

The Strong HB Problem and its Applications

Anonymized for submission

May 1, 2009

Abstract

The HB problem first introduced by Blum and Hopper has been the basis for extremely lightweight authentication protocols for RFID tags [18, 19]. In this paper we introduce a variant of this problem which we call the *strong HB problem*. We analyze the strong HB problem and give some arguments that support its hardness. We then use the strong HB assumption in two applications of independent interest.

First, while the HB problem has been the basis of several lightweight protocols for RFID authentication, none of these protocols have proofs of security against fully adaptive man-in-the-middle attacks. We improve on the HB[#] protocol [15] using the strong HB assumption. Our protocol is two rounds less than HB[#], with similar efficiency otherwise, and can be proven secure against man-in-the-middle attacks. In addition, we create a related-key secure non-adaptive MAC based on our improved version of the HB[#].

1 Introduction

In the learning parity with noise (LPN) problem [19, 18], an adversary attempts to learn the value of a bit vector \mathbf{x} , after being given many pairs (\mathbf{b}_i, z_i) , where $z_i = \langle x, \mathbf{b}_i \rangle \oplus e_i$ where with probability p , $e_i = 1$, otherwise $e_i = 0$. The LPN problem is NP-hard [17] and the task of solving the LPN problem over a randomly selected set of pairs \mathbf{b}_i, z_i has been the basis of many cryptographic protocols [19, 18, 15, 6]. The HB problem, first introduced by Blum and Hopper [18], is equivalent to this “random” LPN problem. In the HB problem, the adversary’s goal is to produce a correct bit z^* given a random \mathbf{b}^* after being given many pairs \mathbf{b}_i, z_i for random vectors \mathbf{b}_i .

In this paper, we discuss a variant of the HB problem that we call the “strong HB problem,” in which the adversary’s goal is to produce a correct (\mathbf{b}^*, z^*) pair for an *adversarially chosen* \mathbf{b}^* . This is a straightforward and natural extension of the HB problem, but its hardness, and its use in cryptographic applications, are unexplored.

Hardness. We give several arguments to support the claim that the strong HB problem is hard, although it remains open to reduce the SHB problem to another known assumption. First, we note that the only difference between the SHB problem and the HB problem amounts to whether or not the adversary is allowed to select its own challenge vector or not. The case of the adversarially chosen input has been handled in the context of the polynomial reconstruction problem; and has been shown to be useful for cryptography and is thought to be hard [23].

Next, we give a reduction from an adversary who can attack SHB very successfully (probability near 1) to an adversary who can solve the random LPN problem. Furthermore, we consider a broad class of algorithms, including all methods we are aware of for solving LPN / HB [5], and show that

no such method will be successful against SHB with a polynomial number of queries. Finally we argue that the SHB problem is randomly self-reducible. On the basis of this evidence, we propose that the hardness of the SHB problem is a reasonable assumption.

Applications. We present two interesting applications based on the SHB assumption which demonstrate its utility. The first is a highly efficient, lightweight authentication protocol well-suited to use by RFID tags. The second is a highly efficient related-key secure non-adaptive MAC.

Authentication protocols used by RFID tags are expected to guarantee both security and anonymity. These issues are complicated by the fact that RFID tags have extremely limited computational abilities. Thus, a strong authentication protocol must be capable of being implemented using a small amount of memory and little computation whatsoever, yet still provide sufficient, (and in some ways contradictory), security guarantees.

Blum and Hopper [18] created a very efficient protocol for anonymous lightweight authentication. Their protocol, and two subsequent ones for RFID tags [19, 15], were based on the hardness of the HB problem. These protocols are very efficient, but are not proven secure against a fully adaptive man-in-the-middle adversary. We show how to use the SHB problem to bridge the gap in RFID authentication by presenting the HB^S protocol, which is lightweight, requires two fewer rounds¹ than $HB^\#$, and is proven secure against a fully adaptive man in the middle adversary.

In addition, we construct an XOR related-key secure, non-adaptive MAC in the standard model. This MAC derives directly from the RFID authentication protocol; specifically, the fact that the protocol uses only two rounds is key in converting it into a MAC and the MAC gains its related key security from the simple algebraic structure of the SHB problem. We discuss possible additional efficiency gains to both the RFID and MAC protocols in Appendix A.

2 Prior Work

Cryptography and computational learning theory are two avenues of theoretical computer science which have affected each other. Various intractability results in computational learning theory utilize cryptographic constructions in their proofs [29, 22, 1]. In addition, several learning problems which are thought to be hard have been used to create cryptographic constructions [4, 22, 23, 15, 18, 19]. The two most prominent learning problems used to create cryptographic protocols are the “learning parity with noise” problem and the “polynomial reconstruction problem”. These problems are quite similar in nature. Both concern themselves with learning some target function f , given many samples of the form $f(x_i)$, where each sample is perturbed with some probability. For the LPN problem, the target function is linear, while in the polynomial reconstruction problem the target function is a polynomial of fixed degree. The learning parity with noise problem and the resulting HB and HB+ protocols have been the basis of several authentication protocols, [18, 19, 15] as well as a public key cryptosystem [28]. Similarly, the polynomial reconstruction, (PR), problem has been the basis of a public key encryption scheme as well as a commitment scheme and a blockcipher [23, 2]. Attacks on the various constructions and the underlying learning problems have been proposed. [24, 13, 9, 16, 5, 27].

RFID tags are extremely lightweight, extremely cheap computational devices which are capable of interacting with a remote reader. The lightest weight tags do not even contain their own

¹Regarding the $HB^\#$ protocol as *effectively* requiring four rounds, since RFID tags cannot initiate a protocol on their own.

power sources, instead they are only activated by the signal from the reader. As these tags can be remotely read, and are used for many applications such as product tracking, secure entry, electronic passports and other related protocols, issues of authentication and anonymity exist. Various authentication protocols based off of standard cryptographic primitives have been developed, each providing different efficiency/security guarantees [6, 20, 11, 31, 12, 8, 30, 7, 25]. These protocols, such as the “randomized hash lock” scheme, can be shown to be secure against a fully adaptive man in the middle attack. On the other hand, protocols based off of the HB problem do not require the use of expensive primitives, and as such are very efficient, however they have not been shown to be secure against the same class of adversaries. The original protocol, HB, is easily seen to be insecure against an adversary who can act as a valid reader [18]. HB⁺, deals with this difficulty [19], however HB⁺ has been shown to be insecure against an adversary who can modify messages sent by a valid reader during an instance of the protocol, and who can see if that instance is accepted by the reader. HB[#] is immune to this attack, however it was not proven to be secure against a fully adaptive man in the middle adversary [15].

Previous work on related key security has been mostly concerned with demonstrating insecurity/reduced security in protocols under related-key attacks, though some positive results have been achieved. Many constructions have been shown to have a reduced level of security under related key attacks, such as NMAC [10] and AES [34, 33]. Bellare and Kohno [3] demonstrate that it is impossible to create many different cryptographic primitives that are secure under unlimited related key assumptions. As a positive result, they demonstrate that under a weaker model of a related key adversary, one can construct a related key secure PRP or PRF in the ideal cipher model. Lucks, in [26], gives a construction of blockcipher which is related key secure against an adversary which can only change part of the key, as well some constructions of related key secure primitives under certain number-theoretic assumptions.

3 Definitions

If $f : \mathbb{N} \rightarrow \mathbb{R}$ is a function, we say that f is *negligible* if for all c , there is an n_0 such that for all $n > n_0$, $f(n) < \frac{1}{n^c}$.

We use \mathcal{A} to denote an adversary; adversaries are typically assumed to be probabilistic polynomial-time.

If \mathbf{x}, \mathbf{g} are vectors of length k , let $\langle \mathbf{x}, \mathbf{g} \rangle$ denote the inner product of \mathbf{x} and \mathbf{g} . When the context is clear, let z_i denote the i 'th bit of vector \mathbf{z} . We denote \mathbf{x} randomly selected from $\{0, 1\}^k$ by $\mathbf{x} \leftarrow \{0, 1\}^k$, and \mathbf{x} sampled from a random variable B by $\mathbf{x} \leftarrow B$.

LPN problem. If given enough values, $\mathbf{b}_1, z_1, \mathbf{b}_2, z_2, \dots, \mathbf{b}_n, z_n$ where $z_i = \langle \mathbf{x}, \mathbf{b}_i \rangle$ one can find \mathbf{x} using standard linear algebra. However, if $z_i = \langle \mathbf{x}, \mathbf{b}_i \rangle \oplus e_i$ where e_i is a random bit which may be 1 or 0, finding \mathbf{x} from this transcript is NP-hard. This is called the “learning parity with noise”, (LPN) problem.

Let p be a fixed probability such that $0 < p < 1/2$. Let B_p be a random distribution that outputs 1 with probability p , and 0 with probability $1 - p$. Let $B_{p,n}$ be the distribution that outputs a n -bit vector, each bit being a sample from B_p .

Let $L_{\mathbf{x},p}$ be the distribution which when sampled selects $\mathbf{b} \leftarrow \{0, 1\}^k$ and $e \leftarrow B_p$, and outputs (\mathbf{b}, z) where $z = \langle \mathbf{x}, \mathbf{b} \rangle \oplus e$. Let $L_{\mathbf{x},p,q}$ be the distribution which when sampled outputs

q samples of $L_{\mathbf{x},p}$. We will denote one sample of $L_{\mathbf{x},p,q}$ as a matrix \mathbf{A} and a vector \mathbf{z} : $\exists \mathbf{e} \leftarrow B_{p,q}$ where $\mathbf{A}\mathbf{x} \oplus \mathbf{e} = z$.

Definition 3.1 (Random LPN problem) Define $\text{ADV}_{\text{LPN}}(\mathcal{A}, k)$ to be

$$\Pr[\mathbf{x} \leftarrow \{0, 1\}^k; 1^q \leftarrow \mathcal{A}(1^k); (\mathbf{A}, \mathbf{z}) \leftarrow L_{\mathbf{x},p,q}; \mathbf{x}' \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{z}) : \mathbf{x}' = \mathbf{x}]$$

We say that the (probability p) random LPN problem is hard if the maximum advantage $\text{ADV}_{\text{LPN}}(\mathcal{A}, k)$ over all probabilistic polynomial-time \mathcal{A} is negligible in k .

Since the information provided to the adversary cannot be influenced by the adversary, our definition simplifies the problem into an off-line problem, where the adversary merely specifies how many pairs to receive.

HB problem. The HB problem is a simple variant of the random LPN problem, in which the adversary is not asked to reconstruct \mathbf{x} , but is rather asked to create z^* correctly given a random \mathbf{b}^* .

Definition 3.2 (HB problem) Define $\text{ADV}_{\text{HB}}(\mathcal{A}, k)$ to be

$$\Pr[\mathbf{x} \leftarrow \{0, 1\}^k; 1^q \leftarrow \mathcal{A}(1^k); (\mathbf{A}, \mathbf{z}) \leftarrow L_{\mathbf{x},p,q}; \mathbf{b}^* \leftarrow \{0, 1\}^k; z^* \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{z}, \mathbf{b}^*) : z^* = \langle \mathbf{x}, \mathbf{b}^* \rangle] - \frac{1}{2}$$

We say that the (probability p) HB problem is hard if the maximum advantage $\text{ADV}_{\text{HB}}(\mathcal{A}, k)$ over all probabilistic polynomial-time \mathcal{A} is negligible in k .

It is known that the LPN problem is hard if and only if the HB problem is hard [18].

3.1 Strong HB Problem

In the HB problem, the adversary \mathcal{A} is asked to compute the correct parity for randomly chosen vectors. We can also allow the adversary to be partially adaptive and compute the parity for vectors of the adversary's choosing. With this in mind, we define the *strong* HB problem.

Definition 3.3 (Strong HB problem) Define $\text{ADV}_{\text{SHB}}(\mathcal{A}, \alpha, k)$ to be $\Pr[\text{WIN}_{\text{SHB}}(\mathcal{A}, k)] - (1 - p - \alpha)$, where $\text{WIN}_{\text{SHB}}(\mathcal{A}, k)$ is defined to be

$$\Pr[\mathbf{x} \leftarrow \{0, 1\}^k; 1^q \leftarrow \mathcal{A}(1^k); (\mathbf{A}, \mathbf{z}) \leftarrow L_{\mathbf{x},p,q}; (\mathbf{b}^*, z^*) \leftarrow \mathcal{A}(\mathbf{A}, \mathbf{z}) : z^* = \langle \mathbf{x}, \mathbf{b}^* \rangle \text{ and } \forall i, \mathbf{b}^* \neq \mathbf{b}_i]$$

We say that the (probability p) strong HB problem is hard if there exists a non-negligible α such that the maximum advantage $\text{ADV}_{\text{SHB}}(\mathcal{A}, p', k)$ over all probabilistic polynomial-time adversaries \mathcal{A} is negligible in k .

Our use of α in this definition may seem strange; why not upper bound the adversary's advantage to be negligibly close to random guessing? As we discuss in the next section, the SHB problem can be solved with probability non-negligibly greater than $1/2$ so we cannot use that lower bound. It is sufficient for our purposes however, to guarantee that the adversary's advantage is non-negligibly less than $1 - p$. As such we say the SHB problem is hard as long as the adversary's success is close to $1 - p - \alpha$ where α represents this non-negligible gap. In the next section we conjecture that the SHB problem is hard for $\alpha = p - 2p^2$. Any smaller, but non-negligible α will suffice for our applications.

4 Hardness of the SHB problem

In this section we discuss the SHB problem. We give some arguments supporting our conjecture that the SHB problem is hard, and analyze other aspects of the problem. Informally, our argument that the SHB problem is hard consists of two parts. The first part gives an upper limit on the probability that the adversary can solve the SHB problem, by showing a reduction from the modified problem to LPN. The second part examines known (non-polynomial) attacks on HB, HB⁺, and LPN, and shows how these attacks cannot be generalized to an efficient attacks on SHB. Since we have observed no known attacks on the SHB problem in previous literature, this demonstrates that new techniques will be necessary to solve this problem. We also argue that the SHB problem is randomly self-reducible.

Maximum success probability. We now give an upper limit on the ability for an adversary to solve the SHB problem by giving a reduction from the modified SHB problem to the LPN problem. Informally, this reduction is relatively simple. If, over different instances of the SHB problem, \mathcal{A} returns many solutions \mathbf{b}_i^*, z_i^* for *different* vectors \mathbf{b}_i^* , and if each solution is correct with overwhelming probability, with high probability these vectors form a basis, and thus we can recover \mathbf{x} using standard linear algebra. The difficulty is in that an adversary \mathcal{A} may not return distinct vectors \mathbf{b}_i^* . \mathcal{A} may have a vector \mathbf{b}^* “hardwired” as it were. We demonstrate how to resolve this difficulty.

Theorem 4.1 *Let \mathcal{A} be a probabilistic polynomial-time adversary such that $\Pr[\text{WIN}_{\text{SHB}}(\mathcal{A})] = 1 - \nu(k)$ where ν is negligible. Then there exists \mathcal{A}' such that $\text{ADV}_{\text{LPN}}(\mathcal{A}', k)$ is non-negligible.*

Proof. We define the operation of \mathcal{A}' . When \mathcal{A} requests one sample of $L_{\mathbf{x},p,q}$, \mathcal{A}' receives one sample from $L_{\mathbf{x},p,q}$, (\mathbf{A}, \mathbf{z}) . \mathcal{A}' randomly selects a $k \times k$ matrix \mathbf{R} and a matrix $\mathbf{A}' : \mathbf{A}'\mathbf{R} = \mathbf{A}$. \mathcal{A}' then returns \mathbf{A}' , \mathbf{z} to \mathcal{A} . This is a valid sample from $L_{\mathbf{R}\mathbf{x},p,q}$ because $\mathbf{A}\mathbf{x} \oplus \mathbf{e} = \mathbf{z} = (\mathbf{A}'\mathbf{R})\mathbf{x} \oplus \mathbf{e} = \mathbf{A}'(\mathbf{R}\mathbf{x}) \oplus \mathbf{e}$. \mathcal{A} will then return a pair \mathbf{b}^*, z^* such that $\langle \mathbf{b}^*, \mathbf{R}\mathbf{x} \rangle = z^*$ with all but negligible probability. Since $\langle \mathbf{b}^*, \mathbf{R}\mathbf{x} \rangle = (\mathbf{R}\mathbf{x})^T \mathbf{b}^* = \mathbf{x}^T \mathbf{R}^T \mathbf{b}^* = (\mathbf{b}^*)^T \mathbf{R}\mathbf{x} = \langle \mathbf{x}, \mathbf{R}^T \mathbf{b}^* \rangle$ we now have the inner product of a randomly selected vector, (as \mathbf{R} was randomly selected), and \mathbf{x} . \mathcal{A}' then retrieves q new samples and re-runs \mathcal{A} until it has repeated this procedure $2k$ times. Note that the probability that \mathcal{A} returns even one answer that is incorrect is at most $2k\nu(k)$, which is negligible.

The $2k$ vectors returned by \mathcal{A} , $\mathbf{b}'_i = \mathbf{R}_i \mathbf{b}_i^*$ are k distinct random vectors such that with overwhelming probability $\mathbf{A}'\mathbf{x} = \mathbf{z}$ where \mathbf{A}' is the matrix where each row is an \mathbf{b}'_i vector, and the i 'th bit of \mathbf{z} is z_i . With overwhelming probability, some k of these vectors will be linearly independent, and thus \mathcal{A}' can solve for \mathbf{x} using Gaussian elimination.

Thus, $\text{ADV}_{\text{LPN}}(\mathcal{A}', k)$ is near 1, which is non-negligible. □

Theorem 4.1 assures us that the SHB problem is hard to solve extremely well, but does not give us a useable upper bound. Ideally, we would like to bound the adversary’s success in the SHB problem to be negligibly close to 1/2, the same as the HB problem. The next theorem shows that we cannot upper bound the adversary’s success probability to be that low.

Theorem 4.2 *Let (\mathbf{b}_i, z_i) be a sequence of pairs produced by $L_{\mathbf{x},p,q}$. For all sets of indices i_1 through i_s , $\langle \mathbf{b}_{i_1} \oplus \dots \oplus \mathbf{b}_{i_s}, \mathbf{x} \rangle = z_{i_1} \oplus \dots \oplus z_{i_s}$ with probability $\frac{1}{2} + \frac{1}{2}(1 - 2p)^s$.*

Consider an adversary which takes two samples $\mathbf{b}_i, z_i, \mathbf{b}_j, z_j$ and computes $\mathbf{b}_i \oplus \mathbf{b}_j, z_i \oplus z_j$ as its answer. Based on Theorem 4.2 this will be right with probability $1 - 2p + 2p^2$ which is non-negligibly greater than $1/2$.

Known attacks. We now examine previous attacks on the LPN, HB and HB⁺ problems and see how these attacks might be applied to SHB. Attacks which give lower bounds on the difficulty of the LPN and HB problems [13, 9, 16, 5, 27, 14], (though they do not break it), work by attempting to exploit the linear structure of the LPN and HB problems. They attempt to find many equations of the vectors \mathbf{b}_i , each equation containing a small number of vectors, such that $\mathbf{b}_1 \oplus \mathbf{b}_2 \oplus \dots \oplus \mathbf{b}_s = \mathbf{c}$ where \mathbf{c} is equal to a desired vector, usually a canonical basis vector. With many equations being equal to the same \mathbf{c}_i , this can give us $\langle \mathbf{c}_i, \mathbf{x} \rangle$ with high probability due to Theorem 4.2. The algorithm then uses these vectors \mathbf{c}_i to find \mathbf{x} . For instance, the attack of [5] on the LPN problem attempts to create the canonical basis vectors, (vectors where all but 1 element is 0), using this process. To solve the SHB problem, we do not need to find a set of specific solutions $\mathbf{c}_i, \langle \mathbf{c}_i, \mathbf{x} \rangle$ we merely require any one vector, \mathbf{c} . We now show that this methodology is incapable of providing a poly-time attack on SHB.

Theorem 4.3 *Let \mathcal{A} receive q samples of the form $\mathbf{b}_i, \mathbf{z}_i = \langle \mathbf{b}_i, \mathbf{x} \rangle \oplus \mathbf{e}$ where $\mathbf{a}_i \leftarrow \{0, 1\}^k$ as part of his attempt to break the SHB problem. The probability that two equations $\mathbf{a}_{i_1} \oplus \mathbf{a}_{i_2} \oplus \dots, \mathbf{a}_{i_s} = \mathbf{c}$ and $\mathbf{a}_{i'_1} \oplus \dots \oplus \mathbf{a}_{i'_s} = \mathbf{c}$ exist, each of size s or less where $s = O(\text{polylog}(k))$ is negligible.*

Proof. There are $\sum_{i=1}^s \binom{q}{i}$ different equations of size up to s given q samples. As an upper bound to the probability, we assume that each equation produces a different value \mathbf{c}_j . From [32] we can conclude that:

$$\sum_{i=1}^s \binom{q}{i} \leq s \binom{q}{s} \leq sq^s \leq s2^{ls}$$

where $l = O(\text{polylog}(k))$ as if l is not poly-logarithmic then q is exponential in k which makes \mathcal{A} exponential in the security parameter. The probability that two equations output the same value \mathbf{c} is 2^{-k} as the vectors \mathbf{a}_i are randomly selected. Thus the probability that two equations exist that both equal a vector \mathbf{c} is less $2^{-k+2\text{polylog}(k)}$ which is negligible. \square

What this shows is that if the adversary's plan of attack against SHB is to gather "votes" for the value of $\langle \mathbf{x}, \mathbf{c} \rangle$, by finding equations of values that xor to \mathbf{c} , and if the adversary has only polynomially many samples, then either \mathbf{c} will have a short equation but very likely only *one*, or \mathbf{c} will be the result of multiple equations, but all such equations will have more than polylogarithmic \mathbf{b}_i values involved. In the former case, Theorem 4.2 shows that the adversary, with a single equation, has probability $\frac{1}{2} + \frac{1}{2}(1 - 2p)^s \leq \frac{1}{2} + \frac{1}{2}(1 - 2p)^2$ of success. In the latter, each equation gives a negligible advantage over $\frac{1}{2}$, so the adversary would need to examine exponentially many such colliding equations, which would be impossible for a polynomial-time adversary.

Thus, this argument shows that if an adversary can solve the Strong HB-Problem that adversary will need to utilize a genuinely new technique as they cannot depend on amplifying the advantage gained through any linear equations, the technique used by all other known attacks on related problems.

This leads us to propose the following conjecture concerning the SHB problem:

Conjecture 4.4 (Hardness of the SHB Problem) *The SHB problem is hard for $\alpha = (1 - p) - \frac{1}{2} + \frac{1}{2}(1 - 2p)^2 = p - 2p^2$*

Uniform hardness. In order to support the claim that the SHB problem is useful for cryptography, we need to justify that it is hard on average. Our arguments so far do not establish this, but we now argue that the SHB problem is randomly self-reducible. We call one sample from $L_{\mathbf{x},p,q}$, as an instance of the SHB or LPN problems.

Lemma 4.5 (Random self-reducibility) *Any instance \mathbf{A}, \mathbf{z} of the strong HB problem can be transformed into an instance of the strong HB problem with random secret \mathbf{x} and biased error vector \mathbf{e} , such that a correct solution of the resulting problem can be translated into a correct solution of the original problem.*

Proof. The proof is a similar technique as is used in the previous reduction. Given a SHB problem instance \mathbf{A}, \mathbf{z} , select a random invertible matrix \mathbf{R} and a matrix $\mathbf{A}' : \mathbf{A}'\mathbf{R} = \mathbf{A}$. The rows of \mathbf{A}' and the corresponding entries of \mathbf{z} are then randomly shuffled. Since \mathbf{R} is a random invertible matrix and \mathbf{x} is a random vector, $\mathbf{R}\mathbf{x}$ is a new randomly selected vector. As such, $\mathbf{A}'(\mathbf{R}\mathbf{x}) \oplus \mathbf{e} = \mathbf{z}$ is a valid instance with a randomly selected vector $\mathbf{R}\mathbf{x}$. The shuffling of the rows of \mathbf{A} and the entries in \mathbf{z} can be viewed as randomizing the error vector \mathbf{e} .

When an adversary \mathcal{A} returns a result $\mathbf{b}^*, z^* : z^* = \langle \mathbf{R}\mathbf{x}, \mathbf{b}^* \rangle$ we can change the result to $\mathbf{R}^T \mathbf{b}^*, z^*$, a correct solution to the original problem instance. \square

Theorem 4.6 (Uniform hardness) *Suppose \mathcal{A} is a probabilistic polynomial-time adversary such that $\Pr[\text{WIN}_{\text{SHB}}(\mathcal{A})|\mathbf{A}, \mathbf{z}] > p_0$ for some p_0 and a non-negligible fraction of possible \mathbf{A}, \mathbf{z} pairs (over all pairs and all \mathbf{x} .) Then there exists a \mathcal{A}' such that for every \mathbf{A}, \mathbf{z} , $\Pr[\text{WIN}_{\text{SHB}}(\mathcal{A}')|\mathbf{A}, \mathbf{z}] > p_0$.*

Proof. Let \mathcal{A}' be an adversary which receives an instance of the SHB problem, \mathbf{A}, \mathbf{z} where $\mathbf{z} = \mathbf{A}\mathbf{x} \oplus \mathbf{e}$ for some \mathbf{x} and \mathbf{e} . For each row of \mathbf{A} , \mathcal{A}' takes n other rows at random and sums them together, producing a new row of the matrix \mathbf{A}' . The corresponding entry z_i is computed by adding together the corresponding n bits of \mathbf{z} together. The noise rate is now set to be $p' = \frac{1}{2} - \frac{1}{2}(1 - 2p)^{n+1}$. The resulting instance is a new, random instance of the SHB problem and thus with non-negligible probability can be solved by \mathcal{A} . Since the “random” instance of the problem utilizes the same secret \mathbf{x} vector as the “real” instance, the solution provided by \mathcal{A} is a solution for the instance given to \mathcal{A}' . \square

5 Fully Secure RFID Protocol

In this section we give a construction of a fully secure RFID protocol. Our protocol is very efficient, requiring little more than the inner product of numerous bit vectors. Our definition of a fully secure protocol is very similar to [20] and is stronger than the security model of the $HB^\#$ protocol[15].

Before we give a definition of an RFID protocol we state our model describing how an RFID protocol governs the interactions between readers and tags. This model consists of tags and a single reader, all of which are PPT machines. We have that each tag \mathcal{T}_i maintains a key K_i as well as a state S_i^0 . The reader \mathcal{R} maintains a list of tag/key pairs \mathcal{T}_i / K_i which is unavailable to the adversary. Each reader \mathcal{R} begins a protocol session, (using a function `BeginReader`), with a tag \mathcal{T}_i by sending its first message a_1, σ , where σ is the session ID. Given a message a_i and a session identifier σ , \mathcal{T}_i utilizes the function `TagResponse` to return a message b_i, σ . The reader responds to b_i, σ with a_{i+1}, σ utilizing the function `ReaderResponse`. It should be noted that we require that the

first message of an RFID protocol come from the reader. This is due to the fact that RFID chips often do not contain their own power sources, and thus cannot send a message without receiving one from the reader. We also allow both `TagResponse` and `ReaderResponse` to depend on the protocol ID σ , and all previous messages sent under that protocol.

We call any sequence of messages $\sigma, a_1, b_1, \dots, a_n, b_n$, where each a_i/b_i was sent during σ , a transcript and we denote a specific transcript via $S_{n,\sigma}$. The reader has a function `Output`, (which has access to the list of tag/key pairs as well as previous sent/received messages), and which on input $S_{\sigma,n}$ outputs 0 or a set of “valid” tag numbers. We say a transcript $S_{\sigma,n}$ is accepting if the reader began a session σ and $\text{Output}(S_{\sigma,n}) \neq 0$. We assume that the reader has a fixed number n such that after \mathcal{R} receives n messages from a tag, it runs `Output` on the resulting transcript and either accepts or rejects. With this model in mind, we define an RFID protocol P as the tuple of functions `BeginReader`, `TagResponse`, `ReaderResponse` and `Output`.

Definition 5.1 (Accurate) *We say an RFID protocol is accurate if with overwhelming probability over the possible transcripts $S_{\sigma,n}$ `Output` returns either 0, or a single number k .*

In order to define the security properties of an RFID protocol, we now define the interactions an adversary can have between the reader and a given set of tags. We allow any adversary \mathcal{A} to perform the following interactions between the adversary, the reader, and the tags:

1. `BeginReader`: \mathcal{A} requests the reader to begin a protocol session σ . The adversary records the reader’s first message σ, a_1 .
2. `BeginTag`(i, σ, a_1): \mathcal{A} asks the tag \mathcal{T}_i to begin a new session σ and sends a_1 as the first message.
3. `Corrupt`($j, \mathcal{K}'_j, \mathcal{S}'_j$): The adversary receives key and state $\mathcal{K}_j, \mathcal{S}_j$ from \mathcal{T}_j and \mathcal{T}_j sets its key as \mathcal{K}'_j and state as \mathcal{S}'_j (Note: $\mathcal{K}'_j, \mathcal{S}'_j$ can depend on $\mathcal{K}_j, \mathcal{S}_j$).
4. `ReaderMessage`: \mathcal{A} sends message σ, b_i to the reader, and receives σ, a_{i+1} .
5. `TagMessage`(j, \mathbf{a}_i, σ): \mathcal{A} sends message σ, a_i to tag \mathcal{T}_j and receives response σ, b_i .

It should be noted that \mathcal{A} does not get access to the results of `Output` for any transcript. This is a reasonable restriction, as for many RFID protocols, the results of a valid acceptance/rejection of a tag cannot be observed by the adversary.

We now define the following two games.

Unforgeability Game:

In the first phase, \mathcal{A} interacts between the tags and the reader by invoking the above interactions up to q times. Denote the set of tags that were not corrupted by \mathcal{A} as $\mathcal{T}_{incorrupted}$.

In the second stage, \mathcal{A} is barred from sending any `TagMessage`, `BeginTag` and `Corrupt` messages and sends a new `BeginReader` message to the reader. Denote all messages sent from \mathcal{A} to the reader in this stage as b_i^* and all messages from the reader to \mathcal{A} in this stage as a_i^* .

The reader outputs $\text{Output}(S_{\sigma,n}^*)$ where $S_{\sigma,n}^* = a_1^*, b_1^*, \dots, a_n^*, b_n^*$. The adversary succeeds if the output is $j \neq 0$ and $\mathcal{T}_j \in \mathcal{T}_{incorrupted}$.

Anonymity Game:

In the first phase, \mathcal{A} interacts with the reader and tags q' times. Denote the set of tags left incorrupted by \mathcal{A} as $\mathcal{T}_{incorrupted}$.

\mathcal{A} selects two tags $\mathcal{T}_1, \mathcal{T}_0 \in \mathcal{T}_{incorrupted}$. A random bit b is flipped and we denote \mathcal{T}_b as \mathcal{T}^* .

\mathcal{A} again interacts with the reader and tags $q'' : q = q' + q''$ times utilizing the previous protocols with the exceptions that it can no longer interact with \mathcal{T}_0 or \mathcal{T}_1 , and cannot $\text{Corrupt}(\mathcal{T}^*)$.

\mathcal{A} outputs a bit and succeeds if the bit $b' = b$.

We define $\text{ADV}_{\text{ANON}}(P, \mathcal{A}, q, t)$ as the probability an adversary which interacts q times and takes t time succeeds in the anonymity game, and we define $\text{ADV}_{\text{UNFORG}}(P, \mathcal{A}, q, t)$ as the probability that an adversary which takes t time and interacts q times succeeds in the unforgeability game for a given protocol P .

Definition 5.2 (Un-Forgeability) *An adversary \mathcal{A} is considered to have broken the unforgeability property of an RFID protocol P if $\text{ADV}_{\text{UNFORG}}(P, \mathcal{A}, q, t)$ is non-negligible for some polynomial q, t .*

Definition 5.3 (Anonymity) *An adversary \mathcal{A} is considered to have broken the anonymity of an RFID protocol P if $\text{ADV}_{\text{ANON}}(P, \mathcal{A}, q, t)$ is non-negligible for some q, t .*

Definition 5.4 (Fully Secure) *An RFID protocol is fully secure if it is both anonymous and un-forgeable.*

5.1 HB^S Protocol

We now give a construction of a fully secure, accurate, RFID authentication protocol. Let the reader \mathcal{R} maintain a list of key tag pairs, $\mathcal{T}_i, \mathcal{K}_i$ where $\mathcal{K}_i = X, Y$ where X and Y are random $s(k)$ by k matrices for some polynomial $s(k)$. The protocol goes as follows:

- The reader \mathcal{R} sends a k bit vector \mathbf{a} to \mathcal{T} .
- \mathcal{T} returns $\mathbf{b}, (\mathbf{c} = \mathbf{X}\mathbf{a} \oplus \mathbf{Y}\mathbf{b} \oplus \mathbf{e})$ to \mathcal{R} where \mathbf{e} is an $s(k)$ bit vector that has each bit set to 1 with probability p .
- \mathcal{R} goes through its list of keys and if there is one key $\mathbf{X}_i, \mathbf{Y}_i : \|\mathbf{X}_i\mathbf{a} \oplus \mathbf{Y}_i\mathbf{b} \oplus \mathbf{c}\| \leq u$, where $\|\mathbf{x}\|$ denotes the Hamming weight of \mathbf{x} and $u \approx ps(k)$ \mathcal{R} accepts the tag as \mathcal{T}_i . Otherwise \mathcal{R} rejects the tag as a forgery attempt.

We prove the security of this construction in relation to the strong HB problem.

Theorem 5.5 (Un-Forgeable) *Let \mathcal{A} be an adversary which breaks the Un-Forgeability property. Then there exists \mathcal{A}' which solves the strong HB problem.*

Proof. \mathcal{A}' has access to the distribution $L_{\mathbf{x}, p, q}$ and as such can be considered to have access to the distribution $L_{\mathbf{x}, p}$. \mathcal{A}' will create his “test” RFID key by selecting a pair of random matrices \mathbf{X}^* and \mathbf{Y}^* where all but one row of \mathbf{Y}^* is defined. Denote the undefined row by j . \mathcal{A}' also creates many other valid RFID keys $\mathbf{X}_i, \mathbf{Y}_i$. \mathcal{A}' will now act as a reader or as a tag, in response to the various messages sent by \mathcal{A} . When \mathcal{A}' receives $\text{BeginTag}(j, \mathbf{a}_{1, \sigma}, \sigma)$ for a never before seen j from \mathcal{A} , \mathcal{A}' either chooses to set the key for \mathcal{T}_j as $\mathbf{X}_j, \mathbf{Y}_j$ or to set the key as $\mathbf{X}^*, \mathbf{Y}^*$ where \mathcal{A}' makes the latter decision only once. Once \mathcal{A}' sets the key for a tag \mathcal{T}_* as $\mathbf{X}^*, \mathbf{Y}^*$, \mathcal{A}' answers

$\text{TagMessage}(*, \mathbf{a}_{1,\sigma}, \sigma)$ queries by selecting a vector \mathbf{b}_i from some tuple in Trans and returning \mathbf{z}_i where $\forall i \neq j \ z_i = \langle \mathbf{a}_{1,\sigma}, \mathbf{x}_i \rangle \oplus \langle \mathbf{b}_i, \mathbf{y}_i \rangle \oplus e_i$ where at most $s(k) - u + 1$ bits of e_i are 1 and where $z_j = \langle \mathbf{a}_{1,\sigma}, \mathbf{x}_j \rangle \oplus g_j$. It is clear that for all messages where \mathcal{A}' responds using key $\mathbf{X}_j, \mathbf{Y}_j$, \mathcal{A}' responds as a valid tag, and when \mathcal{A}' uses key $\mathbf{X}^*, \mathbf{Y}^*$, \mathcal{A}' 's response is the response of a valid tag using key $\mathbf{X}^*, \mathbf{Y}^*$ where the j 'th row of \mathbf{Y}^* is \mathbf{x} . \mathcal{A}' responds to BeginReader by selecting a random vector \mathbf{a}_1 and random id σ . There is no ReaderMessage query to respond to as the reader sends one message in the protocol. The correctness of \mathcal{A}' as a reader comes from the fact that its BeginReader responses are identically distributed to the reader, and the fact that \mathcal{A}' always gives correct responses as a tag. Thus \mathcal{A}' can simulate a valid reader and tag.

When \mathcal{A} attempts to forge it returns \mathbf{b}', \mathbf{c} in response to \mathcal{A}' 's message \mathbf{a}' . \mathcal{A}' computes $\mathbf{c} \oplus \mathbf{X}^* \mathbf{a}' \oplus \mathbf{Y}^* \mathbf{b}'$ for all bits except the j 'th, and checks to see if less than u bits are incorrect. If that is the case, then \mathcal{A} has created a successful forgery for \mathcal{T}_* . If not, then \mathcal{A}' discards all tuples $\mathbf{b}_i, \langle \mathbf{b}_i, \mathbf{x} \rangle \oplus e_i$ that it received from $L_{\mathbf{x},p,q}$ and begins again. Since \mathcal{A} can only interact with a polynomial number of tags, (and cannot forge a tag that he has never interacted with), and \mathcal{A}' randomly selects a tag to use its “test” RFID key, the probability of \mathcal{A} successfully forging the tag needed by \mathcal{A}' is non-negligible.

If \mathcal{A}' receives a successful forgery \mathbf{c} for \mathcal{T}_* , \mathcal{A}' can compute $\mathbf{c} \oplus \mathbf{X}^* \mathbf{a}' = \mathbf{Y}^* \mathbf{b}' \oplus \mathbf{e}'$ for some error vector \mathbf{e}' where the j 'th bit of this vector is $\langle \mathbf{x}, \mathbf{b}' \rangle$ with probability p' . We argue that $p' \geq 1 - p$. This follows from the fact that j is randomly selected without regards to any incorrect bits returned by \mathcal{A} in its forgery, and there can be at most u incorrect bits. Thus the probability that the j 'th bit of $\mathbf{c} \oplus \mathbf{X}^* \mathbf{a}'$ is equal to $\langle \mathbf{b}', \mathbf{x} \rangle$ is equal to the probability that \mathcal{A} “chose” that bit to be correct, which is $\approx \frac{s(k)-u}{s(k)} = 1 - p$.

Thus, when \mathcal{A} returns a valid forgery on the “correct” tag, with probability $\approx 1 - p \geq 1 - p - \epsilon$, \mathcal{A}' receives a correct inner product and thus solves the SHB Problem. \square

Theorem 5.6 *For all transcripts $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and for all p , the probability that for two random keys $\mathbf{X}_i, \mathbf{Y}_i, \mathbf{X}_j, \mathbf{Y}_j$ there exist two correctly weighted error vectors $\mathbf{e}_i, \mathbf{e}_j$: $\mathbf{X}_i \mathbf{a} \oplus \mathbf{Y}_i \mathbf{b} \oplus \mathbf{e}_i = \mathbf{X}_j \mathbf{a} \oplus \mathbf{Y}_j \mathbf{b} \oplus \mathbf{e}_j = \mathbf{c}$ is negligible. Thus this protocol is accurate.*

Proof. This comes directly from the un-forgeability property and the fact that only a polynomial number of keys are stored in the reader’s database. If the theorem is false, then an adversary could forge any tag that exists in the reader’s database by selecting a random key and answering queries from the reader using that key. With non-negligible probability over random keys, there is a correctly weighted error vector \mathbf{e}_i such that the transcript is a valid transcript from the tag in the database. \square

Before we show that this protocol is anonymous, we cite a theorem concerning the SHB distribution from [21].

Theorem 5.7 (Pseudo-Random) *For a randomly selected secret \mathbf{x} , the distribution $(\mathbf{b}, z) : \mathbf{b} \leftarrow \{0, 1\}^k, e \leftarrow B_p, z = \langle \mathbf{b}, \mathbf{x} \rangle \oplus e$ is pseudorandom.*

Theorem 5.8 *This protocol satisfies the anonymity property*

Proof. This proof comes from Theorem 5.7, the randomness and independence of each row of \mathbf{X} and \mathbf{Y} , and the fact that for each value $\mathbf{X}_i \mathbf{a}_i \oplus \mathbf{Y}_i \mathbf{b}_i \oplus \mathbf{e}_i$, either the \mathbf{a}_i or \mathbf{b}_i vector is selected

randomly. Thus the \mathbf{c}_i vectors which are returned by either \mathcal{A} or \mathcal{A}' are pseudorandom as each bit is pseudorandom and independent. \square

These theorems show that our protocol is fully secure, and accurate with a failure rate related to the probability of more than u bits being set to 1 when each bit is selected with probability p . In addition, this shows that if the SHB problem is indeed hard, the $HB^\#$ protocol of [15] can be run in two rounds, and is secure in the full model which solves two open questions of that paper.

6 HB^S -MAC

We now construct HB^S -MAC, a very efficient, related-key secure, non-adaptive nonce based MAC whose security is based off of the SHB problem.

This construction is based off the HB^S construction described earlier. The main observation is that in the earlier RFID protocol, the fact that the vector \mathbf{a}' sent by \mathcal{A}' was random, was never used in the proof. Thus for any vector \mathbf{a}' it should be hard for the adversary to forge in the RFID protocol. Thus we can use \mathbf{a} as the message. This MAC is very efficient and similar in construction to MAC's which utilize a nonce, though in our case we require the nonce to be random.

Definition 6.1 (Φ Related-Key Secure Non-Adaptive MAC) *A Φ related key secure MAC is a trio of functions KeyGen , MAC , Verify that possess the following properties (Denote M_K as the oracle which, on input m returns $\text{MAC}(m, K)$ and let ν be negligible):*

$\text{KeyGen}(1^k)$ returns key K .

$\text{MAC}(m, K)$ returns a tag Γ .

$\text{Verify}(m, \Gamma, K)$ returns a bit, such that $\text{Verify}(m, \text{MAC}(m, K), K) = 1$.

Security: $\forall \mathcal{A} \in PPT, \exists \nu$ negligible: $\Pr[K \leftarrow \text{KeyGen}(1^k); \mathcal{A}^{M_K} \rightarrow m, \Gamma, f : \text{Verify}(m, \Gamma, f(K)) = 1] \leq \nu(k)$ where \mathcal{A} never queries M on m and where M_K returns $\Gamma = \text{MAC}(m, f(K))$ on input (m, f) for function $f \in \Phi$.

We call such a MAC “non-adaptive” in the sense that it does not necessarily provide security against an adversary capable of arbitrary verification queries. Note that for a *deterministic* MAC, in which re-computing the MAC constitutes verification, non-adaptive security implies “adaptive security”, or the normal notion of security with regards to MAC protocols. However, this is not necessarily the case for MACs that are not deterministic, such as ours.

We now restate our previous construction as a MAC.

MAC Construction

$\text{KeyGen}(1^k) = \mathbf{X}, \mathbf{Y}$ where \mathbf{X} and \mathbf{Y} are random $s(k)$ by k matrices.

$\text{MAC}(m, \mathbf{X}, \mathbf{Y})$ treats m as a vector \mathbf{m} , selects a random vector \mathbf{b} , and a randomized vector \mathbf{e} where each bit of \mathbf{e} is 1 with probability p and returns $\mathbf{b}, \mathbf{X}\mathbf{m} \oplus \mathbf{Y}\mathbf{b} \oplus \mathbf{e}$.

$\text{Verify}(m, \Gamma, \mathbf{X}, \mathbf{Y})$ checks if $\mathbf{c} = \mathbf{X}\mathbf{m} \oplus \mathbf{Y}\mathbf{b} \oplus \Gamma$ has hamming weight $\leq u$. If so, Verify outputs 1, otherwise 0.

Theorem 6.2 *The above construction is a Φ related-key secure MAC where $\Phi = \{f : f(x) = x \oplus f\}$.*

Proof. We first prove the construction is a secure MAC. The proof will follow similarly to the proof of Theorem 5.5. Given \mathcal{A} which violates the security of the MAC we construct \mathcal{A}' to solve the SHB problem. \mathcal{A}' given a sample from $L_{\mathbf{x},p,q}$ will act as M_K . \mathcal{A}' creates a random key \mathbf{X}, \mathbf{Y} , where a random row j of \mathbf{Y} is undefined. When \mathcal{A}' receives a message \mathbf{m} , \mathcal{A}' creates $\Gamma = \mathbf{X}\mathbf{m} \oplus \mathbf{g} \oplus \mathbf{e}_i$ where $g_j = z_k$ is retrieved from $L_{\mathbf{x},p}, \forall i \neq j, g_i = \langle \mathbf{b}_k, \mathbf{y}_i \rangle$ where \mathbf{y}_i is the i 'th row of \mathbf{Y} and where $e_j = 0, \forall i \neq j e_i$ is one with probability p . It is easy to see that \mathcal{A}' is simulating M using key \mathbf{X}, \mathbf{Y} where the j 'th row of \mathbf{Y} is \mathbf{x} .

When \mathcal{A} outputs $\mathbf{m}, \mathbf{b}', \Gamma$ \mathcal{A}' checks if $\Gamma \oplus \mathbf{X}\mathbf{m} \oplus \mathbf{Y}\mathbf{b}'$, (leaving the j 'th bit of $\mathbf{Y}\mathbf{b}'$ undefined), has Hamming weight $\leq u$. If not, \mathcal{A}' discards the samples from $L_{\mathbf{x},p}$ it used in its simulation then restarts \mathcal{A} . If so, then Γ is a valid forgery of message \mathbf{m} . Similarly to Theorem 5.5 we argue that if Γ is valid forgery, the j 'th bit of Γ is correct with probability p . This comes from the fact that j is randomly chosen, independently of any actions by \mathcal{A} and at most $u \approx ps(k)$ positions of Γ are "incorrect". Thus \mathcal{A}' solves the SHB problem.

We next prove that the construction is xor related-key secure by giving a reduction from an adversary which breaks the xor related-key security, to one who breaks its security as a normal MAC. Let \mathcal{A} be the adversary which breaks the xor related-key security. For any string f , we break down f into $\mathbf{F}_x, \mathbf{F}_y$ to represent the offset matrices \mathcal{A} wants for each section of the key. When \mathcal{A} makes a query m, f , \mathcal{A}' , the adversary which will break the normal MAC security of the construction, queries its oracle on m , receiving m, b, Γ . \mathcal{A}' then computes $\Gamma \oplus \mathbf{F}_x\mathbf{m} \oplus \mathbf{F}_y\mathbf{b} = \mathbf{X}\mathbf{m} \oplus \mathbf{F}_x\mathbf{m} \oplus \mathbf{Y}\mathbf{b} \oplus \mathbf{F}_y\mathbf{b} \oplus \mathbf{e} = (\mathbf{X} \oplus \mathbf{F}_x)\mathbf{m} \oplus (\mathbf{Y} \oplus \mathbf{F}_y)\mathbf{b} \oplus \mathbf{e}$ and returns this value to \mathcal{A} . It is clear that this is a valid MAC of message m under nonce \mathbf{b} and key $\mathbf{X} \oplus \mathbf{F}_x, \mathbf{Y} \oplus \mathbf{F}_y$. When \mathcal{A} returns $\mathbf{m}, \mathbf{b}, \Gamma, f$, \mathcal{A}' computes $\Gamma \oplus \mathbf{F}_x\mathbf{m} \oplus \mathbf{F}_y\mathbf{b}$ which is a valid MAC of message m using nonce b and key $\mathbf{X} \oplus \mathbf{F}_x \oplus \mathbf{F}_x = \mathbf{X}, \mathbf{Y} \oplus \mathbf{F}_y \oplus \mathbf{F}_y = \mathbf{Y}$, as long as \mathcal{A} creates a successful forgery. □

It is interesting to note how we gain related key security in our non-adaptive MAC construction. Our MAC is related key secure in that it has a certain type of related key *insecurity*. Namely, we find that for a given message, nonce and tag tuple under one key, it is easy to find valid tags for that message and nonce under and chosen offset of the key.

7 Conclusion

In this paper we defined the *strong* HB problem (SHB), a variant of HB for adaptive adversaries. We gave arguments to support the claim that SHB is hard, although it remains open to reduce SHB to another known assumption.

We further gave two highly efficient applications of the SHB assumption, both of which may be of independent interest (see Appendix A for discussion of efficiency). The HB^S protocol is the first protocol in HB family of protocols to achieve security against a fully adaptive man-in-the-middle attack. In addition we create $\text{HB}^S\text{-MAC}$, an xor related-key secure non-adaptive MAC proven secure in the standard model.

References

- [1] Dana Angluin and Michael Kharitonov. Why won't membership queries help? In *23rd ACM Symposium on Theory of Computing*, pages 444–454, 1991.
- [2] Daniel Augot and Matthieu Finiasz. A public key encryption scheme based on the polynomial reconstruction problem. In *Advances in Cryptology – Proceedings of EUROCRYPT 2003*, page 645, 2003.
- [3] M. Bellare and T. Kohno. A Theoretical Treatment of Related-Key Attacks: PKA-PRPs, RKA-PRFs, and Applications. In E. Biham, editor, *Advances in Cryptology – EUROCRYPT '03*, volume 2656 of LNCS, pages 491–506, 2003.
- [4] A. Blum, M. Furst, M. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Advances in Cryptology – Proceedings of CRYPTO '93*, pages 278–291, 1993.
- [5] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *JACM*, 50(4):506–519, 2003.
- [6] 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, 2nd International Workshop*, pages 28–33, 2006.
- [7] Mike Burmester, Breno de Medeiros, and Rossana Motta. Robust, Anonymous RFID Authentication with Constant Key-Lookup. In *Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS 2008, Tokyo, Japan*, pages 283–291, 2008.
- [8] Mike Burmester, Tri van Le, and Breno de Medeiros. Provably Secure Ubiquitous Systems: Universally Composable RFID Authentication Protocols. Cryptology ePrint Archive, Report 2006/131, 2006. <http://eprint.iacr.org/>.
- [9] Jose Carrijo, Rafael Tonicelli, Hideki Imai, and Anderson C A Nascimento. A Novel Probabilistic Passive Attack on the Protocols HB and HB^+ . Cryptology ePrint Archive, Report 2008/231, 2008. <http://eprint.iacr.org/>.
- [10] Scott Contini and Yiqun Lisa Yin. Forgery and partial key-recovery attacks on hmac and nmac using hash collisions. In *Advances in Cryptology - ASIACRYPT 2006*, pages 37–53, 2006.
- [11] Ivan Damgård and Michael Østergaard Pedersen. RFID Security: Tradeoffs between Security and Efficiency. In *Topics in Cryptology – CT-RSA 2008, The Cryptographers' Track at the RSA Conference 2008 Proceedings*, pages 318–332, 2008.
- [12] Martin Feldhofer, Sandra Dominikus, and Johannes Wolkerstorfer. Strong Authentication for RFID Systems using the AES algorithm. In *Cryptographic Hardware and Embedded Systems*, pages 357–370, 2004.
- [13] Marc P. C. Fossorier, Miodrag J. Mihaljevi, Hideki Imai, Yang Cui, and Kanta Matsuura. A Novel Algorithm for Solving the LPN problem and its Application to Security Evaluation of

- the HB protocol for RFID authentication. In *Progress in Cryptology – INDOCRYPT 2006, 7th International Conference on Cryptology in India, Proceedings*, pages 48–62, 2006.
- [14] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. An active attack against HB⁺ - a provably secure lightweight authentication protocol. In *IEE Electronic Letters 41: 21*, pages 1169–1170, 2005.
- [15] Henri Gilbert, Matthew J.B. Robshaw, and Yannick Seurin. HB#: Increasing the security and efficiency of HB⁺, 2008.
- [16] Zbigniew Golebiewski, Krzysztof Majcher, Filip Zagorski, and Marcin Zawada. Practical Attacks on HB and HB⁺ Protocols. Cryptology ePrint Archive, Report 2008/241, 2008. <http://eprint.iacr.org/>.
- [17] Johan Hastad. Some optimal inapproximability results. In *Proceedings of STOC 1997*, pages 1–10, 1997.
- [18] N. Hopper and M. Blum. Secure human identification protocols. In *Advances in Cryptology - ASIACRYPT 2001*, pages 52–56, 2001.
- [19] Ari Juels and Stephen Weis. Authenticating pervasive devices with human protocols. In *Advances in Cryptology – CRYPTO 2005*, pages 293–308, 2005.
- [20] Ari Juels and Stephen Weis. Defining strong privacy for RFID. In *Pervasive Computing and Communications Workshops*, pages 342–347, 2007.
- [21] Jonathan Katz. Efficient cryptographic protocols based on the hardness of learning parity with noise. In *IMA Int. Conf.*, pages 1–15, 2007.
- [22] Michael Kearns and Leslie G. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. In *21st ACM Symposium on Theory of Computing*, pages 433–444, 1989.
- [23] Aggelos Kiayias and Moti Yung. Cryptographic hardness based on the decoding of Reed-Solomon codes with applications. In *Proceedings of ICALP 2002, LNCS 2380*, pages 232–243, 2002.
- [24] Aggelos Kiayias and Moti Yung. Cryptanalyzing the polynomial-reconstruction based public-key system under optimal parameter choice. *Des. Codes Cryptography*, 43(2-3):61–78, 2007.
- [25] Jooyoung Lee and Yongjin Yeom. Efficient RFID authentication protocols based on pseudorandom sequence generators. Cryptology ePrint Archive, Report 2008/343, 2008. <http://eprint.iacr.org/>.
- [26] Stefan Lucks. Ciphers secure against related-key attacks. In *Fast Software Encryption 2004*, 2004.
- [27] Vadim Lyubashevsky. The Parity Problem in the Presence of Noise, Decoding Random Linear Codes, and the Subset Sum Problem. In *APPROX-RANDOM 2005: Approximation, Randomization and Combinatorial Optimization*, pages 378–389, 2005.

- [28] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1997.
- [29] Shafi Goldwasser Oded Goldreich and Silvio Micali. How to construct random functions. In *Journal of the ACM*, volume 33, pages 792–807, 1986.
- [30] Tri van Le, Mike Burmester, and Breno de Medeiros. Forward-Secure RFID Authentication and Key exchange. Cryptology ePrint Archive, Report 2007/051, 2007. <http://eprint.iacr.org/>.
- [31] Stephen Weis, Sanjay Sarma, Ronald Rivest, and Daniel Engels. Security and privacy aspects of low-cost radio frequency identification systems. In *Conference on Security in Pervasive Computing SPC 2003*, volume LNCS, pages 454–469, 2003.
- [32] Thomas Worsch. Lower and Upper Bounds for (sums of) Binomial Coefficients. 1994.
- [33] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Improved related-key impossible differential attacks on reduced-round aes-192. In *Selected Areas in Cryptography*, pages 15–27, 2007.
- [34] Wentao Zhang, Lei Zhang, Wenling Wu, and Dengguo Feng. Related-key differential-linear attacks on reduced aes-192. In *Progress in Cryptology - INDOCRYPT 2007*, pages 73–85, 2007.

A Toeplitz Matrices and Efficiency

In the above RFID protocol/MAC constructions, the keys \mathbf{X} and \mathbf{Y} will contain thousands of bits (as the dimension of each matrix will need to be hundreds of bits in length). This may make them too bulky to be stored in an RFID chip. To solve this problem in the $\text{HB}^\#$ protocol, a construction was proposed that utilizes random *Toeplitz* matrices as the keys. Toeplitz matrices are matrices M where $M_{i,j} = M_{i',j'}$ whenever $j - i = j' - i'$. Thus a Toeplitz matrix can be described by vector containing its first column and row, which drastically reduces the memory requirements. In addition, Toeplitz matrices have the following property [15].

Lemma A.1 *Let \mathcal{P} be the family of m by k Toeplitz matrices where \mathcal{P}_s is the Toeplitz matrix defined by the $k + m - 1$ -bit vector s . For any vector \mathbf{a} , $\mathbf{a}\mathcal{P}$ is uniformly distributed over k . In fact, $\mathbf{a}\mathcal{P}_s$ can be written as the inner product of s and a matrix derived from \mathbf{a} which has rank k .*

It is an open problem to prove that either our protocols, or the $\text{HB}^\#$ protocol in [15] are provably secure when the keys are random Toeplitz matrices.

If we use Toeplitz matrices, the key size for our MAC and our RFID protocol would be $2(m + n - 1)$, otherwise, the key would be of length $2mn$. Either way, computing the MAC or verifying it (or computing any step in the RFID protocol) would require just two $m \times n$ matrix multiplications, which is certainly feasible for RFID tags.