

Privacy-Compliant RFID Authentication Protocols

Gildas Avoine

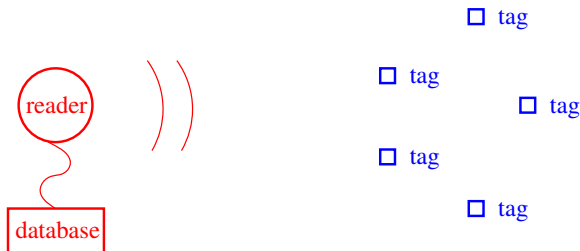
MIT, USA

<http://www.avoine.net>



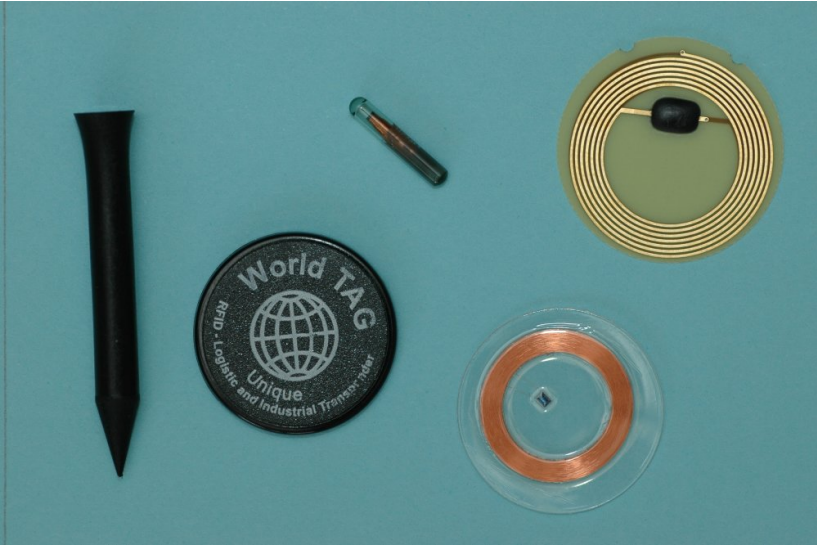
RFID and Authentication? Er... What's that?

RFID is a technology to identify (**authenticate** in our case) an object or a person, remotely, using a small transponder (**tag**) that is attached or incorporated to the object or person to identify.



Willingness to develop **small** and **cheap** RFID tags

Examples of Tags



- Tags are passive
- Communication range is centimeters to meters
- Small Memory
- No public-key cryptography
- Not tamper-resistant
- Tags answer without the agreement of their bearers
- Tags cannot be switched-off

Authentication in Daily Lives



Proving to somebody else (**verifier**) that you are the person you claim to be (**prover**).

Verifier

Prover

← Identifier, Password

```
graph LR; Prover -- "Identifier, Password" --> Verifier;
```

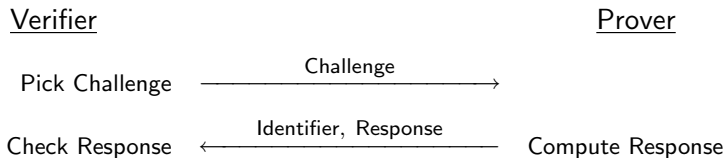
Verifier

Prover

← Identifier, Password

Do not resist to replay attacks!

Authentication by Challenge/Response



- Authentication **without revealing the identifier**.

- Authentication **without being tracked**.

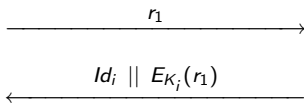
(Given a set of interactions between provers and a verifier, an adversary must not be able to find any correlation in this set)

- **Secure** i.e. prover authenticated iff prover knows secret
- Resistant to **compromized keys**
- **Efficient** i.e. reasonable amount of computation (prov/verif)
- **Lightweight** i.e. implementable on low-capabilities provers
- **Privacy-compliant** (identifier hidden, non-traceability)

Efficient Challenge-Response Protocols

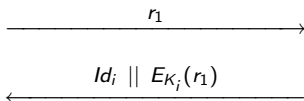
Verifier (Id_i, K_i)

Prover (Id_i, K_i)



Verifier (Id_i, K_i)

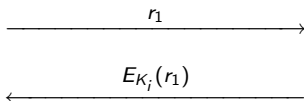
Prover (Id_i, K_i)



database	
Id_1	K_1
Id_2	K_2
\vdots	\vdots
Id_i	K_i
\vdots	\vdots
Id_n	K_n

Verifier (Id_i, K_i)

Prover (Id_i, K_i)

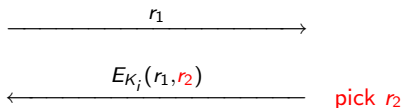


database	
Id_1	K_1
Id_2	K_2
\vdots	\vdots
Id_i	K_i
\vdots	\vdots
Id_n	K_n

- The verifier must **check every key** of the database (exhaustive search).

Verifier (Id_i, K_i)

Prover (Id_i, K_i)

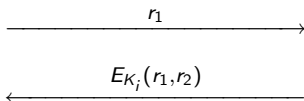


database	
Id_1	K_1
Id_2	K_2
\vdots	\vdots
Id_i	K_i
\vdots	\vdots
Id_n	K_n

- The verifier must **check every key** of the database (exhaustive search).
- The prover must **randomize** his response.

Verifier (Id_i, K_i)

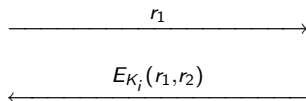
Prover (Id_i, K_i)



database	
Id_1	K_1
Id_2	K_2
\vdots	\vdots
Id_i	K_i
\vdots	\vdots
Id_n	K_n

Verifier (Id_i, K_i, s_i^1)

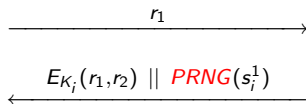
Prover (Id_i, K_i, s_i^1)



database		
Id_1	K_1	s_1^1
Id_2	K_2	s_2^1
\vdots	\vdots	\vdots
Id_i	K_i	s_i^1
\vdots	\vdots	\vdots
Id_n	K_n	s_n^1

Verifier (Id_i, K_i, s_i^1)

Prover (Id_i, K_i, s_i^1)



database		
Id_1	K_1	s_1^1
Id_2	K_2	s_2^1
\vdots	\vdots	\vdots
Id_i	K_i	s_i^1
\vdots	\vdots	\vdots
Id_n	K_n	s_n^1

Computations Needed to Identify one Tag

$$\begin{array}{rcccccccc} s_1^1 & \rightarrow & s_1^2 & s_1^3 & \dots & \dots & s_1^{m-1} & s_1^m \\ s_2^1 & \rightarrow & s_2^2 & s_2^3 & \dots & \dots & s_2^{m-1} & s_2^m \\ \vdots & \rightarrow & \dots & \dots & \dots & \dots & \dots & \vdots \\ s_n^1 & \rightarrow & s_n^2 & s_n^3 & \dots & \dots & s_n^{m-1} & s_n^m \end{array}$$

- Complexity is mn .
- Done during a **precalculation** phase.
- Storing the whole table is **not practicable**.

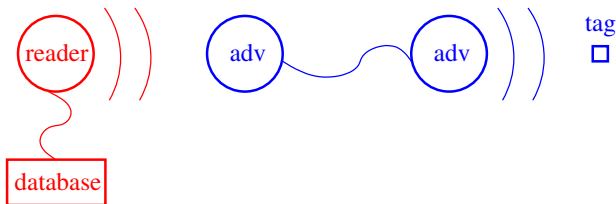
- The **general idea** of time-memory trade-offs is to reduce the time complexity of a given problem by adding memory.
- **Hellman's** trade-off (1980): $T \propto N^2/M^2$ e.g. $T = M = N^{2/3}$.
- Inverting a **one-way function** using an exhaustive search in order to find a preimage.
- Famous example: Windows **password cracking** (EPFL, 2003).

Scheme	Time (millisecond)
Basic Challenge-Response	60
With Precomputation (342 MB)	0.1
With Precomputation (1.25 GB)	0.002

$$(n = 2^{20}, m = 2^{10})$$

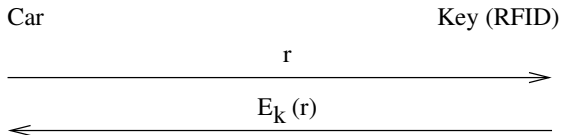
Other (Funny) Problems Related to Authentication

The reader believes that the tag is **within its electromagnetic field** while it is not the case (the attacker behaves as an **extension cord**).



Countermeasures consist in computing the **round trip time**.

Digital Signature Transponder manufactured by Texas Instrument, used in automobile ignition keys (about **130 millions** DST modules).



Cipher uses **40 bit keys**: attack in **less than 1 minute**

- 1 Recovering the cryptographic key
- 2 Impersonating the car ignition key
- 3 Impersonating the SpeedPass payment card

- Seminars April-June at UCL
- Special Issue on RFID in MISC
- RFID Security published by Springer-Verlag
- <http://lasecwww.epfl.ch/~gavoine/rfid>