# Privacy-Aware Multi-Context RFID Infrastructure using Public Key Cryptography[*]

Selim Volkan Kaya, Erkay Savaş, Albert Levi, Özgür Erçetin

Sabancı University, Istanbul, Turkey
selimvolkan@su.sabanciuniv.edu, {erkays,levi,oercetin}@sabanciuniv.edu

**Abstract.** We propose a novel RFID infrastructure design, which foresees the usage of a single RFID tag within different contexts and for multiple purposes. We show that an infrastructure for multi-purpose RFID tags to be used in different contexts can be implemented in a privacy-preserving manner. We address security attacks such as impersonation, tracking, and replay. We also introduce spatio-temporal attacks as an important threat against privacy. We propose a methodology to thwart or alleviate these kinds of attacks. We develop our multi-context RFID infrastructure relying on usage of public key cryptography (PKC), which presents more scalable solutions in the sense that the backend servers can identify the tags 75 times faster than best symmetric cipher based systems when there are a million tags in the system. We demonstrate that the requirements for PKC are comparable to those for other cryptographic implementations based on symmetric ciphers proposed for RFID use.

**Keywords:** RFID, privacy, security, public key cryptography, spatio-temporal attacks

## 1. Introduction

Remote identification of objects based on radio signals is welcomed with an enthusiastic acceptance in various numbers of applications due to ease of use and efficiency. Compared to previous technologies for object identification such as barcodes and smart cards, RFID is a non-sight of vision technology. The amount and variety of information that can be stored in an RFID tag are unimaginable in the traditional technologies. These features render the use of RFID tags as popular (and inevitable to a great extent) in large and diverse set of applications such as supply chain, toll collection, payment tokens etc. A common characteristic of these RFID-based applications is that the tags are used for a single purpose and in a single context, in the sense that only designated readers can challenge/query the tags. This does not necessarily prevent the other unauthorized readers from participating in privacy-violating activities such as tracking the movements of the tags, hence individuals associated with them.

[*] The work described in this paper is supported by the Scientific and Technological Research Council of Turkey under project number 105E089.

Therefore, usage of RFID in a single context puts certain limitations on the versatility of tags while adding to privacy problems since existence of many tags may provide new opportunities for illegally tracking objects or individuals. For instance, if an object may need to be identified by different readers for different purposes, multiple RFID tags are required for the same object. However, this approach has some drawbacks. Firstly, multiple tags attached to an object increase the cost. Secondly, privacy breaches are more likely to appear since more tags mean more possible privacy vulnerabilities. Thirdly, management of multiple tags is more difficult since all tags need to function properly and securely. And finally, reaching different kinds of data about an object with multiple tags hinders more scalable and practical solutions.

To overcome problems mentioned above, we propose a multi-purpose RFID infrastructure, where one RFID tag can be interrogated by various readers with different motivations. This infrastructure seems to be more adaptive to real life situations since an object is related to multiple parties in some way as a result of cooperative and collaborative structure of the society. For instance, an RFID tag to identify individuals can be queried by different sites for different purposes as illustrated in Figure 1. Police department should be able to identify each person to find out whether that person has a crime record or not. Hospital should be able to identify each person in case of health emergency to learn about medications that person use and her previous health record. In order to decide whether to give visa or not, visa office should be able to identify each person to get information about that person's previous visits and check social statues, bank account and crime records. Security department of the building where that person works needs to identify the person to give her access to the building. We can easily extend this example to tens of different motivations, where identification of each person or objects (e.g. cars) is needed. From the scenarios outlined above, each party with a reader queries RFID tag to retrieve information of interest for the person or object whose identification information is stored in the tag. In this context, there must be rules and limitations that govern what kind of information and in which circumstances this information can be obtained for the individuals.

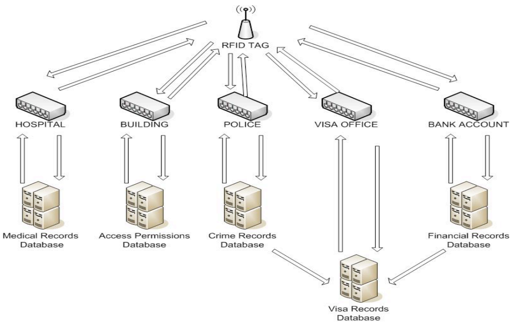


**Fig. 1.** Multi-Context RFID Infrastructure

Nevertheless, increasing the versatility of RFID tags by letting multi-purpose access by different parties emphasizes the importance of privacy. This is because of the fact that RFID tags provide access to a huge amount of information which may result in the compromise of individual privacy if access rights of each party are not determined appropriately and if infrastructure is vulnerable to security attacks. We addressed several security attacks, and showed that these attacks are prevented using our proposed multi-purpose RFID infrastructure. Some of these attacks are *traceability, impersonation of readers and tags, replay attacks, spatio-temporal attacks, and cryptanalytic attacks*.

Public key cryptography (PKC) is a powerful technique which proves to be indispensable for the proposed infrastructure, where the aforementioned attacks for RFID systems can be thwarted without an adverse effect on the scalability and openness of the overall system. However, public key cryptography is known for its excessive need for resources, which do not exist in resource-constrained RFID tags. Nonetheless, public key cryptography can still be considered in RFID applications provided that the PKC algorithms are efficiently realized in hardware with considering power and chip area constraints as ultimate design criteria. Especially, NTRU public key cryptosystem [14] offers great advantages in terms of power, memory requirements, performance results, and security level, and it proves to be a feasible solution for even the Class 2 RFID tags.

## 2. Related Work

Different methods proposed in the literature to provide privacy in RFID systems basically take one of the three main approaches. The first approach is to use hash functions for their low-cost and computational efficiency. Hash lock scheme [1] is the most primitive representative of this approach and it works by locking RFID tag by the time it receives required response from the interrogator. However, the hash lock scheme fails to prevent tracking since the hashed values of tag IDs can be used as metaIDs, which remain constant over time, to trace each tag. To prevent tracking, randomized hash lock scheme [1] is introduced, where the hashed value of tag ID is changed for each read request by concatenating a random value to tag ID before hashing. The main drawback of this scheme is that reader should try all possible tag IDs with the random number, sent with the hash value, to identify the correct tag. Both hash lock and randomized hash lock schemes provide no forward secrecy since if a tag is compromised, all previous communications of that tag could be determined. To prevent tracking and provide forward secrecy, hash chain scheme [2] is introduced. However hash chain scheme requires strict synchronization between tag and reader which makes it vulnerable to de-synchronization attacks [16].

Second approach is to use tree structure [10] to store secrets for each tag. In the tree structure, a tag, which has a unique path to the root, keeps multiple secrets defined by this unique path. In the tree structure, secrets of the interrogated tag could be obtained by using depth-first search. However, that the tree structure causes overlap among secrets of tags means compromising one tag yielding secrets of other tags [4].

Third approach is to use symmetric encryption for challenge-response based authentication defined in [7]. As reported in [9], symmetric encryption is feasible in RFID systems. However, using symmetric keys comes with a price. Compromise of the secret key even in one tag will affect the whole system if every tag uses the same secret to authenticate itself [3]. On the other hand, if each tag uses a different secret, as challenge-response protocol in [10], then the problem of matching a secret with a tag during authentication process becomes an issue since brute force search is needed to search entire space of secrets to find out the owner of the secret [4], that limits the scalability when there are many tags in the system. Therefore, as shown subsequent sections, the PKC offers great advantages in terms of both scalability and security.

## 3. Our Approach

The following sub-sections present a detailed explanation of the multi-context RFID infrastructure, proposed protocol definition, and security analysis of our approach for the most common attacks that can be applied in RFID systems.

### 3.1 Multi-Purpose RFID Infrastructure

As stated in Section 1, one RFID tag can be used in several applications for identification. For that purpose, we propose a decentralized RFID infrastructure consisting of sub-domains. Each sub-domain is responsible for only one purpose and has a trusted backend server, in which information related to that sub-domain is stored. Information of each RFID tag is distributed vertically to each sub-domain. In other words, each sub-domain stores different attribute values corresponding to the same tags. For instance, if we revisit the example given in Section 1, a person has data on her crime, health, social and financial records kept on trusted back-end servers of the corresponding sub-domains. For a reader to be able to query a tag, reader's interest is first determined and then added to the sub-domain corresponding to its interest in advance. A reader can belong in multiple sub-domains as shown in Figure 1. For example, a reader in the visa office may want to access to data kept in financial and crime sub-domains. After being added to a sub-domain, access control of that sub-domain is updated for that new reader. Access control lists for a reader is built based on spatio-temporal constraints as well as roles and identifications.

Decentralized multi-context RFID infrastructure has several additional advantages. First of all, vertical partitioning of the data among sub-domains reduces security risk if one sub-domain is compromised. Secondly, management of the data is easier in decentralized model since data size is reduced for each sub-domain. And finally, access control model is simpler in decentralized model since sub-domains are given only necessary portions of the information and readers join in the sub-domains according to their interests. If a reader requests a query, sub-domain only needs to check location and time constraints in the access control list than determining allowed attribute values and locations for that user.

The access control model is based on three pieces of information: 1) the authenticated ID of the reader, 2) verifiable location of the reader during tag interrogation, and 3) verifiable time of the interrogation. During the interrogation process, the reader should present these three pieces of information to the backend server. The authentication of the reader to the backend sever can be done in a straightforward manner using symmetric or public key cryptography, which is not explicitly shown in the protocol. The reader can obtain the verifiable time and location information in several ways. If the RFID tag is mounted on an object, where the resources are abundant, such as automobiles, this information can be supplied by the tag itself. In such environments, an active tag can be used such that it maintains a built-in clock and can get the location information from a nearby GPS device. If the tag is a passive device and there is no means of getting location information, the infrastructure must provide the verifiable location and time service. If the reader is a wireless device connected to mobile (cellular) network, time and location information of the reader during tag interrogation can be supplemented by the network with certain precision.

Since the backend servers are trusted in our model they share the same public key and private key pair. The tags know the public key of the backend servers. In the protocol definition below, we only showed one backend server for sake of simplicity, and we assume that there is a secured internetwork connecting the backend servers.

### 3.2 Protocol Definition

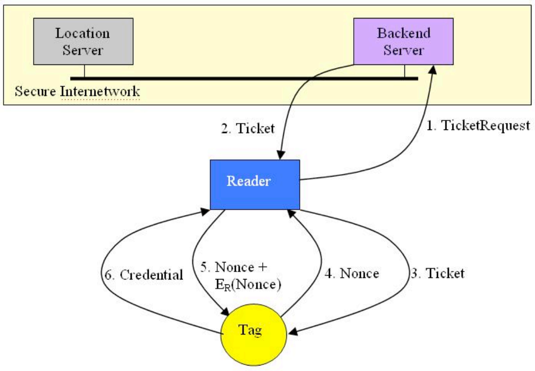
Figure 2 outlines the proposed protocol that shows the communication between tag, reader, and back-end server in this architecture. The numbers in the figure indicate the order of steps. The protocol can be divided into two phases as explained below.

**Phase I:** The reader acquires a ticket from the backend server to query tags in its reading range to obtain data pertaining to them. The ticket is the encryption of reader ID, current time and location of the reader under the secret key known only to the backend server. The backend server obtains the reader location from the location server in a secure way. Since reader is a self-powered device and connected to the network, the location server can locate the reader when asked. The reader can use the same ticket to query different tags for a certain period of time, which is determined by the backend server depending on the reader, its location and the time. Thus, the reader does not have to repeat the first two steps every time it wants to query tags.

**Phase II:** The reader sends the Ticket (message 3) to start the query process. The tag responds by sending a randomly generated “Nonce” (message 4). The Nonce is a challenge for the reader to prove its claimed identity and its location. The reader responds the challenge by sending the encryption of the Nonce (message 5) under its secret (or private) key, known to itself and to the backend server, concatenated with the Nonce. In the final stage of the Phase II, the tag encrypts its ID (TID), the Nonce, $E_R$(Nonce), and the “Ticket” using the public key of the backend server and sends it to the reader. The resulting ciphertext is called as the “Credential”, and used by the reader to access to the data pertaining to the tag in the backend server. Since there are IDs of both tag and reader in the Credential, the backend server can perform selective disclosure of data based on the access rights of the reader, which may vary depending on the time and location.

## 3.3 Security Analysis of the Protocol

In this section, we analyze security of our protocol against spatio-temporal, impersonation, and replay attacks, and tracking.
**Assumptions:** The protocol assumes that there are trusted location servers that can either track the readers or locate them when asked by the backend servers. The protocol uses a public-key encryption scheme to form the credential for the reader in Phase II of the protocol. Although public key cryptography is known to be resource-hungry, in subsequent section we discuss that a certain class of public key cryptosystem can efficiently be used in our scheme. Only tags need to know public key of back-end servers. We assume that there is a dedicated secure and authenticated channel between a reader and backend server, and opponents cannot clone a reader, which is a reasonable assumption since it is always possible to build tamperproof hardware to protect confidential information (e.g. secret keys) in the reader.



Ticket = $E_S$(RID+LOC+TIME) [+: concatenation]
Nonce = random number
Response = Nonce + $E_R$(Nonce)
Credential = $E_{pub}$(Ticket+TID+Nonce+$E_R$(Nonce))

**Fig. 2.** Multi-Context RFID Protocol Definition

**Spatio-temporal Attacks**: A reader can access to data pertaining to a tag only in certain locations and certain times of the day. For instance, a reader in a hospital should not be able to interrogate a tag outside of hospital. It would be a privacy breach if readers can do so since different readers combine information pertaining to a tag. Continuing the hospital example, police reader and hospital reader can combine the information and they both get unauthorized access to private information about individuals. Moreover, access rights of the reader may change depending on the time and its location.

The location of a reader can be measured by location servers using several methods with different precisions. If the reader is connected to cellular network, methods such as radiolocation or triangulation can be used. A secure and efficient method that can be applied in any wireless network uses approximate location estimation technique based on a distance-bounding protocol [17]. Granularity in spatial access rights of a reader is limited by the precision provided by location service. Since the reader may change its position and therefore location measurement becomes invalid, a measurement should have a validity period. When this period expires, the location measurement must be repeated. Therefore, the location information is always stored with the time of the measurement.

The reader uses the location information and measurement time to obtain data pertaining to a tag. The reader needs not see the location information and measurement time, and furthermore the reader must be prevented from counterfeiting them. Therefore, these two pieces of information along with the reader ID is encrypted by backend server's key and given to the reader as a Ticket. Therefore, a malicious reader cannot modify and fabricate a Ticket or pass it to another reader.

A reader with a legitimate ticket queries a tag, which, in turn, embeds the ticket in the Credential, which is sent back to the reader. The reader delivers the credential to the backend server to access the data pertaining to the interrogated tag. The server compares the reader ID, location and measurement time against the access rights of the reader. There may be several cases when the backend server denies access to the reader: 1) reader has no access rights at all, 2) the reader has no rights in the specified location in the ticket, and 3) the measurement time of the location expired. The expiration time of location information is determined depending on how mobile the reader is and known to both the reader and the backend server. For highly mobile readers a relatively short expiration time of the location should be selected. The reader can use the same credential to access data until the expiration date is over. The backend server will deny access if a reader tries to use the Credential beyond the expiration time.

**Impersonation Attacks**: A malicious reader cannot impersonate another reader and get access to information pertaining to a tag, to which it is not allowed, since there is a secure and authenticated channel between the reader and backend server. Our protocol also prevents impersonation of tags since no tag ID is sent in clear, but encrypted within the credential with public key of the backend server. Thus, no malicious tag, in collaboration with a malicious reader, can claim to be an honest tag as tag IDs are known only to the backend servers and to the tags. Note that Tag IDs must be generated in a secure way so that it is not feasible to fabricate one by unauthorized parties.

**Replay Attacks**: The proposed protocol is safeguarded against replay attacks by means of measurement time of location that serves as a timestamp and random nonce values that change in every tag query. A reader, however, can use the same credential repeatedly until the measurement time expires, which is, in fact, one of the aims of the proposed protocol. A reader cannot eavesdrop and use a credential intended for another reader since the credential contains ID of the intended reader. As mentioned before readers maintain secure and authenticated channels with backend server and therefore, they cannot impersonate other readers.

**Traceability**: The fact that a reader or a group of collaborating readers is able to track a tag violates the privacy of the tag. If the tag ID is sent in clear or the tag responds to reader's queries always the same way (e.g. sending the same Nonce or same Credential in messages 4 and 6), then readers can violate the privacy by tracking the tags. However, the tag IDs are always encrypted and both Nonce and Credential change in every transaction in nondeterministic way. Only way that a reader can track a tag is that when it collaborates with the backend server. In our assumptions, the backend server is trusted and when it is compromised all system security is lost.

## 4. Public Key Cryptography versus Secret Key Cryptography in RFID Tags

The foremost motivation of using public key cryptography in RFID-based systems is for better scalability when the privacy is of a concern. Pseudonym based schemes [2], [18], [10], where tags respond with different, random looking pseudonyms at each read, have been received wide acceptance in the research community. Informally speaking, a pseudonym is obtained by encrypting tag ID padded with a random string using a symmetric key algorithm (e.g. AES). For better security, each tag has a separate key rather than a single key used by all the tags in the system. In order to link a pseudonym to a tag ID, one must posses the secret key used in pseudonym generation process. Since the secret key is only known to tag itself and the backend server, and the pseudonym changes at every read in unpredictable manner, readers can link a pseudonym neither to a tag ID nor to any other pseudonym generated by the same tag. Consequently, readers rely on the backend servers to link pseudonyms to tag IDs.

One major drawback with pseudonym based schemes using symmetric key cryptography is that the backend servers have to perform decryption operation with the corresponding key to link a pseudonym to a tag ID. The backend server may have to try all possible keys to decrypt the pseudonym since every tag is stipulated to use a different key for security reasons, resulting in a serious scalability problem as the number of tags increase. Assuming N is the number of tags, the number of symmetric encryption operations to be performed is in the order O(N) in [2], $O(N^{2/3})$ in [18], and O(log N) in [10] and [19]; not to mention the storage requirements. Any information attached to or included in the pseudonym that facilitates the linking operation at backend server before decryption operation would also benefit readers for tracking tags. An obvious remedy for better scalability is to use public key cryptography, where tags generate pseudonyms by encryption with the public key of the backend server. When compared to symmetric key cryptography based systems, the advantage of public key cryptography is self-evident, and can be seen in Table 1.

**Table 1.** Comparison of Proposed Scheme to Symmetric Key Based Schemes

| Scheme | Time Complexity | Storage Complexity |
|---|---|---|
| [2] | O(N) | O(N) |
| [18] | $O(N^{2/3})$ | $O(N^{2/3})$ |
| [10] | O(log N) | O(1) |
| [19] | O(log N) | O(1) |
| Proposed | O(1) | O(1) |

The scheme in [10] fails to scale well due to the fact that it necessitates at least 3 and possibly as many as O(log N) rounds of communication between tag and reader.

The scheme in [19], on the other hand, requires that tag store O(log N) keys and perform O(log N) symmetric key operations, which renders the original scheme impractical in RFID systems. The optimized scheme of [19] reduces tag storage, computation and communication overhead at the cost more computation at the backend server. We substantiate this discussion using concrete figures on a realistic example below.

One other caveat before giving the example is about possible confusion due to Table 1. The comparison in the table would be at least unfair (if not misleading), if we failed to adjust complexity of a symmetric key operation with respect to that of a public key operation; the latter is known to be categorically much more expensive than the former. In what follows, we give a fair comparison of our scheme against symmetric cipher based systems.

In order to compare the performance of NTRU public key cryptosystem against that of a fast symmetric cipher, we run a unoptimized implementation of NTRU decryption operation on a 2.8 GHz Intel P4 machine running Windows XP and used the crypto++ package [20], which is the most widely known cryptographic library. ARC4 algorithm, which is one of the fastest symmetric key algorithms in crypto++ package, turns out to decrypt 68 times faster than NTRU decryption. Although this figure seems to favor the symmetric key cryptography, the NTRU cryptosystem is superior so far as the computation cost at the backend server is concerned. Assuming that there are $2^{20}$ (about a million) tags, number of operations needed in proposed schemes and scheme in [19] is given in Table 2.

**Table 2.** Comparison of Proposed Scheme to Symmetric Key Based Schemes

| **Scheme** | **Number of decryption operations** | **Adjusted number of decryption operations** | **Speedup over [19]** |
|---|---|---|---|
| [19] | 5120 ARC4 decryption | 5120 | 1 |
| Proposed | 1 NTRU decryption | 68 | 75 |

As can be observed in Table 2 using public key cryptography considerably benefits the RFID system as far as the scalability is concerned.

Yet, we still need to show that NTRU encryption operation can be implemented in RFID tags and its implementation in ASIC consumes only comparable resources to those needed by symmetric key systems. To this end, we included comparison results of NTRU cryptosystem implementation against one of the state-of-the-art implementation of AES algorithm intended for RFID tags [9] in Table 3.

**Table 3.** Comparison of NTRU against symmetric cipher in RFID tags

| | **NTRU [5]** | | **AES [9]** |
|---|---|---|---|
| Frequency | 500 KHz | 250 KHz | 100 KHz |
| Power | 20 μW | 10 μW | 12.2 μW |
| Time per encryption operation | 58.45 ms<br>(167 bit block) | 116.9 ms<br>(167 bit block) | 10 ms<br>(128 bit block) |
| Gate Count | 3000 | | 3595 |

| | | |
|---|---|---|
| Feature size | 0.13 μm | 0. 35 μm |

In terms of area and power requirements, the NTRU is, in fact, slightly better than AES. On the other hand, AES is significantly faster than NTRU operation as expected (about 29 times faster at 100 KHz). Nevertheless, the time spent on encryption operation in tag can be masked, if reader interrogates other tags meanwhile. Note also that the comparison in Table 3 is not exactly fair since the security levels of two algorithms, feature sizes used in the implementations, and operating frequencies are different. However, these two implementations represent the state-of-the-art and our aim is only to show that the requirements are comparable. For example, another NTRU implementation with higher security [21] (N=503 stronger than 4096 bit RSA) requires about 3000 gates with feature size 0.35 μm which is the same as [9]. Since it does not provide power requirement analysis, we did not include [21] in our comparisons.

## 5. Discussions

As stated in [12], main advantage of PKC is to use asymmetric keys for reader and tag communication which overcomes the main security vulnerability of symmetric key encryption: usage of one master key for both tag and reader. In symmetric encryption, compromise of an RFID tag and master key stored in that tag means compromise of the whole RFID system which depends on that secret. Another (probably more important in our case) advantage of using PKC in RFID applications is that the proposed scheme provides a more scalable solution since the number of decryption that the backend server has to perform is only one per transaction.

Conservative space and power requirements of NTRU with respect to legacy PKC schemes make it more appropriate for RFID applications [5]. Originally, NTRU is proposed to be implemented in software for RFID tags [13] considering the additional cost of adding cryptographic co-processor. However, software implementation of NTRU may not be suitable for applications, where high performance and less power consumption is needed provided that certain amount of increase in manufacturing cost is tolerable. Since hardware implementation of NTRU is possible with about 3000 gates and less than 20 μW power consumption [5], the additional cost of NTRU hardware to RFID tag is affordable. Additionally, increase in manufacturing costs for NTRU specific hardware will be tolerable in the sense that efficient and secure PKC in RFID tags enables multi-purpose RFID tags, which means one powerful tag replacing several standard tags. In other words, increase in quality of RFID tags allows decrease in quantity of tags.

There are also some disadvantages of NTRU over legacy PKC schemes. First of all, the length of ciphertext can be up to seven times of the plaintext size as a result of NTRU encryption [14]. Expansion of ciphertext means more data to be transmitted from tag to reader in step (6) in the proposed protocol, and hence longer transmission time. The NTRU inventors claim that the message expansion can be avoided if the Credential is formed as two parts, which are sent as separate packets. While the message expansion is still necessary for the first part, we can eliminate it from the second part. Further work is needed to assess the efficiency of this technique and to investigate others to shorten the Credential.

Secondly, NTRU will cause decryption failures which will occur with probability of $2^{-40}$ as stated in [15]. Decryption failure causes limitations on security analysis of NTRU, and attacks based on decryption failure can be performed on NTRU cryptosystem.

A malicious party could produce fraudulent clones of a tag by means of physical replication. We did not propose any solution to cloning attacks within our infrastructure.

The cryptographic key lengths that the state-of-the-art implementations of NTRU primitives [5] use can provide only moderate security level. As the use of RFID tags become more widespread, the key lengths will have to be longer in the near future. For example, Another NTRU implementation with higher security [21], where N = 503 (stronger than 4096 bit RSA), requires about 3000 gates. The only reason we did not include this particular implementation is that [21] does not provide power requirements, which is essential in our feasibility analysis.

## 6. Conclusion

We proposed a privacy-aware multi-context RFID infrastructure that employs public key cryptography (PKC). In this infrastructure, different readers can interrogate RFID tags for different purposes. It is not possible for the readers to track RFID tags; therefore their privacy is preserved. During interrogation, tags encrypt their IDs with the public key of the backend server, which performs only one decryption to access the ID of the interrogated tag. In symmetric cipher-based schemes, the backend server has to try many symmetric keys since it cannot know the ID beforehand and choose the corresponding symmetric key. Therefore, employing PKC makes the proposed scheme more scalable compared to other symmetric cipher-based schemes.

We analyzed feasibility of PKC for our protocol. We found that NTRU cryptosystem is suitable PKC in the proposed protocol with power requirement of no more than 20 μW, chip area of about 3300 gates. Analysis of our protocol in terms of user and system memory requirements, and total execution time for NTRU was not given in this study due to space limitations. Further information can be obtained from the extended version of our study in [22].

We also addressed several security attacks such as impersonation, tracking, replay attacks and we showed that the protocol is secure against these attacks. In addition, we introduced a novel type of attack, for which we coined the term spatio-temporal attacks. In this type of attack, the malicious readers can try to interrogate tags beyond their authorized interrogation area and time interval. We showed that ordinary readers or group of readers cannot mount spatio-temporal attacks since special equipment with considerable resources is necessary to mount them.

# 7. References

1. Weis, S., Sarma, S., Rivest, R., Engels, D.: Security and privacy aspects of low-cost radio frequency identification systems.: In SPC'03, LNCS, Vol.2802. Springer-Verlag (2003) 454–469
2. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to Privacy-friendly Tags. In RFID Privacy Workshop. MIT (2003)
3. Menezes, A.J., van Oorschot, P.C., Vanstone, S. A.: Handbook of Applied Cryptography. CRC Press (1997)
4. Avoine, G., Dysli, E., Oechslin, P.: Reducing time complexity in RFID systems.: In SAC'05. LNCS, Vol. 3897. Springer-Verlag (2006) 291 – 306
5. Gaubatz, G., Kaps, J.P., Öztürk, E., Sunar, B.: State of the Art in Ultra-Low Power Public Key Cryptography for Wireless Sensor Networks.: In PerSec'05, Kauai Island, Hawaii (2005)
6. Hoffstein, J., Silverman, J., Whyte, W.: NTRU report 012, version 2. estimated breaking times for NTRU lattices, Technical Report 12, NTRU Cryptosystems, Inc. (2003)
7. National Institute of Standards and Technology (NIST). FIPS-197: Advanced Encryption Standard, November (2001). Available online at http://www.itl.nist.gov/fipspubs/
8. Fujistsu web site, 2006. Referenced 2006 at http://www.fujitsu.com/us/services/edevices/microelectronics/memory/fram/
9. Feldhofer, M., Dominikus, S., Wolkerstorfer, J.: Strong authentication for RFID systems using the AES algorithm.: In CHES'04. LNCS, Vol. 3156. Springer-Verlag (2004) 357–370
10. Molnar, D., Wagner, D.: Privacy and security in library RFID: Issues, practices, and architectures.: In CCS'04. ACM Press, Washington, DC, USA (2004) 210–219
11. Juels, A., Weis, S.: Authenticating pervasive devices with human protocols. In CRYPTO'05. LNCS, Vol. 3621. Springer-Verlag (2005) 293–308.
12. NTRU RFID data sheet, 2006. http://www.ntru.com/products/NtruRFID.pdf
13. NTRU RFID white paper, 2006. www.ntru.com/products/RFID_White_paper_FNL.pdf
14. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A Ring-Based Public Key Cryptosystem.: Algorithmic Number Theory (ANTS III). LNCS, Vol. 1423. Springer-Verlag, Berlin (1998) 267-288
15. Graham, N.H., Nguyen, P., Pointcheval, D., Proos, J., Silverman, J. H., Singer, A., Whyte, W.: The Impact of Decryption Failures on the Security of NTRU Encryption. In CRYPTO'03, Santa Barbara, USA (2003)
16. Dimitriou, T.: A Lightweight RFID Protocol to protect against traceability and cloning attacks. In SecureComm'05 59 – 66
17. Capkun, S., Hubaux, J.P.: Secure positioning of wireless devices with application to sensor networks. In INFOCOM'05
18. Avoine, G., Oeschlin, P.: A Scalable Protocol for RFID Pseudonyms. In Persec (2004)
19. Molnar, D., Soppera, A., Wagner, D.: A Scalable, Delegatable Pseudonym Protocol Enabling Ownership Transfer of RFID Tags. Workshop on RFID and Light-Weight Crypto, July 14-15, Graz, Austria (2005)
20. Dai W.: Crypto++, a Free C++ Library for Cryptography. http://www.eskimo.com/~weidai (2004)
21. O'Rourke, C., Sunar, B.: Achieving NTRU with Montgomery Multiplication. In IEEE Trans. on Comp., vol. 52, No. 4, April (2003)
22. Kaya, S. V., Savas, E., Levi, A., Ercetin, O.: Privacy-Aware Multi-Context RFID Infrastructure using Public Key Cryptography. http://students.sabanciuniv.edu/~selimvolkan/MultiContext_RFID_Framework.pdf (November 19, 2006)