Revisiting RFID Security and Privacy

Ivan Visconti
University of Salerno (ITALY) and UCLA (USA)
Radio Frequency Identification

System Model [Avo05, JW06, Vau07]

Contactless smartcards
- Passive (powered by readers)
- Short communication range (¼ 1m)
- Limited memory (¼ 1 kByte)

Applications
- ePass
- eTicket
- Access Control
- Credit Cards
- Supply Chain

- At most symmetric crypto but no PKC
- (Pseudo) Random Number Generator
- Not tamper-resistant
Outline

1. Requirement Analysis
2. Existing Solutions, Focus on [Va07]
3. Revisiting RFID Security & Privacy
4. Conclusion
Trust and Adversary Model

Physical Attacks
- Timing attacks
- Power analysis,
- etc.

Adversary A
- Passive/Active Protocol Attacks
  - Man-in-the-middle
  - Replay
  - etc.
-Auxiliary Information

Fully trusted

Identify/Authenticate

Tag $T_i$ ($ID_i, K_i$)

Reader R

Backend

Auxiliary Information [JW06, Vau07]: A can observe whether a user is granted access (e.g., if a door opens) after his tag $T_i$ has been identified (A learns whether $T_i$ is valid).
Threats

• **Privacy related**
  – **Identification**: Disclosure of the tag identity
  – **Tracking**: Linkability of the transactions of a tag
=> **Creation (and misuse) of user-profiles**

• **Security related**
  – ** Forgery**: Creation of a new valid tag
  – **Impersonation**: Simulation of a valid tag
  – **Denial of Service**: Destruction of a valid tag
=> **Financial losses**
Requirements

- **Privacy** (even if the tag’s state has been disclosed)
  - **Anonymity**: Confidentiality of the tag identity
  - **Untraceability**: Unlinkability of tag’s transactions

- **Security**
  - **Authentication**: Only valid tags are accepted by reader
  - **Availability**: Manipulations of valid tickets such that honest readers do not accept them should be infeasible

- **Functional**
  - **Efficiency**: Fast identification of cost-efficient tags
  - **Scalability**: Support of a huge amount of tags
Problems

• **Existing solutions** do not satisfy requirements with good tradeoff

• **Existing formal definitions** are not general enough to provide formal framework for RFID security and privacy
Outline

1. Requirement Analysis
2. Existing Solutions, Focus on [Va07]
3. Revisiting RFID Security & Privacy
4. Conclusion
Existing Solutions

• **Practice**
  – Often non-public specifications
  – Usually no privacy aspects considered
    • e.g., [Oys05, Cal08, Son08, NXP08]
  – **Cost-efficient RFIDs lack computational resources**
    • No public-key cryptography available [Atm07, NXP08, Son08]

• **Literature**
  – **Often tradeoff between security and privacy (usually security loss)**
    • Denial-of-service attacks possible (e.g., [HM04, OSK04, Dim06, ACM04])
    • Replay attacks possible (e.g., [Dim06, ACM04])
  – **Do not fulfill functional requirements for practice**
    • Inefficient tag verification (e.g., [WSRE03, MW04, OSK04, Tsu06])
    • Number of tags fixed at system initialization (e.g., [MW04, OSK04, Dim06])
    • Require public-key cryptography (e.g., [BCDF06, Vau07])
Existing Security and Privacy Models

• **Are often incomparable or incompatible**
  – Protocols proven secure and private in one model can be attacked in other models \[HMZH08\]

• **Often do not capture important aspects like**
  – Tag corruption (disclosure of tag state) \[OSK03, Jul04, BLM06\]
  – Auxiliary information (on whether a tag is valid) \[OSK03, Jul04, Avo05, ADO05, BLM06\]
Summing Up

• **The number of models, definitions and proposals for RFID is continuously increasing**
  – Often authors rediscover/miss what has already been done for identification schemes
  – The literature now includes a huge number of incomparable papers and it is not clear which results should be used in practice and which ones should not

• **Revisiting RFID Security & Privacy**
  – We should instead have a common and flexible framework where possibility and impossibility results, open problems and improvements can be formally claimed

• **One step in this direction has been done in [Va07]**
  – Generalizes and classifies most previous privacy notions
  – Defines 8 privacy levels modeling different adversary classes
Adversary Model

Oracles defined in [Vau07]

Access to tags (models reading range)

Communication with reader

Access to auxiliary information

Setup, communication, and corruption of tags
Adversary Model

Oracles defined in [Vau07]

CreateTag

DrawTag

FreeTag

SendTag

CorruptTag

SendReader

Launch

Result

Adv

msg, vtag

S

ID, b

Δ

vtag, vtag...

msg, vtag

msg

msg, vπ

msg

b

π

π

vtag

S

Δ

vtag, vtag...
Privacy Definition [Vau07]

Game\(_0\)  ·  Adversary \(A\)  ·  Game\(_1\)

- **Real RFID Scheme**
- **Simulation of tag and reader communication** (SendTag, SendReader, Result)
- **Real RFID Scheme**

**Winning condition:** \(A\) decides correctly which game is being played

**RFID scheme is private, if** \(A\) **wins with at most probability \(\cdot 1/2\)**

**Intention:** No privacy leakage by tag-reader communication and aux information
Security Definition [Vau07]

Winning condition:
1. Reader identifies tag ID
2. State of tag ID has not been revealed to A (no corruption of ID)
3. Transcript is not just a copy of a Tag-Reader conversation

RFID scheme is secure, if A wins with at most negligible probability

Intention: Infeasibility of tag impersonation
(Replay attacks and tag corruption not considered)
**Privacy Notions and Results** [Vau07]

**Weak**  
A cannot corrupt tags (tamper-evident tags)

**Forward**  
A can only corrupt tags after interacting with the system (e.g., shut-down)

**Destructive**  
A destroys tags by corrupting them (tamper-evident hardware)

**Strong**  
A can corrupt tags at any time during his attack (most available tags)

<table>
<thead>
<tr>
<th></th>
<th>Narrow Weak</th>
<th>Narrow Forward</th>
<th>Narrow Destructive</th>
<th>Narrow Strong</th>
<th>Weak</th>
<th>Forward</th>
<th>Destructive</th>
<th>Strong</th>
</tr>
</thead>
<tbody>
<tr>
<td>SK-Crypto w/ RO</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
<td>?</td>
<td>?</td>
<td>X</td>
</tr>
<tr>
<td>PK-Crypto</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>?</td>
<td>X</td>
</tr>
</tbody>
</table>

A has no auxiliary information
Mutual Authentication [PV08]

• The tag outputs a bit

• [PV08] achieves mutual authentication with
  – Narrow-strong and forward privacy (using PKC)
  – Narrow-destructive privacy (using Random Oracles)
  – Weak privacy (using PRFs)
Outline

1. Requirement Analysis
2. Existing Solutions, Focus on [Va07]
3. Revisiting RFID Security & Privacy
4. Conclusion
Revisiting RFID Security & Privacy 1/3 [SVW09]

• **Formal framework for anonymizer-enabled RFID systems**
  – First general security and privacy model capturing anonymizer-enabled RFID systems

• **Privacy-preserving RFID system**
  – Narrow-strong and forward private tag identification
  – Authentication even if all anonymizers are corrupted
  – Basic availability (DoS requires corruption of anonymizer)
  – Efficient and scalable
  – Provable secure in the anonymizer-enabled model (with RO)
Anonymizer-Enabled RFID Systems [GJJS04, SRS04, ACM05]

Anonymizer

- Ensures untraceability of tag’s transactions with readers
- Frequently interacts with (i.e., anonymizes) tags
- Takes off computational workload (i.e., public-key crypto) from tags

Possible implementations (business models):

- **Public Anonymizer**: Privacy service provider chosen by user
- **Personal Anonymizer**: Software on user’s mobile phone/PDA
Trust and Adversary Model

=> Anonymizer opens new attack surfaces
=> Must be carefully considered when extending existing RFID security and privacy models to support anonymizers!
Anonymizer-Enabled Security and Privacy Model

Our extension

CreateAnon

LaunchAnon

SendAnon

CorruptAnon

Adversary A

T \{vtag_i\} = ID_i

DrawTag

FreeTag

CreateTag

LaunchIdent

SendTag

CorruptTag

SendReader

Result

Oracles defined in [V07]
Anonymization Protocol

**Tag**

- $K$: Authentication secret
- $E = \text{Enc}_{pk_A}(K_A)$
- $K_T$: Randomness
- $F = \text{Enc}_{pk_R}(K_T)$
- $G = \text{Enc}_{pk_R}(ID)$

Choose random $N_T$

- $(K'_T, F^*, G^*, N'_T) = \text{Dec}_{K_A}(S)$
- If $N'_T = N_T$ then
  - $K'_T = K_T \pm_{p} K^*_T$
  - $F' = F \pm_{c} F^*$
  - $G' = G \pm_{c} G^*$

Update $(K_T, F, G)$ with $(K'_T, F', G')$

**Anonymizer**

- $sk_A$: Decryption key
- $pk_A$: Encryption key
- $pk_R$: Reader’s encryption key

Choose random $K^*_T$

- $F^* = \text{Enc}_{pk_R}(K^*_T)$
- $G^* = \text{Enc}_{pk_R}(1_P)$
- $K'_A = \text{Dec}_{sk_R}(E)$
- $S = \text{Enc}_{K'_A}(K^*_T, F^*, G^*, N_T)$
Identification Protocol

Tag

\[ K \quad \text{Authentication secret} \]
\[ K_T \quad \text{Randomness} \]
\[ F = \text{Enc}_{pk_R}(K_T) \]
\[ G = \text{Enc}_{pk_R}(ID) \]

Reader

\[ \text{sk}_R \quad \text{Decryption key} \]
\[ \text{pk}_R \quad \text{Encryption key} \]

\( (ID, K) \) For each tag

Choose random \( N_R \)

\[ D = H(N_R, F, G, K_T, K) \]

\[ D', F, G \]

\[ ID' = \text{Dec}_{sk_R}(G) \]
\[ K'_T = \text{Dec}_{sk_R}(F) \]

Obtain \( (ID', K') \)

\[ D' = H(N_R, F, G, K'_T, K') \]

Accept tag if \( D' = D \)

Else reject tag
Revisiting RFID Security & Privacy 2/3 [DSV09]

• [Va07] does not consider DoS attacks

• [DSV09]: When adding an oracle query to model a DoS attacks no privacy can be achieved
  – create two tags
  – MakeInactive one tag
  – execute a protocol with a random tag
  – distinguish inactive tag by looking at the protocol transcript
    (a similar analysis was used in [Va07] to show that a narrow-destructive protocol is not forward private)

• [DSV09]: new definitions where privacy is disconnected from DoS attacks, and thus potentially achievable
Revisiting RFID Security & Privacy 2/3 [DSV09]

- **[DSV09] semi-destructive privacy (definition):** corrupt is possible when the tag is out of the range of the reader, otherwise it destroys the tag.
- **[DSV09] semi-destructive privacy (protocol):** based on symmetric-key crypto we achieve this notion with an efficient protocol that however makes DoS attacks even easier.

```plaintext
<table>
<thead>
<tr>
<th>Tag</th>
<th>Reader</th>
</tr>
</thead>
<tbody>
<tr>
<td>((K, b))</td>
<td>((\ldots, (ID, K), \ldots))</td>
</tr>
<tr>
<td>(b \in {0, 1}^n)</td>
<td>pick (a \in {0, 1}^n)</td>
</tr>
<tr>
<td>(K' \in {0, 1}^n)</td>
<td>(c = Enc_K(a</td>
</tr>
<tr>
<td>(K \leftarrow K')</td>
<td>()</td>
</tr>
<tr>
<td><strong>if</strong> (b' \neq b)**</td>
<td><strong>if</strong> check fails or not found**</td>
</tr>
<tr>
<td>()</td>
<td>(b' \leftarrow 0^n)**</td>
</tr>
<tr>
<td>()</td>
<td>output (\perp)**</td>
</tr>
<tr>
<td>()</td>
<td>()</td>
</tr>
</tbody>
</table>
```

**Protocol Description:**
- Pick \(b \in \{0, 1\}^n\) and \(K' \in \{0, 1\}^n\).
- Compute \(c = Enc_K(a|b|K')\) and send to the reader.
- The reader checks if \(a = a'\) and replaces \(K\) with \(K''\). If not, it outputs \(\perp\).
- If \(b' \neq b\), the tag is killed.
Revisiting RFID Security & Privacy 3/3 [ASVW09]

- [PV08] shows that mutual authentication is achievable along with several notion of privacy and without paying a significant overhead
  - 1st protocol for weak privacy, 2nd protocol for narrow-destructive privacy, 3rd protocol for forward and narrow-strong privacy (the paper does not include any proof)
- [ASVW09] proves INSTEAD that narrow-forward mutual authentication is impossible
  - This implies that mutual authentication is impossible when tag corruption is possible
  - This implies 2nd and 3rd protocols showed in [PV08] are flawed and moreover can not be fixed
- [ASVW09] reconsiders the [DSV09] definition of privacy where corrupt is possible when tag is off-line
  - We show that mutual authentication is possible under this restricted narrow-forward privacy notion (thus going beyond the above impossibility result)
- [ASVW09] shows that under the restricted narrow-strong notion, mutual authentication is still impossible
  - This implies that the 3rd protocol of [PV08] unfortunately has a second conceptually independent flaw
Revisiting RFID Security & Privacy 3/3 [ASVW09]

- **Where is the problem?**
  - The tag needs to keep some temporary information during the execution of the protocol in order to authenticate the reader
  - Corruption done in the middle of the protocol would reveal such an information
  - The Blinder has no strategy to avoid this attack

- **Is this a form of adaptive corruption?**
  - Basically yes, and identification schemes secure against adaptive attack have not been studied in depth, though some schemes have been published

- **Why don’t we use a standard identification scheme secure under adaptive corruption?**
  - The standard notion of secure computation considers a simulator with access to the adversary. Unfortunately the blinder defined for privacy in [Va07,PV08] instead is a very limited player

- **Summing up:**
  - The requirement of the existence of a blinder in the privacy definitions of [Va07,PV08] plays a critical role in all these negative results
  - This new special definition does not seem to be well motivated
Outline

1. Requirement Analysis
2. Existing Solutions, Focus on [Va07]
3. Revisiting RFID Security & Privacy
4. Conclusion
Conclusion

• The state-of-the art for RFID Security & Privacy is nowadays out of our control
  – Too many incomparable (and often ad-hoc) claims
  – Too much resources spent in rediscovering previously achieved results

• A general framework would allow us to make quicker improvements
  – The approach of [Va07, Pv08] is a good step in this direction, but unfortunately the proposed framework includes too many unjustified limitations
Thank You!

Ivan Visconti
University of Salerno (ITALY) and UCLA (USA)

Relevant co-authors:
Frederik Armknecht (Univ. Bochum, Germany)
Paolo D’Arco (Univ. Salerno, ITALY)
Ahmad-Reza Sadeghi (Univ. Bochum, Germany)
Alessandra Scafuro (Univ. Salerno, ITALY)
Christian Wachsmann (Univ. Bochum, Germany)
Backup Slides
Existing Anonymizer-Enabled Solutions

• Do not fulfill security & privacy requirements
  – Impersonation and denial-of-service attacks possible [GJJS04, SRS04, ACM05]
  – Tracing attacks possible [GJJS04, SRS04]

• Are not covered by existing RFID security and privacy models
  – Security and privacy proofs usually in ad-hoc models not reflecting real-world adversaries against RFID [GJJS04, SRS04, ACM05]
Designing Protocols without RO

• Tag’s secret must be frequently updated to prevent tracking after corruption

• Tag uses different keys, which of the anonymizer knows the “difference”

• One needs key correlation resistant cryptographic primitive (MAC, encryption, etc.) to replace the RO