An elliptic curve and zero knowledge based forward secure RFID Protocol

S. Martínez M. Valls C. Roig F. Giné J.M. Miret

{santi, magda, roig, sisco, miret}@eps.udl.es

Escola Politècnica Superior
Universitat de Lleida,
Spain

III Conference on RFIDSec-2007, Málaga

Outline

Outline

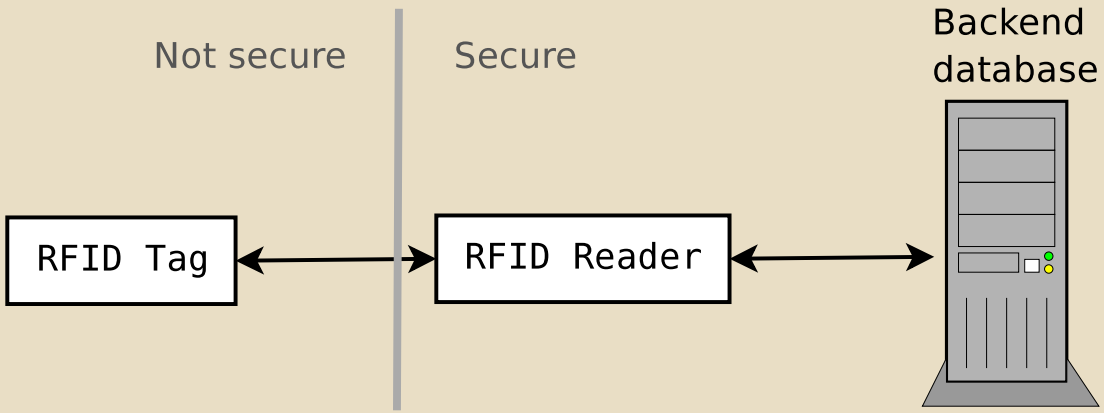
Context

- Great boom of the ubiquity of smaller hardware devices
- These devices need cryptography, but they have difficulties for the use of conventional cryptography
- Elliptic curve cryptography solves this problem allowing shorter keys for equal security

Conventional IFP & DLP	Elliptic Curves ECDLP
512 bits	112 bits
1024 bits	160 bits
2048 bits	224 bits
3072 bits	256 bits
7680 bits	384 bits

Necessary bits for same security levels

Scenario



- Communications between the reader and the database and information in the database are secure
- Communications between reader and tags may be eavesdropped
- For some security aspects it is considered that an attacker can physically read a tag

Goals

Goals

- Prevent leaking of tag information
  - Indistinguishability: output cannot be related to tag's ID
- Prevent user behaviour tracking
  - Variation of the information transmitted by the tag
- Forward security
- Scalability

Forward security

Motivation

- Some tags may incorporate a sensor
- Each sample is encrypted in function of a tag's secret
- When a tag sends its ID, it also sends a sensor value

Problem

- An attacker eavesdrops some encrypted samples
- Later, he obtains the tag, phisically reads it, obtains the secret and decrypts any eavesdropped values
- A system has *forward security* if it is capable of avoid this attack

Outline

Previous Solutions

Kill command

- Tags have a PIN protected *kill* command
- When a tag receives the killing PIN it renders itself permanently inoperative

Sleep command

- Tags have two PIN protected commands (*sleep* & *awake*)
- A tag deactivates when receives the *sleep* command, and reactivates when receives the *awake* command

Anonymous ID

- Tag output is an anonymous ID that was previously generated with cryptography or as a random value
- It allows tracking, since the identifier is fixed

External Encryption

- Tag data are encrypted using an external unit with PKC
- The identifiers seem random each rewrite, but tracking is allowed during a rewriting period

Hash Lock

- Reader has a key k for each tag, each tag stores $metaID = hash(k)$
- In a reading, the tag sends its *metaID*, the reader sends it the key and the tag verifies the hash and sends its real ID

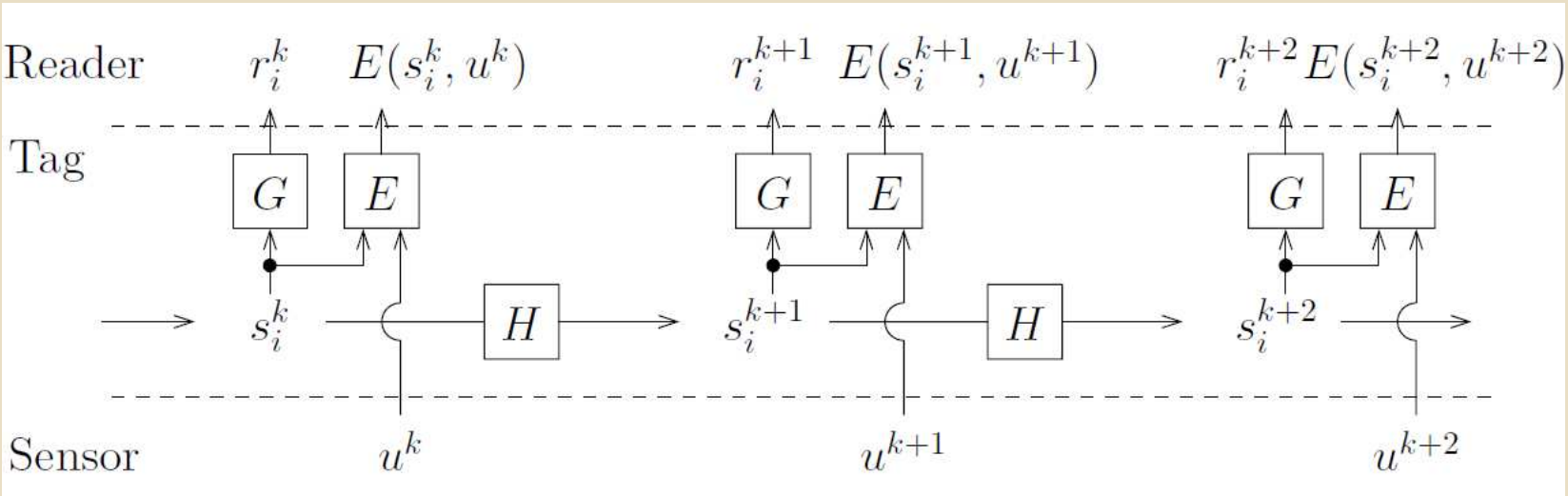
Random Hash lock

- In a reading, the tag computes $c = hash(id|r)$, where r is random and sends (c, r). The database computes the hash for all the identifiers (with r) and sends the correct *id*
- It avoids tracking, but does not have *forward security*

Hash Chains

Hash chains

- Each tag keeps a secret which may be used for sending encrypted data
- When the reader sends an identification request, the tag generates its identifier by means of a hash and changes its secret using another hash



Outline

Proposal (1)

Idea

- Maintain a similar idea to the previous one (change a tag's secret in each reading operation)
- Require the authentication of readers

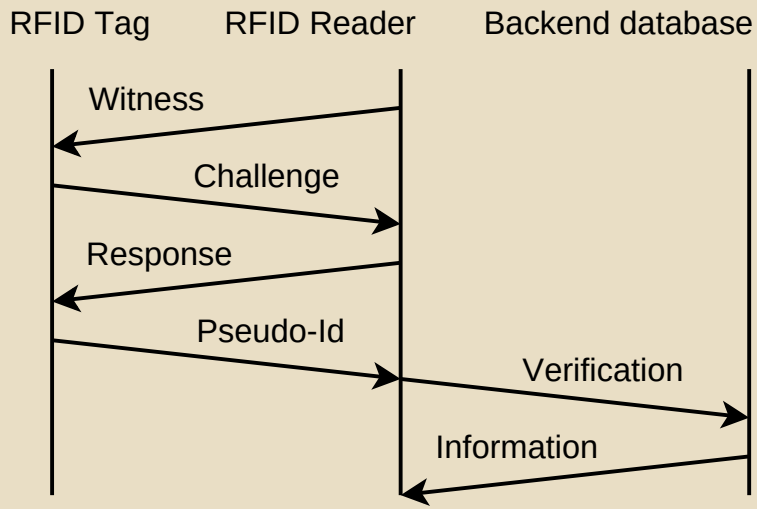
- Use a zero knowledge authentication protocol over elliptic curves (elliptic version of Schnorr's protocol)
- Use the same curve and generator of the ZKAP for changing the tag's secret

Proposal (2)

Protocol

The protocol consists of 4 phases:

- Setup phase
- Reader authentication phase
- Tag identification phase
- Tag verification phase



Protocol: Setup

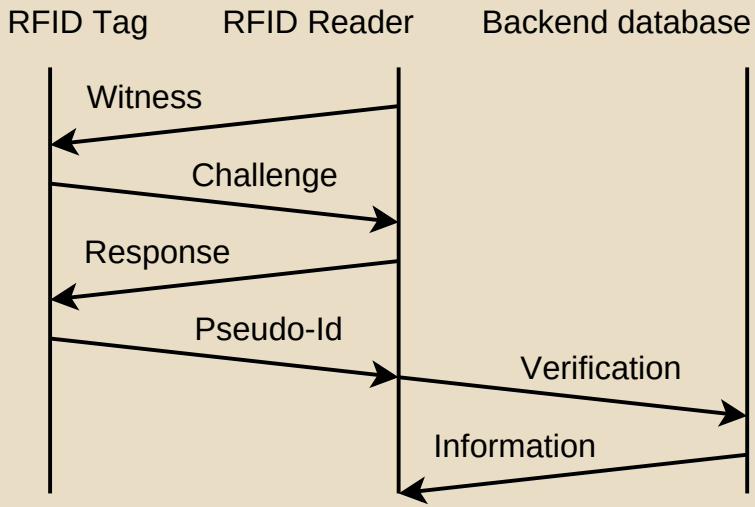
Setup phase

Choose:

- Finite field $\mathbb{F}_q$ and an elliptic curve over this field $E(\mathbb{F}_q)$
- Generator Q of a cyclic subgroup of points of the elliptic curve
- Secret key $s \in [2, \#Q - 1]$ for the reader
- Public key $P \in E(\mathbb{F}_q)$ of the reader, so that $P = sQ$
- An initial secret point $K_1^i \in E(\mathbb{F}_q)$ for each tag T_i

Protocol: Reader Authentication

Reader authentication phase

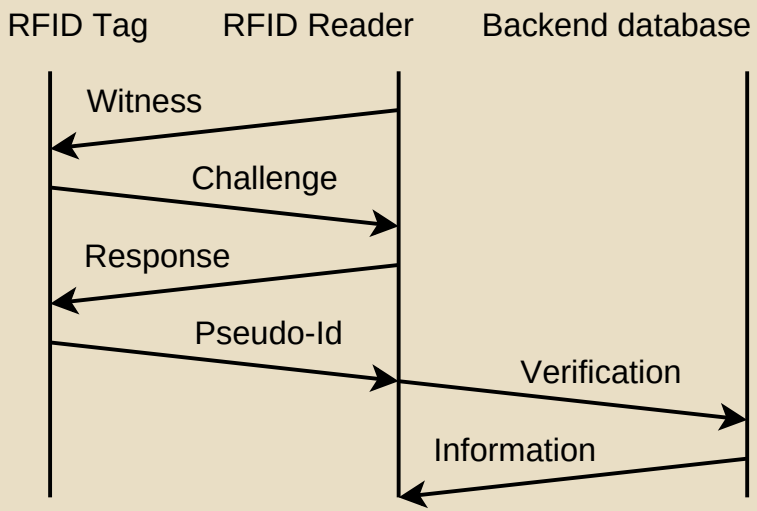


- The reader chooses a random value r (*commitment*)
- The reader computes $W = rQ$ (*witness*) and sends it
- The tag chooses a random c (*challenge*) and sends it
- The reader computes $a = r + cs$ (*response*) and sends it

The tag will accept that the reader is valid if $aQ - cP = W$

Protocol: Tag Identification

Tag identification phase



- The tag computes $id_j^i = LastBits(x(K_j^i)\ bxor\ y(K_j^i))$ (*pseudo-id*)
- The tag computes its next secret point $K_{j+1}^i = x(K_j^i) \cdot Q$
- The tag stores its new secret point
- The tag sends its id_j^i to the reader

Protocol: Tag Verification

For each tag, the database contains a record with the next *pseudo-id*, the current secret, and the product information.

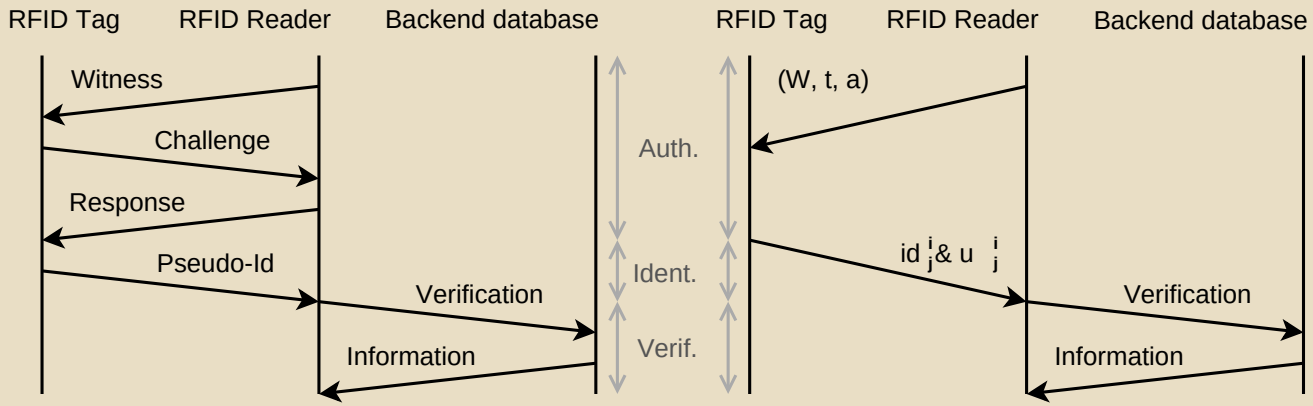
Tag verification phase

- The reader sends the received *pseudo-id* to the database
- The DB searches it in a hash table and extracts the product information
- Then, it changes the secret point in the same manner that the tag does
- Finally, reinserts the information using the next *pseudo-id*
- And sends the product information to the reader

Protocol modifications

Modifications

- The zero knowledge protocol may be done non-interactively, reducing the number of messages. The reader sends an unique message with the *witness*, a *timestamp* and the *response*; the *challenge* is computed by both parts applying a hash to the *witness* and the *timestamp*
- The tag may send values v (e.g. of a sensor) in a secure way using the secret point: $u_j^i = v_j^i \; bxor \; FirstBits(y(K_j^i))$



Outline

Basic Types of Attacks (1)

Sniffing

- The attacker eavesdrops the communications between a reader and a tag
- An sniffing attack is useless due to the use of a zero knowledge protocol in combination with the pseudo-id and the encryption of the sensor data

Tracking of the tags

- Tracking of the behavior of the owner of a tag
- A pseudo-id sniffed at a moment, cannot be related with the information obtained before and the same applies to the encrypted sensor data

Basic Types of Attacks (2)

Spoofing: Impersonation of a reader

- The attacker tries to impersonate a valid reader
- Due to the use of zero knowledge authentication the probability of impersonation of a reader is negligible

Spoofing: Impersonation of a tag

- The attacker tries to impersonate a valid tag
- He needs the current secret K_j^i of the tag to impersonate

Basic Types of Attacks (3)

Replay attacks

- The attacker resends information that he has captured before, eavesdropping a previous session
- Only the interactive version of the protocol avoids entirely this attack

Denial Of Service

- Temporal (or permanent) incapacitation of the system or a part of it
- There is no danger of an attacker performing a denial of service attack of multiple read operations
- In some environments it may require an additional message

Forward security

This property ensures that the revelation of tag's secret will not put in danger the security of previously sent information

- An attacker physically attacks tag T_i and obtains its K_j^i
- In order to decrypt the eavesdropped encrypted sensor values, there are needed the previous secret points, and they cannot be obtained because it implies solving one ECDLP for each one he wants to obtain

Outline

Conclusions

- An efficient and secure RFID protocol has been proposed
- It is used an elliptic curve cryptography protocol with zero knowledge authentication
- It avoids leakage of tag information and tracking of the user's behavior
- The system is scalable and forward secure
- The protocol provides a secure channel between tags and readers

An elliptic curve and zero knowledge based forward secure RFID Protocol

S. Martínez M. Valls C. Roig F. Giné J.M. Miret

{santi, magda, roig, sisco, miret}@eps.udl.es

Escola Politècnica Superior
Universitat de Lleida,
Spain

III Conference on RFIDSec-2007, Málaga