LIGHT-hHB: A LIGHTER VERSION OF hHB FOR RESOURCE-CONSTRAINED IoT DEVICES.

Ahmad Khoureich Ka

Department of Computer Science, Alioune Diop University of Bambey

ABSTRACT

This paper offers a lighter version of the harder HB protocol (hHB) denoted Light-hHB. This proposal uses the same framework as hHB, that is a two stages protocol: the first one for the establishment of a session key between the reader and the tag and the second one similar to HB+ which was proposed in Crypto ’05 by Hopper and Blum. We also introduce in this paper a novel and lightweight key exchange protocol inspired by the BB84 protocol named the non-quantum key exchange protocol. With the use of a practical implementation of the latter protocol in the first stage of Light-hHB, the transmission cost is drastically reduced compared to the one of hHB, which is its main drawback. In the context of resource-constrained IoT devices, Light- hHB is significantly more practical than hHB and achieves the same security goals.

KEYWORDS

BB84, LPN, HB, hHB, Man-In-the-Middle.

1. INTRODUCTION

IoT tags such as RFID are used for animal tracking, anti-theft for merchandise in stores, payment and access control. Some of these uses require security especially authentication. Since the tag can be forged, the design of well-suited authentication protocol that do not leak sensitive information that a malicious person can use is of great need. Well-suited because RFID tags are resource constrained devises, they have no computational power and storage for standard cryptographic primitives (RSA, AES, hash functions, etc.). This has motivated Hopper and Blum to invent a lightweight authentication protocol for low-cost RFID tags called the HB protocol [14]. The HB protocol is only resistant to passive adversary but falls in front of active ones. A passive adversary can only eavesdrop on communications between the tag and the reader, while an active adversary has the ability to tamper with messages exchanged between the tag and the reader. The resistance of the HB protocol against passive attacks lies on the Learning Parity with Noise (LPN) known to be a hard problem [3-5, 16, 25]. To strengthen HB, Juels and Weis introduced the HB+ protocol [15], which is secure against passive and limited active attacks (the adversary is able to query legitimate tags) but not against the man-in-the-middle attacks (MITM) ones e.g., GRS attack [11]. Since that time many researchers have published protocols [6-8, 18, 22, 28] they claim resistant to MITM attacks but many of them have weaknesses [10, 12, 23]. The one that interests us in this paper is the hHB protocol, which is an attempt to strengthen the HB+ protocol against MITM attacks introduced by Khoureich [17]. The hHB protocol has two stages: in the first one the reader sends a session key to the tag and in the second one the reader does r HB+ rounds to authenticate the tag. Although the hHB has explicit security proofs against MITM attacks, its transmission cost is perplexing in regard to the resource constraints of the RFID tags.
In this paper, we propose Light-hHB a new protocol that follows the same framework as hHB, that is a two stages protocol. We also introduce a novel and lightweight key exchange protocol denoted by the non-quantum key exchange protocol inspired by the BB84 quantum key exchange protocol [2] due to Charles H. Bennet and Gilles Brassard. The first stage of Light-hHB is a practical implementation of our non-quantum key exchange protocol. The second stage remains the same as HB'. The overall protocol is lighter than hHB in terms of transmission cost and is secure against MITM attacks.

The organization of this paper is as follows: in section 2, we describe the HB+ protocol and at the same time the LPN problem. Section 3 briefly explains the BB84 quantum key exchange protocol, which inspired us to introduce our non-quantum key exchange protocol in section 4. Section 5 exposes our proposal Light-hHB and at the same time our implementation of the non-quantum key exchange protocol. Finally, section 6 and 7 give respectively security arguments of Light-hHB and the conclusion.

2. HB+ Protocol

The HB+ protocol [15] is an improvement of the HB protocol [14] proposed by Juels and Weis to resist against active attacks. The design of HB+ is based on the Learning from Parity with Noise (LPN) problem. Informally the LPN problem consists of finding the k-bit string x satisfying the following system of equations.

\[
\begin{align*}
    a_0 \cdot x &= z_0 \oplus \nu_0 \\
    \vdots \\
    a_n \cdot x &= z_n \oplus \nu_n
\end{align*}
\]

Where \(a_i \leftarrow \{0,1\}^k\), \(z_i = a_i \cdot x\), \(\nu_i \in \{0,1\}\), with \(Pr[\nu_i = 1] = \epsilon\), \(Pr[\nu_i = 0] = 1 - \epsilon\) and \(\epsilon \in ]0,1/2[\). Algorithms to solve the LPN problem are published in [5, 9, 19] but it remains a hard one since the complexity of those algorithm are exponential.

In HB+ the reader and the tag share two secrets \(x \in \{0,1\}^{k_1}\) and \(y \in \{0,1\}^{k_2}\) and execute \(r\) 3-steps rounds (figure 1):

1. The tag randomly selects a blinding factor \(b \leftarrow \{0,1\}^{k_2}\) and sends it to the reader.
2. The reader responds with a randomly selected challenge vector \(a \leftarrow \{0,1\}^{k_1}\)
3. The tag selects \(\nu\) in respect to \(Ber_\epsilon\) then computes and sends to the reader the bit \(z = a \cdot x \oplus b \cdot y \oplus \nu\). \(Ber_\epsilon\) denotes the Bernoulli distribution with parameter \(\epsilon\) (i.e., \(\nu \leftarrow Ber_\epsilon\), \(Pr[\nu = 1] = \epsilon\) and \(Pr[\nu = 0] = 1 - \epsilon\)).
For each round, the reader verifies if \( z = a \cdot x \oplus b \cdot y \oplus v \). If the latter verification fails for at most \( \varepsilon r \) rounds, then the tag is authenticated. One consequence of the probabilistic nature of the authentication is that an honest tag can be rejected by an honest reader (False Rejection) or a counterfeit tag be accepted (False Acceptance). Fortunately, the false rejection probability \( P_{FR} = \sum_{i=u+1}^{r} \binom{r}{i} e^i (1-\varepsilon)^{r-i} \) and the false acceptance probability \( P_{FA} = \frac{1}{2^r} \sum_{i=0}^{u} \binom{r}{i} \) are negligible probabilities in \( r \).

The HB\(^+\) protocol is secure against passive and limited active attacks (the adversary is able to query legitimate tags) but not against man-in-the-middle ones. A MITM attack named GRS attack [11] has been successfully mounted against HB\(^+\). The GRS attack consists of adding a perturbation \( e_i = (e_i^1, ..., e_i^{|a|}) \) where \( e_i^j = 1 \) and \( e_i^j = 0 \) for \( i \neq j \) in the challenge vector \( a \) and observe the result of the authentication process of an honest tag. This perturbation is effective if \( e_i \cdot x = 1 \). Thus, if the authentication succeeds with greater probability than \( P_{FA} \) it means that the bit at the position \( i \) of \( x \) is 0 otherwise it is 1. The GRS attack is simple and has motivated many researchers to propose solutions for the HB\(^+\) protocol [6–8,18,22] but many of them show weaknesses in their design [10, 12, 23].

4. The BB84 Quantum Key Exchange Protocol

BB84 is a quantum key exchange protocol invented in 1984 by Charles H. Bennett and Gilles Brassard [2]. Several proofs of its unconditional security have been published [20, 21, 27]. Here, we give a brief description of BB84. Two parties, Alice and Bob wish to share a secret key for a cryptographic purpose. They have access to a public quantum channel and to a public classical channel resistant to active attacks, i.e., an adversary cannot tamper with messages but only listen to them. The BB84 protocol they run consists of the following steps:

1. Alice randomly selects a binary string \( \alpha \).
2. She transforms each bit of \( \alpha \) to a qubit by randomly using a basis in \( \{+,\times\} \) (where + and \( \times \) are respectively rectilinear and diagonal photon polarization states) and obtains a qubit string \( |\alpha\rangle \). Let \( T \) be the function that transforms a bit to a qubit under a basis. We have \( T_+ (0) = \rightarrow, T_+ (1) = \downarrow, T_\times (0) = \leftarrow, T_\times (1) = \uparrow \).
3. Alice sends to Bob the qubit string \( |\alpha\rangle \) through the public quantum channel.
4. On receiving \( |\alpha\rangle \) from Alice, Bob measures each qubit by randomly using a basis in \( \{+,\times\} \) and obtains a binary string \( \beta \). Notice that when Bob uses a basis which is different to the one that Alice uses to produce the qubit, he fails to obtain the correct bit with probability 1/2. Also notice that quantum channels are noisy so Bob cannot successfully measure all the qubits sent by Alice.
5. Alice and Bob compare their basis choices over the public channel.
6. Alice extracts her raw key from \( \alpha \) by discarding the bits where her basis choice does not coincide with Bob’s basis choice. Bob will do the same as Alice using the bit string \( \beta \) to extract his raw key. If an adversary has not manipulated the qubits sent over the public quantum channel, the raw keys extracted by the two parties will be equal.
7. In order to verify that no active attack has occurred on the public quantum channel, Bob reveals to Alice (through the public classical channel) some bits randomly selected from his raw key. If Alice confirms the latter bits, then each of them considers the remaining bits of his/her raw key as the secret key. Otherwise, they restart the protocol because the raw key is compromised.
An illustration of the BB84 protocol without noise and attack on the public quantum channel is given in table 1.

<table>
<thead>
<tr>
<th>Over the public quantum channel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Alice’s random bits α</strong></td>
</tr>
<tr>
<td><strong>Alice’s basis choices</strong></td>
</tr>
<tr>
<td><strong>Qubit string sent by Alice</strong></td>
</tr>
<tr>
<td><strong>Bob’s basis choices</strong></td>
</tr>
<tr>
<td><strong>Bits measured by Bob</strong></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Over the public classical channel</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bases comparison</strong></td>
</tr>
<tr>
<td><strong>Raw key</strong></td>
</tr>
<tr>
<td><strong>Bits revealed by Bob to Alice</strong></td>
</tr>
<tr>
<td><strong>Secret key</strong></td>
</tr>
</tbody>
</table>

## 5. OUR NON-QUANTUM KEY EXCHANGE PROTOCOL

The main security argument of the BB84 protocol comes from the fact that an active attacker cannot duplicates the qubits Alice has sent to Bob (no-cloning theorem of quantum mechanics). This is unfeasible in a classical data transmission. In our non-quantum key exchange protocol, we bypass this impossibility by considering the basis choices of each party as pre-shared secrets, thus they will not be revealed in the public channel. We define two basis \( \frac{0}{1} \) and \( \frac{1}{0} \) which Alice will randomly use to transform bits she sends and Bob to measure bits he receives. We define \( T \) the function that transforms a bit to another bit (not qubit) under a basis in \( \{0, 1\} \) as follows: \( T_{\frac{0}{1}}(0) = 0, T_{\frac{1}{0}}(1) = 1, T_{\frac{1}{0}}(0) = 1, T_{\frac{0}{1}}(1) = 0 \). That is if \( e \) is a bit, \( T_e(\sigma) = (\neg e \land 0) \lor (e \land 1) \) and \( T_e(\sigma) = (\neg e \land 1) \lor (e \land 0) \). Meaning that if \( \sigma \in \{0, 1\} \) we have:

\[
T_e(\sigma) = (\neg e \land \sigma) \lor (e \land \neg \sigma) = e \oplus \sigma
\]

We define \( M \) the function that measures a transformed bit under a basis in \( \left\{ \frac{0}{1}, \frac{1}{0} \right\} \) as being equal to \( T \), that is \( M_{\sigma} = T_{\sigma} \).

The non-quantum key exchange (non-QKE) protocol is defined as follows:

1. Each party Alice and Bob knows \( A \) the basis string of Alice and \( B \) the basis string of Bob.
   \[ A \leftarrow \left\{ \frac{0}{1}, \frac{1}{0} \right\}^n \] and \( A \leftarrow \left\{ \frac{0}{1}, \frac{1}{0} \right\}^n \)

2. Alice randomly chooses a binary string \( \alpha = \{0, 1\}^n \), transforms each bit of it using the basis at the same position in \( A \) and sends the resulting binary string to Bob.

3. Upon receiving the bits from Alice, Bob measures each of them using the basis at the same position in \( B \) and obtains a binary string \( \beta \).

4. Alice extracts her secret key \( s_\alpha \) from \( \alpha \) by discarding the bits where \( A \) and \( B \) does not coincide. Bob will do the same as Alice from the binary string \( \beta \) to extract his secret key
If the binary string sent by Alice in the second step is not modified, $s_\alpha$ will be equal to $s_\beta$ and constitute the secret key (see theorem 4.1).

**Theorem 4.1.** If the binary string that Alice sends to Bob in the second step of the non-quantum key exchange protocol is not modified by an active attacker then the extracted keys $s_\alpha$ and $s_\beta$ are equal.

**Proof.** Let $\alpha = \alpha_1, ..., \alpha_n$ and $\beta = \beta_1, ..., \beta_n$ the binary strings as in the non-quantum key exchange protocol. Let $A = A_1, ..., A_n$ and $B = B_1, ..., B_n$ be respectively the basis strings of Alice and Bob. Let $i = \{1, ..., n\}$ such that $A_i = B_i = \frac{\alpha_i}{\beta}, \theta \in \{0, 1\}$. The bit $\beta_i$ measured by Bob satisfies:

$$\beta_i = M_{\theta} (T_{\theta} (\alpha_i)) = M_{\theta} (\alpha_i \oplus \theta) = (\alpha_i \oplus \theta) \oplus \alpha_i \tag{2}$$

This means for each position $i$ where $A_i = B_i$ we have $\alpha_i = \beta_i$, which implies that $s_\alpha = s_\beta$.

**Theorem 4.2.** Let $s_\alpha'$ be the remaining bits of $\alpha$ after the extraction of $s_\alpha$ and $s_\beta'$ be the remaining bits of $\beta$ after the extraction of $s_\beta$. If the binary string that Alice sends to Bob in the second step of the non-quantum key exchange protocol is not modified by an active attacker then $s_\alpha' = -s_\beta'$.

**Proof.** The bits of $s_\alpha'$ and $s_\beta'$ correspond respectively to bits of $\alpha$ and $\beta$ at positions where the basis string $A$ of Alice does not coincide with the basis string $B$ of Bob. Therefore, from equation 2 it becomes clear that $s_\alpha' = -s_\beta'$.

**Theorem 4.3.** Let $n$ be the length of the basis strings $A$ and $B$ in the non-quantum key exchange protocol. If the binary string sent by Alice to Bob in the second step of the protocol is not modified by an active attacker, then the length $N$ of the extracted key satisfies $N = \Theta(1/2)$.

**Sketch of the proof.** Let $A = A_1, ..., A_n$ and $B = B_1, ..., B_n$. Since Alice and Bob select randomly and independently their basis from $\{0, 1\}$, $A$ and $B$ are sequences of independent and identically distributed random variables of expected values $1/2$ and variances $1/4$.

Let $X_i = \begin{cases} 0 & \text{if } A_i \neq B_i \\ 1 & \text{if } A_i = B_i \end{cases}$

where $X_i$ is a random variable with expected value $1/2$ and variances $1/4$. We have $N = X_1 + X_2 + \cdots + X_n$ and by the law of large numbers, for any $\varepsilon > 0$, $\Pr\left(\left|\frac{N}{n} - \frac{1}{2}\right| < \varepsilon\right) \to 1$ as $n \to \infty$.

Meaning that for large $n$ the length of the extracted key (the value of $N$) is around $1/2$.

**Execution example:** An illustration of the non-quantum key exchange protocol is given in table 2. The basis string of Alice is $A = 01^1 10^1 01^0 00^1 11^1 0^{10}$ and that of Bob is $B = 11^0 01^1 11^1 01^1 01^1 0^{10}$.

These basis strings constitute are pre-shared information.
Table 2. Illustration of our non-quantum key exchange protocol.

<table>
<thead>
<tr>
<th></th>
<th>Alice (A, B)</th>
<th>Bob (A, B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice’s random bits α</td>
<td>1 0 1 0 1 0 0 1 1 1 0 0</td>
<td>0 0 0 1 0 1 1 0 1 0 1 1</td>
</tr>
<tr>
<td>Alice’s A bases</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
</tr>
<tr>
<td>Bits transformed and sent by Alice</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
</tr>
<tr>
<td>Bob’s B bases</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
<td>✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱ ✱</td>
</tr>
<tr>
<td>Secret key s₂</td>
<td>1 1 0 1 0</td>
<td>0</td>
</tr>
</tbody>
</table>

Security Analysis: Notice that a basis string in the non-quantum key exchange protocol is a binary string over its complement. So, equivalently we can say that a basis string can be considered as binary string s. Hence, from equation 1 transforming or measuring bits of a binary string c respectively using T or M under a basis string s is equivalent of XORing s and c.

Theorem 4.4. The non-quantum key exchange protocol is secure if the basis string of Alice is renewed before each execution of the protocol.

Proof. Consider the binary string s as the basis string of Alice and α the binary string she randomly chooses in the second step of the non-quantum key exchange protocol. The only data string exchanged between Alice and Bob is s ⊕ α. Because s is renewed before every execution of the protocol and α is randomly selected, s ⊕ α is a Vernam’s ciphertext (a one-time pad ciphertext) which is perfectly secret, therefore conveys no information about s and α that can compromise the extracted key. This completes the proof.

If an adversary modifies the bits sent by Alice to Bob, the secret keys extracted by the two parties will not be the same. Despite that, there is no need for Bob to reveal some bits of his extracted key to Alice in order to detect an active attack as in BB84 because the only information available to an adversary is a one-time pad ciphertext, which leaks no information on any party’s secret key. It is worth noting that the fact that the non-quantum key exchange protocol is secure only if the basis string of Alice is used once is a serious limitation. In the next section, we introduce an efficient implementation of the non-QKE protocol usable in the context of RFID authentication.

5. The hHB Protocol

Here we recall the harder HB+ protocol (hHB in short) suggested by Khourech [17]. As its name suggests (harder HB+), hHB is an attempt to strengthen the HB+ protocol against the man-in-the-middle (MITM) attacks. The intuition behind the design of hHB is to renew the secrets x and y shared by the tag and the reader before each execution of the protocol. It is believed that the renewal of the pre-shared keys can help an HB-like protocol to resist against the MITM attacks. Therefore, the hHB consists of two stages. The first stage is in essence a session key exchange between the tag and the reader and the second one is similar to the HB+ protocol. The main drawback of hHB is the large amount of transmitted data between the reader and the tag concerning the exchange of a random session key in the first stage.
The hHB protocol works as follow:

1. The reader uses a keyed function $f_s$ to substitute the randomly selected bits $\tau$, $\xi_0$ and $\xi_1$ to bit strings $\alpha$, $\beta$ and $\gamma$.
2. Upon receiving the triplet $\alpha$, $\beta$ and $\gamma$, the tag applies the inverse function $f_s^{-1}$ and recovers the bits $\tau$, $\xi_0$ and $\xi_1$. This first step serves to initialize the secret bit string $p_0$.
3. The tag and the reader repeat $n$ times this same step to compute the secret bit string $p_n$ which will be splitted into $x$ and $y$ used in the second stage of the protocol. This latter stage is identical to the HB' protocol.

6. THE LIGHT-hHB PROTOCOL

The Light-hHB protocol we propose here follows the same framework as hHB, it is a two stages protocol. The first stage of Light-hHB implements a lightweight session key exchange protocol based on our non-quantum key exchange protocol.

6.1. First stage of Light-hHB: a lightweight session key exchange protocol

The lightweight session key exchange protocol we introduce here is a practical implementation of the non-quantum key exchange protocol. It constitutes the first stage of Light-hHB where the session keys $x$ and $y$ are computed. The following figure shows its graphical representation.

Figure 2. The hHB authentication protocol.

Figure 3. The first stage of Light-hHB.
1. The tag and the reader share two $k$-bit secrets $s_T$ and $s_R$. This corresponds to the first step of the non-QKE.

2. The reader selects a random $\xi \in \{0,1\}^{k'}$, computes $s'_R = \text{Mix}(s_R \oplus \xi)$ where $\text{Mix}$ is a mixing function. Then the reader sends $\xi$ to the tag. This is an extra step in regard to the non-QKE. The $k'$-bit string $\xi$ and the mixing function $\text{Mix}$ are used for: (1) to randomize the positions where $s_T$ and $s'_R$ have the same bits thus making the extraction of the secret key at the final step of the protocol random. (2) To consider $s'_R$ as a one-time pad even if it is not a perfect one.

3. Upon receiving $\xi$ the tag computes $s'_R = \text{Mix}(s_R \oplus \xi)$. This is of course an extra step in regard to the non-QKE (the tag also needs $s'_R$).

4. The reader selects a random $\alpha \in \{0,1\}^k$ then sends $\alpha' = \alpha \oplus s'_R$ it to the tag. This corresponds to the second step of the non-QKE.

5. Upon receiving $\alpha'$, the tag computes $\beta = \alpha' \oplus s'_T$. This corresponds to the third step of the non-QKE.

6. The reader compares $s_T$ and $s'_R$ and extracts the session key $x$ from $\alpha$, the remaining bits of $\alpha$ constitute the session key $y$. The tag does the same and extracts $x$ from $\beta$, the remaining bits of $\beta$ is equal to $\neg y$. From theorem 4.3, $|x| = \Theta(k/2)$ and $|y| = \Theta(k/2)$.

**Security Analysis:** The only information that an adversary can see are the random $k'$-bit string $\xi$ and $\alpha' = \alpha \oplus s'_R$ where $\alpha$ is a random $k$-bit string. If the mixing function is not linear relative to the XOR operation and introduces much non-linearity between its inputs and outputs, then $s'_R$ can be considered as a one-time pad hence $\alpha'$ reveals nothing useful about $\alpha$ and $s'_R$ to an adversary to find the extracted key $x$.

### 6.2. Second stage of Light-hHB

This stage is identical to the HB+ protocol. The reader and the tag share two secrets $x$ and $y$ obtained from the first stage and execute $r$ rounds of this stage. The lengths of the $x$ and $y$ are not constant meaning that $|x| = \Theta(k/2)$ and $|y| = \Theta(k/2)$ where $k = |s_R| = |s_T|$. This leads to minor changes. Therefore, the three steps in the second stage are as follows (see figure 3 for a graphical representation):

- The tag randomly selects a fixed length blinding factor $b \leftarrow \{0,1\}^{k/2}$ and sends it to the reader. Even if the length of $y$ is not constant, the choice to set the length of $b$ is done in order to lighten the protocol on the tag side. As a consequence of this choice, the length of $y$ used in the computations in this stage is at most $k/2$.
- The reader responds with a randomly selected challenge vector $a \leftarrow \{0,1\}^{\Theta(k/2)}$.
- Instead of computing $a \cdot x \oplus b \cdot y$ as in the HB+ protocol, the tag computes $p(a \cdot x) \cdot s(x \cdot a) \oplus p(b \cdot y) \cdot s(y \cdot b)$ where $p$ and $s$ are very simple and lightweight functions.
see Algorithm 1 and 2. So the tag sends to the reader the bit \( z = p(a \cdot x) \cdot s(x \cdot a) \oplus p(b \cdot y) \cdot s(y \cdot b) \oplus v \)

4. The reader accepts the round if \( z = p(a \cdot x) \cdot s(x \cdot a) \oplus p(b \cdot \neg y) \cdot s(\neg y \cdot b) \oplus v \).

**Algorithm 1** Function \( p \) that returns a prefix of its first argument

```plaintext
function p(u, v)
    set m to the minimum of |u| and |v|
    return prefix of length m of u
end function
```

**Algorithm 2** Function \( s \) that returns a suffix of its first argument

```plaintext
function s(u, v)
    set m to the minimum of |u| and |v|
    return suffix of length m of u
end function
```

8. **SECURITY ARGUMENTS**

8.1. **Security of Light-hHB against active attacks**

In an active attack the adversary interacts with the tag \( q \) times in order to gain some information and then tries to authenticate to the reader.

**Theorem 6.1.** If HB\(^+\) is secure against limited active attacks where the adversary is able to query the legitimate tag, then Light-hHB is also secure against this type of attack.

**Proof (Outline).** The proof is a reduction of HB\(^+\) to Light-hHB and is analogous to the one of hHB in [17] against active attacks.

8.2. **Security of Light-hHB against MITM attacks**

Here, we give a heuristic analysis of the security of Light-hHB against MITM attacks. In a MITM attack the adversary tampers with messages exchanged between the tag and the reader in \( q \) instances of the protocol and observes the effect of his actions on the behavior of the reader (accepting or rejecting the tag). After that, the adversary tries to authenticate to the reader.

**The adversary mounts an attack on the first stage of Light-hHB.** Consider an attack where the adversary modifies a bit of \( \xi \), that is he causes a bit flip in the input of \( Mix \). Since it is required that the mixing function \( Mix \) introduce much non-linearity between inputs and outputs, that modification will be hard to follow in the output \( s'_R \). Therefore, such modification cannot benefit the adversary.

Now suppose that the adversary flips a bit of \( \alpha' \). This will lead to a bit flip in either \( x \) or \( y \). This perturbation may have no effect on the overall authentication process because of the functions \( p \) and \( s \). Therefore, it is unlikely to gain useful information on \( s'_R \) or \( s'_T \) from the manipulation of \( \alpha' \).

**The adversary mounts an attack on the second stage of Light-hHB.** The session keys obtained from the first stage of the protocol have non constant lengths. That is \( |x| = \Theta(k/2) \) and \( |y| = \Theta(k/2) \) where \( k \) is the length of the pre-shared keys \( k = |s_R| = |s_T| \). The challenge vector \( a \) in the second phase of the protocol is randomly picked from \( \{0, 1\}^{\Theta(k/2)} \).
Without loss of generality consider \( |a| \in \left[ \frac{k}{2} - \frac{\delta}{2} \cdot \frac{k}{2} + \frac{\delta}{2} \right] \) and \( |x| \in \left[ \frac{k}{2} - \frac{\delta}{2} \cdot \frac{k}{2} + \frac{\delta}{2} \right] \). We consider three cases depending on whether the length of \( a \) is greater than the length of \( x \).

**Case 1:** \( |a| \leq |x| \). In this case \( p(a \cdot x) = a \) and \( s(x \cdot a) \) is a suffix of \( x \). In a MITM attack an adversary can flip a bit of \( a \) as in the GRS attack [11], that is he perturbs the challenges \( a \) sent by a legitimate reader by sending to the tag \( a \oplus e_i \) where \( e_i = (e_i^1, ..., e_i^{|a|}) \), \( e_i^j = 1 \) and \( e_i^j = 0 \) for \( i \neq j \). The probability that this action perturbs the authentication process is \( \frac{(\delta-1) |x|}{2 \delta |a|} \) and since \( s(x \cdot a) \) is a suffix of \( x \) the adversary has no way of locating the affected bit of \( x \).

**Case 2:** \( |a| = |x| \). In this case \( p(a \cdot x) = a \) and \( s(x \cdot a) = x \). This event happens with probability \( \frac{1}{\delta} \). When an adversary perturbs a bit of \( a \) he can locate the affected bit of \( x \).

**Case 3:** \( |a| \geq |x| \). In this case \( p(a \cdot x) \) is a prefix of \( a \) and \( s(x \cdot a) = x \). When an adversary flips a bit of \( a \), the probability that this action perturbs the authentication process is \( \frac{(\delta-1) |x|}{2 \delta |a|} \). If the perturbation introduced by the adversary has no effect on the authentication process, then \( e_i \cdot x = 0 \), otherwise \( e_i \cdot x = 1 \). Considering the above three cases, the probability that the adversary obtains useful information on \( x \) is less than \( \frac{1}{2} + \frac{1}{2^\delta} \). Since the session key \( x \) is renewed before each execution of the protocol, it is very unlikely that the pre-shared secrets \( s_T \) and \( s_R \) be recovered. Therefore, the advantage of the adversary against Light-hHB is negligible.

#### 9. Design Choices

In this section, we give key lengths and specify our choice for the mixing function \( Mix \).

**Choices for our lightweight session key exchange protocol.** For the mixing function a linear feedback shift register will not suit the security requirement (\( s_R \) to be an OTP) because LFSR(\( s_R \odot \xi \)) = LFSR(\( s_R \)) \odot LFSR(\( \xi \)). A good option for \( Mix \) is the mixing function used by Shamir in SQUASH-128 [26]. Recall that the resistance of SQUASH-128 to some attacks is partly due to that non-linear mixing function [24]. The latter is the 128-bit non-linear feedback shift register (NFSR) of Grain-128 [13]. That NFSR was updated in the new version of Grain-128 [1] and have the feedback function:

\[
\begin{align*}
    b_{i+128} &= b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+67} + b_{i+11} + b_{i+13} + \\
    &+ b_{i+17} b_{i+18} b_{i+27} b_{i+59} + b_{i+40} b_{i+48} + b_{i+61} b_{i+65} + b_{i+68} b_{i+84} + \\
    &+ b_{i+89} b_{i+92} b_{i+93} b_{i+95} + b_{i+23} b_{i+24} b_{i+25} + b_{i+70} b_{i+78} b_{i+82}.
\end{align*}
\]

where \( b_0, ..., b_127 \) is its initial state. We consider \( s_R = Mix(s_R \odot \xi) \) to be the internal state of the NFSR after being initialized with \( s_R \odot 0^{64} || \xi \) and clocked 512 times (to obtain a good non-linearity). This means that we set \( |s_R| = |s_T| = 128 \) and \( |\xi| = 64 \). This NFSR, as stated by its authors [1] introduces much non-linearity that it would not be possible to solve from its output a system of equations in its initial state.

**Settings for the second stage of Light-hHB.** This stage is identical to the HB* protocol. From theorem 4.3 we have the length of \( x \) (obtained from the first stage of the protocol) around 64 bits so, we set \( |a| = \Theta(64) \). The length of \( y \) is also around 64 bits but in order to lighten the protocol on the tag side the length of \( b \) is fixed to 64 bits. These values for the length of \( x \) and \( y \) do not follow the recommendations of Levieil et al [19] but we think it will not affect the security of Light-hHB because \( x \) and \( y \) are one-time secrets and an adversary does not need to “brute force”
or to resolve a variable LPN instance. We also set the number of rounds $r$ of this second stage to 1164 and the threshold $u$ to $0.348 \times r$ thus the probabilities of false acceptance and false rejection will respectively be $2^{-80}$ and $2^{-40}$. With these settings the transmission cost for the establishment of $x$ and $y$ (in the first stage of Light-hHB) is equal to $|\xi| + |\alpha'| = 194$ bits which is far less than the 50115 bits used by the $h$HB reader to transmit the same secrets to the tag. This represents a substantial gain in the transmission cost and makes Light-hHB significantly more practical than $h$HB.

10. CONCLUSIONS

In this paper we have presented a lighter version of the $h$HB protocol named Light-hHB. We have also presented a novel and lightweight key exchange protocol inspired by BB84 denoted the non-quantum key exchange protocol. A practical implementation of the latter protocol is also exposed. With this improvement, Light-hHB is more practical than $h$HB as it reduces drastically the transmission cost and have a better security.

REFERENCES

[10] h. gilbert, m. j. roshaw, and y. seurin. good variants of hb+ are hard to find. in financial cryptography and data security, pages 156–170. springer, 2008.

AUTHORS

Dr Ahmad Khoureich Ka is Maître de conférences at University of Alioune Diop de Bambey since 2007. His research interests include cryptography, and security on power-restrained devices.