A Hash-Based RFID Security Protocol

[1] Subodh S. Bhoite, [2] Prof. Pawar Sanjay Shamrao

[1] *Electronics & Telecommunication, Shivaji University*
1086, A, Ambai Tank Road, Kaizen Park, Kolhapur, India
[1] bhoitess@gmail.com
[2] pawarsanjay2@rediffmail.com

Abstract— **RFID (Radio Frequency Identification) tags are small, wireless electronic devices that help to identify objects and people. Privacy protection and integrity assurance become rather crucial in the RFID systems, because these RFID tags may have a wide transmission range, making them subject to unauthorized scanning by malicious readers and various other attacks. Hence, Ha et al. proposed an RFID protocol and proved that their protocol can provide the forward privacy service. However, in this paper, it is shown that an attacker can track a target tag by observing unsuccessful previous session of the tag. That is, Ha et al.'s RFID protocol fails to provide the forward privacy protection as claimed. Therefore, to overcome the privacy weaknesses of Ha et al.'s RFID protocol, an RFID protocol based on the cryptographic hash functions is proposed. Moreover, the proposed RFID protocol is evaluated according to both the privacy attribute and the implementation performance.**

Keywords— **Put RFID protocol, RFID tag, forward privacy, Cryptographic hash function, security**

I. INTRODUCTION

RFID (Radio Frequency Identification) is a technology that uses small, wireless, low-power RFID tags to automatically identify and track physical objects. An RFID system consists of the tags, one or more readers, and a backend database. For simplicity, the reader and the back-end database can be treated as a single entity, i.e., the reader, because the channel between the reader and the back-end database is assumed to be secure. The tags can be as small as a grain of sand and cost just pennies apiece. Tags are tuned to a particular frequency, and each tag has a unique ID number. When a tag receives its signal from the reader, the tag sends its ID information in response. The distance at which they can receive and broadcast a receivable signal. Usually varies from roughly five centimeters at the least powerful end to several meters at the most powerful end. Typical uses of tags include toll plaza payments, transit system fare cards, stock or inventory labels, passports, and identity cards, etc.

In order to provide a better support for the RFID security applications, a variety of approaches have been proposed to protect the user privacy. Juels [1] surveyed several well known privacy-restoring approaches for the tag use. The solutions on RFID privacy problems can be roughly classified into the physical techniques and the protocol based techniques. The ideas underlying physical technologies include "disabling a tag" [2], "shielding a tag to block its access by a reader" [3], and "redesigning a tag" [4], etc. The idea among the protocol-based techniques is that every tag exchanges messages with the reader through a specification of the RFID security protocol. Most of the research for the protocol-based techniques focuses on two directions: one is to construct RFID security protocols [5] - [10] that are compatible with the constraints of tags; the other is to define privacy models [11]-[15] for the RFID systems. This paper addresses how to strengthen the RFID security protocol for the privacy protection.

Informally, it is widely believed that an RFID security protocol is a privacy protocol, if an attacker, basing on limits. The sessions of the security protocol, cannot distinguish whether he has seen the same tag twice from whether he has seen two different tags. A stronger concept is the so called forward privacy, i.e., corrupting a tag does not link it to its past session. Ha *et al.* [6] proposed an RFID security protocol using the cryptographic hash function and proved that their protocol can provide the forward privacy service. However, in this paper, it is shown that an attacker can track a target tag by observing unsuccessful previous session of the tag. That is, Ha *et al.*'s RFID protocol fails to provide the protection of the forward privacy as claimed. Therefore, an RFID security protocol based on cryptographic hash functions is proposed to overcome the privacy weaknesses of Ha *et al.*'s RFID protocol. Moreover, the proposed RFID protocol is evaluated according to both the privacy attribute and the implementation performance.

II. HASH FUNCTION

A function that maps a message of any length into a fixed-length hash value, which serves as the authenticator. A variation on the message authentication code is the one-way hash function. As with the message authentication code, a hash function [16] accepts a variable-size message M as input and produces a fixed-size output, referred to as a hash code H (M). The hash code is also referred to as a message digest or hash value. The hash code is a function of all the bits of the message and provides an error-detection capability: A change to any bit or bits in the message results in a change to the hash code. Fig.1. illustrates a ways in which a hash code can be used to provide message authentication, as follows:

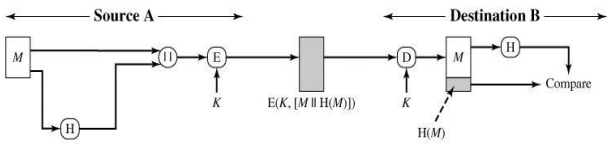

Fig.1 General Hash Code Use

The message plus concatenated hash code is encrypted using symmetric encryption. This is identical in structure to

the internal error control strategy shown in Fig. 1. The same line of reasoning applies: Because only A and B share the secret key, the message must have come from A and has not been altered. The hash code provides the structure or redundancy required to achieve authentication. Because encryption is applied to the entire message plus hash code, confidentiality is also provided. A hash value h is generated by a function H of the form h = H (M)

Where M is a variable-length message and H (M) is the fixed-length hash value. The hash value is appended to the message at the source at a time when the message is assumed or known to be correct. The receiver authenticates that message by re-computing the hash value. Because the hash function itself is not considered to be secret, some means is required to protect the hash value.

III. General Structure of Secure Hash Code

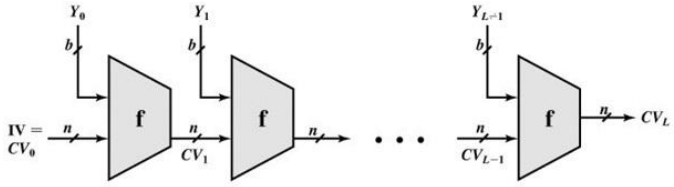

Fig.2. General Structure of Secure Hash Code

Where,

IV = Initial Value L = Number of input block
CVi = Changing variable n = Length of hash code
Yi = ith input block b = Length of input block
f = Compression algorithm

The hash algorithm involves repeated use of a compression function, 'f' that takes two inputs (an n-bit input from the previous step, called the chaining variable, and a b bit block) and produces an n-bit output. At the start of hashing, the chaining variable has an initial value that is specified as part of the algorithm. The final value of the chaining variable is the hash value. Often, b > n hence the term compression. The hash function can be summarized as follows:

CV0 = IV = Initial n-bit value

CVi = f(CVi1, Yi1) $1 \leq i \leq L$

H (M) = CVL

Where the input to the hash function is a message M consisting of the blocks Yo, Y1,...., YL1.

The motivation for this iterative structure stems from the observation by Merkle and Damgard that if the compression functions is collision resistant, then so is the resultant iterated hash function. Therefore, the structure can be used to produce a secure hash function to operate on a message of any length. The problem of designing a secure hash function reduces to that of designing a collision-resistant compression function that operates on inputs of some fixed size.

Cryptanalysis of hash functions focuses on the internal structure of f and is based on attempts to find efficient techniques for producing collisions for a single execution of f. Once that is done, the attack must take into account the fixed value of IV. The attack on f depends on exploiting its internal structure. Typically, as with symmetric block ciphers, f consists of a series of rounds of processing, so that the attack involves analysis of the pattern of bit changes from round to round.

Keep in mind that for any hash function there must exist collisions, because we are mapping a message of length at least equal to twice the block size b (because we must append a length field) into a hash code of length n, where $b \geq n$. What is required is that it is computationally infeasible to find collisions.

IV. Some Notations In RFID Security Protocol

To simplify the discussions of the RFID security protocols, the notations used throughout the paper can be summarized as follows:

R denotes the reader.

T denotes the tag.

ID denotes the current identity of *T*, where the *ID* is a k-bit number.

HID denotes the hashed value of *ID*, where the *HID* is a k-bit number.

PID denotes the previous identity of *T* used in previous session, where the PID is a k-bit number.

rR denotes random number generated by *R*.

rT denotes random number generated by *T*.

Query denotes the request generated by *R*.

SYNC is used to check whether both *T* and *R* succeeded in updating *ID* simultaneously or not. Here, the *SYNC* is defined as a 1-bit number.

H(), H1() represent the secure cryptographic hash functions, which are the computationally efficient functions mapping binary strings of arbitrary length to binary strings of fixed k-bit length, i.e., H: {0, 1}*→{0, 1}k and H1: {0, 1}*→{0, 1}k.

LT(m) represents the left half of the input message m.

RT(m) represents the right half of the input message m.

|| represents the concatenation of two inputs.

?= represents the comparison of two inputs justified.

V. Problem Model For RFID Security Protocol

In general, an RFID system S comprises a fixed tag group TS= {T1, T2, …, Tn} and a reader R as the entities, i.e., S={TS, R}. The reader R has the authentication information for TS, such as ID, session identifier, and state value, etc. The state value is defined as:

Before the RFID security protocol is run for the first time, an initialization phase occurs in both T_l and *R*, where l=1, 2, …,*n*. That is, each $T_l \in TS$ runs a random algorithm $G(1^k)$ to generate and save the secret identity ID_l, and *R* also secretly saves these values in a database field. The state values are perhaps also initialized in both T_l and *R*, where l=1, 2... *n*.

Explanation, Herein, we assume that the RFID security protocol is applied to the single-reader RFID system. That is, a reader *R* owns a group of T_l, where l=1, 2,..., *n*. However, the RFID security protocol can also be easily adapted to the cross-reader RFID system, where several readers may own several groups of tags. Note that the size of the authentication information maintained by each reader cannot be

exponentially increased under the cross-reader setting, because the number of tags in whole RFID system should be polynomially increased only.

VI. Review Of HA ET AL.'S RFID Protocol

In Ha *et al.*'s RFID protocol [6], *R* manages *ID*, *HID*, and *PID* for each *T* in *R*'s field. According to the state of the *T* in the previous session, *R* looks for *ID* in the current session or *PID* in the previous session to identify the corresponding *T*. After authenticating *T*, *R* updates the *ID* of the *T* to achieve the protection of the forward privacy for *T*. Ha *et al.*'s RFID protocol can be briefly shown in Fig. 1. A step-by-step description of Ha *et al.*'s RFID protocol is given below.

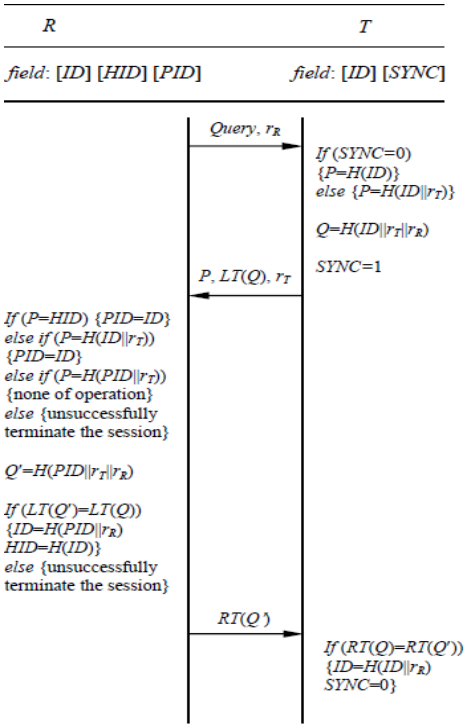

Fig.3 Ha et al.'s RFID protocol.

Step 1. *R* generates a random number *rR* and broadcasts it to *T* with a *Query*.

Step 2. Upon receiving the *Query* and *rR*, *T* generates a random number *rT* and computes the parameter *P* differently according to the state of *SYNC*. If *SYNC*=0, then *P*=*H*(*ID*), otherwise *P*=*H*(*ID*||*rT*). It then computes *Q*=*H*(*ID*||*rT*||*rR*) and sets *SYNC* as 1. And then, *T* sends *P*, *LT*(*Q*), and *rT* to *R* in response to the *Query*.

Step 3. Upon receiving the *P*, *LT*(*Q*), and *rT*, *R* first compares the received *P*=*H*(*ID*) with the *HID* value in the database. If the values match, *R* regards the *ID* as the identity of *T* requesting authentication and sets *PID*=*ID*. The above case takes place, if the previous session is closed normally. If *R* cannot find the *HID* in the first searching case, it then computes *H*(*ID*||*rT*) with the received *rT* and compares it with *P*. If the tag's response message is blocked in the previous session, that is, *SYNC* =1 and two *ID*s in the reader and the tag are not updated, then *R* finds a match with the *ID* of *T* in the second searching case. Here, *R* also need set *PID*=*ID*. However, if *R* cannot find the *ID* of the tag in the above two cases, it then computes *H*(*PID*||*rT*) and compares it with *P*. *R* finds a match with the *PID* of *T*, when *R*'s last messages were blocked in the previous session. If *R* cannot find the identity of *T* in the above three cases, *R* halts the searching of the identity and unsuccessfully terminates the authentication session. If *R* finds the *ID* or *PID* in one of the three searching cases, then it computes *Q'*=*H*(*PID*||*rT*||*rR*) and verifies *LT*(*Q'*)?=*LT*(*Q*). If it is correct, *R* successfully authenticates *T*, transmits *RT*(*Q'*) to *T*, and updates *ID*=*H*(*PID*||*rR*) and *HID*=*H*(*ID*) for the next session.

Step 4. Upon receiving the *RT*(*Q'*), *T* verifies *RT*(*Q'*)?= *RT*(*Q*). If it is correct, *T* successfully authenticates *R*, updates the identity as *ID*=*H*(*ID*||*rR*), and sets the *SYNC* value to 0.

Explanation. Intuitively, the respective computation and verification of the values *H*(*ID*||*rT*||*rR*) and *H*(*PID*||*rT*||*rR*) can achieve the mutual authentication between the legitimate *T* and the legitimate *R*, because the correct *ID* and *PID* are merely shared by them in Ha *et al.*'s RFID protocol. Any attacker without the *ID* or *PID* cannot fabricate the valid values *H*(*ID*||*rT*||*rR*) or *H*(*PID*||*rT*||*rR*) to pass the authentication process due to the cryptographic property of the hash function *H*().

VII. Privacy Weaknesses On HA ET AL.'S RFID Protocol

A. Forward Privacy Weakness

Based on Ha *et al.*'s design idea, their RFID security protocol depends on the update of the *ID* in the *T* during each session to provide the forward privacy service. Even if the attacker can obtain the current *ID* of the *T*, he cannot track the *T* by using the previous session data *Query*, *rR*, *P*, *LT*(*Q*), *rT*, and *RT*(*Q'*). The reason is that the values *P*, *LT*(*Q*), and *RT*(*Q'*) in the previous session are computed by the previous corresponding *ID*, *rT*, and *rR* of the *T*. In order to derive the previous *ID* from the current *ID*, the attacker actually needs to find the preimage *x* that satisfies *H*(*x*)=*y* for a given *y*. However, according to the cryptographic property of the hash function *H*(), it is computationally infeasible to find any input *x*, which hashes to that output *y*.

To provide the protection of the forward privacy, it is an interesting idea to update the *ID* of the *T* in each session. However, Ha *et al.*'s RFID protocol still fails to correctly achieve the forward privacy service. We can outline the following attack called as forward privacy attack-I to undermine the forward privacy protection.

Step 1-I. An attacker passively observes an unsuccessful session run of a target *T*. Alternatively, the attacker actively prevents a target *T* from normally finishing a session run.

Step 2-I. The attacker can link the target *T* to the session run by disclosing the state value *SYNC* of each *T* and verifying whether *SYNC?*=1 in the possible tag group. According to Ha *et al.*'s RFID protocol, the *SYNC* field is set to 0 for an unsuccessful session and 1 for a successful session. Note that the attacker using the practical side channel attacks [17]-[20] may easily obtain the secret information stored in all possible *T*s, i.e., the *ID*s and the *SYNC*s. Therefore, the above attack on

Ha *et al.*'s RFID protocol shows that it is possible to track the previous session of the *T* when the corruption of the *T* is allowed. It means that Ha *et al.*'s RFID protocol violates the definition of the forward privacy. We can give two examples to further explain this attack.

Example 1. Assume that a proximity card containing a tag *T* can be used to open a door. Assume that such RFID system includes Ha *et al.*'s RFID protocol to provide forward privacy protection. An attacker determine whether the tag runs an unsuccessful session, e.g., by observing whether a door is open or not at a tag transit and deducing whether the session run is successful or not. Now, if several possible proximity cards are obtained by the attacker, he can identify the proximity card with the corresponding *T*. The detailed way is to link the *T* with its previous session by checking whether *SYNC?*=1.

VIII. PROPOSED RFID PROTOCOL

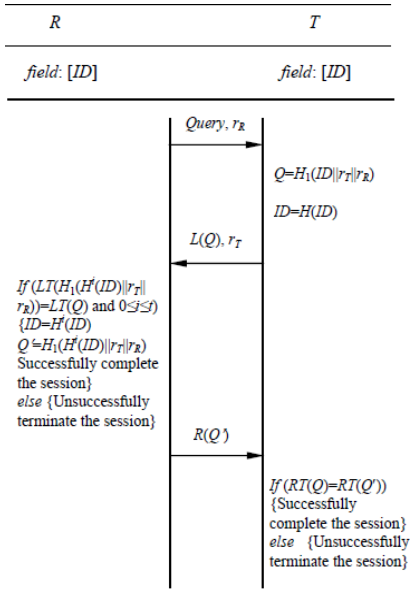

Fig.4. Proposed RFID protocol.

A. *Description of Proposed RFID Protocol*

Let *t* be a fixed constant, e.g., *t*=1000 or 10000, defining the number of the unsuccessful session runs to be allowed for *T*. Let $Hi(x)$ denote iterating the cryptographic hash function *I* times with the input *x*. For example, $H3(x)=H(H(H(x)))$ and $H0(x)=x$. The proposed RFID protocol can be described as follows.

Step 1. *R* generates a random number *rR* and broadcasts it to *T* with a *Query*.

Step 2. Upon receiving the *Query* and *rR*, *T* generates a random number *rT*, computes $Q=H1(ID||rT||rR)$, and updates $ID=H(ID)$. Then, *T* sends $LT(Q)$ and *rT* to *R* in response to the *Query*.

Step 3. Upon receiving the $LT(Q)$ and *rT*, *R* computes $Q=H1(Hi(ID)||rT||rR)$ satisfying $0\le i\le t$ and verifies $LT(Q')?=LT(Q)$. If *R* finds such *ID* value saved in the database, *R* successfully authenticates *T*, and then transmits $RT(Q')$ to *T* and updates $ID=Hi(ID)$ for the next session. Otherwise, *R* unsuccessfully terminates the session.

Step 4. Upon receiving the $RT(Q')$, *T* verifies $RT(Q')?=RT(Q)$. If it is correct, *T* successfully authenticates *R*, otherwise, *T* unsuccessfully terminates the session. Fig. 2 depicts the proposed RFID protocol.

Explanation. Clearly, the proposed RFID protocol achieves the mutual authentication by the computation and verification of the values *Q* and *Q*・ in the respective *T* side and *R* side.

B. Privacy Analysis

Since the state parameter *SYNC* of the *T* reveals its state of the previous session, Ha *et al.*'s RFID protocol cannot provide the protection of the forward privacy, when the *SYNC* stored in the *T* is compromised. Actually, the *ID* of the *T* may also undermine the forward privacy protection, when the session run of the *T* is unsuccessfully terminated in Ha *et al.*'s RFID protocol. The reason is that the *ID* of the *T* is not updated in this case. In the proposed RFID protocol, each *T* keeps its own *ID*, but the state parameter *SYNC* is not used. Moreover, the *ID* of the *T* is updated in each session run of the proposed RFID protocol. It can be intuitively seen that the proposed RFID protocol should correctly provide the forward privacy service. That is, even if the attacker can compromise the *ID* of the *T*, he cannot track the identity of the *T* by linking its previous session, because the computation of the pre-image is intractable for the secure cryptographic hash function.

IX. IMPLEMENTATION PERFORMANCE

In this subsection, the efficiency of the proposed RFID protocol is evaluated by comparing it with the improved OSK RFID protocol [12] and Ha *et al.*'s RFID protocol [7]. The reason is that these RFID security protocols use similar cryptographic tools and attempt to achieve the similar privacy and authentication goals. In the RFID system, the tag is usually treated as the resource-constrained electronic device, but the reader is always powerful enough to provide the security service. Hence, for each RFID security protocol, the estimation of the implementation costs is merely focused on the tag side. As shown in Table I, a brief efficiency comparison is given in the storage, communication, and computation aspects of the three RFID security protocols. Here, assume that the random number and the output of the secure cryptographic hash functions are all 160 bits. The communication cost and the computation cost are estimated from a normal session of the three RFID security protocols. Without loss of accuracy, it is reasonable to merely consider the expensive cryptographic hash computation but do not take count of other inexpensive operations such as concatenation operation and comparison operation. It is shown that Ha *et al.*'s RFID protocol is less efficient than the proposed RFID protocol. But, compared with Ha *et al.*'s RFID protocol and the proposed RFID protocol, the improved OSK RFID protocol is more efficient in communication. However, it needs to point out that Ha *et al.*'s RFID protocol and the proposed RFID protocol provide the mutual authentication

service, but the improved OSK RFID protocol only provides the unilateral authentication service.

TABLE I
EFFICIENCY COMPARISON IN TAG SIDE

Protocol	Storage Cost	Communication Cost	Computation Cost
Ha et al.'s RFID protocol	161 bits	640 bits	3H
Proposed RFID protocol	160 bits	480 bits	2H

X. CONCLUSION

For RFID protocol designers, it is always a hard task to balance the security requirement, the functionality requirement, and the efficiency requirement. So, many design mechanisms appear well motivated at first glance but have unintended consequences. The lessons learned with respect to Ha *et al.*'s RFID protocol demonstrate this point. Ha *et al.*'s RFID protocol may employ the *SYNC* parameter to improve the computational efficiency in the reader side, because the reader *R* firstly compares the received *P* with the *HID* values to identify the tag *T* in the Step 3 of Ha *et al.*'s RFID protocol. None of cryptographic hash computations is required in this searching case. More importantly, this is a general case, when the previous session of the *T* is finished normally. But, it is shown that the *SYNC* parameter simultaneously threatens the forward privacy of Ha *et al.*'s RFID protocol. Hence, an RFID protocol is proposed to fix the privacy vulnerabilities in Ha *et al.*'s RFID protocol. Although the proposed RFID protocol does not satisfy the full spectrum of real-world needs, the research results will promote the designs of the RFID security protocols with strong privacy protection, which are suitable for different security applications.

ACKNOWLEDGMENT

The Acknowledgements give an opportunity for the authors to thank people who helped with the study or preparation of the paper. This includes anyone who provided technical assistance to the authors, took care of the animals, or provided reagents or equipment. The authors may want to thank anyone who had helpful discussions with them or contributed less tangible concepts. This is also where the authors may indicate that the results of this study were presented in another form, such as a poster or abstract or at a symposium. Finally, practical matters such as present addresses of contributing authors are listed here.

REFERENCES

[1] A. Juels, "RFID security and privacy: a research survey," IEEE Journal on Selected Areas in Communications, vol. 24, no. 2, pp. 381–394, Feb. 2006.

[2] A. Weis, S. E. Sarma, R. L. Rivest, and D. W. Engles, "Security and privacy aspects of low-cost radio frequency identification systems," in: D. Hutter, G. Müller, W. Stephan, and M. Ullmann, editors, Proceedings of 1st International Conference on Security in Pervasive Computing, LNCS 2802, Springer–Verlag, pp. 201–212, 2004.

[3] A. Juels, R. L. Rivest, and M. Szydlo, "The blocker tag: selective blocking of RFID tags for consumer privacy," in: Proceedings of 10th ACM Conference on Computer and Communications Security-ACM CCS' 03, pp. 103–111, ACM, 2003.

[4] H. Holtzman, S. H. Lee, and D. Shen, "OpenTag: privacy protection for RFID," *IEEE Pervasive Computing*, vol. 8, no. 2, pp. 71–77, Apr.–Jun. 2009.

[5] H. Y. Chien, "SASI: a new ultra lightweight RFID authentication protocol providing strong authentication and strong integrity," *IEEE Transactions on Dependable and Secure Computing*, vol. 4, no. 4, pp. 337–340, Oct./Dec. 2007.

[6] J. H. Ha, S. J. Moon, J. Y. Zhou, and J. C. Ha, "A new formal proof model for RFID location privacy," in: S. Jajodia and J. Lopez, editors, Proceedings of 13th European Symposium on Research in Computer Security–ESORICS'08, LNCS 5283, Springer–Verlag, pp. 267–281, 2008.

[7] S. Piramuthu, "Lightweight cryptographic authentication in passive RFID-tagged systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 38, no. 3, pp. 360– 376, May 2008.

[8] D. Z. Sun, J. P. Huai, J. Z. Sun, J. W. Zhang, and Z. Y. Feng, "A new design of wearable token system for mobile device security," *IEEE Transactions on Consumer Electronics*, vol. 54, no. 4, pp. 1784–1789, Nov. 2008.

[9] J. Katz, J. S. Shin, and A. Smith, "Parallel and concurrent security of the HB and HB+ protocols," *Journal of Cryptology*, vol. 23, no. 3, pp. 402– 421, Jul. 2010

[10] P. D'Arco and A. D. Santis, "On ultra lightweight RFID authentication protocols," *IEEE Transactions on Dependable and Secure Computing*, vol. 8, no. 4, pp. 548–563, Jul./Aug. 2011.

[11] S. Vaudenay, "On privacy models for RFID," in: K. Kurosawa, editor, Proceedings of 13th International Conference on the Theory and Application of Cryptology and Information Security-ASIACRYPT' 07, LNCS 4833, Springer-Verlag, pp. 68–87, 2007.

[12] C. S. Ma, Y. J. Li, R. H. Deng, and T. Y. Li, "RFID privacy: relation between two notions, minimal condition, and efficient construction," in: Proceedings of 16th ACM Conference on Computer and Communications Security-CCS' 09, pp. 54–65, ACM, 2009.

[13] A. Juels and S. A. Weis, "Defining strong privacy for RFID," *ACM Transactions on Information and System Security*, vol. 13, no. 1, pp. 7:1–7:23, Oct. 2009.

[14] R. H. Deng, Y. J. Li, M. Yung, and Y. L. Zhao, "A new framework for RFID privacy," in: D. Gritzalis, B. Preneel, and M. Theoharidou, editors, Proceedings of 15th European Symposium on Research in Computer Security-ESORICS' 10, LNCS 6345, Springer-Verlag, pp. 1–18, 2010.

[15] D. Z. Sun, An error in "On a new formal proof model for RFID location privacy", Cryptology ePrint Archive, Technical Report 2012/031, 2012.

[16] William Stallings, Cryptography and Network Security Principles and Practices, Fifth Edition, Prentice Hall, 2007, Chapter 11, pp. 327–353