester, fully corresponds to the information that was obtained from the third party source.
- $\neg T$ represents that the verification requirements, as obtained from the requester, do not fully correspond to the information that was obtained from the third party source.

This state can only transition in one of two ways and is depicted as follows:

- (S_6, T, S_7) if information supplied by the requester is fully verified by the third party source.
- $(S_6, \neg T, S_F)$ if information supplied by the requester is not fully verified by the third party source.

4.7 *State S_7: Does the authority level of the requester provide them with sufficient rights to request the action or information?*

State S_7 utilises all the information obtained throughout the model and asks the receiver questions based on the information obtained. This state aims to determine whether the requester has the necessary authority and rights to request the action or the information. The previous model only asked "Does the requester have the necessary authority to request the action or the information?" This state elaborates on this by allowing the receiver to verify whether the requester has sufficient authority to gain access to the request or the information. In this state the alphabet is as follows:

- A represents that the requester has sufficient authority to be allowed to request the receiver to perform the action or to provide the information.
- $\neg A$ represents that the requester does not have sufficient authority and is thus not allowed to request the receiver to perform the action or to provide the information.

This state can only transition in one of two ways and is depicted as follows:

- (S_7, A, S_S) if the requester has sufficient authority for the request to be processed.
- $(S_7, \neg A, S_F)$ if the requester does not have sufficient authority for the request to be processed.

4.8 *State S_F: Halt the request*

This is the negative result state. In this state the request will be halted. In some environments one could opt to rather defer the request to a more authoritative receiver. Deferring the request may lead to the request never being performed, and can be considered as the request having been halted. In the case where the receiver is part of an organisation, there is the option to refer the request to a more authoritative person in the same organisation. This will allow someone else who may be better suited to determine whether to perform or halt the request.

4.9 *State S_S: Perform the request*

This is the positive result state of the model. In this state the receiver is allowed to perform the single request from the requester.

The next section briefly discusses social engineering attack templates, which are then tested against the social engineering attack detection model.

5. APPLICATION OF THE SOCIAL ENGINEERING ATTACK DETECTION MODEL

Previously, social engineering attack templates have been proposed to provide researchers with a set of social engineering attack examples that can be used to verify and make comparisons between models, processes and frameworks within social engineering [12]. All of the templates were derived from real-world social engineering attacks that have been documented in either news articles, technical reports, research reports, films or blogs. The news articles, technical reports, research reports or blogs did not always contain all of the information regarding the social engineering attack. This lack of information was addressed by proposing the templates as a more generalised form of the social engineering attacks provided in the literature [12].

Each template contains the full description of every phase and associated steps of the social engineering attack framework in such a way that each template will provide repeatable results when used for verification and comparison purposes. The templates are also kept as simple as possible so that they can be expanded upon to create more elaborate scenarios with similar principal structures. The templates can also be used to verify or compare other models, processes and frameworks without having to physically perform the attack and potentially cause harm to innocent targets [19]. The rest of this section is dedicated to testing the SEADM against the social engineering attack templates.

In the first scenario, a bidirectional communication template, the social engineer pretends to be someone who works on the management floor and has to convince a cleaner that he is indeed an employee [12]. He requests the cleaner to give him access to the management floor. In the second scenario, a unidirectional communication template, the social engineer attempts to obtain financial gain by sending out paper mail in which the letter requests a group of individuals to make a small deposit into a bank account owned by the attacker [12]. In the third scenario, an indirect communication template, the social engineer attempts to gain unauthorised access to a workstation in an organisation by using a storage medium device [12].

In each scenario the reader is provided with a generic description of the attack as taken from social engineering attack templates. This generic description is then populated with elements, both subjects and objects, from real-world examples of social engineering attacks, as provided in the discussion of the specific social engineering attack template. Using the generic description, the elements from the real-world examples and the fully detailed flow of the attack as provided in each phase and step of the social engineering attack framework, one is able to devise a social engineering attack scenario. This scenario is then reflective of a real-world example of which every phase and step is fully documented as per the social engineering attack framework. Using the proposed social engineering attack templates, one is able to formulate a social engineering attack scenario that always follows the same process, with regards to phases and steps, whilst the social engineering attack is still representative of a real-world scenario.

The remainder of this section is dedicated to mapping the social engineering attack templates to the states of the SEADM and verifying whether the social engineering attack detection model can assist in detecting these attacks. The following discussions will highlight at which points the SEADM could prevent the success of the social engineering attack if it were properly followed, but will make allowances for failure in following the SEADM to fully explore each scenario. Note that the discussion of each state makes implicit reference to the subsidiary questions defined and described for the state in Section 4.

5.1 Bidirectional Communication Scenario

The generic description for this scenario reads as follows: "This template illustrates an SEA where the attacker attempts to gain physical access to a computerised terminal at the premises of an organisation. The assumption is that when the attacker has once gained access to the computerised terminal, he/she is deemed to have been successful. The attacker is now able to install a backdoor onto the computerised terminal for future and further access from the outside." This scenario is populated with elements from a real-world example where the social engineer pretends to be someone who works on the management floor and convinces a cleaner of his supposed role. The goal of the attacker is to manipulate the cleaner into granting the social engineer access to the management floor. This allows the social engineer to gain physical access to the computerised terminals on the management floor [20, 21].

In this scenario a social engineer has to convince the cleaner, the receiver, to believe that he is indeed a staff member. In this scenario, the cleaners have full access to the building, yet, their security awareness is typically quite low. They are not trained to respond to unusual requests such as giving other employees access to the management floor. If the request is successful, access has been gained to the management floor, and a key logger is deployed onto a workstation. This attack is performed using bidirectional communication because the social engineer communicates with the cleaner and convinces him that the social engineer is allowed to have access to the management floor and the workstations.

S_1 **– Understanding the Request**: The request from the social engineer should clearly state that access needs to be gained to the management floor. The social engineer can also justify to the receiver why access is required to further allow the receiver to understand the request. If the receiver understands the request, edge U is followed to S_3. If the receiver does not initially understand the request, the SEADM would move to state S_2 via edge $\neg U$.

S_2 **– Requesting information to fully understand the**

request: In the unlikely event that the request is not initially understood, the social engineer would explain his/her request in more detail. As the request itself is straightforward, if unusual, it is assumed that it will ultimately be understood, resulting in a transition (S_2, I, S_3).

S_3 **– Does the receiver meet the requirements to perform the request?**: In this scenario, the receiver does not have the authority to grant access to the management floor. If the SEADM were followed, appropriate security policy were in place, and the cleaner had sufficient knowledge of said policy, the cleaner would decline the request and refer the social engineer to a more qualified individual, thwarting the attack. This would be reflected in the SEADM by a transition via edge $\neg C$ to the failure state, S_F. For the purposes of discussion, and assuming the cleaner is not properly trained and fears reprisals from a more senior employee, the transition (S_3, C, S_4) is followed.

S_4 **– Does the request have any further requirements that need to be met before the request can be serviced?**: In the scenario, only management and cleaners should have access to the management floor. The requested action is thus not available to the public, or in service of a medical emergency, and is subject to further verification of the requester's identity. If the SEADM were followed at this point, this would result in the transition (S_4, R, S_5).

S_5 **– To what extent is the requester's identity verifiable?**: As bidirectional communication is utilised, it allows for the receiver to communicate back via face to face communication and ask more questions to verify the requester. In this case, the authority principle is utilised and the social engineer mimics an authoritative figure who should have access to the management floor. The pretext utilised during this attack is that the social engineer is part of management and that he or she should have access to the management floor. The receiver is only able to verify the falsified authority level from the social engineer in this scenario. Since only a single verification requirement is met, transition (S_5, V_L, S_F) would be the most appropriate, resulting in the request being deferred to an individual with more authority, thus thwarting the attack. If the social engineer can provide additional false information, and the cleaner decides to err on the side of caution, the transition (S_5, V_M, S_6) may be followed instead. As the social engineer is not known to the cleaner and is not an actual employee, edge V_H should not be followed.

S_6 **– Can you verify the requester's identity from a third party source?**: The receiver will now have the ability to verify the information from another employee on the management floor. In the case that there are no other employees on the management floor, the transition $(S_6, \neg T, S_F)$ will be taken and the social engineering attack will be thwarted. If it is assumed that there are other employees on the management floor who can be contacted to verify the information, the receiver will be able to ask whether the authority level of the social engineer is indeed true. The other employee will deny this and thus the verification process will show that the information provided is not the same as the verification requirements. Consequently, the transition $(S_6, \neg T, S_F)$ will still be taken, and the social engineering attack will be thwarted.

S_7 **– Does the authority level of the requester provide them with sufficient rights to request the action or information?** Assuming the receiver is untrained, the SEADM is not followed, and the requester's identity is incorrectly assumed to be verified, their presumed authority would grant them access to the management floor, resulting in transition (S_7, A, S_S). If the SEADM were followed, however, the attack would move to the failure state S_F twice before this state is first reached.

In this scenario, the SEADM would have detected and thwarted the social engineering attack at two separate points before reaching the final state, S_7, in the model.

5.2 Unidirectional Communication Scenario

The generic description for this scenario reads as follows: "This template illustrates an SEA in which the attacker attempts to obtain financial gain by sending out paper mail. This letter requests a group of individuals to make a small deposit into a bank account owned by the attacker. In this template, the attacker develops a phishing letter that masks the attacker as a charity organisation requesting donations. Once the attacker has received the small deposit from the targeted individual, the SEA is deemed to be successful." This scenario is populated with elements from the real-world example where the social engineer performs a pretext using postal letters. The social engineer pretends to be various officials, internal employees, employees of trading partners, customers, utility companies or financial institutions, and the social engineer solicits confidential information by using a wide range of persuasive techniques [22].

In this scenario, the attacker will develop a phishing letter that masks the attacker as a charity organisation requesting donations. The phishing letter contains the contact details, the logo and the purpose of the charity to improve the authenticity of the letter. This attack uses unidirectional communication and thus the receiver is not able to communicate with the attacker. The rest of this section maps the scenario to the model.

S_1 **– Understanding the Request**: The letter from the social engineer should clearly state that a receiver is requested to make a donation to the specific charity. The letter will include all the required details because this receiver cannot communicate with the social engineer. In the SEADM, this should result in the transition (S_1, U, S_3). In the unlikely event that the receiver does not understand the request, the transition $(S_1, \neg U, S_2)$ will be followed.

S_2 **– Requesting information to fully understand the request**: The social engineer would have tried to ensure that the targeted individual fully understands the request.

As the attack uses unidirectional communication, the receiver is unable to request further information. As a result, in the SEADM, the only available state transition is $(S_2, \neg I, S_F)$, thwarting the attack.

S_3 **– Does the receiver meet the requirements to perform the request?**: The receiver is the owner of their bank account and has the required information to make the deposit. Given that their understanding of the request has been ascertained, and assuming they have a bank account with an available balance and are both capable and authorised to deposit their money into another bank account, the transition (S_3, C, S_4) will be followed.

S_4 **– Does the request have any further requirements that need to be met before the request can be serviced?**: The requested action is to make a deposit into the bank account of the requester. This action is only available to receiver, and is not in service of preventing a life-threatening emergency. Furthermore, this request can be seen as either unusual or new as the requester would not usually receive this specific type of letter from the charity. It is additionally likely that the requester feels uneasy about, or suspicious of, the request, which is itself a reason for refusal. The transition S_3, R, S_4 should thus be followed.

S_5 **– To what extent is the requester's identity verifiable?**: Since unidirectional communication is utilised, the receiver can only verify the identity of the requester using the information provided in the received letter. While the receiver may be aware of the existence of the charity organisation, if the letter provides no additional information for verification purposes (such as contact details), or if information in the charity's official correspondence or online presence is inconsistent with information provided in the fabricated request, the SEADM indicates that the transition (S_5, V_L, S_F) should be followed, thwarting the attack. If the letter contains the actual contact details of the charity organisation, thereby providing some additional verifiable information, the transition (S_5, V_M, S_6) may be followed instead. As the request is unidirectional and does not specify the charity's actual bank details, the transition (S_5, V_H, S_7) should not be followed; edge V_H indicates very high confidence, which should not be followed when a request is unusual and provides limited verifiable information.

S_6 **– Can you verify the requester's identity from a third party source?**: As the charity is a well known charity, the request may be verified by contacting the charity directly. The receiver will make a phone call to the charity to verify the information. If the charity cannot be reached, the transition $(S_6, \neg T, S_F)$ should be followed, thwarting the attack. If the receiver is able to contact the charity directly, the receiver will be able to ask the organisation whether such a letter has in fact been sent out. The charity organisation will deny this and thus the verification process will show that the information provided is not the same as the verification requirements. Consequently, the transition $(S_6, \neg T, S_F)$ will be followed and the social engineering attack will be thwarted.

S_7 **– Does the authority level of the requester provide them with sufficient rights to request the action or information?** Assuming that the SEADM was not followed up to this point, and the receiver believes the information provided in the fabricated request, the receiver may consider the requester to have sufficient authority to make the request. If this is the case, the transition (S_7, A, S_S) may be followed, allowing the attack to succeed. If the SEADM was followed, however, state S_7 would not have been reached, preventing this transition from occurring.

While thwarting this form of attack requires some diligence on the part of the receiver, following the SEADM should prevent the SEA from succeeding.

5.3 Indirect Communication Scenario

The generic description for this scenario reads as follows: "This template illustrates an SEA in which the attacker attempts to gain unauthorised access to a workstation within an organisation by using a storage device. Once the target has plugged the storage device (in this case a USB flash drive) into the targeted workstation, the SEA is deemed to be successful; the attacker is now able to install a backdoor onto the workstation via the storage device. The social engineer can then proceed to use this workstation as a pivot point for any further attacks on the organisation." This scenario is populated with elements from the real-world example where the social engineer attempts to gain unauthorised access to a workstation in an organisation by using a storage medium device [23,24]. This attack is also depicted in a popular television series about penetration testing, Mr. Robot [23].

In this scenario, the organisation does not have a company policy in place that disallows employees plugging storage devices into their workstations. The social engineer will leave the device outside the organisation's building to be found by an employee. The device will be infected with a trojan so that when it is plugged into the workstation, it opens a backdoor for the social engineer to connect to the system remotely. As the storage device is left unattended, this attack utilises indirect communication. The rest of this section maps this scenario to the model.

S_1 **– Understanding the Request**: The storage medium device planted by the social engineer should be marked clearly to indicate that it contains important and/or confidential information. The social engineer expects that the receiver – the employee who finds this device – will either try to return the device to its rightful owner (a benevolent target), or attempt to access the contents of the drive (a malevolent target). The social engineer may attempt to manipulate a specific employee into finding the device, or may settle for any employee. As it is an inherent request, the request should be easily understandable to the target, and the transition (S_1, U, S_3) should be followed in the SEADM.

S_2 **– Requesting information to fully understand the request**: This state would not typically be entered, as the social engineer would have made certain that the storage medium device is deployed at such a location that only individuals who have access to a workstation and who understand how such devices work would find the device. If this is not the case, the attack runs the risk of immediate failure, as the receiver may take the device to a lost-and-found where the attack is indefinitely halted, or to the IT department where the attack may be detected. Either of these cases would result in the transition $(S_2, \neg I, S_F)$, a failure state.

S_3 **– Does the receiver meet the requirements to perform the request?**: As it is assumed that the organisation does not prohibit the use of external USB drives on company workstations, and the receiver has access to a workstation that supports the USB attachment, the receiver is capable of performing the request. In most instances, this would be sufficient for the requirements of performing the implicitly requested action, and the transition (S_3, C, S_4) would be followed. If the drive is labelled as confidential, however, the receiver may be considered to not have sufficient authority to attach the drive. In this instance, the transition $(S_3, \neg C, S_F)$ would be followed in the SEADM, resulting in the attack being halted or thwarted.

S_4 **– Does the request have any further requirements that need to be met before the request can be serviced?**: The implicit request is directed at the receiver to manipulate them into attaching the device to a company worstation, either to return it to its rightful owner or to determine its contents. This action is not available to the public, or in service of preventing a life-threatening emergency. While there are no administrative or procedural reason for refusal due to a lack of specific company policy, the request would be considered unusual, and so the transition (S_4, R, S_5) would be followed.

S_5 **– To what extent is the requester's identity verifiable?**: Since indirect communication is utilised, the only piece of information the receiver has is the physical storage device, which does not provide any external verification information. Following the SEADM, this would result in a transition (S_5, V_L, S_F), resulting in the implicit request being either halted or detected by a more knowledgeable individual who is allowed to safely, and on a secure workstation, verify the contents of the storage device and potentially contact the rightful owner. Neither V_M or V_H should be followed in this instance.

S_6 **– Can you verify the requester's identity from a third party source?**: While this state should not be reached, there is no available third-party to provide verification of the device. The only option in this instance would be to follow transition $(S_6, \neg T, S_F)$, similarly resulting in the failure of the attack.

S_7 **– Does the authority level of the requester provide them with sufficient rights to request the action or information?** If this state were hypothetically reached, the lack of specific organisational policy governing the use of external storage media would provide sufficient authority for the implicit request to be performed, resulting in a transition (S_7, A, S_S) and allowing the attack to succeed. Similar to S_6, however, this state should not be reached if the SEADM is properly followed.

This demonstrates how the SEADM may be successfully utilised to prevent an attack when SEAs use indirect communication.

The preceding three sections have shown, through the use of attack templates, how the SEADM may be applied to help detect and prevent several social engineering attack vectors that utilise bidirectional, unidirectional and indirect communication approaches. The SEADM is, however, only a tool, and is best utilised in conjunction with security awareness and robust security policy to reinforce and guide its appropriate application and use, largely dependent on the security context.

The following section concludes the paper with a summary of the advantages of the underlying finite state machine and how it was still able to thwart social engineering attack templates.

6. CONCLUSION

The protection of information is extremely important in modern society and even though the security around information is continuously improving, a weak point is still human actors who are susceptible to manipulation techniques. This paper explored social engineering as a domain and social engineering attack detection techniques as a process within this domain. To this end, both the previous papers by the authors, *Social Engineering Attack Detection Model: SEADM* [6] and *Social Engineering Attack Detection Model: SEADMv2* [14] were revisited.

Both of the previous iterations of the SEADM focused on expanding the capability of the accuracy of detection. The models were populated with questions catering for attacks utilising either bidirectional communication, unidirectional communication or indirect communication. Previous work did mention that the model is extensible with additional questions, but was unclear on how and where additional questions should be added.

This paper improves on the SEADM by providing the underlying finite state machine, which allows researchers to better understand and utilise the SEADM. Representing the SEADM as a finite state machine allows one to have a more concise overview of the process that is followed throughout the model. The model provided a general procedural template for implementing detection mechanisms for social engineering attacks. The state diagram provides a more abstract and extensible model that highlights the inter-connections between task categories associated with different scenarios. This paper also shows that the finite state machine is both deterministic and correct for all possible input alphabets, simplifying the process of implementing the SEADM model either as a process or in software. The state diagram is currently implemented as a mobile application as part of a social engineering prevention training tool [25].

The improved SEADM was further taken and tested against social engineering attack templates, showing that the SEADM remains capable of thwarting social engineering attacks. Adapting the SEADM to a finite state machine improved the extensibility of the model without negatively impacting on detecting social engineering attacks.

The SEADM, with the underlying finite state machine, can be used as a general framework to protect against social engineering attacks. Even if the model is not adhered to in respect of every request, it will cause one to think differently about requests — and this is already a step in the right direction.

REFERENCES

[1] D. Harley, “Re-floating the titanic: Dealing with social engineering attacks,” in *European Institute for Computer Antivirus Research*, 1998, pp. 4–29.

[2] L. Laribee, “Development of methodical social engineering taxonomy project,” MSc, Naval Postgraduate School, Monterey, California, June 2006.

[3] K. Ivaturi and L. Janczewski, “A taxonomy for social engineering attacks,” in *International Conference on Information Resources Management*, G. Grant, Ed. Centre for Information Technology, Organizations, and People, June 2011, pp. 1–12.

[4] F. Mohd Foozy, R. Ahmad, M. Abdollah, R. Yusof, and M. Mas’ud, “Generic taxonomy of social engineering attack,” in *Malaysian Technical Universities International Conference on Engineering & Technology*, Batu Pahat, Johor, November 2011, pp. 1–7.

[5] P. Tetri and J. Vuorinen, “Dissecting social engineering,” *Behaviour & Information Technology*, vol. 32, no. 10, pp. 1014–1023, 2013.

[6] F. Mouton, L. Leenen, M. M. Malan, and H. Venter, “Towards an ontological model defining the social engineering domain,” in *ICT and Society*, ser. IFIP Advances in Information and Communication Technology, K. Kimppa, D. Whitehouse, T. Kuusela, and J. Phahlamohlaka, Eds. Springer Berlin Heidelberg, 2014, vol. 431, pp. 266–279.

[7] J. W. Scheeres, “Establishing the human firewall: reducing an individual’s vulnerability to social engineering attacks,” Master’s thesis, Air Force Institute of Technology, Wright-Patterson Air Force Base, Ohio, March 2008.

[8] K. D. Mitnick and W. L. Simon, *The art of intrusion: the real stories behind the exploits of hackers, intruders and deceivers.*, W. Publishing., Ed. Indianapolis: Wiley Publishing, 2005.

[9] J. Debrosse and D. Harley, “Malice through the looking glass: behaviour analysis for the next decade,” in *Proceedings of the 19th Virus Bulletin International Conference*, September 2009.

[10] G. L. Orgill, G. W. Romney, M. G. Bailey, and P. M. Orgill, “The urgency for effective user privacy-education to counter social engineering attacks on secure computer systems,” in *Proceedings of the 5th Conference on Information Technology Education*, ser. CITC5 ’04. New York, NY, USA: ACM, 2004, pp. 177–181. [Online]. Available: http://doi.acm.org/10.1145/1029533.1029577

[11] F. Mouton, M. M. Malan, L. Leenen, and H. Venter, “Social engineering attack framework,” in *Information Security for South Africa*, Johannesburg, South Africa, Aug 2014, pp. 1–9.

[12] F. Mouton, L. Leenen, and H. Venter, “Social engineering attack examples, templates and scenarios,” *Computers & Security*, vol. 59, pp. 186 – 209, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404816300268

[13] M. Bezuidenhout, F. Mouton, and H. Venter, “Social engineering attack detection model: Seadm,” in *Information Security for South Africa*, Johannesburg, South Africa, August 2010, pp. 1–8.

[14] F. Mouton, L. Leenen, and H. S. Venter, “Social engineering attack detection model: Seadmv2,” in *International Conference on Cyberworlds (CW)*, Visby, Sweden, October 2015, pp. 216–223.

[15] F. Mouton, M. Malan, and H. Venter, “Development of cognitive functioning psychological measures for the seadm,” in *Human Aspects of Information Security & Assurance*, Crete, Greece, June 2012, pp. 40–51.

[16] D. Gragg, “A multi-level defense against social engineering,” SANS Institute InfoSec Reading Room, Tech. Rep., December 2002.

[17] R. Bhakta and I. Harris, “Semantic analysis of dialogs to detect social engineering attacks,” in *Semantic Computing (ICSC), 2015 IEEE International Conference on*, Feb 2015, pp. 424–427.

[18] S. S. Epp, *Discrete Mathematics with Applications*, 4th ed. Brooks/Cole Publishing Co., 2010.

[19] F. Mouton, M. M. Malan, and H. S. Venter, “Social engineering from a normative ethics perspective,” in *Information Security for South Africa*, Johannesburg, South Africa, August 2013, pp. 1–8.

[20] L. Janczewski and L. Fu, "Social engineering-based attacks: Model and new zealand perspective," in *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on*, Oct 2010, pp. 847–853.

[21] T. Dimkov, A. van Cleeff, W. Pieters, and P. Hartel, "Two methodologies for physical penetration testing using social engineering," in *Proceedings of the 26th Annual Computer Security Applications Conference*, ser. ACSAC '10. New York, NY, USA: ACM, 2010, pp. 399–408. [Online]. Available: http://doi.acm.org/10.1145/1920261.1920319

[22] M. Workman, "A test of interventions for security threats from social engineering," *Information Management & Computer Security*, vol. 16, no. 5, pp. 463–483, 2008.

[23] S. Esmail, "eps1.5_br4ve-trave1er.asf," June 2015, mr. Robot: Season 1, Episode 6. [Online]. Available: http://www.usanetwork.com/mrrobot/episode-guide/season-1-episode-6-eps15br4ve-trave1erasf

[24] M. Jodeit and M. Johns, "Usb device drivers: A stepping stone into your kernel," in *Computer Network Defense (EC2ND), 2010 European Conference on*, Oct 2010, pp. 46–52.

[25] F. Mouton, M. Teixeira, and T. Meyer, "Benchmarking a mobile implementation of the social engineering prevention training tool," in *Information Security for South Africa*, Johannesburg, South Africa, Aug 2017, pp. 106–116.

NOTES

SAIEE AFRICA RESEARCH JOURNAL – NOTES FOR AUTHORS

This journal publishes research, survey and expository contributions in the field of electrical, electronics, computer, information and communications engineering. Articles may be of a theoretical or applied nature, must be novel and must not have been published elsewhere.

Nature of Articles

Two types of articles may be submitted:

- Papers: Presentation of significant research and development and/or novel applications in electrical, electronic, computer, information or communications engineering.
- Research and Development Notes: Brief technical contributions, technical comments on published papers or on electrical engineering topics.

All contributions are reviewed with the aid of appropriate reviewers. A slightly simplified review procedure is used in the case of Research and Development Notes, to minimize publication delays. No maximum length for a paper is prescribed. However, authors should keep in mind that a significant factor in the review of the manuscript will be its length relative to its content and clarity of writing. Membership of the SAIEE is not required.

Process for initial submission of manuscript

Preferred submission is by electronic upload in electronic MS Word and PDF formats. PDF format files should be 'press optimised' and include all embedded fonts, diagrams etc. All diagrams to be in black and white (not colour).

The Managing Editor, SAIEE Africa Research Journal, PO Box 751253, Gardenview 2047, South Africa

Author guidelines and upload link available at http://goo.gl/8OcZVe

E-mail: researchjournal@saiee.org.za

These submissions will be used in the review process. Receipt will be acknowledged by the Editor-in-Chief and subsequently by the assigned Specialist Editor, who will further handle the paper and all correspondence pertaining to it. Once accepted for publication, you will be notified of acceptance and of any alterations necessary. You will then be requested to prepare and submit the final script. The initial paper should be structured as follows:

- TITLE in capitals, not underlined.
- Author name(s): First name(s) or initials, surname (without academic title or preposition 'by')
- Abstract, in single spacing, not exceeding 20 lines.
- List of references (references to published literature should be cited in the text using Arabic numerals in square brackets and arranged in numerical order in the List of References).
- Author(s) affiliation and postal address(es), and email address(es).
- Footnotes, if unavoidable, should be typed in single spacing.
- Authors must refer to the website: http: //www.saiee.org.za/arj where detailed guidelines, including templates, are provided.

Format of the final manuscript

The final manuscript will be produced in a 'direct to plate' process. The assigned Specialist Editor will provide you with instructions for preparation of the final manuscript and required format, to be submitted directly to:
The Managing Editor, SAIEE Africa Research Journal, PO Box 751253, Gardenview 2047, South Africa.
E-mail: researchjournal@saiee.org.za

Page charges

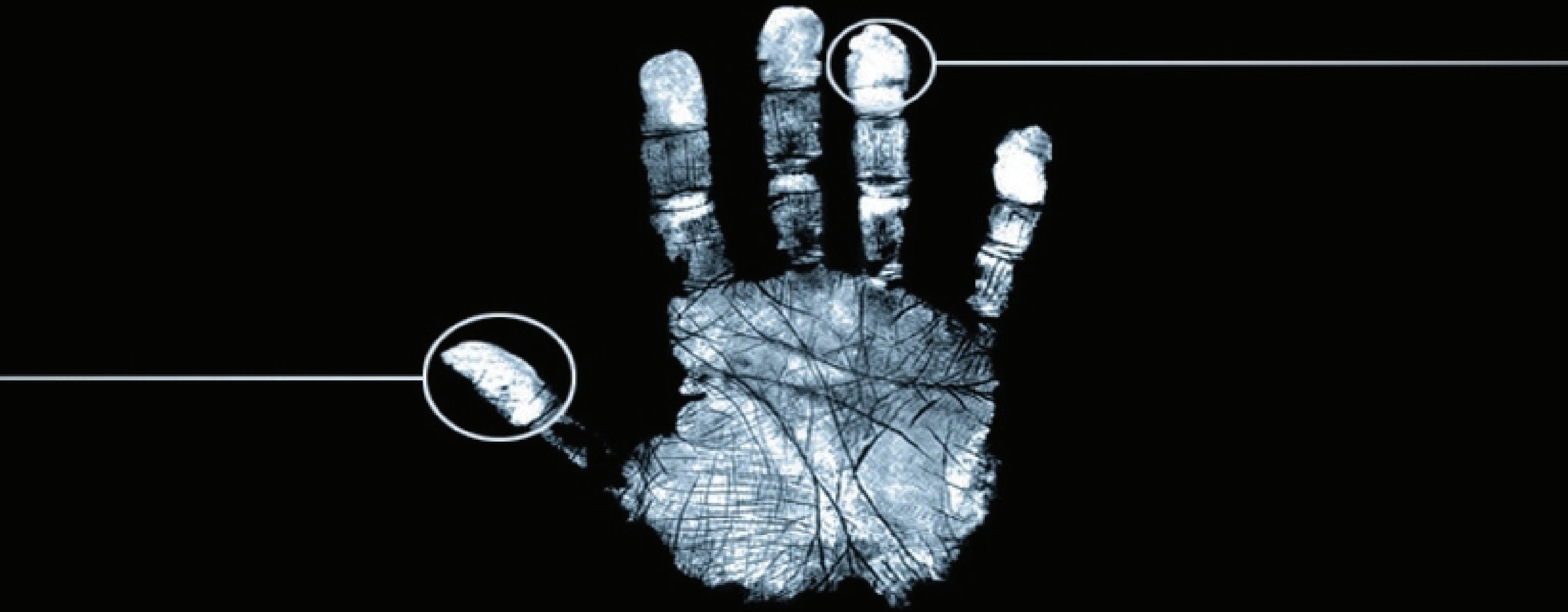
A page charge of R200 per page will be charged to offset some of the expenses incurred in publishing the work. Detailed instructions will be sent to you once your manuscript has been accepted for publication.

Additional copies

An additional copy of the issue in which articles appear, will be provided free of charge to authors.
If the page charge is honoured the authors will also receive 10 free reprints without covers.

Copyright

Unless otherwise stated on the first page of a published paper, copyright in all contributions accepted for publication is vested in the SAIEE, from whom permission should be obtained for the publication of whole or part of such material.



SAIEE

South African Institute for Electrical Engineers (SAIEE)
PO Box 751253, Gardenview, 2047, South Africa
Tel: 27 11 487 3003
E-mail: researchjournal@saiee.org.za | Website: www.saiee.org.za