Data Synchronization in Privacy-Preserving RFID Authentication Schemes

Sébastien CANARD and Iwen COISEL

Orange Labs R&D - Caen - France

RFIDSec 08 - 10th July 2008

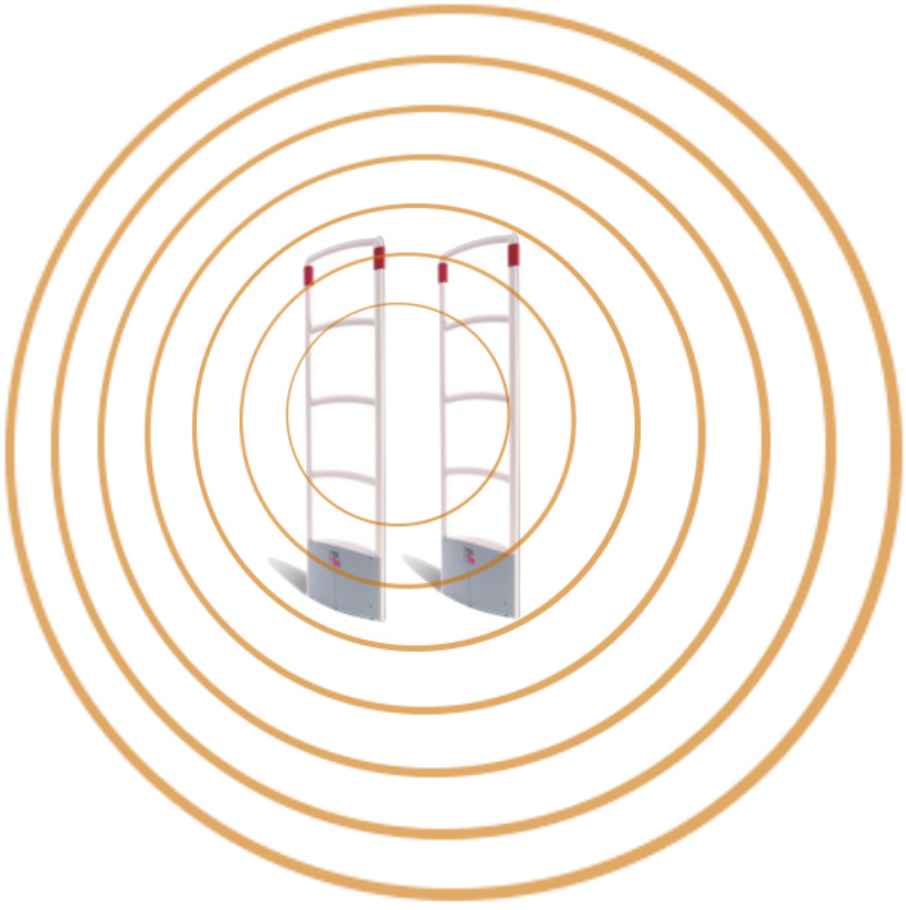
research & development

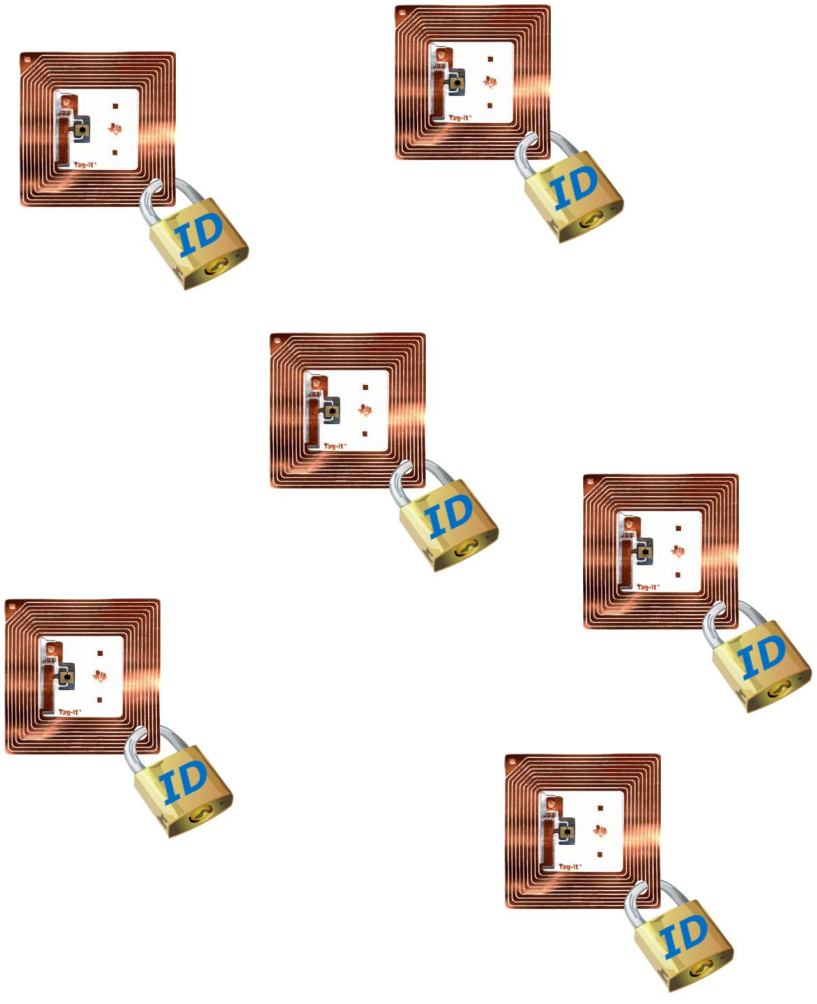
Outline

1. General Context
2. A synchronization problem
3. A New Modelization
4. The C^2 Scheme

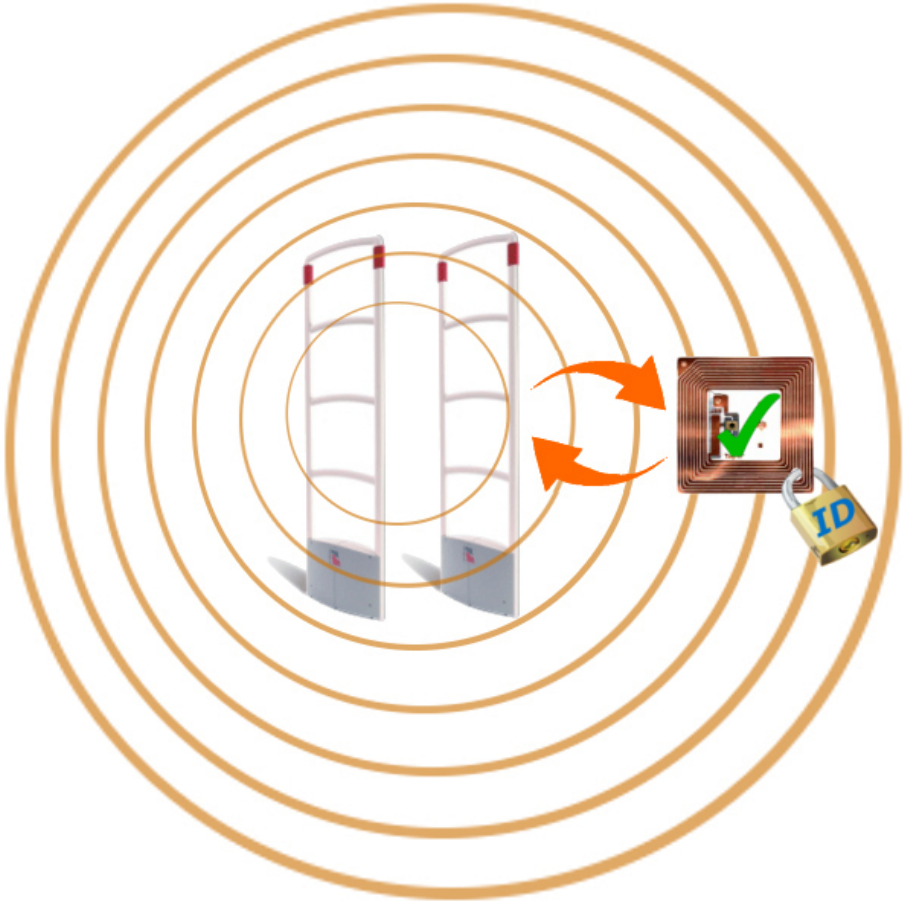
Outline

System



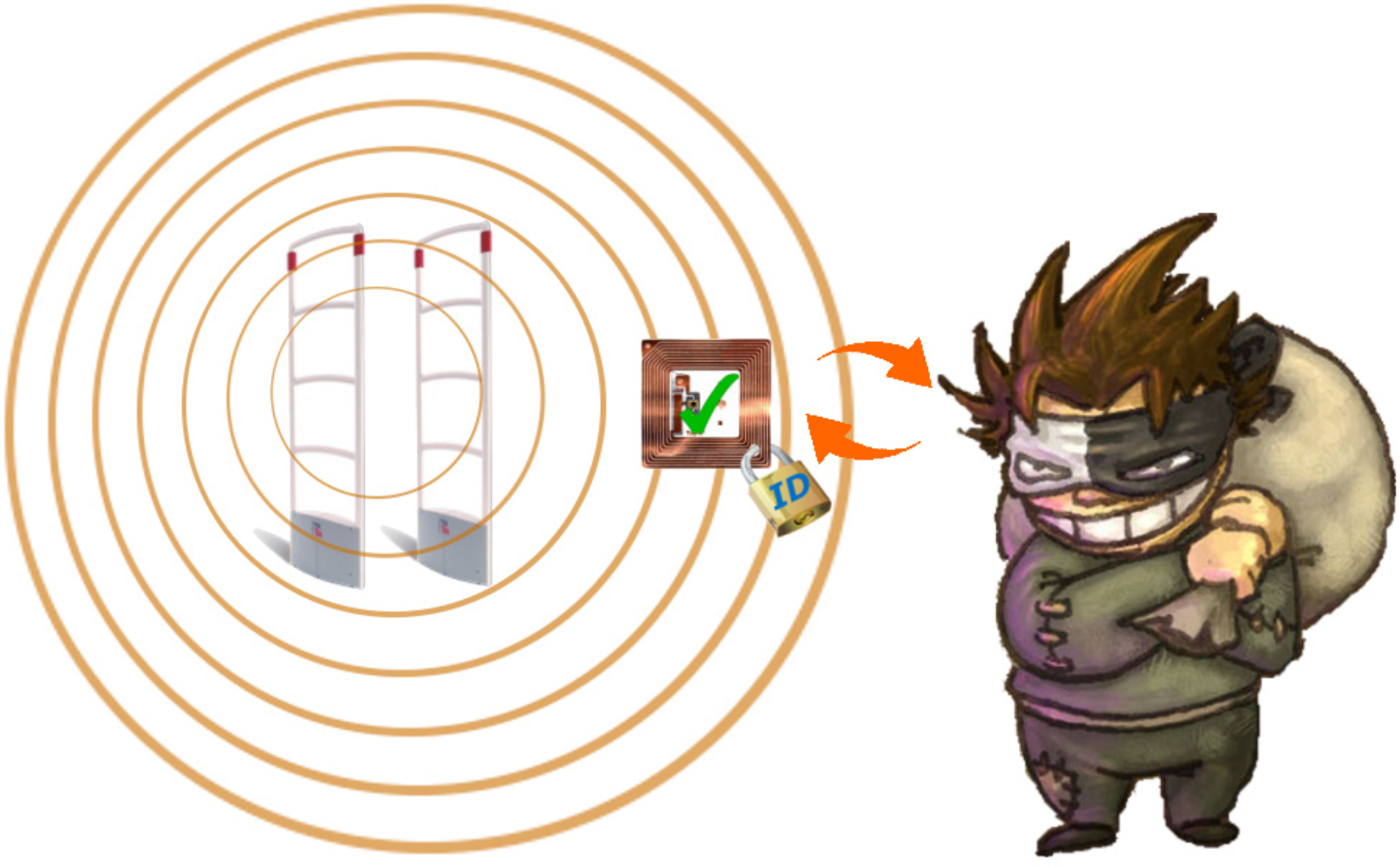


Correctness



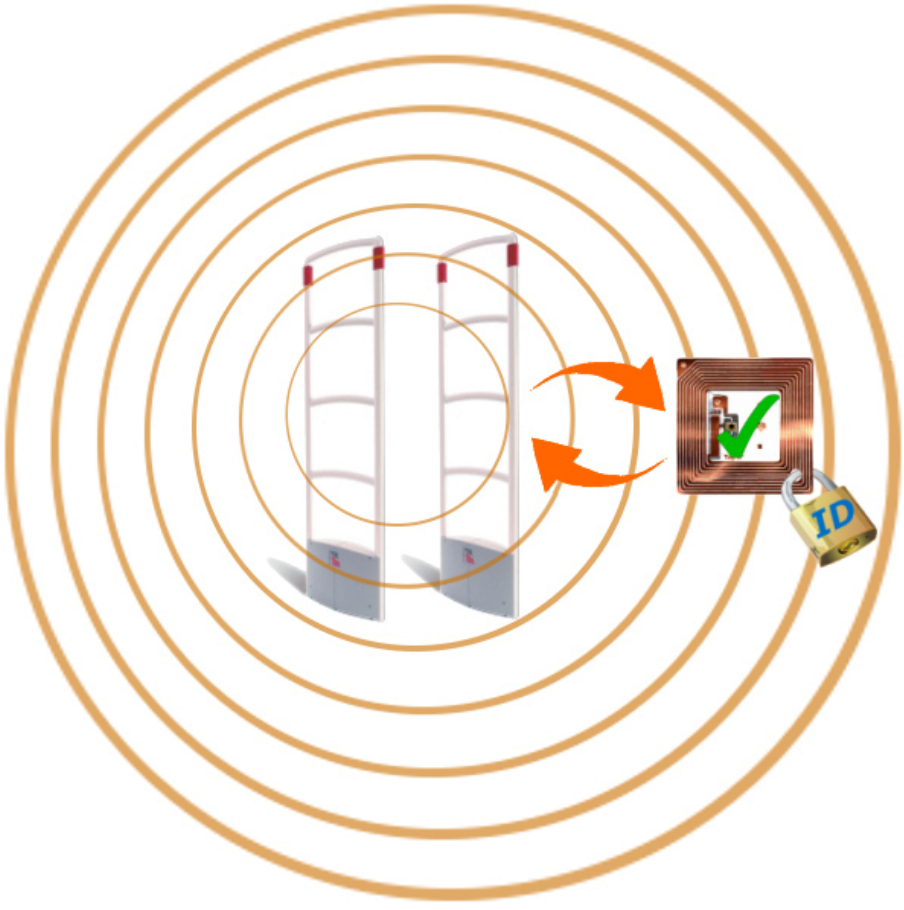
Correct : a legitimate tag is always accepted by a reader.

Strong Correctness



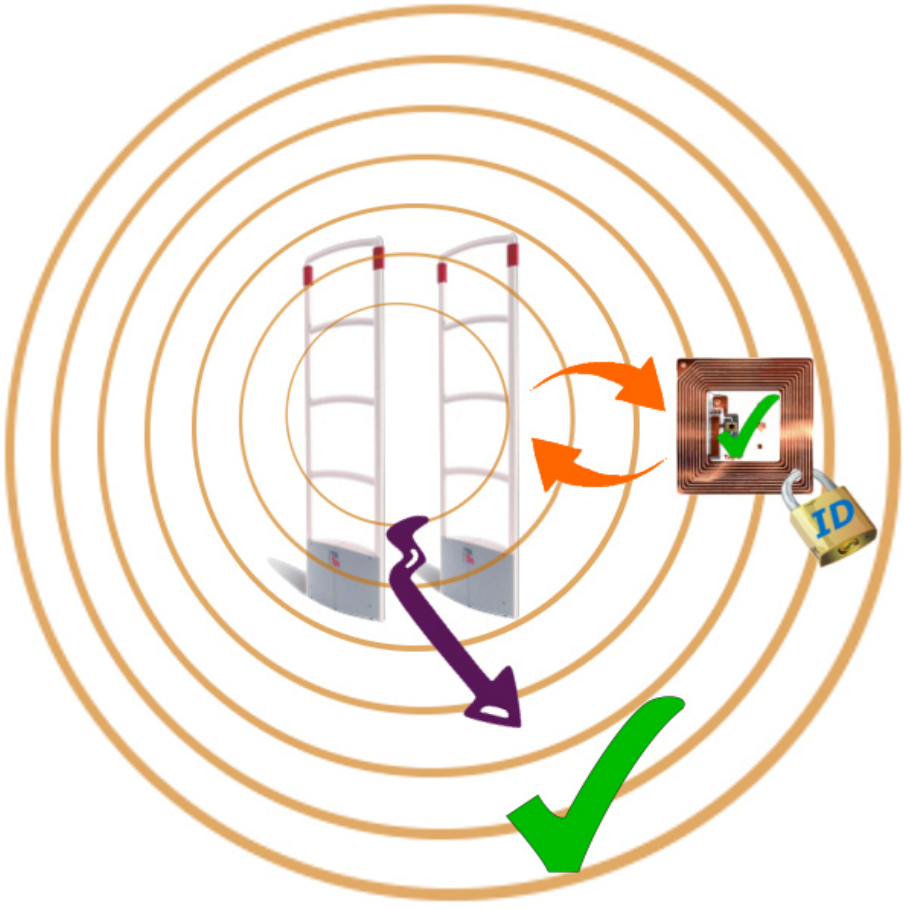
Strong Correct : a legitimate tag is always accepted by a reader, even if an adversary interacts with the system.

Strong Correctness



Strong Correct : a legitimate tag is always accepted by a reader, even if an adversary interacts with the system.

Strong Correctness

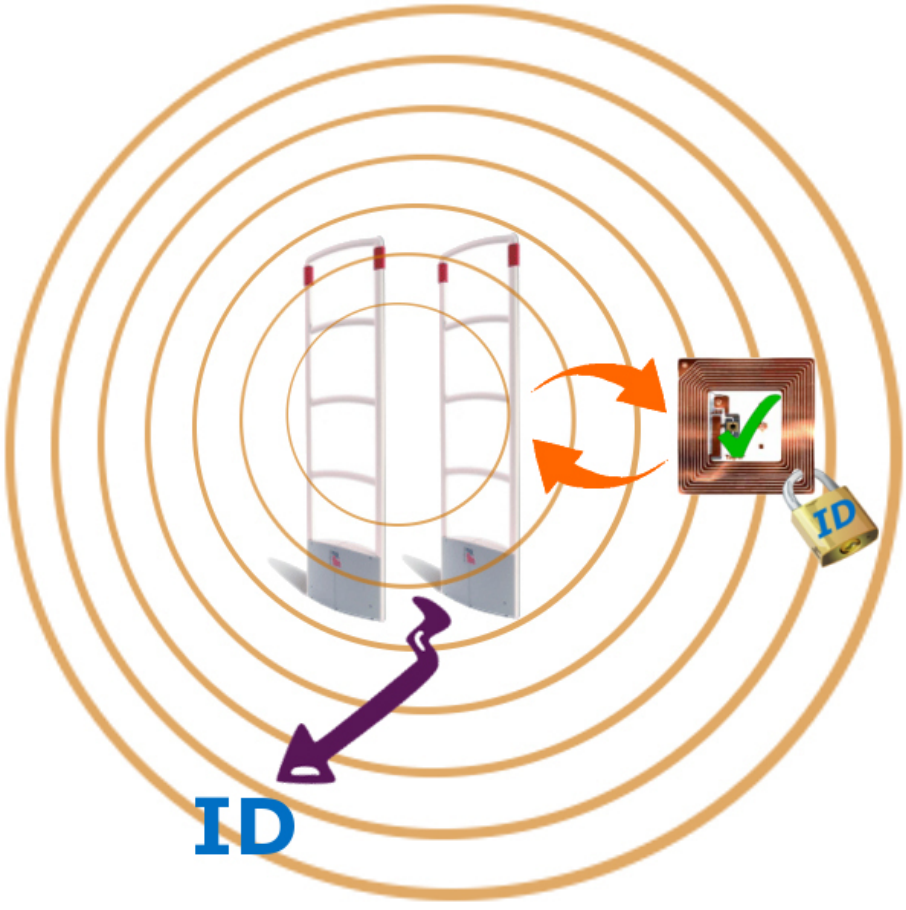


Strong Correct : a legitimate tag is always accepted by a reader, even if an adversary interacts with the system.

Soundness

Sound : an adversary should not be accepted as an uncorrupted tag by a reader.

Privacy - Anonymity

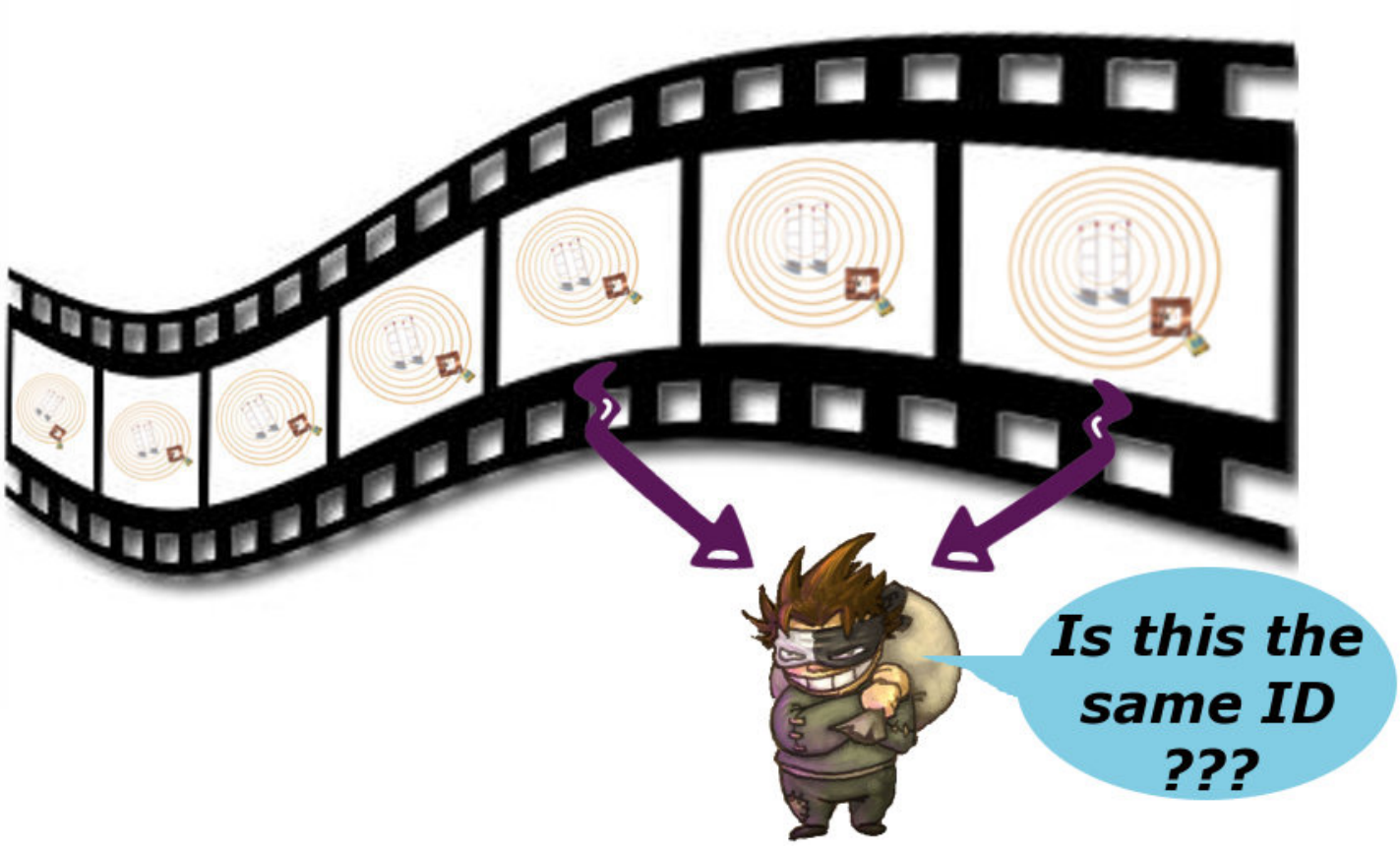


Anonymous : a tag is anonymous for everyone except the reader.

Privacy - Anonymity

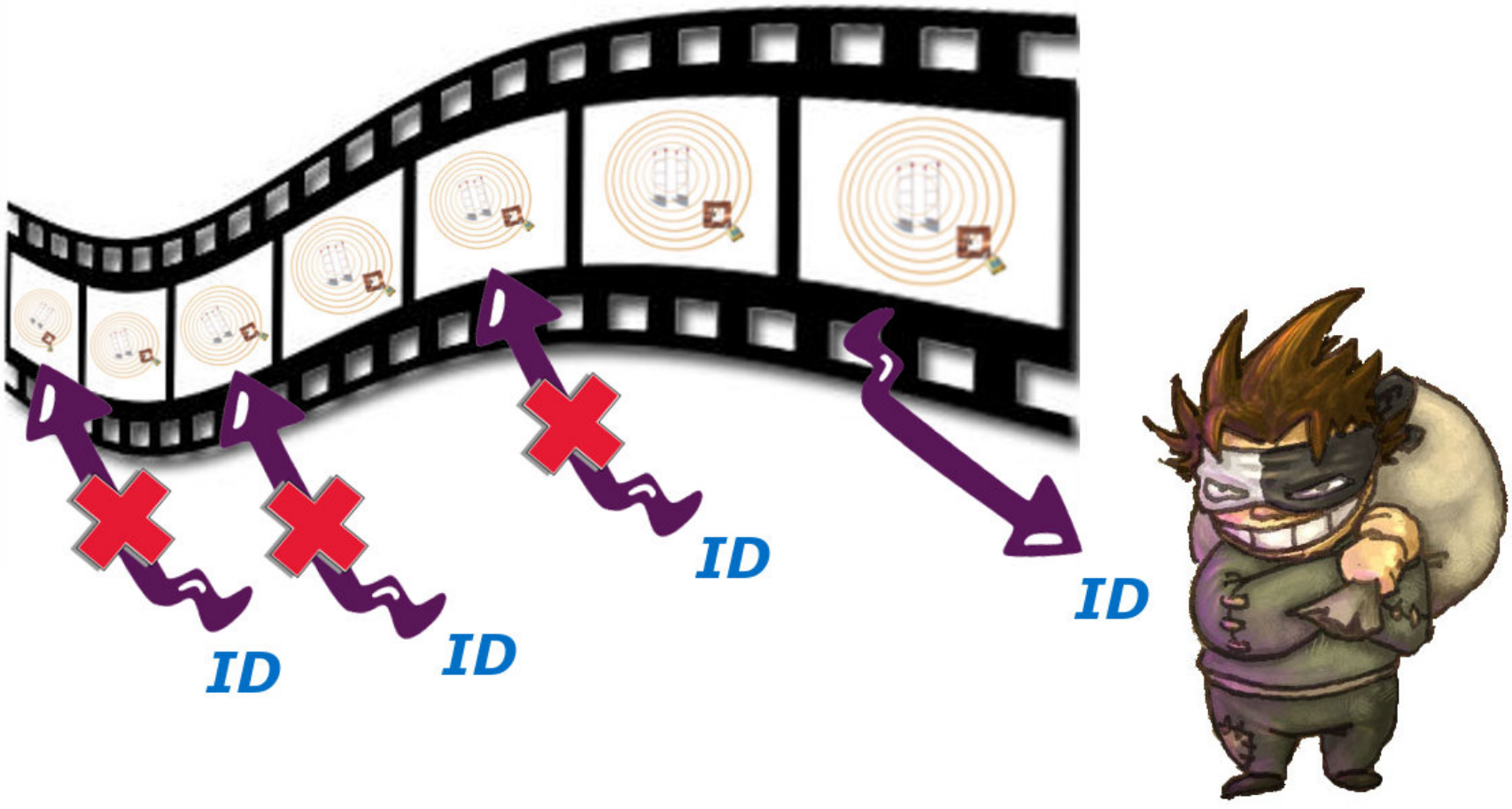
Anonymous : a tag is anonymous for everyone except the reader.

Privacy - Untraceability



Untraceable : an adversary is not able to link different authentications of the same tag.

Privacy - Forward-Privacy

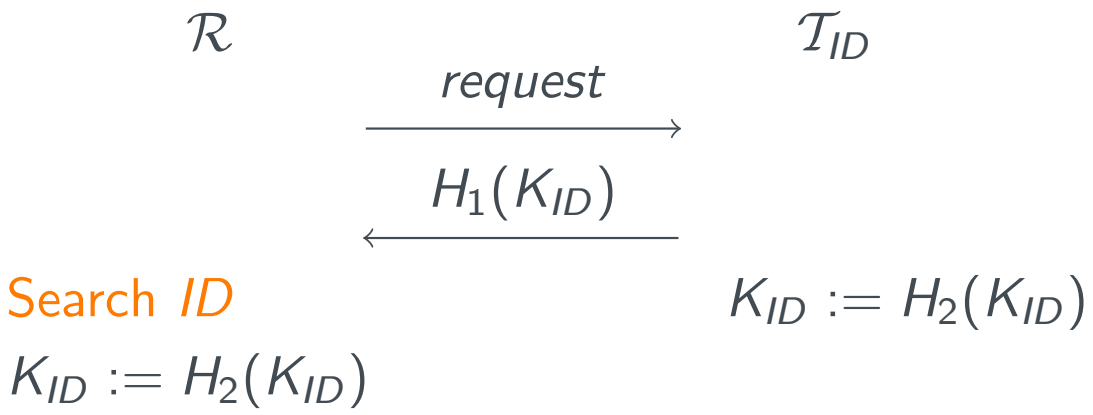


Forward-private : an adversary which obtains the secret data of a given tag is not able to recognize previous authentications of this tag.

Outline

OSK Scheme

Ohkubo, Suzuki and Kinoshita in 2003.



- Correct
- Sound
- Private

OSK Scheme

Ohkubo, Suzuki and Kinoshita in 2003.

$\mathcal{R}$ $\qquad\qquad$ $\mathcal{T}_{ID}$

$\xrightarrow{\quad request \quad}$

$\xleftarrow{\quad H_1(K_{ID}) \quad}$

Search *ID* $\qquad\qquad$ $K_{ID} := H_2(K_{ID})$

$K_{ID} := H_2(K_{ID})$

Search *ID*:

K_{ID_i}

$\downarrow$

$H_1(K_{ID_i})$

OSK Scheme

Ohkubo, Suzuki and Kinoshita in 2003.

$\mathcal{R}$ $\qquad\qquad$ $\mathcal{T}_{ID}$

$\xrightarrow{\quad request \quad}$

$\xleftarrow{\quad H_1(K_{ID}) \quad}$

Search *ID* $\qquad\qquad$ $K_{ID} := H_2(K_{ID})$

$K_{ID} := H_2(K_{ID})$

Search *ID*:

$$K_{ID_i} \xrightarrow{H_2} K_{ID_i}^{(+1)} \cdots \xrightarrow{H_2} K_{ID_i}^{(+j)}$$

$$\downarrow \qquad\qquad \downarrow \qquad\qquad \downarrow$$

$$H_1(K_{ID_i}) \qquad H_1(K_{ID_i}^{(+1)}) \qquad H_1(K_{ID_i}^{(+j)})$$

Attacks against the OSK Scheme

- An adversary can send as many requests as he wants to a tag, which consequently updates its key. Even if it takes some time, the reader is always able to resynchronize both keys.
- An adversary can answer to a request from the reader by sending a random value.

$\Rightarrow$ the search procedure "will never end".

Solutions:

- OSK_m: the search procedure stops if no match is found after m updates of each key.
- OSK-AO: the database is constructed differently (using rainbow table) inducing a faster search procedure .

Problem: these protocols are Desynchronizable .
(= a valid tag can be rejected by a reader)

Outline

Our New Modelization

The Desynchronization Value $(D_{\mathcal{R}}, D_{\mathcal{T}})$:

- $D_{\mathcal{R}}$: maximum number of times that an adversary can update the key stored in DB without updating the one stored in the tag.
- $D_{\mathcal{T}}$: maximum number of times that an adversary can update the key stored in a tag without updating the one stored in DB.

Example:

OSK, OSK$_m$ and OSK-AO:

- the reader cannot be desynchronized $\Rightarrow D_{\mathcal{R}} = 0$.
- a tag can be desynchronized indefinitely $\Rightarrow D_{\mathcal{T}} = \infty$.

Our New Modelization

Formally:

- During the strong correctness experiment, $\mathcal{A}$ interacts with the system and then chooses a legitimate tag ID

$$RK_{ID} = K_{ID}^{j} \quad and \quad TK_{ID} = K_{ID}^{i}$$

- At the end of the experiment, we define both intermediary values:
  - $D_{\mathcal{R},\mathcal{A}} = j - i$
  - $D_{\mathcal{T},\mathcal{A}} = i - j$

Definition

For a given RFID authentication scheme, the *desynchronization value* of a scheme is the couple $(D_{\mathcal{R}}, D_{\mathcal{T}})$ with $D_{\mathcal{R}} = Sup_{\mathcal{A}}(D_{\mathcal{R},\mathcal{A}})$ and $D_{\mathcal{T}} = Sup_{\mathcal{A}}(D_{\mathcal{T},\mathcal{A}})$. The scheme is said $(D_{\mathcal{R}}, D_{\mathcal{T}})$-desynchronizable .

Our New Modelization

The Resynchronization Value $(R_{\mathcal{R}}, R_{\mathcal{T}})$:

- $R_{\mathcal{R}}$: maximum number of times that a key stored in DB can be desynchronized while the corresponding tag is still accepted by the reader.
- $R_{\mathcal{T}}$: maximum number of times that a tag can be desynchronized while it is still accepted by the reader.

Example:

OSK:

- a tag can be resynchronized indefinitely $\Rightarrow R_{\mathcal{T}} = \infty$,
- the reader can not be desynchronized and so, no mechanism to resynchronize it is needed $\Rightarrow R_{\mathcal{R}} = 0$.

OSK$_m$/OSK-AO:

- a tag can be resynchronized only m times $\Rightarrow R_{\mathcal{T}} = m$,

Our New Modelization

Formally:

- We initialize a counter $C = 1$;
- We force the tag (resp. the reader) to update its secret key;
- An authentication protocol between the tag and the reader is launched;
- If the reader accepts the tag, we restart this procedure by incrementing C, else the resynchronization value is equal to $C - 1$.

Definition

For a given RFID authentication scheme, if $D_{\mathcal{R}} \leq R_{\mathcal{R}}$ and $D_{\mathcal{T}} \leq R_{\mathcal{T}}$, the scheme is said *synchronizable*. Else, the scheme is said *desynchronizable*.

For OSK_m and OSK-AO, as $D_{\mathcal{T}} > R_{\mathcal{T}}$, it is desynchronizable .

Our New Modelization

- **Efficiency of the Search Procedure:** for a given scheme, we compute the number of operations (per tag) performed by the reader to accept/reject a tag in the worst case.

Examples:

OSK:

- On reception of a random value, the reader updates "indefinitely" all stored values without finding a match.

OSK_m:

- On reception of a random value, the reader updates m times all stored values without finding a match, inducing $2m+1$ computations of hash function per tag.

OSK-AO:

- On reception of a random value, the reader has to compute the end of each possible chain of the rainbow table and compares them with those stored in the database, inducing $2(t-1)^2/n$ operations per tag.

Results in this model

Protocol	Des.	Res.	Search	Security
OSK	$(\infty, 0)$	$(\infty, 0)$	∞	OK
OSK_m	$(\infty, 0)$	$(m, 0)$	$2m+1$	OK
OSK-AO	$(\infty, 0)$	$(m-1, 0)$	$\frac{2(t-1)^2}{n}$	OK
Dimitriou	$(0, 1)$	$(0, 1)$	2	Traceable [1]
O-FRAP/O-FRAKE	$(0, 1)$	$(0, 1)$	2	No Forward-Privacy [2]

No scheme presents all the requested properties.

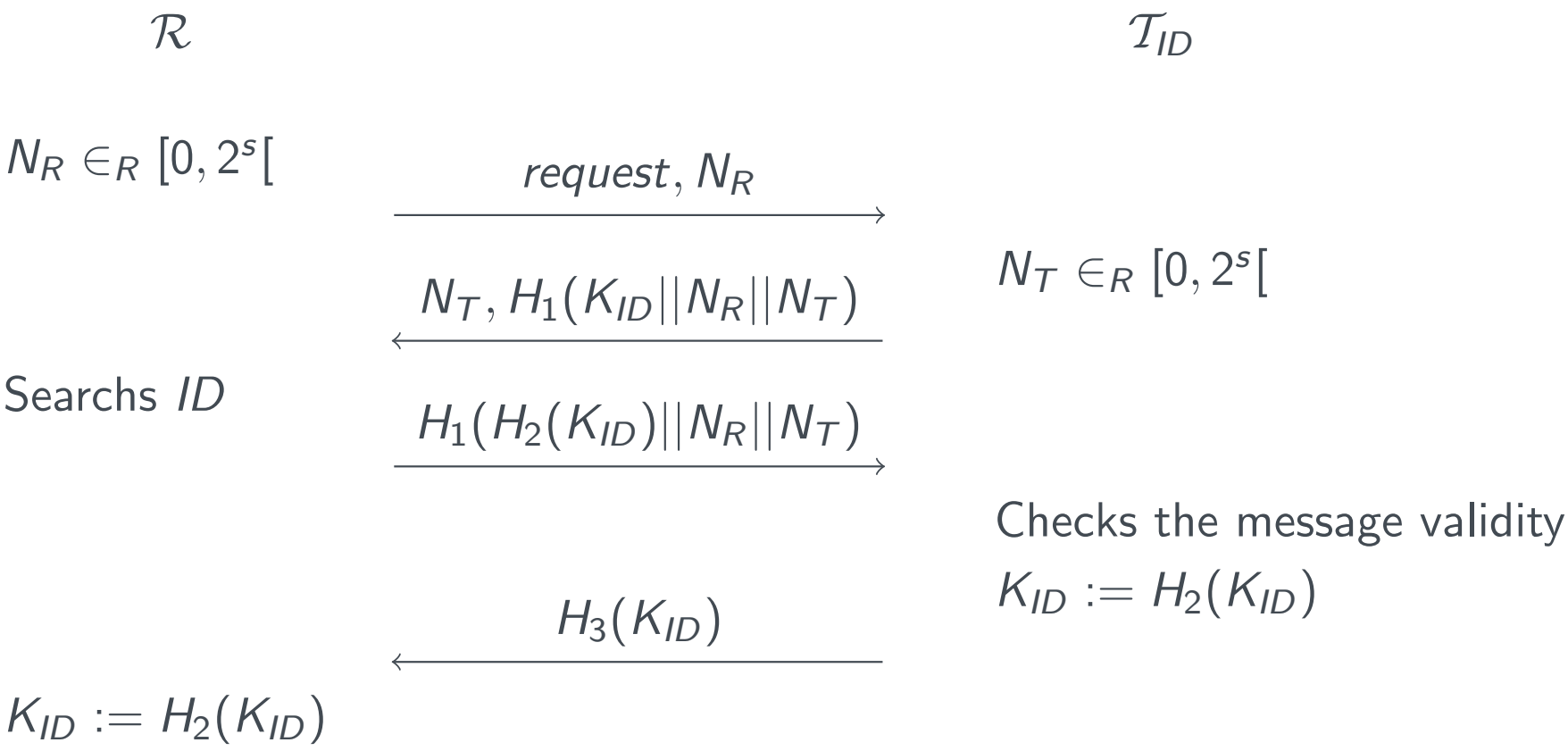
---

[1] This paper

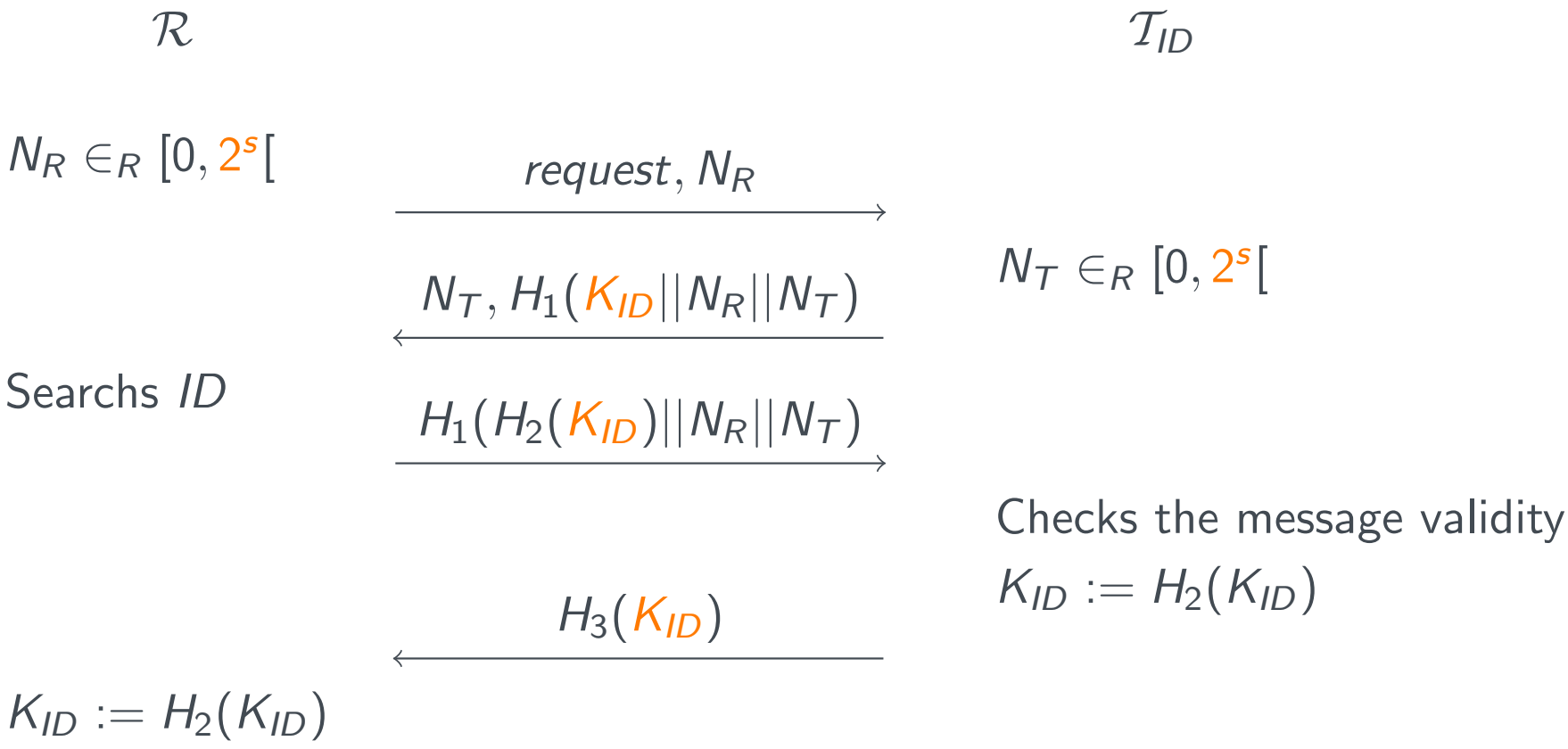
[2] K. Ouafi and R. C.-W. Phan, Traceable Privacy of Recent Provably-Secure RFID Protocols. In ACNS 2008, volume 5037 of LNCS, pages 479-489, 2008.

Outline

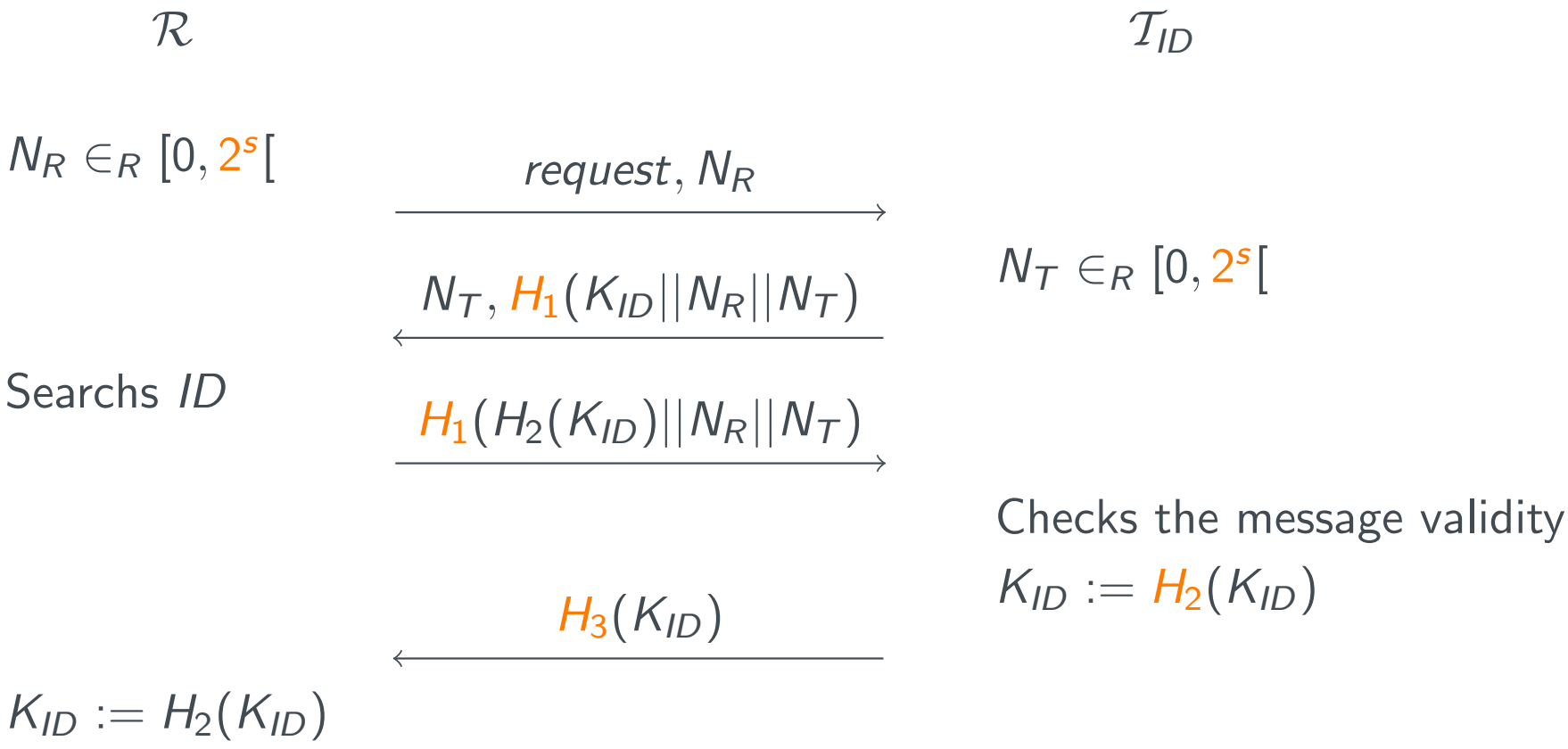
Our New Scheme: The C^2 Scheme



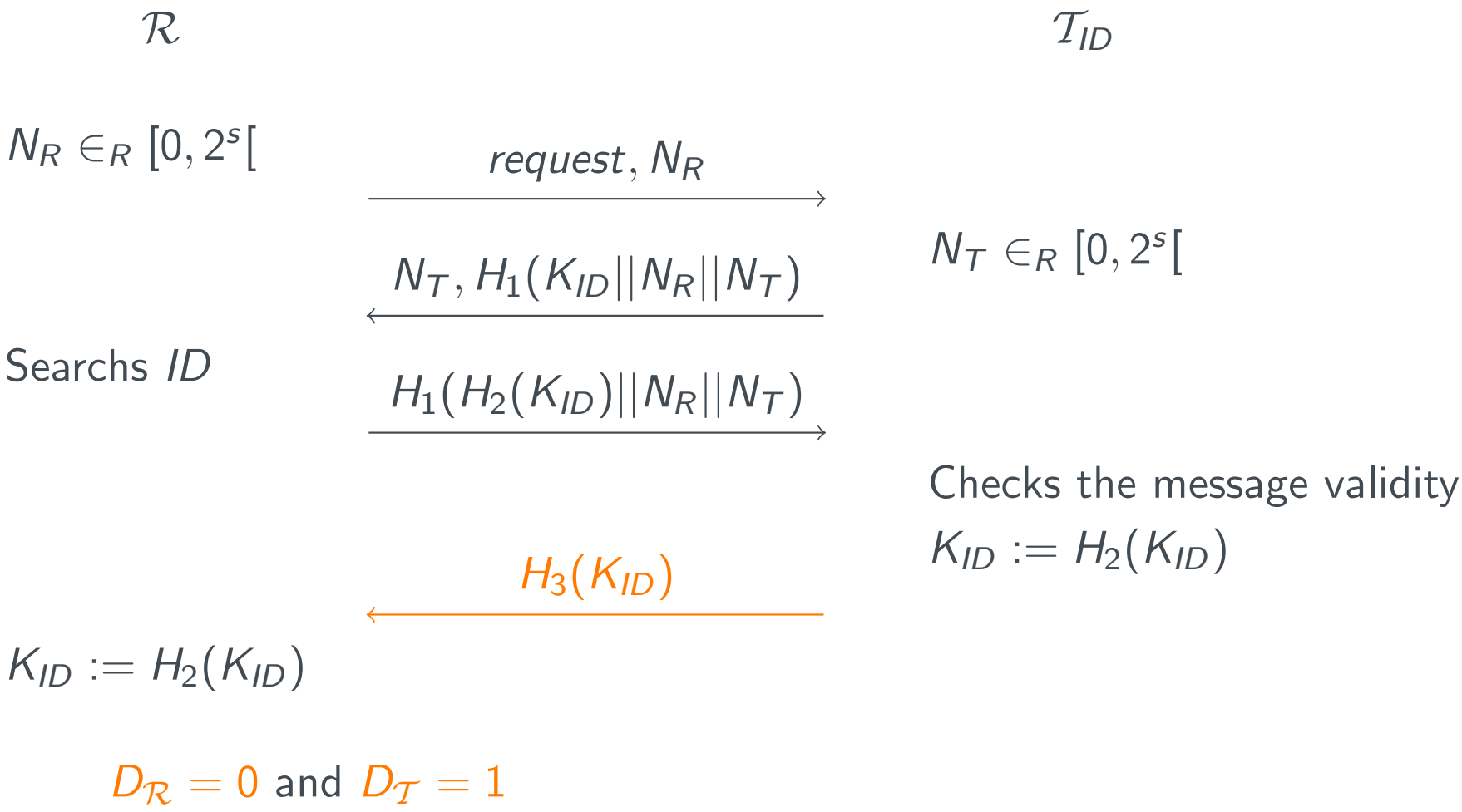
Security Properties - Soundness



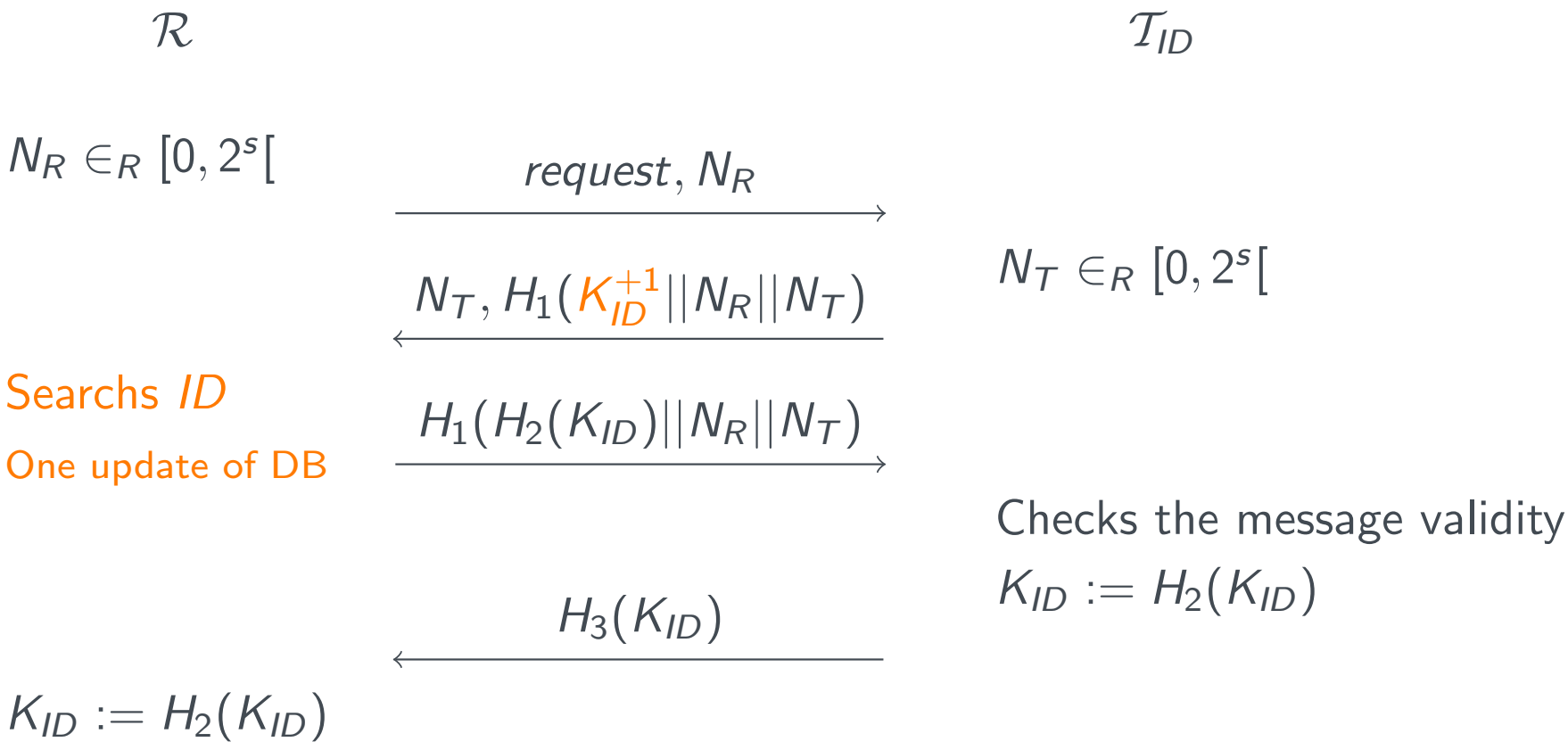
Security Properties - Privacy



Desynchronization Property



Resynchronization Property



$R_{\mathcal{R}} = 0$ and $R_{\mathcal{T}} = 1$. The scheme is synchronizable

Search Procedure Efficiency

$\mathcal{R}$ $\qquad\qquad\qquad\qquad\qquad\qquad$ $\mathcal{T}_{ID}$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad \ldots$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad N_T, r = H_1(K_{ID}||N_R||N_T)$

$\qquad\qquad\qquad\qquad\qquad\qquad\quad \longleftarrow$

Searchs ID:

- $\forall\ i \in [1, n]\ \ do$

  $H_1(K^R_{ID_i}||N_R||N_T) \stackrel{?}{=} r$

- if there is no match $\forall\ i \in [1, n]\ \ do$

  $\tilde{K}^R_{ID_i} := H_2(K^R_{ID_i})$

  $H_1(\tilde{K}^R_{ID_i}||N_R||N_T) \stackrel{?}{=} r$

The search procedure works in 3 operations in the worst case

Conclusion

Our contributions:

- We present new security properties to compare efficiency of RFID protocols.
- We study related work in this new model.
- We present a new privacy preserving authentication protocol with good desynchronization value at the price of some additional computations.

Open Problems:

- Show that at least one desynchronization, of the tag or the reader, is unavoidable when the protocol uses a key-update mechanism.
- Find a search procedure independent of the number of tags of the system.