TOWARD SELF-AUTHENTICABLE WEARABLE DEVICES

FIDEL PANIAGUA DIEZ, DIEGO SUÁREZ TOUCEDA, JOSÉ MARÍA SIERRA CÁMARA, AND SHERALI ZEADALLY

ABSTRACT

Wearable devices communicate among themselves, but they also need to communicate with remote entities through the Internet in order to share information. Given the sensitivity of the information handled, authentication is needed to allow data access only to authorized parties. However, the existing authentication solutions for wearable devices are limited to scenarios where direct communication between the authenticating parties is possible. With this in mind, we propose an authentication protocol that enables secure mutual end-to-end authentication between a wearable device and any other entity such as another wearable device, a personal device (mobile phone), a remote server, or a user's application. Our design uses a point-to-point authentication protocol (end-to-end authentication) regardless of whether other intermediate devices are trusted or not. Finally, we present a security evaluation of the proposed authentication protocol.

INTRODUCTION

According to various studies, the number of wearable devices is expected to continue to grow in the near future. For example, ABI research [1] estimates the global market for wearable devices in health and fitness could reach 170 million devices by 2017. These wearable devices can hold a lot of useful information about their users. Such information includes blood pressure, heart rate, activity tracking, and tastes that can be used for several purposes including health, fitness, recommendations, and so on. However, this information is of little use if wearable devices lack the ability to exchange it with other devices or services. Due to wearable devices' limited range of connectivity, the use of other personal devices as intermediaries to share this information over longer distances must be possible.

Similarly, because of the sensitive information (e.g., medical records or pacemaker configurations) wearable devices may handle, it is essential that such information be protected from unauthorized access or modification. For example, consider a pacemaker updating the actual conditions of its holder in an online medical system of a hospital. Before sending the updated patient's information, the pacemaker has to make sure that it is really communicating with the hospital (hospital authentication) to prevent information leaks. In the same way, before updating the patient's information, the medical system has to make sure that the information is really coming from the patient's pacemaker (pacemaker authentication) to ensure that the patient's medical information is also correctly updated. Furthermore, if, after receiving the updated patient's information, the medical service would like to change the patient's pacemaker's configuration, again, mutual authentication is necessary to protect the patient's health. The medical service would have to check the identity of the pacemaker to know that it is updating the configuration of the right device. The pacemaker would also need to check the identity of the medical service to verify that the new configuration is coming from a trusted source. Therefore, it is necessary that a secure authentication service be associated with any wearable device holding sensitive information.

At present, a few solutions exist that address the problem of a secure authentication service for wearable devices, but they are more concerned about authenticating the user holding the wearable device than the device itself and are limited to scenarios where direct communication between the authenticating parties is possible. Some of them, such as [2], are only oriented to the authentication among wearable devices in the same body area network (BAN), while others, such as [3], consider a bigger scenario where a wearable device needs to also communicate with a remote entity. However, in the latter case, end-to-end authentication between the wearable device and the remote entity is not provided. First, the wearable device authenticates itself with an intermediate device (e.g., a mobile phone) in its BAN that collects the wearable's information; afterward, the intermediate device authenticates itself with the remote entity to send the information collected from the wearable. From our point of view, this approach is not robust enough: wearable devices are not really authenticating against a remote entity, but only to an intermediate device; and the whole process of communication could be compromised in the case of an unreliable intermediate device.We want to go a step beyond, and have an autonomous and independent wearable

Fidel Paniagua Diez, Diego Suárez Touceda, and José María Sierra Cámara are with Carlos III University of Madrid.

Sherali Zeadally is with the University of Kentucky.

1536-1284/15/$25.00 © 2015 IEEE

device that can authenticate end-to-end with a remote entity by itself.

Regarding this concern, other well-known protocols that provide end-to-end authentication (among other security services), such as TLS, are not an option due to the wearable's lack of the necessary resources to both implement a TCP/IP stack and manage their overhead. Nor are other Transport Layer Security (TLS)-based alternatives, such as EAP-TLS for smartcards [4], since they are based on the assumption that the smartcard performs the authentication in conjunction with a Smartcard Interface Entity that does have the necessary resources.

With the aforementioned limitations of current solutions, in this article, we propose an authentication protocol that can perform secure mutual end-to-end authentication between a wearable device and any other system such as another wearable device, an external personal device, an authentication server, or a user's application with which it needs to communicate. Our design is based on a point-to-point protocol where the wearable device is authenticated directly against the other final entity regardless of whether the communication is direct or through other some intermediate (trusted or not) devices (e.g., the user mobile phone) used for forwarding purposes because of the wearable devices' lack of long-range connectivity. Furthermore, although our protocol is initially focused on authentication, it could easily be extended to also include the exchange of a session key that allows the establishment of a secure channel to protect the data in transit.

One of the main objectives of this work is to design a solution that is as independent as possible while leveraging standard and secure technologies, also taking into consideration the device size to make our solution viable for wearable devices: wristband, rings, pacemakers, and so on. With this in mind, we propose the use of a smart card (based on Java Card Technology [5]) within the wearable device in order to make it self-authenticable and provide it with near field communication (NFC) [6] connectivity.

The rest of the article is organized as follows. We briefly present an overview of the technologies used in this article. We describe our proposed authentication protocol, including its architecture and the authentication methods used. The security of our proposal is analyzed. Finally, we conclude the work presented in this article.

Related Work and Technologies

In this section we describe the most important technologies used in our proposed authentication protocol.

Wearable Devices

The idea behind wearable devices is to add technology in everyday life, helping to improve human's day-to-day activities. A few examples of areas where wearable devices are increasingly being deployed and have demonstrated satisfactory results include health, sports, entertainment, and the textile industry.

Early wearable devices were limited in storage capacity and were generally standalone devices. Over the years, they have matured into advanced and useful interoperable wearable devices that support short-range connectivity using standardized protocols such as Bluetooth and NFC. An overview of different wearable devices and their applications is illustrated in Fig. 1.

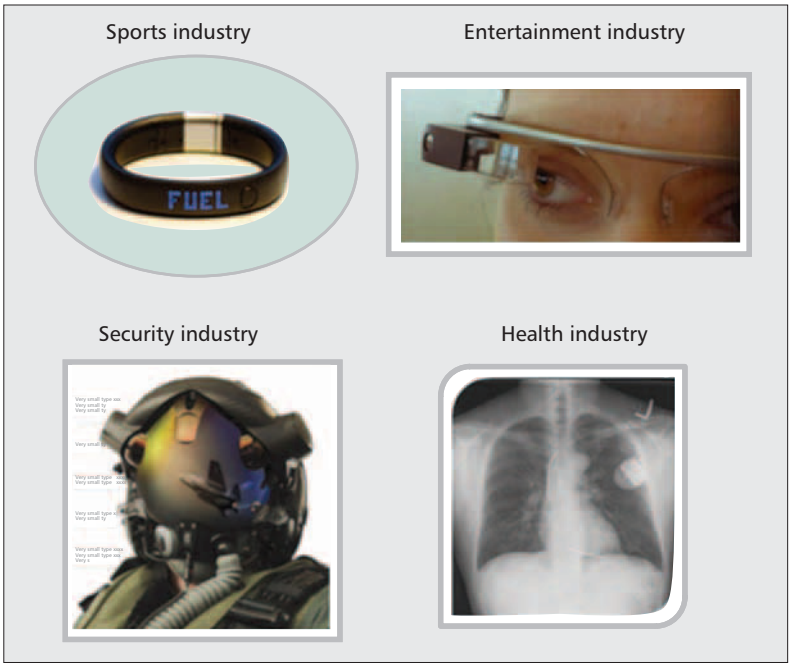

Figure 1. Wearable devices.[1]

Recent research efforts have focused on improving the connectivity range of wearable devices using more powerful personal devices (e.g., mobile phones or tablets) as intermediate devices. However, this new communication method also opens up new security challenges that should be addressed, which involve protecting wearable devices from both malicious intermediate devices and unauthorized access attempts from other entities residing on the Internet.

Near Field Communication

Near field communication [6] is a technology that is currently integrated in many mobile devices that try to simplify life's common tasks such as making transactions, exchanging digital content, or connecting electronic devices. NFC's market penetration is expanding each year, and it is expected that by this year half a billion people worldwide will use it in conjunction with their mobile devices as travel tickets on metros, subways, and buses [7].

NFC operates in the radio frequency of 13.56 MHz with data transmission rates ranging from 106 to 424 kb/s.

Smart Cards

Smart cards are hardware devices used to protect sensitive operations such as electronic payments and access control. They can store sensitive information securely and have cryptographic capabilities.

[1] Figure 1 was created using images from several sources:

Sports industry image: Nike FuelBand by Peter Parkes licensed under CC BY.

Entertainment industry image: Detail of Google Glass by Antonio Zugaldia, reduced from the original, licensed under CC BY.

Health industry: Cardioverter defibrillator by Gregory Marcus licensed under CC BY.

Security industry: The HMDS for the F-35 Lightning II by the United States Marine Corps, in the public domain.

	Remote authentication	Intermediate authentication	Holder authentication
Low	X		
Intermediate	X	X	
High	X	X	X

Table 1. Security level — authentication control.

Smart card characteristics, ranging from physical characteristics to commands to interact with cards, are described in the International Organization for Standards/International Electrotechnical Commission (ISO/IEC) 7816 standard series. Security and commands for information interchange are defined in ISO/IEC 7816-4 [8].

Traditionally, the use of smart cards has been strictly attached to a terminal. Both the terminal and the smart card were involved in the authentication process, working together as a split-supplicant [4]. However, more recent works [9] have proposed using a smart card in an autonomous way (standalone supplicant) where the card is able to take part in the authentication process by itself. This new functionality is based on the Java Card technology, which allows Java-based applications to be run on smart cards, and its autonomy can only be achieved if the smartcard has connectivity.

Point-to-Point Protocol

Point-to-Point Protocol (PPP) [10] is a data link protocol used to establish a direct connection between two nodes. It provides a variety of services (e.g., encapsulating multiprotocol datagrams) once connectivity is achieved.

PPP allows Extensible Authentication Protocol (EAP) packets to be transported over the ISO 7816 protocol by mapping the PPP header to commands proposed in ISO/IEC 7816-4.

Extensible Authentication Protocol

The Extensible Authentication Protocol (EAP) [11] is an authentication framework, and is not a specific authentication mechanism. It provides some common functions and negotiation options of authentication methods called EAP methods. Some of these methods are based on the use of regular passwords (EAP-MD5), the use of one-time passwords (EAP-OTP), or the use of certificates (EAP-TLS). Besides, EAP can be extended with new authentication mechanisms.

Imprinting

The imprinting mechanism was defined in [12]. This mechanism attempts to solve the problem of establishing trust between two devices. It is based on the duckling imprinting phase, that is, the process by which a newborn duck establishes a pattern of recognition for its parents. Imprinting consists of a secure association established when devices are going to be deployed. During this phase both devices share something that allows identification beetween these devices in the future.

Proposed Authentication Protocol

As mentioned previously, the new connectivity capabilities for wearable devices using short-range communication protocols, such as NFC, and the need to increase their communication range to allow Internet connectivity using other personal devices as intermediate devices opens up new security challenges. These intermediate devices may be malicious and try to intercept or modify the information in transit during wearable devices' communications. Once connected to the Internet, the wearable device could become a target for attacks attempting to access or modify the data it holds.

To address the aforementioned challenges, we have proposed a new authentication protocol for wearable devices. The main functionalities of this protocol include:

- A secure protocol for the mutual authentication of wearable devices and any other network element such as another wearable device, a user's application, or a remote server. This protocol can work directly over a short-range connection or use other intermediate devices (trusted or untrusted) for Internet access.
- A secure imprinting mechanism that allows the wearable device to differentiate between trusted and untrusted intermediate devices.
- Storage of different authentication credentials (password, certificates, etc.) within the wearable device. This also opens up the opportunity to use the holder's biometric data obtained by the wearable device itself as a source of authentication.
- Inclusion of cryptographic and logical capabilities within the wearable device, allowing it to make decisions according to the sensitivity of information handled by the wearable device.

In the rest of this section we first analyze the functional and non-functional requirements, and justify the design of the wearable device used. Then we present several application scenarios of our proposed authentication protocol according to the sensitivity level of the information being handled by the wearable device. Finally, the proposed system architecture is presented, followed by a description of the authentication methods available in our proposed protocol.

Wearable Device Requirements and Proposed Design Justification

Several requirements arise for wearable devices in order to solve the challenges raised by the ability to securely exchange information with other devices or services:

1. The first requirement is the limited size of a wearable device. The proposed solution should fit inside a wearable device of any kind using existing technologies.
2. The second requirement arises from the necessity of providing the wearable device with strong authentication capabilities. In order to do so, the wearable device should have the capability to perform cryptographic and logic operations.
3. The third requirement has to do with credentials used to perform the authentication as further described below. The wearable device should have storage capabilities.

4. Finally, the fourth requirement is Internet connectivity. Since a direct connection with the Internet is not possible due to the limited capabilities of a wearable device, we need to use a short-range communication technology that allow the wearable device to use other devices in its communication range as intermediate devices to enable Internet access.

The objective of our solution is to design a protocol as device-independent as possible while exploiting already proven standards that enable its security and interoperability. With this in mind, we propose to include a Java Card based on smart card technology within the wearable device to satisfy requirements 2 (cryptographic and logic capabilities) and 3 (storage capability), as well as requirement 1 (size limit) due to the small size of smart cards. To satisfy requirement 4 (connectivity) and considering the size limit requirement, we have included NFC capabilities within the wearable device. One of the motivations for using NFC instead of other alternatives such as RFID [13] is its resilience against man-in-the-middle attacks.

During the rest of this article we call this device WD-NFC-JCard: wearable device with NFC and Java Card capabilities.

Application Scenarios

Before describing our proposed protocol, we present three different application scenarios according to the sensitivity of information handled by the wearable device. These scenarios are based on Fig. 2, where a WD-NFC-JCard wants to authenticate itself to a remote server using a personal device with Internet connectivity (e.g., a mobile phone) as an intermediate device.

Below, we describe different modes of operation that our proposed architecture provides (a summary is provided in Table 1):

1. Low security level: The information the WD-NFC-JCard holds is unclassified. In this scenario, we can use an untrusted intermediate device (no authentication between the wearable device and the mobile phone is required), and we only perform mutual authentication with the remote server. It is worth noting that even if the intermediate device is not trusted, end-to-end authentication can still be performed in a secure way.
2. Intermediate security level: The information the WD-NFC-JCard holds has a moderate security level. In this scenario we limit the use of intermediate devices to only trusted ones in order to provide an additional level of protection to the information. Therefore, prior to performing the mutual authentication with the remote server, it is necessary to perform a secure pairing (mutual authentication) with the intermediate device.
3. High security level: The information the WD-NFC-JCard holds has a highly classified security level. In this case, before applying the mechanisms (secure pairing and mutual authentication) described in the previous scenario, a PIN input from the holder of the wearable device is required. Due to the fact that some wearable devices do not have a keyboard, a biometric token is also acceptable.

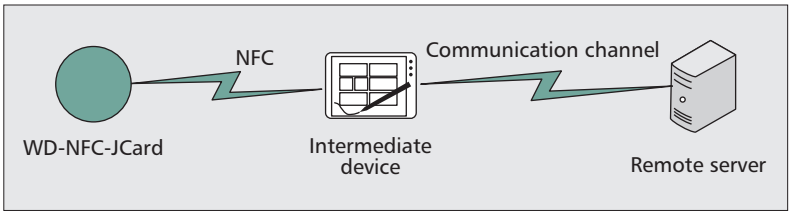

Figure 2. System elements.

Architecture

As mentioned before, when a wearable device needs to communicate with a remote server over the Internet, it uses an intermediate device that is within its communication range. To do so, a communication channel should be established between the wearable device and the intermediate device using the NFC protocol. Once this channel has been established, the intermediate device is responsible for opening the communication channel between itself and the remote server, forwarding the packets in both directions between the wearable device and the remote server, as shown in Figure 2.

This communication method is based on a standalone supplicant, where the WD-NFC-JCard uses the intermediate device only for communication but not to authenticate the remote server. When this communication channel is set up between the WD-NFC-JCard and the remote server, the authentication process starts.

Our proposed authentication protocol is based on the EAP authentication protocol. Since EAP is an authentication framework that does not assume the use of any particular authentication method, we define the specific methods used in our proposed authentication protocol. In addition, the fact that EAP will be running within the WD-NFC-JCard implies that the mobile device does not need to understand the authentication messages (except when its own authentication is also required), but only to forward the messages received in both directions.

As shown Fig. 3, in the WD-NFC-JCard the EAP messages are encapsulated in PPP frames, as explained in [9]. The mapping of the PPP headers with the ISO/IEC 7816-4 commands has been done based on [14]. Finally, the commands are transmitted over the air over 13.56 MHz radio waves between the WD-NFC-JCard and the intermediate device according to the ISO-14443-4 standard [15].

The intermediate device receives and unencapsulates the packets from the WD-NFC-JCARD to the EAP level, and then re-encapsulates and sends them over Internet. If a packet comes from the remote server the encapsulation/unencapsulation process is the exact opposite. The accounting, authorization, and authentication (AAA) protocol is used to exchange the authentication messages between the intermediate device and the remote server.

Finally, the remote server unencapsulates the different protocols until it can retrieve the authentication information encapsulated by the EAP protocol.

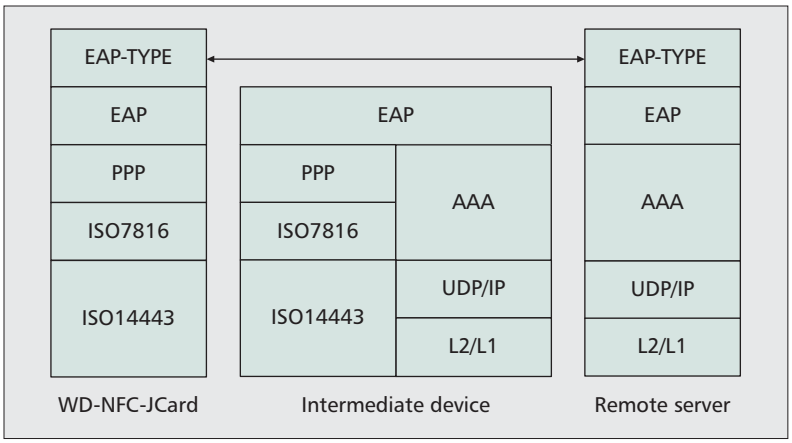

Figure 3. Proposed authentication architecture.

Authentication Methods

We used two different authentication methods in our proposed authentication protocol. First, a method to authenticate the intermediate device and the WD-NFC-JCard is defined. Then a method to authenticate the WD-NFC-JCard and the remote server is defined. In both cases, it should be taken into account that:

- The certificates used in our system are based on the X509v3 standard. Since our authentication methods are algorithm-independent, the size of the certificates would depend mainly on the specific algorithm and key length[2] chosen.
- We describe the cryptographic operations (encrypt, decrypt, sign, verify) that should be performed in each case, but we do not specify the particular algorithm to be used. Any algorithm supported by the smartcard can be chosen depending on the application needs.

Intermediate Device — WD-NFC-JCard Authentication — In this case, the holder of the wearable device is responsible for registering the trusted intermediate device with the WD-NFC-JCard. To do so, pairing the intermediate device with the WD-NFC-JCard is performed using the imprinting mechanism described in [3]. The necessary steps to establish this link between the intermediate device and the WD-NFC-JCard are shown in Fig. 4:

1. NFC communication between the WD-NFC-JCard and the intermediate device is established.
2. If the WD-NFC-JCard has not been paired with any intermediate device before, a pairing process is automatically started by the WD-NFC-JCard by sending a pairing request.
3. The intermediate device sends some unique identifiable information to the WD-NFC-JCard (e.g., its international mobile equipment identity [IMEI], medium access control [MAC], and processor model in a smartphone).
4. The WD-NFC-JCard stores the data received from the intermediate device and creates a pair of private and public keys.
5. A public key certificate (PKC) and a digital signature of the received data are sent from the WD-NFC-JCard to the intermediate device.
6. Finally, the intermediate device verifies the digital signature received, and if everything is correct, the certificate is stored, and a confirmation message is sent to the WD-NFC-JCard.

If the holder of the wearable device wants to link another device in the future, the holder should remove the data stored in the WD-NFC-JCard associated with the old intermediate device and then start the pairing process with the new intermediate device.

In future communications, if a WD-NFC-JCard wants to authenticate the intermediate device as a trusted one, it can use the EAP method described in Fig. 5. The steps to be followed during this authentication method, which we have called Wearable Device Intermediate Device (WBID), are:

1. A communication channel is established between the WD-NFC-JCard and the intermediate device through NFC using PPP configuration messages.
2. The WD-NFC-JCard sends the intermediate device an EAP Authentication message indicating that it wants to start an authentication process.
3. The intermediate device sends an EAP Request message to the WD-NFC-JCard to start the authentication procedure. Within the request, the intermediate device includes a random number encrypted with the public key of the WD-NFC-JCard.
4. The WD-NFC-JCard decrypts the received message and sends an EAP Response to the intermediate device with the number received.
5. If the received number is correct, the intermediate device confirms this with the WD-NFC-JCard. The WD-NFC-JCard has been authenticated.
6. Next, the WD-NFC-JCard sends an EAP Request message to the intermediate device in order to start the authentication procedure in the other direction.
7. The intermediate device sends an EAP Response message to the WD-NFC-JCard with its unique identifiable information (IMEI, MAC, and processor model for a smartphone) encrypted with the WD-NFC-JCard's public key.
8. Finally, the WD-NFC-JCard decrypts the received message and verifies that the data corresponds to that of the intermediate device paired in the imprinting process. If it is correct, the WD-NFC-JCard confirms this with the intermediate device. As a result, both devices have been mutually authenticated.

WD-NFC-JCard — Remote Server Authentication — In this case, the objective is to enable mutual authentication between the WD-NFC-JCard and the remote server. This authentication is achieved through the use of PKCs. These two assumptions are made:

- The public key of the WD-NFC-JCard is known by the authentication server and is trusted.
- The public key of the authentication server is known by the WD-NFC-JCard and is trusted.

The WD-NFC-JCard and authentication server hold a PKC that shall be used during this

[2] As a guide for choosing the key length, we recommend following "NIST Special Publication 800-57: Recommendation for Key Management."

phase of mutual authentication. The EAP name chosen for this authentication method is wearable device remote server (WDRS). It is based on a double challenge-response in four rounds. The general steps (as shown in Fig. 6) to perform this authentication are as follows:

1. A communication channel is established between the WD-NFC-JCard and the intermediate device through NFC using PPP configuration messages.
2. The WD-NFC-JCard sends an EAP Authentication message to the intermediate device indicating that it wants to start an authentication process with a remote server.
3. The intermediate device sends an EAP Request message to the WD-NFC-JCard allowing it to start the authentication process.
4. The WD-NFC-JCard sends an EAP Response message, including its certificate ID, to the remote server via the intermediate device.
5. The intermediate device encapsulates this message and sends it to the remote server.
6. The remote server generates a random number, rS, and initiates the wearable device authentication process by sending a response, including rS, to the WD-NFC-JCard via the intermediate device .
7. The intermediate device unencapsulates the EAP packet received from the remote server and forwards it to the WD-NFC-JCard.
8. The WD-NFC-JCard generates a random number rC and encrypts rS with its private key. Finally, it encapsulates rC and sign (rS) in an EAP Response and sends it to the remote server via the intermediate device.
9. The intermediate device encapsulates this message and forwards it to the remote server.
10. The remote server checks the correctness of the digital signature received. If it is correct, the WD-NFC-JCard is authenticated. Next, it encrypts rC with its private key and encapsulates this digital signature in an EAP Request packet, which is sent to the WD-NFC-JCard through the intermediate device.
11. The intermediate device unencapsulates the EAP packet received from the remote server and sends it to the WD-NFC-JCard.
12. The WD-NFC-JCard checks to see if the digital signature over rC is correct. If it is, the remote server has also been authenticated. Finally, the WD-NFC-JCard sends an EAP SUCCESS message to the remote server via the intermediate device to inform the remote server that the mutual authentication has been successful.
13. The intermediate device encapsulates this message and forwards it to the remote server.

Security Evaluation

In this section we evaluate our proposed authentication protocol, which supports three main functionalities.

The first funtionality is related to the imprinting of the WD-NFC-JCard and the intermediate device. Since the WD-NFC-JCard and the intermediate server do not yet know each other, the owner of the WD-NFC-JCard is responsible for starting this phase with the appropiate devices. This process is protected against enveasdropping attacks by the short-range distance and physical properties of NFC communications.

Once the imprinting process has been executed, the devices will share unique identifiable data between them, which will allow them to mutually authenticate in the future using the second WBID functionality. In this process the intermediate device identity is protected by previously exchanged identifiable data (IMEI, MAC, and processor model for a smartphone), while the identity of the WD-NFC-JCard is protected by the proof of possession of the private key related to its digital certificate. Both mechanisms together prevent the possibility of spoofing attacks. Also, as in the imprinting process, the integrity of the communication is protected by the physical properties of NFC communications.

The third functionality of our protocol is WDAS, which is related to the mutual authentication of the WD-NFC-JCard and the remote server. In this case, both identities are protected

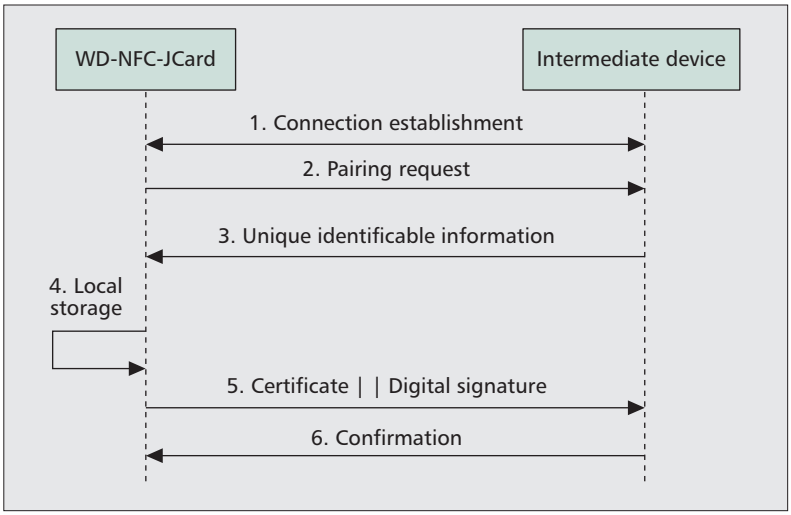

Figure 4. Pairing process.

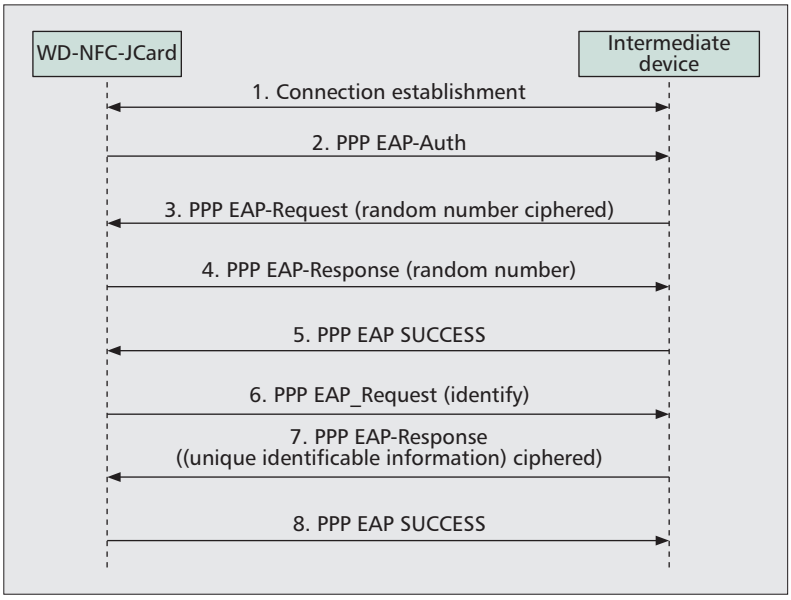

Figure 5. WD-SCard — mobile device authentication flow.

Our proposed protocol is based on a digital signature scheme with a challenge-response mechanism used to authenticate the different entities present in the system. Our proposed authentication protocol is not only secure, but flexible enough to provide different levels of protection based on the sensitivity of the information handled.

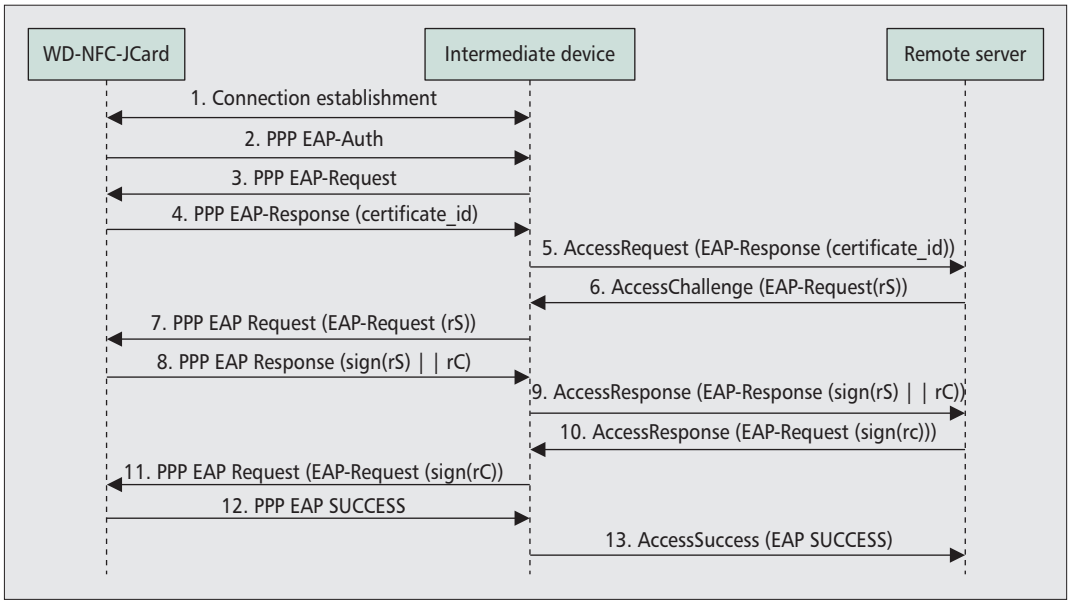

Figure 6. WD-SCard — authentication server authentication flow.

by proof of posession of the private keys related to their digital certificates. Authentication and integrity are therefore provided by digitally signing the exchanged messages. However, the possibility of using an untrusted intermediate device should be clarified. Two possible scenarios arise:

- Although untrustworthy, the intermediate device forwards the packets between the WD-NFC-JCard and the remote server. In this scenario the authentication protocol can carry out the mutual authentication of the involved parties in a secure way.
- A different scenario occurs when the untrustworthy intermediate device refuses to forward the packets between the WD-NFC-JCard and the remote server. In this case, the intermediate device cannot break the security of our authentication protocol (it is not possible to successfully authenticate a party without the necessary credentials); however, it can prevent communication between the involved parties. In this case, the WD-NFC-JCard should find another intermediate device willing to forward the communication.

In relation to the operation and storage of credentials, the use of a smart card increases their security in comparision with a non-hardware-based solution. However, smart cards are not unbreakable, and their resilience against some attacks, such as the side channel attack [16], should be taken into account when choosing the specific one to be used.

Finally, although our protocol is initially focused on authentication, it could easily be extended to include the exchange of a session key that allows the establishment of a secure channel to protect the data in transit.

Conclusion

Secure authentication has not received much attention in the area of wearable devices. However, from our point of view and because of the sensitivity of the information handled by wearable devices, authentication should be an essential service that any wearable device should provide.

With this in mind, in this article we have presented an authentication protocol for wearable devices that allows a secure mutual authentication between a wearable device and any other entity to be performed. Our proposed protocol is based on a digital signature scheme with a challenge-response mechanism used to authenticate the different entities present in the system. Our proposed authentication protocol is not only secure, but flexible enough to provide different levels of protection based on the sensitivity of the information handled.

Acknowledgments

This work is part of the research project SAVIER (Situational Awareness Virtual Environment) supported by Airbus Defense and Space.

References

[1] ABI Research, "Body Area Networks for Sports and Healthcare," *ABI Research*, Sci. Rep., 2012.
[2] Z. Zhang *et al.*, "ECG-Cryptography and Authentication in Body Area Networks," *IEEE Trans. Info. Tech. in Biomedicine*, vol. 16, no. 6, Nov. 2012.
[3] M. Shin, "Secure Remote Health Monitoring with Unreliable Mobile Devices," *J. Biomed. and Biotech.*, 2012.
[4] P. Urien and G. Pujolle, "EAP Smart Card Protocol," IETF Internet Draft 26, 2014.
[5] Java Card Technology, http://www.oracle.com/technetwork/java/javame/javacard/overview/getstarted/index.html.
[6] NFC Forum, http://nfc-forum.org/.
[7] Juniper Research, "Mobile Ticketing for Transport Markets. Airlines, Rail, Metro & Bus 2011–2015," Juniper Research, Sci. Rep., 2011.
[8] "Identification Cards — Integrated Circuit Cards — Part 4: Organization, Security and Commands for Interchange," ISO/IEC 7816-4, 2005.
[9] J. Torres, A. Izquierdo, and J. M. Sierra, "Advances in Network Smart Cards Authentication,Computer Networks," *Computer Networks: The Int'l. J. Computer and Telecommun. Networking*, vol. 51, no. 9, 2007, pp. 2249–61.
[10] W. Simpson, "The Point-to-Point Protocol (PPP)," RFC 1661, 1994.
[11] B. Aboba *et al.*, "Extensible Authentication Protocol (EAP)," RFC 3748, 2004.

[12] F. Stajano and R. Anderson, "The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks," *Security Protocols Wksp*., 1999, pp. 172–94.
[13] Q. Sheng *et al.*, "Ubiquitous RFID: Where Are We?," *Int'l. J. Info. Sys. Frontiers*, vol. 12, no. 5, 2010, pp. 485–90.
[14] J. Torres Márquez, *Nuevo marco de autenticación para tarjetas inteligentes en red. Aplicación al pago electrónico en entornos inalámbricos*, Univ. Carlos III, Madrid, Ph.D. thesis, 2006.
[15] ISO/IEC 14443, "Identification Cards — Contactless Integrated Circuit Cards — Proximity Cards — Part 4: Transmission Protocol," 2008.
[16] F. X. Standaert, "Introduction to Side-Channel Attacks," *Secure Integrated Circuits and Sys.*, 2010, pp. 27–42.

Biographies

Fidel Paniagua Diez (fidel.paniagua@uc3m.es) received a B.Sc. in computer enginnering from Carlos III University of Madrid in 2011. He is currently working at Evalues (IT Security Evaluation Laboratory), a laboratory inside of Carlos III University of Madrid, toward a Ph.D. in computer security. This Ph.D. is part of the research project SAVIER supported by Airbus Defense and Space. His research is focused on access control models. Previously, his research was mainly centered on designing, developing, and evaluating secure communication systems.

Diego Suárez Touceda (diego.suarez@uc3m.es) is a security consultant and researcher at IT Security Evaluation Laboratory (Evalues). He has a Ph.D. in telecommunications engineering, CISSP, and CEH. His work and research are focused on security architectures, network security services, access control systems, and CyberSecurity.

José María Sierra Cámara (sierra@inf.uc3m.es) is a professor in the Computer Science Department of the University Carlos III of Madrid and a visiting researcher at the Massachusetts Institute of Technology. He has a Ph.D. in computer science. His research work is centered in the area of Internet security, which at the present time he is working on and researching. He has participated in numerous R&D projects and has published articles in journals related to security in information technologies.

Sherali Zeadally (szeadally@uky.edu) is an associate professor in the College of Communication and Information, University of Kentucky. He received his Bachelor's degree and doctorate degree, both in computer science, from the University of Cambridge, England, and the University of Buckingham, England, respectively. He is a Fellow of the British Computer Society and the Institution of Engineering Technology, England.