

RFID Authentication Protocol for Low-cost Tags

Boyeon Song and Chris J Mitchell

1 April 2008

Introduction

RFID

Requirements for such RFID Protocols

Privacy, Security and Performance

Related Work

A Novel Authentication Protocol

Design Principle

Preliminaries

Protocol Description

Performance

Conclusion

References

RFID

Automatic Identification and Data Capture (AIDC)

Methods for automatically identifying objects, collecting data about them, and entering that data directly into computer systems

RFID

Automatic Identification and Data Capture (AIDC)

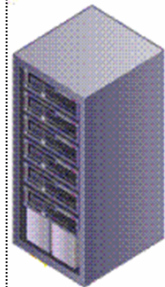
Methods for automatically identifying objects, collecting data about them, and entering that data directly into computer systems

Radio Frequency Identification (RFID)

A wireless AIDC technology that uses radio signals to identify a product, animal or person

RFID System

Server



Reader



Tag



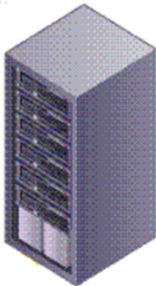
RFID System

Three main components

- ▶ **RFID tags:** an identification device attached to an item, which uses radio frequency (RF) to communicate with RFID readers
- ▶ **RFID readers:** a device that can recognise the presence of RFID tags and read the information supplied by them
- ▶ **A back-end server:** has a database of detailed information regarding a tag (or the item attached to the tag)

Typical RFID Protocol

Server



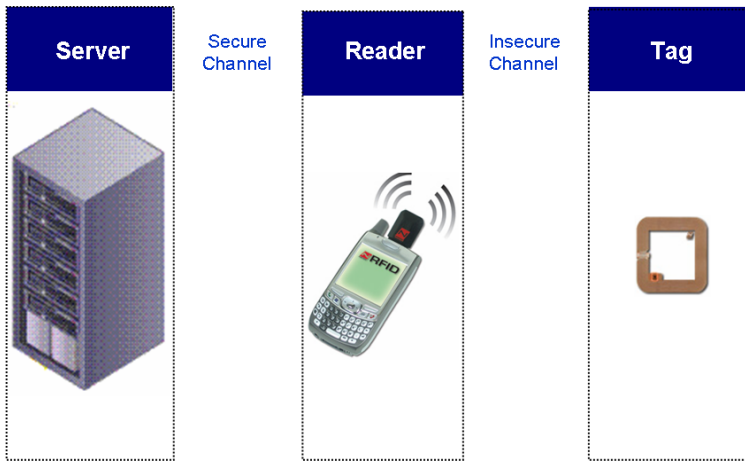
Reader



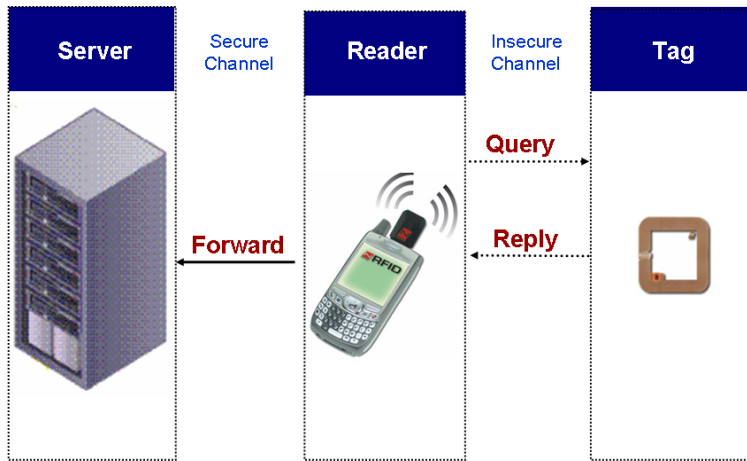
Tag



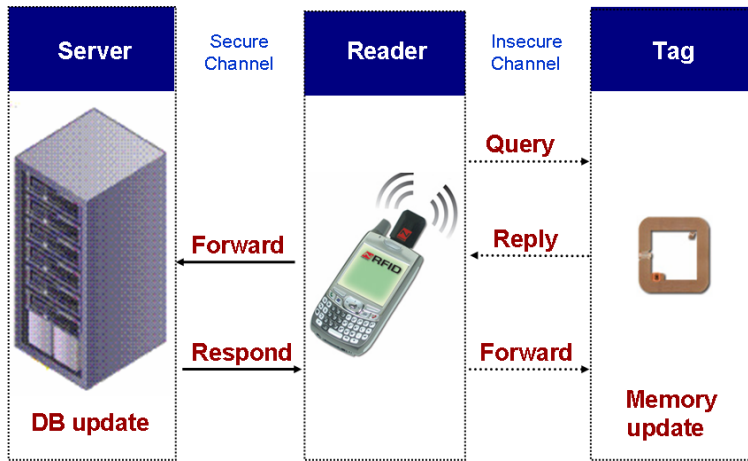
Typical RFID Protocol



Typical RFID Protocol



Typical RFID Protocol



Assumptions in RFID Protocols

Tag

- ▶ has a rewritable, non-temper resistant, and constrained memory capacity.
- ▶ has limited processing power.

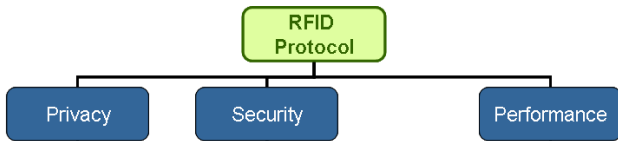
Server

- ▶ has a secure database containing information for tags that it manages.
- ▶ has significantly greater computational ability than a tag.

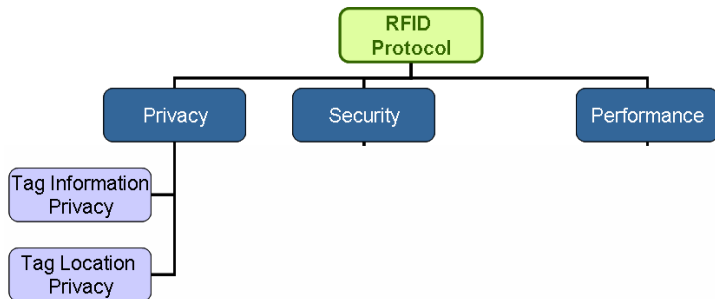
Channels

- ▶ The channel between the server and the reader is secure.
- ▶ The reader and the tags communicate over an insecure channel.

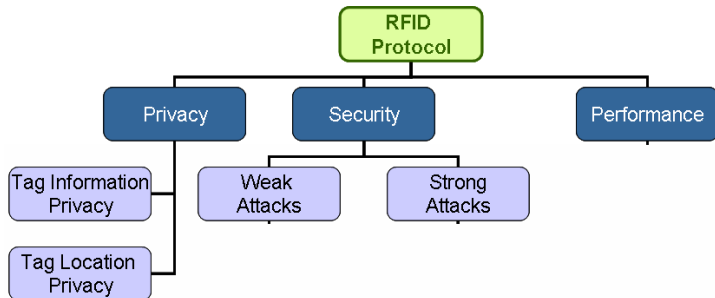
Requirements for RFID Protocols



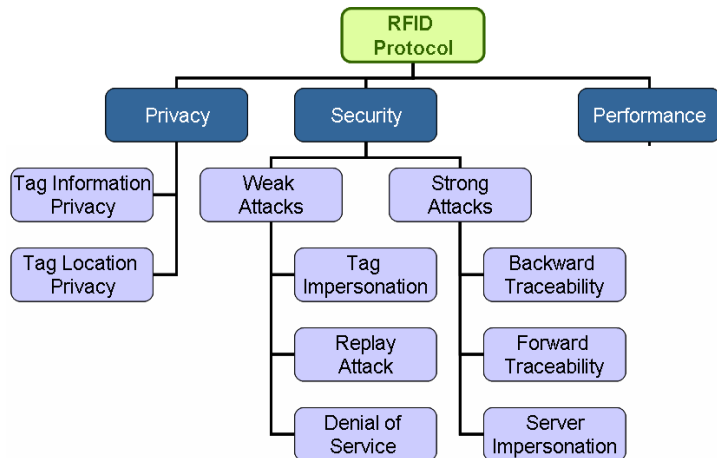
Requirements for RFID Protocols



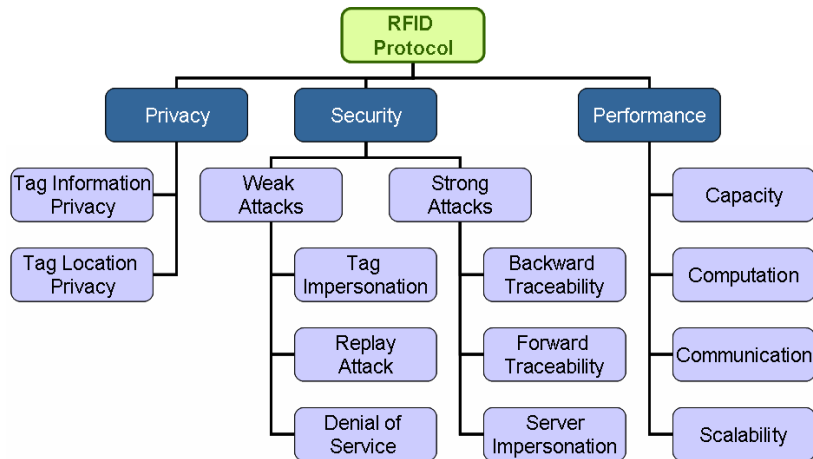
Requirements for RFID Protocols



Requirements for RFID Protocols



Requirements for RFID Protocols



Prior RFID protocols

- ▶ HM scheme [Henrici and Müller, 2004]
- ▶ MW scheme [Molnar and Wagner, 2004]
- ▶ D scheme [Dimitriou, 2005]
- ▶ KN scheme [Karthikeyan and Nesterenko, 2005]
- ▶ DPLK scheme [Duc et al., 2006]
- ▶ CC scheme [Chien and Chen, 2007]
- ▶ LK scheme [Lim and Kwon, 2006]

Privacy and Security Properties of Prior Schemes

<i>Property</i>	HM	MW	D	KN	DPLK	CC	LK
Information Privacy	○	○	○	○	○	○	○
Location Privacy	—	○	—	—	○	○	○
Tag Impersonation	—	○	○	○	—	○	○
Replay attack	—	○	○	—	—	○	○
DoS attack	○	○	—	—	—	○	○
Backward Traceability	—	—	○	—	—	—	○
Forward Traceability	—	—	—	—	—	—	△
Server Impersonation	—	—	—	—	—	—	△

- : provided
- △ : provided under an assumption
- : not provided

Storage of Prior Schemes

<i>Storage</i>	DPLK	CC	LK
Server	$(l + l_1 + l_2) \cdot N$	$(l + 2l_1 + 2l_2) \cdot N$	$2(l + 3l_3 + ml_4) \cdot N$
Tag	$l + l_1 + l_2$	$l + l_1 + l_2$	$l + l_3 + c$

- N : The number of tags
- l : The bit-length of a tag identifier (e.g., $l = 64, 96$ or 128)
- l_1 : The bit-length of a PIN in the DPLK and CC protocols (e.g., $l_1 = 32$)
- l_2 : The bit-length of a session key in the DPLK and CC protocols (e.g., $l_2 = 16$)
- l_3 : The bit-length of a server validator in the LK protocol (e.g., $l_3 = 32$)
- l_4 : The bit-length of a tag secret transmitted in the LK protocol (e.g., $l_4 = 32$)
- m : The maximum number of authentication failures in the LK protocol (e.g., $m = 64$)
- c : The size of a counter in the LK protocol (e.g., $c = 100$)

Computation of Prior Schemes

<i>Computation</i>		DPLK	CC	LK
Server	On receiving the 2nd flow	$(k + 1)F$	kF	$(k_1 + 1)F$
	On sending the 3rd flow	$1F$	$1F$	$1F$
	On updating or refreshing	$1F$	$2F$	$(k_1 + k_2 + m)F$
	Total	$(k + 3)F$	$(k + 3)F$	$(2k_1 + k_2 + m + 2)F$
Tag	On sending the 2nd flow	$2F$	$1F$	$1F$
	On receiving the 3rd flow	$1F$	$1F$	$2F$
	On updating or refreshing	$1F$	$2F$	$1F$
	Total	$4F$	$4F$	$4F$

- F : A computationally complex function
 (such as a Cyclic Redundancy Code (CRC) or Pseudo-Random Function (PRF))
- N : The number of tags
- m : The maximum number of authentication failures in the LK protocol (e.g., $m = 64$)
- n : The length of the backward key chain in the LK protocol (e.g., $n = 2^{20}$)
- k : An integer satisfying $1 \leq k \leq 2N$
- k_1 : An integer satisfying $0 \leq k_1 \leq m - 1$
- k_2 : An integer satisfying $0 \leq k_2 \leq n - 2$

New RFID Protocol

Goals

- ▶ To meet the identified privacy and security requirements
- ▶ To maximise performance efficiency.

Main features

- ▶ A challenge-response approach (replay attack)
- ▶ The server database stores both the most recent and the current data for each tag. (denial-of-service attack)
- ▶ The server and tag update their shared secrets probabilistically using a hash function h and exchanged random numbers r_1 and r_2 . (backward and forward traceability)

Main features

- ▶ A challenge-response approach (replay attack)
- ▶ The server database stores both the most recent and the current data for each tag. (denial-of-service attack)
- ▶ The server and tag update their shared secrets probabilistically using a hash function h and exchanged random numbers r_1 and r_2 . (backward and forward traceability)
- ▶ A random number r_2 generated by a tag serves as a **temporary secret** for the tag.

Main features

- ▶ A challenge-response approach (replay attack)
- ▶ The server database stores both the most recent and the current data for each tag. (denial-of-service attack)
- ▶ The server and tag update their shared secrets probabilistically using a hash function h and exchanged random numbers r_1 and r_2 . (backward and forward traceability)
- ▶ A random number r_2 generated by a tag serves as a **temporary secret** for the tag.
- ▶ A tag only needs to store an **identifier t_i** .

Main features

- ▶ A challenge-response approach (replay attack)
- ▶ The server database stores both the most recent and the current data for each tag. (denial-of-service attack)
- ▶ The server and tag update their shared secrets probabilistically using a hash function h and exchanged random numbers r_1 and r_2 . (backward and forward traceability)
- ▶ A random number r_2 generated by a tag serves as a **temporary secret** for the tag.
- ▶ A tag only needs to store an **identifier** t_i .
- ▶ A server database stores an **identifier** t_i and a **bit-string** u_i used for server validation for each tag, where $t_i = h(u_i)$. (server impersonation attack)

Main features

- ▶ A challenge-response approach (replay attack)
- ▶ The server database stores both the most recent and the current data for each tag. (denial-of-service attack)
- ▶ The server and tag update their shared secrets probabilistically using a hash function h and exchanged random numbers r_1 and r_2 . (backward and forward traceability)
- ▶ A random number r_2 generated by a tag serves as a temporary secret for the tag.
- ▶ A tag only needs to store an identifier t_i .
- ▶ A server database stores an identifier t_i and a bit-string u_i used for server validation for each tag, where $t_i = h(u_i)$. (server impersonation attack)

Notation

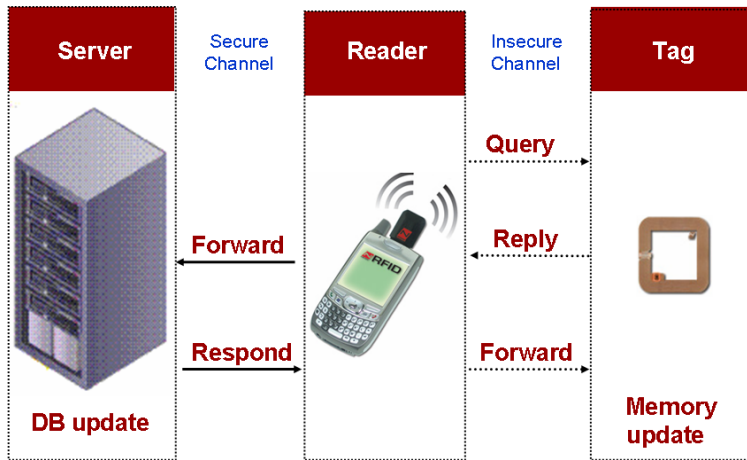
- h A hash function, $h : \{0, 1\}^l \rightarrow \{0, 1\}^l$
- f_k A keyed hash function, $f_k : \{0, 1\}^l \times \{0, 1\}^l \rightarrow \{0, 1\}^l$
(a MAC algorithm)
- \oplus XOR operator
- \parallel Concatenation operator
- \leftarrow Substitution operator
- $x \ggg k$ Right circular shift operator, which rotates all bits of x to the right by k bits, as if the left and right ends of x were joined.
- $x \lll k$ Left circular shift operator, which rotates all bits of x to the left by k bits, as if the left and right ends of x were joined.

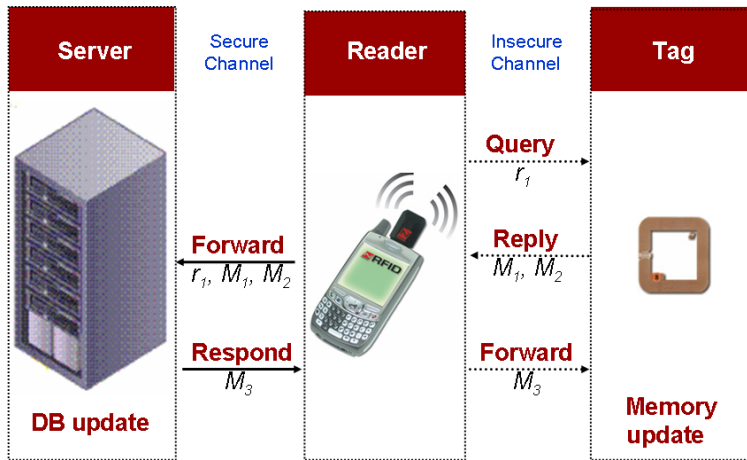
Assumption

- ▶ There are implementations of h and f_k which are suitable for a low cost tag, and sufficiently secure, and resistant to collision.

Assumption

- ▶ There are implementations of h and f_k which are suitable for a low cost tag, and sufficiently secure, and resistant to collision.
 - ▶ [Weis, 2003]: non-linear feedback shift registers for a different candidate paradigm for low cost RFID hash functions
 - ▶ [Yüksel, 2004]: several universal hash functions designed specifically for efficient hardware implementations and ultra-low power devices
 - ▶ [Pramstaller et al., 2006]: a compact hardware implementation of a hash function Whirlpool





Privacy and security

<i>Property</i>	HM	MW	D	KN	DPLK	CC	LK	Sec. 4
Information Privacy	○	○	○	○	○	○	○	○
Location Privacy	—	○	—	—	○	○	○	○
Tag Impersonation	—	○	○	○	—	○	○	○
Replay attack	—	○	○	—	—	○	○	○
DoS attack	○	○	—	—	—	○	○	○
Backward Traceability	—	—	○	—	—	—	○	○
Forward Traceability	—	—	—	—	—	—	△	△
Server Impersonation	—	—	—	—	—	—	△	△

- : provided
 △ : provided under an assumption
 — : not provided

Storage

Storage	DPLK	CC	LK	Sec. 4
Server	$(l + h_1 + h_2) \cdot N$	$(l + 2h_1 + 2h_2) \cdot N$	$2(l + 3h_3 + mh_4) \cdot N$	$4l \cdot N$
Tag	$l + h_1 + h_2$	$l + h_1 + h_2$	$l + h_3 + c$	l

- N : The number of tags
- l : The bit-length of a tag identifier (e.g., $l = 64, 96$ or 128)
- h_1 : The bit-length of a PIN in the DPLK and CC protocols (e.g., $h_1 = 32$)
- h_2 : The bit-length of a session key in the DPLK and CC protocols (e.g., $h_2 = 16$)
- h_3 : The bit-length of a server validator in the LK protocol (e.g., $h_3 = 32$)
- h_4 : The bit-length of a tag secret transmitted in the LK protocol (e.g., $h_4 = 32$)
- m : The maximum number of authentication failures in the LK protocol (e.g., $m = 64$)
- c : The size of a counter in the LK protocol (e.g., $c = 100$)

Computation

<i>Computation</i>		DPLK	CC	LK	Sec. 4
Server	On receiving the 2nd flow	$(k + 1)F$	kF	$(k_1 + 1)F$	kF
	On sending the 3rd flow	$1F$	$1F$	$1F$	—
	On updating or refreshing	$1F$	$2F$	$(k_1 + k_2 + m)F$	$1F$
	Total	$(k + 3)F$	$(k + 3)F$	$(2k_1 + k_2 + m + 2)F$	$(k + 1)F$
Tag	On sending the 2nd flow	$2F$	$1F$	$1F$	$1F$
	On receiving the 3rd flow	$1F$	$1F$	$2F$	$1F$
	On updating or refreshing	$1F$	$2F$	$1F$	$1F$
	Total	$4F$	$4F$	$4F$	$3F$

F : A computationally complex function (such as a CRC, PRF or hash function)

N : The number of tags

m : The maximum number of authentication failures in the LK protocol (e.g., $m = 64$)

n : The length of the backward key chain in the LK protocol (e.g., $n = 2^{20}$)

k : An integer satisfying $1 \leq k \leq 2N$

k_1 : An integer satisfying $0 \leq k_1 \leq m - 1$

k_2 : An integer satisfying $0 \leq k_2 \leq n - 2$

Summary

The proposed protocol

- ▶ the most identified privacy and security properties.
- ▶ the least tag-side memory among the four schemes.
- ▶ fewer complex function invocations than the other three protocols, both for the server and the tag.

- Outline
- Introduction
- Requirements for such RFID Protocols
- Related Work
- A Novel Authentication Protocol
- Conclusion**
- References

Thank you

Any Questions and Comments?





H. Chien and C. Chen.

Mutual authentication protocol for RFID conforming to EPC class 1 generation 2 standards.
Computer Standards & Interfaces, 29(2):254–259, February 2007.



T. Dimitriou.

A lightweight RFID protocol to protect against traceability and cloning attacks.
In *Conference on Security and Privacy for Emerging Areas in Communication Networks — SecureComm*, pages 59–66, Athens, Greece, September 2005. IEEE.



D. N. Duc, J. Park, H. Lee, and K. Kim.

Enhancing security of EPCglobal gen-2 RFID tag against traceability and cloning.
In *Symposium on Cryptography and Information Security — SCIS 2006*, Hiroshima, Japan, January 2006.
The Institute of Electronics, Information and Communication Engineers.



A. Henrici and P. Müller.

Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers.
In R. Sandhu and R. Thomas, editors, *International Workshop on Pervasive Computing and Communication Security — PerSec 2004*, pages 149–153, Orlando, Florida, USA, March 2004. IEEE Computer Society.



S. Karthikeyan and N. Nesterenko.

RFID security without extensive cryptography.
In *Workshop on Security of Ad Hoc and Sensor Networks — SASN '05*, pages 63–67, Alexandria, Virginia, USA, November 2005. ACM Press.



C. Lim and T. Kwon.

Strong and robust RFID authentication enabling perfect ownership transfer.

In P. Ning, S. Qing, and N. Li, editors, *Conference on Information and Communications Security — ICICS '06*, volume 4307 of *Lecture Notes in Computer Science*, pages 1–20, Raleigh, North Carolina, USA, December 2006. Springer-Verlag.



D. Molnar and D. Wagner.

Privacy and security in library RFID: Issues, practices, and architectures.

In B. Pfitzmann and P. Liu, editors, *Conference on Computer and Communications Security — ACM CCS*, pages 210–219, Washington, DC, USA, October 2004. ACM Press.



N. Pramstaller, C. Rechberger, and V. Rijmen.

A compact FPGA implementation of the hash function whirlpool.

In *ACM/SIGDA 14th international symposium on Field Programmable Gate Arrays — FPGA'06*, ACM Press, pages 159–166, New York, 2006.



Stephen Weis.

Security and privacy in radio-frequency identification devices.

Master's thesis, Massachusetts Institute of Technology (MIT), Massachusetts, USA, May 2003.



K. Yüksel.

Universal hashing for ultra-low-power cryptographic hardware applications.

Master's thesis, Dept. of Electrical Engineering, Worcester Polytechnic Institute, Worcester, MA, USA, 2004.