

Authentication in Ubiquitous Computing

Laurent BUSSARD and Yves ROUDIER
Institut Eurécom

Workshop on Security in Ubiquitous Computing
UBICOMP 2002, Göteborg Sweden, 29 Sept 2002

Security in Ubicomp

■ User-centric interactions in Ubicomp

- Intuitive interaction
- Physical entities (artifacts)

■ Security

- Rights, Delegation
- Trust, Ownership
- Non-repudiation of interactions

⇒ **Requires authentication of artifacts**

What are the differences between security in ubicomp and security in distributed systems? What are the differences between security in ubicomp and security in ad-hoc network?

What is specific to ubicomp? We understand ubiquitous computing as an environment where human beings and artifacts can interact intuitively.

What do we need?

- It is necessary to provide rights to an artifact (e.g. my electronic-ring is allowed to do some micro-payments).
- It is necessary to define ownership mechanisms.
- It is necessary to ensure non-repudiation of interactions.
- Etc.

And interactions have to stay intuitive!

To provide those security features, it is necessary to define what is the authentication of an artifact. This presentation will focus on this.

Service Authentication in UbiComp

- **Classical network security**


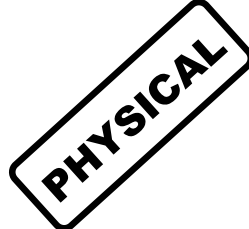
- Authentication of a virtual service
- Verify knowledge of a private key



VIRTUAL

- **Ubiquitous computing**

- Authenticate an artifact offering a service
- Provide rights to a given artifact
- Verifying that a user is present

PHYSICAL

In classical security, there are mainly two types of authentication:

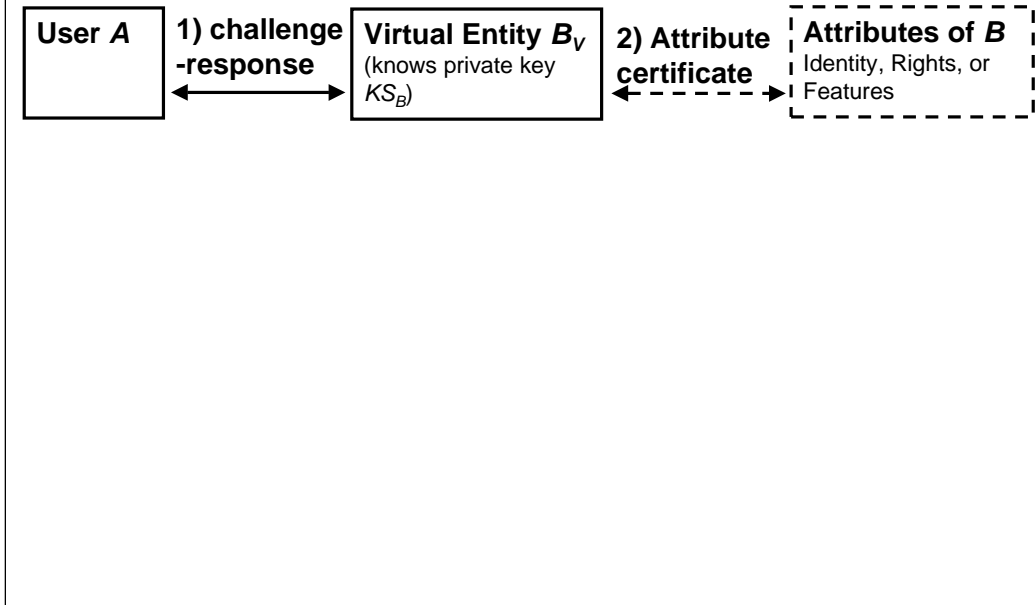
- authentication of a human being: he has to show
 - something he know: password
 - something he is: biometrics
 - something he has: token
- **authentication of virtual services**
 - it has to prove that it knows a given secret (often a private key)

Ubiquitous computing requires a new type of authentication:

- **authentication of artifacts** (delivering a physical service)
 - a physical artifact has to prove that it knows a secret.

In fact the three types are required

The Gap

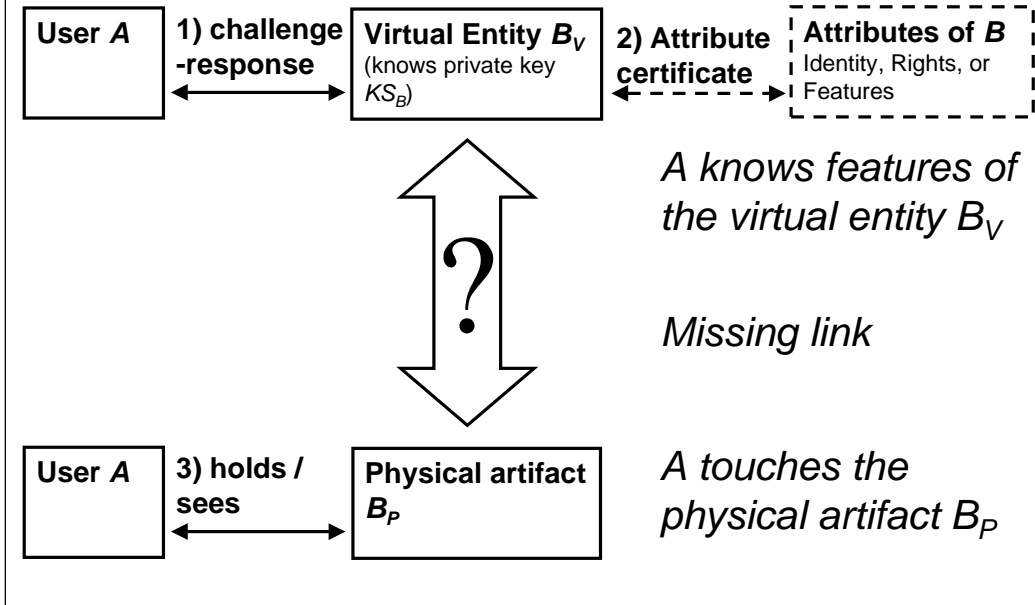


What is missing:

using a challenge-response protocol, it is possible to verify that an entity knowing a secret is involved. A certificate can associate some rights or features to this identity. For instance, when we access a banking service, each message can be authenticated (MAC).

In ubicomp we have to deal with physical services too. An artifact can provide inputs, outputs, be touched, deliver goods, and so on.

The Gap



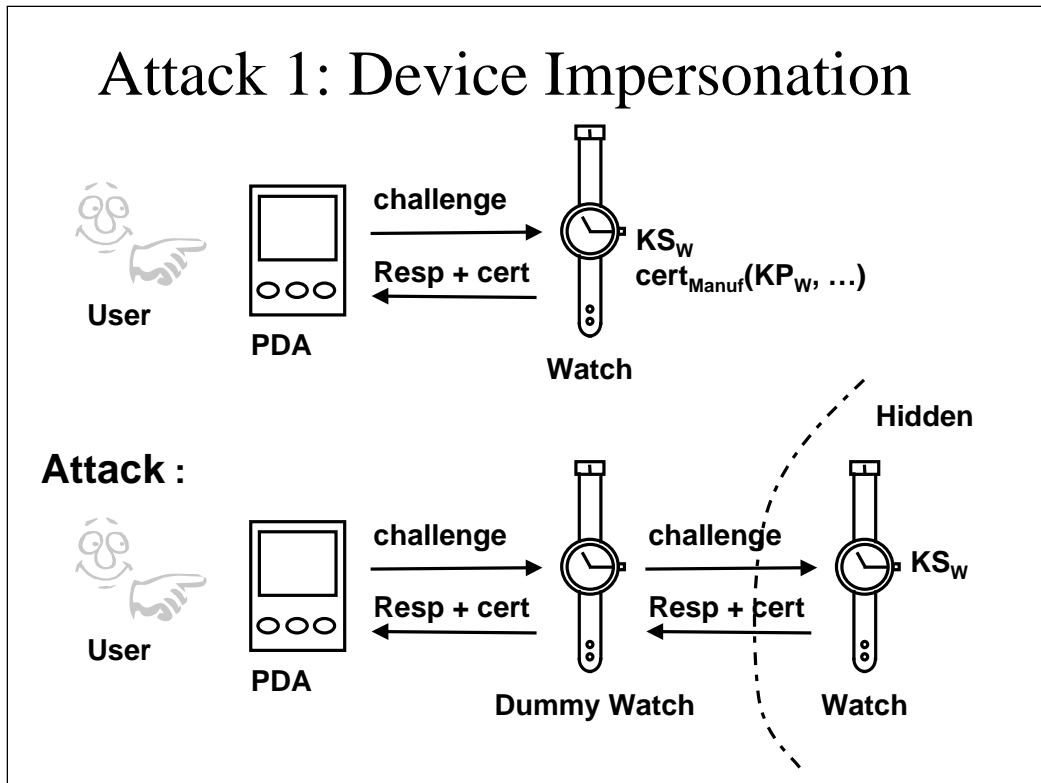
The same user A is touching an artifact but has no way to verify whether the authenticated virtual entity is embodied in this one.

It is necessary to know if a virtual entity is embodied in a given artifact!

For instance I can know that there is a printer from my company somewhere but I cannot verify if it is the one I am touching.

Let's look at some examples:

Attack 1: Device Impersonation

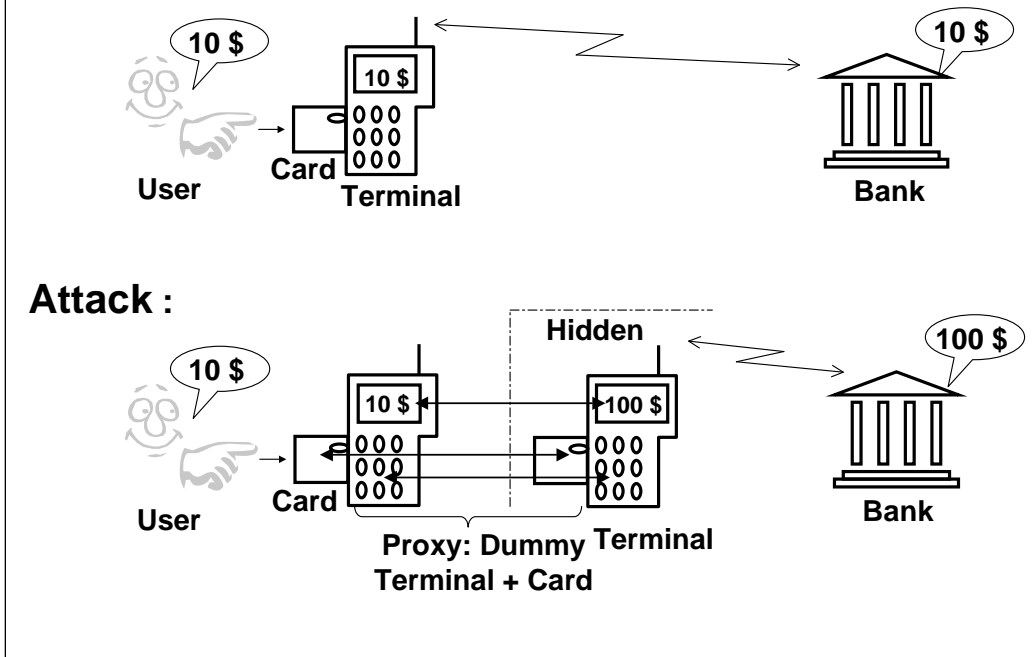


The user buys a ubicomp Swiss watch in the street. He want to verify that it is a real one. Each watch contains a tamper-resistant module protecting its private key. The manufacturer signs a certificate guaranteeing the watch knowing the private key corresponding to a given public key .

The user uses his PDA to view and verify the certificate. The verification is done through a wired or wireless media.

However, an attack can occur: a dummy watches can act as a proxy during the verification, interacting with a real watch that stays in the pocket of the seller.

Attack 2: Device Impersonation



A real world example involving two artifacts: a smart card and a point of sale terminal. The user plugs his smart card and use the keyboard and display of the terminal

Security protocols and crypto: no problem, the smart card verify that is communicating with a terminal certified by a bank, etc.

Artifacts: no problem, both are tamper-resistant.

User: the smart card does not disappear and cannot be stolen.

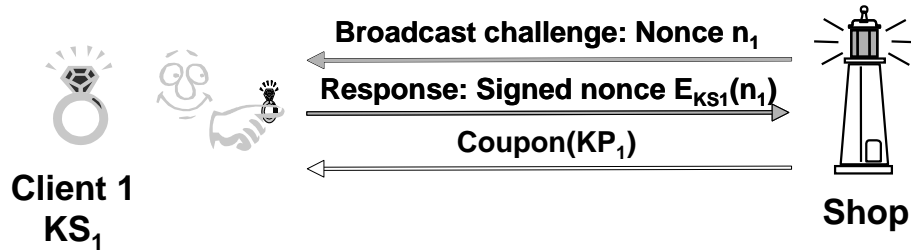
However, an attack can occur:

A dummy terminal (blue) acting as a proxy is presented to the user. The right terminal is authenticated but the user is using another one that can modify the displayed amount, get the PIN code, etc.

It is an “artifact impersonation attack”.

ATMs isolate smart cards during authentication. It is secure but not flexible enough.

Attack 3: P2P Discounts Sharing

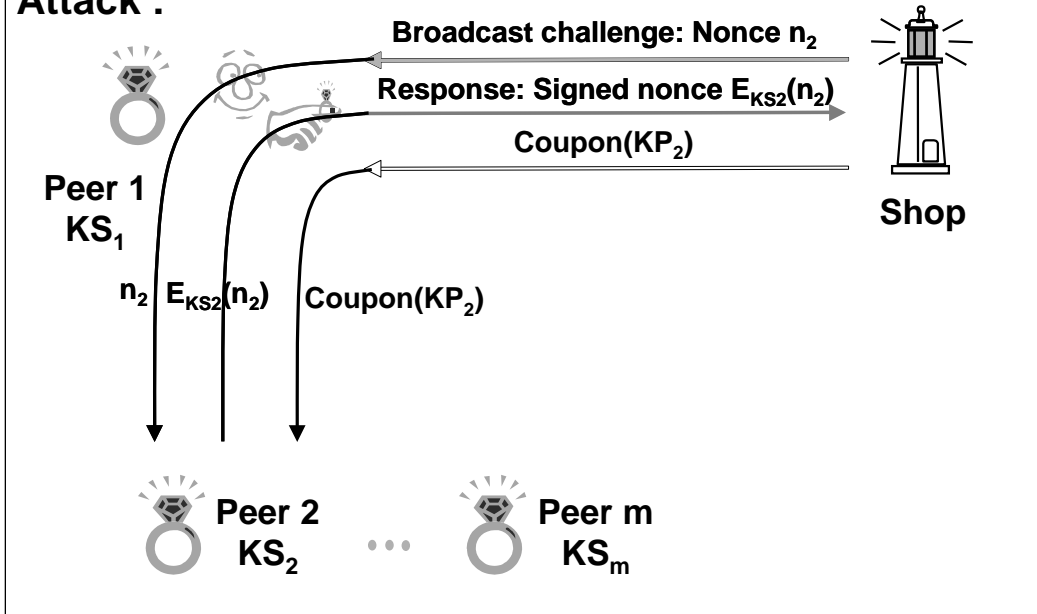


Another example: A shop can offer discounts to clients that come at least once a week.

Nonces are broadcasted to customers with a short-range media (Bluetooth, etc.). Only clients in the shop can receive this challenge and return an appropriated response.

Attack 3: P2P Discounts Sharing

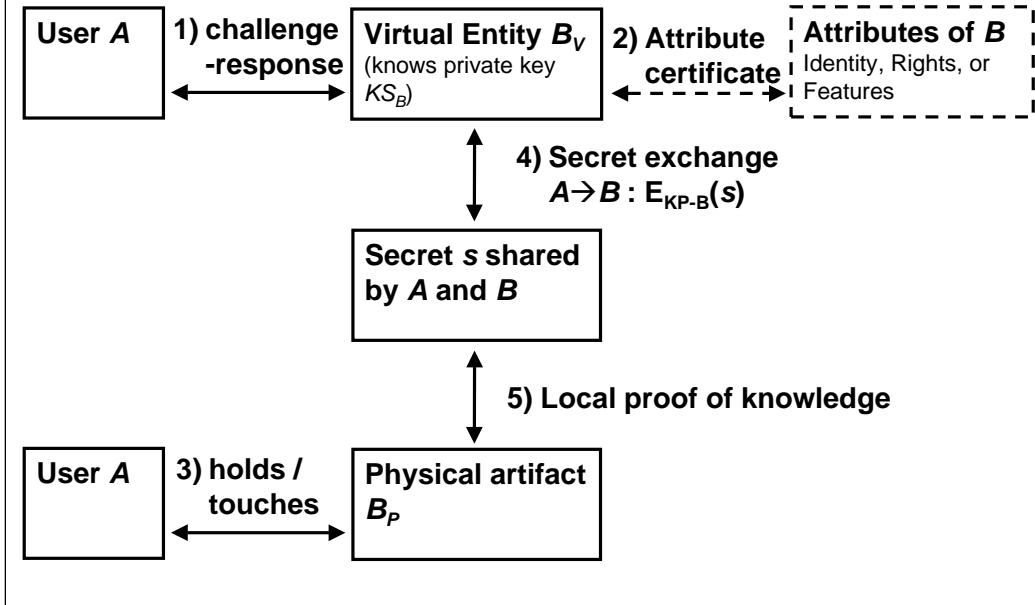
Attack :



Attack: if multiple shops offer this mechanism, group of clients can define a “Napster of discounts”. Each members are online and when a member is in a place offering discounts, he acts as a proxy and sends challenges to each interested member. So the peers can pretend being present in order to get the discount. Peer 1 will receive discounts proposed to peer 2, etc.

Once-more, isolation is not realistic, it is not possible to build a Faraday cage around each shop. We propose an approach based on time.

Filling the Gap



First, a secret is shared with the virtual entity. Next, we use our local proof of knowledge protocol to verify the location of the secret. In other words, we have a probability p that the secret is known in a given area (e.g. one cubic meter).

Local Proof of Knowledge

Time-based approach

- **Dedicated hardware**
 - No application-level approach
- **Simple distance evaluation**
 - contact based approach
- **No cryptography during exchange**
 - Responses pre-computed
- **Simple exchanges**
 - One-bit challenge
 - One-bit response

It has to be very fast

- dedicated hardware,
- static distance evaluation
- no crypto during the protocol.
- simple one-bit exchanges,

Local Proof of Knowledge

- Provides one-bit responses r_0 and r_1 to the dedicated hardware.



Uses the secret to Generate two one-bit responses r_0 and r_1

A share a secret with the entity B and want to verify that it is touching this one.

Two responses r_0 and r_1 are generated from the shared secret.

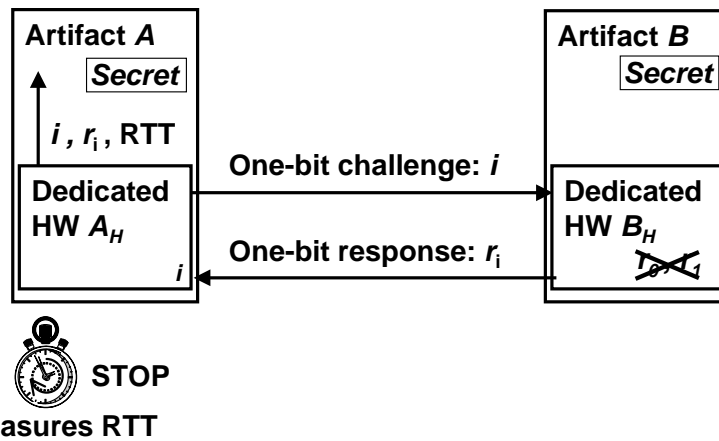
B provides those pre-calculated responses to its hardware.

A choose randomly a challenge (0 or 1)

The local proof of knowledge can start

Local Proof of Knowledge

- Start the fast challenge-response



The dedicated hardware of A starts to measure RTT.

It sends the one-bit challenge to B.

A receives within a few nanoseconds the response and stop the measure.

Finally, the application layer gets the challenge, the response, and the measured RTT in order to verify them.

Local Proof of Knowledge

- **No more Man-in-the-middle attacks**
 - No proxying in between (distance + logic)
 - Cannot get both responses
- **One bit challenge-response**
 - Precise location
 - High probability of successful attack $p = 3/4$
- **Multiple rounds (n)**
 - Precise location
 - Low probability of successful attack $p_n = (3/4)^n$

One bit challenge-response

Short RTT --> few meters (10 ns --> 1,5 m)

High probability of successful attack $p = 3/4$

After one round the authenticated entity has a 25% probability of being the embodied in the artifact that is touched.

Requires multiple rounds (n)

No impact on RTT measured

Probability of successful attack $p_n = (3/4)^n$

$n = 200$ --> $p_n = 10^{-25}$

Man-in-the-middle attacks can't work any more

No proxy in between (distance + logic)

Cannot get both responses

Conclusion: Impact on Usability

Tamper resistance + cryptography not sufficient

- **Changes in previous examples**
 - Point of Sale Terminal: LED on smart card
 - Shop offering discounts: board
- **New user-centric interactions**
 - Touch to authenticate
 - Drag-and-drop
 - Touch to transfer ownership, delegate rights
- **Authentication: a building block for developing**
 - Access control
 - Ownership

To conclude, we have seen that standard security protocols and tamper-resistant artifacts are not sufficient to define authentication in ubiquitous computing.

Our “local proof of knowledge” has to be used to verify that an a secret in locally known.

When it is required (physical and critical services), it has a big impact on usability because it relies on contact or distance measurement.

- In the first example, our dedicated hardware has to be added to both smart card and POS terminal. Moreover the smart card needs a LED to warn the user when there is a problem (do not enter the PIN code, ...).
- In the second example, the shop could offer a board that has to be touched by the electronic rings of users in order to get discounts.

Other types of interaction can be defined. For example we can drag a certificate from an artifact to a PDA in order to know the characteristics of this artifact.

Relationships (use, own, etc.) can be securely and dynamically created between artifacts.

We are currently working on new types of capabilities to define rights in such environments.

Questions ?

Contact: Laurent BUSSARD
bussard@eurecom.fr