# 行政院國家科學委員會專題研究計畫　成果報告

# 無線微型感測網路的安全身分認証與祕密搜尋協定設計與實作(I)

計畫類別： 整合型計畫
計畫編號： NSC94-2213-E-009-089-
執行期間： 94 年 08 月 01 日至 95 年 07 月 31 日
執行單位： 國立交通大學資訊工程學系(所)

計畫主持人： 謝續平

報告類型： 精簡報告

處理方式： 本計畫涉及專利或其他智慧財產權，2 年後可公開查詢

中　華　民　國 95 年 10 月 30 日

# 行政院國家科學委員會專題研究計畫成果報告

無線微型感測網路的安全身分認証與祕密搜尋協定設計與實作

Secure Wireless Sensor Network: Secure, Light-Weight Authentication and Secret Search Protocols for Wireless Sensor Networks

計畫編號：NSC 94-2213-E-009-089
執行期限： 94 年 8 月 1 日 至 95 年 7 月 31 日
全程計畫： 94 年 8 月 1 日 至 95 年 7 月 31 日
主持人：謝續平 教授　國立交通大學資訊工程學系

**中文摘要**

**在此計劃中，我們提出一基於 RFID 與 Sensor nodes(*ARIES*)上的網路架構，其中包含了「雙向認證協定 (*AMULET*) 」以及「密秘搜尋協定 (*ASSART*)」。在 *ARIES* 中，由於我們採用了 RFID-aware sensor nodes 可用來減輕距離的限制；而 *AMULET* 則擁有雙向認證的功能且可大幅減少重新認證的成本；最後在 *ASSART* 中則提供在加密資料中搜尋機制，可用來解決私密性問題，並且資料不會在通訊、查訊過程中被丟棄。**

**關鍵詞：無線射頻識別、無線感測網路、身份認証、私密性、密秘搜尋**

**Abstract**

In this paper, we propose a network architecture consisting of RFIDs and sensor nodes (*ARIES*), a mutual authentication protocol (*AMULET*), and a secret search protocol (*ASSART*). *ARIES* utilizes RFID-aware sensor nodes to alleviate the distance limitation problem. *AMULET* performs mutual authentication and reduces the cost of re-authentication. *ASSART* solves the privacy problem by offering a secret search mechanism over encrypted data, thus preventing data disclosure during communication and query processes.

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

## 1. Introduction

Searching for unencrypted data in a conventional remote database is relatively easy, but it leads to a serious problem: these queries may leak private information during transmission. One possible solution to prevent data leakage is to encrypt the original data and place it in a remote database. However, encryption is almost unaffordable in a network composed of wireless sensor nodes and RFIDs. Redesigning encryption schemes is a challenging task. Furthermore, it is more difficult to search encrypted data.

In a typical application, sensor nodes encrypt data to improve security against intrusions. To search data, a sensor node must first decrypt the data, a process which usually causes significant delay. Moreover, computation-limited, low cost devices,

such as sensor nodes and RFID tags, leave the decrypted data vulnerable to disclosure. In such an exposed environment, it is desirable to develop a new method that performs secret search directly on the ciphertext without decryption, thus, preserving secrecy and avoiding decryption delay.

A RFID tag is a small, low-cost device with limited data storage space. Each tag is assigned an identification (*ID*) to identify itself. Therefore, tagging specific targets with RFID tags allows for individualization and recognition of each target by the attached ID. Through the wireless interface, each tag can report data when queried over radio by a RFID reader. The RFID reader can execute read, write, and overwrite commands on each tag over the wireless interface. However, RFID readers can only recognize tags in close proximity; a data tag that is out of range cannot be read by a reader. This distance limitation severely restricts RFID deployment. Despite equipping readers and tags with longer-range wireless communication capability, RFID readers still have difficulty tracking or monitoring tags at a distance. To solve this distance limitation problem, a wireless sensor network can act as a bridge between the tags and the readers when tracking or monitoring remote targets.

A wireless sensor network [5][10][17] consists of groups of sensor nodes connected by wireless links that perform sensing tasks, such as detecting changes in temperature, pressure, etc. These sensors are employed for specialized tasks like surveillance and security, environmental monitoring, location tracking, warfare, and health care.

Sensor nodes can communicate with RFID tags through the wireless interface. Because sensor nodes are cheap, they can be widely deployed to monitor every target, allowing readers to find targets at a distance. Although the use of sensor nodes solves the distance limitation problem, it introduces additional security challenges.

In the aforementioned environment, the collaboration of sensor nodes and tags can form a dynamic, distributed database, where each sensor node contains a tiny database that tracks the data stored in RFIDs. Since sensor nodes are widely deployed, they form a group of distinctive databases. Simply encrypting the database ensures data security; however, it raises the issue of searching secrets.

For secret searching in wireless sensor networks, the following requirements are considered important:

1) *Secrecy*: Storing data in an encrypted form helps retain its confidentiality. Because sensors are vulnerable, computation-limited, and low cost devices, allowing sensors to decrypt data to perform a search results in unnecessary risk of disclosure. Thus, sensors must execute a secret search directly on

ciphertext, rather than plaintext. Furthermore, data transmitted over a wireless interface is susceptible to exposure. Therefore, sensors must only transmit encrypted data. In summary, the data must remain in an encrypted form and should not be decrypted unless necessary to minimize the possibility of disclosure.

2) *Authentication*: Since the network obtains data from a large number of sensors or tags, attackers can easily acquire readers with the same specifications to extract data stored in the tags. Therefore, both the reader and the tag need to verify the authenticity of its communication counterpart before executing read or write operations.
3) *Integrity*: Assuring data integrity prevents attackers from using unauthorized readers to modify or inject data into databases. Readers or tags must verify data integrity upon receipt of data.
4) *Performance*: Requiring a sensor node to decrypt data before searches causes significant and unnecessary delay. Also, the limited computation capabilities of sensor nodes and tags hinder them from performing complex operations, such as encryption and exponential calculations. Therefore, all operations must be redesigned to fit their computation capabilities.

Previous research focused on authentication. Some papers propose the use of public key infrastructure (PKI) to authenticate two parties through a trusted-third-party. This solution is inadequate for RFID applications since the PKI requires the reader or tag to save private keys and verify the identity of others with the help of the trusted-third-party. Tags have little storage, and they can only transmit data to devices in close proximity. In other words, the trusted-third-party must be located near the tags, which is a difficult requirement to achieve and one that presents other security risks. Moreover, the tag cannot afford the additional computational power required to verify others. Therefore, a PKI scheme is not feasible for RFID applications.

Weis et al.[21] suggest a randomized lock protocol for private authentication in a highly constrained computation and storage environment. However, their scheme is neither private nor secure against passive eavesdroppers. Wagner et al. [3] propose a PRF-based private authentication protocol to improve upon Weis's protocol. Unfortunately, both protocols require re-authentication of a tag even if another authorized reader previously authenticates the tag. These extra steps are computationally wasteful and unnecessary.

Privacy is a major concern encountered in RFID applications [1][4]. A RFID tag may store sensitive data associated with a target, which must

remain private. Since readers, tags, and sensor nodes send messages through a wireless medium, attackers can easily listen in on their communication and extract secret information.

An intuitive way to protect private data is encryption. However, tags and sensor nodes have severely limited storage and computation capability; consequently, traditional cryptographic algorithms are not well-suited for these devices. As a result, we must redesign security mechanisms to support RFID tags and sensor nodes.

A new problem arise from encrypting data: RFID readers cannot easily perform queries on secured data [18]. Many researchers have investigated secret search over encrypted data in an untrusted file server or external memory environment [6][10][13][15][16]. Dawn et al. [2] provide a method for secret searching untrusted servers in a way that prevents disclosure of data when only given the ciphertext. Unfortunately, their scheme requires complex encryption operations unavailable to both tags and sensor nodes. Another solution is to support searching over encrypted data by using multi-party computation and oblivious functions [7][8][9][19][22][23]. However, this solution requires high computation overhead and therefore not applicable in a tag or sensor system.

Our research contribution is threefold. First, we propose an architecture consisting of RFIDs and RFID-aware sensor networks (*ARIES*). This architecture extends RFID's capabilities through a wireless sensor network by utilizing sensor nodes to locate targets at a distance. Second, we design a mutual authentication protocol (*AMULET*) that is feasible for RFIDs and sensor nodes. Moreover, *AMULET* reduces the cost of re-authentication. Third, we present a secret search protocol (*ASSART*) that enables readers to perform searches over encrypted data, allowing data to remain encrypted during transmission or at vulnerable locations. By only using one-way hash functions, pseudo random number generation functions, and XOR operations, *ASSART* accommodates the resource limitations of both tags and sensors.

The remainder of this paper is organized as follows. Section 2 introduces our proposed *ARIES* architecture of RFID and sensor networks, while section 3 presents our *AMULET* mutual authentication protocol for readers and tags. Next, we develop our *ASSART* secret search protocol to query encrypted data in section 4. Finally, section 5 provides proof of our proposed schemes' security, and section 6 concludes our work.

## 2. ARIES

Motivated by the distance limitation problem of RFID readers, we propose an **AR**chitecture of RF**I**Ds and RFID-aware s**E**nsor network**S** (*ARIES*). Sensor nodes can bridge the gap between readers and tags by transmitting commands from

reader to tag or sending tag data to the reader, allowing readers to trace any tag located far away.

Although an *RFID reader* is called a reader by convention, it also has writing capability. Thus, a reader can perform read, write, and overwrite operations on RFID tags through the wireless interface. In our system, readers have access to a shared database storing all authorized *ID*s. To construct a secure channel between readers and tags, the readers share a unique secret key $s$ with each tag. While readers save all tag pairs $(s, ID)$ in the shared database, each tag stores its individual secret key $s$ locally. Additionally, each reader possesses a unique encryption key $EK_i$ to encrypt data, which it saves locally and remotely (on the shared database). $EK_i$ can be used to verify the ownership of encrypted data.

An *RFID tag* is a small, thin, readable, and writeable device that can store limited data. Embedded with a transceiver, each tag can communicate via wireless channels with other devices, such as readers or sensor nodes. Because tags have limited computation capability, intensive operations, such as encryption, are impractical for tags. Therefore, we will introduce new methods in section 3.

An *RFID-aware sensor node* is a tiny device capable of detecting RFID tags. It is also outfitted with a transceiver to communicate with readers and tags through a wireless interface. Like tags, sensor nodes are cheap and widely dispersible.

As mentioned earlier, sensor nodes can compensate for the distance limitation of RFID readers. To reach readers, we assume that the sensor network allows for multi-hop communication. Furthermore, readers, tags, and sensor nodes can maintain secure communications. However, we do not introduce a security scheme between readers and sensors, tags and sensors, or readers and tags; instead, we merely indicate that secure channels exist through shared secret keys or pre-distributed verifiable key pairs [14].

To prevent replay attacks, we assume that each reader, tag, and sensor node has a synchronized timer, allowing them to verify that an authentication process has not expired. Our system only requires loose time synchronization because of infrequent authentication. Because past researchers have investigated time synchronization [11][19], we do not address this issue here.

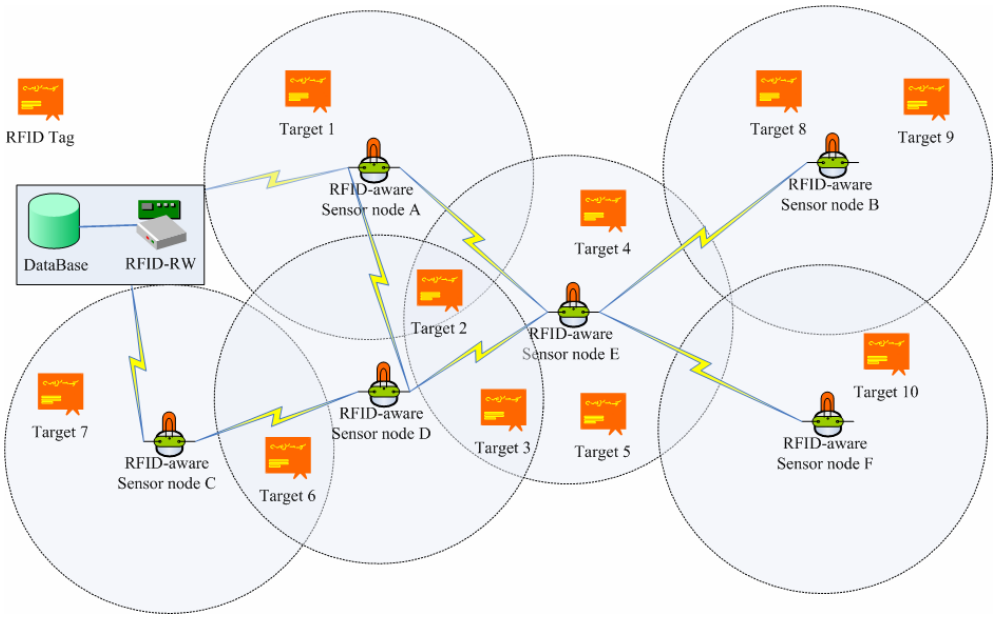


**Figure 1: *ARIES* architecture.**

In our architecture, readers request data from tags via sensor nodes. Figure 1 depicts the RFID readers, RFID tags, and RFID-aware wireless sensor nodes that make up the *ARIES* architecture. The sensor node collects data from tags in its vicinity and stores it in a local tiny database, where each attribute represents characteristics of the target. Table 1 represents a distributed tiny database located on one sensor node.

| Target ID | Sensor ID | Attr 1 | Attr 2 | ---------- | Attr N |
|---|---|---|---|---|---|
| $ID_1$ | *Sensor A* | *Attr*(*A*1) | *Attr*(*A*2) | ---------- | *Attr*(*An*) |
| $ID_2$ | *Sensor A* | *Attr*(*A*1) | *Attr*(*A*2) | ---------- | *Attr*(*An*) |
| $ID_2$ | *Sensor B* | *Attr*(*B*1) | *Attr*(*B*2) | ---------- | *Attr*(*Bn*) |
| ⋮ | | | | | |
| $ID_7$ | *Sensor K* | *Attr*(*K*1) | *Attr*(*K*2) | ---------- | *Attr*(*Kn*) |

**Table 1: Distributed tiny database.**

## 3. AMULET

Authentication is the first step in building a trust relationship between readers and tags. Since readers and tags rely on wireless communication, attackers may eavesdrop on transmitted data and extract passwords. Previous research characterizes RFID communication as asymmetrical in signal strength. That is, attackers have an easier time listening in on signals from reader to tag than on data from tag to reader. Additionally, attackers can easily purchase readers and tags to

perform malevolent operations. Therefore, we propose **A MU**tua**L** auth**E**ntication pro**T**ocol (*AMULET*) for readers and tags to prevent attackers from impersonating authorized entities.

Wagner et al. propose a PRF-based private authentication protocol in [3], which extends Weis's randomized hash lock protocol. Their authentication scheme comprises of a triple of probabilistic polynomial time algorithms $(G, R, T)$ (for Generator, Reader, and Tag). Also, each tag possesses a unique secret $s$ and identification *ID*, and the reader contains a database *D* storing all pairs of $(s, ID)$. In their protocol, each reader needs to authenticate every target, even if another reader previously validates the tag. This redundant authentication imposes unnecessary overhead on low computation power devices.

In our scheme, we assign each tag a unique secret $s$ and identification *ID* and store all the tag pairs $(s, ID)$ in a database *D*. According to the protocol outlined in Figure 2, *AMULET* involves the following steps:

1. To begin the authentication process, the reader chooses a random number $R_1 \in \{0,1\}^n$, checks the current time $T_1$, and calculates $f_s(R_1, T_1)$, where $f_s$ is a pseudo random function (PRF). The reader then sends a *Hello* packet to the tag that includes $R_1$, $T_1$, and $f_s(R_1, T_1)$.
2. When the tag receives a *Hello* packet, it chooses a random number $R_2 \in \{0,1\}^n$, checks the current time $T_2$, and calculates $\alpha = ID \oplus f_s(R_1, R_2, T_2)$. The tag sends a packet containing $R_2$, $T_2$, and $\alpha$ back to the reader and also saves a copy of $R_2$ and $T_2$.
3. Upon receiving $R_2$, $T_2$, and $\alpha$, the reader verifies that $ID = \alpha \oplus f_s(R_1, R_2, T_2)$ and $T_2 > T_1$. It then checks for the current time $T_3$, computes the time difference $T = T_3 - T_2$, calculates $\beta = ID \oplus f_s(R_1, R_2, T)$, and returns an *Ack* (acknowledgement) packet to the tag that includes $T$ and $\beta$. In addition, the reader updates the original tag pair $(s, ID)$ to $(s, ID, R_2, T_2)$.
4. Finally, the tag validates the *Ack* packet by checking that $ID = \beta \oplus f_s(R_1, R_2, T_2)$.

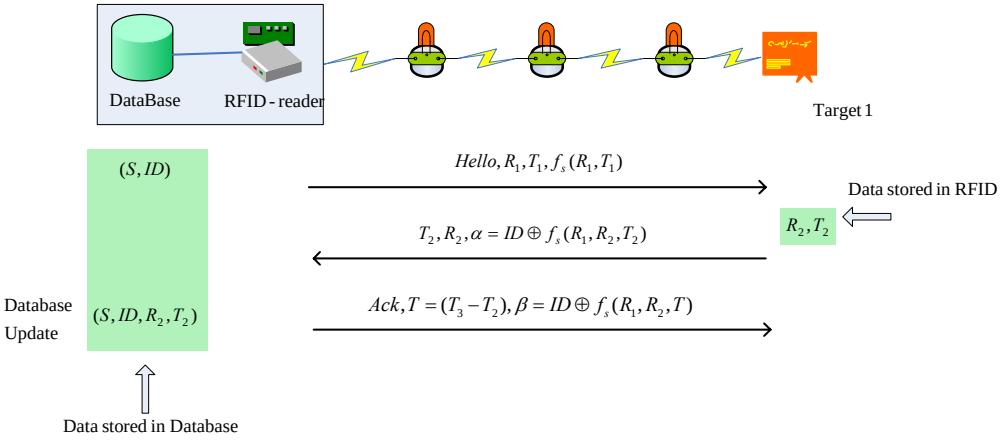


**Figure 2: AMULET architecture.**

*AMULET* can reduce the re-authentication cost when a reader wishes to send commands to an authenticated tag. The reader need not re-authenticate the tag because the database stores the tag's information $(s, ID, R_2, T_2)$. As depicted in Figure 3, the tag can verify future commands by the following two steps:

1. If a new reader queries the database and obtains $(s, ID, R_2, T_2)$ instead of $(s, ID)$, then it recognizes that another reader already authenticated the tag with this *ID*. As a result, it chooses a random number $R_1' \in \{0,1\}^n$, checks for the current time $T_3$, computes the difference in time $T = T_3 - T_2$, and calculates $\beta' = ID \oplus f_s(Cmd', R_1', R_2, T)$. The reader then sends its command $Cmd'$, along with $R_1'$, $T$, and $\beta'$, to the tag.
2. Upon receipt of the $Cmd'$ packet, the tag verifies that $ID = \beta' \oplus f_s(Cmd', R_1', R_2, T)$ before executing $Cmd'$. Otherwise, the tag drops the command.

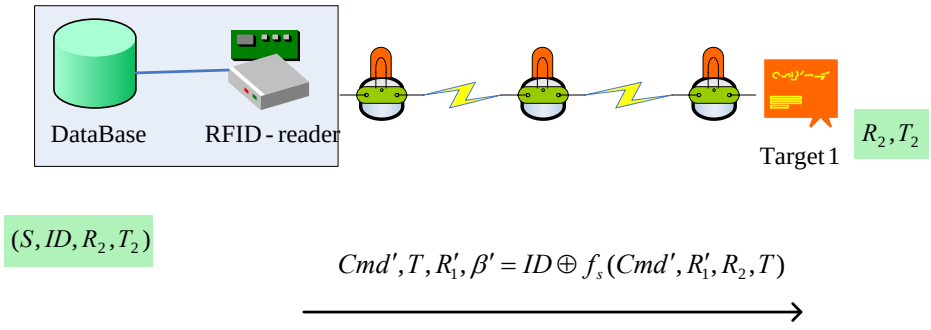


**Figure 3: Commands verification without re-authentication process.**

As previously mentioned, it is harder to eavesdrop on the channel from tag to reader than from the reader to tag; accordingly, *AMULET* provides security against passive eavesdropping on the reader-to-tag link. A common attack to authentication protocols is

man-in-the-middle attack, which *AMULET* naturally resists. Although an attacker can gather $R_1$ and $T_1$ from the reader and $R_2$, $T_2$, and $\alpha = ID \oplus f_s(R_1, R_2, T_2)$ from the tag, it does not possess the secret key *s*, and thus cannot modify or inject its own $\alpha$. Consequently, man-in-the-middle attacks will not succeed against our protocol, and we will formally prove this property in section 4. Furthermore, *AMULET* can defeat replay attacks when tags check that $T$ has not expired and $\beta$ or $\beta'$ is valid for a first-time authentication or re-authentication procedure, respectively.

## 4. ASSART

To preserve data privacy, simply encrypting data prevents attackers from discerning the contents. However, traditional cryptography is not feasible in tags and sensor nodes because of their limited computation capability. Moreover, it is difficult to search encrypted data. To solve this problem, we propose **A S**ecret **S**e**AR**ch pro**T**ocol (*ASSART*), which maintains data in an encrypted form but allows authorized readers to perform searches without disclosing data during transmissions or queries.

In *ASSART*, tags store each characteristic of its associated target as an attribute of the target. We can formally describe a target as $B = (Attr1, Attr2, \cdots AttrN)$, where $N$ is the number of attributes. For example, a tag attached to a book may store the book's ID, title, authors, check-in and check-out time, borrower's ID, etc. Personal attributes like borrower's ID must not be exposed to unauthorized readers or attackers. As shown in Figure 4, *ASSART* involves the following steps:

1. For an attribute $AttrK$, the reader first generates $H^K(s, R_2)$ by iteratively hashing $(s, R_2)$ $K$ times, where K indicates the number of the sequential order of $AttrK$.
2. Next, the reader generates $H^K(EK_i)$ by iteratively hashing $EK_i$ $K$ times.
3. After calculating $f_s(N, R_2, H^K(s, R_2))$, the reader concatenates it with $N$ to form $\lambda = N, f_s(N, R_2, H^K(s, R_2))$.
4. Finally, the reader computes $Attr'K = AttrK \oplus H^K(EK_i) \oplus \lambda$ and overwrites $AttrK$ with $Attr'K$.

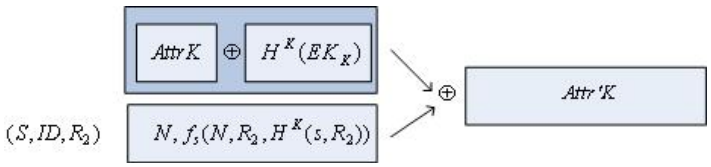


**Figure 4: ASSART operations for attribute *K*.**

Once every attribute is overwritten, attackers will learn nothing from the encrypted data. Since *K* is different for all attributes, each attribute generates a different encrypted attribute value even if some attribute values happen to be the

same. This will keep attributes private. Figure 5 illustrates ASSART's operations.

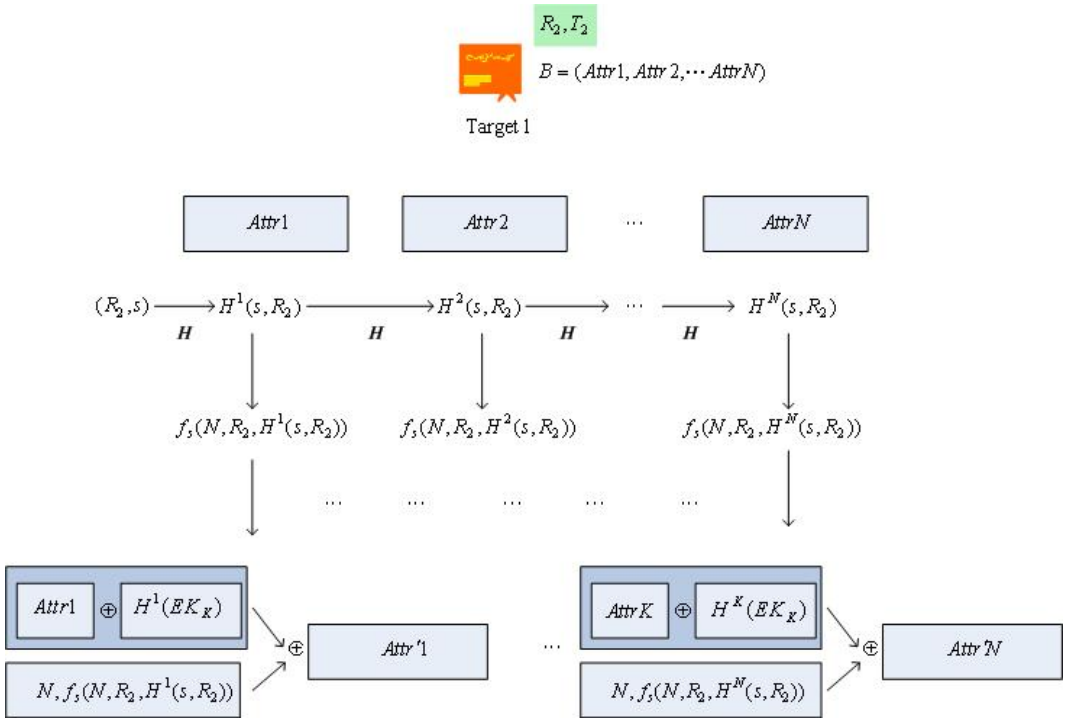


**Figure 5: ASSART operations.**

Authorized readers can inversely-transform $Attr'K$ back to $AttrK$ by computing $AttrK = Attr'K \oplus H^K(EK_i) \oplus N, f_s(N, R_2, H^K(s, R_2))$. Because authorized readers can retrieve $(S, R_2)$ from the database, they can easily calculate $AttrK$ without exposing sensitive and private data during wireless transmission.

A major contribution of *ASSART* is that it ensures the privacy of the remaining attributes in the event that some attributes are compromised. Since $f_s(N, R_2, H^K(s, R_2))$ varies by $K$, $Attr'(K+1) = Attr(K+1) \oplus H^{K+1}(EK_i) \oplus (N, f_s(N, R_2, H^{K+1}(s, R_2)))$ will remain secure even when $f_s(N, R_2, H^K(s, R_2))$ is compromised.

To search for an attribute $AttrK$, the RFID-reader broadcasts an encrypted query $AttrK \oplus H^K(EK_i)$ to all sensor nodes. Next, 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 every value of $K$. The sensor node must calculate an $Attr'K$ for all $K's$ because it does not know the value of $K$. If any sensor node finds a match, it returns $Attr'K$ and $K$ to the RFID-reader. Since data is encrypted, privacy is maintained during the transmission.

## 5. Security Analysis

In this section, we first demonstrate the security of *AMULET* under man-in-the-middle attacks. Second, we provide an analysis that discusses the resources required to break ASSART.

### 5.1 Security of AMULET

We classify man-in-the-middle attacks into three categories: type-1 attack modifies $R_1$ only, type-2 attack modifies $R_2$ only, and type-3 attack modifies $R_1$, $R_2$, and $\alpha$. We will show that these three types of attacks fail against our authentication protocol. Before we begin our proof, we give several definitions below.

**Definition 1:** (Instance) We can formally describe a target by its *ID* and attributes, where $B = (ID_B, Attr1, Attr2, \cdots AttrN)$. An instance $X_B$ is defined as $X_B = (Attr1, Attr2, \cdots, AttrN)$, and a verification function $V_f$ is defined as $V_f(X_B) = \sum_{i=1}^{n} Attri$.

**Definition 2:** (Distinguishable) Two instances of a target are distinguishable if any attribute has different values.

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

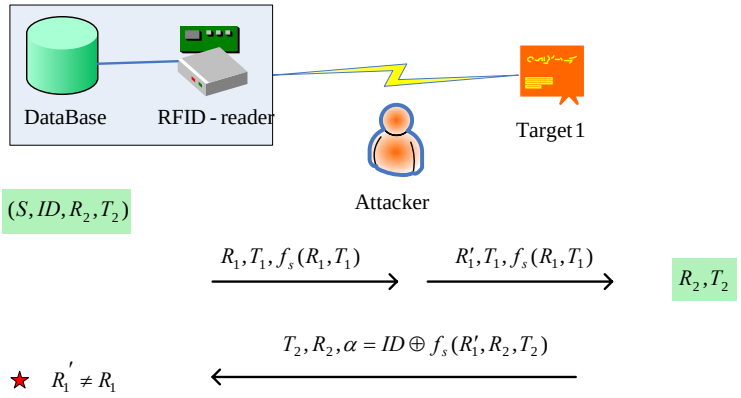


**Figure 6: Type-1 man-in-the-middle attack**

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 sends $R_2$, $T_2$, and $\alpha$ back to the reader. Since $R_1 \neq R_1'$, the reader will find that $f_s(R_1, R_2, T_2) \neq f_s(R_1', R_2, T_2)$. As a result, the readers can prevent type-1 man-in-the-middle attacks.

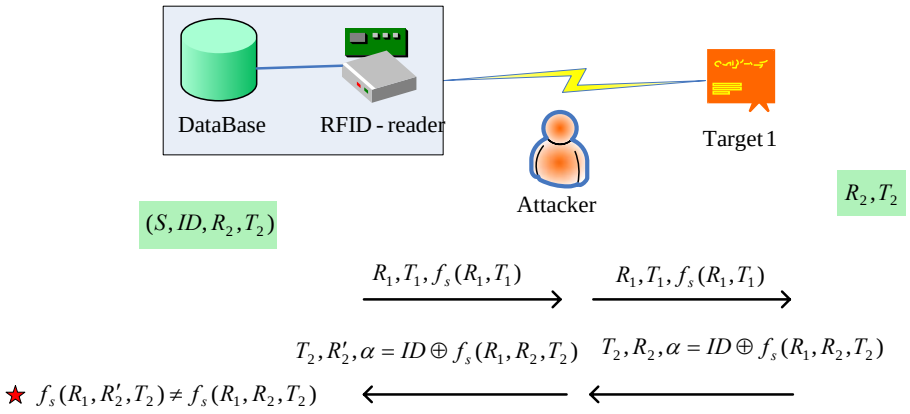


**Figure 7: Type-2 man-in-the-middle attack.**

As depicted in Figure 7, a type-2 attacker eavesdrops on $R_2$, produces a false value $R_2'$, and transmits $R_2$ and $\alpha$ back to the reader. Because $R_2 \neq R_2'$, the reader will find that $f_s(R_1, R_2, T_2) \neq f_s(R_1', R_2, T_2)$, thus thwarting type-2 man-in-the-middle attacks.

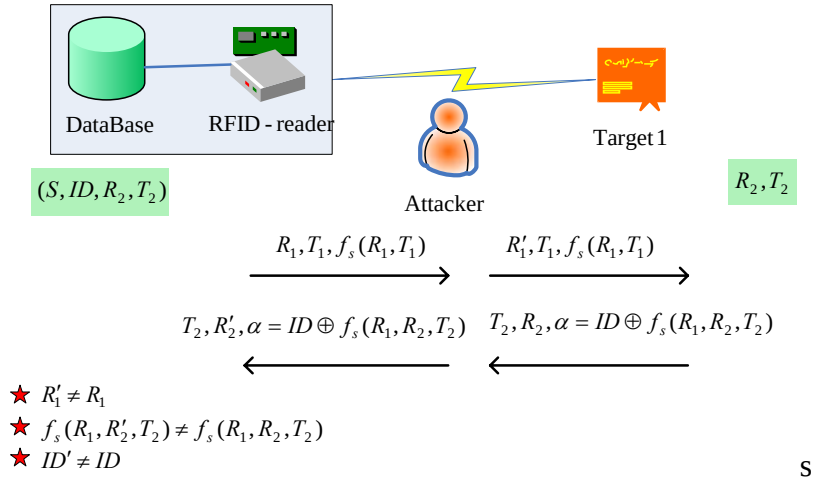


**Figure 8: Type-3 man-in-the-middle attack.**

In Figure 8, a type-3 attacker generates false $R_1'$, $R_2'$, and $\alpha'$ back to the reader and the tag separately. Since $s$ remains secret, the reader will observe that $f_s(R_1, R_2, T_2) \neq f_s(R_1', R_2, T_2)$ and $ID \neq ID'$, causing type-3 man-in-the-middle attacks to fail.

### 5.2 Security of ASSART

We provide a proof of ASSART's security strength in terms of the secrecy of its attributes. By establishing the number of resources required to compromise a system, we can evaluate its security strength. Theorem 1 states that an attacker must have knowledge of both $s$ and $R_2$ to compromise $Attr'K$, where

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

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

**Proof:** Since attackers may extract the values of *N* and *K*, only *s* and $R_2$ must be kept secret. Attackers must know both *s* and $R_2$ to compromise $f_s(N, R_2, H^K(s, R_2))$. Thus, $f_s(N, R_2, H^K(s, R_2))$ is $(s, R_2)$-breakable.

An instance is a collection of all attributes of a tag whose security strength is defined by the number of attributes needed compromise the tag. Thus, as the number of distinguishable attributes increase, the instance will attain a higher security level.

**Theorem 2:** Given an instance of any two attributes *Attr'I*, *Attr'J*, where $I \neq J$, there does not exist a different instance *Attr"I*, *Attr"J*, such that the verification function evaluates to the same value $V_f(Attr'I, Attr'J) = V_f(Attr''I, Attr''J)$.

**Proof:** Let *AttrI*, *AttrJ* be two original attributes such that $I > J$, *Attr'I*, *Attr'J*, be their transformed attributes, and $V_f(Attr'I, Attr'J)$ be the verification of the transformed attributes. We will prove that an attacker cannot generate attributes *Attr"I*, *Attr"J* that satisfies $V_f(Attr'I, Attr'J) = V_f(Attr''I, Attr''J)$.

From equation 1, we know that

$$V_f(Attr'I + Attr'J) =$$

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

An important property of our protocol is that *AttrI* can be used to authenticate *AttrJ* by checking that

$$H^J(s, R_2) = H^{J-I}(H^I(s, R_2))$$

If an attacker generates attributes *Attr"I*, *Attr"J*, $H^I(s, R_2)$ and $H^J(s, R_2)$ can be calculated by the following two equations.

$$Attr''I \oplus AttrI = N, f_s(N, R_2, H^I(s, R_2))$$

$$Attr''J \oplus AttrJ = N, f_s(N, R_2, H^J(s, R_2))$$

Because only authorized readers and tags know *s* and $R_2$, the attacker cannot falsify $H^I(s, R_2)$ and $H^J(s, R_2)$.

The next theorem stipulates that an attacker must compromise all attributes of an instance to deceive readers. If only a portion of the attributes are compromised, the reader can still verify the instance. We will use induction to 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 $V_f(B) = \sum_{i=0}^{n} Attri = Attr'1 + Attr'2 + \cdots + Attr'N$. *B* is *N*-breakable and distinguishable.

**Proof:** Let $B = (Attr1, Attr2, \cdots, AttrN)$ be the original attributes and $B' = (Attr'1, Attr'2, \cdots, Attr'N)$ be the

attributes after transformation.

For $N$ = 2, $B$ is 2-breakable by theorem 2.

Suppose when $N$=$P$, $B$ is $P$-breakable. We want to prove $B$ is $P$-breakable when $N$=$P$+1. Let $B_1 = (Attr1, Attr2, \cdots, AttrN, AttrN+1)$ . From theorem 2, we know that every pair of attributes is distinguishable. Therefore, $AttrN+1$ and $AttrM$ are distinguishable for $M = 1,2,\cdots N$ by verifying $H^{N+1}(s, R_2)$ and $H^1(s, R_2), H^2(s, R_2), \cdots, H^N(s, R_2)$ respectively. Since all $N+1$ attributes are distinguishable, we have shown 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 attribute $AttrK+1$ through equations 5 and 6.

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

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

If both equation 5 and 6 are satisfied, the added attribute $AttrK$ is valid. Otherwise, $AttrK$ is invalid and should be discarded.

## 6. Conclusion

In this paper, we present the *ARIES* architecture to solve the distance limitation problem in RFID applications by utilizing RFID-aware sensor nodes to monitor distant targets. We also propose an authentication protocol, *AMULET*, which mutually authenticates readers and tags. *AMULET* can resist man-in-the-middle attacks and reduce re-authentication overhead. Finally we devise a search protocol, *ASSART*, to perform queries on encrypted data, which prevents the disclosure of information during the transmission or search process. Furthermore, *ASSART* uses a key chain to improve data security. Even if some attributes are compromised, the rest of attributes remain private.

## 7. 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] E. Kusilevitz, and R. Ostrovsky, Replication Is Not Needed: Single Database, Computationally-Private Information Retrieval, In *Proceedings of the 38th Annual Symposium on Foundations of Computer Science*, pp.364-373, 1997.

[5] 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.

[6] F. Dabek, E. Brunskill, M. F. Kaashoek, D. Karger, Building Peer-to-Peer Systems With Chord, A Distributed Lookup Service, *In Proceedings of the. 8th Workshop on Hot Topics in Operating System,* pp.81, May 2001.

[7] Frank, J., Cheeseman, P., and Stutz, J., On The Complexity of Blocks-World Planning, In *Proceedings of Artificial Intelligence*, pp.139—403, 1992.

[8] H.-M. Sun and S.-P. Shieh, An EfficientCconstruction of Perfect Secret Sharing Schemes for Graph-based Access Structures, In *Proceedings of Computers and Mathematics with Applications*, pp.129-135, 1996

[9] H.-M. Sun and S.-P. Shieh, On Dynamic Threshold Schemes, *In Proceedings of Information Processing Letters*, pp.201-206, 1994.

[10] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong, Freenet: A Distributed Anonymous Information Storage and Retrieval System, In *Proceeding of the ICSI Workshop on Design Issues in Anonymity and Unobservability*, pp.311-320, 2000.

[11] 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.

[12] Jonathan Ledlie, Jacob Taylor, Laura Serban, and Margo Seltzer, Self-Organization in Peer-to-Peer Systems, In *Proceedings of 10th SIGOPS European Workshop*, 2002

[13] 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.

[14] 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.

[15] N. Alon, Z. Galil and M. Yung, Efficient dynamic-resharing verifiable secret sharing against mobile adversary, In *Proceedings of European Symposium on Algorithms*, pp.523-537, 1995.

[16] P. Feldman, A practical scheme for non-interactive verifiable secret sharing. In *Proceedings of the 28th IEEE Symposium on Foundations of Computer Science*, pp.427-438, 1987.

[17] Pottie G. J, Wireless Sensor Networks, In *Proceedings of Information Theory Workshop*, pp.139-140, 1998.

[18] Premkumar T. Devanbu and Stuart G. Stubblebine, Stack and Queue Integrity on Hostile Platforms, In *Proceedings of IEEE Transactions on Software Engineering*, pp.100-108, 2002.

[19] 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.

[20] Srisathapornphat, C., Jaikaeo, C., Chien-Chung Shen, Sensor Information Networking Architecture, In *Proceedings of International Workshop on Parallel Processing*, pp.92-95, 2000.

[21] 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.

[22] Y. Gertner, Y. Ishai, and E. Kushilevitz, Protecting Data Privacy in Private Information Retrieval Schemes, In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, pp.151-160, 1998.

[23] Y. Zheng, T. Hardjono and J. Seberry, How to recycle shares in secret sharing schemes, In *Proceedings of Austral Computer Science Communications*, pp.1053-1064, 1992.