A Privacy-Aware Continuous Authentication Scheme for Proximity-Based Access Control

Isaac Agudo, Ruben Rios, Javier Lopez

Network Information and Computer Security (NICS) Lab,
University of Malaga, 29071, Spain

Abstract

Continuous authentication is mainly associated with the use of biometrics to guarantee that a resource is being accessed by the same user throughout the usage period. Wireless devices can also serve as a supporting technology for continuous authentication or even as a complete alternative to biometrics when accessing proximity-based services.

In this paper we present the implementation of a secure, non-invasive continuous authentication scheme supported by the use of Wearable Wireless Devices (WWD), which allow users to gain access to proximity-based services while preserving their privacy. Additionally we devise an improved scheme that circumvents some of the limitations of our implementation.

Keywords: Wearable Wireless Devices, Continuous Authentication, Proximity-based Services, Privacy

1. Introduction

Wearable wireless devices (WWD) are small personal devices with a number of integrated sensors that a user might wear or carry in order to help him/her monitor a wide variety of activities or medical conditions. These wireless-enabled devices allow the information collected to be shared with other devices nearby which are responsible for processing and/or presenting these data in a human-readable way. Therefore, it is not surprising that

Email addresses: isaac@lcc.uma.es (Isaac Agudo), ruben@lcc.uma.es (Ruben Rios), jlm@lcc.uma.es (Javier Lopez)

WWDs are becoming a prevalent technology in e-health (e.g., chronic disease monitoring and elderly assistance) and fitness (e.g., pedometers and hearth-rate monitors) scenarios. This is, however, only the tip of the iceberg compared to the envisioned new application scenarios, as suggested by ABI research (2012) and IMS research (2012).

As these devices become smaller and less costly they can be attached to clothes or even implanted thereby directly associated with an individual, which entails serious privacy risks. For example, Saponas et al. (2007) demonstrated that it is possible to identify and track users wearing Nike+ sensors[1] using a simple device which costs less than 30 euros. The origin of the problem is that these devices are continuously broadcasting a unique identifier which can be easily linked to the real user by a third party. The risk is further exacerbated when the devices are implanted inside the individual, as recently shown by Gollakota et al. (2011), because they cannot be disconnected at will.

But these problems are not new, Di Pietro and Mancini (2003) concluded that the main functionalities of WWDs (i.e. discovery, advertising, and service provisioning) should prevent the use of unique identifiers to protect users privacy but that it was still necessary to find the right balance between privacy and security because a fully anonymous service may encourage users to misbehave. Other wireless technologies have also presented similar privacy problems: Denis Foo Kune and Kim (2012) show that by listening to unencrypted broadcast messages from cellular towers it is possible to leak the location of subscribers. Moreover, Bluetooth (Zafeiropoulos et al., 2010; Hay and Harle, 2009), Wi-Fi (Kao et al., 2010), and RFID (Sadeghi et al., 2009) technologies have been extensively used for tracking purposes.

Although the existence of privacy risks may hinder the adoption of WWDs and detract from all its benefits if not carefully considered, the information collected by these devices might also be exploited for the creation of advanced authentication systems by incorporating them as an additional level of assurance to the process. In particular, proximity-based access control (PBAC) systems could benefit tremendously from the application of WWDs because these wireless-enabled personal devices can serve as an authentication token proving the presence of the individual in a particular location and thus allow for the straightforward and seamless provision of proximity-based

[1] http://nikeplus.nike.com/plus/products

services. Some examples of PBAC services are automatic log-in and log-out from computers, providing access to restricted facilities, and so forth. For PBAC systems to work in practice, users need to be continuously authenticated when moving from one protected resource to another.

The main contribution of this work is the design and implementation of a privacy-aware continuous authentication scheme and architecture for the delivery of proximity-based services. The devised scheme is supported by the use of WWDs that unobtrusively interact with the infrastructure on behalf of the user. In addition we present the design of a more sophisticated solution that overcomes some of the limitations of our prototype implementation.

The rest of this paper is organised as follows. Section 2 describes the main features of proximity-based access control systems by reviewing some previous works in the area. Next, Section 3 presents the architectural components of our system and their relationships. This section also defines various types of adversarial models as well as the security and privacy threats that they introduce. In Section 4 we presents a number of mechanisms used by the WWD platform used in our prototype in order to diminish the threats described in the previous section. Subsequently, Section 5 provides the implementation details of our prototype and discusses some limitations. An improved version of the prototype implementation is described in Section 6. Finally, Section 7 provides some concluding remarks and discusses possible lines of future work .

2. Proximity-Based Access Control

It is difficult to date the first proposal using spatial constrains for access control. We could say that the Spatial Role-based Access Control (SRBAC) model by Hansen and Oleshchuk (2003) is one of the first to extend the RBAC model to include location information in security policy definitions but many proposals followed after that in a short period of time. In this proposal, the authors highlight the importance of providing secure and trusted location data but relegated this function to a “trusted underlying network infrastructure”. GEO-RBAC by Damiani et al. (2007) is another extended RBAC model where roles are activated based on the position of the user. LoT-RBAC by Chandran and Joshi (2005) copes with both location and time constrains in the authorisation process. Those proposals focused on the formal model for authorisation leaving the details on the localisation infrastructure out of the scope.

There are some other proposals that include some prototype or that directly focus on the localisation problem. Proximity-Based Access Control (PBAC) by Gupta et al. (2006) define a formal access control model based on proximity for Smart-Emergency Departments and present a prototype using a proprietary solution based on Ultra-Wide Band RFID. In (Cruz et al., 2008), a location-aware role-based and attribute-based access control system is introduced where authors implement a prototype using the Google Api. In (Kirkpatrick et al., 2011) the authors define the syntax and semantics of Prox-RBAC as well as the protocols and algorithms for enforcing Prox-RBAC policies based on usage control logics. They implemented a prototype using NFC.

Those proposals require the use of a wearable or handled wireless device that is used to check user location continuously to be able to dynamically change user permissions. Each time the user location is updated the WWD has to authenticate against the location verifier, ideally without any or very little human intervention. When using UHF RFID, as in the prototype by Gupta et al. (2006), there no needed intervention, whereas using NFC requires that users place the WWD close to the reader. Each approach has their benefits, while requiring no intervention from the user may feel more user friendly, it may cause unsolicited access to some services as users do not become aware of the interactions between their WWDs and the infrastructure. On the other hand, requiring users to react every time their device interact with the infrastructure might discourage the use of the WWDs.

User location can be determined in many ways. Usually the process is divided in two phases. During the observation phase, which consists of measuring some particular features of the signals (i.e., received signal strength (RSS), angle of arrival of the signal (AoA), and the time of arrival (ToA) (Vossiek et al., 2003)), it is possible to determine the distance from the device to a (set of) reference points (i.e., anchor). After the observation comes the determination of the position, which can be calculated by the device itself (i.e., user-driven) or by a positioning infrastructure (i.e., infrastructure-driven) (Ferreres et al., 2008).

This classification has interesting connotations from the point of view of security and privacy. The user-driven approach is more privacy-friendly because the infrastructure only gets location information when device decides so and to the extent desired by the device. On the downside, users might act in a malicious way and try to fool the system in order to gain access to resources (Capkun and Hubaux, 2006). An infrastructure-based approach is more resilient to attacks but the location information is no longer under the control of the user. Also, an external adversary might try to impersonate the positioning system or simply observe the communications in order to track the users without their explicit consent.

For PBAC scenarios, an infrastructure-based approach appears to be more suitable from a security perspective but it is necessary to deal with some of the privacy issues arising from its use. Moreover, in our application scenario we do not need to know the precise position of the users but that their are in the vicinity of a resource.

3. Problem Statement

This section presents the main assumptions applicable to the rest of this work. In particular it describes the architecture of our system and the adversarial models under consideration.

3.1. System Architecture

The devised system consists of various architectural components, some of which are necessarily separated hardware but others might be logically integrated into the same physical component. An illustration of the elements of the system as well as well as the relationships among them is provided in Figure 1.

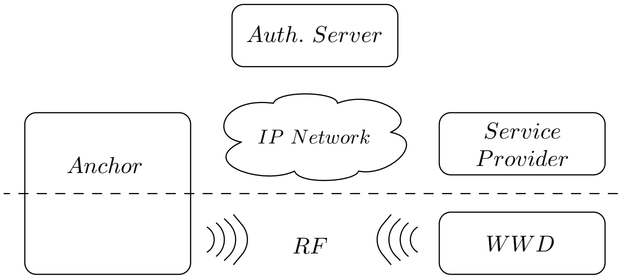

Figure 1: System Architectural Components

Anchors are wireless-enabled devices in charge of detecting WWDs nearby. They also serve as a communication proxy for WWDs, establishing an indirect link between the authentication server and the WWDs. Therefore, the anchors in our system are composed of an RF element and an IP element. The communication between anchors and WWDs takes place in an RF band without any intermediaries, while the communication with the Authentication Server is done by means of an IP network, potentially over the Internet, using a mutually authenticated SSL channel.

The **Authentication Server** is used to authenticate the presence of the users of the system in a particular location, i.e. in the vicinity of a particular anchor. To that end, the Authentication Server associates the information received about a WWDs to the anchors that forwarded it. The functionality of this component might be executed as a standalone service or it might be integrated with the anchor. The first option is more scalable but the integrated version might be useful in settings where the number of resources to be controlled is rather limited and not geographically disperse. In this work we consider the standalone version.

The **Service Provider** offer services to users based on their proximity to particular anchors. Service providers receive events from the Authentication Server regarding their anchors of interest. Events are triggered whenever an user gets in or out of range from a particular anchor. In a simplistic scenario the service provider could integrate an RF antenna and implement a dual functionality as anchor and service provider.

Finally, the **WWDs** involved in the system have a pre-established security association with the authentication server. Additionally, every WWD may create confidential channels with Anchors. The WWDs authenticates against the authentication server via the anchors in their proximity.

3.2. Adversarial Models

The robustness of the system will be determined by the capabilities of the adversarial model under consideration. In general we assume computationally bounded adversaries which cannot break the cryptographic algorithms used by the elements of the system to protect their communications.

In the following we provide an informal definition of the various types of adversarial models that may attempt to attack some of the elements of our system.

Definition 1 (The $\mathcal{ADV}_{\mathcal{HBC}}$ model)**.** *An honest-but-curious adversary ($\mathcal{ADV}_{\mathcal{HBC}}$) is an attacker that does not deviate from the protocol specification by modifying, fabricating or deleting messages from the elements of the system. The actions that are permitted for this type of adversary are:*

- *Store the messages exchanged by the elements of the system, and*

- *Analyse the communication flows.*

An $\mathcal{ADV}_{\mathcal{HBC}}$ is often concerned with being detected while attacking the system. This type of adversary tries to infer any information from the mere observation of the communications and thereby it is usually referred to a passive attacker. Also, we assume $\mathcal{ADV}_{\mathcal{HBC}}$ are external, that is, they are in possession of no cryptographic material to decrypt messages.

Definition 2 (The $\mathcal{ADV}_{\mathcal{MAL}}$ model)**.** *A malicious adversary ($\mathcal{ADV}_{\mathcal{MAL}}$) is an attacker that tries to actively disrupt the normal operation of the protocol. Besides storing and analysing the communications, an $\mathcal{ADV}_{\mathcal{MAL}}$ can perform the following actions:*

- *Fabricate (fake or inconsistent) messages,*
- *Replay previously stored messages, and*
- *Perturb some properties of the wireless signal (i.e., amplification, attenuation, block or delay).*

The strategy of an $\mathcal{ADV}_{\mathcal{MAL}}$ is to deviate from the protocol specification in order to find an attack vector. Therefore, $\mathcal{ADV}_{\mathcal{MAL}}$ can also be referred to as active attackers. Active attacks are usually launched by external adversaries which are not worried of being caught cheating.

Definition 3 (The $\mathcal{ADV}_{\mathcal{DIS}}$ model)**.** *A dishonest adversary ($\mathcal{ADV}_{\mathcal{DIS}}$) is a legitimate user of the system who is occasionally interested in bypassing some of the policies of the system.*

An $\mathcal{ADV}_{\mathcal{DIS}}$ is basically a legitimate user who is capable of completing the authentication protocol but he can occasionally behave as a malicious user ($\mathcal{ADV}_{\mathcal{MAL}}$). This type of adversary is not interesting in interfering with the operation of the system.

Definition 4 (The $\mathcal{ADV}_{\mathcal{CAP}}$ model)**.** *The $\mathcal{ADV}_{\mathcal{CAP}}$ is an attacker that compromises some of the elements of the system in order to gain access to the cryptographic material contained in them.*

We make a distinction between $\mathcal{ADV}_{\mathcal{DIS}}$ and $\mathcal{ADV}_{\mathcal{CAP}}$ even though both are internal attackers. However, the first adversarial model is only interested in fooling the system but the second model is more powerful. In fact, we consider that an $\mathcal{ADV}_{\mathcal{CAP}}$ can compromise not only WWDs but also Anchors. After compromising any of these devices he can retrieve any information intended for the captured device. Also, this type of attacker can perform passive or active attacks to other elements of the system.

Definition 5 (The $\mathcal{ADV}_{\mathcal{COL}}$ model)**.** *A colluders adversarial model ($\mathcal{ADV}_{\mathcal{COL}}$) consists of a group of adversaries that cooperate in order to attack the system.*

The $\mathcal{ADV}_{\mathcal{COL}}$ model may consist of any number of aforementioned adversaries. Here, we concentrate on a special case of collusion where a $\mathcal{ADV}_{\mathcal{DIS}}$ helps an $\mathcal{ADV}_{\mathcal{MAL}}$ to convince the system that the dishonest user is closer than he actually is. This type of collusion is also known as terrorist fraud attack.

3.3. Security and Privacy Threats

The adversarial models described in Section 3.2 introduce a number of security and privacy threats to PBAC systems. Next we provide a list of potential attacks and relate them to the requirements of the system.

- **Identity-related threats**: As in any authentication protocol, the first threats we need to consider are those related with the identities of the elements involved in the protocol.

  I.1 *Impersonation*, is the process by which an entity attempts to pose as another element of the system. This type of attack can be done by using the identifier of another device or by mimicking its behaviour. Our goal is to prevent an attacker from impersonating a legitimate WWD.

- **Privacy-related threats**: the integration of wireless-enabled technologies to seamlessly authenticate WWDs on behalf of their owners pose serious privacy risks. By merely observing the packets being exchanged between the elements of the system an attacker might obtain sensitive information about the users. Since the identifiers of the devices are static and many solutions use them for authentication purposes, the following threats might appear:

  P.1 *Re-identification*, is the process by which apparently anonymous data is linked to its true owner. The identifier of devices interacting with the system can be eventually linked to real individuals.

For example, this can be achieved after physical observation of the owner of the device. Our goal is to avoid messages from revealing the identity of a particular user of the system.

P.2 *Unsolicited Tracking*, refers to the ability to observe the behaviour of an individual for a period of time. This term might refer to the actions or the movements of the user. Tracking might be possible even if re-identification is not. The goal of our scheme is to prevent attackers from linking two different messages belonging to the same user.

- **Location-related threats**: the robustness of the localisation process is a critical factor since the access to resources is not only based on the identity of the users but also on their location. Therefore, it is necessary to consider the error introduced by the underlying technology used to locate the users but most importantly whether the user is capable of fooling the system into believe that he is located somewhere else. There are several types of attack in this respect (Cremers et al., 2012):

  L.1 *Distance fraud*, is an attack in which the goal is to convince the system that the adversary is at a different distance than he really is. This type of attack is normally used to get access to resources that are out of the range of the adversary. The goal is to avoid that a legitimate user convinces the anchor of being in range while being outside of the proximity zone.

  L.2 *Mafia fraud*, consists of making the localisation system believe that a honest user of the system is at a different location. This is a man-in-the-middle attack that is also known as relay attack. By employing this type of attack the user unknowingly grants access to resources to the adversary. The goal is that the attacker cannot convince the authentication server that a legitimate user is in a particular location (close to an anchor) when he is not.

  L.3 *Terrorist fraud*, is the process by which an adversary is helped by a dishonest user to fool the system into believing that the user is at a different location. In some sense this is a form of consensual impersonation. The goal is that the attacker and a dishonest user cannot collude to grant access to the attacker on-behalf of user without physically giving the WWD to the attacker.

We have defined a number of adversarial models and threats but providing a complete solution capable of protecting from all of them is rather difficult, specially given the hardware limitations of WWDs. Therefore, the focus of this work is on the identity- and privacy-related threats inherent to continuous authentication schemes rather than trying to solve location-related problems. Notwithstanding, as we will see in the following sections, we introduce some basic countermeasures in an attempt to hamper the success of the latter threats.

4. Platform Description

For our prototype we use the eZ430-Chronos platform[2] from Texas Instrument as the WWD. It provides a highly integrated, wireless development system which is implemented as a wrist watch that the user can wear without requiring any adaptation. Moreover, the eZ430-Chronos platform presents several advantages that make it the ideal choice for implementing our prototype. It is a relatively affordable WWD hardware[3] that incorporates a well documented programming API and has the support of a wide development community and Texas Instrument experts. Also, there are some community supported firmwares that allow for a customised configuration. In particular, we chose the OpenChronos[4] firmware as the base for our development. The authentication server and the anchors are implemented as services in a computer. The anchors need an RF transmission module operating in the same band as the eZ430-Chronos.

Since our scheme is primarily concerned with allowing a secure, non-invasive continuous authentication of users while preserving their privacy, we identify the following key elements to tackle this problem:

- *Symmetric-key cryptography.* Symmetric key cryptography enables us to establish confidential channels between the elements of the infrastructure but it can also be used for authentication purposes using for example CBC-MAC. The legitimate WWDs and the elements of the tracking system should be able to use a symmetric-key encryption scheme in order to provide both authentication and confidentiality.

[2] http://processors.wiki.ti.com/index.php/EZ430-Chronos

[3] The development kit costs less than 60 euros.

[4] `http://github.com/poelzi/OpenChronos/`

The reason for using symmetric-key encryption instead of asymmetric-key encryption is basically due to the inherent hardware limitations of most Wearable Wireless Devices currently available on the market. The WWD should then include a cryptographic module that make encryption efficient both in time and power consumption. The eZ430-Chronos platform includes a built-in AES co-processor which allows for efficient symmetric-key cryptographic operations.

- *ID removal.* The messages transmitted by the wireless devices contain addressing information. An important mechanism to prevent an observer from using this information to track the device is to remove any identifiers contained in the packets or periodically change them. Moreover, we must prevent the disclosure of the device identifier in networks frames, otherwise the location of the user is continuously exposed. Instead, the identifier must be available but only to authorised entities, that is, to the infrastructure (e.g., using confidential channels). This prevents external users becoming aware of any user or device addresses used. The platform should then expose the lowest medium access network layer of the communication, allowing for custom packet formats where IDs are removed. The network drivers included in the OpenChronos expose the MAC layer and allow developers to work with the wireless interface at a low level. The TI proprietary stack SimpliciTI$^{\text{TM}}$stack[5] can be fully removed and a new stack can be implemented that does not use any ID for addressing.

- *Transmission power adjustment.* By limiting the strength of the wireless signals broadcast by the infrastructure, WWDs would only be able to respond to those signals that come from other neighbouring devices within a short time period. When the transmission power of the antennas is low enough, this can prevent remote users being able to authenticate with a particular node thereby pretending to be somewhere else. Together with timing constrains this can mitigate in some degree attacks that try to alter the legitimate location of the users. Anyway, this is not the primary focus of our work and we acknowledge that this feature can only be considered as very naive solution to the problem

[5] SimpliciTI is a TI proprietary low-power RF protocol aimed at simple, small RF networks

of Distance Bounding presented in Section 3.3. The tools provided by Texas Instrument allow developers to adjust all the parameters of the RF interface. In particular, transmission power can be adjusted to allow a sensitivity of few centimetres.

- *Data freshness.* It is necessary to ensure that the data generated is current to prevent the attacker from using a previous message in response to a present message, and thus succeeding in impersonating a legitimate user of the system. A system clock accurate enough could be used to check time constrains and synchronise to some extent the elements of the system. Random number generator can also help achieving the same features. The eZ430-Chronos includes an internal clock and although it does not include a hardware module for random number generators, their sensors can be used to get some entropy and produce pseudo-random numbers.

- *Tailored user interaction.* Some of the attacks and threats on continuous authentication using wireless devices are based on using the wireless device signals without the user consent. User interaction can be used to get explicit user consent. Instead of requiring the user to input any data the interaction should require a less invasive approach. The WWD should be capable of getting the attention of users and presenting relevant information (e.g. buzzing, vibrating, using an LCD screen) and get their feedback in an user friendly way when needed (e.g. clicking buttons, accelerometer data). The watch includes a buzzer and a LCD screen that can be used to attract users attention and present some relevant information to them. Also, the accelerometers can be used to get user feedback in an non-intrusive way (e.g. waving hands or tapping the screen for confirmation).

These are the building blocks that we use for the development of our prototype implementation. Further details are provided in the following section.

5. Prototype Implementation

This section presents the implementation details of our prototype for the provision of proximity-based services. We also discuss some of the limitations of our prototype at the end of the section.

5.1. Overview

The prototype implementation we devised consists of a one-way authentication mechanism that is based on the use of symmetric key encryption. The users and the Auth. Server share keys, K_i, that allow users to generate a message authentication code (MAC) at both ends of the communication channel. Anchors serve as authentication proxies to enable geographically distributed applications. The behaviour and interactions between the elements of our prototype are depicted in Figure 2. The notation used in this figure is described in Table 1.

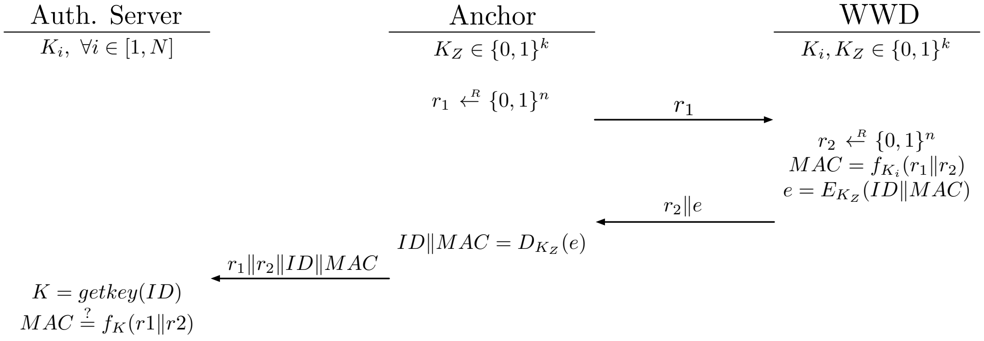
Auth. Server
$K_i,\ \forall i \in [1, N]$
Anchor
$K_Z \in \{0,1\}^k$
WWD
$K_i, K_Z \in \{0,1\}^k$
$r_1 \xleftarrow{R} \{0,1\}^n$
r_1
$r_2 \xleftarrow{R} \{0,1\}^n$
$MAC = f_{K_i}(r_1 \| r_2)$
$e = E_{K_Z}(ID \| MAC)$
$r_2 \| e$
$ID \| MAC = D_{K_Z}(e)$
$r_1 \| r_2 \| ID \| MAC$
$K = getkey(ID)$
$MAC \overset{?}{=} f_K(r1 \| r2)$

Figure 2: Continuous authentication protocol prototype

In our prototype, anchors are periodically broadcasting random numbers, r_1, which are used by WWDs in range together with their own random number, r_2, to generate the MAC code that will be finally verified by the authentication server. Additionally, anchors and WWDs share a symmetric zone key, K_Z, to conceal the device identifier (ID) at the application layer and thereby protect users' privacy[6].

Upon the reception of the reply, the anchor obtains the random value provided by the WWD, r_2, and decrypts the rest of the message, from which it obtains the device identifier, ID, and the MAC. These data together with the original random value r_1 are forwarded to the Auth. Server, which uses the ID to retrieve the key shared with the WWD. Finally, the Auth. Server checks the validity of the received MAC by using the key and the parameters received.

[6] Recall that we remove any identifiers contained in the message headers.

Symbol	Description
k, n	Size of the keys and random numbers used in the protocol
N	Number of legitimate WWDs in the system
K_i	Authentication key shared between the WWD and the Authentication Server
K_Z	Encryption key shared by all legitimate WWDs and anchors used for confidentiality
r_1, r_2	Random numbers
$\{0,1\}^n$	Set of all binary strings of length n
$\|\|$	Concatenation operator
$\xleftarrow{R}$	Random selection operator
$=$	Assignment operator
$\stackrel{?}{=}$	Equality operator
$f_K(\cdot)$	Message Authentication Code function using key K
$E_K(\cdot)$, $D_K(\cdot)$	Encryption and decryption operators respectively using key K
ID	WWD identifier
$getkey(\cdot)$	Method that returns the key associated with a given ID

Table 1: Notation of the protocol prototype

In what follows, we provide more details on the implementation of each of the components of our prototype.

5.2. Implementation details

The service offered by our prototype is the login into a Linux system. While the WWD is in range the session remains open, if the WWD is not responding for a certain period of time the session is terminated.

The authentication server has been implemented in a PC using *python* running a Linux distribution. Also, the anchor and the service provider are implemented as independent services in the same PC, for simplicity in the deployment. The PC is equipped with an USB dongle that is capable of communicating with the WWDs as they are based on the same RF transceiver. Moreover, this PC can communicate with the authentication server using mutually authenticated SSL channels.

Regarding cryptography, we use AES for confidentiality in the wireless channel because of the existence of a cryptographic module in the WWDs (see Section 4). We also take advantage of this module for the implementation of message authentication codes, i.e. $f_K(\cdot) = AES\text{-}CBC\text{-}MAC_K(\cdot)$. Moreover, we have created a small PKI using OpenSSL[7] with a single Certification Authority (CA) that issues certificates for the anchor, the service provider and the authentication server in order to support SSL communications.

[7] `http://www.openssl.org/`

The authentication server waits for SSL connections from anchors and service providers in the system. Only connections authenticated with a valid certificate signed by our CA are accepted. The anchor forwards the packets received from the WWD using the authenticated channel in such a way that for every received packet the authentication server knows which anchor it is by simply checking the client certificate used in the SSL session. When a WWD is authenticated, the authentication server notifies the service provider using also an SSL channel. After a short period of time without notifications from the authentication server, the user is assumed to be away and the service provider locks the system.

In the RF channel, the messages exchanged between WWDs and anchors have the following format $(src, dst, type, payload)$. The element of the message are described below:

- The source address, *src*. Anchors use it as a unique value that identify then, for WWD it is always set to 0xFF.
- The destination address, *dst*. This value is always set to 0xFF.
- The message type, *type*. It is basically used to determine the size of the payload and parse its contents. Currently, there are 2 message types RAND and AUTH.
- The content, *payload*. The content is just a random number for messages of type RAND and for type AUTH it contains an encrypted payload composed of the random number generated by the WWD, its ID and the MAC, everything encrypted with the zone key.

When the anchor needs to access the RF channel it interacts with the USB dongle using a serial API. The API provides methods for sending the random number and receiving AUTH packets that have already been decrypted using the zone key inside the dongle. Hence assuming the device includes some measures for tamper resistance, the zone key never leaves the USB dongle. We perform decryption within the dongle in order to protect the zone key. Moreover, the dongle is equipped with a cryptographic module similar to the one included in the WWD, which ensures an efficient decryption process. As for random numbers, they are generated in the PC and passed to the USB dongle using the serial API. The dongle then assembles the network packet and finally sends the random number over the RF channel. The anchor will only accept AUTH messages for a particular RAND message a small fraction of time, which is defined by the system administrator.

5.3. Analysis and Limitations

The prototype implementation is mainly concerned with the protection against attackers interested in impersonating legitimate users of the system or compromising their privacy.

An $\mathcal{ADV}_{\mathcal{HBC}}$ observes and analyses the communications in the RF channel in order to retrieve any information that allows him to link messages to particular WWDs or users. However, our prototype introduces several mechanisms that prevent honest-but-curious adversaries from successfully re-identifying or tracking users. In particular, WWDs do not include any identifiers at frame level, instead they are cryptographically protected within the packet payload. Moreover, the appearance of the payload changes for every new packet because it depends on two random numbers (r_1 and r_2). Therefore, it is very important to have a good source of entropy for the random number generators.

An $\mathcal{ADV}_{\mathcal{MAL}}$ might be interested in impersonating a legitimate user of the system to gain access to resources or in tracking users without their consent. To that end, he might try to deceive a the legitimate WWD into believe it is communicating with a legitimate anchor. This adversary might perform several types of attack. First, he can continuously transmit the same random number in an attempt to track a WWD. However, if the random numbers generated by the WWD are sufficiently random this type of attack is infeasible. Second, a malicious adversary might generate a significant number of different random numbers to retrieve valid authentication tokens to be latter used with the legitimate anchor. Unfortunately, the use of random numbers generated by the WWD does not protect against this type of threat. Including some kind of timestamping (e.g., a counter) in the packets would solve this problem but it would also introduce additional synchronisation problems, which are left for future work. Third, the adversary might attempt to perform a mafia fraud attack by relaying the random numbers from the anchor to the WWD and also relaying the reply back to the anchor. Two $\mathcal{ADV}_{\mathcal{COL}}$ adversaries (i.e., $\mathcal{ADV}_{\mathcal{MAL}}$ and $\mathcal{ADV}_{\mathcal{DIS}}$) might try to bypass the authentication process in a similar way by performing a terrorist fraud attack. These attacks are successful only if the reply arrives to the anchor immediately after the anchor sends the random number. Our solution does not fully address this threat but in an attempt to diminish it the anchors only accept replies to a particular random number within a very short time period specified with a predefined parameter.

Furthermore, $\mathcal{ADV}_{\mathcal{MAL}}$ and $\mathcal{ADV}_{\mathcal{DIS}}$ can alter some properties of the signal. On the one hand, the former adversary could launch denial-of-service (DoS) attacks by blocking the wireless signals transmitted by WWDs and Anchors. We consider that this type of attack is out of the scope of our solution. On the other hand, an $\mathcal{ADV}_{\mathcal{DIS}}$ might try to amplify the signals to convince the anchor of being in its vicinity. This type of attacks can only be eliminated with a precise clock measuring the time of arrival of the signals but we aim to reduce this threat by adjusting the transmission and by only accepting replies within a short time window.

Finally, an $\mathcal{ADV}_{\mathcal{CAP}}$ capturing anchors or a WWD could extract the cryptographic material from these devices. In case the key used to establish the confidential channel is compromised, all the communications intended for the anchor would be exposed. This poses a serious privacy breach because the ID of the WWDs would be accessible to the attacker. Therefore, it is important to prevent this from happening. In the following section we present an improved solution that removes the need for a confidentiality key in favour of a pseudo-anonymous scheme.

6. Pseudo-Anonymous Continuous Authentication Scheme

This section presents an improved scheme that by relaxing the anonymity requirement is capable to resist the threat of compromised zone keys. First, we provide a detailed description of the pseudo-anonymous scheme and then we discuss on the features and potential extensions to the protocol.

6.1. Overwiew

The pseudo-anonymous scheme consists of two phases. The first phase is more computation intensive but it only takes place when the WWD enters a new zone for the first time. At the end of this phase, both the WWD and the Auth. Server share a temporal key to be used in the following. The second phase, which is more lightweight for both the WWD and the Auth. Server, is used until the device leaves the area. In Figure 3 we show the interactions between the elements of the system and we separate the two phases by a dashed line. The new notation necessary for the pseudo-anonymous scheme is shown in Table 2.

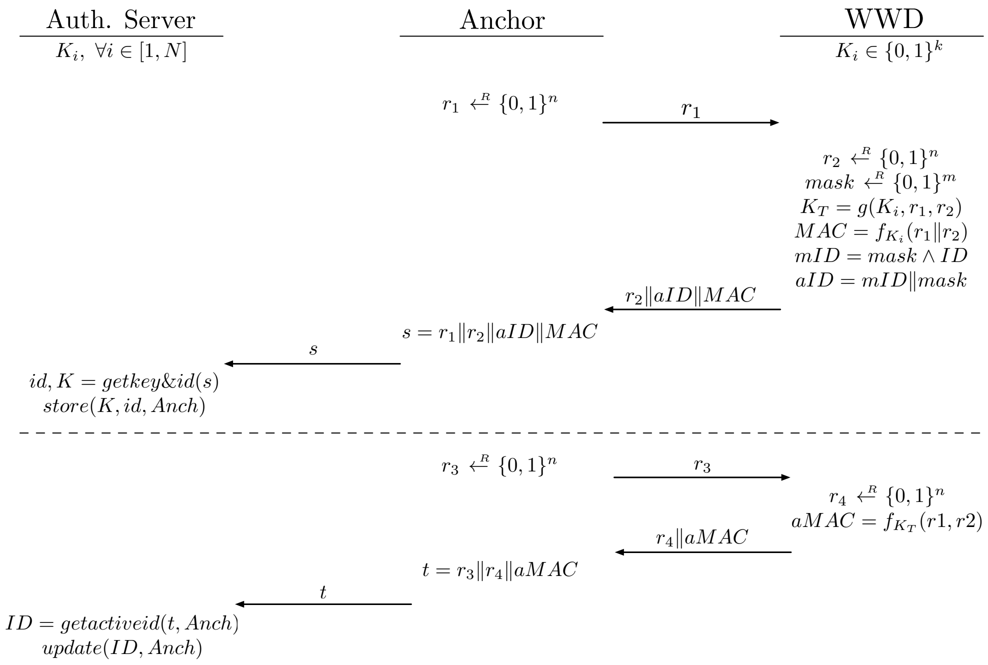
Auth. Server
$K_i,\ \forall i \in [1, N]$
Anchor
WWD
$K_i \in \{0,1\}^k$
$r_1 \overset{R}{\leftarrow} \{0,1\}^n$
r_1
$r_2 \overset{R}{\leftarrow} \{0,1\}^n$
$mask \overset{R}{\leftarrow} \{0,1\}^m$
$K_T = g(K_i, r_1, r_2)$
$MAC = f_{K_i}(r_1 \| r_2)$
$mID = mask \wedge ID$
$aID = mID \| mask$
$r_2 \| aID \| MAC$
$s = r_1 \| r_2 \| aID \| MAC$
s
$id, K = getkey\&id(s)$
$store(K, id, Anch)$
$r_3 \overset{R}{\leftarrow} \{0,1\}^n$
r_3
$r_4 \overset{R}{\leftarrow} \{0,1\}^n$
$aMAC = f_{K_T}(r1, r2)$
$r_4 \| aMAC$
$t = r_3 \| r_4 \| aMAC$
t
$ID = getactiveid(t, Anch)$
$update(ID, Anch)$

Figure 3: Pseudoanonymous Scheme

The communication model is very similar to the prototype implementation but here we use two different authentication codes: MAC and $aMAC$. The former is computed using the authentication key K_i shared between the Auth. Server and each particular WWD. This key is only used when the wireless devices require a new temporal key, K_T. The latter authentication code, $aMAC$, is generated with the agreed temporal key and is used throughout the second phase until the WWD leaves the zone.

Continuous authentication protocols require devices to be periodically proving their identity. We want to avoid exposing the master authentication key K_i too often and for that reason we use a temporal key K_T for authentication purposes. In this way, the K_i is only used at the very beginning of the protocol and if by any chance the temporal key is compromised, it would only be valid while the current session is active.

Also note from Figure 3 that the user ID is neither sent in clear nor encrypted. In order to identify which WWD sent the MAC, the authentication server uses the function *getid&key* that iterates over all the user IDs that match the aID in order to check which of the K_i was used to create

Symbol	Description
K_T	Temporal authentication key generated using $g(\cdot)$
$aMAC$	Message authentication code created with K_T
$mask$	Bit mask of the same size as the ID
mID	Masked ID computed as the bitwise AND operation of the $mask$ and the ID
aID	Pseudo-anonymous ID composed of two elements: $mask$ and mID
$Anch$	The anchor identifier obtained from the SSL connection
$\wedge$	Bitwise AND operator
$g(\cdot)$	Key derivation function
$getid\&key(\cdot)$	Returns the ID and temporal key
$store(\cdot)$	Stores the temporal key for a given ID and Anchor
$getactiveid(\cdot)$	Returns the ID of the WWD sending the $aMAC$
$update(\cdot)$	Refreshes the status of the temporal key

Table 2: Notation of the pseudo-anonymous scheme

the MAC. This process, which is depicted in Algorithm 1, is only executed when the wireless device needs a new temporal key.

Algorithm 1 Pseudo-code algorithm for *getid&key*

```
Input: $aID$, $MAC$, $r_1$, $r_2$
$m \leftarrow mask\_size(aID)$
for $i = 0$ to $2^m$ step 1 do
    $id \leftarrow generate(aID, i)$
    $K_i \leftarrow getkey(id)$
    if $(MAC == f_{K_i}(r_1 \| r_2))$ then
        return $(id, g(K_i, r_1, r_2))$
    end if
end for
return error
```

Clearly the complexity of this method depends on the number of bits used in the aID mask. The more bits we hide the more secure (private) the scheme is but the verification algorithm also becomes more inefficient. It is necessary to find the right balance between the number of bits used in the mask depending on the number of registered devices, the actual length of the IDs, and the computational capabilities of the authentication server. Also, it is important to take into consideration the mIDs already in use by other WWDs in order to minimise the disclosure of information about the original ID of the device.

After a successful authentication in phase 1, the authentication server stores the temporal key of the matching user together with the anchor identifier (*Anch*) that reported the presence of the WWD. The identifier of the anchor can be retrieved from the SSL certificate used in the connection between the anchor and the authentication server.

In phase 2, when the WWD has an active temporal key shared with the authentication server, it replies to the anchor by sending its own random number and the $aMAC$. Then, the authentication server checks, using the *getactiveid* function, whether the WWD still has an active temporal key and in such case refreshes the status of the key. If active keys are not refreshed in a given time frame they are removed from the database of active keys and any subsequent authentication attempt from the corresponding user will fail. In such case, the WWD will need to authenticate using phase 1 in order to get a new temporal key.

Algorithm 2 Pseudo-code algorithm for *getactiveid*

```
Input: $Anch$, $aMAC$, $r_3$, $r_4$
for each $id$ in $active(Anch)$ do
    $K_T \leftarrow getkey(id)$
    if $(aMAC == f_{K_T}(r_3 \| r_4))$ then
        return $id$
    end if
end for
return error
```

The *getactiveid* function, described in Algorithm 2, has lineal complexity on the number of clients active in a particular anchor, which is expected to be relatively low.

6.2. Discusion

The present design removes the need for a confidential channel between the Anchor and the WWD in favour of pseudo-anonymised identifiers. By doing so, the proposed scheme reliefs the system from relying on zone keys that might be captured or compromised. Also, thanks to this, privacy-conscious users only need to fully trust the authentication server to gain access to services. In the prototype implementation, even legitimate anchors, which are intended to serve as mere intermediaries, have access to the WWD identifiers after decrypting the messages received from the users.

One of the main downsides of this scheme with respect to the prototype implementation is that it requires a higher computational burden. Regarding the WWD, only the first phase of the protocol implies an increased number of operations while the second phase is even more lightweight since message confidentiality is no longer necessary. As for the authentication server, in addition to keep a record of the authentication keys for all WWDs, it needs to keep track of all active temporal keys. The bright side is that during phase 2 the complexity of the verification process is reduced because the authentication server only has to search among the number of clients active for a particular anchor.

Clearly, the efficiency of our algorithms can be improved. Since this was not the main goal of our algorithms, in the first phase we used a linear search approach that is dependent on the number of possible matching IDs. However, in a real-world deployment the search space could be huge. Therefore, it is important to explore new alternatives as it would not be difficult to move from the adopted solution to a more time-efficient one. This is left for future work.

7. Conclusion

In this paper we have presented a prototype implementation for proximity-based access control that takes advantage of WWDs to allow for continuous authentication. Our solution is mainly focused on the protection of user privacy while they access services in their vicinity although it also includes some basic countermeasures to reduce other identity- and location-related threats that may be caused by skilled adversaries.

The prototype is based on the eZ430-Chonos platform from Texas Instruments, which features a wireless-enabled wrist watch that is used for the seamless authentication of the user with the infrastructure. The feasibility of our prototype has been empirically in a real-world setting, i.e., a secure and privacy-preserving log-in and log-out service for workstations. Moreover, this solution is extensible to many other application domains such as emergency response scenarios, where a strong authentication with minor user interaction is required.

Besides the prototype implementation we have proposed a pseudo-anonymous solution that overcomes some of the limitations of the prototype. This new scheme exempts the anchor from performing cryptographic operations. But most importantly, in this new version of the protocol the anchors do not need to share any key material with the WWDs, which posed an important threat to the security of the system if they were compromised. Also, this allows the distribution of slave anchors and a single master anchor in charge of relaying the messages from the slaves to the server. Virtually, any wireless device regardless of its computational power would be able to behave as a slave anchor. This setting fits neatly into the Internet of Things paradigm where any physical device is network-enabled.

As for future work, we plan to explore the possibilities that the LCD display and the sensors present in the eZ430-Chronos platform bring to enhance the prototype. For example, the use of the LCD screen and the buttons can be used to prevent some attacks. Also, one aspect we need to carefully study is the use of sensors as a source of entropy for generating random numbers. However, recent works suggest that sensors are not always a good source for entropy. New WWD platforms with random number generators should be explored.

We also plan to extend our scheme beyond proximity detection, allowing different anchors to collaborate with each other in order to pinpoint the location of the users. This is an interesting step towards more resilient solutions to location-related attacks. Furthermore, we are interested in extending our prototype by producing location evidences that can be validated by external entities. Finally, we need to improve the efficiency of the our algorithms to scale adequately with the number of users in the system.

Acknowledgements

This work has been partially funded by the European Commission through the FP7 project NESSoS (FP7 256890) and the Spanish Ministry of Science and Innovation through the ARES (CSD2007-00004) and SPRINT (TIN2009-09237) projects. SPRINT is co-financed by FEDER (European Regional Development Fund). The second author is supported by the Spanish Ministry of Education through the F.P.U. Program. Also, special thanks to the anonymous reviewers for their valuable comments.

References

ABI research . Wearable Sports and Fitness Devices Will Hit 90 Million Shipments in 2017. [online]; 2012. `http://www.abiresearch.com/press/wearable-sports-and-fitness-devices-will-hit-90-mi`.

Capkun S, Hubaux JP. Secure Positioning in Wireless Networks. In: Selected Areas in Communications, IEEE Journal on. volume 24; 2006. p. 221–32.

Chandran S, Joshi J. Lot-rbac: A location and time-based rbac model. In: Ngu A, Kitsuregawa M, Neuhold E, Chung JY, Sheng Q, editors. Web Information Systems Engineering ? WISE 2005. Springer Berlin Heidelberg; volume 3806 of *Lecture Notes in Computer Science*; 2005. p. 361–75.

Cremers C, Rasmussen KB, Schmidt B, Capkun S. Distance Hijacking Attacks on Distance Bounding Protocols. In: IEEE Symposium on Security and Privacy. San Francisco, California, USA: IEEE Computer Society; SP; 2012. p. 113–27. 978-0-7695-4681-0.

Cruz IF, Gjomemo R, Lin B, Orsini M. A location aware role and attribute based access control system. In: Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems. New York, NY, USA: ACM; GIS '08; 2008. .

Damiani ML, Bertino E, Catania B, Perlasca P. Geo-rbac: A spatially aware rbac. ACM Trans Inf Syst Secur 2007;10(1).

Denis Foo Kune John Koelndorfer NH, Kim Y. Location leaks over the GSM air interface. In: Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS'12). 2012. .

Di Pietro R, Mancini L. Security and privacy issues of handheld and wearable wireless devices. Communications of the ACM 2003;46(9):74–9.

Ferreres AIGT, Álvarez BR, Garnacho AR. Guaranteeing the authenticity of location information. IEEE Pervasive Computing 2008;7(3):72–80.

Gollakota S, Hassanieh H, Ransford B, Katabi D, Fu K. They can hear your heartbeats: non-invasive security for implantable medical devices. SIGCOMM Comput Commun Rev 2011;41(4):2–13.

Gupta SKS, Mukherjee T, Venkatasubramanian K, Taylor TB. Proximity Based Access Control in Smart-Emergency Departments. In: Proceedings of the 4th annual IEEE international conference on Pervasive Computing and Communications Workshops. Washington, DC, USA: IEEE Computer Society; PERCOMW '06; 2006. p. 512.

Hansen F, Oleshchuk V. Spatial role-based access control model for wireless networks. In: In Proceedings of the 58th IEEE Vehicular Technology Conference (VTC?03). Vol. 3. IEEE Computer Society. 2003. .

Hay S, Harle R. Bluetooth Tracking without Discoverability. In: Proceedings of the 4th International Symposium on Location and Context Awareness. Berlin, Heidelberg: Springer-Verlag; LoCA '09; 2009. p. 120–37.

IMS research . Wearable Technology Market to Exceed $6 Billion by 2016. [online]; 2012. `http://imsresearch.com/press-release/Wearable_Technology_Market_to_Exceed_6_Billion_by_2016`.

Kao KF, Liao IE, Lyu JS. An indoor location-based service using access points as signal strength data collectors. In: International Conference on Indoor Positioning and Indoor Navigation (IPIN). 2010. p. 1 –6.

Kirkpatrick MS, Damiani ML, Bertino E. Prox-rbac: a proximity-based spatially aware rbac. In: Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems. New York, NY, USA: ACM; GIS '11; 2011. p. 339–48.

Sadeghi AR, Visconti I, Wachsmann C. Location privacy in rfid applications. In: Bettini C, Jajodia S, Samarati P, Wang X, editors. Privacy in Location-Based Applications. Springer Berlin Heidelberg; volume 5599 of *Lecture Notes in Computer Science*; 2009. p. 127–50.

Saponas TS, Lester J, Hartung C, Agarwal S, Kohno T. Devices that tell on you: privacy trends in consumer ubiquitous computing. In: Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium. Berkeley, CA, USA: USENIX Association; SS'07; 2007. p. 5:1–5:16.

Vossiek M, Wiebking L, Gulden P, Wieghardt J, Hoffmann C, Heide P. Wireless Local Positioning. Microwave Magazine, IEEE 2003;4(4):77 – 86.

Zafeiropoulos A, Papaioannou I, Solidakis E, Konstantinou N, Stathopoulos P, Mitrou N. Exploiting Bluetooth for deploying indoor LBS over a localisation infrastructure independent architecture. International Journal of Computer Aided Engineering and Technology 2010;2(2):145–63.