

Security and Privacy Issues in RFID Systems

Gildas Avoine

EPFL, Lausanne, Switzerland

Outline of the Presentation

RFID PRIMER

SECURITY AND PRIVACY THREATS

IMPERSONATION OF TAGS

MALICIOUS TRACEABILITY

COMPUTATION COMPLEXITY

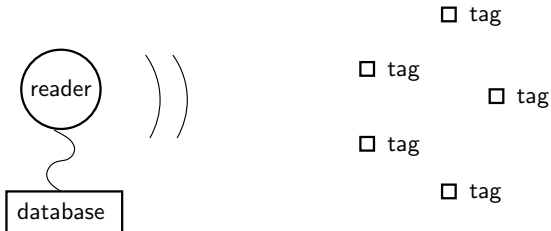
RFID PRIMER

RFID Definition and Architecture

Definition

RFID

Radio Frequency Identification (**RFID**) is a method of remotely **identifying** objects or subjects using transponders (**tags**) queried through a **radio frequency** channel.



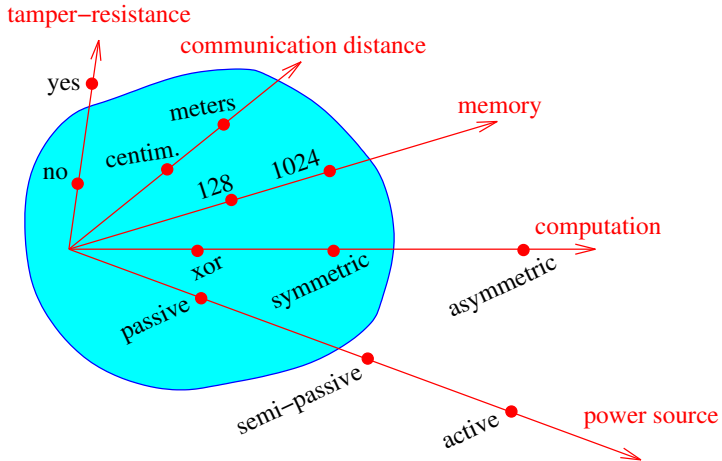
RFID Tags



RFID Readers

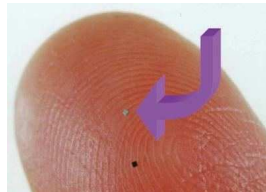


Tag Characteristics



Tag Specificities

- ▷ Tags cannot be **switched-off**
- ▷ Tags answer **without the agreement** of their bearers
- ▷ Increasing of the **communication range**
- ▷ Tags can be almost **invisible**



Daily Life Examples

- ▷ Management of stocks
- ▷ Libraries
- ▷ Anti-counterfeiting
- ▷ Access control
- ▷ Localization of people
- ▷ Electronic documents
- ▷ Counting cattle

SECURITY AND PRIVACY THREATS

Threat Classification – Example: Electronic Passports

- ▷ Denial of service
- ▷ Impersonation
- ▷ Information Leakage
- ▷ Malicious traceability

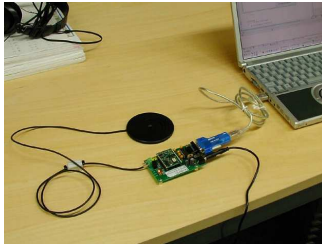


IMPERSONATION OF TAGS

Adversary

Adversary Means and Goals

The adversary can **query** the targetted tag or **eavesdrop** (RFID) communications between the tag and readers. Then the adversary tries to **simulate** the tag in front of a legitimate reader.



Identification vs Authentication

Primal goal of RFID is to provide **security**.

Definition

Authentication

The authentication consists for the reader in obtaining the identity of the tag and a **proof** that the claimed identity is correct.

Primal goal of RFID is to provide **functionality**.

Definition

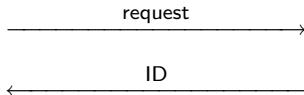
Identification

The identification consists for the reader in obtaining the identity of the tag, but **no proof is required**.

Identification Protocol

System

Tag

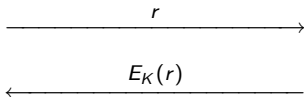


Examples: Counting cattle, localization, stock management.

Authentication Protocol

System (K)

Tag (K)



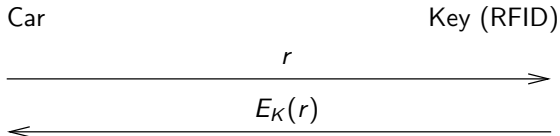
Examples: Access control, e-documents, anti-counterfeiting.

Back to Basics in Cryptography

- ▷ **Symmetric** encryption: sender and receiver use the same **secret key** both to encrypt and decrypt.
- ▷ **Asymmetric** encryption: sender uses the receiver's **public key** to encrypt. Receiver uses his own **private key** to decrypt.

Impersonation (Example: Texas Instrument DST Module)

- ▶ Attack of Bono *et al.* on the Digital Signature Transponder manufactured by TI, used in **automobile ignition key**.



- ▶ Recovering the **40-bit** key requires less than **1 minute** using a time-memory trade-off.

Recovering the cryptographic key / Impersonating the ignition key / Impersonating the SpeedPass card

MALICIOUS TRACEABILITY

Adversary

Adversary Means and Goals

The adversary can **query** the targetted tag and **eavesdrop** (RFID) communications between his target and readers. She tries to **track** people thanks to the RFID tags they carry

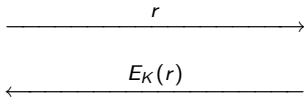
Avoiding Malicious Traceability

- ▶ The information sent back by the tag must be **indistinguishable** (by an adversary) from a random value.
- ▶ The information must be **refreshed** at each new identification.

Challenge-Response Protocol

System (K)

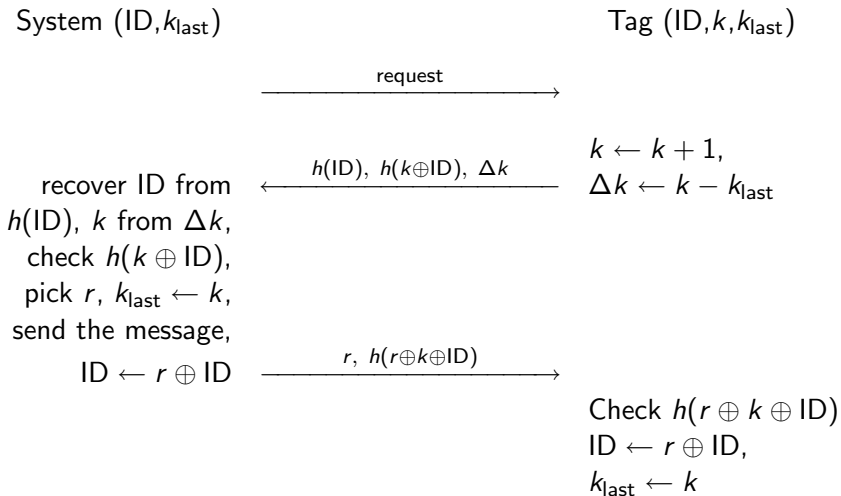
Tag (K)



Protocols

Protocol	Weaknesses pointed out by
[JuelsP03]	[Avoine04], [ZhangK05]
[VadjaB03]	[VadjaB03]
[GolleJJS04]	[Avoine05], [SaitoRS04]
[Juels04]	[Juels04]
[HenriciM04]	[Avoine05]
[SaitoRS04]	[Avoine05]
[JuelsW05]	[GilbertRS05]
[WeisSRE02]	
[OhkuboSK03]	
[FeldhoferDW04]	
[MolnarW04]	
[RheeKKW05]	

Henrici and Müller's Protocol



Attacks on Henrici and Müller's Protocol

- ▷ Attack based on lack of randomness.
 - ▶ Taking advantage of the information supplied by Δk .
- ▷ Attack based on desynchronization.
 - ▶ Desynchronizing the counters shared by tag and system.

Protocols

Protocol	Weaknesses pointed out by
[JuelsP03]	[Avoine04], [ZhangK05]
[VadjaB03]	[VadjaB03]
[GolleJJS04]	[Avoine05], [SaitoRS04]
[Juels04]	[Juels04]
[HenriciM04]	[Avoine05]
[SaitoRS04]	[Avoine05]
[JuelsW05]	[GilbertRS05]
[WeisSRE02]	
[OhkuboSK03]	
[FeldhoferDW04]	
[MolnarW04]	
[RheeKKW05]	

Feldhofer, Dominikus, and Wolkerstorfer's Protocol

System (K)

Tag (K)

\xrightarrow{r}

$\xleftarrow{\sigma = E_K(r, r')}$

find K in its
database s.t.
 $E_K^{-1}(\sigma)$ is valid

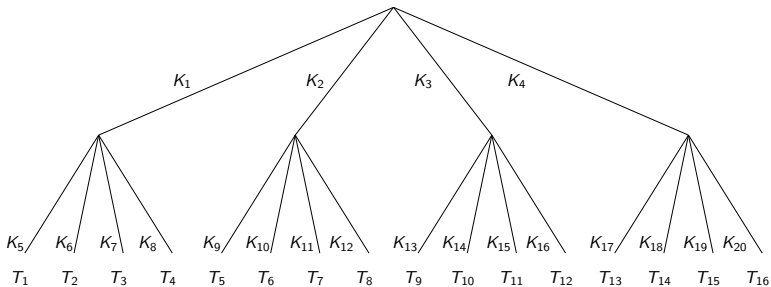
COMPUTATION COMPLEXITY

Computation Complexity of Challenge-Response Protocols

- ▷ An **exhaustive search** in the system's database is required to identify one tag.
- ▷ **Complexity too high** in particular in case of inventory.
- ▷ Is it possible to design an RFID protocol with a complexity **better than linear**?

Molnar and Wagner's Tree-Based Technique

- ▷ Each tag stores $\log_{\delta}(n)$ keys.



- ▷ A challenge-response is applied at each level of the tree.
- ▷ Instead of carrying out **1** exhaustive search in a set of size n , $\log_{\delta}(n)$ exhaustive searches are performed in sets of size δ .

Time-Memory Trade-Off

Idea

The general idea of time-memory trade-offs is to reduce the time complexity of a given problem by adding memory (or the inverse).

Example: Exhaustive search on a one-way function $S : A \rightarrow B$ in order to find preimages ($N := |A|$).

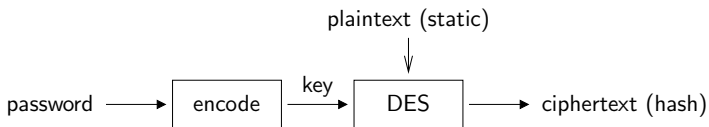
Extreme cases:

On-line computation: N Storage: 0 Pre-computation: 0

On-line computation: 0 Storage: N Pre-computation: N

Hellman's trade-off (1980): $T = N^2/M^2$ e.g. $T = M = N^{2/3}$.

Windows (LM Hash) Passwords



Pre-Computation Phase

- ▷ Invert $S : A \rightarrow B$.
- ▷ Define $R : B \rightarrow A$ an arbitrary (**reduction**) function.
- ▷ Define $f : A \rightarrow A$ s.t. $f = R \circ S$.
- ▷ Chains are generated from arbitrary values in A .

$$\begin{array}{rcccccccc} S_1 & = & X_{1,1} & \xrightarrow{f} & X_{1,2} & \xrightarrow{f} & \dots & \xrightarrow{f} & X_{1,t} & = & E_1 \\ S_2 & = & X_{2,1} & \xrightarrow{f} & X_{2,2} & \xrightarrow{f} & \dots & \xrightarrow{f} & X_{2,t} & = & E_2 \\ & & \vdots & & & & & & & & \vdots \\ S_m & = & X_{m,1} & \xrightarrow{f} & X_{m,2} & \xrightarrow{f} & \dots & \xrightarrow{f} & X_{m,t} & = & E_m \end{array}$$

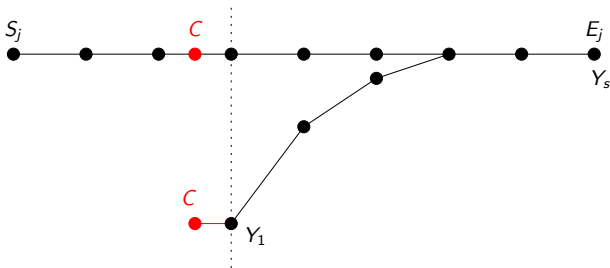
- ▷ Only the **first** and the **last** element of each chain is stored.
- ▷ The generated values should **cover** the set A .
- ▷ Time-memory trade-off techniques are **probabilistic**.

Collisions and Merges

- ▶ Collisions occur during the **pre-computation** phase.
- ▶ We use **several** tables (ℓ) with different reduction functions.
- ▶ We can use one reduction function per column (**rainbow tables**).
 - ▶ If 2 chains collide in different columns, they don't merge.
 - ▶ If 2 chains collide in the same column, the merge can be detected.

On-Line Phase

- ▷ Given one output $C \in B$, we compute $Y_1 := R(C)$ and generate a chain starting at Y_1 : $Y_1 \xrightarrow{f} Y_2 \xrightarrow{f} Y_3 \xrightarrow{f} \dots Y_s$



Cracking Windows (LM Hash) Passwords [Oechslin03]

- ▷ Cracking an alphanumerical password requires
 - ▶ a few **hours** using a classical brute force.
 - ▶ a few **seconds** using a time-memory trade-off.
- ▷ Storing the tables requires about 1 GB.

See <http://ophcrack.sourceforge.net/>

CONCLUSION

Conclusion and Current Trend

- ▶ Lightweight cryptography, human-based cryptography
- ▶ Untraceable identification/authentication protocols
- ▶ Adversary model
- ▶ Distance bounding protocols
- ▶ Side-channel attacks