Desynchronization and traceability attacks on RIPTA-DA protocol

Nasour Bagheri, Praveen Gauravaram*,
Masoumeh Safkhani, Somitra Kr Sanadhya

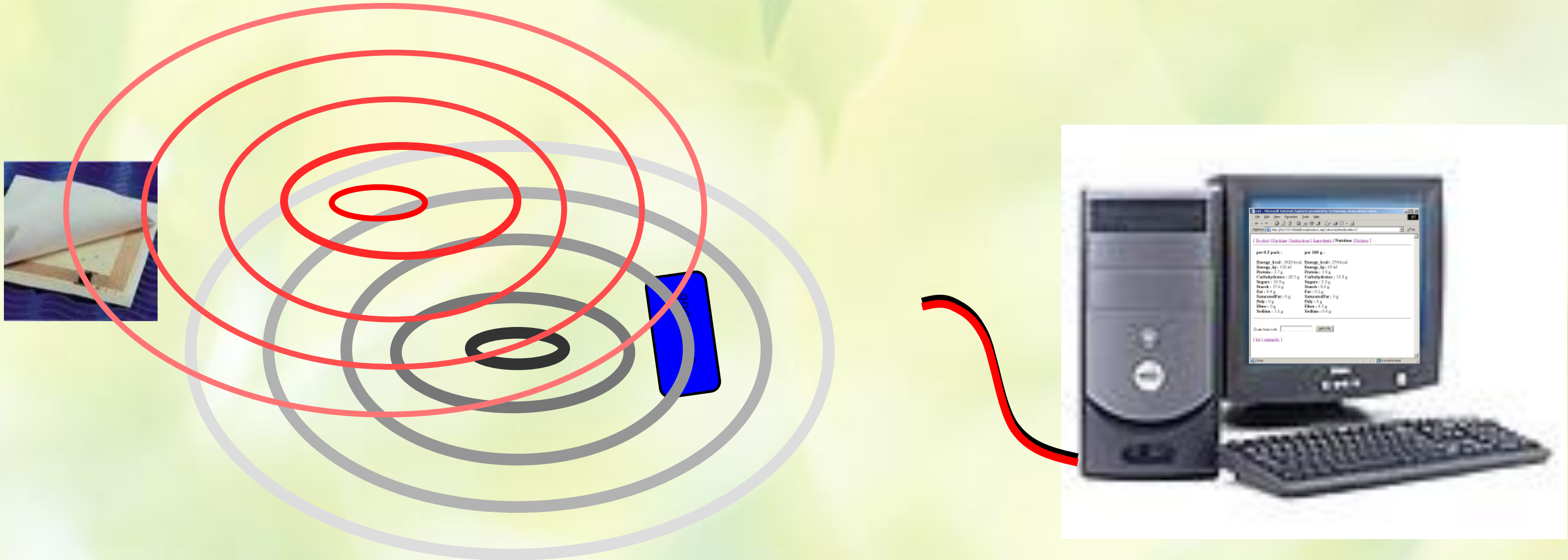
* TCS Innovation Labs, Hyderabad, India

The 9th Workshop on RFID Security, July 9-11, 2013, Graz, Austria

Outline

- RFID Overview
- Research Problem
- Outline of RIPTA-DA Protocol
- Main Weaknesses of RIPTA-DA
- Attacks
- Conclusions

RFID Overview



How Does RFID Work?

Tag

ID: 02.3DFEX4.78AF51

Radio signal

Reader

Database

RFID Applications

Security

Theft Prevention

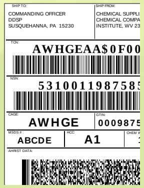
Packaging

Banking

Tracking

E-Passport

Barcode Replacement



SHIP TO:
COMMANDING OFFICER
DDSP
SUSQUEHANNA, PA 15230

SHIP FROM:
CHEMICAL SUPPLI
CHEMICAL COMPA
INSTITUTE, WV 23

TCN: AWHGEAA$0F00

NSN: 531001198758

CAGE: AWHGE

GTIN: 0009875

MSDS # : ABCDE

HCC: A1

AHRIST DATA:

Medicine

Libraries

Access Control

Toll Payment

And many more…

ISO-18000-6c RFID Standard

- EPCglobal Class-1 Gen-2 (EPC C1 G2) is one of the most important standards proposed by EPCglobal .
- This standard was adopted in 2004.
- 18months later (March–April 2006) ratified by ISO.
- Later published as an amendment to ISO-18000-6c standard.

The most important properties :

- Tags are passive.
- Tags operate on the UHF band (860–960 MHz).
- Tags cannot support conventional cryptographic primitives.
- Tags include on chip limited storage and computational resources for security purposes.

RFID Protocols Properties

- Mutual Authentication
- Resistance Against Active and Passive Attacks
- Privacy Preserving
- Resistance Against Traceability
- Resistance Against Secret disclosure Attack
- Resistance Against Desynchronization Attack
- Perfect Forward Secrecy and so on…

Research Problem

- We study the RIPTA-DA (Resisting the Intermittent Position Trace Attacks and Desynchronization Attacks) protocol
- This protocol was designed by Gao et al.
- We show that this protocol does not resist against
  - Secret disclosure attack,
  - Traceability attack, and
  - Desynchronization attacks.

Results

- We present a secret disclosure attack which given 512 consecutive queries to the tag and its responses, retrieves more than n bits out of a $3n$-bit secret key with the success probability of almost 1.

- In addition, given the recovered secret, we present an approach to trace the tag for which the adversary's advantage is 0.738 for each query to the tag.

- We present a desynchronization attack which after two queries to the tag, desynchronizes the tag and the reader with the probability of 1.

- The result of desynchronization attack is that the reader and tag do not authenticate each other anymore.

- These attacks contradicts the claims on the security of the RIPTA-DA protocol against traceability and desynchronization attacks.

Outline of RIPTA-DA protocol

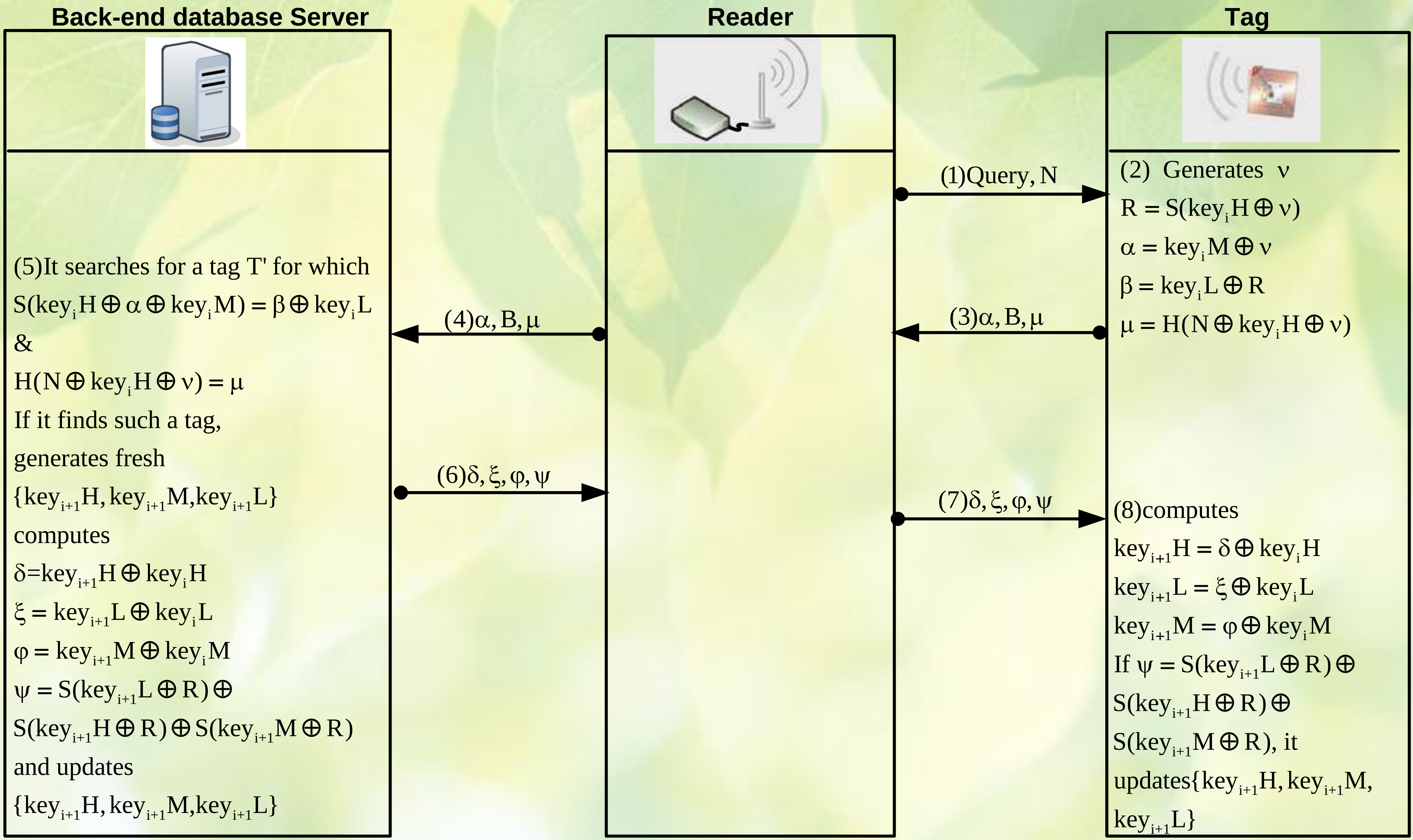
RIPTA-DA Protocol

- In this protocol, the tag and the reader share three n-bit secret keys denoted by $key_i H$, $key_i M$ and $key_i L$ respectively where i is session index.

- To avoid desynchronization attacks, both the tag and the reader keep two records of the secret parameters denoted by $key_i 1$ and $key_i 2$ respectively.

- $flag = 1$ implies that $key_i 1$ group keeps the previous success authentication secret key group and $flag = 0$ implies that $key_i 2$ group keeps the previous successful secret key group.

RIPTA-DA Protocol



Back-end database Server

Reader

Tag

(1)Query, N

(2) Generates ν

$R = S(key_i H \oplus \nu)$

$\alpha = key_i M \oplus \nu$

$\beta = key_i L \oplus R$

$\mu = H(N \oplus key_i H \oplus \nu)$

(3)α, B, μ

(4)α, B, μ

(5)It searches for a tag T' for which

$S(key_i H \oplus \alpha \oplus key_i M) = \beta \oplus key_i L$

&

$H(N \oplus key_i H \oplus \nu) = \mu$

If it finds such a tag,

generates fresh

$\{key_{i+1}H, key_{i+1}M, key_{i+1}L\}$

computes

$\delta = key_{i+1}H \oplus key_i H$

$\xi = key_{i+1}L \oplus key_i L$

$\varphi = key_{i+1}M \oplus key_i M$

$\psi = S(key_{i+1}L \oplus R) \oplus$

$S(key_{i+1}H \oplus R) \oplus S(key_{i+1}M \oplus R)$

and updates

$\{key_{i+1}H, key_{i+1}M, key_{i+1}L\}$

(6)$\delta, \xi, \varphi, \psi$

(7)$\delta, \xi, \varphi, \psi$

(8)computes

$key_{i+1}H = \delta \oplus key_i H$

$key_{i+1}L = \xi \oplus key_i L$

$key_{i+1}M = \varphi \oplus key_i M$

If $\psi = S(key_{i+1}L \oplus R) \oplus$

$S(key_{i+1}H \oplus R) \oplus$

$S(key_{i+1}M \oplus R)$, it

updates$\{key_{i+1}H, key_{i+1}M,$

$key_{i+1}L\}$

Notation

$\chi_{a\sim b}$ indicates a fraction of χ from bth bit to ath bit.

For example:

$$\chi = 1101 \quad 0001 \quad 1010 \quad 0111$$

$$\chi_{9\sim 5} = 01101$$

RIPTA-DA Protocol

The building block denoted by $S(A \oplus B)$ works as below:

$$x = A \oplus B$$

$$y = x^2$$

$$z = (y)_{((B)_{k-1\sim0})\sim(((B)_{k-1\sim0})-(n-1))}$$

$$S(A \oplus B) = z$$

where k is a fixed value and the factory determines this k.

<u>Observation:</u>

$$(B)_{k-1\sim0} = n-1 \Rightarrow y = (x^2)_{(n-1)\sim0} \Rightarrow (y)_1 = 0$$

i.e a specific bit will be 0.

RIPTA-DA Protocol

An example of $S(A \oplus B)$

$n = 16,$

$A = 1011\ 0100\ 1110\ 0101$

$B = 0100\ 0111\ 0101\ 1101$

and $k = 3$, then

$x = A \oplus B = 1111\ 0011\ 1011\ 1000$

$y = x^2 = 1110\ 1000\ 0000\ 0110\ 1101\ 0100\ 0100\ 0000$

$(B)_{3\text{-}1\sim 0} = 101$

$z = (y)_{5\sim\text{-}10} = \boxed{0000\ 00}\ \boxed{1110\ 1000\ 00}$

$S(A \oplus B) = 0000\ 0011\ 1010\ 0000$

Main Weaknesses

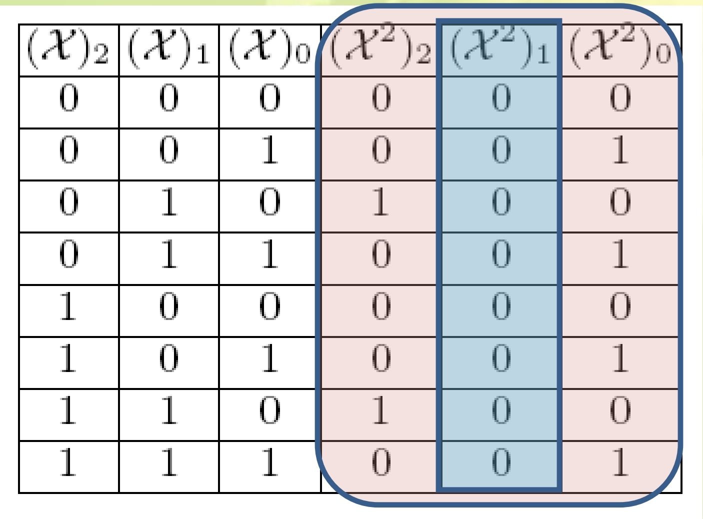
Main Observation

The only source of nonlinearity in RIPTA-DA is square random number function.

Given

$$\chi = (\chi)_{n-1} \,||\, \dots \,||\, (\chi)_1 \,||\, (\chi)_0$$

$$\chi^2 = \chi \times \chi$$



$(\mathcal{X})_2$	$(\mathcal{X})_1$	$(\mathcal{X})_0$	$(\mathcal{X}^2)_2$	$(\mathcal{X}^2)_1$	$(\mathcal{X}^2)_0$
0	0	0	0	0	0
0	0	1	0	0	1
0	1	0	1	0	0
0	1	1	0	0	1
1	0	0	0	0	0
1	0	1	0	0	1
1	1	0	1	0	0
1	1	1	0	0	1

$$(\chi^2)_0 = (\chi)_0$$

$$(\chi^2)_1 = 0$$

$$(\chi^2)_2 = (\chi)_1 \cdot \overline{(\chi)_0}$$

Attacks

Attacks against RIPTA-DA

Secret Disclosure Attack

Traceability Attack

Desynchronization Attack

Secret Disclosure Attack

- The adversary initiates t consecutive sessions
- In j^{th} session, the adversary sends N^j to the tag and receives tag answer which are

$$\alpha^j = key_i M \oplus v^j$$
$$\beta^j = key_i L \oplus R^j$$
$$R^j = S(key_i H \oplus v^j)$$

For $1 \le j,f \le t$

& $1 \le m \le n$

$$(\alpha^j \oplus \alpha^f)_m = (v^j \oplus v^f)_m = (v^j)_m \oplus (v^f)_m$$

- It is possible to determine $(v^j)_m \overset{?}{=} (v^f)_m$ for $1 \le m \le n$
- It is possible to group $v^1, \ldots, v^t$ into 2^k groups denoted by $G1, \ldots, G2^k$, respectively where any entry in a group holds the same value in its k least significant bits, i.e., $(v)_{k-1\sim0}$
- Then we are looking for a group for which the k least significant bits of v are equal to $n-1$

Secret Disclosure Attack

- $(R)_{2\sim0} = (key_i H \oplus v)_{2\sim0} \xrightarrow[(\chi^2)_1=0]{\text{Based on}} (R)_1 = 0, (\beta^j)_1 = (key_i L \oplus R)_1 = (key_i L)_1$
- Based on above, if for a group Gi, $(v)_{k-1\sim0} = 0$, then for all $(\beta)_1$ elements of that group should remain constant
- Given such a group we reveal $(key_i L)_1$ and $(v)_{k-1\sim0} = n - 1$
- Given $(v)_{k-1\sim0}$,we reveal k bits of $(key_i M)$ as $(key_i M)_{k-1\sim0} = ((\alpha^j) \oplus (n-1))_{k-1\sim0}$
- Given $(key_i M)_{k-1\sim0,}$ we reveal $(v)_{k-1\sim0}$ for each group and an extra bit of $key_i L$
- Following this approach, we determine all bits $key_i L$ and also $(key_i M)_{k-1\sim0}$
- Given $key_i L$, we determine R as $R = \beta \oplus key_i L$

Secret Disclosure Attack

- Based on $(\chi^2)_0 = (\chi)_0$ which combined with the extracted $(v)_0$ reveals $(key_i H)_0$

- Given $(key_i H)_0$,$(v)_1$ and $(\chi^2)_2 = (\chi)_0 \cdot \overline{(\chi)_1}$ we can retrieve $(key_i H)_1$

- Continuing this approach it is possible to reveal several other bits of $key_i H$

- The adversary succeeds in her attack if she selects a correct group, as a group for which $(v)_{k-1\sim 0} = n - 1$

Secret Disclosure Attack

- If for a group $(v)_{k-1\sim0} \neq n-1$, all elements of group holds the same $(\beta)_1$ only with $p=2^{-|G|}$.

  $|G|$ denotes the group's cardinality which is approximately $\frac{t}{2^k}$.

- Exclude the correct group, the adversary is expected to receive $(|\#G|-1)\times 2^{-|G|}$ groups that satisfy the given condition on $(\beta)_1$. We call such a group a quasi-correct-group.

- $|\#G|$ denotes the total number of groups, i.e, 2^k.

- The adversary knows the expected value of $(v)_{k-1\sim0}$ for each group.

- $(v)_{k-1\sim0}$ is used to determine the location of $(key_iL)_1$ in each group which can be used to filter wrong guesses.

Secret Disclosure Attack

- The adversary fails if for a wrongly selected group all the given conditions are satised.
- Given that for a quasi-correct-group all bits in the expected location for $(key_iL)_1$ on each group holds with $P= 2^{-|G|}$.
- We have 2^k groups and $(|\#G|-1) \times 2^{-|G|}$ quasi-correct-groups.
- A quasi-correct-group passes all conditions with

$$p = ((|\#G|-1) \times 2^{-|G|}) \times (2^{-|G|})^{\neq|G|} = (2^k - 1) \times 2^{-\frac{t}{2^k}} \times (2^{-\frac{t}{2^k}})^{2^k}$$

- For $k = 7$ and $t = 512$, a quasi-correct-group passes the conditions with $p < 2^{-508}$.
- Hence the adversary's advantage, i.e. $1-p$, for $t \geq 512$ is almost one.

Traceability Attack

Given T, $(key_i H))_{1\sim0}$, $(key_i M))_{k-1\sim0}$ and $key_i L$, to determine whether randomly selected tag T' is the target T, the adversary initiates a session and receives tag response and does

$$R' = key_i L \oplus \beta$$

$$(v')_{k-1\sim0} = (key_i M \oplus \alpha)_{k-1\sim0}$$

If $(v')_{k-1\sim0} >= 2$ then the adversary can determine $((key_i H \oplus v')^2)_{2\sim0}$ and $(key'_i H)_{1\sim0}$

The adversary outputs '1' if $(key_i H)_{1\sim0} = (key'_i H)_{1\sim0}$, otherwise outputs '0'

The adversary's advantage Adv_A to make the correct decision in this attack is

$$Adv_A = \left|Pr[A^{T=T'} \Rightarrow 1] - Pr[A^{T\neq T'} \Rightarrow 1]\right| =$$

$$\left|(1-\frac{2}{2^k})\times 1 + (\frac{2}{2^k})\times\frac{1}{2} - (1-\frac{2}{2^k})\times\frac{1}{4} - (\frac{2}{2^k})\times\frac{1}{2}\right|$$

For $k=7$, Adv_A is approximately 0.74

It is upper bounded by 0.75

Desynchronization Attack

- Given $key_i L$, it is enough to manipulate $key_{i+1} L$ without being detected.

- In the protocol, the adversary changes new value of $key_{i+1} L$ to $key'_{i+1} L$ which is sent in $\xi = key_i L \oplus key_{i+1} L$ from server to the tag.

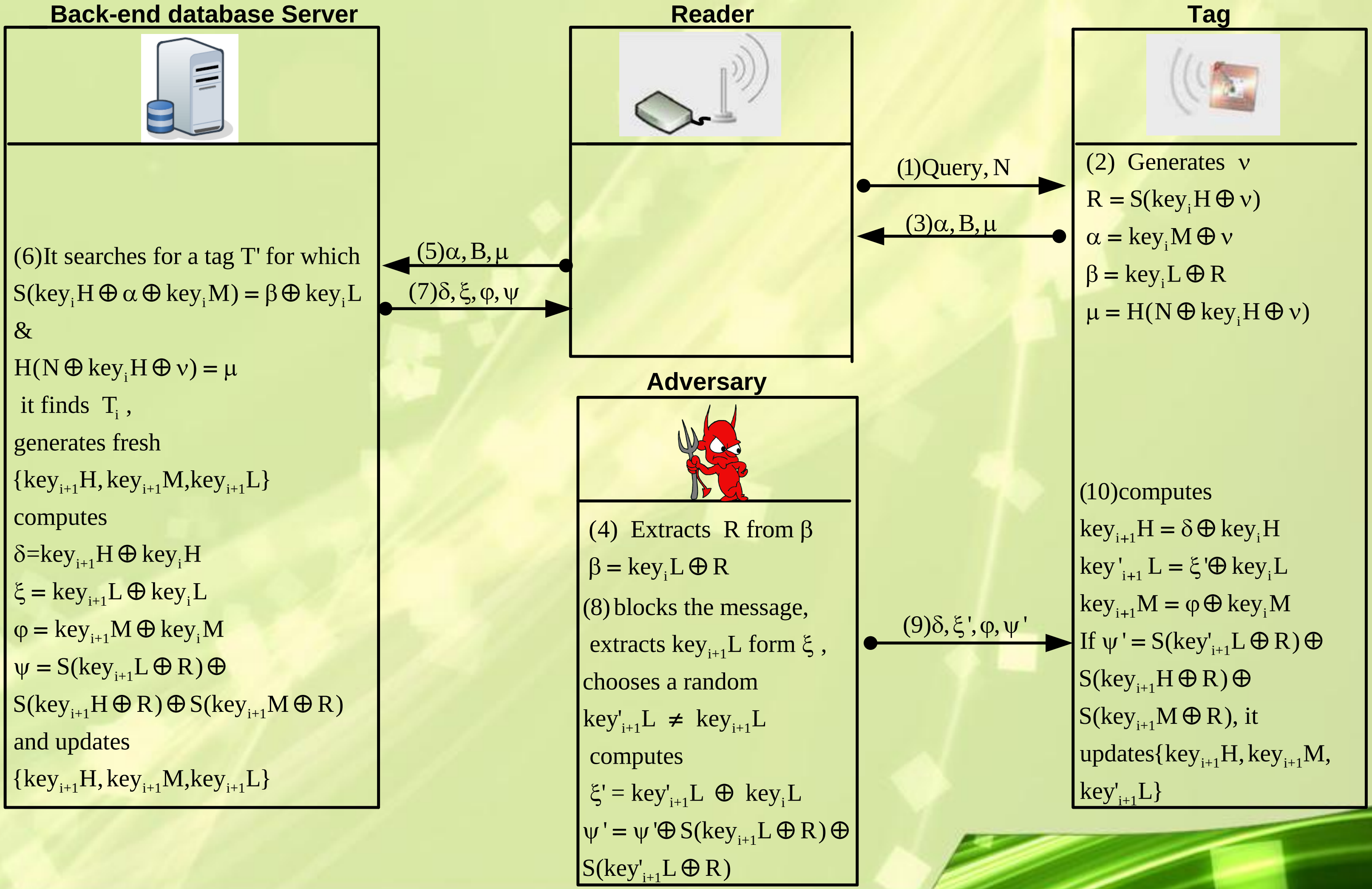
- Tag accepts the manipulated $key'_{i+1} L$. So the server and tag have $key_{i+1} L$ and $key'_{i+1} L$ which are different.

- In the next consecutive session, the adversary changes both value of $key'_i L$ and $key_{i+1} L$ to $key''_i L$ and $key'_{i+1} L$ which is sent in $\xi' = key'_i L \oplus key_{i+1} L$ from server to the tag.

- Tag accepts the manipulated $key''_i L$ and $key'_{i+1} L$. So the server has $key'_i L$ and $key_{i+1} L$ and the tag has $key''_i L$ and $key'_{i+1} L$ which are different.

Desynchronization Attack

Phase 1 (updating key_{i+1}): Adversary forces the tag and the database to update their record of $key_{i+1}L$ to different values as follows:



Back-end database Server

(6)It searches for a tag T' for which
$S(key_i H \oplus \alpha \oplus key_i M) = \beta \oplus key_i L$
&
$H(N \oplus key_i H \oplus \nu) = \mu$
it finds T_i ,
generates fresh
$\{key_{i+1}H, key_{i+1}M, key_{i+1}L\}$
computes
$\delta = key_{i+1}H \oplus key_i H$
$\xi = key_{i+1}L \oplus key_i L$
$\varphi = key_{i+1}M \oplus key_i M$
$\psi = S(key_{i+1}L \oplus R) \oplus$
$S(key_{i+1}H \oplus R) \oplus S(key_{i+1}M \oplus R)$
and updates
$\{key_{i+1}H, key_{i+1}M, key_{i+1}L\}$

Reader

(1)Query, N

(3)α, B, μ

(5)α, B, μ

(7)$\delta, \xi, \varphi, \psi$

Tag

(2) Generates ν
$R = S(key_i H \oplus \nu)$
$\alpha = key_i M \oplus \nu$
$\beta = key_i L \oplus R$
$\mu = H(N \oplus key_i H \oplus \nu)$

Adversary

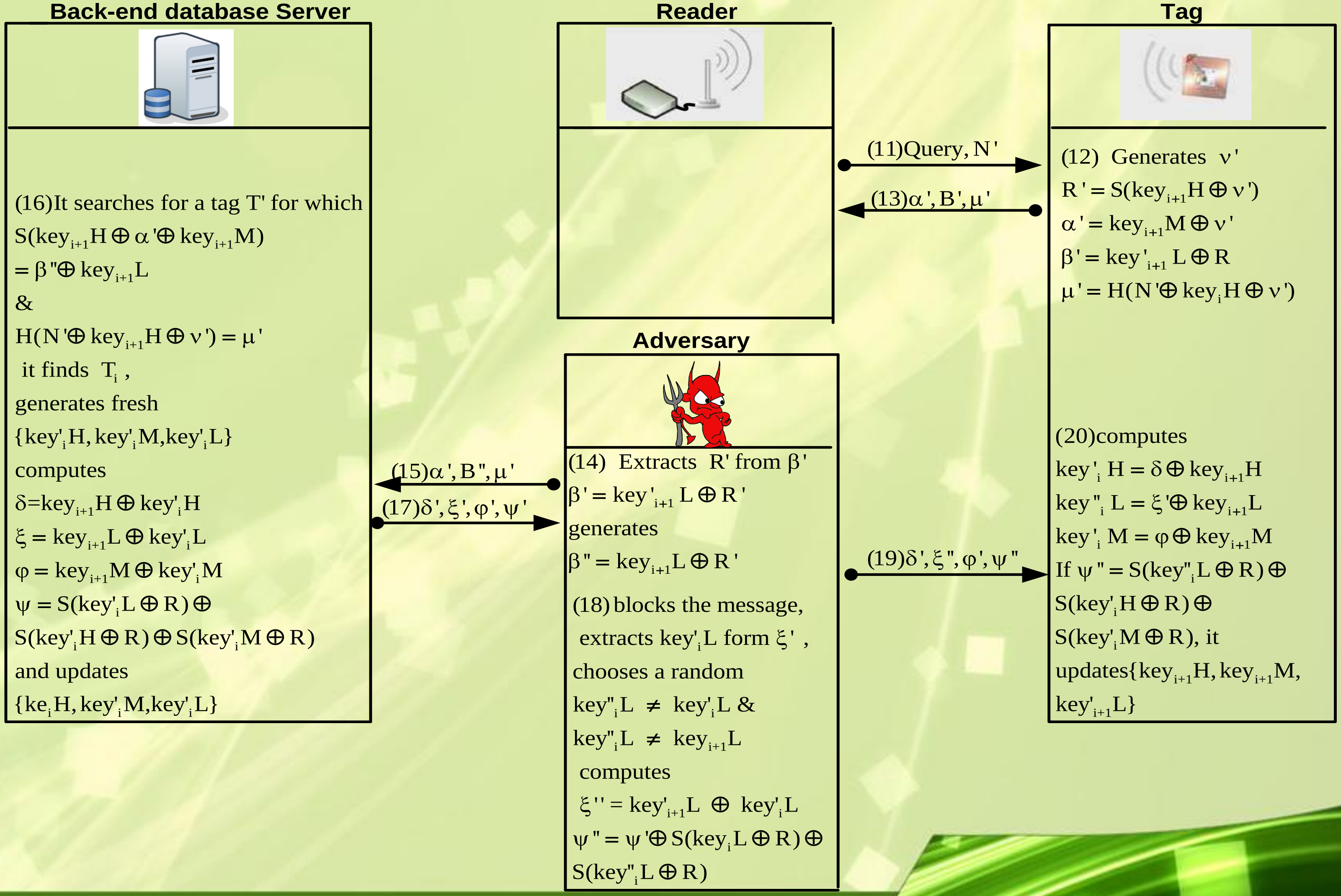
(4) Extracts R from β
$\beta = key_i L \oplus R$
(8) blocks the message,
extracts $key_{i+1}L$ form ξ ,
chooses a random
$key'_{i+1}L \neq key_{i+1}L$
computes
$\xi' = key'_{i+1}L \oplus key_i L$
$\psi' = \psi' \oplus S(key_{i+1}L \oplus R) \oplus$
$S(key'_{i+1}L \oplus R)$

(9)$\delta, \xi', \varphi, \psi'$

(10)computes
$key_{i+1}H = \delta \oplus key_i H$
$key'_{i+1} L = \xi' \oplus key_i L$
$key_{i+1}M = \varphi \oplus key_i M$
If $\psi' = S(key'_{i+1}L \oplus R) \oplus$
$S(key_{i+1}H \oplus R) \oplus$
$S(key_{i+1}M \oplus R)$, it
updates$\{key_{i+1}H, key_{i+1}M,$
$key'_{i+1}L\}$

Desynchronization Attack

Phase 2 (updating key_i): adversary forces the tag and the database to updates their record of key_iL to different values in the next consecutive session



Back-end database Server

Reader

Tag

Adversary

(11)Query, N'

(12) Generates ν'

$R' = S(key_{i+1}H \oplus \nu')$

$\alpha' = key_{i+1}M \oplus \nu'$

$\beta' = key'_{i+1}L \oplus R$

$\mu' = H(N' \oplus key_iH \oplus \nu')$

(13)α', B', μ'

(14) Extracts R' from β'

$\beta' = key'_{i+1}L \oplus R'$

generates

$\beta'' = key_{i+1}L \oplus R'$

(15)α', B'', μ'

(16)It searches for a tag T' for which

$S(key_{i+1}H \oplus \alpha' \oplus key_{i+1}M)$

$= \beta'' \oplus key_{i+1}L$

&

$H(N' \oplus key_{i+1}H \oplus \nu') = \mu'$

it finds T_i ,

generates fresh

$\{key'_iH, key'_iM, key'_iL\}$

computes

$\delta = key_{i+1}H \oplus key'_iH$

$\xi = key_{i+1}L \oplus key'_iL$

$\varphi = key_{i+1}M \oplus key'_iM$

$\psi = S(key'_iL \oplus R) \oplus$

$S(key'_iH \oplus R) \oplus S(key'_iM \oplus R)$

and updates

$\{ke_iH, key'_iM, key'_iL\}$

(17)$\delta', \xi', \varphi', \psi'$

(18) blocks the message,

extracts key'_iL form ξ' ,

chooses a random

$key''_iL \neq key'_iL$ &

$key''_iL \neq key_{i+1}L$

computes

$\xi'' = key'_{i+1}L \oplus key'_iL$

$\psi'' = \psi' \oplus S(key_iL \oplus R) \oplus$

$S(key''_iL \oplus R)$

(19)$\delta', \xi'', \varphi', \psi''$

(20)computes

$key'_i H = \delta \oplus key_{i+1}H$

$key''_i L = \xi' \oplus key_{i+1}L$

$key'_i M = \varphi \oplus key_{i+1}M$

If $\psi'' = S(key''_iL \oplus R) \oplus$

$S(key'_iH \oplus R) \oplus$

$S(key'_iM \oplus R)$, it

updates$\{key_{i+1}H, key_{i+1}M,$

$key'_{i+1}L\}$

Desynchronization Attack

After desynchronization attack:

Records of tag after attack	Records of back-end database after attack
$key_i = \{ key'_i H, key'_i M, \underline{key''_i L} \}$ $key_{i+1} = \{ key_{i+1} H, key_{i+1} M, \underline{key'_{i+1} L} \}$	$key_i = \{ key'_i H, key'_i M, \underline{key'_i L} \}$ $key_{i+1} = \{ key_{i+1} H, key_{i+1} M, \underline{key_{i+1} L} \}$

The success probability of attack is almost 1 whereas the complexity is just two sessions of protocol, given that the adversary has already extracted the related secrets.

Conclusions

- We have shown some security pitfalls in the design of RIPTA-DA protocol.

- We presented three attacks against the protocol.

- It is worth investigating the design and performance aspects of RFID protocols by using standard ciphers such as PRESENT.

Thank you