Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/090203

Multiple Attacks on Authentication Protocols for Low-Cost RFID Tags

Zeeshan Bilal[*], *Keith Martin and Qasim Saeed*

Information Security Group, Royal Holloway University of London, Egham, Surrey, TW20 0EX, United Kingdom

Received: 4 May 2014, Revised: 5 Aug. 2014, Accepted: 7 Aug. 2014
Published online: 1 Mar. 2015

Abstract: In this paper, we present a security analysis of two authentication protocols *SIDRFID* and *DIDRFID*, proposed for low-cost RFID systems. These protocols are considered to employ ultra-lightweight functions and are very efficient. However, we demonstrate design flaws that result in full secret disclosure in both protocols. These disclosure attacks undermine the security of both protocols. Further analysis highlights additional attacks including traceability and reader impersonation.

Keywords: RFID, Ultralightweight, Authentication Protocols, Security

1 Introduction

Radio Frequency Identification (RFID) systems are becoming pervasive in large scale identification applications [1]. The most widely deployed are low-cost RFID systems [2], where tags normally cost a few cents. These are likely to replace bar-codes since RFID tags do not require a line of sight in order to be scanned. However, there are many privacy and security concerns with low-cost RFID systems [1]. The main limiting factor in low-cost RFID tags relates to resource constraints. Since the cost has to be kept low, these tags cannot afford a state-of-the-art CPU, large memory or support large bandwidth. Generally, low-cost RFID tags consist of a few thousand gates, a simple Arithmetic and Logic Unit (ALU) performing simple operations, and no power source.

Authentication schemes in RFID systems are classified into four classes based on cost, available resources and applications [2]. Low-cost RFID systems are covered by the ultra-lightweight class. In [3], two ultra-lightweight authentication protocols are proposed. In one of the protocols, the tag and reader do not share any secrets and use their respective identities as shared secrets. These identities are, therefore, not transmitted in the clear. Moreover, these identities do not update and are static. This protocol is called Ultra-lightweight RFID Protocol with Static Identity (*SIDRFID*). In the other protocol, the tag and reader share a secret key K. After authenticating the reader, the tag sends its unique secret identity IDT. Both K and IDT are updated in each authentication round, therefore, this protocol is called Ultra-lightweight RFID Protocol with Dynamic Identity (*DIDRFID*). Both protocols claim to provide mutual authentication and implement very efficient and extremely lightweight functions. We discuss these protocols in greater depth in Section 2.

Avoine et al. [4] have carried out a security analysis of both protocols. They observe that using a single master key in *SIDRFID* is a single point of failure if compromised. However, they do not elaborate on any specific technique to recover the master key. We show in this paper how to recover this single master key and break the entire *SIDRFID* system. Further, Avoine et al. [4] highlight an attack on the secret key used in *DIDRFID*. This attack involves eavesdropping two rounds of authentication session and L^2 possible guesses (where L is the length of key). We demonstrate a passive full disclosure attack that determines the correct key after eavesdropping approximately $\sqrt{\pi L}$ rounds.

Our security analysis is explained in detail in Section 3 and Section 4 respectively. We also describe further attacks on these protocols including one where an attacker successfully traces a tag.

[*] Corresponding author e-mail: Zeeshan.Bilal.2010@live.rhul.ac.uk

Table 1: Notation

Notation	Description
IDT	Tag's static identity.
$DIDT_i$	Tag's dynamic identity used in i^{th} authentication round.
IDR	Reader's static identity.
K_i	Secret key shared between the tag and reader in i^{th} authentication round.
R_i	Random number generated by reader in i^{th} authentication round.
$\oplus$	Bitwise XOR operation.
$\vee$	Bitwise OR operation.
$\wedge$	Bitwise AND operation.
$A \rightarrow B : M$	A sends to B, message M.
X	A 96-bit string as $x_{95} \cdots x_0$, where x_0 and x_{95} are the least significant and most significant bits respectively.
$\mathrm{HW}(X)$	Hamming weight of bit string X.
$Rot(X,Y)$	Left rotation of argument X by $\mathrm{HW}(Y)$ bits.

2 Two Ultra-lightweight Authentication Protocols

In this section, we summarize the two ultra-lightweight authentication protocols suggested for use in low-cost RFID systems proposed in [3]. These protocols belong to the ultra-lightweight class designed for low-cost RFID tags and claim to provide mutual authentication. Additionally, these protocols claim to resist attacks including traceability, replay, de-synchronization and impersonation. Importantly, the computation cost is kept low by incorporating lightweight functions. In the proposed protocols, the pseudo-random number generator is only installed in the reader. The low-cost tag only performs simple bit-wise operations (XOR,AND,OR) and left rotation of bits $Rot(A,B)$.

2.1 Protocol with Static Identity (SIDRFID)

The protocol assumes that tag and reader each have identities IDT and IDR, respectively, which are secret values shared by each entity (it is assumed that tag and reader have these pre-installed prior to activation of the scheme). The i^{th} round of authentication is as shown in Figure 1 and consists of the following steps:

–Step 1.
–Reader generates R_i.
–Reader computes:

$$S_i = R_i \oplus IDR.$$

–$Reader \rightarrow Tag : S_i$

–Step 2.
–Tag computes:

$$\begin{aligned} R_i &= S_i \oplus IDR, \\ P_i &= IDT \oplus Rot(R_i, IDR), \\ Q_i &= Rot(IDT, IDT) \oplus Rot(R_i, R_i). \end{aligned}$$

–$Tag \rightarrow Reader : (P_i, Q_i)$

–Step 3.
–Reader computes:

$$\begin{aligned} IDT &= P_i \oplus Rot(R_i, IDR), \\ Q_i^{'} &= Rot(IDT, IDT) \oplus Rot(R_i, R_i)). \end{aligned}$$

–Reader authenticates tag as follows:
if $Q_i^{'} = Q_i$ **then**
Tag is authenticated.
else
Protocol is abandoned.
end if

–Step 4.
–In case of successful tag authentication, the reader computes:

$$\begin{aligned} Z_i = &Rot(IDT, IDR \oplus R_i) \\ &\oplus Rot(IDR, IDT \oplus R_i). \end{aligned}$$

–$Reader \rightarrow Tag : Z_i$

–Step 5.
–Tag computes:

$$\begin{aligned} Z_i^{'} = &Rot(IDT, IDR \oplus R_i) \\ &\oplus Rot(IDR, IDT \oplus R_i). \end{aligned}$$

–Tag authenticates reader as follows:

if $Z_i^{'} = Z_i$ **then**
Reader is authenticated.
else
Protocol is abandoned.
end if

2.2 Protocol with Dynamic Identity (DIDRFID)

The protocol assumes that tag and reader share a secret key KS (it is assumed that tag and reader have this pre-installed prior to activation of the scheme). The i^{th} round of authentication is as shown in Figure 2 and consists of the following steps:

–Step 1.
–$Tag \rightarrow Reader : DIDT_i$

–Step 2.
–Reader uses $DIDT_i$ as index to extract the corresponding secret key K_i from the database.
–Reader generates a random number R_i.
–Reader computes:

$$A_i = K_i \oplus R_i,$$
$$B_i = Rot(K_i, K_i) \oplus Rot(R_i, R_i).$$

–$Reader \rightarrow Tag : (A_i, B_i)$

–Step 3.
–Tag computes:

$$R_i = A_i \oplus K_i,$$
$$B_i^{'} = Rot(K_i, K_i) \oplus Rot(R_i, R_i).$$

–Tag authenticates reader as follows:
if $B_i^{'} = B_i$ **then**
Reader is authenticated.
else
Protocol is abandoned.
end if

–Step 4.
–In case of successful reader authentication, the tag computes:

$$C_i = Rot(K_i, R_i) \oplus Rot(R_i, K_i).$$

–$Tag \rightarrow Reader : C_i$

–Step 5.
–Reader computes:

$$C_i^{'} = Rot(K_i, R_i) \oplus Rot(R_i, K_i).$$

–Reader authenticates tag as follows:
if $C_i^{'} = C_i$ **then**
Tag is authenticated.
else
Protocol is abandoned.
end if

–Key Updating Step. After successful mutual authentication, tag and reader update their values:
–Tag and Reader compute:

$$DIDT_{i+1} = Rot(R_i, R_i \vee K_i) \oplus Rot(K_i, R_i \wedge K_i),$$
$$K_{i+1} = Rot(R_i, R_i \wedge K_i) \oplus Rot(K_i, R_i \vee K_i).$$

–Tag and Reader both keep $(DIDT_i, K_i)$ and $(DIDT_{i+1}, K_{i+1})$ in their memory.

3 Security Analysis of SIDRFID

In this section, we carry out a security analysis of *SIDRFID* [3]. Avoine et al. [4] have suggested that *SIDRFID* is a weak protocol because it uses a single master key which in many situations is considered unacceptable. However, there may be applications, such as issuing temporary RFID tags for access control to a team visiting an organization, where use of a single master key may be justified. In such scenarios, we do not need to generate new keys on every access attempt and thus avoid the need for secure distribution of these secret keys to each tag. Nonetheless we show that, even in situations where a fixed master key is justified, the secret entities can be easily recovered thus demonstrating that *SIDRFID* is a very weak protocol.

3.1 Passive Hamming Weight Disclosure (PHWD) Attack

We first present a passive attack which reveals HW(IDR). We make the realistic assumption that the channel between the tag and reader is wireless and insecure. The attacker simply needs to eavesdrop any two rounds of authentication. Moreover, the resources available to the attacker are also limited so it cannot perform heavy computations (a realistic assumption in lightweight cryptography). The attack executes as follows:

–Step 1. Attacker eavesdrops two legitimate authentication rounds to obtain S_1, P_1 and S_2, P_2.

–Step 2. The attacker computes:

$$\begin{aligned} A &= S_1 \oplus S_2, \\ &= (R_1 \oplus IDR) \oplus (R_2 \oplus IDR), \\ &= R_1 \oplus R_2. \end{aligned} \quad (1)$$

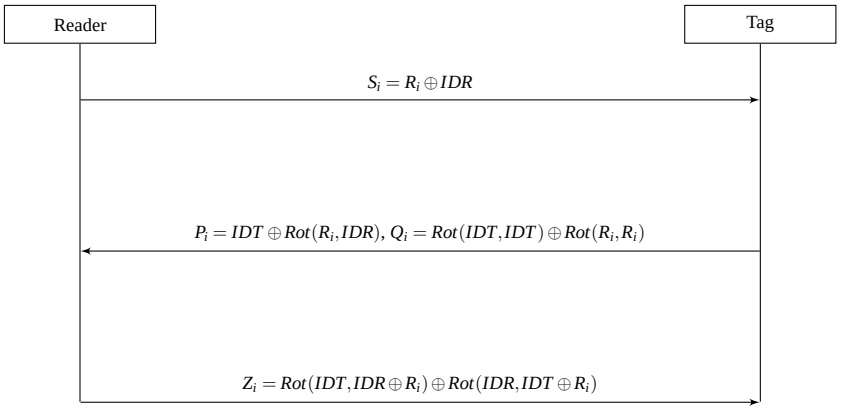

Fig. 1: Protocol with Static Identity SIDRFID.

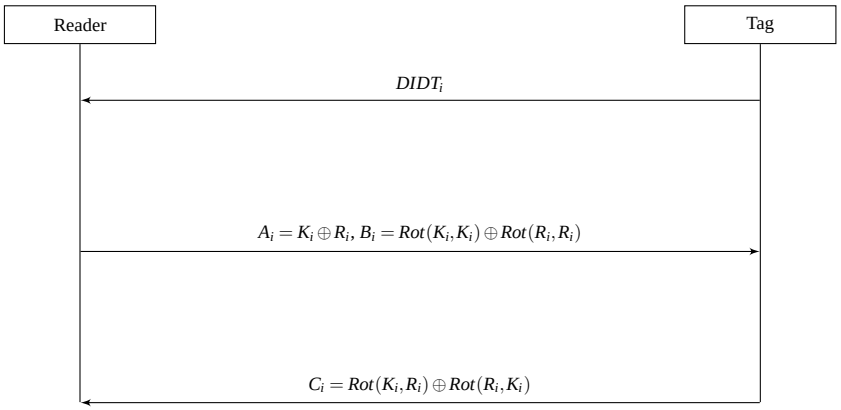

Fig. 2: Protocol with Dynamic Identity DIDRFID.

$$\begin{aligned} B =& P_1 \oplus P_2, \\ =& (IDT \oplus Rot(R_1, IDR)) \\ & \oplus (IDT \oplus Rot(R_2, IDR)), \\ =& Rot(R_1, IDR) \oplus Rot(R_2, IDR), \\ =& Rot(R_1 \oplus R_2, IDR). \end{aligned} \tag{2}$$

From (1) and (2), we get:

$$B = Rot(A, IDR). \tag{3}$$

Since A and B are known from (1) and (2), $HW(IDR)$ can easily be obtained from (3).

After disclosing HW(IDR), an attacker can carry out a selective brute force attack to find the exact value, where each value has correctness probability (considering L as the length of bit string IDR):

$$p = \frac{1}{\binom{L}{HW(IDR)}}.$$

This value is much higher than 2^{-L}, which is the probability of brute force attack success against an L-bit value. If we assume that IDR is similar to those assigned as EPC values (96-bits [5]), IDR consists of only 36 random bits (which we denote IDR^*) and the remaining 60 bits are publicly known (these determine the header, manufacturer and type of item details). This further raises the correctness probability p' of a guess to:

$$p' = \frac{1}{\binom{36}{HW(IDR^*)}},$$

which is substantially fewer trials to conduct.

3.2 Full Disclosure Active (FDA) Attack

We now present a Full Disclosure Active (FDA) attack against $SIDRFID$. We assume that either the attacker is in possession of the tag or there is no restriction on accessing the tag. This attack involves eavesdropping one

round of legitimate communication and 95 chosen public messages sent to the tag (considering the length of variables to be 96 as in the EPC standard [5]). The *FDA* attack is explained as follows:

–**Step 1.** The attacker eavesdrops a legitimate authentication round and records S_1, P_1, Q_1 and Z_1 (described in Section 2.1), where the labels of individuals bits in each of these strings is as for string X in Table 1.

–**Step 2.** The attacker impersonates a legitimate reader and sends S_2, which is a manipulated version of S_1 with the two least significant bits flipped as $s_0^{'}$ and $s_1^{'}$ (the subscript of S represents the round number and subscript of s represents the bit position).

–**Step 3.** Tag computes R_2 as follows:

$$R_2 = S_2 \oplus IDR. \tag{4}$$

Since IDR is fixed, R_2 is the same as R_1 except that the least significant two bits are flipped as $r_0^{'}$ and $r_1^{'}$ as follows:

$$\begin{aligned} R_1 &= r_{95}r_{94}r_{93}\cdots r_2 r_1 r_0, \\ R_2 &= r_{95}r_{94}r_{93}\cdots r_2 r_1^{'} r_0^{'}, \\ M &= R_1 \oplus R_2, \\ &= 00\cdots 011. \end{aligned} \tag{5}$$

Tag now computes P_2 and Q_2 where,

$$\begin{aligned} P_2 &= IDT \oplus Rot(R_2, IDR), \\ Q_2 &= Rot(IDT, IDT) \oplus Rot(R_2, R_2). \end{aligned}$$

and sends them to the attacker.

–**Step 4.** After receiving P_2 and Q_2, the attacker computes:

$$\begin{aligned} N &= P_1 \oplus P_2, \\ &= (IDT \oplus Rot(R_1, IDR)) \\ &\quad \oplus (IDT \oplus Rot(R_2, IDR)), \\ &= Rot(R_1, IDR) \oplus Rot(R_2, IDR), \\ &= Rot(R_1 \oplus R_2, IDR), \\ &= Rot(M, IDR). \end{aligned} \tag{6}$$

Since N and M are known in (6), HW(IDR) can be calculated.

–**Step 5.** The attacker now computes:

$$\begin{aligned} T &= Q_1 \oplus Q_2, \\ &= (Rot(IDT, IDT) \oplus Rot(R_1, R_1)) \\ &\quad \oplus (Rot(IDT, IDT) \oplus Rot(R_2, R_2)), \\ &= Rot(R_1, R_1) \oplus Rot(R_2, R_2). \end{aligned} \tag{7}$$

–**Step 6.** R_2 is same as R_1 except that the least two bits are flipped as $r_0^{'}$ and $r_1^{'}$, as discussed before for deriving (5). The two least significant bits of R_1, will either be the same or different with probability one half. The attacker thus analyzes (7) according to two conditions as follows:

1.**Case 1.** The two flipped bits of R_1 are different, which results in:

$$HW(R_1) = HW(R_2).$$

This simplifies (7) as follows:

$$\begin{aligned} W &= Rot(R_1 \oplus R_2, R_1), \\ &= Rot(M, R_1). \end{aligned} \tag{8}$$

Since M is a string of all 0's except for two consecutive 1's in the least significant positions (as described for (5)), W will also consist of all 0's except for two 1's at two consecutive positions in the string. The position of the first 1 starting with the least significant bit as zero determines HW(R_1). The attacker marks the least significant bit of R_1 as x and the next bit as $x^{'}$ (in this case the first two LSBs are inverses of each other).

2.**Case 2.** The two flipped bits of R_1 are the same which results in either:

$$HW(R_1) = HW(R_2) + 2,$$

or

$$HW(R_1) = HW(R_2) - 2.$$

Since HW(R_1) $\neq$ HW(R_2), this does not simplify (7). In this case the string T will be a random string of 0's and 1's without any pattern. The attacker marks the least significant bit of R_1 as x and the next bit as x, since both bits are either 0 or 1.

–**Step 7.** The attacker continues sending the next chosen plaintext S_3 by flipping (s_0, s_2). The resultant string T in this case will reveal whether r_2 is the same as r_0.

if $r_2 = r_0$ **then**
  $r_2 = x$
else
  $r_2 = x^{'}$
end if

In general, the attacker continues sending chosen plaintexts by flipping two bits (s_0, s_k) where $k = 1\cdots 95$ as shown in Figure 3. For the k^{th} round of authentication, the string T in (7) reveals two bits of R_1, (r_0, r_k), to be either the same or otherwise.

–**Step 8.** At the end of this attack, R_1 is represented as a string of x and $x^{'}$ with known HW(R_1) from (8). The attacker now replaces x's with 1's and $x^{'}$'s with 0's, or

vice versa according to HW(R_1).

–**Step 9.** The only non-trivial value will be when HW(R_1) = 48. In this case, x can either be a 1 or a 0, thus, R_1 has two possible values. In this case, the attacker uses the eavesdropped legitimate round of Step 1 and checks which of the two possible values of R_1 satisfies the values of the public messages S_1,P_1 and Q_1.

–**Step 10.** Once we get the value of R_1, we can easily determine *IDR* and *IDT* from any of the public messages. It now becomes very easy to launch multiple attacks on a tag including tag cloning, tag tracking and inventorying [1].

3.3 Other Attacks

We have just shown a full disclosure attack which completely disrupts the authentication process in *SIDRFID*. We now highlight further weaknesses in the design of this protocol which can be exploited to launch multiple attacks.

–**Traceability Attack.** We assume that a low-cost RFID tag is unable to keep track of the current status in an authentication round. It thus replies to every query sent by a compatible reader. In *SIDRFID*, the public messages P and Q are different in every authentication round because of the different random R's generated by the reader. The attacker thus eavesdrops one round of authentication and keeps on sending the same S, thus forcing the tag to calculate similar public messages. This will facilitate tracking of a particular tag.

–**Reader Impersonation.** The order of authentication is important in RFID authentication protocols and can counter several active attacks. The reader should be authenticated first so the tag may transmit its secret information only to a legitimate reader. The wrong order of authentication leads to a reader impersonation attack. An attacker can eavesdrop a legitimate authentication round. The attacker can then impersonate a legitimate reader and replay the eavesdropped response as legitimate and get itself authenticated. This attack is possible because secret values are not updated in each fresh round of authentication.

–**Identification of Reader.** *SIDRFID* does not specify how the tag determines which *IDR* is to be used to generate the public values. Therefore, a further limitation of this protocol is that it can only be implemented in scenarios where there is only one particular reader (or many readers with the same *IDR* value).

4 Security Analysis of DIDRFID

In this section, we carry out a security analysis of *DIDRFID* [3]. Avoine et al. [4] presented a key guessing attack against *DIDRFID*. This attack requires eavesdropping two authentication session and a total of L^2 possible guesses, where L is the length of the secret key. Whilst this is a serious attack, we present another variant of full disclosure attack which uniquely determines the key. This further demonstrates that *DIDRFID* is a very weak protocol.

4.1 Passive Weight Disclosure (PWD) Attack

We assume that the channel between the tag and reader is wireless and insecure. This attack first obtains HW(K) which we will then show allows us to uniquely determine the correct secret K.

The details of this protocol are given in Section 2.2 and our attack, which extracts the secret key K, is as follows:

–**Step 1.** Attacker scans the communication channel until he observes that the message B_i in (9) sent by reader to tag (forward channel) is equal to the message C_i in (10) sent by tag to reader (backward channel).

$$B_i = Rot(K_i, K_i) \oplus Rot(R_i, R_i), \tag{9}$$

$$C_i = Rot(K_i, R_i) \oplus Rot(R_i, K_i). \tag{10}$$

It is evident from (9) and (10) that $B_i = C_i$ when:

$$HW(K_i) = HW(R_i). \tag{11}$$

–**Step 2.** The probability P of meeting the condition in (11) for two random L bits values is as follows:

$$P = \sum_{i=0}^{L} \frac{\binom{L}{i}^2}{(2^L)^2}. \tag{12}$$

–**Step 3.** Once the condition in (11) is satisfied, attacker re-writes (9) and (10) as follows:

$$B_i = Rot(K_i \oplus R_i, K_i), \tag{13}$$

$$C_i = Rot(K_i \oplus R_i, K_i). \tag{14}$$

–**Step 4.** Since message A is:

$$A_i = K_i \oplus R_i. \tag{15}$$

as described in Section 2.2, (13) and (14) can be written as:

$$B_i = C_i = Rot(A_i, K_i). \tag{16}$$

Since A_i, B_i and C_i are known, HW(K_i) can be computed from (16) and thus HW(R_i) from (11).

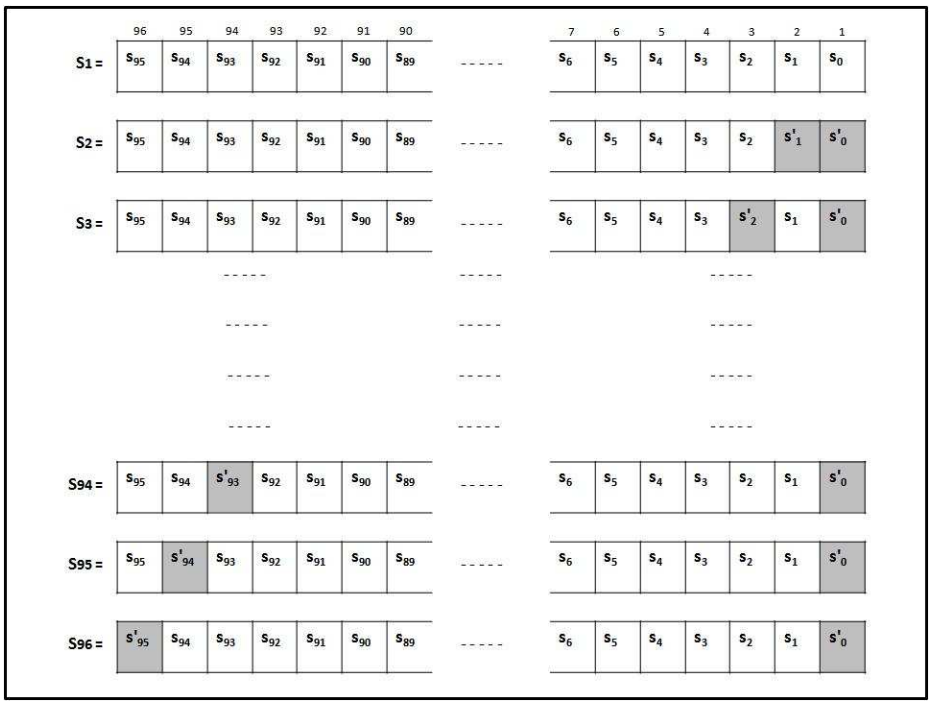

Fig. 3: Full Disclosure Attack.

–**Step 5.** Since message A_i and HW(K_i,R_i) are known, attacker uses (15) to infer the following information:

$$HW(A_i) = HW(R_i) + HW(K_i) - 2j. \quad (17)$$

where j determines the number of 1's in K_i overlapping with R_i at the same bit positions.

–**Step 6.** Attacker determines j using (17) to infer the following information:

$$HW(R_i \vee K_i) = HW(A_i) + j, \quad (18)$$

$$HW(R_i \wedge K_i) = j. \quad (19)$$

–**Step 7.** We now XOR the update equations as given in Section 2.2 as follows:

$$\begin{aligned} DIDT_{i+1} \oplus K_{i+1} &= Rot(R_i \oplus K_i, R_i \vee K_i) \\ &\quad \oplus Rot(R_i \oplus K_i, R_i \wedge K_i), \\ &= Rot(A_i, R_i \wedge K_i) \\ &\quad \oplus Rot(A_i, R_i \vee K_i). \end{aligned} \quad (20)$$

Since DID_{i+1}, A_i are public values and we use (18) and (19) to deduce the correct K_{i+1}.

4.2 Comparison between Our Attack and Avoine's Attack

The complexity of revealing the secret K for both attacks depends on the number of bits of secret K. The number of operations in Avoine's attack corresponds to the number of guesses before revealing the correct K. Avoine's attack thus requires a total of L^2 guesses and eavesdropping of two rounds of $DIDRFID$ authentication sessions.

Our attack requires a small number of rounds to be eavesdropped, but once this is done there is no further "guesswork" required since the key K is then revealed. The number of rounds are approximated as $\sqrt{\pi L}$.

From (12), the approximate number of eavesdropped rounds corresponds to $\frac{1}{P}$, in other words:

$$r = \sum_{i=0}^{L} \frac{(2^L)^2}{\binom{L}{i}^2}. \quad (21)$$

Putting $m = n = p = L$ in Vandermondes convolution formula (also called ChuVandermonde formula) see [6, 7] we see that:

$$\sum_{i=0}^{L} \binom{L}{i}^2 \approx \binom{2L}{L}. \quad (22)$$

Table 2: Comparison between Our Attack and Avoine Attack.

Type of Attack	No of Rounds to be Eavesdropped	No of Guesses before Revealing Secret Key
Avoine Attack	2	L^2
Our Attack	$\sqrt{\pi L}$ approx.	1

From Stirling's approximation see [8]:

$$\binom{2L}{L} \approx \frac{4^L}{\sqrt{\pi L}}. \quad (23)$$

Hence it follows that:

$$r \approx \sqrt{\pi L}. \quad (24)$$

We note that for the case of EPCglobal tag, $L = 96$ and hence $r = 17$. Since eavesdropping the tag-reader channel is easy, our attack can be very effective in dense reader environments where tags can be read multiple times. In other cases an ongoing authentication round can be interrupted and repeated until $B_i = C_i$. The relationship between these two attacks is summarized in Table 2.

4.3 Traceability Attack

We note an additional weakness of *DIDRFID*. If the final message C_i sent by the tag does not reach the reader due to a transmission error, or the attacker disrupts it, the reader does not recognize the updated value $DIDT_{i+1}$. The reader in this case asks for older values of $DIDT_i$ (this is not mentioned in [3]). In such a scenario, the attacker can track the tag by eavesdropping $DIDT_i, A_i, B_i$ and then disrupting message C_i. The attacker can then repeatedly ask for an older value $DIDT_i$ and send A_i, B_i in response, thus tracking the tag.

5 Conclusion

We have carried out a security analysis of the two RFID authentication protocols proposed in [3]. Earlier analysis carried out by Avoine et al. [4] on *SIDRFID* mentions only that the use of single master key is a potential weakness. We have shown how to recover this single master key, thus allowing this weakness to be fully exploited. Similarly, the attack on *DIDRFID* presented in [4] can successfully guess the correct key in L^2 attempts (where L is the length of key). We have presented another variant of a full disclosure attack which only requires the attack to eavesdrop approximately $\sqrt{\pi L}$ rounds but performs no further computation in order to disclose the secret key. We conclude that both *SIDRFID* and *DIDRFID* are both extremely weak protocols.

Acknowledgement

The authors are grateful to the anonymous referee for a careful checking of the details and for helpful comments that improved this paper.

References

[1] A. Juels, "RFID Security and Privacy: A Research Survey," *Journal on Selected Areas in Communications*, **24**, 381–394 2006, IEEE.

[2] H.-Y. Chien, "SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity," *Transactions on Dependable & Secure Computing*, **4**, 337–340, 2007, IEEE.

[3] A. Juels, Minimalist Cryptography for Low-Cost RFID Tags, Security in Communication Networks, **3352**, 149-164 (2005).

[4] G. Avoine and X. Carpent, "Yet Another Ultralightweight Authentication Protocol that is Broken," in *Radio Frequency Identification. Security and Privacy Issues*. Nijmegen, Netherlands: Springer, 2013, pp. 20–30.

[5] *EPC Radio-Frequency Identity Protocols Class-1 Generation-2 UHF RFID Protocol for Communications at 860 MHz-960 MHz, Version 1.2.0*, GS1 EPCGlobal, October 2008, http://www.gs1.org/gsmp/kc/epcglobal/uhfc1g2.

[6] J. M. Gutiérrez, M. A. Hernández, P. J. Miana, and N. Romero, "New Identities in the Catalan Triangle," *Journal of Mathematical Analysis and Applications*, **341**, 52–61, 2008, Elsevier Inc.

[7] R. P. Stanley, *Enumerative Combinatorics*. Cambridge University Press, 2011, no. 49.

[8] W. Feller, "Stirling's Formula," *An Introduction to Probability Theory and Its Applications*, **1**, 50–53, 1968, New York: Wiley.

Zeeshan Bilal holds a BE (Avionics) from NUST Pakistan, an MIT from IQRA University Pakistan, and an MS (Information Security) from NUST Pakistan. After receiving his BE degree with distinction, Zeeshan worked as an aeronautical engineer. He completed his masters in IT from IQRA University and MS in Information Security from NUST with CGPA of 4.00/4.00 in both programs. He received President's Gold Medals for his achievements. During his MS Thesis, he worked on authentication protocols for RFIDs. He has been selected for a PhD. He is working under the supervision of Prof

Keith Martin and his advisor is Dr Kostas Markantonakis. His areas of interest include Cryptography, Network Security and Wireless Systems Security.

Keith Martin is Director of the Information Security Group at Royal Holloway, University of London. He received his BSc (Hons) in Mathematics from the University of Glasgow in 1988 and a PhD from Royal Holloway in 1991. Between 1992 and 1996 he held a Research Fellowship at the University of Adelaide, investigating mathematical modelling of cryptographic key distribution problems. In 1996 he joined the COSIC research group of the Katholieke Universiteit Leuven in Belgium, working on security for third generation mobile communications. Keith rejoined Royal Holloway in January 2000, became a Professor in Information Security in 2007 and Director of the Information Security Group in 2010. Keith's current research interests include key management, cryptographic applications and securing lightweight networks. He is the author of the recently published Everyday Cryptography by Oxford University Press. As well as conventional teaching, Keith is a designer and module leader on Royal Holloways distance learning MSc Information Security programme, and regularly presents to industrial audiences and schools.

Qasim Saeed received BE (Avionics) and MS (Information Security) NUST, Pakistan. He is currently studying for a PhD under the supervision of Dr Colin Walter in the area of security aspects of the Near Field Communication (NFC) technology.