ABSTRACT

Due to practical demands, usability in security systems is reconsidered by researchers in recent years. Given the three prevailing short-range communication technologies - Radio Frequency Identification (RFID), Near Field Communication (NFC), and Bluetooth, Zero-Interaction Authentication (ZIA) is proposed for a subset of their application scenarios. Nevertheless, relay attacks especially “ghost-and-leech” have emerged as threats against ZIA model.

In this paper, we will present the co-location-based solution to defend ZIA against relay attacks. The idea is to verify the co-location of two principals involved in authentication by comparing their contextual features. The contextual features (known as context tags) are extracted and synthesized from the sensing data of related contextual attributes. We will present 6 categories of context tags (audio, light, GPS, acceleration, wireless broadcast traffic, and nearby device-IDs) with principles, proof-of-concept experiments and specific application scenarios. We will also evaluate the advantages and limitations of each context tag, and discuss probable problems for our solution. Finally, we will introduce our experiment on BlueProximity, a Bluetooth computer lock application, in order to demonstrate the probable attacks and the feasibility of contextual co-location-based authentication.

Keywords
Zero-Interaction Authentication, relay attacks, context tags, co-location, correlation

1. INTRODUCTION

Due to practical demands for ease-of-use, usability in security systems is reconsidered by researchers in recent years. Zero-Interaction Authentication [4] is one of the concepts proposed to improve the ease-of-use in security system design. It describes a model of authentication without explicit labor of manual interaction between a user and the service portal. There are three fundamental short-range communication technologies: Radio Frequency Identification (RFID) [8], Near Field Communication (NFC) [7] and Bluetooth [2]. RFID consists of readers and tags and works in a challenge-response manner to provide contactless authentication. NFC is a promising standard based on RFID enabling short-range communication (<10cm) and contactless transactions between two handsets. These two short-range communication technologies are so widely applied in public and private services: credit card payment, electronic passports, transport cards, door access cards, and vehicle keys. Bluetooth is a short-range communication (1-10m) technology defined in standard IEEE 802.15.1, and is used for data exchange with key generation based on PIN.

However, potential threats have emerged with relay attacks. The probable attacks can be "ghost-and-leech", thus the secure identity information in a RFID tag can be used to cheat in authentication by an attacker, when the tag and reader are actually not co-located. This calls for new techniques to prevent the authentication process from such relay attacks while preserving excellent usability. The traditional solution - Distance Bounding Protocol, a protocol that calculates proximity by measuring Round Trip Time, works fine but is restricted by time delay and hardware. So here we focus on the automatic authentication method based on proximity.

The contextual method that we focus on in this paper is to prove co-location of two devices by comparing the contextual features (i.e. context tags) derived from environmental attributes. These context tags are required to be unique for different locations, and are generated and captured by the sensors equipped in daily handsets (smartphones, tablets). This method offers prominent benefits: it requires no extra actions for the users to complete the authentication procedure since being close is enough, so you will be able to pay in the supermarket by showing your NFC smartphone near to the reader instead of manually inputting PIN code; the context tags sensed from environmental attributes make the system intrinsically difficult to attack, because the complexity and dynamics of environment restricted the similarity; location privacy is better protected by avoiding direct use of location information.

We present the contextual co-location methods as well as related applications in the following sections. Section 2 intro-
roduces the problem and motivations for this paper, including Zero-Interaction Authentication, relay attacks and distance bounding defense. Section 3 illustrates the system requirements, and presents the various context tags classified by ambient attributes and two proof-of-concept experiments. Section 4 compares aforementioned context tag techniques. Section 5 describes my experiment on BlueProximity. And finally Section 6 concludes this paper.

2. ZERO-INTERACTION AUTHENTICATION

In this section, we present the problem and motivations for this paper, including Zero-Interaction Authentication, relay attacks, and distance bounding defense.

Zero-Interaction Authentication (ZIA) [4] is 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 contactless key, or unlocking a laptop everytime the user returns from a break. In ZIA, the user holds a token as a contactless key to decrypt the encrypted system (a locked car or a locked laptop) via a wireless link in a short range.

Figure 1: ZIA: token authentication model [4].

Figure 1 describes the general model of ZIA process. To initiate a token, the user need authenticate a PIN to the token, and bind the token with target device to ensure the token is the only one responsive. Then each principal of pair authenticates its identity to each other, and establishes a session key. When the user with token leaves out of the range, the paired device is automatically locked. And when the user with token returns, the decryption process starts from authentication and session key establishment.

According to the standard assumption of RFID (ISO 14443) [9, 8], the allowed upper boundary of the distance between tag and reader is 10cm, and the time-out condition for data transmission is 5 sec. So there are space and enough time to construct a basic relay attack. In this attack, the attacker places a "leech" near to the tag, and places a "ghost" near to the reader. The leech pretends to be a reader for the genuine tag, while the ghost pretends to be a tag for the genuine reader. Both leech and ghost are responsible for forwarding data without processing. Since leech and ghost communicate via fast digital channel, the distance can be much longer than the genuine pair. And with repeaters between leech and ghost, the "ghost-and-leech" attack can be achieved for unlimited distance in theory.

The distance limitation between tag and leech probably lead to failure in attacking, since leech has to be too close to the tag. However, according to Kfir and Wool’s work [9], they made a demo with NFC devices extending the leech-tag distance to 50cm and the ghost-reader distance to 50m. Now it’s safe and feasible to open the car door with ghost-and-leech attack even when the owner is actually miles away. And for Bluetooth-enabled devices, the range of communication can be 10-100 meters, which is a loose distance limitation. In pairing of Bluetooth devices, a master key is generated as the shared secret key, and packets are encrypted for communication between the two paired devices [2].

So how to defeat ghost-and-leech attacks? Here are two optional solutions. The first and straightforward one is to add user’s involvement. This solution is suitable for users with NFC smartphones. After the first challenge-response trip, NFC portal gives a prompt on screen. Only after the user has checked the transaction details and confirmed when the payment is committed. Its weakness is obvious: the user needs extra actions.

An alternative approach is distance bounding protocols [5, 15, 13]. In distance bounding protocols (see Figure 3), there is a pair of principals "verifier" and "prover". Both principals are mutually trusted. The verifier sends a challenge message to prover, and prover sends the response back after a short interval of processing time $T_p$. On receiving the response, it gets the measured round-trip time $RTT$. So the distance between verifier and prover is

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

and $c$ is the speed of signal propagation. It emphasizes the thinking of proving proximity, which can defeat ghost-and-leech attacks.

3. CONTEXT TAGS

In contrast to distance bounding defense, we use locally generated context tags to test co-location. Since the two principals are assumed trusted (paired), the "ghost-and-leech" attack is defeated. The general model of proving co-location is shown in Figure 4.

We need to identify the requirements for context tag tech-
niques before going into details. The requirements [7] listed below are derived and summarized from the related references.

- **Unforgeability**: It is difficult to fake the context for an attacker. This ensures that the system can hardly be compromised by constructing the corresponding environmental attributes at another location. In this sense, complexity, dynamics and noise for a specific context tag improves the level of Unforgeability. A typical example is indoor temperature, steady and simple, which can be copied to another location for relay attack.

- **Efficiency**: The amount of computation needed for measuring sensor data, extracting feature values and making comparison is relatively small.

- **Robustness**: Robustness refers to the ability to tolerate errors in making co-location decisions. We can use detection rate to identify the percentage of correct decision among all sample pairs. There are false positive errors (i.e. each device resides at a remote location while the system judges it as co-located) and false negative errors (i.e. both devices stay co-located while the system judges it as remotely located). The system is considered robust if the detection rate is approaching 100 percent and error rates are relatively small.

- **Usability**: Remember the ease of use when choosing contextual attributes, considering environmental and handset restrictions. Here are two negative examples. One example is to use odor sensors to capture odor tags, which requires extra components.

In the following subsection, we present the various context tags classified with ambient attributes: audio, light, GPS data, acceleration, wireless broadcast traffic, and nearby device IDs.

### 3.1 Ambient Audio

Ambient audio can be a suitable source for generating context tags, since the audio fingerprint is different at different locations, according to Halevi et al.’s work [7]. Besides, ambient audio conditions are influenced by noise, thus are full of dynamics, which adds to complexity and security. So firstly we decide to use microphones in smartphones to record audio signals. For any pair of smartphones, audio recording starts simultaneously for a short interval.

After acquiring audio tags, we need transfer the sample time-based sequence to meaningful values that are easy for processing. Here are the correlation techniques employed by Halevi et al. [7] to derive the similarity or distance:

- **Time-based**: Given audio signal sample $X_i$ and $X_j$, which are normalized by setting the sum of energy to 1, $S$ stands for similarity, and $D$ stands for difference.

  For correlation method:
  \[
  S(i, j) = \max(CrossCorr(X_i, X_j))
  \]
  \[
  D(i, j) = 1 - S(i, j)
  \]

  And for difference method:
  \[
  D(i, j) = ||X_i, X_j||
  \]
  \[
  S(i, j) = 1 - D(i, j)
  \]

- **Frequency-based**: Fast Fourier Transform (FFT).

- **Time-Frequency-based**: Given 2 normalized audio signal samples $X_i$ and $X_j$, $S$ stands for similarity, and $D$ stands for difference.

  \[
  D(i, j) = \sqrt{(D_{time}(i, j))^2 + (D_{freq}(i, j))^2}
  \]
  \[
  S(i, j) = 1 - D(i, j)
  \]

Another method of generation audio tags is acoustic fingerprinting, a widely-used technique to identify a certain piece of sound effects or music from crowded libraries. Acoustic fingerprinting is the method of extracting characteristic patterns from audio sequence. Although studies on music fingerprinting operate on music properties like amplitude, rhythm, contour and pitch, the ambient audio sequence calls for more general techniques. According to Schurmann et al.’s work [16] to establish audio fingerprinting-based authentication, they developed a energy-based fingerprinting method for audio sequences, based on Haitsma and Kalker’s
robust fingerprinting algorithm [6]. The general idea is to generate a fingerprint of 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. Firstly, divide the audio sequence $S$ into $n$ frames $S_i, i \in 0, \ldots, n - 1$ on which respectively applied discrete Fourier Transformation (DFT). Secondly, split each frame $S_i$ linearly and evenly into $m$ non-overlapping frequency bands $S_{ij}, j \in 0, \ldots, m - 1$. Thirdly, establish an energy matrix $E$ with the sum of energy values for each $S_{ij}$ as elements: $E_{ij} = \sum S_{ij}[k]$. Fourthly, the fingerprint is generated from the differences between two successive $E_{ij}$:

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

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

Halevi et al.’s experiments indicates zero error, while Schurmann et al.’s experiment on comparing synchronously sampled data indicates around 70 percent similarity. So audio correlation approach seems better than audio fingerprinting approach according the the results. It is noise that increases errors in the audio fingerprint bit sequence. Besides, the typical and suitable scenarios for audio context tags are the places with strong background audio sources, e.g., cafe, concert hall, CD store. This is because the audio measurement tends to be larger in such scenarios than quiet places or places with evenly distributed noise.

### 3.2 Ambient Light

Light illuminance is also a variable changing at different locations inside a room, so it can be a candidate for context tags [7]. 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 between location $i$ and $j$ is:

$$D(i, j) = |L_i - L_j|$$

Halevi et al.’s work showed the availability to use ambient light as an acceptable context tag, but still errors that happen in experiments undermines the robustness of such a technique. So light is not as a strong context tag as audio.

### 3.3 GPS

According to our daily experience, normal GPS is not a smart tool to locate positions. Much 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: there is a slow start process for GPS taking history point as initial value; GPS responds faster for mobile objects than static ones; the military level of GPS, which we seldom use, has better performance than the ordinary level. Narayanan et al.’s report [14] proves the unforgeability with more than 4 satellites. We can reach better performance by installing extra components like professional GPS receivers, but that undermines the usability.

Ma et al.’s work [10] on location-aware RFID security demonstrates how to utilize location information from GPS as a context tag. The authors’ work is similar to our proposed work to some extent, since they use GPS modules to provide context comparison enhancing simple RFID authentication against relay attacks. They selected longitude, latitude and speed together as the context tag. According to NMEA standard, they extract position, velocity and time values from GPRMC (recommended minimum specific GPS/Transit data) sentence, with GPGGA (Global Positioning System Fix Data) sentence serves as a fix. And they get Doppler-speed accurately from GPRMC. In the proof-of-concept experiment, they used a GPS receiver module instead of smart handsets to acquire GPS information. The GPS receiver module has the feature of rapid satellite acquisition, ensuring acceptable performance compared to built-in GPS module in smartphones. The average of every 10 readings from GPS is taken to decrease errors. The experiment tries to compare location vectors acquired from GPS with the data written into EEPROM (known as “valid” points), but the idea is the same: use threshold to compare and prove proximity. The threshold is set as the square region with the valid point as the center. According to the result of the selective unlocking experiment [10], location information acquired from GPS is a strong and accurate context tag.

It is overkill to prove co-location using concrete GPS location coordinates. Besides, GPS location coordinates are not available in many scenarios like indoor environment. So people are motivated to develop a variation, i.e. utilizing GPS raw data to generate context tags. GPS raw data are always available whenever the GPS receiver is on. Multiple attributes can be extracted from GPS raw data: satellite ID, signal strength, speed, time, etc. It is reasonable to make the list of multiple attributes (at least satellite ID and signal strength) the context tag.

### 3.4 Acceleration

The first strategy of using acceleration tags is recording background acceleration. It is specially suitable for applications on transportation, and different transportation modes features unique traces of environmental acceleration. V.Manzoni et al.’s technical report [11] introduces an innovative system CO2GO for estimating real-time personal CO$_2$ emissions. It is a featured project developed by SENSEable city lab of MIT. CO2GO classifies transportation into eight modes and computes CO$_2$ emissions based on the transportation mode and distances. The system features in automatic transportation mode inference, which utilizes data processed from accelerometers and GPS receivers to determine the identity. Google Nexus One with Android 2.2 samples the accelerometer and GPS data, which is then processed with Fast Fourier Transformation (FFT). Supervised machine learning algorithms based functional trees are adopted to extract the patterns for eight modes from processed accelerometer traces (see Figure 5).
Thus we choose an alternative approach, i.e. taking the list of visible WIFI access point IDs along with radio signal strength (RSS) as the context tag. According to Narayanan et al.’s statistics [14], on average, around half of the number of IDs in the visible list is different from that at a different location. But we can still improve the complexity of our context tags with RSS. So WIFI ID can be a moderate candidate for proximity testing. Bluetooth and cellular IDs along with RSSs follows the similar way of WIFI, and can be considered moderate context tags.

There are still weaknesses for nearby device ID and RSS tags. WIFI, Bluetooth are restrained to indoor scenario. And all three types of context tags are influenced by the density of nearby devices. Besides, since the context tag is a list of IDs and RSSs instead of a value sequence, granularity is crucial when defining co-location. Moreover, given granularity, low false positive errors are guaranteed; but false negative errors can be high, because even two devices at the same location may witness different lists of nearby device IDs and RSSs.

### 3.7 Experiments

Here we present two demonstrations of proof-of-concept experiments with three essential techniques involved: correlation, confusion matrix, and error analysis. Correlation (sometimes difference) is used for measuring the similarity (or distance) between two data sequences. Confusion matrix is a tool of machine learning used for establishing the threshold if the diagonal is distinct from the rest elements. And in error analysis, we need false positive errors and false negative errors. A false positive error occurs when two objects with an actual distance beyond the threshold are considered co-located by the program. A false negative error occurs when two objects actually co-located are considered away from each other by the program.

Our subjects are T. Halevi et al.’s experiments [7] on ambient audio and light. The researchers applied audio- and light-based proximity proving mechanisms to NFC model in payment scenario. Since primitive NFC payment is prone to relay attacks, the researchers added encrypted location segments to both user-side and merchant-side terminals. The location segments are generated locally from equipped sensors and are authenticated through the third party server of bank.

For the audio experiment, the researchers used two Nokia N97s with programmes to capture contextual sound via built-in microphones.

1. **Technique selection**: The researchers established “dataset 1” with groups of 1-second recordings, with each group recorded at a different location scenario. Then they applied different correlation techniques (time-based, frequency-based, time-frequency-based) to compute the similarities. The detection results are generated by comparing the similarity from different location pairs with the similarity from the same location pair. According to the final result, time-frequency-based correlation technique had the best detection rate (53%) and is selected for the following experiments.

2. **Classification formula**: The author establish "dataset
In this paper, we have reviewed the categories of context tags: audio, light, GPS data, acceleration, wireless broadcast traffic, and nearby device IDs. For each of them, we have identified the structure of such tags, the extraction or fingerprinting techniques, and how they are applied in experiments.

As indicated by Table 1, the best context tags that we recommend for security authentication are ambient audio, GPS, acceleration (shaking) and wireless broadcast traffic. Audio fingerprint, light and nearby WiFi/Bluetooth/cellular IDs+RSSs are moderate context tags with acceptable small errors according to the experiments.

Moreover, there are limitations for recommended context tags.

- Audio is a strong and robust context tag, and suitable for most circumstances, but the typical scenarios are public areas like cafe, concert hall, library, and restaurant that are tested in various experiments. Some quiet places with even and static distribution of audio signals may extract fingerprints of mainly noise, and thus prone to relay attacks.

- Light is strong in certain cases like restaurants and department stores, but may witness errors in cases like car dealers, according to the experiment. And light illuminance can be affected by the orientation of hands, which increases the false negative errors.

- GPS is strong and accurate as a context tag when directly used, since locations are absolutely different by location vector. However, the restriction of hardware makes it slow in acquisition data from satellites, while adding extra hardware undermines usability. GPS raw data is feasible in most scenarios, but may experience similar problems as nearby devices, e.g. high false negatives.

- Acceleration (shaking) is a strong, unforgeable context tag, but calls for extra user interaction.

- Wireless Broadcast Traffic is strong and robust, although limited to the same wireless network for normal usage. It also calls for sufficient density of access points, and can be applied to campus, libraries and other public areas.

- Nearby device-IDs + RSSs are moderate candidates for context tags, but still density of nearby devices is required. They may experience high false negative errors.

4.2 Probable problems

Replay attack is a probable problem if no timing information is encrypted when sending context vectors. The attacker can record such a message and use it afterwards to cheat the device making comparison. This can be solved by adding timing information to the message like

\[(ID1, ID2, contextvector, timestamp)_K\]

where K is the shared key used for encryption. However, the comparison of timestamps calls for time synchronization. So the better solution would be challenge and response with a nonce (random number for each round) incorporated:

\[Challenge : (ID2, ID1, N_1)_K\]

\[Response : (ID1, ID2, contextvector, N_1)_K\]

Another problem is faking context attack. Contextual proving proximity is inherently against relay attacks. But in some cases, it is possible to construct a fake context similar to the valid context, or only faking essential environmental attributes. This is the case when unforgeability is broken. The cost of such attacks depends on the complexity of contextual attributes.
<table>
<thead>
<tr>
<th>Context Tag</th>
<th>Evaluation</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ambient Audio</td>
<td>strong (correlation), moderate (fingerprint)</td>
<td>scenarios with audio sources</td>
</tr>
<tr>
<td>Ambient Light</td>
<td>moderate</td>
<td>influenced by orientation of devices</td>
</tr>
<tr>
<td>GPS</td>
<td>strong</td>
<td>hardware; high false negatives</td>
</tr>
<tr>
<td>Acceleration</td>
<td>strong</td>
<td>need user interaction (shaking)</td>
</tr>
<tr>
<td>Wireless Broadcast Traffic</td>
<td>strong</td>
<td>density of access points</td>
</tr>
<tr>
<td>Nearby Device-IDs + RSSs</td>
<td>moderate (WIFI, Bluetooth, cellular)</td>
<td>density of access points, high false negatives</td>
</tr>
</tbody>
</table>

Table 1: Comparison of context tags

4.3 Combination of context tags
We have mentioned the need for combined context tags, which can decrease the possibility of errors and increase the cost of faking environment attacks. We need to develop an approach to ensure better performance of promising combination. The probable solution is machine learning. What we have to do is to feed the vector model with datasets and generate the suitable coefficients for each attributes.

4.4 Location Privacy
Location privacy is inherently protected in proximity-based authentication using context tags (except direct GPS). Communication between a RFID reader and a tag is encrypted with the shared key. Besides, the context vector in the message is the representation of transformed environmental attributes instead of direct location. So it’s very difficult to derive the actual location from such context vectors like audio fingerprints.

Nevertheless, location information is exposed to probable attackers in the direct usage of GPS. The attacker can eavesdrop the message when devices are communicating with satellites. In this sense, the actual location can be derived from the analysis of captured messages, and location privacy is undermined. So direct GPS should be avoided in the application with heavy concern about location privacy.

5. OUR EXPERIMENTS
In this section, we present our experiment on BlueProximity [1]. BlueProximity is an ubuntu application that enables the user to automate locking/unlocking his laptop screen. Figure ?? shows the preference setting panels. The user is required to bind a handset Bluetooth ID (i.e. Bluetooth MAC address) to his laptop in panel (a). The user can also customize the parameters (distance thresholds and confirmation durations for locking and unlocking respectively) in panel (b). When BlueProximity service is working on the background, the user with his handset (bound to laptop) can easily unlock his laptop by approaching it and lock his laptop by leaving it.

Here are the principles under the hood: the laptop periodically checks the received signal strength indicator (RSSI) of the handset, and estimates the distance. Once the distance is estimated larger (or smaller) than the distance threshold for the confirmation duration, the laptop status is set locked (or unlocked). Since there is no pairing between the handset and the laptop, BlueProximity is still not a ZIA application.

5.1 Experiment Design

Our Contextual Co-location Detection Project is designed to demonstrate that BlueProximity is vulnerable to attacks, even with pairing, and contextual co-location-based approach can defend BlueProximity against such attacks. Our experiment is planned in three phases: faking ID attack, relay attack, and contextual co-location-based authentication.

There are three principals in faking ID attack: Victim (a laptop with Ubuntu OS), Attacker (a laptop with Ubuntu OS), and Token (a handset with the Bluetooth module). We have to mention that Bluetooth ID (i.e. MAC address) can be faked using a third-party library "bdaddr". Since Victim and Token are working in unpaired mode, no shared key is established to guarantee the mutual trust between Token
and Victim. This is why proximity can be easily compromised by faking ID. The goal of faking ID attack experiment is to cheat BlueProximity and use Attacker to unlock Victim when Token is away, which can be realized by changing Attacker’s Bluetooth ID (i.e. Bluetooth MAC address) to Token’s. The model of attack is shown in Figure 7. After setting up BlueProximity on Victim, we bind Token’s MAC address to Victim. When Victim’s screen is locked, Attacker gets Token’s Bluetooth MAC address, and pretends to be Token by changing its Bluetooth MAC address. Thus Victim’s BlueProximity gets cheated and its screen is unlocked.

In the second phase of experiment, we plan to conduct “ghost-and-leech” attack to BlueProximity with pairing. As shown in Figure 8, we use two laptops as Leech and Ghost, who will play the roles as in the aforementioned RFID relay attack model. Token and Victim are defined as phase 1. Ghost acts as the fake Token to Victim, and Leech acts as the fake Victim to Token. The communication between Token and Victim is received and retransmitted by Ghost and Leech. BlueProximity should be upgraded by adding key establishment between Victim and Token beforehand. Since the challenge and response of checking Bluetooth ID are both encrypted with shared key, Leech and Ghost should only forward the encrypted messages without trying decryption. Thus the route via Leech and Ghost is established, and Ghost is considered as the RSSI source by Victim. Now we can unlock the screen of Victim by placing Ghost close to Victim.

In the third phase, we plan to design a protocol for the complete contextual co-location mechanism in BlueProximity authentication. Co-location is no longer proved by unilaterally measured RSSI, but by comparing the sensor-generated context vectors from Token and Victim. Besides, the pairing process (shared key establishment) is also needed to encrypt the communication. After key establishment, the first step in protocol is checking Bluetooth identity of Token. Then the second step is to generate context information from sensors equipped locally on both sides. The third step is that Token sends its context vector to Victim via a secure channel based on pairing. And the final step is to compare the context vectors in Victim to prove co-location. The authentication is achieved if the context vectors are matched. The general model as shown in Figure 9 will be tested against “ghost-and-leech” attacks in the further experiment, although in theory the protocol can defeat relay attacks.

5.2 Deployment and Results

Currently, we have finished the first phase of the experiment - faking ID attack. We prepared two laptops as Attacker and Victim, and a Nexus 7 tablet as Token. BlueProximity is available in Ubuntu Software Center. We tuned up the distance threshold and duration parameters so that fluctuation of locking/unlocking screen was restrained. Normally, the distance threshold was 8 for locking, and 3 for unlocking; the duration was 3-5 for both actions. BlueProximity worked well on my laptop.

After testing BlueProximity itself, we were to deploy Attacker in our scenario. We installed a bluetooth library “libbluetooth-dev”, and compiled a third-party library for changing MAC address “bdaddr”. We made a shell script “fakeid.sh” and deployed it on Attacker. When Token was away, we faked ID by executing the script: “sudo ./fakeid.sh xx:xx:xx:xx:xx:xx” where the address of Token was acquired...
from Bluetooth visible list. Then, Victim’s screen was unlocked successfully when placing Attacker close to Victim. In this sense, faking ID attack was done successfully.

We noticed that the victim can only recognize "Token" device by MAC address after faking, and the devices with identical MAC addresses are considered as one object by BlueProximity. So even when Token and Attacker were present within the threshold, only one was visible for Victim.

6. CONCLUSION

In this paper, we presented the contextual co-location-based authentication approach to defend ZIA from relay attacks. The co-location is proved by comparing the corresponding context tags derived from the sensing data of related environmental attributes, which are categorized as audio, light, GPS, acceleration, wireless broadcast traffic, and nearby device-IDs. According to the proof-of-concept experiments and evaluations, most of the context tag techniques in corresponding scenarios are sufficiently effective and robust to meet the requirements. Generally, audio, GPS, and wireless broadcast traffic are strong context tags for proving co-location, while light and nearby device-IDs are considered moderate. In this sense, the proposed solution utilizing context tags can well protect ZIA from relay attacks.

Besides, we also presented our experiment on BlueProximity, with the design for three phases and the results of Faking-ID attack. Faking-ID attack is conducted successfully since BlueProximity lacks pairing. And we will demonstrate the GPS-based approach of proving co-location in the next experiment.

7. REFERENCES