Mutual Authentication Protocol
for Low-cost RFID

Jeongkyu Yang\textsuperscript{1}, Jaemin Park\textsuperscript{2}, Hyunrok Lee\textsuperscript{2}, Kui Ren\textsuperscript{3}, Kwangjo Kim\textsuperscript{2}

\textsuperscript{1}Korea Minting and Security Printing Corporation (KOMSCO)
\textsuperscript{2}Information and Communication University (ICU)
\textsuperscript{3}Worcester Polytechnic Institute (WPI)
<table>
<thead>
<tr>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
</tr>
<tr>
<td>2. Preliminaries</td>
</tr>
<tr>
<td>3. Proposed Scheme</td>
</tr>
<tr>
<td>4. Correctness</td>
</tr>
<tr>
<td>5. Security &amp; Performance Analysis</td>
</tr>
<tr>
<td>6. Comparison</td>
</tr>
<tr>
<td>7. Conclusion</td>
</tr>
</tbody>
</table>
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.
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

<table>
<thead>
<tr>
<th>Design</th>
<th>Dynamic Power $\mu$W</th>
<th>Leakage Power $\mu$W</th>
<th>Circuit Area gates</th>
<th>Maximum Delay ns</th>
<th>speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>WH-64</td>
<td>452.3 100</td>
<td>9.36 100</td>
<td>1701 100</td>
<td>1.35 1.0</td>
<td></td>
</tr>
<tr>
<td>WH-32</td>
<td>217.5 48</td>
<td>4.81 51</td>
<td>873 51</td>
<td>1.31 1.0</td>
<td></td>
</tr>
<tr>
<td>WH-16</td>
<td>126.2 28</td>
<td>2.32 25</td>
<td>460 27</td>
<td>0.76 1.8</td>
<td></td>
</tr>
</tbody>
</table>
2. Preliminaries (2/4)

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

- **B. Server**

- Reader

- **Tag** (hash)

- 1. Query
- 2. metaID
- 3. metaID
- 4. (key, ID)
- 5. key
- 6. ID

- Secure Channel
- Insecure Channel

- 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])

- **B. Server**
  - 1. Query
  - 3. Get all IDs
  - 4. $ID_1, ID_2, ..., ID_n$

- **Reader**
  - 2. $r.h(ID_k || r)$

- **Tag**
  - (hash, RNG)
  - 5. $ID_k$

- **Secure Channel**
- **Insecure Channel**

- $r$ 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]*)

- **B. Server (RNG)**
  - 3. $h(ID), h(TID \oplus ID), \Delta TID$
  - 4. $R, h(R \oplus TID \oplus ID)$

- **Reader**
  - 2. $h(ID), h(TID \oplus ID), \Delta TID$
  - 5. $R, h(R \oplus TID \oplus ID)$

- **Tag (hash)**
  - 1. Query
  - 1. Query

**HID, ID, TID, LST, AE, DATA**

**Insecure Channel**

**Secure Channel**

- 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**

  - $T$: RF tag, or transponder.
  - $R$: RF tag reader, or transceiver.
  - $B$: Back-end server, it has a database.
  - $D$: A database of $B$.
  - $C$: Chip serial number that is embedded into $T$ 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 $T$, 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 $R$ and $B$.
  - $k_1$: Shared random secret between $T$ and $B$.
  - $k_2$: Shared random secret between $T$ and $B$.
  - $RNG$: Random Number Generator.
  - $r$: Random number generated by $RNG$ of $R$.
  - $S$: Keyed one-way hash value of $h_k(r)$.
  - $\oplus$: Exclusive-or (XOR) function.
  - $\equiv$: 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 $T$ 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 $T$.
  - $DATA$: A field for all other application related data of $T$. 
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

\[ (h(), h_k(), \oplus) \]

\[ k_1, k_2, C \]

\[ RTB \]

\[ k_1 \]

\[ \oplus \]

\[ ID' \]

\[ k_2 \]

\[ \oplus \]

\[ ID \]

Verify \( S = ? h_k(r) \) 
(abort if not) 
then 
Retrieve \( <k_1, k_2, C> \) 
from \( <T_1, T_2, CN> \in D \) 
Verify \( ID = ? h(k_1 \oplus h_k(r) \oplus C) \) 
(abort if not) 
then \( ID' = h(k_2) \)

\[ k_1 \leftarrow k_1 \oplus ID' \]
\[ k_2 \leftarrow k_2 \oplus ID \]

\[ (RNG, h_k()) \]

\[ r, S = h_k(r) \]

query with \( S \)

\[ ID \]

1) challenge

2) T-R response

3) R-B response

4) R-B reply

5) R-T reply

\[ ID', E_{h_k(S)}(DATA) \]

\[ D_{h_k(S)}(DATA) \]

Insecure Channel

Insecure Channel

\[ ID = h(k_1 \oplus S \oplus C) \]

\[ k_1, k_2, C \]

\[ T \]

\[ h() \]

\[ h() \]

\[ h() \]

\[ (h(), \oplus) \]

\[ ID \]

\[ k_1 \]

\[ \oplus \]

\[ k_2 \]

\[ ID \]

Verify \( ID' = ? h(k_2) \) 
(abort if not) 
then 
\[ k_1 \leftarrow k_1 \oplus ID' \]
\[ k_2 \leftarrow k_2 \oplus ID \]
4. Correctness (1/4)

- **GNY Logic [20]**

- **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**

<table>
<thead>
<tr>
<th>$(X, Y)$</th>
<th>Concatenation of formulae</th>
<th>${X}^K \cdot {X}^{K^{-1}}$</th>
<th>Symmetric encryption and decryption</th>
</tr>
</thead>
<tbody>
<tr>
<td>$P \triangleright X$</td>
<td>$P$ possesses or is capable of possessing formula, $X$</td>
<td>$P \triangleright X$</td>
<td>$P$ conveyed $X$.</td>
</tr>
<tr>
<td>$P \bowtie X$</td>
<td>$P$ believes $X$.</td>
<td>$#(X)$</td>
<td>The formula $X$ is fresh. $X$ has not been before the current run of the protocol.</td>
</tr>
<tr>
<td>$P \lhd X$</td>
<td>$P$ is told $X$. $P$ has a received a message containing $X$ and $P$ can read and repeat $X$.</td>
<td>$P \lhd X$</td>
<td>$P$ is told formula $X$, not conveyed by $P$ during the current protocol run.</td>
</tr>
<tr>
<td>$X \sim C$</td>
<td>Message $X$ has the extension $C$. The precondition for $X$ being conveyed is $C$.</td>
<td>$P \Rightarrow X$</td>
<td>$P$ has jurisdiction over $X$. The principal $P$ is an authority on $X$.</td>
</tr>
<tr>
<td>$\phi X$</td>
<td>Formula $X$ is recognizable</td>
<td>$P \leftrightarrow Q$</td>
<td>$K$ is a suitable secret for $P$ and $Q$. It may be used as a key or as a proof of identity.</td>
</tr>
<tr>
<td>$P \overset{K}{\Rightarrow} Q$</td>
<td>$K$ 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.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. Correctness (3/4)

- **Proof Goals**

  1. $B \equiv T \vdash H(K1^i \oplus H_K(N_R))$
  2. $T \equiv B \vdash H(K2^i)$
  3. $R \equiv R \overset{K}{\leftarrow} B$
  4. $B \equiv R \overset{K}{\rightarrow} B$
  5. $R \equiv R \overset{K_{RB}}{\leftarrow} B$
  6. $B \equiv R \overset{K_{RB}}{\rightarrow} B$

- (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** $T \triangleleft \star (H(K2^i)) \leadsto T \ni K2^i$

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

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

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

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

37) $\frac{T \equiv \sharp(H(K2^i)) \wedge T \ni B \sim \sharp(H(K2^i))}{T \equiv B \sim \sharp(H(K2^i))}$: Applying V35, and applying F1, Freshness Rule. This is the proof for P2, $T \ni B \sim \sharp(H(K2^i))$ applying A10, V21, and the freshness $\sharp(H(K2^i), R_{RB})$ is straightforward, and applying I1, Message Interpretation Rule.

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

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

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
    - $E_{hk(S)}(DATA)$ by $B$, and $D_{hk(S)}(DATA)$ 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 $2^{1/2}L$ bits, when $h, h_k : \{0, 1\}^* \rightarrow \{0, 1\}^{1/2L}$ 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

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User data confidentiality</td>
<td>×</td>
<td>Δ</td>
<td>Δ</td>
<td>○</td>
</tr>
<tr>
<td>Tag anonymity</td>
<td>×</td>
<td>Δ</td>
<td>Δ</td>
<td>○</td>
</tr>
<tr>
<td>Data integrity</td>
<td>Δ</td>
<td>Δ</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Mutual authentication</td>
<td>Δ</td>
<td>Δ</td>
<td>Δ</td>
<td>○</td>
</tr>
<tr>
<td>Reader authentication</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>Man-in-the-middle attack prevention</td>
<td>Δ</td>
<td>Δ</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>Replay attack prevention</td>
<td>Δ</td>
<td>Δ</td>
<td>○</td>
<td>○</td>
</tr>
<tr>
<td>Forgery Resistance</td>
<td>×</td>
<td>×</td>
<td>×</td>
<td>○</td>
</tr>
<tr>
<td>Data Recovery</td>
<td>×</td>
<td>×</td>
<td>○</td>
<td>○</td>
</tr>
</tbody>
</table>

†† Notation

- ○ satisfied
- Δ partially satisfied
- × not satisfied
## 6. Comparison (2/2)

### Performance Comparison

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Hash Operation</td>
<td>$T$</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>$n$</td>
<td>3</td>
<td>$2n$</td>
<td></td>
</tr>
<tr>
<td>No. of Keyed Hash Operation</td>
<td>$R$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
</tr>
<tr>
<td>No. of RNG Operation</td>
<td>$T$</td>
<td>$-$</td>
<td>1</td>
<td>$-$</td>
<td>$-$</td>
</tr>
<tr>
<td></td>
<td>$R$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
<td>$-$</td>
</tr>
<tr>
<td>No. of Encryption</td>
<td>$B$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
</tr>
<tr>
<td>No. of Decryption</td>
<td>$R$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>1</td>
</tr>
<tr>
<td>Number of Authentication Steps</td>
<td>6</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
</tr>
<tr>
<td>Required Memory Size</td>
<td>$T$</td>
<td>$1\frac{1}{2}L$</td>
<td>$1L$</td>
<td>$3L$</td>
<td>$2\frac{1}{2}L$</td>
</tr>
<tr>
<td></td>
<td>$R$</td>
<td>$-$</td>
<td>$-$</td>
<td>$-$</td>
<td>$1\frac{1}{2}L$</td>
</tr>
<tr>
<td></td>
<td>$B$</td>
<td>$2\frac{1}{2}L$</td>
<td>$1\frac{1}{2}L$</td>
<td>$9L$</td>
<td>$8L$</td>
</tr>
</tbody>
</table>

†† Notation $-$ not required

- $L$ bits is assumed for the sizes of all components between protocols
- The outputs of hash function is $\frac{1}{2}L$ 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

B

(h(), h_k(), ⊕)

R

(RNG, h_k())

T

(h(), ⊕)

\[ k_1, k_2, C \]

\[ k_{1}, k_{2}, C \]

\[ k_1 \leftarrow k_1 \oplus ID' \]

\[ k_2 \leftarrow k_2 \oplus ID \]

Verify \( S =? h_k(r) \)

(abort if not)

then

Retrieve \( <k_1, k_2, C> \)

from \( <T_1, T_2, CN> \in D \)

Verify \( ID =? h(k_1 \oplus h_k(r) \oplus C) \)

(abort if not)

then \( ID' = h(k_2) \)

\[ h \]

\[ r, S = h_k(r) \]

query with \( S \)

1) challenge

ID, S, r

2) T-R response

ID

3) R-B response

ID', E_{h_k(S)}(DATA)

4) R-B reply

ID, S, r

5) R-T reply

\[ D_{h_k(S)}(DATA) \]

Insecure Channel

Insecure Channel

\[ ID = h(k_1 \oplus S \oplus C) \]

Verify \( ID' =? h(k_2) \)

(abort if not)

then

\[ k_1 \leftarrow k_1 \oplus ID' \]

\[ k_2 \leftarrow k_2 \oplus ID \]