A Secure RFID Ticket System for Public Transport

Kun Peng and Feng Bao

Institute for Infocomm Research, Singapore

Abstract. A secure RFID ticket system for public transport is proposed in this paper. It supports security properties including secure authentication, unforgeability, correct billing and privacy and can prevent various attacks. It consists of two protocols, both following three principles necessary for secure RFID ticket system. The first protocol is very efficient and suitable for applications with critical requirement on efficiency. The second protocol does not need any trust assumption and is suitable for applications with critical requirement on security.

1 Introduction

RFID ticket system has been widely applied to public transport. Each user of the public transport system holds a RFID card as his ticket. When the user enters and leaves the public transport system, his card is read by a checking (verifying) machine and his authentication is checked. Only valid users are allowed to enjoy transport service. After each ride, a user's credit is charged. He can top up his credit when necessary. Security of existing RFID ticket systems is weak and needs improving. The existing RFID ticket schemes, including efficient solutions without computation in RFID cards [3,5], solutions authenticating RFID cards through an identity-linked key [9,1,6,7,8], solutions based on dynamic identity [4,10] or more recent solutions like [11], cannot achieve strong enough security in practical applications. Some of them are efficient, but need strong trust on the participants; some of them are too inefficient when handling real-time tasks and thus impractical; some of them ignore important security properties. In one word, they cannot prevent the attacks at a practical cost in a practical environment.

In this paper, two new RFID ticket protocols are proposed. Both of them follow three principles emphasized in this paper: one-time secret for authentication, secure database to store information and simple billing system to simplify soundness and privacy. The first protocol does not need the RFID cards to carry out any computation, so is very efficient. Although it is more secure than many existing RFID ticket schemes, it needs to trust the vender machines, so still needs to improve its security mechanism in applications with critical security requirements. The second protocol removes the trust assumption by letting the RFID cards to carry out some necessary computations. Efficiency improving methods like allocating costly computations to participants with greater computation capability and carrying out costly computations before hand guarantee that its efficiency is high enough for its supposed applications.

S. Foresti and S. Jajodia (Eds.): Data and Applications Security XXIV, LNCS 6166, pp. 350–357, 2010.

2 New Schemes

Two new protocols for RFID ticket in public transport are proposed in this section. Before describing the two protocols, three designing principles are proposed and explained. We believe that they are necessary in secure RFID ticket schemes. Then the two protocols, denoted as Protocol 1 and Protocol 2 respectively, are proposed. Both of them follow the three principles. Protocol 1 is very efficient but depends on a trust assumption. Protocol 2 is less efficient but more secure and its efficiency is still practical.

2.1 Three Principles

There are three designing principles we think necessary in design of secure RFID ticket schemes. They are one-time secret, information stored in a database and simple billing mechanism.

Firstly, our analysis leads to a result: unless interactive asymmetric cipher based cryptographic techniques like zero knowledge proof is employed, a RFID card must use different secret information for each authentication operation in a RFID ticket based public transport system.

- If a RFID card uses the same symmetric cipher based secret information for each authentication, it can be linked to the unique authentication secret and thus can be traced.
- A unique authentication secret for a RFID card is liable to replay attack. The unique authentication secret may be reused.

As we stated before, with practical limitations to the computation capability, asymmetric cipher is too costly for authentication, which must be real-time. So to achieve privacy and prevent replay attack, one-time secret is necessary in authentication.

Secondly, we demonstrate that in a RFID ticket based public transportation system there must be a database.

- Although information in the system can be stored in the RFID cards, as stated before, it is difficult to prevent the card from being tampered with if the card owner colludes. Moreover, if the verifiers need to write some information (e.g. billing information) to the cards, a corrupted verifier may write invalid information to the RFID cards. So information on the IFID cards is not reliable and a database is needed to store some important information.
- As costly asymmetric cipher cannot be employed in authentication and one-time authentication secret must be employed, a database is needed to store some verification information, against which the one-time authentication secrets can be tested. For each valid one-time authentication secret, there is corresponding verification record in the database. A one-time authentication secret is accepted if and only if a corresponding verification record can be found in the database. After a one-time authentication secret is successfully verified, its corresponding verification record is deleted from the database so that it cannot be reused.

– If the credit and billing information of a RFID ticket is stored in the RFID card, it may be tampered with as stated before. So we suggest to store it in the same database together with the verification records. Various security mechanisms including access control, file protection, encryption, audit, encryption and sharing of power among multiple parties can be employed in the database to guarantee integrity (and privacy when necessary) of the credit and billing information.

Receiving an authentication query, the database searches its records for the corresponding verification information. The search must be efficient as it must be real-time. So computation in the search must be strictly limited although the database manager may have greater computation capability than the users. So the number of cryptographic operations should be strictly controlled and a large number of cryptographic operations like in [9] must be avoided.

Thirdly, we find that billing system should employ the so-called simple billing mechanism. In a simple billing mechanism, each trip in the public transport system is charged a same amount of credit no matter how long it is. As discussed before, the credit and billing information should be stored in a database. If charging of a RFID card is measured by the length of the trip, the authority in charge of the database must calculate the chargement according to the starting place and the ending place of the trip and thus can collect the card owner's location information and trace him. So for the sake of privacy of RFID card users and to prevent tracing attack, simple billing mechanism should be employed. Actually, this billing mechanism has been widely employed in public transport systems in many cities in the world.

2.2 Protocol 1

Protocol 1 is a simple and secure RFID based protocol for public transport especially suitable for low-capability RFID cards. In Protocol 1, a RFID card does not need to perform any computation. However, the vender machines must be trusted. The RFID card defined suitable for public transport is used. The three principles, one-time secret, database and simple billing mechanism are employed in Protocol 1. When a user buys a new RFID card or tops up his RFID card at a vender machine, the vender machine receives his payment and calculates how many times the user is allowed to use the public transport service according to the simple billing mechanism. Suppose the user's payment enables t times of service. The vender machine generates $2t$ verification tokens and stores them in the users' RFID card. The vender machine generates a verification record for each token and then stores the $2t$ tokens in the database, each in a random different place. Each trip costs a user two tokens in his card. Each of the two tokens is verified against the corresponding record in the database, which is deleted afterward. Detailed description of the protocol is as follows.

1. A database is set up. The vender machines have the right to insert records to the database. The check machines in the transport stations are verifiers and can query the database for a record. The database can automatically delete

a record after it matches a query. Necessary security measures like access control, file protection, encryption, audit, sharing of power are employed to guarantee that the data in the database is confidential and integrated.

2. A public one-way collision-resistent hash function $H()$ is set up to be used.
3. Initiation and top up
   A user accesses a vender machine to fill up a new RFID card or top up an old RFID card. The user pays money for t time usage of the public transport system. The vender machine operates as follows.
   (a) The vender machine randomly chooses $2t$ integers $a_1, a_2, \ldots, a_{2t}$ in Z_L where L is a security parameter decided by setting of $H()$ and system parameters like the number of users and the size of the public transport system.
   (b) The vender machine writes $a_2, a_3, \ldots, a_{2t}$ to the memory space for secret data in the RFID card. It writes a_1 to the outward readable memory space of the RFID card.
   (c) The vender machine separately inserts $H(a_1), H(a_2), \ldots, H(a_{2t})$ into the database. The vender machine does not send the records for the tokens of a card together in a batch so that neither the receiving database nor an eavesdropper can tell which records belong to the same RFID card. Instead some of the records of a newly updated RFID card is submitted, being mixed with the unsubmitted records of earlier updated RFID cards. The unsubmitted records of the newly updated RFID card will be submitted later, being mixed with the records of later updated RFID cards. This data insertion mechanism is called SMI (separate and mixed insertion). As no user will use up all his tokens just after buying them, SMI does not affect usability of the tokens.
4. Using the public transport
   (a) When a user enters the public transport system, he puts his RFID card on a checking machines (verifier or called reader). The checking machine reads the token in the outward readable memory space of the RFID card. Suppose the token it obtains is u.
   (b) The checking machine queries the database to search for $H(u)$. If $H(u)$ is a record in the database, the RFID card passes the authentication and the user is allowed to enter.
   (c) If the searched item is a record in the database, the database automatically deletes the record.
   (d) After being allowed in, the RFID card removes one token from its memory space for secret data and puts it in its outward readable memory to replace the used token.
   (e) When a user leaves the public transport system, he puts his RFID card on a checking machines. The checking machine reads the token in the outward readable memory space of the RFID card. Suppose the token it obtains is u'.
   (f) The checking machine queries the database to search for $H(u')$. If $H(u')$ is a record in the database, the RFID card passes the authentication the user is allowed to leave. Otherwise the user is punished.

(g) If the searched item is a record in the database, the database automatically deletes the record.
(h) After being allowed to leave, the RFID card removes one token from its memory space for secret data and puts it in its outward readable memory to replace the used token.

Protocol 1 can prevent some usual attacks.

- As one-time secret is used for authentication, replay attack cannot work no matter the attacker is the user himself or a third party.
- Forging credit attack is prevented as the credit information is stored in a secure database.
- Forging attack against authentication is difficult. With the assumption that the data stored in the RFID memory for secret information is unreadable without the card user's cooperation, the verifiers cannot obtain any token from any RFID card before it is used. As $H()$ is one-way, even if the communication between the vender machines and the database is intercepted by an attacker, it cannot find any token from the intercepted verification information. So unless the card owner or the vender machine is the attacker, forging attack against authentication cannot work. Even if a card owner launches a forging attack, takes out some tokens from his RFID card and copies it into a forged card, he does not benefit from the attack and cannot double use any of his tokens as the tokens are one-time secrets. So the only harmful forging attack is launched by the vender machine, who should be trusted not to record the tokens it generates and use them to launch a forging attack.
- As one-time secret tokens are used in authentication and submission of verification records from the vender machines to the database is through SMI, no one can link different tokens for a user unless a vender machine colludes with multiple verifiers. The only possible tracing attack is a collusion between a vender machine and multiple verifiers. The vender machine records the tokens of a user and share them with many verifiers. The verifiers look for the revealed tokens and record the locations they appear. If the vender machines are assumed to be trusted, no tracing attack can work.

2.3 Protocol 2

Protocol 1 is very efficient as it only employs hash function and does not need any asymmetric cipher. The RFID cards in it do not even need to carry out any computation. However, the vender machines must be trusted in it. Although techniques like tamper-resistent device may be applied to the vender machines to stengthen security and reduce the risk, the trust assumption is still too strong in some circumstances. So Protocol 2 is designed to remove the trust assumption. In Protocol 2, the RFID cards need to perform some computations when being filled up or topped up. As the computation does not need to be real-time and mostly depends on symmetric cipher, it is acceptable. Especially, some costly computation can be performed before hand so that when they are needed they

are ready for use already. In protocol 2, a RFID card generates the tokens it buys itself using a hash chain. It then submits the end of the hash chain to the vender machine it uses, who forwards the end of the hash chain to the database. The vender machine helps the RFID card to encrypt the other nodes of the hash chain, which are stored in the RFID card as tokens. Detailed description of the protocol is as follows.

1. A database is set up. Multiple authorities are in charge of the database and for security they share the power of managing the database. The authorities receive records from the vender machines and insert them into the database. They also handle the verifiers' queries and search the database to answer them. As mentioned before, necessary security measures for database are employed to guarantee security of the database. The database authorities set up a Paillier encryption algorithm and they share the private key. Decryption is feasible only if the number of cooperating authorities is over a threshold. To learn more details about secure sharing of private key in Paillier encryption, interested readers are referred to [2].
2. A one-way collision-resistent hash function $H'()$ is set up and published.
3. Initiation and topping up
   A user accesses a vender machine to fill up a new RFID card or top up an old RFID card. The user pays money for t time usage of the public transport system. He and the vender machine operate as follows.
   (a) The vender machine generates $2k$ probabilistic random encryptions of 0 using the database authorities' public key: $e_0, e_1, \ldots, e_{2t-1}$. Note that although the generation needs $2k$ modulo exponentiations, it can be performed before hand so that when they are needed they are ready for use already. So it does not affect efficiency. The vender machine gives $e_0, e_1, \ldots, e_{2t-1}$ to the RFID card.
   (b) The RFID card chooses a seed s from the input space of $H'()$ and calculates tokens $a_i = H'(a_{i-1})$ for $i = 1, 2, \ldots, 2t$ where $a_0 = s$.
   (c) The RFID card calculates $b_i = a_i e_{\pi(i)}$ for $i = 0, 1, \ldots, 2t-1$ where $\pi()$ is a permutation of $\{0, 1, \ldots, 2t-1\}$. It stores $b_0, b_1, \ldots, b_{2t-2}$ in the memory space for secret data in his RFID card. $b_0, b_1, \ldots, b_{2t-2}$ are stored in their order. More precisely, the memory space for secret data is a stack such that b_0 is in the bottom of the stack, b_1 is on top of b_0, b_2 is on top of b_1, ..., b_{2t-2} is on top of b_{2t-3}. The RFID card stores b_{2t-1} in the outward readable memory space of the RFID card.
   (d) The RFID card submits a_{2t} to the vender machine.
   (e) The vender machines submits $(a_{2t},\ 2t)$ to the database.
   (f) The database authorities store $(a_{2t}, 2t)$ in the database as a record.
4. Using the public transport
   (a) When a user enters the public transport system, he puts his RFID card on a checking machines (verifier or called reader). The checking machine reads the data in the outward readable memory space of the RFID card. Suppose the data it obtains is v.

(b) The checking machine sends v to the database.
(c) The database authorities cooperate to decrypt v and obtain the message in it r. Details of distributed Paillier decryption can be found in [2]. If $H'(r)$ is in a record in the database and the other item in the same record (indicating the left credits the user has) is larger than zero,
   i. the RFID card passes the authentication and the user is allowed to enter;
   ii. the database aothorities replace $H'(r)$ with r;
   iii. 1 is subtracted from the other item in the same record.
   Otherwise, entry is rejected.
(d) After being allowed in, the RFID card removes one token on the top of its memory stack for secret data and puts it in its outward readable memory to replace the used token.
(e) When a user leaves the public transport system, he puts his RFID card on a checking machines. The checking machine reads the data in the outward readable memory space of the RFID card. Suppose the data it obtains is v'.
(f) The checking machine sends v' to the database.
(g) The database authorities cooperate to decrypt v' and obtains the message in it r'. If $H'(r')$ is a record in the database and the other item in the same record (indicating the left credits the user has) is larger than zero,
   i. the RFID card passes the authentication and the user is allowed to leave;
   ii. the database aothorities replace $H'(r')$ with r';
   iii. 1 is subtracted from the other item in the same record.
   Otherwise the user is punished.
(h) After being allowed to leave, the RFID card removes one token on the top of its memory stack for secret data and puts it in its outward readable memory to replace the used token.

Protocol 2 can prevent usual attacks

- As one-time secret is used for authentication, replay attack cannot work no matter the attacker is the user himself or a third party.
- Forging credit attack is prevented as the credit information is stored in a secure database.
- Forging attack against authentication is difficult. With the assumption that the data stored in the RFID memory for secret information is unreadable without the card user's cooperation, the verifiers cannot obtain any token from any RFID card before it is used. The tokens are generated by the RFID cards themselves and are unknown to the vender machines. So without any trust on the vender machines it is guaranteed that the tokens are not revealed. The vender machines only know the end of each hash chain, whose other nodes are the authentication secrets. As $H()$ is one-way, the vender machines cannot obtain any authentication secrets. For the same reason, even if an end of a hash chain sent from a vender machine to the database

is intercepted by an attacker, it cannot find any secret token. So unless the card owner is the attacker, forging attack cannot work. Even if a card owner launches a forging attack, takes out some tokens from his RFID card and copies it into a forged card, he does not benefit from the attack and cannot double use any of his tokens as the tokens are one-time secrets.
– As one-time secret used for authentication are in ciphertext, the verifiers cannot link the one-time tokens of the same user although they are in a hash chain. So even if a vender machine colludes with multiple verifiers, no tracing attack can work as none of them knows the plaintexts of the one-time tokens.

3 Conclusion

Each of the two protocols has its advantages and suitable application circumstance. Protocol 1 is suitable for applications requiring high efficiency and tolerating some trust assumption, while Protocol 2 is suitable of applications requiring high security and less critical with efficiency. The two protocols show that the three principles can be applied to design secure RFID ticket systems in practice. The two new RFID ticket protocols in this paper are secure and efficient and are respectively suitable for two different kinds of public transport applications.

References

1. Dimitriou, T.: A lightweight rfid protocol to protect against traceability and cloning attacks. In: ICSPEACN 2005, pp. 59–66 (2005)
2. Fouque, P., Poupard, G., Stern, J.: Sharing decryption in the context of voting or lotteries. In: Frankel, Y. (ed.) FC 2000. LNCS, vol. 1962, pp. 90–104. Springer, Heidelberg (2001)
3. Camenisch, J., Ateniese, G., de Medeiros, B.: Untraceable rfid tags via insubvertible encryption. In: ACM CCS 2005, pp. 92–101 (2005)
4. Henrici, D., Muller, P.: Hash-based enhancement of location privacy for radio-frequency identification devices using varying identifiers. In: IEEE CPCC 2004, pp. 149–153 (2004)
5. Juels, A., Pappu, R.: Squealing euros: Privacy protection in rfidenabled banknotes. In: Wright, R.N. (ed.) FC 2003. LNCS, vol. 2742, pp. 103–121. Springer, Heidelberg (2003)
6. Lim, C., Kwon, T.: Strong and robust rfid authentication enabling perfect ownership transfer. In: Ning, P., Qing, S., Li, N. (eds.) ICICS 2006. LNCS, vol. 4307, pp. 1–20. Springer, Heidelberg (2006)
7. Molnar, D., Wagner, D.: Privacy and security in library rfid: Issues, practices, and architectures. In: ACM CCS 2006, pp. 210–219 (2006)
8. Song, B., Mitchell, C.: Rfid authentication protocol for low-cost tags. In: ACM CWNS 2008, pp. 140–147 (2008)
9. Rivest, R., Weis, S., Sarma, S., Engels, D.: Security and privacy aspects of low-cost radio frequency identification systems. In: ICSPC 2003, pp. 50–59 (2003)
10. Tsudik, G.: Ya-trap: Yet another trivial rfid authentication protocol. In: IEEE PCCW 2006, pp. 640–643 (2006)
11. Sadeghi, A., Visconti, I., Wachsmann, C.: User privacy in transport systems based on rfid e-tickets. In: PILBA 2008 (2008)