Department of Computer Science

Submitted in part fulfilment for the degree of
MSc in Cybersecurity.

Investigating the Security of p≡p 's Trustword PGP Fingerprint Encoding

Aidan Fray

DRAFT PROCESSED 27th July 2019

Supervisor: Siamak Fayyaz Shahandashti

Acknowledgements

I would like to thank my goldfish for all the help it gave me writing this document.

As usual, my boss was an inspiring source of sagacious advice.

Contents

List of Figures

List of Tables

1 Introduction

The increasing use of public key cryptography by instant messaging and secure email means key fingerprint verification is an ever more important task. One of the most significant risks to the security of the communication channel is a Man-in-the-middle (MiTM) attack. A successful MiTM attack can circumvent the encryption as it allows an attacker to read all the encrypted data. A countermeasure for this is the verification of each parties' fingerprint. **[...]**

2 Literature Review

Fingerprints are used to compare the similarity of two keys quickly. The comparison occurs between two digests of a secure one-way hash function. Historically these have been represented as a hexadecimal string whereon verification fingerprints are compared between two substrates, for example, a monitor screen and a business card. This process is known as the *“authentication ceremony.”*

Prior research has shown that the average human can only hold around 7-digits worth of data in their working memory[1]. This rules out the possibility of comparing complete digests. For example, SHA-1 is 40 hex digits (160-bit) and, therefore, difficult to effectively compare. Therefore, if human interaction is required; there is a need for schemes that work effectively with consideration to individual limitations.

Research in this area has taken various encoding schemes and compared their fallibility to impersonated attacks. The main elements of comparison have been the “accuracy of attack detection" and “time to compare". In this paper, these are considered the metrics of “security” and “usability”, respectively.

Definition of terms

Throughout the paper, various terms are used to define elements of the authentication ceremony relevant to fingerprints verification. Explicit descriptions are, therefore, stated in this section. These are used to remove ambiguity and underlying associations readers may have for the words chosen.

Encoding schemes - This is the physical method of encoding used to represent the fingerprint. For example, “Hexadecimal” or “Words” are a unique way to represent a fingerprint and thus, are encoding schemes.

Method of comparison - This is how the user assesses the encoding scheme. The most common example and the one used as default is “Compare-and-Confirm” (CaC). CaC, as the name suggests, is the *comparison* of two fingerprints on different devices or mediums that are then *confirmed*. The following sections explain alternative methods of comparisons.

2.0.1 Authentication ceremony performance

Encoding scheme performance

Results from the literature consistently show the effectiveness of language-based encodings such as Words or Sentences with accuracies ranging from 94% [2][3][4]. In all cases, these were the best schemes from the sets assessed. The exception to this is the work performed by **Hsiao, *et al.***[5] in 2009 with Words achieving an abnormal accuracy of 63.00%.

Aside from textual representations were graphical schemes. Examples of schemes assessed were: Random Art[6], Flag[7], T-Flag[8], Vash[1], OpenSSH visual host key and Unicorns[2], among others. These schemes had mixed accuracy with ranges as large as 50% - 94% in work by **Hsiao, *et al.***[5]. The only other paper assessing graphical representations was the work of **Tan *et al.***[3] where they also achieved mixed results with accuracies ranging from 46% to 90%.

Scheme	Hsu-Chun[5]	Kainda[4]	Dechand[2]	Tan[3]
Hexadecimal			11.20	9.00
Numerical		6.00	10.60	9.00
Base32	3.51	6.00	10.20	
Words	4.63	7.00	8.70	7.00
Scentences		11.00	12.30	8.00
Chinese Symbols	5.01			
Japanese Symbols	5.07			
Korean Symbols	4.92			
Random Art	3.21			
Flag	4.28			
T-Flag	4.00			
Flag Ext.	4.02			
OpenSSH‘				5.00
Unicorns				3.00
Vash				3.00

Table 2.1: Timing results in seconds for the related schemes

In terms of usability of graphical schemes, the literature concurred on their

[1] https://github.com/thevash/vash

[2] https://unicornify.pictures/

high usability. The comparison speed of these schemes were all among the quickest (See Table 2.1 for an overview of timings). Other work also indirectly concurred with graphical encodings having significantly quicker comparison times compared to non-graphical schemes [2][4]. In terms of research into the performance of graphical schemes, the literature does not contain an extensive review with literature only containing two papers. There is also no overlap in the schemes assessed with each paper reviewing a unique set. This is, therefore, a promising candidate for further research.

One unique paper in this research area was the work by **M. Shirvanian *et al.***[9] produced in the context of secure messaging pairing. This paper was unique for several reasons. First was to consideration for “remote-vs-proximity” pairing where this is the first consideration of this aspect found in the literature. There is room for further research to compare encoding schemes in the context of “remote-vs-proximity.” Another unique aspect was the end-to-end encryption context of the study.

The findings from the paper showed a high false negative rate for all the schemes; this is a consideration also missing from the literature. Alongside this, results for usability were lower in a remote setting for all the schemes. The author, however, comments on the expected nature of this result. Images were assessed as being the most secure method of authentication in the remote setting but voted as the method with the worst usability. This conclusion is highly inconsistent with all other work in this area. However, this could be due to the unique setting of remote verification, resulting in distinct results from user studies. Without further work in this area, it is difficult to validate these results conclusively.

Alongside this, it was shown in work by **Hsiao, *et al.***[5] that age and gender do not affect the accuracy of the scheme. However, younger participants were considerably faster. Furthermore, findings also showed that language comprehension helped in discerning small differences between schemes encoded in Chinese, Japanese, Korean or English. Subsequently, knowledge of the language did not assist in differentiating more significant changes in the schemes as these had high accuracy regardless. These were interesting and unique considerations. Further work could aim to corroborate these conclusions.

Tables 2.1 & 2.2 contain the accuracy results of all papers assessed; this is to aid in visual comparison. Each paper used a different metric for measuring accuracy. Therefore, all results have been translated into “overall accuracy.”

Scheme	Hsu-Chun[5]	Kainda[4]	Dechand[2]	Tan[3]	M. Shirvanian[9]
Hexadecimal			89.56%	79.00%	
Numerical		100.00%	93.66%	65.00%	97.33%
Base32	86.00%	90.00%	91.50%		
Words	63.00%	100.00%	94.25%	94.00%	
Scentences		100.00%	97.01%	94.00%	
Chinese Symbols	59.00%				
Japanese Symbols	57.00%				
Korean Symbols	54.00%				
Random Art	94.00%				
Flag	50.00%				
T-Flag	85.00%				
Flag Ext.	88.00%				
OpenSSH‘				90.00%	
Unicorns				46.00%	
Vash				88.00%	

Table 2.2: Accuracy of correct comparison for the encoding schemes assessed

Methods of comparison performance

Aside from “Compare-and-Confirm” (CaC), there is “Compare-and-Select” (CaS) and “Compare-and-Enter” (CaE). CaS is the method where one device displays the fingerprint, and the other user is provided with several options. The user then has to choose the correct value from the list of candidates. If there is no match, the user must deny the connection attempt. The creation of CaS was due to concerns that CaC would be “too easy” for users leading to complacency and errors[10]. CaE is designed for scenarios where both devices might not have a display, i.e. pairing between a phone and a keyboard. One device displays the checksum. This checksum is entered into the other device. The first device then compares the entered string and checks for a match.

Research has been performed assessing the performance of these schemes and how they affect the ultimate security of the authentication ceremony. The literature agrees on CaC being the best overall scheme to compare fingerprints [3][10] with CaS being highlighted for its poor security and usability. CaE has had contrasting results. [10] discarded it after one round due to “poor usability.” However, [3] considered it the best method overall for usability and security. These conflicting results, therefore, show polarisation in the results of CaE. However, this could be due to the different overall use-cases of the studies. Validation of results from either study would, therefore, be an area of additional study.

Experimentation methodology comparison

To further look into the validity of previously discussed results, it is necessary to asses how the respective studies reached these conclusions. Areas for consideration are scheme entropy, attacker strength and participant attributes.

The starkest limitations of the literature are the range of participants and encoding entropy. The worst studies tested only 22-bits of entropy. This makes it difficult to compare results directly. One of the papers this most effects is the early work by **Hsiao, *et al.***[5] where their highest entropy is 28-bits. This level of entropy was inadequate even at the time of publication. There is an attempt by the authors' to address this issue in the later stages of the paper where they write *"[...] increasing entropy is not a solution because it sacrifices usability and accuracy. With more entropy, representations will contain longer sequences of characters or more minute details which will lead to increased time and errors during comparisons."*. This statement is backed up with no empirical evidence, and the authors' fail to consider how the low entropy would affect the overall security of the schemes.

Attacker strength is also another metric used to compare results concluded by each paper. Some papers failed to address their attacker strength consideration directly, but the overall strength has been inferred from the changes made to their schemes. For example, if they decided to change a single character in a 40-digit (160-bit) SHA-1 hex digest, they are indirectly stating that the attacker can control 39-digits (156-bit). To achieve this, the attacker would have to compute 2^{156} SHA-1 compressions to find a key match. In the literature, this element ranged from 2^{28} to around 2^{242}; however, this has some relation to the size of the encoding schemes used. This is a substantial range that makes it ultimately challenging to compare and confer results confidently. Further work could exclusively look into the affects attacker strength has on the success of attacks. Moreover, all the papers assessed failed to fully consider the feasibility of attacks in terms of computer and storage requirements. This again, shows gaps in the literature.

All of the studies considered demographical data when presenting their results. Their average ages were all around ~ 35 years old with the majority of participants educated with at least a bachelor degree. This was alongside the equal split between male and female participants.

One consideration of note is that made by Dechand *et al.* [2] where they briefly consider medical conditions such as ADHD and reading or visual disorders and the way they affect the comparison's effectiveness. They highlight a slight reduction in overall accuracy, although due to their small sample size, they cannot conclusively validate these results. This is an aspect unique to all literature. Further work would be required to produce conclusive results. Therefore, highlighting a gap in the literature.

One glaring issue with the demographical health of **Ersin Uzun *et al.***[10] study is the use of two entirely different groups of participants. Not only were their demographics different, but they were from different countries (America and Finland). Different cultures contain inherent biases and assumptions. Moreover, more concerns are raised around the dual study design of this paper with two completely different set of participants. Changes were made pragmatically regarding the results from the first round of 40 participants. These were then re-assessed and the results were directly compared. This puts huge doubts on the validity of the results with no consideration made by the authors to control external factors that may affect the performance of the method of comparison.

Hexadecimal and numerical schemes were not included as encoding schemes in work by **Hsiao, *et al.***[5] (some of the most widespread encoding schemes). This decision was based on their claims on similarities to Base32 and “well-known deficiencies" in the excluded schemes. This point was provided with no further justification or quantified in any way. It is also not consistent with available research, for example, in *“Empirical Study of Textual Key-Fingerprint Representations”*[2] it was shown numerical representations performed significantly better than that of Base32.

	Hsu-Chun[5]	Kainda[4]	Dechand[2]	Tan[3]	M. Shirvanian[9]
Attacker Strength[3]	$\sim 2^{28}$	$\sim 2^{40}$	2^{80}	2^{60}	$\sim 2^{242}$
Entropy Range	22-28 bits	20-40 bits	122 bits	128 bits	160-256 bits
No° Participants	436	30	1001	661	25

Table 2.3: Paper attribute comparison

Table 2.3 has been provided to visually compare the differing aspects of the papers’ parameters. Clearly it can be seen from the table the large ranges in participant size, entropy and attacker strength.

Topic conclusion

Overall, this review has identified several key areas suitable for further work. The first is the performance assessment of graphical encoding schemes. Recreation of pre-existing results from [5][3] is required to corroborate current conclusions and validate results. Another gap in the research is the consideration into the utilization of encoding schemes in realistic conditions, i.e. “remove vs. proximity." This topic was initially covered by [9] but their scope was limited. Further work, therefore, could increased the scope and touch upon a large number of schemes in these settings. The final aspect for further work is the limited consideration into the feasibility of attacks on encoding schemes. All of the papers assessed simulated attacks and had minimal consideration for the execution of these attacks. Therefore, further work could delve into the implementation of such attacks and their feasibility in terms of compute and storage complexity.

2.0.2 Encoding schemes

Another area of research is investigations into the actual physical encodings of the hash digest. This section will briefly discuss the current research available on the creation and security of actual encoding schemes. The actual details of the operation of the schemes are outside the scope of this literature review, therefore, minimal attention will be allocated to these details.

Some of the oldest preliminary work into visual encoding schemes was performed by **Adrian Perrig *et al***[6]. in the creation of their scheme "Random Art" in 1999. The motivation for creating such a scheme was the perceived flaws in the ways humans verify and compare written information. As mentioned in previous sections visual encoding schemes have been shown to have mixed success, with low security being one of their most alarming flaws. This research laid the foundation for further work in analysing the security of visual encoding schemes.

Further research into the creation of unique visual hash schemes have been performed by **C Ellison *et al.*** [7] (Flag), **Yue-Hsun Lin *et al.***[8] (T-Flag) and work by **M. Olembo *et al.***[11]. Each publication has provided a new way to visually represent a key fingerprint. Alongside the academic literature, there are more informally presented methods of visual fingerprints such as Unicorns[4] and Robots[5]. This list is by no means exhaustive but is used to depict the amount of research and work invested into graphical hash representations.

One paper of note is the preliminary work performed by **D Loss *et al.***[12] in their *"An analysis of the OpenSSH fingerprint visualization algorithm"* where their aim was to spur on further research with their initial findings into the security of the OpenSSH scheme. The authors claim that the use of the algorithm in OpenSSH is only heuristically defined and there is a need for a formal proof of its security.

The paper proposed a number of ways to generate similar fingerprints. The methods proposed were: Naive brute force, Graph Theory, and brute force of a full visual set. They were only able to produce only very basic results and have proposed a large amount of potential further work. Since the

[4] https://unicornify.pictures/

[5] https://github.com/e1ven/Robohash

paper's publication in 2009, there seems to have been no research building on the work of the authors. This highlights a current gap in the available literature.

Minimal research has also focused on basic textual fingerprint representations and their respective security. Work by **A. Karole** and **N. Saxena**[13] looked into ways to improve the security of a textual representation. This research aim was to improve the secure device pairing process of comparing two numerical values. The devices used (Nokia 6030b; Mid-range devices at the time of publication) and the SAS compared results in findings that are not directly applicable in a fingerprint comparison context.

A more specific subsection of textual fingerprints is the use of words and sentences to encode hash digests. Some of the first work in this area was produced by **Juola** and **Zimmermann** [14] and their work in generating a word list where phonetic distinctiveness was prioritised. Each word is mapped to a single byte. The unique aspect of the word list is the separation of "even" and "odd" words where "even" byte positions are sample from the even-list and "odd" from the odd-list. This effectively creates two subword lists. The maximisation of linguistic distinctiveness of these word lists were maximised through the use of a Genetic algorithm. The paper also includes a study on effective measures of "linguistic distances" of words and provided an in-depth discussion into these areas.

Overall the paper provides a foundation for formalising the creation of effective wordlists. A limitation is the lack of empirical data gathered on the performance. However, this was later evaluated in work by Dechand *et al.* [2] and shown to be an effective encoding scheme.

Other research of note is work by **M. Goodrich *et al.***[15] called *Loud and Clear: Human-Verifiable Authentication Based on Audio*. As the name suggests the authors were researching ways to improve current methods of secure device pairing. The unique aspect of this work is the use of a Text-to-Speech system reading out syntactically correct English sentences. The sentences are based on a MadLibs[6] where static placeholders were replaced with potential words.
The work into a potential wordlist can be seen as an extension to the work performed by Juola and Zimmermann[14] as they aimed to emulate the techniques used in PGPfone. The paper's finding are limited by the lack of empirical data backing up claims made by the author as the systems performance and security are only theoretically assessed.

Aside from this research, there have been further informal implementations of fingerprint encodings. The first being by **Michael Rogers**[7]. Rogers' implementation is a program designed to map fingerprints to pseudo-

[6] https://en.wikipedia.org/wiki/Mad_Libs
[7] https://github.com/akwizgran/basic-english

random poems. This implementation was again, empirically evaluated by Dechand *et al.*[2]. Older work by **N. Haller** with the S/KEY[16] shows the implementation of a system designed to represent a hash as a series of six short words. However, this system is designed for a one-time-password purpose and only provides word mappings for basic human usability of the password and not within a fingerprint verification context. Therefore, the wordlist has not been designed with pronounceability in mind.
A very recent implementation of a word list can be found in Pretty Easy Privacy (pEp) implementation of TrustWords[8]. The unique aspect of TrustWords is its mapping of a single word to 16-bits. In comparison to other literature, this is the highest number of bits-per-word seen. Full mappings (no duplication of words) would, therefore, require 2^{16} words in the dictionary and arguably is higher than most users vocabulary. this deviation from the norm has not been currently backed up by research. Moreover the main RFC documentation still remains in a draft stage and states *"It is for further study, what minimal number of words (or entropy) should be required."*. These aspects clearly highlight on a gap in the current literature.

Topic conclusion

In conclusion to this topic, the current research has primarily focused on the research and creation of visual representations. Research for textual fingerprints is fragmented and incomplete with work Juola and Zimmermann [14] and M. Goodrich *et al.*[15] providing meaningful research to build upon in terms of word a sentence based encodings. The fragmentation of this research leaves room for further work into this topic area. Alongside this, findings from the previous sections research shows that human language based encodings provided the best usability and, therefore, should be a target for further research looking to improve upon their security and usability.

2.0.3 Attacks on encoding schemes

This area of research studies ways to physically execute attacks on fingerprint encoding schemes. This differs from previously examined work due to papers discussing the performance and fallibility of encoding schemes simulated the attack without consideration for how the attack would be performed. Research in this area is scant, with lots of research attention being directed towards the security of Man-in-the-Middle (MITM) attacks and not the encoding schemes themselves.

Research in 2002 by **Konrad Rieck**[17] is the first formalisation of at-

[8] https://tools.ietf.org/html/draft-birk-pep-trustwords-03

tacks on fingerprint representations. The paper titled *"Fuzzy Fingerprints Attacking Vulnerabilities in the Human Brain"* aimed to look into ways users check hexadecimal encoded OpenSSH fingerprint representations. The author created an elegant way to 'weight' more important chunks of the digest. The bytes furthest to the right and left of the digests provided the highest weight. The weight was the smallest in the centre of the digest. This provides a way to score digests and determine the best partial collisions found. For example with the target fingerprint: `9F23` a partial match `9313` is given a score of 45% even though only two characters were matching. This is due to the weightings.

The paper contains an implementation with a "1.2GHz CPU" being able to obtain 130,000 H/s (With MD5). In comparison to this, a mid-range Intel i5-3320M CPU can today obtain 111,700,000 MD5 H/s. This shows that the results obtained from the paper are significantly outdated. However, even with the low hash rate, the author was able to obtain some promising results. Figure 2.1 contains the best example used.

```
TARGET: d6:b7:df:31:aa:55:d2:56:9b:32:71:61:24:08:44:87
MATCH:  d6:b7:8f:a6:fa:21:0c:0d:7d:0a:fb:9d:30:90:4a:87
```

Figure 2.1: Best match obtained after a few minutes of hashing

Overall the paper shows an interesting way to create partial fingerprint matches but is not quantified by any empirical evidence gathered on real world users. This, therefore, highlights on gaps in the coverage of this literature.

The only other relevant research on this topic is the work by **M Shirvanian *et al.***[18] and their paper *"Wiretapping via Mimicry: Short Voice Imitation Man-in-the-Middle Attacks on Crypto Phones"*. Further research in the area of "human voice impersonation" has received lots of attention [19][20][21]. This paper was chosen over other alternatives due to is specific use of encoding schemes in its evaluation.

In this paper, the authors develop a way to impersonate users when authenticating Short-authentication-Strings (SAS) in pairing of Crypto-phones. To achieve this impersonating they propose two methods: "Short voice reordering attack" where an arbitrary SAS string is recreated by re-ordering snipped obtained from eavesdropping a previous connection and "Short voice morphing attacks" whereby the use of previously eavesdropped audio snippets the attacker can morph their own voice to match that of the victim. With these methods, they aimed to attack encodings of Numbers, PGP word list (previously discussed work by Juola and Zimmermann [14]) and MadLib (M. Goodrich *et al.*[15] work also previously discussed). The effectiveness of these attacks were evaluated with a study involving 30 participants.

Results from the paper show the effectiveness of these methods. Compared to the baseline of the attacker's voice replacing the victim where this performed with a $\sim$18% success rate. Morphing gained an overall success rate of 50.58% and Reordering a very impressive 78.23% success rate. Showing that these attacks provide an improvement on top of the naive implementation.

One of the biggest limitations addressed by the authors was the reduction in success rates as the size of the authentication string grew. The morphing and reordering attacks become increasingly ineffective as the user has more time to detect imperfections. This is not quantified by the author and the extent of this degradation is never empirically discussed. Therefore, the results from this study are only effective and applicable in a SAS context.

Topic Conclusion

Overall the literature for this subtopic remains sparse and incomplete. Further suggested work could look into the feasibility of generating partial collisions for all textual representations alongside quantified effectiveness on users. With the possibility to concentrate on a few selected implementations. The work would aim to focus on the various physical methods used and their feasibility. This is one area the previous literature has failed to cover and has only theoretically quantified attacker strength without consideration for the actual real-world cost of these attacks.

2.1 Overall Summary

TODO: Create an overall summary of all the gaps identified

2.2 Research Questions

TODO: - Backup choice of questions up using my previous discussion.

3 Background

3.1 Problem Context

The project aims to asses the security of p≡p 's fingerprint representation "Trustwords". Trustwords aim to sacrifice security for increased usability. The encoding scheme has been designed to assist this by having the user compare a reduced set of words by about half in comparison to other schemes. To achieve this a larger number of bits are required per word. In this case Trustwords maps 16bits per word providing 65537 different combinations. This is, however, the largest number of bits per words (As show in the Literature Review in Chapter 2) and pushes the boundaries of a language's usable vocabulary.

This abnormal design choice has not received a security assessment to back up claims. This paper, therefore, aims to rectify this research gap.

However, in order to fully understand the project goal it is important to understand public-key cryptography, the role and functioning of p≡p and the role fingerprint comparison has in preventing Man-in-the-middle attacks.

TODO: Maybe research questions here instead of in the literature review

3.2 Public-key Cryptography Key Exchange

Asymmetric cryptography facilitates the secure encryption of messages in end-to-end encryption (E2EE), verification of digital signatures and sharing of pre-communication secrets, among others. The use-case of E2EE messages will be the primary focus of this paper.

The asymmetry stems from the use of "Public" and "Private" keys. The Public key is used to encrypt data that only the respective Private key can decrypt. This means the keys required to communicate can be easily shared across insecure channels.

To construct an E2EE connection the initial stage will be a key exchange using the key types discussed previously. This is the sharing of a pre-shared secret between two verified parties, this will then be used to encrypt subsequent messages with symmetric encryption[1]. This, however, hinges on the verification of the initial parties, if one party impersonates another and gets to view the pre-shared secret all communication becomes decryptable. Therefore, the correct identification of parties is crucial to maintaining secure communication channels. The exploitation of this is known as a Man-in-the-middle (MiTM) attack. This can be bidirectional where the attacker can sit in the middle of a communication channel and decrypt all correspondence. Figure 3.1 contains a visual representation of this attack.

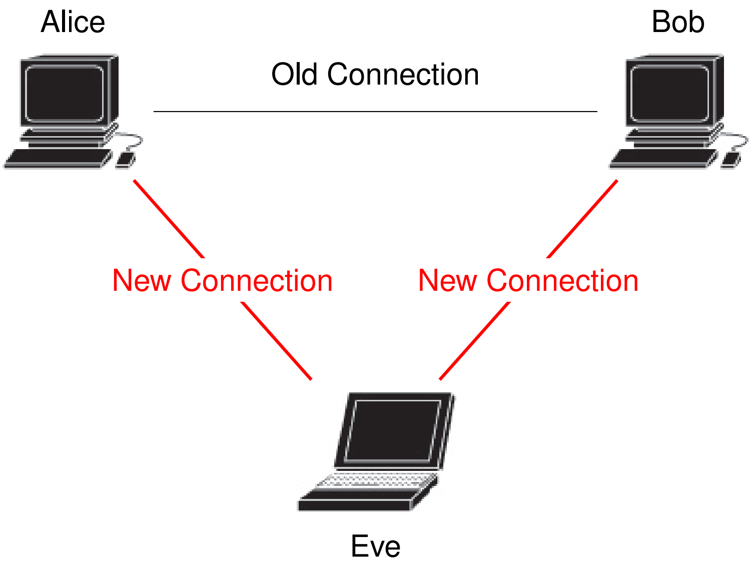


Figure 3.1: Photo depicting a MITM attack

Due to the assumption that the attacking party (Eve) does not have the same public-private key pair as either party (Alice or Bob) the fingerprint of the public key can be used for identification. There are, however, issues in how a user should link a key's fingerprint to real world entity. This paper will, however, assume that users know the respective parties' correct fingerprint, as the protocols used to achieve this are outside the project's scope.

The fingerprint of the key is generated by running the main components of the key through a secure one way hash function[2]. This process produces a digest of fixed length that can be used to compare keys. Therefore, the comparison of expected and actual fingerprints is used to detect MiTM attacks and prevent impersonation.

[1] This is due to the speed increase of symmetrical over asymmetrical encryption

[2] Explain this?

3.3 PGP

Pretty Good Privacy (PGP) is a specific implementation of public key cryptography discussed in the previous section. Its main use-case is for securing message based communications but can also be used for file or hard drive encryption.

Each element of a PGP key is split into what is know as a 'packet'. The 'fingerprint' packet v4 contains a version number, timestamp and the main elements of the keys algorithm (i.e. RSA exponent and modulus) formatted as MPIs[3]. This data is then preceded with PGP packet header (used to specify length and version numbers). This is all then utilized as input into the one-way hash function SHA-1 to produce a 160-bit digest.

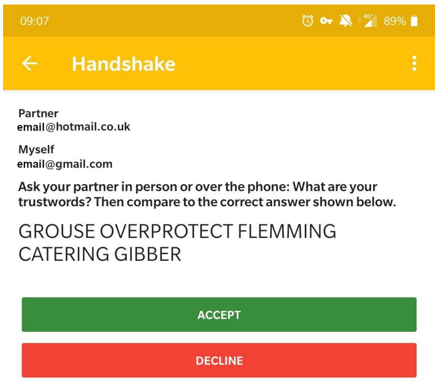
TODO: Generate an image of PGP v4 fingerprint packet

[3] Multiprecision integers are unsigned integers used to hold large integers such as the ones used in cryptographic calculations.

3.4 Pretty Easy Privacy

Pretty Easy Privacy (p≡p) is a data encryption system that utilises PGP encryption to provide E2EE on all common channels of communication such as email of SMS. p≡p 's design principles state that above all the systems should be easy to install, use and understand.

p≡p deals with the threat of MiTM attacks by having users compare the respective key fingerprints encoded as a set of words. Figure 3.2 shows the p≡p Android implementation of trustwords. The user pair of users then authenticate the words on an OOB (Out of Band) channel such as a phone call or in-person communication. If both users decide the words match they will accept or decline respectively.



Partner
email@hotmail.co.uk

Myself
email@gmail.com

Ask your partner in person or over the phone: What are your trustwords? Then compare to the correct answer shown below.

GROUSE OVERPROTECT FLEMMING CATERING GIBBER

Figure 3.2: Trustword fingerprint verification

The trustwords are unique for each pair of keys, this is the alternative of having a static key for each user. The justification behind the unique set for each user pair is in an attempt to motivate users to perform the authentication ceremony. The unique combined fingerprint is believed to reduced complacency due to inherent change **[CITE ME]**.

Each key pair is the exclusive-or of each key's fingerprint, this combined values is then broken down into 16-bit chunks and mapped to words via a pre-defined dictionary.

$$KeyFingerprint_1 \oplus KeyFingerprint_2 = TrustwordsFingerprint$$

Figure 3.3: Creation of the combined Trustword fingerprint

Due to the number of words required (2^{16}) alongside the design choice to exclude slang and profanities the english Trustword dictionary requires dual-mapping of a section of words. Approximately 13633/65536 (20.8%) of words are re-mapped in the dictionary leaving 51903 unique words. The re-mapping is also done on a loop with it remaining alphabetical. Figure 3.4 shows the position in the dictionary where this occurs. This predictability within the dictionary will be explored later in the paper.

```
     [...]
52127 ZYGOTE
52128 ZYGOTIC
52129 ZYMURGY
52130 AACHEN
52131 AARDVARK
52132 AAREN
     [...]
```

Figure 3.4: Re-mapping position in Trustword dictionary

3.5 OpenCL

3.6 Similarity metrics (Soundex etc)

4 Design

4.1 Design of GreenOnion

And the way it works. Need to explain it is based on Scallion but improved with the use of C++ alongside a bloom filter.

5 Experiments

5.1 Scallion vs GreenOnion

5.2 Experiment 1

5.3 Experiment 2

6 Conclusion

A Encoding Scheme Results

Bibliography

[1] G. A. Miller, ‘The magical number seven, plus or minus two: Some limits on our capacity for processing information.’, *Psychological review*, vol. 63, no. 2, p. 81, 1956.

[2] S. Dechand, D. Schürmann, K. Busse, Y. Acar, S. Fahl and M. Smith, ‘An empirical study of textual key-fingerprint representations’, in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 193–208.

[3] J. Tan, L. Bauer, J. Bonneau, L. F. Cranor, J. Thomas and B. Ur, ‘Can unicorns help users compare crypto key fingerprints?’, in *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, ACM, 2017, pp. 3787–3798.

[4] R. Kainda, I. Flechais and A. Roscoe, ‘Usability and security of out-of-band channels in secure device pairing protocols’, in *Proceedings of the 5th Symposium on Usable Privacy and Security*, ACM, 2009, p. 11.

[5] H.-C. Hsiao, Y.-H. Lin, A. Studer, C. Studer, K.-H. Wang, H. Kikuchi, A. Perrig, H.-M. Sun and B.-Y. Yang, ‘A study of user-friendly hash comparison schemes’, in *2009 Annual Computer Security Applications Conference*, IEEE, 2009, pp. 105–114.

[6] A. Perrig and D. Song, ‘Hash visualization: A new technique to improve real-world security’, in *International Workshop on Cryptographic Techniques and E-Commerce*, 1999, pp. 131–138.

[7] C. Ellison and S. Dohrmann, ‘Public-key support for group collaboration’, *ACM Transactions on Information and System Security (TISSEC)*, vol. 6, no. 4, pp. 547–565, 2003.

[8] Y.-H. Lin, A. Studer, Y.-H. Chen, H.-C. Hsiao, L.-H. Kuo, J. M. McCune, K.-H. Wang, M. Krohn, A. Perrig, B.-Y. Yang *et al.*, ‘Spate: Small-group pki-less authenticated trust establishment’, *IEEE Transactions on Mobile Computing*, vol. 9, no. 12, pp. 1666–1681, 2010.

[9] M. Shirvanian, N. Saxena and J. J. George, ‘On the pitfalls of end-to-end encrypted communications: A study of remote key-fingerprint verification’, in *Proceedings of the 33rd Annual Computer Security Applications Conference*, ACM, 2017, pp. 499–511.

[10] E. Uzun, K. Karvonen and N. Asokan, ‘Usability analysis of secure pairing methods’, in *International Conference on Financial Cryptography and Data Security*, Springer, 2007, pp. 307–324.

[11] M. M. Olembo, T. Kilian, S. Stockhardt, A. Hülsing and M. Volkamer, ‘Developing and testing a visual hash scheme.’, in *EISMC*, 2013, pp. 91–100.

[12] D. Loss, T. Limmer and A. von Gernler, *The drunken bishop: An analysis of the openssh fingerprint visualization algorithm*, 2009.

[13] A. Karole and N. Saxena, ‘Improving the robustness of wireless device pairing using hyphen-delimited numeric comparison’, in *2009 International Conference on Network-Based Information Systems*, IEEE, 2009, pp. 273–278.

[14] P. Juola, ‘Whole-word phonetic distances and the pgpfone alphabet’, in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96*, IEEE, vol. 1, 1996, pp. 98–101.

[15] M. T. Goodrich, M. Sirivianos, J. Solis, G. Tsudik and E. Uzun, ‘Loud and clear: Human-verifiable authentication based on audio’, in *26th IEEE International Conference on Distributed Computing Systems (ICDCS’06)*, IEEE, 2006, pp. 10–10.

[16] N. Haller, ‘The s/key one-time password system’, 1995.

[17] K. Rieck, ‘Fuzzy fingerprints attacking vulnerabilities in the human brain’, *Online publication at http://freeworld. thc. org/papers/ffp. pdf*, 2002.

[18] M. Shirvanian and N. Saxena, ‘Wiretapping via mimicry: Short voice imitation man-in-the-middle attacks on crypto phones’, in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2014, pp. 868–879.

[19] D. Mukhopadhyay, M. Shirvanian and N. Saxena, ‘All your voices are belong to us: Stealing voices to fool humans and machines’, in *European Symposium on Research in Computer Security*, Springer, 2015, pp. 599–621.

[20] S. Chen, K. Ren, S. Piao, C. Wang, Q. Wang, J. Weng, L. Su and A. Mohaisen, ‘You can hear but you cannot steal: Defending against voice impersonation attacks on smartphones’, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)*, IEEE, 2017, pp. 183–195.

[21] Z. Wu, N. Evans, T. Kinnunen, J. Yamagishi, F. Alegre and H. Li, ‘Spoofing and countermeasures for speaker verification: A survey’, *speech communication*, vol. 66, pp. 130–153, 2015.