

Secret Searching in Wireless Sensor Networks with RFIDs

Shih-I Huang Shiuhpyng Shieh
Dept. of Comp. Sci. & Info. Eng.
National Chiao Tung University, Hsinchu, Taiwan 300
{sihuang,ssp}@csie.nctu.edu.tw

Abstract

In this paper, we propose a network architecture with RFIDs and sensor nodes (*ARIES*), a mutual authentication protocol (*AMULET*), and a secret search protocol (*ASSART*). In *ARIES*, the distance limitation can be relieved with the help of widely deployed RFID-aware sensor nodes. *AMULET* can perform mutual authentication and reduce the cost for re-authentication. *ASSART* solves the privacy problem by offering a secret search mechanism over encrypted data. In this way, data will not be disclosed during communication and query processes.

Keywords: RFID, Wireless Sensor Networks, Authentication, Privacy, Secret Search

1. Introduction

To search unencrypted data in a conventional remote database is relatively easy, but it leads to a problem that these searching queries may leak private information during transmission. One possible solution to prevent data leaking is to encrypt original data and put encrypted data in remote database. However, encryption causes problems when performing queries.

In a network composed of wireless sensor nodes and RFIDs, encryption is almost unaffordable. How to re-design encryption schemes is a challenging task. In these environments, the collaboration of sensor nodes and tags can form a dynamic, distributed database, where each sensor node contains a tiny database, and each element of the database is composed of data stored in RFIDs. Since sensor nodes are widely deployed, they form a group of distinctive databases. To solve security problems mentioned above, a simple way is to encrypt and store data in each database. However, it raises the secret searching problem when authorized readers want to search a specific target in encrypted form.

With highly constrained computation capability and storage, Weis et al.[11] suggest a randomized lock protocol for private authentication. However, their scheme is neither private nor secure against passive eavesdroppers. Wagner et al. [3] propose a PRF-based private authentication protocol to improve Weis's protocol. However, in both protocols, the tag needs to be re-authenticated even it has been authenticated by one authorized reader beforehand. This is computationally waste and unnecessary.

One acute secret search problem occurs when data is encrypted and stored in tags. Though data security can be attained, data cannot retrieved by performing search queries merely by plaintexts [1]. Many researchers have investigated secret search over encrypted problem when using an untrusted file server or external untrusted memory [7]. One of the premier research works [2] provides secret search in the sense that the untrusted servers cannot learn anything about the plaintext when only given the ciphertext. In their scheme, data needs to be encrypted beforehand with complex encryption operations. This is unavailable for both tags and sensors. Therefore, this scheme is not well-suitable for RFID applications. Another secret search solution is to support searching over encrypted data by using multi-party computation and oblivious functions [5][9]. This solution requires high computation overhead, and therefore is not applicable in sensor architecture.

Our contributions are threefold. First, we propose an architecture of RFIDs and

RFID-aware sensor networks (*ARIES*). Second, we design a mutual authentication protocol (*AMULET*), which is feasible for RFIDs and sensor nodes. Third, a secret search protocol (*ASSART*) is proposed for readers to search secret over private data in an encrypted form.

The remainder of this paper is organized as follows. Section 2 introduces our proposed architecture of RFID and sensor networks. Also, a mutual authentication protocol for readers and tags (*AMULET*) is proposed to prevent passive eavesdropping. In section 3, a secret search protocol (*ASSART*) is proposed to query encrypted data. No private data will be leaked during wireless transmission. Section 4 gives proofs to our proposed schemes, and section 5 concludes our work.

2. *ARIES*

To solve distance limitation problem, we propose an ARchitecture of RFIDs and RFID-aware sEnsor networkS (*ARIES*). Sensor nodes can read data on tags as a bridge between readers and tags. In this architecture, every target can be traced even they are located far away from readers; therefore, finding misplaced targets, such as books, can be solved.

A *RFID-reader* is a device can perform read, write or overwrite operations on RFID tags through wireless interface. All *readers* share a database storing all authorized *IDs*. The readers share a unique secret key s with each tag, and s is saved in both tag and the database shared by readers. That is, readers save all pairs of (s, ID) in the shared database, and each tag save its individual secret key s . Each reader also saves an unique encryption key EK_i to encrypt data and no other devices know this key. Since EK_i is privately saved, it can be used to verify the ownership of encrypted data.

A *tag* is a small, thin, readable, and writeable device which can store limited data. One restriction of tags is that each tag has only limited computation capability. Computation intensive operations, such as encryption, are inadequate for tags.

To reduce the effort for building secure channels between readers and sensor nodes, tags save two secret keys. One key is used to build secure channel with readers, the other is used to build secure channel with sensor nodes.

A RFID-aware sensor node is a tiny device capable of detecting RFID tags. It has a RFID-aware sensor, and uses a transceiver and a receiver to communicate with readers and tags through wireless interface. Sensor nodes are cheap and can be widely dispersed. Each sensor saves two secret keys shared with readers and tags.

To prevent replay attacks, we assume that each reader and sensor node has a synchronized timer. Therefore, re-authentication processes can use the timer to verify whether current re-authentication process is expired or not. The synchronization does not need to perform frequently because authentication may not operate constantly. Loosely time-synchronization would be secure enough for authentication. Since much research has investigated on time synchronization [6][9], we do not intend to spend time on this issue.

One restriction needs to solve is distance limitation. Since readers cannot be widely deployed, sensor nodes can make up for this need. We assume that every sensor can multihopped to authorized readers and every sensor node has a secret key SR shared with readers. With SR , readers and sensor nodes can maintain secure communication. We do not intend to introduce the security algorithm between readers and sensors. Instead, we merely indicate that the channel between readers and sensor nodes are secure by storing shared secret keys or pre-distributing verifiable key pairs [8].

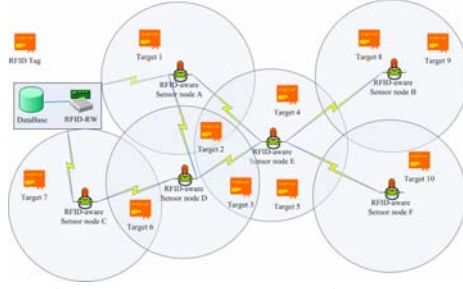


Figure 1: ARIES architecture.

Figure 1 shows *ARIES* architecture of using RFID and sensor wireless sensor networks. In the architecture, sensor nodes can collect data stored on tags, and as a whole can be viewed as a distributed database with tiny databases. Each attribute of the distributed database represents characteristics of the target. As an example, the distributed tiny database can be represented as following table (Table 1):

Target ID	Sensor ID	Attr1	Attr 2	Attr N
ID_1	Sensor A	Attr(A1)	Attr(A2)	Attr(An)
ID_2	Sensor A	Attr(A1)	Attr(A2)	Attr(An)
ID_2	Sensor B	Attr(B1)	Attr(B2)	Attr(Bn)
.....					
ID_2	Sensor K	Attr(K1)	Attr(K2)	Attr(Kn)

Table 1: An example of the distributed tiny database.

2.1 AMULET

Authentication is the first necessary process to build trust relationship between readers and tags. Because the communication between reader and tag is wireless, there is a possibility for attackers to eavesdrop the transmitted data, including password. Much research has shown that the RFID communication is an asymmetry in signal strength. That is, it will be much easier for attackers to eavesdrop on signals from reader to tag than on data from tag to reader. With this property, we propose **A** **M**utual authEntication proTocol for readers and tags (*AMULET*) to enhance passwords wireless communication between RFID tags and readers.

At setup time, we give each tag a unique secret s and identification ID , and the reader has a database D storing all pairs of (s, ID) . As the protocol shown in Figure 2, *AMULET* involves the following steps:

1. The reader chooses a random number $R_1 \in \{0,1\}^n$, current time T_1 , and calculates $f_s(R_1, T_1)$. All R_1 , T_1 and $f_s(R_1, T_1)$ are then sent to the tag, including a *Hello* bit to indicate the beginning of the authentication process.
2. When the tag receives a *Hello* packet, it chooses a random number $R_2 \in \{0,1\}^n$, current time T_2 , and calculates $\alpha = ID \oplus f_s(R_1, R_2, T_2)$. f_s is a pseudo random function (PRF). The R_2 , T_2 and α are then sent back to the reader. The tag also saves one copy of R_2 and T_2 in its storage.
3. Whenever the reader receives R_2 , T_2 and α , firstly it checks whether $ID = \alpha \oplus f_s(R_1, R_2, T_2)$ or not. If the condition is satisfied and $T_2 > T_1$, it then picks current time T_{now} , calculates $\beta = ID \oplus f_s(R_1, R_2, (T_{now} - T_2))$ and sends both T_{now} and β together to the tag along with an *Ack* bit indicating the acknowledgement. Meanwhile, the reader updates original (s, ID) pair to (s, ID, R_2, T_2) .
4. The tag then verify it by checking $ID = \beta \oplus f_s(R_1, R_2, T_2)$ and $f_s(R_1, R_2, (T_{now} - T_2))$. If both conditions are satisfied, the authentication succeeds.

Since it is harder to eavesdrop on the channel from tag to reader than from the reader to

tag, *AMULET* can provide security against passive eavesdropping on the reader-to-tag link.

A common attack to authentication protocols is man-in-the-middle attack. *AMULET* has the nature to resist such an attack. In *AMULET*, the reader will send a random number R_1 to the tag. Then the tag chooses $R_2 \in \{0,1\}^n$ and current time T_2 ; calculates $\alpha = ID \oplus f_s(R_1, R_2, T_2)$; send them all back to the reader. If an attacker tries to perform man-in-the-middle attack, he can eavesdrop on R_1 , R_2 , T_1 , T_2 and $\alpha = ID \oplus f_s(R_1, R_2, T_2)$. However, since the attacker does not know the secret key s , he can not modify $f_s(R_1, R_2, T_2)$. Therefore, man-in-the-middle attacks will not success. Therefore, our protocol can provide security against man-in-the-middle attacks. We will formally prove this property in section 4.

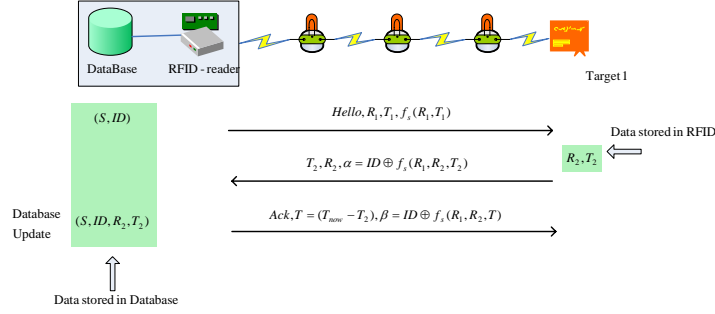


Figure 2: AMULET architecture.

Whenever a reader wants to send commands to authenticated tag, the authentication process does not need to rerun again. After authentication process, reader's database will update original (s, ID) pair to (s, ID, R_2, T_2) . As shown in Figure 3, the authenticated command can be verified by the following two steps:

1. A new reader can query the database and retrieve (s, ID, R_2, T_2) . When a reader retrieves (s, ID, R_2, T_2) instead of (s, ID) , it then knows that the tag with this ID is authenticated by another reader before. As a result, it chooses a random number $R'_1 \in \{0,1\}^n$, current time T_{now} and calculates $\beta' = ID \oplus f_s(R'_1, R_2, T_{now} - T_2)$. The reader then sends its command Cmd' along with R'_1 and β' to the tag.
2. When the tag receives Cmd' , R'_1 and β' , it checks if $ID = \beta' \oplus f_s(R'_1, R_2, (T_{now} - T_2))$ and $f_s(R'_1, R_2, (T_{now} - T_2)) = \beta' \oplus ID$. If both conditions are satisfied, then the tag can execute Cmd' . Otherwise, the tag drops it directly.

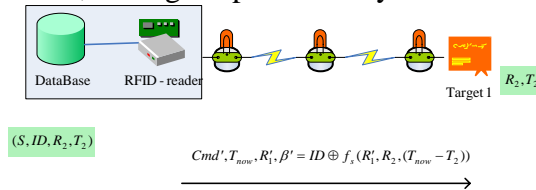


Figure 3: Commands verification without re-authentication process.

In this way, the re-authentication cost can be reduced. Also, replay attacks can be solved by checking whether T_{now} is expired and $f_s(R'_1, R_2, (T_{now} - T_2))$ is sustained or not. Therefore, both man-in-the-middle attacks and replay attacks cannot succeed.

3. ASSART

To maintain data privacy, one approach is to encrypt all attributes so that attackers cannot decrypt it and get that data. However, traditional cryptographic encryption is not feasible in tags and sensor nodes because their computation capability is limited. Moreover, it is hard to search on encrypted data. To solve the problem, we propose **A Secret SeARch proTocol** (*ASSART*). *ASSART* saves original data into a secret form, and allow authorized readers to search secret data on tags so that no private information will be disclosed during transmission or query processes.

Basically each tag can store data related to the target. Each characteristic of the target corresponds to an attribute of the target. As an example, a tag attached to a book may store the book ID, book title, authors, check-in and check-out time, borrower IDs, etc. Therefore, the status of a target can be formally described as $B = (Attr1, Attr2, \dots, AttrN)$, where N is the number of attributes. Some attributes are personal privacy and should not be exposed to unauthorized readers or attackers.

As shown in Figure 4, ASSART involves the following steps:

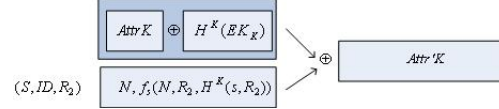


Figure 4: ASSART operations for attribute K .

1. For an attribute $AttrK$, the reader first generates $H^K(s, R_2)$ by iteratively hashing (s, R_2) for K times, where K indicates the number of sequential order of $AttrK$.
2. The reader generates $H^K(EK_i)$ by iteratively hashing EK_i for K times.
3. The reader calculates $f_s(N, R_2, H^K(s, R_2))$ and concatenates it with N . Let $\lambda = N, f_s(N, R_2, H^K(s, R_2))$.
4. The transformed attribute $Attr'K = AttrK \oplus H^K(EK_i) \oplus \lambda$. The reader then overwrites $AttrK$ by $Attr'K$.

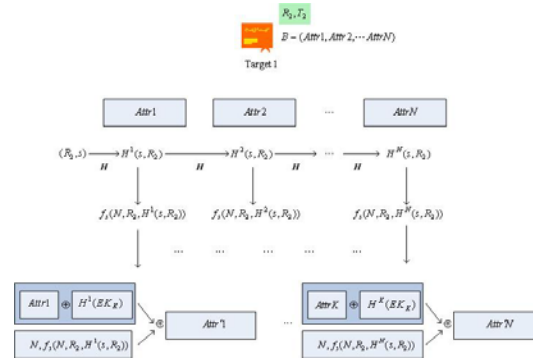


Figure 5: ASSART operations.

After every attribute is overwritten, attackers will learn nothing from the encrypted data even they have readers. Since K is different in all attributes, each attribute generates different $H^K(s, R_2)$. With different $H^K(s, R_2)$, even if some attribute values happen to be the same, it will generate different encrypted attribute values. This will keep attributes private and secure. Figure 5 shows the overview of ASSART operations.

For authorized readers, whenever a transformed attribute $Attr'K$ is given, they can inversely-transform it to $AttrK$ by computing:

$$AttrK = Attr'K \oplus H^K(EK_i) \oplus N, f_s(N, R_2, H^K(s, R_2)).$$

Since authorized readers can retrieve (S, R_2) from the database, they can easily calculate $AttrK$ without exposing sensitive and private data during wireless transformation.

A major contribution of ASSART is that it can maintain the rest of attributes private, even if some attributes are compromised. Since $f_s(N, R_2, H^K(s, R_2))$ varies according to different K , $Attr'(K+1) = Attr(K+1) \oplus H^{K+1}(EK_i) \oplus (N, f_s(N, R_2, H^{K+1}(s, R_2)))$ will remain secure even when $f_s(N, R_2, H^K(s, R_2))$ is compromised. This will keep the rest of attributes uncompromised.

For searching an attribute $AttrK$ in a sensor node, the RFID-reader does not send $AttrK$ in plaintext during query process. Instead, the RFID-reader broadcasts

$AttrK \oplus H^K(EK_i)$ to all sensor nodes. Then each sensor node calculates $Attr'K$ by $Attr'K = AttrK \oplus H^K(EK_i) \oplus N, f_s(N, R_2, H^K(s, R_2))$ with its own s , R_2 and all K 's. The reason why each sensor node needs to calculate $Attr'K$ with all K 's is because the sensor node does not know what K exactly is. If any sensor node finds the $Attr'K$, it returns $Attr'K$, $K = 1, 2, \dots, n$, to the RFID-reader. Since data are encrypted, data secrecy and privacy can be attained during transmission processes.

4. Security Analysis

In this section, we first show *AMULET* is secure under man-in-the-middle attacks. Second, security analysis in terms of the resources needed to break the secret search protocol is discussed.

We classify man-in-the-middle attacks into three types. Type-1 attack modifies R_1 only, type-2 attack modifies R_2 only, and type-3 attack modifies R_1 , R_2 , and α . We will show that these three types of attacks cannot work in our authentication protocol. Before we start our proof, several definitions and theorems is given below.

Def 1: (Instance) Since the status B of a target is composed of the target's ID and attributes, B can be formally described as $B = (ID_B, Attr1, Attr2, \dots, AttrN)$. An instance X_B is defined as $X_B = (Attr1, Attr2, \dots, AttrN)$, and a verification function V_f is defined as

$$V_f(X_B) = \sum_{i=1}^n Attr_i.$$

Def 2: (Distinguishable) Two instances of a target are distinguishable if they are different in any attributes.

Def 3: (R -Breakable) Let an instance $X_B = (Attr1, Attr2, \dots, AttrN)$. If X_B can be derived from R ($R \leq N$) attributes, then it is R -Breakable. Under the same condition, a system is R -Breakable if it needs R resources to break the system.

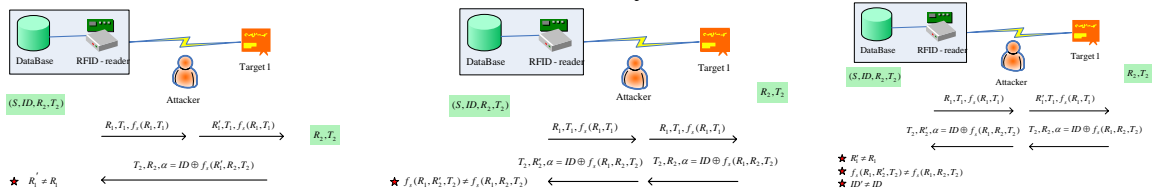


Figure 6: Type-1, Type-2 and Type-3 man-in-the-middle attacks (left to right)

Type-1 attacker, shown in Figure 6, eavesdrops on R_1 , generates a false value R_1' and delivers it to the tag. The tag then uses R_2 to generate $\alpha = ID \oplus f_s(R_1', R_2, T_2)$ and transmits R_2 , T_2 and α back to the reader. Since $R_1 \neq R_1'$, the reader will find that $f_s(R_1, R_2, T_2) \neq f_s(R_1', R_2, T_2)$. Therefore the readers can prevent type-1 man-in-the-middle attacks.

Type-2 attacker eavesdrops on R_2 , generates a false value R_2' , and transmits R_2' and α back to the reader. Since $R_2 \neq R_2'$, the reader will find that $f_s(R_1, R_2, T_2) \neq f_s(R_1, R_2', T_2)$. Therefore the readers can prevent type-2 man-in-the-middle attacks.

Type-3 attacker generates false R_1' , R_2' , and α' back to the reader and the tag separately. Since s is kept in secret, the reader will find that $f_s(R_1, R_2, T_2) \neq f_s(R_1', R_2', T_2)$ and $ID \neq ID'$. Therefore, type-3 man-in-the-middle attacks will fail.

For *ASSART* protocol, we will prove their security strength in terms of the secrecy of attributes. In *ASSART*, the $Attr'K$ can be calculated by the following equation

$$Attr'K = AttrK \oplus H^K(EK_i) \oplus (N, f_s(N, R_2, H^K(s, R_2))) \quad (\text{Eq. 1})$$

Theorem 1 proves that it takes both s and R_2 to compromise $Attr'K$.

Theorem 1: Let $T = f_s(N, R_2, H^K(s, R_2))$. T is (s, R_2) -breakable.

Proof: Since N and K may be eavesdropped by attackers, only s and R_2 are kept secret. Therefore, attackers need to compromise both s and R_2 to compromise T . Therefore T is (s, R_2) -breakable.

To evaluate the security strength of a system, an approach is to verify the number of resources needed to compromise the system. It is shown in the theorem that it needs both s and R_2 to compromise an attribute.

An instance is a collection of all attributes of a tag. The security strength of an instance is defined as the number of attributes needed to be compromised. As more attributes are distinguishable, higher security level can be attained.

Theorem 2: For all $Attr'I, Attr'J$, where $I \neq J$ and X is an instance composed of $Attr'I$ and $Attr'J$, $X = V_f(Attr'I, Attr'J)$. There does not exist $Attr''I, Attr''J$, such that $V_f(Attr'I, Attr'J) = V_f(Attr''I, Attr''J)$

Proof: Let $AttrI$ and $AttrJ$ are two original attributes and $I > J$. $Attr'I$ and $Attr'J$ are transformed attributes, and $X = V_f(Attr'I, Attr'J)$. If the attacker generates $Attr''I$ and $Attr''J$, we will prove that it cannot satisfies $X = V_f(Attr''I + Attr''J)$.

Since $X = V_f(Attr'I + Attr'J)$, therefore

$$X = (AttrI \oplus H^I(EK_i) \oplus (N, f_s(N, R_2(H^I(s, R_2)))) + (AttrJ \oplus H^J(EK_i) \oplus (N, f_s(N, R_2(H^J(s, R_2))))).$$

One major property in our data transformation protocol is that $AttrI$ can be used to authenticate $AttrJ$ by checking following equation.

$$H^J(s, R_2) = H^{J-I}(H^I(s, R_2)) \quad (\text{Eq. 2})$$

Therefore if attacker generate $Attr''I$ and $Attr''J$, $H^I(s, R_2)$ and $H^J(s, R_2)$ can be calculated by the following two equations.

$$Attr''I \oplus AttrI = N, f_s(N, R_2, H^I(s, R_2)) \quad (\text{Eq. 3})$$

$$Attr''J \oplus AttrJ = N, f_s(N, R_2, H^J(s, R_2)) \quad (\text{Eq. 4})$$

Since s and R_2 are kept secret only between authorized readers and tags, the attackers cannot generate false $H^I(s, R_2)$ and $H^J(s, R_2)$. Therefore, two attributes are distinguishable.

Since any two attributes are distinguishable, even attackers can generate different attributes, the attackers cannot cheat readers.

Next, we will show that an instance of a target B is N -breakable and distinguishable, where N is the number of attributes of B .

Theorem 3: Let $V_f(B) = \sum_{i=0}^n Attr_i = Attr'1 + Attr'2 + \dots + Attr'N$. B is N -breakable and distinguishable.

Proof: Let $B = (Attr1, Attr2, \dots, AttrN)$ and B' be the data after transformation and $B' = (Attr'1, Attr'2, \dots, Attr'N)$. We prove N -breakable property of an instance by induction.

Let $N = 2$. According to theorem 2, it is show that B is 2-breakable.

Suppose when $N=P$, B is P -breakable. We want to prove B is P -breakable when $N=P+1$.

Without lost of generosity, we assume that $B = (Attr1, Attr2, \dots, AttrN)$ and B is P -breakable.

Without lost of generosity, let $B_1 = (Attr1, Attr2, \dots, AttrN, AttrN+1)$. From theorem 2, we know that every two attributes are distinguishable. Therefore, $AttrN+1$ and $AttrM$ are distinguishable for $M = 1, 2, \dots, N$ by verifying $H^{N+1}(s, R_2)$ and $H^1(s, R_2), H^2(s, R_2), \dots, H^N(s, R_2)$ respectively. Since all $N+1$ attributes are distinguishable, it is proved that an instance of a target is N -breakable.

If the new attribute $AttrK$ is inserted between $Attr1$ to $AttrN$, $AttrK$ can be verified

by both its previous attribute $Attr(K-1)$ and its following attributes $Attr(K+1)$ by the following equations:

$$H(H^{K-1}(s, R_2)) = H^K(s, R_2) \quad (\text{Eq. 5})$$

$$H(H^K(s, R_2)) = H^{K+1}(s, R_2) \quad (\text{Eq. 6})$$

If both eq.5 and eq.6 are satisfied, the added attributed $Attr(K)$ is valid. Otherwise, $Attr(K)$ is invalid and should be discarded.

Theorem 3 indicates that attackers need to compromise all attributes to cheat readers. If only a portion of attributes are compromised, still the reader can verify it.

5. Conclusion

In this paper, we present an *AREIS* architecture to solve the distance limitation problem in RFID applications. Targets at a distance still can be monitored with the assistance of RFID-aware sensor nodes. An authentication protocol, *AMULET*, is also proposed to mutually authenticate readers and tags. *AMULET* can resist man-in-the-middle attacks and reduce re-authentication overhead. Finally we propose a search protocol, *ASSART*, to search on secret data. Information will not be disclosed during the search process. *ASSART* uses a key chain to improve data security. Even if some attributes are compromised; the rest of attributes are still kept in private.

6. References

- [1] B. Chor, O. Goldreich, E. Kushilevitz, and M. Sudan, Private Information Retrieval, In *Proceedings Journal of the ACM*, pp.965-981, 1998.
- [2] D. Song, D. Wagner, and A. Perrig, Practical Techniques for Searches on Encrypted Data., In *Proceedings of IEEE Symposium on Security and Privacy*, pp.44-55, 2000.
- [3] David Molnar and David Wagner, Privacy and Security in Library RFID Issues, Practices, and Architectures, In *Proceedings of ACM Conference on Computer and Communication Security*, pp.210-219, 2004.
- [4] Estrin, D., Govindan, R., Heidemann, J., Kumar, Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp.263-270, 1999.
- [5] H.-M. Sun and S.-P. Shieh, An Efficient Construction of Perfect Secret Sharing Schemes for Graph-based Access Structures, In *Proceedings of Computers and Mathematics with Applications*, pp.129-135, 1996
- [6] Jana van Greunen and Jan Rabaey, Lightweight Time Synchronization for Sensor Networks, In *Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp.11-19, 2003.
- [7] K. Bennett, C. Grothoff, T. Horozov, and I. Patrascu, Efficient Sharing of Encrypted Data, In *Proceedings of the 7th Australian Conference on Information Security and Privacy*, pp.107-120, 2002.
- [8] Laurent Eschenauer, Virgil D. Gligor, A key-management scheme for distributed sensor networks, In *Proceedings of the 9th ACM Conference on Computer and Communication Security*, pp.41-47, 2002.
- [9] Saurabh Generiwal, Ram Kumar and Mani B. Srivastava, Time-sync Protocol for Sensor Networks, In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp.138-149, 2003.
- [10] Srisathapornphat, C., Jaikao, C., Chien-Chung Shen, Sensor Information Networking Architecture, In *Proceedings of International Workshop on Parallel Processing*, pp.92-95, 2000.
- [11] Stephen A. Weis, Sanjay E. Sarma, Ronald L. Rivest, and Daniel W. Engels, Security and Privacy Aspects of Low-Cost Radio Frequency Identification Systems, In *Proceedings of Pervasive Computing*, pp.201-212, 2004.