Revisiting Context-Based Authentication in IoT

To appear at 55th Design Automation Conference (DAC), San Francisco, June 2018

Markus Miettinen, Thien Duc Nguyen,
Ahmad-Reza Sadeghi
Technische Universität Darmstadt
Mornewegstraße 30
64372, Darmstadt, Germany
{markus.miettinen,ducthien.nguyen,ahmad.sadeghi}@
trust.tu-darmstadt.de

N. Asokan
Aalto University
Konemiehentie 2
01520, Espoo, Finland
asokan@acm.org

ABSTRACT

The emergence of IoT poses new challenges towards solutions for authenticating numerous very heterogeneous IoT devices to their respective trust domains. Using passwords or pre-defined keys have drawbacks that limit their use in IoT scenarios. Recent works propose to use contextual information about ambient physical properties of devices' surroundings as a shared secret to mutually authenticate devices that are co-located, e.g., the same room. In this paper, we analyze these context-based authentication solutions with regard to their security and requirements on context quality. We quantify their achievable security based on empirical real-world data from context measurements in typical IoT environments.

ACM Reference format:
Markus Miettinen, Thien Duc Nguyen, Ahmad-Reza Sadeghi and N. Asokan. 2018. Revisiting Context-Based Authentication in IoT. In *Proceedings of Design Automation Conference, San Francisco, USA, June 2018 (DAC'18)*, 7 pages.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The emergence of the Internet of Things (IoT) is rapidly and drastically increasing the number of connected devices. Hence, there is an increasing need for reliable and usable solutions for provisioning security associations among devices belonging to the same trust domain (e.g., Smart Home, Smart Office, etc.). At the same time, state-of-the-art techniques can't provide adequate authentication solutions in such scenarios. Firstly, device pairing protocols like, e.g., Bluetooth pairing tend to quickly encounter usability limitations in settings with many devices, as it is tedious (and error-prone) to use a relatively laborious authentication process for every device separately. Secondly, solutions based on pre-shared keys or certificates can't be applied in practice due to the huge number of IoT device manufacturers that would need to set up a common key pool or PKI. Manufacturer-specific pre-shared keys also do not address the problem adequately, since it is not possible to use them to distinguish between devices belonging to different trust domains (e.g., Smart Home devices of different neighbors). Also, users may not want to rely on solutions based on centralized key management, due to privacy concerns.

As a solution for IoT device pairing scenarios, several previous works [7, 8, 10, 11] proposed to use common contextual features observed by co-located devices as a shared secret to enable them to authenticate their co-presence in the same contextual environment, e.g., in the same physical space like a room. The underlying assumption is that the ability to observe common contextual features like audio is spatially and temporally limited, either by mutual distance or environmental perimeters like walls. This can be utilized to distinguish between the devices to be paired and other devices. The related pairing can be either a one-time user-initiated process, or, performed implicitly by utilizing the sustained co-presence of devices in mutual proximity as a means to identify devices belonging to the same trust domain.

Goals and Contributions. In this paper, we revisit the schemes that have been proposed for context-based device pairing. We analyze their applicability to IoT scenarios and the security assurance that they provide. Concretely, we provide following contributions:

- A unified model of the use of context as a shared secret in authentication applications (Sect. 3),
- A security analysis of proposed schemes taking the entropy loss incurred by used error-correction schemes and privacy amplification into account (Sect. 4), and,
- An empirical evaluation of the security of context-based pairing based on real-world context data from environments relevant to IoT (Sect. 5).

2 CONTEXT-BASED PAIRING SCHEMES

2.1 System Model

Context-based pairing can be applied in situations in which two IoT devices A and B do not have a prior security association and want to establish one because they belong to the same *trust domain* $\mathcal{D}$. A trust domain denotes a set of devices that are intended to be able to communicate with one another and form collaborative (trusted) ensembles. Typically, devices owned by the same person or organization form such a trust domain. We also assume that there is *a priori* no key management infrastructure for authenticating the membership of devices A and B in the same trust domain $\mathcal{D}$.

In all context-based pairing approaches [8–11], A and B utilize measurements of physical features of their ambient surroundings

DAC'18, June 2018, San Francisco, USA
© 2017 Copyright held by the owner/author(s).
ACM ISBN 978-x-xxxx-xxxx-x/YY/MM.
https://doi.org/10.1145/nnnnnnn.nnnnnnn

observed with their on-board sensors for deriving a *context fingerprint w*. This fingerprint is subsequently used to establish a shared secret between the devices. These approaches are either based on demonstrative identification via proximity or implicit context-based authentication as we describe in the following.

2.2 Demonstrative Identification via Proximity

In these scenarios, pairing is a one-time operation during which the user *demonstratively identifies* [1] the devices to be paired by placing them close to each other. Usability considerations dictate that pairing completes within a few seconds as it is unacceptable for users to maintain A and B in close proximity for longer periods. This approach is amenable to mobile devices like smartphones that are relatively easy to place in any desired constellation. It requires active involvement of the user to explicitly initiate pairing and make sure that no other adversarial devices are within pairing distance d of either device A or B. Pairing can thus not be automated, as otherwise devices might pair with any devices sufficiently close to them. Especially in mobile scenarios, e.g., in crowded public transport systems this would lead user's devices to potentially establish pairings with devices of complete strangers just happening to stand nearby the user.

ProxiMate by Mathur *et al.* [8] is a scheme that uses fluctuations in a radio signal that A and B jointly observe to extract random secret bits to be used as a shared secret. Its security is based on the fact that these fluctuations are correlated between A and B only if they are located within half the wavelength λ of the used RF frequency of each other. Beyond this distance, no correlation exists.

The scheme by Schürmann and Sigg [10] extracts entropy from ambient audio and bases its security on the assumption that only if A and B are located close to each other they can observe similar audio environment. They extract context fingerprints by observing significant changes in the sound energy levels at different frequency bands in order to extract a maximum amount of entropy. In their approach, both A and B extract context fingerprints w and w', respectively, based on their context observations. A uses its fingerprint w to 'hide' a randomly selected secret s in a *fuzzy vault* [5] based on a Reed-Solomon error-correcting code. The check-in function of the fuzzy vault provides error-correcting information P, which A transmits to B. Using P and a fingerprint w' sufficiently similar to w, i.e., within Hamming distance $dist(w, w') \leq t$, B is able to retrieve secret s from the fuzzy vault. In a similar fashion, also the scheme by Mathur *et al.* uses an error-correcting Golay code to enable B to correct deviations between w' and w and subsequently use the corrected fingerprint as the shared secret between A and B.

2.3 Implicit Context-Based Authentication

A scheme utilizing implicit context-based authentication was first introduced by Miettinen *et al.* [9]. It allows establishing security associations between devices that are *permanently* located in the proximate context of each other. The underlying assumption is that all such devices belong to the same trust domain $\mathcal{D}$. In this approach A and B repeatedly monitor their context and iteratively execute a pairing protocol, which will succeed if the context observations of A and B are similar enough, e.g., if A and B are located in the same room, or fail otherwise. After a sufficient number of successful pairing iterations, A and B will accept the established pairing as authentic.

A challenge for the implicit context-based authentication scenario are devices not belonging to trust domain $\mathcal{D}$ that might be temporarily present in the context C (e.g., a visitor's smartphone). Therefore the implicit scheme requires *sustained* presence from devices by repeating authentication iterations over a prolonged period of time longer than the reasonable assumed duration of a visiting device's visit. This does, of course, not preclude A or B from granting *guest-level* access to the counterpart already after one or a few successful authentication iterations. However, full access to trust domain $\mathcal{D}$ would be granted only after a sufficient number of successful iterations.

3 ADVERSARY MODEL AND SECURITY GOALS

We consider the following adversary model. Given two legitimate IoT devices A and B belonging to the domain $\mathcal{D}$, the adversary $\mathcal{E}$ is a device that is not in the same proximate context C as A and B. Depending on the pairing scheme, *proximate context* may either denote close proximity in terms of physical distance d, or, the physical space that encloses both devices and is separated from the outside space by an enclosure like the walls of a room. In particular, we assume the adversary $\mathcal{E}$ to have following properties:

- It is equipped with the same contextual sensors as legitimate devices A and B.
- It can wirelessly communicate with both A and B in the same way as A and B with each other.

Impersonation. In an impersonation attack, adversary $\mathcal{E}$ that does not belong to the same trust domain $\mathcal{D}$ as A attempts to convince device A that it is a legitimate device $B \in \mathcal{D}$ and establish a successful pairing with it. This can happen if $\mathcal{E}$ can fabricate context observations that are similar enough to those of A that it will lead to successful authentication.

Man-in-the-Middle. If $\mathcal{E}$ can successfully execute the impersonation attack simultaneously with both A and B, it will gain the ability to perform man-in-the-middle attacks against A and B, i.e., completely controlling the communications between them.

In the schemes presented above, the context measurements of A and B are used to derive a shared secret s to be used in two alternative ways: either as an *authentication token*, or, directly as a *cryptographic key*. From the point of view of the adversary $\mathcal{E}$, s has to fulfill following requirements, depending on its use.

Use as Authentication Token. It is necessary that s has sufficient entropy to resist an on-line guessing attack by $\mathcal{E}$. A should implement strict rate-limiting for the number of permissible authentication attempts for each set of context observations, since re-trying does not help if the used context data do not change. It is therefore sufficient for s to have a min-entropy of approximately 20 bits, i.e., $\widetilde{\mathbf{H}}_\infty(S) \geq 20$, where S denotes the probability distribution from which s is drawn. This achieves a comparable resilience against guessing attacks as in the PIN-based Bluetooth pairing protocol, which can be considered a widely accepted industry standard for device pairing applications.

Use as Cryptographic Key. In schemes where the shared secret s is used directly as a cryptographic key, the requirements are much stricter. Not only has the min-entropy $\widetilde{\mathbf{H}}_\infty(S)$ to be sufficient to withstand off-line known-plaintext attacks, but, also the probability distribution S from which s is drawn, needs to be sufficiently indistinguishable from the uniform distribution in order for s to be considered a good cryptographic key.

4 SECURITY OF PAIRING SCHEMES

Recent context-based pairing schemes proposed in literature [8–10] use error-correcting codes to derive the shared secret s from context observations. None of these works, however, provide a quantitative empirical evaluation of their security under practical real-world requirements. In the following, we analyze the factors influencing security that these context-based pairing schemes can provide and evaluate their effectiveness in a real-world setting that is typical for IoT environments.

4.1 Context-Based Authentication

Since context observations in practice always are influenced by random errors arising from, e.g., context sensors' hardware or random fluctuations in the monitored context parameter, the observations of devices A and B will be similar but not identical. To compensate for these deviations, error-correcting codes like Golay or Reed-Solomon are used to perform *information reconciliation* [3] to 'correct' the context fingerprints of A and B to be identical.

The process of context-based authentication is shown in Fig. 1. First, device A derives a *context fingerprint* w which represents a quantization of its context observations. How this quantization is done is specific to each scheme and depends on the used context modality. In the subsequent discussion, we will simply refer to this process as *context fingerprinting*. Subsequently, A derives error-correcting information P from its fingerprint with the help of an appropriate error-correcting code (ECC), and sends it to B. Using this information B can adjust any deviations in its own context fingerprint w' in comparison to w, as long as the Hamming distance of its fingerprint w' to A's fingerprint w is within the error-correcting capability t of the used ECC, i.e., $dist(w, w') \leq t$. The resulting adjusted fingerprint $w^* = w$ can then either directly be used as the authentication token s or utilized further for deriving the cryptographic key s.

For deriving a cryptographic key s from the context fingerprints, A and B need to employ *privacy amplification*, as the error-correcting information P may provide partial information about the fingerprint w to adversary $\mathcal{E}$. The privacy amplification step will take fingerprint w about which $\mathcal{E}$ has partial information and output a secret s of which $\mathcal{E}$ has virtually no information. In addition, privacy amplification is used to make sure that the distribution S from which s is drawn is arbitrarily indistinguishable from the uniform distribution.

4.2 Entropy Loss

The shared secret s is derived from the context fingerprint w. Therefore its secrecy is dependent on the entropy of w from the point of view of adversary $\mathcal{E}$. This is measured in terms of its min-entropy $\widetilde{\mathbf{H}}_\infty(W|P)$, where W is the probability distribution of the fingerprints w and P denotes the error-correcting information revealed by A. Min-entropy is a measure of the 'worst-case' entropy, i.e., it measures the entropy of values of w that are easiest to guess for $\mathcal{E}$. It is therefore a good measure for the security of the scheme, since it considers the most favorable outcome for $\mathcal{E}$.

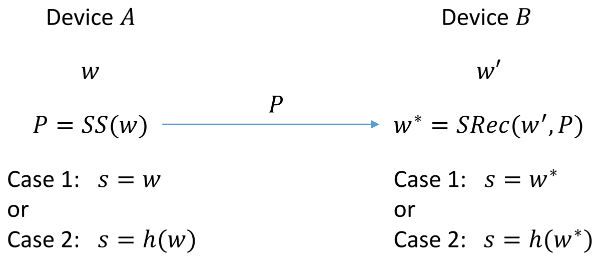


Figure 1: The context-based authentication approach for using context fingerprint w as an authentication token (Case 1) or for deriving a cryptographic key (Case 2).

Information reconciliation. When device A reveals the error-correcting information P for its fingerprint w this inevitably leaks some information about w. The extent of this entropy loss depends on the used error-correcting scheme. In the Schürmann and Sigg [10] scheme this is realized through a *fuzzy vault* [5] that utilizes *fuzzy commitments* [6]. Fuzzy commitments are equivalent to a *secure sketch* [4] utilizing the so-called code-offset construction, in which P is obtained by adding fingerprint w to the codeword $C(s)$ of secret s, i.e., $P = w \oplus C(s)$. We therefore analyze the entropy loss incurred by the error-correction with the help of secure sketches, as these can be generalized also to other schemes utilizing ECCs.

A secure sketch as introduced by Dodis *et al.* [4] is a pair of efficient algorithms $SS(\cdot)$ and $SRec(\cdot, \cdot)$ such that the secure sketching operation $SS(w) = P$ provides error-correcting information P that can be used to reconstruct w using the operation $SRec(w', P) = w$ given a value w' that is sufficiently similar to w, i.e., $dist(w, w') \leq t$. For secure sketches based on $[n, k, 2t + 1]$ ECCs it can be shown [4] that the entropy loss incurred by revealing the error-correcting information P is bounded by $(n - k)$, where n denotes the length, k the dimension and t the error-correcting capability of the ECC. The selection of the code to be used for information reconciliation is dependent on the amount of error-correction t that is required. In general, an ECC with higher error-correction capability will also incur a higher entropy loss.

Privacy Amplification. If the reconciled context fingerprint w is used to derive a cryptographic key, privacy amplification is needed to obtain a secret s over which $\mathcal{E}$ does not have even partial information. This is not considered in the Schürmann and Sigg scheme. Mathur *et al.* mention the need for privacy amplification, but do not take the entropy loss caused by it into account. To this end, a universal hash function $h(\cdot)$ can be used on the fingerprint w to generate a close-to-uniformly distributed secret, of which the adversary $\mathcal{E}$ does not have any information. According to the generalized

Leftover Hash Lemma (LHL) [2], the privacy amplification will incur $\log \epsilon^{-1}$ bits of entropy loss, where ϵ is a security parameter determining how indistinguishable the output is from the uniform distribution.

5 EVALUATION

To evaluate the feasibility of context-based authentication for IoT devices in a real-world setting that is applicable to typical smart home appliances like smart light bulbs, smart power plugs, IP cameras, etc., we performed two longitudinal experiments in domestic and office environments, representing typical deployment environments for IoT devices. In both experiments, data collection was performed continuously over a time period of 30 days in order to capture typical variations in contextual activity caused by daily and weekly differences in routines. In our evaluation we focus on the audio modality, as it is readily available and the required sensors relatively inexpensive to integrate in devices.

We focus on two measures of fitness: the *false accept rate* (FAR) and the *false reject rate* (FRR). FAR measures the rate at which fingerprints of adversary $\mathcal{E}$ will be falsely accepted by A as genuine, enabling thus an impersonation attack. FRR in contrast, measures the rate at which fingerprints of a genuine device B will be falsely be rejected by A. As FAR is a measure of the security of the scheme and FRR for its usability in practice, a good context-authentication scheme will seek to minimize both of these measures.

5.1 Data Collection

For data collection we used recent models of Android smartphones for which we had developed a data collection app recording the ambient sound energy level in the context every 100 ms. In each experiment we considered two different settings: one with two and another with three co-located devices marking IoT devices in the same trust domain $\mathcal{D}$, and one adversary device $\mathcal{E}$. In total the dataset covered therefore 12 distinct devices over a period of 30 days, covering more than 8000 hours of context measurements.

In each experiment, data collection devices were installed in the target environment at a distance of 2-3 meters from each other in order to model the relative positioning of typical co-located IoT devices. Adversarial devices were placed in adjacent rooms. However, due to practical constraints in the experimental set-up in the Home environment, the contextual isolation of the adversary device $\mathcal{E}$ was not as good as in the office environment, as the adjacent room was connected by a light-weight door that had to be opened from time to time. This allowed us, however, to analyze what impact the quality of the contextual separation has on the security of the context-based pairing.

5.2 Context Quantization

We utilize a fingerprint quantization scheme based on detecting prominent peaks in the audio measurements and using *list-encoding* to generate context fingerprints w. List-encoding is an efficient way of transforming continuous measurements into binary fingerprints as, e.g., Mathur *et al.* [8] have shown. In contrast to their scheme, which used minima and maxima of observed RF-measurements to encode "1" and "0" bits of the fingerprint, respectively, we slightly modified their scheme, as the audio signal doesn't contain clear minima. In our scheme A detects significant peaks in the audio measurements and uses these to encode "1" bits of its context fingerprint w. To encode "0" bits, A will randomly pick a roughly equal amount of non-peak observations at a minimum distance of 500 ms from any observed peaks and use these to encode zero bits. For the resulting fingerprint w, A will then derive the error-correcting information P and sends it along with the timestamps ts_i of the observations used to encode the fingerprint bits to B.

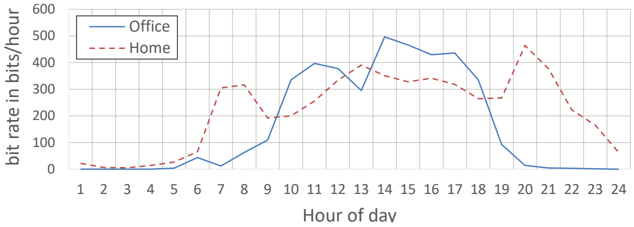


Figure 2: Bitrate of fingerprint extraction during different times of day

Device B will then use the timestamps ts_i to decode its fingerprint w' based on its own context measurements. It will decode each ts_i corresponding to a peak within a distance of 500 ms as a "1" bit and as a "0" bit if it does not correspond to a peak within this time window.

5.3 Contextual Activity

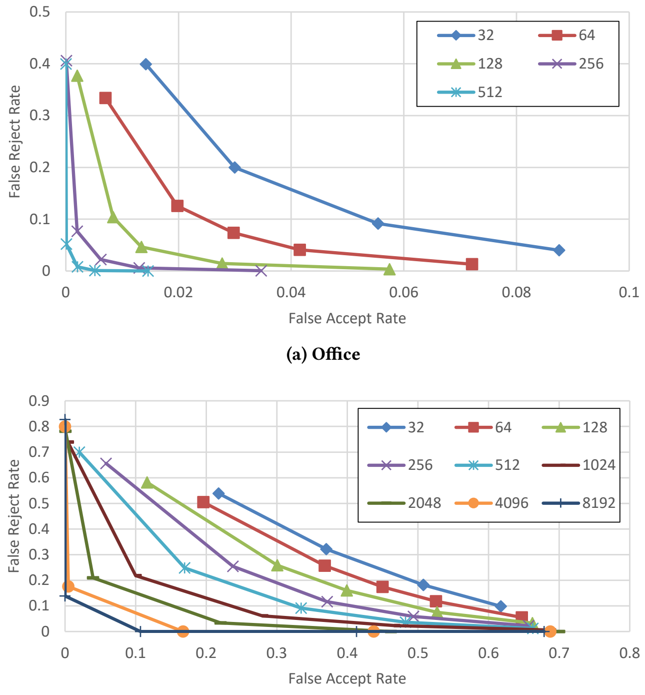
As fingerprint extraction is dependent on observed contextual activity, the amount of fingerprint bits that can be obtained from the context typically varies depending on the hour of day. The average hourly bitrate during different times of the day for the evaluation data is shown in Fig. 2. We focus our analysis therefore on the active hours of the day, i.e., on the hours between 6.a.m. to 9 p.m. in the Home environment and between 9 a.m. and 6 p.m. in the Office environment. During these active times the average bit rate in the Home environment was 309 bits per hour, whereas it was 368 bits per hour in the Office environment.

5.4 Similarity of Fingerprints

In the Home environment, average similarity of fingerprints extracted during the active hours of the day is constantly over 92%, the average being 93.2%. For the Office environment, during the active office hours on weekdays, even higher similarity can be reached, being constantly at least 94%, the average being 95.2%.

The fingerprint similarities for adversary devices are in both scenarios consistently lower than 90%, 86.1% in the Home scenario and 67.9% in the Office scenario on average. This difference shows the impact that the lower quality of contextual separation in the Home experiment has. These results suggest that in both environments, an ECC with error-correcting capability of ca. 10% would be sufficient to allow co-located devices to successfully pair, while adversarial devices would not be able to do so.

However, the above figures apply only to the *average* case. Our evaluation revealed that another factor, which earlier works [8, 10] have not explicitly taken into account has to be considered, namely the inherent variation in the similarity of context fingerprints. Our



(a) Office

(b) Home (with insufficient contextual separation)

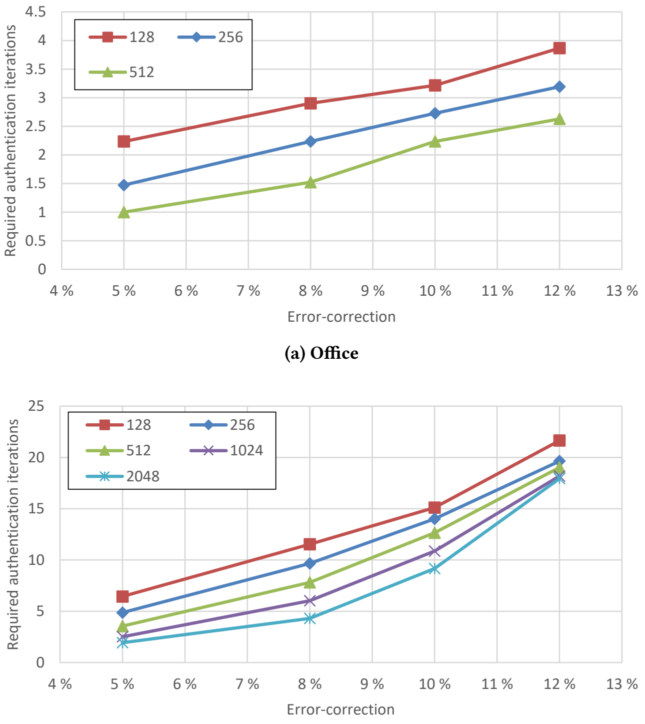
Figure 3: FAR vs. FRR for error-correction levels 5%, 8%, 10%, 12% **and** 15% **for different fingerprint lengths.**

data show that from time to time the fingerprint of adversary $\mathcal{E}$ is in fact sufficiently similar to the fingerprint of A, thus enabling $\mathcal{E}$ to falsely authenticate with A. Two factors affect the probability of this happening: 1) higher error-correcting capability t increases the probability that $\mathcal{E}$'s fingerprint will be accepted, while 2) longer fingerprints average out short-term fluctuations in fingerprint similarity, thus reducing $\mathcal{E}$'s success probability. Figure 3 shows the impact of these factors on the FAR and FRR values.

Due to the better contextual separation in the Office experiment, the FAR/FRR values (Fig. 3a) are clearly lower than in the Home experiment (Fig. 3b). For short fingerprint lengths, the FAR is relatively high, e.g., ranging from 1.4% to 8.6% for fingerprints of length 32 bits. Increasing the fingerprint length effectively reduces FAR, so that close-to-optimal performance can be achieved with a 512-bit fingerprint length with a FAR of 0.2% and FRR of 0.8% at an error-correction level of 10%.

The values for the Home experiment in Fig. 3b show how crucial contextual separation is for the security of the scheme. For short fingerprint lengths, $\mathcal{E}$ has a relatively high success probability of 21.8% to 61.7%. In this experiment, even using extremely long fingerprints of 8192 bits would bring down the FAR to only 10.7%.

From Fig. 3 we can, however, see that even under favorable conditions, the adversary has a non-negligible chance of succeeding. This means that in order to further decrease the FAR for increased security, one would need to adopt the approach proposed by Miettinen *et al.* (Sect. 2.3), where the authentication is iteratively repeated, in order to increase the confidence in the counterpart's authenticity. The number of authentication iterations required is dependent on the FAR of the used ECC. Figure 4 shows the amount or required iterations for reaching a FAR of 2^{-20} (comparable security to Bluetooth pairing) for the different ECCs in the examined environments. We can see that, e.g., at the 10% error-correction level, 3 − 4 iterations in the Office environment would be required, while 10 − 16 repetitions would be needed in the Home environment.



(a) Office

(b) Home (with insufficient contextual separation)

Figure 4: Required number of authentication iterations to reach FAR of 2^{-20} **for different fingerprint lengths.**

5.5 Entropy Analysis

As discussed in Sect. 4.2, an $[n, k, 2t + 1]$-code will incur an $(n - k)$-bit entropy loss during the information reconciliation stage. The higher the required error-correcting capability is, the larger also the entropy loss. From this point of view, Reed-Solomon (RS) codes provide an optimal trade-off between error-correction capability and entropy loss, as for each symbol of error-correction capability, the code will incur an entropy loss of two symbols, i.e., in practice an error-correction capability of t bits will incur $2t$ bits of entropy loss. This assumes an approach used, e.g., by Schürmann and Sigg [10], where fingerprint bits are encoded with the help of symbols of the RS-code. Our evaluation shows that an error-correction capability of ca. 10% is required to enable A and B to perform successful pairing with low FRR. The fingerprint w would therefore need to have initially at least 25 bits of min-entropy to retain a leftover entropy of 20 bits after the information reconciliation step with

20% of entropy loss. As discussed in Sect. 3, this would be sufficient for using the fingerprint as an authentication token.

For deriving a cryptographically strong secret of 128 bits, also the entropy loss incurred by privacy amplification needs to be taken into account. As discussed in Sect. 4.2, this amounts to $\log \epsilon^{-1}$ bits, where ϵ is a parameter defining the desired indistinguishability of S, the distribution of the secrets s, from the uniform distribution. For, e.g., $\epsilon = 2^{-20}$ this would result in additional 20 bits of entropy loss associated with the privacy amplification step. To retain a min-entropy of 128 bits after information reconciliation and privacy amplification, the min-entropy of the context fingerprint would therefore need to be at least $\frac{128+20}{80\%} = 185$ bits, if a Reed-Solomon error-correcting code with 10% error-correction capability is used.

5.6 Duration of Pairing

Based on our evaluation data the best strategy for adversary $\mathcal{E}$ to guess A's fingerprint is to use its own fingerprint, as on average 86.1% of fingerprint bits in the Home environment and 67.9% of the fingerprint bits in the Office environment will be identical with A's fingerprint bits. Therefore, the amount of entropy of each fingerprint bit from $\mathcal{E}$'s point of view is only 0.24 bits in the Home and 0.32 bits in the Office environment. Obtaining sufficient min-entropy, i.e. 25 bits, for an authentication token will therefore require $\lceil \frac{25}{0.24} \rceil = 105$ fingerprint bits in the Home and $\lceil \frac{25}{0.32} \rceil = 79$ fingerprint bits in the Office environment, on average. At average bit generation rates of 309 and 368 bits per hour, the required time for acquiring sufficient bits would therefore be 20.4 min in the Home and 12.9 min in the Office environment.

Similarly, for obtaining the required 185 bits of min-entropy for a cryptographic secret would require $\lceil \frac{185}{0.24} \rceil = 771$ fingerprint bits in the Home and $\lceil \frac{185}{0.32} \rceil = 579$ fingerprint bits in the Office environment. The respective required times to harvest this entropy would accordingly be 149.7 minutes in the Home and 94.4 minutes in the Office environment.

5.7 Summary

Our evaluation shows that using context measurements for establishing a shared secret is possible, given sufficient time to harvest entropy from the ambient environment. However, for any contexts where a complete contextual separation from the outside environment can't be guaranteed, the authentication process has to be repeated a sufficient number of times to bring down the false accept rate to an acceptable level (cf. Fig. 4). Therefore, an approach along the lines of [9], in which initially only basic level access is granted and additional privileges only added as more successful authentication iterations are completed should be followed in applying context-based pairing in real-world environments.

6 RELATED WORK

Earlier proposals for context-based pairing have focused on using RF-signals. AMIGO by Varshavsky *et al.* [11] aimed at authenticating the co-presence of devices by comparing the received signal strength indicators (RSSI) of WiFi data packets. This approach was subsequently extended by Kalamandeen *et al.*'s Ensemble [7], which not only observed incoming packets, but utilized also transmissions by an ensemble of trusted wearable devices to verify proximity of devices. However, subsequent work has shown that RSSI values are relatively predictable and can potentially be inferred or influenced by an adversary remotely, if he is aware of the positions of A and B. Mathur *et al.* [8] therefore subsequently introduced the ProxiMate system as discussed in Sect. 2.2, which relies on physical properties of the RF-field for secrecy. These approaches are, however, only applicable for demonstrative identification via proximity, as the devices have to be very close to one another (e.g., 15 - 35 cm) for authentication to be possible. Their applicability for large-scale authentication of numerous IoT devices, e.g., in a Smart Home environment, is questionable, as the user needs to separately point out each and every device, which might be tedious, e.g., in the case of numerous smart light bulbs installed in the ceiling of an apartment.

The scheme of Schürmann and Sigg [10] proposes to use audio in the proximate context as a source for a shared secret, as discussed in Sect. 2.2. However, they do not provide a quantitative analysis of the security of their scheme, discussing the entropy loss associated with the use of ECCs. They also propose to use the random secret s selected by A directly as a secret key without privacy amplification, thereby not taking into account that the secrecy of s depends only on the min-entropy of the used w, over which $\mathcal{E}$ obtains partial information due to the released error-correcting information P.

A similar approach is taken by the scheme of Miettinen *et al.* [9] (Sect. 2.3) that proposes an implicit context-based authentication scheme based on audio and luminosity. In this scheme, an initial strong *unauthenticated* shared secret is established between A and B, which is subsequently iteratively evolved by repeated context-based authentication steps in order to gradually establish confidence in the authenticity of the counterpart. Our evaluation shows that this indeed is necessary, unless complete contextual isolation of the target context from adversary $\mathcal{E}$ can be guaranteed.

7 CONCLUSION

Context-based pairing for authentication of IoT devices can provide significant usability benefits as compared to traditional solutions like, e.g. Bluetooth pairing. Applying it in practice, however, has caveats that have not been sufficiently considered in earlier proposals [8–11]. Firstly, one has to consider and quantify the entropy losses related to the applied error-correction and privacy amplification in order to estimate a sufficient amount of entropy to be harvested from the environment. In addition, our evaluation shows that one also has to have a good understanding about the performance of the fingerprinting approach as well as the level of contextual separation that the target environment provides. Therefore, before deployment of context-based pairing solutions, sufficient understanding about the target contexts should be acquired in order to make informed decisions about relevant parameters like error-correction level, used fingerprint lengths and number of required authentication iterations, so that the used approach can in fact provide sufficient security in a real-world setting.

REFERENCES

[1] Dirk Balfanz, Diana K. Smetters, Paul Stewart, and H. Chi Wong. Talking to Strangers: Authentication in Ad-Hoc Wireless Networks. In *NDSS, 2002.*

[2] Boaz Barak, Yevgeniy Dodis, Hugo Krawczyk, Olivier Pereira, Krzysztof Pietrzak, François-Xavier Standaert, and Yu Yu. Leftover Hash Lemma, Revisited. In *Proc. 31st Annual Cryptology Conference (CRYPTO 2011).*

[3] Gilles Brassard and Louis Salvail. Secret-Key Reconciliation by Public Discussion. In *Proc. Workshop on the Theory and Application of Cryptographic Techniques Lofthus (EUROCRYPT '93), Norway, May 23–27, 1993*.

[4] Yevgeniy Dodis, Leonid Reyzin, and Adam Smith. Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data. In *Proc. Intl. Conf. on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2004), Interlaken, Switzerland, May 2-6, 2004*.

[5] Ari Juels and Madhu Sudan. 2006. A Fuzzy Vault Scheme. *Designs, Codes and Cryptography* 38, 2 (01 Feb 2006), 237–257. https://doi.org/10.1007/s10623-005-6343-z

[6] Ari Juels and Martin Wattenberg. 1999. A Fuzzy Commitment Scheme. In *Proc. ACM CCS, 1999*.

[7] Andre Kalamandeen, Adin Scannell, Eyal de Lara, Anmol Sheth, and Anthony LaMarca. Ensemble: Cooperative Proximity-based Authentication. In *Proc. 8th Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys '10), 2010*.

[8] Suhas Mathur, Robert Miller, Alexander Varshavsky, Wade Trappe, and Narayan Mandayam. ProxiMate: Proximity-based Secure Pairing Using Ambient Wireless Signals. In *Proc. 9th Intl. Conf. on Mobile Systems, Applications, and Services (MobiSys '11), 2011*.

[9] Markus Miettinen, N. Asokan, Thien Duc Nguyen, Ahmad-Reza Sadeghi, and Majid Sobhani. Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices. In *Proc. ACM CCS, 2014*.

[10] Dominik Schürmann and Stephan Sigg. 2013. Secure Communication Based on Ambient Audio. *IEEE Trans. Mob. Comput.* 12, 2 (2013), 358–370.

[11] Alex Varshavsky, Adin Scannell, Anthony LaMarca, and Eyal de Lara. Amigo: Proximity-Based Authentication of Mobile Devices. In *UbiComp 2007: Ubiquitous Computing*. Lecture Notes in Computer Science, Vol. 4717. Springer.