Desynchronization Attack on RAPP Ultralightweight Authentication Protocol

Zahra Ahmadian, Mahmoud Salmasizadeh, and Mohammad Reza Aref

***Abstract*—RAPP (RFID Authentication Protocol with Permutation) is a recently proposed efficient ultralightweight authentication protocol. The operation used in this protocol is totally different from the other existing ultralightweight protocols due to the use of new introduced data dependent permutations and avoidances of modular arithmetic operations and biased logical operations such as AND and OR. The designers of RAPP claimed that this protocol resists against desynchronization attacks since the last messages of the protocol is sent by the reader and not by the tag. This letter challenges this assumption and shows that RAPP is vulnerable against desynchronization attack. This attack has a remarkable probability of success and is effective whether Hamming weight-based or modular-based rotations are used by the protocol.**

***Index Terms*—RAPP, RFID security, Ultralightweight protocols, desynchronization attack.**

I. Introduction

Ultralightweight authentication protocols are an important class of lightweight authentication protocols in which the operations used in the tag side is restricted to simple bitwise operations (like XOR, AND, OR, rotation, modular addition, etc.). Since the introduction of this class by Peris-Lopez et al. [1], [2], [3] all of the proposals put forward in this area such as SASI [4], Gossamer [5], LMAP^{++} [6] and Yeh et al. [7] have made use of the mentioned operations. The exclusive or wide use of T-functions in these protocols is widely criticized since it makes some successful cryptanalysis possible [8], [9] and [10].

Recently, Tian et al. proposed a new ultraligthweight authentication protocol [11] whose operations is totally different from the other existing ultralightweight protocols. This protocol avoids modular arithmetic operations and biased logical operations such as AND and OR any more. Instead, the authors introduced a new data dependent permutation as an ultralightweight operation which is unbalanced and breaks the orders of the bits. In fact, one of the most important features of this operation is its completely non-triangular property which is highly recommended in [8].

So far two attacks have been proposed on RAPP. The first one [12] is a traceability attack which exploits Hamming weight-invariant property of the foregoing permutation. The next one [13] is an active full disclosure attack which requires about 2^{30} counterfeit authentication session with the victim tag.

Zahra Ahmadian and Mohammad Reza Aref are with Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran. Mahmoud Salmasizadeh is with Electronic Research Center, Sharif University of Technology, Tehran, Iran. (E-mail: ahmadian@ee.sharif.edu, {salmasi, aref}@sharif.edu)

In [11] the authors claimed that RAPP resists against desynchronization attacks since the last messages exchanged is transmitted by the reader rather than by the tag. So, both the old and new states of the protocol should be stored in the reader rather than the tag. We show that this assumption is not valid and the attacker can still force the tag to stay it its previous state by blocking the last messages of the protocol. Then convince it to update its state to a different state from the reader's new and old states, resulting a desynchronization between two parties. This attack works for the both definition of the rotation (hamming weight-based and modular-based).

II. RAPP Specifications

We first introduce two function used in the RAPP protocol.

Definition 1. *Let X be a L-bit word, X_i be the i^{th} bit of X where $0 \leq i \leq L-1$, and X_0 and X_{L-1} be the LSB and MSB of the word X, respectively. ($X = X_{L-1} \ldots X_1 X_0$). Suppose A and B are two L-bit words and $Hw(B) = m$ where $Hw(\cdot)$ is the Hamming weight function. Moreover $B_i = 1$ if $i \in I_1 = \{k_1, k_2, \ldots k_m\}$ and $B_i = 0$ if $i \in I_0 = \{k_{m+1}, k_{m+2}, \ldots k_L\}$, where*

$$k_m > k_{m-1} > \ldots > k_2 > k_1$$
$$k_L > k_{L-1} > \ldots > k_{m+2} > k_{m+1}$$

The permutation of A according to B, denoted as $Per(A,B)$, is defined as:

$$Per(A,B) =$$
$$A_{k_m} A_{k_{m-1}} \ldots A_{k_2} A_{k_1} A_{k_{m+1}} A_{k_{m+2}}, \ldots A_{k_{L-1}} A_{k_L}$$

For example for $A = 1101101$ and $B = 0110111$, $m = 5, I_0 = \{6,3,0\}, I_1 = \{5,4,2,1\}$ thus $Per(A,B) = A_5 A_4 A_2 A_1 A_0 A_3 A_6 = 1010111$.

Definition 2. *Suppose A and B are two L-bit words, we define Hamming weight-based rotation by $Rot(A,B) = A \lll Hw(B)$ and modular-based rotation by $Rot(A,B) = A \lll (B \; mod \; L)$ where $\lll$ is left rotation.*

In [11] it is stated that between these two options, Hamming weight-based rotation is selected for RAPP, since despite the modular-rotation, the parameter that controls the rotation is uniform in this case.

In RAPP, each tag has a unique identity ID which is shared with the reader as well as its index pseudonym, IDS and three session keys, $K1, K2, K3$. All variables are L-bit length. Although the exact value of L is not specified in [11], we assume $L = 96$ whenever required, which is the most common bit length value for the variables of a tag (e.g GID-96, SGTIN-96, GIAI-96, etc.) [14]. The state of the protocol, $(IDS, K1, K2, K3)$, is updated at the end of each successful authentication session in both parties. In order to resist desynchronization attacks, the reader keeps the backup of its previous state $(IDS^{old}, K1^{old}, K2^{old}, K3^{old})$, too. The protocol works as follows:

$$1.\ Reader \to Tag : Hello$$
$$2.\ Tag \to Reader : IDS$$

The reader uses the received IDS as a search index to to extract the secret information linked to the tag, i.e. $(ID, K1, K2, K3)$. Then the reader generates a L-bit nonce, $n1$ and computes messages A and B as follows.

$$3.\ Reader \to Tag : A, B$$

$$A = Per(K2, K1) \oplus n1$$
$$B = Per(K1 \oplus K2, Rot(n1, n1)) \oplus Per(n1, K1)$$

Upon receiving the messages A and B, the tag first extracts $n1$ from A then verifies the value of B. If the verification succeeds, the reader is authenticated and the tag computes the response value C as follows.

$$4.\ Reader \to Tag : C$$

$$C = Per(n1 \oplus K1, n1 \oplus K3) \oplus ID$$

Upon receiving C, the reader uses its local values to verify its correctness. If the tag is authenticated, the reader generates L-bit nonce $n2$ and sends the messages D and E as follows. It updates its pseudonym and secret keys while keeping its current state as well.

$$5.\ Reader \to Tag : D, E$$

$$D = Per(K3, K2) \oplus n2$$
$$E = Per(K3, Rot(n2, n2)) \oplus Per(n1, K3 \oplus K2)$$

The tag extracts $n2$ and verifies E. If it is correct, the tag updates its state in the same vein as the reader (but it does not keep the current state any more). For more details of the pseudonym and key updating see the specification of RAPP [11].

III. A DESYNCHRONIZATION ATTACK ON RAPP

It is claimed in [11] that RAPP resists desynchronization attack since despite other existing ultralightweight protocols, the last messages exchanged in this protocol are sent by the reader rather than the tag. In this section we outline a desynchronization attack on this protocol that is effective whether the protocol uses Hamming weight-based or modular-based rotation.

A. Attack description

First, assume two parties are synchronized in the state $\mathcal{S}^{(1)}$. To resist possible desynchronization attacks, the reader keeps also the previous state $\mathcal{S}^{(0)}$. The aim of the attacker is to force the tag to update its state to, say $\mathcal{S}^{(2')}$ while the reader has updated its state to $\mathcal{S}^{(2)}$ and keeping $\mathcal{S}^{(1)}$ as its old state. The attack is outlined as follows:

The attacker allows the two parties to run a genuine session except the last messages D and E and saves all the messages of this session, i.e. IDS, A, B, C, D, E. She blocks ?? the last messages to prevent the tag updating its state. Thus, the tag stays at $\mathcal{S}^{(1)}$ while the reader has updated its states to $\mathcal{S}^{(2)}$ and keeping $\mathcal{S}^{(1)}$ as its old state. Now, the attacker should convince the tag to update its state to any value other than $\mathcal{S}^{(2)}$.

in the next phase, the attacker runs counterfeit sessions with the victim tag. She uses the same messages as the previous incomplete session except D which is replaced by $D' = D \oplus e_i \oplus e_{i+1}$ for some appropriate i, where e_i is zero vector of length n that contains a 1 in the i^{th} position (i.e. $e_i = 2^i$). In more details, the attacker first sends a "Hello" message to the tag to which the tag answers with its IDS. Then the attacker transmits A and B which is definitely accepted by the tag and the tag replies with the same C again. In fourth message, the attacker sends D' and E to the tag.

Upon receiving D' and E, the tag extracts $n2' = n2 \oplus e_i \oplus e_{i+1}$ and checks the correctness of E. E is accepted if $Per(K3, Rot(n2', n2')) = Per(K3, Rot(n2, n2))$. If $Rot(n2, n2) = Rot(n2', n2') \oplus e_0 \oplus e_1$. i.e. $Rot(n2, n2)$ and $Rot(n2', n2')$ differ only in the two LSBs, we have $Per(K3, Rot(n2', n2')) = Per(K3, Rot(n2, n2))$ with probability of $\frac{1}{2}$ (this will be discussed more precisely in the following subsections).

To achieve such situation, the attacker repeats this scenario for some appropriate i, depending on the definition of rotation, and stops till she receives a *new* IDS in the next query meaning that the tag has accepted the invalid message D' and its state has been updated to the wrong state $\mathcal{S}^{(2')}$.

The success probability the attack is considered separately for the two definitions of rotation.

B. Hamming weight-based rotation: $Rot(X, Y) = X \lll Hw(Y)$

In the following, we discuss the success ratio of the attacker to desynchronize the tag when the rotation operation is defined as $Rot(X, Y) = X \lll Hw(Y)$. We first state some lemmas.

Lemma 1. *Let X be a L-bit word. Then, $Pr(Hw(X) = Hw(X \oplus e_i \oplus e_{i+1})) = \frac{1}{2}$. This happens if $X_i \neq X_{i+1}$.*

Proof: This can easily be verified. ■

Lemma 2. *Let $Z = Per(X, Y)$ and $Z' = Per(X, Y')$ where $Y' = Y \oplus e_1 \oplus e_0$. Then $Pr(Z = Z') = \frac{1}{2}$. This happens if $X_0 = X_1$.*

Proof: Let $I_1 = \{k_1, k_2, \ldots k_m\}$ and $I_0 = \{k_{m+1}, k_{m+2}, \ldots k_L\}$ be the set of indexes whose corresponding bits in Y are equal to 1 and 0 respectively, as defined in

Definition 1. Then,

$$Z = Per(X,Y) =$$
$$X_{k_m}X_{k_{m-1}}\dots X_{k_2}X_{k_1}X_{k_{m+1}}X_{k_{m+2}}\dots X_{k_{L-1}}X_{k_L}$$

Let's consider three separate cases:

- If the two last bits of Y are not the same, i.e. $Y_1Y_0 = 01$ in which $k_1 = 0, k_{m+1} = 1$ or $Y_1Y_0 = 10$ in which $k_1 = 1, k_{m+1} = 0$. in both cases we can conclude that

$$I'_1 = \{k_{m+1}, k_2, \dots k_m\}$$
$$I'_0 = \{k_1, k_{m+2}, \dots k_L\}$$

  where I'_0 and I'_1 are the two sets of indexes corresponding to Y'. Moreover $k_{m+1} < k2$ and $k_1 < k_{m+2}$. Thus,

$$Z' = Per(X,Y') =$$
$$X_{k_m}X_{k_{m-1}}\dots X_{k_2}X_{k_{m+1}}X_{k_1}X_{k_{m+2}}\dots X_{k_{L-1}}X_{k_L}$$

- If $Y_1Y_0 = 11$, then $k_1 = 0, k_2 = 1$, and we can conclude that

$$I'_1 = \{k_3, \dots k_m\}$$
$$I'_0 = \{k_1, k_2, k_{m+1}, \dots k_L\}$$

  Moreover $k_1 < k_2 < k_{m+1} < k_{m+2}$. Thus,

$$Z' = Per(X,Y') =$$
$$X_{k_m}X_{k_{m-1}}\dots X_{k_1}X_{k_2}X_{k_{m+1}}X_{k_{m+2}}\dots X_{k_{L-1}}X_{k_L}$$

- If $Y_1Y_0 = 00$, then $k_{m+1} = 0, k_{m+2} = 1$. Thus

$$I'_1 = \{k_{m+1}, k_{m+2}, k_1, \dots k_m\}$$
$$I'_0 = \{k_{m+3}, \dots k_L\}$$

  Moreover $k_{m+1} < k_{m+2} < k_1 < k_2$. Thus,

$$Z' = Per(X,Y') =$$
$$X_{k_m}X_{k_{m-1}}\dots X_{k_2}X_{k_1}X_{k_{m+2}}X_{k_{m+1}}\dots X_{k_{L-1}}X_{k_L}$$

Therefore, in the all three cases the two *edge* parts of Z' do not change and in the *middle* part, the positions of two bits of X are exchanged. ($X_{k_{m+1}}, X_{k_1}$ in the first case, X_{k_1}, X_{k_2} in the second case, and $X_{k_{m+1}}, X_{k_{m+2}}$ in the third case which are the two LSBs of X in the all cases, i.e. X_0 and X_1).

Therefore, $Z = Z'$ if $X_0 = X_1$, which holds with probability of $\frac{1}{2}$. ■

Now we turn to our main problem. $D' = D \oplus e_i \oplus e_{i+1}$ implies that $n2' = n2 \oplus e_i \oplus e_{i+1}$, where $n2'$ is the nonce that victim tag extracts from the last messages of the counterfeit authentication session.

Let $r = Hw(n2)$ and $r' = Hw(n2')$ be the quantities which controls the amount of rotations in genuine and counterfeit authentication sessions respectively. According to Lemma 1, $Pr(r = r') = \frac{1}{2}$.

Now we can conclude:

$$\begin{aligned} Rot(n2', n2') &= (n2 \oplus e_i \oplus e_{i+1}) \lll r' \\ &= (n2 \lll r) \oplus e_{i+r} \oplus e_{i+1+r} \end{aligned} \quad (1)$$

provided that $r = r'$. The subscripts in (1) are taken in modulo L. Therefore, when the attacker tries all i, it yields $i = -r \ mod \ L$ for some $0 \le i \le L-1$. This causes $Rot(n2', n2') = Rot(n2, n2) \oplus e_0 \oplus e_1$ implying $Per(K3, Rot(n2', n2')) = Per(K3, Rot(n2, n2))$ with probability of $\frac{1}{2}$, according to Lemma 2.

To summarize, the attacker succeeds to desynchronize the tag, if the following events occur:

$\mathcal{A}$: $n2_i = n2_{i+1}$ which implies $r' = r$;
$\mathcal{B}$: $K3_0 = K3_1$ which implies
$Per(K3, Rot(n2, n2) \oplus e_0 \oplus e_1) = Per(K3, Rot(n2, n2))$.

Thus, the success probability, $P_{succ,Hw}$, is equal to:

$$P_{succ,Hw} = P(\mathcal{A}) \cdot P(\mathcal{B}) = \frac{1}{4}$$

C. Modular-based rotation: $Rot(X,Y) = X \lll (Y \ mod \ n)$

In this subsection we show that the attack is successful even if the rotation operation is defined as $Rot(X,Y) = X \lll (Y \ mod \ L)$. First, some lemmas are stated.

Lemma 3. *Let X be a L-bit word. Then,*

$$X \oplus e_i = \begin{cases} X + 2^i & \textit{if } X_i = 0, \\ X - 2^i & \textit{if } X_i = 1, \end{cases}$$

Proof: This can easily be verified. ■

Lemma 4. *Let $L = 96$ and $5 \le i \le L-1$. Then,*

$$2^i \ mod \ L = \begin{cases} 32 & \textit{if i is odd}, \\ 64 & \textit{if i is even}, \end{cases}$$

Proof: This can easily be proved by induction on i. ■

Corollary 1. *Let X be a L-bit word and $5 \le i \le L-2$. Then $(X + 2^i + 2^{i+1}) \ mod \ L = X \ mod \ L$.*

Here, we have again $n2' = n2 \oplus e_i \oplus e_{i+1}$ since $D' = D \oplus e_i \oplus e_{i+1}$. Let $r = n2 \ mod \ L$ be the quantity which controls the amount of rotation. r depends on all bits of $n2$ since $n = 96$ is not an integer power of 2. Thus, flipping bit i will affect the amount of rotation. In the counterfeit session:

$$\begin{aligned} r' &= n2' \ mod \ L \\ &= (n2 \oplus e_i \oplus e_{i+1}) \ mod \ L \\ &= (n2 \pm 2^i \pm 2^{i+1}) \ mod \ L \\ &= r + (\pm 2^i \pm 2^{i+1}) \ mod \ L \end{aligned} \quad (2)$$

By Lemma 3, if $n2_i = n2_{i+1}$, the second term of (2), $\pm 2^i \pm 2^{i+1}$ will be equal to either $2^i + 2^{i+1}$ or $-2^i - 2^{i+1}$, where in both cases is equal to zero in modulo L, according to Corollary 1. Thus we have $Pr(r = r') = \frac{1}{2}$ and we can conclude again:

$$\begin{aligned} Rot(n2', n2') &= (n2 \oplus e_i \oplus e_{i+1}) \lll r' \\ &= (n2 \lll r) \oplus e_{i+r} \oplus e_{i+1+r} \end{aligned} \quad (3)$$

provided that $r = r'$. Therefore, when the attacker tries all $5 \le i \le L-2$, $Pr(i = -r \ mod \ L) = \frac{L-6}{L} = 0.94$. If this happens, it holds that $Rot(n2', n2') = Rot(n2, n2) \oplus e_0 \oplus e_1$ which implies $Per(K3, Rot(n2', n2')) = Per(K3, Rot(n2, n2))$ with probability of $\frac{1}{2}$, according to Lemma 2.

To summarize, the attacker succeeds to desynchronize the tag, if the following event occurs in addition to events $\mathcal{A}$ and $\mathcal{B}$,

$$\mathcal{C} : i = -r \; mod \; L \text{ for some } i, 5 \leq i \leq L - 2;$$

Thus, the success probability, $P_{succ,mod}$, becomes:

$$\begin{aligned} P_{succ,mod} &= P(\mathcal{A}) \cdot P(\mathcal{B}) \cdot P(\mathcal{C}) \\ &= \frac{1}{2} \times \frac{1}{2} \times 0.94 \\ &= 0.23. \end{aligned}$$

Which shows that with a slight reduction in the probability of success, the desynchronization attack still works on RAPP with the modular-based rotation.

IV. Conclusion

In this letter, we proposed a desynchronization attack on the recently proposed ultralightweight authentication protocol RAPP [11]. The attack is effective for both definitions of rotations (i.e. Hamming weight-based and modular-based rotation) and its probability of success is about $\frac{1}{4}$.

Though this work shows a weakness of this protocol, the main contribution of [11], i.e. the introduction the random permutation as a new ultralightweight operation opens up new horizons in design of ultralightweight authentication protocols. One may achieve stronger properties by combining the traditional operations with this new one.

References

[1] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda, ”LMAP: A Real Lightweight Mutual Authentication Protocol for Low-Cost RFID Tags,” Proc. Second Workshop RFID Security, July 2006.

[2] P. Peris-Lopez, J.C. Hernandez-Castro, J.M. Estevez-Tapiador, and A. Ribagorda, ”M2AP: A Minimalist Mutual-Authentication Protocol for Low-Cost RFID Tags,” Proc. Ubiquitous Intelligence and Computing, pp. 912-923, 2006.

[3] 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,” Proc. OTM 06 Workshop, pp. 352-361, 2006.

[4] H. Chien, ”SASI: A New Ultralightweight RFID Authentication Protocol Providing Strong Authentication and Strong Integrity, IEEE Trans. Dependable and Secure Computing, vol. 4, no. 4, pp. 337-340, Oct.-Dec. 2007.

[5] P. Peris-Lopez, J. C. Hernandez-Castro, J. M. E. Tapiador, and A. Ribagorda, ”Advances in ultralightweight cryptography for low-cost RFID tags: Gossamer protocol,” in Proc. International Workshop on Information Security Applications 2008, pp. 5668, 2008.

[6] T. Li, ”Employing lightweight primitives on low-cost rfid tags for authentication,” Proc. IEEE Vehicular Technology Conference, (VTC ’08), pp. 1-5, Sept. 2008.

[7] K.H. Yeh, N.W. Lo, and E. Winata, ”An Efficient Ultralightweight Authentication Protocol for RFID Systems,” In Proc. of RFIDSec Asia 2010, Cryptology and Information Security Series, vol. 4, pp. 4960, 2010.

[8] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, ”Recursive Linear and Differential Cryptanalysis of Ultralightweight Authentication Protocols,” Cryptology ePrint Archive, Report 2012/489.

[9] G. Avoine, X. Carpent, and B. Martin,”Privacy-Friendly Synchronized Ultralightweight Authentication Protocols in the Storm,” Journal of Network and Computer Applications, vol. 35, pp. 826-843, 2012.

[10] P. D’ Arco and A. De Santis, ”On Ultralightweight RFID Authentication Protocols,” IEEE IEEE Trans. Dependable and Secure Computing, vol. 8, no. 4, Jul.-Aug. 2011.

[11] Y. Tian, G. Chen, and J. Li. A New Ultralightweight RFID Authentication Protocol with Permutation. IEEE Communications Letters, Vol. 16, No. 5, May 2012, pp.702-705.

[12] G. Avoine, X. Carpent, ”Yet Another Ultralightweight Authentication Protocol that is Broken,” in pre-proceeding of RFIDsec 2012.

[13] W. Shao-hui, H. Zhijie, L. Sujuan, C. Dan-wei, ”Security Analysis of RAPP An RFID Authentication Protocol based on Permutation,” Cryptology ePrint Archive, Report 2012/327, 2012.

[14] GS1 EPCglobal. EPCglobal Tag Data Standards Version 1.4, http://www.epcglobalinc.org/standards/