Time Measurement Threatens Privacy-Friendly RFID Authentication Protocols

Gildas Avoine[1], Iwen Coisel[2] and Tania Martin[1]

1: Information Security Group - Université Catholique de Louvain
2: Crypto Group - Université Catholique de Louvain

RFIDSec 2010

The Privacy of an RFID Authentication Scheme

- Interest relative to the application
  - not really necessary in inventory management
  - essential in passport context to protect user's identity and also to prevent anybody to trace him

- Lots of sensitive applications
  - medical supplies
  - transport cards
  - luxury items
  - ...

$\Rightarrow$ Real necessity of a privacy analysis

We here focus on traceability

Privacy vs Time Measurement

Several privacy models exist [A05,JW07,LBM07,V07,CCG10]

- Juels and Weis : possible to know the result of a protocol
- Vaudenay : tags are not necessary in the adversary's field

How long it takes to a reader to identify a tag ? None of them

It's not (only) an implementation issue

Privacy vs Time Measurement

Several privacy models exist [A05,JW07,LBM07,V07,CCG10]

- Juels and Weis : possible to know the result of a protocol
- Vaudenay : tags are not necessary in the adversary's field

How long it takes to a reader to identify a tag ? None of them

It's not (only) an implementation issue

Contributions :

- Point out this threatens
- Formalize it
- Attacks some protocols
- Present some countermeasures

Outline

Outline

Vaudenay's Model [Vau07]

List of oracles given to an adversary $\mathcal{A}$

- CREATETAG : adds a new legitimate tag.
- DRAWTAG : tag enters in the adversary's field
- FREE : tags goes out of the adversary's field
- EXECUTE : returns transcripts.
  - LAUNCH
  - SENDTAG
  - SENDREADER
  - RESULT
- CORRUPT : returns tag's key set.

Vaudenay's Model [Vau07]

Considering the CORRUPT oracle, 3 adversary's ability :

- WEAK : no CORRUPT allowed
- FORWARD : CORRUPT "stops" the system
- STRONG : CORRUPT has no effect

Considering the RESULT oracle, 2 adversary's ability :

- NARROW : no RESULT allowed

Adversary classes ordered by power P

$$\begin{array}{ccccc} \text{STRONG} & \Rightarrow & \text{FORWARD} & \Rightarrow & \text{WEAK} \\ \Downarrow & & \Downarrow & & \Downarrow \\ \text{N-STRONG} & \Rightarrow & \text{N-FORWARD} & \Rightarrow & \text{N-WEAK} \end{array}$$

Vaudenay's Model [Vau07]

Experiment of $\mathcal{A}$

1. $\mathcal{A}$ interacts with the whole system
2. $\mathcal{A}$ submits an hypothesis
3. $\mathcal{A}$ obtains Tab and returns 0/1

The protocol is said P-private if $\mathcal{A}^{sim}$ has the same success probability as $\mathcal{A}$:

$$|Pr[\mathcal{A} \rightarrow 1] - Pr[\mathcal{A}^{sim} \rightarrow 1]| < \epsilon(k)$$

$$\begin{array}{ccccc} \text{STRONG} & \Rightarrow & \text{FORWARD} & \Rightarrow & \text{WEAK} \\ \Downarrow & & \Downarrow & & \Downarrow \\ \text{N-STRONG} & \Rightarrow & \text{N-FORWARD} & \Rightarrow & \text{N-WEAK} \end{array}$$

Time-Privacy

To capture the time notion in an authentication protocol

- TIMER : outputs the time δ taken by the reader for its overall computations during a given protocol instance

Possible to define the TIMEFUL-Privacy

- Adds a new ability $\Rightarrow$ more powerful
- At each level $X \in \{\text{STRONG, FORWARD, WEAK}\}$:

$$\begin{array}{ccc} \text{TIMEFUL-}X & \Rightarrow & X \\ \Downarrow & & \Downarrow \\ \text{TIMEFUL-NARROW-}X & \Rightarrow & \text{NARROW-}X \end{array}$$

Outline

Context of the Study

Several key infrastructures possible

	secret-key	public-key
master	X	Yes
particular	Yes	Yes

Considering Vaudenay's generic scheme [Vau07]

- Authentication : encryption of $\mathcal{ID}||K||a$
- Verification : decryption of the message + authenticity of K

$\Rightarrow$ constant-time authentication

Particular secret-key infrastructure

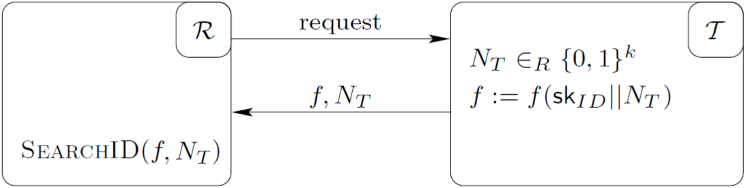
- Each tag owns a particular secret-key
- The reader does not know which key to use

$\Rightarrow$ SearchID *procedure*

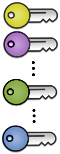
WSRE Protocol

Protocol proposed by Weis, Sarma, Rivest and Engels [WSRE03]

- Each tag owns a secret key sk_{ID} ;
- f is a pseudo-random function ;



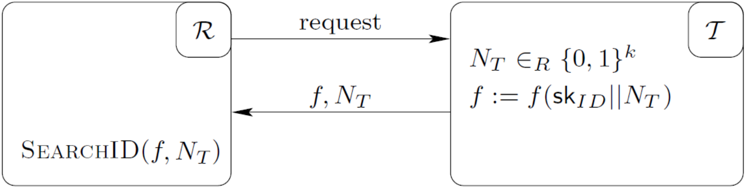
$\textsc{SearchID}$ procedure : brute-force search



WSRE Protocol

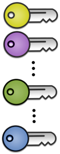
Protocol proposed by Weis, Sarma, Rivest and Engels [WSRE03]

- Each tag owns a secret key sk_{ID} ;
- f is a pseudo-random function ;



SEARCHID procedure : brute-force search

- Best case : 1 computation
- Average : $n/2$ computations
- Worst case : n computations



WSRE Protocol

A time-attack on WSRE

- $\mathcal{A}$ creates 2 legitimate tags and affects them : t_1 and t_2
- $\mathcal{A}$ calls $\textsc{Execute}(t_1)$ and $\textsc{Execute}(t_2)$: (π_1, tr_1), (π_2, tr_2)
- $\mathcal{A}$ calls $\textsc{Timer}(\pi_1)$ and $\textsc{Timer}(\pi_2)$: δ_1 and δ_2
- $\mathcal{A}$ frees both tags, and reaffects only one of them : t_3
- $\mathcal{A}$ calls $\textsc{Execute}(t_3)$: (π_3, tr_3)
- $\mathcal{A}$ calls $\textsc{Timer}(\pi_3)$: δ_3
- If $\delta_3 = \delta_1$, then $t_1 = t_3$, else $t_2 = t_3$

$$\Rightarrow Pr[\mathcal{A} \rightarrow 1] = 1$$

WSRE Protocol

A time-attack on WSRE

- $\mathcal{A}$ creates 2 legitimate tags and affects them : t_1 and t_2
- $\mathcal{A}$ calls $\textsc{Execute}(t_1)$ and $\textsc{Execute}(t_2)$: (π_1, tr_1), (π_2, tr_2)
- $\mathcal{A}$ calls $\textsc{Timer}(\pi_1)$ and $\textsc{Timer}(\pi_2)$: δ_1 and δ_2
- $\mathcal{A}$ frees both tags, and reaffects only one of them : t_3
- $\mathcal{A}$ calls $\textsc{Execute}(t_3)$: (π_3, tr_3)
- $\mathcal{A}$ calls $\textsc{Timer}(\pi_3)$: δ_3
- If $\delta_3 = \delta_1$, then $t_1 = t_3$, else $t_2 = t_3$

$$\Rightarrow Pr[\mathcal{A} \rightarrow 1] = 1$$

For the simulation, the output of $\textsc{Timer}(\pi_3)$ is guessed

$$\Rightarrow Pr[\mathcal{A}^{Sim} \rightarrow 1] = 1/2$$

WSRE is NOT TIMEFUL-WEAK-private.

Several Attacks

Ohkubo, Suzuki and Kinoshita [OSK03]

- NARROW-FORWARD private
- Not TIMEFUL-WEAK private
- Desynchronisation helps to distinguish two tags

Undesynchronizable schemes [D05, LBM07, CC08, ...]

- Only one possible desynchronization
- WEAK private
- Not TIMEFUL-WEAK private

Outline

Presentation

Major concern = SEARCHID procedure

Example for WSRE

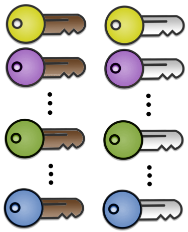
- Always waiting until the worst case (n computations)
  - "Always" applicable
  - Not efficient
- Random SEARCHID instead of a linear one
  - More efficient : $n/2$ computations in average for each tag

Countermeasures

- Not possible to link a time length to a tag
- Optimally : time length independent of n

Undesynchronizable Schemes

Tags can be desynchronized once
⇒ 2 possible keys per legitimate tag



- Worst case : $2n$ computations (instead of n)
- Random Search
  - Synchronized tag : $n/2$ computations
  - Desynchronized tag : $3n/2$ computations

  ⇒ $\mathcal{A}$ can distinguish 2 tags
- New Random Search
  - Random among the whole set of keys (current and old/next ones)
  - Average time for all tags : n computations

Precomputation Solution

No random values in OSK

$\Rightarrow$ Precomputation of "all" answers possible : $n.m$ answers

- Balanced Binary Search
  - SEARCHID efficient : $O(\log n)$
  - really dynamic : tags can be added infinitely
- Rainbow Table [AO05,ADO05]
  - Database size reduced
  - Efficiency of SEARCHID depends on the time-memory trade-off
  - But not dynamic
  - But requires database update (instead of tag update)

Outline

Conclusion

- Point a new threaten : computation time of the reader
- Model a new TIMEFUL adversary
- Lots of protocols are not TIMEFUL private
- Hopefully counter-measures are possible
  - Should not (only) be an implementation consideration
  - Constant-Time authentication exists
  - Still some progress to do to comply efficiency and small database

Conclusion

- Point a new threaten : computation time of the reader
- Model a new TIMEFUL adversary
- Lots of protocols are not TIMEFUL private
- Hopefully counter-measures are possible
  - Should not (only) be an implementation consideration
  - Constant-Time authentication exists
  - Still some progress to do to comply efficiency and small database

Thank You