Practical Attacks on HB and HB+ Protocols

Zbigniew Golębiewski, Krzysztof Majcher, Filip Zagórski, Marcin Zawada

Practical Attacks on HB and HB+ Protocols *

Zbigniew Gołębiewski, Krzysztof Majcher, Filip Zagórski, and Marcin Zawada

Institute of Mathematics and Computer Science, Wrocław University of Technology
{zbigniew.golebiewski,k.majcher,filip.zagorski,marcin.zawada}@pwr.wroc.pl

Abstract. HB and HB+ are a shared secret-key authentication protocols designed for low-cost devices such as RFID tags. HB+ was proposed by Juels and Weis at Crypto 2005. The security of the protocols relies on the "learning parity with noise" (LPN) problem, which was proven to be NP-hard.
The best known attack on LPN by Levieil and Fouque [13] requires sub-exponential number of samples and sub-exponential number of operations, which makes that attack impractical for the RFID scenario (one cannot assume to collect exponentially-many observations of the protocol execution).
We present a passive attack on HB protocol in detection-based model which requires only linear (in the length of a secret key) number of samples. Number of performed operations is exponential, but attack is efficient for some real-life values of the parameters, i. e. noise $\frac{1}{8}$ and key length 152-bits. Passive attack on HB can be transformed into active one on HB+.
Keywords: lightweight cryptography, RFID, authentication, LPN problem, learning parity with noise, HB, HB+

1 Introduction

The HB/HB+ Scheme HB [12] is a lightweight secret-key protocol for RFID tag identification. It is based on the human-to-computer authentication protocol designed by Hopper and Blum (HB, [11]). The security of the HB/HB+ schemes is provable – it is based on the "learning parity with noise" (LPN) problem, which was proved to be NP-hard [2].

Previous Attacks Over the last years, several attacks on the LPN problem have been proposed. Most of them (e.g. *LF2* from [13] or [14]) are tune-ups of the BKW algorithm (Blum, Kalai, Wasserman 2003) [3].

The BKW algorithm takes a sub-exponential (in the size of the secret key) number of samples and then tries to find out a secret key by adding up sample vectors to obtain vectors from a canonical basis of a vector space. An algorithm

* This paper was supported by funds from Polish Ministry of Science and Higher Education – grant No. N N206 2573 35 .

proposed in [14] manages from a small number of samples to generate exponentially many of them and then use the BKW algorithm.

HB+ is vulnerable to man-in-the-middle attack proposed by Gilbert, Robshaw, and Silbert [7]. Since then, many other schemes have been proposed to design an LPN-based protocol which is secure against man-in-the-middle attacks [4, 15, 9] and some of them have been already broken – see [6, 8, 16].

Our Results We present a different approach to attack LPN problem. Our attack has worse asymptotic time-complexity that the best algorithms solving LPN problem, but it is efficient for real-life sizes of the parameters – parameters that may be used in RFID devices. Some ideas of our algorithm are already used as a sub-protocol in [6] to attack LPN-based protocols.

We need only to collect two successful executions of the HB protocol in order to start an attack (while other solutions need much larger samples). We assume that a single execution of the HB protocol uses parameters suggested in [13], i. e. number of bits sent during a single execution of the protocol is $O(n^2)$, where n is the length of a secret key. Number of bits required by the best known algorithm is $\Omega(2^n)$ while we need only to collect $O(n^3)$ bits (i. e. $O(n)$ protocol executions).

Our first implementation of the algorithm breaks 88-bit HB with noise parameter $\frac{1}{4}$ and 152-bit HB with noise parameter $\frac{1}{8}$. We estimate that algorithm presented is able to practically break HB for noise parameter $\frac{1}{4}$ for keys of the length up to $n = 96$.

Let us also notice that the presented passive attack on HB can be transformed into active one on HB+.

2 Description of the HB and HB+ protocols

The HB Protocol. The Tag and the Reader share public values: $n, \varepsilon, \eta(\varepsilon)$ and a secret key x of the length n. The protocol proceeds in r 2-move rounds as shown in Figure 1: the tag generates a challenge $a^{(i)} \in_R \{0,1\}^n$ and sends it to the tag; the tag computes $(\mathbf{a^{(i)}} \cdot \mathbf{x}) \oplus \nu^{(i)}$, where $\nu^{(i)} \sim Ber\,(\varepsilon)$. The reader authenticates the tag if the number of i's, for which $z^{(i)} \neq a^{(i)} \cdot x$ does not exceed ηr.

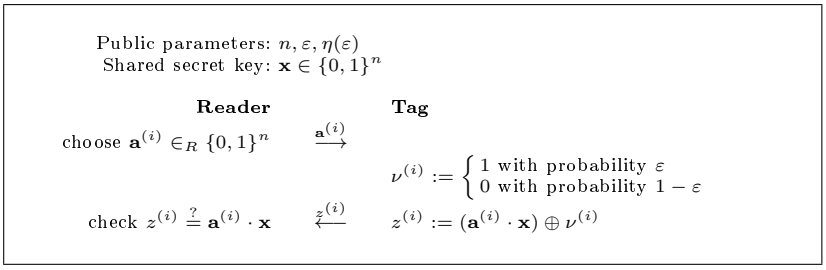

Fig. 1. The i-th round of HB protocol

Efficiency of the HB Efficiency of the HB protocol depends on three values: n, ε, r (in fact $r = r(n, \eta)$). The number of bits sent during an authentication process by the reader is equal to $N_r(n, \eta) = n \cdot r(\eta)$, the tag responds with $N_t = r(\eta)$ bits. Unfortunately, the simplicity in hardware design influences on the communication complexity. The number of bits sent required by a reliable authentication, according to [13], are presented in the table below (all values in KB, $1KB = 8192b$).

Table 1. Number of bits sent during the authentication (in KB)

n	$\eta = 1/20$	$\eta = 1/8$	$\eta = 1/4$
128	4	7	18
512	16	28	73

So, for some parameters of the HB/HB+ protocol, it may take seconds to authenticate even an expensive tag. The meaning of the "high-speed data rate" for RFIDs depends on the manufacturer and varies usually from $20KB/s$ to $40KB/s$. Low-end RFIDs are even 10-times slower.

This leads to the observation that for cheap RFIDs a key length and a number of rounds and thus a noise parameter ε should be adjusted at the relatively low level.

The HB+ Protocol. The HB+ was proposed as a protocol robust against active attacks (while HB is immune against passive attacks). Use of the blinding factor y turns an active attack on HB+ into a passive attack on HB.

In the HB+ scheme the tag and the reader share public values: $n, \varepsilon, r(n, \varepsilon)$ and secret keys x, y. The protocol proceeds in r 3-move rounds as shown on Figure 2.

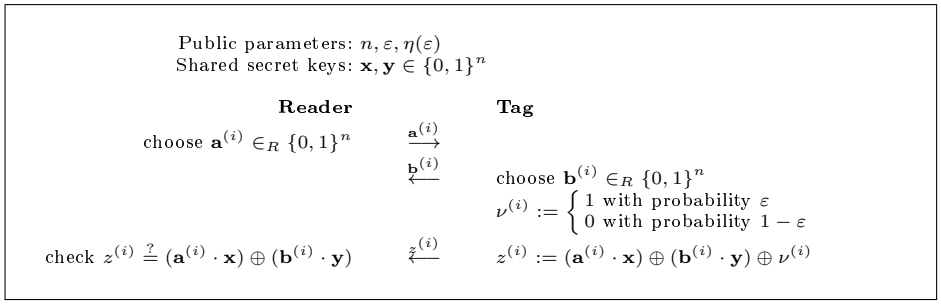

Fig. 2. The i-th round of HB+ protocol

Let us notice that if an attacker wants to break actively the HB+ tag i. e. by sending appropriate values of a, she has to be able to passively break HB.

3 Passive Attacks on HB protocol

Basic notation Let $\mathbf{x} \in \{0,1\}^n$ be a n-bit shared secret between the tag and reader. Suppose that a passive adversary has collected m authentications of the HB protocol. Let us consider that $\mathbf{A} = \{\mathbf{a}_i \in \{0,1\}^n : i = 1, \ldots, m\}$ be a matrix of challenges sent by a reader (each challenge is a row of the matrix) and let $\mathbf{z} = \{z_i \in \{0,1\} : i = 1, \ldots, m\}$ be a vector of collected responses for the tag.

The subset $\mathbf{B} = \{\mathbf{b}_i \,|\, i = 1, \ldots, n\} \subseteq \mathbf{A}$ is called a *basis* of $\{0,1\}^n$ treated as an n-dimensional vector space over $GF(2)$ if vectors $\mathbf{b}_i$ for $i = 1, \ldots, n$ are linearly independent and span whole space $\{0,1\}^n$.

Problem We re-formulate the HB protocol as follows. The reader sends matrix $\mathbf{A}$ of challenges to the tag. The tag responses with a vector $\mathbf{z} = (\mathbf{A} \cdot \mathbf{x}) \oplus \mathbf{v}$, where $\mathbf{v}$ is m-bit vector of "noise". Then the reader checks if $|(\mathbf{A} \cdot \mathbf{x}) \oplus \mathbf{z}| \leq \eta \cdot m$, where $|\cdot|$ is the Hamming weight.

During eavesdropping, an attacker collects samples $\mathcal{S} = (\mathbf{A}, \mathbf{z})$ as a matrix $\mathbf{A}$ of challenges and vector of responses $\mathbf{z}$, therefore the problem of breaking HB is: to find a vector $\mathbf{x}'$ such that $|(\mathbf{A} \cdot \mathbf{x}') \oplus \mathbf{z}| \leq \eta \cdot m$.

Further we show that such $\mathbf{x}'$ has to be equal to the secret-key $\mathbf{x}$ with high probability. This problem is know in the literature as the Learning Parity in the present of Noise (LPN problem).

k-Basis Property Let us assume that we have collected m samples $\mathcal{S} = (\mathbf{A}, \mathbf{z})$ of the HB protocol. Further, we have found such a matrix $\mathbf{B} \subseteq \mathbf{A}$ of size $n \times n$ with vector of responses $\mathbf{z_B}$ such that $\mathbf{B}$ is a basis and vector $\mathbf{z_B}$ has all correct responses ($\mathbf{z_B} = \mathbf{B} \cdot \mathbf{x}$). In such a case we can easily find a secret $\mathbf{x}$. Since we have a system of linear equations over $GF(2)$, thus we can solve it very fast by Gaussian elimination. Let us notice that linear equations have exactly one solution since $\mathbf{B}$ form a basis. The secret can also be found by possessing inverse matrix of $\mathbf{B}$ as follows: $\mathbf{x} = \mathbf{B}^{-1} \cdot \mathbf{z_B}$. However situations that we are capable to find such a basis are quite rare. Thus we introduce the notion of k-basis. A k-basis is a basis with exactly k responses wrong.

Definition 1. *A k-**basis** for a HB protocol instance $(n, \mathbf{x}, \varepsilon, \eta, r)$ and samples $\mathcal{S} = (\mathbf{A}, \mathbf{z})$ is a subset $\mathbf{B} \subseteq \mathbf{A}$ with a vector of responses $\mathbf{z_B}$ which satisfies the following conditions:*

- *$\mathbf{B}$ is a basis of an n-dimensional vector space $\{0,1\}^n$,*
- *$|(\mathbf{B} \cdot \mathbf{x}) \oplus \mathbf{z_B}| = k$.*

We call a *k-basis test* a procedure of verification if both conditions of the definition of k-basis hold.

3.1 Simple Walker Algorithm

Our first algorithm (called a Simple Walker Algorithm) is quite simple probabilistic algorithm which implements the idea presented in previous subsection

i. e. one collects samples and then finds 0-basis. As we mentioned before, possessing 0-basis is equivalent to finding a secret key $\mathbf{x}$. Simple Walker Algorithm can be treated as a slightly different version of the natural brute-force algorithm. However simulations show that even such simple algorithm works quite well in practical settings and there is still room for improvements. In Algorithm 1 we presents pseudo-code of the algorithm which finds secret $\mathbf{x}$ and needs only $m \geq n + C$ samples of the "single authentication step of HB protocol" (Fig. 1), where C is a small constant needed to assure that one can find a basis of n dimensional space in a set of $m = n + C$ vectors each of the length n (for more information see [10]). Input of the algorithm is a set of samples $\mathcal{S} = (\mathbf{A}, \mathbf{z}), n, \varepsilon, \eta)$ and the output is a secret vector $\mathbf{x}$.

Algorithm 1 SimpleWalker$(\mathcal{S} = (\mathbf{A}, \mathbf{z}), n, \varepsilon, \eta)$

1: $m \leftarrow$ length of the vector $\mathbf{z}$
2: find subset $\mathbf{B} \subseteq \mathbf{A}$ such that it is a basis with responses $\mathbf{z_B}$
3: $\mathbf{A'} \leftarrow \mathbf{A} \cdot \mathbf{B}^{-1}$
4: **repeat**
5: $\quad \boldsymbol{\nu} \leftarrow$ choose a random vector $\in \{0,1\}^n$, provided that $|\boldsymbol{\nu}| \sim Bin(n, \varepsilon)$ (i.e. the number of 1's in $\boldsymbol{\nu}$ is Binomially distributed with parameters n, ε)
6: $\quad \mathbf{z'_B} \leftarrow \mathbf{z_B} \oplus \boldsymbol{\nu}$
7: **until** $|(\mathbf{A'} \cdot \mathbf{z'_B}) \oplus \mathbf{z}| \leq \eta \cdot m$
8: **return** $\mathbf{x} \leftarrow \mathbf{B}^{-1} \cdot \mathbf{z'_B}$

The SimpleWalker could be impractical even a few years ago, but since it can be very easily implemented in distributed fashion collecting even small number of samples and access to computers, an adversary can easily find a secret key. Moreover, remarkable progress in multi-core processors makes the HB protocol even more vulnerable to SimpleWalker. Further, it is worth to mention that SimpleWalker has very low memory requirements.

3.2 k-Basis Walker Algorithm

The main drawback of SimpleWalker algorithm is that it is purely probabilistic and do not try to take advantage of using data that were already computed in previous attempts. Therefore, we introduce second algorithm k-BasisWalker. Let us describe the main idea behind the algorithm. The algorithm takes a set of the samples $\mathcal{S} = (\mathbf{A}, \mathbf{z})$, where $\mathbf{A}$ is a set of challenges, $\mathbf{z}$ is a set of responses. Then it divides $\mathcal{S}$ into two parts $(\mathbf{U}, \mathbf{z_U})$ and $(\mathbf{V}, \mathbf{z_V})$. The samples $(\mathbf{U}, \mathbf{z_U})$ are used as a "universe" from which the algorithm picks at random potential 0-Basis. It is called *the testing set* while the samples $(\mathbf{V}, \mathbf{z_V})$ are used for k-Basis-testing and are called *the observations set*. It is important to make this division correctly i.e. in a way that does not change the fraction of incorrect responses. For instance, let $\mathcal{W}(\mathbf{z})$ denote an expected percentage of the incorrect bits in vector $\mathbf{z}$, then the division of the samples set has to satisfy: $\mathcal{W}(\mathbf{z}) = \mathcal{W}(\mathbf{z_U}) = \mathcal{W}(\mathbf{z_V})$.

Notice, that performing such division can be done as follows. Let α be some adjustable parameter. Thus, if one eavesdropped l correct executions of the HB protocol then $\lfloor \alpha l \rfloor$ of these executions could be treated as $(\mathbf{U}, \mathbf{z_U})$ and the rest of executions $\lceil (1-\alpha) l \rceil$ could be treated as $(\mathbf{V}, \mathbf{z_V})$.

As we show later, we need that the sample set has to contain at least $|\mathbf{A}| \geq n + \frac{n}{1-\eta} = O(n)$ vectors. The size of the testing set should be at least of the size of the length of authentication packet n. We also need about $\frac{n}{1-\eta}$ samples (vectors) to be sure that in the sample space there exists at least one 0-Basis. For parameters suggested in [13] and small keys (length smaller than 128), it occurs that our algorithm needs to collect observations from only 2 successful executions of the HB.

Input of the algorithm is a set of samples $\mathcal{S} = (\mathbf{A}, \mathbf{z}), n, \varepsilon, \eta, k)$ and the output is a secret vector $\mathbf{x}$.

Algorithm 2 k-BasisWalker$(\mathcal{S} = (\mathbf{A}, \mathbf{z}), n, \varepsilon, \eta, k)$

1: divide $(\mathbf{A}, \mathbf{z})$ into $(\mathbf{U}, \mathbf{z_U})$ and $(\mathbf{V}, \mathbf{z_V})$
2: $m \leftarrow$ length of the vector $\mathbf{z_V}$
3: **loop**
4: **repeat**
5: $\mathbf{B} \in_R \mathbf{U}$ draw at random n row vectors and the corresponding vector $\mathbf{z_B} \subseteq \mathbf{z_U}$
6: **until** $\mathbf{B}$ is a basis of an n-dimensional vector space $\{0,1\}^n$
7: $\mathbf{V}' \leftarrow \mathbf{V} \cdot \mathbf{B}^{-1}$
8: **for** $1 \leq i_1 \leq n$ **do**
9: $\boldsymbol{\nu} \leftarrow n$-bits vector with 1 at position i_1
10: $\mathbf{z'_B} \leftarrow \mathbf{z_B} \oplus \boldsymbol{\nu}$
11: **if** $|(\mathbf{V}' \cdot \mathbf{z'_B}) \oplus \mathbf{z_V}| \leq \eta \cdot m$ **then**
12: **return** $\mathbf{B}^{-1} \cdot \mathbf{z'_B}$
13: **end if**
14: **end for**
15: $\vdots$
16: **for** $1 \leq i_1 < i_2 < \ldots < i_k \leq n$ **do**
17: $\boldsymbol{\nu} \leftarrow n$-bits vector with 1's at positions $i_1, i_2, \ldots, i_k$
18: $\mathbf{z'_B} \leftarrow \mathbf{z_B} \oplus \boldsymbol{\nu}$
19: **if** $|(\mathbf{V}' \cdot \mathbf{z'_B}) \oplus \mathbf{z_V}| \leq \eta \cdot m$ **then**
20: **return** $\mathbf{B}^{-1} \cdot \mathbf{z'_B}$
21: **end if**
22: **end for**
23: **end loop**

After execution of the above algorithm we get a basis $\mathbf{B}$ and the corresponding set of responses $\mathbf{z_B}$. Because one has to check if the 0-Basis test holds, one has to find a representation of the testing-vectors. It takes a while, so is worth to use the same basis several times. To find a representations of test vectors in a basis $\mathbf{B}$ it takes $O(n^2 \cdot |\mathbf{V}|)$, so it is worth to check if the set $\mathbf{B}$ is 1-Basis or 2-Basis, because checking i-Basis property requires $\binom{n}{i} \cdot |\mathbf{B}|$ operations.

Let us call by 012-*Basis Walker Algorithm (*012-*BWA, BWA)* a modification of the 0-*Basis Walker Algorithm* which checks also 1- and 2-Basis property for every picked set. As we will see later this has a good influence on the efficiency of the algorithm.

4 Algorithm Analysis

First, let us find a probability that k-BasisWalker finds a k-basis (line 4–6 of the Algorithm 2).

Lemma 1. *Let $(n, \mathbf{x}, \varepsilon, \eta, r)$ be a instance of HB protocol. Let $\mathcal{S} = (\mathbf{A}, \mathbf{z})$ be a sample of the HB protocol divided into $(\mathbf{U}, \mathbf{z_U})$ and $(\mathbf{V}, \mathbf{z_V})$ such that $|\mathbf{U}| = t$. Then, the probability that the matrix $\mathbf{B}$ picked uniformly at random from $\mathbf{U}$ is k-basis equals to*

$$p_k = p_B \binom{n}{k} \sum_{j=k}^{\lfloor \eta \cdot t \rfloor} \binom{t-n}{j-k} \varepsilon^j (1-\varepsilon)^{t-j}, \tag{1}$$

where $p_B \approx 0.2887$ denote the probability that randomly chosen set $\mathbf{B}$ is a basis of an n-dimensional vector space $\{0,1\}^n$.

Proof. The probability that random chosen set $\mathbf{B}$ of size n from $\mathbf{U}$ is k-basis can be calculated as follows. Let C_j denotes an event that there are exactly j incorrect responses in $(\mathbf{U}, \mathbf{z_U})$. Then from the Bernoulli trails we have: $\Pr[C_j] = \binom{t}{j}\varepsilon^j(1-\varepsilon)^{t-j}$. Let A be an event that $\mathbf{B} \in_R \mathbf{U}$ is a basis and B_k be an event that $\mathbf{B}$ has exactly k incorrect responses. Thus, the probability that $\mathbf{B}$ is k-basis is equal to $p_k = \Pr[A \wedge B_k]$. By the law of total probability we obtain that $\Pr[A \wedge B_k] = \sum_{j=k}^{\lfloor \eta \cdot t \rfloor} \Pr[A \wedge B_k \wedge C_j] = \sum_{j=k}^{\lfloor \eta \cdot t \rfloor} \Pr[A|B_k \wedge C_j] \cdot \Pr[B_k|C_j] \cdot \Pr[C_j]$.

Since $\mathbf{B}$ is k-basis, then $\mathbf{U}$ must have at least k incorrect responses, thus the sum starts from $j = k$. The upper bound of the sum is $\lfloor \eta \cdot t \rfloor$ because we assume that $\mathbf{U}$ comes from successful authentications. The probability p_B that set B is a basis of $\{0,1\}^n$ is independent on the choices of the responses and $p_B \approx 0.2887$ has been already calculated in the paper [5]. Thus $\Pr[A|B_k \wedge C_j] = p_B$. The probability that one taking n bits from the vector of $\mathbf{z_U}$ responses of length t, takes exactly k wrong responses is equal to $\Pr[B_k|C_j] = \binom{t-j}{n-k} \cdot \binom{j}{k} \cdot \binom{t}{n}^{-1}$. After elementary simplifications, we obtain that $\frac{\binom{t-j}{n-k}\binom{j}{k}}{\binom{t}{n}} \cdot \binom{t}{j} = \binom{n}{k} \cdot \binom{t-n}{j-k}$.

Thus, the proof of the lemma follows. □

The expected value and the variance of basis that should be tested. Let X_k denote a random variable that counts the number of basis that should be tested before at most one k-basis is found. It easy to see that the variable X_k is geometrically distributed with the success probability $p_{X_k} = \sum_{i=0}^{k} p_i$. Thus the expected value for geometrically distributed random variable is $\mathbf{E}[X_k] = 1/p_{X_k}$ and the variance $\mathbf{Var}[X_k] = (1 - p_{X_k})/p_{X_k}^2$. Notice that we are not interested in asymptotic

behavior, since in practice the size of the secret key is at most 512. Thus, in Table 2 we present only the numerical results of $\mathbf{E}[X_k]$ for different value of protocol parameters.

Table 2. The expected number of basis that should be tested in case of the 012-BWA.

	Size of a sample $m = 3 \cdot n$		Size of a sample $m = n^2$	
n	$\varepsilon = 0.125$, $\eta = 0.256$	$\varepsilon = 0.25$, $\eta = 0.348$	$\varepsilon = 0.125$, $\eta = 0.256$	$\varepsilon = 0.25$, $\eta = 0.348$
48	68	24172	44	5271
64	348	$1.39 \cdot 10^6$	167	146704
80	1963	$9.04 \cdot 10^7$	694	$4.55 \cdot 10^6$
96	11865	$6.33 \cdot 10^9$	3062	$1.51 \cdot 10^8$
112	75287	$4.67 \cdot 10^{11}$	14108	$5.30 \cdot 10^9$
128	495413	$3.59 \cdot 10^{13}$	67206	$1.92 \cdot 10^{11}$
144	$3.35 \cdot 10^6$	$2.84 \cdot 10^{15}$	328581	$7.21 \cdot 10^{12}$
160	$2.32 \cdot 10^7$	$2.30 \cdot 10^{17}$	$1.64 \cdot 10^6$	$2.76 \cdot 10^{14}$

Finding Wrong Secrets Now we deal with the problem of getting secret keys different from the searched ones. In the Lemma 3 we show how often a "bad" basis passes the test.

Lemma 2. *Let* $\mathbf{x}$ *be* n*-bit secret key. Let* $\mathbf{A}$ *be a matrix of challenges and* $\mathbf{z_A}$ *be a* m*-bit vector of responses. We assume that* $\mathbf{B} \subseteq \mathbf{A}$ *is a* k*-basis with vector* $\mathbf{z_B} \subseteq \mathbf{z_A}$ *of responses and* $\mathbf{x}' = \mathbf{B}^{-1} \cdot \mathbf{z_B}$ *is a potential secret key. Then*

$$\Pr[(\mathbf{a} \cdot \mathbf{x}) \oplus \nu \neq \mathbf{a} \cdot \mathbf{x}'] = \begin{cases} \varepsilon & \text{if } k = 0, \\ \frac{1}{2} & \text{if } k \geq 1, \end{cases} \tag{2}$$

where $\mathbf{a} \in_R \{0,1\}^n$ *and* ν *is* $0-1$ *random variable such that* $\Pr[\nu = 1] = \varepsilon$.

Proof. By the law of total probability we get

$$\Pr[(\mathbf{a} \cdot \mathbf{x}) \oplus \nu \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] = \\ \Pr[\mathbf{a} \cdot \mathbf{x} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] \cdot \Pr[\nu = 0] + \Pr[\mathbf{a} \cdot \mathbf{x} = \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] \cdot \Pr[\nu = 1].$$

Notice that if $\mathbf{B}$ is 0-basis then $\mathbf{x} = \mathbf{B}^{-1} \cdot \mathbf{z_B}$. Thus $\Pr[\mathbf{a} \cdot \mathbf{x} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] = 0$ and $\Pr[\mathbf{a} \cdot \mathbf{x} = \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] = 1$. Therefore for 0-basis we obtain

$$\Pr[(\mathbf{a} \cdot \mathbf{x}) \oplus \nu \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] = \Pr[\nu = 1] = \varepsilon \ .$$

Let $k > 0$. Consider that $\mathbf{B}$ is k-basis. We need to calculate the probability $\Pr[\mathbf{a} \cdot \mathbf{x} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}]$. Let $\mathbf{z}_{corr} = \mathbf{B} \cdot \mathbf{x}$ be a vector of correct responses. Then

$$\begin{aligned} \Pr[\mathbf{a} \cdot \mathbf{x} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] &= \Pr[\mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z}_{corr} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] \\ &= \Pr[\mathbf{a} \cdot \mathbf{B}^{-1} \cdot (\mathbf{z}_{corr} - \mathbf{z_B}) \neq 0] \ . \end{aligned}$$

Since $\mathbf{z}_{corr} - \mathbf{z_B}$ has 1's on $k \geq 1$ positions and $\mathbf{B}^{-1}$ has linearly independent vectors. Then, by fact that if $(X_i)_{i=1,\ldots,k}$ are independent random 0-1 variables $\Pr[X_i = 1] = 1/2$, then $\Pr[\bigoplus_{i=1}^{k} X_i \neq 0] = 1/2$. Moreover, notice that $\Pr[X_i \oplus \nu = 0] = \Pr[X_i = 0] \cdot \Pr[\nu = 1] + \Pr[X_i = 1] \cdot \Pr[\nu = 0] = (1/2) \cdot \varepsilon + (1-\varepsilon) \cdot (1/2) = 1/2$. Therefore $\Pr[\mathbf{a} \cdot \mathbf{x} \neq \mathbf{a} \cdot \mathbf{B}^{-1} \cdot \mathbf{z_B}] = \frac{1}{2}$. Thus, the proof is complete. □

Lemma 3. *Let $\mathcal{S} = (\mathbf{A}, \mathbf{z})$ be a sample of the HB protocol divided into $(\mathbf{U}, \mathbf{z_u})$ and $(\mathbf{V}, \mathbf{z_v})$ such that $|\mathbf{V}| = m$. Let $\mathbf{B} \subseteq \mathbf{U}$ be a k-basis for $k \geq 1$ with vector $\mathbf{z_B}$ of responses. Thus $\mathbf{x}' = \mathbf{B}^{-1} \cdot \mathbf{z_B}$ is a wrong secret key. Then the probability that $\mathbf{x}'$ passes a test $|(\mathbf{V} \cdot \mathbf{x}') \oplus \mathbf{z}| \leq \eta \cdot m$ is given by $\frac{1}{2^m} \cdot \sum_{i=0}^{\eta \cdot m} \binom{m}{i}$.*

Proof. By Lemma 2 for $k \geq 1$, we obtain that single vector gives us a correct response with probability 1/2. Then the probability p_i that exactly i vectors from $\mathbf{V}$ disagree is equal to $\binom{m}{i} \left(\frac{1}{2}\right)^i \left(1 - \frac{1}{2}\right)^{m-i} = \binom{m}{i} \left(\frac{1}{2}\right)^m$. Therefore, the probability that at most $\eta \cdot m$ out of m vectors passes a test we can obtain by adding the probabilities p_i for $i = 0, 1, \ldots, \eta \cdot m$. □

5 Experimental Results

We have implemented and tested our algorithm for several values. We have broken HB for the parameter $\varepsilon = 0.125, \eta = 0.256, n = 144$ and it took about 3 hours on home PC. For the parameters $\varepsilon = 0.25, \eta = 0.348, n = 80$ it takes on average 10 hours on home PC.

This results and the values in the Table 2 suggest, that we are able to break $n = 96$ bit version of 0.25-HB and $n = 154$ bit version of 0.125-HB protocol.

Parallelization of the presented algorithm is very easy. We are currently working on the CUDA-version of the implementation. First results show that even a cheap GPU allow for about 8-time speedup of a protocol compared to the execution times run on CPU. The new graphic cards that have been recently appeared on the marked can run 8-times more threads than the one which we used for "pre"-testing. Use of GPUs allows to break keys that are few bits longer (≈ 10).

6 Conclusions

We have shown a passive attack for the HB protocol which allow to perform an active attack for HB+ scheme (not man-in-the middle). Our attack needs only $O(n)$ eavesdropped pairs of challenge-response, where n is the length of a secret key, while the best known algorithm *LF2* ([13]) needs exponential number of samples.

References

1. *Approximation, Randomization and Combinatorial Optimization, Algorithms and Techniques, 8th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems, APPROX 2005 and 9th InternationalWorkshop*

on Randomization and Computation, RANDOM 2005, Berkeley, CA, USA, August 22-24, 2005, Proceedings, volume 3624 of *Lecture Notes in Computer Science*. Springer, 2005.

2. E. R. Berlekamp, R. J. McEliece, and H. C. A. van Tilborg. On the inherent intractability of certain coding problems. In *IEEE Trans. Info. Theory*, pages 384–386, 1978.
3. A. Blum, A. Kalai, and H. Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *Journal of the ACM, vol. 50, no. 4*, pages 506–519, 2003.
4. Julien Bringer, Hervé Chabanne, Tom A. M. Kevenaar, and Bruno Kindarji. Extending match-on-card to local biometric identification. In *COST 2101/2102 Conference*, volume 5707 of *Lecture Notes in Computer Science*, pages 178–186. Springer, 2009.
5. Jacek Cichon, Marek Klonowski, and Miroslaw Kutylowski. Privacy protection for rfid with hidden subset identifiers. In *Pervasive Computing*, 2008.
6. Dmitry Frumkin and Adi Shamir. Un-trusted-hb: Security vulnerabilities of trusted-hb. Cryptology ePrint Archive, Report 2009/044, 2009.
7. H. Gilbert, H. Sibert, and M. Robshaw. An active attack against a provably secure lightweight authentication protocol. In *IEEE Electronic Letters 41*, pages 1169–1170, 2005.
8. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good variants of hb^{+} are hard to find. In Gene Tsudik, editor, *Financial Cryptography*, volume 5143 of *Lecture Notes in Computer Science*, pages 156–170. Springer, 2008.
9. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. $\mathrm{Hb}^{\#}$: Increasing the security and efficiency of hb^{+}. In *EUROCRYPT*, volume 4965 of *Lecture Notes in Computer Science*, pages 361–378. Springer, 2008.
10. Zbigniew Golebiewski, Krzysztof Majcher, and Filip Zagórski. Attacks on ckk family of rfid authentication protocols. In David Coudert, David Simplot-Ryl, and Ivan Stojmenovic, editors, *ADHOC-NOW*, volume 5198 of *Lecture Notes in Computer Science*, pages 241–250. Springer, 2008.
11. Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. *Lecture Notes in Computer Science*, 2248, 2001.
12. Ari Juels and Stephen A. Weis. *Authenticating Pervasive Devices with Human Protocols*, volume 3621. November 2005.
13. Éric Levieil and Pierre-Alain Fouque. An improved lpn algorithm. In Roberto De Prisco and Moti Yung, editors, *SCN*, volume 4116 of *Lecture Notes in Computer Science*, pages 348–359. Springer, 2006.
14. Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM* [1], pages 378–389.
15. J. Munilla and A. Peinado. Hb-mp: A further step in the hb-family of lightweight authentication protocols. *Comput. Netw.*, 51(9):2262–2267, 2007.
16. Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of hb# against a man-in-the-middle attack. In Josef Pieprzyk, editor, *ASIACRYPT*, volume 5350 of *Lecture Notes in Computer Science*, pages 108–124. Springer, 2008.