Exploiting Vulnerabilities of HB-MP

José Carrijo [1], **Rafael Tonicelli** [1]

[1]Department of Electrical Engineering, University of Brasilia
Campus Darcy Ribeiro, 70910-900, Brasilia, DF, Brazil

carrijo@redes.unb.br, tonicelli@redes.unb.br

Abstract. *HB-MP is a prominent member of the HB-family: a family of authentication protocols specially designed for RFID tags. We present two different cryptanalytic methods on HB-MP: (1) a passive attack based solely on the eavesdropping of legitimate authentication procedures; (2) an active attack, where the adversary has control over the RFID tag and is allowed to change the content of chosen memory areas of the device.*

1. Introduction

Recent years have witnessed unprecedented advances in mobile computing. A relevant advance in the field is the use of low-cost computing devices to perform cryptographic tasks. In this context, particular interest has been paid to providing authentication functionalities by means of RFID (Radio Frequency Identification) tags. RFID systems are composed of a reader, numerous tags, and a central database that stores the information about the objects identified by the tags. Important applications for those systems include: public transportation passes, pet identification, secure passport systems, anti-counterfeiting tags for medicines, warehouse inventory control, among others.

Despite all the potential applications, some weaknesses arise from the use of RFID tags. Since they are resource-constrained devices, they present reduced power and computational resources. Besides that, reader and tags communicate through wireless links, what makes them more vulnerable to both active and passive attacks. As a consequence, the development of lightweight cryptographic primitives suitable for those devices has become an active research area.

In 2001, Hopper and Blum [6] introduced the first authentication protocol appropriate for RFID tags, the so-called HB, which was based on a well-known intractability assumption: the LPN (Learning Parity with Noise) problem. The LPN problem is NP-complete and only requires the use of inner products and binary vectors, what makes it an attractive choice for constrained devices. This problem consists of determining a secret value $\mathbf{x}$ given "noisy" inner products of $\mathbf{x}$ with a sequence of randomly-chosen vectors. Subsequently, in 2005, Juels and Weis [7] proved that HB is only secure against a passive (eavesdropping) adversary. In order to provide resistance against active attacks, they developed HB^{+}, which was a modified version of HB.

Later, new attacks against the protocol HB^{+} have been added to the cryptanalytic literature. In [4], Gilbert *et al.* demonstrated that HB^{+} was vulnerable to certain man-in-the-middle attacks. Another key result was independently achieved by Carrijo *et al.* [2], and Golebiewski *et al.* [5]. They developed a probabilistic passive attack against HB^{+} that was efficient and required a reasonably feasible amount of captured transcripts.

Recently, Munilla and Peinado [9] proposed a new authentication protocol, named HB-MP, which is an enhancement of HB^{+}. Their intention was to overcome all the known weaknesses presented by the previous members of the HB-family without increasing its complexity. So, it was originally claimed that HB-MP could provide a more efficient performance and an increased resistance against the active attacks applied to the HB-family.

Contributions. To the best of the authors' knowledge, there are only two published results regarding the cryptanalysis of the protocol HB-MP. Gilbert *et al.* [3] showed that HB-MP is vulnerable to a passive impersonation attack, where an adversary who eavesdrops an entire authentication procedure is able to impersonate a legitimate RFID tag. The other cryptanalytic method is due to Leng *et al.* [8]. They presented an effective man-in-the-middle attack against HB-MP where the adversary utilizes the predictable rotation of the secret key. In this paper, we offer two alternative approaches on disrupting HB-MP:

- A flexible and entirely passive attack that actually recovers the shared secret keys. It presents a computational effort of 2^m, where m denotes the length of the challenges used.
- An active attack based on physical assumptions which presents increased efficiency. Remarkably, the amount of information and the computational complexity of this second attack are negligible. For instance, it allows one to recover a secret key of 2,048 bits by capturing only 29 complete authentication procedures.

Organization. This paper is organized as follows. In section 2, we briefly review the protocol HB-MP. In section 3, we outline the vulnerability of HB-MP. Section 4 describes the proposed passive attack. Section 5 describes the proposed active attack. Section 6 presents the performance results of our attacks. Section 7 presents HB-MP^{+}: an evolved and more resistant version of HB-MP.

2. Description of HB-MP

In this section, we present a brief description of the protocol HB-MP.

It is assumed that the tag and the reader communicate by means of a wireless insecure channel and share two secret keys, **x** and **y**, such that **x** and **y** $\in \{0,1\}^k$. We also introduce the following notation:

Symbol	Meaning
Rot(**w**,t)	bitwise left rotate operator, where the binary string **w** is rotated t positions.
xm	m-bit string composed of the m least significant bits of **x**. That is, **xm** $= \lfloor \mathbf{x} \rfloor_m$.
v_i	a noise bit, such that $Pr[v_i = 1] = \eta$, where $\eta \in \left[0, \frac{1}{2}\right]$.

The protocol is as follows:

Protocol HB-MP

1. For $i = 1$ to r
   (a) The reader randomly chooses a m-bit string $\mathbf{a}_i \in \{0,1\}^m$, and sends it to the tag.
   (b) The tag computes $\mathbf{x} = Rot(\mathbf{x}, \mathbf{y}[i])$, where $\mathbf{y}[i]$ denotes the i-th bit of the binary string $\mathbf{y}$.
   (c) The tag computes $z_i = \mathbf{a}_i \odot \mathbf{xm} \oplus v_i$, where $\odot$ is the inner product, v_i is the noise bit value at round i, and $\oplus$ is the bitwise XOR operation.
   (d) The tag finds a m-bit string, namely $\mathbf{b}_i$, such that $z_i = \mathbf{b}_i \odot \mathbf{xm}$. Once found, the value $\mathbf{b}_i$ is sent to the reader by the tag.
   (e) The reader computes the secret key used in this round as $\mathbf{x} = Rot(\mathbf{x}, \mathbf{y}[i])$.
   (f) The reader computes $z_i = \mathbf{b}_i \odot \mathbf{xm}$ and compares it with $z_i^* = \mathbf{a}_i \odot \mathbf{xm}$.
2. The reader accepts the authentication as valid if $z_i^* \neq z_i$ in less than ηr rounds.

3. Predictable Rotations: a vulnerability on HB-MP

Protocol HB-MP was specially designed to resist against man-in-the-middle attacks. In order to prevent such attacks, the key $\mathbf{x}$ is rotated in function of the key $\mathbf{y}$ and HB-MP can be viewed as an authentication protocol that uses a different key $\mathbf{xm}_i$ in each round i. However, the structure of HB-MP hides an implicit vulnerability: for each new authentication procedure, the secret key $\mathbf{xm}_i$ used in the i-th round is the same.

The explanation for this fact resides in the synchronization problem. For instance, suppose that a reader previously authenticated a tag in a r-round protocol execution. As a result, the key $\mathbf{x}$ was rotated in function of the first r bits of the key $\mathbf{y}$. If a new tag enters the electromagnetic field of the reader, before the new authentication procedure begins, the reader must restart, since the new tag does not have access to the updated value of $\mathbf{x}$.

As a result, in a cryptanalytic method, an adversary can eavesdrop n legitimate authentication procedures, each of them composed of r rounds, and build a matrix $\mathbf{T}$ as illustrated below. The term $(\mathbf{a}_i^u, \mathbf{b}_i^u)$ denote the challenge/response pair captured in the i-th round of the u-th authentication procedure.

$$\mathbf{T} = \begin{pmatrix} (\mathbf{a}_1^1, \mathbf{b}_1^1) & (\mathbf{a}_2^1, \mathbf{b}_2^1) & (\mathbf{a}_3^1, \mathbf{b}_3^1) & \cdots & (\mathbf{a}_i^1, \mathbf{b}_i^1) & \cdots & (\mathbf{a}_r^1, \mathbf{b}_r^1) \\ (\mathbf{a}_1^2, \mathbf{b}_1^2) & (\mathbf{a}_2^2, \mathbf{b}_2^2) & (\mathbf{a}_3^2, \mathbf{b}_3^2) & \cdots & (\mathbf{a}_i^2, \mathbf{b}_i^2) & \cdots & (\mathbf{a}_r^2, \mathbf{b}_r^2) \\ (\mathbf{a}_1^3, \mathbf{b}_1^3) & (\mathbf{a}_2^3, \mathbf{b}_2^3) & (\mathbf{a}_3^3, \mathbf{b}_3^3) & \cdots & (\mathbf{a}_i^3, \mathbf{b}_i^3) & \cdots & (\mathbf{a}_r^3, \mathbf{b}_r^3) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \underbrace{(\mathbf{a}_1^n, \mathbf{b}_1^n)}_{1\text{-st round}} & (\mathbf{a}_2^n, \mathbf{b}_2^n) & (\mathbf{a}_3^n, \mathbf{b}_3^n) & \cdots & \underbrace{(\mathbf{a}_i^n, \mathbf{b}_i^n)}_{i\text{-th round}} & \cdots & (\mathbf{a}_r^n, \mathbf{b}_r^n) \end{pmatrix}$$

Thus, the column vector $\mathbf{T}[1, \ldots, n][i] = [(\mathbf{a}_i^u, \mathbf{b}_i^u)]_{u=1}^n$ can be used by the cryptanalyst to derive secret information on $\mathbf{xm}_i$ (the key used in round i).

4. Description of the Passive Attack

Since HB-MP makes use of two secret keys with length k, a brute force attack has a computational complexity of 2^{2k}. It is known that, in r rounds, the key $\mathbf{x}$ is rotated in function of the first r bits of the key $\mathbf{y}$. Thus, for $k \geq m + r$, we effectively use

$$\begin{cases} r \text{ bits of the key } \mathbf{y}. \\ m + Hwt(\lfloor \mathbf{y} \rfloor_r) \text{ bits of the key } \mathbf{x}, \text{ where } 0 \leq Hwt(\lfloor \mathbf{y} \rfloor_r) \leq r. \end{cases}$$

The term $Hwt(\mathbf{c})$ denotes the Hamming weight of the string $\mathbf{c}$.

We also remark that it doe not matter for the cryptanalyst whether there was a rotation in the first round. I.e., from the cryptanalytic point of view, we are not concerned on finding the bit $\mathbf{y}[1]$ unless $r > k$. The computational complexity of a brute force attack is then reduced to 2^{m+r-1}. Our cryptanalytic method presents a complexity of 2^m.

In order to understand our passive attack, one should recall the Law of Large Numbers and also notice that the generation of noise bits v_i^u corresponds to the execution of i.i.d. (independent identically distributed) random variables V_i^u. Each of this random variables is described by a Bernoulli distribution Ber_η. Thus, according to the Law of Large Numbers, for a fixed i, with overwhelming probability, $\frac{1}{n} \sum_{u=1}^{n} V_i^u \rightarrow \eta$ with n large. Equivalently,

$$\lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} V_i^u - \eta\right| \geq \varepsilon\right) = 0 \quad \text{and} \quad \lim_{n\to\infty} P\left(\left|\frac{1}{n}\sum_{i=1}^{n} V_i^u - \eta\right| < \varepsilon\right) = 1$$

Let the terms $\mathbf{z}(\cdot,\cdot)$ and $\mathbf{z}^*(\cdot,\cdot)$ be defined as two-variable functions such that $\mathbf{z}\left(\mathbf{s}, i\right) = \left(\mathbf{b}_i^u \odot \mathbf{s}\right)_{u=1}^n$ and $\mathbf{z}^*\left(\mathbf{s}, i\right) = \left(\mathbf{a}_i^u \odot \mathbf{s}\right)_{u=1}^n$, where $\mathbf{s}$ is a binary string with length m and i is the index of a given round. Their vector representation is given below.

$$\mathbf{z}(\mathbf{s}, i) = \begin{bmatrix} \mathbf{b}_i^1 \odot \mathbf{s} \\ \vdots \\ \mathbf{b}_i^n \odot \mathbf{s} \end{bmatrix} \quad \text{and} \quad \mathbf{z}^*(\mathbf{s}, i) = \begin{bmatrix} \mathbf{a}_i^1 \odot \mathbf{s} \\ \vdots \\ \mathbf{a}_i^n \odot \mathbf{s} \end{bmatrix}$$

According to the previous observation, if $\mathbf{s}$ is the key used in the i-th round of the several authentication procedures, then, with overwhelming probability, the strings $\mathbf{z}(\mathbf{s}, i)$ and $\mathbf{z}^*(\mathbf{s}, i)$ will, on average, differ in approximately ηn positions. Analogously, $Hwt[\mathbf{z}^*(\mathbf{s}, i) \oplus \mathbf{z}(\mathbf{s}, i)] \approx \eta n$.

We assume that the attacker previously eavesdropped n ($n > m$) successful authentication procedures of r rounds and later built a matrix of transcripts $\mathbf{T}$. At first (**algorithm 1**), the attacker tests the universe of possible keys and select those that present the expected statistical behavior of a potential key. After finding a initial set of potential keys, the cryptanalyst iteratively tests each one of these (**algorithm 2**). For each potential key, there are three possibilities: (1) no rotation happened; (2) a bit 0 was rotated; (3) a bit 1 was rotated. If neither of these possibilities was satisfied, the key is discarded.

Assume that the symbol "||" denote the concatenation operator (for instance, $\mathbf{x}||bit = \{\mathbf{x}[n], \ldots, \mathbf{x}[1], bit\}$). Also consider an arbitrarily small parameter ϵ that can be adjusted by the attacker. The cryptanalytic method is as follows.

Algorithm 1 Initial Set of Potential Keys

Require: matrix of transcripts $\mathbf{T}[1,\ldots,n][1]$.
Ensure: two sets of potential keys, $\mathcal{X}$ and $\mathcal{Y}$.

for $k = 1$ to $2^m - 1$ **do**
  $\mathbf{x} \leftarrow Binary(k)$

  if $Hwt\left[\mathbf{z}^*\left(\mathbf{x},1\right) \oplus \mathbf{z}\left(\mathbf{x},1\right)\right] = \eta n + \epsilon$ **then**
    Include $\mathbf{x}$ into a set of potential keys $\mathcal{X}$.
    Create an associate key $\mathbf{y} \in \mathcal{Y}$.
  end if

end for

return the sets $\mathcal{X}$ and $\mathcal{Y}$

Algorithm 2 Determination of the Key Pair

Require: matrix of transcripts $\mathbf{T}[1,\ldots,n][2,\ldots,r]$ and two sets of potential keys, $\mathcal{X}$ and $\mathcal{Y}$.
Ensure: the key pair.

for $i = 2$ to r **do**
  for all $\mathbf{x} \in \mathcal{X}$ **do**
    if $Hwt\left[\mathbf{z}^*\left(\mathbf{x},i\right) \oplus \mathbf{z}\left(\mathbf{x},i\right)\right] = \eta n + \epsilon$ **then**
      Keep $\mathbf{x}$ in the set $\mathcal{X}$.
      $\mathbf{y}[i] \leftarrow 0$.

    else if $Hwt\left[\mathbf{z}^*\left(\lfloor \mathbf{x}||0 \rfloor_m, i\right) \oplus \mathbf{z}\left(\lfloor \mathbf{x}||0 \rfloor_m, i\right)\right] = \eta n + \epsilon$ **then**
      $\mathbf{x} \leftarrow \mathbf{x}||0$.
      $\mathbf{y}[i] \leftarrow 1$.

    else if $Hwt\left[\mathbf{z}^*\left(\lfloor \mathbf{x}||1 \rfloor_m, i\right) \oplus \mathbf{z}\left(\lfloor \mathbf{x}||1 \rfloor_m, i\right)\right] = \eta n + \epsilon$ **then**
      $\mathbf{x} \leftarrow \mathbf{x}||1$.
      $\mathbf{y}[i] \leftarrow 1$.

    else
      Discard $\mathbf{x}$ from $\mathcal{X}$
      Discard $\mathbf{y}$ from $\mathcal{Y}$
    end if

  end for
end for

return the the key pair $(\mathbf{x}, \mathbf{y})$.

5. Description of the Active Attack

This section describes our active attack against HB-MP. The cryptanalytic method here presented is based on a new paradigm. This new model of attack relies on some simple physical assumptions about the level of control that the adversary has over the device. By admitting some additional functionalities to the adversary, we demonstrate that it is possible to develop an extremely efficient attack against HB-MP.

The assumptions we make are very simple and feasible on a realistic scenario. They are as follows:

- The adversary is allowed to eavesdrop n authentication procedures, each one of them composed of r rounds. As before, we use $(\mathbf{a}_i^u, \mathbf{b}_i^u)$ to denote the transcript captured in the i-th round of the u-th authentication procedure.
- The adversary has the capability of modifying chosen memory areas of the RFID tag. Specifically, we admit that he/she can insert a binary string into the memory area designated for the binary strings $\mathbf{b}_i^u$.
- The adversary is able to operate the RFID tag as many times he/she needs.

5.1. Main Idea behind our active attack

In this part, we give the intuition behind our active attack.

At a first insight one should notice that:

$$\mathbf{b}_i^u \odot \mathbf{xm}_i = \mathbf{a}_i^u \odot \mathbf{xm}_i \oplus v_i^u.$$

By adding up the term $\mathbf{b}_i^u \odot \mathbf{xm}_i$ on both sides of the equality, we obtain:

$$(\mathbf{a}_i^u \oplus \mathbf{b}_i^u) \odot \mathbf{xm}_i \oplus v_i^u = 0.$$

Developing the left side of the equality, we achieve:

$$\left(\bigoplus_{j=1}^{m} \mathbf{a}_i^u[j] \cdot \mathbf{b}_i^u[j] \cdot \mathbf{xm}_i[j] \right) \oplus v_i^u = 0.$$

One can easily observe that, if $\mathbf{b}_i^u[j] = 1$, then:

$$\underbrace{(1 \cdot \mathbf{b}_i^u[j])}_{1} \cdot \mathbf{xm}_i[j] = \underbrace{(0 \cdot \mathbf{b}_i^u[j])}_{0} \cdot \mathbf{xm}_i[j] \quad \text{if and only if} \quad \mathbf{xm}_i[j] = 0.$$

Conclusion: Let $\mathbf{a}_i^u[j] = \mathbf{b}_i^u[j] = 1$. If the bit $\mathbf{xm}_i[j] = 0$, the action of flipping the bit $\mathbf{a}_i^u[j]$ to zero does not interfere in the authentication process. Otherwise (if $\mathbf{xm}_i[j] = 1$), the action of flipping the bit $\mathbf{a}_i^u[j]$ to zero will cause interference in the authentication process.

In the light of the previous conclusion, we define an algorithm that flips the i-th bit of a given string (**algorithm 3**) and present a brief sketch of our active attack.

Algorithm 3 FLIP$_i(\mathbf{w}, bit)$: Flip the i-th bit of the string $\mathbf{w}$ to bit, such that $bit \in \{0,1\}$.
Require: $\mathbf{w}$ such that $\mathbf{w} \in \{0,1\}^m$.
Ensure: $\overline{\mathbf{w}}$, such that $\overline{\mathbf{w}}[i] = bit$, and for all $j \neq i$, $\overline{\mathbf{w}}[j] = \mathbf{w}[j]$.

Sketch of the Active Attack: Determination of $\mathbf{xm}_i$

Inputs: A matrix of transcripts **T**.

Output: The key $\mathbf{xm}_i$ used in the i-th round.

STEP 1 Compute $u \leftarrow 1$.

STEP 2 For a fixed i, compute $\mathbf{c} \leftarrow \mathbf{a}_i^u \cdot \mathbf{b}_i^u$ in order to know the bit-positions where both $\mathbf{a}_i^u$ and $\mathbf{b}_i^u$ equal 1.

STEP 3 If $c[j] = 1$ and the j-th bit of the key $\mathbf{xm}_i$ is unknown, compute $\overline{\mathbf{a}}_i^u \leftarrow \text{FLIP}_j(\mathbf{a}_i^u, 0)$ and execute steps **A** to **D** for λ times.

A Send the modified challenge $\overline{\mathbf{a}}_i^u$ to the tag.

B After receiving the modified challenge, the tag initiates the authentication procedure. So, the tag will look for a m-bit binary string $\overline{\mathbf{b}}_i^u$ such that $\overline{\mathbf{b}}_i^u \odot \mathbf{xm}_i = \overline{\mathbf{a}}_i^u \odot \mathbf{xm}_i \oplus v_i^u$.

C By this time, insert into the tag's memory the original value $\mathbf{b}_i^u$.

D The tag sends to the adversary $\overline{\mathbf{b}}_i^u$. After receiving the response, compare $\mathbf{b}_i^u$ and $\overline{\mathbf{b}}_i^u$.

STEP 4 If $|\mathbf{b}_i^u = \overline{\mathbf{b}}_i^u| \approx (1-\eta)\lambda$, compute $\mathbf{xm}_i[j] \leftarrow 0$. Else, compute $\mathbf{xm}_i[j] \leftarrow 1$.

STEP 5 If there are yet bits to be determined, repeat all the process (**Steps 2** to **4**) for $u \leftarrow u + 1$.

As can be seen in our sketch, we assume that the adversary previously eavesdropped n authentication procedures. After acquiring a collection of transcripts $(\mathbf{a}_i^u, \mathbf{b}_i^u)$ and building a matrix **T**, the adversary can interact with the tag in the following way:

- He/she finds a bit position j where $\mathbf{a}_i^u[j] = \mathbf{b}_i^u[j] = 1$. Then, the adversary flips to zero the j-th bit of $\mathbf{a}_i^u$ and sends its modified version $\overline{\mathbf{a}}_i^u$ to the tag.

- The adversary injects $\mathbf{b}_i^u$ into the tag.
- After receiving $\overline{\mathbf{a}}_i^u$, the tag looks for a binary string $\overline{\mathbf{b}}_i^u$ such that $\overline{\mathbf{b}}_i^u \odot \mathbf{xm}_i = \overline{\mathbf{a}}_i^u \odot \mathbf{xm}_i \oplus v_i^u$. As a consequence of the injection, the first string to be tested by the tag will be the string $\mathbf{b}_i^u$.
- The tag sends to the adversary the string $\overline{\mathbf{b}}_i^u$. The adversary compares $\mathbf{b}_i^u$ and $\overline{\mathbf{b}}_i^u$.
- This process is repeated for λ times. If $|\mathbf{b}_i^u = \overline{\mathbf{b}}_i^u| \approx (1-\eta)\lambda$, the action of flipping the j-th bit of $\mathbf{a}_i^u$ did not affect the authentication procedure and $\mathbf{xm}_i[j] = 0$ with high probability. Otherwise, $\mathbf{xm}_i[j] = 1$ with high probability.

5.2. Active Attack

We now present our active attack by means of a detailed set of pseudocodes.

Algorithm 4 presents the procedure ExtractBit( ), which receives as one of its inputs the collection of transcripts $\{(\mathbf{a}_i^u, \mathbf{b}_i^u)\}$ and operates on a matrix $\mathbf{X}$. This matrix stores at its i-th row the key $\mathbf{xm}_i$ used in the i-th round. The values ϵ and λ are error and accuracy parameters, respectively. The greater the value of λ, the more accurate is the determination of the key $\mathbf{x}$.

Algorithm 4 ExtractBit($\mathbf{X}, (\mathbf{a}_i^u, \mathbf{b}_i^u), j, \epsilon, \lambda$): Recover j-th bit of the secret key $\mathbf{xm}_i$ used in round i.

Require: A $r \times m$ matrix $\mathbf{X}$ that stores in the i-th row the value of the key $\mathbf{xm}_i$ used in the i-th round. A collection of transcripts $(\mathbf{a}_i^u, \mathbf{b}_i^u)$. An error parameter ϵ. An accuracy parameter λ. The position value j / $j \in [1, m]$.

Ensure: The j-th bit of the secret key used in round i, namely $\mathbf{xm}_i[j]$.

$\overline{\mathbf{a}}_i^u \leftarrow \text{FLIP}_j(\mathbf{a}_i^u, 0)$
$counter \leftarrow 0$

for $s = 1$ to λ **do**
  Send $\overline{\mathbf{a}}_i^u$ to the tag
  Inject $\mathbf{b}_i^u$ into the tag
  $\overline{\mathbf{b}}_i^u \leftarrow \text{RESPONSE}(\overline{\mathbf{a}}_i^u)$
  if $\mathbf{b}_i^u = \overline{\mathbf{b}}_i^u$ **then**
    $counter \leftarrow counter + 1$
  end if
end for

if $counter = \lambda(1 - \eta) + \epsilon$ **then**
  $\mathbf{X}[i][j] \leftarrow 0$
else
  $\mathbf{X}[i][j] \leftarrow 1$
end if

Algorithm 5 presents the main function of our attack: BreakHBmp( ). This algorithm outputs the collection of keys $\mathbf{xm}_i$ used in each round.

Algorithm 5 BreakHBmp($(\mathbf{a}_i^u, \mathbf{b}_i^u), \epsilon, \lambda$): Provide a $r \times m$ matrix $\mathbf{X}$ that stores the r keys $\mathbf{xm}_i$ used in the r rounds.

Require: A collection of transcripts $(\mathbf{a}_i^u, \mathbf{b}_i^u)$ / $1 \leq i \leq r$ and $1 \leq u \leq n$. An error parameter ϵ. A precision parameter λ.

Ensure: The j-th bit of the secret key used at round i, namely $\mathbf{xm}_i[j]$.

```
Create a r × m matrix X

for i = 1 to r do
   UnknownBits ← {1, 2, ..., m}
   u ← 1
   while UnknownBits ≠ ∅ do
      for j = 1 to m do
         c[j] ← a_i^u[j] · b_i^u[j]
         if (c[j] = 1) and (j ∈ UnknownBits) then
            ExtractBit(X, (a_i^u, b_i^u), j, ε, λ)
            Discard element j from UnknownBits
         end if
         u ← u + 1
      end for
   end while
end for

return (X)
```

The method for recovering the key **y** is explained in **algorithm 6**.

Algorithm 6 Determination of the Key **y**

Require: The collection of round keys $\mathbf{xm}_i, i = 1, \ldots, r$.
Ensure: The secret key **y**.

```
for i = 1 to i = r do
   if xm_i = xm_{i+1} then
      y[i + 1] ← 0
   else
      y[i + 1] ← 1
   end if
end for

return the the key y.
```

6. Performance Results

In this section, we investigate the performance of the proposed attacks.

At first, we analyze the passive attack on HB-MP. The protocol HB-MP can be viewed as an authentication process that makes use of r different keys of m bits, one key per round. The method consists of generating all the possible m-bit strings and observing their statistical behavior. The keys that present the desired statistical behavior are included into a set of candidate keys. At each round of our iterative attack, we make this set more precise until the determination of the key pair (**x**,**y**). As can be seen, the determination of the initial set of potential key is the part that most demands computational effort. Consequently, one may dramatically reduce the complexity of our attack by using some alternative method to generate the initial set of probable keys. For instance, one could combine the passive and active attacks here presented. The attacker can use the first challenge/response pairs of each authentication, $\{(\mathbf{a}_1^u, \mathbf{b}_1^u)\}_{u=1}^n$, to recover the key $\mathbf{xm}_1$ used in the first round and then use the passive attack to recover the remaining bits of **x** and the key **y**. This fact proves the flexibility of the passive attack. We also remark that a suitable amount of protocol samples to apply the passive attack is $2k$ complete authentication procedures.

Regarding the active attack, we analyze the number of required authentication procedures to recover the secret key pair. In order to evaluate the feasibility of this attack, we simulated it on a computer system. On table 1, we present the relation between m and the average number of protocol samples that are necessary to determine the pair of secret keys. This table was achieved by running our simulation program on a conventional computer system.

Key Length in bits (m)	**Required Amount of Protocol Samples** ($\rho(m)$)
128	19
256	21
512	24
1,024	26
2,048	29
8,192	33

Table 1. Relation between the key length m and the average amount of protocol samples $\rho(m)$ needed to perform the active attack.

Clearly, the required number of authentication procedures is a function of m (the length of each round key), i.e., $n = \rho(m)$. Remarkably, for $m =$ 8,192 bits, on average, the attacker only needs 33 authentication procedures to recover the secret key pair.

7. How to Prevent such Attacks

We have shown that the predictability of the round keys used in the protocol HB-MP constitutes an important weakness of this protocol. Since HB-MP uses deterministic keys, an eavesdropper may intercept various protocol executions and collect a considerable amount of protocol samples, all of them related to the same keys.

A good strategy for preventing the attack is to add randomness to the computation of the round key. Remarkably, a recent member of the HB-family satisfies this require-

ment: HB-MP$^+$ designed by Leng *et al.* [8], whose original intention was to defend against GRS-MiTM attacks [4]. The core idea of this authentication protocol is to use random round keys by means of a one-way function as a way of defending against GRS-MiTM attacks. This new feature also makes the protocol resistant against the previously described attacks.

Define the binary string $\mathbf{x} \in \{0,1\}^k$ as a secret key shared between tag and reader. Also assume that tag and reader know a non-linear one-way function $f(\cdot)$ such that $f : \{0,1\}^m \times \{0,1\}^k \rightarrow \{0,1\}^m$. The protocol HB-MP$^+$ is as follows:

Protocol HB-MP$^+$

1. For $i = 1$ to r
   (a) The reader randomly chooses a m-bit string $\mathbf{a}_i \in \{0,1\}^m$, and sends it to the tag.
   (b) The tag computes the round key $\mathbf{x}_s = f(\mathbf{a}_i, \mathbf{x})$.
   (c) The tag computes $z_i = \mathbf{a}_i \odot \mathbf{x}_s \oplus v_i$.
   (d) The tag finds a m-bit string, namely $\mathbf{b}_i$, such that $z_i = \mathbf{b}_i \odot \mathbf{x}_s$. Once found, the value $\mathbf{b}_i$ is sent to the reader by the tag.
   (e) The reader computes the secret key used in this round as $\mathbf{x}_s = f(\mathbf{a}_i, \mathbf{x})$.
   (f) The reader computes $z_i = \mathbf{b}_i \odot \mathbf{x}_s$ and compares it with $z_i^* = \mathbf{a}_i \odot \mathbf{x}_s$.
2. The reader accepts the authentication as valid if $z_i^* \neq z_i$ in less than ηr rounds.

Figure 1 illustrates the protocol.

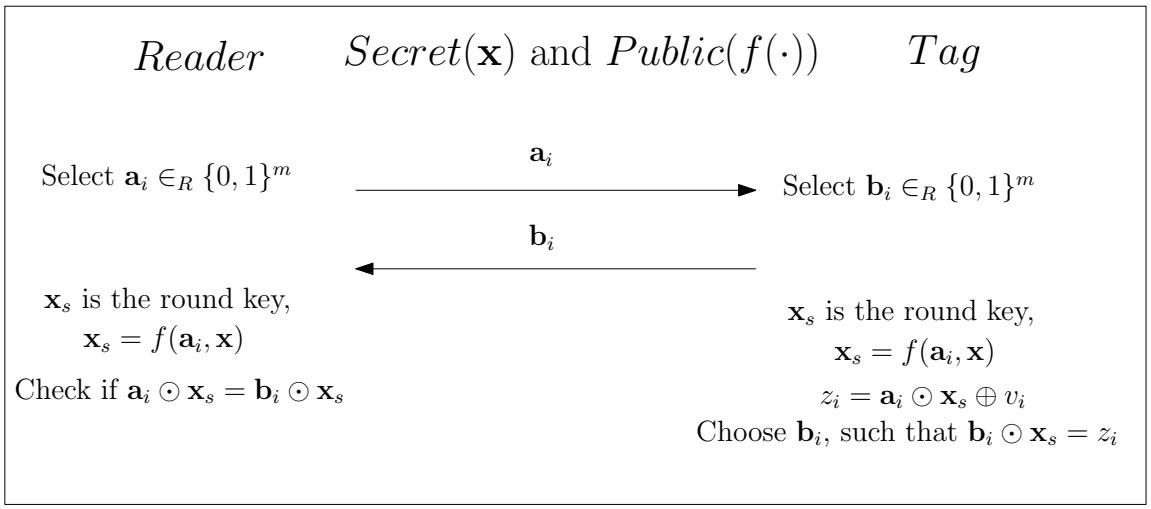

Figure 1. Description of the protocol HB-MP$^+$.

HB-MP$^+$ solves the synchronization and predictability problems pointed out in section 3. Besides that, the use of random round keys restrict the amount of transcripts related to a given key, increasing its resistance against a great variety off cryptanalytic methods. Nevertheless, it is important to emphasize that the non-linear function $f(\cdot)$ used in the protocol is not a concrete function. It just exists in an abstract form. This function has to meet several requirements, such as:

- *Compatibility with resource-constrained devices*: this property assures that $f(\cdot)$ can be implemented in RFID tags.

- *One-Wayness*: if an adversary retrieves the round key $\mathbf{x}_s$ (or even several round keys), he won't be able to derive the key $\mathbf{x}$ in polynomial time, since $f(\cdot)$ is a one-way function. The same is not true for HB-MP, where the key $\mathbf{x}$ can be trivially derived from the round keys $[\mathbf{xm}_i]_{i=1}^{r}$.

Due to all those listed features, the design of a practical implementation of HB-MP$^+$ appears as a challenging issue in pervasive computing cryptography.

8. Conclusions

In this paper we provided two different attacks against the protocol HB-MP.

We proposed a passive attack against HB-MP that considerably reduces the computational effort of a brute force attack.

We also introduced an active attack that is based on a paradigm completely different from other previous results in the literature. We demonstrated that it is possible to design an efficient attack on a realistic scenario by admitting some additional capabilities to the adversary. We also proved, on a practical basis, the feasibility of this cryptanalytic method.

Additionally, this paper presented HB-MP$^+$: an improved and more resistant version of HB-MP. This recent authentication protocol eliminates its predecessor's vulnerabilities by using random round keys. As a result, HB-MP$^+$ is secure against the proposed cryptanalytic techniques.

References

[1] M. Blum, A. Kalai, H. Wasserman. Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. Journal of the ACM 50, 4 (July 2003), pp. 506–519, 2003.

[2] J. Carrijo, R. Tonicelli, H. Imai and A. C. A. Nascimento. A Novel Probabilistic Passive Attack on the Protocols HB and HB^+. IEICE Trans. on Fundamentals of Electronics, Communications and Computer Science, Vol.E92-A, Number 2, pages 658–662, 2009.

[3] H. Gilbert, M. Robshaw, H. Seurin. Good Variants of HB^+ are Hard to Find. 2008 Financial Cryptography Conference, Lecture Notes in Computer Science vol. 5143, Springer-Verlag, pages 156–170, 2008.

[4] H. Gilbert, M. Robshaw, H. Silvert. An active attack against HB^+ - a provable secure lightweight protocol. Cryptology ePrint Archive, Report 2005/237, 2005, available at http://eprint.iacr.org/2005/237.pdf.

[5] Z. Golebiewski, K. Majcher, F. Zagorski, and M. Zawada. Practical Attacks on HB and HB^{+} Protocols. Cryptology ePrint Archive, Report 2008/241, 2008, available at http://eprint.iacr.org/2008/241.pdf.

[6] N. J. Hopper and M. Blum. Secure Human Identification Protocols. Advances in Cryptology – ASIACRYPT 2001, Lecture Notes in Computer Science vol. 2248, Springer-Verlag, pages 52–66, 2001.

[7] A. Juels and S. A. Weis. Authenticating pervasive devices with Human Protocols. Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science vol. 3621, Springer-Verlag, pages 293–308, 2005.

[8] X. Leng, K. Mayes, K. Markantonakis. HB-MP^{+} Protocol: An Improvement on the HB-MP Protocol. 2008 IEEE International Conference on RFID, pages 118–124, 2008.

[9] J. Munilla and A. Peinado. A further step in the HB-family of lightweight authentication protocols. Computer Networks, vol. 51, Elsevier, pages 2262–2267, 2007.