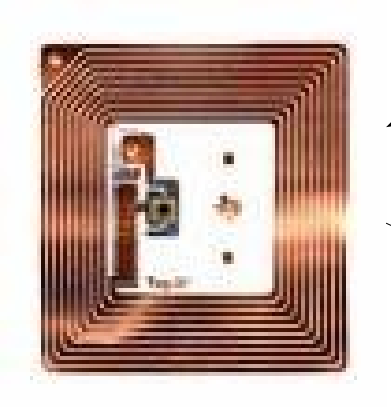
Serverless Search and Authentication Protocols for RFID

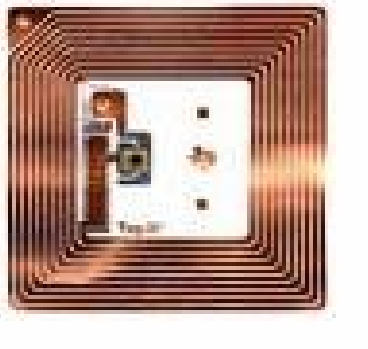
Chiu C. Tan, Bo Sheng and Qun Li
Department of Computer Science
College of William and Mary

What is RFID?

ID #:12345678
Name: Fischer, B



Using RFID



Tag

Query

Data

Reader

Basic Requirements for RFID

- Authentication
  - We want reader to distinguish a fake RFID tag from a real RFID tag.
- Privacy
  - We want the information from the RFID tag to be read only by authorized RFID readers.
  - This also includes location information about the tag .

Why authentication and privacy?

No Authentication

Accept

Query

Fisher, B

Real Tag

Query

Fisher, B

Fake Tag

Accept ???

No Privacy

Query

Fisher, B

Law Enforcement

Query

Fisher, B ??

Paparazzi

Location Privacy

Query

Query

Fisher, B

Fisher, B

Using a Server (adapted Dimitriou 2005)

Query

$h(ID_i), N, h_{ID_i}(N)$

$h(ID_i), N, h_{ID_i}(N)$

Fake tag does not know ID_i

$h(ID_i)$ not found in server

Unauthorized reader will not get any data from server.

Tag uses a different N each time

Cannot track movement of tag.

1. Find ID that matches $h(ID_i)$
2. Verify $h_{ID_i}(N)$ matches.
3. Returns information to reader.

Our paper

- Serverless solution, while still providing authentication and privacy.
  - Server solutions requires constant connection between reader and server. Not always possible !
- Ability to search,i.e. how to find 1 tag from a larger group of tags.
  - Authentication. Reader can detect a fake tag.
  - Privacy. Unauthorized reader cannot get back useful information from searching for a tag

Remainder of this talk

- Introduce our serverless solution.
- Introduce the problem of searching RFID tags.
- Conclude.

First dig at the problem…

(adapted from Weis et. al. 2003)

Query

$N\,,\ ID \oplus h_t(N)$

Secret t

Secret		*ID*
t_1	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
t_n	$\Leftrightarrow$	ID_n

Access List

For every entry inside access list

Apply $h_t()$ to N

Do $(ID \oplus h_t(N)) \oplus h_t(N)$ to get ID

Check against ID in access list

Checking Authentication

Query

$N\,,\, ID \oplus h_x(N)$

Secret x

Secret		*ID*
t_1	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
t_n	$\Leftrightarrow$	ID_n

Access List

Reader checks access list

Secret x not in access list.
Reader rejects tag !

Checking Privacy

Unauthorized

Query

$N\,,\, ID \oplus h_t(N)$

Secret t

Blank

Access List

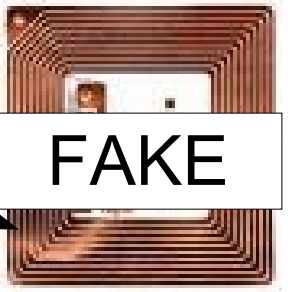
Unauthorized reader does not have secret t.
Cannot obtain ID !

Every tag reply uses a different random N.
Cannot track movements of the tag!

However, do not lose the reader

Secret		ID
t_1	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
t_n	$\Leftrightarrow$	ID_n

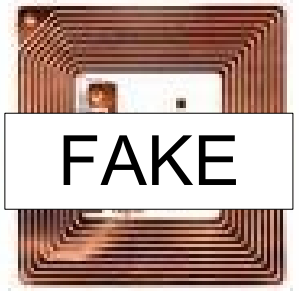
Access List



Secret t, ID

Query

$N\,,\,ID \oplus h_t(N)$



Secret t, ID

Fools legitimate reader !

If you first don't succeed .....

- Problem: Reader knows tag secret t
- Idea : Let the reader check if tag knows secret t, without telling reader t.
- Use 1-way hash function.
- Create reader id for each reader, r.
- Instead of telling reader t, use f(r,t) . f() is a 1-way hash function.

If you first don't succeed .....

Query, r

$f(r,t) \oplus ID$

ID,t, f()

Secret		*ID*
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

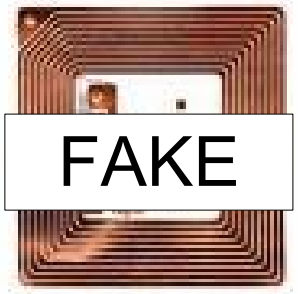
Access List

For every entry inside access list
Do $(f(r,t) \oplus ID) \oplus f(r,t)$ to get ID
Check against ID in access list

Stealing the reader

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

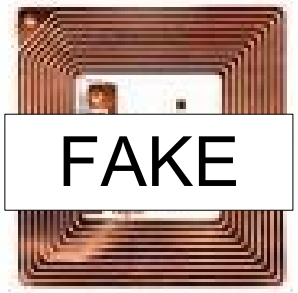
Access List



Secret f(r,t)

Query, s

$f(r,t) \oplus ID$



Secret f(r,t)

$f(s,t) \Leftrightarrow ID$

Access List

Reader s expects f(s,t).
Reader not fooled !

However .....

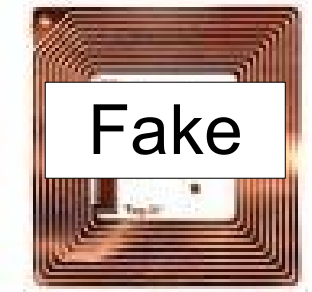
Query, r

$f(r,t) \oplus ID$

ID,t, f()

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

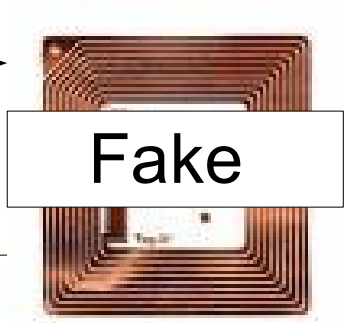


$f(r,t) \oplus ID$

When reader queries again …

Query, r

$f(r,t) \oplus ID$



$f(r,t) \oplus ID$

Checks access list

$Secret$		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

Fake tag fools legitimate reader r !

Third times a charm ...

- Problem:
  - Reader always issues same query, susceptible to replay attack.
  - Tag response always the same, vulnerable against tracking.
- Idea: Use random numbers to differentiate query and response.

Third times a charm ...

$$Query \rightarrow$$

$$\leftarrow n_t$$

$$r, n_r \rightarrow$$

$$\leftarrow h(f(r,t), n_r, n_t) \oplus ID$$

ID,t, f(),h()

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

Check against access list.

Eavesdropping now gets

$$Query$$

$$n_t$$

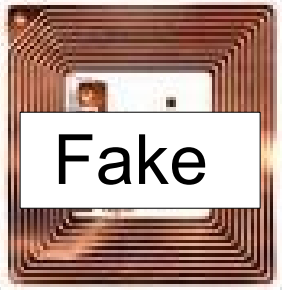
$$r, n_r$$

$$h(f(r,t), n_r, n_t) \oplus ID$$

ID,t, f(),h()

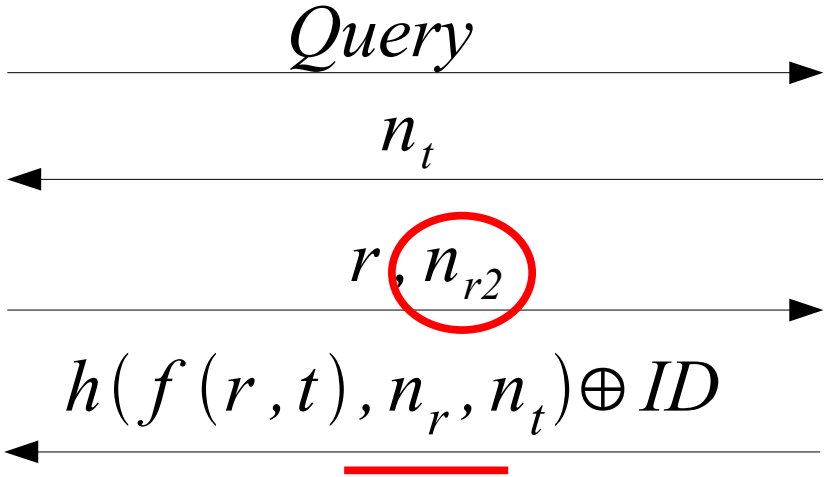
$Secret$		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

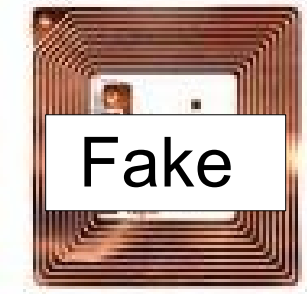
Access List



$$h(f(r,t), n_r, n_t) \oplus ID$$

When reader queries again





$h(f(r,t), n_r, n_t) \oplus ID$

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

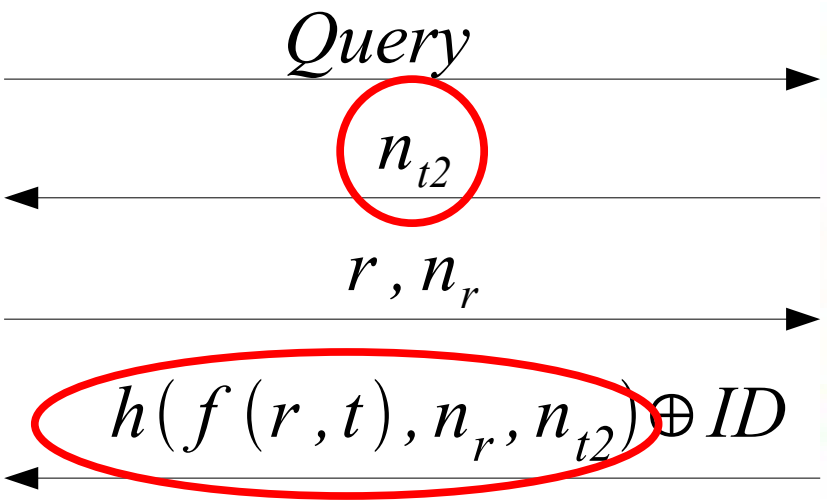
Access List

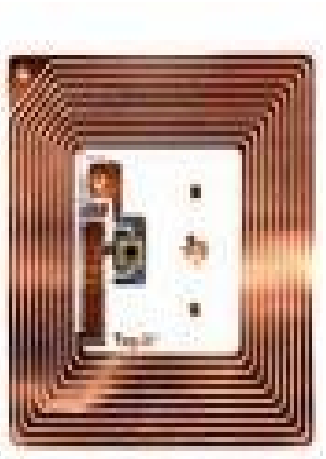
Reader will pick a different random number each time.

When checked against access list, reader will not get back ID.

Reader not fooled !

When unauthorized reader queries





ID,t, f(),h()

Tag uses a different random number each time

Unauthorized reader repeats the same query at different places

Gets back a different reply.

No tracking !

A little bit faster ….

Problem : Access list could have many entries. Slow.

Return the first m bits of h(f(r,t))

$$h(f(r,t))_m,$$
$$h(f(r,t),n_r,n_t)\oplus ID$$

ID,t, f(),h()

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
⋮	$\Leftrightarrow$	⋮
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

Tag returns the first m bits of h(f(r,t))

Reader can pre-compute this value ahead of time. Not affected by random numbers.

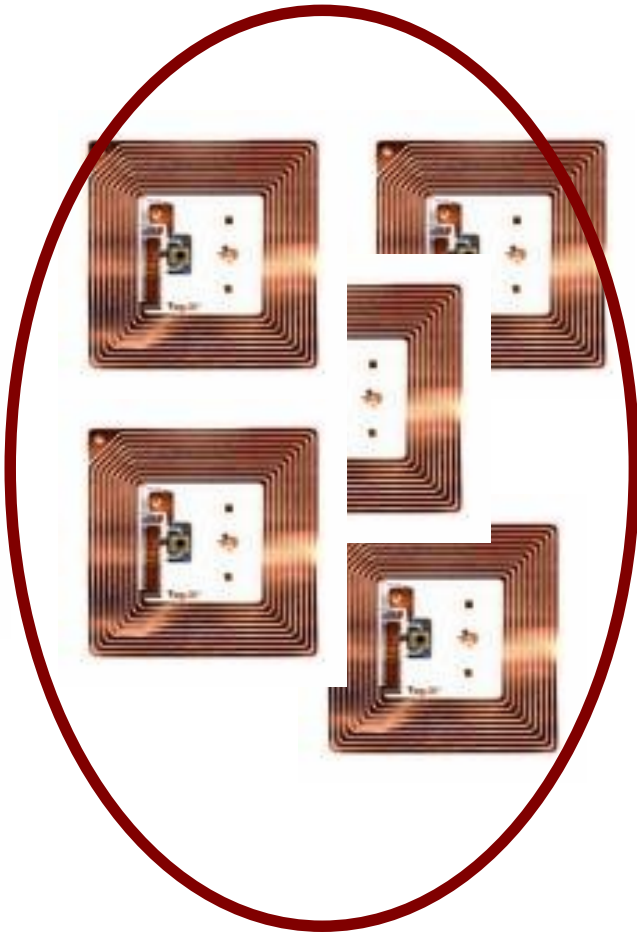
Onto searching

- RFID search problem:
  - Find 1 particular tag from a collection of tags, while still providing authentication and privacy.
- Simple solution:
  - Collect IDs from every tag, using technique presented earlier.
  - Find the tag you want.
- Not efficient.

Needle in a haystack

ID_4 ?

ID_4



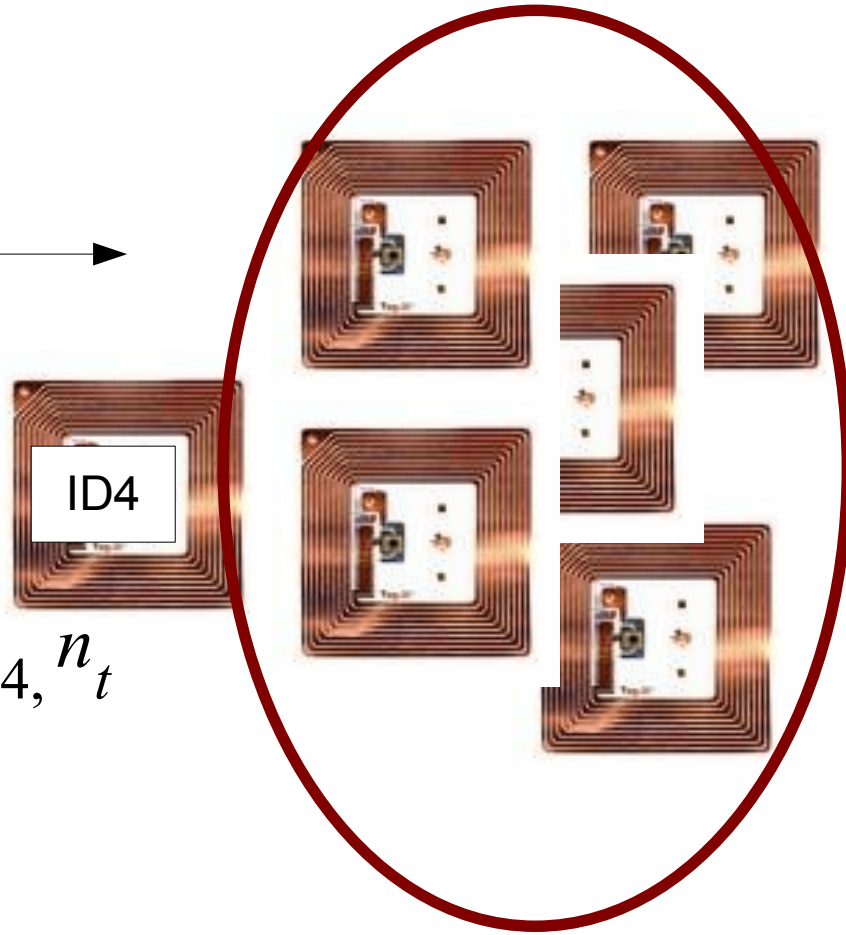
Reader cannot distinguish fake tag.
An unauthorized reader can still query.
No protection against eavesdropper.

Keep silent

Use the same trick as before

$h(f(r,t_4),n_r)\oplus ID_4$

n_r, r



$h(f(r,t_4),n_t,n_r)\oplus ID_4, n_t$

Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

Verify own ID by deriving
$h(f(r,t),n_r)$, then XOR to get ID_4

Seems to work .....

- We basically invert the same techniques presented earlier on.
- One major exception ....

What went wrong?

$$h(f(r,t_4),n_r)\oplus ID_4$$

$$n_r, r$$

$$h(f(r,t_4),n_t,n_r)\oplus ID_4, n_t$$

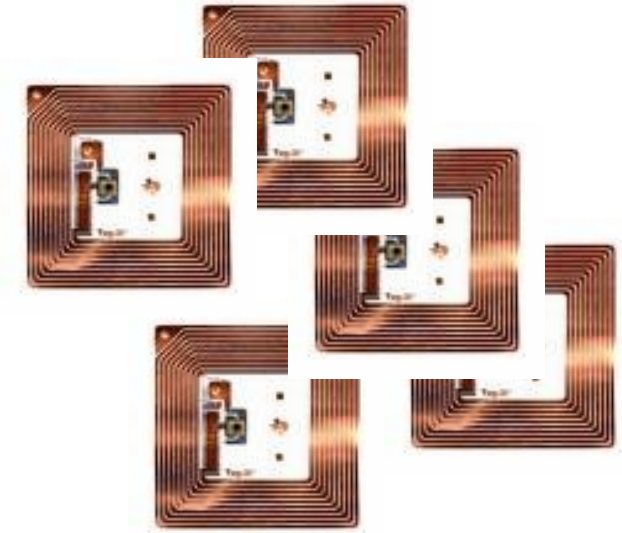
$$h(f(r,t_4),n_r)\oplus ID_4, n_r, r$$

$$h(f(r,t_4),n_t,n_r)\oplus ID_4, n_t$$

What went wrong?

$$h(f(r,t_4),n_r)\oplus ID_4$$

$$n_r, r$$



$$h(f(r,t_4),n_{t2},n_r)\oplus ID_4, n_t$$

Does not matter if tag changes random number

Still able to track, since only 1 tag will reply !

What went wrong?

- Basic nature of search. We expect to find 1 tag from a group of tags.
- In other words, always have 1 tag replying.

The very act of replying identifies the tag !

Possible techniques

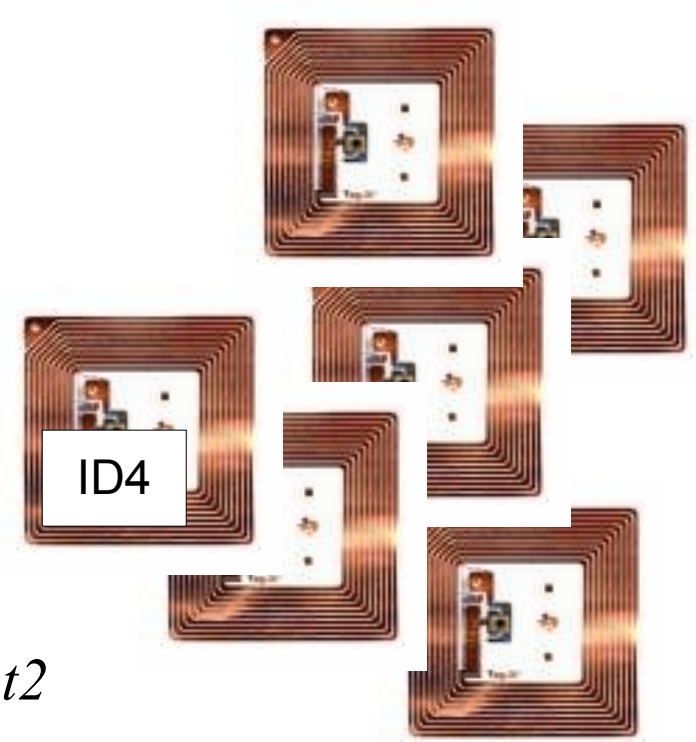
1. Try to prevent readers from repeatedly using the same random number.
2. Try to create a query that can be satisfied by more than 1 tag.
3. Try to generate noise to mask reply.

Possible solution

$$ID_{4m}, r, n_r \rightarrow$$

$$\leftarrow h(f(r,t_4), n_r, n_t) \oplus ID_4, n_t$$

$$\leftarrow h(f(r,t_5), n_r, n_{t2}) \oplus ID_5, n_{t2}$$



Secret		ID
$f(r,t_1)$	$\Leftrightarrow$	ID_1
$\vdots$	$\Leftrightarrow$	$\vdots$
$f(r,t_n)$	$\Leftrightarrow$	ID_n

Access List

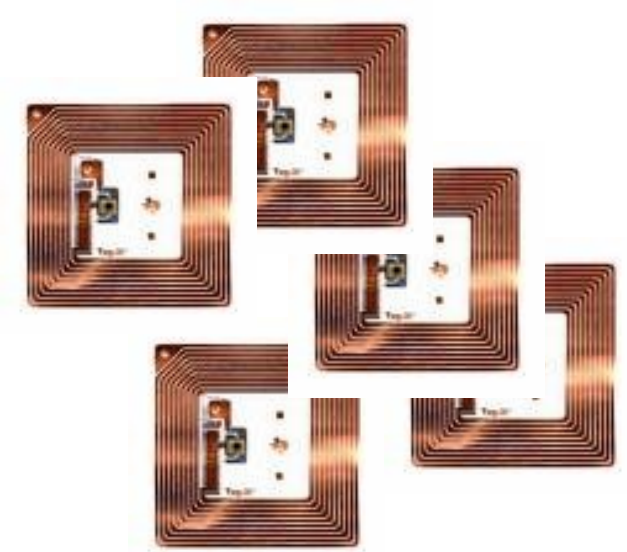
Check the first m bits

Reader receives more than 1 tag reply

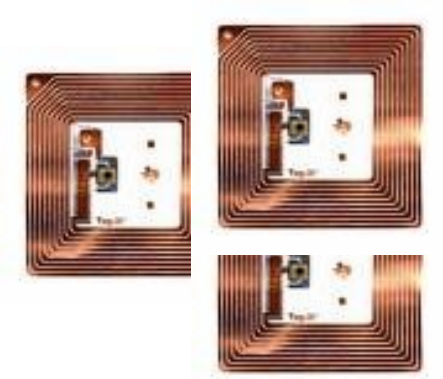
Reader obtains ID using access list and checks.

Under this scheme

ID_{4m}, r, n_r



$h(f(r, t_4), n_{t2}, n_r) \oplus ID_4, n_t$



Eavesdropper receives multiple replies.
Each tag chooses a different random number

Cannot perform tracking !

To summarize …

- Provide authentication and privacy protections for RFID.
- Done without need for persistent connections with central server.
- Examined security considerations when searching for RFID tags.
- Suggested solutions for secure RFID search

Acknowledgments

- We like to thank reviewers for their helpful comments.