Pervasive and Mobile Computing 16 (2015) 187–204

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

Using contextual co-presence to strengthen Zero-Interaction Authentication: Design, integration and usability

Hien Thi Thu Truong [a,*], Xiang Gao [a], Babins Shrestha [b], Nitesh Saxena [b], N. Asokan [a,c], Petteri Nurmi [a]

[a] University of Helsinki, Finland
[b] University of Alabama at Birmingham, USA
[c] Aalto University, Finland

ARTICLE INFO

Article history:
Available online 29 October 2014

Keywords:
Relay attack
Multiple sensors
Zero interaction authentication

ABSTRACT

Zero-Interaction Authentication (ZIA) refers to approaches that authenticate a user to a verifier (terminal) without any user interaction. Currently deployed ZIA solutions are predominantly based on the terminal detecting the proximity of the user's personal device, or a security token, by running an authentication protocol over a short-range wireless communication channel. Unfortunately, this simple approach is highly vulnerable to low-cost and practical *relay attacks* which completely offset the usability benefits of ZIA. The use of contextual information, gathered via on-board sensors, to detect the co-presence of the user and the verifier is a recently proposed mechanism to resist relay attacks.

In this paper, we systematically investigate the performance of different sensor modalities for co-presence detection with respect to a standard Dolev–Yao adversary. *First*, using a common data collection framework run in realistic everyday settings, we compare the performance of four commonly available sensor modalities (WiFi, Bluetooth, GPS, and audio) in resisting ZIA relay attacks, and find that WiFi is better than the rest. *Second*, we show that, compared to any single modality, fusing *multiple modalities* improves resilience against ZIA relay attacks while retaining a high level of usability. *Third*, we motivate the need for a stronger adversarial model to characterize an attacker who can compromise the integrity of context sensing itself. We show that in the presence of such a powerful attacker, each individual sensor modality offers very low security. Positively, the use of multiple sensor modalities improves security against such an attacker *if* the attacker cannot compromise multiple modalities simultaneously.

Finally, based on our analysis, we integrate our contextual co-presence detection system with a real-world ZIA application, BlueProximity [1], so as to enhance its security against relay attacks. We describe the design of the *BlueProximity++* application and present results from a small-scale user study that evaluated its effectiveness.

1. Introduction

In proximity-based "zero interaction authentication" (ZIA) [1] systems, a verifier device authenticates the presence of a prover device in physical proximity of the verifier while requiring *no additional interaction* by the user of the prover device.

* Corresponding author.
E-mail addresses: htruong@cs.helsinki.fi (H.T.T. Truong), xzgao@cs.helsinki.fi (X. Gao), babins@uab.edu (B. Shrestha), saxena@uab.edu (N. Saxena), asokan@acm.org (N. Asokan), ptnurmi@cs.helsinki.fi (P. Nurmi).

http://dx.doi.org/10.1016/j.pmcj.2014.10.005
1574-1192/

The zero interaction requirement is intended to improve usability of access control systems. For example, BlueProximity [2] allows a user to unlock the idle screen lock in her computer merely by physically approaching the computer while in possession of a mobile phone, previously paired with the computer, without having to perform any other action, such as typing in a password. Motivated by these usability considerations, there are many examples of ZIA systems, such as "Passive keyless entry and start" systems like "Keyless-Go"[1] PhoneAuth [3], and access control systems based on wearable devices [4].

Although the security research community no longer takes security and usability to be mutually contradictory goals [5], simultaneously accomplishing security and usability goals continues to be a challenge. Under the standard Dolev–Yao adversary model [6], an attacker is assumed to have complete control over the communication channel. In such a model, naïve ZIA schemes are vulnerable to *relay attacks* where a pair of colluding attackers relays messages between a legitimate user and verifier, thereby fooling the verifier into incorrectly concluding that the user is in close proximity. Relay attacks have been demonstrated to be practical for various short range wireless communication technologies like Bluetooth [7,8], RFID [9] and NFC [10], making this vulnerability a serious threat.

The commonly proposed defense against such relay attacks, while preserving zero-interaction, is to use *distance bounding* techniques [11]. Distance bounding assumes that the prover and verifier share a security association. The prover is required to respond to a series of rapid-fire challenges from the verifier, which can then calculate a lower bound for the distance to the prover by measuring the elapsed time between sending a challenge and receiving a correct response. Distance bounding needs to be implemented at the lowest possible layer in the communication stack because even a small error in estimating processing time at the prover side can lead to large deviations in the distance bound. Therefore implementing distance bounding on commodity devices like ordinary smartphones might be a challenge.

An alternative approach is to leverage the fact that two co-present devices will "see" (almost) the same ambient environment. Modern computing devices are equipped with many "sensors" like microphones, wireless networking interfaces, global positioning system (GPS) receivers and so on. A device can extract information from such a sensor that is characteristic of that context. By having two mutually trusting devices exchange and compare context information, they can determine if they are co-present or not. This approach has recently been proposed for *single* sensor modalities, including WiFi [12,13], audio [14,15] Bluetooth and GPS [16].

Although these prior works constitute an important step towards addressing the hard problem of resisting relay attacks using off-the-shelf hardware, they leave several important questions unexplored, which we address in this paper. *First*, we compare the performance of different sensor modalities in resisting relay attacks against ZIA based on contextual co-presence. Although standalone evaluations of different modalities individually have been reported in prior work, they cannot be used for a fair comparison given that the data assessing each modality was collected in disparate settings. *Second*, we investigate whether the combination ("fusion") of multiple sensor modalities will perform better than using individual modalities in isolation. Prior work did not address this question. *Third*, we explore the question of finding the appropriate adversary model for ZIA based on contextual co-presence. While the Dolev–Yao model is sufficient for relay attacks in general, the use of contextual co-presence raises the possibility of an attacker who can subvert the integrity of context sensing (by faking signals in the context). Previous works have mainly considered resistance against false authentications in benign settings [14] or with respect to specific attack scenarios [13].

Based on our analysis, we designed and implemented *BlueProximity++* which incorporates contextual co-presence determination into the existing BlueProximity [2] ZIA application, commonly used for unlocking a PC or laptop based on proximity with a mobile phone. We conducted a small-scale user study which indicates that the inclusion of contextual co-presence provides comparable usability, while at the same time significantly improving security of ZIA.

Contributions: This paper makes the following contributions:

1. We present the first "fair" comparison of four sensor modalities commonly available on commodity smartphones – audio, WiFi, Bluetooth and GPS – under the *same settings*. We show that, in the usage contexts captured in our data, the use of WiFi for contextual co-presence outperforms the other modalities in resisting relay attacks (Section 5.3). Our analysis is based on a dataset collected from multiple users and devices, in a combination of predefined scenarios and everyday situations, using a common data collection framework we developed (Section 3). We make the dataset and framework freely available for research purposes.[2]
2. We demonstrate that fusing multiple modalities is effective: it can improve security, while maintaining a very similar level of usability as proximity-based ZIA mechanisms (Section 5.3).
3. Using a simple model for adversaries who can compromise the integrity of context sensing, we show that individual modalities provide low security against such adversaries. Fusion can improve security *if* the adversary cannot compromise multiple sensor modalities simultaneously. Our results call for extensions of the Dolev–Yao model that incorporate integrity of context sensing as part of the model (Section 6).
4. We present BlueProximity++ which, to our knowledge, is the first ZIA app to incorporate multi-modal contextual co-presence for access control.

1 http://techcenter.mercedes-benz.com/_en/keylessgo/detail.html.

2 http://se-sy.org/projects/coco.

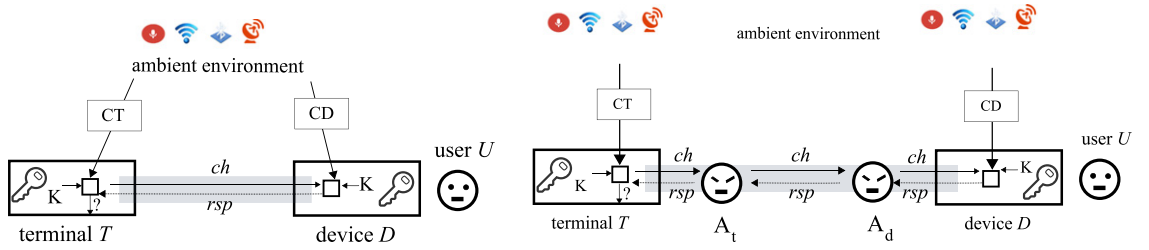


Fig. 1. System model for ZIA with contextual co-presence.

2. Background

ZIA: A ZIA scheme involves a user U who intends to authenticate to a verifier terminal T (e.g., a PC, car or gate) using a device D (e.g., a phone or smart key). U does not explicitly take part in the authentication process other than by approaching T while carrying D. ZIA is triggered by the devices sensing each other over a short-range wireless communication channel like Bluetooth. T will authenticate U by running a standard challenge–response based entity authentication protocol with D over the proximity communication channel. D and T pre-share a key K, which allows D to authenticate to T in the entity authentication protocol.

Standard adversary model: The goal of the adversary A against the ZIA protocol is to fool T into concluding that U is nearby and thus needs access to T even when U is far away (and not intending to authenticate). A possesses standard Dolev–Yao capabilities [6]: it has complete control of the communication channel over which the authentication protocol between T and D is run but does not have physical possession of D nor is able to compromise either D or T. However, we allow A to be in close physical proximity of D as well as T, even when D and T are far apart and U does not intend for D to authenticate to T. A could take the form of a "ghost-and-leech" [8] duo (A_d, A_t) such that A_d is physically close to D and A_t is physically close to T, and A_d and A_t communicate over a high-bandwidth connection. Such an adversary pair can completely compromise the security of ordinary ZIA schemes by simply initiating a protocol session between D and T, relaying messages (e.g., the challenge and response) between them, leading T to conclude that D is in proximity.[3]

ZIA enhanced with contextual co-presence: The contextual co-presence approach to ZIA aims to prevent such a relay attack. Fig. 1 shows the system model for ZIA based on contextual co-presence. When D sends a ZIA trigger to T it responds with a challenge ch. D and T then initiate context sensing for a fixed duration t. D appends ch to the sensed context information CD and computes an authenticated encryption of the result using key K to create the response rsp, which is sent to T. In the meantime, T finishes sensing its own context CT and compares it with CD extracted from rsp. T can conclude that D is in proximity if CT and CD are sufficiently similar. Note that context sensing is not run continually, but only when an authentication request takes place, implying a minimal energy overhead from the inclusion of sensing. When multiple (n) sensor modalities are used, CD and CT are vectors of the form $CD = CD_1, CD_2, \ldots, CD_n, CT = CT_1, CT_2, \ldots, CT_n$. T compares each CT_i with received CD_i in making the co-presence decision. In such a ZIA scheme enhanced with the contextual defense, A still cannot manipulate the authentication protocol between D and T. However, as we will later explore in Section 6, A may undermine the integrity of context sensing.

3. Data collector

Evaluating the suitability of contextual co-presence for ZIA requires collecting ground truth data about co-presence together with multimodal sensor data. To gather this kind of data, we have developed a data collection framework as an application installed on user devices. Our goal in developing this application is to have an easy-to-use, non-intrusive tool that allows potentially a large set of users to collect co-presence ground truth data. We also wanted a tool that can be easily re-purposed to conduct real-world and controlled experiments. Existing context sensing frameworks such as SensorDrone[4] are not suitable because they are designed for data collection on individual devices, making it cumbersome to collect co-presence data from multiple devices. Concretely we aimed for 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.

[3] We have developed a proof-of-concept to demonstrate the feasibility of such a relay attack against BlueProximity (a practical ZIA instantiation) using off-the-shelf hardware/software.

[4] http://www.sensorcon.com/sensordrone/.

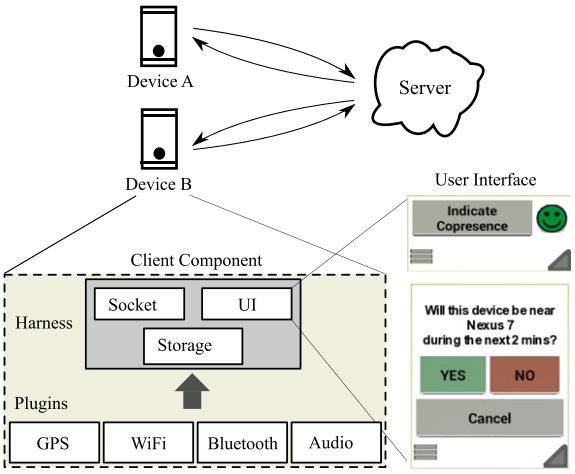


Fig. 2. Data collector architecture.

3.1. Design and usage

Fig. 2 depicts the architecture of the data collector. It consists of the back-end synchronization server, the pair of clients, and the communication between server and clients.

Server facilitates the "binding" of two devices of the same user. It provides a communication channel between a pair of bound devices for forwarding control messages to synchronize data collection. It also stores the collected data samples.

Client records and uploads sensor data, and provides the UI via which the user indicates co-presence ground truth. The client software framework consists of a harness with common functionality (communication with Server, UI, etc.) and a plugin interface for integrating sensing modules for different sensor modalities. Clients use the communication channel via Server to synchronize sensing.

Usage: A data collection user begins by binding two devices with the help of Server. Once the devices are bound, they maintain an open connection to Server. A user can provide co-presence ground truth via the UI shown in Fig. 2.[5] It can be launched either manually (at any time) by the user activating the "Indicate co-presence" button or periodically (once every 30 min by default; the frequency is configurable by the user). The UI is launchable only if the peer device (and hence Server) are reachable. When the user indicates ground truth in one device, sensing is initiated on both devices. The resulting sensor data (collection of samples) and the ground truth are sent to Server for persistent storage with a unique sequence number. The data is then deleted on devices. The data collector app minimizes energy consumption by initiating context scans only when ground truth is provided by the user.

3.2. Sensor data

We currently have plugins for GPS, WiFi, Bluetooth and audio modalities. These modalities were chosen as they are widely available on contemporary smartphones.

GPS raw data: We record the identifiers of visible GPS satellites and the "signal strength" for each of them in the form of signal-to-noise ratio (SNR). The identifier is the "pseudo-random noise code" (PRN) which is an integer ($1 \ldots 32$). Each data sample consists of multiple records taken at the rate of one every second over a two minute period. Each record contains the set of identifiers and SNRs observed at that instant. The SNR ranges from 0 to 100. Where a location fix is available, we record longitude, latitude, altitude and accuracy.

WiFi: For each visible WiFi access point (AP), we record the list of link-layer addresses (BSSID) and the associated received signal strength indicators (RSSI), supported capabilities and the frequency of the WiFi channel advertised by that AP. RSSI ranges from -100 to -20 dBm.

Bluetooth: For each visible Bluetooth device, we record the identifier (BDADDR) and received signal strength indicator (RSSI). RSSI ranges from -100 to -20 dBm.

Audio: Ambient audio is recorded in standard PCM format (wav file) without compression. Each PCM wave is sampled in 44 100 Hz with 16-bit encoding. Because raw audio is sensitive, by default, we do not store raw audio on Server. Instead, we extract certain features (as described in Section 4). Users however have the option of changing this default to let their client(s) upload raw audio to Server.

[5] Using Bluetooth signal strength to determine co-presence (as done by BlueProximity) is problematic because it has a high false negative rate. Instead, we let users decide what scenario constitutes as a co-present scenario.

4. Experimental setup

We carried out an extensive empirical investigation of the effectiveness of different sensor modalities, both individually and in different combinations, in strengthening ZIA solutions against relay attacks. In this section, we describe the data we collected, and the features used in our analyses.

4.1. Data collection

Everyday dataset:
Using our data collection framework, five testers collected data for 15 days in mid 2013. All five were university employees in the 25–35 age group (four male, one female). They were not asked for any information about their home environments. No pre/post questionnaire was required. Hardware variations across devices are well-known to cause significant changes in sensor measurements. To ensure robustness of results with respect to device variations, we collected data using tablets and phones from different manufacturers and with different models: tablets (Google Nexus 7, Samsung Galaxy Tab, Acer Iconia, Asus Transformer) and phones (Samsung Galaxy SII, SIII).

We gave no specific instructions to the testers about what scenarios or locations in which they should collect data. Consequently, the resulting dataset is *uncontrolled*, consisting of data collected in various everyday settings and locations (e.g., university campus, labs, libraries, cafeteria, home, streets). Data collection was done in two different cities: Birmingham, Alabama, USA and Helsinki, Finland. This dataset contains 2303 samples, of which 1140 samples (49.5%) are from co-present devices and 1163 (50.5%) from non co-present devices. Each sample contains data from sensor modalities available at the time on the respective devices (2117 with audio, 1600 with Bluetooth, 782 with GPS and 2269 with WiFi). For each sample, we scan all available sensors simultaneously: 2 min for GPS scanning, 10 scans for WiFi (about 30 s), 10 s for recording ambient audio, and 10 scans for Bluetooth (up to 12 s for each scan).

Ethical considerations: The data collection was carried in conformance to local IRB guidelines by members of the research groups who gave explicit consent. The data is released for research purposes in anonymized form. The anonymization was carried out by (a) replacing each device identifier with its SHA-1 value, (b) replacing the pair of GPS co-ordinates from the two devices in a sample by the great-circle distance between the pair, and (c) raw audio data is replaced by relevant features as discussed below.

4.2. Features

We investigated various possible features that can be extracted from the data in different sensor modalities, finally settling on the most promising features as discussed below.

4.2.1. Features for Bluetooth, WiFi, GPS

Although the three sensors (GPS, Bluetooth, WiFi) involving radio-frequency (RF) emissions considered in our analysis are different, fundamentally they have the same inherent characteristics. We therefore chose to represent them by a common set of features. Let a record from an RF sensor modality be of the form (m, s) where m is an identifier of a sensed device and s is the associated signal strength. Let S_a and S_b denote the set of records sensed by a pair of bound devices A and B, and let n_a and n_b denote the number of different beacons (i.e., WiFi access points, GPS satellites or Bluetooth devices) observed by devices a and b. We define the following sets:

$$S_a = \{(m_i^{(a)}, s_i^{(a)}) \mid i \in \mathbb{Z}_{n_a-1}\}.$$
$$S_b = \{(m_i^{(b)}, s_i^{(b)}) \mid i \in \mathbb{Z}_{n_b-1}\}.$$
$$S_a^{(m)} = \{m \,\forall (m, s) \in S_a\}, \qquad S_b^{(m)} = \{m \,\forall (m, s) \in S_b\}.$$
$$S_\cap = \{(m, s^{(a)}, s^{(b)}) \,\forall m | (m, s^{(a)}) \in S_a, (m, s^{(b)}) \in S_b\}.$$
$$S_\cup = S_\cap \cup \{(m, s^{(a)}, \theta) \,\forall m | (m, s^{(a)}) \in S_a, m \notin S_b^{(m)}\} \cup \{(m, \theta, s^{(b)}) \,\forall m | (m, s^{(b)}) \in S_b, m \notin S_a^{(m)}\},$$
θ is modality-specific (see below).
$$S_\cap^{(m)} = \{m \,\forall m | (m, s^{(a)}, s^{(b)}) \in S_\cap\}.$$
$$S_\cup^{(m)} = \{m \,\forall m | (m, s^{(a)}, s^{(b)}) \in S_\cup\}.$$
$$L_a^{(s)} = \{s^a | (m, s^{(a)}, s^{(b)}) \in S_\cap\}.$$
$$L_b^{(s)} = \{s^b | (m, s^{(a)}, s^{(b)}) \in S_\cap\}.$$

$S_\cap$ consists of devices seen by both A and B; $S_\cup$ represents all devices seen by A or B with θ filled in as the "signal strength" for devices that are *not* seen by either device.

We consider a total of six features, five of which have been selected from state-of-the-art co-presence detection systems (NearMe [12], Amigo [13] and RF-based place learning schemes [17]).

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: $\sum_{i=1}^{T} f_i$. Here T is the scanning time (seconds)

$$f_i = 1 \quad \text{if} \quad S_{a_i}^{(m)} \neq \emptyset,\ S_{b_i}^{(m)} \neq \emptyset, \\ (S_{a_i}^{(m)} \subseteq S_{b_i}^{(m)} \text{ or } S_{a_i}^{(m)} \supseteq S_{b_i}^{(m)})$$

$f_i = 0$ otherwise. S_{a_i}, S_{b_i} are the set of records by A and B respectively at the ith second.

WiFi: Features 1–5 are used. Since we do multiple scans in each sample, in line with current best practices, we use the mean value of RSSI for a BSSID (m) from all of the scans as the signal strength (s) value. θ is -100.

Bluetooth: Features 1, 3 are used with BDADDR as identifier (m) and average RSSI as signal strength (s). θ is -100.

GPS: All features are used with PRN as identifier (m) and mean SNR as signal strength (s). θ is 0.

Note that feature 6 is used only for GPS. This is because the set of satellites visible to a device varies greatly depending on the sensitivity of GPS hardware. Thus, one device may see a subset of the satellites seen by the another co-present device. In such cases, metrics like Jaccard distance perform poorly whereas the subset count could perform better. When GPS co-ordinates are available for A and B in a sample, we also use the orthodromic distance [18] as a feature.

4.2.2. Features for audio

We consider two features proposed by Halevi et al. [14], which were found to be the most robust among all algorithms tested: Schurmann and Sigg [15], SoundSense [19], Shazam audio fingerprinting [20], and Sound of Silence [21]. 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 two features that we consider are defined as follows:

- Max cross correlation:

$$M_{\text{corr}}(a, b) = \text{Max}(\textit{cross correlation}(X_a, X_b)).$$

- Time frequency distance:

$$D(a, b) = \sqrt{(D_{c,\text{time}}(a, b))^2 + (D_{d,\text{freq}}(a, b))^2}$$

where, $D_{c,\text{time}}(a, b) = 1 - M_{\text{corr}}$, $D_{d,\text{freq}}(a, b) = \|FFT(X_a) - FFT(X_b)\|$ is the Euclidean norm of the distance.

Here X_a and X_b denote the raw (16-bit PCM) audio signals recorded by A and B and FFT(X_a), FFT(X_a) denotes the Fast Fourier Transforms of the corresponding signals.

5. Analysis and results

5.1. Analysis methodology and metrics

We treat contextual co-presence detection as a classification task. All our experiments have been performed using ten-fold cross-validation and Multiboost [22], 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 2×2 confusion matrix, containing the number of True Positives (TP), True Negatives (TN), False Positives (FP) and False Negatives (FN). Positive and negative classes represent co-presence and non-co-presence, respectively.

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 and the Matthews' correlation coefficient (MCC).

The F-measure (Fm) uses precision ($\frac{\text{TP}}{\text{TP+FP}}$) and recall ($\frac{\text{TP}}{\text{TP+FN}}$) 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 ith class and N being the total number of samples, c is the number of classes.

Table 1
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

Table 2
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

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 2×2 contingency table: $|\text{MCC}| = \sqrt{\frac{\chi^2}{n}}$. It can be calculated directly from the confusion matrix as: $|\text{MCC}| = \frac{\text{TP}*\text{TN}-\text{FP}*\text{FN}}{\sqrt{(\text{TP}+\text{FP})*(\text{TP}+\text{FN})*(\text{TN}+\text{FP})*(\text{TN}+\text{FN})}}$. 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.

5.2. 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 s) 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 s, we only consider the sensor data recorded by the device in a sample within the first n s. Table 1 shows the results for the uncontrolled dataset for different time budgets. Although the overall performance is reasonable with a 5-s 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-s budget the performance is significantly better than with a 5-s budget as more data is captured by sensors, but it flattens out thereafter. Consequently, we fix a 10-s time budget for all subsequent analyses.

5.3. Performance of single and 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 multiple context modalities. The results of this investigation are shown in Table 2. For a given sensor modality, we only consider samples that have data from that sensor. To facilitate comparison,

Table 3
Single modalities comparison based on a subset of samples containing all sensor modalities.

Sensor modalities	Audio	Bluetooth	GPS	WiFi
%FN	28.0	28.0	21.7	1.2
%FP	10.9	4.0	123.0	2.3
MCC	0.623	0.705	0.552	0.964
Fm	0.807	0.842	0.776	0.982

Table 4
Performance for different modality bands, #IDs = number of device IDs seen and N: number of samples.

Audio						Bluetooth					
RMS[a]	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
Rest	1198	23.89	16.51	0.594	0.795	Rest	1049	10.34	0.72	0.883	0.941
GPS						WiFi					
#IDs	N	%FN	%FP	MCC	Fm	#IDs	N	%FN	%FP	MCC	Fm
≤5	243	29.8	19.1	0.514	0.755	≤5	212	20.0	34.6	0.460	0.730
Rest	539	20.0	25.0	0.549	0.774	Rest	2091	0.4	1.4	0.981	0.990

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

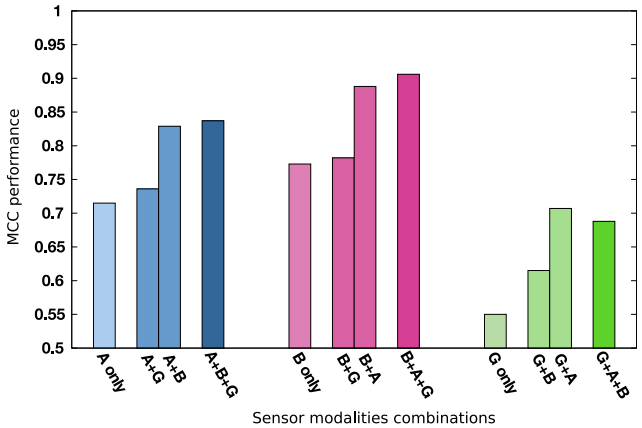


Fig. 3. MCC comparison for three modalities Audio (A)—Bluetooth (B)—GPS (G) and their combinations.

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 performs 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). We repeated the analysis using only those samples that have all modalities represented. As can be seen in Table 3, the relative performance of the modalities remains unchanged.

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 Fig. 3 we can observe that the performance of these modalities improves when they are fused with another modality.

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"). Table 4 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, or, alternatively the ZIA scheme can be adopted according to the characteristics of the current context data. Second, the performance of RF-based schemes, particularly GPS and Bluetooth, naturally improves when more radio sources (GPS satellites, Bluetooth beacons, or WiFi APs) are visible—but within our 10s time budget, GPS performs poorly because the vast majority of the samples contain only one visible satellite. Similarly, we can observe that most of the samples only have few Bluetooth devices visible, whereas WiFi AP points are much more prevalent.

Table 5
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 6
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

Table 7
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

5.4. Small-scale controlled dataset

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) 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 5 shows the performance of the classification in controlled dataset for different sensor modalities (single, and all together). As seen from Table 6, 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.

The performance of WiFi exceeds other modalities, providing near perfect results for the uncontrolled dataset. As we have demonstrated in Table 4 above, the performance of WiFi (naturally) depends on the amount of access points that can be observed, suggesting that generally there are sufficiently many APs available in everyday usage contexts. Another possible explanation 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 m 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 7 summarizes the results.

5.5. Effect of personalized training model

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 against hardware variations. Table 8 summarizes the results for three users (uncontrolled dataset) 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.

Table 8
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

5.6. 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.

6. Adversarial analysis

So far, we assumed the Dolev–Yao [6] 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. [23] 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 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 [14]. The occasional exceptions involved testing resistance against a few types of attacks interfering with context sensing [13]. 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.

A model for a context adversary: Faking 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).

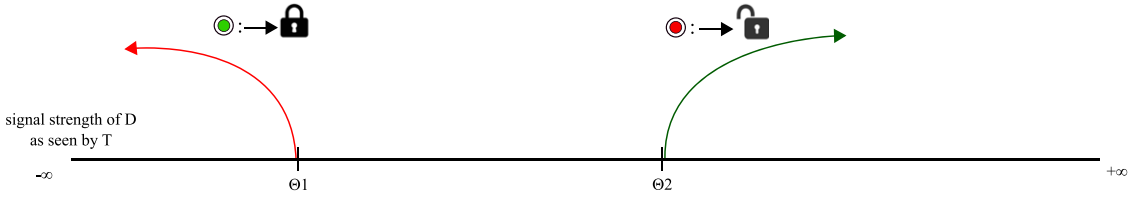
While this is still a very powerful attacker, analyzing our features for classification with respect to such an attacker may give some insights into the relative security of different sensor modalities.

Analysis: For RF-based sensors, the context adversary as defined above can be modeled 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 modeled replacing X_b by $X_a + X_b$. To estimate the effect of such an adversary, we took the following approach. We used 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.

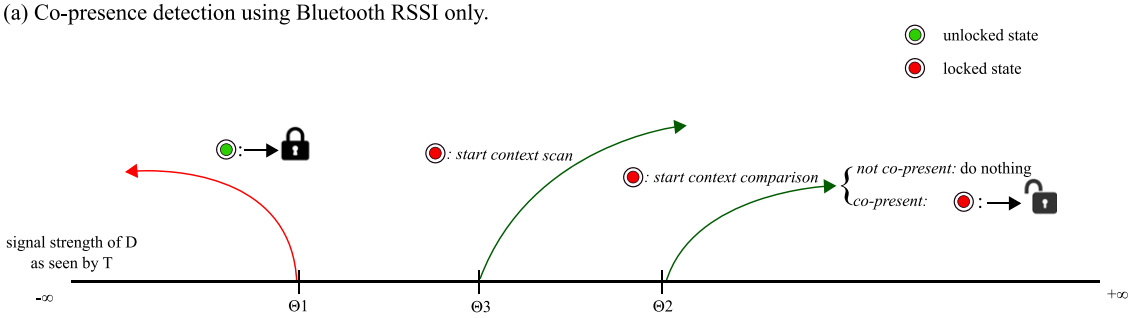
The results for WiFi, Bluetooth and audio are shown in Table 9. (We did not include GPS in this analysis because GPS performed poorly to begin with and spoofing GPS is likely to be harder than the other modalities. Nevertheless, we expect the adversary model to hold for GPS as well and is likely to yield 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 9 shows the difference in false positive rate with respect to the same modalities in the absence of the attacker.

Table 9
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 2	−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



(a) Co-presence detection using Bluetooth RSSI only.



(b) Co-presence detection using context comparison and Bluetooth RSSI.

Fig. 4. Usage of RSSI-based triggers in BlueProximity++.

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.

7. Application integration: BlueProximity++

To demonstrate the practical feasibility of our results, we have extended the BlueProximity [2] system with contextual co-presence detection. We refer to the extended system as BlueProximity++. In the following we detail the design of BlueProximity++ and present results from a small-scale usability study. We also discuss lessons learned from the study and the resulting design improvements we have implemented.

7.1. Design

Access control triggers: BlueProximity determines co-presence solely based on Bluetooth RSSI (from the device D as measured by the terminal T). Fig. 4(a) shows how BlueProximity uses Bluetooth RSSI to trigger access control (i.e., locking and unlocking) events: T locks if the RSSI falls below a threshold θ_1 and unlocks if it rises above a threshold θ_2. A user can visually change the thresholds from a configuration menu.

In BlueProximity++, we augment the RSSI-based co-presence detection by adding support for context comparison. Fig. 4(b) illustrates how the RSSI thresholds are used in this case. The locking threshold remains as before. We add a new threshold θ_3. When RSSI rises above θ_3, T starts scanning its context using all available sensors. T also instructs D to starts context scanning. When the RSSI crosses θ_3, T starts the context authentication process using the most recent context data (if the most recent context data is stale, a new scan is triggered). Context authentication use co-presence detection using the classification model which resulted from the analysis in Section 5.3 which returns co-presence status, as "co-present" or "not co-present", which are then used for the locking and unlocking access control decisions. For example, T is unlocked only when its RSSI for D is greater than θ_3 and both D and T are contextually co-present.

Corrective feedback: The classification model we use for contextual co-presence has a very low, although still non-zero, false classification rate. Furthermore, it was trained using the data from all users in our earlier dataset (Section 4.1). The classification resulting from the model may not match what a particular user perceives as co-presence. In addition, the Bluetooth RSSI measurements are not perfectly reliable. All of these can result in incorrect authorization decisions.

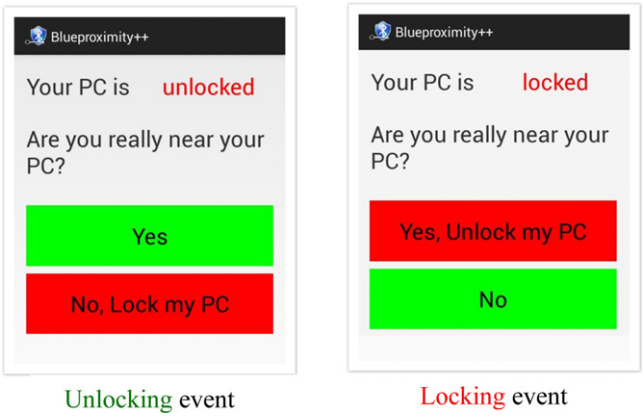


Fig. 5. UI for user notifications of authorization decisions and corrective feedback.

We notify the user of authorization decisions from the user interface of D and allow the user to provide corrective feedback. Fig. 5 shows the notification and feedback UI. In the case of a false negative (T locks when D is co-present), the user can give corrective feedback ("Yes, Unlock my PC") which will result in T being unlocked. Similarly, in the case of a false positive event, corrective feedback ("No, Lock my PC") will result in T being locked.

Ground truth elicited from corrective feedback is stored in databases at T and D. In the future, we can use data to update the model, thereby personalizing it and reducing subsequent classification errors.

D–T **communication channel**: T needs to communicate with D in order to co-ordinate context scanning, send notifications and corrective feedback. We decided not to use Bluetooth for this channel because Bluetooth range is limited to 5–10 m. Co-ordination of anticipatory scanning, and the provision of unlocking notification (and false positive feedback) may occur when the distance between D and T is greater than the Bluetooth range. Instead, we used a proxy server similar to the one in our data collector architecture (Fig. 2). However, the proxy server only provides a communication channel between D–T; no context data or other personal information is stored there. The communication is secured using a shared key established during a binding process.

Configurability: By default BlueProximity++ will use Bluetooth RSSI and context comparison for co-presence detection. But it allows user to configure co-presence determination to be based on Bluetooth RSSI only (i.e., falling back to the functionality of vanilla BlueProximity) In addition it supports all BlueProximity configuration options.

BlueProximity++ consists of two components: A Linux software package for T written in Python and an Android application for D written in Java. We describes details of data flow within BlueProximity++ in the Appendix.

7.2. User study

We conducted a small-scale user study to evaluate the usability of BlueProximity++. Our main objective was to gain indicative (rather than definitive) answers to the following questions:

- How usable is BlueProximity++ in general?
- Does the choice of method for co-presence detection, namely, "Bluetooth RSSI only" (RSSI) or "Context comparison and Bluetooth RSSI" (CC + RSSI) have an impact on usability?

7.2.1. Description

Participants: We recruited ten participants in total (7 from Finland and 3 from the US). All were researchers in different branches of computer science. All had relatively high level of computer literacy (the average self evaluation of computer skills was 8.6 ($\sigma = 1.5$) on a 0–10 scale). All used a personal computer and phone on a daily basis. Most used a password to protect their personal computers. Five participants had a high level of concern for security of their personal computers ("very much"), while three had a moderate level of concern ("somewhat"), and two had a little security concern ("a little"). They had different national backgrounds (French, Finnish, Chinese, Indian, Bangladesh, American, Italian) but were mostly young and male. Table 10 summarizes the demographic details.

Materials: Each participant used a Linux personal computer as their primary computing device. Each participant either owned an Android phone or was given one for the duration of the user study. The participants completed the following five online questionnaires at different stages of the study: the consent questionnaire, the demography questionnaire, the comparison questionnaire, the System Usability Scale (SUS) questionnaire [24] and the open-ended feedback questionnaire. SUS questionnaire is used to calculate an aggregate usability score (out of 100) for the system tested. All questionnaires

Table 10
Demographics of participants.

Information	Details
Age	25–31 years old
Gender	Female (1), Male (9)
Nationality	7 countries
Education	M.Sc. (7), Ph.D.(3)

used are available at the project website.[6] In response to completing the study, each participant was given a voucher valued 25€ (or US $30) at the end of the study.

Design: We used a within-subjects design to see if the choice of co-presence detection technique had an impact on perceived usability. To avoid any potential learning biases, participants were divided into two groups of five. Each participant used BlueProximity++ for a period (up to two weeks) that was divided into two rounds. During round 1, those in *group I* used BlueProximity++ configured to use the RSSI-only mode for co-presence detection. Those in *group II* used BlueProximity++ configured to use context comparison and RSSI (CC + RSSI). During round 2, the configuration was switched. The participants were instructed to respond to the access control notifications as much as possible, especially if they disagreed with the decision.

Procedure: The user study was carried out according to local IRB guidelines. It was conducted in two rounds, sandwiched between three face-to-face meetings with the participants as follows:

- *First meeting*: we introduced BlueProximity++ to all participants, helped them install it and instructed them on basic usage. They were asked to fill in the "demography questionnaire".
- *Second meeting*: After the first round, we collected logged data from devices of the participants. We then changed the configuration of their application to switch the co-presence detection technique it was using. Participants were asked to fill in the "round 1 SUS questionnaire" for the setting they were using during the first round.
- *Third meeting*: after the second round, we again collected logged data and had the participants fill in the "round 2 SUS questionnaire" for the second round setting. They also filled in the "comparison questionnaire" which asked for user preference regarding the two methods. In addition, participants were debriefed in a free-form post-study session where they reported how they used the system during the study.

Communication and support during the study were done via email or in person. After the completion of the user study, we asked the participants to provide us with their overall impression by completing the "open-ended feedback questionnaire".

We encountered two classes of technical problems during the study. The first was related to the operation of BlueProximity++ on diverse Linux platforms. All participants used their own personal computer for the study. They had a wide range of Linux distributions and windowing systems. For example, the screen-saver command that BlueProximity++ needs to use to lock or unlock the display differs depending on the windowing system (such as Gnome, Unity or KDE). These issues were diagnosed and fixed during the first round.

Second, some participants did not always have Internet connectivity on their devices which disconnected *D*–*T* communication channel. Recall that this channel is used to convey access control decisions to the user and for getting corrective feedback. It also is critical when contextual co-presence detection is used, since *D* needs to transfer the gathered context data to *T*. When the *D*–*T* channel is disrupted, contextual co-presence results in false negatives.

7.2.2. Results

SUS scores: Table 11 presents the SUS scores of using BlueProximity++ with the two different techniques for co-presence detection. Fig. 6 shows the distribution of SUS scores in both cases. There is a large variance of SUS score across different participants. The average SUS score is slightly below the level required to consider the system easy to use for both cases.

The SUS scores for the two co-presence detection techniques, summarized in Table 11, are similar. We compared the SUS scores of BlueProximity in the two cases (RSSI-only and CC + RSSI) using a Wilcoxon rank-sum test. No significant effects were found ($Z = 0.49$, $p = 0.63$). To ensure no order effects (i.e., learning effects due to the order in which a participant tried the two methods) were present, we also carried out a repeated measures ANOVA (RM-ANOVA) using *Round* and *Method* as independent variables. Neither main effects (i.e., influence of the round or the selection of the method) nor interaction effects (i.e., combined influence of round and method selection) were found statistically significant.

Note that while the interaction effect was found marginally significant, this was due to two persons in the same group assigning lower scores for all methods due to technical problems resulting from lack of connectivity. The comparable level of usability between the two co-presence detection methods, RSSI-only and CC + RSSI, can also be seen in the responses to the "comparison questionnaire" (which asked participants to directly compare the usability in the two rounds: Four preferred RSSI-only method, while five preferred CC + RSSI one was neutral). Based on this, we may conclude that both systems had similar usability, i.e., the inclusion of contextual co-presence provides comparable usability while at the same time providing better level of security.

[6] http://se-sy.org/projects/coco/userstudy.html.

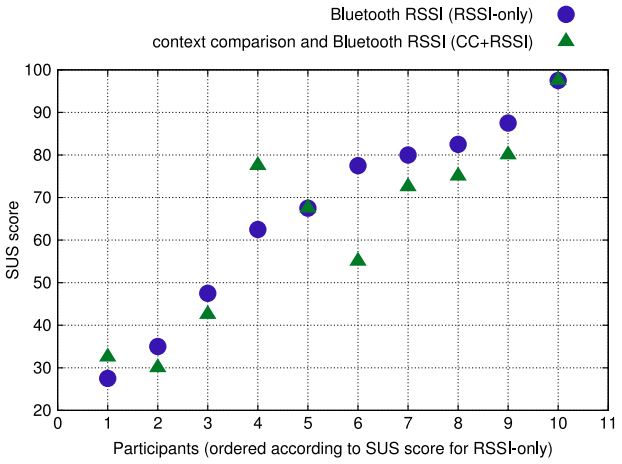


Fig. 6. SUS score distribution.

Table 11
Summary of the SUS scores for the two co-presence detection techniques.

	RSSI-only Mean (std dev)	CC + RSSI Mean (std dev)
Average SUS score	67 (23)	63 (22)

Table 12
Comparison with ground truth.

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

Comparison with ground truth: Recall that we instrumented BlueProximity++ to collect ground truth as part of corrective feedback (as described in Section 7.1). Table 12 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 perceptions as we saw above. However, the initial technical problems (cf. Section 7.2.1) would have impacted the ground truth information.

Qualitative insights: We summarize some relevant information and insights from the responses to the "open-ended feedback questionnaire":

Energy consumption: Six of ten participants did not notice any difference in battery usage of their phones with BlueProximity++ installed compared to their regular usage before the user study. Two participants indicated perceiving greater battery use. The rest indicated that they did not have a baseline for such a comparison. We did not conduct any quantitative measurement of energy consumption in this user study.

Locking/unlocking policies: One participant indicated that he prioritizes zero-interaction *de*-authentication over Zero-Interaction Authentication: "*I would think it could be nice to automatically lock (and only lock) the screen when mobile is going away.*". Another participant who shares an office with four other people would like locking to happen more quickly, "*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*". While participants could change the default RSSI thresholds for the various triggers (and thus change the distance at which locking and unlocking happens), a better alternative would be to use classification techniques that can be updated online, thereby personalizing the model.

Comparison to password-based authentication: Four participants prefer ZIA over typing in a password to unlock a PC. Three prefer passwords over ZIA while the remaining three thought that both ZIA and passwords are needed. One participant said: "*my mother and sister always leave their mobile phone somewhere, and it could be easily taken and used to unlock their computer*

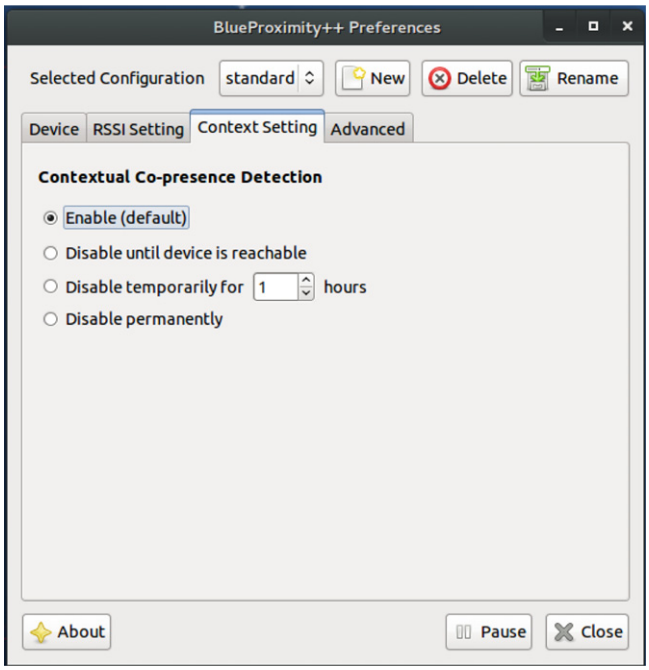


Fig. 7. Settings for co-presence based on context comparison.

without them knowing it if someone wanted to". The only female participant echoed these sentiments indicating that at home, she did not always have her phone 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 indicates that people do not always carry phones with them [25]. These observations highlight that a wearable device with sensing capabilities, such as smart watches, glasses or wristbands, that are now becoming available, may be better candidates for ZIA.

7.3. Design improvements

Based on the results of our user study, we made a number of improvements for BlueProximity++. Notable among them are:

- **Bluetooth** *D–T* **channel**: In addition to the proxy based *D–T* channel, we now use a direct Bluetooth *D–T* channel as well to provide better fault tolerance. Small amounts of data (e.g., notifications) are sent via both channels. Voluminous data (e.g., audio context data) is sent only via the proxy.
- **Finer-grained control for context comparison**: Users can now disable contextual co-presence detection either temporarily or until *D* becomes reachable to *T* (Fig. 7). This can be used in cases where the user does not have connectivity via the proxy (e.g., while traveling). It is also possible for the user to turn off BlueProximity++ completely, thereby falling back to standard interactive authentication by typing in a password to unlock their computers.

8. Related work

We review prior work on utilizing contextual information for relay attack resilience.

Relay attack resilience: Distance bounding [11] techniques are commonly used to avoid relay attacks [26,27]. As we saw in Section 1, it may not be realistic for commodity devices.

Halevi et al. [14] developed techniques using ambient audio for co-presence detection. Their experiments were done using identical device models rather than using different device models as in our work. Our results show that in such divergent scenarios, their techniques perform less well than in [14]. Nevertheless, their techniques are the best among different audio techniques we tested for ambient audio.

Narayanan et al. [28] studied the use of various modalities for private proximity detection and concluded that on WiFi broadcast packets and WiFi access point IDs are likely to perform best. Our systematic experiments confirm that WiFi access point IDs perform well. We ruled out the use of WiFi broadcast packets because they are not accessible to applications in ordinary smartphone platforms.

Krumm et al. [12] proposed "NearMe" which uses WiFi similarity features for proximity detection. They built a model using data collected in an office building environment and tested in a cafeteria environment. They conjecture that their

approach generalizes well to other settings. Our analysis with the uncontrolled dataset collected from diverse environments confirms that WiFi access point and signal strength information works well for general settings.

Czeskis et al. [29] proposed "secret handshakes" to avoid ghost-and-leech attack by limiting the context where the contactless card communicates with the reader. They used only accelerometer data as contextual information.

Pairing using contextual information: Secure pairing using contextual information is a harder problem in that it requires the two devices being paired to extract sufficient entropy (e.g., 128 bits) from the context to serve as a cryptographic key. In contrast, contextual co-location determination does not require secrecy for the context information. There has been significant work in secure device pairing using contextual information such as WiFi or audio. Schurmann et al. [15] presented an approach that uses binary fingerprints from ambient audio to establish a secure channel between two co-present devices. Varshavsky et al. [13] presented Amigo to authenticate co-present devices using various features extracted from the WiFi environment. All such previous work on pairing has focused on a single sensor modality. In contrast, we consider the use of multiple modalities simultaneously.

9. Summary and discussion

In this paper, we addressed the issue of using different sensor modalities for co-presence detection to be used in applications that need ZIA. To the best of our knowledge, our work is the first that fairly compares performance of different modalities and shows that the use of multiple modalities can improve security of co-presence detection (relay attack resilience) without significantly degrading its usability. We used our insights to enhance an existing ZIA scheme to design and implement the first ZIA scheme that integrates contextual co-presence detection.

Energy consumption: One potential concern in adding context sensing to ZIA is the effect on battery consumption. However, this is not a serious problem for the following reasons. First, context sensing is initiated only when D makes an authentication request in proximity of T, *and* T is in the "locked" state". Second, a pre-requisite for context sensing is an authenticated trigger sent from D to T, which precludes denial of service attacks. Third, although a relay attacker could cause repeated triggers, this can be resisted by system design, such as introducing exponential back off after a small number of failed authentication attempts. A large majority of our user study participants also did not perceive any noticeable difference in battery consumption when using context sensing.

Limitations: Our data collection and analysis is targeted for evaluating contextual co-presence techniques for the particular use case of ZIA. As such, our results (such as the effectiveness of WiFi as a sensor modality) may not generalize to other applications of contextual co-presence detection. We let users decide what constitutes co-presence, which is a reasonable approach for evaluating ZIA but may not be so for other applications. On the positive side, our data collection framework can be easily adapted to collect data for different scenarios, such as indicating ground truth in terms of the exact distance rather than as a Boolean value.

The accuracy required for co-presence detection varies from application to application. Apart from the small-scale controlled experiment in Section 5.4, we did not focus on estimating the exact granularity of co-presence in terms of distance. Other work, such as NearMe [12], suggest that co-presence can be determined reliably for distances of around 20 m.

It is possible that co-presence determination is inconclusive, for example because sufficient sensor data is not available in a given situation. In such a case, depending on the application, it may be reasonable to trade off usability for security by relaxing the "zero interaction" requirement, e.g., by asking the user U to confirm co-presence on device D (recall that D and T share a security association).

The environments where our participants collected data might not represent all possible environments that other people have. For example (as seen from Table 2) our participants took samples in environments with significant presence of WiFi APs. The environments where our participants collected data may not represent all possible environments that other people have. For example (as seen from Table 3) our participants took samples in environments with significant presence of WiFi APs. These environments are generally representative of home, office, and other types of indoor scenarios, and as such representative of usage situations of BlueProximity and similar applications. However, in other types of applications, such as in car unlock systems, the radio environment is likely to be sparser and alternative context sources may be required. We also note that the performance of all radio frequency techniques is sensitive to the number of reference points (devices, access points, satellites) that can be observed. In particular, our results seem to suggest that WiFi is current the best option in terms of prevalence rate, and further evaluations are needed to understand the exact characteristics of the underlying technologies.

The usability evaluation of our authentication app, BlueProximity++, was done on a small population high levels of self-reported computer skills.

Extensions: It is natural to consider the use of other forms of sensor modalities. Indeed, in a recent paper [30], we investigated the use of sensor modalities that represent the *physical* ambient environment. We are expanding the data collection to include more users so that we can resolve the question of whether personalized models are more effective for contextual co-presence detection than common, offline models. We are incorporating our model into BlueProximity and plan to conduct a user study to evaluate its usability. Our adversarial analysis is intended as a first step—the question of how to characterize a context adversary formally yet realistically remains open.

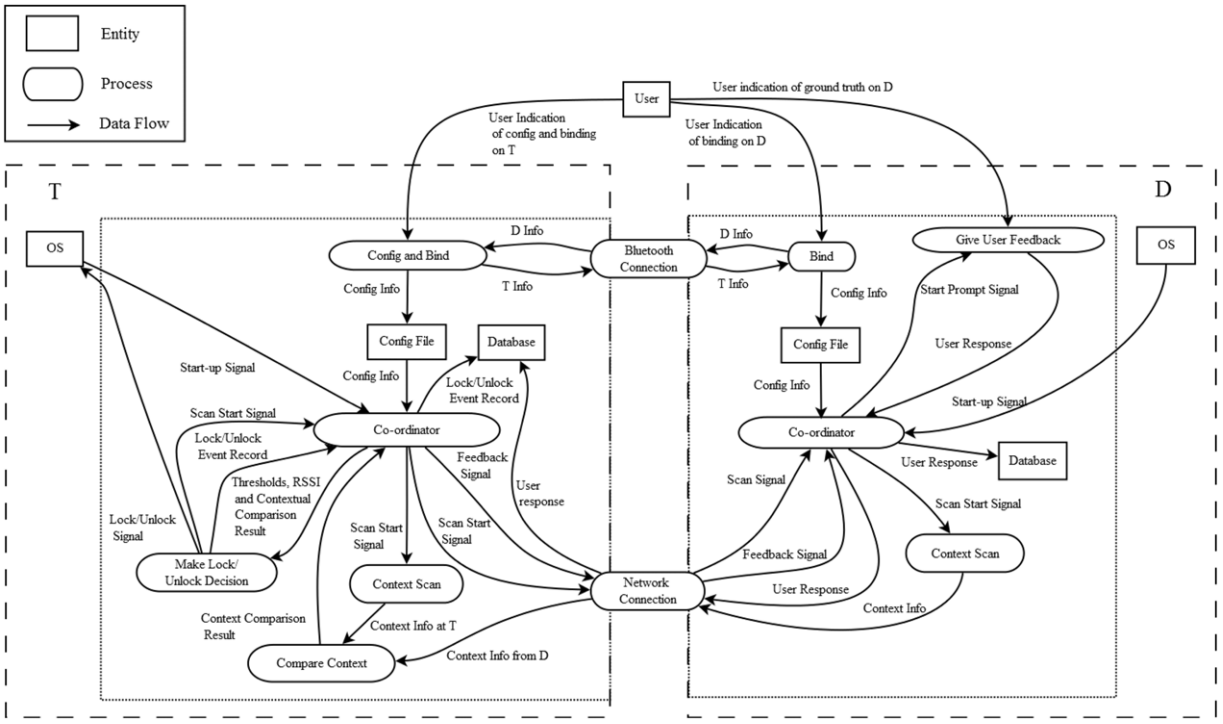


Fig. 8. Data flow in BlueProximity++.

Acknowledgments

This paper is an extended version of our previous paper [31]. We thank the user study participants for their time and feedback. Our work was partially supported by a Google Faculty Research Award, a donation from Nokia, Academy of Finland (*Contextual Security* project-274951) and US NSF grant CNS-1201927. Xiang Gao and Petteri Nurmi were supported by TEKES as part of the Internet of Things and Data to Intelligence programs, respectively, of DIGILE (Finnish Strategic Centre for Science, Technology and Innovation in the field of ICT and digital business).

Appendix. BlueProximity++ data flow diagram

The data flow diagram provides an overview of the system-wide data flow in BlueProximity++ (see Fig. 8). The regular rectangles represent the entities as data source or sink; the round-cornered rectangles represent processes operating on or transporting data; and arrows represent 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 on T and to the *Context Scan* process on D (via *Network Connection* and *Co-ordinator* at T). The *Compare Context* process gets the resulting *context info* from both-sides 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.

References

[1] M.D. Corner, B.D. Noble, Zero-interaction authentication, in: Proc. 8th Annual International Conference on Mobile Computing and Networking, MobiCom'02, ACM, New York, NY, USA, 2002, pp. 1–11. URL: http://doi.acm.org/10.1145/570645.570647.
[2] BlueProximity, SourceForge Project. http://sourceforge.net/projects/blueproximity/.
[3] A. Czeskis, M. Dietz, T. Kohno, D. Wallach, D. Balfanz, Strengthening user authentication through opportunistic cryptographic identity assertions, in: Proc. 2012 ACM Conference on Computer and Communications Security, CCS'12, ACM, New York, NY, USA, 2012, pp. 404–414. URL: http://doi.acm.org/10.1145/2382196.2382240.

[4] B. Tognazzini, The Apple iWatch, Blog posting in AskTOG: Interaction Design Solutions for the Real World, Febraury 2013. http://asktog.com/atc/apple-iwatch/.
[5] K.-P. Yee, Aligning security and usability, IEEE Secur. Privacy 2 (5) (2004) 48–55.
[6] D. Dolev, A.C.-C. Yao, On the security of public key protocols, IEEE Trans. Inform. Theory 29 (2) (1983) 198–207.
[7] A. Levi, E. Cetintas, M. Aydos, C. Koc, M. Caglayan, Relay attacks on bluetooth authentication and solutions, in: Computer and Information Sciences, ISCIS, 2004.
[8] Z. Kfir, A. Wool, Picking virtual pockets using relay attacks on contactless smartcard, in: Proc. First International Conference on Security and Privacy for Emerging Areas in Communications Networks, SECURECOMM'05, IEEE Computer Society, Washington, DC, USA, 2005, pp. 47–58. http://dx.doi.org/10.1109/SECURECOMM.2005.32.
[9] A. Francillon, B. Danev, S. Čapkun, Relay attacks on passive keyless entry and start systems in modern cars, in: Proc. Network and Distributed System Security Symposium, NDSS, 2011.
[10] L. Francis, G. Hancke, K. Mayes, K. Markantonakis, Practical NFC peer-to-peer relay attack using mobile phones, in: Proc. 6th International Conference on Radio Frequency Identification: Security and Privacy Issues, RFIDSec'10, Springer-Verlag, Berlin, Heidelberg, 2010, pp. 35–49.
[11] S. Brands, D. Chaum, Distance-bounding protocols, in: Workshop on the Theory and Application of Cryptographic Techniques on Advances in Cryptology, EUROCRYPT'93, Springer-Verlag New York, Inc., 1994, pp. 344–359. URL: http://dl.acm.org/citation.cfm?id=188307.188361.
[12] J. Krumm, K. Hinckley, The NearMe wireless proximity server, in: Proc. Ubiquitous Computing, UbiComp, 2004.
[13] A. Varshavsky, A. Scannell, A. LaMarca, E. De Lara, Amigo: proximity-based authentication of mobile devices, in: Proc. 9th International Conference on Ubiquitous Computing (UbiComp), Springer, 2007, pp. 253–270.
[14] T. Halevi, D. Ma, N. Saxena, T. Xiang, Secure proximity detection for NFC devices based on ambient sensor data, in: Proc. 17th European Symposium on Research in Computer Security (ESORICS), in: Lecture Notes in Computer Science, vol. 7459, Springer, 2012.
[15] D. Schurmann, S. Sigg, Secure communication based on ambient audio, IEEE Trans. Mob. Comput. 12 (2) (2013) 358–370. http://dx.doi.org/10.1109/TMC.2011.271.
[16] D. Ma, N. Saxena, T. Xiang, Y. Zhu, Location-aware and safer cards: enhancing RFID security and privacy via location sensing, IEEE Trans. Dependable Secure Comput. 10 (2) (2013) 57–69. http://dx.doi.org/10.1109/TDSC.2012.89.
[17] O. Dousse, J. Eberle, M. Mertens, Place learning via direct WiFi fingerprint clustering, in: 2012 IEEE 13th International Conference on Mobile Data Management, Vol. 0, 2012, pp. 282–287. http://doi.ieeecomputersociety.org/10.1109/MDM.2012.46.
[18] W. Gellert, S. Gottwald, M. Hellwich, The VNR Concise Encyclopedia of Mathematics, second ed., Van Nostrand Reinhold, New York, 1989.
[19] H. Lu, W. Pan, N.D. Lane, T. Choudhury, A.T. Campbell, SoundSense: scalable sound sensing for people-centric applications on mobile phones, in: Proc. 7th International Conference on Mobile Systems, Applications, and Services, 2009, pp. 165–178.
[20] A. Wang, The Shazam music recognition service, Commun. ACM 49 (2006) 44–48.
[21] W.-T. Tan, M. Baker, B. Lee, R. Samadani, The sound of silence, in: Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems, SenSys'13, ACM, New York, NY, USA, 2013, pp. 19:1–19:14. URL: http://doi.acm.org/10.1145/2517351.2517362.
[22] G.I. Webb, MultiBoosting: a technique for combining boosting and wagging, Mach. Learn. 40 (2) (2000) 159–196. http://dx.doi.org/10.1023/A:1007659514849.
[23] N.O. Tippenhauer, K.B. Rasmussen, C. Pöpper, S. Čapkun, Attacks on public WLAN-based positioning systems, in: Proc. 7th International Conference on Mobile Systems, Applications, and Services, MobiSys'09, ACM, New York, NY, USA, 2009, pp. 29–40. URL: http://doi.acm.org/10.1145/1555816.1555820.
[24] J. Brooke, SUS: a quick and dirty usability scale, in: P.W. Jordan, B. Weerdmeester, A. Thomas, I.L. Mclelland (Eds.), Usability Evaluation in Industry, Taylor and Francis, London, 1996.
[25] A.K. Dey, K. Wac, D. Ferreira, K. Tassini, J.-H. Hong, J. Ramos, 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, ACM, New York, NY, USA, 2011, pp. 163–172. URL: http://doi.acm.org/10.1145/2030112.2030135.
[26] G. Hancke, M. Kuhn, An RFID distance bounding protocol, in: Security and Privacy for Emerging Areas in Communications Networks, 2005. SecureComm 2005, 2005, pp. 67–73. http://dx.doi.org/10.1109/SECURECOMM.2005.56.
[27] J. Reid, J.M.G. Nieto, T. Tang, B. Senadji, Detecting relay attacks with timing-based protocols, in: Proc. 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS'07, ACM, New York, NY, USA, 2007, pp. 204–213. http://dx.doi.org/10.1145/1229285.1229314.
[28] A. Narayanan, N. Thiagarajan, M. Lakhani, M. Hamburg, D. Boneh, Location privacy via private proximity testing, in: Proc. Network and Distributed System Security Symposium, NDSS, 2011.
[29] A. Czeskis, K. Koscher, J.R. Smith, T. Kohno, RFIDs and secret handshakes: defending against ghost-and-leech attacks and unauthorized reads with context-aware communications, in: Proc. 15th ACM Conference on Computer and Communications Security, CCS'08, ACM, New York, NY, USA, 2008, pp. 479–490. http://dx.doi.org/10.1145/1455770.1455831.
[30] B. Shrestha, N. Saxena, H.T.T. Truong, N. Asokan, Drone to the rescue: relay-resilient authentication using ambient multi-sensing, in: Proc. Eighteenth International Conference on Financial Cryptography and Data Security 2014.
[31] H.T.T. Truong, X. Gao, B. Shrestha, N. Saxena, N. Asokan, P. Nurmi, Comparing and fusing different sensor modalities for relay attack resistance in zero-interaction authentication, in: IEEE International Conference on Pervasive Computing and Communications, PerCom 2014, Budapest, Hungary, March 24–28, 2014, 2014, pp. 163–171.