NLHB: A Non-Linear Hopper-Blum Protocol

Mukundan Madhavan and Andrew Thangaraj
Dept of Electrical Engg, IIT Madras
Chennai, India
Email: andrew@ee.iitm.ac.in

Yogesh Sankarasubramanian and Kapali Viswanathan
HP Labs India
Bangalore, India
Email: yogesh,kapali@hp.com

***Abstract*—The Hopper-Blum (HB) protocol, which uses noised linear parities of a shared key for authentication, has been proposed for light-weight applications such as RFID. Recently, algorithms for decoding linear codes have been specially designed for use in passive attacks on the HB protocol. These linear coding attacks have resulted in the need for long keys in the HB protocol, making the protocol too complex for RFID in some cases. In this work, we propose the NLHB protocol, which is a non-linear variant of the HB protocol. The non-linearity is such that passive attacks on the NLHB protocol continue to be provably hard by reduction. However, the linear coding attacks cannot be directly adapted to the proposed NLHB protocol because of the non-linearity. Hence, smaller key sizes appear to be sufficient in the NLHB protocol for the same level of security as the HB protocol. We construct specific instances of the NLHB protocol and show that they can be significantly less complex for implementation than the HB protocol, in spite of the non-linearity. Further, we propose an extension, called the NLHB+ protocol, that is provably secure against a class of active attack models.**

I. Introduction

The HB protocol was proposed in [1] as a low-complexity authentication algorithm that can be computed by human users. Its security is based upon the hardness of the "Learning Parity in Noise" (LPN) problem [2], which is known to be NP-Hard. Though the protocol is secure against passive attacks, it was found to be vulnerable to active attacks [3]. Juels and Weis [3] proposed the HB$^+$ protocol as an alternative that could resist certain active attacks. The added complexity of the HB$^+$ protocol rendered it more suitable for low-complexity RFID tags rather than human users.

Cryptanalysis of the HB authentication protocol has resulted in efficient solutions to the LPN problem. Notably, Levieil and Fouque [4] proposed the LF2 algorithm, which is an improved form of the BKW algorithm [5] for solving the LPN problem. Later, Carrijo *et al.* [6] proposed a probabilistic passive attack against HB and HB$^+$ protocols. These new solutions have significantly reduced the effective key-size of the HB protocol family that depend on the hardness of decoding linear codes for security against passive adversaries.

In this paper, we define and consider the UNLD problem, which is a decoding problem for a specific class of non-linear codes. We prove hardness of UNLD by reducing the LPN problem to the UNLD problem. Following this, we propose the NLHB protocol, which is a carefully constructed variant of the HB protocol. The basic idea behind the NLHB protocol is the use of a carefully-chosen non-linear Boolean function on the linear parities generated in the HB protocol. We prove the passive attack security of NLHB by reducing the UNLD problem to the passive attack problem. So, security of NLHB is based on the UNLD problem. On the practical side, the use of the non-linear function considerably weakens the effectiveness of passive attacks like LF2 [4] that depend on the linearity of the parities. Therefore, key efficiency is higher in NLHB when compared to HB. For implementation, we demonstrate a certain quadratic form chosen from the general family of functions that we propose for the NLHB, which presents a specific low-cost candidate for the protocol. Using this candidate function, the complexity of the NLHB protocol is low enough that it can be implemented in low-cost devices such as RFIDs. Finally, we show that the Prover/Verifier complexity of NLHB protocol can be lower than that of the HB protocol because of the use of smaller keys.

Active attacks similar to those on the HB protocol are possible on the basic NLHB protocol. We demonstrate that the basic NLHB protocol can be extended to an NLHB$^+$ protocol, in the spirit of HB$^+$, for security in some active attack models. We show that the reductions for the HB$^+$ protocol as shown in [7], [8] work for the NLHB$^+$ protocol as well.

In summary, the main contribution of this paper is a low-cost, provably-secure extension of the HB protocol through the use of simple non-linear functions on parities that has better resistance to known passive attacks on the HB family resulting in higher key efficiency and cheaper implementations. Also, the NLHB protocol can be modified in the spirit of the several known modifications of the HB protocol to obtain better security against different classes of active attacks.

The paper is organized as follows. In Section II, we give a brief introduction to the HB and HB$^+$ protocols, related security models and the LPN problem. In Section III, we describe the UNLD problem, a type of non-linear code decoding problem and prove its NP-Hardness. This is followed by a description of the NLHB protocol and its security proofs. Section IV contains discussions on the resistance of the protocol to passive attacks and its Prover complexity. in Section V, we propose the NLHB$^+$ protocol and give its security proofs . Section VI concludes the paper.

II. The HB And HB$^+$ Protocols

The HB protocol is a symmetric-key authentication protocol. The Prover and Verifier share a random k-bit secret key $\mathbf{s}$. The protocol has two public probability parameters $\epsilon, \epsilon' \in \ ]0, \frac{1}{2}[$ such that $\epsilon < \epsilon'$. To authenticate, the Verifier

sends a random k-bit challenge vector $\mathbf{a}$. The Prover, in turn, calculates the binary dot-product $\mathbf{s}.\mathbf{a}$ and replies to the Verifier with $z = \mathbf{s}.\mathbf{a} + v$, where v is a Bernoulli random variable that takes the value 1 with probability ϵ and $+$ denotes XOR addition. This process is repeated n times. At the end of n repetitions, the Verifier returns an "Accept" message if at most $\epsilon' n$ responses are "wrong", i.e, different from dot-products of the secret and the corresponding challenges.

This process, which constitutes one authentication session can be parallelized as shown in Figure 1. In the parallelized

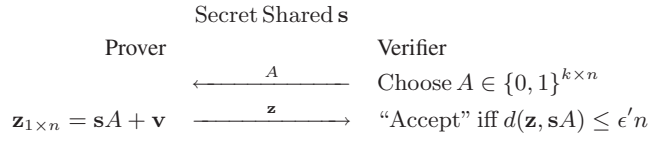

Fig. 1. Parallelized version of the HB protocol

form, the Verifier challenges the Prover with a random $k \times n$ matrix, to which the Prover responds with $\mathbf{z} = \mathbf{s}A + \mathbf{v}$. Here, the bits of the vector $\mathbf{v}$ are *iid* binary random variables with $\Pr(1) = \epsilon$. Such a vector $\mathbf{v}$ is called a Bernoulli noise vector with parameter ϵ. All bit operations are over the binary field $GF(2)$ in this article. The Verifier responds with "Accept" if $d(\mathbf{z}, \mathbf{s}A) \leq \epsilon' n$, where $d(\cdot, \cdot)$ denotes Hamming distance. The parameters ϵ,ϵ', and n are fixed so that both the probability of rejecting an honest Prover as well as the probability of positively authenticating an attacker giving random responses are negligible [4, Fig. 2].

The HB Protocol has been proved secure in the Passive attack model as defined below.

Definition 1: (*Passive attack model* [3], [7]) In this model, the adversary algorithm is two-phased. In the first phase (called the query phase), the adversary has access to the transcripts from several authentication sessions between an honest Prover and Verifier. In the second phase (called the cloning phase), the adversary tries to impersonate an honest Prover to the Verifier.

A. HB$^+$ Protocol

The HB protocol is not secure against active attacks as shown in [3]. To counter active attacks, the HB$^+$ protocol was proposed in [3]. Instead of a single secret, the Prover and Verifier share two k-bit secret keys $\mathbf{s}_1$ and $\mathbf{s}_2$.

The parallel version of the HB$^+$ protocol is shown in Fig. 2. In its parallel form, the HB$^+$ protocol can be described as follows. The Prover starts an authentication session by sending a random "blinding" matrix B to the Verifier, which in turn replies with a random challenge matrix A. On receiving A, the Prover responds with $\mathbf{z} = \mathbf{s}_1 B + \mathbf{s}_2 A + \mathbf{v}$. Here, A and B are $k \times n$ matrices, and $\mathbf{v}$ has the same definition as in the HB protocol. The Verifier responds with an "Accept" decision if $d(\mathbf{z}, \mathbf{s}_1 B + \mathbf{s}_2 A) \leq \epsilon' n$.

The HB$^+$ protocol is secure against both passive attacks as well as active attacks in a model known as the "DET" attack model.

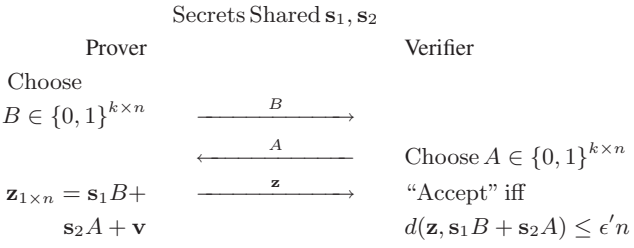

Fig. 2. Parallelized version of the HB$^+$ protocol

Definition 2: (*DET Attack Model* [3], [8]) In this model, attacks are two-phased. In the first (query) phase, the adversary can interact with an honest Prover several times. In the second (cloning) phase, the adversary interacts with the Verifier and attempts impersonation.

In settings where the Verifier reports repeated authentication failures from a Prover, the "DET" model is found to be suitable and more practical [3].

B. The LPN Problem and Passive Attacks

Definition 3: (*LPN Problem* [3]) Let $\mathbf{s}$ be a random binary k-bit vector. Let $\epsilon \in]0, \frac{1}{2}[$ be a constant error parameter. Let A be a random $k \times n$ matrix, and let $\mathbf{v}$ be a random n-bit vector such that $\mathrm{wt}(\mathbf{v}) \leq \epsilon n$, where $\mathrm{wt}(\mathbf{v})$ denotes the Hamming weight of $\mathbf{v}$. Given A, ϵ and $\mathbf{z} = (\mathbf{s}A) + v$, find a k-bit vector $\mathbf{s}'$ such that $d(\mathbf{z}, \mathbf{s}'A) \leq \epsilon n$.

The LPN problem has been proved to be both NP-Hard [2] and is conjectured to be average-case hard [1]. The LPN problem can be reduced to forging the HB protocol with a high probability of success in the passive attack model [3] [7], and this proves passive attack security for the HB protocol. The best-known passive attacks on HB employ algorithms to solve the LPN problem [4] [6].

The basic idea behind the LF2 attack in [4] is to add columns of A (and corresponding noisy responses) so that the resulting columns are non-zero only in a small set of rows. Since only the key bits corresponding to these rows will affect the new responses (during matrix multiplication), the attacker can now find these key bits alone by doing exhaustive search over a smaller key space relevant to this new set of equations. Thus the whole key is found in parts. (Refer [4] for more details). Later, a probabilistic attack on the LPN problem was proposed in [6]. The basic idea here is to pick a few bits out of the n response bits and find the key through Gaussian elimination, hoping that the picked response bits are all noise-free.

As a consequence of these attacks, a LPN instance using as many as 512 bits of secret can be attacked with a complexity of just 2^{80} operations. This results in the reduction of effective key size for the HB protocol - to get 80-bit security, 512 bits of key are required. A key size of 512 bits limits the applicability of the HB protocol in several RFID applications.

III. The UNLD Problem and the NLHB Protocol

The main idea in this paper is to replace the linear parity generation part sA in the HB protocol with a non-linear version $f(sA)$ for a suitable public function $f : \{0,1\}^n \rightarrow \{0,1\}^D$ for an integer D. The following characteristics are desirable for such a function f:

1) The function f, assumed to be public, must allow for the reduction of hardness of decoding problems to the passive attacks.
2) The function f must be simple enough to implement on low-cost devices.
3) The function f must provide better resistance to known passive attacks that solve the LPN problem.
4) The function f should allow extensions such as HB$^+$ for security against active attacks.

We now describe a specific class of non-linear Boolean vector functions for this purpose, and discuss some of its properties that will be used in the security reductions.

A. The Function f

Let $D = n - p$ for a positive integer p. For $\mathbf{x} \in \{0,1\}^n$, $\mathbf{y} = [y_1\ y_2 \cdots y_D] = f(\mathbf{x})$ is defined by

$$y_i = x_i + g([x_{i+1}, .., x_{i+p}]), \tag{1}$$

where $g : \{0,1\}^p \rightarrow \{0,1\}$ is Boolean function containing strictly non-linear terms in the variables $\{x_{i+1}, .., x_{i+p}\}$. One of the important properties of the function f is the following: for uniformly distributed $\mathbf{x} \in \{0,1\}^n$, $f(\mathbf{x})$ is uniformly distributed in $\{0,1\}^D$. A proof of this provided can be found in [9], and is omitted here.

For $p = 3$, a specific example for the function f is as follows:

$$y_i = x_i + x_{i+1}x_{i+2} + x_{i+2}x_{i+3} + x_{i+3}x_{i+1}, 1 \leq i \leq D. \tag{2}$$

As we can see, such functions add very low additional complexity (only 3 AND gates and 3 XOR gates in this case) to an implementation of a linear HB protocol.

B. UNLD Problem

The set of vectors $\{f(\mathbf{s}A) : \mathbf{s} \in \{0,1\}^k\}$ can be viewed as a non-linear code. We now define the UNLD problem, which (in words) is the problem of decoding this class of non-linear codes.

Definition 4 (UNLD Problem): Let $\mathbf{s}$ be a random k-bit binary vector. Let $\epsilon \in]0, \frac{1}{2}[$ be a constant error parameter. Let A be a random $k \times n$ binary matrix and let $\mathbf{v}$ be a random D-bit vector such that $\mathrm{wt}(\mathbf{v}) \leq \epsilon D$, where $\mathrm{wt}(\mathbf{v})$ denotes the Hamming weight of $\mathbf{v}$. Given A, ϵ and $\mathbf{z} = f(\mathbf{s}A) + \mathbf{v}$, find the k-bit vector $\mathbf{s}$.

We prove the hardness of the UNLD problem by reducing a random instance of the NP-Hard LPN problem to the UNLD problem. To show the reduction, we construct an algorithm S, which can solve a random LPN instance, when given access to an algorithm X that can solve the UNLD problem.

Theorem 1 (LPN reduces to UNLD): Let A be a random $k \times n$ matrix, $\mathbf{v}'$ be a $(n-p)$-bit Bernoulli noise vector, and $\mathbf{s}$ be a random k-bit vector. Suppose there exists a probabilistic polynomial-time (PPT) algorithm X with input $\langle A, \mathbf{y}_{(n-p)} = f(\mathbf{s}A) + \mathbf{v}' \rangle$ that can output $\mathbf{s}$ with probability at least δ. Then, there also exists a PPT algorithm S that can solve a random LPN problem instance $\langle G_{k \times n'}, \mathbf{z} = \mathbf{m}G + \mathbf{v} \rangle$ for randomly chosen $\mathbf{m}$, Bernoulli noise vector $\mathbf{v}$ and $n' \leq \frac{(n-1)}{p}, k < n'$ with probability at least δ.

Proof: Let $\mathbf{z} = [z_1, ..., z_{n'}]$ and $\mathbf{v} = [v_1, ..., v_{n'}]$ be the constituent bits of the vectors described above. The algorithm S, having access to algorithm X works as follows to solve a random LPN instance $\langle G, \mathbf{z} \rangle$ passed to it.

1) Pick r_i for $1 \leq i \leq n' - 1$ such that $r_i \geq (p-1), \sum_{i=1}^{n'-1} r_i = n - p - n'$.
2) Insert r_i Bernoulli bits between bit z_i and z_{i+1} of $\mathbf{z}$ for $1 \leq i \leq n'-1$. This gives rise to the vector $\mathbf{y}_{(n-p)} = [z_1,\ b_1\, b_2 \ldots b_{r_1},\ z_2,\ b_{r_1+1} \ldots b_{r_1+r_2},\ z_3,\ .....b_{n-p-n'},\ z_{n'}]$.
3) Insert r_i columns of zeros in between columns i and $i+1$ of G $(1 \leq i \leq n' - 1)$ to get the matrix A. Insert p columns of zeros after the last column of A. Now, the dimension of A is $k \times n$ and A is of the form $A = [\mathbf{g}_1\underline{00}..\underline{0}\mathbf{g}_2\underline{00}..\underline{0}.....\mathbf{g}_{n'}\underline{00}..\underline{0}]$, where $\mathbf{g}_i$ are the columns of G.
4) Pass $\langle A, \mathbf{y} \rangle$ to X and get back $\mathbf{m}'$.
5) Return $\mathbf{m}'$ as the estimate of the LPN secret $\mathbf{m}$.

We now show that S succeeds with probability at least δ. Consider the vector $\overline{\mathbf{x}} = \mathbf{m}A$. We can see that $\overline{\mathbf{x}} = [x_1 00..0 x_2 00..0 x_3 00.0....x_{n'} 00..0]$, where $[x_1, x_2, ..., x_{n'}]$ are the bits of $\mathbf{x} = \mathbf{m}G$. We also see that, since g has only non-linear terms (i.e each term in g is some kind of product of at least two input bits) and $r_i \geq (p-1)$, the vector $f(\overline{\mathbf{x}})$ can be written as $f(\overline{\mathbf{x}}) = [x_1 00..0 x_2 00..0 x_3 .....00 x_{n'}]$, as all the product terms from g go to zero.

So, the vector $\mathbf{y}$ is of the form $f(\overline{\mathbf{x}}) + \mathbf{v}'$, where $\mathbf{v}' = [v_1,\ b_1\, b_2 \ldots b_{r_1},\ v_2,\ b_{r_1+1} \ldots b_{r_1+r_2},\ v_3,\ .....b_{n-p-n'},\ v_{n'}]$. Here, v_i are the Bernoulli bits since they are part of the LPN noise vector $\mathbf{v}$ and b_i are picked to be Bernoulli bits. In other words, $\mathbf{y} = f(\mathbf{m}A) + \mathbf{v}'$, where $\mathbf{v}'$ is a Bernoulli noise vector. Hence, by definition, X will return $\mathbf{m}$ with probability at least δ. Since S succeeds whenever X succeeds, the probability of success of S is at least δ. ■

C. NLHB Protocol

In the parallel version of the proposed NLHB protocol, the Prover and Verifier share a k-bit secret $\mathbf{s}$. The Verifier transmits a random $k \times n$ challenge matrix A to the Prover. On receiving this, the Prover computes a D-bit vector $\mathbf{z} = f(\mathbf{s}A) + \mathbf{v}$, where $\mathbf{v}$ is a D-bit Bernoulli noise-vector with parameter ϵ. The Verifier returns "Accept", if $d(\mathbf{z}, f(\mathbf{s}A)) \leq \epsilon' D$. The parameters (D, ϵ, ϵ') have to satisfy the conditions satisfied by the HB protocol parameters (n, ϵ, ϵ') (See [4]).

D. Security Proofs For NLHB In Passive Model

The proof of security for NLHB in the passive model involves reductions from the UNLD problem to the forging

of the NLHB protocol in the passive model. The proofs are broadly based on the proof of security given for the HB protocol in [7] [8], with suitable modifications and additions to support the function f. The proofs and details are provided in a longer version of this article available on line [9]. We provide a brief overview for completeness.

If the i^{th} bit of $\mathbf{s}$, $s_i = 1$, modification of the i-th row of the challenge matrix A by adding a random vector results in a uniform distribution for $\mathbf{s}A + \mathbf{v}$. However, if $s_i = 0$, the distribution is different from uniform. Detecting this change in distribution is the central idea used in security reductions for the HB protocol [3] [7]. The following lemma establishes the above result for NLHB using the properties of the function f.

Lemma 1: Let A be a randomly chosen $k \times n$ matrix. Let $\mathbf{s}$ be a random k-bit binary secret vector. Further, assign $\mathbf{z} = f(\mathbf{s}A) + \mathbf{v}$, where the bits of $\mathbf{v}$ are *i.i.d* Bernoulli distributed. Now, let $\mathbf{c}$ be a randomly chosen (independent of all other factors) n-bit binary vector. For an arbitrary $1 \leq i \leq k$, let A' denote the matrix formed by modifying only the i-th row of A as $(A)_i = (A)_i + \mathbf{c}$. If hyb_i denotes the distribution of the bit-string $\langle A', \mathbf{z} \rangle$, then $hyb_i = U_{kn+D}$ if $s_i = 1$.

Lemma 1 enables the use of techniques used in the security reductions of HB to be used for security reductions in NLHB. The security reductions are given in the following two steps.

1) In the first step, we prove a reduction from the UNLD problem to the problem of distinguishing between U_{kn+D}, the distribuion representing $(kn + D)$-length uniformly distributed bitstrings and $\mathcal{A}_{\mathbf{s},\epsilon,f}$, the distribution followed by the strings from the NLHB protocol transcripts, A and $f(\mathbf{s}A) + \mathbf{v}$ concatenated together.
2) In the second step, we provide a reduction from the problem of distinguishing between the distributions to the problem of forging the NLHB protocol.

For more details, see [9].

IV. Implementation and Efficiency

Using the specific f in (2), we will show how the existing LF2 attack [4] on LPN is ineffective on the NLHB protocol. Let $\mathbf{x} = [x_1, ..., x_n] = \mathbf{s}A = [\mathbf{s}.\mathbf{a}_1, ..., \mathbf{s}.\mathbf{a}_n]$, where $[\mathbf{a}_1, ..., \mathbf{a}_n]$ are columns of A. Let $\mathbf{y} = f(\mathbf{x})$. Then, the passive adversary to NLHB has access to $\mathbf{z} = \mathbf{y} + \mathbf{v}$.

The LF2 (or BKW) algorithm works by repeatedly adding the columns of the matrix A and obtaining the response corresponding to this new matrix by adding the responses corresponding to the individual columns. We examine the result when the attacker does one column addition. Let the attacker modify A into $A' = [\mathbf{a}_1, ..., \mathbf{a}_j + \mathbf{a}_k, ..., \mathbf{a}_n]$, i.e, he adds the k^{th} column to the j^{th} column. The corresponding matrix product between $\mathbf{s}$ and A' will be $\overline{\mathbf{x}} = [x_1, x_2, ..., x_j + x_k, ..., x_n]$, i.e $\overline{\mathbf{x}}$ has the same bits as $\mathbf{x}$ except at the j^{th} position, where it is $x_j + x_k$. Let $\overline{\mathbf{y}} = f(\overline{\mathbf{x}})$ and let $E_i = y_i + \overline{y}_i$. As can be seen, $E_i = 0$ for all i except for $i \in \{j-3, j-2, j-1, j\}$. The following relationships are readily established.

$$\begin{aligned} E_{j-3} &= x_{j-1}x_k + x_k x_{j-2}, \\ E_{j-2} &= x_{j-1}x_k + x_k x_{j+1}, \\ E_{j-1} &= x_{j+1}x_k + x_k x_{j+2}, \\ E_j &= x_k. \end{aligned}$$

Each error term above is an unknown bit to the attacker, since he does not have access to either a noised or unnoised version of these terms. So, the attacker has to guess the error bits $E_{j-3}, E_{j-2}, E_{j-1}, E_j$ that need to be added to the new response to get the right responses corresponding to the new matrix A'. The amount of uncertainty involved in guessing these bits can be found from the entropy of $[E_{j-3}, E_{j-2}, E_{j-1}, E_j]$. Since the bits x_i are uniformly distributed, it can easily be seen that this entropy is equal to 2.5 bits. So each time a column is added, the attacker has to guess 2.5 bits on an average. Since there are many such additions needed in the LF2 attack, this attack does not easily extend longer to the NLHB protocol.

In Table I, we give the values of the entropy of error bits in the LF2 attack for different function choices with $p = 2, 3, 4$. For $p = 3, 4$, there are other functions that achieve the maximum entropy of 2.5 and 3, respectively. As we can see, the entropy increases with increase in p, meaning that LF2 attacks are harder for higher p. Similar arguments can be given for the infeasibility of the Imai [6] attack, that also relies heavily on linearity. The infeasibility of existing passive attacks on the related HB protocol indicates that the NLHB protocol can achieve 80-bit security using key sizes smaller than 512 bits, which is the number of key bits needed by the HB protocol for 80-bit security. In Table IV, we compare the number of bit multiplications and additions needed by an implementation of NLHB protocol with the function f in (2) with that of a comparable HB protocol. We see that prover complexity in NLHB is significantly lesser than in a comparable HB protocol.

V. NLHB$^+$ Protocol

Though the NLHB protocol is secure against a passive adversary, it is not secure against an active attacker. An efficient active attack similar to the one demonstrated against HB can also be mounted on the NLHB protocol. So, in the spirit of the HB$^+$ protocol, we propose the NLHB$^+$ protocol to provide security in the DET model.

Figure 3 shows the NLHB$^+$ protocol. Here, the Prover and Verifier share two secrets $\mathbf{s}_1$ and $\mathbf{s}_2$. The authentication session is started when the Prover transmits a random $k \times n$ blinding matrix B to the Verifier, which responds with a random $k \times n$ challenge matrix A. The Prover responds with $\mathbf{z} = f(\mathbf{s}_1 B) + f(\mathbf{s}_2 A) + \mathbf{v}$, where f and $\mathbf{v}$ are as defined in the NLHB protocol. The Verifier replies with "Accept" if $d(\mathbf{z}, f(\mathbf{s}_1 B) + f(\mathbf{s}_2 A)) \leq \epsilon' D$. NLHB$^+$ depends on the hardness of the UNLD problem for security against active and passive attacks. In addition, NLHB$^+$ is secure against active attacks in the "DET" model as outlined in the next section.

p	Function Achieving Maximum entropy for given p	Maximum Entropy Achieved for given p
2	$y_i = x_i + x_{i+1}x_{i+2}$	2
3	$y_i = x_i + x_{i+1}x_{i+2} + x_{i+1}x_{i+3}$	2.5
4	$y_i = x_i + x_{i+1}x_{i+4} + x_{i+2}x_{i+3}$	3

TABLE I
MAXIMUM ENTROPY ACHIEVED OVER ALL FUNCTIONS FOR A GIVEN p AND THE FUNCTION ACHIEVING THIS MAXIMUM

.

	k	ϵ	ϵ'	Size of Challenge Matrix	Length Of Prover Response	Scalar Multiplications	Scalar Additions
HB	512	.25	.348	512×1164	n=1164	595968	594804
NLHB	128	.25	.348	128×1167	D=1164	152868	151701

TABLE II
COMPARISON OF PROVER/VERIFIER COMPLEXITIES BETWEEN NLHB AND HB FOR f WITH $p = 3$, FALSE-REJECT PROBABILITY $P_{FR} = 2^{-40}$ AND FALSE-ACCEPT PROBABILITY $P_{FA} = 2^{-80}$ AND 80-BIT SECURITY.

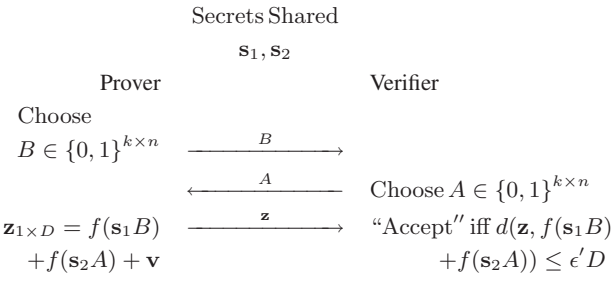

Fig. 3. Parallelized version of the NLHB$^+$ protocol

A. Security Proof for NLHB$^+$ In the "DET" Model

The security proof for NLHB$^+$ in the "DET" model is from the problem of differentiating the distributions $\mathcal{A}_{\mathbf{s},\epsilon,f}$ and U_{kn+D} of Section III-D to active attacks on the NLHB$^+$ protocol. The strategy for the proof is broadly based on the proofs given in [8]. See [9] for details.

1) Algorithm Z_+: This is a two-phased polynomial-time NLHB$^+$ adversary. In its query phase, it takes a $k \times n$ random matrix B as input, responds with a challenge matrix A (which can be non-random) and receives $\mathbf{z} = f(\mathbf{s}_1 B) + f(\mathbf{s}_2 A) + \mathbf{v}$ for secrets $\mathbf{s}_1$ and $\mathbf{s}_2$. In the challenge phase, it sends a random blinding matrix $\hat{B}$ to the NLHB$^+$ Verifier, receives a challenge matrix $\hat{A}$ from the Verifier and generates a response $\hat{\mathbf{z}}$ that can generate "Accept" from the NLHB$^+$ Verifier.
2) $Adv_{Z_+}^{\mathrm{NLHB}^+ attack}(k, \epsilon, u, f)$ denotes the probability of success of Z_+. The advantage is a function of the parameters k, ϵ, u, f.

Theorem 2: If for some polynomial-time adversary Z_+, $Adv_{Z_+}^{\mathrm{NLHB}^+ attack}(k, \epsilon, u, f)$ is non-negligible, then the UNLD problem can be efficiently solved.

The important steps in the proof are simulating a prover to the adversary Z_+ in the query phase and using rewinding of Z_+ to achieve our purpose in its challenge phase. Details can be found in [9].

VI. CONCLUSION AND FUTURE WORK

In this paper, we have proved the hardness of a non-linear decoding problem that we call the UNLD problem and proposed the NLHB and NLHB$^+$ authentication protocols, which are variants of the HB and HB$^+$ protocols. These new protocols have better passive attack security than the HB and HB$^+$ protocols. They are light-weight and have lesser complexity of implementation than a comparable HB protocol for the same security level.

In the future, it would be interesting to examine if any non-linear adaptations of existing attacks are possible on the NLHB protocol family. Also, it would be interesting to see if the MIM attacks [10], [11] (part of a prevention-based attack model) on the HB family of protocols can be prevented by making appropriate changes to the NLHB protocol.

REFERENCES

[1] N. Hopper and M. Blum, "A secure human-computer authentication scheme," Carnegie Mellon University, Tech. Rep. CMU-CS-00-139, 2000.
[2] E. Berlekamp, R. McEliece, and H. van Tilborg, "On the inherent intractability of certain coding problems (corresp.)," *Information Theory, IEEE Transactions on*, vol. 24, no. 3, pp. 384–386, May 1978.
[3] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Advances in Cryptology CRYPTO 2005*, ser. Lecture Notes in Computer Science, vol. 3621, 2005, pp. 293–308.
[4] É. Levieil and P.-A. Fouque, "An improved LPN algorithm," in *Security and Cryptography for Networks (SCN) 2006*, ser. Lecture Notes in Computer Science, vol. 4116. Springer-Verlag, 2006, pp. 348–359.
[5] A. Blum, A. Kalai, and H. Wasserman, "Noise-tolerant learning, the parity problem, and the statistical query model," *J. ACM*, vol. 50, no. 4, pp. 506–519, 2003.
[6] J. Carrijo, R. Tonicelli, H. Imai, and A. C. A. Nascimento, "A novel probabilistic passive attack on the protocols HB and HB+," *IEICE Transactions*, vol. E92-A, no. 2, pp. 658–662, 2009.
[7] J. Katz and J. S. Shin, "Parallel and concurrent security of the HB and HB+ protocols," in *Advances in Cryptology - EUROCRYPT 2006*, ser. Lecture Notes in Computer Science, vol. 4004. Springer Berlin / Heidelberg, 2006, pp. 73–87.
[8] J. Katz and A. Smith, "Analyzing the HB and HB+ protocols in the "large error" case," Cryptology ePrint Archive, Report 2006/326, 2006, http://eprint.iacr.org/2006/326.pdf.
[9] M. Madhavan, A. Thangaraj, Y. Sankarasubramaniam, and K. Viswanathan, "NLHB: A non-linear Hopper Blum protocol," ArXiv.org, 2010, http://arxiv.org/abs/1001.2140.
[10] H. Gilbert, M. Robshaw, and H. Sibert, "Active attack against HB+: a provably secure lightweight authentication protocol," *Electronics Letters*, vol. 41, no. 21, pp. 1169–1170, Oct. 2005.
[11] 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*, ser. Lecture Notes in Computer Science, vol. 5350. Springer Berlin / Heidelberg, 2008, pp. 108–124.