ICU/KOREA

AUTO–ID LABS

An Authentication Framework for Integrating RFID Systems

Sungbae Ji, Hyunrok Lee, Sungjune Yoon, and Kwangjo Kim
Auto-ID Lab, ICU, Korea

Auto-ID Labs White Paper WP-SWET-025

Software & Network

Summary

It is difficult to combine heterogeneous RFID systems which have different security requirements and authentication protocols. In this paper, we propose a new RFID authentication framework based on EAP. Due to the properties of RFID, EAP cannot be directly applied to the RFID systems. Using our framework, we can integrate proper authentication protocols for many different RFID applications into a single RFID authentication system.

Keywords

RFID, Authentication Framework, Authentication Schemes, EAP

1. Introduction

Radio Frequency IDentification (RFID) technology has been widely used for tracking and inventorying systems in supply chains, check-out/in systems in libraries, electronic payment systems, and electronic passports. However there are sensitive issues that should be dealt with in order to adopt secure RFID systems.

Feldhofer *et al.* [FEL04] described three security issues in RFID systems: privacy compromise, forgery of tags, and unauthorized access to tags. In addition to these issues, the problems of eavesdropping, tag spoofing, tag cloning, and replay attacks were pointed out in other literatures [WEI04, SAR03, VAJ03]. These security problems can be solved through proper authentication mechanisms. Recently, many authentication protocols are proposed such as hash-based authentication schemes [WEI04, OHK04, HEN04], PIN-based scehme [JUE05], and block-cipher-based scheme [FEL04], but there is no universal authentication protocol that can be applied to many types of systems. In fact, it is inevitable to have various authentication protocols because each application has its own security requirements. Moreover, each application requires different types of RFID tags because of their different characteristics. Therefore, any authentication protocols other than the embedded authentication protocols cannot be used, and heterogeneous RFID systems cannot be integrated into a single RFID authentication system.

A universal authentication framework, EAP (Extensible Authentication Protocol) [ABO05] seems to be a solution for this problem because of its extensibility and similar architectural model. However, there are a few limitations of applying EAP directly into RFID systems such as bulk-reading capabilities, traceability and privacy issues, and extremely low resources. These features make EAP inapplicable to RFID systems.

1.1. Our Contributions

In this paper, we will propose a universal authentication framework suitable for RFID systems based on EAP. When applying our authentication framework to RFID systems, a specific authentication method including a vendor-specific method that a tag supports can be chosen by a negotiation process. As a result, many types of tags and authentication methods can be combined into a single authentication system as long as a middleware supports the same authentication methods, and an owner of a tag can use an authentication method they want to use.

2. Background and Related Work

Each RFID application must meet with its security requirements according to its own goal, and various types of RFID tags used for the applications also have its own security features (*e.g.* encryption algorithms, hash functions, and PINs) as well as its technical features (*e.g.* radio frequencies, computational resources, and data I/O rates). For the authentication, each system can utilize a suitable authentication protocol based on the capacity of tags and the security goal. Following subsections, we discuss why we require an authentication framework for RFID systems with various applications, tags, and authentication schemes.

2.1. RFID Application Requirements

Phillips *et al.* [PHI05] categorized RFID systems into three applications. Logistical applications are the RFID applications used for inventorying and tracking products in supply chains. Because the products are required to be rather tracked and physically secured, strong authentication mechanisms in RFID systems are not necessarily required. Low-latency, high potential read rates and bulk-reading capabilities are much more important.

On the other hand, consumer applications need to protect consumers' privacy. When an unauthorized person or a device tries to access a smart card, an electronic passport, or a consumer's belongings, a certain level of authentication is required to protect privacy.

Vertical applications is somewhere between logistical applications and consumer applications and requires specific security features according to their goals. For example, RFID-enabled banknotes [JUE03] require limited access control to protect consumers' privacy and tracking capabilities to monitor illegal transactions as well.

2.2. RFID Tag Classifications by Characteristics

Various RFID applications need various types of RFID tags. Based on the band of radio frequency, RFID tags are categorized into low-frequency (LF), high-frequency (HF), and ultra-high-frequency (UHF) tags. Generally, passive tags are widely used in logistical applications because they are cheap and small. Unlike passive tags, active tags require a power source and can have more computational resources.

EPC Gen 2 Class 1 UHF tags [EPC05] are passive identity tags designed to be used for supply chain and logistical applications, and thus they are simple and cheap. They have a PRNG (pseudo random number generator), CRC (cyclic redundancy check) error detection, a kill function (32-bit kill PIN), and password-protected access control (32-bit access PIN).

Smart cards and RFID-enabled passports are typical examples of RFID tags using ISO/IEC 14443 [ISO14443] and 15693 [ISO15693] air interface. Because security features are not defined in the ISO/IEC standards, each vendor implements its own proprietary authentication and access protocol using cryptographic primitives such as DES, 3DES, AES, RSA, SHA-1, *etc*.

2.3. RFID Authentication Schemes

Due to their simplicity, many hash-based authentication schemes have been proposed for low-cost tags. Hash-lock [WEI04] reveals keys in plaintexts and cannot protect location privacy because a tag is authenticated using a fixed metaID. In randomized hash lock [WEI04], a tag responds with randomized ID value but it is still traceable and not scalable. Hash chain [OHK04] burdens the back-end database with a series of hash computations, and tag should have two different hash functions. Hash-based ID variation scheme [HEN04] is untraceable and secure against replay attack but spoofing attack is possible.

BasicTagAuth+ protocol [JUE05] is designed to be used for EPC tags. This protocol employs a kill PIN of EPC Gen 2 Class 1 UHF tags. Because it assumes untrustworthy readers, authentication is performed between tags and a verifier while limiting intervention from readers.

Feldhofer *et al.* implemented encryption-only AES algorithm as an 8-bit architecture in hardware. They achieved low power consumption and low die-size circuit enough to satisfy the restriction of passive RFID tags. This result shows that AES encryption algorithm can be used as a cryptographic primitive for RFID authentication protocols.

3. RFID Authentication Framework Requirements

3.1. System Architecture and Requirements

An RFID system architecture consists of RFID tags, RFID readers, and a middleware (or a backend server). Usually, RFID authentication is done between a middleware and tags. An RFID reader relays messages between tags and a middleware and does not need to concern credentials used in authentication methods and policy decision. The followings are our basic assumption and requirements for RFID authentication framework in this paper.

3.1.1. Assumptions

- ***Pre-established secure channel.*** A middleware authenticates RFID readers beforehand and establishes a secure channel with each reader via pre-shared key or current well-known security mechanism such as SSL/TLS. If a wired reader is attached to a host computer, the host computer is authenticated instead of the reader. In case of wireless/mobile readers (e.g. PDA), it can be authenticated after connected to WLAN securely using EAP method and WPA2.
- ***Reliable transport of low layer.*** The underlying air interface protocol covers unreliable transport more than an expected level so that it guarantees a high probability of successful detection. Class 1 Generation 2 UHF Air Interface Protocol Standard provides higher reliability in radio communications and collision avoidance than previous RFID air interface.

3.1.2. Requirements

- ***Multiple instances in the tags.*** Tags can store the instances of authentication protocols and distinguish them. In order to support multiple readers, each of multiple states is maintained in the memory of tags until the authentication processes is finished. Because the lack of memory, instances are stored in the memory circularly (*e.g.* round robin queuing).
- ***Timer-driven authentication in the middleware.*** A middleware maintains each instance from each valid response on a timer-driven basis because multiple tags can respond to a request message. The terminated instances are discarded.
- ***Pass-through behavior.*** RFID readers act as “pass-through agents" without authentication method layer functionalities so that they are compatible with multiple authentication methods.
- ***Mutual authentication support.*** Authentication framework should support mutual authentication as well in order to prevent the accesses from unauthorized readers.

- ***Authentication method negotiation.*** To support multiple authentication methods, negotiaion should be provided in the framework.

3.2. EAP and Its Limitation in RFID Systems

EAP is usually used in wireless LANs but not limited to wireless LANs. It is, in fact, a universal authentication framework which can be used any network environment [ABO05]. It supports multiple authentication methods. There are more than 40 authentication methods defined in RFCs. Each method takes different approaches to authenticate EAP peer only or both EAP peer and authenticator in mutually. EAP-MD5, LEAP, EAP-TLS, EAP-TTLS, PEAP are widely used in WLANs and they have different performances and security features using various cryptographic primitives. Through the EAP negotiation, a peer and an authenticator can choose what they want among various authentication methods.

In addition to this property, the similarity between RFID systems and WLANs seems to fit EAP into a RFID authentication framework. They are using RF signal to communicate between RFID tags and readers; and between access points (APs) and supplicants. Moreover, readers and APs are the interfaces of the RFID applications and Internet services respectively. Mutual authentications in WLANs are performed between an authentication server (e.g. RADIUS server) and supplicants not between APs and supplicants because of rogue AP problems. In the same manner, authentications are done between a middleware and tags, and readers act as "pass-through agents." Therefore, RFID readers do not need to concern credentials used in authentication methods and policy decision.

From this point of view, Dantu *et al.* [DAN07] examined and evaluated existing EAP methods for RFID. However, they overlooked the main difference between WLANs and RFID systems. In WLAN, EAP methods are used in the IEEE 802.1X phase not only for authentication but also generation of key materials. Using a key derived from IEEE 802.1i 4-way handshake, data can be encrypted. Unlike continuous data exchange in WLANs, passive RFID tags respond only when a reader requests in the authentication process. That means RFID systems do not need a key or key material for further continuous data exchange. Moreover, there are some features they did not consider seriously when applying EAP to RFID systems. In the following section, we point out limitations of applying EAP into RFID systems.

3.2.1. Limitation

- ***Bulk-reading capability.*** In EAP, an authentication is performed with an EAP peer one-to-one, and each entity proceeds to the next authentication step synchronously, so-called

"lock-step protocol." Unlike EAP, some RFID applications need bulk-reading capability. When a request is fired, a middleware should expect 0 or more responses from tags and keep authentication state per tag with timer.

- ***Traceability and privacy issues.*** An **Identity** (**Type 1**) exchange sent in cleartext is optional in EAP, but it is "recommended to be used primarily." On the other hand, in RFID systems, it is better to design a method-specific identity exchange not to expose the identity of tag, especially in consumer application. Because obtaining the identity of a tag is the only goal in RFID application, **Identity** type should be used in the authentication method carefully.
- ***Extremely low resources.*** Passive RFID tags have extremely low power because they induce current from incoming RF signals. Because outgoing RF communication consumes the large portion of available power, the shorter packet length is more suitable for RFID applications. EAP is designed to provide 1020 octets of MTU (Maximum Transmission Unit) and fragmentation. However, the MTU size for RFID systems should be relatively small, and RFID cannot consider fragmentation.

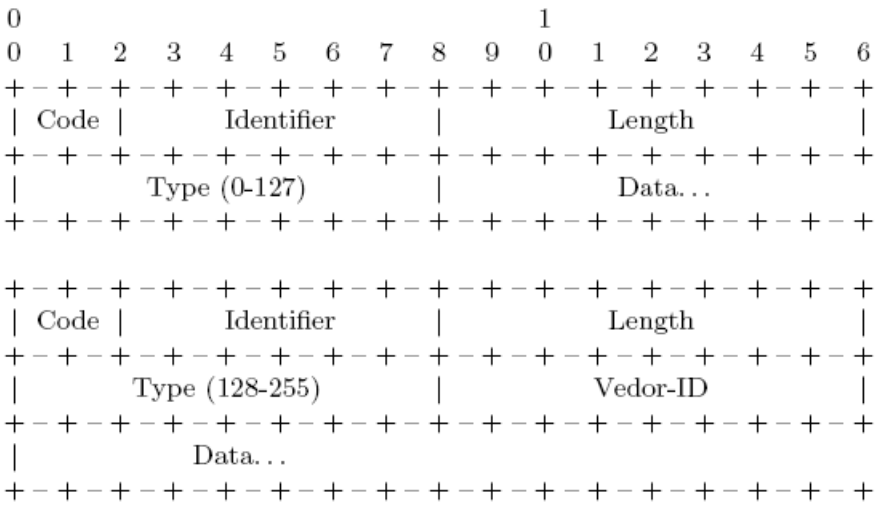

Fig. 1: Message Frame Format

4. RFID Authentication Framework based on EAP

The authentication process always starts from RFID reader's **Request** so as to be applied to passive RFID tags. A passive tag receives this **Request**, induces current needed to operate itself, and sends **Response** to the reader. Readers have a PRNG and use it to make a random number required in the initial **Request** message of the default authentication method. Except for this initial message, readers act as pass-through agents.

As shown in Figure 1, the message frame format of proposed authentication framework is designed based on EAP. However, some fields are changed and all over fields are shortened after considering the feature of RFID technology and the applicability of EAP to RFID applications.

The 2-bit **Code** field is assigned as shown in Table 1. **Success** and **Failure** in EPC are combined into **Notification** and divided in the **Type** field. When a tag receives notification, it can release the authentication instance in its memory. The **Command** can be used by a middleware in order to access tags directly with or without authenticating tags.

Code	
0	Request
1	Response
2	Notification
3	Command

Table 1: Code Field

The **Identifier** field is the same as the EAP Identifier field except the length. Because this field has no cryptographic meaning, 6-bit long Identifier is enough to match Response packets with Request packets.

The **Length** field is the packet length in octets. This means the length of packet cannot exceed 255 bytes, which is long enough to be used for RFID applications.

An 8-bit **Type** field value (Table 2) is selected among 256 values corresponding to the **Code** field. **Identity** is used for null authentication, which requests the Identity of tags but not authenticates them or which can be used in another authentication method. **Nak** is a type for authentication negotiation when a received **Type** of **Request** is not supported by a tag. **Success** or **Failure** notify whether a performed authentication is done successfully or not. Other than these types, 124 defined standard authentication types and 128 vendor-specific types can be supported. Whenever a vendor-specific type is used, an 8-bit **Vendor-ID** is followed to specify which vendor's tag is used. This **Vendor-ID** should be globally unique and can be supported upto 256 vendors.

Type	
0	Identity (Null Authentication)
1	Nak (Response Only)
2	Success (Notification Only)
3	Failure (Notification Only)
4 – 127	Standard Types
128 – 255	Vendor-specific Types

Table 2: Authentication Type

Type - Code	Data
Identity - Request	0 octet
Identity - Response	ID (e.g. 12 octets for EPC)
Nak - Response	*s* + 2*v* octets (*s* = # of standard types, *v* = # of vendor-specific types)
Success - Notification	optional commands to access a tag's memory (0 or more octets)
Failure - Notification	0 octet
Type(4-255) - Request	0 or more octets depending on authentication methods
Type(4-255) - Response	0 or more octets depending on authentication methods

Table 3: Data Field determined by Type-Code

The **Data** field is 0 or more octets depending on the **Code** and **Type** field. The **Data** field corresponding to each **Type-Code** case is shown in Table 3. The **Data** field of **Nak-Response** message is filled with authentication types that a tag wants to use. Vendor-specific types always accompany a **Vendor-ID**. When there is no desired authentication type, **Nak-Respone** will have 0 octet for the **Data** field. **Success-Notification** can include optional commands to access a tag's memory (*e.g.* to update a shared secret or a key, or to obtain additional information about the product).

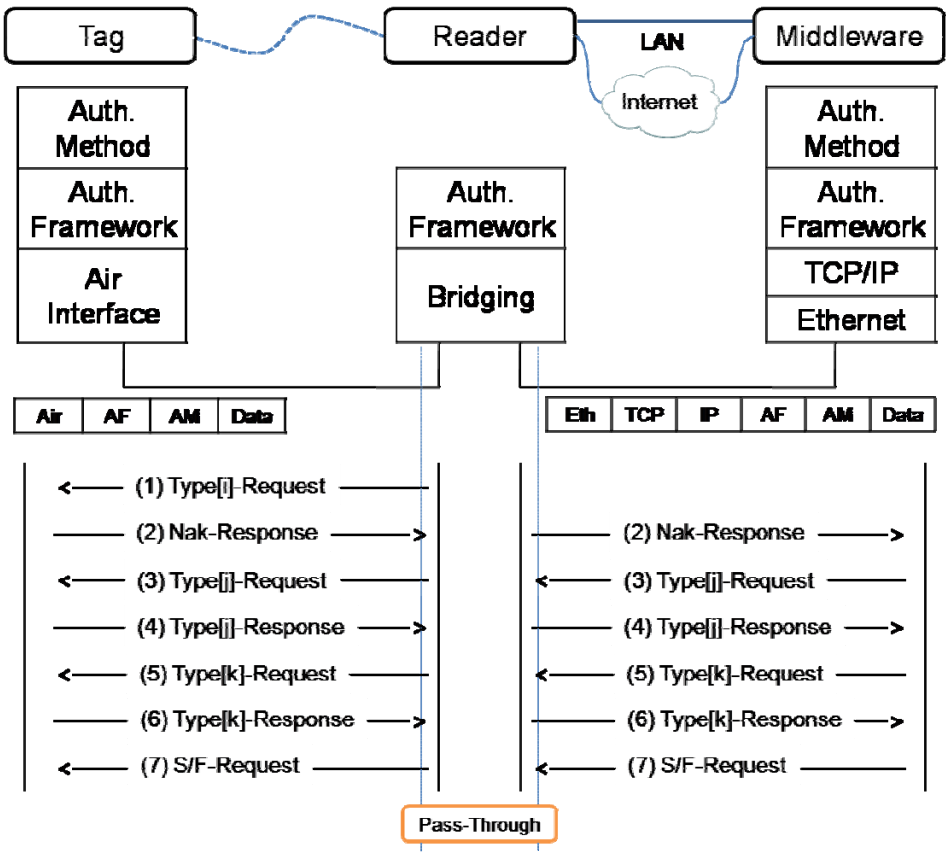

Fig. 2: RFID Authentication Framework and Authentication Flow

Figure 2 shows an RFID system. A tag and a middleware have three layers: authentication method (AM) layer, authentication framework (AF) layer, and underlying network layers. On the other hand, a reader has only two layers and bridges underlying network layers. Each layer encapsulates upper-layer data by adding header or footer. In our framework, AF header is (**Code**, **Identifier**, **Length**), and AM header is (**Type**) or (**Type**, **Vendor-ID**). A typical authentication flow of our proposed framework is as follows.

(1) Reader initiates default authentication methods, **Type**[*i*] where $i \in \{0, 4{:}255\}$. Identifier value is incremented from the previous value.($a{:}b = \{a, a+1, \ldots, b\}$ where $a < b$)

(2) When a tag does not support the default **Type**, it can send **Nak** = **Type**[1]. The Data field of **Nak** will be filled with authentication type set, TypeSet where **Type**[1] $\notin$ TypeSet. (2) and (3) can be omitted if the default authentication method is satisfied with a tag. In this case, go to (4) and assume $j = i$.

(3) Middleware selects an element **Type**[*j*] $\in$ TypeSet and re-initiate authentication process. **Identifier** value is not incremented.

(4) If a tag receives (3) with the same **Identifier** after sending (2), it responds to (3) with **Type**[*j*]-**Response**. Otherwise, (3) with the same **Identifier** is discarded. Depending on the **Type**[*j*], (3)-(4) is more than one round trip.

(5) (5)-(6) is not necessarily required, but another authentication type can be jointly used in this framework. Usually, in this case, **Type**[0] = **Identity** is jointly used. Middleware sends **Type**[*k*]-**Request** where $k \in \{0, 4{:}255\}$, $k \neq i$, and $k \neq j$.

(6) Tag sends corresponding **Type**[*k*]-**Response** to (5). Depending on the **Type** [*k*], (5)-(6) is more than one round trip.

(7) If authentication fails in the middle of authentication, middleware sends **Failure-Notification** immediately. If authentication ends successfully, middleware sends **Success-Notification**.

5. Concluding Remarks

In this paper, we propose a universal authentication framework suitable for RFID systems based on EAP. When applying this framework to an RFID system, authentication methods which tags support can be selected through the negotiation, and vendor-specific authentication methods can be also used in the system. Therefore, various types of tags and authentication methods will be integrated into a single authentication system as long as the middleware supports the authentication methods. Moreover, the owner of a tag can make the tag to use a specific authentication method appropriate for their application after ownership is transferred.

For a further research, we will present actual authentication methods using distinct cryptographic primitives such as a hash function, a PIN, and a block cipher, demonstrate them in this authentication framework through an implementation, and verify its wide applicability. We will also research ownership transfer in this authentication framework for solving key pre-distribution problems of the symmetric encryption based authentication.

References:

[FEL04] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer, Strong Authentication for RFID Systems Using the AES Algorithm, CHES 2004, LNCS 3156, pp. 357-370, 2004.

[PHI05] Ted Phillips, Tom Karygiannis, and Rick Kuhn, Security Standards for the RFID Market, Security & Privacy Magazine, IEEE, vol. 3, no. 6, pp. 85-89, Nov-Dec. 2005.

[JUE05] Ari Juels, Strengthening EPC Tags Against Cloning, ACM Workshop on Wireless Security (WiSe), pp.67-76. 2005.

[ABO04] Bernard Aboba, Larry J. Blunk, John R. Vollbrecht, James Carlson, and Henrik Levkowetz, Extensible Authentication Protocol (EAP), RFC 3748, IETF, 2004.

[JUE03] Ari Juels and Ravikanth Pappu, Squealing Euros: Privacy Protection in RFIDEnabled Banknotes, LNCS 2742, pp. 103-121, 2003.

[EPC05] EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz - 960 MHz Version 1.0.9. 2005.

[WEI04] Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, Security in Pervasive Computing, LNCS 2802, pp. 201-212, 2004.

[OHK04] Miyako Ohkubo, Koutarou Suzuki, and Shingo Kinoshita, Hash-Chain Based Forward-Secure Privacy Protection Scheme for Low-Cost RFID, Proceedings of the SCIS 2004, pp.719-724, 2004.

[HEN04] Dirk Henrici and Paul Müller, Hash-Based Enhancement of Location Privacy For Radio-Frequency Identification Devices Using Varying Identifiers, PerSec, 2004.

[TRA05] Ken Traub, Greg Allgair, Henri Barthel, Leo Burstein, John Garrett, Bernie Hogan, Bryan Rodrigues, Sanjay Sarma, Johannes Schmidt, Chuck Schramek, Roger Stewart, and KK Suen, The EPCglobal Architecture Framework, EPCglobal, 2005.

[SAR03] Sanjay Sarma, Stephen Weis, and Daniel Engels, Radio-Frequency Identification: Security Risks and Challenges, Cryptobytes, RSA Laboratories, 2003.

[VAJ03] István Vajda and Levente Buttyán, Lightweight authentication protocols for lowcost RFID tags, Workshop on Security in Ubiquitous Computing, 2003.

[ISO14443] ISO/IEC 14443-2, Identification cards - Contactless integrated circuit(s) cards - Proximity cards (PICCS) - Part 2: Radio frequency power and signal interface, 2001.

[ISO15693] ISO/IEC 15693-2, Identification cards - Contactless integrated circuit(s) cards - Vicinity cards (VICCs) - Part 2: Air interface and initialisation, 2000.

[DAN07] Ram Dantu, Gabriel Clothier, and Anuj Atri, EAP methods for wireless networks, Computer Standards & Interfaces 29, pp. 289-301, 2007