Analysis and Enhancement of Desynchronization Attack on an Ultralightweight RFID Authentication Protocol

Da-Zhi Sun, Zahra Ahmadian, Yue-Jiao Wang, Mahmoud Salmasizadeh, and Mohammad Reza Aref

***Abstract*—As low-cost RFID tags become more and more ubiquitous, it is necessary to design ultralightweight RFID authentication protocols to prevent possible attacks and threats. We reevaluate Ahmadian *et al.*'s desynchronization attack on the ultralightweight RFID authentication protocol with permutation (RAPP). Our results are twofold: (1) we demonstrate that the probability of the desynchronization between the tag and the reader is 15/64 instead of 1/4 as claimed, when RAPP uses Hamming weight-based rotation; (2) we further improve the original attack and make the desynchronization more efficient.**

***Index Terms*—RFID security, cryptography, ultralightweight protocol, authentication, desynchronization attack.**

I. Introduction

Radio Frequency Identification (RFID) has been widely treated as a promising automatic identification technology. However, the RFID system raises serious authentication and privacy concerns. The main reasons resulted in the security threats include not only low computational power and small size of the tag but also the wireless communication between the reader and the tag. The RFID system typically composes of a group of tags, a reader, and a back-end database.

The work of Dr. Da-Zhi Sun was supported in part by the National Natural Science Foundation of China under Grant Nos. 61003306 and 61272106. The work done by Zahra Ahmadian, Mahmoud Salmasizadeh, and Mohammad Reza Aref was supported in part by the Iranian National Science Foundation (INSF) under Contract No. 88114/46 and INSF cryptography chair and in part by the Office of Vice-President for Science and Technology, I.R. Iran.

D.-Z. Sun is with the School of Computer Science and Technology, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, P.R. China, and also with Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, P.R. China (e-mail: sundazhi@tju.edu.cn).

Z. Ahmadian is with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: ahmadian@ee.sharif.edu).

Y.-J. Wang is with the School of Computer Science and Technology, Tianjin University, No. 92 Weijin Road, Nankai District, Tianjin 300072, P.R. China (e-mail: jiaojiao7835@126.com).

M. Salmasizadeh is with the Electronic Research Center, Sharif University of Technology, Tehran, Iran (e-mail: salmasi@sharif.edu).

M.-R. Aref is with the Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran (e-mail: aref@sharif.edu).

From the view of the security analysis, the reader and the back-end database are usually considered as a single entity, called the reader. In practice, the RFID security protocol is able to neutralize the potential threats.

As low-cost tags become more and more ubiquitous, it is necessary to design ultralightweight RFID authentication protocols to provide the security service and the privacy protection. The ultralightweight RFID authentication protocol only requires some bit operations, such as AND, OR, NOT, and XOR, etc., and excludes more expensive operations, such as public key functions, random number generators, and hash functions. In 2012, Tian *et al.* [1] presented a new ultralightweight authentication protocol with permutation called as RAPP. RAPP tries to provide the mutual authentication between the reader and the tag. The attractive advantage of RAPP is that the computational cost in the tag side is trivial. Therefore, RAPP is ideally suited for the low-cost tags. The designers also claimed that RAPP can resist desynchronization attacks, since the last message of RAPP is sent by the reader rather than by the tag. However, Ahmadian *et al.* [2] subsequently proposed a novel desynchronization attack on RAPP, which tries to deceive the tag into an abnormal state. As a result, the proposed attack desynchronizes the secret keys shared in the tag and the reader and therefore renders future successful protocol session impossible.

We reevaluate Ahmadian *et al.*'s desynchronization attack. We demonstrate that the success probability of the proposed attack is overestimated, when RAPP uses Hamming weight-based rotation. That is, the probability of the desynchronization between the tag and the reader is 15/64 instead of 1/4 as claimed. Moreover, we show that the original desynchronization attack can be improved to reduce the number of unsuccessful efforts, and the repetition of the improved attack can increase the probability of the desynchronization. But, the improved attack cannot avoid the failure case just like the original attack. Our analysis results will be beneficial to design more secure ultralightweight RFID authentication protocols.

II. Review of RAPP

For a self-contained discussion, we briefly review the required parts of RAPP and use the same symbols and notions as in [2]. The full technique details of RAPP can be found in [1], [2]. RAPP

involves three crucial operations: bitwise XOR ⊕, permutation Per(·,·), and left rotation Rot(·,·). Per(·,·) and Rot(·,·) are the crucial operations and can be described as follows.

Definition 1. Suppose A and B are two L-bit words, i.e., $A=A_{L-1}\ldots A_1A_0$ and $B=B_{L-1}\ldots B_1B_0$. Suppose $Hw(B)=m$, where $Hw(\cdot)$ denotes the Hamming weight function. Moreover, we define $B_i=1$ if $i\in I_1=\{k_m, \ \ldots, k_2, k_1\}$ and $B_i=0$ if $i\in I_0=\{k_{m+1}, \ldots, k_{L-1}, k_L\}$, where $0\le i\le L-1$, $k_m>\ldots>k_2>k_1$, and $k_{m+1}<\ldots<k_{L-1}<k_L$. The permutation Per($A$, B) can be defined as:

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

Example 1. For $A=A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0=1011101010$ and $B=B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0=0110111001$, we have $m=Hw(B)=6$, $I_1=\{8, 7, 5, 4, 3, 0\}$, and $I_0=\{1, 2, 6, 9\}$. Thus, Per(A, B)$=A_8A_7A_5A_4A_3A_0A_1A_2A_6A_9=0110101011$.

Definition 2. Suppose A and B are two L-bit words. We define Hamming weight-based rotation by Rot(A, B)$=A<<<Hw(B)$, where <<< is left rotation.

Example 2. For $A=A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0=1101101010$ and $B=B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0=0110111001$, we have $m=Hw(B)=6$. Thus, Rot(A, B)$=A<<<Hw(B)=A_3A_2A_1A_0A_9A_8A_7A_6A_5A_4=1010110110$.

In RAPP, each tag has a unique identity ID and shares four dynamic parameters, i.e., IDS, $K1$, $K2$, and $K3$, with the back-end database of the reader. Fig. 1 illustrates the authentication session of RAPP. The reader uses the nonces $n1$ and $n2$ to prevent the replay attack. After authentication process, both the reader and the tag perform the key update as follows.

IDS^*=Per(IDS, $n1\oplus n2$)$\oplus K1\oplus K2\oplus K3$;

$K1^*$=Per($K1$, $n1$)$\oplus K2$;

$K2$=Per($K2$, $n2$)$\oplus K1$;

$K3$=Per($K3$, $n1\oplus n2$)$\oplus IDS$;

$IDS=IDS^*$;

$K1=K1^*$.

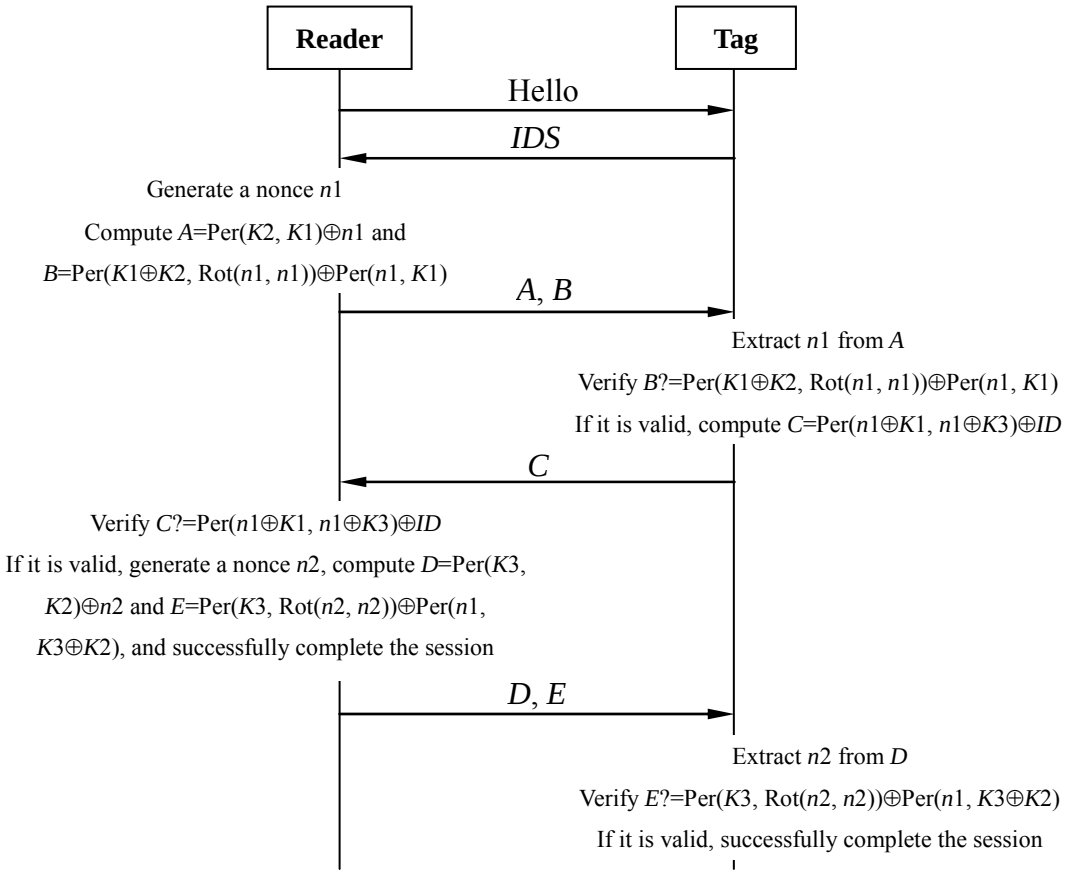

Fig.1. Authentication session of five-pass RAPP protocol.

III. Flaws in Desynchronization Attack on RAPP

The idea of the desynchronization attack [2] is to modify the authentic value D to the counterfeit value D' in the last message of RAPP. It means that the tag extracts a counterfeit nonce $n2'=D'\oplus\text{Per}(K3, K2)$ instead of the authentic nonce $n2=D\oplus\text{Per}(K3, K2)$ in the authentication session. As a result, the key update in the tag side will perhaps be wrong, because $n2'\neq n2$.

To achieve the desynchronization, the proposed attack obviously requires a necessary condition. That is, the values D' and E should also pass the tag's authentication process. The values D' and E are accepted if $\text{Per}(K3, \text{Rot}(n2', n2'))=\text{Per}(K3, \text{Rot}(n2, n2))$. Let X be any L-bit word, i.e., $X=X_{L-1}\ldots X_1X_0$. X_0 is the Least Significant Bits (LSB) of the word X, and X_1X_0 is the

two LSBs of the word X. Based on **Definition 1**, Ahmadian *et al.*'s analysis [2] shows that Per($K3$, Rot($n2'$, $n2'$))=Per($K3$, Rot($n2$, $n2$)) with probability 1/2, when Rot($n2'$, $n2'$) and Rot($n2$, $n2$) differ only in their two LSBs. Therefore, the attacker can merely change two consecutive bits of the authentic value D to generate the counterfeit value D'. By **Definition 2**, the change of two consecutive bits probably causes that Rot($n2'$, $n2'$) and Rot($n2$, $n2$) differ only in their two LSBs. The attacker can iteratively try different counterfeit values D' to make above case happen. According to Ahmadian *et al.*'s result, the following events should simultaneously occur to ensuring the desynchronization, i.e., Per($K3$, Rot($n2'$, $n2'$))=Per($K3$, Rot($n2$, $n2$)).

$\mathcal{A}$: $n2_{i+1}=n2_i$ for some appropriate i.

$\mathcal{B}$: $K3_1=K3_0$.

By **Definition 2**, it is clear that the definition of the event $\mathcal{A}$ is invalid. We can correct the event $\mathcal{A}$ as follows.

$\mathcal{A}'$: $n2_{i+1}\neq n2_i$ for some appropriate i.

Let Pr(Ev) denote the probability that the event Ev occurs. In [2], the success probability of the desynchronization attack, conditioned that $i=-r \bmod L$, is estimated by

$$P_{succ,\,Hw} = \Pr(\mathcal{A})\Pr(\mathcal{B}) = \frac{1}{2}\frac{1}{2} = \frac{1}{4}, \tag{2}$$

where $r=Hw(n2)$ and $0\le i\le L-1$ for the Hamming weight-based rotation.

We must argue that the probability $P_{succ,\,Hw}$ merely represents the probability of the key update using the counterfeit nonce $n2'$, but is not equal to the probability of the successful desynchronization between the tag and the reader. To confirm this fact, we focus on the following state of the proposed attack. Although the tag and the reader respectively use the counterfeit nonce $n2'$ and the authentic nonce $n2$, the parameters IDS, $K2$, and $K3$ in both the tag side and the reader side remain the same after the key update. In this state, the tag and the reader are still synchronizing after the proposed attack. It need point out that the updating parameter $K1$ does not use the counterfeit nonce $n2'$ at all. We need the following property of the permutation Per($\cdot$,$\cdot$) to determine the necessary events for above state and estimate the related probabilities

using the counterfeit nonce $n2'$.

Lemma 1. Suppose A and B are two L-bit words and $Hw(B)=m$. We write $A=A_{L-1}\ldots A_{i+1}A_i\ldots A_1A_0$ and $B=B_{L-1}\ldots B_{i+1}B_i\ldots B_1B_0$, where $0\le i\le L-2$. Let B' be another L-bit word merely differentiated from B in two consecutive bits. We can write $B'=B'_{L-1}\ldots B'_{i+1}B'_i\ldots B'_1B'_0=B_{L-1}\ldots\neg B_{i+1}\neg B_i\ldots B_1B_0$, where $\neg$ is the bit NOT operator. Let $Z=\mathrm{Per}(A, B)$ and $Z'=\mathrm{Per}(A, B')$. If $A_{i+1}=A_i$ and $B_{i+1}\neq B_i$, we have $Z=Z'$.

Proof. Since B_{i+1}, $B_i\in\{0, 1\}$ and $B_{i+1}\neq B_i$, we know either $B_{i+1}=1$ and $B_i=0$ or $B_{i+1}=0$ and $B_i=1$. Without loss of generality, consider $B_{i+1}=1$ and $B_i=0$. For $B=B_{L-1}\ldots B_{i+1}B_i\ldots B_1B_0$, suppose $B_t=1$ if $t\in I_1=\{k_m, \ldots, k_f=i+1, \ldots, k_2, k_1\}$ and $B_t=0$ if $t\in I_0=\{k_{m+1}, \ldots, k_g=i, \ldots, k_{L-1}, k_L\}$, where $k_m>\ldots>k_f>\ldots>k_2>k_1$, $k_{m+1}<\ldots<k_g<\ldots<k_{L-1}<k_L$, $1\le f\le m$, and $m+1\le g\le L$. According to **Definition 1**, we have

$$\begin{aligned}\mathrm{Per}(A, B) &= A_{k_m}\ldots A_{k_f}\ldots A_{k_2}A_{k_1}\,A_{k_{m+1}}\ldots A_{k_g}\ldots A_{k_{L-1}}A_{k_L}\\ &= A_{k_m}\ldots A_{i+1}\ldots A_{k_2}A_{k_1}\,A_{k_{m+1}}\ldots A_i\ldots A_{k_{L-1}}A_{k_L}\end{aligned}. \tag{3}$$

For the corresponding $B'=B'_{L-1}\ldots B'_{i+1}B'_i\ldots B'_1B'_0=B_{L-1}\ldots\neg B_{i+1}\neg B_i\ldots B_1B_0$, we have $B'_t=1$ if $t\in I'_1=\{k_m, \ldots, k_f=i, \ldots, k_2, k_1\}$ and $B'_t=0$ if $t\in I'_0=\{k_{m+1}, \ldots, k_g=i+1, \ldots, k_{L-1}, k_L\}$. According to **Definition 1**, we get

$$\begin{aligned}\mathrm{Per}(A, B') &= A_{k_m}\ldots A_{k_g}\ldots A_{k_2}A_{k_1}\,A_{k_{m+1}}\ldots A_{k_f}\ldots A_{k_{L-1}}A_{k_L}\\ &= A_{k_m}\ldots A_i\ldots A_{k_2}A_{k_1}\,A_{k_{m+1}}\ldots A_{i+1}\ldots A_{k_{L-1}}A_{k_L}\end{aligned}. \tag{4}$$

Since $A_{i+1}=A_i$, we can obtain $\mathrm{Per}(A, B)=\mathrm{Per}(A, B')$. □

We illustrate **Lemma 1** as follows.

Example 3. For $A=A_9A_8A_7A_6A_5A_4A_3A_2A_1A_0=1011101010$ and $B=B_9B_8B_7B_6B_5B_4B_3B_2B_1B_0=01\mathbf{10}111001$, we have known $\mathrm{Per}(A, B)=A_8A_7A_5A_4A_3A_0A_1A_2A_6A_9=0110101011$ in **Example 1**. Consider $B'=B'_9B'_8B'_7B'_6B'_5B'_4B'_3B'_2B'_1B'_0=B_9B_8\neg B_7\neg B_6B_5B_4B_3B_2B_1B_0=01\mathbf{01}111001$. We have $m'=Hw(B')=6$, $I'_1=\{8, \mathbf{6}, 5, 4, 3, 0\}$, and $I'_0=\{1, 2, \mathbf{7}, 9\}$. Thus, $\mathrm{Per}(A, B')=A_8A_6A_5A_4A_3A_0A_1A_2A_7A_9=0110101011=\mathrm{Per}(A, B)$. We can see $A_7=A_6$ and $B_7\neq B_6$.

Now, consider the computations for the key update. To desynchronize the reader and the tag, the event $\mathcal{A}'$ and $\mathcal{B}$ should happen at the same time, i.e., $n2_{i+1}\neq n2_i$ and $K3_1=K3_0$. If $\mathrm{Per}(IDS, n1\oplus n2)=\mathrm{Per}(IDS, n1\oplus n2')$, we have $IDS^*=\mathrm{Per}(IDS, n1\oplus n2)\oplus K1\oplus K2\oplus K3=\mathrm{Per}(IDS, n1\oplus n2')\oplus$

$K1 \oplus K2 \oplus K3$. In Table I, it shows that $n1_{i+1} \oplus n2_{i+1} \neq n1_i \oplus n2_i$, when $n1_{i+1}=n1_i$. By **Lemma 1**, it therefore requires $n1_{i+1}=n1_i$ and $IDS_{i+1}=IDS_i$ to ensure Per(IDS, $n1 \oplus n2$)=Per(IDS, $n1 \oplus n2'$). By **Lemma 1**, it similarly requires $n1_{i+1}=n1_i$ and $K3_{i+1}=K3_i$ to ensure $K3$=Per($K3$, $n1 \oplus n2$)$\oplus IDS$= Per($K3$, $n1 \oplus n2'$)$\oplus IDS$. And, $K2_{i+1}=K2_i$ is sufficient to ensure $K2$=Per($K2$, $n2$)$\oplus K1$=Per($K2$, $n2'$)$\oplus K1$. So, besides the events $\mathcal{A}'$ and $\mathcal{B}$, the following events $\mathcal{D}$, $\mathcal{E}$, $\mathcal{F}$, and $\mathcal{G}$ should happen at the same time, when the tag uses the counterfeit nonce $n2'$ but still has the same IDS, $K1$, $K2$, and $K3$ with the reader.

$\mathcal{D}$: $n1_{i+1}=n1_i$.

$\mathcal{E}$: $IDS_{i+1}=IDS_i$.

$\mathcal{F}$: $K2_{i+1}=K2_i$.

$\mathcal{G}$: $K3_{i+1}=K3_i$.

Suppose each bit of an L-bit word takes on the values 1 and 0 with same probability 1/2. The probability of the pseudo desynchronization using the proposed attack can therefore be estimated by

$$P_{pseudosucc,\,Hw} = \Pr(\mathcal{A}')\Pr(\mathcal{B})\Pr(\mathcal{D})\Pr(\mathcal{E})\Pr(\mathcal{F})\Pr(\mathcal{G}) = \frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2}\frac{1}{2} = \frac{1}{64}. \quad (5)$$

Correspondingly, the probability of real desynchronization using the proposed attack is

$$P'_{succ,\,Hw} = P_{succ,\,Hw} - P_{pseudosucc,\,Hw} = \frac{1}{2} - \frac{1}{64} = \frac{15}{64}. \quad (6)$$

TABLE I

TRUTH TABLE FOR THE DESYNCHRONIZATION ATTACK

$n1_{i+1}$	$n1_i$	$n2_{i+1}$	$n2_i$	$n1_{i+1} \oplus n2_{i+1}$	$n1_i \oplus n2_i$
0	0	0	1	0	1
0	0	1	0	1	0
1	1	0	1	1	0
1	1	1	0	0	1
0	1	0	1	0	0
0	1	1	0	1	1
1	0	0	1	1	1
1	0	1	0	0	0

IV. Improvement of Desynchronization Attack on RAPP

To reduce the number of the unsuccessful efforts, we can improve the desynchronization attack. And then, we analyze its probability more accurately. Actually, the basic steps of the desynchronization attack in [2] are unchanged here. Although the attack is the same completely, a more accurate analysis is given here.

Also thanks to **Lemma 1**, the improved desynchronization attack need not try all consecutive bits any more. However, in [2], it is emphasized that the attack should be repeated for all i to ensure that the condition $i=-r \bmod L$ is satisfied somewhere. Certainly, we can further use the repetition of the improved attack to increase the success probability. Assume that the improved attack terminates in the m^{th} effort. Clearly, in this situation, all previous $m-1$ unsuccessful efforts can be observed and roughly determined by the unchanged *IDS*. We can estimate that the success probability after m efforts is

$$\begin{aligned} P_{succ,\,rep} &= \Pr(\mathcal{A}')\Pr(\mathcal{B})\sum_{n=1}^{m}\left(1-\Pr(\mathcal{A}')\Pr(\mathcal{B})\right)^{n-1} - P_{pseudosucc,\,Hw}\sum_{n=1}^{m}\left(1-\Pr(\mathcal{A}')\Pr(\mathcal{B})\right)^{n-1} \\ &= \frac{1}{4}\sum_{n=1}^{m}\left(1-\frac{1}{4}\right)^{n-1} - \frac{1}{64}\sum_{n=1}^{m}\left(1-\frac{1}{4}\right)^{n-1} = \frac{15}{16}\left(1-\left(\frac{3}{4}\right)^{m}\right) \end{aligned}, \qquad (7)$$

which is equal to about 92% for $m=13$ but does not exceeds 15/16. Recall that in [2] the success probability of the attack was estimated 25% for $m=96$.

V. Conclusion

We have pointed out that the success probability in [2] is inaccurate, because the key update mechanism is omitted in the probability analysis. Furthermore, we have shown that the improvement on the original attack can make the desynchronization more efficient. As also claimed in [2], the desynchronization between the tag and the reader can be verified by the new *IDS* in the next protocol session. However, due to our analysis in Section III, the result of the next protocol session need be observed to determine whether the desynchronization between the tag and the reader is really successful. This operation is necessary for not only the original attack but also the improved attack.

It is an intractable task to design the ultralightweight RFID authentication protocol, because

the security engineer must cope with the trade-offs among security, cost, and performance. In RAPP, the implementation cost in the tag side can certainly be reduced due to the permutation Per(·,·) instead of the cryptographic hash function, compared with the hash-based RFID protocols. However, Per(·,·) should similarly provide the security properties of the cryptographic hash function. Based on our analysis, Per(·,·) is weaker than the cryptographic hash function from the viewpoint of the security, and therefore RAPP suffers from the desynchronization problem. Per(·,·) should be redesigned in the future. We hope that our research result is helpful to security engineers, who are responsible for the design and development of the RFID authentication systems.

REFERENCES

[1] Y. Tian, G.L. Chen, and J.H. Li, "A new ultralightweight RFID authentication protocol with permutation," *IEEE Communications Letters*, vol. 16, no. 5, pp. 702–705, May 2012.

[2] Z. Ahmadian, M. Salmasizadeh, and M.R. Aref, "Desynchronization attack on RAPP ultralightweight authentication protocol," *Information Processing Letters*, vol. 113, no. 7, pp. 205–209, Apr. 2013.