Wrong Siren! A Location Spoofing Attack on Indoor Positioning Systems: The Starbucks Case Study

Junsung Cho, Jaegwan Yu, Sanghak Oh, Jungwoo Ryoo, JaeSeung Song, and Hyoungshick Kim

Thanks to indoor proximity technologies, it is possible to introduce location-based smart services to customers, for example, transmitting identifiable signals that represent the locations of stores. The authors investigate a potential security risk involved in such technologies: physical signals used as identifiers can be captured and forged easily with today's widely available IoT software for implementing location spoofing attacks.

ABSTRACT

The Internet of Things interconnects a mass of billions devices, from smartphones to cars, to provide convenient services to people. This gives immediate access to various data about the objects and the environmental context — leading to smart services and increased efficiency. A number of retail stores have started to adopt IoT enabled services to attract customers. In particular, thanks to indoor proximity technologies, it is possible to introduce location-based smart services to customers, for example, transmitting identifiable signals that represent the locations of stores. In this article, we investigate a potential security risk involved in such technologies: physical signals used as identifiers can be captured and forged easily with today's widely available IoT software for implementing location spoofing attacks. We highlight this security risk by providing a case study: an in-depth security analysis of the recently launched Starbucks service called `Siren Order`.

INTRODUCTION

Tracking the physical locations of objects (e.g., a user's smartphone) could be applied to the Internet of Things (IoT) to make them more convenient and attractive to users. There are many practical applications utilizing the geographical locations of things; some applications allow customers to locate various points of interests (POIs) including retail stores, tourist attractions, public transportation stations, and so on; other applications focus on marketing and help vendors push advertisements to potential clients when they are within a specific range of a geographic location.

For example, in order to help their customers avoid queues, Starbucks Korea recently introduced a mobile pre-ordering and payment service called `Siren Order`. This service allows customers to remotely place their orders and pay in advance for those orders using their smartphones without contacting a cashier at a Starbucks store. For this service, a customer's Starbucks app needs to identify the particular Starbucks store where the customer wants to pick up the order. Unfortunately, GPS does not often work well for this scenario when the customer is already inside a building (i.e., the Starbucks store). Therefore, an indoor positioning system can alternatively be used for this kind of pre-ordering/payment service.

A large number of available sensors built into a thing (e.g., smartphone) — RF technology such as Wi-Fi, Bluetooth, and RFID, ultrasound, GPS, infrared, and magnetic fields — can be used for tracking people and objects within a geographical space [1]. For instance, IndoorAtlas (http://www.indooratlas.com, accessed 10 October 2016) uses magnetic technology, Wi-Fi, and Bluetooth to provide an indoor positioning service. Skyhook (http://www.skyhookwireless.com, accessed 10 October 2016) uses GPS and Wi-Fi to deploy geofences. Recently, Broadcom (http://www.broadcom.com, accessed 10 October 2016) developed an indoor positioning technology using fifth generation (5G) Wi-Fi (802.11ac).

Despite the benefits of indoor positioning systems for both customers and retailers, this technology may pose serious security and privacy threats. Several studies [2, 3] demonstrated that indoor positioning systems might be vulnerable to location spoofing attacks at the physical layer. Tippenhauer *et al.* [4] particularly introduced several kinds of attacks targeted at WLAN-based positioning systems through the security analysis of a WLAN-based positioning system such as Skyhook. They showed that Skyhook is vulnerable to location spoofing attacks by jamming and replaying localization signals to deceive WLAN clients into believing that they are at a position which is different from their actual physical position, and suggested some mitigation techniques (e.g., using the unique characteristics of access points).

In this article, we demonstrate that a different type of indoor positioning system using high-frequency audio signals can also be vulnerable to similar location spoofing attacks, through a deep analysis of the `Siren Order` service in Starbucks stores. We found that an attacker can easily record the unique audio signal used for identifying a Starbucks store, and then broadcast that signal in another store to deceive victims into placing their orders at the place where the attacker is located. Therefore, the item being ordered can be intercepted by an attacker. Such attacks might, in turn, negatively influence customers'

Digital Object Identifier:
10.1109/MCOM.2017.1600595CM

Junsung Cho, Jaegwan Yu, Sanghak Oh, and Hyoungshick Kim are with Sungkyunkwan University; Jungwoo Ryoo is with Pennsylvania State University; JaeSeung Song is with Sejong University.

0163-6804/17/$25.00 © 2017 IEEE

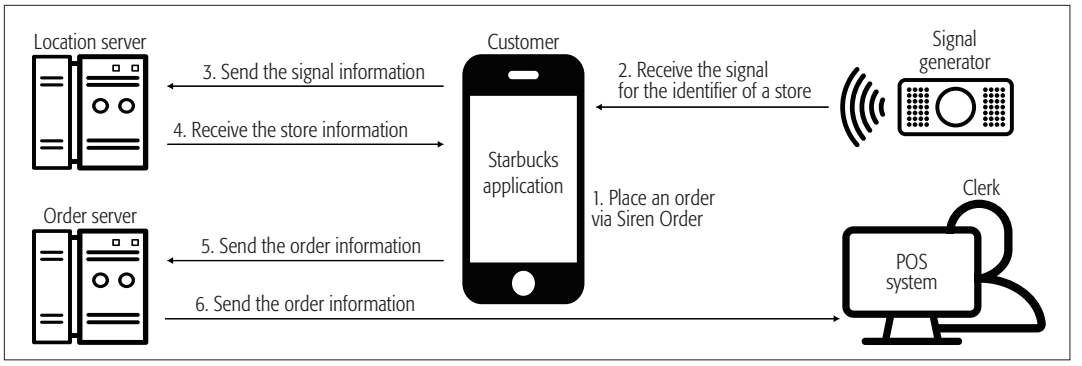

Figure 1. Overview of the process of `Siren Order`.

attitude and behavior toward indoor positioning systems and may seriously damage the reputation of the company using the system. We demonstrated the feasibility of a successful attack exploiting the real-world service called `Siren Order`. This implies that many real-world indoor positioning systems might be badly designed without considering security threats at the physical layer. To improve the status quo, we suggest practical ways to address such vulnerabilities.

The remainder of this article presents our in-depth security analysis and discusses `Siren Order`. We first explain how `Siren Order` works in detail, and then discuss the feasibility of a location spoofing attack against that service.

What Is Siren Order?

Starbucks Korea launched a new mobile pre-ordering service, called `Siren Order`, with the Starbucks mobile app, which has been made available for both iOS and Android platforms. The goal of this service is to allow customers to order in advance, saving them waiting time before picking up their order at a store location. Unlike `Mobile Order & Pay`, which was launched in the United States, using smartphones' GPS functionality to identify the Starbucks store nearest to a customer's location, an indoor positioning system is used to implement the `Siren Order` service. Even when a customer inside a Starbucks store tries to place an order through the Starbucks app, the `Siren Order` service (i.e., the Starbucks mobile app) can identify in which Starbucks store the customer placed the order.

For the `Siren Order` service, high-frequency audio signals that are mostly inaudible to human ears have been used. This technology has some benefits compared to conventional RF-based indoor positioning systems. In general, audio signals are easily absorbed into walls. That is, user locations can be determined at room-level precision with high accuracy because those signals cannot pass through walls or windows. This is very useful to precisely identify in which store the customer is actually located.

Figure 1 shows the overall process of `Siren Order`. The `Siren Order` system consists of five components: a customer's Starbucks app, location server, order server, point-of-sale (POS) system, and signal generator. A typical use of this system would be as follows:

1. A customer places an order via the Starbucks app and pays for the selected item.
2. The app turns on the microphone in the customer's smartphone and then records the audio signals, which come from the signal generator installed in a Starbucks store (see an example in Fig. 2).
3. When the recording ends, the app analyzes the captured audio signal and submits a query with the signal data to the location server.
4. After receiving that query, the location server finds the Starbucks store matched with the signal data, and sends the Starbucks store information to the Starbucks app.
5. After receiving the query response, the Starbucks app sends the order information to the order server.
6. Finally, this order information is processed at the order server and relayed to the POS system at the Starbucks store for placing the order to the cashier at the store.

Figure 2. Signal generator.

We collected audio signals from four different Starbucks stores and found that the audio signals used in `Siren Order` typically range from 18 to 20 kHz, which humans cannot hear. The collected audio signals have uniquely different periodic patterns, although all patterns are commonly repeated every 1.25 s (i.e., five time units). Figure 3 shows one of the audio signals recorded in a Starbucks store. As shown in Fig. 3, one period of the signal is composed of two parts — start flag (the first time unit) and store ID (the remaining time units).

In general, audio signals are easily absorbed into walls. That is, user locations can be determined with room-level precision with high accuracy because those signals cannot pass through walls or windows. This is very useful to precisely identify which store the customer is actually located at.

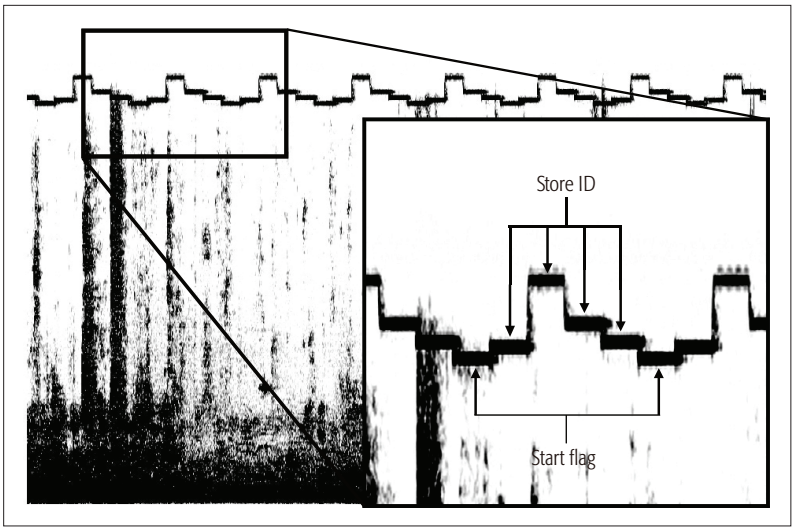

Figure 3. A recorded audio signal in a Starbucks store.

Implementation of a Location Spoofing Attack

We describe our implementation of a location spoofing attack against `Siren Order`. As mentioned earlier, a signal generator at a Starbucks store continuously emits a unique audio signal to represent the store's identifier. The goal of our attack is to deceive a victim's Starbucks app at store S_1 into believing that the app is at store S_2 in which an attacker is located. When an order is placed at S_2 instead of S_1, the attacker can illegally intercept the item that the victim ordered in store S_1. Therefore, such attack attempts will inevitably harm the reputation of Starbucks since the attacker can control customers' orders freely and/or disrupt the whole service.

Figure 4 illustrates an overview of our attack. In our attack, there are two attackers: attacker A_1 in store S_1 and attacker A_2 in store S_2. A_2 has recorded the signal transmitted from S_2, and delivers it to A_1 via any communication channel. After receiving the signal from A_2, A_1 broadcasts it again (i.e., by playing the recorded signal through an audio player) to its neighbors (i.e., potential victims) in S_1. To succeed in this attack, a victim's device must receive A_1's signal instead of the authentic signal transmitted from S_1's signal generator. This can be achieved simply by jamming at the physical layer (e.g., loudly playing the signal to represent S_2's identifier). If A_1's signal is more powerful than the signal from the transmitter at S_1, the attacker can interfere and overpower the signal from S_1. As a result, a victim's Starbucks app in S_1 receives the attacker's signal representing S_2's identifier and unknowingly transmits that signal to the location server with which the Starbucks app communicates. Thereafter, the location server finds the store information about S_2 in response to the received signal and replies to the victim's Starbucks app; the Starbucks app blindly believes that it is in S_2. Therefore, if the victim places an order through her Starbucks app, this order is processed at S_2 in spite of the user's original intent (to place the order at S_1) in which attacker A_2 is located. This is a typical scenario for our location spoofing attack.

As a proof of concept, we performed a location spoofing attack on real Starbucks stores. In our implementation, we used QuickTime Player (https://support.apple.com/kb/PH22585, accessed 10 October 2016) for recording signals and Adobe Audition CC (http://www.adobe.com/products/audition.html, accessed 10 October 2016) for filtering out unnecessary signals, which are widely affordable and popular.

In our experiment, we first recorded a signal in Starbucks store A and then applied a bandpass filter (in Adobe Audition CC) between 18 and 20 kHz to the recorded signal data to isolate the high-frequency part, which is a typical range used for `Siren Order`. In another Starbucks store, B, two participants were recruited to play the roles of "victim" and "attacker," respectively. The attacker simply amplified the audio signal (previously recorded at store A) and broadcasted it to overpower the signal data emitted from store B's generator. When the victim was located around the attacker (e.g., within about 3 m), the victim's Starbucks app believed that the victim was in store B. Finally, we confirmed that location spoofing attacks can be successfully performed in real-world settings when the victim tried to place an order through his Starbucks app; his order was inappropriately placed at store B, although he was in store A (our demonstration video clip is available at https://youtu.be/oN9kB169lvE, accessed 10 October 2016).

The main goal of this experiment is not to damage Starbucks' business or reputation. We conducted this experiment to show the feasibility of location spoofing attacks on new indoor positioning systems through a case study. We already reported the discovered problem to the Starbucks developers and suggested a fix based on our observations.

Countermeasures

How can we fix this problem in indoor positioning systems? In this section, we discuss some possible mitigation techniques against such attacks.

Freshness of Audio Signals

Location spoofing is basically a kind of *replay attack*. Therefore, we need to verify the freshness of messages to prevent location spoofing attacks. A number of distance-bounding protocols have already been proposed for this purpose. Brands and Chaum [5] proposed the first distance-bounding protocol against a type of replay attack called Mafia fraud [6]. Hancke and Kuhn [7] also proposed a distance-bounding protocol against a terrorist fraud [6], which was a modified version of Mafia fraud. Furthermore, Reid *et al.* [8] proposed an advanced distance-bounding protocol based on a symmetric key cryptosystem, taking advantage of the security strengths of Brands' and Chaum's protocol and the efficiency of Hancke's and Kuhn's protocol. However, those distance-bounding protocols are not suitable for the indoor positioning system in `Siren Order` where one-way communication from a signal generator to a Starbucks app is only allowed because in the aforementioned protocols, challenge-response message pairs should be repeatedly exchanged to obtain meaningful statistical information about the physical distance between the sender and the recipient. To over-

come this limitation in our application, we present a distance-bounding protocol based on a synchronized timestamp.

Our main idea is to include a timestamp in the signals used for an indoor positioning system to limit the lifetime of recorded signals. We briefly describe this with the following notation. In a protocol that is used by S_1 and S_2, "$S_1 \rightarrow S_2$: x" implies that S_1 sends message x to S_2. The symbols G, a, and S represent the signal generator, Starbucks app, and Starbucks server, respectively. E is a symmetric encryption algorithm (e.g., AES). $k_{S_1S_2}$ is a secret symmetric session key to be shared by two parties S_1 and S_2. For data input x, $E_k(x)$ denotes the data value resulting from E's encryption operation on x using the encryption key k. t_P is a timestamp generated by a party P. id_G is a signal to identify a signal generator G installed at a Starbucks store. Notation $||$ denotes the concatenation operation. We assume that an encryption key k_{GS} is securely shared between G and S, and G and a have a synchronized time clock that can be maintained via coordinated universal time (UTC). A reliable connection to the Internet is needed for G and a to use a clock synchronization mechanism on the Internet. This assumption could be acceptable because it is expected that most sensor devices such as G would be connected to the Internet in the near future.

Unlike the existing system, in our proposed protocol, G generates its timestamp t_G and broadcasts the encrypted signal $E_{k_{GS}}(id_G||t_G)$ instead of the plaintext signal id_G in its Starbucks store as follows:

$$G \rightarrow A: \quad E_{k_{GS}}(id_G||t_G)$$

After receiving $E_{k_{GS}}(id_G||t_G)$ from G, a immediately generates its own timestamp t_A and then relays $E_{k_{GS}}(id_G||t_G)$ with the generated t_A to S. We assume that the communication channel between G and S is securely protected. This assumption is practical and reasonable because G and S communicate via the Internet against an attacker who can eavesdrop any wireless signals in the Starbucks store.

$$A \rightarrow S: \quad E_{k_{GS}}(id_G||t_G) \;||\; t_A$$

After receiving $E_{k_{GS}}(id_G||t_G) \;||\; t_A$ from a, S decrypts the encrypted part $E_{k_{GS}}(id_G||t_G)$ only with the shared key k_{GS} and verifies its freshness. For the verification, S calculates the time difference between t_G and t_A. If the difference is less than a pre-determined threshold δ, the received query message is accepted, and the corresponding Starbucks store information is sent to a; otherwise, this query is rejected. If the Starbucks customer relays an outdated message $E_{k_{GS}}(id_G||t_G)$ (which has been replayed by a location spoofing attack) to S, the time difference between t_G and t_A would be quite large.

Suffice it to say that it is important to choose a proper δ to make location spoofing attacks difficult while guaranteeing a low false alarm rate for legitimate customers. We claim that a considerable amount of processing time will be required to perform a location spoofing attack in this scenario. If an attacker tries a location spoofing attack, the attacker's timestamp can be approximately calculated as follows:

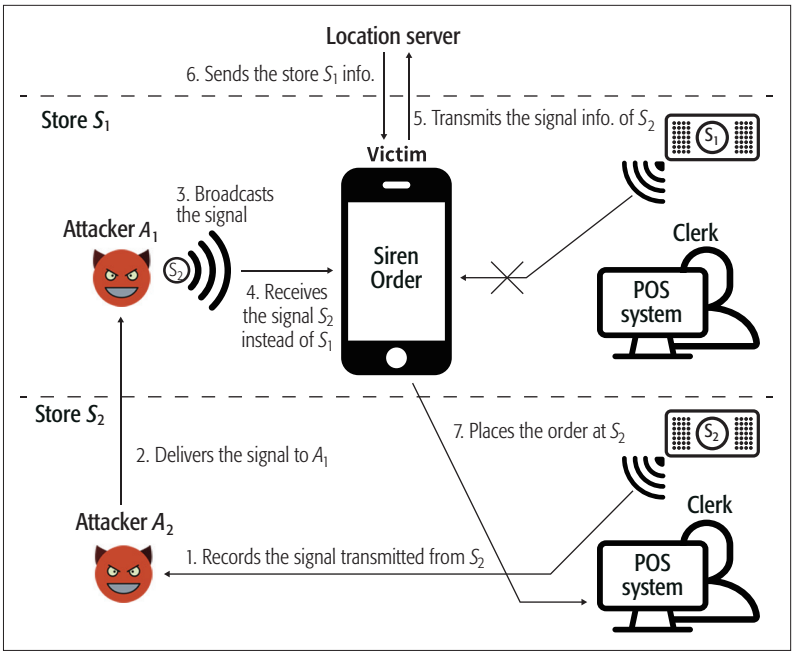

Figure 4. Overview of the location spoofing attack on `Siren Order`.

In this equation, t_{sound_1} is the amount of time taken from a signal generator to an attacker's recording device; t_{record} is the amount of time taken for recording the audio signal in a digital format; $t_{internet}$ is the amount of time taken to deliver a recorded signal from an attacker A_2 in store S_2 to another attacker A_1 in store S_1; and t_{sound_2} is the amount of time from an attacker's audio player to a victim's Starbucks app. Note that t_A can also be represented as $t_G + t_{sound_1}$, which might be significantly less than t_{attack}. To prevent location spoofing attacks, we need to find a proper threshold δ that satisfies the following equation. To simplify the equation, we assume that t_{sound_1} is equal to t_{sound_2} as follows:

$$t_{sound} < \delta < 2 \cdot t_{sound} + t_{record} + t_{internet}$$

Now suppose that the distance from a signal generator to a customer's smartphone is 10 m. In this case, if we assume that the speed of sound is 343.2 m/s, t_{sound} can approximately be calculated to be roughly 29.1 ms. To show that there is a practically reasonable δ for the proposed mitigation technique in a real-world situation (i.e., $2 \cdot t_{sound} + t_{record} + t_{internet} \gg 29.1$ ms), we conducted a simple experiment with two laptops with a non-congested 100 Mb/s Wi-Fi connection to a LAN connected to the Internet via a Gigabit-speed link. The first and second laptops were used to simulate attackers A_1 and A_2, respectively, in Fig. 4. We used an audio streaming application named Nicecast to efficiently deliver the recorded audio signal from the first laptop to the second laptop. We recorded the input sound stream and receiver's output sound stream synchronously. A short audio signal was generated and delivered to simulate a location spoofing attack. After receiving the sound signal, the second laptop produced the same sound signal from its speaker. We measured the total processing time for those steps to approximately measure $2 \cdot t_{sound} + t_{record} + t_{internet}$.

In order to deploy our mitigation methods in such existing IoT platforms, a platform has to support at least two features: location and security. As these widely used IoT platforms support location and security functions, our mitigation methods can easily be integrated into existing IoT platforms.

We repeated this 20 times to obtain statistically meaningful results. The mean time spent on each simulation was 2.1 s, ranging from 1.9 s to 2.9 s, which implies that there is a significant gap between t_{sound} (29.1 ms) and $2 \cdot t_{sound} + t_{record} + t_{internet}$ (2.1 s). Therefore, in practice, we can find a reasonable δ to mitigate location spoofing attacks.

However, efficient and accurate time synchronization is not easy in the real world. For example, Network Time Protocol (NTP) [9] provides limited accuracy because the packet propagation delay varies depending on network conditions. Fortunately, our experimental results (2.1 s vs. 29.1 ms) show that the proposed method does not require a highly accurate time synchronization model. An inaccuracy of a few milliseconds, which could be incurred by NTP, seems well tolerated in the proposed solution.

Transaction Authentication

The main problem, or the reason for this attack, is the absence of a verification process when an order is picked up. We can simply fix this problem by introducing an additional procedure for transaction authentication. That is, we require that a customer provides a proof of transaction before picking up an order. It is a secure way to authenticate whether someone who is trying to pick up the order is the legitimate customer of the order being placed.

For example, when a customer places an order via `Siren Order`, the customer's Starbucks app can generate a 4-digit random number as a one-time password and send it to a clerk through the `Siren Order` service. This number is then required to pick up the order for the purpose of verifying the customer who placed the order. This technique helps protect the customer's order against an attacker who wants to steal the ordered product. It is very difficult for an attacker to obtain the randomly generated number, although capturing any signals in the air is possible. Without modifying the existing system, this verification procedure might be added with a software patch to the Starbucks app. However, it is likely to degrade the usability of the `Siren Order` service as customers and clerks should check the validity of the generated random number for each order. Therefore, we need to conduct a user study to investigate the usability of this newly proposed procedure.

Conclusion

In recent years, indoor positioning systems are gaining popularity in the market to provide the location information of people and devices in a building. Several different types of technologies have been introduced, but their security issues have not been explored thoroughly.

In this article, we point out a security risk called *location spoofing* associated with indoor positioning systems by providing a proof-of-concept case study that implements a well designed location spoofing attack against the Starbucks pre-order service called `Siren Order`, which can cause severe disruption in the service. To mitigate such attacks, we discuss two possible mitigation strategies.

There are many IoT platforms, for example, Mobius based on oneM2M global IoT standards [10] and IoTivity open source platform based on OCF (https://openconnectivity.org, accessed 10 October 2016). In order to deploy our mitigation methods into such existing IoT platforms, a platform has to support at least two features: location and security. As these widely used IoT platforms support location and security functions, our mitigation methods can easily be integrated into existing IoT platforms.

As part of our future work, we plan to implement the proposed mitigation techniques and further investigate the performance and usability of those solutions by conducting user studies.

Acknowledgments

This work was supported in part by the NRF Korea (No. 2014R1A1A1003707), the ITRC (IITP-2016-R0992-16-1006), and ICT R&D program (No. B0717-16-0116 , No. B0184-15-1001). The authors would like to thank all the anonymous reviewers for their valuable feedback.

References

[1] Y. Gu, A. Lo, and I. Niemegeers, "A Survey of Indoor Positioning Systems for Wireless Personal Networks," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 1, 2009, pp. 13–32.
[2] L. Lazos, R. Poovendran, and S. Capkun, "ROPE: Robust Position Estimation in Wireless Sensor Networks," *Proc. 4th Int'l. Symp. Info. Processing in Sensor Networks*, 2005.
[3] S. Capkun and J. Hubaux, "Secure Positioning of Wireless Devices with Application to Sensor Networks," *Proc. 24th Annual Conf. IEEE Comp. Commun. Societies*, 2005.
[4] N. O. Tippenhauer *et al.*, "Attacks on Public WLAN-Based Positioning Systems," *Proc. 7th Int'l. Conf. Mobile Systems, Applications, and Services*, 2009.
[5] S. Brands and D. Chaum, "Distance-Bounding Protocols," *Proc. Wksp. Theory and Application of of Cryptographic Techniques*, 1993.
[6] Y. Desmedt, "Major Security Problems with the 'Unforgeable' (feige)-fiat-shamir proofs of Identity and How to Overcome Them," *SecuriCom*, 1988.
[7] G. P. Hancke and M. G. Kuhn, "An RFID Distance Bounding Protocol," *Proc. 1st Int'l. Conf. Security and Privacy for Emerging Areas in Commun. Networks*, 2005.
[8] J. Reid *et al.*, "Detecting Relay Attacks with Timing-Based Protocols," *Proc. 2nd ACM Symp. Info., Comp. and Commun. Security*, 2007.
[9] D. Mills *et al.*, "RFC 5905: Network Time Protocol version 4: Protocol and Algorithms Specification," IETF tech. rep., 2010.
[10] J. Swetina *et al.*, "Toward a Standardized Common M2M Service Layer Platform: Introduction to oneM2M," *IEEE Wireless Commun.*, vol. 21, no. 3, June 2014, pp. 20–26.

Biographies

Junsung Cho (js.cho@skku.edu) received his B.S. degree from the Department of Computer Engineering, Korea University of Technology and Education, in 2014. He is currently a graduate student with the Department of Computer Science and Engineering, Sungkyunkwan University, Korea, supervised by Hyoungshick Kim. His current research interests include usable security, mobile security, IoT security, and security engineering.

Jaegwan Yu (jaegwan@skku.edu) received his B.S. degree from the Department of Electrical and Information Engineering, Korea University, in 2015. He is currently a graduate student with the Department of Platform Software, Sungkyunkwan University, supervised by Hyoungshick Kim. His current research interests include network security, software security, and security engineering.

Sanghak Oh (osh09@skku.edu) received his B.S. degree from the Department of Software, Sungkyunkwan University, in 2015. He is currently a graduate student with the Department of Platform Software, Sungkyunkwan University, supervised by Hyoungshick Kim. His current research interests include network security, software security, and security engineering.

Jungwoo Ryoo [M] (jryoo@psu.edu) is a professor of information sciences and technology at Pennsylvania State University.

His research interests include information security and assurance, software engineering, and computer networking. He received a Ph.D. in computer science from the University of Kansas.

JAESEUNG SONG (jssong@sejong.ac.kr) is an assistant professor in the Computer and Information Security Department at Sejong University. He holds the position of oneM2M Test Working Group Chair. Prior to his current position, he worked for NEC Europe Ltd. and LG Electronics in various positions. He received a Ph.D. from Imperial College London in the Department of Computing, United Kingdom. He holds B.S. and M.S. degrees in computer science from Sogang University. He is a member of IEEE.

HYOUNGSHICK KIM (hyoung@skku.edu) received his B.S. degree from the Department of Information Engineering, Sungkyunkwan University, his M.S. degree from the Department of Computer Science, Korea Advanced Institute of Science and Technology, Daejeon, and his Ph.D. degree from the Computer Laboratory, University of Cambridge, United Kingdom, in 1999, 2001, and 2012, respectively. He is currently an assistant professor with the Department of Software, Sungkyunkwan University. His current research interests include usable security and security engineering.