Mutual Authentication Protocol for Low-cost RFID

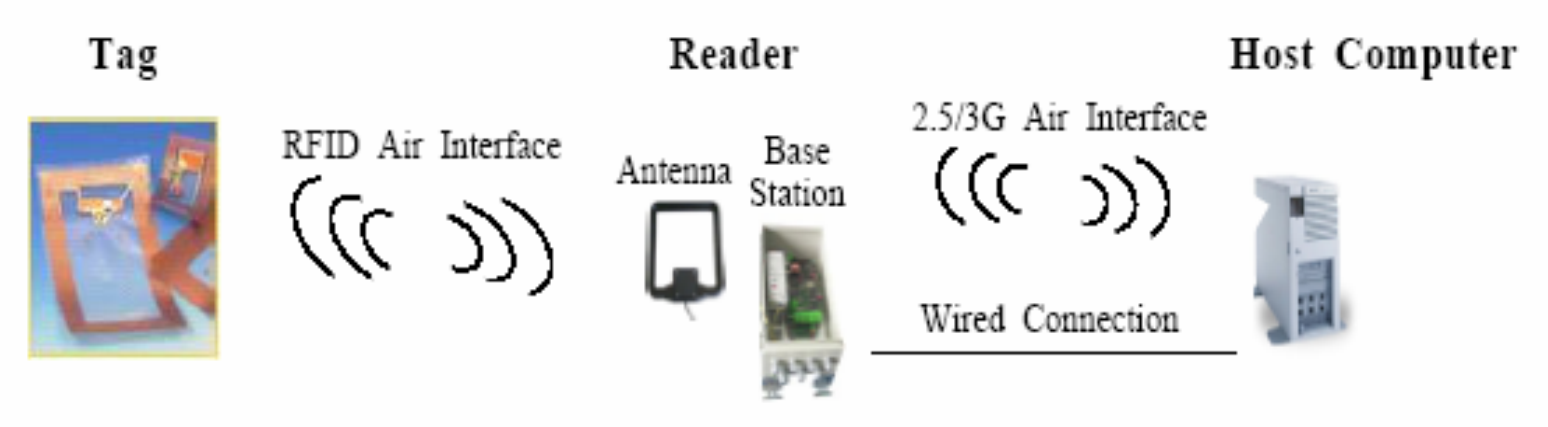
Jeongkyu Yang[1], Jaemin Park[2], Hyunrok Lee[2], Kui Ren[3], Kwangjo Kim[2]

[1]Korea Minting and Security Printing Corporation (KOMSCO)
[2]Information and Communication University (ICU)
[3]Worcester Polytechnic Institute (WPI)

Contents

1. Introduction (1/4)

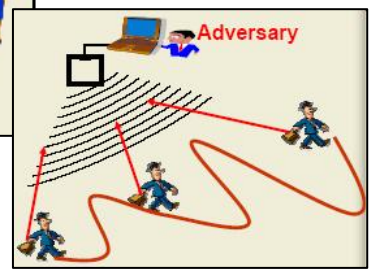
- Typical RFID System



- Characteristics
  - **ISO (Int. Standard), EPC (De-facto Standard)**
  - **Air interface – 13,56 MHz, 915 MHz, *etc.***
  - **Asymmetric communication channel**
  - **Collision avoidance IDs, Lock & Un-lock Mechanism, Pwd. Mgt.**
  - **Tag cost**
    - » To 5-cents tag, the IC cost < 2 cents

1. Introduction (2/4)

- Leakage of personal belongs data
  - **Leak data regarding belongings without awareness of user.**

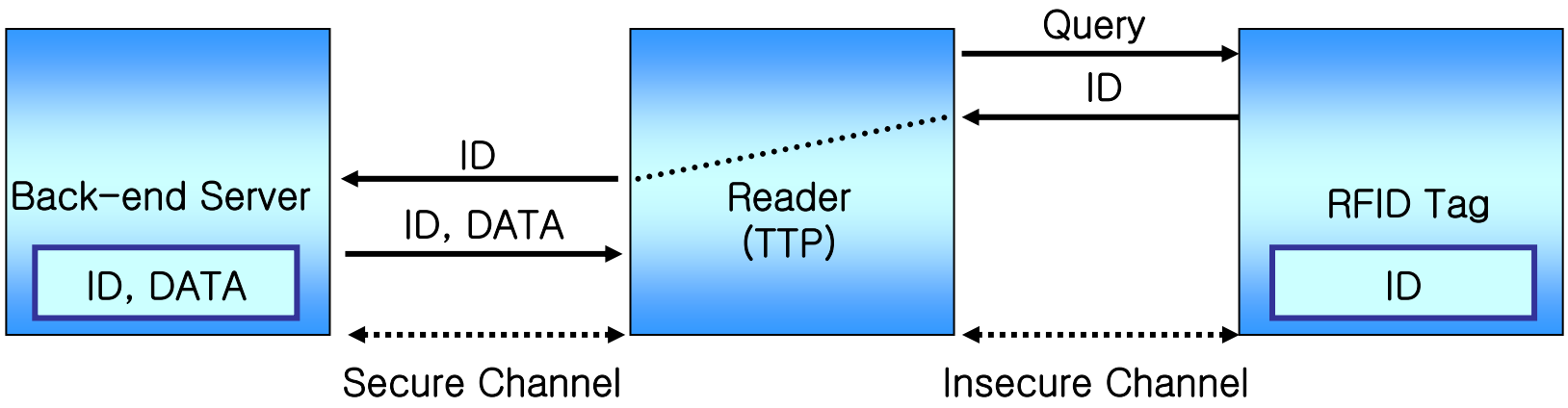


- Illegal ID tracking
  - **Monitor tag owners activities.**

- Attacks
  - **Eavesdropping**
  - **Man-in-the-middle attack (Impersonation, Spoofing)**
  - **Replay attack**
  - **Data loss (DoS, Message hijacking)**
  - **Forgery (Decoy Tag, *etc.*)**
  - **Physical attack**

1. Introduction (3/4)

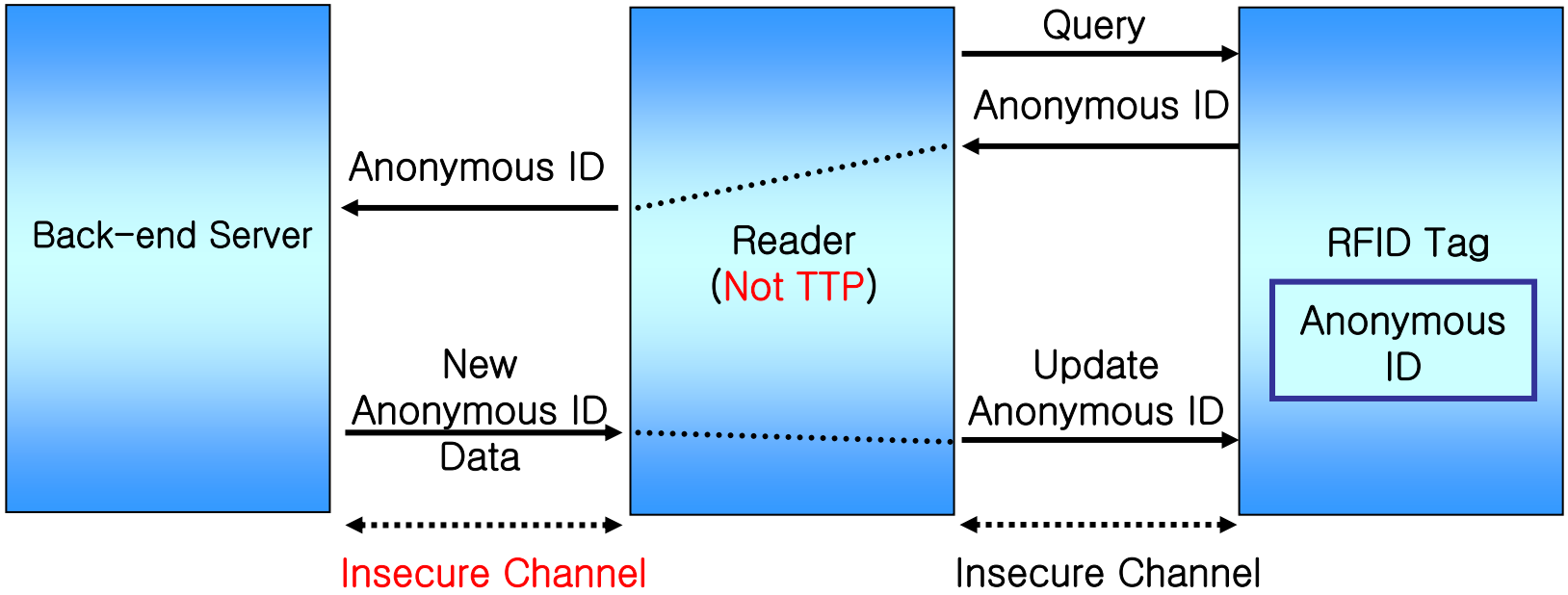
- RFID authentication



- **Low-cost RFID system environment**
  - » **Light-weight primitives**
- **Privacy protection for the tag bearers**
  - » **Data privacy & location privacy must be guaranteed.**
- **Security measure**
  - » **Mutual authentication is needed.**

1. Introduction (4/4)

- Secure authentication protocol for low-cost RFID system
  - **Using a rewritable memory like EEPROM, hash in tags**



  - **Meet low-cost RFID environment**
  - **Guarantee privacy for tag bearers**
  - **Satisfy confidentiality, anonymity, and integrity**
  - **Robust against attacks**

2. Preliminaries (1/4)

One-way hash function

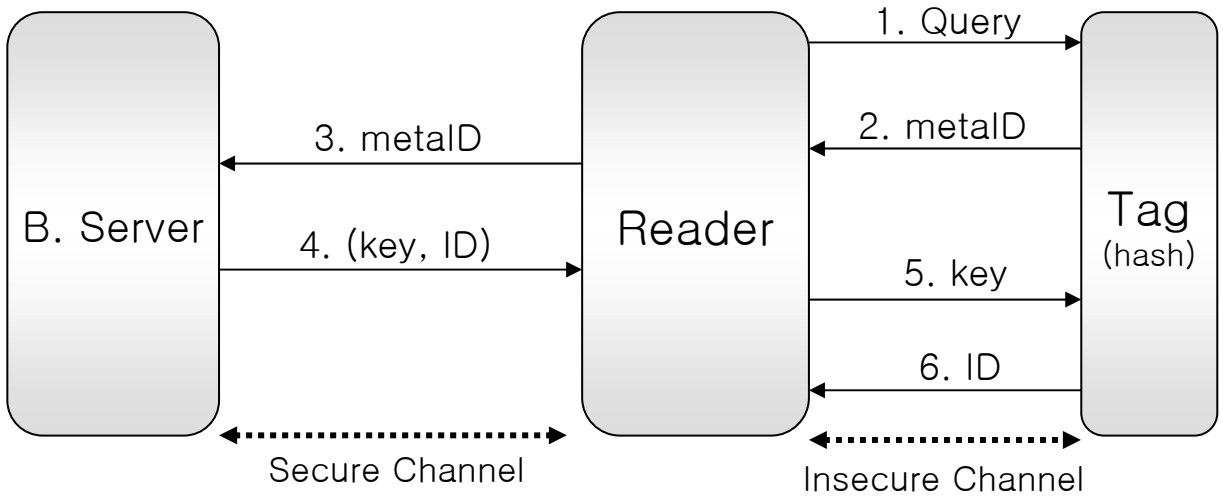
- Constrained resources of a tag
  - » # of gates is **7.5~15 K**, 100-bit EPC chip requires 5~10 K
  - » # of gates available for security < **2.5~5 K**

- Hash implementation
  - » **Ultra-Low-Power Universal Hash Functions, *by Yüksel et al. in CNDS 2004.* [9]**

Design	Dynamic Power		Leakage Power		Circuit Area		Maximum Delay	
	μW	%	μW	%	gates	%	ns	speedup
WH-64	452.3	100	9.36	100	1701	100	1.35	1.0
WH-32	217.5	48	4.81	51	873	51	1.31	1.0
WH-16	126.2	28	2.32	25	460	27	0.76	1.8

2. Preliminaries (2/4)

❑ Hash-lock Scheme (*Weis et al., SPC 2003.* [14])



metaID, key, ID

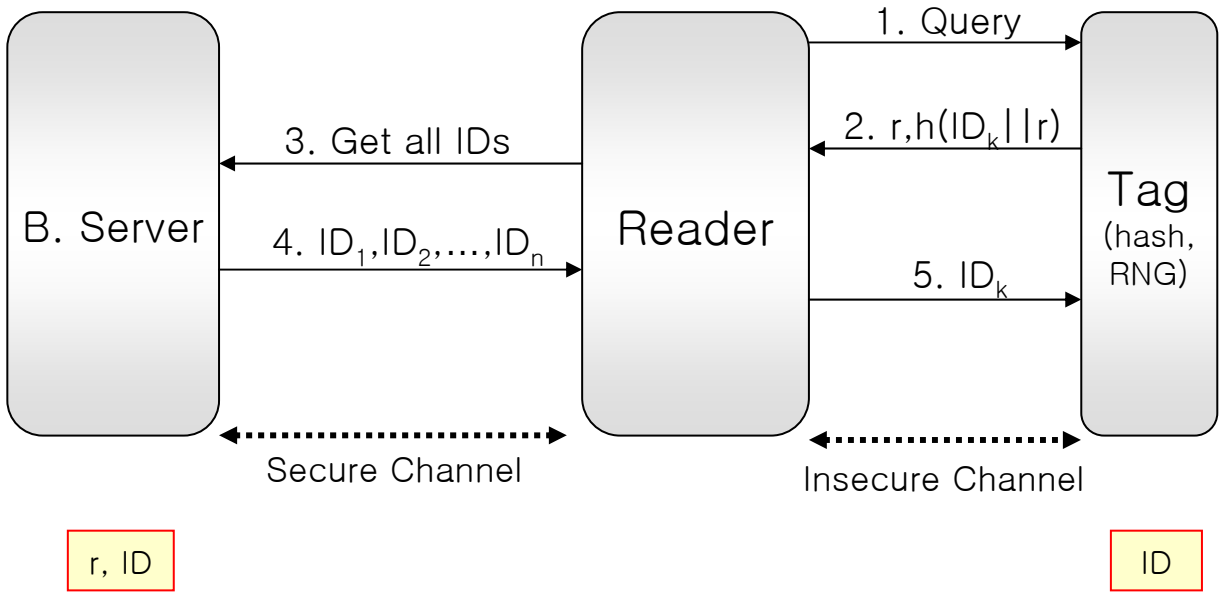
metaID, ID

metaID = H(key), where H is a hash function

- The **metaID** itself is constant and will be the target of tracking.

2. Preliminaries (3/4)

Extended Hash-lock Scheme (*Weis et al., SPC 2003.* [14])

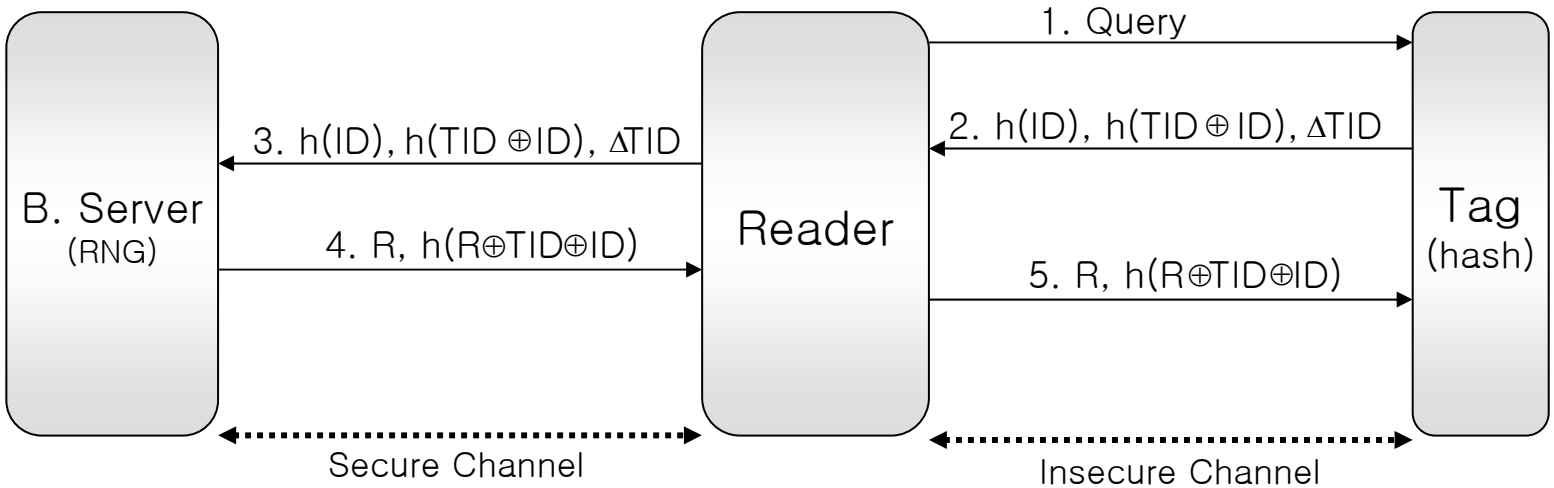


r is generated by RNG of tag

- ID is randomized, but cannot prevent man-in-the-middle attack.
- Implementation issues on **RNG** for each tag.

2. Preliminaries (4/4)

❑ Hash-based Varying Identifier (*Herici et al., PerSec'04.* [4])



HID, ID, TID, LST, AE, DATA

HID, ID, TID, LST, AE, DATA

ID, TID, LST

R is generated by RNG of Back-end Server

- ID is randomized, but cannot prevent man-in-the-middle attack.
- Tag anonymity cannot be guaranteed until the next session.

3. Proposed Scheme (1/5)

- Notations

$\mathcal{T}$	RF tag, or transponder.
$\mathcal{R}$	RF tag reader, or transceiver.
$\mathcal{B}$	Back-end server, it has a database.
D	A database of $\mathcal{B}$.
C	Chip serial number that is embedded into $\mathcal{T}$ during manufacturing.
$E_k()$	Symmetric-key cryptosystem based encryption function with the secret key, k.
$D_k()$	Symmetric-key cryptosystem based decryption function with the secret key, k.
$h()$	One-way hash function.
$h_k()$	Keyed hash function with the secret key k.
ID	Temporary identification value of $\mathcal{T}$, it is used to make the shared secret k_2 randomized.
ID'	Temporary value to be used to make the shared secret k_1 randomized.
k	Secret key shared between $\mathcal{R}$ and $\mathcal{B}$.
k_1	Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.
k_2	Shared random secret between $\mathcal{T}$ and $\mathcal{B}$.
RNG	Random Number Generator.
r	Random number generated by RNG of $\mathcal{R}$.
S	Keyed one-way hash value of $h_k(r)$.
$\oplus$	Exclusive-or (XOR) function.
$\stackrel{?}{=}$	Verification operator to check whether the left side are valid for the right side or not.
$\leftarrow$	Update operator from the right side to the left side.
HID	A field for the temporary identification value of $\mathcal{T}$ and used as a primary index.
T_1	A field for the shared random secret, k_1.
T_2	A field for the shared random secret, k_2.
AE	A field for the pointer linking a pair of records each other to counteract for the data loss.
CN	A field for the chip serial number, C, of $\mathcal{T}$.
$DATA$	A field for all other application related data of $\mathcal{T}$.

3. Proposed Scheme (2/5)

- Assumptions
  - **Hash Function**
    - » Has desirable security like 1st, 2nd preimage resistance, and collision avoidance.
  - **Tag (T)**
    - » Has a hash function, XOR gate, and the capability to keep state during a single session.
    - » Is passive and has re-writable memory like EPC class 2 of EPC Global.
  - **Reader (R)**
    - » Is not a TTP and has enough computational power.
    - » Has a *RNG* and a keyed one-way hash function with symmetric key between the reader and the back-end server.
  - **Back-end Server (B)**
    - » Has sufficient capability to manage symmetric-key cryptosystem.
  - **Insecure channel between reader and back-end server**

3. Proposed Scheme (3/5)

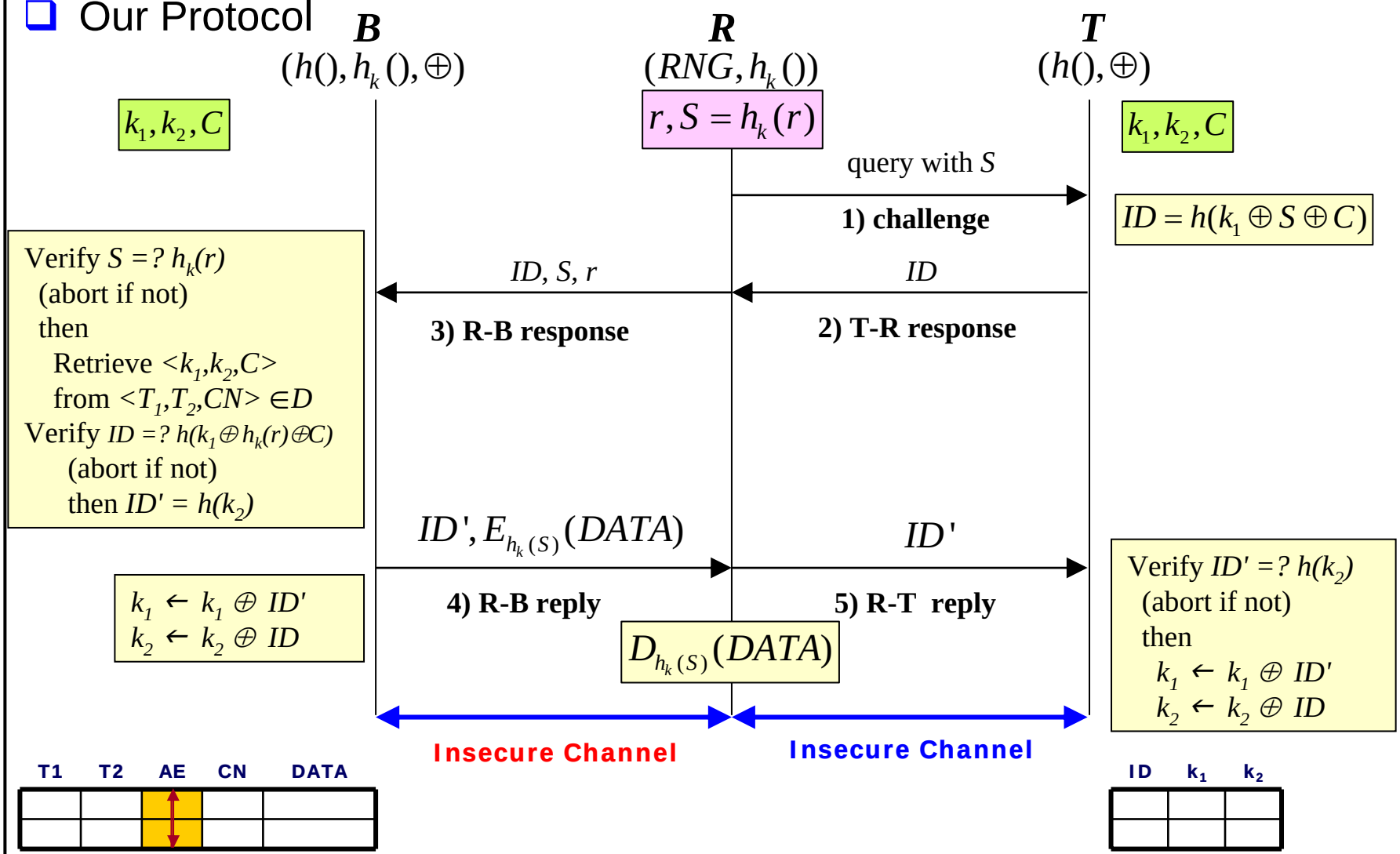
- Attacking Model
  - **Man-in-the-middle attack**
    - » The attacker can impersonate as a legitimate R and get the information from T. ➔ He can impersonate as the legitimate T responding to R.
  - **Replay attack**
    - » The attackers eavesdrop the response message from T, and can retransmit the message to the legitimate R.
  - **Forgery**
    - » The simple copy of T information by eavesdropping.
  - **Data loss**
    - » DoS, power interruption, and hijacking, *etc.*
  - **Do not consider the physical attack**

3. Proposed Scheme (4/5)

- Security Requirement
  - **Data confidentiality**
    - » To prevent the data privacy of T from the insecure data
  - **Tag anonymity**
    - » To prevent the location privacy of tag bearers
  - **Data integrity**
    - » Data integrity between T and B against data loss
    - » Linkage between the authentication info. of T and T itself ➔ Simple forgery is prevented
  - **Detection for an illegitimate R**
    - » Replay attack and Man-in-the-middle attack are prevented.

3. Proposed Scheme (5/5)

- Our Protocol



4. Correctness (1/4)

GNY Logic [20]

- L. Gong, R. Needham and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols", *1990 IEEE Computer Society Synopsis on Research in Security & Privacy*.

Correctness Proof of Our Scheme

- We applied the reasoning process of GNY logic to prove correctness of our protocol.
- Correctness of proof goals means two entities, T and B, share two secrets for every session and those secrets are fresh.
- Besides, two entities, R and B, shared the keys for providing reader authentication and secure message exchange.
- The proof goals are accomplished by the verification steps.

4. Correctness (2/4)

- **Used GNY Constructs**

(X, Y)	Concatenation of formulae	$\{X\}K$ $\{X\}K^{-1}$	Symmetric encryption and decryption
$P \ni X$	P possesses or is capable of possessing formula. X	$P \mid\!\sim X$	P conveyed X.
$P \mid\!\equiv X$	P believes X.	$\sharp(X)$	The formula X is fresh. X has not been before the current run of the protocol.
$P \lhd X$	P is told X. P has a received a message containing X and P can read and repeat X.	$P \lhd \star(X)$	P is told formula X, not conveyed by P during the current protocol run.
$X \rightsquigarrow C$	Message X has the extension C. The precondition for X being conveyed is C.	$P \Rightarrow X$	P has jurisdiction over X. The principal P is an authority on X.
ϕX	Formula X is recognizable	$P \overset{K}{\longleftrightarrow} Q$	K is a suitable secret for P and Q. It may be used as a key or as a proof of identity.
$P \overset{K}{\Leftrightarrow} Q$	K is a secret known only to P and Q, and possibly to principals trusted by them. Only P and Q may use X to prove their identities to one another. Often, K is fresh as well as secret.		

4. Correctness (3/4)

Proof Goals

1. $B \models T \mid\sim \sharp(H(K1^i \oplus H_K(N_R)))$	2. $T \models B \mid\sim \sharp(H(K2^i))$
3. $R \models R \overset{K}{\longleftrightarrow} B$	4. $B \models R \overset{K}{\longleftrightarrow} B$
5. $R \models R \overset{K_{RB}}{\longleftrightarrow} B$	6. $B \models R \overset{K_{RB}}{\longleftrightarrow} B$

- (1) and (2) for shared secrets between tag and back-end server
  - » (1) for the message from T
  - » (2) for the message from B
- (3-6) for shared keys between reader and back-end server
  - » (3) and (4) for a keyed hash function
  - » (4) and (6) for message encryption and decryption

4. Correctness (4/4)

❑ Verification

Message 5 $T \lhd \star(H(K2^i)) \rightsquigarrow T \ni K2^i$

32) The extension to the message, $T \ni K2^i$, is valid because it holds when the message is sent as is evident from the initial assumptions, A5.

33) $T \lhd H(K2^i)$: Applying T1, Being-Told Rule.

34) $T \ni H(K2^i)$: Applying P1, Possession Rule.

35) $\frac{T \mid\!\equiv \sharp(K2^i) \wedge T \ni H(K2^i)}{T \mid\!\equiv \sharp(H(K2^i))}$: Applying A7, and applying F10, Freshness Rule.

36) $\frac{T \lhd \star(H(K2^i)) \wedge T \ni K2^i \wedge T \mid\!\equiv T \overset{K2^i}{\longleftrightarrow} B \wedge T \mid\!\equiv \sharp(H(K2^i))}{T \mid\!\equiv B \mid\!\sim H(K2^i)}$: Applying A5, A9, V33, and applying I3, Message Interpretation Rule.

37) $\frac{T \mid\!\equiv \sharp(H(K2^i)) \wedge T \mid\!\equiv B \mid\!\sim H(K2^i)}{T \mid\!\equiv B \mid\!\sim \sharp(H(K2^i))}$: Applying V35, and applying F1, Freshness Rule. This is the proof for P2, $T \mid\!\equiv B \mid\!\sim \sharp(H(K2^i))$.

applying A16, V27, and the freshness $\sharp(H(K2^i), K_{RB})$ is straightforward, and applying I1, Message Interpretation Rule.

29) $R \mid\!\equiv B \mid\!\sim R \overset{K_{RB}}{\longleftrightarrow} B$: Applying I7, Message Interpretation Rule.

30) $\frac{R \mid\!\equiv B \mid\!\sim R \overset{K_{RB}}{\longleftrightarrow} B \wedge B \mid\!\equiv R \overset{K_{RB}}{\longleftrightarrow} B}{R \mid\!\equiv R \overset{K_{RB}}{\longleftrightarrow} B}$: Applying A20, and applying J1, Jurisdiction Rule. This is the proof for P5, $R \mid\!\equiv R \overset{K_{RB}}{\longleftrightarrow} B$.

31) We omit the proof for P6 since, for the encrypted message with the key, K_{RB}, there in no further message exchange after this step. That is, the encrypted message of the entity, B, is replied to R and decrypted by R. Thus, the proof is not needed at this moment.

5. Analysis (1/3)

- Security Analysis
  - **Data confidentiality**
    - » On data privacy of tag bearers
      - ✓ T does store no privacy information of tag bearers.
      - ✓ All messages from T is hashed, so eavesdropping is meaningless.
    - » On Application data
      - ✓ $E_{h_k}(S)(DATA)$ by B, and $D_{h_k}(S)(DATA)$ by R
      - ✓ $h_k(S)$: randomly created shared key between R and B
  - **Tag anonymity**
    - » All outputs of T are anonymous for every read attempt with r of R.
    - » Freshness of k_1 and k_2 is guaranteed for each session.
    - » Location privacy is protected.

5. Analysis (2/3)

- Security Analysis (cont.)
  - **Data Integrity**
    - » Synchronization between T and B by mutual authentication
    - » Providing data recovery using a pair of DB records of B
    - » Providing Linkage between the authentication info. of T and T itself using the chip S/N
  - **Availability**
    - » Man-in-the-middle attack prevention ➔ Step 3, and step 5
    - » Unauthorized reader detection ➔ From step 1 to step 3
    - » Replay attack prevention ➔ Step 3 for B, and step 5 for T
    - » Forgery resistance ➔ C of ID by B
    - » Data recovery ➔ Step 4

5. Analysis (3/3)

- Performance Analysis
  - **Computational Overhead**
    - ✓ ***T*** **needs only 2 hash calculation**
    - ✓ **Encryption & decryption for insecure channel** ***B*** ***needs*** ***2n*** **of hash calculation, when** ***n*** **is number of** ***T***
  - **Storage Overhead**
    - » ***T*** **needs only 2½L bits, when** $h, h_k : \{0,1\}^* \rightarrow \{0,1\}^{\frac{1}{2}L}$ **and** $r \in_U \{0,1\}^L$
  - **Communication Overhead**
    - » **Message exchange: total – 5, between** ***T*** **and** ***R*** **- 2**
  - **Cost Overhead**
    - » 1.7 K-gate/hash + several hundreds gates/XOR < 2.5 K ~ 5 K-gate
      ➔ Feasible to 5 cents tag

6. Comparison (1/2)

- Security Comparison

Protocol	HLS [18]	EHLS [18]	HBVI [7]	Our Scheme
User data confidentiality	×	△	△	○
Tag anonymity	×	△	△	○
Data integrity	△	△	○	○
Mutual authentication	△	△	△	○
Reader authentication	×	×	×	○
Man-in-the-middle attack prevention	△	△	×	○
Replay attack prevention	△	△	○	○
Forgery Resistance	×	×	×	○
Data Recovery	×	×	○	○

†† Notation

○ satisfied △ partially satisfied × not satisfied

6. Comparison (2/2)

- Performance Comparison

Protocol	Entities	HLS [18]	EHLS [18]	HBVI [7]	Our Scheme
No. of Hash Operation	$\mathcal{T}$	1	2	3	2
	$\mathcal{B}$	¬	n	3	2n
No. of Keyed Hash Operation	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	¬	1
No. of RNG Operation	$\mathcal{T}$	¬	1	¬	¬
	$\mathcal{R}$	¬	¬	¬	1
	$\mathcal{B}$	¬	¬	1	¬
No. of Encryption	$\mathcal{B}$	¬	¬	¬	1
No. of Decryption	$\mathcal{R}$	¬	¬	¬	1
Number of Authentication Steps		6	5	5	5
Required Memory Size	$\mathcal{T}$	$1\frac{1}{2}L$	$1L$	$3L$	$2\frac{1}{2}L$
	$\mathcal{R}$	¬	¬	¬	$1\frac{1}{2}L$
	$\mathcal{B}$	$2\frac{1}{2}L$	$1\frac{1}{2}L$	$9L$	$8L$

†† Notation ¬ not required

n number of tags L size of required memory

- ***L*** bits is assumed for the sizes of all components between protocols
- The outputs of hash function is ½***L*** bits
- Comparison for ***DATA*** is excluded since its size is depended on application.

7. Conclusion

- RFID will be important for the future ubiquitous society. However, RFID systems are vulnerable to many security risks and imply potential privacy problems.

- Different from previous results, our protocol is firstly proposed on the assumption that the communication channel between reader and back-end server is insecure and reader is not TTP.

- As based on strong mutual authentication between entities, our protocol is robust enough for security vulnerabilities and privacy problems, and is very feasible for low-cost RFID environment since tag only has a hash function with small memory size.

Thanks for your attention!

Q&A

3. Proposed Scheme (5/5)

- Our Protocol