

Parallel and Concurrent Security of the HB and HB⁺ Protocols*

JONATHAN KATZ^{†‡}

JI SUN SHIN[†]

ADAM SMITH[§]

Abstract

Hopper and Blum (Asiacrypt 2001) and Juels and Weis (Crypto 2005) recently proposed two shared-key authentication protocols — HB and HB⁺, respectively — whose extremely low computational cost makes them attractive for low-cost devices such as radio-frequency identification (RFID) tags. The security of these protocols is based on the conjectured hardness of the “learning parity with noise” (LPN) problem, which is equivalent to the problem of decoding random binary linear codes. The HB protocol is proven secure against a passive (eavesdropping) adversary, while the HB⁺ protocol is proven secure against active attacks.

In this paper, we revisit the security analysis of these protocols and give simpler proofs of security that also have a number of technical advantages with respect to prior work. Most significantly, we prove security for *parallel* or *concurrent* executions, meaning that the protocols can be parallelized to run in fewer rounds. We also explicitly address the dependence of the soundness error on the number of iterations.

*The results of this work appeared in preliminary form in [26] and [27]. Some of this research was performed while J.K. and A.S. were visiting the Institute for Pure and Applied Mathematics (IPAM) at UCLA.

[†]Dept. of Computer Science, University of Maryland. {jkatz,sunny}@cs.umd.edu.

[‡]Research supported by NSF Trusted Computing grants #0310499, #0310751 and #0617306 and NSF CAREER award #0447075.

[§]Dept. of Computer Science and Engineering, The Pennsylvania State University. asmith@cse.psu.edu. Research supported in part by NSF CCF grant #0729171.

1 Introduction

Low-cost, resource-constrained devices such as radio-frequency identification (RFID) tags or sensor nodes demand extremely efficient algorithms and protocols. Securing such devices is a challenge since, in many cases, “traditional” cryptographic protocols are simply too computationally-intensive to be utilized. With this motivation in mind, Juels and Weis [25] — building upon work of Hopper and Blum [21, 22] — investigate two highly-efficient, shared-key (unidirectional) authentication protocols suitable for an RFID *tag* identifying itself to a tag *reader*. (We will sometimes refer to the tag as a *prover* and the tag reader as a *verifier*.) These protocols are extremely lightweight, requiring both parties to perform only a relatively small number of primitive bit-wise operations such as “XOR” and “AND,” and can thus be implemented using fewer than the 3-5K gates required to implement a block cipher such as DES or AES [25].

The two authentication protocols studied by Juels and Weis are both proven secure via reduction to the “learning parity with noise” (LPN) problem [2, 3, 4, 8, 20, 28, 21, 22, 35], which is related to the hardness of decoding a random linear code; a formal definition of the LPN problem as well as evidence for its difficulty are reviewed in Section 2.1. The first protocol (the HB protocol [21, 22]) is proven secure against a *passive* (eavesdropping) adversary, while the second (called HB⁺) is proven secure against the stronger class of *active* adversaries. In each case, Juels and Weis focus on a single, “basic authentication step” of the protocol and prove that a computationally-bounded adversary cannot succeed in impersonating a tag in this case with probability noticeably better than 1/2; that is, a single iteration of the protocol has *soundness error* 1/2. The implicit assumption is that repeating these “basic authentication steps” sufficiently-many times yields a protocol with negligible soundness error.

Difficulties and limitations. There are, however, some subtle limitations of the security proofs given by Juels and Weis. Most serious, perhaps, is a difficulty explicitly highlighted by Juels and Weis and regarded by them as a potential barrier to usage of the HB⁺ protocol in practice [25, Section 6]: the proof of security for HB⁺ requires that the adversary’s interactions with the tag (i.e., when the adversary is impersonating a tag reader) be *sequential*. Besides leaving in question the security of HB⁺ under *concurrent* executions, this also means that the HB⁺ protocol itself (which, recall, consists of sufficiently-many repetitions of an underlying basic authentication step) requires very high round complexity since the multiple iterations of the basic authentication step cannot be *parallelized* but must instead be performed sequentially. The difficulty and importance of proving security of various identification protocols under concurrent or parallel composition is well-understood, and many results are known: for example, the (black-box) zero-knowledge property of an identification protocol is not preserved under parallel [16] or concurrent [7] composition (though it is preserved under sequential composition [18]), whereas witness indistinguishability *is* preserved in these cases [10]. Unfortunately, the HB⁺ protocol is not known to satisfy either zero knowledge or witness indistinguishability and so such results are of no help here.

An additional difficulty, not explicitly mentioned in [25], is that it is unclear what is the exact relationship between the soundness error and the number of repetitions of the basic authentication step; this is true for both the HB and HB⁺ protocols, regardless of whether the repetitions are carried out in parallel or sequentially.¹ This is related to the more general question of *hardness amplification* (i.e., analyzing the difficulty of solving *multiple* instances of a problem compared to

¹Indeed, as we have noted, Juels and Weis [25] only prove soundness 1/2 for a basic authentication step and never make any claims regarding the security of multiple iterations (for either HB or HB⁺).

the difficulty of solving a *single* such instance) which has been studied in many different contexts [36, 17, 1, 15, 34, 6] and is surprisingly non-trivial to answer. Unfortunately, there does not seem to be any prior work that applies in our setting. Specifically:

- For the HB and HB⁺ protocols it is not possible to efficiently verify whether a given transcript is “successful” without possession of the secret key; thus, Yao’s “XOR-lemma” [36, 17] and related techniques that require efficient verifiability do not apply.
- Work on hardness amplification for “weakly-verifiable puzzles” [6] does not apply either. Although the HB/HB⁺ protocols can be viewed as efficiently-verifiable puzzles, existing results [6] only apply to *completely independent* instances of the “puzzle.” In particular, existing results imply that running the basic authentication step of the HB protocol n times *using n independent keys* yields soundness roughly $(1/2)^n$, but say nothing about running n iterations using the *same* key (which is the case we are interested in).
- The HB/HB⁺ protocols are *computationally*-sound only, and thus known results [15, Appendix C] [34] on soundness reduction for interactive proof systems (which apply only when soundness holds even against an all-powerful cheating prover) do not apply either.
- Limited positive results regarding soundness reduction for computationally-sound protocols exist [1, 33], but these results apply *only* when the verifier does not hold a secret key (or, more generally, when the verifier does not share state across different iterations). These results are therefore of no help when the same secret key is used across all iterations.

An additional difficulty in our setting is that HB and HB⁺ protocols do not have perfect completeness; indeed, crucial to both the HB and HB⁺ protocols is that the honest prover injects “noise” into its answers and so even the honest prover does not succeed with probability 1. This was not explicitly addressed in the security proofs of [25], either, and introduces additional complications.

1.1 Our Contributions

In this work we address the difficulties and open questions mentioned above, and show the following results: (1) the HB⁺ protocol remains secure under arbitrary concurrent interactions of the adversary with the honest prover/tag, and so in particular the iterations of the HB⁺ protocol can be parallelized; furthermore, (2) our security proofs explicitly incorporate the dependence of the soundness error on the number of iterations as well as on the error introduced by the honest prover.

Besides the results themselves, we believe the techniques and proofs given here are of independent interest for future work on cryptographic applications of the LPN problem. Our main technical tool is a result due to Regev [35] (see also [3]) showing that the hardness of the LPN problem implies the pseudorandomness of a certain distribution. Using this, we give proofs which we believe are substantially *simpler* than those given in [25], and also more *complete* in that, in contrast to [25], they explicitly deal with the dependence of soundness on the number of iterations and also the issues arising due to non-perfect completeness. Our proofs also use bounds from coding theory [19, 23, 24] in a novel way.

1.2 Additional Discussion

The problem of secure authentication using a shared, secret key is well understood, and many widely-known solutions based on, e.g., block ciphers are available. The aim of the line of research

considered here, as in [25], is to develop protocols which are exceptionally efficient (i.e., potentially more efficient than hardware implementations of block ciphers such as DES or AES) while still guaranteeing some useful level of provable security. Of course, the protocols described here are far from solving the problem completely. For example, Gilbert, Robshaw, and Silbert [12] have recently shown a man-in-the-middle attack on the HB⁺ protocol. Although their attack would be devastating if carried out successfully, the possibility of such an attack does not mean that it is useless to explore the security of the HB/HB⁺ protocols in weaker attack models. For one, man-in-the-middle attacks can be difficult to carry out. Especially in the case of RFID, where communication is inherently short range, it appears much more difficult to mount a man-in-the-middle attack than an active attack.² (The reader is referred to the work of Wool, et al. [29, 30], for an illuminating discussion on the feasibility of various attacks in RFID systems.) Juels and Weis further note [25, Appendix A] that the man-in-the-middle attack of [12] does not apply in a *detection-based* system where numerous failed authentication attempts immediately raise an alarm. Our work can thus be viewed as quantifying more precisely the tradeoff between efficiency and privacy provided by the HB/HB⁺ protocols.

Beyond our concrete results, we also hope that the techniques introduced in this paper will prove useful in analyzing future variants of the HB/HB⁺ protocols, as well as other protocols based on the LPN problem.

2 Definitions and Preliminaries

We formally define the LPN problem and state and prove the main technical lemma on which we rely. We also define our notion(s) of security for identification; these are standard, but some complications arise due to the fact that the HB/HB⁺ protocols do not have perfect completeness.

2.1 The LPN Problem

A function $\varepsilon : \mathbb{N}^+ \rightarrow \mathbb{R}^+ \cup \{0\}$ is *negligible* if it is asymptotically smaller than any inverse polynomial, i.e., if for every polynomial p there exists a K such that $k > K$ implies $\varepsilon(k) \leq 1/p(k)$. We use k for the security parameter, so that all parties are assumed to run in time polynomial in k and the soundness error should be negligible in k . We let PPT stand for “probabilistic polynomial time”.

If $\mathbf{s}, \mathbf{a}_1, \dots, \mathbf{a}_\ell$ are binary vectors of length k , let $z_i = \langle \mathbf{s}, \mathbf{a}_i \rangle$ denote the dot product of \mathbf{s} and \mathbf{a}_i (modulo 2). Given the values $\mathbf{a}_1, z_1, \dots, \mathbf{a}_\ell, z_\ell$ for randomly-chosen $\{\mathbf{a}_i\}$ and $\ell = \Theta(k)$, it is possible to efficiently solve for \mathbf{s} using standard linear-algebraic techniques. However, in the presence of *noise* where each z_i is flipped (independently) with probability ε , finding \mathbf{s} becomes much more difficult. We refer to the problem of learning \mathbf{s} in this latter case as *the LPN problem*.

For the formal definition, let Ber_ε be the Bernoulli distribution with parameter $\varepsilon \in (0, \frac{1}{2})$ (so if $\nu \sim \text{Ber}_\varepsilon$ then $\Pr[\nu = 1] = \varepsilon$ and $\Pr[\nu = 0] = 1 - \varepsilon$), and let $A_{\mathbf{s}, \varepsilon}$ be the distribution defined by:

$$\left\{ \mathbf{a} \leftarrow \{0, 1\}^k; \nu \leftarrow \text{Ber}_\varepsilon : (\mathbf{a}, \langle \mathbf{s}, \mathbf{a} \rangle \oplus \nu) \right\}.$$

Also let $A_{\mathbf{s}, \varepsilon}$ denote an oracle which outputs (independent) samples according to this distribution.

²Though there have been claims of being able to read some RFID tags over as much as 69 feet, the maximum distance from which many commonly-used cards can be read appears to be almost two orders of magnitude lower [29]. Note further that a man-in-the-middle attack requires the ability to *send* data to the tag (and reader).

For some fixed value of k , algorithm M is said to (t, q, δ) -solve the LPN_ε problem if

$$\Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k : M^{A_{\mathbf{s}, \varepsilon}}(1^k) = \mathbf{s} \right] \geq \delta,$$

and furthermore M runs in time at most t and makes at most q queries to its oracle.³ In asymptotic terms, in the standard way, the LPN_ε problem is “hard” if every probabilistic polynomial-time algorithm M solves the LPN_ε problem with only negligible probability (where the algorithm’s running time and success probability are functions of k).

The error parameter ε is usually taken to be a fixed constant independent of k , as will be the case here. The value of ε to use depends on a number of tradeoffs and design decisions: although, roughly speaking, the LPN_ε problem appears to become “harder” as ε increases, a larger value of ε also implies that the honest prover is rejected more often (as will become clear when we describe the HB/HB^+ protocols, below). Our results are meaningful for all $\varepsilon \in (0, \frac{1}{2})$.

The hardness of the LPN_ε problem has been studied in many previous works. It can be formulated also as the problem of decoding a random linear code [2, 35], and is \mathcal{NP} -complete [2] as well as hard to approximate within a factor better than 2 (where the optimization problem is phrased as finding an \mathbf{s} satisfying the most equations) [20]. These worst-case hardness results are complemented by numerous studies of the average-case hardness of the problem [3, 4, 8, 28, 21, 22, 35]. Most relevant for our purposes is that the current best-known algorithms for solving the LPN_ε problem [4, 31, 11] for any constant ε require $t, q = 2^{\Theta(k/\log k)}$. We refer the reader to [25, Appendix D] and [31, 11] for more exact estimates of the running time of this algorithm, as well as suggested practical values for k .

2.2 A Technical Lemma

In this section we prove a key technical lemma: hardness of the LPN_ε problem implies “pseudo-randomness” of $A_{\mathbf{s}, \varepsilon}$. Specifically, let U_{k+1} denote the uniform distribution on $(k+1)$ -bit strings. The following lemma shows that oracle access to $A_{\mathbf{s}, \varepsilon}$ (for randomly-chosen \mathbf{s}) is indistinguishable from oracle access to U_{k+1} . A proof of the following is essentially in [35, Sect. 4], although we have fleshed out some of the details and worked out the concrete parameters of the reduction.

Lemma 1 *Say there exists an algorithm D making q oracle queries, running in time t , and with*

$$\left| \Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k : D^{A_{\mathbf{s}, \varepsilon}}(1^k) = 1 \right] - \Pr \left[D^{U_{k+1}}(1^k) = 1 \right] \right| \geq \delta.$$

Then there exists an algorithm M making $q' = \mathcal{O}(q \cdot \delta^{-2} \log k)$ oracle queries, running in time $t' = \mathcal{O}(t \cdot k \delta^{-2} \log k)$, and such that

$$\Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k : M^{A_{\mathbf{s}, \varepsilon}}(1^k) = \mathbf{s} \right] \geq \delta/4.$$

(We remark that various tradeoffs are possible between the number of queries/running time of M and its success probability in solving LPN_ε ; see [35, Sect. 4]. We aimed for simplicity in the proof rather than trying to optimize parameters.)

Proof. Set $N = \Theta(\delta^{-2} \log k)$. Algorithm $M^{A_{\mathbf{s}, \varepsilon}}(1^k)$ proceeds as follows:

³Our formulation of the LPN problem follows, e.g., [35]; the formulation in, e.g., [25] allows M to output any \mathbf{s} satisfying at least a $(1 - \varepsilon)$ fraction of the equations returned by $A_{\mathbf{s}, \varepsilon}$. It is easy to see that for q large enough these formulations are equivalent as with overwhelming probability there will be a unique such \mathbf{s} .

1. M chooses random coins ω for D and uses these for the remainder of its execution.
2. M runs $D^{U_{k+1}}(1^k; \omega)$ for a total of N times to obtain an estimate p for the probability that D outputs 1 in this case. (The probability here is over the responses from the oracle.)
3. M obtains $q \cdot N$ samples $\{(\mathbf{a}_{1,j}, z_{1,j})\}_{j=1}^q, \dots, \{(\mathbf{a}_{N,j}, z_{N,j})\}_{j=1}^q$ from $A_{\mathbf{s}, \varepsilon}$.
4. For $i = 1$ to k :
 - (a) Run $D(1^k; \omega)$ for a total of N times, using a fresh set of samples $\{(\mathbf{a}_j, z_j)\}_{j=1}^q$ to answer the q oracle queries of D each time. Answer the j^{th} oracle query of D in each iteration by choosing a random bit c_j and returning $(\mathbf{a}_j \oplus (c_j \cdot \mathbf{e}_i), z_j)$, where \mathbf{e}_i is the vector with 1 at position i and 0s elsewhere. Obtain an estimate p_i for the probability that D outputs 1 in this case. (Note that the samples that M obtains in step 3 are re-used for different values of i .)
 - (b) If $|p_i - p| \geq \delta/4$ set $s'_i = 0$; else set $s'_i = 1$.
4. Output $\mathbf{s}' = (s'_1, \dots, s'_k)$.

Let us analyze the behavior of M . First note that, by a standard averaging argument, with probability at least $\delta/2$ over choice of \mathbf{s} and random coins ω it holds that

$$\left| \Pr \left[D^{A_{\mathbf{s}, \varepsilon}}(1^k; \omega) = 1 \right] - \Pr \left[D^{U_{k+1}}(1^k; \omega) = 1 \right] \right| \geq \delta/2, \quad (1)$$

where the probabilities are taken over the answers D receives from its oracle. We restrict our attention to \mathbf{s}, ω for which Eq. (1) holds and show that in this case M outputs $\mathbf{s}' = \mathbf{s}$ with probability at least $1/2$. The theorem follows.

Setting $N = \Theta(\delta^{-2} \log(k))$, we can ensure that

$$\left| \Pr \left[D^{U_{k+1}}(1^k; \omega) = 1 \right] - p \right| \leq \delta/16 \quad (2)$$

except with probability at most $O(1/k)$. Next focus on a particular iteration i of steps 4(a) and 4(b). Letting hyb_i denote the distribution of the answers returned to D in this iteration, we again have

$$\left| \Pr \left[D^{\text{hyb}_i}(1^k; \omega) = 1 \right] - p_i \right| \leq \delta/16 \quad (3)$$

except with probability at most $O(1/k)$. Applying a union bound (and setting parameters appropriately) we see that Eqs. (2) and (3) hold (the latter for all $i \in [k]$) with probability at least $1/2$. We assume this to be the case for the rest of the proof, and show that when this occurs then M always outputs $\mathbf{s}' = \mathbf{s}$.

Let $\mathbf{s} = (s_1, \dots, s_k)$. We claim that if $s_i = 0$ then $\text{hyb}_i = A_{\mathbf{s}, \varepsilon}$, while if $s_i = 1$ then $\text{hyb}_i = U_{k+1}$. To see this note that when $s_i = 0$ the answer $(\mathbf{a}_j \oplus (c_j \cdot \mathbf{e}_i), z_j)$ returned to D is distributed exactly according to $A_{\mathbf{s}, \varepsilon}$ since $\langle \mathbf{s}, \mathbf{a}_j \rangle = \langle \mathbf{s}, \mathbf{a}_j \oplus (c_j \cdot \mathbf{e}_i) \rangle$ regardless of c_j . On the other hand, if $s_i = 1$ then $\langle \mathbf{s}, \mathbf{a}_j \rangle$ is a random bit, independent of $\mathbf{a}_j \oplus (c_j \cdot \mathbf{e}_i)$, and hence z_j is also a random bit.

It follows that if $s_i = 0$ then

$$\left| \Pr \left[D^{\text{hyb}_i}(1^k; \omega) = 1 \right] - \Pr \left[D^{U_{k+1}}(1^k; \omega) = 1 \right] \right| \geq \delta/2$$

(by Eq. (1)), and so $|p_i - p| \geq \frac{\delta}{2} - 2 \cdot \frac{\delta}{16} = \frac{3\delta}{8}$ (using Eqs. (2) and (3)) and $s'_i = 0 = s_i$. When $s_i = 1$ then

$$\Pr \left[D^{\text{hyb}_i}(1^k; \omega) = 1 \right] = \Pr \left[D^{U_{k+1}}(1^k; \omega) = 1 \right],$$

and so $|p_i - p| \leq 2 \cdot \frac{\delta}{16} = \frac{\delta}{8}$ (again using Eqs. (2) and (3)) and $s'_i = 1 = s_i$. Since this holds for all $i \in \{1, \dots, k\}$, we conclude that $\mathbf{s}' = \mathbf{s}$. \blacksquare

2.3 Overview of the HB/HB⁺ Protocols, and Security Definitions

Recall that we let k denote our security parameter. The HB and HB⁺ protocols as analyzed here consist of $n = n(k)$ *parallel* iterations of a “basic authentication step.” We describe the basic authentication step for the HB protocol, and defer a discussion of the HB⁺ protocol to Section 4.1. In the HB protocol, a tag \mathcal{T} and a reader \mathcal{R} share a random secret key $\mathbf{s} \in \{0, 1\}^k$; a basic authentication step consists of the reader sending a random challenge $\mathbf{a} \in \{0, 1\}^k$ to the tag, which replies with $z = \langle \mathbf{s}, \mathbf{a} \rangle \oplus \nu$ for $\nu \sim \text{Ber}_\varepsilon$. The reader can then verify whether the response z of the tag satisfies $z \stackrel{?}{=} \langle \mathbf{s}, \mathbf{a} \rangle$; we say the iteration is *successful* if this is the case. See Figure 1.

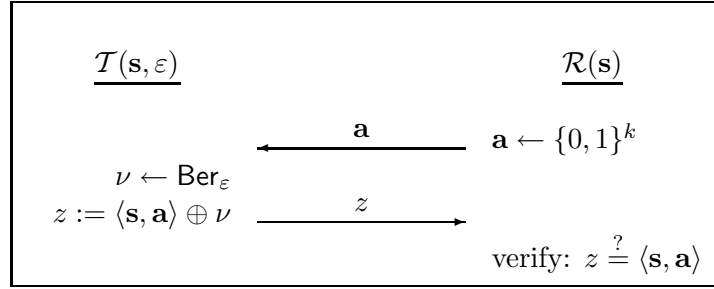


Figure 1: The basic authentication step of the HB protocol.

Even for an honest tag a basic iteration is unsuccessful with probability ε . For this reason, a reader accepts upon completion of all n iterations of the basic authentication step as long as the number of unsuccessful iterations is not “too high”. More precisely, let $u = u(k)$ be such that $\varepsilon \cdot n \leq u$; then the reader accepts as long as the number of unsuccessful iterations is at most⁴ u . (Overall, then, the entire HB protocol is parameterized by ε, n , and u .) For an honest tag, each iteration is independent of the others and so the completeness error ε_c (i.e., the probability that an honest tag is rejected) can be calculated using a Chernoff bound. In particular, for any positive constant δ , setting $u = (1 + \delta)\varepsilon n$ suffices to achieve ε_c negligible in n .

By sending random responses in each of the n iterations, an adversary trying to impersonate a valid tag succeeds with probability

$$\delta_{\varepsilon, u, n}^* \stackrel{\text{def}}{=} 2^{-n} \cdot \sum_{i=0}^u \binom{n}{i};$$

⁴Note in particular that if u is set to *exactly* $\varepsilon \cdot n$ then the completeness error will be rather high. One can imagine changing the protocol so that the tag introduces *at most* $\varepsilon \cdot n$ errors (and iterations are no longer independent); see Section 5 for discussion of this point.

that is, $\delta_{\varepsilon, u, n}^*$ is the *best* possible soundness error we can hope to achieve for the given setting of the parameters. Asymptotically, as long as $u \leq (1 - \delta) \cdot n/2$ for some positive constant δ , the success of this trivial attack will be negligible in n . (This can again be analyzed using a Chernoff bound.)

Let $\mathcal{T}_{\mathbf{s}, \varepsilon, n}^{\text{HB}}$ denote the tag algorithm in the HB protocol when the tag holds secret key \mathbf{s} (note that the tag algorithm is independent of u), and let $\mathcal{R}_{\mathbf{s}, \varepsilon, u, n}^{\text{HB}}$ similarly denote the algorithm run by the tag reader. We denote a complete execution of the HB protocol between a party $\hat{\mathcal{T}}$ and the reader \mathcal{R} by $\langle \hat{\mathcal{T}}, \mathcal{R}_{\mathbf{s}, \varepsilon, u, n}^{\text{HB}} \rangle$ and say this equals 1 iff the reader accepts.

For the case of a passive attack on the HB protocol, we imagine an adversary \mathcal{A} running in two stages: in the first stage the adversary obtains polynomially-many transcripts⁵ of (honest) executions of the protocol by interacting with an oracle $\text{trans}_{\mathbf{s}, \varepsilon, n}^{\text{HB}}$ (this models eavesdropping); in the second stage, the adversary interacts with the reader and tries to impersonate the tag. We define the adversary's advantage as

$$\text{Adv}_{\mathcal{A}, \text{HB}}^{\text{passive}}(\varepsilon, u, n) \stackrel{\text{def}}{=} \Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k; \mathcal{A}^{\text{trans}_{\mathbf{s}, \varepsilon, n}^{\text{HB}}}(1^k) : \langle \mathcal{A}, \mathcal{R}_{\mathbf{s}, \varepsilon, u, n}^{\text{HB}} \rangle = 1 \right].$$

We say the HB protocol is *secure against passive attacks* (for a particular setting of ε and $u = u(k)$, $n = n(k)$) if for all PPT adversaries \mathcal{A} we have that $\text{Adv}_{\mathcal{A}, \text{HB}}^{\text{passive}}(\varepsilon, u, n)$ is negligible in k .

As we will describe in Section 4.1, the HB^+ protocol uses two keys $\mathbf{s}_1, \mathbf{s}_2$. We let $\mathcal{T}_{\mathbf{s}_1, \mathbf{s}_2, \varepsilon, n}^{\text{HB}^+}$ denote the tag algorithm in this case, and let $\mathcal{R}_{\mathbf{s}_1, \mathbf{s}_2, \varepsilon, u, n}^{\text{HB}^+}$ denote the algorithm run by the tag reader. For the case of an active attack on the HB^+ protocol, we again imagine an adversary running in two stages: in the first stage the adversary interacts polynomially-many times with the honest tag algorithm (with concurrent executions allowed), while in the second stage the adversary interacts only with the reader. The adversary's advantage in this case is

$$\text{Adv}_{\mathcal{A}, \text{HB}^+}^{\text{active}}(\varepsilon, u, n) \stackrel{\text{def}}{=} \Pr \left[\mathbf{s}_1, \mathbf{s}_2 \leftarrow \{0, 1\}^k; \mathcal{A}^{\mathcal{T}_{\mathbf{s}_1, \mathbf{s}_2, \varepsilon, n}^{\text{HB}^+}}(1^k) : \langle \mathcal{A}, \mathcal{R}_{\mathbf{s}_1, \mathbf{s}_2, \varepsilon, u, n}^{\text{HB}^+} \rangle = 1 \right].$$

We say the HB^+ protocol is *secure against active attacks* (for a particular setting of ε and $u = u(k)$, $n = n(k)$) if for all PPT adversaries \mathcal{A} we have that $\text{Adv}_{\mathcal{A}, \text{HB}^+}^{\text{active}}(\varepsilon, u, n)$ is negligible in k .

We remark that allowing the adversary to interact with the reader multiple times (even concurrently) does not give the adversary any additional advantage other than the fact that, as usual, the probability that the adversary succeeds in at least one impersonation attempt scales linearly with the number of attempts.

3 Security of the HB Protocol against Passive Attacks

Recall from the previous section that the HB protocol is parameterized by ε (a measure of the noise introduced by the tag), u (which determines the completeness error ε_c as well as the best achievable soundness), and n (the number of iterations of the basic authentication step given in Figure 1). We stress that these n iterations are run *in parallel*, so the entire protocol requires only two rounds.

⁵Note in particular that the adversary is assumed not to learn whether or not the reader accepts. Since, as discussed earlier, the parameters can be set such that the reader accepts an honest tag with all but negligible probability, this makes no difference as far as asymptotic security is concerned.

Theorem 2 Assume the LPN_ε problem is hard, where $0 < \varepsilon < \frac{1}{2}$. Let $n = \Theta(k)$ and $u = \varepsilon^+ \cdot n$, where ε^+ is a constant satisfying $\varepsilon < \varepsilon^+ < \frac{1}{2}$. Then the HB protocol with these settings of the parameters has negligible completeness error, and is secure against passive attacks.

A standard Chernoff bound shows that the completeness error is negligible for the given setting of the parameters. Therefore, we focus only on the security of the protocol against passive attacks. We deal first with the case $\varepsilon < \varepsilon^+ < 1/4$ since this case admits a significantly simpler analysis. We then show how to extend the proof to the case $\varepsilon < 1/2$.

Claim 3 Say there exists an adversary \mathcal{A} eavesdropping on at most q executions of the HB protocol, running in time t , and achieving $\text{Adv}_{\mathcal{A}, \text{HB}}^{\text{passive}}(\varepsilon, u, n) = \delta$. Then there exists an algorithm D making $(q + 1) \cdot n$ oracle queries, running in time $O(t)$, and such that

$$\left| \Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k : D^{A_{\mathbf{s}, \varepsilon}}(1^k) = 1 \right] - \Pr \left[D^{U_{k+1}}(1^k) = 1 \right] \right| \geq \delta - \varepsilon_c - 2^{-n} \cdot \sum_{i=0}^{2u} \binom{n}{i}.$$

Asymptotically, for any $\varepsilon < \varepsilon^+ < \frac{1}{4}$ and n, u as in Theorem 2, the final two terms of the above expression are negligible. Thus, the claim together with Lemma 1 proves Theorem 2 for this case.

Proof. D , given access to an oracle returning $(k + 1)$ -bit strings (\mathbf{a}, z) , proceeds as follows:

1. D runs the first phase of \mathcal{A} . Each time \mathcal{A} requests to view a transcript of the protocol, D obtains n samples $\{(\mathbf{a}_i, z_i)\}_{i=1}^n$ from its oracle and returns these to \mathcal{A} .
2. When \mathcal{A} is ready for the second phase, D again obtains n samples $\{(\bar{\mathbf{a}}_i, \bar{z}_i)\}_{i=1}^n$ from its oracle. D sends the challenge $(\bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_n)$ to \mathcal{A} and receives in return a response $Z' = (z'_1, \dots, z'_n)$.
3. D outputs 1 iff $\bar{Z} \stackrel{\text{def}}{=} (\bar{z}_1, \dots, \bar{z}_n)$ and Z' differ in at most $2u$ entries.

When D 's oracle is U_{k+1} , it is clear that D outputs 1 with probability exactly $2^{-n} \cdot \sum_{i=0}^{2u} \binom{n}{i}$ since \bar{Z} is in this case uniformly distributed and independent of everything else. On the other hand, when D 's oracle is $A_{\mathbf{s}, \varepsilon}$ then the transcripts D provides to \mathcal{A} during the first phase of \mathcal{A} 's execution are distributed identically to real transcripts in an execution of the HB protocol. Let $Z^* \stackrel{\text{def}}{=} (\langle \mathbf{s}, \bar{\mathbf{a}}_1 \rangle, \dots, \langle \mathbf{s}, \bar{\mathbf{a}}_n \rangle)$ be the vector of correct answers to the challenge $(\bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_n)$ sent by D in the second phase. Then with probability at least δ it holds that Z' and Z^* differ in at most u entries (since \mathcal{A} successfully impersonates the tag with this probability). Also, since \bar{Z} is distributed exactly as the answers of an honest tag, \bar{Z} and Z^* differ in at most u positions except with probability at most ε_c . It follows that with probability at least $\delta - \varepsilon_c$ the vectors Z' and \bar{Z} differ in at most $2u$ entries, and so D outputs 1 with at least this probability. ■

We next consider the general case of $\varepsilon < 1/2$. We do not provide concrete bounds in this case, though such bounds may be derived easily from the proof that follows.

Proof (of Theorem 2). Fix some PPT adversary \mathcal{A} attacking the HB protocol, and let $\delta \stackrel{\text{def}}{=} \text{Adv}_{\mathcal{A}, \text{HB}}^{\text{passive}}(\varepsilon, u, n)$. We construct a PPT adversary D attempting to distinguish whether it is given oracle access to $A_{\mathbf{s}, \varepsilon}$ or to U_{k+1} (as in Lemma 1). Relating the advantage of D to the advantage of \mathcal{A} gives the stated result.

The first two steps of our algorithm D are identical to those in the previous proof, and only the third step differs. For convenience we repeat the first two steps here. D , given access to an oracle returning $(k + 1)$ -bit strings (\mathbf{a}, z) , proceeds as follows:

1. D runs the first phase of \mathcal{A} . Each time \mathcal{A} requests to view a transcript of the protocol, D obtains n samples $\{(\mathbf{a}_i, z_i)\}_{i=1}^n$ from its oracle and returns these to \mathcal{A} .
2. When \mathcal{A} is ready for the second phase, D again obtains n samples $\{(\bar{\mathbf{a}}_i, \bar{z}_i)\}_{i=1}^n$ from its oracle. D sends the challenge $(\bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_n)$ to \mathcal{A} and receives in return a response $Z' = (z'_1, \dots, z'_n)$.
3. D outputs 1 iff $\bar{Z} \stackrel{\text{def}}{=} (\bar{z}_1, \dots, \bar{z}_n)$ and Z' differ in at most $u' \stackrel{\text{def}}{=} \varepsilon^{++} \cdot n$ entries, where ε^{++} is a constant satisfying $\varepsilon^+ - 2\varepsilon^+\varepsilon + \varepsilon < \varepsilon^{++} < \frac{1}{2}$. (Note that for $\varepsilon < 1/2$, $\varepsilon^+ < 1/2$, we have

$$\begin{aligned} \varepsilon^+ - 2\varepsilon^+\varepsilon + \varepsilon &= \varepsilon^+ \cdot (1 - 2\varepsilon) + \varepsilon \\ &< \frac{1}{2} \cdot (1 - 2\varepsilon) + \varepsilon = \frac{1}{2}, \end{aligned}$$

and so ε^{++} in the desired range exists.)

When D 's oracle is U_{k+1} , it is clear that D outputs 1 with probability $2^{-n} \cdot \sum_{i=0}^{u'} \binom{n}{i}$ since \bar{Z} is in this case uniformly distributed and independent of everything else. Since $u' < n/2$, this quantity is negligible in k for the given settings of the other parameters.

When D 's oracle is $A_{s,\varepsilon}$ then the transcripts D provides to \mathcal{A} during the first phase of \mathcal{A} 's execution are distributed identically to real transcripts in an execution of the HB protocol. Letting $Z^* \stackrel{\text{def}}{=} (\langle s, \bar{\mathbf{a}}_1 \rangle, \dots, \langle s, \bar{\mathbf{a}}_n \rangle)$ be the vector of correct answers to the challenge $(\bar{\mathbf{a}}_1, \dots, \bar{\mathbf{a}}_n)$ sent by D in the second phase, it follows that with probability δ (i.e., the impersonation probability of \mathcal{A}) the vector of responses Z' given by \mathcal{A} differs from Z^* in at most u entries. We show below that conditioned on this event, Z' and \bar{Z} differ in at most u' entries with all but negligible probability. Thus, D outputs 1 in this case with probability negligibly close to δ . We conclude from Lemma 1 that δ must be negligible.

Let $\mathbf{wt}(Z)$ denote the Hamming weight of a vector Z ; i.e., $\mathbf{wt}(Z)$ is the number of entries of Z equal to 1. The distance between two vectors Z_1, Z_2 is exactly $\mathbf{wt}(Z_1 \oplus Z_2)$. We show that, conditioned on $\mathbf{wt}(Z' \oplus Z^*) \leq u$, we have $\mathbf{wt}(Z' \oplus \bar{Z}) \leq u'$ with all but negligible probability.

Write $Z' = Z^* \oplus \mathbf{w}$ for some vector \mathbf{w} of weight at most $u = \varepsilon^+ n$. The vector \bar{Z} is generated by the following process: choose an error vector \mathbf{e} by setting each position of \mathbf{e} (independently) to 1 with probability ε , and then set $\bar{Z} = Z^* \oplus \mathbf{e}$. We see that the probability that \bar{Z} differs from Z' in at most u' entries is precisely the probability that

$$\mathbf{wt}(Z' \oplus \bar{Z}) = \mathbf{wt}(\mathbf{w} \oplus \mathbf{e}) \leq u'.$$

The random variable $\mathbf{wt}(\mathbf{w} \oplus \mathbf{e})$, where \mathbf{w} is fixed, is the sum of n independent indicator random variables, one for each position of the vector $\mathbf{w} \oplus \mathbf{e}$. The expectation of $\mathbf{wt}(\mathbf{w} \oplus \mathbf{e})$ is

$$\begin{aligned} \mathbf{wt}(\mathbf{w}) \cdot (1 - \varepsilon) + (n - \mathbf{wt}(\mathbf{w})) \cdot \varepsilon &\leq \varepsilon^+ n \cdot (1 - \varepsilon) + (n - \varepsilon^+ n) \cdot \varepsilon \\ &= (\varepsilon^+ - 2\varepsilon^+\varepsilon + \varepsilon) \cdot n. \end{aligned}$$

Since ε^{++} is a constant strictly larger than $(\varepsilon^+ - 2\varepsilon^+\varepsilon + \varepsilon)$, the Chernoff bound implies that $\mathbf{wt}(\mathbf{w} \oplus \mathbf{e}) \leq \varepsilon^{++} n$ with all but negligible probability. \blacksquare

4 The HB⁺ Protocol

4.1 Description of the Protocol

It is easy to see that the HB protocol is insecure against an active adversary. To achieve security against active attacks, Juels and Weis propose to modify the HB protocol by having the tag and

reader share *two* (independent) keys $\mathbf{s}_1, \mathbf{s}_2 \in \{0, 1\}^k$. A basic authentication step now consists of three rounds: first the tag sends a random “blinding factor” $\mathbf{b} \in \{0, 1\}^k$; the reader replies with a random challenge $\mathbf{a} \in \{0, 1\}^k$ as before; and finally the tag replies with $z = \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle \oplus \nu$ for $\nu \leftarrow \text{Ber}_\varepsilon$. As in the HB protocol, the tag reader can verify whether the response z satisfies $z \stackrel{?}{=} \langle \mathbf{s}_1, \mathbf{b} \rangle \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle$, and we again say the iteration is *successful* if this is the case. See Figure 2.

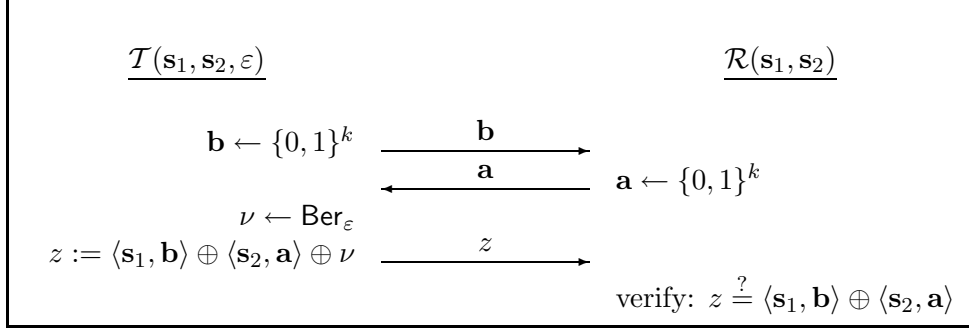


Figure 2: The basic authentication step of the HB^+ protocol.

The actual HB^+ protocol consists of n parallel iterations of the basic authentication step (and so the entire protocol requires only three rounds). The protocol also depends upon a parameter \mathbf{u} as in the case of the HB protocol, and this will again affect the completeness error as well as the best achievable soundness.

4.2 Security of the HB^+ Protocol against Active Attacks

We now prove security of the HB^+ protocol against active attacks.

Theorem 4 *Assume the LPN_ε problem is hard, where $0 < \varepsilon < \frac{1}{2}$. Let $n = \Theta(k)$ and $\mathbf{u} = \varepsilon^+ \cdot n$, where ε^+ is a constant satisfying $\varepsilon < \varepsilon^+ < \frac{1}{2}$. Then the HB^+ protocol with these settings of the parameters has negligible completeness error, and is secure against active attacks.*

A standard Chernoff bound shows that the completeness error is negligible for the given setting of the parameters. Therefore, we focus only on the security of the protocol against active attacks. As in the previous section, we deal first with the case of $\varepsilon < \varepsilon^+ < 1/4$. In that case, we also assume for simplicity that $n \ll k$ (specifically, that $k - n = \Theta(k)$). We then extend the proof to handle any $\varepsilon < 1/2$, and arbitrary n .

Claim 5 *Say there exists an adversary \mathcal{A} interacting with the tag in at most q executions of the HB^+ protocol (possibly concurrently), running in time t , and achieving $\text{Adv}_{\mathcal{A}, \text{HB}^+}^{\text{active}}(\varepsilon, \mathbf{u}, n) = \delta$. Then there exists an algorithm D making $q \cdot n$ oracle queries, running in time $O(t)$, and such that*

$$\left| \Pr \left[\mathbf{s} \leftarrow \{0, 1\}^k : D^{\mathcal{A}, \varepsilon}(1^k) = 1 \right] - \Pr \left[D^{U_{k+1}}(1^k) = 1 \right] \right| \geq \delta^2 - \frac{2^n}{2^k} - 2^{-n} \cdot \sum_{i=0}^{2\mathbf{u}} \binom{n}{i}.$$

Asymptotically, when $\varepsilon < \varepsilon^+ < \frac{1}{4}$, the number of iterations n satisfies $n = \delta \cdot k$ for constant $\delta < 1$, and \mathbf{u} is as in Theorem 4, the final two terms of the above expression are negligible. Thus, the claim together with Lemma 1 proves Theorem 4 in this case.

Proof. D , given access to an oracle returning $(k+1)$ -bit strings (\mathbf{b}, \bar{z}) , proceeds as follows:

1. D chooses $\mathbf{s}_2 \in \{0, 1\}^k$ uniformly at random.
2. D runs the first phase of \mathcal{A} . To simulate a basic authentication step, D obtains a sample (\mathbf{b}, \bar{z}) from its oracle and sends \mathbf{b} as the initial message. \mathcal{A} replies with a challenge \mathbf{a} , and then D responds with $z = \bar{z} \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle$. Note that since D does not rewind \mathcal{A} here, there is no difficulty in simulating the n parallel executions of the basic authentication step (nor in simulating concurrent executions of the entire protocol).
3. When \mathcal{A} begins the second phase of its attack, it first sends an initial message $\mathbf{b}_1, \dots, \mathbf{b}_n$ (we now explicitly consider all n parallel iterations of the protocol rather than focusing on a single basic authentication step). In response, D chooses random $\mathbf{a}_1^1, \dots, \mathbf{a}_n^1 \in \{0, 1\}^k$, sends these challenges to \mathcal{A} , and records \mathcal{A} 's response z_1^1, \dots, z_n^1 . Then D rewinds \mathcal{A} , chooses random $\mathbf{a}_1^2, \dots, \mathbf{a}_n^2 \in \{0, 1\}^k$, sends these to \mathcal{A} , and records \mathcal{A} 's response z_1^2, \dots, z_n^2 .
4. Let $z_i^\oplus \stackrel{\text{def}}{=} z_i^1 \oplus z_i^2$ and set $Z^\oplus \stackrel{\text{def}}{=} (z_1^\oplus, \dots, z_n^\oplus)$. Let $\hat{\mathbf{a}}_i = \mathbf{a}_i^1 \oplus \mathbf{a}_i^2$ and $\hat{z}_i = \langle \mathbf{s}_2, \hat{\mathbf{a}}_i \rangle$, and set $\hat{Z} \stackrel{\text{def}}{=} (\hat{z}_1, \dots, \hat{z}_n)$. D outputs 1 iff Z^\oplus and \hat{Z} differ in at most $2u$ entries.

Let us analyze the behavior of D :

Case 1: Say D 's oracle is U_{k+1} . In step 2, above, since \bar{z} is uniformly distributed and independent of everything else, the answers z that D returns to \mathcal{A} are uniformly distributed and independent of everything else. It follows that \mathcal{A} 's view throughout the entire experiment is independent of the secret \mathbf{s}_2 chosen by D .

The $\{\hat{\mathbf{a}}_i\}_{i=1}^n$ are uniformly and independently distributed, and so except with probability $\frac{2^n}{2^k}$ they are linearly independent and non-zero (this is a standard combinatorial result that is easy to prove). Assuming this to be the case, \hat{Z} is uniformly distributed over $\{0, 1\}^n$ from the point of view of \mathcal{A} . But then the probability that Z^\oplus and \hat{Z} differ in at most $2u$ entries is exactly $2^{-n} \cdot \sum_{i=0}^{2u} \binom{n}{i}$. We conclude that D outputs 1 in this case with probability at most $\frac{2^n}{2^k} + 2^{-n} \cdot \sum_{i=0}^{2u} \binom{n}{i}$.

Case 2: Say D 's oracle is $A_{\mathbf{s}_1, \varepsilon}$ for randomly-chosen \mathbf{s}_1 . In this case, D provides a perfect simulation for the first phase of \mathcal{A} . Let ω denote all the randomness used to simulate the first phase of \mathcal{A} (namely, the keys $\mathbf{s}_1, \mathbf{s}_2$, the randomness of \mathcal{A} , and the randomness used to respond to \mathcal{A} 's queries). For a fixed such ω , let δ_ω denote the probability, over random choice of $\mathbf{a}_1, \dots, \mathbf{a}_n$, that \mathcal{A} successfully impersonates the honest tag in the second phase. The probability that \mathcal{A} successfully responds to both sets of queries $\mathbf{a}_1^1, \dots, \mathbf{a}_n^1$ and $\mathbf{a}_1^2, \dots, \mathbf{a}_n^2$ sent by D is thus δ_ω^2 . The overall probability that \mathcal{A} successfully responds to both sets of queries is then given by

$$\mathbf{E}_\omega(\delta_\omega^2) \geq (\mathbf{E}_\omega(\delta_\omega))^2 = \delta^2,$$

using Jensen's inequality (here \mathbf{E}_ω denotes the expectation over the choice of ω).

Assuming \mathcal{A} does respond successfully to both sets of D 's challenges, this means that (z_1^1, \dots, z_n^1) differs in at most u entries from the correct answer

$$\text{ans}^1 \stackrel{\text{def}}{=} (\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^1 \rangle, \dots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^1 \rangle)$$

and also (z_1^2, \dots, z_n^2) differs in at most u entries from the correct answer

$$\text{ans}^2 \stackrel{\text{def}}{=} (\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^2 \rangle, \dots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^2 \rangle).$$

But then $(z_1^1, \dots, z_n^1) \oplus (z_1^2, \dots, z_n^2) = Z^\oplus$ differs in at most $2u$ entries from

$$\begin{aligned} \text{ans}^1 \oplus \text{ans}^2 &= (\langle \mathbf{s}_2, \mathbf{a}_1^1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^2 \rangle, \dots, \langle \mathbf{s}_2, \mathbf{a}_n^1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^2 \rangle) \\ &= (\langle \mathbf{s}_2, (\mathbf{a}_1^1 \oplus \mathbf{a}_1^2) \rangle, \dots, \langle \mathbf{s}_2, (\mathbf{a}_n^1 \oplus \mathbf{a}_n^2) \rangle) = \hat{Z}. \end{aligned}$$

We conclude that D outputs 1 in this case with probability at least δ^2 . This completes the proof of the claim. \blacksquare

We next consider the general case of $\varepsilon < 1/2$ and arbitrary n . We do not provide concrete bounds in this case, though such bounds may be derived easily from the proof that follows.

We first state the following lemma, based on the classic Johnson bound from coding theory. To get some intuition for the lemma, note that any two n -bit strings in a Hamming ball B of radius αn , for $\alpha < \frac{1}{2}$, are at distance at most $2\alpha n$ from each other. However, two strings chosen uniformly and independently from B will have much smaller distance with high probability. The following lemma shows that this is true for any distribution over B :

Lemma 6 (Distributional form of Johnson bound) *Let α, α^+ be constants such that $0 < \alpha < \alpha^+ < \frac{1}{2}$ and $\alpha^+ > \frac{1}{2} \cdot (1 - (1 - 2\alpha)^2)$. Then there exists a constant $C = C(\alpha, \alpha^+)$ such that for any n , and any distribution \mathcal{D} over a Hamming ball of radius $\alpha \cdot n$ in $\{0, 1\}^n$, we have:*

$$\Pr_{\Delta^1, \Delta^2 \leftarrow \mathcal{D}} [\text{wt}(\Delta^1 \oplus \Delta^2) < \alpha^+ n] \geq C.$$

Proof. Without loss of generality, assume the Hamming ball is centered at the origin. Let $\delta = 1 - 2\alpha^+$, and let $\gamma = 1 - 2\alpha$. Note that $\delta < \gamma^2$ by hypothesis. Set $c \stackrel{\text{def}}{=} \left\lceil \frac{1-\delta}{\gamma^2-\delta} + 1 \right\rceil$.

We show that for two vectors Δ^1, Δ^2 chosen independently according to distribution \mathcal{D} , we have $\text{wt}(\Delta^1 \oplus \Delta^2) < \alpha^+ n$ with (constant) probability at least $\frac{1}{c^2}$. Assume not. Then

$$\Pr[\Delta^1, \Delta^2 \leftarrow \mathcal{D} : \text{wt}(\Delta^1 \oplus \Delta^2) < \alpha^+ n] < \frac{1}{c^2}.$$

But then, by a union bound, $\Pr[\Delta^1, \dots, \Delta^c \leftarrow \mathcal{D} : \exists i \neq j \text{ s.t. } \text{wt}(\Delta^i \oplus \Delta^j) < \alpha^+ n] < \frac{1}{2}$. In particular, there exist c vectors $\Delta^1, \dots, \Delta^c$ in the support of \mathcal{D} whose pairwise distances are all at least $\alpha^+ n = \frac{1}{2} \cdot (1 - \delta) \cdot n$. Furthermore, each Δ^i has weight at most $\frac{1}{2} \cdot (1 - \gamma) \cdot n$ since it lies in the support of \mathcal{D} . However, the Johnson bound [23, 24] (our notation was chosen to be consistent with the formulation in [19, Theorem 1]), which gives bounds on the size of bounded-weight codes of certain minimum distance, shows that no such set $\{\Delta^i\}_{i=1}^c$ exists. \blacksquare

Proof (of Theorem 4). Fix a PPT adversary \mathcal{A} , and let $\delta_{\mathcal{A}} \stackrel{\text{def}}{=} \text{Adv}_{\mathcal{A}, \text{HB}^+}^{\text{active}}(\varepsilon, u, n)$. We construct a PPT adversary D attempting to distinguish whether it is given oracle access to $A_{\mathbf{s}, \varepsilon}$ or to U_{k+1} (as in Lemma 1). Relating the advantage of D to the advantage of \mathcal{A} gives the stated result.

The first three steps of our algorithm D are identical to those in the previous proof, and only the last step differs. For convenience we repeat all the steps here. D , given access to an oracle returning $(k+1)$ -bit strings (\mathbf{b}, \bar{z}) , proceeds as follows:

1. D chooses $\mathbf{s}_2 \in \{0, 1\}^k$ uniformly at random.
2. D runs the first phase of \mathcal{A} . To simulate a basic authentication step, D obtains a sample (\mathbf{b}, \bar{z}) from its oracle and sends \mathbf{b} as the initial message. \mathcal{A} replies with a challenge \mathbf{a} , and then D responds with $z = \bar{z} \oplus \langle \mathbf{s}_2, \mathbf{a} \rangle$.

3. When \mathcal{A} begins the second phase of its attack, it first sends an initial message $\mathbf{b}_1, \dots, \mathbf{b}_n$. In response, D chooses random $\mathbf{a}_1^1, \dots, \mathbf{a}_n^1 \in \{0, 1\}^k$, sends these challenges to \mathcal{A} , and records \mathcal{A} 's response z_1^1, \dots, z_n^1 . Then D rewinds \mathcal{A} , chooses random $\mathbf{a}_1^2, \dots, \mathbf{a}_n^2 \in \{0, 1\}^k$, sends these to \mathcal{A} , and records \mathcal{A} 's response z_1^2, \dots, z_n^2 .
4. Let $z_i^\oplus := z_i^1 \oplus z_i^2$ and set $Z^\oplus \stackrel{\text{def}}{=} (z_1^\oplus, \dots, z_n^\oplus)$. Let $\hat{\mathbf{a}}_i = \mathbf{a}_i^1 \oplus \mathbf{a}_i^2$ and $\hat{z}_i = \langle \mathbf{s}_2, \hat{\mathbf{a}}_i \rangle$, and set $\hat{Z} \stackrel{\text{def}}{=} (\hat{z}_1, \dots, \hat{z}_n)$. D outputs 1 iff Z^\oplus and \hat{Z} differ in strictly fewer than $u' = \varepsilon^{++}n$ entries, for some constant $\varepsilon^{++} < \frac{1}{2}$ to be fixed later.

Let us analyze the behavior of D :

Case 1: Say D 's oracle is U_{k+1} . Let A be an $n \times k$ matrix whose rows are the $\hat{\mathbf{a}}_i$. Viewing \mathbf{s}_2 and \hat{Z} as column vectors, we see that $\hat{Z} = A \cdot \mathbf{s}_2$. As in the proof of D 's oracle is U_{k+1} , the adversary \mathcal{A} has no information about \mathbf{s}_2 and therefore, from the point of view of the adversary, \hat{Z} is a random element in the column space of A . Furthermore, D outputs 1 exactly when Z^\oplus is within distance u' of \hat{Z} . We want to argue that this happens with low probability.

Translating the above to the language of coding theory, A defines a random, linear code of dimension k and length n , and \hat{Z} is a random codeword in this code. The following lemma implies that, with all but negligible probability over choice of A , any string Z^\oplus is far from a randomly-selected codeword with all but negligible probability.

Lemma 7 *Let $\alpha < \frac{1}{2}$ be a constant, and let \mathcal{C} be a random $[n, k]$ -code generated by an $n \times k$ binary matrix A with entries chosen uniformly at random. With probability $2^{-\Omega(n)}$ over choice of A , there does not exist a Hamming ball of radius $\alpha \cdot n$ that contains a $2^{-\Omega(k)}$ fraction of the codewords in \mathcal{C} .*

We defer the proof of Lemma 7 until we finish the proof of the theorem.

Case 2: Say D 's oracle is $A_{\mathbf{s}_1, \varepsilon}$ for randomly-chosen \mathbf{s}_1 . Exactly as in the proof of Claim 5, we have that \mathcal{A} responds correctly to both sets of queries $\mathbf{a}_1^1, \dots, \mathbf{a}_n^1$ and $\mathbf{a}_1^2, \dots, \mathbf{a}_n^2$ with probability at least $\delta_{\mathcal{A}}^2$. We show next that conditioned on both challenges being answered successfully (and for appropriate choice of ε^{++}), Z^\oplus differs from \hat{Z} in fewer than u' entries with *constant* probability. Putting everything together, we conclude that D outputs 1 in this case with probability $\Omega(\delta_{\mathcal{A}}^2)$. It follows from Lemma 1 that $\delta_{\mathcal{A}}$ must be negligible.

We now prove the above claim regarding the probability that Z^\oplus differs from \hat{Z} in fewer than u' entries. Set ε^{++} so that $\frac{1}{2} > \varepsilon^{++} > \frac{1}{2} \cdot (1 - (1 - 2\varepsilon^+)^2)$. Fixing all the randomness used in the simulation of the first phase of \mathcal{A} defines a function $f_{\mathcal{A}}$ from queries $\mathbf{a}_1, \dots, \mathbf{a}_n$ to vectors (z_1, \dots, z_n) given by the response function of \mathcal{A} in the second phase. Define the function f_{correct} that returns the “correct” answers for a particular query; i.e.,

$$f_{\text{correct}}(\mathbf{a}_1, \dots, \mathbf{a}_n) \stackrel{\text{def}}{=} (\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1 \rangle, \dots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n \rangle)$$

(recall that $\mathbf{b}_1, \dots, \mathbf{b}_n$ are the vectors sent by \mathcal{A} in the first round). Define

$$\Delta(\mathbf{a}_1, \dots, \mathbf{a}_n) \stackrel{\text{def}}{=} f_{\mathcal{A}}(\mathbf{a}_1, \dots, \mathbf{a}_n) \oplus f_{\text{correct}}(\mathbf{a}_1, \dots, \mathbf{a}_n),$$

and say a query $\mathbf{a}_1, \dots, \mathbf{a}_n$ is *good* if $\text{wt}(\Delta(\mathbf{a}_1, \dots, \mathbf{a}_n)) \leq u$. A query $\mathbf{a}_1, \dots, \mathbf{a}_n$ is good if \mathcal{A} 's response is within distance u of the “correct” response, that is, \mathcal{A} successfully impersonates the tag in response to such a query.

Let \mathcal{D} denote the distribution over $\Delta(\mathbf{a}_1, \dots, \mathbf{a}_n)$ induced by a uniform choice of a good query $\mathbf{a}_1, \dots, \mathbf{a}_n$ (we assume at least one good query exists since we are only interested in analyzing this case). To see how this maps on to the reduction being analyzed above, note that conditioning on the event that \mathcal{A} successfully responds to queries $\mathbf{a}_1^1, \dots, \mathbf{a}_n^1$ and $\mathbf{a}_1^2, \dots, \mathbf{a}_n^2$ is equivalent to choosing these two queries uniformly from the set of good queries. Setting $\Delta^1 \stackrel{\text{def}}{=} \Delta(\mathbf{a}_1^1, \dots, \mathbf{a}_n^1)$ and Δ^2 analogously, we have

$$\begin{aligned} \Delta^1 \oplus \Delta^2 &= f_{\mathcal{A}}(\mathbf{a}_1^1, \dots, \mathbf{a}_n^1) \oplus f_{\text{correct}}(\mathbf{a}_1^1, \dots, \mathbf{a}_n^1) \oplus f_{\mathcal{A}}(\mathbf{a}_1^2, \dots, \mathbf{a}_n^2) \oplus f_{\text{correct}}(\mathbf{a}_1^2, \dots, \mathbf{a}_n^2) \\ &= Z^\oplus \oplus f_{\text{correct}}(\mathbf{a}_1^1, \dots, \mathbf{a}_n^1) \oplus f_{\text{correct}}(\mathbf{a}_1^2, \dots, \mathbf{a}_n^2). \end{aligned}$$

Furthermore,

$$\begin{aligned} &f_{\text{correct}}(\mathbf{a}_1^1, \dots, \mathbf{a}_n^1) \oplus f_{\text{correct}}(\mathbf{a}_1^2, \dots, \mathbf{a}_n^2) \\ &= (\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^1 \rangle, \dots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^1 \rangle) + (\langle \mathbf{s}_1, \mathbf{b}_1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^2 \rangle, \dots, \langle \mathbf{s}_1, \mathbf{b}_n \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^2 \rangle) \\ &= (\langle \mathbf{s}_2, \mathbf{a}_1^1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_1^2 \rangle, \dots, \langle \mathbf{s}_2, \mathbf{a}_n^1 \rangle \oplus \langle \mathbf{s}_2, \mathbf{a}_n^2 \rangle) \\ &= (\langle \mathbf{s}_2, (\mathbf{a}_1^1 \oplus \mathbf{a}_1^2) \rangle, \dots, \langle \mathbf{s}_2, (\mathbf{a}_n^1 \oplus \mathbf{a}_n^2) \rangle) = \hat{Z}. \end{aligned}$$

So $\Delta^1 \oplus \Delta^2 = Z^\oplus \oplus \hat{Z}$, and we see that Z^\oplus and \hat{Z} differ in fewer than u' entries exactly when $\text{wt}(\Delta^1 \oplus \Delta^2) < u'$.

Now, by definition of a good query, each vector in the support of \mathcal{D} has weight at most $u = \varepsilon^+ n$. By Lemma 6, with constant probability over Δ^1, Δ^2 generated independently according to \mathcal{D} , we have $\text{wt}(\Delta^1 \oplus \Delta^2) < u'$ (note that u' and u were chosen to satisfy the conditions of the lemma). This concludes the proof of Theorem 4. \blacksquare

We conclude with the deferred proof of Lemma 7.

Proof (of Lemma 7). Suppose there is a ball B of radius αn that contains K codewords of \mathcal{C} . We first show that there is a ball B^+ of slightly larger radius, centered at a codeword, that captures at least γK points of \mathcal{C} (for some constant γ). We then show that, for a random linear code, any such B^+ typically captures only an exponentially small (in k) fraction of the codewords of \mathcal{C} .

First, choose a constant $\alpha^+ < \frac{1}{2}$ such that $(\frac{1}{2} - \alpha^+) < (\frac{1}{2} - \alpha)^2$. Let $\gamma = C(\alpha, \alpha^+)$, as defined in Lemma 6. We claim that there exists a codeword $x^* = Ar$ in B such that the ball B^+ of radius $\alpha^+ n$ centered at x^* contains at least $\gamma \cdot K$ codewords. Assume for contradiction that no such x^* exists. Then for a *random* pair of codewords x, y contained in B , the probability that x, y differ in fewer than $\alpha^+ n$ coordinates is less than γ . But this contradicts Lemma 6.

We now show that γK cannot be too large. We use two facts about random linear codes: (1) the distance distribution is the same for every codeword x^* in the code, so we may take $x^* = 0^n$ without loss of generality; and (2) the locations of codewords are pairwise independent. We can thus apply the Chebyshev bound to bound the probability that $\gamma \cdot K$ codewords lie within the ball B^+ of radius $\alpha^+ n$ centered at 0^n . Let X denote the fraction of codewords in B^+ . The expectation of X is $\frac{|B^+|}{2^n}$ and its variance is at most $\frac{|B^+|}{2^n \cdot (2^k - 1)}$. The probability that X exceeds its expectation by more than, say, $2^{-k/2}$, is at most $\frac{|B^+|}{2^n} = 2^{-\Omega(n)}$. The total number of points in any ball B of radius αn is thus at most $\frac{1}{\gamma 2^{k/2}}$, with probability $1 - 2^{-\Omega(n)}$ over the choice of A . \blacksquare

5 Conclusions and Open Questions

The main technical results of this paper are the first rigorous proofs of (1) security of the HB^+ protocol against active attacks, even under parallel and concurrent executions; and (2) “hardness amplification” for the HB and HB^+ protocols as the number of iterations of the basic authentication step increases. Our proofs are also the first to explicitly take into account the non-zero completeness error and the impact this has on the security of the protocol as a whole.

We believe our proofs are remarkably simple, and view this as an additional contribution of this work (rather than as a drawback!). Indeed, we expect there will be further applications of Lemma 1 to the analysis of other cryptographic constructions based on the LPN problem, and hope this paper inspires and aids others in exploring such applications.

It would be very interesting to see an efficient protocol based on the LPN problem that is provably resistant to man-in-the-middle attacks such as those of Gilbert et al. [12]. Though much recent work [5, 9, 13, 14, 32] (subsequent to the results described here) addresses this problem, none of these provides a provably-secure solution to the problem in its full generality. It would also be useful to improve the concrete security reductions obtained here, or to propose new protocols with tighter security reductions. As one possible approach toward this goal, one can imagine changing the HB/HB^+ protocols so that the tag always introduces *at most* $\varepsilon \cdot n$ errors, rather than introducing errors in each of the n iterations with independent probability ε .⁶ (A related idea, in a different context, was explored in [3]; their analysis does not seem to apply to our setting.) This would give protocols with perfect completeness, and would improve the concrete security bounds as well since the upper bound u could be set to exactly $\varepsilon \cdot n$. On the other hand it is not clear what can be said of the hardness of the natural variant of the LPN problem such protocols would be based on.

References

- [1] M. Bellare, R. Impagliazzo, and M. Naor. Does Parallel Repetition Lower the Error in Computationally-Sound Protocols? *38th IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 374–383, 1997.
- [2] E.R. Berlekamp, R.J. McEliece, and H.C.A. van Tilborg. On the Inherent Intractability of Certain Coding Problems. *IEEE Trans. Info. Theory* 24: 384–386, 1978.
- [3] A. Blum, M. Furst, M. Kearns, and R. Lipton. Cryptographic Primitives Based on Hard Learning Problems. *Adv. in Cryptology — Crypto '93*, LNCS vol. 773, Springer-Verlag, pp. 278–291, 1994.
- [4] A. Blum, A. Kalai, and H. Wasserman. Noise-Tolerant Learning, the Parity Problem, and the Statistical Query Model. *J. ACM* 50(4): 506–519, 2003.
- [5] J. Bringer, H. Chabanne, and E. Dottax. HB^{++} : A Lightweight Authentication Protocol Secure Against Some Attacks. In P. Georgiadis, J. Lopez, S. Gritzalis, and G. Marias, editors, *Proceedings of SecPerU 2006*, 28–33, IEEE Computer Society Press, 2006.

⁶Note that introducing *exactly* $\varepsilon \cdot n$ errors in the n iterations is insecure.

- [6] R. Canetti, S. Halevi, and M. Steiner. Hardness Amplification of Weakly Verifiable Puzzles. *2nd Theory of Cryptography Conference (TCC 2005)*, LNCS vol. 3378, Springer-Verlag, pp. 17–33, 2005.
- [7] R. Canetti, J. Kilian, E. Petrank, and A. Rosen. Black-Box Concurrent Zero-Knowledge Requires (Almost) Logarithmically Many Rounds. *SIAM J. Computing* 32(1): 1–47, 2002.
- [8] F. Chabaud. On the Security of Some Cryptosystems Based on Error-Correcting Codes. *Adv. in Cryptology — Eurocrypt '94*, LNCS vol. 950, Springer-Verlag, pp. 131–139, 1995.
- [9] D.N. Duc and K. Kim. Securing HB^+ Against GRS Man-in-the-Middle Attack. In *Institute of Electronics, Information and Communication Engineers, Symposium on Cryptography and Information Security*, Jan. 23–26, 2007.
- [10] U. Feige and A. Shamir. Witness Indistinguishability and Witness Hiding Protocols. *22nd ACM Symposium on Theory of Computing*, ACM, pp. 416–426, 1990.
- [11] M. Fossorier, M.J. Mihaljevic, H. Imai, Y. Cui, and K. Matsuura. An Algorithm for Solving the LPN Problem and Its Application to Security Evaluation of the HB Protocols for RFID Authentication. *Proc. INDOCRYPT 2006*, p. 48-62.
- [12] H. Gilbert, M. Robshaw, and H. Silbert. An Active Attack against HB^+ : A Provably Secure Lightweight Authentication Protocol. *IEEE Electronics Letters* 41(21): 1169–1170, 2005.
- [13] H. Gilbert, M.J.B. Robshaw, and Y. Seurin. Good Variants of HB^+ are Hard to Find. In *Proceedings of Financial Crypto 2008*, to appear.
- [14] H. Gilbert, M.J.B. Robshaw, and Y. Seurin. $HB^\#$: Increasing the Security and Efficiency of HB^+ . In *Proceedings of Eurocrypt 2008*, to appear. Full version available from <http://eprint.iacr.org/2008/028>
- [15] O. Goldreich. *Modern Cryptography, Probabilistic Proofs, and Pseudorandomness*. Springer-Verlag, 1998.
- [16] O. Goldreich and H. Krawczyk. On the Composition of Zero-Knowledge Proof Systems. *SIAM J. Computing* 25(1): 169–192, 1996.
- [17] O. Goldreich, N. Nisan, and A. Wigderson. On Yao’s XOR-Lemma. Available at <http://eccc.uni-trier.de/eccc-reports/1995/TR95-050/>.
- [18] O. Goldreich and Y. Oren. Definitions and Properties of Zero-Knowledge Proof Systems. *J. Cryptology* 7(1): 1–32, 1994.
- [19] V. Guruswami and M. Sudan. Extensions to the Johnson Bound. Unpublished manuscript, 2001. Available at <http://citeseer.ist.psu.edu/guruswami01extensions.html>.
- [20] J. Håstad. Some Optimal Inapproximability Results. *J. ACM* 48(4): 798–859, 2001.
- [21] N. Hopper and M. Blum. A Secure Human-Computer Authentication Scheme. Technical Report CMU-CS-00-139, Carnegie Mellon University, 2000.

- [22] N. Hopper and M. Blum. Secure Human Identification Protocols. *Adv. in Cryptology — Asia-crypt 2001*, LNCS vol. 2248, pp. 52–66, 2001.
- [23] S.M. Johnson. A New Upper Bound for Error-Correcting Codes. *IRE Trans. Information Theory* 8(3): 203–207, 1962.
- [24] S.M. Johnson. Improved Asymptotic Bounds for Error-Correcting Codes. *IEEE Trans. Info. Theory* 9(3): 198–205, 1963.
- [25] A. Juels and S. Weis. Authenticating Pervasive Devices with Human Protocols. *Adv. in Cryptology — Crypto 2005*, LNCS vol. 3621, Springer-Verlag, pp. 293–308, 2005. Updated version available at: <http://www.rsasecurity.com/rsalabs/staff/bios/ajuels/publications/pdfs/lpn.pdf>.
- [26] J. Katz and J.-S. Shin. Parallel and Concurrent Security of the HB and HB⁺ Protocols. *Adv. in Cryptology — Eurocrypt 2006*.
- [27] J. Katz and A. Smith. Analyzing the HB and HB⁺ Protocols in the “Large Error” Case. Available at <http://eprint.iacr.org/2006/326>.
- [28] M. Kearns. Efficient Noise-Tolerant Learning from Statistical Queries. *J. ACM* 45(6): 983–1006, 1998.
- [29] Z. Kfir and A. Wool. Picking Virtual Pockets using Relay Attacks on Contactless Smartcard Systems. Available at <http://eprint.iacr.org/2005/052>.
- [30] I. Kirschenbaum and A. Wool. How to Build a Low-Cost, Extended-Range RFID Skimmer. Available at <http://eprint.iacr.org/2006/054>.
- [31] E. Levieil and P.-A. Fouque. An Improved LPN Algorithm. *Security and Cryptography for Networks (SCN 2006)*, LNCS vol. 4116, pp. 348–359, 2006.
- [32] J. Munilla and A. Peinado. HB-MP: A Further Step in the HB-family of Lightweight Authentication Protocols. *Computer Networks* 51, 2262–2267, 2007.
- [33] R. Pass and M. Venkatasubramanian. An Efficient Parallel Repetition Theorem for Arthur-Merlin Games. *39th ACM Symposium on Theory of Computing*, ACM, pp. 420–429, 2007.
- [34] R. Raz. A Parallel Repetition Theorem. *SIAM J. Computing* 27(3): 763–803, 1998.
- [35] O. Regev. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography. *37th ACM Symposium on Theory of Computing*, ACM, pp. 84–93, 2005.
- [36] A. C.-C. Yao. Theory and Applications of Trapdoor Functions. *23rd IEEE Symposium on Foundations of Computer Science*, IEEE, pp. 80–91, 1982.