Date of acceptance Grade

Instructor

Strengthening Zero-Interaction Authentication Using Contextual Co-presence Detection

Xiang Gao

Helsinki May 28, 2014

MSc Thesis

UNIVERSITY OF HELSINKI
Department of Computer Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Computer Science

Tekijä — Författare — Author
Xiang Gao

Työn nimi — Arbetets titel — Title
Strengthening Zero-Interaction Authentication Using Contextual Co-presence Detection

Oppiaine — Läroämne — Subject
Computer Science

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
MSc Thesis	May 28, 2014	66 pages + 11 appendices

Tiivistelmä — Referat — Abstract

Designing systems that balance security and usability is a desirable but challenging goal. *Zero-Interaction Authentication (ZIA)* is one example of an effort for making security easy to use. It attempts to improve usability by avoiding explicit user interaction for the authentication process, but instead resorting to automatically determining if the principals are co-present. Nevertheless, current ZIA models, which detect co-presence by measuring observed signal strengths in some form of local wireless communication, suffer from relay attacks, where attackers fool the principals by relaying the authentication messages even if the messages exchanged in the authentication protocols are cryptographically secured.

Contextual Co-presence Detection is an alternative technique to detect co-presence. The main idea is that co-present principals should observe similar ambient context. Although prior work has studied the use of single sensor modalities (audio, Bluetooth, GPS and WiFi) for perceiving ambient context, there were (a) no fair comparisons of how different sensor modalities perform and (b) no studies about whether fusing multiple sensor modalities would increase performance.

In this thesis, we built a data collection framework that allowed an individual user to easily collect ground truth data about co-presence of a pair of devices. We applied standard classification techniques to this data. Our results demonstrate WiFi (the set of visible WiFi access points and their respective signal strengths) and that fusing multiple sensor modalities improves performance in terms of security and usability. We then extended a real ZIA application (BlueProximity) with support for contextual co-presence detection, and conducted a small-scale user study to evaluate the usability of contextual co-presence detection as compared to co-presence detection using signal strength only. Our study did not find evidence that the addition of contextual co-presence may harm usability.

ACM Computing Classification System (CCS):
D.2.4 [Security and Protection][Authentication]

Avainsanat — Nyckelord — Keywords
zero interaction authentication, relay attack

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — övriga uppgifter — Additional information

Contents

1 Introduction

Nowadays, usability in security systems has been discussed and explored by academia and practitioners. Zero-Interaction Authentication (ZIA) [CN02] is an example of effort to improve the ease-of-use in security system design. ZIA refers to the authentication process that authenticates the user (prover) to the system (verifier) by detecting the co-presence relation without explicit user interaction in addition to using a standard cryptographic authentication protocol. The concept of co-presence refers to the scenario where two principals reside near to each other within a predefined distance. Co-presence between the user and a device, as an alternative to biometric-based authentication, is not in our scope of research.

ZIA is supported by the short range communication technologies, such as Radio Frequency Identification (RFID) [Jue06], Near Field Communication (NFC) [HMSX12] as an extension of RFID, and Bluetooth [LcA+04, KW05]. An RFID system consists of readers and tags that work in a challenge-response manner to provide contactless identification information in the range of 0.1m to 200m. NFC is developed as an extension of RFID enabling short-range communication (limited to 10cm) and contact-less transactions between two handsets. RFID and NFC are widely applied in public and private services [RFI, NFC]: credit card payment, electronic passports, transport cards, door access cards, and vehicle keys. Alternatively, Bluetooth, as defined in Bluetooth SIG adopted specifications [SIG], works in the range of 10 - 60 meters, and is mainly adopted in data exchange applications where the communication channel is secured with a shared session key.

Current ZIA models, however, are vulnerable to relay attacks. The typical example, "ghost-and-leech" attack, is a kind of man-in-the-middle attack with two colluding attackers fool the verifier by forwarding packets between the prover and the verifier.

This calls for enhancements to protect the authentication process from such relay attacks while preserving usability. One of the traditional solutions - distance bounding [BC94], a protocol that calculates proximity by measuring Round-Trip Time (RTT), is only acceptable in limited scenarios where time delay is minimal and hardware is specially designed. Contextual co-presence detection is an alternative solution: co-presence is determined by comparing the context information sensed by both principals. Co-present principals observe similar shared context, whereas devices that are not co-present observe different contexts. Devices can observe the nearby context by using sensors for sensing ambient environment.

Common sensors (e.g., microphones, WiFi and Bluetooth interfaces, GPS receivers) in commodity computing devices can be used for collecting ambient context information. Existing contextual co-present approaches investigated single sensor modalities including audio [HMSX12, SS13], WiFi [KH04, VS07, NT11], Bluetooth [NT11] and GPS [MPSX12]. However, these single sensor modalities are potentially vulnerable to attacks of manipulating the context [TRPv09]. In order to improve the security of contextual approaches while still maintaining the usability, one approach is fusing the context information from multiple sensor modalities. We conjecture that the context features sensed from multiple sensor modalities make the system intrinsically difficult to attack since the complexity and dynamics of environment makes it harder to manipulate multiple modalities simultaneously.

Our contributions:

1. We compared the performance of four ubiquitous sensor modalities - audio, WiFi, Bluetooth, GPS - used for contextual co-presence detection. To do so, we designed a data collection framework and launched a data collection to collect co-present data. Our comparison reveals the difference in resisting relay attacks for single sensor modalities.

2. We built a contextual co-presence decision model by fusing multiple sensor modalities for ZIA. Our data analysis demonstrates that the fusion approach improves security against relay attacks without sacrificing usability.

3. We built a demonstrative system, which we call BlueProximity++, for contextual co-presence detection augmented ZIA model. We conducted a small-scale user study to evaluate the performance of our model in practical scenarios.

These contributions resulted from a joint work by members of Secure Systems group in University of Helsinki (Finland) and University of Alabama at Birmingham (USA). I was responsible for conducting the data collection and implementing the data collection framework, designing and implementing the demonstrative system BlueProximity++, and conducting the user study for evaluation. I also contributed to the data analysis.

The thesis is structured as follows. Section 2 introduces the background information, including Zero-Interaction Authentication, relay attacks, and ubiquitous sensor modalities. Section 3 presents the related work on relay attack resilience for ZIA models, and existing context-based authentication approaches. Section 4 presents

the statement of the problem we address and its requirements. Section 5 describes the data collection framework designed for our research. Section 6 and 7 presents the data collection and analysis process to get the contextual co-presence decision model. Section 8 proposes an initial adversarial model and the resilience of our approach against such attacks in such a model. Section 9 describes our demonstrative system BlueProximity++ and the small-scale user study to evaluate its practical performance. Section 10 evaluates our contextual co-presence detection approach. And finally Section 11 concludes this thesis by summarizing our contributions and discussing the future plan.

2 Background

2.1 Zero-Interaction Authentication

Zero-Interaction Authentication (ZIA) [CN02] was developed to solve the problem of frequent re-authentication and to improve the usability. This problem resides in scenarios like frequently opening the door of a car using a contact-less key, or unlocking a laptop every time the user returns from a break. In ZIA, the user holds a device as a contact-less key to unlock a locked system (a locked car or a locked laptop) via a short-range wireless communication channel.

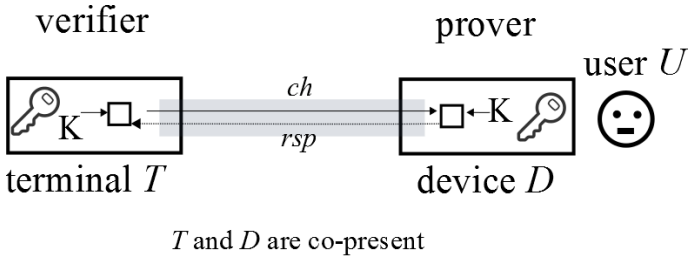


Figure 1: System model for ZIA.

Figure 1 describes the conventional ZIA model, where the user U tries to authenticate to a terminal T by approaching it with a device D. We assume that D is always carried by U. Before using ZIA, the user U is required to set up an initial security context by "binding" or "registering" the *prover* D to the *verifier* T. This setup procedure results in a *security association* between T and D, e.g., a shared secret key K. ZIA is triggered when D approaches T:

- T authenticates D by running a traditional challenge-response entity authentication protocol based on the pre-established shared security association;
- T verifies that D is at a short physical distance from it (i.e. D and T are co-present).

ZIA is applied in many access control scenarios. For example, *BlueProximity* [Blu], an open source Linux application, is designed to enable automatic locking/unlocking T's screen by verifying the co-presence of D based on *Bluetooth received signal strength indicator (RSSI)*, without user interactions such as typing a password. The principle under the hood is that T detects the Bluetooth signal strength of D as the estimation of physical distance. There are other examples such as *Keyless Go* [Key] (an authorization system for opening/closing a vehicle door), *PhoneAuth* [CDK+12] (a two-factor authorization system for web use), and potential proximity-based access control systems using wearable devices [Tog13].

2.2 Relay Attacks

ZIA model is vulnerable to relay attacks such as "ghost-and-leech" (as shown in Figure 2). We assume the terminal T and the device D are actually far away (not intending to trigger authentication). In this scenario, the attacker uses a *leech* (A_d plays as a terminal) near D, and a *ghost* (A_t plays as a device) near T. Both A_t and A_d are responsible for relaying encrypted authentication information (challenge *ch* and response *rsp*) via a high bandwidth connection. As a result, T concludes that D is in close proximity because the authentication protocol and the proximity distance measure both succeed. Security is therefore compromised.

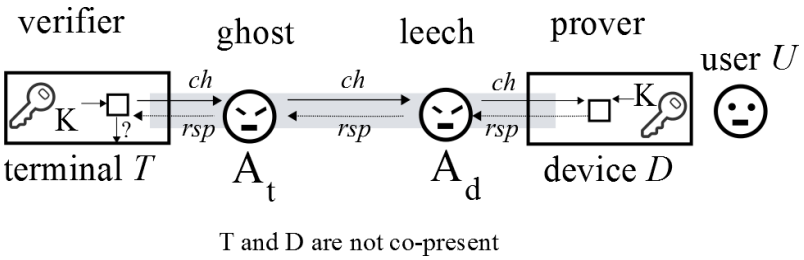


Figure 2: Relay attack for ZIA model [KW05].

The distance limitation between A_d and D might lead to attack failure, since A_d may catch the user's attention. However, for RFID-based applications, the distance

between A_d and D can be extended up to 50 meters [KW05], enough to achieve the relay attacks. And for Bluetooth-based applications, the range of communication can be 10-60 meters, which is a loose distance limitation.

2.3 ZIA Enhanced with Contextual Co-presence

Figure 3 depicts the approach of contextual co-presence in ZIA, as a promising resistance against relay attacks. The principle under the hood is that co-present entities should observe similar shared context.

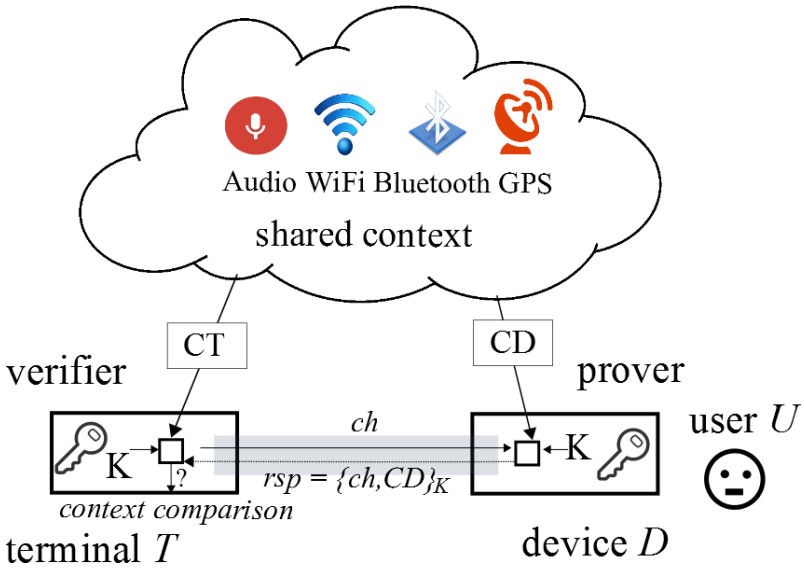


Figure 3: System model for ZIA with contextual co-presence.

In this approach, the communication channel between T and D is secured with a pre-established shared key or some other form of a shared security association. D triggers T to send a challenge message ch, and start context scanning on both sides. The result CD and CT are generated from ambient sensor modalities. When context scanning is done, D sends a response rsp by encrypting ch and CD with the key K. Upon receiving rsp from D, T compares the local result CT with CD. T concludes that D is co-present if CT and CD are similar, and the authentication is established successfully. Otherwise, D is considered not co-present with T, and the authentication request is declined.

When multiple sensor modalities are used to scan the context, CD and CT are vectors. In prior work on context-based solutions for co-presence detection, some ambient sensor modalities, such as ambient audio [HMSX12, SS13], WiFi [KH04,

VS07, NT11], Bluetooth [NT11], GPS [MPSX12], ambient light [HMSX12], acceleration [MG07], etc, were investigated. However, all existing solutions are based on single modalities.

3 Related Work

We investigated prior work on resilience against relay attacks for ZIA model. In this section, we present the straight-forward solution - distance bounding, and elaborate the family of contextual co-presence solutions.

3.1 Distance Bounding

In distance bounding protocols [FDC10, RC10, MJ07] (Figure 4), there is a pair of principals as known as *verifier* and *prover*. Both principals share a security association that they can use to authenticate messages from each other. The *verifier* sends a challenge message to the prover, and the *prover* computes a response for the challenge based on the shared security association and sends the response back after a short interval of processing time T_p. On receiving the response, the *verifier* checks that the response corresponds to the challenge as specified by the authentication protocol and gets the measured round-trip time RTT. So the *lower distance bound* between *verifier* and *prover* is

$$distance\ bound = \frac{RTT - T_p}{2} \cdot c$$

and c is the speed of signal propagation (i.e. the speed of electromagnetic waves). The computed bound is a measure of proximity, which can defeat ghost-and-leech attacks.

In practice, the processing time T_p should be minimized since a small error in T_p will lead to a large deviation in the resulting estimation of the distance bound, given the multiplier c. So the protocol is required to be implemented at the lowest layer of the communication stack. It limits the application of the protocol to implementation in hardware or firmware, therefore is not suitable for commodity mobile devices.

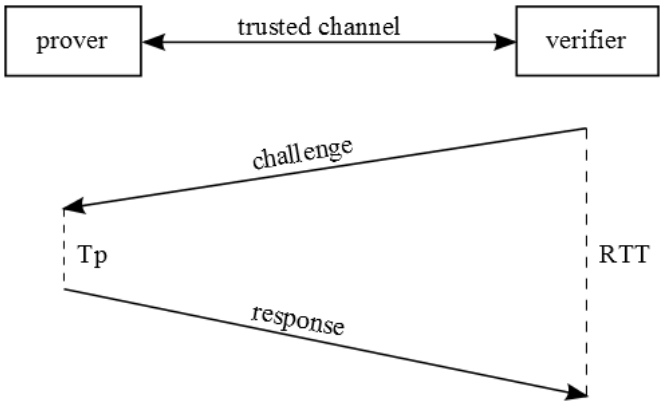


Figure 4: Distance bounding defense.

3.2 Contextual Co-presence

We investigated the prior work of relay attack resistance using contextual co-presence, and the co-presence detection techniques. We categorized them based on the sensor modalities used.

3.2.1 Ambient Audio

We present the two strategies for detecting co-presence based on ambient audio information: using raw recordings in Halevi et al.'s work [HMSX12], and fingerprinting techniques in Schurmann et al.'s work [SS13].

According to Halevi et al.'s work [HMSX12], they use microphones to record audio raw signals. For any pair of mobile devices, audio recording starts simultaneously for a short interval. After acquiring audio recordings, they transform the time-domain signals to meaningful features (similarity and distance) that are directly used for making proximity decisions. The following feature extraction techniques are employed in Halevi et al.'s work for for computing the similarity and distance:

- **Time-based feature extraction**: Two time-domain raw audio signals denoted by X_i and X_j are *normalized* to make total energy equal to 1: $\hat{X}_i = \frac{X_i}{||X_i||}$ and $\hat{X}_j = \frac{X_j}{||X_j||}$. The similarity S and the distance D are computed in two ways: (i) In the correlation-based method, cross-correlation [XCO] is used to compute S. Cross-correlation is a mathematical means for measuring the similarity of two waveforms, preferable for comparing two signals with uncontrolled time-

lag. The time-based similarity of signals X_i and X_j using correlation-based method is denoted by $S_{c,time}(i,j)$, and the corresponding distance is similarly denoted by $D_{c,time}(i,j)$.

$$S_{c,time}(i,j) = max(Cross\text{-}Correlation(\hat{X}_i, \hat{X}_j)),$$

$$D_{c,time}(i,j) = 1 - S_{c,time}(i,j).$$

(ii) In the distance-based method, Euclidean norm [Nor] is used to compute D. The time-based distance of signals X_i and X_j using distance-based method is denoted by $D_{d,time}(i,j)$, and the corresponding similarity is similarly denoted by $S_{d,time}(i,j)$.

$$D_{d,time}(i,j) = ||\hat{X}_i - \hat{X}_j||,$$

$$S_{d,time}(i,j) = 1 - D_{d,time}(i,j).$$

- **Frequency-based feature extraction**: Two time-domain raw audio signals denoted by X_i and X_j are converted to frequency domain using Fast Fourier Transform (FFT) [FFT]. Then $FFT(X_i)$ and $FFT(X_j)$ are normalized to $\widehat{FFT(X_i)}$ and $\widehat{FFT(X_j)}$ in the same way as the time-based approach. The similarity S and the distance D are computed in two ways: (i) In the correlation-based method, cross-correlation is used to compute S. The frequency-based similarity of signals X_i and X_j using correlation-based method is denoted by $S_{c,freq}(i,j)$, and the corresponding distance is similarly denoted by $D_{c,freq}(i,j)$.

$$S_{c,freq}(i,j) = max(Cross\text{-}Correlation(\widehat{FFT(X_i)}, \widehat{FFT(X_j)})),$$

$$D_{c,freq}(i,j) = 1 - S_{c,freq}(i,j).$$

(ii) In the distance-based method, Euclidean norm [Nor] is used to compute D. The frequency-based distance of signals X_i and X_j using distance-based method is denoted by $D_{d,freq}(i,j)$, and the corresponding similarity is similarly denoted by $S_{d,freq}(i,j)$.

$$D_{d,freq}(i,j) = ||\widehat{FFT(X_i)} - \widehat{FFT(X_j)}||,$$

$$S_{d,freq}(i,j) = 1 - D_{d,freq}(i,j).$$

- **Time-Frequency-based feature extraction**: Time-based and frequency-based features are combined to compute the distance D and the similarity S.

The time-frequency-based distance of signals X_i and X_j is denoted by $D(i,j)$, and the corresponding similarity is similarly denoted by $S(i,j)$.

$$D(i,j) = \sqrt[2]{(D_{c,time}(i,j))^2 + (D_{d,freq}(i,j))^2},$$

$$S(i,j) = 1 - D(i,j).$$

An alternative approach of generating audio context information is acoustic fingerprinting. *Acoustic fingerprinting* is the method of extracting characteristic patterns from audio sequence. It is widely adopted in identifying a certain piece of sound effects or music from large audio databases. Although studies on music fingerprinting operate on music properties like amplitude, rhythm, contour and pitch, the ambient audio sequence calls for more general techniques. Schurmann et al. [SS13] developed an energy-based fingerprinting method for audio sequences to establish audio fingerprinting-based authentication, based on Haitsma and Kalker's robust fingerprinting algorithm [HK02] (essentially dynamic programming). They used the fingerprints to establish a secure channel between two co-present devices. The general idea is to generate a fingerprint as a bit sequence from an audio sequence based on the differences of energy between all consecutive frequency bands.

There are four steps before comparing fingerprints from the original audio sequence: (1) dividing the audio sequence S into n frames $S_i, i \in 0, ..., n-1$ on each of which Discrete Fourier Transformation (DFT) is applied; (2) splitting each frame S_i linearly and evenly into m non-overlapping frequency bands $S_{ij}, j \in 0, ..., m-1$; (3) establishing an energy matrix E with the sum of energy values for each S_{ij} as elements: $E_{i,j} = \sum S_{ij}[k]$; (4) generating the fingerprint from the differences between two successive $E_{i,j}$:

$$f(n) = \begin{cases} 1, & (E_{i,j} - E_{i,j+1}) - (E_{i-1,j} - E_{i-1,j+1}), \\ 0, & otherwise. \end{cases}$$

Afterwards, they use Hamming distance to represent the difference between two fingerprints (bit sequences) and set a proper threshold for proving proximity.

Halevi et al.'s experiments indicate zero error (Table 2 of [HMSX12]), but under the condition that co-present devices are of identical models. Schurmann et al.'s experiment on comparing synchronously sampled data indicates around 70 percent similarity. Although we cannot compare both strategies fairly, we can still find the factors impacting the fingerprinting performance. It is level of ambient noise that

increases errors in the audio fingerprint bit sequence. Besides, in places with strong background audio sources, e.g., cafes, concert halls, CD stores, audio fingerprinting works well. This is because the audio amplitude measurement tends to be larger in such scenarios than in quiet places or places with evenly distributed noise.

3.2.2 WiFi

We investigated related work using WiFi to detect co-presence or generate secret keys. In general, prior work uses WiFi in two strategies: using WiFi scan results (information of nearby WiFi access points (APs)), and using WiFi broadcast traffic (extracting information from raw packets). We present the representative work from Narayanan et al. [NT11], Krumm et al. [KH04], and Varshavsky et al. [VS07].

Narayanan et al. [NT11] studied the strategy to use the list of WiFi access point IDs nearby to prove co-presence. According to their statistics, on average, around half of the number of AP IDs in the visible list is different from that at a different location.

Narayanan et al. [NT11] also conducted experiments to evaluate the performance of using WiFi broadcast packets for detecting co-presence. They derive various independent elements that contain different values, such as IP addresses, and packet sequence tokens. According to the experiment done at Stanford, the entropy in WiFi broadcast packets sensed at a location and a certain time reflects the unforgeability of presence in that location and time, which is more than enough to be used to distinguish co-presence and non-copresence. WiFi broadcast traffic calls for sufficient density of nearby devices, and is applicable to indoor scenarios like campus and libraries. However, WiFi broadcast traffic is not applicable to commodity mobile devices where accessing WiFi packets is not possible from an ordinary application.

Krumm et al. [KH04] proposed "NearMe" which uses WiFi signatures (MAC addresses and received signal strength values) features for co-presence detection. They built a model using data collected in an office building environment and tested in a cafeteria environment. They conjectured that their approach generalizes well to other settings.

Varshavsky et al. [VS07] presented "Amigo" to authenticate co-present devices using various features extracted from the WiFi environment. In Amigo, two devices perform a Diffie-Hellman key exchange after which each device monitors the radio environment and generates a signature based on data observed and sends it to the

other device for co-presence verification. Amigo also systematically introduced a set of synthetic features (elaborated in Section 6.2).

3.2.3 Bluetooth

Bluetooth inquiry results can be utilized to generate useful context features similar to those used by Narayanan et al. [NT11] and Varshavsky et al. [VS07]. Bluetooth MAC address along with RSSIs can be extracted from the inquiry results as meta context information for nearby devices, and the list of such meta information can be used to detect co-presence in the same way of using WiFi access points.

Bluetooth has the similar limitations as WiFi APs. In addition, most Bluetooth devices are personal devices, invisible by default to public due to privacy concerns. This limitation reflects the most common situation that co-present devices observe only each other.

3.2.4 GPS

Civilian GPS is not a reliable means to locate positions indoors, since the satellite signal is often blocked or shielded by building structures [KBG+10]. Often, the user has to wait for seconds and move fast in order to refresh the location displayed on Google map. There are three reasons for the weakness: GPS positioning starts with a slow initialization process taking the history point as the initial value; mobile GPS receivers can compute their location faster than static receivers; the military level signals of GPS provides better performance than the civilian level signals. Here we emphasized that GPS's performance of locating static objects is important in most application scenarios where two static devices authenticate by comparing their location information. Narayanan et al.'s report [NT11] proved the unforgeability with more than four satellites. We can reach better performance by installing extra components like professional GPS receivers, but that undermines the usability.

Ma et al.'s work [MPSX12] on location-aware RFID security demonstrated how to utilize location information from GPS on co-present devices to defend against relay attacks. They used external GPS modules to provide context comparison enhancing simple RFID authentication. They selected longitude, latitude and speed together as the target context information. According to the National Marine Electronics Association (NMEA) specification [NME], they extracted position, velocity and time values from the Recommended minimum specific GPS/Transit data (GPRMC), with

Global Positioning System fix data (GPGGA) serving as a fix. They extracted the accurate speed value directly from GPRMC. The speed value is computed in the GPS unit based on Doppler Effect when the device is moving. In the proof-of-concept experiment, they used a external GPS receiver module instead of embedded modules in smart handsets to acquire GPS information. The GPS receiver supports rapid satellite acquisition, ensuring acceptable performance compared to built-in GPS module in smartphones. In their approach, they attempt to authorize the device by detecting the proximity between the sensed location and the preset legitimate locations. The device is authorized only if its location vector falls into a square region centered at a legitimate location. According to the experiment result of the location-based selective authorization, location information acquired from GPS is strong and accurate context information for detecting co-presence.

GPS location coordinates are not available in many scenarios like indoor environment. Inspired by Narayanan et al.'s work [NT11] on similar RF sensor modalities (e.g. WiFi and Bluetooth), we conjectured that GPS raw data can be used to generate context features for co-presence detection. GPS raw data, specifically the GPS satellites in view (GPGSV) [NME] message provides essential information about the visible satellites: Pseudo-Random Noise code (PRN, the unique identifier of a satellite), Elevation (the vertical angle in 0 - 90 degree), Azimuth (the horizontal angle from the true north in 0 - 359 degree), and Signal-Noise Ratio (SNR, indicating the received signal strength). GPGSV information is always available when the GPS receiver is on, and is updated every second. The main idea of using GPS raw data for co-presence detection is that devices at different locations observe different information of the satellites in view. It seems promising to use the list of PRNs and SNRs to distinguish co-presence and non-copresence, but it still calls for experiments to evaluate the performance.

3.2.5 Other Sensor Modalities

Ambient Light

Halevi et al. [HMSX12] proposed an alternative co-presence detection approach based on ambient light. They count on the fact that light illuminance that varies at different locations inside a room. However, light illuminance differs from ambient audio in two ways: light illuminance is heavily influenced by the direction and bearing of the smartphone, so in theory the probable error is larger than ambient audio; the light condition observes very slow fluctuation, so the mathematical step

mentioned for ambient audio can be altered to the mean of illuminance data over the sampling interval. So the distance of illuminance ($D(i,j)$) between location i and j is:

$$D(i,j) = |\overline{L_i} - \overline{L_j}|,$$

where L_i and L_j denote the illuminances at location i and j respectively.

Halevi et al. showed that although ambient light can be used as context information in detecting co-presence, errors observed in their experiments undermine the robustness of such technique. Ambient light is not considered as strong a sensor modality as ambient audio.

Acceleration

Accelerometers are widely deployed in most commodity mobile devices. They are passive sensors accessible all the time, and are efficient in scanning time and energy-consumption. We investigated related work in [CKSK08, MG07, HMS01] using acceleration to resist relay attacks.

Czeskis et al. [CKSK08] presented "context-aware communication" for RFID authentication between reader and tag to defend against "ghost-and-leech" attacks. Context-aware communication refers to the limitation that communication between RFID reader and tag is allowed only if the action of the tag matches predefined pattern. For example, when unlocking a car, the user has to insert the key into the slot and twist it, which makes a defined action pattern. The authors deployed accelerometers in RFID tags to collect acceleration information and makes decision based on fast gesture recognition results. They also designed "secret handshakes" to distinguish the predefined actions from daily gestures.

An alternative strategy is to detect co-presence by testing the device acceleration when the user is asked to shake both handsets with one hand. Holmquist et al. [HMS01] proposed a proof-of-art application named "Smart-Its Friends" to establish connection by detecting contextual co-presence. The principle is to sample the acceleration measurement periodically and match the sample with preset patterns. The user has to shake two devices in his hand to establish the dedicated connection. R.Mayrhofer and H.Gellersen [MG07] extended their work by applying the same technique in secure authentication.

Although shaking seems to add extra manual interactions to authentication, it is considered intuitive action when people meet each other. Besides, shaking acceler-

ation is difficult to be forged since acceleration values and pattern vary with time and people.

4 Problem Statement and Requirements

We proposed the following research questions:

1. How do different sensor modalities commonly available on commodity computing devices perform when used for contextual co-presence detection to improve the security of ZIA?
2. Does fusing multiple sensor modalities improve performance over single sensor modalities?
3. Does the use of contextual co-presence impact usability of ZIA compared to standard approaches for detecting co-presence?

The resulting solution targets the following requirements:

(i) Improves security compared with prior work on single sensor modalities.

(ii) Maintains acceptable usability.

(iii) Being robust to the variance of hardware and the difference of user perception of co-presence.

We planned our research roadmap as follows:

1. Collecting sensor data with ground truth. A data collection framework is developed to support the data collection.
2. Analyzing data to obtain classification results of single modalities and multiple modalities.
3. Establishing a specific adversarial model for security assessment of contextual co-presence.
4. Extending a ZIA application with contextual co-presence detection, and conducting a small-scale user study for usability assessment.

5 Framework for Data Collection

In order to collect a large sensor dataset (including ground truth), we developed a data collection framework. It was motivated by our special requirement that we need to collect data from two devices, either co-present or not co-present, at the same time. Our goal was to have an easy-to-use, non-intrusive application that enables collecting sensor data along with co-presence ground truth automatically in a large scale. The application was implemented extensible and maintainable to be re-purposed for other controlled experiments.

Concretely, the framework complied with the following characteristics:

- A framework with a plug-in mechanism that allows later addition of new sensor modalities;
- The possibility for a user to indicate whether two devices are co-present or not by providing input on only one of them.
- A balance between collecting ample data without imposing excessive battery consumption while still letting the user to temporarily disable data collection.

5.1 Design

As Figure 5 depicts, the architecture is composed of a back-end synchronization server (*Server*), a pair of devices (*Device A* and *Device B*) with a *client component* on each device, and *communication channels* between the server and the devices.

5.1.1 Server

Server is designed to maintain the "binding" relation of the two devices belonging to one user, and to synchronize data collection tasks by routing control messages. It is the relay point of the communication channel between a pair of devices. The server also provides the service to store the collected data samples on server-side database.

The server runs as a daemon service. A thread pool is implemented to maintain the raw TCP connection from clients. Server and client exchange control messages via the communication channel. We implemented message handling callbacks on the basis of TCP socket communication. All messages are serialized in JSON format, with the essential key "id" mapped to the role constants (e.g., REQ_TASK refers

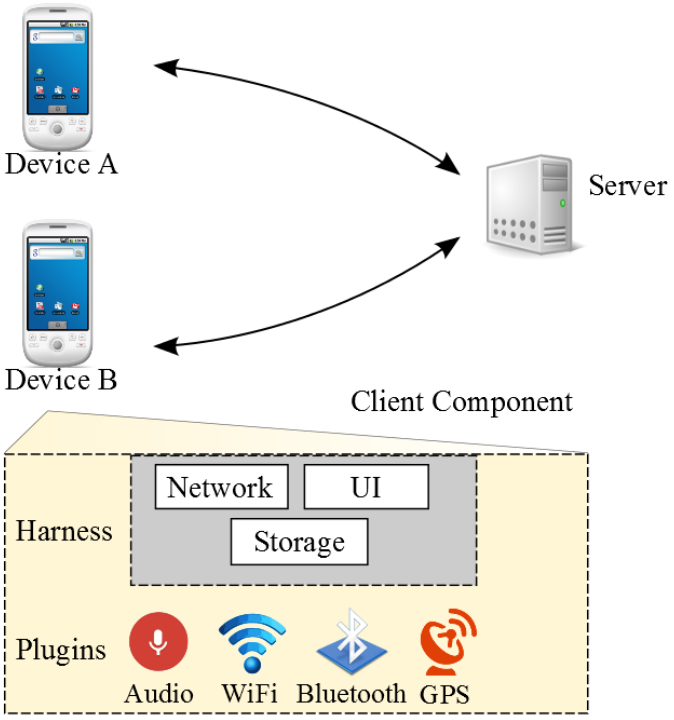


Figure 5: Data collection framework architecture.

to the message of making task requests), and optional keys mapped to extra flags (e.g., in REQ_TASK, UUID refers to the universal unique string representing the identity of device).

5.1.2 Client Component

Client component runs on devices to scan sensor data. It provides a user interface for user to indicate co-presence ground truth.

The client component consists of a harness with common functionalities, and a plugin interface for incorporating specific sensing modules for different sensors. Two devices belonging to one user use the communication channel via server to synchronize sensing tasks.

On the client-side, *Harness* consists of *Network* module (maintaining TCP communication with the server), *UI* module (providing the user interface for task control and ground-truth indication), and *Storage* module (caching sensing data from all modalities). We implemented plugins for different modalities: *GPS plugin* for GPS satellite

information, WiFi plugin for WiFi AP scanning, *Bluetooth plugin* for Bluetooth device discovery, and *Audio plugin* for sound recording. The extensible architecture enables incorporating additional modalities (e.g., humidity, temperature).

5.1.3 Context Scanning Task

Binding procedure

Binding is the prerequisite procedure for context scanning tasks. The purpose is to establish an association (namely a bind) between peer devices by exchanging meta information and registering the bind on the server. To elaborate the details of binding procedure (Figure 6), we suppose *Device A* and *Device B* are not paired to other devices. *Device A* sends a request message *REQ_ GETQ* with its *UUID* (Universally Unique Identifier) and *Name* (customizable by the user, by default the model name) to *Server* (step 1). Upon reception, *Server* generates a queue number (i.e. *QNum* in the figure, a 4-digit random integer) to the pending bind (Device A and an empty placeholder), and sends the queue number back to *Device A* (as *ACK_ GETQ*) (step 2). *Server* keeps a pending queue number valid for 5 minutes. Then, the user inputs the readable queue number into *Device B*, which sends a *REQ_ VALQ* message with its *UUID*, *Name* and queue number up to *Server* (step 3). Finally, *Server* validates the received queue number (step 4): if it matches a pending queue number, then *Device A* and *Device B* are successfully bound; otherwise, binding procedure fails. Hereafter , we use the phrase "bound devices" to refer to two devices A and B that have been "paired" using the aforementioned procedure.

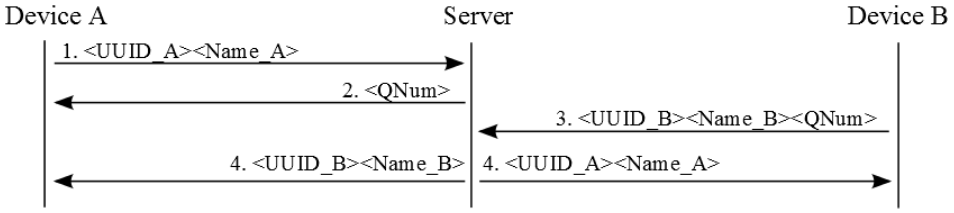


Figure 6: Binding procedure between two devices.

Task

Figure 7 shows the procedures of a context scanning task. A task is triggered by sending a request message from either of the bound devices (suppose it's device A) to the server (step 1). The request message also contains the ground truth of

co-presence and the *modality mask* (a bitwise mask of the modalities enabled on local device) of device A. Upon reception, the server sends another request message to device B (step 2). If device B is idle, it replies with its own modality mask to the server (step 3). Then on the server, the masks from A and B are intersected to generate the common modality mask, and the server sends the trigger message containing the common modality mask to both devices simultaneously (step 4). On receiving the trigger message, A and B starts local context scanning using the modalities enabled in the common modality mask (step 5). After completing the local context scanning, each device uploads its local scanning results to the server (step 6), when a context scanning task is completed.

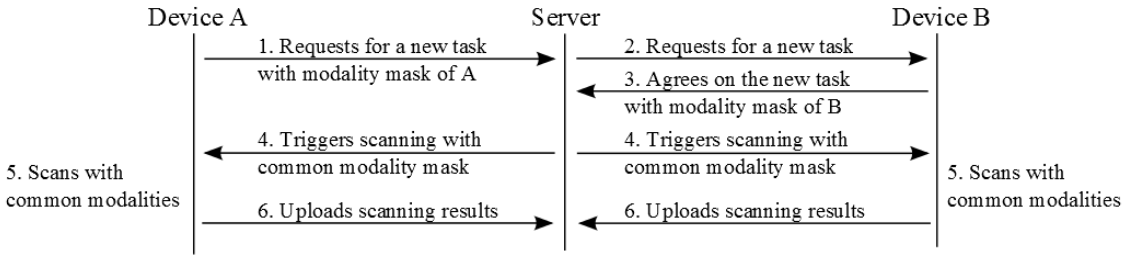


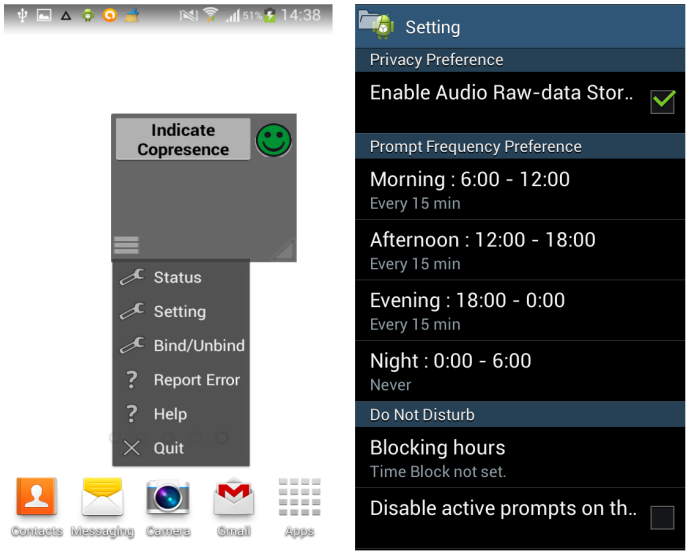
Figure 7: Context scanning task procedures.

Time Synchronization

Time synchronization is essential to guarantee the validity of sensing data from bound devices. We investigated standard time synchronization techniques, e.g. the Network Time Protocol (NTP) [SS13] maintaining a reference clock and a layered hierarchy. However, we adopted an alternative *relative time synchronization* mechanism in our framework because it was simpler to implement and was just enough for our data collection tasks. In our mechanism, the server triggers a task by sending a command to a pair of bound devices simultaneously. On receiving the command, each device starts context scanning. Each device also maintains a periodic ping-echo heartbeat to the server, not only to keep track of the status of the server and peer device, but also to measure the average round-trip time $avgRTT$ between the device and the server. Then the device time is synchronized to the server by eliminating the transmission delay of $avgRTT/2$.

5.2 Usage

Setup: The Android client application starts with a minimalist floating window over the screen (of any other applications) as shown in Figure 8(a). This is the main user interface pertaining to configurations and data collection tasks.



a. Floating windows with dropdown menu b. Setting panel

Figure 8: Data collection client: setting UIs.

The user is supposed to bind two Android devices before data collection. When a new client connects to the server, the device information is automatically registered to server, and a queue number is generated on user's first device. The user has to input the queue number into the second device to authorized the binding relation.

The user can configure related preferences in Setting panel shown in Figure 8(b). Due to privacy concerns, we disable, by default, storing audio raw data on the server, but the user can toggle the option to enable it. Besides, the user can customize the "Do not disturb" options to mute the periodic prompts (with vibration, as reminder for starting new data collection tasks).

Tasks: Before the routine data collection tasks, the user is encouraged to confirm that the peer client is reachable and all sensor modalities are enabled. We provided "Status UI" for the real-time status report of all sensor modalities and the reachability of the server and peer client. As shown in Figure 9, the user is recommended to turn on all the sensors marked with red "Disabled" sign before launching new tasks.

Figure 10 illustrates the user involvement in triggering a new task. A green smily face means both clients are ready. In this case, the user can indicate the ground truth

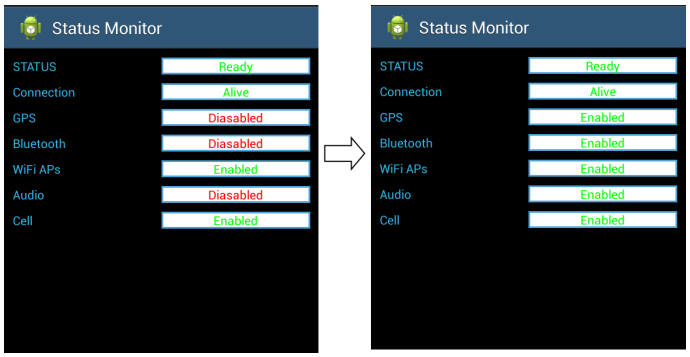


Figure 9: Data collection client: status UI.

of co-presence manually or in response to a periodic prompt (vibration reminder). By clicking the green "Yes" or red "No" button, the corresponding co-present/non-co-present ground truth is recorded and a subsequent context scan is triggered from the server simultaneously. Scanning lasts for 2 minutes after which ground truth and sensor data are uploaded to the server, and both clients return to "ready" status.

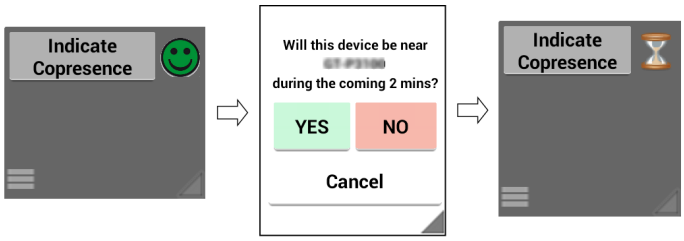


Figure 10: Data collection client: task UIs.

5.3 Sensor Data

We currently have plugins for GPS, WiFi, Bluetooth, and Audio modalities. These modalities were chosen as they are widely deployed in commodity computing devices.

GPS raw data: We extracted the information of the visible GPS satellites from received GPGSV messages (see Section 3.2.4). We recorded the identifier and "signal strength" for each of the satellites in view. The identifier, i.e. the Pseudo-Random Noise code (PRN) is an integer ranging from 1 to 32, and "signal strength" in the form of signal-noise ratio (SNR) is an integer ranging from 1 to 100. The list of such records was updated every second. And the resulting sample consisted of approximately 120 such lists within 2 minutes.

WiFi access points: We extracted the meta attributes from the result of WiFi scanning for access points. Each entry pertaining to one access point consists of the MAC address (BSSID) and the Received Signal Strength Identifier (RSSI). BSSID is a string with 6 hexadecimal characters, and RSSI is an integer ranging from -100 to -20 dBm (the empirical upper bound observed from Android devices). The empirical period for one scan is around 1 second. Each resulting sample consists of 10 consecutive scan records.

Bluetooth inquiry results: We extracted the meta attributes from the result of Bluetooth inquiries (with RSSIs). Each entry pertaining to one Bluetooth device nearby consists of the MAC address (BDADDR) and RSSI. BDADDR is a string with 6 hexadecimal characters, and RSSI is an integer ranging from -100 to -20 dBm (the empirical bounds observed from Android devices). Unlike WiFi scanning, Bluetooth inquiry works in a broadcast-response manner, taking by default 10.24 seconds as one cycle for results. The resulting sample consisted of the inquiry results of 10 consecutive cycles within 2 minutes.

Ambient raw audio: We recorded ambient audio in uncompressed PCM format (wave file). The sampling rate was 44100Hz, the encoding width was 16-bit, and the duration was 10 seconds. Due to privacy concern, the raw audio were not directly uploaded to server. Instead, we extracted certain features (as detailed in Section 6.2).

6 Data Collection

In this section, we describe the data collection along with the resulting dataset, and summarize the features that we adopted in data analysis.

6.1 Description

Setup: During the middle of 2013, we conducted the data collection to gather raw data for analysis. Five researchers (three in Helsinki, Finland, and the other two in Birmingham, Alabama, the US) in the group participated in the data collection using our data collection framework. All participants were instructed to use their own devices in any scenarios without restrictions. The overall *uncontrolled* setting was deliberately designed to ensure the robustness of the resulting dataset across

various users and scenarios. Table 1 described the uncontrolled setting for data collection.

Type	Information
Time	June - July, 2013
Participants	5 from Finland and the US
Devices	Phones & Tablets (various models: Google Nexus 7, Samsung Galaxy Tab 2, Acer Iconia Tab, Asus Transformer, Samsung Galaxy S3)
Places	Not pre-determined, depending on places where participants collected data, e.g., university campus, labs, libraries, cafeteria, home, streets

Table 1: Uncontrolled setting for data collection.

Result: The resulting dataset contains 2303 samples in total. We maintained a balanced ground truth distribution, with 49.5% samples were co-present, while 50.5% were not co-present. Not all samples were collected with all 4 sensor modalities. According to our observation, missing sensor modalities were caused by the following reasons: the participants sometimes forgot to enable all sensor modalities when collecting data; there were no Bluetooth devices nearby or the nearby devices were not discoverable; audio recording was sometimes disabled due to privacy concern; the GPS receiver was in slow start phase to download almanac and ephemeris data before capturing GPS satellite signals.

As summarized in Figure 11, most samples contained WiFi and Audio data (2269 with WiFi, and 2117 with audio), and Bluetooth and GPS data are limited but enough for analysis (1600 with Bluetooth, and 782 with GPS).

As mentioned in Section 5.3, the time budget for each sensor modality is different: 2 minutes for GPS scanning, 20-30 seconds for WiFi (10 consecutive rounds in practice), 10 seconds for recording ambient audio burst, and 2 minutes for Bluetooth (up to 10 consecutive rounds).

Privacy: Prior to the campaign, all the participants consented to our privacy policy. The collected dataset is anonymized and released on request for research purposes. And the participants had the right to inspect and withdraw his/her data from our database. We anonymized all sensor raw data as a preprocessing procedure in three ways: replacing device identifiers (including MAC addresses) with their SHA-1 hashed strings; replacing the GPS co-ordinate tuples (longitude, latitude) with the Euclidean distance estimation on earth surface; replacing the raw audio data (wave files) with relevant synthetic features as summarized below.

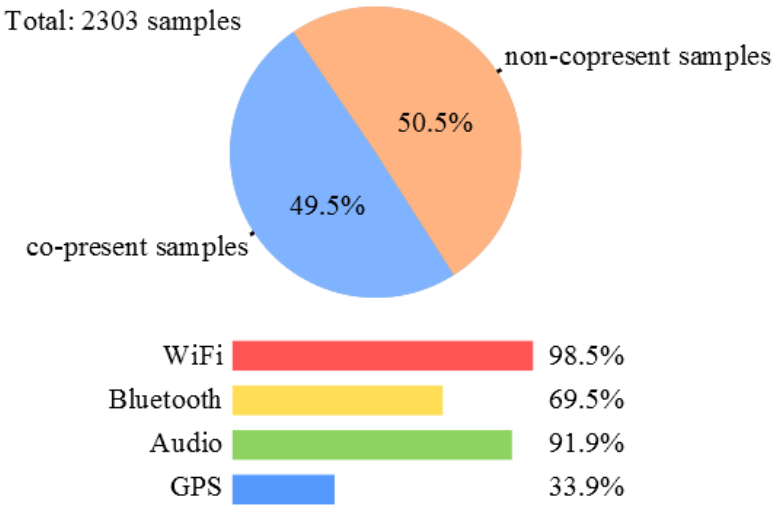


Figure 11: Dataset distribution based on ground truth and sensor modalities.

6.2 Features

In this section, we summarize the promising features from among those discussed in Section 3.2. These features were extracted from the dataset in a preprocessing step before the classification described in Section 7.

6.2.1 Features for WiFi, Bluetooth, GPS

We investigated the sensors with radio-frequency (RF) emissions, capable of sensing nearby devices in a preset range, namely scanning for beacons (i.e., WiFi APs, Bluetooth devices in range, and GPS satellites in view). Such a beacon can be represented as a tuple of entity identifier along with the associated signal strength value. We defined the notations of related attributes as in Table 2.

Then we defined the following sets as an intermediate step:

1. Set of tuple (m, s) sensed by device A and B respectively (denoted by S_a, S_b): $S_a = \{(m_i^{(a)}, s_i^{(a)}) \mid i \in \mathbb{Z}_{n_a-1}\}$, $S_b = \{(m_i^{(b)}, s_i^{(b)}) \mid i \in \mathbb{Z}_{n_b-1}\}$.

2. Set of beacon identifiers sensed by device A and B respectively (denoted by $S_a^{(m)}, S_b^{(m)}$):
$S_a^{(m)} = \{m \; \forall (m, s) \in S_a\}$, $S_b^{(m)} = \{m \; \forall (m, s) \in S_b\}$.

Table 2: Notations for RF sensor modalities.

Notation	Information
a, b	Identities of a pair of bound devices (A, B) which initiate sensing.
m	Identifier of a sensed beacon.
s	Associated signal strength.
θ	Sensor-specific default value of associated signal strength.
n	Number of beacons sensed by one device.
r	Rank of the associated signal strength in the set of beacon records sorted in ascending order.
$m^{(x)}$	Identifier of a beacon sensed by device x, $x \in \{a, b\}$.
$s^{(x)}$	Associated signal strength of a beacon sensed by device x, $x \in \{a, b\}$.
$r^{(x)}$	Rank of the associated signal strength in the set of beacon records sorted in ascending order sensed by device x, $x \in \{a, b\}$.
n_x	Number of beacons sensed by device x, $x \in \{a, b\}$.
S_x	Set of beacon records sensed by device x, $x \in \{a, b\}$.

3. Intersection of S_a and S_b (denoted by $S_\cap$): consists of devices seen by both A and B
   $S_\cap = \{(m, s^{(a)}, s^{(b)}) \ \forall m | (m, s^{(a)}) \in S_a, (m, s^{(b)}) \in S_b\}$.

4. Union of S_a and S_b (denoted by $S_\cup$): consists of devices seen by A or B with θ filled in as the "signal strength" for devices that are *not* seen by either device.
   $S_\cup = S_\cap \cup \{(m, s^{(a)}, \theta) \ \forall m | (m, s^{(a)}) \in S_a, m \notin S_b^{(m)}\}$
   $\cup \{(m, \theta, s^{(b)}) \ \forall m | (m, s^{(b)}) \in S_b, m \notin S_a^{(m)}\}$.

5. Set of beacon identifiers in $S_\cap$ and $S_\cup$ (denoted by $S_\cap^{(m)}, S_\cup^{(m)}$):
   $S_\cap^{(m)} = \{m \ \forall m | (m, s^{(a)}, s^{(b)}) \in S_\cap\}$, $S_\cup^{(m)} = \{m \ \forall m | (m, s^{(a)}, s^{(b)}) \in S_\cup\}$.

6. Set of beacon associated signal strength values for device A and B respectively (denoted by $L_a^{(s)}, L_b^{(s)}$):
   $L_a^{(s)} = \{s^{(a)} | (m, s^{(a)}, s^{(b)}) \in S_\cap\}$, $L_b^{(s)} = \{s^{(b)} | (m, s^{(a)}, s^{(b)}) \in S_\cap\}$.

Finally, we synthesized the following features (the first five are previously proposed features in [DEM12, VS07, KH04]):

1. Jaccard distance: $1 - \frac{|S_\cap^{(m)}|}{|S_\cup^{(m)}|}$
2. Mean of Hamming distance: $\frac{\sum_{k=1}^{|S_\cup|} |s_k^{(a)} - s_k^{(b)}|}{|S_\cup|}$
3. Euclidean distance: $\sqrt{\sum_{k=1}^{|S_\cup|} (s_k^{(a)} - s_k^{(b)})^2}$

4. Mean exponential of difference: $\frac{\sum_{k=1}^{|S_\cup|} e^{|s_k^{(a)} - s_k^{(b)}|}}{|S_\cup|}$

5. Sum of squared of ranks: $\sum_{k=1}^{|S_\cap|}(r_k^{(a)} - r_k^{(b)})^2$
   where, $r_k^{(a)}$ (respectively $r_k^{(b)}$) is the rank of $s_k^{(a)}$ ($s_k^{(b)}$) in the set L_a (L_b) sorted in ascending order.

6. Subset count: $\sum_{i=1}^{T} f_i$.
   where T is the sensing duration (in seconds), and

$$f_i = \begin{cases} 1, & \text{if } S_{a_i}^{(m)} \neq \emptyset,\ S_{b_i}^{(m)} \neq \emptyset,\ (S_{a_i}^{(m)} \subseteq S_{b_i}^{(m)} \text{ or } S_{a_i}^{(m)} \supseteq S_{b_i}^{(m)}). \\ 0, & \text{otherwise.} \end{cases}$$

   S_{a_i}, S_{b_i} are the set of records by A and B respectively at the i^{th} second.

By testing these candidate features with our dataset, we chose the ones with better performance (more distinguishable features between co-presence and non-copresence) for the three different modalities:

WiFi: Features 1-5 are used. Identifier (m) is BSSID; and (s) is assigned with the mean of RSSIs for the same MAC address in consecutive multiple cycles. θ is -100 (dBm).

Bluetooth: Features 1,3 are used. Identifier (m) is BDADDR; and (s) is assigned with the average RSSI in consecutive multiple cycles. θ is -100 (dBm).

GPS: Feature 1-6 are used. Identifier (m) is the PRN; and (s) is assigned with the average SNR. θ is 0.

Feature 6 is specially developed for GPS. There is a strong and practical motivation: the GPS SNR is highly dependent on receiver hardware. Even in co-presence, the weaker device (with limited sensitivity) may see only a subset of the stronger one's observation. It is witnessed for many times during the data collection campaign, and features like Jaccard distance perform poorly whereas subset count performs better. When GPS coordinates are available for both devices, we also use the orthodromic distance [GGH89] as an additional feature.

6.2.2 Features for Ambient Audio

We decided to use two features proposed by Halevi et al. [HMSX12], which were found to be the most robust among all algorithms tested: Schurmann and Sigg [SS13],

SoundSense [LPL+09], and Shazam audio fingerprinting [Wan06]. The other features either required careful synchronization between the two audio samples or were highly sensitive to variations in the microphone characteristics of the devices. The chosen features are defined as follows:

- Max cross-correlation:
  $M_{corr}(a,b) = max(Cross\text{-}Correlation(\hat{X}_a, \hat{X}_b))$
- Time-Frequency distance:
  $D(a,b) = \sqrt{(D_{c,time}(a,b))^2 + (D_{d,freq}(a,b))^2}$
  where $D_{c,time}(a,b) = 1 - M_{corr}(a,b)$, and $D_{d,freq}(a,b) = ||\widehat{FFT(X_a)} - \widehat{FFT(X_b)}||$ (i.e., Euclidean norm of the difference).

Here X_a and X_b denote the raw (16-bit PCM) audio signals recorded by A and B. $\hat{X}_a$ and $\hat{X}_b$ denote the normalized signals from X_a and X_b. $\widehat{FFT(X_a)}$ and $\widehat{FFT(X_b)}$ denote the normalized Fast Fourier Transform of the corresponding time-domain signals X_a and X_b.

7 Analysis and Results

7.1 Analysis Methodology and Metrics

We treated contextual co-presence detection as a classification task. All our experiments have been performed using ten-fold cross-validation and Multiboost [Web00], a state-of-the-art algorithm widely used for different types of context recognition tasks, as the classification algorithm. In all experiments, decisions trees (J48 Graft) are used as the weak learners. From each experiment, we record the 2x2 confusion matrix, containing the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN).

The classification performance of contextual co-presence detection directly influences both the security and usability of the underlying ZIA mechanism. In particular, the security of the system is determined by the FP rate as it indicates the probability of T concluding that D (and hence U) is co-present erroneously. Usability, on the other hand, is represented by the FN rate as it determines the probability of T not being able to authenticate U even though U is co-present. In addition to evaluating the FP and FN rates, we consider two metrics for the overall classification

performance: (macro-averaged) F-measure [VR79] and the Matthews' correlation coefficient (MCC) [Mat75].

The F-measure (Fm) uses precision ($\frac{TP}{TP+FP}$) and recall ($\frac{TP}{TP+FN}$) for each class. $Fm_i = 2 \cdot \frac{precision_i \cdot recall_i}{precision_i + recall_i}$, $Fm = \frac{\sum_{i=1}^{c} w_i \cdot Fm_i}{\sum_{i=1}^{c} w_i}$, where i is the class index, $w_i = n_i/N$ with n_i being the number of samples of the i^{th} class and N being the total number of samples, c is the number of classes.

MCC is an approximate statistical measure for deciding whether the prediction is significantly more correlated with the data than a random guess. MCC is related to chi-square statistic for a 2x2 contingency table: $|MCC| = \sqrt{\frac{\chi^2}{n}}$. It can be calculated directly from the confusion matrix as: $|MCC| = \frac{TP*TN-FP*FN}{\sqrt{(TP+FP)*(TP+FN)*(TN+FP)*(TN+FN)}}$. It takes values between -1 and +1, with +1 representing perfect prediction, and -1 total disagreement between prediction and ground truth while 0 represents no better than random guess.

7.2 Effect of Time Budget

Although we collected data for two minutes in each sample, the realistic time budget for ZIA is much smaller (typically 5-15 seconds) due to usability reasons (e.g., being able to unlock a terminal or a door quickly). To see the effect of sampling time on the performance of classification, we consider the performance with different time budgets. For a time budget of n seconds, we only consider the sensor data recorded by the device in a sample within the first n seconds. Table 3 shows the results for the uncontrolled dataset for different time budgets. Although the overall performance is reasonable with a 5-second limit (FN=8.95%; FP=7.14%, Fm=0.921, MCC=0.841), data was often missing from different sensor modalities: among 2303 instances, 80% is without GPS data, 37% without WiFi data, 40% without Bluetooth and 8% without Audio. With a 10-second budget the performance is significantly better than with a 5-second budget as more data is captured by sensors, but it flattens out thereafter. Consequently, we fix a 10-second time budget for all subsequent analyses.

7.3 Single vs Multiple Modalities

Next we focus on investigating the effectiveness of single modality based co-presence detection, and on assessing the potential improvements provided by the fusion of

Table 3: Overall performance vs. time budget

Time Budget (s)	5	8	10	12	15
%FN	8.95	2.19	1.67	1.40	1.49
%FP	7.14	2.67	1.98	2.15	2.15
MCC	0.841	0.951	0.966	0.964	0.964
Fm	0.921	0.976	0.983	0.982	0.982

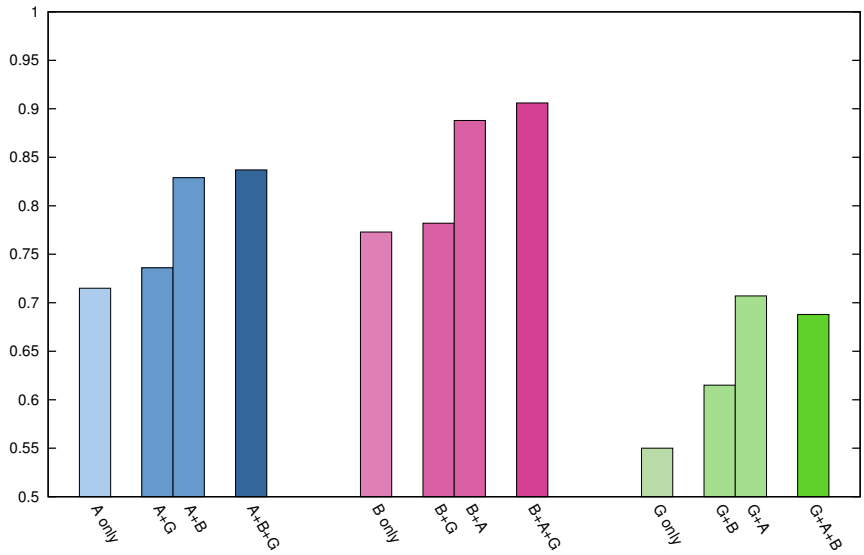


Figure 12: MCC comparison for three modalities Audio (A) - Bluetooth (B) - GPS (G) and their combinations.

multiple context modalities. The results of this investigation are shown in Table 4. For a given sensor modality, we only consider samples that have data from that sensor. To facilitate comparison, we study the fusion of modalities for the same set of samples in each case. Among individual modalities (column 2) WiFi performs best (Fm = 0.989, MCC = 0.978) and GPS worst (Fm = 0.776, MCC = 0.550). Bluetooth and Audio exhibit similar performance with the former (Fm = 0.885, MCC = 0.773) slightly better than the later (Fm = 0.857, MCC = 0.715).

The results for Bluetooth, audio and GPS clearly demonstrate that relying solely on any single one of these modalities is not sufficient for satisfying the usability and security requirements of ZIA. Moreover, from Figure 12 we can observe that

Table 4: Individual modalities vs Fusion of modalities; (A) Audio, (B) Bluetooth, (G) GPS, (W) WiFi

All samples containing Audio (sample size = 2117)								
	A only	A+B	A+G	A+W	A+B+G	A+B+W	A+G+W	A+B+G+W
FN(%)	19.9	12.49	20.41	1.52	12.59	1.52	1.73	1.62
FP(%)	9.28	5.21	7.07	1.59	4.33	1.77	1.59	1.77
MCC	0.715	0.829	0.736	0.969	0.837	0.967	0.967	0.966
Fm	0.857	0.914	0.866	0.984	0.918	0.983	0.983	0.983

All samples containing Bluetooth (sample size = 1600)								
	B only	B+A	B+G	B+W	B+A+G	B+A+W	B+G+W	B+A+G+W
FN(%)	15.54	7.64	18.25	0.74	6.78	0.49	0.49	0.37
FP(%)	7.35	3.55	4.18	1.27	2.66	1.01	1.14	1.01
MCC	0.773	0.888	0.782	0.980	0.906	0.985	0.984	0.986
Fm	0.885	0.944	0.886	0.990	0.952	0.992	0.992	0.993

All samples containing GPS (sample size = 782)								
	G only	G+A	G+B	G+W	G+A+B	G+A+W	G+B+W	G+A+B+W
FN(%)	23.6	14.89	25.28	1.97	18.54	1.69	2.53	1.97
FP(%)	21.36	14.32	13.85	3.52	12.91	3.99	3.52	3.76
MCC	0.55	0.707	0.615	0.944	0.688	0.941	0.938	0.941
Fm	0.776	0.854	0.808	0.972	0.845	0.971	0.969	0.971

All samples containing WiFi (sample size = 2269)								
	W only	W+A	W+B	W+G	W+A+B	W+A+G	W+B+G	W+A+B+G
FN(%)	0.36	0.27	0.45	0.45	0.18	0.18	0.27	0.18
FP(%)	1.83	1.83	1.83	1.83	1.83	1.83	1.92	1.83
MCC	0.978	0.979	0.977	0.977	0.980	0.980	0.978	0.980
Fm	0.989	0.989	0.989	0.989	0.990	0.990	0.989	0.990

the performance of these modalities improves when they are fused with another modality.

7.4 Band Analysis

To see if the performance of an individual modality varied greatly depending on the sampled values, we analyzed the performance separately for samples with values in different ranges ("bands"). Tables 5 shows the results. A band consists of those samples where the records from both devices fall in the range corresponding to that band (e.g., there were 551 samples in which both devices saw only one other Bluetooth device). Several conclusions can be drawn from the table. First, the performance is significantly worse in some bands (e.g., "< 2" for Bluetooth). In a practical ZIA implementation, samples falling in such bands can be filtered out when evaluating contextual co-presence. Second, the performance of GPS naturally improves when more satellites are visible – but within our 10s time budget, GPS performs poorly because the vast majority of the samples contain only one visible satellite.

7.5 Controlled Setting

To assess the robustness of the results with respect to common sources of noise in sensor measurements, such as variations in device placement (pocket vs bag) and variations in the characteristics of the ambient environment (noisy vs quiet), we supplemented the everyday dataset with a limited dataset collected from predefined settings. This *controlled dataset* was collected in order to determine if there was any potential systematic bias as to how our testers collected the data in the uncontrolled dataset. The controlled dataset, contains 94 samples (44 from co-present devices and 50 from non co-present devices) which were collected by two users. All were taken in noisy environments (in crowded areas and noisy streets). In each sample, one device was within an enclosure (pocket or backpack) while the other was exposed (e.g., in the user's hands).

Table 6 shows the performance of the classification in controlled dataset for different sensor modalities (single, and all together). The results do not indicate any clear systematic difference between the two datasets in terms of the classification performance, suggesting that generally the evaluated context sensing mechanisms are robust across variations in environmental characteristics and in device placements.

Table 5: Performance for different modality bands, #IDs = number of device IDs seen and N: number of samples.

Audio						**Bluetooth**						**GPS**					
RMS[a]	N	%FN	%FP	MCC	Fm	#IDs	N	%FN	%FP	MCC	Fm	#IDs	N	%FN	%FP	MCC	Fm
≤ 500	919	11.14	4.09	0.855	0.933	< 2	551	38.12	4.01	0.642	0.826	≥ 1	757	27.12	21.84	0.511	0.757
rest	1198	23.89	16.51	0.594	0.795	rest	1049	10.34	0.72	0.883	0.941	≥ 5	314	19.85	15.30	0.647	0.828
												≥ 10	39	6.25	4.35	0.894	0.949

[a]RMS refers to audio signal's root mean square level.

Table 6: Controlled setting (sample sizes in brackets)

	Single modality				All modalities			
	%FN	%FP	MCC	Fm	%FN	%FP	MCC	Fm
Audio(74)	18.18	16.67	0.644	0.825	4.55	3.33	0.917	0.960
Bluetooth(94)	4.44	2.04	0.936	0.968	4.44	0	0.958	0.979
GPS(37)	18.18	26.67	0.552	0.784	4.55	0	0.946	0.973
WiFi(88)	4.44	2.33	0.932	0.966	4.44	2.33	0.932	0.966

Table 7: Controlled vs. Uncontrolled settings

	Controlled				Uncontrolled			
	Single		All		Single		All	
	MCC	Fm	MCC	Fm	MCC	Fm	MCC	Fm
Audio	0.644	0.825	0.917	0.960	0.715	0.857	0.966	0.983
BT	0.936	0.968	0.958	0.979	0.773	0.885	0.986	0.993
GPS	0.552	0.784	0.946	0.973	0.55	0.776	0.941	0.971
WiFi	0.932	0.966	0.932	0.966	0.978	0.989	0.980	0.990

The performance of WiFi exceeds other modalities, providing near perfect results for the uncontrolled dataset. One possible reason is that in most of the samples in this dataset, the two devices are either very close or very far from other. This is reasonable since our focus is on preventing relay attacks where the common case is for the attacker to attempt relaying when the two legitimate devices are far apart. However, it is reasonable to ask whether the FP rate of WiFi will remain as high when the non co-present devices are much closer to each other. To investigate this issue, we conducted another small-scale controlled experiment where we collected data from four devices. Pairs of devices were placed in two offices that were approximately 15 meters apart, and 100 samples containing all sensor modalities were recorded for a duration of two hours, in which 50% is from the co-present pair and 50% from the non co-present pair. The results show that (a) WiFi performance degrades slightly with FP% rising from 1.83% to 7.14% and (b) the fusion of multiple sensor modalities does improve the FP rate (to 4.76%) compared to using WiFi alone.

Table 8 summarizes the results.

Table 8: Performance for low-distance non co-presence

Modalities	FN(%)	FP(%)	MCC	Fm
WiFi only	10.0	7.14	0.826	0.913
All	4.0	4.76	0.912	0.957

7.6 Personalization

So far, we used data from all users to create a common user-independent model. A natural question is whether a user-specific model would perform better. To see this, we separated data by individuals and used them to train "personalized" models. Note that a personalized model is trained using data from only two devices, whereas the common model was computed using data from multiple pairs of devices. Accordingly, the user-specific evaluation also assesses the robustness of our results hardware variations. Table 9 summarizes the results for three users (uncontrolled data set) with the most data. Since a personalized model is more cumbersome (it would require each user to train the model), it has to be significantly better than the common model to justify its use, which is not the case based on our results.

7.7 Summary

We showed that WiFi is the most effective sensor modality for resisting relay attacks against ZIA schemes based on contextual co-presence detection. We also showed that for all combinations of sensor modalities, fusing all available modalities will improve security (low false positives) of such ZIA schemes while retaining the high level of usability (low false negatives) characteristic of ZIA.

8 Adversarial Analysis

So far, we assumed the Dolev-Yao [DY83] adversary model. However, the Dolev-Yao model is intended for analyzing traditional communication protocols. Attacks against the integrity of context sensing are known. For example, Tippenhauer et al. [TRPv09] showed how to defeat WiFi-based positioning systems with inexpensive equipment. Our proof-of-concept attack against BlueProximity was based on changing the Bluetooth device address on the Bluetooth controller on a PC. It is not difficult to imagine an attacker capable of generating same dominant sound near a

Table 9: Analysis of personalized model for individual users, blanks indicate insufficient data.

Modalities	**User1**					**User 2**					**User 3**				
	N	%FN	%FP	MCC	Fm	N	%FN	%FP	MCC	Fm	N	%FN	%FP	MCC	Fm
Personalized Model: Trained and tested with personal data															
Audio	494	0.76	0.85	0.984	0.992	228	21.55	18.58	0.599	0.799	209	6.88	18.37	0.737	0.905
Bluetooth	435	0.77	0	0.99	0.995	198	3	4.08	0.929	0.965	133	-	-	-	-
GPS	52	31.58	15.15	0.539	0.787	20	-	-	-	-	59	-	-	-	-
WiFi	496	0.76	0	0.992	0.996	229	0.86	0.88	0.983	0.991	219	1.25	1.67	0.966	0.986
All	496	0.76	0	0.992	0.996	229	0.86	0.88	0.983	0.991	220	0.63	3.33	0.966	0.986
Common Model: Trained with all data and tested with personal data															
All	496	0	0	1	1	229	0	2.65	0.974	0.987	220	0	3.33	0.977	0.991

pair of devices in two different locations. All this demonstrate the need for a stronger adversary model that would cover the capability for interfering with context sensing.

Prior work on contextual co-presence largely limited their security analysis to benign failures only [HMSX12]. The occasional exceptions involved testing resistance against a few types of attacks interfering with context sensing [VS07]. In contrast, we argue that there is a need for a precise but realistic formulation of a contextual adversary without having to spell out specific attacks. Once such an adversary model is defined, different contextual co-presence schemes can be analyzed with respect to such an adversary.

8.1 Adversarial Model

Manipulating contextual information may require conspicuous equipment (like fake access points) or actions (like playing loud music). Observe that D is usually carried by the human user U whereas T may be unattended. We therefore postulate that A_t, the attacker near T, can more easily interfere with the context sensing of T undetected than can A_d with D. Furthermore, we assume that it is infeasible for an attacker to *suppress* existing context signals. Therefore, one way to characterize the context attacker is as follows:

- A_d can perfectly measure the context information that D would sense,
- A_t can fool T into sensing any context information it chooses; Specifically A_t can receive context information from A_d, reproduce it perfectly near T; and
- A_t (A_d) cannot suppress any other ambient context information from being sensed by T (D).

While this is still a very powerful attacker, analyzing our features for classification with respect to such an attacker may give some insights into the relative security of different sensor modalities.

8.2 Results

For RF-based sensors, the context adversary as defined above can be modelled by replacing S_b with $S_a \cup \{(m, s)\ \forall (m, s) \in S_b, m \notin S_a^{(m)}\}$. For audio, since raw audio data is additive, the adversary can be modelled replacing X_b by $X_a + X_b$. To estimate the effect of such an adversary, we took the following approach. We used

Table 10: Performance in adversarial setting

Modalities	**Audio**				**Bluetooth**				**WiFi**			
	FN(%)	FP(%)	MCC	Fm	FN(%)	FP(%)	MCC	Fm	FN(%)	FP(%)	MCC	Fm
Single modality	16.14	**100**	-0.298	0	15.17	**99.11**	-0.268	0.281	0.45	**75.17**	0.365	0.556
Difference from Table. 4	-16.77	+91.23	-0.905	-0.857	-0.37	+91.76	-1.041	-0.604	+0.09	+73.34	-0.613	-0.433
Fused of multi-modalities	1.75	**3.01**	0.952	0.976	0.37	**1.22**	0.984	0.992	0.45	**65.8**	0.444	0.625

our uncontrolled dataset with ten-fold validation. Training is done using the nine folds of the dataset as before. But the test data was transformed as described above to model the effect of the context adversary.

The results for WiFi, Bluetooth and audio are shown in Table 10. (We did not include GPS in this analysis because GPS performed poorly to begin with and spoofing GPS is likely to be harder than the other modalities. Nevertheless, we expect the adversary model to hold for GPS as well and is likely to yield similar results.) The first and the third row show the performance of individual and multiple sensor modalities in the presence of the context attacker. All individual modalities are insecure with respect to such an attacker. *If* we can assume that the attacker is capable of compromising only one sensor modality at a time, the use of multiple modalities restores security in the case of audio and Bluetooth, thanks to the effect of WiFi. In the case of WiFi itself, the fusion of the other modalities results in only a modest increase in security. The second row of Table 10 shows the difference in false positive rate with respect to the same modalities in the absence of the attacker. False positive rate of Bluetooth and Audio has comparable increases (+91.76% and +91.23% respectively) while the increase in WiFi is a more modest 73.34%, implying that although the powerful context attacker is very successful across the board, WiFi performs somewhat better than the other modalities against such an attacker.

9 Application: BlueProximity++

We developed BlueProximity++ application by incorporating our contextual co-presence detection module into BlueProximity [Blu]. We wanted the authentication process to be more secure against relay attacks without impairing the easy-to-use characteristics. This section presents the design and implementation of BlueProximity++, and a small-scale user study.

9.1 Design

9.1.1 Access Control Scheme

BlueProximity++ supports two access control schemes: the fallback scheme using Bluetooth RSSI only, and the advanced scheme where contextual co-presence is

incorporated. The two schemes were designed for the subsequent usability evaluation to make comparisons.

If the fallback scheme is enabled, as presented in Figure 13, The terminal T triggers lock event when the measured Bluetooth RSSI of the device D decreases below the *locking threshold* $\Theta 1$; T triggers unlock event when the measured Bluetooth RSSI of D increases above the *unlocking threshold* $\Theta 2$.

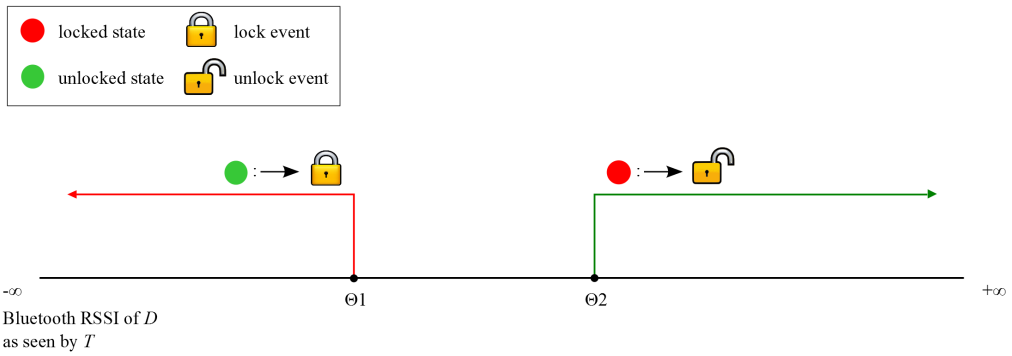


Figure 13: BlueProximity++: co-presence detection using Bluetooth RSSI only.

Figure 14 shows our advanced scheme for BlueProximity++. We added an additional threshold $\Theta 3$ *for triggering context scans*, keeping $\Theta 1$ and $\Theta 2$ untouched. When the measured Bluetooth RSSI of D measured by T rises above $\Theta 3$, T starts scanning context information on both T and D (triggered via a communication channel). When the Bluetooth RSSI of D rises above $\Theta 2$, T starts the contextual authentication process with the previous context scan result from both T and D (if the previous context scan result is missing or expired, a new context scan session is triggered). The contextual authentication process feeds the previous scan result into an off-line classification model (mentioned in Section 7), and gets the co-presence indicator as output. T's screen unlocks upon "co-present" result (i.e., authenticated successfully), otherwise it stays locked (i.e., authentication fails).

Fluctuations in the Bluetooth signal has a significant influence on the performance of BlueProximity++. Bluetooth RSSI measurement is highly sensitive to other physical factors [Gow12, Dar12] such as shadow fading (e.g. with water or human body as obstacles), multi-path fading (e.g. in a long and narrow space), and antenna polarization. To mitigate the fluctuation, we attempted to smooth the Bluetooth RSSI of D measured by T. We conducted experiments to evaluate the fluctuation and designed a smoothing algorithm. As a result, we observed steadier Bluetooth RSSI curves after adopting the smoothing algorithm (see Appendix B).

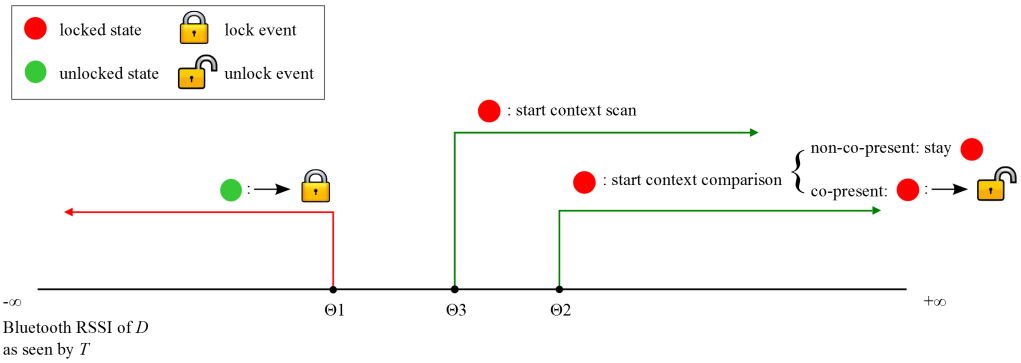


Figure 14: BlueProximity++: co-presence detection using context comparison and Bluetooth RSSI.

9.1.2 Corrective Feedback Scheme

We added support of user feedback in BlueProximity++ to gather data for user study and later performance evaluation, and also to give the means for users to correct wrong access control decisions. We considered the following key factors that introduce authentication errors:

1. The contextual co-presence classification model we used gives a low but non-negligible rate of false prediction.

2. The off-line classification model resulting from the generalized dataset might not fit well with personalized perception of "co-presence" from specific users.

3. Disruptions in Bluetooth caused by many external factors (shadow fading, multi-path fading, antenna polarization, etc.) can lead to false negatives.

In the feedback scheme, the user receives the notification of the current access control events on D's UI. The notification (Figure 15) is designed to ask for user indication of ground truth (whether or not being near T). By clicking the green button, the user provides the affirmative feedback (true positive or true negative). By clicking the red button, the user provides the corrective feedback (false positive or false negative) which triggers the corrective access control events on T as expected. All such events and user feedback are recorded in local databases for further data analysis and evaluation.

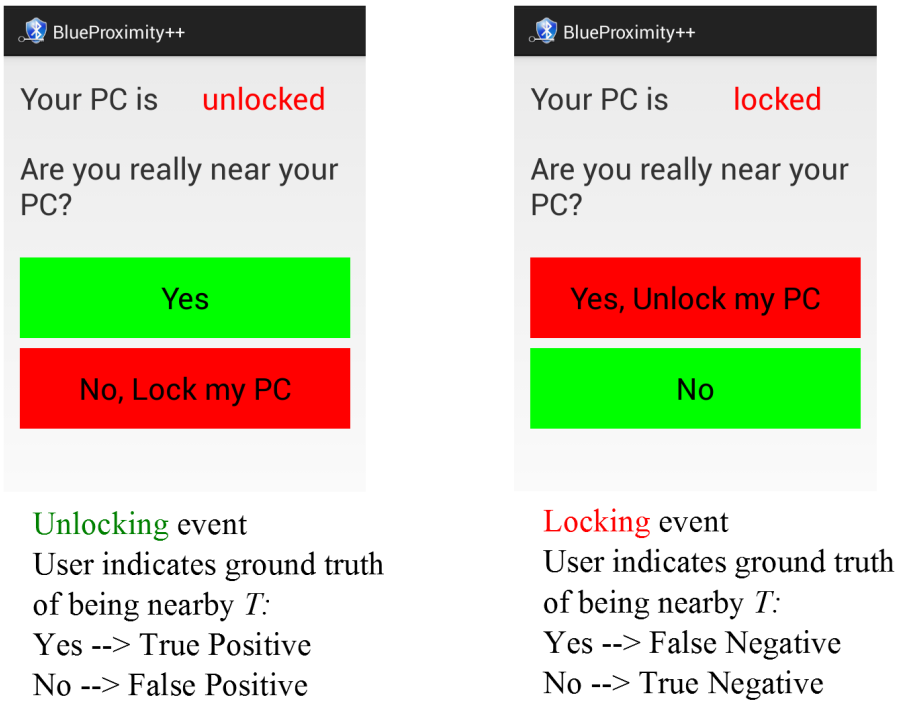


Figure 15: Notification of access control events, asking for ground truth feedback.

9.1.3 Architecture

BlueProximity++ application comprises two parts: the terminal part written in Python and C running on a Linux PC (T), and the device part on D written in Java running on an Android device (D). Figure 16 depicts the high-level design of BlueProximity++ architecture.

On T, the application provides the major functionalities: detecting Bluetooth signal from D, conducting context scans, and maintains a communication channel with D.

On D, the application maintains a communication channel with T and a local context scan module.

The *communication channels* between T and D include the Bluetooth connection reserved for binding T and D for one user in the initialization phase, and a communication channel via a proxy server on the Internet reserved for exchanging commands to trigger context scan and messages for user feedback. It is designed both for user study and the corrective feedback for giving false negative and false positive user indications.

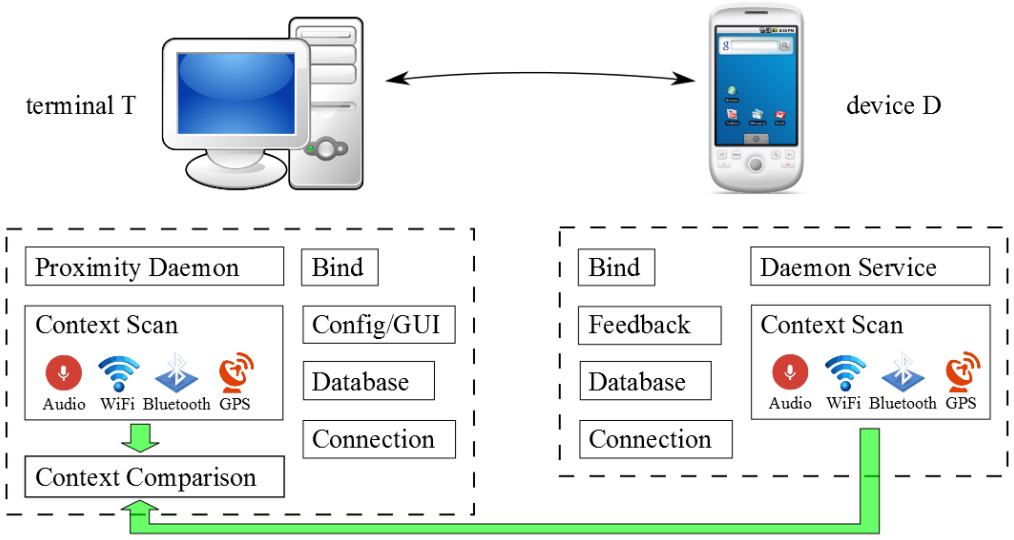


Figure 16: Architecture of BlueProximity++.

9.2 Implementation

On T, *Proximity Daemon* is a coordinator service running in the background, monitoring the Bluetooth RSSI dynamics, and harnessing all other modules. *Context Scan* module triggers contextual utilities (audio, WiFi, Bluetooth, GPS) on T to start new scanning tasks. Local scanning results are stored in a sensor object, ready for context comparison after D's scanning result is received. *Config* module is designed for synchronizing the configuration options in a file for initialization, coupled with a GUI is designed for configuration. *Bind* module is designed to handle the binding procedure with D via a Bluetooth RFCOMM connection. *Connection* module handles the communication channel with D for sending context scan commands, accepting scan results, and exchanging lock/unlock messages for user feedback. There is a Sqlite database recording all lock/unlock events and user feedback, dedicated for user study.

On D, *Daemon Service* is running as a foreground service, harnessing all other modules. *Feedback* module is the activity for collecting user indication of ground truth data, triggered by arriving notification message of lock/unlock events. *Context scan*, *Bind*, and *Connection* module functions the same way as their counterpart on T. A *TinyDB* persistent database is used for recording binding meta data and maintaining a local copy of user feedback.

Since Bluetooth cannot support reliable data transmission beyond its working distance (i.e. 10 meters for most Bluetooth devices following the Bluetooth 2.1 specification), we added the network communication channel. The network communication channel maintains a communication via a proxy reachable via the Internet, to enable the bound T and D to communicate with each other even when they are out of Bluetooth range. It is implemented using the Advanced Message Queuing Protocol (AMQP) [AMQ]. Messages are sent and received in different message queues (tagged with unique "channel identifiers") through a proxy where messages are routed in a publish-subscribe pattern. And messages in the queues are encrypted using AES-CBC and verified using message authentication code (HMAC-SHA-256). The shared key was generated and agreed in a previous binding procedure via Bluetooth channel.

To elaborate how different modules in BlueProximity++ collaborate to achieve tasks, we are going to describe the details of the following three tasks: binding, locking, and unlocking.

9.2.1 Binding

Figure 17 illustrates the procedure to bind D to T. On D, Bluetooth is temporarily set to be discoverable, and an idle port is used to register a service on the Bluetooth Service Discovery Protocol (SDP) server (step 0). The *Service UUID* is known to BlueProximity++ on both T and D. Then on T, BlueProximity++ starts to inquire nearby candidate devices which are displayed to the user (step 1). After user selection, the device name and MAC address are confirmed, and the port number is found by matching the service list with known *Service UUID* (step 2). Soon after that, Bluetooth RFCOMM connection is initiated by T (step 3) and confirmed by D (step 4). By this time, the port (of D) taken for binding has been released and used for this connection, which guarantees that no other services will preempt the port.

T and D need exchange meta information (step 5) to complete binding. Here, device name, Bluetooth MAC address, port number, and device UUID are exchanged. Besides, T generates a new salt value (a random number) and sends it to D. The shared key is generated by concatenating UUID of T, UUID of D, and the salt. The pair of "channel identifiers" are generated by hashing the concatenation of both UUIDs in different orders. Both the shared key and "channel identifiers" are prepared for further network communication channel.

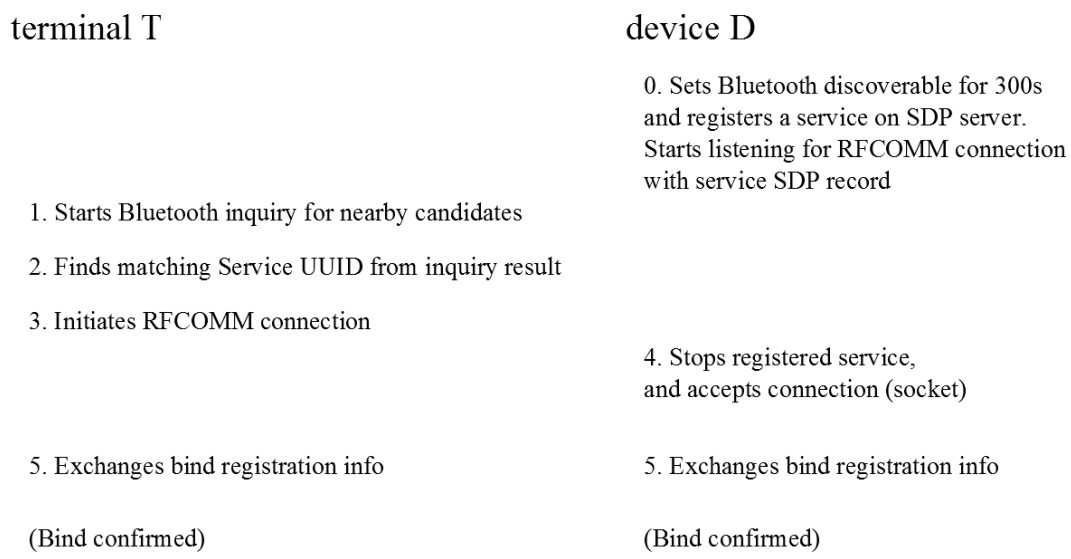


Figure 17: BlueProximity++ Binding procedure.

9.2.2 Locking

To minimize the energy consumption, we designed the locking mechanism without contextual scanning and comparison, since we emphasized on authentication instead of de-authentication (but considered to incorporate co-presence detection into locking decision in the future). We defined the locking distance as the threshold for Bluetooth RSSI, because we assume Bluetooth RSSI is positively correlated with physical distance in steady environment.

The Locking algorithm runs in the background continuously. It does nothing if the screen is already locked. If it is in the unlocked state and the RSSI goes below the locking threshold and stays below for a sustained period ($D_{confirm}$ iterations), then the screen is locked.

Algorithm 1 shows the details of making locking screen decisions. S is the variable for the realtime Bluetooth RSSI of D measured by T. $State$ is the variable for the state of T's screen, with two possible values $UNLOCKED$ (i.e., screen is unlocked) and $LOCKED$ (i.e., screen is locked). $\Theta 1$ refers to the customizable threshold for triggering lock events. $D_{confirm}$ is the customizable length of the sustained period, and acc is the accumulator the sustained period.

Algorithm 1 Locking

```
Θ1 ← locking threshold for RSSI
D_confirm ← confirming window
while true do
    S ← Bluetooth RSSI of D measured by T
    State ← screen state
    if State = UNLOCKED then
        if S ≥ Θ1 then
            acc ← 0
        else
            acc ← acc + 1
            if acc ≥ D_confirm then
                acc ← 0
                trigger lock event
                State ← LOCKED
    sleep for 1s
```

9.2.3 Unlocking

We incorporated the contextual co-presence detection when designing the unlocking mechanism. The actual context comparison should take place right before D comes into the unlocking range of T. To maximize the chances of both T and D having sufficiently recent context scans at this point, it is desirable to trigger context scans at a lower threshold. It would be better to learn from the user behavior, but we assigned a fixed coefficient in current release, i.e. the threshold to trigger context scan is double the unlocking threshold.

The Unlocking algorithm runs in the background continuously. It does nothing if the screen is already unlocked. If it is in the locked state, the RSSI goes above the unlocking threshold and stays above for a sustained period ($D_{confirm}$ iterations), and the recent context comparison indicates co-presence, then the screen is unlocked.

We assigned a timeout value to each contextual comparison result, i.e. the decision (co-presence or non-copresence). So in the main thread, the following conditions should be tested: whether the decision is available (not empty), whether the decision has expired, whether there is already a context scan running.

Algorithm 2 Unlocking

```
Θ2 ← unlocking threshold for RSSI
D_confirm ← confirming window
Θ3 ← threshold for triggering newcontext scanning
prevResult ← previous contextual comparison result
while true do
    S ← Bluetooth RSSI of D measured by T
    State ← screen state
    if State = LOCKED then
        if S < Θ2 then
            acc ← 0
        else
            acc ← acc + 1
            if acc ≥ D_confirm then
                acc ← 0
                if prevResult is null or prevResult expired then
                    if context scan is not running then
                        trigger new context scan and update prevResult
                else if prevResult = co-presence then
                    acc ← 0
                    trigger unlock event
                    State ← UNLOCKED
                continue
        if S >= Θ3 then
            if prevResult is null or prevResult is expired then
                if context scan is not running then
                    trigger new context scan and update prevResult
    sleep for 1s
```

Algorithm 2 explains the details of making unlocking screen decisions. $\Theta 2$ refers to the customizable threshold for triggering unlock events. $\Theta 3$ refers to the threshold for triggering a new context scan (currently hardcoded as double $\Theta 2$).

9.3 Usage

We provide Debian and Android packages. BlueProximity++ registers itself as a start-up program after installation. The first step for a new setup is binding T and D. User are required to operate on both T and D to complete binding procedures. After binding, meta information for each other will be registered and displayed on user interfaces, as shown in Figure 18.

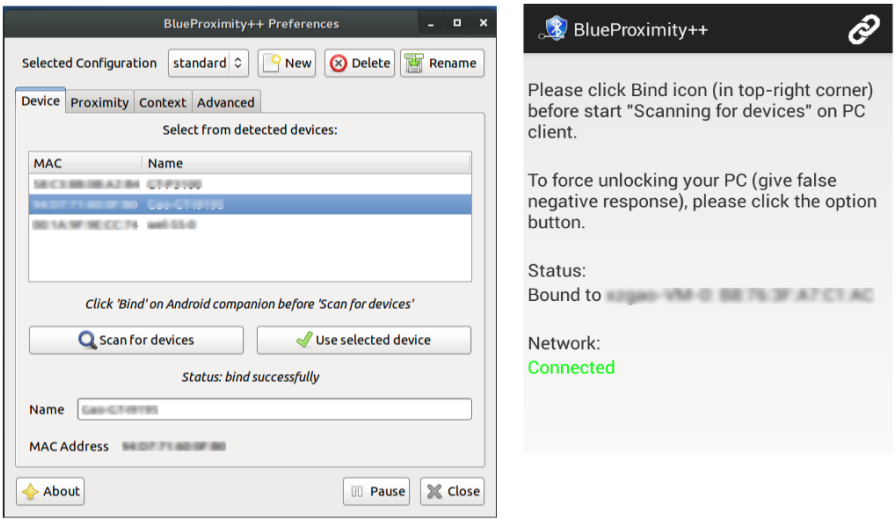


Figure 18: Configuration UIs for binding.

Users are allowed to switch between two modes - with or without contextual co-presence detection - by toggling the preference entry in context setting user interface. Originally, we integrated a simple toggle button in *Proximity Details* as shown in Figure 19. By default, contextual co-presence detection is enabled in BlueProximity++.

After the user study, we decided to update the configuration design to provide fine-grained control of context settings. As shown in Figure 20, we designed a standalone tab *Context Setting* with a group of radio buttons. The default option remains the same. Users are allowed to disable the contextual co-presence detection temporarily for a given period or until D is reachable (in case of network failure). We also provide two baselines in context setting: the fallback co-presence detection with

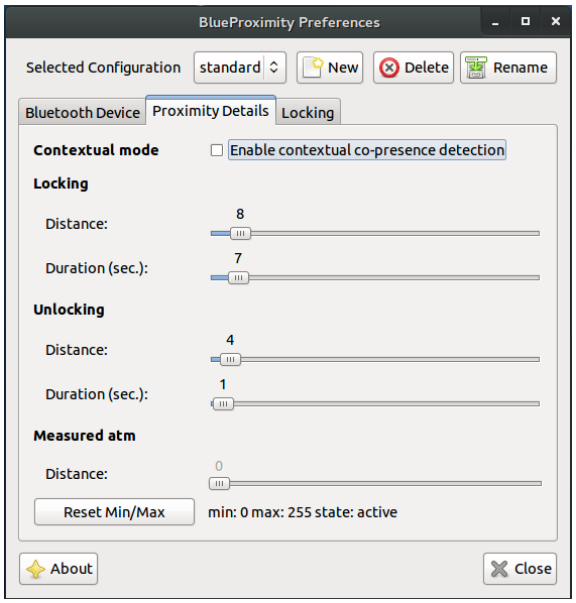


Figure 19: Original design of context setting UI.

Bluetooth RSSI only via the *Disable permanently* button, and completely disabling Zero-Interaction Authentication via the *Pause* button.

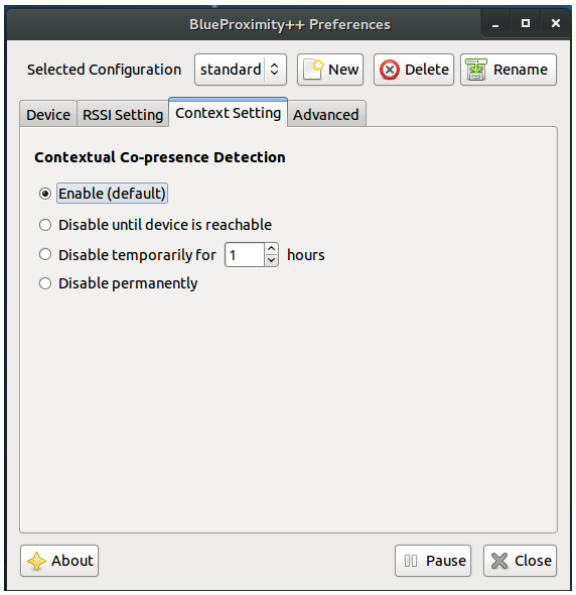


Figure 20: Current fine-grained context setting UI.

In normal use cases, when the user (with registered device) come inside the nearby range of D, T's screen will automatically unlock within a few seconds; and when the user leaves far away from T's nearby range, T's screen will automatically lock within a few seconds. Upon locking or unlocking events, the user will receive prompts from

D, asking for ground truth indication. As shown in Figure 21, the prompt shows the current specific screen state "locked" or "unlocked", and asks the user whether being inside the nearby range. By clicking the green or red buttons, the user is providing the ground truth data of the current event as user feedback. The responses "Yes" or "No" indicates True/False Positive/Negatives of co-presence.

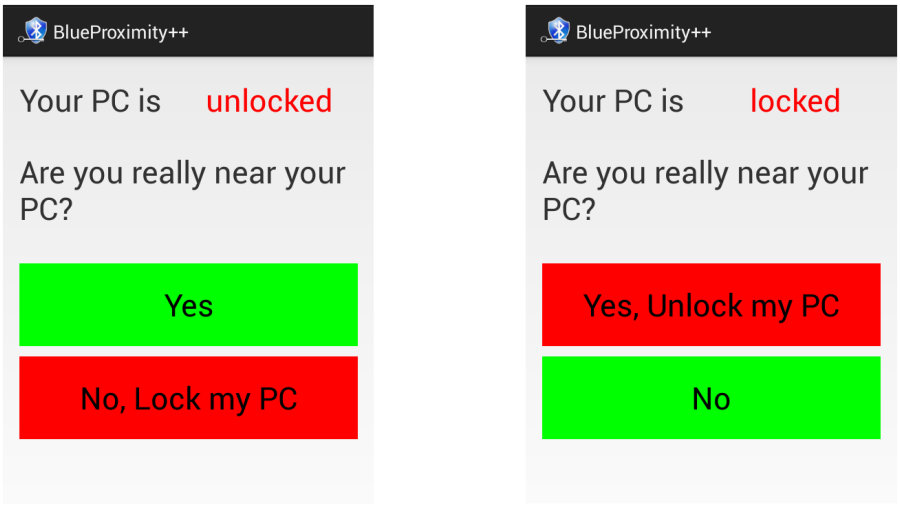


Figure 21: Prompts for user feedback.

Additionally, we provide an additional fallback solution for false negative cases (screen stays locked while the user is nearby for a few seconds). Users can unlock T's screen manually from the menu button of the Android app of BlueProximity++. This is also recorded as the user feedback in database for user study.

9.4 User Study

After implementing BlueProximity++, we launched a user study to assess its usability. The user study was designed to evaluate the usability of BlueProximity++, and to compare the two schemes (i.e., "Bluetooth RSSI only" or "Context comparison and Bluetooth RSSI") in terms of usability.

9.4.1 Description

Participants: We invited ten participants: seven of them are from Helsinki, Finland, and three are from Birmingham, the United States. There is a broad distribution of nationalities: Bangladesh, China, Finland, France, India, Italy, and the US.

But the gender and age distribution are quite biased due to the limited scale. All the participants are with higher educational backgrounds, and they shared the similar professional background in Computer science as researchers or students. They are considered as professionals in computer skills (with the average of self evaluation scoring 8.6 ($\sigma = 1.5$) on a 0-10 scale). Each participant possessed a personal computer and a mobile device for daily use (mostly). All the participants used passwords to secure their personal computers. Five participants indicated highly concerned attitude about the security of their personal computers ("very much"), three showed moderately concerned attitude ("somewhat"), while two had little concern about the security ("a little"). The demographics of the participants are summarized in Table 11.

Type	Information
Age	25-31 years old
Gender	Female (1), Male (9)
Nationality	7 countries
Education	M.Sc (7), Ph.D (3)
Computer Skill	8.6 ($\sigma = 1.5$) on a 0-10 scale
Use of Mobile Device	Daily(9), Several times a week(1)
Use of Personal Computer	Daily(8), Several times a week(2)
Use of Password	Yes(10)
Security Concern	Very much(5), Somewhat(3), A little(2)

Table 11: Demographics of participants

Materials: Each participant used a Linux personal computer as the primary working terminal (T) and an Android phone or tablet (either their own or provided by us) as device (D). The participants were required to fill in all five documents for different phases: a consent form and a demography questionnaire prior to the user study, a System Usability Scale (SUS) questionnaires [Bro96] and a comparison questionnaire right after the user study, and an open-ended feedback questionnaire as an reflection. SUS questionnaire is an standardized method to generate the aggregate score of usability (out of 100) for the target system in a user study. The questionnaires are attached in the Appendices section. At the end of the study, each participant was given a voucher or movie ticket valued 25€ (or US $30) as the reward.

Design: A within-subjects design was adopted to test the influence of different co-presence detection schemes on usability perceived by participants. In order to

mitigate potential learning effects, we decided to divide all participants into two balanced groups, and divide the time of study into two equal rounds. Before the first round, participants were randomly assigned to form two groups of five (*Group I* and *Group II*). During the first round, those in *group I* were required to use BlueProximity++ with only Bluetooth-signal-strength-based co-presence detection. Those in *group II* were required to use BlueProximity++ with contextual and Bluetooth-signal-strength-based co-presence detection. The choice of co-presence detection schemes was set up via the configuration user interface. During the second round, the preference of co-presence detection schemes was toggled to switch to the other group. The participants were asked to try responding to all access control events by indicating ground truth via the notifications on mobile devices.

Procedures: The user study was organized in two rounds (roughly one week for each round), and three face-to-face meetings were arranged with different purposes:

- *Orientation meeting*: Before the 1st round, the participants were asked to sign the consent form, and fill in the demography questionnaire anonymously. Then we helped the participants setup BlueProximity++ on their devices, and guided them to familiarize the usage of BlueProximity++.

- *First post-test meeting*: After the first round, the participants were asked to fill in the post-test SUS questionnaire about their experience during the previous week. The group ID was attached in each questionnaire. Then we gathered the history data of access control events and feedback from the participants. Finally, as the setup procedure for the second round, we switched the preference of co-presence detection schemes in their configurations.

- *Second post-test meeting*: After the second round, the participants were asked to fill in another post-test SUS questionnaire. Same as in the previous meeting, we gathered history data from the participants. Additionally, they were asked to fill in the comparison questionnaire to compare the perceived usability between the two rounds.

- After the user study, we additionally asked the participants to provide their overall feedback via the open-ended questionnaire.

Support: During the user study, we provided technical support via email or in person, responding to participant feedback promptly. We met with the following major technical issues:

- *Incompatibility with environment*: BlueProximity++ was designed for all Debian-based Linux distributions, but was actually implemented and tested on Ubuntu 12.04 only. It had not become an issue until we met with a variety of Linux distributions and desktop systems on participants' own computers. For instance, one of the participants used Gnome 3 desktop where gnome-screen-saver support (the built in interface to trigger screen lock) was deprecated. We fixed this issue by integrating another DBus command, supporting a broader range of desktop systems. Incompatibility was fixed during the head of the first round.

- *Internet connectivity*: Some participants had problems when network connection was disrupted. The communication channel between T and D is essential for transmitting both context scan results as while as user responses. In case of disconnection, the results of contextual co-presence would be always false negative (i.e. screen stays locked when nearby).

- *Suspension*: Originally, terminal suspension was not taken into account when designing BlueProximity++. However, as a participant reported, the application became inactive after the laptop recovered from suspension. We found that the locked/unlocked state became inconsistent because the power manager had ignored the pending lock/unlock commands in suspension, but the state boolean was changed after triggering lock/unlock commands in our implementation. We fixed this issue by adding a periodic detection of screen state to synchronize the screen state after recovering from suspension.

9.4.2 Results

SUS scores: The SUS scores of using BlueProximity++ with the two different schemes of co-presence detection are summarized in Table 12. And the distributions are visualized in Figure 22. We observed a large variance of SUS score among different participants. The SUS score average for both schemes are slightly under the required level beyond which the system is considered easy to use for both cases.

As summarized in Table 12, the SUS scores for the two co-presence detection schemes are quite close. We used the Wilcoxon rank-sum test to estimate the similarity. The difference was not statistically significant ($Z = 0.49, p = 0.63$). The similarity of usability is also observed in the participants' responses to the comparison questionnaire: four preferred the simple scheme based on Bluetooth RSSI only, five preferred

the advanced scheme combined with context comparison, and one remained neutral. With these results, we conclude that there is not sufficient evidence to conclude that the two schemes resulted in different perceived usability.s

Table 12: SUS scores for the two co-presence detection schemes

	Scheme: Bluetooth RSSI only mean (std dev)	**Scheme: Context comparison + Bluetooth RSSI mean (std dev)**
Average SUS score	67 (23)	63 (22)

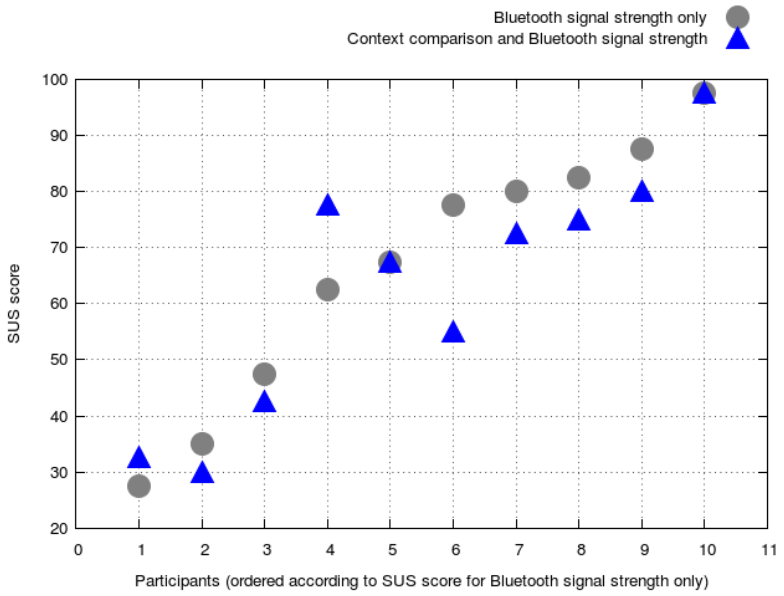


Figure 22: SUS score distribution

Comparison with ground truth:

BlueProximity++ provides the functionality to collect ground truth as part of corrective feedback (as described in Section 9.1). Table 13 shows how the two co-presence detection mechanisms performed in relation to the ground truth. We do not report true positive/true negative figures because in the post-study debriefing several participants indicated that they provided ground truth response only when the access control decision was incorrect. To compare the two co-presence detection methods, we consider the overall number of incorrect decisions, i.e., we compare the proportion of false negative and false positive decisions across all participants. Using Z-test for two population proportions, we found that the differences between the two mechanisms are not significant (at $p < 0.05$) which is in line with the user percep-

tions as we saw above. However, the initial technical problems (cf. Section 9.4.1) would have impacted the ground truth information.

Table 13: Comparison with ground truth

BlueProximity++	**#Total**	**#FP(%)**	**#FN(%)**
Scheme: Bluetooth RSSI only	832	77(9.25%)	86(10.34%)
Scheme: Context comparison + Bluetooth RSSI	774	64(8.27%)	96(12.4%)
Z-test 2 population proportions		Z=0.7 p=0.48 (p>0.05)	Z=-1.3 p=0.19 (p>0.05)

9.4.3 Qualitative Insights

By summarizing the responses of the open-ended questionnaires, we got the following qualitative insights:

Energy consumption: During the user study, six of ten participants did not notice any difference in battery usage of their Android devices compared with their regular usage without BlueProximity++. Two participants claimed their perceived higher battery usage. And the remaining two indicated their lack of battery usage baseline for comparison.

Locking/unlocking policies: One participant reported that he prioritizes zero-interaction *de*-authentication rather than authentication: *"I would think it could be nice to automatically lock (and only lock) the screen when mobile is going away."*. Another participant sharing an office with four other colleagues expected shorter time delay before triggering the locking events: *"I feel that I would like the laptop to lock earlier than it does now so that the app can be used even in a somehow crowded environment. At the moment the laptop locks when I'm a bit too far for my likings"*. Although participants could tune up the relevant thresholds to finally change the physical distance at which locking and unlocking events are triggered, it would be a better solution to adopt an online classification model trained and updated periodically to comply with personal preferences.

Comparing with password-based authentication: Four participants prefer zero-interaction authentication than unlocking a PC screen with passwords, while three prefer using passwords than zero-interaction authentication. And the remaining three thought that both methods are needed. One participant commented: *"my mother and sister always leave their mobile phone somewhere, and it could be easily taken and used*

to unlock their computer without them knowing it if someone wanted to". The only female participant claimed her inconvenience at home where she did not always take the device with her *"usually the locking would trigger when I wandered into a different room with my laptop at home and forgot to bring my mobile device with me"*. Previous research also revealed that people do not always carry phones with them [DWF+11]. These observations provided another insight that the prevailing wearable devices with sensing capabilities (e.g., smart watches, glasses, wristbands, or rings) could be the better candidates for ZIA.

9.4.4 Design Improvements

We made several improvements of BlueProximity++ on the basis of the user study results. The remarkable changes are described as blow.

Bluetooth channel for context data: Due to the Internet connection issue, we decided to use Bluetooth as an alternative channel in addition to the AMQP channel via a proxy. In the improved design, access control event notifications as well as context scan results are sent on both channels, when both T and D ar inside the working range of Bluetooth (normally 10m).

Fine-grained control of co-presence detection: As mentioned in Figure 20 of Section 9.3, we designed a standalone tab "Context Setting" with a group of radio buttons. The default option remains the same. Users are allowed to disable the contextual co-presence detection temporarily for a given period or until D is reachable (in case of network failure). We also provide two baselines in context setting: the fallback co-presence detection with Bluetooth RSSI only via the "Disable permanently" button, and completely disabling Zero-Interaction Authentication via the "Pause" button.

10 Evaluation

Our approach using contextual co-presence detection with multiple sensor modalities to enhance ZIA satisfies the preset requirements in Section 4:

(i) Our approach was designed and proved to be more secure than existing solutions using single modalities. In our solution, ZIA gets additional security benefits by fusing multiple sensor modalities (as in Section 7); and the small-scale user study indicates similar improvement of security level(Section 9.4).

(ii) Our approach brought a comparable level of usability, retaining the benefit of ease-to-use from ZIA (Section 7, 9.4).

(iii) Our approach is applicable to a variety of ZIA scenarios. It was designed for users with various commodity computing devices.

11 Conclusion and Future Work

We presented an approach - using contextual co-presence detection with multiple sensor modalities - to strengthen ZIA models against relay attacks without sacrificing the usability benefits. We compared the performance of various single sensor modalities (audio, WiFi, Bluetooth, GPS), and proved the improvement of security level by fusing multiple sensor modalities. We also implemented a demonstrative ZIA system augmented with our contextual co-presence approach, and evaluated the performance in a user study.

We plan to launch a larger-scale user study in the near future. We will use the lessons learned from the recent user study, make the system more robust to the diverse Linux platforms and Internet connection exceptions. We will also maintain a real-time feedback mechanism for the future user study by recording each user's ground truth responses on the server. We will become more responsive to user inactivities when we find limited ground truth responses from the server-side record.

For the development of contextual co-presence models: we built the demonstrative system with the global classification model, which inevitably annoys users with different preferences and in different application scenarios. So we plan to adopt a personalized training model to the system. The model will be trained on-line with daily ground truth responses from the user to make it adaptive to personalized user preferences. We also plan to incorporate support for sensors in our system, for example, using Sensordrone [Sen] and other sensors such as accelerometers. And incorporating contextual co-presence detection into de-authentication would be another good try for our next step. This feature is requested by some users, to improve security by enforcing T to lock when T and D are in proximity but in different environment (e.g. T is left inside a room and D is taken outside but still being nearby).

Acknowledgments

This thesis is a summary of my work in the Contextual Co-presence Detection (CoCo) Project at the Secure Systems group at the University of Helsinki (Finland), and with the collaborators in University of Alabama at Birmingham (USA). I thank the group members that contributed to project management and data analysis: Hien Truong, N. Asokan, Babins Shrestha, Nitesh Saxena, and Petteri Nurmi. I also thank the user study participants for their time and feedback. My work was supported by TEKES as part of the Internet of Things program. The thesis work was supervised by Prof. N. Asokan and Dr. Hien Truong.

Publications

The thesis work contributed to the following publications:

1. Hien Thi Thu Truong, Xiang Gao, Babins Shrestha, Nitesh Saxena, N.Asokan and Petteri Nurmi. Comparing and Fusing Different Sensor Modalities for Relay Attack Resistance in Zero-Interaction Authentication. *in Proceedings of the 12th International Conference on Pervasive Computing and Communications (PerCom'14)*, Budapest, Hungary, March 2014.

2. Hien Thi Thu Truong, Xiang Gao, Babins Shrestha, Nitesh Saxena, N.Asokan and Petteri Nurmi. Using Contextual Co-Presence to Strengthen Zero-Interaction Authentication: Design, Integration and Usability, *Pervasive and Mobile Computing journal (PMC)*. (submitted)

References

AMQ Amqp, Advanced Message Queuing Protocol. URL http://www.amqp.org/.

BC94 Brands, S. and Chaum, D., Distance-bounding protocols. *in Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT'93)*, 1994, pages 344–359, URL http://dx.doi.org/10.1007/3-540-48285-7_30.

BKP03 Bardram, J., Kjær, R. and Pedersen, M., Context-aware user authentication-supporting proximity-based login in pervasive computing. *in Proceedings of the 5th International Conference on Ubiquitous Computing (UbiComp'03)*, 2003, pages 107–123, URL http://dx.doi.org/10.1007/978-3-540-39653-6_8.

Blu BlueProximity, SourceForge Project. URL http://sourceforge.net/projects/blueproximity/.

Bro96 Brooke, J., SUS: A quick and dirty usability scale. In *Usability evaluation in industry*, JJordan, P. W., Weerdmeester, B., Thomas, A. and Mclelland, I. L., editors, Taylor and Francis, London, 1996, pages 189–194, URL http://www.usabilitynet.org/trump/documents/Suschapt.doc.

CDK+12 Czeskis, A., Dietz, M., Kohno, T., Wallach, D. and Balfanz, D., Strengthening user authentication through opportunistic cryptographic identity assertions. *in Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12)*, New York, New York, USA, 2012, ACM Press, page 404, URL http://dx.doi.org/10.1145/2382196.2382240.

CKSK08 Czeskis, A., Koscher, K., Smith, J. R. and Kohno, T., RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications. *in Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08)*, New York, New York, USA, 2008, ACM Press, page 479, URL http://dx.doi.org/10.1145/1455770.1455831.

CN02 Corner, M. D. and Noble, B. D., Zero-interaction authentication. *in Proceedings of the 8th International Conference on Mobile Computing and Networking (MobiCom'02)*, New York, New York, USA, 2002, ACM Press, page 1, URL http://dx.doi.org/10.1145/570645.570647.

Cov06 Covell, M., Content fingerprinting using wavelets. *in Proceedings of the 3rd European Conference on Visual Media Production (CVMP'06), Part of the 2nd Multimedia Conference 2006*. IEE, 2006, pages 198–207, URL http://dx.doi.org/10.1049/cp:20061964.

CSR11 Chandrasekhar, V., Sharifi, M. and Ross, D., Survey and Evaluation of Audio Fingerprinting Schemes for Mobile Query-by-Example Applications. *in Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR'11)*, number Ismir, 2011, pages 801–806, URL http://ismir2011.ismir.net/papers/OS10-2.pdf.

Dar12 Dargie, W., Evaluation of the reliability of RSSI for indoor localization. *in Proceedings of the 3rd International Conference on Wireless Communications in Underground and Confined Areas (ICWCUCA'12)*. IEEE, August 2012, pages 1–6, URL http://dx.doi.org/10.1109/ICWCUCA.2012.6402492.

DEM12 Dousse, O., Eberle, J. and Mertens, M., Place Learning via Direct WiFi Fingerprint Clustering. *in Proceedings of the 13th IEEE International Conference on Mobile Data Management (MDM'12)*. IEEE, July 2012, pages 282–287, URL http://dx.doi.org/10.1109/MDM.2012.46.

DWF+11 Dey, A. K., Wac, K., Ferreira, D., Tassini, K., Hong, J.-H. and Ramos, J., Getting closer: An empirical investigation of the proximity of user to their smart phones. *in Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp'11)*, New York, NY, USA, 2011, ACM, pages 163–172, URL http://doi.acm.org/10.1145/2030112.2030135.

DY83 Dolev, D. and Yao, A., On the security of public key protocols. *IEEE Transactions on Information Theory*, 29,2(1983), pages 198–208. URL http://dx.doi.org/10.1109/TIT.1983.1056650.

EP05 Eagle, N. and Pentland, A., Social Serendipity: Mobilizing Social Software. *IEEE Pervasive Computing*, 4,2(2005), pages 28–34. URL http://dx.doi.org/10.1109/MPRV.2005.37.

FDC10 Francillon, A., Danev, B. and Capkun, S., Relay attacks on passive keyless entry and start systems in modern cars. *in Proceedings of the 18th Network and Distributed System Security Symposium (NDSS'11)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science, 2010, URL http://dx.doi.org/10.3929/ethz-a-006708714.

FFT Fast fourier transform, from Wikipedia. URL https://en.wikipedia.org/wiki/Fast_Fourier_Transform.

FHMM10 Francis, L., Hancke, G., Mayes, K. and Markantonakis, K., Practical NFC peer-to-peer relay attack using mobile phones. *in Proceedings of the 6th International Conference on Radio Frequency Identification: Security and Privacy Issues (RFIDSec'10)*, Berlin, Heidelberg, 2010, Springer-Verlag, pages 35–49, URL http://dx.doi.org/10.1007/978-3-642-16822-2_4.

GGH89 Gellert, W., Gottwald, S. and Hellwich, M., *The VNR concise encyclopedia of mathematics*. Van Nostrand Reinhold New York, second edition, 1989. URL http://dx.doi.org/10.1007/978-94-011-6982-0.

GMA11 Gupta, A., Miettinen, M. and Asokan, N., Using context-profiling to aid access control decisions in mobile devices. *in Proceedings of IEEE 2011 International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, March 2011, pages 310–312, URL http://dx.doi.org/10.1109/PERCOMW.2011.5766891.

GMAN12 Gupta, A., Miettinen, M., Asokan, N. and Nagy, M., Intuitive Security Policy Configuration in Mobile Devices Using Context Profiling. *in Proceedings of the 4th International Conference on Privacy, Security, Risk and Trust (PASSAT'12) and the 4th International Confernece on Social Computing (SocialCom'12)*. IEEE, September 2012, pages 471–480, URL http://dx.doi.org/10.1109/SocialCom-PASSAT.2012.60.

Gow12 Gowda, P. L., *Exploring bluetooth for received signal strength indicator-based secret key extraction*. Ph.D. thesis, University of

Utah, 2012. URL http://content.lib.utah.edu/utils/getfile/collection/etd3/id/2052/filename/2032.pdf.

HK02 Haitsma, J. and Kalker, T., A highly robust audio fingerprinting system. *in Proceedings of the 3rd International Society for Music Information Retrieval Conference (ISMIR'02)*, 2002, pages 107–115, URL http://ismir2002.ismir.net/proceedings/02-FP04-2.pdf.

HK03 Haitsma, J. and Kalker, T., A Highly Robust Audio Fingerprinting System With an Efficient Search Strategy. *Journal of New Music Research*, 32,2(2003), pages 211–221. URL http://dx.doi.org/10.1076/jnmr.32.2.211.16746.

HK05 Hancke, G. and Kuhn, M., An RFID Distance Bounding Protocol. *in Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*. IEEE, 2005, pages 67–73, URL http://dx.doi.org/10.1109/SECURECOMM.2005.56.

HMS01 Holmquist, L., Mattern, F. and Schiele, B., Smart-its friends: A technique for users to easily establish connections between smart artefacts. *in Proceedings of the 3rd International Conference on Ubiquitous Computing (UbiComp'01)*, 2001, pages 116–122, URL http://dx.doi.org/10.1007/3-540-45427-6_10.

HMSX12 Halevi, T., Ma, D., Saxena, N. and Xiang, T., Secure Proximity Detection for NFC Devices Based on Ambient Sensor Data. *in Proceedings of the 17th European Symposium on Research in Computer Security (ESORICS'12)*, volume 7459, 2012, pages 1–18, URL http://dx.doi.org/10.1007/978-3-642-33167-1_22.

Jue06 Juels, A., RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications*, 24,2(2006), pages 381–394. URL http://dx.doi.org/10.1109/JSAC.2005.861395.

KBG+10 Kjærgaard, M. B., Blunck, H., Godsk, T., Toftkjær, T., Christensen, D. L. and Grønbæk, K., Indoor Positioning Using GPS Revisited. *in Proceedings of the 8th International Conference on Pervasive Computing (Pervasive'10)*, Floréen, P., Krüger, A. and Spasojevic, M., editors, volume 6030, Berlin, Heidelberg, May 2010, Springer

Berlin Heidelberg, pages 38–56, URL http://dx.doi.org/10.1007/978-3-642-12654-3_3.

Key KeylessGo, Mercedes-Benz commercial product. URL http://techcenter.mercedes-benz.com/_en/keylessgo/detail.html.

KH04 Krumm, J. and Hinckley, K., The NearMe Wireless Proximity Server. *in Proceedings of the 6th International Conference on Ubiquitous Computing (UbiComp'04)*, 2004, pages 283–300, URL http://dx.doi.org/10.1007/978-3-540-30119-6_17.

KREA11 Kostiainen, K., Reshetova, E., Ekberg, J.-E. and Asokan, N., Old, new, borrowed, blue – a perspective on the evolution of mobile platform security architectures. *in Proceedings of the 1st ACM Conference on Data and Application Security and Privacy (CODASPY'11)*, New York, New York, USA, 2011, ACM Press, page 13, URL http://dx.doi.org/10.1145/1943513.1943517.

KW05 Kfir, Z. and Wool, A., Picking Virtual Pockets using Relay Attacks on Contactless Smartcard. *in Proceedings of the 1st International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05)*, number c. IEEE, 2005, pages 47–58, URL http://dx.doi.org/10.1109/SECURECOMM.2005.32.

LcA+04 Levi, A., Çetintas, E., Aydos, M., Koç, c. K. and Çaglayan, M. U., Relay Attacks on Bluetooth Authentication and Solutions. *in Proceedings of the 19th International Symposium of Computer and Information Sciences (ISCIS'04)*, 2004, pages 278–288, URL http://dx.doi.org/10.1007/978-3-540-30182-0_29.

LPL+09 Lu, H., Pan, W., Lane, N. D., Choudhury, T. and Campbell, A. T., SoundSense: scalable sound sensing for people-centric applications on mobile phones. *in Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (Mobisys'09)*, New York, New York, USA, 2009, ACM Press, page 165, URL http://dx.doi.org/10.1145/1555816.1555834.

Mat75 Matthews, B., Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)*

- *Protein Structure*, 405,2(1975), pages 442 – 451. URL http://dx.doi.org/10.1016/0005-2795(75)90109-9.

MG07 Mayrhofer, R. and Gellersen, H., Shake well before use: Authentication based on accelerometer data. *in Proceedings of the 5th International Conference on Pervasive computing (Pervasive'07)*, 2007, pages 144–161, URL http://dx.doi.org/10.1007/978-3-540-72037-9_9.

MHK+14 Miettinen, M., Heuser, S., Kronz, W., Sadeghi, A.-R. and Asokan, N., ConXsense - Context Sensing for Adaptive Usable Access Control. *in Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security (ASIACCS'14)*, June 2014, pages –, URL http://arxiv.org/abs/1308.2903.

MJ07 Murdoch, S. D. and J., S., Keep your enemies close: Distance bounding against smartcard relay attacks. *in Proceedings of the 16th USENIX Conference on Security Symposium (USENIX Security'07)*, 2007, pages 87–102, URL https://www.usenix.org/legacy/events/sec07/tech/drimer/drimer.pdf.

MMV+11 Mathur, S., Miller, R., Varshavsky, A., Trappe, W. and Mandayam, N., ProxiMate: proximity-based secure pairing using ambient wireless signals. *in Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services (MobiSys'11)*, New York, New York, USA, 2011, ACM Press, page 211, URL http://dx.doi.org/10.1145/1999995.2000016.

MPSX12 Ma, D., Prasad, A. K., Saxena, N. and Xiang, T., Location-aware and safer cards: enhancing RFID security and privacy via location sensing. *in Proceedings of the 5th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WISEC'12)*, New York, New York, USA, April 2012, ACM Press, page 51, URL http://dx.doi.org/10.1145/2185448.2185457.

NFC NFC, from Wikipedia. URL http://en.wikipedia.org/wiki/Near_field_communication.

NME Nmea, NMEA 0183 Standard. URL http://www.nmea.org/content/nmea_standards/nmea_0183_v_410.asp.

Nor Euclidean norm, from Wikipedia. URL https://en.wikipedia.org/wiki/Norm_%28mathematics%29.

NSHJ12a Nguyen, N., Sigg, S., Huynh, A. and Ji, Y., Pattern-Based Alignment of Audio Data for Ad Hoc Secure Device Pairing. *in Proceedings of the 16th International Symposium on Wearable Computers (ISWC'12)*. IEEE, June 2012, pages 88–91, URL http://dx.doi.org/10.1109/ISWC.2012.14.

NSHJ12b Nguyen, N., Sigg, S., Huynh, A. and Ji, Y., Using ambient audio in secure mobile phone communication. *in Proceedings of IEEE 2012 International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*. IEEE, March 2012, pages 431–434, URL http://dx.doi.org/10.1109/PerComW.2012.6197527.

NT11 Narayanan, A. and Thiagarajan, N., Location privacy via private proximity testing. *in Proceedings of the 18th Network and Distributed System Security Symposium (NDSS'11)*, 2011, URL http://www.isoc.org/isoc/conferences/ndss/11/pdf/1_3.pdf.

QND14 Quach, Q., Nguyen, N. and Dinh, T., Secure Authentication for Mobile Devices Based on Acoustic Background Fingerprint. *in Proceedings of the 5th International Conference on Knowledge and Systems Engineering (KSE'13)*, Huynh, V. N., Denoeux, T., Tran, D. H., Le, A. C. and Pham, S. B., editors, volume 244, Cham, 2014, Springer International Publishing, pages 375–387, URL http://dx.doi.org/10.1007/978-3-319-02741-8.

RC10 Rasmussen, K. B. and Capkun, S., Realization of RF Distance Bounding. *in Proceedings of the 19th USENIX Conference on Security Symposium (USENIX Security'10)*, 2010, pages 25–25, URL http://www.usenix.org/events/sec10/tech/full_papers/Rasmussen.pdf.

RFI RFID, from Wikipedia. URL http://en.wikipedia.org/wiki/Radio-frequency_identification.

RQSL12 Riva, O., Qin, C., Strauss, K. and Lymberopoulos, D., Progressive Authentication: Deciding When to Authenticate on Mobile Phones. *in Proceedings of the 21st USENIX Conference on Security Symposium (USENIX Security'12)*, 2012, pages

15–15, URL https://www.usenix.org/system/files/conference/usenixsecurity12/sec12-final154.pdf.

SBG99 Schmidt, A., Beigl, M. and Gellersen, H.-W., There is more to context than location. *Computers & Graphics*, 23,6(1999), pages 893–901. URL http://dx.doi.org/10.1016/S0097-8493(99)00120-X.

SDY12 Stephan, S., Dominik, S. and Yusheng, J., PINtext: A Framework for Secure Communication Based on Context. *in Proceedings of the 8th International ICST Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous'11)*, 2012, pages 314–325, URL http://dx.doi.org/10.1007/978-3-642-30973-1_31.

SEKA11 Saxena, N., Ekberg, J.-E., Kostiainen, K. and Asokan, N., Secure Device Pairing Based on a Visual Channel: Design and Usability Study. *IEEE Transactions on Information Forensics and Security*, 6,1(2011), pages 28–38. URL http://dx.doi.org/10.1109/TIFS.2010.2096217.

Sen Sensordrone. URL http://www.sensorcon.com/sensordrone/.

SIG SIG, B., Bluetooth Specification Adopted Documents. URL https://www.bluetooth.org/en-us/specification/adopted-specifications.

Sig11 Sigg, S., Context-based security. *in Proceedings of the 5th ACM International Workshop on Context-Awareness for Self-Managing Systems (CASEMANS'11)*, New York, New York, USA, September 2011, ACM Press, pages 17–23, URL http://dx.doi.org/10.1145/2036146.2036150.

SJNH12 Sigg, S., Ji, Y., Nguyen, N. and Huynh, A., AdhocPairing: Spontaneous audio based secure device pairing for Android mobile devices. *in Proceedings of the 4th International Workshop on Security and Privacy in Spontaneous Interaction and Mobile Phone Use (IWSSI/SPMU'12), in conjunction with Pervasive'12*, 2012, URL http://www.medien.ifi.lmu.de/iwssi2012/papers/iwssi-spmu2012-sigg.pdf.

SS13 Schurmann, D. and Sigg, S., Secure Communication Based on Ambient Audio. *IEEE Transactions on Mobile Computing*, 12,2(2013), pages 358–370. URL http://dx.doi.org/10.1109/TMC.2011.271.

SSTA14 Shrestha, B., Saxena, N., Truong, H. T. T. and Asokan, N., Drone to the Rescue: Relay-Resilient Authentication using Ambient Multi-Sensing. *in Proceedings of the 18th International Conference on Financial Cryptography and Data Security (FC'14)*, 2014, pages –, URL http://se-sy.org/projects/coco/FC.pdf.

SUVA11 Saxena, N., Uddin, M. B., Voris, J. and Asokan, N., Vibrate-to-unlock: Mobile phone assisted user authentication to multiple personal RFID tags. *in Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom'11)*. IEEE, March 2011, pages 181–188, URL http://dx.doi.org/10.1109/PERCOM.2011.5767583.

SW81 Smith, T. and Waterman, M., Identification of common molecular subsequences. *Journal of Molecular Biology*, 147,1(1981), pages 195–197. URL http://dx.doi.org/10.1016/0022-2836(81)90087-5.

TMOA12 Tiwari, M., Mohan, P., Osheroff, A. and Alkaff, H., Context-centric security. *in Proceedings of the 7th USENIX Conference on Hot Topics in Security (HotSec'12)*, 2012, pages 9–9, URL https://www.usenix.org/conference/hotsec12/workshop-program/presentation/tiwari.

Tog13 Tognazzini, B., The Apple iWatch | askTog, 2013. URL http://asktog.com/atc/apple-iwatch/.

TPRC11 Tippenhauer, N. O., Pöpper, C., Rasmussen, K. B. and Capkun, S., On the requirements for successful GPS spoofing attacks. *in Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'11)*, New York, New York, USA, October 2011, ACM Press, page 75, URL http://dx.doi.org/10.1145/2046707.2046719.

TRPv09 Tippenhauer, N. O., Rasmussen, K. B., Pöpper, C. and Čapkun, S., Attacks on public WLAN-based positioning systems. *in Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services (Mobisys'09)*, New York, New York, USA, 2009, ACM Press, page 29, URL http://dx.doi.org/10.1145/1555816.1555820.

VHKR10 Vallivaara, I., Haverinen, J., Kemppainen, A. and Roning, J., Simultaneous localization and mapping using ambient magnetic field. *in Pro-*

ceedings of IEEE 2010 International Conference on Multisensor Fusion and Integration (MFI'10). IEEE, September 2010, pages 14–19, URL http://dx.doi.org/10.1109/MFI.2010.5604465.

VR79 Van Rijsbergen, C. J., *Information Retrieval*. Butterworth-Heinemann, Newton, MA, USA, second edition, 1979. URL http://www.dcs.gla.ac.uk/Keith/Preface.html.

VS07 Varshavsky, A. and Scannell, A., Amigo: Proximity-based authentication of mobile devices. *in Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp'07)*, 2007, pages 253–270, URL http://dx.doi.org/10.1007/978-3-540-74853-3_15.

Wan06 Wang, A., The Shazam music recognition service. *Communications of the ACM*, 49,8(2006), page 44. URL http://dx.doi.org/10.1145/1145287.1145312.

Web00 Webb, G. I., MultiBoosting: A Technique for Combining Boosting and Wagging. *Machine Learning*, 40,2(2000), pages 159–196. URL http://dx.doi.org/10.1023/A:1007659514849.

XCO Cross-correlation, from Wikipedia. URL https://en.wikipedia.org/wiki/Cross-correlation.

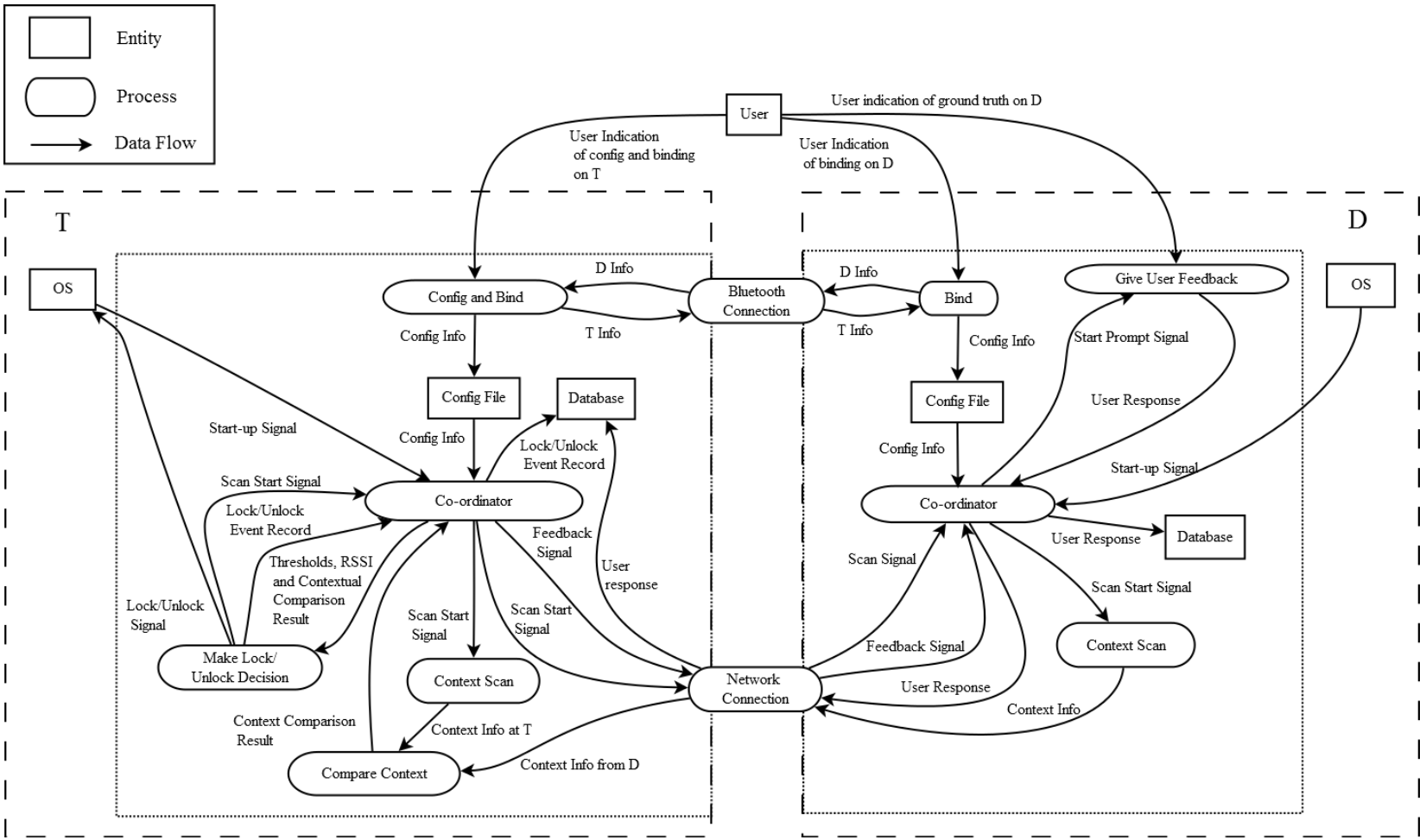
Yee04 Yee, K., Aligning security and usability. *IEEE Security & Privacy Magazine*, 2,5(2004), pages 48–55. URL http://dx.doi.org/10.1109/MSP.2004.64.

Appendices

A BlueProximity++: Data Flow Diagram

The data flow diagram provides an overview of the system-wide data flow in BlueProximity++. The regular rectangles represent the entities as data source or sink; the round-cornered rectangles represent the processes operating on or transporting data; and the arrows represent the directional data flows. More details are as follows:

- The data flows between *User* and *Config File*s are transient procedures to get user preferences as well as to bind T and D by exchanging meta information via a Bluetooth connection. As a result of the binding process, D and T share a key and agree on a "channel identifier" to uniquely identify that D-T channel.
- The routine data flow between T and D via the proxy server are handled by *Network Connection*. Such data flow is tagged with the channel identifier and encrypted using the previously agreed key.
- *Co-ordinator* at T triggers a new context scan by sending *Scan Start Signal* to *Context Scan* process at T and to the *Context Scan* process at D (via *Network Connection* and *Co-ordinator* at T). The *Compare Context* process gets the resulting *context info* from both T and D as input, and produces the comparison result for the *Co-ordinator* at T, which together with the information from *Config File* is used by *Make Lock/Unlock Decision* process for access control decisions.
- *Co-ordinator* at T records lock/unlock events in its local *Database*, and sends *Feedback Signal* to D for user notification and responses. Upon receiving *Feedback Signal*, *Co-ordinator* at T fires the *Start Prompt Signal*; the ground truth responses received from *User* are then routed back to T and stored in the *Database* at T.


Entity
Process
Data Flow
User
User indication of ground truth on D
User Indication of config and binding on T
User Indication of binding on D
T
D
OS
Config and Bind
D Info
Bluetooth Connection
T Info
Config Info
Config File
Database
Start-up Signal
Lock/Unlock Event Record
Scan Start Signal
Co-ordinator
Feedback Signal
User response
Thresholds, RSSI and Contextual Comparison Result
Lock/Unlock Signal
Make Lock/ Unlock Decision
Scan Start Signal
Context Scan
Context Comparison Result
Context Info at T
Compare Context
Context Info from D
Network Connection
Bind
Give User Feedback
Start Prompt Signal
User Response
Scan Signal
Feedback Signal
User Response
Context Info

B BlueProximity++: Smoothing Bluetooth RSSI

Using BlueProximity [Blu] to lock/unlock the terminal T, we observed that T locks screen sometimes even though device D was in proximity. Since locking or unlocking depends on Bluetooth RSSI, we conducted a small experiment to measure its reliability. The idea was to measure the temporal dynamics of Bluetooth RSSI at fixed distances. We used a Linux laptop (Thinkpad X230, Ubuntu 12.04) as T which is bound to a smartphone (Samsung I9195, Android 4.2.2) as D to run the application. T and D both support Bluetooth 3.0. We fixed a distance 1m between T and D to observe the fluctuation of Bluetooth RSSI of D that T can measure during three minutes.

Figure 23 shows the results of our experiment. Given a fixed time-window, for example within 1 minute, we observed some peaks. For instance, if we take -6dB as the locking threshold, from 60 to 120 second, RSSI levels are almost above -6dB except some peak points at -13dB and -9dB.

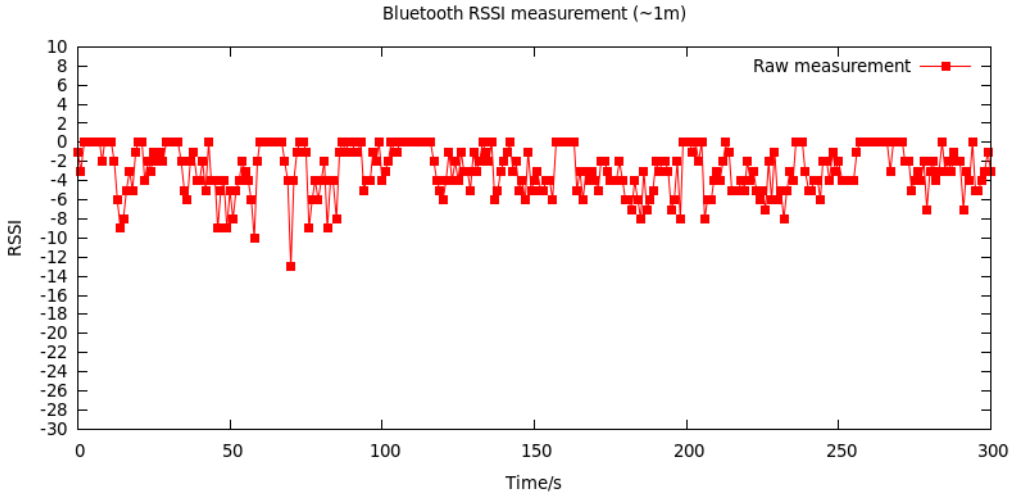


Figure 23: Bluetooth RSSI for 1m: raw measurement.

According to related work [Gow12, Dar12], Bluetooth signal is considered sensitive to many physical factors such as shadow-fading (e.g. with water or human body as obstacles), multi-path fading (e.g. in a long and narrow space), and antenna polarization. Instead of circumventing such physical factors, we attempted to mitigate the fluctuation by smoothing RSSI values of D that T can measure to reduce

fluctuation. Our algorithm for smoothing RSSI is described as follows. T runs the algorithm when a new RSSI (denoted by S) is measured.

(1) Eliminating a suspicious outlier: If the difference between S and the previous RSSI S_{prev} goes beyond an empirical threshold δ, T consider S suspicious and uses S_{prev} as the current RSSI. Otherwise, T assigns S to S_{prev}.

(2) Calculating the average: T calculates the mean of a buffer B (with size of W, including S_{prev} and the previous $(W-1)$ RSSIs). The result is denoted by S_{mean}.

(3) Selecting from the candidates: T selects the maximum between S_{prev} and S_{mean} as the smoothing result at this moment.

Algorithm 3 Smoothing a new Bluetooth RSSI of D measured by T

```
Require: global W, δ, B, S_prev, S_sus
  procedure SmoothRSSI(S)
    if B is empty then
      S_prev ← S, S_sus ← 0
    else if S − S_prev ≤ δ then
      if S_sus = 0 then
        S_sus ← S
      else
        S_prev ← S, S_sus ← 0
    else
      S_prev ← S, S_sus ← 0
    if B is full then
      remove the oldest element from B
    append S_prev to the end of B
    S_mean ← mean of B
    S_ret ← max(S_prev, S_mean)
    return S_ret
```

Algorithm 3 explains the details of smoothing the Bluetooth RSSIs of D measured by T. B is the buffer array of previous RSSIs with size of W. δ is the threshold for verifying outliers. S is the current raw RSSI of D measured by T, S_{prev} is the previous RSSI after eliminating outliers, and S_{sus} is the candidate of RSSI outlier. S_{mean} is the resulting average of buffer. S_{ret} is the final result of smoothing the current RSSI.

Figure 24 shows our smoothing results (the green curve) with the empirical parameters $W = 5, \delta = -6$. Intuitively, the green curve is more smooth that the red curve for raw RSSI values. For every 5-second window, the peaks are closer to the mean of RSSI values. Our locking/unlocking based on RSSI benefits from this improvement.

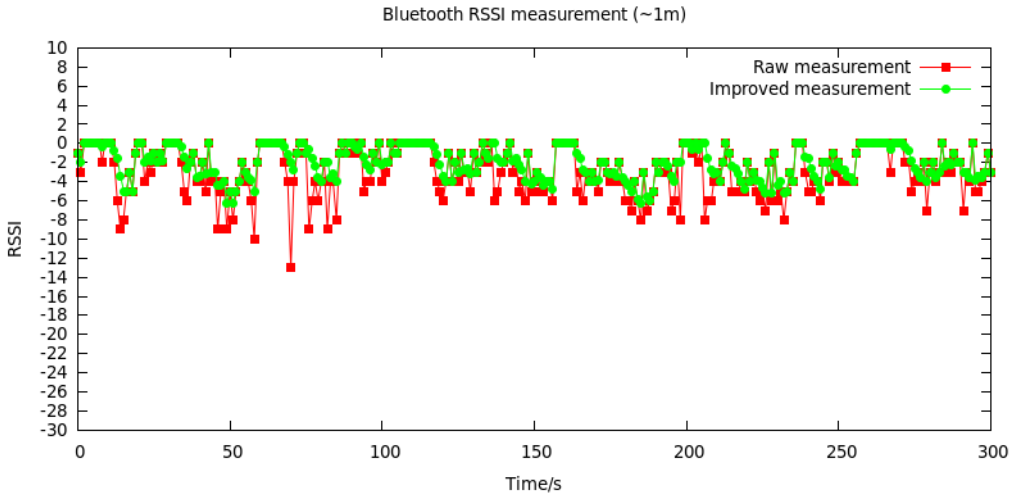


Figure 24: Bluetooth RSSI for 1m: improved measurement.

C User Study Questionnaires

1. Consent form

User Study Informed Consent Form

You have been selected to participate in a demonstration of a screen lock application. Through this demonstration, we hope to collect user responses about positive and negative screen lock events in real life.

Should you choose to participate, you will be asked to complete the tasks that we will or have described to you to the best of your abilities. This will include taking the Android device with you, giving responses once you are notified, and unlocking screen via your Android device instead of typing password.

Your identity will be completely confidential. Only cursory information about your identity (such as gender, age group, etc.) will be used. Your name will not be revealed without your consent.

Your data (nearby WiFi, Bluetooth devices, and short bursts of ambient audio) that the app collected will be anonymized and used for our research. Data is available for inspection upon request.

Copies of this form are available upon request.

Signing below indicates that the participant has read, understands, and agrees to the terms stated above.

Signature of Participant _____ Date _____

2. Demography questionnaire

Pre-Test Questionnaire :

1. Age : _____
2. Gender : ☐ M ☐ F
3. Nationality: _____
4. What academic program are you enrolled in? _____
5. At which of the following level are you studying at present?
   [ ] Undergraduate
   [ ] Graduate
   [ ] Ph.D
6. On a scale of 1 to 10, how would you rate yourself with respect to your computer skills, 1 being a novice and 10 an expert?

1	2	3	4	5	6	7	8	9	10

7. How often do you use mobile devices (such as mobile phones)?
   [ ] Daily
   [ ] Several times a week
   [ ] Once a week
   [ ] Less than once a week
8. How often do you use a laptop or desktop computer?
   [ ] Daily
   [ ] Several times a week
   [ ] Once a week
   [ ] Less than once a week
9. Do you lock your laptop or desktop with a password:
   Yes ☐ No ☐
10. How concerned you are about the security of your laptop or desktop?
    [ ] Very much
    [ ] Somewhat
    [ ] A little
    [ ] Not at all

3. Post-test SUS questionnaires

Post-Test Questionnaire – System I (the first unlocking mechanism used)

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Other Comments:

_____

_____

_____

_____

Post-Test Questionnaire – System II (the second unlocking mechanism used)

	Strongly disagree				Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5
2. I found the system unnecessarily complex	1	2	3	4	5
3. I thought the system was easy to use	1	2	3	4	5
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5
5. I found the various functions in this system were well integrated	1	2	3	4	5
6. I thought there was too much inconsistency in this system	1	2	3	4	5
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5
8. I found the system very cumbersome to use	1	2	3	4	5
9. I felt very confident using the system	1	2	3	4	5
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5

Other Comments:

_____

_____

_____

_____

4. Comparison questionnaire

Post-Test Questionnaire – Comparison

Which system did you prefer?

☐ System I

☐ System II

Other Comments:

_____

_____

_____

_____

_____

_____

_____

_____

_____

5. Open-ended questionnaire

Please provide as detailed responses as you can. Thanks!

1. Do you feel that the app meets your security needs?

2. Did you notice any difference in battery consumption when you were using the app compared to earlier?

3. What did you like most about the app?

4. What did you like the least?

5. When we have a new version of the app, would you want to try it out?

6. Do you think the app are better than using passwords?

7. Were there situations where you wanted to unlock your desktop/laptop, but your phone was not near you or out of power?