RMAC – A Lightweight Authentication Protocol for Highly Constrained IoT Devices

Ahmad Khoureich Ka

Department of Computer Science, University of Alioune Diop de Bambey,Senegal

Abstract. Nowadays, highly constrained IoT devices have earned an important place in our everyday lives. These devices mainly comprise RFID (Radio-Frequency IDentification) or WSN (Wireless Sensor Networks) components. Their adoption is growing in areas where data security or privacy or both must be guaranteed. Therefore, it is necessary to develop appropriate security solutions for these systems. Many papers have proposed solutions for encryption or authentication. But it turns out that sometimes the proposal has security flaw or is ill-suited for the constrained IoT devices (which has very limited processing and storage capacities). In this paper we introduce a new authentication protocol inspired by Mirror-Mac (MM) which is a generic construction of authentication protocol proposed by Mol *et al*. Our proposal named RMAC is well suited for highly constrained IoT devices since its implementation uses simple and lightweight algorithms. We also prove that RMAC is at least as secure as the MM protocol and thus secure against man-in-the-middle attacks.

Key words: IoT, MAC, authentication, lightweight protocol, Xor-Cascade Encryption.

1 Introduction

The Internet of Things (IoT), refers to a wide variety of devices that can collect, share data and more to connect to the internet. These devices mainly comprise RFID or WSN components [1]. Data security and privacy constitute one of the biggest issues with the IoT since extremely sensitive data can be collected and shared anonymously [1]. Therefore, in order to maintain the growing interest (and trust) in connected objects 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, classical cryptographic primitives require important processing and storage capacities that constrained IoT devices do not have [2]. Therefore, it is necessary to invent secure and lightweight solutions. But designing such lightweight protocols is not easy, since they must take into account security, hardware efficiency and energy consumption. Nevertheless, several solutions for lightweight authentication protocol have been proposed. We can mention HB-like protocols [3, 4, 5], Message Authentication Codes [6, 7, 8] exploiting the difficulty of the LPN problem and others [9, 10, 11, 12, 13, 14, 2].

In this paper, we propose a new 3-round MAC authentication protocol named RMAC (stands for Random MAC). One might ask why a new lightweight authentication protocol since there is a bunch of proposals. The answer is simply because many of the aforementioned proposals have been broken [15, 16, 17, 18, 19, 20] or have an storage and transmission cost unacceptably high [4, 6] for highly constrained IoT devices. This new protocol is inspired by the MM proposal [21], i.e. a 2-round authentication protocol based on weak MACs and provably secure against man-in-the-middle attacks. It also exploits the well-studied r-round Xor-Cascade Encryption, i.e. a framework for designing block ciphers.

The main idea behind the MM proposal is that the prover responds to the verifier only if the pair (M, τ_1) received is correct (see figure 1). Since the MAC is unforgeable under random-message attack (uf-rma), any modification of the pair (M, τ_1) stops the execution of the protocol. Therefore, the attacker has very little chance to have any information about the keys used in the authentication protocol. But the weakness of this method is that the attacker can use an isolated prover as a verification oracle because he can send a pair (M, τ_1) to the prover and if the latter responds he will know that the pair is good and at the same time he will have for free $\mathsf{TAG}_{K_2}(M)$ for an M of his choice. The RMAC protocol that we propose here eliminates this shortcoming.

The r-round Xor-Cascade Encryption is a framework for designing block ciphers from random permutations [22, 23, 24, 25]. RMAC implements a 2-round Xor-Cascade Encryption as a message authentication code. We know that the 2-round Xor-Cascade Encryption is secure up to $2^{\kappa+n/2}$ query complexity [22]. That is an adversary cannot gain useful

information about the key with less than $2^{\kappa+n/2}$ queries to both the 2-round Xor-Cascade Encryption itself and to its inner permutations. But this threshold can be easily reached when the attacker has at his disposal a genuine prover. Therefore, in order to overcome this weakness, we execute a key establishment protocol before the core authentication protocol to renew the key at each authentication session.

This paper is structured as follows. A brief introduction is done in section 1 followed by some definitions in section 2. Section 3 presents existing work. Our RMAC protocol is described in section 4 followed by the security arguments that weighs in its favor in section 5. Finally, a conclusion is given in section 6.

2 DEFINITIONS

2.1 r-round Xor-Cascade Encryption

The r-round Xor-Cascade Encryption (which can also be seen as a Generalized Even-Mansour Cipher[1]) can be considered as a framework for building block ciphers from a set of random permutations. Consider an ensemble $\mathcal{P} = \{P_i\}_{i\in\{0,1\}^\kappa}$ of random permutations of $\{0,1\}^n$, the r-round 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 $x \in \{0,1\}^n$,

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

A considerable number of papers [22, 23, 24, 25] 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 r-round Xor-Cascade Encryption approaches $2^{\kappa+n}$ when r is increasing. Also other papers explored attacks on these constructions [27, 28, 29, 30].

2.2 Message Authentication Codes

A message authentication code (MAC) is a triple of probabilistic polynomial-time algorithms (KGEN, TAG, VRFY) such that:

1. KGEN is the key generation algorithm. It takes as input a security parameter 1^n and outputs a key K from a specified keyspace $\mathcal{K}$.
2. TAG is the MAC tag generation algorithm (may be randomized). It takes as input a key K and a message M from a specified message space $\mathcal{M}$ and outputs a MAC tag $\tau \leftarrow \mathsf{TAG}_K(M)$.
3. VRFY is the verification algorithm (assumed to be deterministic). It takes as input a key K, a message M and a MAC tag τ and outputs a bit $b = \mathsf{VRFY}_K(M,\tau)$. If the TAG algorithm is a cipher as in our RMAC protocol, VRFY outputs 1 if $\tau = \mathsf{TAG}_K(M)$ or 0 otherwise.

The security of a MAC is related to its resistance against forgery. The strongest notion of MAC security is suf-cma, that is strongly unforgeable under chosen-message attack. It refers to MACs for which any adversary has negligible chance to generate a valid MAC tag for a new message (a message whose MAC tag is not previously seen by the adversary) even if she has seen MAC tags for messages of its choosing. Our RMAC protocol is based on a weaker MAC (uf-rma MAC) that is a MAC which is unforgeable only under random-message attack. That is, the MAC is unforgeable if the adversary does not have the ability to perform chosen-message attack. The adversary can only see MAC tags for random messages (messages for which she has no control).

2.3 MITM secure authentication protocol

Man-in-the-middle (MITM) attacks are the most powerful attacks against authentication protocols [6]. The MITM adversary is allowed to interact several times (at will and even concurrently) with the prover and the verifier. An authentication protocol achieves MITM security if any MITM adversary cannot bring the verifier to accept.

[1] The Generalized Even-Mansour Cipher is a generalization of the one-round Even-Mansour schema [26].

3 EXISTING WORK

A number of works has been done on lightweight authentication protocols. There is HB-like protocols [3, 4, 5] which take advantage of the difficulty of solving the learning parity with noise (LPN) problem. The probabilistic nature of the verifier's final response (accepting or rejecting the prover) in HB-like protocols is generally exploited to develop attacks against them [17, 18, 19]. In addition, despite their attractive design, which implies low computing resource requirements, their communication cost 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 [2] based on the Rabin encryption scheme, and others based on the LPN problem [6, 7, 31]. The proposals of Kiltz *et al.* [6] are MAC-based authentication protocols exploiting the difficulty of the LPN problem. These protocols have the advantage of having a tight reduction to the LPN problem and therefore secure against man-in-the-middle attacks. But they suffer from the large size of their keys and their large communication complexity. All these drawbacks make these protocols poorly suited for highly constrained IoT devices. More recent proposals have been made [9, 10, 11, 12] but it turns out that [9, 10] fail to achieve the claimed security level [15, 16]. The proposal of Mol *et al.* named Mirror-Mac (MM) [21] has caught our attention. MM is a generic construction of a 2-round MITM secure protocol (see figure 1). Mol *et al.* have proven that when instantiated with an uf-rma (unforgeable under random-message attack) 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 negligible chance to make the verifier to accept. Our proposal RMAC is inspired by MM.

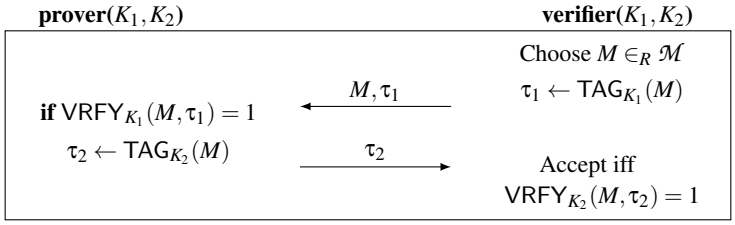

Fig. 1. The generic MM construction of a 2-round MITM secure protocol proposed by Mol *et al.* [21]. using MAC = (KGen,TAG,VRFY) where the keys K_1 and K_2 are generated by KGen, TAG takes as input a key K and a message M in a message space $\mathcal{M}$ and outputs $\mathsf{TAG}_K(M)$ and VRFY takes as input a key K, a message M and a MAC tag τ in the tag space $\mathcal{T}$ then outputs a decision $\mathsf{VRFY}_K(M, \tau) \in \{0, 1\}$.

4 THE NEW PROTOCOL

Traditionally, a MAC authentication is a 2 steps protocol. The verifier and the prover share a common secret K. The verifier sends a challenge M to the prover that calculates and returns the corresponding tag $\tau \leftarrow \mathsf{TAG}_{K_1}(M)$ to the verifier. The latter checks if the received tag is correct $\mathsf{VRFY}_{K_1}(M, \tau) = 1$ to accept the prover. For such a scheme to be secure it is necessary that the MAC be suf-cma since the adversary has a direct access to it and can make chosen-message attacks. Such MACs are generally constructed from pseudorandom functions unusable in a highly constrained environment. A solution for highly constrained IoT devices is to use a MAC consisting of lightweight algorithms (certainly less sure than conventional MACs) and embed it in a protocol from which the adversary will not have a direct access to make chosen-message attacks. Therefore, the adversary is forced to be passive. It's that solution that we implement with this new protocol (RMAC).

RMAC 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 $n = 64$. RMAC is based on the two-round generic construction of a man-in-the-middle secure authentication protocol denoted MM introduced by Mol *et al.* [21]. But in our protocol the challenge is encrypted using a one-time pad prior to sending it to the prover and whatever the result of the verification of the pair $(\mathsf{pfx}_n(\beta), \tau_1)$ by the prover an n-bit string is returned to the verifier (see figures 1 and 2 for a graphical comparison of MM and

RMAC). The RMAC protocol also uses a new implementation of a Xor-Cascade Encryption consisting of only two rounds as an uf-rma message authentication code. Using this small number of rounds guaranties low-latency and low-cost hardware implementation [32]. In the rest of this paper, we call Small-Cipher or SC that implementation of the 2-round Xor-Cascade Encryption.

We know from [22] that the 2-round Xor-Cascade Encryption is secure up to $2^{\kappa+n/2}$ query complexity. When such construction is implemented in highly constrained devices (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.

4.1 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 prover begins by drawing uniformly at random α from $\{0,1\}^{2n}$ and then sends it to the verifier. They both compute $S' = \mathsf{MixBits}(S,\alpha)$ where MixBits is a mixing function. After that, the verifier draws uniformly at random β from $\{0,1\}^{2n}$, computes $\beta' = \beta \oplus S'$ and sends the result to the prover. 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 [14] (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
/* wt denotes Hamming weight                                   */
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
```

4.2 Design Details of The Core Authentication Protocol

The prover and the verifier share a long-lived key S and two session keys K_1 and K_2 obtained from the lightweight key establishment protocol. The verifier draws a random $2n$-bit string β, computes $\beta' = \beta \oplus S'$ and $\tau_1 = \mathsf{SC}(\mathsf{pfx}_n(\beta))$ (where S' is computed during the session key establishment protocol, $\mathsf{pfx}_n(\beta)$ is the prefix of length n of β) then sends

(β', τ_1) to the prover. Upon receiving (β', τ_1), the prover checks whether τ_1 is equal to $\mathsf{SC}(\mathsf{pfx}_n(\beta))$, and if so, sends $\tau_2 = \mathsf{SC}(\tau_1)$ to the verifier, if not, sends a random n-bit string to the verifier. The latter accepts the prover if and only if $\tau_2 = \mathsf{SC}(\tau_1)$.

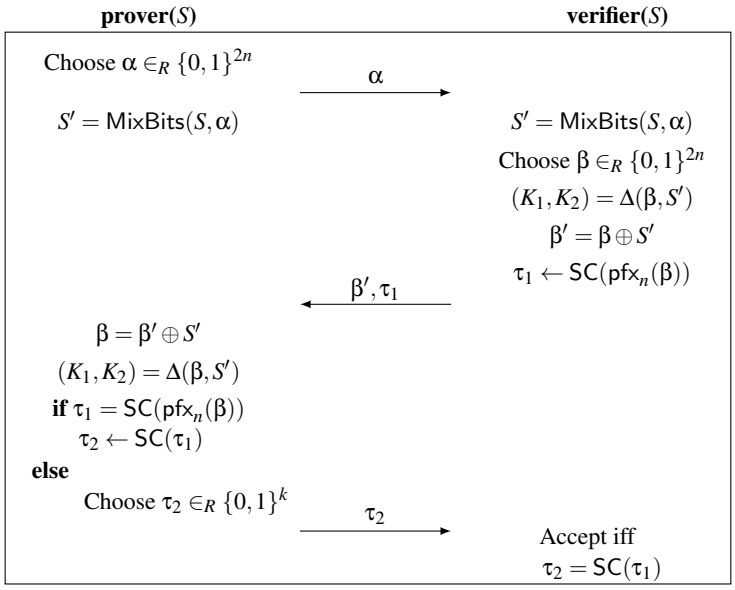

Fig. 2. The RMAC Authentication Protocol using SC with keys K_1, K_2, S_1 and S_2. $\mathsf{pfx}_n(\beta)$ is the prefix of length n of β.

Now we present our Small-Cipher[2] (SC) which is an extremely lightweight implementation of the 2-round Xor-Cascade Encryption. It is used as a message authentication code (MAC) by RMAC. Its inner permutation is an SP-network denoted RBOX (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 value of bits 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 state but here we introduce keyed one.

In the rest of this section, we successively present the S-box, the P-box, RBOX (the Small-Cipher inner permutation) 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 [33]. 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 P-box. 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 preserves 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.

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. 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.

[2] It must be clear that SC is used as an uf-rma MAC for our authentication protocol and we do not claim to offer it as block cipher for constrained environments.

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
```

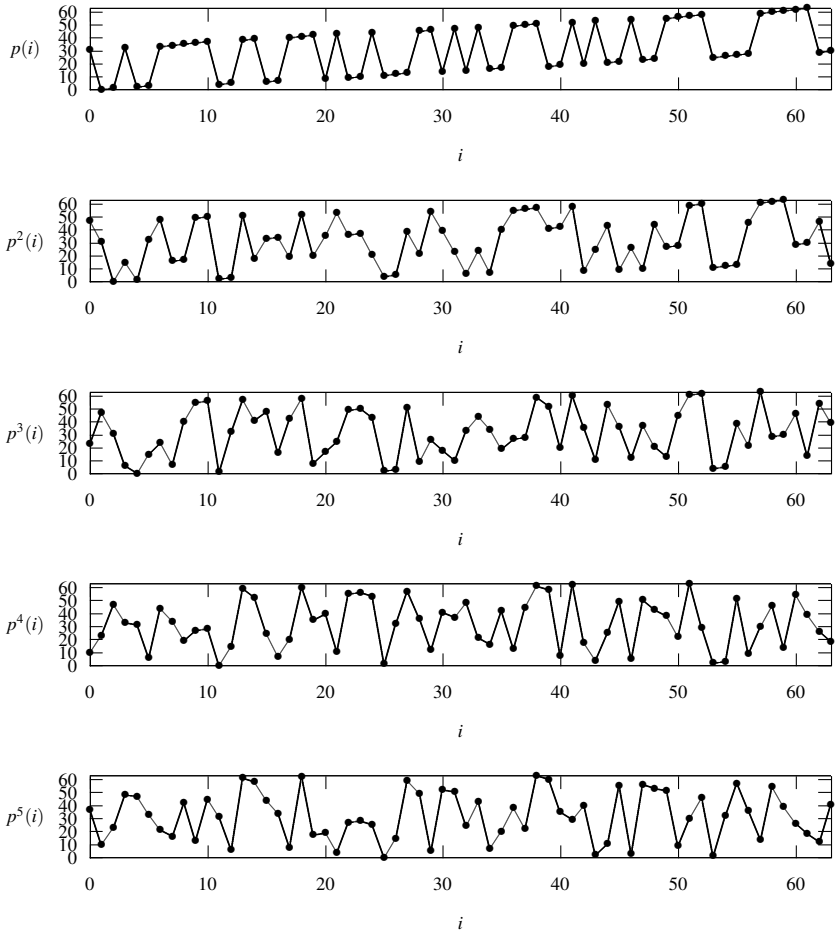

Fig. 3. Improvement of the diffusion power by iterating L_K where K is the 64-bit string $\texttt{93E6748D4E52787C}_{16}$. From top to bottom we have the 1st to the 5th iteration of L_K.

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.

The RBOX. 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 $\mathsf{RBOX}_{K_1K_2}$ 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 $\mathsf{RBOX}_{K_1K_2}$).

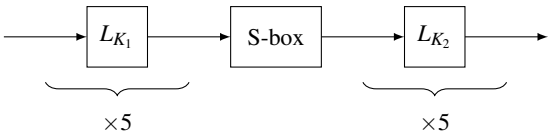

Fig. 4. $\mathsf{RBOX}_{K_1K_2}$. The P-boxes L_{K_1} and L_{K_2} are iterated five times.

The Small-Cipher (SC). In an initial phase, both the prover and the verifier hold the same $2n$-bit secret key S. Let $S_1 = \mathsf{pfx}_n(S)$ 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 the lightweight key establishment protocol the two parties obtained two n-bit session keys K_1 and K_2. Since it is only required in the iterated Even-Mansour cipher to have the 3-round keys to be 2-wise independent [34]. Then, by using the two independent n-bit permutations $\mathsf{RBOX}_{K_1S_1}$ and $\mathsf{RBOX}_{S_2K_2}$ and the round keys $(K_1, K_1 \oplus K_2, K_2)$ we have our implementation of the 2-round Xor-Cascade Encryption depicted in Figure 5.

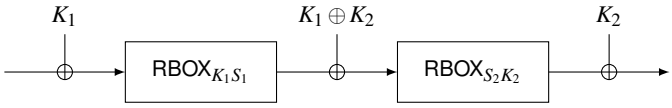

Fig. 5. Small-Cipher using keys K_1, K_2, S_1 and S_2

A comparison of RMAC with other authentication protocols is given in table 1.

Table 1. Storage and transmission cost of some authentication protocols. *Rabin crypto.* stands for Rabin cryptosystem, *qSDH* for q-Strong Diffie-Hellman, *PUF* for Physical Unclonable Function and *2-XC* for 2-round Xor Cascade Encryption. Values are given in bits.

Protocol	Based on	Parameters	Key storage	Transmission cost
HB^+ [3]	LPN	$K_x = 80; K_y = 512$	592	690252
$\mathrm{HB}^\#$ [4]	LPN	$K_X = 80; K_Y = 512; m = 1164$	2918	1756
SQUASH-128 [2]	Rabin crypto.	$K = 64$	64	96
$\mathrm{MM}_{\mathsf{qSDH}}$ [21]	qSDH	$(g,K) \in \mathbb{G} \times \mathbb{Z}_p$	?	$(M, \tau_1, \tau_2) \in \mathbb{Z}_p \times \mathbb{G}^2$
ETAP[11]	Dynamic Token	$a = b = 16, u = 10, v = 6$	288	64
H protocol[12]	PUF	P_n, FID, K_n	288	576
RMAC	2-XC	$S = 128$	128	384

5 SECURITY ARGUMENTS

In this section, we present the security analysis of SC and provide security proofs for RMAC.

5.1 Security of SC

The Structure of SC. SC is an implementation of the 2-round Xor-Cascade Encryption using RBOX as its inner permutation. The 2-round Xor-Cascade Encryption is secure up to $2^{\kappa+n/2}$ query complexity [22]. For SC, $\kappa = 2n$ thus

SC is theoretically secure up to $2^{5n/2}$ queries to the underlying RBOXes and to SC itself. This bound is theoretical since the RBOXes cannot be considered as random permutations. This is why it is advantageous to change the RBOXes keys at each execution of RMAC. Also, note that through RMAC it is difficult for an adversary to make a chosen plaintext attack (and even more difficult a chosen ciphertext attack) against SC.

Resistance to Linear and Differential Cryptanalysis. Linear and differential cryptanalysis [35, 36] 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 in the linear approximation or in the differential characteristic as we traverse the SP-network (the so-called active S-boxes). Theorem 1 implies that, using the iterated linear layer L_K with a secret key K makes complex to follow a bit of state through the SP-network. Therefore, it is very unlikely to determine the active S-boxes which are crucial to linear and differential cryptanalysis.

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.*

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

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.

For $t = 0$ we have $p^0(i) = i$. Thus equation 2 is true for $t = 0$.

Now we assume $t > 0$ and show that equation 2 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 2 is true, we have:

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

This final equation completes the proof.

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 $p^r(i)$ for $0 \leq r \leq t-1$ 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 $1-k_{p^r(i)}$ for $0 \leq r \leq t-1$ can be considered as a random variable with equal probability of taking value 0 or 1. We know from [37] 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 $1-k_{p^u(i)}=0$.

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 672 multivariate equations in 256 variables. The number of equations is not impressive. However, the difficulty of using an algebraic attack against SC relies on the fact 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, it is very unlikely that algebraic attack be effective against SC.

5.2 Security of RMAC

Security of The Lightweight Key Establishment Protocol Any modification on α will only change the way that 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 prover by the verifier. Since β is drawn uniformly at random from $\{0,1\}^{2n}$, the attacker derives no benefit from its actions on α.

Security of The Core Authentication Protocol The following theorem states that RMAC has the same resistance to MITM attacks as MM.

Theorem 2. *If MM instantiated with an uf-rma MAC is a man-in-the-middle secure authentication protocol, then RMAC instantiated with the same type of MAC is a man-in-the-middle secure authentication protocol.*

Proof. For this proof, we use the reduction technique. That is from an instance of the MM protocol we simulate an instance of the RMAC protocol. And show that if there is a MITM adversary that has a non-negligible advantage on RMAC, it can be used to mount a MITM attack against MM with a non-negligible success probability.

Now, let $\mathcal{A}$ be a MITM probabilistic polynomial-time adversary attacking RMAC with a non-negligible success probability. We construct a MITM probabilistic polynomial-time adversary $\mathcal{A}'$ that attempts to attack MM. So, from an instance of the MM protocol, $\mathcal{A}'$ simulates for $\mathcal{A}$ an instance of RMAC as follows:

1. $\mathcal{A}'$ begins by drawing uniformly at random α from $\{0,1\}^{2n}$ and sends it to $\mathcal{A}$.
2. Upon receiving (r,τ_1) from the MM verifier, $\mathcal{A}'$ draws uniformly at random r' from $\{0,1\}^n$ and sends $(r||r',\tau_1)$ to $\mathcal{A}$. Since in the RMAC protocol $\mathsf{pfx}_n(\beta')$ is unrelated to $\mathsf{sfx}_n(\beta')$ and β' is obtained from a one-time pad of two unknown bit strings, the view of $\mathcal{A}$ in this step is identically distributed to the view it has from the second step of RMAC.
3. $\mathcal{A}'$ finishes the simulation by forwarding τ_2 received from the MM prover to $\mathcal{A}$, or if it doesn't receive nothing, sends a uniformly and randomly selected n-bit string to $\mathcal{A}$. Since the MAC is uf-rma, the adversary is not allowed to see MAC tags for chosen messages. Therefore, it will not be able to distinguish $\tau_1 = \mathsf{TAG}_{K_1}(r)$ from $\tau_2 = \mathsf{TAG}_{K_1}(\tau_1)$. Thus, the view of $\mathcal{A}$ in this step is identically distributed to the view it has from the final step of RMAC.

We claim that this simulation is correct since the view of $\mathcal{A}$ when used as a sub-routine by $\mathcal{A}'$ is identically distributed to the view it has when it interacts directly with RMAC. In conclusion, $\mathcal{A}$ has the same advantage over RMAC as $\mathcal{A}'$ has over MM (which is negligible [21]).

Encapsulating the challenge in β' using the one-time pad encryption reduces the security requirements on SC. Therefore, even if the attacker succeeded in finding a valid message-tag pair (m, τ), it has very little chance to reach the next step of the protocol since it will not know how to encapsulate m in β'. Another aspect that reinforces the security of our protocol is that the prover returns a response to the verifier regardless the outcome of the verification $(\mathsf{pfx}_n(\beta), \tau_1)$. This prevents the attacker from using the RMAC prover as a verification oracle since it has no way of detecting a change in the behavior of the RMAC prover following the outcome of the verification of the pair (β', τ_1). This is a one more security element that our protocol RMAC 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 uf-rma MAC) at least as secure as MM which is a man-in-the-middle secure authentication protocol.

6 CONCLUSION

In this paper, we have presented RMAC a new lightweight MAC authentication protocol for highly constrained IoT devices. RMAC consists of ultra-lightweight algorithms and takes advantage — but also adds some extra steps — of the design of MM, a two-round generic construction secure against man-in-the-middle attacks introduced by Mol *et al*. It also uses a new implementation of the extensively studied 2-round Xor-Cascade Encryption as a message authentication code. Although SC is not intended to be a block cipher for resource constrained devices, we have shown that it is resistant to linear or differential cryptanalysis and to algebraic attacks. When viewed as a generic construction (SC replaced by an unforgeable under random-message attack MAC), RMAC is proven to be a secure MITM three rounds authentication protocol.

References

[1] L. David, R. Ammar, and M. Monique. "The Internet of Things". In: *The Internet Protocol Journal* 15 (2012), pp. 10–19. ISSN: 1944-1134. URL: https://www.cisco.com/c/dam/en_us/about/ac123/ac147/archived_issues/ipj_15-3/ipj_15-3.pdf.

[2] A. Shamir. "SQUASH – A New MAC with Provable Security Properties for Highly Constrained Devices Such as RFID Tags". In: *Fast Software Encryption*. Ed. by K. Nyberg. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 144–157. ISBN: 978-3-540-71039-4.

[3] A. Juels and S. A. Weis. "Authenticating Pervasive Devices with Human Protocols". In: *Advances in Cryptology – CRYPTO 2005*. Ed. by V. Shoup. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 293–308. ISBN: 978-3-540-31870-5.

[4] H. Gilbert, M. J. B. Robshaw, and Y. Seurin. "Increasing the Security and Efficiency of HB$^+$". In: *Advances in Cryptology – EUROCRYPT 2008*. Ed. by N. Smart. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 361–378. ISBN: 978-3-540-78967-3.

[5] P. Rizomiliotis and S. Gritzalis. "GHB#: A Provably Secure HB-Like Lightweight Authentication Protocol". In: *Applied Cryptography and Network Security*. Ed. by F. Bao, P. Samarati, and J. Zhou. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 489–506. ISBN: 978-3-642-31284-7.

[6] E. Kiltz et al. "Efficient Authentication from Hard Learning Problems". In: *Journal of Cryptology* 30.4 (Oct. 1, 2017), pp. 1238–1275. ISSN: 1432-1378. DOI: 10.1007/s00145-016-9247-3. URL: https://doi.org/10.1007/s00145-016-9247-3.

[7] S. Heyse et al. "Lapin: An Efficient Authentication Protocol Based on Ring-LPN". In: *Fast Software Encryption*. Ed. by A. Canteaut. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 346–365. ISBN: 978-3-642-34047-5.

[8] V. Lyubashevsky and D. Masny. "Man-in-the-Middle Secure Authentication Schemes from LPN and Weak PRFs". In: *Advances in Cryptology – CRYPTO 2013*. Ed. by R. Canetti and J. A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 308–325. ISBN: 978-3-642-40084-1.

[9] K. Fan et al. “Lightweight RFID Protocol for Medical Privacy Protection in IoT”. In: *IEEE Transactions on Industrial Informatics* 14.4 (Apr. 2018), pp. 1656–1665. ISSN: 1551-3203. DOI: 10.1109/TII.2018.2794996.

[10] D. Liu et al. *Compact-LWE: Enabling Practically Lightweight Public Key Encryption for Leveled IoT Device Authentication*. Cryptology ePrint Archive, Report 2017/685. 2017. URL: https://eprint.iacr.org/2017/685.

[11] M. Chen, S. Chen, and Y. Fang. “Lightweight Anonymous Authentication Protocols for RFID Systems”. In: *IEEE/ACM Transactions on Networking* 25.3 (June 2017), pp. 1475–1488. ISSN: 1063-6692. DOI: 10.1109/TNET.2016.2631517.

[12] H. Xu et al. “A Lightweight RFID Mutual Authentication Protocol Based on Physical Unclonable Function”. In: *Sensors* 18.3 (2018). ISSN: 1424-8220. DOI: 10.3390/s18030760. URL: http://www.mdpi.com/1424-8220/18/3/760.

[13] J. Y. Lee, W. C. Lin, and Y. H. Huang. “A lightweight authentication protocol for Internet of Things”. In: *2014 International Symposium on Next-Generation Electronics (ISNE)*. May 2014, pp. 1–2. DOI: 10.1109/ISNE.2014.6839375.

[14] P. Peris-Lopez et al. “Advances in Ultralightweight Cryptography for Low-Cost RFID Tags: Gossamer Protocol”. In: *Information Security Applications*. Ed. by K.-I. Chung, K. Sohn, and M. Yung. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 56–68. ISBN: 978-3-642-00306-6.

[15] S. F. Aghili and H. Mala. *Security Analysis of Fan et al. Lightweight RFID Authentication Protocol for Privacy Protection in IoT*. Cryptology ePrint Archive, Report 2018/388. 2018. URL: https://eprint.iacr.org/2018/388.

[16] D. Xiao and Y. Yu. “Cryptanalysis of Compact-LWE and Related Lightweight Public Key Encryption”. In: *Sec. and Commun. Netw.* 2018 (Mar. 2018). ISSN: 1939-0114. DOI: 10.1155/2018/4957045. URL: https://doi.org/10.1155/2018/4957045.

[17] H. Gilbert, M. J. B. Robshaw, and Y. Seurin. “Good Variants of HB + Are Hard to Find”. In: *Financial Cryptography and Data Security*. Ed. by G. Tsudik. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 156–170. ISBN: 978-3-540-85230-8.

[18] D. Frumkin and A. Shamir. *Un-Trusted-HB: Security Vulnerabilities of Trusted-HB*. Cryptology ePrint Archive, Report 2009/044. 2009. URL: https://eprint.iacr.org/2009/044.

[19] K. Ouafi, R. Overbeck, and S. Vaudenay. “On the Security of HB# against a Man-in-the-Middle Attack”. In: *Advances in Cryptology - ASIACRYPT 2008*. Ed. by J. Pieprzyk. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 108–124. ISBN: 978-3-540-89255-7.

[20] K. Ouafi and S. Vaudenay. “Smashing SQUASH-0”. In: *Advances in Cryptology - EUROCRYPT 2009*. Ed. by A. Joux. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 300–312. ISBN: 978-3-642-01001-9.

[21] P. Mol and S. Tessaro. “Secret-Key Authentication Beyond the Challenge-Response Paradigm: Definitional Issues and New Protocols”. In: 2012. URL: http://www.cs.ucsb.edu/~tessaro/papers/auth.pdf.

[22] P. Gaži and S. Tessaro. “Efficient and Optimally Secure Key-Length Extension for Block Ciphers via Randomized Cascading”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 63–80. ISBN: 978-3-642-29011-4.

[23] J. Lee. “Towards Key-Length Extension with Optimal Security: Cascade Encryption and Xor-cascade Encryption”. In: *Advances in Cryptology – EUROCRYPT 2013*. Ed. by T. Johansson and P. Q. Nguyen. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 405–425. ISBN: 978-3-642-38348-9.

[24] P. Gaži. “Plain versus Randomized Cascading-Based Key-Length Extension for Block Ciphers”. In: *Advances in Cryptology – CRYPTO 2013*. Ed. by R. Canetti and J. A. Garay. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 551–570. ISBN: 978-3-642-40041-4.

[25] S. Chen et al. “Minimizing the Two-Round Even-Mansour Cipher”. In: *Advances in Cryptology – CRYPTO 2014*. Ed. by J. A. Garay and R. Gennaro. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 39–56. ISBN: 978-3-662-44371-2.

[26] S. Even and Y. Mansour. “A construction of a cipher from a single pseudorandom permutation”. In: *Journal of Cryptology* 10.3 (June 1, 1997), pp. 151–161. ISSN: 1432-1378. DOI: 10.1007/s001459900025. URL: https://doi.org/10.1007/s001459900025.

[27] J. Daemen. “Limitations of the Even-Mansour construction”. In: *Advances in Cryptology — ASIACRYPT ’91*. Ed. by H. Imai, R. L. Rivest, and T. Matsumoto. Berlin, Heidelberg: Springer Berlin Heidelberg, 1993, pp. 495–498. ISBN: 978-3-540-48066-2.

[28] A. Biryukov and D. Wagner. “Advanced Slide Attacks”. In: *Advances in Cryptology — EUROCRYPT 2000*. Ed. by B. Preneel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 589–606. ISBN: 978-3-540-45539-4.

[29] O. Dunkelman, N. Keller, and A. Shamir. “Minimalism in Cryptography: The Even-Mansour Scheme Revisited”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 336–354. ISBN: 978-3-642-29011-4.

[30] I. Dinur et al. “Cryptanalysis of Iterated Even-Mansour Schemes with Two Keys”. In: *Advances in Cryptology – ASIACRYPT 2014*. Ed. by P. Sarkar and T. Iwata. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 439–457. ISBN: 978-3-662-45611-8.

[31] Y. Dodis et al. “Message Authentication, Revisited”. In: *Advances in Cryptology – EUROCRYPT 2012*. Ed. by D. Pointcheval and T. Johansson. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 355–374. ISBN: 978-3-642-29011-4.

[32] J. Borghoff et al. “PRINCE – A Low-Latency Block Cipher for Pervasive Computing Applications”. In: *Advances in Cryptology – ASIACRYPT 2012*. Ed. by X. Wang and K. Sako. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 208–225. ISBN: 978-3-642-34961-4.

[33] A. Bogdanov et al. “PRESENT: An Ultra-Lightweight Block Cipher”. In: *Cryptographic Hardware and Embedded Systems - CHES 2007*. Ed. by P. Paillier and I. Verbauwhede. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 450–466. ISBN: 978-3-540-74735-2.

[34] S. Chen and J. Steinberger. “Tight Security Bounds for Key-Alternating Ciphers”. In: *Advances in Cryptology – EUROCRYPT 2014*. Ed. by P. Q. Nguyen and E. Oswald. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 327–350. ISBN: 978-3-642-55220-5.

[35] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Berlin, Heidelberg: Springer-Verlag, 1993. ISBN: 0-387-97930-1.

[36] M. Matsui. “Linear Cryptanalysis Method for DES Cipher”. In: *Advances in Cryptology — EUROCRYPT ’93*. Ed. by T. Helleseth. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994, pp. 386–397. ISBN: 978-3-540-48285-7.

[37] T. H. Cormen et al. *Introduction to Algorithms, Third Edition*. 3rd. The MIT Press, 2009. ISBN: 0262033844, 9780262033848.