Authentication in Ubiquitous Computing
(Extended Abstract)

Laurent Bussard and Yves Roudier

Institut Eurecom, Sophia Antipolis – France

{Bussard, Roudier}@eurecom.fr

Introduction

Invisible and ubiquitous computing aims at defining environments where human beings can interact in an intuitive way with surrounding objects. Those objects, which can be personal digital assistants, electronic rings, doors or even clothes, offer embedded chips with computation power and communication facilities and are generally called artifacts. Because virtual electronic services are embodied in artifacts, real and virtual worlds are interlaced. It has to be taken into account when defining security.

Numerous security problems have been solved in contexts involving only logical entities [4], which we will call virtual in the rest of this paper. For instance, a remote server can easily be authenticated and the rights of an entity, which knows a given private key, can be verified. This paper shows why common network security approaches are not sufficient to ensure authentication in ubiquitous computing. Asymmetric cryptography (challenge-response protocol and certificates) can be used to prove the identity or the rights of a virtual entity. Moreover, critical operations can be protected by tamper-resistant hardware. Unfortunately, this is not sufficient to authenticate artifacts that are embodied entities. Indeed, it is necessary to verify physical properties such as location. This paper focuses on a new basic security building block: the authentication of an artifact, which is based on a dedicated challenge-response protocol. This protocol is combined with standard security mechanisms to verify that an artifact has some rights and/or features.

The first section presents two man-in-the-middle attacks that can occur in ubiquitous computing. Section 2 describes related works and finally, section 3 introduces our proposal to authenticate artifacts.

1 Man-in-the-Middle Attacks in Ubiquitous Computing

It is more and more necessary to “authenticate artifacts”. When appliances offer physical services such as playing music or delivering goods or money, the user has to verify that the appliance he is holding or touching will really deliver the service. In other words, he has to authenticate the appliance. Otherwise he could pay for a service provided to someone else. When a user has to provide a secret (e.g. password, PIN-code) to an artifact or has to delegate it some rights, it is also mandatory to authenticate the artifact.

Ubiquitous Computing Man-in-the-middle attacks occur when actors, which can be artifacts or users, forward challenges and responses in order to simulate the presence of other actors. This section describes the attack and presents an example based on Point of Sale (POS) terminals. Figure 1a) shows a regular scenario in which a client plugs his credit card and uses inputs and outputs of the terminal. Even with correct security protocols and tamper-resistant point of sale terminals, a masquerade attack is possible (Figure 1b): a dummy terminal is proposed to the client and his inputs and outputs are modified before being redirected. A dummy credit card is plugged in the real terminal and acts as a proxy. Mutual authentication between the right terminal and the user’s credit card succeed but the user is not holding this right terminal. As a result, the attacker can modify the transaction without tampering with the terminal and without stealing the card. This attack, which cannot occur in virtual context, is possible because there is no way for the card to verify if it is plugged in the right terminal. In other words, there is a gap between the virtual entity that is embedded in *B* and authenticated and the artifact that is held by the user and can be *B* or *E*.

A similar attack can be mounted against a tamper-resistant appliance offering services to visitors. For example, suppose that a shop offers a discount to any customer coming frequently enough. In this shop, a short-range local transmitter broadcasts random challenges periodically. Each visitor can return his idcertificate and a challenge signed with his private key. The shop is then able to list the users that are present and that will receive the discount. Unfortunately, any visitor can forward challenges to other remote users and build a peer-to-peer location sharing system in which any member of the group can pretend to be present in order to get discounts.

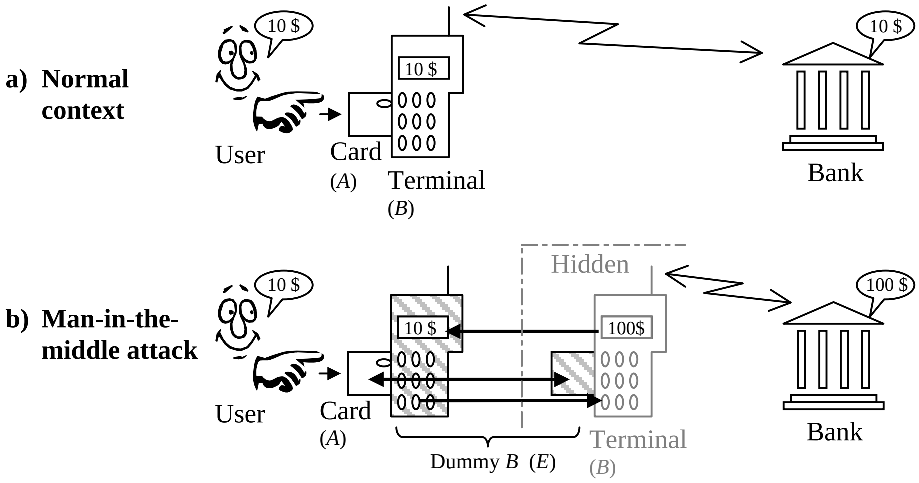

Fig. 1. An example: attack against a tamper-resistant Point of Sale terminal. In b), a dummy terminal *E* is used for the transaction when the original one is hidden. Inputs and outputs are redirected and modified.

As a result, man-in-the-middle attacks allow the impersonation of artifacts and users. It is already a relevant attack against point of sale terminals and will become more frequent when numerous micro-payments and rights delegations will occur daily within ubiquitous computing. It is necessary to defeat that kind of attack.

2 Related Works

Nomadic users generally carry a small trusted device (e.g. smart card or cell-phone) that cannot offer all necessary virtual and physical services. Thus it is necessary to rely on services proposed by the surrounding environment. Different approaches deal with virtual services in ubiquitous computing:

Discovery approaches such as JINI and Salutation [6] fit well virtual services but are not adapted to physical services. For example, it is not intuitive enough to select a printer in a list when it stands in front of the user and could be touched. Moreover, when security is required to print a confidential document or retrieve money, it does not offer a way to verify that the selected device is the expected one.

It is not possible to avoid physical services because trusted mobile devices cannot deliver cash print document, or heat a room. Before ordering cash, it is necessary to verify that automatic teller machine (ATM) are not faked ones. Even when a client uses his own cell-phone’s keyboard and display, he has to verify that the money will be delivered at the right place. Different approaches deal with the authentication of artifacts in ubiquitous computing:

An obvious solution to verify that an artifact knows a secret is to ensure that it cannot communicate with other devices during the challenge-response. For example, ATMs isolate credit cards during the authentication process. This is difficult to implement when it is necessary to protect the environment against visitors. Indeed, it does not seem realistic to install a Faraday cage around shops. This approach is not flexible enough to fit ubiquitous computing.

It is possible to check whether an artifact knows a local physical characteristic. For instance, the Smart-its project [3] bases mutual authentication on movement patterns. Two devices that are shacked together share a specific knowledge that can be used during authentication. This approach is user-friendly but the shared data are not secret enough to be used in critical situation.

The environment can offer an infrastructure to securely locate artifacts. For example, [1] proposes security beacons that exchange data with the artifacts surrounding them. Short-range communications are used to ensure that only local artifacts receive the challenges. This approach requires a physical infrastructure and is not resistant to peer-to-peer attacks.

The next section proposes a user-friendly approach that ensures authentication of artifacts and users.

3 Local Proof of Secret

This section presents a protocol able to verify that a secret is locally known in order to forbid man-in-the-middle attacks in ubiquitous computing.

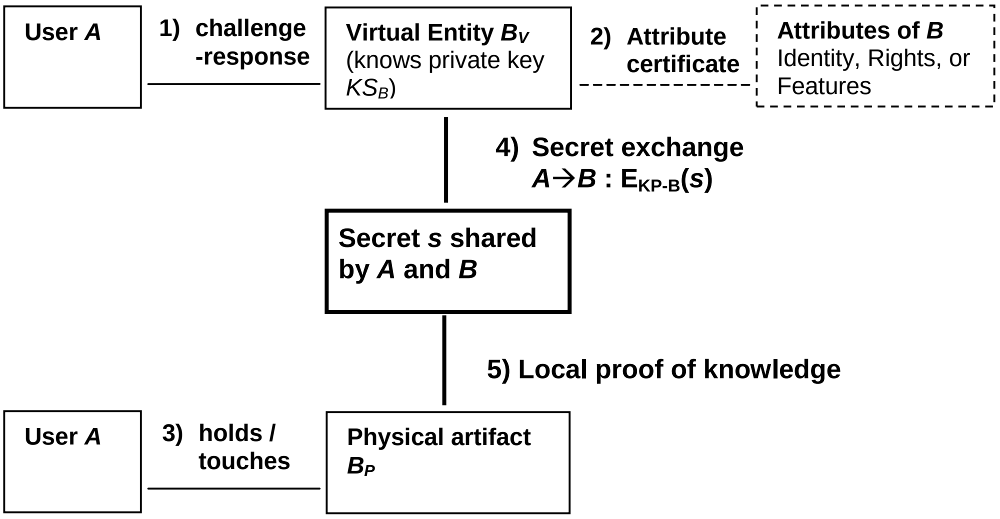
User *A*
1) challenge -response
Virtual Entity B_V (knows private key KS_B)
2) Attribute certificate
Attributes of *B* Identity, Rights, or Features
4) Secret exchange $A \rightarrow B : E_{KP\text{-}B}(s)$
Secret *s* shared by *A* and *B*
5) Local proof of knowledge
User *A*
3) holds / touches
Physical artifact B_P

Fig. 2. Local proof of secret for authenticating artifacts.

Figure 2 shows how a user *A* can authenticate (label 1) a virtual entity B_V. For instance, he can use a terminal to send a challenge to B_V and verify that the response is right (signed by the private key KS_B of *B*). A trusted third party can certify (label 2) that *B* has some properties. For example, *A* can verify and display an attribute certificate describing *B*. In the second part of Figure 2, *A* sees or touches (label 3) a physical artifact B_P. It is yet necessary to verify that the authenticated virtual entity B_V is embodied in the artifact B_P that is held by *A*. In (label 4), *A* generates a new secret and encrypts it with the public key KP_B of the entity pretending to be in front of him. Only the owner of the private key can share this secret with *A* as long as it does not disclose it. Finally, in (label 5), Our local proof of secret protocol is used to verify that the artifact knows the secret.

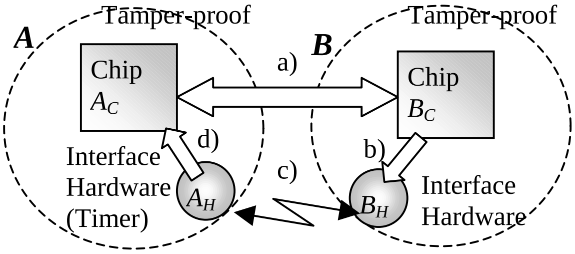

Fig. 3. Specific hardware enabling local proof of knowledge.

We propose to base the local proof of secret on a message round trip time (*RTT*) measurement. If a user could check in one nanosecond that an artifact knows a secret, it could not be farther than fifteen centimeters (due to the physical limit imposed by the speed of light). To reach such a high performance, it is not possible to rely on application layer. The exchange has to occur at the physical layer and the messages have to be as short as possible. One-bit challenges and one-bit responses are exchanged by simple dedicated hardware (logical gates). As a first step, physical contact between artifacts has been chosen because it does not require any distance measurement. Touching artifacts with an electronic ring [2] representing users, is a user-friendly way of authentication. Moreover, it fits well point of sale terminal scenarios.

Figure 3 presents the hardware architecture required to deploy fast challenge-response protocols. All involved artifact (i.e. *A* and *B*) are tamper proof modules offering computation (A_C and B_C), communication facilities (label a), and specific interface hardware (A_H and B_H). Interface hardware ensures fast exchange of two one-bit messages (label c). To avoid masquerade attacks, the response to the challenge that is not chosen has to be erased. The interface hardware is very simple and based on a few logical gates.

3.1 Protocol Description

Table 1 describes a proposal for such a local proof of knowledge. The artifact *A* want to check that it is directly in contact with a given artifact *B*. For example, a user can have to look at the rights or features of an artifact. He can use his e-ring *A* to touch the artifact *B* in order to get an authorization or attribute certificate and drop it in his PDA to view this one. A Public Key Infrastructure (PKI) is not mandatory, key management can be based on trust relationships [5]. Our local proof of knowledge protocol ensures that the received certificate corresponds to the artifact *B* that was touched by the e-ring *A*.

Table 1. Local proof of secret protocol. The bold arrow (➔) means that one-bit messages are exchanged through the dedicated interface.

```
1)  $A_C \rightarrow B_C$    $E_{KP-B}(N)$              ( Share secret $N$ with a virtual entity knowing $KS_B$)
Loop $R$ times  ($r=0..R-1$)                 with $|N| \geq 2 \cdot R$
    Init the hardware with the responses to the two possible challenges
    2)  $B_C \rightarrow B_H$        $resp_0 = N[\, 2 \cdot r \,] \in \{0,1\}$ ;  $resp_1 = N[\, 2 \cdot r+1 \,] \in \{0,1\}$
    Dedicated hardware exchange two bits (challenge and response)
    3.1) $A_H$              Randomly chooses a one-bit challenge  $j \in \{0,1\}$, Starts measuring $RTT$
    3.2) $A_H$ ➔ $B_H$        $j$                    } Fast exchange lasting RTT
    3.3) $A_H$ ⬅ $B_H$        $resp_j$               }
         $B_H$              Suppresses $resp_{j \oplus 1}$       (only one response can be released)
    Result verification
    4.1) $A_H \rightarrow A_C$       $j$, $resp$ ,$RTT$    (challenge $j$, received response $resp_j$, and measured $RTT$)
    4.2) $A_C$              Verifies that response $resp$ is equal to $N[2 \cdot r+j\,]$ and verify $RTT$
end Loop
```

During Step 1, the artifact *A* shares a secret with a virtual entity that pretends to be embedded in *B*. In step 2, the artifact (*B*) initializes its dedicated hardware (B_H) with the responses ($resp_0$ and $resp_1$) to the two possible challenges ($j=0$ or $j=1$) and announces that it is ready. The local proof of knowledge can begin between the two dedicated interfaces. In step 3.2, *A* sends one challenge bit to *B* that responds immediately (step 3.3) by sending the corresponding bits ($resp_0$ or $resp_1$). Dedicated hardware offers a fast response to the challenge and ensures that only one of the two responses can be delivered. Given fast logical gates and optionally fast light emitting diodes it is possible to expect a few nanoseconds round trip time. The received response and the measured time is returned by the specific hardware so that the application layer can verify their validity and choose to continue or stop the protocol (steps 4.1 and 4.2). The whole process is performed *R* times to reach an acceptably low probability of a successful attack. After executing this protocol, *A* knows that it has been in contact with *B*.

This approach also forbids “Peer-to-Peer Location Sharing” when users cannot know the shared secret. It is a realistic assumption when the secret is protected within the tamper-proof hardware that already protects the private key.

3.2 Security Evaluation

Attacks against the physical interface must be taken into account. It is necessary to use two different channels for sending 0 or 1 responses so that the round trip time can always be measured. It is not possible any more for the attacker to get the response in real time because any logical operation and any enlargement of the path increase the round trip time. Due to the required time measurement precision, the probability of a successful attack is very high. Indeed, the attacker can get one of the two possible responses before the exchange by sending randomly challenge. If the other challenge is chosen during the protocol, the attacker can also guess the value of the corresponding response.

With p_{out} the probability of having the right response bit (*$resp_0$* or *$resp_1$*) and p_{val} the probability of guessing the right response when the other challenge is asked, the probability p of a successful attack during a step is $p = p_{\text{out}}+(1-p_{\text{out}})\cdot p_{\text{val}}$. This scheme is based on Boolean challenges thus the probability of guessing which challenges will be sent is $p_{\text{out}}=1/2$. Using Boolean responses ensures that the probability of guessing a response is $p_{\text{val}}=1/2$, thus $p =3/4$. The protocol fails as soon as one response is not right. Each step is independent and thus attacks against the protocol have to be successful R times. The overall probability of successful attack is: $p_R = p^R = (3/4)^R$. For example, with $R=100$ rounds, the probability of a successful attack is thus $p_R=¾^{100}=3\cdot 10^{-13}$. The number of rounds has no impact on the precision of round trip time measurement because it is done independently during each round.

Conclusion

Standard authentication protocols cannot be straightforwardly used in ubiquitous computing environments. Mixing physical and virtual entities compels to redefining security. This paper focuses on authentication of artifacts. Possible attacks are presented and a solution based on dedicated hardware is proposed. It ensures authentication of artifacts in user-centric ubiquitous computing environments.

When physical services are provided (deliver money, print a confidential document, etc.), we assume that the user is in front of the device and can touch it. This approach allows intuitive and secure interactions between artifacts and/or human beings. It enables the verification of identity, rights and/or features of an artifact that has been touched. When users have to be localized for avoiding “peer-to-peer location sharing” attacks, the room can offer boards that have to be touched once to get the services or discounts.

This time-based solution is not restricted to contact-based approaches and we could imagine using laser or radio carriers. However, we focused on contact-based approaches for the following two reasons: Due to the high probability of successful attacks during each round, the protocol cannot be rendered fault-tolerant. And, the contact-based approach ensures that the distance between two artifacts and the resulting round trip time can be easily and statically evaluated.

Finally, this protocol is a basic security building block for defining access control ownership, non-repudiation and other high-level security features that are required in ubiquitous computing.

References

1. Kindberg, T. & Zhang, K. “Context authentication using constrained channels”. HP Labs Tech. report HPL-2001-84. 2001.
2. S.Meloan. “*Inside the Java Ring event”*. http://java.sun.com/features/1998/07/ring-project.html
3. L.E. Holmquist, F. Mattern, B. Schiele, P. Alahuhta and M. Beigl and H.W. Gellersen. “*Smart-Its Friends: A Technique for Users to Easily Establish Connections between Smart Artefacts”*, Proc. of UBICOMP 2001, Atlanta, GA, USA, Sept. 2001.
4. A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone “*Handbook of Applied Cryptography*”, CRC Press, 1996.
5. L. Kagal, T. Finin and, A. Joshi. *Trust-Based Security in Pervasive Computing Environments*. In IEEE Computer Volume 24, Number 12, pages 154-157. December 2001.
6. G.G. Richard, *Service Advertisement and Discovery: Enabling Universal Device Cooperation,* IEEE Internet Computing, vol. 4, no. 5, September/October 2000.