Context-Based Zero-Interaction Pairing and Key Evolution for Advanced Personal Devices

Markus Miettinen
Technische Universität Darmstadt
markus.miettinen@trust.cased.de

N. Asokan
Aalto University and University of Helsinki
asokan@acm.org

Thien Duc Nguyen
Technische Universität Darmstadt
ducthien.nguyen@trust.cased.de

Ahmad-Reza Sadeghi
Technische Universität Darmstadt
ahmad.sadeghi@trust.cased.de

Majid Sobhani
Technische Universität Darmstadt
majid.sobhani@trust.cased.de

ABSTRACT

Solutions for pairing devices without prior security associations typically require users to actively take part in the pairing process of the devices. Scenarios involving new types of devices like Internet-of-Things (IoT) appliances and wearable devices make it, however, desirable to be able to pair users' personal devices without user involvement.

In this paper, we present a new approach for secure zero-interaction pairing suitable for IoT and wearable devices. We primarily require pairing to happen between "correct" devices – the devices that the user intends to pair. Our pairing scheme identifies the correct devices based on measuring *sustained co-presence* over time. We do this by having the devices compute a fingerprint of their ambient context using information gathered through commonly available sensor modalities like ambient noise and luminosity. We introduce a novel robust and inexpensive approach for fingerprinting contexts over time. Co-present devices will observe roughly similar context fingerprints that we use in a key evolution protocol to gradually increase the confidence in the authenticity of the correct devices. Our experiments show the effectiveness of this approach for zero-interaction pairing.

Categories and Subject Descriptors

K.6.5 [**Management of computing and information systems**]: Security and Protection—*Authentication*

Keywords

contextual security; context-based pairing; zero-interaction

CCS'14, November 3–7, 2014, Scottsdale, Arizona, USA.

http://dx.doi.org/10.1145/2660267.2660334.

1. INTRODUCTION

Traditional approaches for key agreement between personal devices without any prior security association (also known as "pairing") typically rely on some form of active user involvement to authenticate the key agreement. For example, the user may be asked to compare authentication strings displayed on the devices or to bring the devices close enough so that they can communicate via a Near Field Communication channel [17]. Such measures are required to thwart man-in-the middle attacks targeting the initial key agreement. Relying on user involvement to authenticate pairing is cumbersome, error-prone and does not scale well. It is therefore desirable to devise *zero-interaction* pairing mechanisms which do not require *any* user interaction.

In this paper, we consider the challenge of zero-interaction pairing for two particularly important emerging classes of personal devices: Internet-of-Things (IoT) appliances and wearables. There has been an increasing interest in both of these classes accompanied by a steady stream of product announcements and media coverage. IoT devices are intelligent network-enabled appliances utilizing connectivity and local computation to enable richer functionality and improved user experience. Examples of IoT devices include Nest smoke detectors and thermostats [12], the Oral-B connected toothbrush [13], the Bee+ smart injection tracker for diabetic patients [21], and the Spotter smart home sensor [14]. According to a recent Gartner forecast, the total installed base of IoT devices will grow to 26 billion units by 2020 [8]. Such devices will therefore play a significant role in the future end-user computing infrastructures. Similarly, new wearable devices include wristbands used for activity monitoring (e.g., LG LifeBand Touch, POLAR Loop Activity Tracker), augmented reality gadgets like the Google Glass near-eye display device, smart watch devices (e.g., Samsung Galaxy Gear) and many more. It is estimated that by 2017, 50% of all smartphone app interactions will involve wearable devices [7], emphasizing the important role that wearables are expected to play in future smartphone usage scenarios.

Both IoT devices and wearables process sensitive information and critical operations. Thus securing their communications is essential. On the other hand, in both cases ordi-

nary users may own and manage many devices. The devices themselves may not have any user interfaces. Therefore, zero-interaction pairing will greatly improve the usability of configuring these devices.

The security goal of pairing personal devices is to ensure that the key agreement takes place between the devices owned by the user. In traditional pairing schemes, users are required to *demonstratively* identify the correct devices [1]. The requirement of zero-interaction, however, rules out demonstrative identification.

Existing pairing solutions that do not require direct user involvement can be broadly divided into two classes: key predistribution and context-based pairing approaches. Key predistribution-based approaches (e.g., [6, 2, 10, 18]) are mainly intended for digital sensor network (DSN) scenarios and require key material to be distributed to all network nodes before their deployment in the field. In IoT scenarios, however, such predistribution is not feasible due to the overwhelmingly large number of devices deployed and the fact that there are hundreds if not thousands of different device vendors that do not necessarily share any security associations with each other. Furthermore, in our scenarios multiple authentication domains may exist in overlapping physical spaces, such as the IoT domains of two neighboring apartments. Therefore the pairing solution must be capable of automatically distinguishing between such overlapping domains.

Context-based pairing approaches (e.g., [20, 16]), on the other hand, use *co-presence* of devices to identify the devices to be paired. These schemes leverage the fact that co-present devices will perceive roughly the same ambient context via their on-board sensors – thus each device takes a momentary snapshot of its ambient context using a given sensor modality (e.g., acoustic or electromagnetic) and uses the resulting "context fingerprint" to authenticate key agreement. Relying on a one-shot fingerprint for zero-interaction pairing has some drawbacks in the scenarios we consider. First, to ensure security the context fingerprint must have sufficient entropy (e.g., 128 bits). This imposes strict requirements on the fingerprinting technique such as the need for tight time synchronization between devices (as in [16]) or access to low-level information like raw WiFi packets that is typically not available to apps in commodity devices (as in [20]). Second, momentary co-presence of two devices does not always imply that the devices belong to the same user.

Our goal and contributions. In this paper, we present a novel approach for zero-interaction pairing that is suitable for IoT and wearable device scenarios. Unlike previous schemes, we identify correct devices based on the notion of *sustained co-presence*: our scheme uses sensed context fingerprints to evolve the pairing key periodically in a way that is only possible for devices co-present over extended periods of time. This is based on the intuition that in the long run, a user's personal devices are much more likely to be co-present with one another as compared to other users' devices. We use readily available context sensor modalities like audio and luminosity. An initial (potentially insecure) pairing is gradually strengthened using a key evolution approach that step-by-step establishes and increases the authenticity of correct peers, while making it increasingly difficult for wrong devices to maintain an authenticated pairing with the user's devices.

The context fingerprints we use are based on observable changes in the average luminosity and noise levels of the devices' ambient context over a longer time period. Use of longer time periods implies that our fingerprinting scheme does not require tight time synchronization and is thus robust. Fingerprints are used to authenticate each key evolution step.

Our main contributions are the following:

- We describe a **robust context-based shared entropy extraction** scheme for audio and luminosity modalities and demonstrate its effectiveness using real context data (Section 4).
- We incorporate the entropy extraction into a novel **key evolution** approach for automatically pairing personal devices of the user (Section 3) and reason about its security (Section 5). The key evolution ensures that pairing succeeds between devices that exhibit sustained co-presence, which is typical for personal devices in IoT and wearable device scenarios.

The rest of this paper is structured as follows: In Sect. 2, we describe the context-based pairing scenario and problem setting. In Sect. 3 we describe the key evolution approach, which utilizes a fingerprinting scheme presented and evaluated in Sect. 4. An analysis of the security properties of our approach is presented in Sect. 5. We conclude the paper with a presentation of related work in Sect. 6 and a conclusion in Sect. 7.

2. PROBLEM DESCRIPTION

We focus on the problem of pairing between two devices. By "pairing", we mean the process of setting up a shared security association (e.g., a shared symmetric key) between the devices. Pairing must be established only between the devices that the user intends to pair, i.e., devices belonging to the same user. We refer to these as the *correct peers*. Conversely devices owned by other users are *wrong peers*. The security goal of pairing is to ensure that only a pairing between correct peers is accepted as *genuine*. Our approach is to develop a context-based pairing scheme to this end. "Context" means here the ambient environment of a device. A device can characterize its context by using *context data* that can be sensed using on-board sensor modalities. In this paper, we use ambient audio (sensed by microphones) and luminosity (sensed by lux sensors) for characterizing a context.

We consider the pairing problem in two particular scenarios: a static setting primarily concerning IoT devices installed at the user's home, and, a mobile setting for wearable personal devices.

IoT Scenario. The IoT scenario is shown in Fig. 1. A user has installed some IoT devices d_1 and d_2. Her neighbor also has an IoT device $\mathcal{A}$ in his apartment. IoT devices are typically equipped with WiFi or Bluetooth connectivity and hence may be placed within each other's communication range. We can assume that all devices are able to communicate with one another and are equipped with context sensors to sense contextual parameters. Over time, the devices d_1 and d_2 in the user's home should establish a secure pairing between each other *without user interaction*, while making sure that a trusted pairing is not erroneously established with device $\mathcal{A}$.

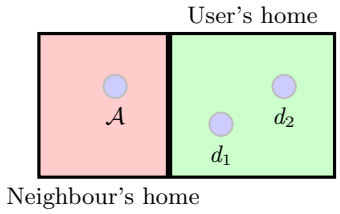

Figure 1: Scenario 1: Pairing of personal IoT devices

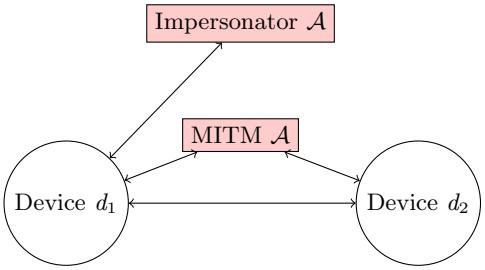

Figure 2: Scenario 2: Pairing of personal wearable devices

Wearable Scenario. The second scenario is concerned with the secure pairing of personal wearable devices, as depicted in Fig. 2. The user has a smartphone d_1 and buys a smart watch d_2, turns it on and starts using it. The newly activated smart watch d_2 actively searches for smartphones nearby and establishes an initial pairing with all such devices that it can find nearby. Similarly d_1 will accept any initial pairing from any wearable device that contacts it.

In order to establish the authenticity of such initial pairings, the two devices then attempt to evolve their pairing key using a key evolution protocol during a *gestation period*. If at the end of the gestation period the two devices have sufficient confidence in the authenticity of each other belonging to the same domain, i.e., being owned by the same user, they accept the pairing key as genuine. Otherwise, they discard the pairing the key. If the wearable device d_2 accepts a key, it stops making new key pairing requests, since it is already associated with the correct user's smartphone. Also in this scenario, we need to make sure that the pairing key of a device $\mathcal{A}$ not belonging to the user is not erroneously accepted as genuine.

2.1 Threat Model and Assumptions

The threat we are concerned with is that an adversary device $\mathcal{A}$ succeeds in making a legitimate device d_1 accept a pairing with $\mathcal{A}$ as genuine.

IoT Scenario. In the IoT scenario, the adversary $\mathcal{A}$ is an IoT device in the neighbor's apartment. This device can be benign, just trying to pair with other devices it can discover in its proximity, or, malicious, if infected with malware, aggressively trying to pair with and infiltrate any IoT networks it can discover. The wrong peer $\mathcal{A}$ is permanently near device d_1 and can communicate with it over a wireless link, but it is not able to monitor d_1's ambient context, since it is separated from it by a solid wall.

If the neighboring apartment where the wrong device $\mathcal{A}$ is located has large windows facing the same direction as the user's apartment, $\mathcal{A}$ may have visibility to any changes in the outside luminosity affecting the lighting conditions in the user's apartment, but it is not able to directly observe the lighting conditions in the ambient context of d_1. Specifically, $\mathcal{A}$ is not capable of mounting targeted attacks, i.e., attacks that are executed by directly monitoring the target apartment where d_1 is located, e.g., from another apartment over the street. Since $\mathcal{A}$ is assumed to be a regular IoT device, it neither has the directional high-fidelity sensors required for monitoring a specific target over large distances nor the functional logic for mounting such attacks.

Wearable Scenario. In the wearable device scenario, the adversary $\mathcal{A}$ is either a malicious attacker trying to play a man-in-the-middle attack on the user and his wearable device, in order to obtain sensitive information exchanged between them, or it could be just someone else's device searching for its own peer device. We assume that the wrong peer $\mathcal{A}$ is from time to time present in the same context as d_1, e.g., while d_1 is visiting a place where also $\mathcal{A}$ is located (cf. Fig 2) and hence $\mathcal{A}$ can observe the same contextual parameters as d_1. However, $\mathcal{A}$ is not able to follow the user constantly. The amount of time $\mathcal{A}$ is able to monitor d_1's context is therefore limited and significantly smaller than the time that d_1 and d_2 spend co-located in the same contexts. We follow a standard Dolev-Yao adversary model [4] and assume that $\mathcal{A}$ has full control over all communication channels.

In both scenarios we assume that the user's own devices d_1 and d_2 consistently spend most of the time in the same contexts. In the IoT scenario, the context is spatially static, e.g., the user's home, since typically IoT devices are household appliances (smart TVs, smart thermostats, etc.) that are relatively static objects. Wearable devices like a smart watch, on the other hand, are continuously carried by the user and are therefore sharing the same, although changing context during the day as the user moves around.

2.2 Objectives

Our main objectives are as follows.

Authenticated pairing. User devices (IoT devices and wearables) securely establish authenticated pairings with the correct peer devices, i.e., a user's device d_1 accepts a pairing with d_2 after a gestation period if, and only if, d_1 and d_2 are owned by the same user and thus are co-present for longer periods of time. Authenticated pairings are not established with wrong peers $\mathcal{A}$, including attacker devices playing man-in-the middle and impersonation attacks against the correct peers.

Zero-interaction. The pairing must happen *without user interaction*, i.e., based solely on information that the involved devices can communicate and sense from their ambient context without human involvement.

2.3 Solution Approach

Previous approaches for context-based pairing presented in literature use context information to establish a one-shot secure pairing [20, 16]. The security of these approaches depends on the assumption that the adversary is not present sufficiently close in the context of the user devices d_1 and d_2 when the pairing is performed and thus unable to observe the same contextual parameters as d_1 and d_2. These approaches rely, however, on the user to visually determine that no adversary $\mathcal{A}$ is present in the proximity of the devices d_1 and d_2

before the pairing is initiated. In a zero-interaction setting, this is not possible. For example, in a situation in which wearable devices are taken into use at a moment when several parties are present in the same room, an adversary $\mathcal{A}$ might very well be present.

Therefore, we follow a more in-depth defense strategy by utilizing a *key evolution approach* described in Sect. 3. In our approach, the target device d_1 is initially entitled to establish pairings with all other devices in proximity, including correct and wrong peers. These pairings are, however, assigned an *authenticity rating* that is initially zero, meaning that the authenticity of the counterpart has not been verified. Key evolution is then used to gradually increment the authenticity rating of correct peers, so that over time only pairings with correct peers will be accepted as genuine.

In earlier approaches, context fingerprints used for one-shot pairing must have sufficient entropy (e.g. 128 bits). Obtaining a sufficient amount of entropy from a short context snapshot requires therefore tight time synchronization between the devices d_1 and d_2 to be paired [16]. On commodity devices achieving sufficiently accurate synchronization might not be technically feasible. To overcome this limitation, we utilize a more robust fingerprinting approach that operates on longitudinal context measurements and is thus not as sensitive to time synchronization issues. The fingerprinting scheme used is described in Sect. 4.

3. CONTEXT-BASED KEY EVOLUTION

Our key evolution approach is based on the assumption that two devices that have established an initial pairing can utilize the common information about their ambient context observed over time to iteratively evolve their pairing key. With each successful iteration, the belief in the authenticity of the counterpart is increased, since the protocol is designed in a way that makes it hard for devices not continuously sharing the same context to execute it successfully.

In the approach, both peers extract *context fingerprints* from their surroundings by continuously monitoring their context. If the peers spend prolonged periods of time in the same context, observing the same contextual information, the fingerprints they extract will be similar as well. We will define the extraction of fingerprints in section 4.

The key evolution approach utilizes three conceptual components: *key evolution*, *key confirmation*, and, *key acceptance*. Key evolution and key confirmation are executed iteratively between the peer devices in what we call a *key evolution step*: evolving the pairing key and verifying the success or failure of each key evolution. After a sufficient number of key evolution steps have been performed, key acceptance is used to ultimately determine, whether a pairing counterpart is a correct or wrong peer.

To perform a key evolution based on context fingerprints, we require a fuzzy commitment scheme that is ideal w.r.t. the *hiding property*. Such a scheme is able to transform a secret value s into a commitment / opening value pair (δ, λ), such that δ does not reveal any information about the secret s, and all pairs $(\delta, \hat{\lambda})$ will reveal s if the Hamming distance $\texttt{Ham}(\lambda, \hat{\lambda}) \leq t$, but it is not feasible to find an opening value λ', for which $\texttt{Ham}(\lambda, \lambda') > t$, such that (δ, λ') would reveal the secret s. In this scheme, the value t is a parameter and denotes the maximum Hamming distance that the scheme allows for an opening value $\hat{\lambda}$ to have from λ, so that the secret s is revealed. In other words, if party d_1 commits to a secret s to obtain a commitment / opening pair $(\delta, \lambda) \leftarrow \texttt{Commit}(s)$, and a subsequent opening of the commitment by party d_2 yields $\hat{s} \leftarrow \texttt{Open}(\delta, \hat{\lambda})$, then $s = \hat{s}$ iff $\texttt{Ham}(\lambda, \hat{\lambda}) \leq t$.

We could utilize any key agreement scheme that provides such a fuzzy commitment (e.g., [3]), but for the purpose of this paper, we adopt and adapt the approach of Schürmann and Sigg in [16]. It is based on the *fuzzy vault* construction of Juels and Sudan [9]. It utilizes the error-correcting properties of Reed-Solomon codes [15] to enable two peers to agree on a common key, if the context fingerprints that peers extracted from information in their ambient context differ in at most t bits. The value of t depends on the parameterization of the Reed-Solomon code and can thus be selected freely based on the number of bit errors to be expected between the fingerprints of legitimate peers. The details of the key evolution approach are shown in Fig. 3.

3.1 Key Evolution

Initially, two devices d_1 and d_2 look for other devices to pair with. When they encounter each other for the first time, they establish an initial pairing key $K^0_{d_1,d_2}$ (e.g., by using a Diffie-Hellman key exchange). This initial key agreement is unauthenticated, i.e., neither device knows, whether the pairing counterpart belongs to the same owner or not. Our goal is to use subsequent key evolution to determine whether the pairing counterpart belongs to the same user or not.

Device d_1 initiates the protocol by sending a key evolution request `EVO_REQ` to device d_2. The request contains timestamps t_1 and t_2, specifying the starting and ending times on which to synchronize the generation of the context fingerprints. From the context observations $C_{d_1}(t_1, t_2)$ and $C_{d_2}(t_1, t_2)$ falling between the specified timestamps, both peers extract context fingerprints $F_{C_{d_1}} = \phi(C_{d_1}(t_1, t_2))$ and $F_{C_{d_2}} = \phi(C_{d_2}(t_1, t_2))$, respectively, by applying a fingerprint extraction function $\phi(\cdot)$ on the collected context sequences. The extraction function is defined in Def. 3 in Sect. 4.

After generating the fingerprints, device d_1 selects a random key evolution diversifier $K_r \in \mathbb{F}^m_{2^k}$, and uses the fuzzy commitment scheme to transform it into a commitment / opening value pair $(\delta, \lambda) \leftarrow \texttt{Commit}(K_r)$. The opening value $\lambda \in \mathbb{F}^n_{2^k}$ is calculated as the codeword for K_r using Reed-Solomon (RS) encoding: $\lambda \leftarrow \texttt{RS}(2^k, m, n, K_r)$. The commitment value δ is then calculated as the difference of the fingerprint $F_{C_{d_1}}$ and the codeword λ: $\delta = F_{C_{d_1}} \ominus \lambda$, where $\ominus$ denotes subtraction in the field $\mathbb{F}^n_{2^k}$.

Device d_1 then transmits the commitment value δ to device d_2, which in turn obtains an opening value $\hat{\lambda}$ and retrieves the key evolution diversifier by opening the commitment of d_1 : $K_r{}' \leftarrow \texttt{Open}(\delta, \hat{\lambda})$. It does so by decoding the opening value using the Reed-Solomon decoding function. Given that the fuzzy commitment scheme fulfils the hiding property requirement, $K_r = K_r{}'$ only if $\texttt{Ham}(\lambda, \hat{\lambda}) \leq t$. Since $\hat{\lambda}$ is calculated as $\hat{\lambda} = F_{C_{d_2}} \ominus \delta$, and δ as $\delta = F_{C_{d_1}} \ominus \lambda$, it means that the fingerprints $F_{C_{d_1}}$ and $F_{C_{d_2}}$ can differ in at most t bits, which in this case is the maximum number of bits the RS coding can correct. Otherwise, d_2 will not be able to open the commitment correctly, and the retrieved key derivation keys will not be identical, i.e., $K_r \neq K_r{}'$.

3.2 Key Confirmation

To determine whether the key evolution was successful, both devices calculate candidate pairing keys by using a key derivation function `KDF` applied on the old pairing key $K^i_{d_1,d_2}$

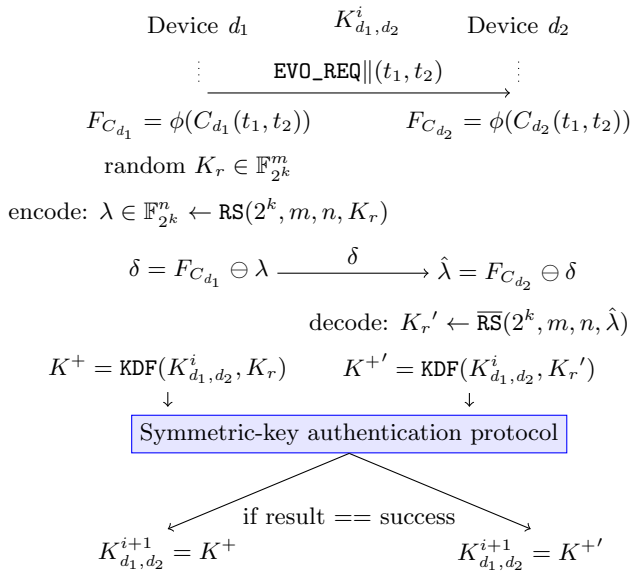

Figure 3: The Key Evolution Protocol

and the key evolution diversifier, i.e., K_r or $K_r{}'$, respectively. The peers then execute a symmetric-key authentication protocol with the candidate keys to determine, whether they are identical. The used protocol needs to be tolerant to offline guessing attacks. For example, a password-authenticated key-exchange scheme [22] can be used (although they are intended for long-lived short shared secrets). If the protocol succeeds, the key evolution step is considered successfully completed and the peers start using the candidate keys as their new pairing keys, i.e., device d_1 sets $K^{i+1}_{d_1,d_2} = K^+$ and device d_2 sets $K^{i+1}_{d_1,d_2} = K^{+\prime}$.

3.3 Key Acceptance

To ultimately determine whether a pairing counterpart is a correct or a wrong peer, we apply the following strategy: assuming that a wrong peer $\mathcal{A}$ is spatially or temporally limited in its ability to continuously monitor the context of a target device d_1, it is likely that $\mathcal{A}$ will fail in key evolution much more often than a correct peer d_2, who is predominantly co-present in d_1's context. By keeping track of the number of successful key evolutions each pairing counterpart is able to follow, it becomes therefore possible to distinguish the correct peer d_2 from wrong peer $\mathcal{A}$.

We need to take into account that wrong peers may appear at any point in the pairing and key evolution process. A wrong peer $\mathcal{A}$ may initiate the pairing first and impersonate a correct peer d_2 which will come into communication range only later, or, $\mathcal{A}$ may appear after an initial pairing with the correct peer d_2 has already been established, and may claim to be d_2. Since we assume that there is no prior security association between any of the devices, we can't distinguish with certainty whether the former or the latter device requesting the pairing is the correct personal device d_2. Therefore, we need to initially accept all pairing requests for a particular device identity and use the key evolution protocol to verify, which device actually is the authentic one.

To be able to distinguish different devices from each other, we assign a key chain identifier $ID^X_d = \mathtt{Hash}(K^0_d)$ for each device d. The chain identifier is a hash value of the key K^0_d derived during the initial unauthenticated pairing with d, and X is the identity that d claims to represent. We denote the set of all devices d claiming identity X with D_X. We evolve the pairing key K^i_d independently for each device's key chain and keep track of the number of successful key evolution steps associated with each key chain identifier ID^X_d as well as the total number of successful key evolution steps for the claimed identity X. The ratio of successful key evolution steps for each key chain identifier ID^X_d to the total number of successful key evolution steps for the related identity X becomes therefore a measure for the authenticity of the device associated with that key chain identifier.

Definition 1 (Authenticity rating α). *Let $\gamma(ID^X_d)$ denote the number of successful key evolution steps that a device has performed with a peer device d with key chain identifier ID^X_d under the claimed identity X. The authenticity rating $\alpha(ID^X_d)$ is the ratio of successful key evolution steps for the key chain ID^X_d to the overall number of successful key evolution steps for identity X:*

$$\alpha(ID^X_d) = \frac{\gamma(ID^X_d)}{\sum_{d_i \in D_X} \gamma(ID^X_{d_i})}. \quad (1)$$

The key evolution is performed during predetermined key evolution cycles. During each key evolution cycle, device d_1 will try to perform key evolution for an identity X with each device $d \in D_X$ claiming to represent that identity. If these attempts succeed, the count of successful key evolution steps for identity X is incremented. Our key evolution approach is designed in a way that only devices d that are in the same context as the target device d_1 for the majority of the time during a key evolution cycle will succeed in the key evolution step. Thus, since correct peers are significantly more often in the same context than wrong peers, the value $\gamma(ID^X_d)$ for any correct peers d will, over time, grow larger than for any wrong peers that will inevitably 'miss' such key evolution steps during which the wrong peers are not in the same context as d_1, or are unable to observe d_1's context.

The context-based pairing approach can therefore be summarized as follows:

1. Establish pairing key with device d claiming to be X. Assign initial authenticity rating $\alpha(ID^X_d) = 0$ to it.

2. Monitor the context and regularly evolve pairing keys with other devices based on derived context information.

3. After the number of successful key evolution steps for identity X reaches a specified threshold value α_{thr}, check if the acceptance criteria listed below hold. If either criterion does not hold, the key evolution process is continued, and acceptance criteria re-evaluated after each successful key evolution step for identity X.

The first acceptance criterion requires that in order to be accepted as genuine, a peer device d's pairing key under an identity X needs to have a sufficient authenticity rating $\alpha(ID^X_d)$, and this rating has to be higher than any other peer's rating for X by a specified margin α_{marg} in order to make the determination of the correct peer unambiguous.

CRITERION 1 (AUTHENTICITY DOMINANCE). *Let $\alpha_{min}, \alpha_{marg} \in [0,1]$ denote a minimal authenticity threshold and an authenticity margin, respectively. If there is a device $d \in D_X$ claiming to represent identity X, such that $\alpha(ID_d^X) > \alpha_{min} \wedge \forall d_i \in D_X, d_i \neq d : \alpha(ID_d^X) \cdot \alpha_{marg} > \alpha(ID_{d_i}^X)$, accept d as authentic, if also criterion 2 for d holds. If $|D_X| = 1$, i.e., there is only one device d claiming the identity X, d is accepted as authentic, if $\alpha(ID_d^X) > \alpha_{min}$, and criterion 2 holds for it.*

Threshold α_{min} determines the minimal authenticity rating required for a peer to be considered genuine. Margin factor $\alpha_{marg} \in [0,1]$ determines, how much the authenticity rating of a correct peer has to dominate over the authenticity ratings of all other peers in order for it to be considered genuine.

The second criterion requires that the key evolution with a device needs to be attempted at least ρ_{min} times, before the authenticity rating can be regarded as representative. In addition, the number of key evolution cycles during which the key evolution is attempted needs to cover at least a fraction of ρ_{cov} of the total key evolution cycles after the initial pairing. Otherwise, an adversary $\mathcal{A}$ in the wearable scenario could just selectively attempt key evolution only when d_1 is in its context and thereby slowly accumulate a high authenticity rating even though it only occasionally shares the same context with d_1.

CRITERION 2 (CONFIDENCE). *Let ρ_d denote the number of key evolution cycles during which device d has attempted key evolution and ρ_d^* the total amount of key evolution cycles since establishing the initial pairing for d. Let also $\rho_{min} \in \mathbb{N}^+$ denote a key evolution attempt threshold and $\rho_{cov} \in [0,1]$ a key evolution coverage threshold. The pairing of a device d is accepted as genuine only if $\rho_d > \rho_{min}$ and $\frac{\rho_d}{\rho^*_d} > \rho_{cov}$.*

Once a device d's pairing is accepted as genuine, there are two options: other key chains may be removed and pairing stopped (e.g., when a smartwatch has found its host smartphone), or, the accepting device may continue to evolve pairing keys for other devices (e.g., in the case of smart TV that can accommodate multiple remote controls).

4. ROBUST CONTEXT FINGERPRINTS

We apply our context fingerprinting method on two different contextual modalities: ambient noise and light. As mentioned before, the fingerprinting scheme is inspired by Schürmann and Sigg [16], but it is different in several ways: The scheme in [16] requires tight time synchronization, whereas our scheme does not. Their scheme is intended to extract enough entropy within a very short time to be used as a cryptographic key, whereas our fingerprints have a longitudinal orientation. Finally, our scheme is equally applicable to both audio and luminosity and the fingerprints represent more sustained changes in the contextual characteristics of the ambient context over several hours. Thereby, the fingerprints will also capture phenomena originating from the user's actions (such as switching on the lights, chatter, silence, etc.). These events are inherently random and therefore difficult to predict even for advanced attackers that may try to utilize profiled information about the target context in attempting to fabricate context fingerprints.

Using a longitudinal approach in fingerprint generation and key evolution has also the advantage that the scheme is more robust against attackers that are occasionally co-located with the paired devices. This is different to earlier approaches, where the security of the pairing is dependent on the fact that the attacker is not sharing the context with the paired devices at the time of pairing [20, 16]. Our longitudinal approach, on the other hand, can gracefully handle situations in which the attacker is occasionally in the same context with the paired devices, as we will show in Sect. 5.

In our scheme, the devices are continuously monitoring their context by scanning context snapshots $c_w(t)$. Every f seconds, a snapshot of w seconds is recorded. Each snapshot consists of a sequence of measurements m_i in a particular contextual modality like ambient luminosity or noise level, such that $c_w(t) = (m_i, m_{i+1}, \ldots, m_{i+n})$, where the timestamp associated with an individual measurement m_i is denoted with $t(m_i)$, and, $t(m_{i+n}) - t(m_i) = w$. Since the used snapshot length w is fixed and usually clear from the context, we omit it in the following and denote a context snapshot just with $c(t)$ for better readability.

We average the measurements within each context snapshot $c(t)$ and denote the snapshot's average value as

$$\bar{c}(t) = \frac{\sum_{m_i \in c(t)} m_i}{|\{m_i \in c(t)\}|}, \tag{2}$$

where $|\cdot|$ denotes set cardinality.

Based on a sequence of context snapshots $C(t, t+nf) = (c(t), c(t+f), c(t+2f), \ldots, c(t+nf))$, we calculate its context fingerprint as a sequence of bits, in which each bit denotes the change of the snapshot's average value in comparison with the previous snapshot's average. The fingerprint bit corresponding to a context snapshot is set to "1" if the relative change between the snapshot's average value and the previous snapshot's average value is larger than a specified relative threshold Δ_{rel} and if the difference between the values exceeds an absolute threshold value Δ_{abs}. Otherwise, the bit is "0".

DEFINITION 2. *Let $C(t, t+nf)$ be a sequence of context snapshots, i.e., $c(t_i) \in C(t, t+nf), t < t_i \leq t+nf, n \in \mathbb{N}^+$. We define the* fingerprint bit $b(t_i)$ *corresponding to each snapshot $c(t_i)$ as*

$$b(t_i) = \begin{cases} 1, & |\frac{\bar{c}(t_i)}{\bar{c}(t_i - f)} - 1| > \Delta_{rel} \wedge |\bar{c}(t_i) - \bar{c}(t_i - f)| > \Delta_{abs} \\ 0, & \textit{otherwise}. \end{cases} \tag{3}$$

DEFINITION 3. *We define the* fingerprint $\phi(C(t, t+nf))$ *of a sequence of context snapshots $C(t, t+nf), n \in \mathbb{N}^+$ as*

$$\phi(C(t, t+nf)) = (b(t), b(t+f), \ldots, b(t+nf)). \tag{4}$$

The rationale for our notion of fingerprints is that two devices that share the same context for an extended period of time will also experience changes in context parameters in a similar way. For example, if the user switches on the lights in a room, the increase in luminosity in the room will be sensed by all devices located inside the room, whereas other devices not in the same room will not be able to sense it. Therefore, bits generated this way will be shared only with the co-located devices. The same applies to fingerprints based on audio. The alternating patterns between chatter, silence and possible other persistent ambient sounds will generate

fingerprint bits in a way that is similar between devices in the same audio context (e.g., the same room). Devices outside the audio context will, however, not be able to sense these changes.

The same logic applies also to mobile personal devices like wearables, which are usually always carried together. Even though the context in which the devices are located may change as the user moves, the changes will be sensed in a similar way by both devices.

We will evaluate our fingerprint extraction scheme in both static and mobile scenarios in Sect. 4.1.

4.1 Implementation and Evaluation

To analyze the feasibility of our approach, we performed several experiments in different contexts investigating, how similar fingerprints extracted from ambient luminosity and noise levels are in real contextual settings.

4.1.1 System Set-Up

To simulate the capability of IoT and wearable devices to sense their ambient context and to use the context information for key evolution, we used Android OS smartphones (Samsung Galaxy Nexus, Nexus S and Galaxy S III devices) running dedicated context data collection software. The collection software on each device continuously measured the luminosity and noise levels in the device's context and routinely sent the collected data to a server for off-line data analysis. In these experiments we used a static placement of the test devices to simulate IoT device pairing scenarios, whereas for personal wearable pairing scenarios test persons carried the data collection devices with them.

In both settings, the orientation of the luminosity sensors of the devices impacts the magnitude of observed luminosity readings. However, since our method for deriving fingerprint bits from luminosity readings is not based on absolute luminosity values, but on relative changes in the ambient illumination, the exact placement and orientation of the devices plays only a minor role.

4.1.2 IoT Scenario

In this scenario, we investigated whether IoT devices located in the same room can successfully establish similar enough fingerprints to be used for context-based key evolution. We tested the scenario in different set-ups and locations over several months, varying the placement of the devices with regard to each other and within the room. Table 1 shows one example of the placement of devices in two settings at two different locations: office and home.

In the office setting, two devices simulating correct peers were placed on the wall of an office room, three meters apart from each other. Other smartphones simulating wrong peers were placed in nearby rooms, but without direct visibility to the room with the correct peers. In the home setting, the correct peers were placed in the living room of the test participant's house. A smartphone simulating a wrong peer in a neighboring apartment was placed in another room of the house, but on a different floor.

To eliminate effects that possible differences in the orientation of windows of the rooms could have on lighting conditions, we selected rooms that had relatively large windows facing the same direction, allowing outdoor light to illuminate all rooms used in the experiment in a similar way during daytime. In addition, to obtain a baseline measurement of the outdoor lighting conditions that affect the illumination of the room with the correct peers, we dedicated in each scenario one device for measuring the direct outdoor light falling into the room.

Table 1: Placement of test devices in an IoT scenario

Device	Placement
Office setting	
Device d_1	User's office
Device d_2	User's office
$\mathcal{A}_1$	Outdoor light
$\mathcal{A}_2$	Adjacent office
$\mathcal{A}_3$	Coffee room, one room apart
Home setting	
Device d_1	Living room, ground floor
Device d_2	Living room, ground floor
$\mathcal{A}_1$	Outdoor light
$\mathcal{A}_2$	Studio, 2^{nd} floor

Table 2: Average fingerprint similarity between the co-located and adversary devices in the IoT scenario

Average fingerprint similarity with co-located devices	Luminosity	Audio
Office setting, 8 a.m. to 6 p.m.		
d_1 and d_2	95.0 %	91.8 %
$\mathcal{A}_1$	70.0 %	-
$\mathcal{A}_2$	88.7 %	71.7 %
$\mathcal{A}_3$	68.3 %	62.6 %
Home setting, 6 a.m. to 10 p.m.		
d_1 and d_2	82.9 %	87.5 %
$\mathcal{A}_1$	70.8 %	-
$\mathcal{A}_2$	70.6 %	77.0 %

Results. We collected luminosity and audio measurements during the course of several weeks. We extracted context-based fingerprints based on a time window of $w = 120$ seconds for each device and compared the average bit differences of the fingerprints of correct and wrong peers. In both settings, hardly any bits were generated during nighttime. We will show in Sect. 5 that fingerprints generated from nighttime data contain only very little entropy and can therefore not be used for fingerprint generation. Therefore, we concentrate our analysis in the office setting during business hours between 8 a.m. and 6 p.m. and in the home setting during active hours of a household between 6 a.m. and 10 p.m. The results are shown in Tab. 2.

In the office setting, the co-located devices clearly show the largest bit similarity in their respective fingerprints. For the luminosity data, the difference between the co-located devices d_1 and d_2 and the adversary device $\mathcal{A}_2$ in the adjacent office is relatively small, i.e., only 6.3%. This is so because the lighting conditions affecting the rooms are almost identical and the effect of sunlight dominates the overall lighting conditions during business hours[1].

For audio, the differences are clearer. Adversary $\mathcal{A}_2$ in the adjacent office has only 71.7 % similarity compared to 91.8 %

[1]The measurements were done less than two months from the summer solstice in the northern hemisphere, i.e., the brightest time in the year. The influence of sunlight is likely to be be smaller during other times of the year.

for the co-located devices d_1 and d_2. This is so, even though the doors of the rooms in question to a common hallway were mostly kept open, so that some parts of the acoustic environment could be shared by the devices in these rooms. However, adversary device $\mathcal{A}_3$ located in the coffee room was farther away, so that it was acoustically more clearly decoupled from the co-located devices. Therefore the similarity percentage of its fingerprints to the fingerprints of the co-located devices is significantly lower, i.e., 62.6 %.

In the home setting, the results were similar. Here, the similarity between co-located devices was on the average 82.9 % for luminosity and 87.5 % for audio. There was also a clear difference to the adversary devices, which could only achieve bit similarity values of 70.8 % for luminosity and 77.0 % for audio fingerprints.

4.1.3 Wearable Device Scenario

In this scenario, we simulated the contextual environment that typical wearable devices are confronted with. We did this by equipping test users with smartphones, each playing the role of a wearable device. We considered two alternative settings: a 'smart watch' scenario, in which one device plays the role of a smart watch, and the other device is used like a regular smartphone. The other, 'cycling' scenario, simulates the use of wearable devices as fitness gadgets.

In the smart watch scenario, users were equipped with two smartphones which they carried with them continuously. One of the devices simulated a smart watch that is worn on the user's wrist. It was therefore placed in a translucent carrying pouch so that its light sensor was constantly exposed to the ambient light. The other device was used like a regular smartphone.

In the 'cycling' scenario, we used two smartphones to simulate wearable fitness gadgets, currently one of the most popular classes of wearable devices. In our scenario, we considered a cyclist, who is using a heart rate monitor to record his physical performance and a near-eye display device to visually follow the key characteristics of his workout, including the heart rate. The smartphone playing the role of a near-eye display device was attached on the side of the bicycle helmet of the cyclist, with the light sensor showing outwards. The other device played the role of the smart heart rate sensor. It was placed in a translucent carrying pouch on the chest of the cyclist, facing forward, which is also a typical placement for heart rate sensors. In the cycling scenario, ambient light and noise data were collected during the workouts of the cyclist.

Results. We collected traces from co-located devices carried by test persons in a number of mobile and static contexts: walking, in public transport, as well as stays in the home and office contexts. Since the mobility of the user introduces a significant amount of changes into the devices' contexts, the bit similarity of fingerprints from the co-located devices was relatively high, 92.6 % on average (minimum 87.3 %, maximum 96.7 %). This provides a good basis for successful key evolution between the co-located devices. In the wearable device scenario, however, also the presence of wrong peer devices in the context plays a role. We analyze the effect of such devices on the key evolution scheme in more detail in Section 5.1.

For the cycling scenario we collected 10 traces of context measurements captured along a back-and-forth journey on a fixed route of approximately 10 miles. The exercises were spread out over several weeks, encompassing varying road and weather conditions ranging from rainy, overcast to sunny days. Since the contextual environment changes in this scenario much faster than in the static scenario, we chose a shorter time window $w = 5$ sec and higher sampling rate $f = 5$ sec for luminosity-based fingerprint generation. Using this fingerprinting scheme, the fingerprints for the exercises contained 665 to 784 bits. For audio data, a slightly longer time window of $w = 6$ sec was used, giving us fingerprints of 501 to 550 bits. The bit similarity between the fingerprints of the co-located devices d_1 and d_2 was on the average 68.6 % for luminosity-based fingerprints (minimum 62.8 %, maximum 74.5 %) and 65.9 % for audio-based fingerprints (minimum 63.6 %, maximum 67.1 %).

4.1.4 Context Replay Attacks

In addition to testing the bit similarities of co-located devices we also examined the effect of context *replay attacks* by analyzing whether an attacker, who knows the route that a user is going to use could record the context parameters along this route and use this recording to produce a context fingerprint that could fool a target device into believing that the attacker has been sharing the same context. We therefore used the set of fingerprints from the cycling scenario generated on different days on the same route and measured their bit similarity. We did this in order to find out what an attacker in the optimal (worst) case could achieve. We iterated over all exercise fingerprints, using one fingerprint at a time as the target device d_1's fingerprint and the remaining fingerprints as fingerprints of the adversary $\mathcal{A}$. Since the fingerprints were recorded at different times, it was not clear how to optimally align them. We therefore calculated the bit difference for each target-attacker fingerprint pair for all possible overlapping alignments of the fingerprints and used the minimal bit difference to choose the optimal alignment. We then averaged the bit similarity values over the adversarial fingerprints with optimal alignments.

The fingerprint similarity of simulated replay attacks with the target devices was 59.5 % (minimum 55.9 %, maximum 62.3 %) for luminosity-based fingerprints and 56.4 % (minimum 55.0 %, maximum 56.8 %) for audio-based fingerprints. The margin between the actually co-located device pair d_1 and d_2 to the replayed adversarial fingerprints was on the average 52.5 bits (minimum 4 bits, maximum 104 bits) for luminosity and 42.8 bits (minimum 27 bits, maximum 51 bits) for audio in favor of the co-located pair.

The clear margins between the bit similarities of co-located and attacker fingerprints suggest that it's in most cases possible to define a parameter value t for the fuzzy commitment scheme so that co-located peers will be able to successfully perform key evolution steps, while blocking most attackers from doing so. The analysis here also involves two rather optimistic assumptions in favor of the attacker, namely that the attacker is able to record the trace in exactly the same way as the targeted user and that the attacker is able to guess the optimal alignment of his fingerprint with the target user's fingerprint. In practice, it is unlikely that an attacker would be able to always guess the optimal alignment, or record context traces with the same rhythm and speed as the targeted users. Hence in practice the margins in favor of the correct peers will be much larger than the ones presented above.

5. SECURITY ANALYSIS

In our analysis we assume that the legitimate personal devices of the user have not been compromised and execute context sensing, key agreement, key evolution and authentication protocols as specified.

Let us consider an adversary $\mathcal{A}$ impersonating X, having a key chain identity $ID^X_{\mathcal{A}}$. The only way for $\mathcal{A}$ to get its pairing with a target device d_1 accepted as genuine is to achieve a high enough authenticity rating $\alpha(ID^X_{\mathcal{A}})$. To do this, it needs to successfully participate in the key evolution process. $\mathcal{A}$ can do so in two cases:

1. $\mathcal{A}$ is in the same context as d_1 and can generate context fingerprints $F_{C_{\mathcal{A}}}$ sufficiently similar to the fingerprint $F_{C_{d_1}}$ of d_1, i.e., $\texttt{Ham}(F_{C_{d_1}}, F_{C_{\mathcal{A}}}) \leq t$, or,
2. $\mathcal{A}$ is not in the same context as d_1 but is able to *fabricate* context fingerprints $F'_{C_{\mathcal{A}}}$ sufficiently similar with the fingerprint of d_1, i.e., $\texttt{Ham}(F_{C_{d_1}}, F'_{C_{\mathcal{A}}}) \leq t$.

We will first analyze the effect of $\mathcal{A}$ being in the same context with d_1 on his authenticity rating, and then analyze the success probability of the attacker fabricating context fingerprints.

5.1 Attacker in Same Context as Target

Let us denote with θ the probability that $\mathcal{A}$ is able to extract a fingerprint $F_{C_{\mathcal{A}}}$ having $\texttt{Ham}(F_{C_{d_1}}, F_{C_{\mathcal{A}}}) \leq t$ and with β the probability that a co-located device d_2 extracts a fingerprint $F_{C_{d_2}}$ having $\texttt{Ham}(F_{C_{d_1}}, F_{C_{d_2}}) \leq t$.

If $\mathcal{A}$ is present in the same context as d_1, $\mathcal{A}$ can extract fingerprints $F_{C_{\mathcal{A}}}$ that have the same probability to have a bit difference of t or lower than the co-located device d_2. Therefore, $\theta = \beta$. However, to simplify the analysis, let us assume a perfect attacker that *always* succeeds in key evolution when it is in the context, i.e., $\theta = 1$.

We denote with n the number of key evolution steps that a correct peer d_2 will attempt during a specific time period and with m the number of key evolution steps that the wrong peer $\mathcal{A}$ will attempt during the same time. To maximally increase its authenticity rating, $\mathcal{A}$ will attempt to do key evolution steps every time it is co-located with the target device in the same context. When $\mathcal{A}$ is not in the same context, the probability to successfully evolve the key is less than β. Therefore $\mathcal{A}$ will not participate in key evolution. Since d_2 is a benign peer, it will regularly attempt to evolve its pairing key with d_1 each time it is co-located with it.

According to Def. 1 the authenticity rating $\alpha(ID^X_{d_2})$ of the correct peer d_2 will be higher, if it has performed more successful key evolution steps than the attacker $\mathcal{A}$, i.e., if $\gamma(ID^X_{d_2}) > \gamma(ID^X_{\mathcal{A}})$. Since $\gamma(ID^X_{d_2}) = \beta \cdot n$ and $\gamma(ID^X_{\mathcal{A}}) = \theta \cdot m = m$, and, if $\beta \cdot n > m$ holds, then attacker $\mathcal{A}$ will never be able to obtain an authenticity rating that is higher than the rating of the correct peer. However, since we assume that $n >> m$, and, in particular $\beta \cdot n > m$, it is clear that the attacker will not succeed in getting his pairing accepted as genuine in our scheme.

5.2 Attacker not in Same Context as Target

The fuzzy commitment scheme used in the key evolution protocol provides the hiding property as mentioned in Sect. 3. $\mathcal{A}$ will not be able to reveal the correct key derivation key K_r and thus participate successfully in the key evolution protocol if it can't find a context fingerprint $F'_{C_{\mathcal{A}}}$ that has a Hamming distance of at most t bits to the fingerprint $F_{C_{d_1}}$ of the targeted device d_1. We focus our analysis therefore on examining whether an attacker is able to fabricate fingerprints $F'_{C_{\mathcal{A}}}$ satisfying this criterion.

When the attacker $\mathcal{A}$ has no access to actual context measurements based on which $F_{C_{d_1}}$ is extracted, $\mathcal{A}$ has the following options for fabricating the fingerprint $F'_{C_{d_1}}$: a random guess, a profiling-based guess, or, the use of partial information.

Random Guess.

In a random guess, the probability to guess one bit correctly is 0.5. Consequently, for a fingerprint of length k, the likelihood for a successful guess is therefore 2^{-k}. The success probability is negligibly small for typical fingerprints of tens or hundreds of bits. For example, using a fingerprinting window of $w = 120$ sec and fingerprinting periods of two hours, one already gets fingerprints of 60 bits, which would be excessively difficult to guess with random guesses.

Profiling-Based Guesses.

An obvious improvement to this attack would be to use profiled information about the distribution of bits to improve $\mathcal{A}$'s chances to fabricate valid fingerprints $F'_{C_{\mathcal{A}}}$. Depending on the used fingerprinting parameters, the type of context in question as well as the time of the day, the distribution of fingerprint bits in the fingerprint changes. For example, during nighttime, when it typically is silent and dark, and thus no significant changes in the context parameters take place, an overwhelming majority of bits in d_1's fingerprint $F_{C_{d_1}}$ will be "0" bits, with only a few "1" bits (if any) in-between. Fig. 4 shows for the office context and the audio modality examined in our evaluation experiments the distribution of "1" vs. "0" bits changing according to the time of day. If $\mathcal{A}$ can obtain such profile information about a context where d_1 is going to be, $\mathcal{A}$ can utilize the profiled information in fabricating fingerprints $F'_{C_{\mathcal{A}}}$ that are more likely to have a lower Hamming distance $\texttt{Ham}(F_{C_{d_1}}, F_{C_{\mathcal{A}}})$ to the fingerprint $F_{C_{d_1}}$ of device d_1 extracted in that context.

The strength of a fingerprint against profiling-based guessing attacks can be analyzed by looking at the *surprisal* associated with individual bits $b \in F_{C_{d_1}}$ using a frequentist interpretation of probability. We do this by calculating the occurrence probability of a particular bit b in the fingerprint during a specific time of the day. Thus, the probability of a particular bit (i.e., "1" or "0") is equal to the fraction of that bit's occurrences in the context fingerprints during that time of the day. Given the occurrence probability of a particular bit, we define the surprisal associated with individual bits as follows.

DEFINITION 4 (SURPRISAL σ OF A FINGERPRINT BIT). *Let B be a random variable modelling the occurrence of a bit as a fingerprint bit in fingerprint F. The* surprisal σ *associated with the occurrence of a fingerprint bit $b \in \{0, 1\}$ is the self-information of this bit $\sigma(b) = I(b) = log(\frac{1}{P(B=b)}) = -log(P(B = b))$, and is measured in bits.*

DEFINITION 5 (SURPRISAL OF A FINGERPRINT). *The surprisal $\sigma(F)$ of a fingerprint F is the sum of the surprisal values of its individual bits, i.e.,*

$$\sigma(F) = \sum_{b \in F} \sigma(b).$$

For example, we can see based on Fig. 4 that context measurements made during the night are not suitable for generating hard-to-guess context fingerprints. This is because, e.g., during the time window of 2 a.m. to 4 a.m. on the average only 1 % of the extracted fingerprint bits are "1" bits. Each of these bits has a surprisal value of 6.3 bits per fingerprint bit, but the remaining "0" bits only have a surprisal of 0.02 bits per fingerprint bit. This means that, e.g., a 60-bit fingerprint extracted during this time frame would have only 5.8 bits of total surprisal on the average. In other words, the attacker would have a $2^{-5.8} \approx 1.8\%$ chance of guessing the exactly correct context fingerprint by utilizing profiling information.

However, since we are using a fuzzy commitment scheme, the attacker $\mathcal{A}$ does not even have to guess the exact fingerprint. Any fingerprint $F'_{C_\mathcal{A}}$ that is within Hamming distance t of the fingerprint $F_{C_{d_1}}$ of target device d_1, will enable the attacker $\mathcal{A}$ to open the commitment and retrieve the correct key derivation key K_r.

From the evaluation results (cf. Tab. 2) we can see that in the office environment, and for the audio modality, the context fingerprints of co-located devices will deviate on the average in ca. 10 % of the bits. In order to let correct peers perform key evolution successfully, we need to tune the parameter t of the fuzzy commitment scheme so that it allows fingerprints deviating in 10 % of the bits to still open the fuzzy commitments. Coming back to our example of Fig. 4, this would mean that for fingerprints of 60 bits we would need to set $t = 6$ bits. $\mathcal{A}$ could use this to his advantage and fabricate during nighttime context fingerprints $F'_{C_\mathcal{A}} = \{0\}^{60}$ containing nothing but "0" bits. The fuzzy commitment scheme would correct the errors this fingerprint has with regard to d_1's fingerprint $F_{C_{d_1}}$, since it on the average contains less than $t = 6$ "1" bits during the night. The attacker $\mathcal{A}$ could thus successfully evolve the key nearly every time just by targeting fingerprints generated from nighttime context data.

To thwart such attacks against fingerprints with very low surprisal values, we need to add an additional requirement: only fingerprints F_{C_d} having sufficient total surprisal may be taken into account when evaluating authenticity ratings.

REQUIREMENT 1 (SURPRISAL THRESHOLD σ_{thr}). *When d_1 calculates the number of successful key evolution steps with another device d, only such evolution steps may be taken into account that have been based on fingerprints $F_{C_{d_1}}$ having a surprisal value $\sigma(F_{C_{d_1}}) > t + \sigma_{marg}$.*

Here σ_{marg} denotes a surprisal margin required in addition to the bits that the fuzzy commitment scheme will correct. In effect, σ_{marg} defines, how hard it is for an attacker to guess a fingerprint $F'_{C_\mathcal{A}}$ that is required for successful key evolution.

Use of Partial Information.

In addition to profiling-based guesses, the attacker $\mathcal{A}$ may be in the position to utilize partial information about the context fingerprint $F_{C_{d_1}}$ of the target device d_1. Such partial information may be available to the attacker based on the fact that the contextual separation between the attacker's context and the target context, where d_1 is located, is not complete. In the case of the luminosity, this may be caused by the fact that outdoor light is influencing the lighting conditions in both the target device d_1's context, as well as the context of the attacker $\mathcal{A}$. In the case of audio, partial information may be because of acoustic events that are heard in both contexts.

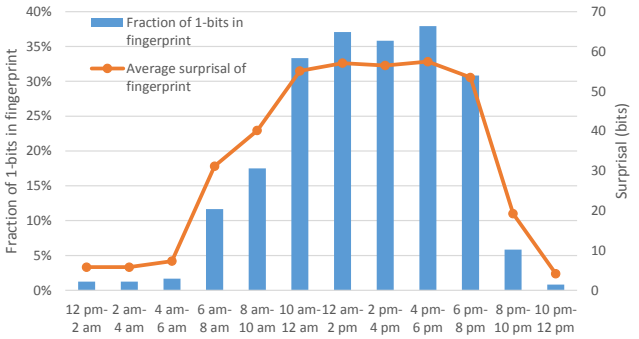

Figure 4: Distribution of bits and surprisal of fingerprints in office context depending on time of day (audio)

The existence of such partial information has the effect that the fingerprints $F_{C_{d_1}}$ and $F_{C_\mathcal{A}}$ share common bits attributable to this partial information. The effect of the partial information is significant. If one looks at the bit similarities of adversarial devices to the co-located ones, we can see that the attacker devices share ca. 65 - 85 % of common bits, depending on the placement of the attacker devices.

The partial information plays therefore in the attacker's favor. If $\mathcal{A}$ can assume that his fingerprint $F_{C_\mathcal{A}}$ contains partial information about the target device d_1's fingerprint $F_{C_{d_1}}$, $\mathcal{A}$ can use its own fingerprint $F_{C_\mathcal{A}}$ as a basis for fabricating a fake fingerprint $F'_{C_\mathcal{A}}$. If we denote the bit difference of $\mathcal{A}$'s fingerprint $F_{C_\mathcal{A}}$ and the target fingerprint $F_{C_{d_1}}$ with t', then $\mathcal{A}$ needs to guess only $\Delta t = t' - t$ bit modifications to $F_{C_\mathcal{A}}$ correctly to fabricate a fake fingerprint $F'_{C_\mathcal{A}}$ having $\texttt{Ham}(F_{C_{d_1}}, F'_{C_\mathcal{A}}) \leq t$, and thus allowing $\mathcal{A}$ to participate successfully in the key evolution.

Since the $\mathcal{A}$ does not know which of the bits in $F_{C_\mathcal{A}}$ differ from d_1's fingerprint $F_{C_{d_1}}$, $\mathcal{A}$ needs to guess a set of at least Δt bit positions to correct in $F_{C_\mathcal{A}}$ in order to obtain a fingerprint $F'_{C_\mathcal{A}}$ having $\texttt{Ham}(F_{C_{d_1}}, F'_{C_\mathcal{A}}) \leq t$. This means he needs to select Δt bits from the total set of t' bits differing with $F_{C_{d_1}}$ and flip them. $\mathcal{A}$ can select these with a probability of

$$P(\Delta t \textit{ successful corrections}) = \frac{\binom{t'}{\Delta t}}{\binom{|F_C|}{\Delta t}}, \tag{5}$$

where $|F_C|$ denotes the bit length of the used fingerprints.

Consider as an example 360-bit fingerprints corresponding to, e.g., 6 hours of 1-minute observations. Assume that $\mathcal{A}$ has a fingerprint $F_{C_\mathcal{A}}$ that has a bit difference of 15 %, i.e., $t' = 54$ bits. Assume also that the fuzzy commitment scheme corrects up to 10 % of bit differences, i.e., $t = 36$ bits. How difficult is it for $\mathcal{A}$ to guess a fingerprint $F'_{C_{d_1}}$ with $\texttt{Ham}(F_{C_{d_1}}, F'_{C_\mathcal{A}}) \leq t = 36$? To do this, the attacker would need to correct $\Delta t = 18$ bits. We can calculate the success probability for $\mathcal{A}$ as $\frac{\binom{54}{18}}{\binom{360}{18}} \approx 9.27 \times 10^{-17}$. This is equivalent to ca. 53 bits of entropy and demonstrates that guessing correct fingerprints will be excessively difficult for

the attacker, if the used fingerprints are long enough. Note that the length of the used fingerprints can be freely chosen depending on the security requirements of a specific use case. The only limiting factor is the time required to acquire the context measurements for generating the fingerprints.

Some of the changes in d_1's ambient context, especially in the ambient luminosity, can originate from environmental changes that can also be observable by the attacker $\mathcal{A}$ in a close-by room (e.g., if direct sunlight is suddenly obscured by a cloud). $\mathcal{A}$ could utilize this information to give more confidence to bits b in its fingerprint $F_{C_\mathcal{A}}$ that $\mathcal{A}$ knows to originate from such environmental events. Thus, he could limit the search space of bit positions to be flipped to fabricate $F'_{C_\mathcal{A}}$, thus decreasing the *effective* length $|F_C|$ of the fingerprint in Eq. 5 and thereby improving his chances for success. However, in our attacker model, $\mathcal{A}$ is an off-the-shelf IoT device, and does in general not have the technology to interpret the causes behind changes in sensor readings in an automated way. Therefore it wouldn't be straightforward for $\mathcal{A}$ to distinguish which changes in the sensor readings are caused by such changes in the environment that are observable also in d_1's context and which are not. On the other hand, should such technology become available in the future, it could not only be used by $\mathcal{A}$ to improve its guesses, but also by d_1 to defend against guessing. While generating its fingerprint, d_1 could keep track of the number of fingerprint bits $b \in F_{C_{d_1}}$ that were influenced by changes in the environment outside of its proximate context. The target device d_1 could then disregard such key evolution steps, for which the number of influenced fingerprint bits is too high.

6. RELATED WORK

There are various approaches proposed to establish a secure pairing between devices. These approaches can be broadly divided into two main categories: utilizing key predistribution mainly addressing nodes in digital sensor networks (DSN), and utilizing context information for key establishment or co-presence verification.

Key predistribution-based approaches. A scheme for key distribution in DSNs based on predistributing keys to nodes was presented by Eschenauer and Gligor [6]. Their scheme ensured that when deployed, each sensor node shares a key with a neighboring node. Chan et al. [2] extend this basic scheme and design three enhanced key pre-distribution schemes: the q-composite scheme, multipath reinforcement scheme, and, random pairwise key predistribution scheme. Liu et al. [10] propose key predistribution schemes based on preassignment of polynomial shares to sensor network nodes: a random subset assignment scheme and a hypercube-based key predistribution scheme.

Traynor et al. [18] extend the key predistribution schemes by removing the assumption of homogeneous sensor nodes and key predistribution by introducing unbalanced probabilistic key distribution. They also extend their approach to hybrid settings in which key distribution centers (KDC) may be available.

However, all of the above schemes are mainly targeted at DSNs deployed in a geographically limited area. Hence they are as such not applicable nor scalable in practice to our setting, which involves arbitrary subsets of devices coming from a pool of potentially millions of IoT and wearable devices deployed anywhere on the planet. Also, in contrast to DSNs that typically are deployed by a single or a few organizations sharing trust associations, IoT and wearable devices are expected to come from hundreds if not thousands of different manufacturers. It is highly unlikely that all potential IoT and wearable device vendors would share mutual security associations necessary for pre-keying of devices. These factors make any solutions based on predistributing keys between devices infeasible to deploy in practice.

Therefore, our approach presented in this paper does not utilize key predistribution to devices, but builds on utilizing ambient context information for evolving a pairing key between devices consistently sharing the same context.

Context information-based approaches. Varshavsky et al. [20] proposed to use the fluctuations in the received signal strength of WiFi broadcast packets for verifying the immediate proximity of the to-be-paired parties. This approach is, however not suitable for IoT scenarios: it is unlikely to work in situations in which peers are located farther away from each other than one meter, due to the local nature of the fluctuations in WiFi signals. The authors also acknowledge that it does not protect against man-in-the middle attacks that are mounted by an attacker immediately behind a wall to the user's location, since WiFi signals are unaffected by some wall materials.

Narayanan et al. [11] propose a similar approach, in which WiFi broadcast packets are monitored to determine *location tags* that peers can compare to determine whether they are co-located or not. Their solution addresses, however, the problem of privacy-preserving determination of co-location and does not address the problem of pairing previously unknown peers with each other, whereas we explicitly address the problem of pairing personal devices.

Schürmann and Sigg [16] propose to use audio for generating a shared secret between co-located peers to be used as a pairing key. They record audio samples and calculate audio fingerprints based on them. Using Reed-Solomon encoding for fuzzy extraction of a common key they show that in various audio environments, cryptographic keys can be derived from the surrounding audio context. Their approach attempts to extract a large amount of entropy from a short audio snapshot and requires therefore very exact temporal alignment of the sound samples, which is difficult to achieve with off-the shelf devices. Our approach is different, since our method does not require exact temporal alignment, and it operates on longitudinal data, extracting entropy from the user's context over a longer period of time. Contrary to the approaches in [20, 11, 16], our approach can also handle situations in which an adversary is present in the correct peers' context without breaking the authenticity of the pairing.

The problem of zero-interaction authentication utilizing contextual proofs of presence has been discussed by Truong et al. [19]. While their work also addresses a zero-interaction scenario, their problem is different: they consider the problem of co-location verification using context information in a setting, in which both endpoints are trusted and already have an established security association, whereas our approach addresses the problem of pairing devices that do not have any prior security associations with each other.

7. CONCLUSION

We have presented a novel key evolution approach for pairing personal IoT and wearable devices. The approach builds

on a robust scheme for extracting shared entropy from the ambient context of such devices. We have also evaluated the approach based on experiments with luminosity and ambient noise in a number of different environments. These results should be understood as indicative, primarily establishing the overall feasibility of our proposed approach. Currently we are working on more large-scale testing in different scenarios and different contexts, which we think is of importance for further research in this area.

Acknowledgments

This work was supported in part by the Intel Institute for Collaborative Research in Secure Computing (ICRI-SC) and the Academy of Finland ("Contextual Security" project).

We thank Jan-Erik Ekberg for originally suggesting the idea of strengthening a shared key between two devices *over time* [5]. We would also like to thank our shepherd Florian Kerschbaum and the anonymous reviewers for their insightful feedback.

8. REFERENCES

[1] D. Balfanz, D. K. Smetters, P. Stewart, and H. C. Wong. Talking to strangers: Authentication in ad-hoc wireless networks. In *Proc. Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, Feb. 2002.

[2] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Proc. 2003 IEEE Symposium on Security and Privacy*, pages 197–213, May 2003.

[3] Y. Dodis, J. Katz, L. Reyzin, and A. Smith. Robust fuzzy extractors and authenticated key agreement from close secrets. In C. Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 232–250. Springer Berlin Heidelberg, 2006.

[4] D. Dolev and A. C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, Mar 1983.

[5] J.-E. Ekberg. Key establishment in constrained devices. graduate seminar paper in T-110.7290 - Research Seminar on Network Security, Oct. 2006. `http://www.tcs.hut.fi/Studies/T-79.7001/2006AUT/seminar-papers/Ekberg-paper-final.pdf`.

[6] L. Eschenauer and V. D. Gligor. A key-management scheme for distributed sensor networks. In *Proc. 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 41–47, New York, NY, USA, 2002. ACM.

[7] Gartner. Gartner says by 2017, mobile users will provide personalized data streams to more than 100 apps and services every day, Jan. 2014. `http://www.gartner.com/newsroom/id/2654115` [Referenced 2014-04-28].

[8] Gartner. Gartner says the internet of things installed base will grow to 26 billion units by 2020, 2014. `http://www.gartner.com/newsroom/id/2636073` [Referenced on 2014-04-28].

[9] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, 2006.

[10] D. Liu, P. Ning, and R. Li. Establishing pairwise keys in distributed sensor networks. *ACM Trans. Inf. Syst. Secur.*, 8(1):41–77, Feb. 2005.

[11] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, and D. Boneh. Location privacy via private proximity testing. In *Proc. Network and Distributed System Security Symposium (NDSS)*, San Diego, CA, USA, Feb. 2011.

[12] Nest Labs. Nest thermostat and nest smoke and CO alarm, 2014. `http://nest.com/` [Referenced on 2014-04-28].

[13] Oral-B. ORAL-B® debuts world's first available interactive electric toothbrush at mobile wold congress 2014, 2014. `http://connectedtoothbrush.com/` [Referenced 2014-04-28].

[14] Quirky. Spotter multipurpose sensor, 2014. `https://www.quirky.com/shop/609-spotter-multi-purpose-sensor` [Referenced 2014-04-28].

[15] I. Reed and G. Solomon. Polynomial codes over certain finite fields. *Journal of the Society for Industrial and Applied Mathematics*, 8(2):300–304, 1960.

[16] D. Schürmann and S. Sigg. Secure communication based on ambient audio. *IEEE Transactions on Mobile Computing*, 12(2):358–370, Feb 2013.

[17] J. Suomalainen, J. Valkonen, and N. Asokan. Security associations in personal networks: A comparative analysis. In F. Stajano, C. Meadows, S. Capkun, and T. Moore, editors, *Security and Privacy in Ad-hoc and Sensor Networks*, volume 4572 of *Lecture Notes in Computer Science*, pages 43–57. Springer Berlin Heidelberg, 2007.

[18] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, and T. La Porta. Efficient hybrid security mechanisms for heterogeneous sensor networks. *IEEE Transactions on Mobile Computing*, 6(6):663–677, June 2007.

[19] H. T. T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, and P. Nurmi. Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication. In *IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, Budapest, Hungary, Mar. 2014.

[20] A. Varshavsky, A. Scannell, A. LaMarca, and E. Lara. Amigo: Proximity-based authentication of mobile devices. In J. Krumm, G. Abowd, A. Seneviratne, and T. Strang, editors, *UbiComp 2007: Ubiquitous Computing*, volume 4717 of *Lecture Notes in Computer Science*, pages 253–270. Springer Berlin Heidelberg, 2007.

[21] Vigilant. Vigilant unveils smart IoT innovation for diabetic patients, Feb. 2014. `http://vigilant.ch/en/News/Company_News/2014/0221/53.html` [Referenced 2014-08-23].

[22] T. D. Wu. The secure remote password protocol. In *Proc. Network and Distributed Systems Security Symposium (NDSS)*, pages 97–111, San Diego, CA, USA, Mar. 1998.