R-MAC – A lightwheight authentication protocol for RFID Tags

Ahmad Khoureich Ka

Dept. of Computer Science, Université Alioune Diop de Bambey, Sénégal,
ahmadkhoureich.ka@uadb.edu.sn

Abstract. Nowadays, RFID systems have earned an important place in our everyday lives. Its adoption is growing in areas where data security or privacy or both must be guaranteed. Since the RFID tag can be cloned, it is necessary to develop appropriate security solutions for these systems. Fortunately, many papers have proposed solutions for encryption or authentication. But it turns out that sometimes the proposal has security flaw or is too complicated for the RFID tag (which has very limited computing resources). In this paper we introduce a new authentication protocol using a new approach. Our proposal named R-MAC is well suited for RFID tags since it uses simple and lightweight constructions. R-MAC is at least as secure as the Mirror-Mac protocol introduced by Petros Mol *et al.*

Keywords: RFID, MAC, authentication, lightweight protocol, Xor-Cascade Encryption.

1 Introduction

In order maintain the large-scale adoption of RFID technologies in healthcare, in the retail supply chain and in the automotive industry to name a few, security in their implementation must be taken into account. Unfortunately RFID tags have very weak computing power so that classical cryptographic primitives do not suit them. Therefore it is necessary to invent secure and lightweight cryptographic primitives, which fit perfectly to the context of RFID systems. But designing such lightweight protocols is not easy, since they must fit on resource limited devices but analyzed or attacked using all available means. Nevertheless, several solutions for authentication and encryption (following different approaches) suited for RFID tags have been proposed. For authentication protocols we can mention the family of HB protocols [21,22,18,26,23,32], Message Authentication Codes (MACs) exploiting the difficulty of the LPN problem [3,21,31] and others based on mathematical problems [33]. For lightweight ciphers we have LPN-C [17], PRESENT [4], PRINCE [5] to name a few.

In this paper we propose a new 3-round MAC authentication protocol named R-MAC (stands for Random MAC) embedding a lightweight secret key sharing protocol. This proposal is not meant to say that the existing authentication protocols are not good but to enrich the offering of secure protocol appropriate

for resource-limited devices. Our R-MAC protocol exploits (improves as well) some theoretical but provably secure constructions such as the Mirror-Mac (a 2-Round MIM-Secure Protocol) introduced by Petros et al [28] and the Xor-Cascade Encryption [16,25,15,6] (used as framework for building block ciphers). We also introduce in this paper a Small-Cipher called SC which not only enjoys simple and lightweight algorithms but is also resistant to linear and differential cryptanalysis and to algebraic attacks. R-MAC uses SC as a building block.

This paper is structured as follows. A brief introduction and a presentation of existing works are done respectively in sections 1 and 2. Our R-MAC protocol is described in section 3 followed by the security arguments that weighs in its favor in section 4. Finally, a conclusion is given in section 5.

2 Existing Work

A number of works has been done on low-cost authentication protocols. We can mention the large family of HB-like protocols [21,22,18,26,23,32] which take advantage of the difficulty of solving the LPN problem (acknowledged as being an NP-complete problem [3,21,31]). The probabilistic nature of the reader's final response (accepting or rejecting the tag) in HB protocols is generally exploited to develop attacks against them [19,14,29]. In addition, despite their attractive design, which implies low computing resource requirements, their communication cost (between the tag and the reader) is often very high. Thus it is hard to see an efficient HB-like protocol secure against man-in-the-middle attacks. However, there are other lightweight authentication protocols based on MACs. For example SQUASH [33] based on the Rabin encryption scheme, and others based on the LPN problem [24,20,11]. But the one that most caught our attention is the Mirror-Mac (MM) protocol from Petros *et al* [28]. MM is a generic construction of a 2-round MIM secure protocol (see figure 1). Petros et al have proven that when instantiated with a suf-rmcc (strongly unforgeable under random-message-chosen-challenge) MAC, MM is secure even if the adversary interacts at will with an arbitrary number of both prover and verifier instances. That is the adversary has negligence chance to make the reader accept. Our proposal R-MAC uses MM as a framework with some little changes.

3 The New Protocol

R-MAC is an authentication protocol that uses a 128-bit key and operates with 64-bit challenge and 64-bit response. In the rest of the paper we set $n = 64$.

The core of the new protocol is an implementation of the 2-Xor-Cascade Encryption [16].

The security of Xor-Cascade Encryption (which can also be seen as a Generalized Even-Mansour Cipher[1]) is widely studied in many papers [16,25,15,6].

[1] The Generalized Even-Mansour Cipher is a generalization of the one-round Even-Mansour schema [13].

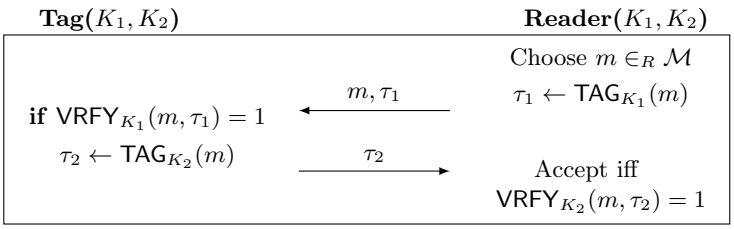

Fig. 1. The generic construction of a 2-round MIM secure protocol called MM (Mirror-Mac) using MAC = (KGen,TAG,VRFY) where the keys K_1 and K_2 are generated by KGen, TAG takes as input a key K and a message m in a message space $\mathcal{M}$ and outputs $\mathsf{TAG}_K(m)$ and VRFY takes as input a key K, a message m and a tag τ in the tag space $\mathcal{T}$ then outputs a decision $\mathsf{VRFY}_K(m, \tau) \in \{0, 1\}$.

Consider an ensemble $\mathcal{P} = \{P_i\}_{i \in \{0,1\}^\kappa}$ of random permutations of $\{0,1\}^n$, the r-Xor-Cascade Encryption $\mathsf{XC}^r_\mathcal{P}$ with $r \leq 2^\kappa$ defines a block cipher with message space $\{0,1\}^n$ and key space $\{0,1\}^{(r+1)n+\kappa r}$ as follow: given a key (k, w) where $k = (k_0, k_1, \cdots k_r) \in (\{0,1\}^n)^{r+1}$, $w = (i_1, i_2, \cdots i_r) \in (\{0,1\}^\kappa)^r$ and a message $x \in \{0,1\}^n$,

$$\mathsf{XC}^r_\mathcal{P}(k, w, x) = k_r \oplus P_{i_r}(k_{r-1} \oplus P_{i_{r-1}}(\cdots P_{i_2}(k_1 \oplus P_{i_1}(k_0 \oplus x)) \cdots)) \ .$$

A considerable number of papers [16,25,15,6,7] have shown that, in the model where the adversary is given oracle access to inner permutations $P_i \in \mathcal{P}$ of her choice and their inverses and to the outer permutation $\mathsf{XC}^r_\mathcal{P}$, the security of the Xor-Cascade Encryption approaches $2^{\kappa+n}$ when r is increasing. Also other papers explored attacks on these constructions [9,2,12,10].

We know from [16] that the 2-Xor-Cascade Encryption is secure up to $2^{\kappa+n/2}$ query complexity. When such construction is implemented in RFID tags (this must be a lightweight implementation) with a fixed key, the security threshold of $2^{\kappa+n/2}$ queries can be easily reached. Thus in order to make such an attack difficult and burdensome, we renew the key with a lightweight key establishment protocol between the tag and the reader before running the core protocol.

3.1 The Lightweight key Establishment Protocol

The lightweight key establishment protocol we describe here is run before the core authentication protocol. Figure 2 shows how the key establishment protocol works. The two parties share a secret key S of size $2n$. The tag begins by drawing uniformly at random α from $\{0,1\}^{2n}$ and then sends it to the reader. They both compute $S' = \mathsf{MixBits}(S, \alpha)$ where MixBits is a mixing function. After that, the reader draw uniformly at random β from $\{0,1\}^{2n}$, computes $\beta' = \beta \oplus S'$ and sends the result to the tag. Finally the two parties derive two n-bit long secret keys K_1 and K_2 by using $\Delta(\beta, S')$. The function Δ acts as a comb on β with n teeth randomly spaced, the space between the teeth is define by S' (see algorithm

1). Any secure lightweight mixing function can be used but here we propose to use the mixing function introduced in the Gossamer protocol because it has an extremely lightweight nature [30] (see below).

The MixBits function

```
Z = MixBits(X,Y)
Z = X;
for(i=0; i<32; i++) {
   Z = (Z>>1) + Z + Z + Y ;
}
```

Algorithm 1: $\Delta(X, S)$. Deriving two bit strings K_1 and K_2 from X using S as a comb.

Input: Two $2n$-bit strings $X = x_1 \dots x_{2n}$ and $S = s_1 \dots s_{2n}$
Output: Two n-bit strings K_1 and K_2
Processing:

```
Initialize K1 ← null
Initialize K2 ← null
Initialize b ← 0
if wt(S) > n then b ← 1

for i = 1 ... 2n do
    if s_i = b and |K1| < n then
        K1 ← K1||x_i
    else K2 ← K2||x_i
```

3.2 Design Details of The Core Authentication Protocol

The core authentication protocol is almost similar to the Mirror-Mac protocol [28] except that in our protocol the challenge is encapsulated in β' using a one-time pad and whatever the result of the verification of the pair $(\mathsf{prf}_n(\beta), \tau_1)$ by the tag a bit string is returned to the reader (see figure 2). The tag and the reader share a long-lived key S and two session keys K_1 and K_2 obtained from the lightweight key establishment protocol. The reader draws a random $2n$-bit string β, computes $\beta' = \beta \oplus S'$ and $\tau_1 = \mathsf{SC}(\mathsf{prf}_n(\beta))$ (where S' is computed during the session key sharing, $\mathsf{prf}_n(\beta)$ is the prefix of length n of β and SC is the Small-Cipher we introduced below) then sends (β', τ_1) to the tag. Upon receiving (β', τ_1), the tag checks whether τ_1 is equal to $\mathsf{SC}(\mathsf{prf}_n(\beta))$, and if so, sends $\tau_2 = \mathsf{SC}(\tau_1)$ to the reader, if not, sends a random n-bit string to the reader. The latter accepts the tag if and only if $\tau_2 = \mathsf{SC}(\tau_1)$.

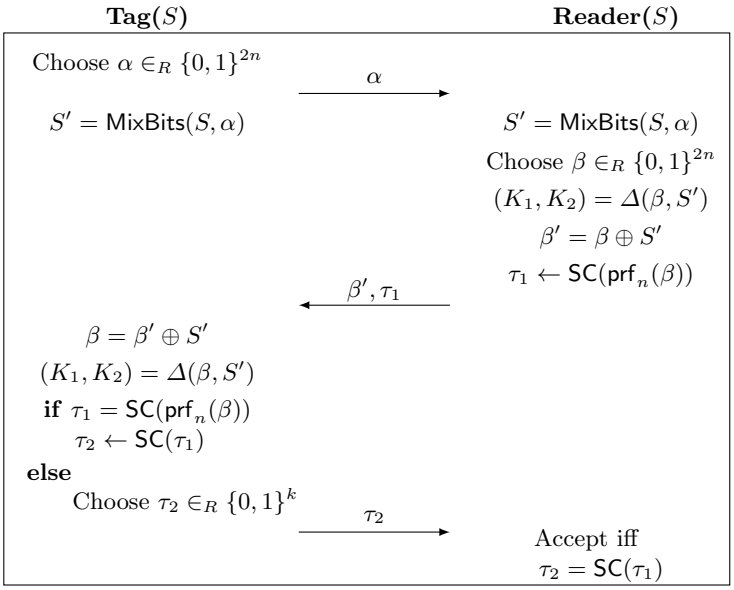

Fig. 2. The R-MAC Authentication Protocol using SC with keys K_1, K_2, S_1 and S_2. $\mathsf{prf}_n(\beta)$ is the prefix of length n of β.

Now we present our Small-Cipher SC designed to be extremely lightweight and secure. It does not have all the requirement of a full-fledged cipher because it is used only for encryption. SC is an implementation of the 2-Xor-Cascade Encryption. Its inner permutation is an SP-network denoted R-BOX (stands for Random-Box). Basically an SP-network applies to its input (a plaintext and a key) many rounds of transformation each consisting of a random substitution of bits value along the input text (using an S-box), a permutation of bit positions (using a P-box) and a key mixing. The P-boxes of SP-networks are usually a fixed permutation of the bit positions of the state but here we introduce keyed one.

Now, we successively present the S-box, the P-box, the Small-Cipher inner permutation (R-BOX) and Small-Cipher itself.

The S-box. It consists of the 4-bit to 4-bit S-box borrowed from the ultra-lightweight block cipher PRESENT [4]. This S-box is designed with hardware efficiency in mind for resource-limited devices. The following table recalls its action in hexadecimal notation.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S-Box[x]	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

The Pbox. Let $L : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be the bit positions permutation we introduce here. That is for every $K \in \{0,1\}^n$, $L(K,\cdot)$ or $L_K(\cdot)$ is a permutation on $\{0,1\}^n$ that preserve the hamming weight of its input. The way that $L_K(\cdot)$ computes the image of an input $X \in \{0,1\}^n$ is given by algorithm 2.

Algorithm 2: $L_K(X)$. Bit positions permutation.

Input: Two n-bit strings $K = k_1 \dots k_n$ and $X = x_1 \dots x_n$
Output: An n-bit strings Y
Processing:

```
Initialize Y_0 ← null
Initialize Y_1 ← null
for i = 1 ... n do
    if k_i = 0 then
        Y_0 ← Y_0||x_i
    else Y_1 ← Y_1||x_i
Y ← Y_0||Y_1
```

For a randomly chosen $K \in \{0,1\}^n$, the first plot from the top of figure 3 shows how L_K maps the original position of a bit of state to its new position. Note that the diffusion power of L_K is weak because some streak of consecutive bits of state are not perturbed despite its action. We can also see from the plotting two groups of points forming two superimposed slopes. Bits of K, which are equal to 1, give the group at the top and bits of K, which are equal to 0, give the group at the bottom. In order, to achieve a better diffusion, we can iterate L_K a number of time. Figure 3 shows the improvement obtained by iterating L_K. Lemma 1 gives the relation that exists between the original position of a bit and its final position after t iterations of L_K. Also it becomes clear from lemma 1 that the more we iterate L_K the more the final position of a bit is unrelated to its original position but only depends on K.

Lemma 1. *Let $t \geq 0$ be an integer, i be the position of a bit of state and $p^t(i)$ its position after t iterations of L_K.*

$$p^t(i) = i \prod_{r=0}^{t-1} (1 - k_{p^r(i)}) + \sum_{r=0}^{t-1} \left[\left(k_{p^r(i)}(n - w + 2w_{p^r(i)}) - w_{p^r(i)} \right) \prod_{u=r+1}^{t-1} (1 - k_{p^u(i)}) \right] . \quad (1)$$

Where $k_{p^r(i)}$ is the bit at position $p^r(i)$ of K, w the Hamming weight of K and $w_{p^r(i)}$ the Hamming weight of the prefix of length $p^r(i)$ of K.

Proof. We prove lemma 1 using the induction principle.
Basis: For $t = 0$ we have $p^0(i) = i$. Thus equation 1 is true for $t = 0$.

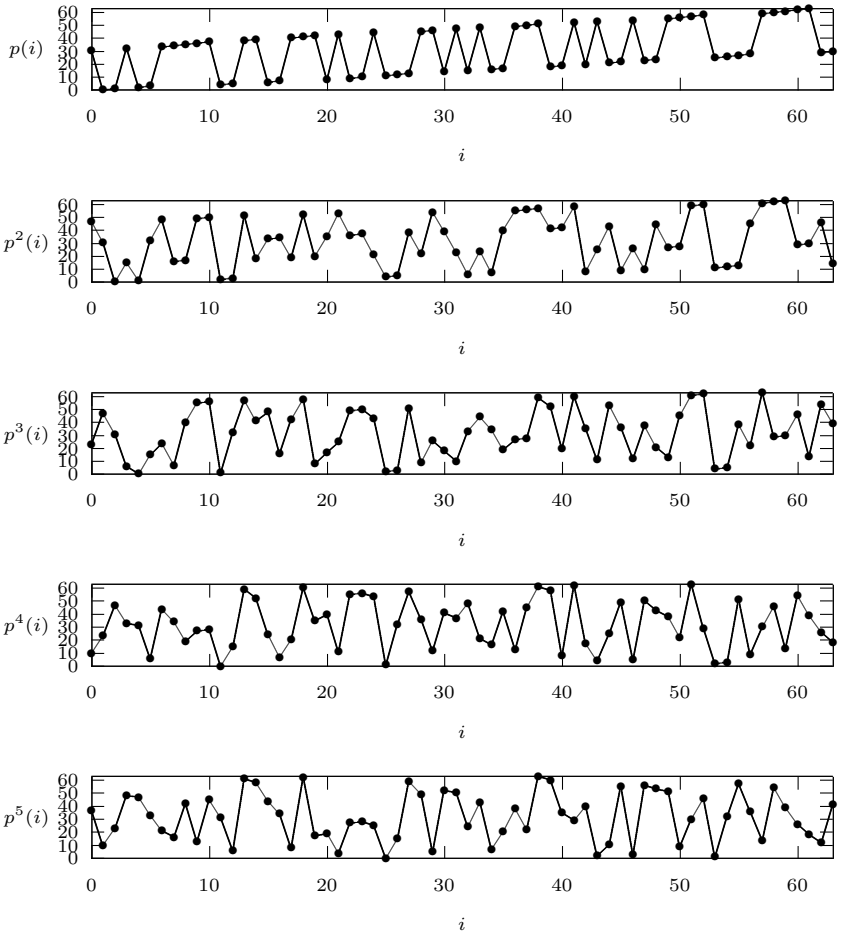

Fig. 3. Improvement of the diffusion power by iterating L_K where K is the 64-bit string $\texttt{93E6748D4E52787C}_{16}$. From top to bottom we have the 1^{st} to the 5^{th} iteration of L_K.

Induction: Now we assume $t > 0$ and show that equation 1 holds for $t+1$ iterations of L_K. From algorithm 1 we have:

$$p(i) = \begin{cases} i - w_i & \text{if } k_i = 0 \\ n - w + w_i & \text{otherwise} \end{cases}$$

which leads to:

$$p(i) = i(1-k_i) + k_i(n - w + 2w_i) - w_i \ .$$

So

$$p^{t+1}(i) = p^t(i)(1-k_{p^t(i)}) + k_{p^t(i)}(n - w + 2w_{p^t(i)}) - w_{p^t(i)} \ .$$

By supposing that equation 1 is true, we have:

$$\begin{aligned}
p^{t+1}(i) = \Bigg[& i \prod_{r=0}^{t-1}(1-k_{p^r(i)}) \\
& + \sum_{r=0}^{t-1} \left[\left(k_{p^r(i)}(n-w+2w_{p^r(i)}) - w_{p^r(i)}\right) \prod_{u=r+1}^{t-1} (1-k_{p^u(i)}) \right] \Bigg] (1-k_{p^t(i)}) \\
& + k_{p^t(i)}(n-w+2w_{p^t(i)}) - w_{p^t(i)} \\
= \ & i(1-k_{p^t(i)}) \prod_{r=0}^{t-1}(1-k_{p^r(i)}) \\
& + (1-k_{p^t(i)}) \sum_{r=0}^{t-1} \left[\left(k_{p^r(i)}(n-w+2w_{p^r(i)}) - w_{p^r(i)}\right) \prod_{u=r+1}^{t-1} (1-k_{p^u(i)}) \right] \\
& + k_{p^t(i)}(n-w+2w_{p^t(i)}) - w_{p^t(i)} \\
= \ & i \prod_{r=0}^{t}(1-k_{p^r(i)}) + \sum_{r=0}^{t-1} \left[\left(k_{p^r(i)}(n-w+2w_{p^r(i)}) - w_{p^r(i)}\right) \prod_{u=r+1}^{t} (1-k_{p^u(i)}) \right] \\
& + k_{p^t(i)}(n-w+2w_{p^t(i)}) - w_{p^t(i)} \\
= \ & i \prod_{r=0}^{t}(1-k_{p^r(i)}) + \sum_{r=0}^{t} \left[\left(k_{p^r(i)}(n-w+2w_{p^r(i)}) - w_{p^r(i)}\right) \prod_{u=r+1}^{t} (1-k_{p^u(i)}) \right] \ .
\end{aligned}$$

This final equation completes the proof. □

The R-BOX. It is composed of the S-box presented earlier surrounded by two iterated P-Box layers. Since our P-boxes are keyed, let K_1 and K_2 be two n-bit keys, therefore $\mathsf{R\text{-}BOX}_{K_1K_2}$ consists of five iterations of L_{K_1} followed by the S-box and five iterations of L_{K_2} (see figure 4 for a depiction of R-BOX).

Small-Cipher. In an initial phase, both the tag and the reader hold the same $2n$-bit secret key S. Let $S_1 = \mathsf{prf}_n(S)$ and $S_2 = \mathsf{sfx}_n(S)$ be respectively the prefix of length n of S and the suffix of length n of S. From the execution of

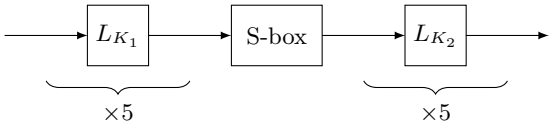

Fig. 4. R-BOX$_{K_1K_2}$. The P-boxes L_{K_1} and L_{K_2} are iterated five times.

the lightweight key establishment protocol the two parties obtained two n-bit session keys K_1 and K_2. Since it's only required in the iterated Even-Mansour cipher to have the 3-round keys to be 2-wise independent [7]. Then, by using the two independent n-bit permutations R-BOX$_{K_1S_1}$ and R-BOX$_{S_2K_2}$ and the round keys $(K_1, K_1 \oplus K_2, K_2)$ we have our implementation of the 2-Xor-Cascade Encryption depicted in Figure 5.

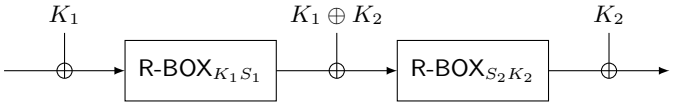

Fig. 5. Our Small-Cipher using keys K_1, K_2, S_1 and S_2

Table 1 provides a comparison of R-MAC with other authentication protocols.

Table 1. Security assumption, storage and transmission cost of some authentication protocols. *Rabin crypto* stands for Rabin cryptosystem, *qSDH* for q-Strong Diffie-Hellman and *2-XC* for 2-Xor Cascade Encryption. Values are given in bits.

Protocol	Assumption	Key(s)	Key storage	Transmission cost
HB$^+$ [22]	LPN$_{n,\varepsilon}$	$K_x = 80; K_y = 512$	592	690252
HB$^\#$ [18]	LPN$_{n,\varepsilon}$	$K_X = 80; K_Y = 512$	2918	1756
SQUASH-128 [33]	Rabin crypto	$K = 64$	64	96
MM$_{\text{qSDH}}$ [28]	qSDH	$(g, K) \in \mathbb{G} \times \mathbb{Z}_p$	?	$(m, \tau_1, \tau_2) \in \mathbb{Z}_p \times \mathbb{G}^2$
R-MAC	2-XC	$S = 128$	128	384

4 Security Arguments

In this section, we present security arguments of our R-MAC authentication protocol.

4.1 Security of The Lightweight Key Establishment Protocol

Any modification on α will only change how K_1 and K_2 are extracted from β. A modification of α will also change the value of β' (which is a one-time pad encryption of β) transmitted to the tag by the reader. Since β is drawn uniformly at random from $\{0,1\}^{2n}$, the attacker derives no benefit from its actions on α.

4.2 Security of The Core Authentication Protocol

The core authentication protocol is almost similar to the MM protocol introduced in [28] except that for our protocol the challenge is encapsulated in β' using a one-time pad and whatever the result of the verification of the pair $(\mathsf{prf}_n(\beta), \tau_1)$ by the tag an n-bit string is returned to the reader. That is not the case of the MM protocol where the tag responds to the challenge of the reader only if the verification is correct (see figure 1). Encapsulating the challenge in β' using the one-time pad encryption reduces the security requirements on SC which is used here as a MAC and must be strongly unforgeable under random-message-chosen-challenge attacks [28]. Therefore even if the attacker succeeded in finding a valid message-tag pair (m, τ), she has very little chance to reach the next step of the protocol since she will not know how to encapsulate m in β'. Another aspect that reinforces the security of our protocol is that the tag returns a response to the reader regardless the outcome of the verification $(\mathsf{prf}_n(\beta), \tau_1)$. This prevents the attacker from using the tag as a verification oracle since he has no way of detecting a change in the behavior of the tag following the result of checking the submitted pair (m, τ). This is one more security element that our protocol has over the MM protocol. All this, allows us to say that our protocol can be seen as a generic construction (SC can be replaced by a weak MAC) at least as secure as MM which is a Man-in-the-Middle secure protocol based on weak MACs.

4.3 Security of SC using R-BOX

The Structure of SC. Our Small-Cipher (SC) is an implementation of the 2-Xor-Cascade Encryption using R-BOX as the inner permutation. The 2-round Xor-Cascade Encryption is secure up to $2^{\kappa+n/2}$ query complexity [16]. For SC, $\kappa = 2n$ thus SC is theoretically (since the R-BOXes cannot be considered as random permutations) secure up to $2^{5n/2}$ queries to the underlying R-BOXes and to SC itself. This is why it is advantageous to change the R-BOXes keys for at each execution of R-MAC.

Resistance to Linear and Differential Cryptanalysis. Linear and differential cryptanalysis [1,27] are among the most famous tools used to analyze the security of a block cipher. But those cryptanalysis tools are heavily dependent on the S-boxes involved (the so-called active S-boxes) in the linear approximation or the differential characteristic as we traverse the SP-network. Theorem 1 implies that, using the linear layer L_K with a secret key K makes complex to follow a bit of state and its mutations through the network. Hence linear and differential cryptanalysis are unhelpful against SC.

Theorem 1. *If K is secret then after three iterations of L_K the position of a bit of state is no longer related to its original position but depends only on fixed unknown data (as they are determined by K).*

Proof (Sketch of the proof). We show that the term $i\prod_{r=0}^{t-1}(1-k_{p^r(i)})$ on the right-hand side of the equation given by lemma 1 vanishes after 3 iterations of L_K.

The positions $p^u(i)$ for $0 \leq u \leq t-1$ of bits of K are not independent but the corresponding bits are independent since all bits of K are drawn uniformly at random from $\{0,1\}$. Hence $1-k_{p^u(i)}$ for $0 \leq u \leq t-1$ can be considered as a random variable with equal probability of taking value 0 or 1. We know from [8] that when we draw uniformly at random t bits, the longest streak of consecutive 1 we expect to have is $\Theta(\log_2\ t)$. Therefore, for $t \geq 3$ there is necessarily some u in the set of integer $\{0,\cdots,t-1\}$ for which $1-k_{p^r(i)}=0$. □

Resistance to Algebraic Attacks. Algebraic attacks are known plaintext attacks. The adversary expresses the whole cipher as a system of multivariate algebraic equations and then tries to solve it using known plaintext-ciphertext pairs in order to recover the secret key. SC is a 2-round cipher that uses a 4-bit to 4-bit S-box and operates on 64-bit block. Each S-box can be described by 21 equations in 8 variables (4 inputs and 4 outputs). Therefore SC can be expressed as a system of by 672 multivariate equations in 256 variables. The number of equations is not impressive. However, the difficulty of using the algebraic attack against SC is that it will not be easy (as theorem 1 implies) to bind the input variables of the S-boxes of the first round to the bits of the plaintext, to bind the output variables of the S-boxes of the first round to the input variables of the S-boxes of the second round and to bind the output variables of the S-boxes of the second round to the bits of the ciphertext. Consequently algebraic attacks are not effective against SC.

5 Conclusion

In this paper we have presented R-MAC a new lightweight MAC authentication protocol. R-MAC takes advantage of secure constructions such as Mirror-Mac [28] and the extensively studied Xor-Cascade Encryption. It also incorporates the new and ultra-lightweight Small-Cipher (introduced in this paper) which is an SP-network with a keyed linear layer. This design choice of Small-Cipher makes difficult to perform linear and differential cryptanalysis and algebraic attacks.

References

1. Eli Biham and Adi Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, London, UK, UK, 1993.

2. Alex Biryukov and David Wagner. *Advanced Slide Attacks*, pages 589–606. Springer Berlin Heidelberg, Berlin, Heidelberg, 2000.
3. Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Advances in cryptology CRYPTO'93*, pages 278–291. Springer, 1994.
4. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin, and C. Vikkelsoe. *PRESENT: An Ultra-Lightweight Block Cipher*, pages 450–466. Springer Berlin Heidelberg, 2007.
5. Julia Borghoff, Anne Canteaut, Tim Güneysu, Elif Bilge Kavun, Miroslav Knezevic, Lars R. Knudsen, Gregor Leander, Ventzislav Nikov, Christof Paar, Christian Rechberger, Peter Rombouts, Søren S. Thomsen, and Tolga Yalçın. *PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications*, pages 208–225. Springer Berlin Heidelberg, 2012.
6. Shan Chen, Rodolphe Lampe, Jooyoung Lee, Yannick Seurin, and John Steinberger. *Minimizing the Two-Round Even-Mansour Cipher*, pages 39–56. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
7. Shan Chen and John Steinberger. *Tight Security Bounds for Key-Alternating Ciphers*, pages 327–350. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
8. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.
9. Joan Daemen. *Limitations of the Even-Mansour construction*, pages 495–498. Springer Berlin Heidelberg, Berlin, Heidelberg, 1993.
10. Itai Dinur, Orr Dunkelman, Nathan Keller, and Adi Shamir. *Cryptanalysis of Iterated Even-Mansour Schemes with Two Keys*, pages 439–457. Springer Berlin Heidelberg, Berlin, Heidelberg, 2014.
11. Yevgeniy Dodis, Eike Kiltz, Krzysztof Pietrzak, and Daniel Wichs. *Message Authentication, Revisited*, pages 355–374. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
12. Orr Dunkelman, Nathan Keller, and Adi Shamir. Minimalism in cryptography: The even-mansour scheme revisited. In *Proceedings of the 31st Annual International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'12, pages 336–354, Berlin, Heidelberg, 2012. Springer-Verlag.
13. Shimon Even and Yishay Mansour. A construction of a cipher from a single pseudorandom permutation. *Journal of Cryptology*, 10(3):151–161, Jun 1997.
14. Dmitry Frumkin and Adi Shamir. Un-trusted-HB: Security vulnerabilities of trusted-HB. *IACR Cryptology ePrint Archive*, page 44, 2009.
15. Peter Gaži. *Plain versus Randomized Cascading-Based Key-Length Extension for Block Ciphers*, pages 551–570. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
16. Peter Gaži and Stefano Tessaro. *Efficient and Optimally Secure Key-Length Extension for Block Ciphers via Randomized Cascading*, pages 63–80. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
17. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. *How to Encrypt with the LPN Problem*, pages 679–690. Springer Berlin Heidelberg, 2008.
18. Henri Gilbert, Matthew JB Robshaw, and Yannick Seurin. : Increasing the security and efficiency of HB^+. In *Advances in Cryptology–EUROCRYPT 2008*, pages 361–378. Springer, 2008.
19. Henri Gilbert, Matthew JB Robshaw, and Yannick Seurin. Good variants of HB^+ are hard to find. In *Financial Cryptography and Data Security*, pages 156–170. Springer, 2008.

20. Stefan Heyse, Eike Kiltz, Vadim Lyubashevsky, Christof Paar, and Krzysztof Pietrzak. *Lapin: An Efficient Authentication Protocol Based on Ring-LPN*, pages 346–365. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
21. Nicholas J Hopper and Manuel Blum. Secure human identification protocols. In *Advances in cryptology–ASIACRYPT 2001*, pages 52–66. Springer, 2001.
22. Ari Juels and Stephen A Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology–CRYPTO 2005*, pages 293–308. Springer, 2005.
23. Ahmad Khoureich Ka. hHB: A harder HB$^+$ protocol. In *SECRYPT 2015 - Proceedings of the 12th International Conference on Security and Cryptography*, pages 163–169, 2015.
24. Eike Kiltz, Krzysztof Pietrzak, Daniele Venturi, David Cash, and Abhishek Jain. Efficient authentication from hard learning problems. *Journal of Cryptology*, 30(4):1238–1275, Oct 2017.
25. Jooyoung Lee. *Towards Key-Length Extension with Optimal Security: Cascade Encryption and Xor-cascade Encryption*, pages 405–425. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.
26. Xuefei Leng, Keith Mayes, and Konstantinos Markantonakis. HB-MP$^+$ protocol: An improvement on the HB-MP protocol. In *IEEE International Conference on RFID 2008*, pages 118–124. IEEE, 2008.
27. Mitsuru Matsui. *Linear Cryptanalysis Method for DES Cipher*, pages 386–397. Springer Berlin Heidelberg, Berlin, Heidelberg, 1994.
28. Petros Mol and Stefano Tessaro. Secret-key authentication beyond the challenge-response paradigm: Definitional issues and new protocols. 2012.
29. Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB$^\#$ against a man-in-the-middle attack. In *Advances in Cryptology–ASIACRYPT 2008*, pages 108–124. Springer, 2008.
30. Pedro Peris-Lopez, Julio Cesar Hernandez-Castro, Juan ME Tapiador, and Arturo Ribagorda. Advances in ultralightweight cryptography for low-cost rfid tags: Gossamer protocol. In *International Workshop on Information Security Applications 2008*, pages 56–68. Springer, 2008.
31. Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In *Proceedings of the Thirty-seventh Annual ACM Symposium on Theory of Computing*, STOC '05, pages 84–93. ACM, 2005.
32. Panagiotis Rizomiliotis and Stefanos Gritzalis. GHB$^\#$: A provably secure HB-like lightweight authentication protocol. In *ACNS 2012*, pages 489–506. Springer, 2012.
33. Adi Shamir. Squash–a new mac with provable security properties for highly constrained devices such as rfid tags. In *Fast Software Encryption*, pages 144–157. Springer, 2008.