On the Authentication of RFID Systems with Bitwise Operations

Basel Alomair
Network Security Lab
University of Washington
Email: alomair@u.washington.edu

Radha Poovendran
Network Security Lab
University of Washington
Email: rp3@u.washington.edu

***Abstract*—Due to the stringent computational capabilities of low-cost RFID tags, many lightweight authentication protocols have been proposed recently aiming to achieve secure authentication via bitwise operations. Following each proposal, a series of papers have been published to point out serious limitations on the security of such protocols. In this paper, we provide a detailed analysis of the security of bitwise authentication protocols in the presence of active adversaries. We divide bitwise operations into two main categories and address the security limitations of each category. Our work aims to provide guidelines for protocol designers in order to avoid pitfalls that can dangerously undermine the security of the designed protocols.**

***Index Terms*—Low-cost computing, bitwise operations, authentication, active adversary.**

I. Introduction

With the ubiquity of RFID devices growing rapidly, counterfeiting is becoming an increasingly alarming security threat. Consider an RFID tag used as an electronic key for building access control. The security of the building will rely heavily on the authenticity of the electronic key. Authenticating a counterfeited tag will lead to granting access to an illegitimate person, possibly malicious. In a different scenario, consider the rapidly spreading application of fast credit card payments based on RFID system. Transmitting the credit card number in clear text over the radio waves is obviously unacceptable. It gives eavesdroppers an easy-to-implement opportunity to counterfeit a credit card transmission. Consequently, a proper authentication mechanism is critical for the successful commercialization of RFID systems.

Entity authentication is a well-established field in cryptography with exhaustively analyzed solutions, including but not limited to [1]–[5]. In most scenarios, however, low-cost RFID tags lack the computational capabilities to perform sophisticated cryptographic operations proven to achieve secure entity authentication. Thus, leading to the search for special cryptographic primitives suitable for such devices with stringent computational capabilities.

The term *lightweight* computations has been implicitly mapped, in many places, to bitwise operations [6]–[10]. Compared to non-bitwise operations appearing in the literature of cryptography, such as modular addition, modular multiplication, modular exponentiation, etc., bitwise operations can be performed very efficiently with digital hardware [11].

Recently, many lightweight authentication protocols have appeared in the literature to accommodate the computational capabilities of low-cost RFID tags, including but not limited to [6]–[10]. Unfortunately, every authentication protocol claimed to achieve security by means of bitwise operations has been followed by a paper, or more, describing possible attacks on that protocol.

Contributions. In this paper, we provide a comprehensive analysis of the bitwise operations appearing in lightweight authentication protocols for low-cost RFID tags. Based on their properties, bitwise operations are divided into two main categories: many-to-one and one-to-one bitwise operations. We analyze bitwise operations in the presence of active adversaries. We show that many-to-one bitwise operations exhibit an undesirable information leakage property that allows active adversaries to extract secret keys with only linear number of interactions.[1] For the one-to-one bitwise operations, we show that an active adversary can modify exchanged messages in a way undetected, with probability *one*, by the legitimate receiver. We then extend our results to binary functions composed by any complex combination of bitwise operations.

We hope that this work will provide protocol designers with some insight about the properties of bitwise operations and, eventually, lead to more carefully designed protocols. We believe that most of the previously proposed protocols that have been broken or shown to have serious flaws could have been avoided if the results of this work have been available to the designers of those protocols.

Organization. The rest of the paper is organized as follows. Section II discusses relevant works. In Section III we describe our model and list the definitions and notations that will be used for the rest of the paper. Section IV describes in detail the studied bitwise operations and their vulnerabilities to active attacks. The final results and the extension to arbitrary binary functions appear in Section V. In Section VI, we conclude our paper.

[1] Linear in the length of the key.

II. Related Work

In [6], Vajda and Buttyán proposed a set of lightweight protocols to authenticate RFID tags with limited computational capabilities. Although the authors provided security analysis of their protocols, Defend *et al.* [12] described possible attacks on Vajda and Buttyán's XOR and SUBSET protocols which exploited vulnerabilities of the bitwise operations.

Juels and Wies [7] extended the human-to-computer authentication protocol proposed by Hopper and Blum in [13]. The original Hopper and Blum (HB) protocol is secure against passive attacks [13]; its security is based on a mathematical problem that is known to be NP-Hard, namely, the Linear Parity with Noise (LPN) problem [14]. The HB^+ protocol of Juels and Wies [7] uses a Fiat-Shamir commit-challenge-respond identification protocol [15] to extend the HB protocol in order to make it secure against active attacks. Shortly after Juels and Wies published their work, Gilbert *et al.* [16] exploited a property of the bitwise operations to present an active attack on the HB^+ protocol.

Peris-Lopez *et al.* proposed a sequence of lightweight mutual authentication protocols for RFID systems. They first proposed LMAP: A Real Lightweight Mutual Authentication Protocol for Low-cost RFID tags [8], followed by an improved protocol called M^2AP: A Minimalist Mutual-Authentication Protocol for Low-cost RFID Tags [9], and finally the EMAP: An Efficient Mutual Authentication Protocol for Low-cost RFID Tags [10]. In [17], Li and Wang showed active attacks on the LMAP and the M^2AP protocols. Li and Deng [18] presented an active attack on EMAP. Alomair *et al.* [19] and Barasz *et al.* [20] further exploited properties of the bitwise operations to present passive attacks on those protocols.

III. Preliminaries and Assumptions

In this section, we describe our assumptions on the communication model and list the important notations and definitions that will be used in the subsequent sections.

A. Notations

We start by providing a list of used notations.

- For two bit strings x and y of equal lengths, $(x \wedge y)$, $(x \vee y)$, $(x \oplus y)$, and $(x \cdot y)$ represent the result of the bitwise AND, OR, XOR, and binary inner product of x and y, respectively.
- For any bit b, $\neg b$ represents the complement of b.
- For a bit strings x, $x^{(i)}$ represents the i^{th} bit of x.
- For a bit strings x, $\text{MAC}_{k,*}(x)$ represents the keyed Message Authentication Code (MAC) of message x with a secret key k, using bitwise operation $*$.

B. Communication Model

At the heart of any entity authentication or secure key exchange protocol there is a MAC. An entity authentication or a secure key exchange attempt consist of at least two main parts. The first part contains information about the identity to be authenticated and/or the key to be exchanged, while the second part serves as an integrity check for the information provided by the first part. In typical scenarios, the first part will contain the identity to be authenticated and/or the key to be exchanged, and the second part will be the MAC of the first part. In symmetric key systems, given the identity, the verifier can access a shared secret key. By means of the secret key, the legitimate receiver can authenticate the identity of the other party and/or the exchanged key.

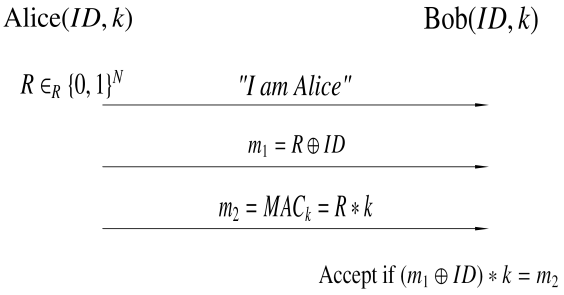

Fig. 1. A single protocol run between Alice and Bob. Alice initiates the protocol by broadcasting her name in clear text. Then she sends her secret ID XORed with a randomly generated string R. In the final message, Alice authenticate herself by providing a keyed MAC of the previous message.

Obviously, an adversary with access to the secret key can impersonate the identity of the legitimate user. Therefore, one of the most important goals of any authentication scheme is to protect the secret keys of legitimate users.

For the rest of the paper, we will assume that legitimate users share a secret key k of length N-bits. To best illustrate our results, we will assume that any entity authentication or key distribution attempt consist of three messages. In the first message, the initiator transmits a plaintext message to the receiver containing her name. The second message consists of the ID of the initiator combined with a random string R. The ID of the initiator is assumed to be secret; hence, the role of the random string R is to preserve the privacy of the ID and, depending on application, R can be the distributed key. The random string R must be a non-predictable nonce; predictable nonces do not induce the required randomness. The third message is a keyed MAC of the second message evaluated with the N-bit long shared key k using a bitwise function. For historical reasons, the initiator is called Alice and the receiver is called Bob.

The operations that will be analyzed are the bitwise AND, OR, XOR, and the binary inner-product. Formally, the message m_1 and its keyed MAC m_2 are given by:

$$m_1 = ID \oplus R, \tag{1}$$

$$m_2 = MAC_{k,*}(m_1) = m_1 * k, \tag{2}$$

where $*$ will be specified by the bitwise operation to be used. Figure 1 depicts a single protocol run between Alice and Bob.

C. Adversarial Model

We assume all messages are transmitted through a public channel that both legitimate users and the adversary share. We also assume that the adversary has a full control over the channel; that is, the adversary can modify messages

transmitted over the channel as well as block, record, modify, and replay exchanged messages.

D. Definitions

Before we can give our definition of secure entity authentication and secure key distribution, we need the definition of negligible functions as appeared in [21].

Definition 1 (Negligible Functions): A function $\gamma : \mathbb{N} \rightarrow \mathbb{R}$ is said to be negligible if for any nonzero polynomial p, there exists N_0 such that for all $N > N_0$, $|\gamma(N)| < \frac{1}{|p(N)|}$. That is, the function is said to be negligible if it converges to zero faster than the reciprocal of any polynomial function.

We now present an informal definition of secure entity authentication in the shared key setup that serves our purpose.

Definition 2 (Secure Entity Authentication): An entity authentication in the shared key setup is secure if it satisfies the following three properties:

1) If the messages communicated between legitimate users are conveyed faithfully without any modification, the entity must be authenticated.
2) The probability of authenticating an illegitimate user (without the shared key) is negligible.
3) The probability of recovering the secret keys of legitimate users by an adversary is negligible.

The definition of secure key distribution is similar to the definition of secure entity authentication.

Definition 3 (Secure Key Distribution): A key distribution in the shared key setup is secure if it satisfies the following three properties:

1) If the messages communicated between legitimate users are conveyed faithfully without any modification, the distributed key must be authenticated.
2) The probability of authenticating a key from an illegitimate user (without the shared key) is negligible.
3) The probability of recovering the secret keys of legitimate users by an adversary is negligible.

In the following section we show that entity authentication or key distribution based solely on bitwise operations fail to meet the security conditions of Definition 2 or Definition 3, respectively.

IV. Entity Authentication and Key Distribution Based on Bitwise Operations

In this section we analyze the security of authentication protocols based solely on bitwise operations.

A. The Bitwise AND and OR Operations

Assume a user Alice with a unique identity ID and a secret key k is attempting to be authenticated to another user Bob with a shared secret key. Alice generates a random number R and XORs it with her ID to compute a message m_1. The bitwise AND operation is then performed between the random string R and the secret key k to produce m_2 as below.

$$m_1 = ID \oplus R, \tag{3}$$

$$m_2 = MAC_{k,\wedge}(R) = R \wedge k. \tag{4}$$

Upon receiving m_1 and m_2, Bob extracts the random string R by XORing m_1 with Alice's ID. The random string R is authenticated by performing a bitwise AND between the extracted R and Bob's version of the secret key k, and the result is compared to m_2. That is, Bob performs the following integrity check:

$$(ID \oplus m_1) \ \wedge \ k \stackrel{?}{=} m_2. \tag{5}$$

If the integrity check of equation (5) is passed, then R as well as Alice are authenticated. Obviously, if neither m_1 nor m_2 have been modified, the integrity check of equation (5) will be satisfied and Alice will be authenticated. Thus, the first conditions of Definition 2 and Definition 3 are satisfied.

The many-to-one property of the AND function, however, produces a big security threat when used for authentication. Observe that when the i^{th} bit of the key k is *zero*, the i^{th} bit of $(R \wedge k)$ is *zero* regardless of the value of the corresponding i^{th} bit of R. Therefore, modifying a bit in m_1 that corresponds to a *zero* bit in the secret key k will go undetected by Bob, leading to a possible authentication of a modified R.

Given the key k is uniformly distributed, the probability of any bit being *zero* is equal to $\frac{1}{2}$. Consequently, flipping an arbitrary bit of m_1 will go undetected with probability $\frac{1}{2}$. Thus, violating the second condition of Definitions 2 and 3 of secure entity authentication and key distribution.

A carefully designed series of active attacks can further lead to catastrophic results for legitimate users. Consider an active adversary intercepting the exchanged messages between Alice and Bob. The adversary then flips the least significant bit of m_1 (that is, $m_1^{(0)} = \neg m_1^{(0)}$), and transmits the modified m_1 to Bob. As illustrated above, if the least significant bit of the secret key is *zero*, the least significant bit of m_2 will be *zero* regardless of the least significant bit of m_1. Therefore, if Bob authenticates the modified m_1, the adversary can infer that the least significant bit of the secret key is *zero*. Otherwise, if Bob detects the modification, the adversary can infer that the least significant bit of the secret key is *one*. This shows how an active adversary, with a single interaction with the receiver, can extract the least significant bit of the secret key.

The adversary can then replay the attack with the second least significant bit flipped. If the modification is undetected, the second least significant bit of the secret key is *zero*; otherwise, it is *one*. The adversary continues by changing one bit at a time and launching a new authentication attempt. At the N^{th} attempt, the most significant bit of the secret key is extracted and, thus, the secret key has been fully recovered by the active adversary. Therefore, the third condition of Definitions 2 and 3 is violated.

The bitwise OR operation exhibits the same information leakage property as the AND operation. Observe that when the i^{th} bit of the key k is *one*, the i^{th} bit of $(R \vee k)$ is *one* regardless of the value of the corresponding i^{th} bit of R, and the same analysis follows.

B. The Binary Inner-Product Operation

In CRYPTO 2005, Juels and Weis [7] extended the human-to-computer authentication protocol proposed by Hopper and Blum in [13] to devise HB$^+$, an authentication protocol resilient to active attacks. For simplicity, we will describe the basic Hopper and Blum (HB) protocol and show how an active adversary can attack it; the same attack can be applied to the HB$^+$ protocol.

As before, let Alice and Bob share a secret key k. When Alice initiates the conversation, Bob challenges her with a random N-bit string a. Alice computes the binary inner-product $b = k \cdot a$, and sends the result back to Bob. Obviously, the result of the binary inner-product is a single bit, called the parity bit in [7]. Upon receiving the parity bit b, Bob computes the inner-product $b' = k \cdot a$, compares it to the received parity bit b, and accepts if $b' = b$. Since the output of the inner-product is a single bit, an adversary impersonating Alice without knowing the secret key k will be successful with probability $\frac{1}{2}$. Performing this procedure r times, however, reduces the impersonator's probability of success to $(\frac{1}{2})^r$.

However, an eavesdropper capturing $O(N)$ valid challenge-response pairs between Alice and Bob can easily compute the secret key using Gaussian elimination [7]. To overcome the problem of key exposure by eavesdroppers, the HB protocol resorts to the Linear Parity with Noise (LPN) technique. More specifically, Alice intentionally sends the wrong response with constant probability $\eta \in (0, \frac{1}{2})$. Alice is authenticated if fewer then ηr of her responses are incorrect.

The HB protocol is vulnerable to active attacks by malicious authenticators. A malicious authenticator playing the role of Bob, can launch a series of adaptive (non-random) challenges in order to extract Alice's secret key. We will not discuss this attack here since it has been already mitigated in the HB$^+$ protocol. The reader can refer to [7] for more details. In what follows, we will describe an active attack that can be launched on both HB and HB$^+$ to extract the secret key.

The source of weakness in the binary inner-product operation comes from the same property of the bitwise AND and OR operations elaborated earlier; namely, the many-to-one property. Just as in the bitwise AND operation, if the i^{th} bit of the secret key is *zero*, no matter what the value of the corresponding i^{th} bit of the challenge, the binary multiplication of the two bits is *zero*. Therefore, an active adversary can proceed exactly as in the bitwise AND scenario by sequentially flipping bits of Bob's challenge a. If flipping the i^{th} bit of the challenge does not affect the authentication process, the adversary can conclude that the i^{th} bit of Alice's secret key is *zero*. On the other hand, if flipping the i^{th} bit of the challenge cause the authentication process to fail, the adversary can conclude that the i^{th} bit of Alice's secret key is *one*. Therefore, performing this attack for all $i,\ 0 \leq i \leq N-1$, will lead to the full disclosure of Alice's secret key.[2]

Obviously, as a direct consequence of the attack described above, the binary inner-product operation fails to satisfy the second and third conditions of Definitions 2 and 3 of secure entity authentication and key distribution.

[2]This attack was first published in [16].

C. The Bitwise XOR Operation

The bitwise XOR operation is more information-theoretically secure than the bitwise AND, OR, and binary inner-product operations. This extra security is due to the one-to-one property of the bitwise XOR function. Because of the one-to-one property, every possible input is mapped to precisely one output. Consequently, any modification of m_1, which will lead to the modification of R, will be detected by its MAC, i.e, m_2. However, easy to mount active attacks are still possible, as detailed below.

As in previous cases, we will assume that Alice, with a unique identity ID and a secret key k, is attempting to be authenticated to Bob. Alice generates a random number R and XORs it with her ID to compute a message m_1. The bitwise XOR operation is then performed between the random string R and the secret key k to produce m_2 as below.

$$m_1 = ID \oplus R, \tag{6}$$

$$m_2 = MAC_{k,\oplus}(R) = R \oplus k. \tag{7}$$

Upon receiving the message m_1 and its MAC m_2, Bob performs a bitwise XOR between m_1 and Alice's ID to extract R. Bob authenticates the extracted R by XORing it with his version of the secret key k and compares the result with m_2; that is, Bob performs the following integrity check:

$$(ID \oplus m_1) \ \oplus \ k \stackrel{?}{=} m_2. \tag{8}$$

Obviously, if neither m_1 nor m_2 have been modified, the integrity check of equation (8) will be satisfied and Alice will be authenticated. Thus, the first condition of Definition 2 is satisfied.

Consider an active adversary attempting to recover Alice's secret key. Trying to implement the same attack used against the bitwise AND, OR, and inner-product operations, the adversary modifies a single bit of m_1 and observes Bob's response. However, given that XOR is a one-to-one function, combined with the fact that any modification of m_1 will result in a modification of R, any modification of m_1 will be detected by Bob with probability *one*. Thus, unlike the previous operations, the adversary cannot gain any information about Alice's secret key by modifying bits of m_1 and observing Bob's response.

For any bit string b, the following are satisfied by the XOR function:

$$b \oplus b = 0, \tag{9}$$

$$b \oplus 0 = b, \tag{10}$$

$$b \oplus 1 = \neg b. \tag{11}$$

Since the adversary is assumed to have full control over the communication channel, and since m_1 and m_2 are transmitted over the channel, the adversary can modify any bits of m_1 and m_2 of her choice. For any non-zero perturbation string b, as a direct consequence of equation (11), $b \oplus m_1 \neq m_1$.

Consider modifying m_1 to $m_1' = m_1 \oplus b$, for a non-zero string b. The extracted R' will be:

$$R' = m_1' \oplus ID = (m_1 \oplus b) \oplus ID = R \oplus b. \quad (12)$$

The extracted R' is then authenticated by XORing it with the secret key as follows:

$$R' \oplus k = (R \oplus b) \oplus k = m_2 \oplus b. \quad (13)$$

Equation (13) implies that $m_2' = m_2 \oplus b$ will be authenticated by Bob. Therefore, all the adversary has to do in order to make Bob authenticate a modified R is disturb both messages m_1 and m_2 by the same non-zero string.

V. Generalization and Discussion

In what follows, we will discuss the results obtained from the previous section and then generalize them to any arbitrary binary function.

A. Many-to-One Operations

As can be seen in the previous section, the many-to-one property of the bitwise AND, OR, and binary inner-product operations can be a source of strength and weakness at the same time. Its strength comes from the fact that, since it is not a one-to-one function, modifying a single bit of m_1 (consequently, the corresponding bit in R is flipped), does not necessarily imply that the corresponding bit of m_2 is flipped. For example, when the bitwise AND operation is used, assume the i^{th} bit of m_1 has been modified and, hence, the i^{th} bit of the extracted R is flipped. The i^{th} bit of m_2 might or might not be flipped, depending on the i^{th} bit of the secret key k (this ambiguity is a consequence of the many-to-one property.) Hence, unlike the case where XOR is used, the adversary cannot make Bob authenticate modified messages with probability *one* by flipping the i^{th} bits of both m_1 and m_2.

However, there are only two possible choices in the binary field, zero and one. Thus, the ambiguity gained by the many-to-one property does not provide sufficient security. This is because, in the worst case, the effect on the i^{th} bit of m_2, caused by modifying the i^{th} bit of m_1, can be guessed correctly with probability $\frac{1}{2}$.[3]

The fact that modifying a bit of m_1 may or may not result in a modification of the corresponding bit of m_2, depending on the corresponding bit of the secret key k, can lead to a critical information leakage. An active adversary can modify a bit of m_1 and observe the response of the authenticator. With the knowledge of the bitwise operation used and the result of the authentication attempt, the corresponding bit of the secret key is exposed (see Section IV for details). Therefore, with only $O(N)$ interactions, where N is the length of the secret key, the secret key can be fully recovered.

The following result is a direct consequence of the above discussion.

[3]Given the realistic assumption that the secret key is uniformly distributed.

(a) Many-to-One Functions

k	ID	Output
0	0	0
0	1	0
1	0	0
1	1	0

k	ID	Output
0	0	0
0	1	0
1	0	1
1	1	0

k	ID	Output
0	0	0
0	1	0
1	0	1
1	1	1

k	ID	Output
0	0	1
0	1	0
1	0	0
1	1	0

k	ID	Output
0	0	1
0	1	1
1	0	0
1	1	0

k	ID	Output
0	0	1
0	1	0
1	0	1
1	1	1

k	ID	Output
0	0	0
0	1	1
1	0	0
1	1	0

k	ID	Output
0	0	0
0	1	0
1	0	0
1	1	1

k	ID	Output
0	0	0
0	1	0
1	0	1
1	1	1

k	ID	Output
0	0	1
0	1	1
1	0	0
1	1	0

k	ID	Output
0	0	1
0	1	1
1	0	0
1	1	1

(b) One-to-One Function

k	ID	Output
0	0	1
0	1	0
1	0	1
1	1	0

k	ID	Output
0	0	0
0	1	1
1	0	1
1	1	0

k	ID	Output
0	0	1
0	1	0
1	0	0
1	1	1

k	ID	Output
0	0	0
0	1	1
1	0	0
1	1	1

Fig. 2. All possible binary functions of two binary inputs.

Corollary 1: In an interactive protocol, if an exchanged quantity is authenticated with secret parameter by performing a many-to-one bitwise operation, the secret key can be fully exposed by at most $O(length\ of\ the\ key)$ malicious interactive attempts.

B. One-to-One Operations

The one-to-one property of the bitwise XOR operation implies that any modification of m_1 will result in modification of the corresponding bits of m_2. This provides good protection against any information leakage of the secret key. The same property, however, allows the adversary to know the exact effect of any modification of m_1 on the corresponding bits of m_2. Thus, allowing the adversary to modify m_2, cancelling the effect of modifying m_1. Therefore, compared to many-to-one operations, a single active attack attempt can be made successful with probability *one* (see Section IV for details).

The following result is a direct consequence of the above discussion.

Corollary 2: In an interactive protocol, if an exchanged quantity is authenticated with secret parameter by performing a one-to-one bitwise operation, the quantity can be maliciously modified in a way that makes it authenticated with probability one.

C. Arbitrary Binary Functions

Single bitwise operations, as appeared in Section IV, are not usually used for authentication protocols. A complex combination of bitwise operations is normally used to increase security. However, any combination of bitwise operations, regardless of its complexity, can be reduced to a single binary function described by a truth table. Figure 2 lists all possible binary functions of two inputs. As can be seen in the figure, any possible binary function will belong to one of two categories: many-to-one or one-to-one. Thus, having the same security properties of either the AND operation or the XOR operation.

Therefore, designing a lightweight protocol by combining multiple bitwise operations does not improve the security of the protocol.

D. Other Attacks

Depending on the design of specific bitwise protocols, different techniques can be used to improve the efficiency of the attack. In many cases, the many-to-one property can lead to possible information leakage by passively observing protocol runs between legitimate users. In [20], Barasz *et al.* described a passive attack on a bitwise authentication protocol that requires the attacker to passively eavesdrop on *few* authentication runs. Although no complexity analysis of their attack is provided, their attack shows how the properties of the bitwise operations can be exploited by a passive eavesdropper to extract information about secret parameters.

In another paper, a comprehensive work was published by Alomair *et al.* [19] where the authors described two passive attacks on two bitwise based protocols for mutually authenticating RFID systems. The authors of [19] provided a concrete complexity analysis of their passive attacks showing that the eavesdropper needs only to observe $O(\log N)$ protocol runs between legitimate users to extract the secret ID of the RFID tag. This shows that our upper bound on the complexity of key recovery attacks of Lemma 1 is rather loose and carefully designed attacks can perform much better.

VI. Conclusion

In this paper, we studied the use of basic bitwise operations for the authentication of low-cost RFID devices. We showed that relying only on bitwise operation for authentication cannot lead to secure authentication in the presence of an active adversary, contradicting the claims of many proposals appearing in the literature. We divided bitwise operations into two main categories: many-to-one bitwise operations and one-to-one bitwise operations. We showed that relying on many-to-one bitwise operations for message authentication will lead to the full exposure of the secret key in the presence of an active adversary. When a one-to-one bitwise function is used for message authentication, we showed that an active adversary can cause a modified message to be authenticated with probability one. Our work shows the necessity to incorporate non-bitwise operations (such as cryptographic hash functions, lightweight encryption schemes suitable for RFID system [22], etc.) in authentication protocols to avoid easy-to-implement active attacks.

References

[1] W. Diffie, P. Oorschot, and M. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes and Cryptography*, vol. 2, no. 2, pp. 107–125, 1992.

[2] M. Burrows, M. Abadi, and R. Needham, "A logic of authentication," *ACM Transactions on Computer Systems (TOCS)*, vol. 8, no. 1, pp. 18–36, 1990.

[3] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "Systematic Design of Two-Party Authentication Protocols," *Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology*, pp. 44–61, 1991.

[4] M. Bellare and P. Rogaway, "Entity authentication and key distribution," in *Advances in Cryptology-CRYPTO*, vol. 773. Springer, 1993, pp. 232–249.

[5] J. Clark and J. Jacob, "A survey of authentication protocol literature," 1997.

[6] I. Vajda and L. Buttyán, "Lightweight authentication protocols for low-cost RFID tags," in *Second Workshop on Security in Ubiquitous Computing – Ubicomp 2003*, Seattle, WA, USA, October 2003.

[7] A. Juels and S. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology – CRYPTO'05*, ser. Lecture Notes in Computer Science, V. Shoup, Ed., vol. 3126, IACR. Santa Barbara, California, USA: Springer-Verlag, August 2005, pp. 293–308.

[8] P. Peris-Lopez, J. C. Hernandez-Castro, J. Estevez-Tapiador, and A. Ribagorda, "LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags," Printed handout of Workshop on RFID Security – RFIDSec 06, Ecrypt, Graz, Austria, July 2006.

[9] ——, "M2AP: A minimalist mutual-authentication protocol for low-cost RFID tags," in *International Conference on Ubiquitous Intelligence and Computing – UIC06*, ser. Lecture Notes in Computer Science, vol. 4159. Springer-Verlag, September 2006, pp. 912–923.

[10] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. Estevez-Tapiador, and A. Ribagorda, "EMAP: An efficient mutual authentication protocol for low-cost RFID tags." in *OTM Federated Conferences and Workshop: IS Workshop – IS'06*, ser. Lecture Notes in Computer Science, vol. 4277. Springer-Verlag, November 2006, pp. 352–361.

[11] S. Sarma, S. Weis, and D. Engels, "RFID systems and security and privacy implications," in *Cryptographic Hardware and Embedded Systems – CHES 2002*, ser. Lecture Notes in Computer Science, B. Kaliski, c. Kaya ço, and C. Paar, Eds., vol. 2523. Redwood Shores, CA, USA: Springer-Verlag, August 2002, pp. 454–469.

[12] B. Defend, K. Fu, and A. Juels, "Cryptanalysis of two lightweight RFID authentication schemes," in *International Workshop on Pervasive Computing and Communication Security – PerSec 2007*, IEEE. New York, USA: IEEE Computer Society Press, March 2007, pp. 211–216.

[13] N. Hopper and M. Blum, "Secure human identification protocols," in *Advances in Cryptology - ASIACRYPT 2001*, ser. Lecture Notes in Computer Science, vol. 2248/2001. Springer, 2001.

[14] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (Corresp.)," *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 384–386, 1978.

[15] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," in *Advances in Cryptology-CRYPTO*, vol. 86. Springer, 1986, pp. 186–194.

[16] H. Gilbert, M. Robshaw, and H. Sibert, "An active attack against HB^+ – a provably secure lightweight authentication protocol," Manuscript, France Telecom, July 2005.

[17] T. Li and G. Wang, "Security analysis of two ultra-lightweight RFID authentication protocols," in *IFIP SEC 2007*. Sandton, Gauteng, South Africa: IFIP, May 2007.

[18] T. Li and R. H. Deng, "Vulnerability analysis of EMAP - an efficient RFID mutual authentication protocol," in *Second International Conference on Availability, Reliability and Security – AReS 2007*, Vienna, Austria, April 2007.

[19] B. Alomair, L. Lazos, and R. Poovendran, "Passive attacks on a class of authentication protocols for RFID," in *International Conference on Information Security and Cryptology – ICISC*, ser. Lecture Notes in Computer Science, K.-H. Nam and G. Rhee, Eds., vol. 4817. Seoul, Korea: Springer-Verlag, November 2007, pp. 102–115.

[20] M. Bárász, B. Boros, P. Ligeti, K. Lója, and D. Nagy, "Passive attack against the M2AP mutual authentication protocol for RFID tags," in *First International EURASIP Workshop on RFID Technology*, Vienna, Austria, September 2007.

[21] O. Goldreich, *Foundations of Cryptography*. Cambridge University Press, 2001.

[22] M. Feldhofer, S. Dominikus, and J. Wolkerstorfer, "Strong authentication for RFID systems using the AES algorithm," in *Workshop on Cryptographic Hardware and Embedded Systems – CHES 2004*, ser. Lecture Notes in Computer Science, M. Joye and J.-J. Quisquater, Eds., vol. 3156, IACR. Boston, Massachusetts, USA: Springer-Verlag, August 2004, pp. 357–370.