# Light-hHB: A New Version of hHB with Improved Session Key Exchange

Ka Ahmad Khoureich*

**Abstract**

This paper offers a new version of the hHB protocol 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$^{+}$. 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 RFID tags, 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

The rapid progress we see today in the use of the RFID chip is due to its advantages over barcodes (timeliness in data collection, no need of human involvement, read/write for tags, etc.). RFID tags 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 tools (RSA, AES, hash functions. etc.). This has motivated Hopper and Blum to invent the HB protocol [15], a lightweight authentication protocol for low cost RFID tags that has inspired many researchers to propose HB-like protocols. The HB protocol is only resistant to passive adversary but falls in front of active ones. Its resistance against passive attacks lies on the Learning Parity with Noise (LPN) known to be a hard problem [3–5, 14, 15, 18, 27]. To strengthen HB, Juels and Weis introduce the HB$^{+}$ protocol [16], which is secure against passive and active attacks [16,17] but not against MITM ones e.g. GRS attack [11]. Since that time many researchers have published protocols [6–8, 20, 24] they claim resistant to MITM attacks but many of them have weaknesses [10, 12, 25]. The one that interests us in this paper is the hHB protocol, which is an attempt to strengthen the HB$^{+}$ protocol against the man-in-the-middle (MITM) attacks introduced by Khoureich [19]. The hHB protocol has two stages;

*Dept. of Computer Science, Alioune Diop University of Bambey, Senegal. ahmadkhoureich.ka@uadb.edu.sn

in the first one the reader sends a session key to the tag and in the second one the reader do $r$ 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. Finaly section 6 and 7 gives respectively security arguments of Light-hHB and the conclusion.

# 2 HB$^+$ Protocol

The HB$^+$ protocol is an improvement of the HB protocol [14] proposed by Juels and Weis [16]. HB$^+$ resists to passive and active attacks [16, 17]. It is a lightweight protocol with a very simple design, see figure 1. The resistance of HB$^+$ to active attacks comes from the introduction of a random blinding factor $b$. Informaly the LPN is the problem of finding the $k$-bit string $x$ from the following system of equations.

$$\begin{cases} a_0 \cdot x & = z_0 \oplus \nu_0 \\ \cdots & \\ a_n \cdot x & = z_n \oplus \nu_n \end{cases}$$

where $a_i \leftarrow \{0,1\}^k$, $z_i = a_i \cdot x$, $\nu_i \in \{0,1\}$ with $\Pr[\nu_i = 1] = \varepsilon$, $\Pr[\nu_i = 0] = 1 - \varepsilon$ and $\varepsilon \in ]0, 1/2[$. Algorithms [5, 9, 21] to solve the LPN problem has been published but it remains a hard one. 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):

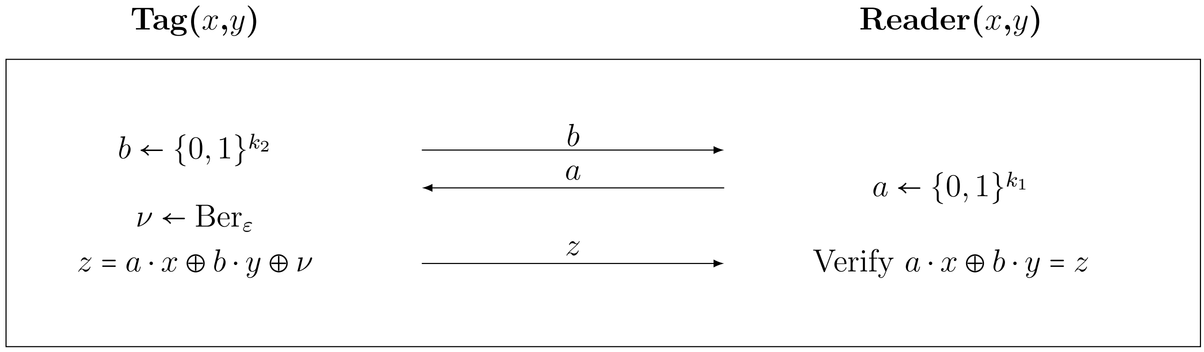
**Tag**$(x,y)$
**Reader**$(x,y)$
$b \leftarrow \{0,1\}^{k_2}$
$b$
$a$
$a \leftarrow \{0,1\}^{k_1}$
$\nu \leftarrow \mathrm{Ber}_\varepsilon$
$z$
$z = a \cdot x \oplus b \cdot y \oplus \nu$
Verify $a \cdot x \oplus b \cdot y = z$

Figure 1: A round of the HB$^+$ Protocol.

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 $\mathsf{Ber}_\varepsilon$ then computes and sends to the reader the bit $z = a \cdot x \oplus b \cdot y \oplus \nu$. $\mathsf{Ber}_\varepsilon$ denotes the Bernoulli distribution with parameter $\varepsilon$, (i.e. $\nu \leftarrow \mathsf{Ber}_\varepsilon$, $\Pr[\nu = 1] = \varepsilon$ and $\Pr[\nu = 0] = 1 - \varepsilon$).

The reader recognizes the outcome *yes* or *no* (Verify $a \cdot x \oplus b \cdot y = z$) of each round and if the number of *no* does not exceed a threshold $\mathsf{u}$ (chosen greater than $\varepsilon r$) the tag is authenticated. One consequence of the probabilistic nature of the authentication is that a honest tag can be rejected by a honest reader (False Rejection) or a counterfeit tag be accepted (False Acceptance). Fortunately, *false rejection* and *false acceptance* happen with negligible probabilities in $k_1$ (because $r = r(k_1)$): $P_{FR} = \sum_{i=\mathsf{u}+1}^{r} \binom{r}{i} \varepsilon^i (1-\varepsilon)^{r-i}$ and $P_{FA} = \frac{1}{2^r} \sum_{i=0}^{u} \binom{r}{i}$.

The HB$^+$ protocol is secure against passive and active attacks 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 $e_i$ (the vector with all 0s but 1 at position $i$) in the challenge vector $a$ and observe the result of the authentication process of a 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, 20, 24] but many of them show weaknesses in their design [10, 12, 25].

# 3 The BB84 Quantum Key Exchange Protocol

BB84 is a quantum key exchange protocol invented in 1984 by Charles H. Bennet and Gilles Brassard [2]. Several proofs of its unconditional security have been published [22, 23, 29]. 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 (an adversary can't tamper with messages but only eavesdrop). 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 $\{+, \times\}$[1], and obtains a qubit string $|\alpha\rangle$. Let $T$ be the function that transforms a bit to a qubit under a basis. We have $T_+(0) = \leftrightarrow$, $T_+(1) = \updownarrow$, $T_\times(0) = \swarrow\!\!\!\nearrow$ and $T_\times(1) = \nwarrow\!\!\!\searrow$.

3. Alice sends to Bob the qubit string $|a\rangle$ 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 can't successfully measure all the qubits sent by Alice.

[1] + and × are respectively rectilinear and diagonal photon polarization states.

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. Il 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.

| Over the public quantum channel | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alice's random bits $\alpha$ | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| Alice's basis choices | + | × | + | + | + | × | + | × | + | × | + | + |
| Qubit string sent by Alice | ↕ | ⤢ | ↕ | ↔ | ↕ | ⤢ | ↔ | ⤡ | ↕ | ⤡ | ↔ | ↔ |
| Bob's basis choices | × | × | + | + | × | × | × | + | × | + | + | × |
| Bits measured by Bob | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 |
| Over the public classical channel | | | | | | | | | | | | |
| Bases comparison | | OK | OK | OK | | OK | | | | | OK | |
| Raw key | | 0 | 1 | 0 | | 0 | | | | | 0 | |
| Bits revealed by Bob to Alice | | 0 | | | | 0 | | | | | | |
| Secret key | | | 1 | 0 | | | | | | | 0 | |

Table 1: Illustration of the BB84 protocol without noise and attack on the public quantum channel.

# 4 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 $\{\frac{0}{1}, \frac{1}{0}\}$ as follows: $T_{\frac{0}{1}}(0) = 0$, $T_{\frac{0}{1}}(1) = 1$, $T_{\frac{1}{0}}(0) = 1$ and $T_{\frac{1}{0}}(1) = 0$. That is if $e$ is a bit, $T_{\frac{0}{1}}(e) = (\neg e \wedge 0) \vee (e \wedge 1)$ and $T_{\frac{1}{0}}(e) = (\neg e \wedge 1) \vee (e \wedge 0)$. This means for $\sigma \in \{0, 1\}$:

$$T_{\frac{\sigma}{\neg\sigma}}(e) = (\neg e \wedge \sigma) \vee (e \wedge \neg\sigma) = e \oplus \sigma \tag{1}$$

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

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

1. Each party Alice and Bob knows $\mathcal{A}$ the basis string of Alice and $\mathcal{B}$ the basis string of Bob. $\mathcal{A} \leftarrow \{\frac{0}{1}, \frac{1}{0}\}^n$ and $\mathcal{B} \leftarrow \{\frac{0}{1}, \frac{1}{0}\}^n$

2. Alice randomly chooses a binary string $\alpha \leftarrow \{0,1\}^n$, transforms each bit of it using the basis at the same position in $\mathcal{A}$ 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 $\mathcal{B}$ and obtains a binary string $\beta$.

4. Alice extracts her secret key $s_\alpha$ from $\alpha$ by discarding the bits where $\mathcal{A}$ and $\mathcal{B}$ does not coincide. Bob will do the same as Alice from the binary string $\beta$ to extract his secret key $s_\beta$. 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 $\alpha = \alpha_1, \dots, \alpha_n$ and $\beta = \beta_1, \dots, \beta_n$ the binary strings as in the non-quantum key exchange protocol. Let $\mathcal{A} = \mathcal{A}_1, \dots, \mathcal{A}_n$ and $\mathcal{B} = \mathcal{B}_1, \dots, \mathcal{B}_n$ be respectively the basis strings of Alice and Bob. Let $i \in \{1, \dots, n\}$ such that $\mathcal{A}_i = \mathcal{B}_i = \frac{\theta}{\neg\theta}$, $\theta \in \{0,1\}$. The bit $\beta_i$ measured by Bob satisfies:

$$\begin{aligned} \beta_i &= M_{\frac{\theta}{\neg\theta}}(T_{\frac{\theta}{\neg\theta}}(\alpha_i)) = M_{\frac{\theta}{\neg\theta}}(\alpha_i \oplus \theta) \\ &= (\alpha_i \oplus \theta) \oplus \theta = \alpha_i \end{aligned} \tag{2}$$

This means for each position $i$ where $\mathcal{A}_i = \mathcal{B}_i$ 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 $s'_\beta$ 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 $s'_\alpha = \neg s'_\beta$.*

*Proof.* The bits of $s'_\alpha$ and $s'_\beta$ correspond respectively to bits of $\alpha$ and $\beta$ at positions where the basis string $\mathcal{A}$ of Alice does not coincide with the basis string $\mathcal{B}$ of Bob. Therefore from equation 2 it becomes clear that $s'_\alpha = \neg s'_\beta$. □

**Size of the extracted key.** The size of the extracted key is around $n/2$.

**Theorem 4.3.** *Let $n$ be the length of the basis strings $\mathcal{A}$ and $\mathcal{B}$ 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 size $N$ of the extracted key satisfies $N = \Theta(n/2)$.*

*Sketch of the proof.* Let $\mathcal{A} = \mathcal{A}_1, \dots, \mathcal{A}_n$ and $\mathcal{B} = \mathcal{B}_1, \dots, \mathcal{B}_n$. Since Alice and Bob select randomly and independently their basis from $\{\frac{0}{1}, \frac{1}{0}\}$, $\mathcal{A}$ and $\mathcal{B}$ are sequences of independent and identically distributed random variables of expected values 1/2 and variances 1/4. Let

$$X_i = \begin{cases} 0 & \text{if } \mathcal{A}_i \neq \mathcal{B}_i \\ 1 & \text{if } \mathcal{A}_i = \mathcal{B}_i. \end{cases}$$

$X_i$ is a random variable with expected value 1/2 and variance 1/4. We have $N = X_1 + X_2 + \ldots + X_n$ and by the law of large numbers, for any $\varepsilon > 0$, $Pr(|\frac{N}{n} - \frac{1}{2}| < \varepsilon) \to 1$ as $n \to \infty$. This means that for large $n$ the size of the extracted key (the value of $N$) is around $n/2$. □

**Execution example:** An illustration of the non-quantum key exchange protocol is given in table 2. The basis string of Alice is $\frac{1\,0\,1\,1\,1\,0\,1\,0\,0\,1\,1\,1}{0\,1\,0\,0\,0\,1\,0\,1\,1\,0\,0\,0}$ and that of Bob $\frac{1\,1\,0\,0\,1\,1\,1\,0\,1\,0\,0\,1}{0\,0\,1\,1\,0\,0\,0\,1\,0\,1\,1\,0}$. The basis strings constitute pre-shared information.

| Alice $(\mathcal{A}, \mathcal{B})$ | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Alice's random bits $\alpha$ | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| Alice's $\mathcal{A}$ bases | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ |
| Bits transformed and sent by Alice | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| Bob's $\mathcal{B}$ bases | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ |
| Secret key $s_\alpha$ | 1 | | | | 1 | | 0 | 1 | | | | 0 |
| Bob $(\mathcal{A}, \mathcal{B})$ | | | | | | | | | | | | |
| Bits received by Bob $\beta$ | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 |
| Bob's $\mathcal{B}$ bases | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ |
| Bits measured by Bob | 1 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 1 | 0 |
| Alice's $\mathcal{A}$ bases | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{0}{1}$ | $\frac{0}{1}$ | $\frac{1}{0}$ | $\frac{1}{0}$ | $\frac{1}{0}$ |
| Secret key $s_\beta$ | 1 | | | | 1 | | 0 | 1 | | | | 0 |

Table 2: Illustration of our non-quantum key exchange protocol.

**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 a 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 $\alpha$ 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 $s \oplus \alpha$. Because $s$ is renewed before every execution of the protocol and $\alpha$ is randomly selected, $s \oplus \alpha$ is a Vernam's ciphertext (a message XORed with a one-time pad) which is perfectly secret, therefore conveys no information about $s$ and $\alpha$ 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 Vernam's 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 Light-hHB protocol

As a new version of the hHB protocol suggested by Khoureich [19], the Light-hHB protocol we propose here follows the same framework. That is a two stages protocol; the first stage which in essence is a session key exchange between the tag and the reader and the second one being the HB$^+$ protocol. As its name suggests (harder HB$^+$), hHB is an attempt to strengthen the HB$^+$ protocol against the man-in-the-middle (MITM) attacks.

Our motivation to design a new version of hHB comes from its severe drawback which is the large amount of transmitted data between the reader and the tag concerning the exchange of a random session key (see conclusion of [19]). So, for the first stage of Light-hHB we introduce a lightweight session key exchange protocol based on our non-quantum key exchange protocol.

## 5.1 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 is intended to constitute the first stage of Light-hHB and works as follows (see figure 2 for a graphical representation):

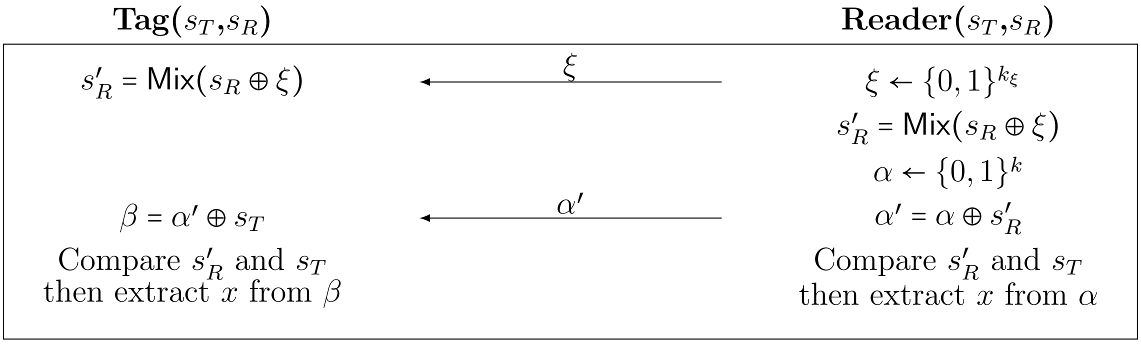
**Tag**($s_T$,$s_R$)
**Reader**($s_T$,$s_R$)
$s'_R = \mathsf{Mix}(s_R \oplus \xi)$
$\xi$
$\xi \leftarrow \{0,1\}^{k_\xi}$
$s'_R = \mathsf{Mix}(s_R \oplus \xi)$
$\alpha \leftarrow \{0,1\}^k$
$\beta = \alpha' \oplus s_T$
$\alpha'$
$\alpha' = \alpha \oplus s'_R$
Compare $s'_R$ and $s_T$ then extract $x$ from $\beta$
Compare $s'_R$ and $s_T$ then extract $x$ from $\alpha$

Figure 2: Our lightweight session key exchange protocol.

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 $\xi \in \{0,1\}^{k_\xi}$, computes $s'_R = \mathsf{Mix}(s_R \oplus \xi)$ where 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 $k_\xi$-bit string $\xi$ and the mixing function $\mathsf{Mix}$ 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 $s'_R = \mathsf{Mix}(s_R \oplus \xi)$. 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 tag do the same and extracts the session key $x$ from $\beta$. From theorem 4.3, $|x| = \Theta(k/2)$. This corresponds to the final step of the non-QKE.

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

## 5.2 Second stage of Light-hHB

This stage is identical to the HB$^+$ protocol. The secrets $x$ and $y$ are obtained from the first stage. $x$ is the extracted key at the sixth step of the lightweight session key exchange protocol and $y$ the remaining bits of $\beta$ after the extraction of $x$. From theorem 4.2 we see that the $y$ obtained by the tag is the complement of the $y$ obtained by the reader. So when the tag considers $y$, the reader considers $\neg y$ in the second stage of Light-hHB. The lengths of the secrets $x$ and $y$ are not fixed ($|x| = \Theta(k/2)$ 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):

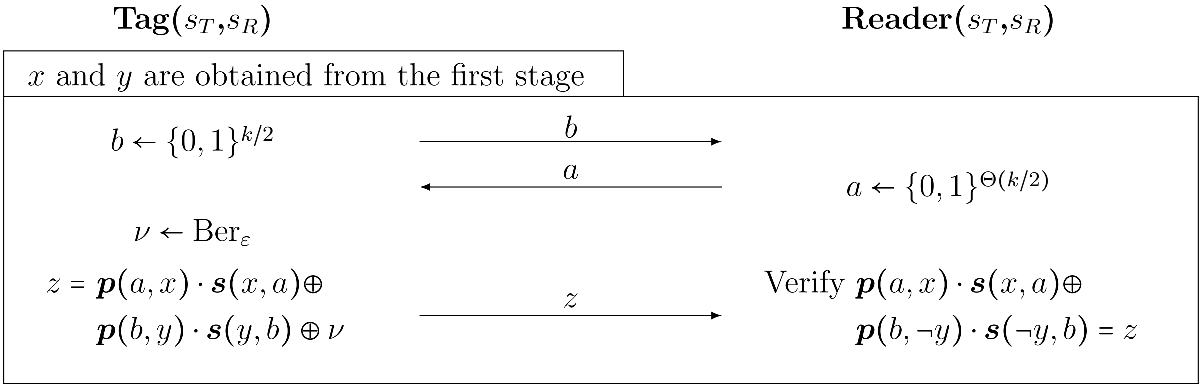
**Tag**$(s_T, s_R)$
**Reader**$(s_T, s_R)$
$x$ and $y$ are obtained from the first stage
$b \leftarrow \{0,1\}^{k/2}$
$b$
$a$
$a \leftarrow \{0,1\}^{\Theta(k/2)}$
$\nu \leftarrow \mathrm{Ber}_\varepsilon$
$z = \boldsymbol{p}(a,x) \cdot \boldsymbol{s}(x,a) \oplus \boldsymbol{p}(b,y) \cdot \boldsymbol{s}(y,b) \oplus \nu$
$z$
Verify $\boldsymbol{p}(a,x) \cdot \boldsymbol{s}(x,a) \oplus \boldsymbol{p}(b,\neg y) \cdot \boldsymbol{s}(\neg y,b) = z$

Figure 3: One round of the second stage of Light-hHB authentication protocol.

1. 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 random around $k/2$ for each round, the choice to fixe the length of $b$ is done in order to lighten the protocol on the tag, As a consequence of this choice, the length of $y$ used in the computations in this stage is at most $k/2$.

2. The reader responds with a randomly selected challenge vector $a \leftarrow \{0,1\}^{\Theta(k/2)}$, that is the length of $a$ is random around $k/2$ for each round.

3. Instead of computing $a \cdot x \oplus b \cdot y$ as in the HB$^+$ protocol, the tag compute $\boldsymbol{p}(a,x) \cdot \boldsymbol{s}(x,a) \oplus \boldsymbol{p}(b,y) \cdot \boldsymbol{s}(y,b)$ where $\boldsymbol{p}$ and $\boldsymbol{s}$ are very simple and lightweight functions, see Algorithm 1 and 2. So the tag sends to the reader the bit $z = \boldsymbol{p}(a,x) \cdot \boldsymbol{s}(x,a) \oplus \boldsymbol{p}(b,y) \cdot \boldsymbol{s}(y,b) \oplus \nu$.

4. The reader accepts the round if $z = \boldsymbol{p}(a,x) \cdot \boldsymbol{s}(x,a) \oplus \boldsymbol{p}(b,\neg y) \cdot \boldsymbol{s}(\neg y,b)$. Recall that the $y$ obtained by the reader from the first stage is the complement of the $y$ obtained by the tag.

**Algorithm 1** Function $\boldsymbol{p}$ that returns a prefix of its first argument

```
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 $\boldsymbol{s}$ that returns a suffix of its first argument

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

# 6 Security Arguments

## 6.1 Security of Light-hHB against active attacks

Active attacks are ones where the adversary can interact 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 active attacks then Light-hHB is also secure against active attacks.*

*proof (Outline).* The proof is a reduction of HB$^+$ to Light-hHB and is analogous to the one of hHB against active attacks [19]. □

### 6.2 Security of Light-hHB against MITM attacks

Here, we give a heuristic analysis of the security of Light-hHB against MITM attacks. MITM attacks are ones where the adversary can tamper with messages exchanged between the tag and the reader in $q$ instances of the protocol and observes the effect on his actions (in order to deduce information) on the behaviour of the reader (accepting or rejecting the tag). And after that, 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 $\mathsf{Mix}$. Since it is required that the mixing function $\mathsf{Mix}$ introduce much non-linearity between inputs and outputs, that modification will be hard to follow in the output $s'_R$. Therefore, we believe that such modification can not benefit the adversary.

Suppose now that the adversary flips a bit of $\alpha'$. This will lead to a bit flip in either $x$ or $y$. The effects of this perturbation is difficult to follow because the overall authentication process can succeed (if the perturbation does not reach $\boldsymbol{s}(x, a)$ or $\boldsymbol{s}(y, b)$) or fail. Therefore, we believe that it is improbable to gain useful informations on $s'_R$ or $s_T$ from the manipulation of $\alpha'$.

**The adversary mounts an attack on the second stage of Light-hHB.** We have shown in section 5.1, that the first stage of the protocol is secure if the mixing function is not linear relative to the XOR operation and introduces much non-linearity between its inputs and outputs. That is an attacker cannot obtain information on the established key by only eavesdropping on messages exchanged between the tag and the reader. Once the key is established, it is up to the user (here the HB$^+$ protocol) to keep it secret. We know that HB$^+$ is weak against MITM attacks [11], so, suppose that by some mean the adversary obtains a bit of $\boldsymbol{s}(x, a)$. The same following heuristic reasoning applies when the adversary obtains a bit of $\boldsymbol{s}(y, b)$. Since the size of $x$ is random around $k/2$ for each instance of the protocol and the application of functions $\boldsymbol{p}$ and $\boldsymbol{s}$ respectively on $a$ and $x$, the attacker would not be able to find the position of that bit in $x$. Thus it is not possible for him to know any bit of $\alpha$ or $\beta$. Notice that knowing a bit of $\beta$ leads to a bit of $s_T$ and a bit of $\alpha$ leads to a bit of $s'_R$. Notice also that with some bits of $s'_R$ (which we believe difficult to obtain) an adversary can try to recover the input of the mixing function. With the requirement that $\mathsf{Mix}$ introduce much non-linearity between inputs and outputs, trying to recover $\xi \oplus s_R$ from $s'_R$ will not be easy.

# 7 Design choices

In this section, we give key sizes and specify our choice for the mixing function $\mathsf{Mix}$.

**Choices for our lightweight session key exchange protocol.** Lets denote by $k_s$ the size of the pre-shared keys $s_T$, $s_R$ and by $k_\xi$ the size of $\xi$. For the mixing function a linear feedback shift register will not suit the security requirement ($s'_R$ to be a OTP) because $\mathrm{LFSR}(\xi \oplus s_R) = \mathrm{LFSR}(\xi) \oplus \mathrm{LFSR}(s_R)$. A good option for $\mathsf{Mix}$ is the mixing function used by Shamir in SQUASH-128 [28]. Recall that the resistance of SQUASH-128 to some attacks e.g. [26] is partly due to that non-linear mixing function. 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:

$$\begin{aligned} b_{i+128} &= b_i + b_{i+26} + b_{i+56} + b_{i+91} + b_{i+96} + b_{i+3}b_{i+67} + b_{i+11}b_{i+13} + \\ & b_{i+17}b_{i+18}b_{i+27}b_{i+59} + b_{i+40}b_{i+48} + b_{i+61}b_{i+65} + b_{i+68}b_{i+84} + \\ & b_{i+88}b_{i+92}b_{i+93}b_{i+95} + b_{i+22}b_{i+24}b_{i+25} + b_{i+70}b_{i+78}b_{i+82}. \end{aligned}$$

where $b_0, \ldots, b_{127}$ is its initial state. We consider $s'_R = \mathsf{Mix}(s_R \oplus \xi)$ to be the internal state of the NFSR after being initialized with $s_R \oplus 0^{64} \| \xi$ and clocked 512 times (to obtain a good non-linearity). This means we set $k_s$ to 128 and $k_\xi$ to 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 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* [21] 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 $\mathsf{u}$ 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 $(|\xi| + |\alpha'|) = 192$ bits, which is negligible compared to the 50115 bits used by the hHB 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 hHB.

# 8 Conclusion

In this paper we have presented a new version of the hHB 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. Light-hHB follows the same framework as hHB and exploits the non-quantum key exchange protocol in its first stage. With this improvement, Light-hHB is more practical than hHB as it reduces drastically the transmission cost and have the same level of security.

# References

[1] M. Agren, M. Hell, T. Johansson, and W. Meier. Grain-128a: a new version of grain-128 with optional authentication. *International Journal of Wireless and Mobile Computing*, 5(1):48–59, 2011.

[2] C. H. Bennett and G. Brassard. Quantum cryptography: Public-key distribution and coin tossing. In *IEEE International Conference on Computers, Systems and Signal Processing*, page 175–179. IEEE, 1984.

[3] E. R. Berlekamp, R. J. McEliece, and H. C. Van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[4] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in cryptology—CRYPTO'93*, pages 278–291. Springer, 1994.

[5] A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

[6] J. Bringer and H. Chabanne. Trusted-HB: a low-cost version of HB$^+$ secure against man-in-the-middle attacks. *arXiv preprint arXiv:0802.0603*, 2008.

[7] J. Bringer, H. Chabanne, and E. Dottax. HB$^{++}$: a lightweight authentication protocol secure against some attacks. In *Security, Privacy and Trust in Pervasive and Ubiquitous Computing, 2006. SecPerU 2006. Second International Workshop on*, pages 28–33. IEEE, 2006.

[8] D. N. Duc and K. Kim. Securing HB$^+$ against GRS man-in-the-middle attack. In *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security*, 2007.

[9] M. P. Fossorier, M. J. Mihaljević, H. Imai, Y. Cui, and K. Matsuura. An algorithm for solving the LPN problem and its application to security evaluation of the HB protocols for RFID authentication. In *Progress in Cryptology-INDOCRYPT 2006*, pages 48–62. Springer, 2006.

[10] D. Frumkin and A. Shamir. Un-trusted-HB: Security vulnerabilities of trusted-HB. *IACR Cryptology ePrint Archive*, 2009:44, 2009.

[11] H. Gilbert, M. Robshaw, and H. Sibert. Active attack against HB$^+$: a provably secure lightweight authentication protocol. *Electronics Letters*, 41(21):1169–1170, 2005.

[12] H. Gilbert, M. J. Robshaw, and Y. Seurin. Good variants of HB$^+$ are hard to find. In *Financial Cryptography and Data Security*, pages 156–170. Springer, 2008.

[13] M. Hell, T. Johansson, A. Maximov, and W. Meier. A stream cipher proposal: Grain-128. In *IEEE International Symposium on Information Theory (ISIT 2006)*. Citeseer, 2006.

[14] N. J. Hopper and M. Blum. A secure human-computer authentication scheme. In *Technical Report CMU-CS-00-139*. Carnegie Mellon University, 2000.

[15] N. J. Hopper and M. Blum. Secure human identification protocols. In *Advances in cryptology—ASIACRYPT 2001*, pages 52–66. Springer, 2001.

[16] A. Juels and S. A. Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology–CRYPTO 2005*, pages 293–308. Springer, 2005.

[17] J. Katz and J. S. Shin. Parallel and concurrent security of the HB and HB$^+$ protocols. In *Advances in Cryptology-EUROCRYPT 2006*, pages 73–87. Springer, 2006.

[18] M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

[19] K. A. Khoureich. hHB: a harder HB$^+$ protocol. Available at http://eprint.iacr.org/2014/562.

[20] X. Leng, K. Mayes, and K. Markantonakis. HB-MP$^+$ protocol: An improvement on the HB-MP protocol. In *RFID, 2008 IEEE International Conference on*, pages 118–124. IEEE, 2008.

[21] É. Levieil and P. A. Fouque. An improved LPN algorithm. In *Security and Cryptography for Networks*, pages 348–359. Springer, 2006.

[22] H.-K. Lo and H. F. Chau. Unconditional security of quantum key distribution over arbitrarily long distances. *science*, 283(5410):2050–2056, 1999.

[23] D. Mayers. Unconditional security in quantum cryptography. *Journal of the ACM (JACM)*, 48(3):351–406, 2001.

[24] J. Munilla and A. Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51(9):2262–2267, 2007.

[25] 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*, pages 108–124. Springer, 2008.

[26] K. Ouafi and S. Vaudenay. Smashing squash-0. In *Advances in Cryptology-EUROCRYPT 2009*, pages 300–312. Springer, 2009.

[27] O. Regev. On lattices, learning with errors, random linear codes, and cryptography. *Journal of the ACM (JACM)*, 56(6):34, 2009.

[28] A. Shamir. Squash–a new mac with provable security properties for highly constrained devices such as rfid tags. In *Fast Software Encryption*, pages 144–157. Springer, 2008.

[29] P. W. Shor and J. Preskill. Simple proof of security of the bb84 quantum key distribution protocol. *Physical review letters*, 85(2):441, 2000.