HB^N: An HB-like protocol secure against man-in-the-middle attacks

Carl Bosley* Kristiyan Haralambiev † Antonio Nicolosi ‡

August 5, 2011

Abstract

We construct a simple authentication protocol whose security is based solely on the problem of Learning Parity with Noise (LPN) that is secure against Man-in-the-Middle attacks. Our protocol is suitable for RFID devices, whose limited circuit size and power constraints rule out the use of more heavyweight operations such as modular exponentiation. The protocol is extremely simple: both parties compute a noisy bilinear function of their inputs. The proof, however, is quite technical, and we believe that some of our technical tools may be of independent interest.

1 Introduction

Motivation. Many cryptographic tasks originate from the necessity to reproduce in cyber space security properties that exist in the physical world. Examples in point include digital signatures (non-repudiation) or public-key encryption (drop-boxes). Among the basic cryptographic goals, authentication has the potential to straddle the physical and cyber world, and enable authentication cryptographically strong authentication of physical things.

For moderately powerful devices like smartphones, or even battery-operated sensors, existing authentication protocols often suffice. Computationally weak devices such as RFID devices and batteryless contactless smartcards, however, require more lightweight, dedicated solutions.

RFID devices are quickly becoming popular in many applications. They are used throughout the supply chain for inventory management. RFID can be used to replace physical keys for access control. Banking and financial institutions have also started to embrace them for account management. Mass transit authorities in several metropolitan areas have taken to used them to replace tokens; similarly, RFID-mediated access to toll roads is the norm all over the world.

RFID devices can do all this, silently. Unfortunately, this silence leaves them vulnerable to stealth queries from malicious entities. This introduces an array of security risks, including unauthorized access, fraudulent account usage, as well as privacy risks, such as stealth tracking.

Learning parity with noise. The LPN problem was introduced in the machine learning community by Angluin and Laird [AL87]. It soon became notorious for having no efficient noise-tolerant algorithm. It was proven by Kearns [Kea93] that the class of noisy parity concepts (LPN) is not learn-able within the statistical query model. Work on LPN-based protocols began with the HB protocol of Hopper and Blum [HB01], which was later proven to be secure against Passive attacks assuming the hardness of LPN.

HB-type protocols. The original motivation for the HB protocol was to enable unaided human authentication: the goal was for the protocol to be simple enough to be carried out without the help of a computational device. Subsequent work has found that the key sizes and error rates required to ensure security may be too large for humans to employ with ease comparable to, say, password-based authentication. Nevertheless, as noted by Juels and Weis [JW05], HB-type protocols are lightweight enough to be potentially applicable in the RFID setting. Indeed, constraints on power consumption and circuit size (1,000–4,000 transistors) for RFID devices makes it problematic to deploy conventional cryptographic algorithms like AES or modular exponentiation on these devices; HB-type protocols, on the other hand, have very simple circuit representations. For example, the interaction between the prover, or tag $\mathcal{T}$, and the verifier, or reader $\mathcal{R}$, in the HB protocol consists of two messages: first, $\mathcal{R}$ sends a random challenge $\mathbf{a} \in \mathbb{F}_2^n$. Next, $\mathcal{T}$ samples $e \in \mathbb{F}_2$ according to the Bernoulli distribution Ber_ε (i.e. $\Pr[e = 1] = \varepsilon$). $\mathcal{T}$ sends $z = \mathbf{a}^\top \mathbf{x} + e$ to $\mathcal{R}$, where $\mathbf{x} \in \mathbb{F}_2^n$ is a key shared between $\mathcal{T}$ and $\mathcal{R}$. $\mathcal{R}$ accepts if $z = \mathbf{a}^\top \mathbf{x}$. The basic protocol has soundness $\frac{1}{2}$ and completeness $1 - \varepsilon$, but this can be improved via sequential or parallel composition (*cf.* Section 2.3).

*Dept. of Computer Science, Stevens Institute. bosley@cs.stevens.edu.

†Dept. of Computer Science, New York University. haralambiev@cs.nyu.edu.

‡Dept. of Computer Science, Stevens Institute. nicolosi@cs.stevens.edu

In [JW05], Juels and Weis also introduced HB^+, which was shown to be secure in a slightly stronger security model (known as Active security) than the original HB protocol. Gilbert, Robshaw, and Seurin ([GRS05]) showed that HB^+ is vulnerable to a man-in-the-middle attack. A number of variants of HB^+ were proposed to remedy this defect, including HB^{++} [BCD06], HB^* [DK08], HB-MP [MP07], HB-MP' [LMM08], and Trusted-HB [BC08]. However, all of these were proven insecure. Gilbert, Robshaw, and Seurin ([GRS08a]) extended their attack on HB^+ to break HB^{++}, HB^*, HB-MP, HB-MP', and Frumkin and Shamir [FS09] showed that Trusted-HB is insecure.

Gilbert, Robshaw, and Seurin [GRS08b] introduced $\mathsf{HB}^\#$, which was secure against the same attack that succeeded against HB^+. However, Oaufi et al [OOV08] presented an Man-in-the-Middle attack on $\mathsf{HB}^\#$.

Katz, Shin, and Smith [KSS10] provided the first proof of security for HB and HB^+ for any error rate $\varepsilon < 1/2$, via black box reductions. However, for HB^+ the reduction used rewinding, so that it achieved active security $\sqrt{\varepsilon}$ assuming LPN is hard for noise rate ε.

Pietrzak then introduced Subspace LWE [Pie10], a more flexible formulation of LPN that is nevertheless equivalent to LPN. In a major advance, Kiltz et al. [KPC$^+$11] built on Subspace LWE [Pie10] to construct a two-round Active-secure protocol, as well as two secure MACs, which imply two-round Man-in-the-Middle-secure protocols. However, both Man-in-the-Middle-secure constructions require the use of an (almost) Pairwise Independent Permutation on approximately $O(n^2)$ bits. Furthermore, the first MAC's security reduction is loose, achieving security $\sqrt{\varepsilon}$, while the second construction is much more complicated and requires a longer key.

1.1 Our Contribution

Our protocol, like the original HB protocol, is extremely simple: instead of computing a noisy linear function $\mathbf{a}^\top\mathbf{x} + e$, the parties compute a noisy *bilinear* function $\mathbf{a}^\top\mathbf{X}\mathbf{b} + e$ of their joint inputs $\mathbf{a}, \mathbf{b}$. As described in Section 3, this can be done in either 2 or 3 rounds.

However, the Man-in-the-Middle security proof is quite technically involved, particularly in the understanding of the noise distributions. We develop some technical tools, including the LSN (Learning Subspaces with Noise) problem, which we believe will be of independent interest.

Another new technique that may be useful elsewhere is the probabilistic scheme for the Verifier, which was not present in earlier protocols. Our Verifier simply adds noise mirroring the noise from the Tag. This eliminates a major difficulty in earlier protocols, for which deterministic verification was often exploited to design attacks.

Interestingly, although its simplicity was obscured by notation, a similar bilinear protocol was proposed in Section 5.2 of [KPC$^+$11] and proven to be Man-in-the-Middle-secure. However, [KPC$^+$11] used more heavyweight tools such as Waters' technique for converting a selectively secure MAC to a fully secure MAC.

1.2 Outline

We describe LPN, HB and HB^+, and the Passive, Active, and Man-in-the-Middle security models in Section 2. In Section 3, we describe the HB^N protocol family. In order to analyze the security of HB^N, we first need to develop new tools for precisely manipulating error distributions, including the LSN (Learning Subspaces with Noise) problem, which we present in Section 4. Finally, in Section 5, we prove that HB^N is secure against Man-in-the-Middle attacks.

2 Preliminaries

2.1 Notation

We write $x \xleftarrow{\$} X$ to denote the process of assigning a value sampled from the distribution X to the variable x. If S is a finite set, we write $s \xleftarrow{\$} S$ to denote assignment to s of a value sampled from the uniform distribution on S. We use $[n]$ to denote the set $\{1, 2, \dots, n\}$. Vice versa, we will abuse set-notation to identify a distribution X with its support; for example, we write $x \in X$ to denote that x is in the support of X. If $\mathcal{A}$ is a probabilistic algorithm, we let $\mathcal{A}(x)$ denote the output distribution of $\mathcal{A}$ on input x, and write $y \xleftarrow{\$} \mathcal{A}(x)$ to denote the process of running algorithm $\mathcal{A}$ on input x and assigning its output to y. We write:

$$\Pr[x_1 \xleftarrow{\$} X_1, x_2 \xleftarrow{\$} X_2(x_1), \dots, x_n \xleftarrow{\$} X_n(x_1, \dots, x_{n-1}) : \phi(x_1, \dots, x_n)]$$

to denote the probability that the predicate $\phi(x_1, \dots, x_n)$ is true, when for all $i \in [n]$, x_i is drawn from distribution X_i, possibly depending on the values drawn for $x_1, \dots, x_{i-1}$. When $n = 1$, $\hat{x} \in X_1$, and $\phi(x_1)$ is of the form "$x_1 = \hat{x}_1$", we use the shorthand $\Pr[\hat{x}_1 \xleftarrow{\$} X]$ to denote $\Pr[x_1 \xleftarrow{\$} X_1 : x_1 = \hat{x}_1]$. For two probability distributions X_1, X_2, we write $X_1 \equiv X_2$ if and only if $\forall \hat{x} \in X_1 \cup X_2, \Pr[\hat{x} \xleftarrow{\$} X_1] = \Pr[\hat{x} \xleftarrow{\$} X_2]$.

Let $\mathbb{F}_q$ represent the finite field with q elements. We denote the uniform distribution over $\mathbb{F}_2^n$ by $\mathsf{U}_{n\times n}$, and the Bernoulli distribution with bias ε by Ber_ε. (Recall that Ber_ε is the distribution over $\mathbb{F}_2$ with $\Pr[1 \xleftarrow{\$} \mathsf{Ber}_\varepsilon] = \varepsilon$, $\Pr[0 \xleftarrow{\$} \mathsf{Ber}_\varepsilon] = 1-\varepsilon$.) We use the binary operator $\oplus\colon \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$ to represent finite field addition, and for $b \in \mathbb{F}_2$, we let $\overline{b} = 1 \oplus b$ be the complement of b. For an event S, $\overline{S}$ represents its complement, the event that S does not occur.

We denote column vectors by lower-case bold letters such as $\mathbf{x}$, and matrices by upper-case bold letters such as $\mathbf{X}$. We denote the transpose of $\mathbf{X}$ by $\mathbf{X}^\top$. For a matrix $\mathbf{A} \in \mathbb{F}_2^{m\times n}$, $\mathsf{rank}(\mathbf{A})$ denotes the rank of $\mathbf{A}$. $\ker(\mathbf{A}) = \{\mathbf{x} : \mathbf{A}\mathbf{x} = 0\}$ denotes the kernel of $\mathbf{X}$, the set of all vectors orthogonal to $\mathbf{A}$, and $\mathsf{Im}(\mathbf{A}) = \{\mathbf{y} : \exists \mathbf{x} \text{ s.t. } \mathbf{A}\mathbf{x} = \mathbf{y}\}$ denotes the image of $\mathbf{A}$, the set of all linear combinations of columns of $\mathbf{A}$. $\mathbf{I}_n$ denotes the $n \times n$ identity matrix.

We will often consider column vectors $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^\ell$ as matrices in $\mathbb{F}_2^{\ell\times 1}$. Considering $\mathbf{x}, \mathbf{y}$ as matrices allows us to extend operations on matrices to vectors. For example, we can form the outer product $\mathbf{x}\mathbf{y}^\top \in \mathbb{F}_2^{\ell\times\ell}$, and form the kernel $\ker(\mathbf{x})$. The dot product of two column vectors $\mathbf{x}, \mathbf{y}$ can be written as the matrix multiplication $\mathbf{x}^\top\mathbf{y}$. For a vector $\mathbf{x}$, we denote the scalar i-th element of $\mathbf{x}$ by $\mathbf{x}_i$. $\mathbf{0}^n$ denotes the all-zero column vector of length n. $\mathbf{e}^{(i,\ell)} \in \mathbb{F}_2^\ell$ denotes the i-th vector of the canonical basis, for which $\mathbf{e}_i^{(i)} = 1$, and $\mathbf{e}_j^{(i)} = 0$ for $j \neq i$. In practice, when the dimension can be determined from context, we drop it, letting $\mathbf{e}^{(i)} = \mathbf{e}^{(i,\ell)}$. For a vector $\mathbf{x}$, let $|\mathbf{x}|$ denote the number of nonzero entries of $\mathbf{x}$.

We denote an arbitrary polynomial function of n by $\mathsf{poly}(n)$. We write $f = \mathsf{negl}$ to mean that f is negligible as a function of n, that is, $f = o(n^{-c})$ for any constant $c > 0$.

2.2 Learning Parity with Noise (LPN)

Roughly speaking, the problem of Learning Parity with Noise amounts to distinguishing two distributions over $\mathbb{F}_2^n \times \mathbb{F}_2$: the uniform distribution and the LPN *distribution*. For a random secret vector $\mathbf{x} \in \mathbb{F}_2^n$, the LPN distribution is in turn defined in terms of its sampling algorithm $\mathsf{LPN}_\varepsilon^\mathbf{x}$, shown in Algorithm 2.2. Algorithm $\mathsf{LPN}_\varepsilon^\mathbf{x}$ is initialized with a uniform secret vector $\mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n$. Thereafter, whenever an LPN sample is requested, the algorithm chooses random $\mathbf{a} \xleftarrow{\$} \mathbb{F}_2^n$ and $e \xleftarrow{\$} \mathsf{Ber}_\varepsilon$ and outputs $(\mathbf{a}, b)$, where $b = \mathbf{a}^\top\mathbf{x} \oplus e$. For $\varepsilon = \frac{1}{2}$, LPN becomes the uniform distribution.

```
function $\mathsf{LPN}_\varepsilon^\mathbf{x}$
    $\mathbf{a} \xleftarrow{\$} \mathbb{F}_2^n$
    $e \xleftarrow{\$} \mathsf{Ber}_\varepsilon$
    $b = \mathbf{a}^\top\mathbf{x} + e$
    return $(\mathbf{a}, b)$
```

Algorithm 1: LPN

We will use the decisional version of the LPN hardness assumption, which is defined using an indistinguishability game. It has been shown [KSS10] that hardness of the decisional version is equivalent (up to polynomial factors) to hardness of recovering the entire key. The decisional variant of LPN is hard if it is difficult to distinguish between an oracle with distribution $\mathsf{LPN}_\varepsilon^\mathbf{x}$ versus an oracle with a random distribution $\mathsf{U}_n \times \mathsf{U}_1$, which (by Corollary 8) can be represented as $\mathsf{LPN}_{1/2}^\mathbf{x}$. More formally, the advantage of an algorithm $\mathcal{A}$ against LPN for a given (ε, n) is defined using a game in which the adversary attempts to guess which oracle was selected:

Definition 1. *The decisional* LPN *assumption states that for all efficient adversaries* $\mathcal{A}$*,* $\mathsf{Adv}_\mathcal{A}^\mathsf{LPN}(\varepsilon, n) \leq \varepsilon_\mathsf{LPN} = \mathsf{negl}$*, where* $\mathsf{Adv}_\mathcal{A}^\mathsf{LPN}(\varepsilon, n)$ *is defined as*

$$\mathsf{Adv}_\mathcal{A}^\mathsf{LPN}(\varepsilon, n) = \left| \Pr\left[\begin{array}{l} \mathbf{x} \xleftarrow{\$} \mathbb{F}_2^n, b \xleftarrow{\$} \mathbb{F}_2, \\ \mathcal{O}_b = \begin{cases} \mathsf{LPN}_{1/2}^\mathbf{x} & \textit{if } b = 0 \\ \mathsf{LPN}_\varepsilon^\mathbf{x}, & \textit{if } b = 1 \end{cases}, \\ \hat{a} \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_b}(1^n) \end{array} : \hat{a} = b \right] - \frac{1}{2} \right| \qquad (1)$$

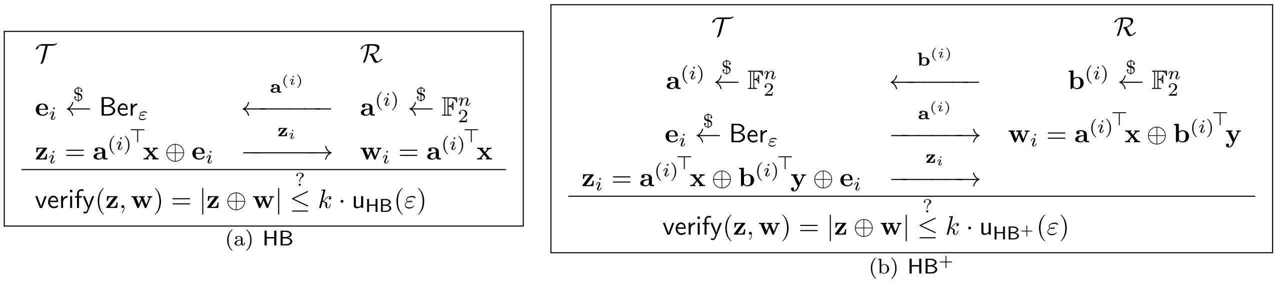
2.3 HB and HB$^+$ protocols

The HB, HB$^+$ protocols consist of $k = \mathsf{poly}(n)$ iterations of what is known as a "basic authentication step". The protocols are executed by two parties: the tag $\mathcal{T}$, who wishes to authenticate, and the reader $\mathcal{R}$, who verifies the tag. [1] The key for HB is a vector $\mathbf{x}$ of length n, where n is the security parameter. For HB$^+$, the key consists of two vectors $\mathbf{x}, \mathbf{y}$ of length n. For $i \in [k]$, $\mathbf{a}^{(i)}, \mathbf{b}^{(i)} \in \mathbb{F}_2^n$ are column vectors used in the execution. In HB, as shown in Figure 1(a), a tag $\mathcal{T}$ and a reader $\mathcal{R}$ share a random secret key $\mathbf{x} \in \mathbb{F}_2^n$. In the i-th round authentication step, the reader sends a random challenge $\mathbf{a}^{(i)} \in \mathbb{F}_2^n$

[1] $\mathcal{T}$ is also known as the prover $\mathcal{P}$, and $\mathcal{R}$ as the verifier $\mathcal{V}$.

to the tag, and the tag replies with $\mathbf{z}_i = \mathbf{a}^{(i)^\top}\mathbf{x} \oplus \mathbf{e}_i$, where $\mathbf{e}_i \xleftarrow{\$} \mathsf{Ber}_\varepsilon$. HB^+ adds a second secret $\mathbf{y}$ and a third round, as shown in Figure 1(b).

In both HB and HB^+, at the end of k rounds, $\mathcal{R}$ checks to see what fraction of answers $\mathbf{z}_i$ were correct. If more than $k \cdot \mathsf{u}(\varepsilon)$ are correct, for $\mathsf{u}(\varepsilon)$ some function of ε, then $\mathsf{verify}(\mathbf{z}, \mathbf{w})$ returns true, and the reader accepts. Otherwise, the reader rejects. k and $\mathsf{u}(\varepsilon)$ should be set high enough to allow the honest tag to authenticate w.h.p., but low enough that a malicious third party should not be able to authenticate by randomly guessing. In particular, as noted in [KSS10], for both HB and HB^+, $\mathsf{u}(\varepsilon) = (1+\delta)\varepsilon$ suffices to achieve completeness error negligible in the security parameter, for any positive constant δ.


$\mathcal{T}$
$\mathcal{R}$
$\mathbf{e}_i \xleftarrow{\$} \mathsf{Ber}_\varepsilon$
$\xleftarrow{\mathbf{a}^{(i)}}$
$\mathbf{a}^{(i)} \xleftarrow{\$} \mathbb{F}_2^n$
$\mathbf{z}_i = \mathbf{a}^{(i)^\top}\mathbf{x} \oplus \mathbf{e}_i$
$\xrightarrow{\mathbf{z}_i}$
$\mathbf{w}_i = \mathbf{a}^{(i)^\top}\mathbf{x}$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}}(\varepsilon)$
$\mathcal{T}$
$\mathcal{R}$
$\mathbf{a}^{(i)} \xleftarrow{\$} \mathbb{F}_2^n$
$\xleftarrow{\mathbf{b}^{(i)}}$
$\mathbf{b}^{(i)} \xleftarrow{\$} \mathbb{F}_2^n$
$\mathbf{e}_i \xleftarrow{\$} \mathsf{Ber}_\varepsilon$
$\xrightarrow{\mathbf{a}^{(i)}}$
$\mathbf{w}_i = \mathbf{a}^{(i)^\top}\mathbf{x} \oplus \mathbf{b}^{(i)^\top}\mathbf{y}$
$\mathbf{z}_i = \mathbf{a}^{(i)^\top}\mathbf{x} \oplus \mathbf{b}^{(i)^\top}\mathbf{y} \oplus \mathbf{e}_i$
$\xrightarrow{\mathbf{z}_i}$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}^+}(\varepsilon)$

(a) HB

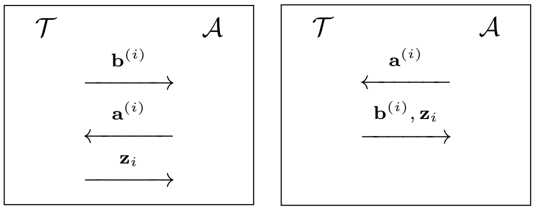
(b) HB$^+$

2.4 Security Models

In this subsection we present several natural security models that have been used for authentication and for HB-type protocols in particular. The more general models are $\mathsf{Passive}$, Active, and $\mathsf{Man\text{-}in\text{-}the\text{-}Middle}$. Additionally, several works have used an intermediate model, $\mathsf{GRS\text{-}MIM}$, which is stronger than Active yet weaker than the full $\mathsf{Man\text{-}in\text{-}the\text{-}Middle}$ model.

$\mathsf{Passive}$ Model: In Phase I, the attacker can only observe the interactions between $\mathcal{T}$ and $\mathcal{R}$.

Active Model: In Phase I, as shown in Figure 1, the tag interacts with the attacker, who is free to choose non-random $\mathbf{a}$. However, $\mathbf{b}$ remains randomly chosen. Note that the attacker does not have access to a reader, and thus is unaware of the results of the reader's verification step.

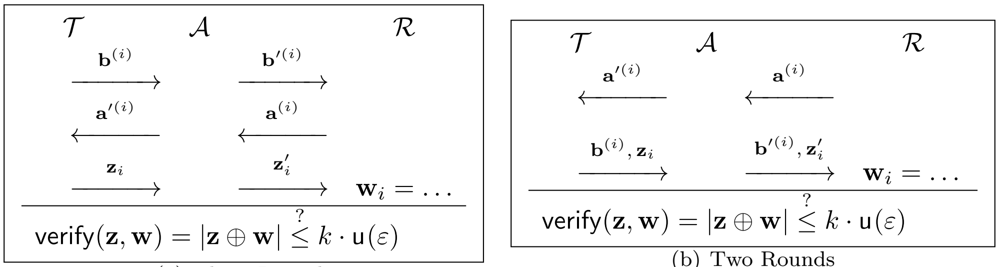

$\mathcal{T}$
$\mathcal{A}$
$\xrightarrow{\mathbf{b}^{(i)}}$
$\xleftarrow{\mathbf{a}^{(i)}}$
$\xrightarrow{\mathbf{z}_i}$
$\mathcal{T}$
$\mathcal{A}$
$\xleftarrow{\mathbf{a}^{(i)}}$
$\xrightarrow{\mathbf{b}^{(i)}, \mathbf{z}_i}$

(c) Three Rounds

(d) Two Rounds

Figure 1: Active

$\mathsf{Man\text{-}in\text{-}the\text{-}Middle}$ Model: In Phase I, the attacker may eavesdrop on and modify any message, as shown in Figure 2. Additionally, the attacker learns the decisions made by the reader's verification step.


$\mathcal{T}$
$\mathcal{A}$
$\mathcal{R}$
$\mathbf{b}^{(i)}$
$\mathbf{b}'^{(i)}$
$\mathbf{a}'^{(i)}$
$\mathbf{a}^{(i)}$
$\mathbf{z}_i$
$\mathbf{z}'_i$
$\mathbf{w}_i = \ldots$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}(\varepsilon)$
$\mathcal{T}$
$\mathcal{A}$
$\mathcal{R}$
$\mathbf{a}'^{(i)}$
$\mathbf{a}^{(i)}$
$\mathbf{b}^{(i)}, \mathbf{z}_i$
$\mathbf{b}'^{(i)}, \mathbf{z}'_i$
$\mathbf{w}_i = \ldots$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}(\varepsilon)$

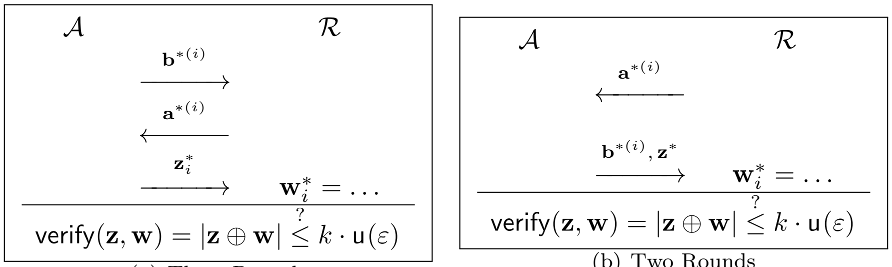
(a) Three Rounds

(b) Two Rounds

Figure 2: Man-in-the-Middle

GRS-MIM **Model:** The GRS-MIM model of Gilbert, Robshaw, and Seurin [GRS08b] is a variant of the Man-in-the-Middle model, in which the adversary is not allowed to modify $\mathbf{z}_i$. That is, $\forall i, \mathbf{z}_i = \mathbf{z}'_i$. GRS-MIM includes the attack on HB$^+$, so that HB$^+$ is not secure in the GRS-MIM model. The restriction $\mathbf{z}_i = \mathbf{z}'_i$ is unrealistic in practice, but GRS-MIM was used by a number of recent works in an attempt to improve on HB$^+$, due to the difficulty of proving security in the full Man-in-the-Middle model. However, GRS-MIM-security does not imply Man-in-the-Middle-security, and indeed, GRS-MIM-secure protocols have been successfully attacked in the full model [OOV08].

Phase II. In all three models, the goal of the attacker $\mathcal{A}$ is to authenticate successfully to the reader $\mathcal{R}$ in k rounds of Phase II, as shown in Figure 3. $\mathcal{A}$ is successful iff $\mathsf{verify}(\mathbf{z})$ returns true and $\mathbf{b}^* \neq \mathbf{0}$ in all k rounds.


$\mathcal{A}$
$\mathcal{R}$
$\mathbf{b}^{*(i)}$
$\mathbf{a}^{*(i)}$
$\mathbf{z}^*_i$
$\mathbf{w}^*_i = \ldots$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}(\varepsilon)$
$\mathcal{A}$
$\mathcal{R}$
$\mathbf{a}^{*(i)}$
$\mathbf{b}^{*(i)}, \mathbf{z}^*$
$\mathbf{w}^*_i = \ldots$
$\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}(\varepsilon)$

(a) Three Rounds

(b) Two Rounds

Figure 3: Phase II (All Models)

3 Our protocol

We present the HB$^\mathsf{N}$ protocol, in 2-round and 3-round variants. Our secret key will be a matrix $\mathbf{X} \in \mathbb{F}_2^{n \times n}$. As before, $\mathbf{a}^{(i)}, \mathbf{b}^{(i)} \in \mathbb{F}_2^n$ are column vectors used in the execution. The protocol consists of the key generation step KeyGen and the authentication step Auth.

KeyGen. $\mathsf{KeyGen}(1^n)$ produces a matrix $\mathbf{X} \overset{\$}{\leftarrow} \mathbb{F}_2^{n \times n}$.

Auth. HB$^\mathsf{N}$ can be run in serial or in parallel. We describe the serial version first, and then modify the notation for the parallel version. The tag $\mathcal{T}_\varepsilon^\mathbf{X} = (\mathsf{Tb}(), \mathsf{Tz}^\mathbf{X}(\cdot,\cdot,\cdot))$ authenticates to the reader $\mathcal{R}_\varepsilon^\mathbf{X} = (\mathsf{Ra}(), \mathsf{Rw}^\mathbf{X}(\cdot,\cdot,\cdot))$ by performing k rounds of the protocol, as shown in Figure 4. Let $\mathsf{Ra}() = \mathsf{Tb}() = \mathsf{ab}()$, and $\mathsf{Rw}(\cdot,\cdot,\cdot) = \mathsf{Tz}(\cdot,\cdot,\cdot) = \mathsf{wz}(\cdot,\cdot,\cdot)$, as shown in Algorithm 2.

In each of k rounds, which can be executed in serial or in parallel, $\mathcal{T}_\varepsilon(\mathbf{X})$ draws $(\mathbf{b}^{(i)}, \mathbf{f}_i) \overset{\$}{\leftarrow} \mathsf{Tb}()$, while $\mathcal{R}$ draws $(\mathbf{a}^{(i)}, \mathbf{e}_i) \overset{\$}{\leftarrow} \mathsf{Ra}()$. $\mathcal{T}$ sends $\mathbf{b}^{(i)}$ to $\mathcal{R}$, while $\mathcal{R}$ sends $\mathbf{a}^{(i)}$ to $\mathcal{T}$. This can be done in either order: if $\mathcal{T}$ sends first, the protocol becomes 3 rounds, while if $\mathcal{R}$ sends first, the protocol becomes 2 rounds. Finally, $\mathcal{T}_\varepsilon(\mathbf{X})$ computes $\mathbf{z}_i = \mathsf{Tz}^\mathbf{X}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{f}_i)$ and sends to $\mathcal{R}_\varepsilon^\mathbf{X}$. $\mathcal{R}_\varepsilon(\mathbf{X})$ computes $\mathbf{w}_i = \mathsf{Rw}^\mathbf{X}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{f}_i)$. At the end of k rounds, $\mathcal{R}$ computes $|\mathbf{z} \oplus \mathbf{w}|$ and to determine what fraction of responses were correct. $\mathcal{R}$ also tests to ensure that $\forall i \in [k], \mathbf{b}^{(i)} \neq \mathbf{0}^n$. If all $\mathbf{b}^{(i)}$ are nonzero and more than $k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) = k(1+\delta)(\varepsilon \oplus \varepsilon)$ for some completeness parameter δ, the reader accepts. [2]

Parallel version. We can use matrix notation to simplify working with HB^N in parallel, as shown in Figure 5 and Algorithm 3. Let $\mathbf{A}, \mathbf{B} \in \mathbb{F}_2^{n \times k}$ be matrices for which $\forall i \in [k], \mathbf{A}\mathbf{e}^{(i)} = \mathbf{a}^{(i)}, \mathbf{B}\mathbf{e}^{(i)} = \mathbf{b}^{(i)}$. That is, the columns of $\mathbf{A}, \mathbf{B}$ respectively are the vectors $\mathbf{a}^{(i)}, \mathbf{b}^{(i)}$ respectively. Then in the two-round version, for example, $\mathcal{R}$ sends the challenge $\mathbf{A} \overset{\$}{\leftarrow} \mathbb{F}_2^{n\times k}$. $\mathcal{T}$ replies with $\mathbf{B} \overset{\$}{\leftarrow} \mathbb{F}_2^{n\times k}$ and $\mathbf{z} = \mathsf{diag}(\mathbf{A}^\top \mathbf{X} \mathbf{B}) \oplus \mathbf{e}$, where $\mathbf{e} \overset{\$}{\leftarrow} \mathsf{Ber}_\varepsilon^n$. $\mathcal{R}$ computes $\mathbf{w} = \mathsf{diag}(\mathbf{A}^\top \mathbf{X} \mathbf{B}) \oplus \mathbf{f}$, where $\mathbf{f} \overset{\$}{\leftarrow} \mathsf{Ber}_\varepsilon^n$, and accepts iff $\forall i \in [k], \mathbf{B}\mathbf{e}^{(i)} \neq \mathbf{0}^n$ and $|\mathbf{z} \oplus \mathbf{w}| \leq \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon)$.

$$\begin{array}{lcl}
\mathcal{T} & & \mathcal{R} \\
(\mathbf{a}^{(i)}, \mathbf{e}_i) \overset{\$}{\leftarrow} \mathsf{ab}() & \xrightarrow{\mathbf{b}^{(i)}} & (\mathbf{b}^{(i)}, \mathbf{f}_i) \overset{\$}{\leftarrow} \mathsf{ab}() \\
\mathbf{z}_i = \mathsf{wz}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{e}_i) & \xleftarrow{\mathbf{a}^{(i)}} & \mathbf{w}_i = \mathsf{wz}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{f}_i) \\
 & \xrightarrow{\mathbf{z}_i} & \\
\hline
\multicolumn{3}{l}{\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) \wedge \forall i, \mathbf{b}^{(i)} \neq \mathbf{0}^n}
\end{array}$$

(a) Three Rounds

$$\begin{array}{lcl}
\mathcal{T} & & \mathcal{R} \\
(\mathbf{a}^{(i)}, \mathbf{e}_i) \overset{\$}{\leftarrow} \mathsf{ab}() & \xleftarrow{\mathbf{a}^{(i)}} & (\mathbf{b}^{(i)}, \mathbf{f}_i) \overset{\$}{\leftarrow} \mathsf{ab}() \\
\mathbf{z}_i = \mathsf{wz}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{e}_i) & \xrightarrow{\mathbf{b}^{(i)}, \mathbf{z}_i} & \mathbf{w}_i = \mathsf{wz}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}^{(i)}, \mathbf{f}_i) \\
\hline
\multicolumn{3}{l}{\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) \wedge \forall i, \mathbf{b}^{(i)} \neq \mathbf{0}^n}
\end{array}$$

(b) Two Rounds

Figure 4: HB^N (Serial notation)

$$\begin{array}{lcl}
\mathcal{T} & & \mathcal{R} \\
(\mathbf{A}, \mathbf{e}) \overset{\$}{\leftarrow} \mathsf{ab}(k) & \xrightarrow{\mathbf{B}} & (\mathbf{B}, \mathbf{f}) \overset{\$}{\leftarrow} \mathsf{ab}(k) \\
\mathbf{z} = \mathsf{wz}^{\mathbf{X}}(\mathbf{A}, \mathbf{B}, \mathbf{e}) & \xleftarrow{\mathbf{A}} & \mathbf{w} = \mathsf{wz}^{\mathbf{X}}(\mathbf{A}, \mathbf{B}, \mathbf{f}) \\
 & \xrightarrow{\mathbf{z}} & \\
\hline
\multicolumn{3}{l}{\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) \wedge \forall i, \mathbf{B}\mathbf{e}^{(i)} \neq \mathbf{0}^n}
\end{array}$$

(a) Three Rounds

$$\begin{array}{lcl}
\mathcal{T} & & \mathcal{R} \\
(\mathbf{A}, \mathbf{e}) \overset{\$}{\leftarrow} \mathsf{ab}(k) & \xleftarrow{\mathbf{A}} & (\mathbf{B}, \mathbf{f}) \overset{\$}{\leftarrow} \mathsf{ab}(k) \\
\mathbf{z} = \mathsf{wz}^{\mathbf{X}}(\mathbf{A}, \mathbf{B}, \mathbf{e}) & \xrightarrow{\mathbf{B}, \mathbf{z}} & \mathbf{w} = \mathsf{wz}^{\mathbf{X}}(\mathbf{A}, \mathbf{B}, \mathbf{f}) \\
\hline
\multicolumn{3}{l}{\mathsf{verify}(\mathbf{z}, \mathbf{w}) = |\mathbf{z} \oplus \mathbf{w}| \overset{?}{\leq} k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) \wedge \forall i, \mathbf{B}\mathbf{e}^{(i)} \neq \mathbf{0}^n}
\end{array}$$

(b) Two Rounds

Figure 5: HB^N (Parallel notation)

```
function $\mathsf{wz}^{\mathbf{X}}(\mathbf{a}, \mathbf{b}, e)$
    return $\mathbf{a}^\top \mathbf{X} \mathbf{b} \oplus e$
function $\mathsf{ab}()$
    return $(\mathbf{a}, e) \overset{\$}{\leftarrow} (\mathsf{U}_n, \mathsf{Ber}_\varepsilon)$
```

Algorithm 2: Algorithms for HB^N (Serial notation)

```
function $\mathsf{wz}^{\mathbf{X}}(\mathbf{A}, \mathbf{B}, \mathbf{e})$
    return $\mathsf{diag}(\mathbf{A}^\top \mathbf{X} \mathbf{B}) \oplus \mathbf{e}$
function $\mathsf{ab}(k)$
    return $(\mathbf{A}, \mathbf{e}) \overset{\$}{\leftarrow} (\mathsf{U}_{n\times k}, \mathsf{Ber}_\varepsilon^k)$
```

Algorithm 3: Algorithms for HB^N (Parallel notation)

4 Learning Subspaces with Noise (LSN)

Outline. In this section, we present a new conceptual tool in for analyzing HB-like protocol, the LSN (Learning Subspaces with Noise) problem, as shown in Algorithm 4.3. The security of LSN is equivalent to that of LPN. First, in Section 4.1, we introduce a new (to our knowledge) compact notation for precisely working with sums of random variables over $\mathbb{F}_2$, in order to simplify working with LPN and LSN. Next, in Section 4.2, we establish several fundamental properties of LPN. We work with LSN itself in Section 4.3.

[2] δ also governs the soundness of the protocol, which will be discussed in Section 5.6.

4.1 Working with probability distributions of additive variables over $\mathbb{F}_2$

We will need to analyze sums of noise distributions. Our task will be made easier by the use of a compact and flexible notation describing our distributions. At the most basic level, we need to understand the sum of two different Bernoulli distributions, $\mathsf{Ber}_\delta \oplus \mathsf{Ber}_\gamma$. Intuitively, noise is additive, and bounded above by $\delta + \gamma$. However, it is also possible for errors to cancel. Indeed,

$$\begin{aligned}\Pr[1 \leftarrow \mathsf{Ber}_\delta \oplus \mathsf{Ber}_\gamma] &= \Pr[1 \leftarrow \mathsf{Ber}_\delta \wedge 0 \leftarrow \mathsf{Ber}_\gamma] + \Pr[0 \leftarrow \mathsf{Ber}_\delta \wedge 1 \leftarrow \mathsf{Ber}_\gamma] \\ &= \delta(1-\gamma) + \gamma(1-\delta) = \delta + \gamma - 2\gamma\delta \end{aligned} \tag{2}$$

We would like to define an operator that adds these distributions, in the same sense that $\oplus$ is the additive operator over $\mathbb{F}_2$. We can describe each distribution X by a single scalar, $\delta_X = \Pr[X = 1]$, with δ_X an element of the closed interval $[0, 1]$. So, given $\oplus : \mathbb{F}_2 \times \mathbb{F}_2 \to \mathbb{F}_2$, we define an induced operator $\oplus^* : [0, 1] \times [0, 1] \to [0, 1]$ which adds distributions:

$$\mathsf{Ber}_{\gamma \oplus^* \delta} = \mathsf{Ber}_\gamma \oplus \mathsf{Ber}_\delta$$

It follows from Equation 2 that for all $\gamma, \delta \in [0, 1]$, $\oplus^*$ must satisfy $\gamma \oplus^* \delta = \delta + \gamma - 2\gamma\delta$. This is sufficient to uniquely define the operator. $\oplus^*$ acts similarly to the familiar binary operator $\oplus$: it is associative, commutative, and obeys the equalities $0 \oplus^* x = x$ and $1 \oplus^* x = 1 - x$ for all $x \in [0, 1]$. For this reason, we drop the $*$ and simply refer to our operator as $\oplus$. We also observe that we can extend the complement operator $\bar{\cdot}$ to all of $[0, 1]$, so that for all $\delta \in [0, 1]$, $\overline{\delta} = 1 \oplus \delta$. In summary, we have defined $\oplus, \bar{\cdot}$ so that

$$\begin{aligned} \forall \delta \in [0,1],\ \overline{\delta} &\doteq 1 \oplus \delta = 1 - \delta \\ \forall \gamma, \delta \in [0,1],\ \gamma \oplus \delta &\doteq \overline{\delta} \cdot \gamma + \delta \cdot \overline{\gamma} = (1-\delta)\gamma + (1-\gamma)\delta = \gamma + \delta - 2\gamma \cdot \delta \end{aligned} \tag{3}$$

Other useful facts about $\oplus$ over $[0, 1]$ that we will use in the following are:

Fact 2. $\forall \varepsilon \in [0, 1], \frac{1}{2} \oplus \varepsilon = \frac{1}{2}$.

Fact 3. $\forall \hat{b} \in \mathbb{F}_2, \Pr[e \xleftarrow{\$} \mathsf{Ber}_\varepsilon : e = \hat{b}] = \hat{b} \oplus \overline{\varepsilon} = \begin{cases} \varepsilon & \text{if } \hat{b} = 1 \\ 1 - \varepsilon & \text{if } \hat{b} = 0 \end{cases}.$

The presence of the complement operator is due to the convention of parameterizing the Bernoulli distribution by $\Pr[\mathsf{Ber}_\varepsilon = 1] = \varepsilon$. If $\Pr[\mathsf{Ber}_\varepsilon = 0]$ was used instead, we would obtain the simpler expression $\hat{b} \oplus \varepsilon$. For this reason, we have chosen to complement the error term ε rather than the desired bit $\hat{b}$.

Fact 4. *Let* $\varepsilon^{\oplus n} = \overbrace{\varepsilon \oplus \varepsilon \oplus \ldots \oplus \varepsilon}^{n}$. *Then* $\varepsilon^{\oplus n} = \frac{1-(1-2\varepsilon)^n}{2}$.

Fact 4 tells us that noise behaves multiplicatively rather than additively. The reason it appears additive for small noise rates corresponds to the approximation $\exp(x) \approx 1 + x$ for small x. More precisely, the scaled distance from $\frac{1}{2}$ behaves multiplicatively:

Fact 5. *For all* $\delta, \tau \in [0, 1]$, $\frac{1}{2}(1 - \delta) \oplus \frac{1}{2}(1 - \tau) = \frac{1}{2}(1 - \delta\tau)$.

4.2 Learning Parity with Noise (LPN)

Next we establish a characterization of the LPN distribution in Lemma 6 and examine its consequences.

Lemma 6. $\forall (\hat{\mathbf{a}}, \hat{b}) \in \mathbb{F}_2^n \times \mathbb{F}_2, \Pr[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{LPN}_\varepsilon^{\mathbf{x}}] = (\hat{\mathbf{a}}^\top \mathbf{x} \oplus \hat{b} \oplus \overline{\varepsilon})2^{-n} = \begin{cases} \varepsilon 2^{-n} & \text{if } \hat{\mathbf{a}}^\top \mathbf{x} \neq \hat{b} \\ (1-\varepsilon)2^{-n} & \text{if } \hat{\mathbf{a}}^\top \mathbf{x} = \hat{b} \end{cases}.$

Proof. Since $\mathbf{a}, e$ are chosen independently, we have:

$$\begin{aligned} \Pr[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{LPN}_\varepsilon^{\mathbf{x}}] &= \Pr[(\mathbf{a}, b) \leftarrow \mathsf{LPN}_\varepsilon^{\mathbf{x}} : \mathbf{a} = \hat{\mathbf{a}}] \cdot \Pr[e \leftarrow \mathsf{Ber}_\varepsilon : e = \hat{\mathbf{a}}^\top \mathbf{x} \oplus \hat{b}] \\ &= \Pr[\hat{\mathbf{a}} \xleftarrow{\$} \mathbb{F}_2^n] \cdot \Pr[e \leftarrow \mathsf{Ber}_\varepsilon : e = \hat{\mathbf{a}}^\top \mathbf{x} \oplus \hat{b}] \\ &= 2^{-n}(\hat{\mathbf{a}}^\top \mathbf{x} \oplus \hat{b} \oplus \overline{\varepsilon}) \end{aligned} \tag{4}$$

Equation 4 follows from Fact 3. □

Summing over all $\hat{\mathbf{a}} \in \mathbb{F}_2^n$ yields the following corollary:

Corollary 7. $\forall \mathbf{x} \neq \mathbf{0}^n$, $\Pr[(\mathbf{a}, b) \leftarrow \mathsf{LPN}_\varepsilon^{\mathbf{x}} : b = 0] = \frac{1}{2}$.

Setting $\varepsilon = \frac{1}{2}$ in Lemma 6 and using Fact 2 yields the following corollary:

Corollary 8. $\Pr[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{LPN}^{\mathbf{x}}_{1/2}] = 2^{-n-1}$. *Equivalently,* $\mathsf{LPN}^{\mathbf{x}}_{1/2} \equiv \mathsf{U}_n \times \mathsf{U}_1$.

Finally, a useful consequence of the random self-reducibility properties of the LPN problem is that, given any LPN distribution for any fixed key $\mathbf{x}$, we can produce an LPN distribution with a random key and the same ε:

Corollary 9. *For any $\varepsilon \in [0, 1]$ and $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^n$, the distribution $\mathsf{LPN}^{\mathbf{x}\oplus\mathbf{y}}_{\varepsilon}$ can be efficiently sampled given $\mathbf{y}$ and oracle access to $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$.*

Proof. Consider the "translated" distribution $\mathsf{Tr\text{-}LPN}^{\mathbf{x}}_{\varepsilon}$ defined as follows: draw a sample $(\mathbf{a}, b)$ from $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$, and return $(\mathbf{a}, b \oplus \mathbf{a}^{\top}\mathbf{y})$. Then:

$$\begin{aligned}\Pr[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{Tr\text{-}LPN}^{\mathbf{x}}_{\varepsilon}] &= \Pr[(\hat{\mathbf{a}}, \hat{b} \oplus \hat{\mathbf{a}}^{\top}\mathbf{y}) \leftarrow \mathsf{LPN}^{\mathbf{x}}_{\varepsilon}] \\ &= 2^{-n}(\hat{\mathbf{a}}^{\top}\mathbf{x} \oplus \hat{\mathbf{a}}^{\top}\mathbf{y} \oplus \hat{b} \oplus \bar{\varepsilon}) \\ &= \Pr[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{LPN}^{\mathbf{x}\oplus\mathbf{y}}_{\varepsilon}]\end{aligned}$$

□

Corollary 9 says that we can "duplicate" an LPN distribution $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$: we can use some of its samples as is, from the original distribution, and at the same time use the remaining samples as if they came from an entirely different LPN distribution with the same ε (even for unknown ε). Furthermore, if the "translation" vector $\mathbf{y}$ is uniformly random, then the original LPN distribution $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$ and its "translate" $\mathsf{LPN}^{\mathbf{x}\oplus\mathbf{y}}_{\varepsilon}$ are independent.

Lemma 10. *Given a challenge oracle $\mathcal{O}_b = \begin{cases} \mathsf{LPN}^{\mathbf{r}}_{1/2} & \text{if } b = 0 \\ \mathsf{LPN}^{\mathbf{r}}_{\varepsilon}, & \text{if } b = 1 \end{cases}$, we can construct ℓ separate challenge oracles, $(\mathcal{O}^{(1)}_b, \ldots, \mathcal{O}^{(\ell)}_b) = \begin{cases} (\mathsf{LPN}^{\mathbf{z}^{(1)}}_{1/2}, \ldots, \mathsf{LPN}^{\mathbf{z}^{(\ell)}}_{1/2}) & \text{if } b = 0 \\ (\mathsf{LPN}^{\mathbf{z}^{(1)}}_{\varepsilon}, \mathsf{LPN}^{\mathbf{z}^{(\ell)}}_{\varepsilon}) & \text{if } b = 1 \end{cases}$*

Proof of Lemma 10. Let $\mathbf{y}^{(i)} \xleftarrow{\$} \mathbb{F}_2^n, \forall i \in [\ell]$. Repeated applications of Corollary 9 yield new oracles $(\mathsf{LPN}^{\mathbf{z}^{(1)}}_{\rho}, \ldots, \mathsf{LPN}^{\mathbf{z}^{(\ell)}}_{\rho})$, where $\mathbf{z}^{(i)} = \mathbf{y}^{(i)} \oplus \mathbf{r}$. Since the $\mathbf{z}^{(i)}$ are independently and uniformly distributed for all $i \in [\ell]$, and since $\rho = \frac{1}{2}$ for $b = 0$ and $\rho = \varepsilon$ for $b = 1$, this establishes the lemma. □

4.3 Learning Subspaces with Noise (LSN)

Next, we introduce $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$, which uses $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$ to produce a biased halfspace distribution: $\mathbf{a}$ is chosen randomly subject to the condition that $\mathbf{a}^{\top}\mathbf{x}$ is distributed according to $\mathsf{Ber}_{\rho} \oplus \mathsf{Ber}_{\varepsilon}$. In particular, for $\mathsf{LSN}^{\mathbf{x}}_{0,\varepsilon}$, $\mathbf{a}^{\top}\mathbf{x} \equiv \mathsf{Ber}_{\varepsilon}$. We derive an expression for the distribution of $(\mathbf{a}, b) \xleftarrow{\$} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ in Lemma 12 intermediate results. We consider the case $\varepsilon = \frac{1}{2}$ in Corollary 13. Next we consider the conditional distribution of b given $\mathbf{a}^{\top}\mathbf{x} = \hat{a}$ in Corollary 15. Finally, we establish a connection between hardness of $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ and $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$.

```
function $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$
    return $\mathsf{LSN}_{\rho}(\mathsf{LPN}^{\mathbf{x}}_{\varepsilon})$

function $\mathsf{LSN}_{\rho}(\mathsf{Samp})$
    $i = 0$
    $\hat{b} \xleftarrow{\$} \mathsf{Ber}_{\rho}$
    repeat
        $(\mathbf{a}^{(i)}, b_i) \xleftarrow{\$} \mathsf{Samp}()$
        $i \leftarrow i + 1$
    until $b_i = \hat{b}$
    return $(\mathbf{a}^{(i)}, b_i)$
```

Algorithm 4: LSN

The algorithm $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$, shown in Algorithm 4.3, is constructed from the oracle $\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$. $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ first uses its own randomness to draw $\hat{b} \xleftarrow{\$} \mathsf{Ber}_{\rho}$. Next, for $i \geq 0$ it repeatedly obtains $(\mathbf{a}^{(i)}, b_i) \xleftarrow{\$} \mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$. The algorithm waits until $b_i = \hat{b}$, and then outputs $(\mathbf{a}^{(i)}, b_i)$. The algorithm runs in expected polynomial time.

The distribution of $(\hat{\mathbf{a}}, \hat{\mathbf{b}}) \xleftarrow{\$} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ can be computed from ρ and the distribution of $(\hat{\mathbf{a}}, \hat{\mathbf{b}}) \xleftarrow{\$} \mathsf{LPN}^{\mathbf{x}}_{\rho,\varepsilon}$:

Lemma 11. $\Pr[(\hat{\mathbf{a}}, \hat{b}) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}] = 2\Pr[\hat{b} \overset{\$}{\leftarrow} \mathsf{Ber}_\rho] \cdot \Pr[(\hat{\mathbf{a}}, \hat{b}) \overset{\$}{\leftarrow} \mathsf{LPN}^{\mathbf{x}}_{\varepsilon}]$.

Proof of Lemma 11. The algorithm $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ progresses through a series of rounds. In each round, $\mathsf{LSN}_{\rho,\varepsilon}$ samples $(\mathbf{a}^{(i)}, b_i) \overset{\$}{\leftarrow} \mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$. The algorithm terminates by returning $(\mathbf{a}^{(i)}, b_i)$ when it finds $b_i = \hat{b}$. To model its distribution, we define a series of events. Let $R_{\hat{b}}$ be the event that $\hat{b} = b_i$. Let $S^{(i)}$ be the event that, given that the algorithm is active during round i, the algorithm terminates by returning $(\mathbf{a}^{(i)}, b_i)$ in round i, for $i \geq 0$. Finally, let $T^{(i)}_{(\hat{\mathbf{a}},\hat{b})}$ be the event that $(\hat{\mathbf{a}}, \hat{b}) \overset{\$}{\leftarrow} \mathsf{LPN}^{\mathbf{x}}_{\varepsilon}$ in round i. It follows that

$$\Pr[(\hat{\mathbf{a}}, \hat{b}) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}] = \sum_{i=0}^{\infty} \left(\Pr[S^{(i)}] \prod_{j<i} \Pr[\overline{S^{(j)}}] \right) \left(\Pr[R_{\hat{b}}] \cdot \Pr[T^{(i)}_{(\hat{\mathbf{a}},\hat{b})}] \right) \tag{5}$$

$$= \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i \left(\Pr[\hat{b} \overset{\$}{\leftarrow} \mathsf{Ber}_\rho] \cdot \Pr[T^{(i)}_{(\hat{\mathbf{a}},\hat{b})}] \right) \tag{6}$$

$$= (\hat{b} \oplus \overline{\rho}) \sum_{i=0}^{\infty} \left(\frac{1}{2}\right)^i \left(\Pr[T^{(i)}_{(\hat{\mathbf{a}},\hat{b})}] \right) \tag{7}$$

$$= 2(\hat{b} \oplus \overline{\rho}) \cdot \Pr[T^{(0)}_{(\hat{\mathbf{a}},\hat{b})}] \tag{8}$$

Equation 5 follows from summing over all $i \geq 0$ and all bits $\hat{b} \in \mathbb{F}_2$ the probability that $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ terminates in round i with output $(\hat{\mathbf{a}}, \hat{b})$. Equation 6 follows from Corollary 7 and from the definition of LSN in Algorithm 4.3. Equation 7 follows from Fact 3. Equation 8 follows from the geometric series formula and from $\forall i, \Pr[T^{(i)}_{(\hat{\mathbf{a}},\hat{b})}] = \Pr[T^{(0)}_{(\hat{\mathbf{a}},\hat{b})}]$. □

Next, we apply Lemma 6 to derive the probability distribution of LSN.

Lemma 12. *For all* $\hat{\mathbf{a}} \in \mathbb{F}_2^n, \hat{a} \in \mathbb{F}_2, \hat{b} \in \mathbb{F}_2$,

(a) $\Pr\left[(\hat{\mathbf{a}}, \hat{b}) \leftarrow \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}\right] = (\hat{b} \oplus \overline{\rho})(\hat{b} \oplus \hat{\mathbf{a}}^\top \mathbf{x} \oplus \overline{\varepsilon})2^{-n+1}$

(b) $\Pr\left[(\mathbf{a}, b) \leftarrow \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : \hat{\mathbf{a}} = \mathbf{a}\right] = (\hat{\mathbf{a}}^\top \mathbf{x} \oplus \rho \oplus \overline{\varepsilon})2^{-n+1}$

(c) $\forall \mathbf{x} \neq \mathbf{0}^n, \Pr\left[(\mathbf{a}, b) \leftarrow \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : (\mathbf{a}^\top \mathbf{x}, b) = (\hat{a}, \hat{b})\right] = (\hat{b} \oplus \overline{\rho})(\hat{b} \oplus \hat{\mathbf{a}}^\top \mathbf{x} \oplus \overline{\varepsilon})$

(d) $\forall \mathbf{x} \neq \mathbf{0}^n, \Pr\left[(\mathbf{a}, b) \leftarrow \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : \mathbf{a}^\top \mathbf{x} = \hat{a}\right] = (\hat{a} \oplus \rho \oplus \overline{\varepsilon})$

Proof of Lemma 12. Lemma 12(a) follows immediately from Lemma 6 applied to Lemma 11. Lemma 12(b) follows from Equation 3 applied to $\delta = \hat{b} \oplus \overline{\rho}$, $\gamma = \hat{b} \oplus \hat{\mathbf{a}}^\top \mathbf{x} \oplus \varepsilon$. Lemma 12(c) and Lemma 12(d) follow from Lemma 12(a) and Lemma 12(b), respectively, from summing over all $\hat{\mathbf{a}}$ such that $\hat{\mathbf{a}}^\top \mathbf{x} = \hat{a}$ and noting that $|\ker(\mathbf{x})| = |\mathbb{F}_2^n \setminus \ker(\mathbf{x})| = 2^{n-1}$. □

Since $\forall x, x \oplus \frac{1}{2} = \frac{1}{2}$, we obtain the following corollary of Lemma 12(a).

Corollary 13. *For* $\mathbf{x} \neq 0$, $\forall(\hat{\mathbf{a}}, \hat{b})$, $\Pr\left[(\hat{\mathbf{a}}, \hat{b}) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\frac{1}{2}}\right] = (\hat{b} \oplus \overline{\rho})2^{-n}$.

Corollary 14. $\left\{\mathbf{a}^\top \mathbf{x} : (\mathbf{a}, b) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}\right\} \equiv \mathsf{Ber}_{\rho \oplus \varepsilon}$.

Proof. The corollary follows from combining Lemma 12(d) and Fact 3 and noting that $\overline{\rho \oplus \varepsilon} = \rho \oplus \overline{\varepsilon}$. □

Combining Lemma 12(c) and Lemma 12(d), we can obtain the conditional probability of obtaining $(\mathbf{a}, \hat{b})$ given $\mathbf{a}^\top \mathbf{x} = \hat{a}$.

Corollary 15. *Let* $p^{\rho,\varepsilon}_{\hat{b}|\hat{a}} = \Pr_{\mathbf{a}^\top \mathbf{x} = \hat{a}}[(\mathbf{a}, b) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : b = \hat{b}]$ *be the conditional probability of obtaining* $\hat{b}$ *from* $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ *subject to the condition* $\mathbf{a}^\top \mathbf{x} = \hat{a}$. *Then* $\forall(\hat{b}, \hat{a})$, $p^{\rho,\varepsilon}_{\hat{b}|\hat{a}} = \dfrac{(\hat{b} \oplus \overline{\rho})(\hat{b} \oplus \hat{a} \oplus \overline{\varepsilon})}{\hat{a} \oplus \rho \oplus \overline{\varepsilon}}$.

Proof of Corollary 15.

$$\Pr_{\mathbf{a}^\top \mathbf{x} = \hat{a}}[(\mathbf{a}, b) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : b = \hat{b}] = \frac{\Pr[(\mathbf{a}, b) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : \mathbf{a}^\top \mathbf{x} = \hat{a} \wedge \hat{b} = b]}{\Pr[(\mathbf{a}, b) \overset{\$}{\leftarrow} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} : \mathbf{a}^\top \mathbf{x} = \hat{a}]} \tag{9}$$

$$= \frac{(\hat{b} \oplus \overline{\rho})(\hat{b} \oplus \hat{a} \oplus \varepsilon)}{\hat{a} \oplus \rho \oplus \overline{\varepsilon}} \tag{10}$$

Equation 9 follows from Bayes' rule. Equation 10 follows from Lemma 12(c) and Lemma 12(d). □

In particular, for $\varepsilon = \rho$ and $\hat{a} = 1$, $\left\{ b : (\mathbf{a}, b) \xleftarrow{\$} \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon} \wedge \mathbf{a}^{\top}\mathbf{x} = 1 \right\} \equiv \mathsf{Ber}_{\frac{1}{2}}$, which will make LSN useful in the security proof.

Corollary 16. *For all bits $\hat{b} \in \mathbb{F}_2$, $p^{\varepsilon,\varepsilon}_{\hat{b}|1} = \frac{1}{2}$.*

Proof.

$$
\begin{aligned}
p^{\varepsilon,\varepsilon}_{\hat{b}|1} &= \frac{(\hat{b} \oplus \overline{\varepsilon})(\hat{b} \oplus \varepsilon)}{\rho \oplus \varepsilon} && (11)\\
&= \frac{\varepsilon(1-\varepsilon)}{2\varepsilon - 2\varepsilon^2}\\
&= \frac{1}{2}
\end{aligned}
$$

Equation 11 follows from Corollary 15.

□

Hardness of LSN. Hardness of LSN can be defined using an indistinguishability game. More formally, the advantage of an algorithm $\mathcal{A}$ is defined using a game in which the adversary attempts to guess whether the oracle is $\mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ or $\mathsf{U}_n \times \mathsf{Ber}_\rho$, which is perfectly equivalent, by Corollary 13, to $\mathsf{LSN}^{\mathbf{x}}_{\rho,\frac{1}{2}}$.

$$
\mathsf{Adv}^{\mathsf{LSN}}_{\mathcal{A}}(\rho,\varepsilon,n) = \left| \Pr\left[\begin{array}{l} \mathbf{x} \xleftarrow{\$} \mathsf{KG}, b \xleftarrow{\$} \mathbb{F}_2, \\ \mathcal{O}_b = \begin{cases} \mathsf{LSN}^{\mathbf{x}}_{\rho,\frac{1}{2}} & \text{if } b = 0 \\ \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}, & \text{if } b = 1 \end{cases}, \\ \hat{b} \xleftarrow{\$} \mathcal{A}^{\mathcal{O}_b}() \end{array} : \hat{b} = b \right] - \frac{1}{2} \right| \tag{12}
$$

For given bitlength n and noise rate ε, and for arbitrary ρ, hardness of LSN and of LPN are directly related:

Lemma 17. *For any ρ, ε, if there exists a probabilistic polynomial time adversary $\mathcal{A}$ achieving $\mathsf{Adv}^{\mathsf{LSN}}_{\mathcal{A}}(\rho,\varepsilon,n) \geq \delta$, then there exists a probabilistic polynomial time adversary $\mathcal{B}$ for which $\mathsf{Adv}^{\mathsf{LPN}}_{\mathcal{B}}(\varepsilon,n) \geq \delta$.*

Proof of Lemma 17. Let $\mathcal{B}^{\mathcal{O}} = \mathcal{A}^{\mathsf{LSN}_\rho(\mathcal{O})}$. That is, $\mathcal{B}$ runs $\mathcal{A}$ and gives $\mathcal{A}$ access to an oracle LSN_ρ applied to $\mathcal{B}$'s oracle $\mathcal{O}$. Since $\mathsf{LSN}_\rho(\mathsf{LPN}^{\mathbf{x}}_{\varepsilon}) \equiv \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}$ and $\mathsf{LSN}_\rho(\mathsf{LPN}^{\mathbf{x}}_{1/2}) \equiv \mathsf{U}_n \times \mathsf{Ber}_\rho$ by Corollary 8, $\mathsf{Adv}^{\mathsf{LPN}}_{\mathcal{B}}(\varepsilon,n)$ can be expressed as

$$
\begin{aligned}
\mathsf{Adv}^{\mathsf{LPN}}_{\mathcal{B}}(\varepsilon,n) &= \left| \Pr\left[\begin{array}{l} \mathbf{x} \xleftarrow{\$} \mathsf{KG}, b \xleftarrow{\$} \mathbb{F}_2, \\ \mathcal{O}_b = \begin{cases} \mathsf{LSN}^{\mathbf{x}}_{\rho,\frac{1}{2}} & \text{if } b = 0 \\ \mathsf{LSN}^{\mathbf{x}}_{\rho,\varepsilon}, & \text{if } b = 1 \end{cases}, \\ \hat{b} \xleftarrow{\$} \mathcal{B}^{\mathcal{O}_b}() \end{array} : \hat{b} = b \right] - \frac{1}{2} \right| \\
&= \mathsf{Adv}^{\mathsf{LSN}}_{\mathcal{A}}(\rho,\varepsilon,n).
\end{aligned}
$$

□

We will not need the reverse direction, but it is possible to show that LSN for an n-bit secret is at least as hard as LPN with a secret of length $n-1$ using Subspace LWE [Pie10]. Thus, LSN and LPN are essentially equivalent up to a 1 bit change in secret length.

5 Proof of Man-in-the-Middle-security

Let S_Γ be the event that the Reader accepts in the challenge phase of Game Γ. For any efficient adversary $\mathcal{A}$ and any game Γ, we define the advantage $\mathsf{Adv}^{\Gamma}_{\mathcal{A}} = \Pr[S_\Gamma]$. More generally, a game Γ in our sequence consists of the adversary's interactions with a tag $\mathcal{T} = \mathcal{T}^{\mathbf{X}}_{\Gamma}$ and a Phase I reader $\mathcal{R} = \mathcal{R}^{\mathbf{X}}_{\Gamma}$ using secret $\mathbf{X}$, and a Phase II reader $\mathcal{R}^{*\mathbf{X}_0} = \mathcal{R}^{*\mathbf{X}_0}_{\Gamma}$ using secret $\mathbf{X}_0$, which will not necessarily equal $\mathbf{X}$. We define the advantage of an adversary against $(\mathcal{T}^{\mathbf{X}}, \mathcal{R}^{\mathbf{X}}, \mathcal{R}^{*\mathbf{X}_0})$, for the two-round prtocol, as follows:

$$
\mathsf{Adv}^{(\mathcal{T}^{\mathbf{X}},\mathcal{R}^{\mathbf{X}},\mathcal{R}^{*\mathbf{X}_0})}_{\mathcal{A}} = \left| \Pr\left[\begin{array}{l} \mathbf{X} \xleftarrow{\$} \mathsf{KeyGen}_{\mathsf{HB^N}}, \\ s \xleftarrow{\$} \mathcal{A}_1^{\mathcal{T}^{\mathbf{X}},\mathcal{R}^{\mathbf{X}}}(1^n), \\ \mathbf{A}^* \xleftarrow{\$} \mathcal{R}^*_1, \\ (\mathbf{z}^*, \mathbf{B}^*) \xleftarrow{\$} \mathcal{A}_2(s, \mathbf{A}^*) \\ \mathbf{w}^* \xleftarrow{\$} \mathcal{R}_2^{*\mathbf{X}_0}(\mathbf{B}^*) \end{array} \quad : \quad \begin{array}{l} \forall i \in [k], \mathbf{B}^*\mathbf{e}^{(i)} \neq \mathbf{0}^n, \\ |\mathbf{z}^* \oplus \mathbf{w}^*| \leq k \cdot \mathsf{u}_{\mathsf{HB^N}}(\varepsilon) \end{array} \right]\right|
$$

For the 3-round protocol, $\mathcal{A}_2$ must output $\mathbf{B}^*$ before $\mathcal{A}_3$ receives $\mathbf{A}^*$. Note that we have split the Phase II Reader into two parts $\mathcal{R}_1^*$, and $\mathcal{R}_2^*$. The former does not require the key $\mathbf{X}_0$, while the latter does. Let $S_{\mathsf{HB^N}}$ be the event that the Reader accepts in the challenge phase $\mathsf{HB^N}$. For any efficient adversary $\mathcal{A}$, we define the adversary's advantage, $\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HB^N}} = \Pr[S_{\mathsf{HB^N}}]$, Our main result will be the following.

Theorem 18. *For any efficient adversary $\mathcal{A}$,*

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HB^N}} \leq 2m \cdot \varepsilon_{\mathsf{LPN}} + \mathsf{negl}$$

Outline. Theorem 18 will follow from Theorem 19 combined with the Hoeffding-Chernoff bound. In Section 5.1, we state Theorem 19 and Corollaries 20–22, which describe the sequence of games used for proving Theorem 19. We prove Corollary 20 in Section 5.2 via interpolating games. We prove Corollary 21, which allows us to replace keys by nearby keys, in Section 5.3. In Section 5.4, we state and prove Theorem 27, a technical result on randomness of bilinear functions. We apply Theorem 27 in Section 5.5 to prove Corollary 22. Finally, in Section 5.6, we complete the proof of Theorem 19 and calculate explicit soundness and completeness parameters in order to prove Theorem 18.

5.1 Sequence of Games

Theorem 19 uses a sequence of games $\mathrm{A}_{0,0}$ (in which the simulator runs the $\mathsf{HB^N}$ protocol) through $\mathrm{A}_{4,m-1}$ to show that $|\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HB^N}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}}| \leq 2m \cdot \varepsilon_{\mathsf{LPN}}$.

Theorem 19. *For all efficient $\mathcal{A}$ and for $m = n + \omega(\log n)$, $k = n - \omega(\log n)$, $k = \omega(\log n)$,*

$$|\mathsf{Adv}_{\mathcal{A}}^{A_{0,0}} - \mathsf{Adv}_{\mathcal{A}}^{A_{4,m-1}}| \leq 2m \cdot \varepsilon_{\mathsf{LPN}} + 2^{k-n} + k2^{-m} = 2m \cdot \varepsilon_{\mathsf{LPN}} + \mathsf{negl}$$

Game Definitions. For almost all games, $\mathbf{z}_i \overset{\$}{\leftarrow} \mathsf{Tz}^{\mathbf{X}}(\cdot,\cdot,\mathbf{e}_i), \mathbf{w}_i \overset{\$}{\leftarrow} \mathsf{Rw}^{\mathbf{X}}(\cdot,\cdot,\mathbf{f}_i), \mathbf{w}_i^* \overset{\$}{\leftarrow} \mathsf{Rw}^{*\mathbf{X}_0}(\cdot,\cdot,\mathbf{f}_i^*)$ remain the same, although the random inputs $\mathbf{e}_i, \mathbf{f}_i, \mathbf{f}_i^*$ may vary. The single exception is $\mathrm{A}_{4,m-1}$, in which $\mathbf{w}^* \overset{\$}{\leftarrow} \mathbb{F}_2$ is computed without the use of any key.

Thus, all the initial games can be completely described by $(\mathbf{X}, \mathbf{X}_0, \mathsf{Ra}(), \mathsf{Tb}(), \mathsf{Ra}^*())$, the Phase I and II keys and the sampling algorithms for $(\mathbf{a}, f), (\mathbf{b}, e), (\mathbf{a}^*, f^*)$ respectively. For all games, Figure 6 lists changes between games. $\mathbf{x}$ is the secret used by the oracle $\mathcal{O} = \mathsf{LPN}_{\varepsilon}^{\mathbf{x}}$ that interacts with the simulator. Every game in the sequence generates $\mathbf{s}^{(j)}, \mathbf{r}^{(j)}, \mathbf{t}^{(j)}, \mathbf{T}_j$ in the same way: $\forall j \in [m], \mathbf{s}^{(j)}, \mathbf{t}^{(j)} \overset{\$}{\leftarrow} \mathbb{F}_2^n, \mathbf{r}^{(j)} = \mathbf{t}^{(j)} \oplus \mathbf{x}, \mathbf{T}_j = \sum_{i=0}^{j-1} \mathbf{r}^{(i)} {\mathbf{s}^{(i)}}^{\top}$.

Game	Phase I key $\mathbf{X}$	$\mathsf{Ra}()$	$\mathsf{Tb}()$	Ra^*	Phase II key $\mathbf{X}_0$
$\mathrm{A}_{0,0}$	$\mathbf{X}_0 \overset{\$}{\leftarrow} \mathsf{KeyGen}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathbf{X}_0$
$\vdots$					
$\mathrm{A}_{1,j}$	$\mathbf{X}_j \overset{\$}{\leftarrow} \mathsf{KeyGen}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{r}^{(j)}}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{s}^{(j)}}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{x}}$	$\mathbf{X}_j \oplus \mathbf{T}_j$
$\mathrm{A}_{2,j}$	$\mathbf{X}_{j+1} \overset{\$}{\leftarrow} \mathsf{KeyGen}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{r}^{(j)}}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{s}^{(j)}}$	$\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{x}}$	$\mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}$
$\mathrm{A}_{3,j}$	$\mathbf{X}_{j+1} \overset{\$}{\leftarrow} \mathsf{KeyGen}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}$
$\vdots$					
$\mathrm{A}_{4,m-1}$	$\mathbf{X}_m \overset{\$}{\leftarrow} \mathsf{KeyGen}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathsf{LSN}_{\varepsilon,\frac{1}{2}}$	$\mathbf{w}^* \overset{\$}{\leftarrow} \mathbb{F}_2^k$

Figure 6: Summary of Games

Transitions between games. The proof of Theorem 19 is built from a sequence of games with several types of transitions, which are proven in Corollaries 20–22.

Changing Sampling of $\mathbf{b}^{(i)}, \mathbf{a}^{(i)}, \mathbf{a}^{*(i)}$. The first transition type, Corollary 20, hinges upon the computational hardness of LSN (and hence of LPN by Lemma 17). The transitions between Games $\mathrm{A}_{0,0}$-$\mathrm{A}_{1,0}$ change how $\mathbf{b}^{(i)}, \mathbf{a}^{(i)}, \mathbf{a}^{*(i)}$ are sampled using LSN, and the transitions to and from $\mathrm{A}_{3,j}$ change how $\mathbf{b}^{(i)}, \mathbf{a}^{(i)}$ are sampled using LSN. Corollary 20 will follow from the construction of interpolating games.

Corollary 20. *The game pairs $(A_{0,0}, A_{1,0}), (A_{2,\mathrm{j}}, A_{3,\mathrm{j}}), (A_{3,\mathrm{j}}, A_{1,\mathrm{j+1}})$, are equivalent:* $|\Pr[S_{A_{0,0}}] - \Pr[S_{A_{1,0}}]| \leq \varepsilon_{\mathsf{LPN}}$, $|\Pr[S_{A_{2,j}}] - \Pr[S_{A_{3,j}}]| \leq \varepsilon_{\mathsf{LPN}}$, $|\Pr[S_{A_{3,j}}] - \Pr[S_{A_{1,j}}]| \leq \varepsilon_{\mathsf{LPN}}$.

Switching the key from $\mathbf{X}_j$ to $\mathbf{X}_{j+1}$. In Games $\mathrm{A}_{1,j}$-$\mathrm{A}_{2,j}$, we use Corollary 21 to replace the Phase I and Phase II keys with nearby keys.

Corollary 21. *$\mathrm{A}_{1,j}$ and $\mathrm{A}_{2,j}$ are equivalent: $|\Pr[S_{A_{1,j}}] - \Pr[S_{A_{2,j}}]| = 0$*

The proof of Corollary 21 requires Lemma 23, a technical result related to LSN. Lemma 23 uses the LSN distribution to annihilate the adversary's contribution $\mathbf{s}^\top\mathbf{b}'$ to $\mathbf{w}$ corresponding to $\mathbf{r}\mathbf{s}^\top$. Corollary 21 will then follow from several applications of Lemma 23. This key lemma is actually the raison d'être of LSN, although we envision it being useful in other applications as well.

A sufficiently random Phase II key yields $\mathbf{w}^*$ indistinguishable from random. For the final step, Corollary 22 establishes that for sufficiently large m, no adversary can achieve advantage non-negligibly greater than the advantage of the adversary which simply chooses $\mathbf{z}^*$ at random.

Corollary 22. $|\mathsf{Adv}_{\mathcal{A}}^{A_{4,m-1}} - \mathsf{Adv}_{\mathcal{A}}^{A_{3,m-1}}| \leq k2^{-m} + 2^{k-n}$*, which is negligible for $k < n - \omega(\log n), m = \omega(\log n)$.*

5.2 Interpolating Games: Proof of Corollary 20

We define interpolating games as shown in Figure 7.

Game $\alpha_{\Gamma_1,\Gamma_2}$	$\mathbf{X}$	$\mathsf{Ra}()$	$\mathsf{Tb}()$	$\mathsf{Ra}^*()$	$\mathbf{X}_0$
$\alpha_{\mathrm{A}_{0,0},\mathrm{A}_{1,0}}$	$\mathbf{X}_0 \xleftarrow{\$} \mathsf{KeyGen}$	$\mathcal{O}_b^{(1)}$	$\mathcal{O}_b^{(2)}$	$\mathcal{O}_b^{(3)}$	$\mathbf{X}_0$
$\alpha_{\mathrm{A}_{2,j},\mathrm{A}_{3,j}}$	$\mathbf{X}_{j+1} \xleftarrow{\$} \mathsf{KeyGen}$	$\mathcal{O}_b^{(1)}$	$\mathcal{O}_b^{(2)}$	$\mathcal{O}_b^{(3)}$	$\mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}$
$\alpha_{\mathrm{A}_{3,j},\mathrm{A}_{1,j+1}}$	$\mathbf{X}_{j+1} \xleftarrow{\$} \mathsf{KeyGen}$	$\mathcal{O}_b^{(1)}$	$\mathcal{O}_b^{(2)}$	$\mathcal{O}_b^{(3)}$	$\mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}$

Figure 7: Interpolating Games

Proof of Corollary 20. For any adversary $\mathcal{A}$, consider the adversary $\mathcal{B}^{\mathcal{O}}$ which constructs game $\alpha_{\Gamma_1,\Gamma_2}$ from its $\mathsf{LSN}_{\varepsilon,\delta}$ oracle (where $\delta = \frac{1}{2}$ when $b = 0$, and $\delta = \varepsilon$, otherwise) as follows. $\mathcal{B}^{\mathcal{O}}$ uses $\mathcal{O}_b^{(i)}$ from Lemma 10 with $\ell = 3$. $\mathcal{B}^{\mathcal{O}}$ then constructs a game using $\mathbf{X}_j \xleftarrow{\$} \mathsf{KeyGen}$ as both the Phase I and Phase II secret, generating $\mathsf{Ra}()$ from $\mathcal{O}_b^{(1)}$ and $\mathsf{Tb}()$ from $\mathcal{O}_b^{(2)}$, and Ra^* from $\mathcal{O}_b^{(3)}$. It then runs $\mathcal{A}$, returning 1 if $\mathcal{A}$ is accepted (i.e. $|\mathbf{z}^* \oplus \mathbf{w}^*| \leq k \cdot \mathsf{u}_{\mathsf{HB^N}}(\varepsilon)$), and 0 otherwise. Then it follows by construction and Lemma 17 that $\forall(\Gamma_1,\Gamma_2) \in \{(\mathrm{A}_{0,0},\mathrm{A}_{1,0}),(\mathrm{A}_{2,j},\mathrm{A}_{3,j}),(\mathrm{A}_{3,j},\mathrm{A}_{1,j+1})\}$ and $\forall\mathcal{A}$,

$$\begin{aligned}|\mathsf{Adv}_{\mathcal{A}}^{\Gamma_1} - \mathsf{Adv}_{\mathcal{A}}^{\Gamma_2}| &= \mathsf{Adv}_{\mathcal{B}}^{\mathsf{LSN}} \\ &\leq \varepsilon_{\mathsf{LPN}}\end{aligned}$$

□

5.3 Key Switch: the Technical Details

Next, we prove Corollary 21. Corollary 21 is in some sense the core of the security proof: it allows us to change the Phase I and Phase II keys so that they differ by a rank 1 matrix, while the protocol remains indistinguishable from the real protocol. Its proof is based on the following technical lemma.

Lemma 23. *Given any $(\mathbf{Y}_0,\mathbf{x},\mathbf{y}) \in \mathbb{F}_2^{n\times n} \times \mathbb{F}_2^n \times \mathbb{F}_2^n$, let $\mathbf{Y}_1 = \mathbf{Y}_0 \oplus \mathbf{x}\mathbf{y}^\top$. For any $\mathbf{b}' \in \mathbb{F}_2^n$, and for $(\mathbf{a},e)$ sampled according to $\mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{x}}$, the random variables $W_0 \doteq \mathsf{wz}^{\mathbf{Y}_0}(\mathbf{a},\mathbf{b}',e)$ and $W_1 \doteq \mathsf{wz}^{\mathbf{Y}_1}(\mathbf{a},\mathbf{b}',e)$ induced by $(\mathbf{a},e)$ are identically distributed.*

Proof. Define the random variables $G_{\hat{a}}$ with distribution $\mathsf{Ber}_{p_{1|\hat{a}}^{\varepsilon,\varepsilon}}$. Recall from Corollary 15 that $G_{\hat{a}}$ describes the marginal distribution on e conditioned on $\mathbf{a}^\top\mathbf{x} = \hat{a}$, where $(\mathbf{a},e) \xleftarrow{\$} \mathsf{LSN}_{\varepsilon,\varepsilon}^{\mathbf{x}}$. We can then write

$$\begin{align}
\mathsf{wz}^{\mathbf{Y}_1}(\mathbf{a}, \mathbf{b}', e) &= \mathbf{a}^{\top}\mathbf{X}_1\mathbf{b}' \oplus e \notag\\
&\equiv \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus \mathbf{a}^{\top}\mathbf{x}\mathbf{y}^{\top}\mathbf{b}' \oplus G_{\mathbf{a}^{\top}\mathbf{x}} \tag{13}\\
&= \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus \begin{cases} G_0 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 0 \\ \mathbf{y}^{\top}\mathbf{b}' \oplus G_1 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 1 \end{cases} \notag\\
&\equiv \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus \begin{cases} G_0 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 0 \\ \mathbf{y}^{\top}\mathbf{b}' \oplus \mathsf{Ber}_{\frac{1}{2}} & \text{if } \mathbf{a}^{\top}\mathbf{x} = 1 \end{cases} \tag{14}\\
&\equiv \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus \begin{cases} G_0 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 0 \\ \mathsf{Ber}_{\frac{1}{2}} & \text{if } \mathbf{a}^{\top}\mathbf{x} = 1 \end{cases} \tag{15}\\
&\equiv \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus \begin{cases} G_0 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 0 \\ G_1 & \text{if } \mathbf{a}^{\top}\mathbf{x} = 1 \end{cases} \tag{16}\\
&= \mathbf{a}^{\top}\mathbf{X}_0\mathbf{b}' \oplus G_{\mathbf{a}^{\top}\mathbf{x}} \notag\\
&\equiv \mathsf{wz}^{\mathbf{Y}_0}(\mathbf{a}, \mathbf{b}', e) \notag
\end{align}$$

Equation 13 follows from conditioning on $\mathbf{a}^{\top}\mathbf{x}$. Equations 14 and 16 follow from Corollary 16. Equation 15 follows from Fact 2. □

Next, we apply the lemma to Phase I and Phase II.

Corollary 24. *When* $\mathsf{Ra} = \mathsf{LSN}^{\mathbf{r}^{(j)}}_{\varepsilon,\varepsilon}$*,* $\mathsf{Tb} = \mathsf{LSN}^{\mathbf{s}^{(j)}}_{\varepsilon,\varepsilon}$*,* $\mathsf{Ra}^* = \mathsf{LSN}^{\mathbf{x}}_{\varepsilon,\varepsilon}$*, and* $\mathbf{r}^{(j)} = \mathbf{t}^{(j)} \oplus \mathbf{x}$*, the games constructed from Phase I and II key pairs* $(\mathbf{X}, \mathbf{Y})$ *and* $(\mathbf{X} \oplus \mathbf{r}^{(j)}\mathbf{s}^{(j)^{\top}}, \mathbf{Y} \oplus \mathbf{x}\mathbf{s}^{(j)^{\top}})$ *are indistinguishable.*

Proof of Corollary 24. The proof follows from three applications of Lemma 23: two for Rw and Tz in Phase I, and one for Rw^* in Phase II.

$$\begin{align}
\mathsf{Rw}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) &\equiv \mathsf{wz}^{\mathbf{X}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) \notag\\
&\equiv \mathsf{wz}^{\mathbf{X}\oplus\mathbf{r}^{(j)}\mathbf{s}^{(j)^{\top}}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) \tag{17}\\
\mathsf{Tz}^{\mathbf{X}}(\mathbf{a}'^{(i)}, \mathbf{b}^{(i)}, \mathbf{e}_i) &\equiv \mathsf{wz}^{\mathbf{X}^{\top}}(\mathbf{b}^{(i)}, \mathbf{a}'^{(i)}, \mathbf{e}_i) \notag\\
&\equiv \mathsf{wz}^{\mathbf{X}^{\top}\oplus\mathbf{s}^{(j)}\mathbf{r}^{(j)^{\top}}}(\mathbf{b}^{(i)}, \mathbf{a}'^{(i)}, \mathbf{e}_i) \tag{18}\\
&\equiv \mathsf{wz}^{\mathbf{X}\oplus\mathbf{r}^{(j)}\mathbf{s}^{(j)^{\top}}}(\mathbf{a}'^{(i)}, \mathbf{b}^{(i)}, \mathbf{e}_i) \tag{19}\\
\mathsf{Rw}^{*\mathbf{Y}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) &\equiv \mathsf{wz}^{\mathbf{Y}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) \notag\\
&\equiv \mathsf{wz}^{\mathbf{Y}\oplus\mathbf{x}\mathbf{s}^{(j)^{\top}}}(\mathbf{a}^{(i)}, \mathbf{b}'^{(i)}, \mathbf{f}_i) \tag{20}
\end{align}$$

Equation 17 follows from Lemma 23 applied to $(\mathbf{X}, \mathbf{r}^{(j)}, \mathbf{s}^{(j)})$. Equation 18 follows from Lemma 23 applied to $(\mathbf{X}^{\top}, \mathbf{s}^{(j)}, \mathbf{r}^{(j)})$. Equation 19 follows from the bilinearity of $\mathsf{Tz}()$. Equation 20 follows from Lemma 23 applied to $(\mathbf{Y}, \mathbf{x}, \mathbf{s}^{(j)})$. □

Finally, we need a result that the joint distribution obtained from Equations 17, 19, and 20 is equivalent to the distribution in $\mathrm{A}_{2,j}$.

Corollary 25. *With notation as in Figure 6*

$$(\mathbf{X}_{j+1}, \mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}) \equiv (\mathbf{X}_j \oplus \mathbf{r}^{(j)}\mathbf{s}^{(j)^{\top}}, \mathbf{X}_j \oplus \mathbf{T}_j \oplus \mathbf{x}\mathbf{s}^{(j)^{\top}})$$

Corollary 25 will follow from the following technical lemma.

Lemma 26. *Let* $(\mathbf{X}_1, \mathbf{X}_2) \xleftarrow{\$} D$*, and choose* $\mathbf{X} \xleftarrow{\$} U_{n\times n}$ *independently of* $(\mathbf{X}_1, \mathbf{X}_2)$*. Then the following distributions are equivalent:*

$$(\mathbf{X} \oplus \mathbf{X}_1, \mathbf{X} \oplus \mathbf{X}_2) \equiv (\mathbf{X}, \mathbf{X} \oplus \mathbf{X}_1 \oplus \mathbf{X}_2).$$

Lemma 26 establishes that the uniform distribution, when used as above, gives us a certain translation-invariance property which, in turn, will be used to hide whether our key switching happens in Phase I or Phase II.

Proof of Lemma 26.

$$\begin{align*}
\Pr[(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2) = (\mathbf{X} \oplus \mathbf{X}_1, \mathbf{X} \oplus \mathbf{X}_2)] &= \Pr[\mathbf{X} = \mathbf{X}_1 \oplus \hat{\mathbf{X}}_1 \wedge \mathbf{X} = \mathbf{X}_2 \oplus \hat{\mathbf{X}}_2)] \\
&= 2^{-n^2} \Pr[\mathbf{X}_1 \oplus \hat{\mathbf{X}}_1 = \mathbf{X}_2 \oplus \hat{\mathbf{X}}_2] \tag{21} \\
&= 2^{-n^2} \Pr[\mathbf{X}_1 \oplus \hat{\mathbf{X}}_1 \oplus \mathbf{X}_2 = \hat{\mathbf{X}}_2] \\
&= \Pr[\hat{\mathbf{X}}_1 = \mathbf{X} \wedge \hat{\mathbf{X}}_2 = \mathbf{X} \oplus \mathbf{X}_1 \oplus \mathbf{X}_2] \tag{22} \\
&= \Pr[(\hat{\mathbf{X}}_1, \hat{\mathbf{X}}_2) = (\mathbf{X}, \mathbf{X} \oplus \mathbf{X}_1 \oplus \mathbf{X}_2)]
\end{align*}$$

Equation 21 and Equation 22 follow from independence of $\mathbf{X}$ from $\mathbf{X}_1, \mathbf{X}_2$ respectively. □

Proof of Corollary 25.

$$\begin{align*}
(\mathbf{X}_j \oplus \mathbf{r}^{(j)} \mathbf{s}^{(j)^\top}, \mathbf{X}_j \oplus \mathbf{T}_j \oplus \mathbf{x}\mathbf{s}^{(j)^\top}) &= (\mathbf{X}_j \oplus (\mathbf{x} \oplus \mathbf{t}^{(j)}) \mathbf{s}^{(j)^\top}, \mathbf{X}_j \oplus \mathbf{T}_j \oplus \mathbf{x}\mathbf{s}^{(j)^\top}) \tag{23} \\
&\equiv (\mathbf{X}_j, \mathbf{X}_j \oplus \mathbf{T}_j \oplus \mathbf{t}^{(j)} \mathbf{s}^{(j)^\top}) \tag{24} \\
&= (\mathbf{X}_j, \mathbf{X}_j \oplus \mathbf{T}_{j+1}) \tag{25} \\
&= (\mathbf{X}_{j+1}, \mathbf{X}_{j+1} \oplus \mathbf{T}_{j+1}) \tag{26}
\end{align*}$$

Equation 23 follows from the definition of $\mathbf{r}^{(j)}$. Equation 24 follows from Lemma 26 applied to $D = \left\{ \mathbf{r}^{(j)} \mathbf{s}^{(j)^\top}, \mathbf{x}\mathbf{s}^{(j)^\top} \right\}$. Equation 25 follows from the definition of $\mathbf{T}_j$. Equation 26 follows from a simple relabeling. □

Proof of Corollary 21. Corollary 21 now follows immediately from Corollary 24 applied to $(\mathbf{X}_j, \mathbf{X}_j \oplus \mathbf{T}_j)$ and from Corollary 25. □

5.4 A Theorem for Products of Random Matrices

For a random $n \times k$ matrix $\mathbf{A}$, let $S_{\mathbf{A}}$ be the event that $\mathsf{rank}(\mathbf{A}) < k$. For a given $n \times k$ matrix $\hat{\mathbf{B}}$ (with no zero columns) and a random $n \times m$ matrix $\mathbf{S}$, let $T_{\hat{\mathbf{B}}^\top \mathbf{S}}$ be the event that some row of $\hat{\mathbf{B}}^\top \mathbf{S}$ is all zero, i.e. $\exists i$ such that $\mathbf{S}^\top \hat{\mathbf{B}} \mathbf{e}^{(i)} = \mathbf{0}^m$. The main result of this section is the following theorem.

Theorem 27. *Let* $\mathbf{A} \xleftarrow{\$} \mathbb{F}_2^{n \times k}$ *and* $\mathbf{R}, \mathbf{S} \xleftarrow{\$} \mathbb{F}_2^{n \times m}$*, for* $m \geq k$*. Given any* $\hat{\mathbf{B}} \in \mathbb{F}_2^{n \times k}$ *such that* $\forall i \in [k], \hat{\mathbf{B}} \mathbf{e}^{(i)} \neq \mathbf{0}^n$*, we have:*

(a) $\Pr[S_{\mathbf{A}}] \leq 2^{k-n}$*,* $\Pr[T_{\hat{\mathbf{B}}^\top \mathbf{S}}] \leq k 2^{-m}$

(b) $\forall \hat{\mathbf{z}} \in \mathbb{F}_2^k$*,* $\Pr[\mathsf{diag}(\mathbf{A}^\top \mathbf{R} \mathbf{S}^\top \hat{\mathbf{B}}) = \hat{\mathbf{z}} \mid \overline{S_{\mathbf{A}}}, \overline{T_{\hat{\mathbf{B}}^\top \mathbf{S}}}] = 2^{-k}$

Roughly speaking, Theorem 27 states that $\mathbf{A}$ and $\hat{\mathbf{B}}^\top \mathbf{S}$ are "degenerate" only with negligible probability, and if $\mathbf{A}, \hat{\mathbf{B}}^\top \mathbf{S}$ are nondegenerate, then $\mathsf{diag}(\mathbf{A}^\top \mathbf{R} \mathbf{S}^\top \hat{\mathbf{B}})$ is uniformly distributed.

Proof of Theorem 27(a). First consider $S_{\mathbf{A}}$. We find that

$$\begin{align*}
\Pr[\mathsf{rank}(\mathbf{A}) < k] &= \Pr[\exists \mathbf{x} \in \mathbb{F}_2^k \setminus \left\{ \mathbf{0}^k \right\} : \mathbf{A}\mathbf{x} = \mathbf{0}^n] \tag{27} \\
&\leq \sum_{\mathbf{x} \in \mathbb{F}_2^k \setminus \left\{ \mathbf{0}^k \right\}} \Pr[\mathbf{A}\mathbf{x} = \mathbf{0}^n] \tag{28} \\
&= \sum_{\mathbf{x} \in \mathbb{F}_2^k \setminus \left\{ \mathbf{0}^k \right\}} \prod_{i \in [n]} \Pr[(\mathbf{e}^{(i)^\top} \mathbf{A}) \mathbf{x} = 0] \tag{29} \\
&= (2^k - 1) \cdot \prod_{i \in [n]} \frac{2^{k-1}}{2^k} \tag{30} \\
&\leq 2^{k-n}
\end{align*}$$

Equation 27 and Equation 30 both follows from $\mathsf{rank}(\mathbf{M}) + \mathsf{rank}(\ker(\mathbf{M})) = k$, for $\mathbf{M} = \mathbf{A}, \mathbf{x}$ respectively. Equation 28 follows from the union bound. Equation 29 follows from independence of the rows $\mathbf{e}^{(i)^\top} \mathbf{A}$ of $\mathbf{A}$.

Next, we consider $T_{\hat{\mathbf{B}}^\top \mathbf{S}}$:

$$\Pr[\exists i \in [k], \mathbf{S}^\top \hat{\mathbf{B}} \mathbf{e}^{(i)} = \mathbf{0}^m] \leq \sum_{i \in [k]} \Pr[\mathbf{S}^\top \hat{\mathbf{B}} \mathbf{e}^{(i)} = \mathbf{0}^m] \tag{31}$$

$$= \sum_{i \in [k]} \prod_{j \in [m]} \Pr[{\mathbf{e}^{(j)}}^\top \mathbf{S} \hat{\mathbf{B}} \mathbf{e}^{(i)} = 0] \tag{32}$$

$$= \sum_{i \in [k]} \prod_{j \in [m]} \frac{1}{2}$$

$$= k 2^{-m}$$

Equation 31 follows from the union bound. Equation 32 follows from independence of the rows ${\mathbf{e}^{(j)}}^\top \mathbf{S}$ of $\mathbf{S}$. □

We move on to Theorem 27(b). We will need the following two lemmata.

Lemma 28. *Let* $\mathbf{R} \xleftarrow{\$} \mathbb{F}_2^{n \times m}$*. Then* $\forall \hat{\mathbf{A}} \in \mathbb{F}_2^{n \times k}$ *with* $\mathsf{rank}(\hat{\mathbf{A}}) = k \leq n$, $\forall \hat{\mathbf{Y}} \in \mathbb{F}_2^{k \times m}$, $\Pr[\hat{\mathbf{A}}^\top \mathbf{R} = \hat{\mathbf{Y}}] = 2^{-km}$.

Proof. Each column $\hat{\mathbf{Y}} \mathbf{e}^{(i)} = \hat{\mathbf{A}}^\top (\mathbf{R} \mathbf{e}^{(i)})$ is an independently random element of $\mathsf{Im}(\hat{\mathbf{A}}^\top)$. Since $\hat{\mathbf{A}}$ has full rank, $\mathsf{Im}(\hat{\mathbf{A}}^\top)$ contains all of $\mathbb{F}_2^k$, so that each column is a uniformly random k-bit vector. □

Lemma 29. *Let* $\mathbf{Y} \xleftarrow{\$} \mathbb{F}_2^{k \times m}$, $\mathbf{S} \xleftarrow{\$} \mathbb{F}_2^{n \times m}$*. Given any* $\hat{\mathbf{z}} \in \mathbb{F}_2^k$ *and* $\hat{\mathbf{B}} \in \mathbb{F}_2^{n \times k}$ *so that* $\forall i \in [k], \mathbf{S}^\top \hat{\mathbf{B}} \mathbf{e}^{(i)} \neq \mathbf{0}^n$,

$$\Pr[\mathsf{diag}(\mathbf{Y} \mathbf{S}^\top \hat{\mathbf{B}}) = \hat{\mathbf{z}}] = 2^{-k}$$

Proof. For all $i \in [k]$, let $\mathbf{y}^{(i)} = \mathbf{Y}^\top \mathbf{e}^{(i)}$ and $\mathbf{x}^{(i)} = \mathbf{S}^\top \hat{\mathbf{B}} \mathbf{e}^{(i)}$. Then

$$\Pr[\mathsf{diag}(\mathbf{Y} \mathbf{S}^\top \hat{\mathbf{B}}) = \hat{\mathbf{z}}] = \prod_{i=1}^{k} \Pr[{\mathbf{y}^{(i)}}^\top \mathbf{x}^{(i)} = \hat{\mathbf{z}}_i] \tag{33}$$

$$= \prod_{i=1}^{k} \frac{\left| \left\{ \mathbf{y}^{(i)} : {\mathbf{y}^{(i)}}^\top \mathbf{x}^{(i)} = \hat{\mathbf{z}}_i \right\} \right|}{|\mathbb{F}_2^n|} \tag{34}$$

$$= \prod_{i=1}^{k} \frac{2^{n-1}}{2^n} \tag{35}$$

$$= 2^{-k}$$

Equation 33 follows from expressing the diagonal of the product $\mathbf{Y} \mathbf{S}^\top \hat{\mathbf{B}}$ in terms of $\mathbf{Y}$ and $\mathbf{S}^\top \hat{\mathbf{B}}$. Equation 34 follows from independence of the $\mathbf{y}^{(i)}$. Equation 35 follows from $|\ker(\mathbf{x}^{(i)})| = 2^{n-1} = |\mathbb{F}_2^n \setminus \ker(\mathbf{x}^{(i)})|$ for $\mathbf{x}^{(i)} \neq \mathbf{0}^k$. □

Theorem 27(b) now follows immediately from Lemma 28 and Lemma 29.

5.5 Proof of Corollary 22

We use Theorem 27 to prove Corollary 22, which states that the adversary in $\mathrm{A}_{3,m-1}$ cannot do non-negligibly better than randomly guessing.

Proof Corollary 22. Let $\mathbf{R}, \mathbf{S}$ be the matrices formed by taking $\mathbf{r}^{(j)}, \mathbf{s}^{(j)}$ as columns respectively: $\forall j \in [m], \mathbf{R} \mathbf{e}^{(j)} = \mathbf{r}^{(j)}, \mathbf{S} \mathbf{e}^{(j)} = \mathbf{s}^{(j)}$. Then

$$\begin{aligned} \sum_{j=1}^{m} \mathbf{r}^{(j)} {\mathbf{s}^{(j)}}^\top &= \sum_{j=1}^{m} (\mathbf{R} \mathbf{e}^{(j)}) (\mathbf{S} \mathbf{e}^{(j)})^\top \\ &= \mathbf{R} \left(\sum_{j=1}^{m} \mathbf{e}^{(j)} {\mathbf{e}^{(j)}}^\top \right) \mathbf{S}^\top \\ &= \mathbf{R} \mathbf{I}_m \mathbf{S}^\top \\ &= \mathbf{R} \mathbf{S}^\top \end{aligned}$$

Since $\mathbf{R}, \mathbf{S}$ are not used in Phase I, we can treat them as random variables, so that $\mathrm{A}_{3,m-1}$ now looks as follows:

$$\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}} = \left| \Pr \left[\begin{array}{l} \mathbf{X}_m \xleftarrow{\$} \mathsf{KeyGen}(1^n), \\ s \xleftarrow{\$} \mathcal{A}_1^{\mathcal{T}^{\mathbf{X}_m}, \mathcal{R}^{\mathbf{X}_m}}(1^n), \\ \hat{\mathbf{A}} \xleftarrow{\$} \mathbb{F}_2^{n \times k}, \\ (\mathbf{z}^*, \hat{\mathbf{B}}) \xleftarrow{\$} \mathcal{A}_2(s, \hat{\mathbf{A}}), \\ \mathbf{R}, \mathbf{S} \xleftarrow{\$} \mathbb{F}_2^{n \times m}, \\ \mathbf{f} \xleftarrow{\$} \mathsf{Ber}_\varepsilon^n, \\ \mathbf{w}^* = \mathsf{diag}(\hat{\mathbf{A}}^\top \mathbf{X}_m \hat{\mathbf{B}}) \oplus \mathsf{diag}(\hat{\mathbf{A}}^\top \mathbf{R} \mathbf{S}^\top \hat{\mathbf{B}}) \oplus \mathbf{f} \end{array} : \begin{array}{l} \forall i \in [k], \hat{\mathbf{B}}\mathbf{e}^{(i)} \neq \mathbf{0}^n, \\ |\mathbf{z}^* \oplus \mathbf{w}^*| \leq k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) \end{array} \right] \right| \tag{36}$$

For any vector $\hat{\mathbf{z}} \in \mathbb{F}_2^k$ of guesses made by the adversary, it follows from Theorem 27(a) that $\Pr[S_{\mathbf{A}} \vee T_{\hat{\mathbf{B}}^\top \mathbf{S}}] \leq 2^{k-n} + k2^{-m}$. It follows from Theorem 27(b) that conditioned on $\overline{S_{\mathbf{A}}} \wedge \overline{T_{\hat{\mathbf{B}}^\top \mathbf{S}}}$, the distribution of $\mathbf{w}^*$ in $\mathrm{A}_{3,m-1}$ obeys

$$\begin{aligned} \mathbf{w}^* &= \mathsf{diag}(\hat{\mathbf{A}}^\top \mathbf{X}_m \hat{\mathbf{B}}) \oplus \mathsf{diag}(\hat{\mathbf{A}} \mathbf{R} \mathbf{S}^\top \hat{\mathbf{B}}) \oplus \mathbf{f} \\ &= \mathsf{diag}(\hat{\mathbf{A}}^\top \mathbf{X}_m \hat{\mathbf{B}}) \oplus \mathsf{Ber}_{\frac{1}{2}}^k \oplus \mathbf{f} \\ &= \mathsf{Ber}_{\frac{1}{2}}^k \end{aligned} \tag{37}$$

Equation 37 follows from Fact 2. Since $\mathsf{Ber}_{\frac{1}{2}}^k$ is the distribution of $\mathbf{w}^*$ in $\mathrm{A}_{4,m-1}$, Corollary 22 follows immediately from Equation 37 and Theorem 27(a).

□

5.6 Soundness and Completeness

We now have all the ingredients required to prove Theorem 19 and Theorem 18, and to determine appropriate parameters to optimize soundness and completeness.

Proof of Theorem 19.

$$\begin{aligned} |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{0,0}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}}| \leq{}& |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{0,0}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{1,0}}| + \sum_{i=0}^{m-2} \left(+ |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{3,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{1,i+1}}| \right) \\ &+ \sum_{i=0}^{m-1} \left(|\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{1,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{2,i}}| + |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{2,i}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{3,i}}| \right) + |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{3,m-1}}| \qquad (38) \\ \leq{}& 2m \cdot \varepsilon_{\mathsf{LPN}} + 2^{k-n} + k2^{-m} \qquad (39) \end{aligned}$$

Equation 38 follows from the triangle inequality, and Equation 39 follows from Corollaries 20–22. □

In $\mathrm{A}_{4,m-1}$, since $\mathbf{w}^* \xleftarrow{\$} \mathbb{F}_2^k$, for any distribution of $\mathbf{z}^*$, the distribution $\mathbf{z}^* \oplus \mathbf{w}^*$ is uniformly random. Therefore

$$\begin{aligned} \Pr[|\mathbf{z}^* \oplus \mathbf{w}^*| \leq k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon)] &= \Pr[|\mathbf{w}^*| \leq k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon)] \\ &\leq 2^{-k\left(\left(\frac{1}{2} - (1+\delta)\right)(\varepsilon \oplus \varepsilon)\right)^2} \end{aligned} \tag{40}$$

Equation 40 follows from the well-known Hoeffding-Chernoff bound, $\Pr[X \leq (1-\mu) \cdot X] \leq e^{-\mu^2 k}$, for $X = \sum_{i=1}^k X_k$ with $X_i \in [0,1]$ for all $i \in [k]$. Recall that with $\mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) = (1+\delta)(\varepsilon \oplus \varepsilon)$, HB^N achieves completeness $e^{-\delta^2 k}$, i.e. an honest $\mathcal{T}$ fails with probability at most $e^{-\delta^2 k}$. If we set δ so that $\mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon) = (\varepsilon \oplus \varepsilon)(1+\delta) = \frac{1}{2}(1-\delta)$, we obtain the same bound of $e^{-\delta^2 k}$ for both soundness and completeness. $(\varepsilon \oplus \varepsilon)(1+\delta) = \frac{1}{2}(1-\delta)$ results in $\delta = \frac{\frac{1}{2} - (\varepsilon \oplus \varepsilon)}{\frac{1}{2} + (\varepsilon \oplus \varepsilon)} = \frac{1-4\varepsilon+4\varepsilon^2}{1+4\varepsilon-4\varepsilon^2}$. As a result, we obtain

$$\begin{aligned} \Pr[|\mathbf{z} \oplus \mathbf{w}| \leq k \cdot \mathsf{u}_{\mathsf{HB}^\mathsf{N}}(\varepsilon)] &\leq 2^{-k\delta^2} \\ &= 2^{-k\left(\frac{1-4\varepsilon+4\varepsilon^2}{1+4\varepsilon-4\varepsilon^2}\right)^2} \end{aligned} \tag{41}$$

If ε is a constant, for example, the bound is $2^{-O(k)}$, which is negligible for $k = \omega(\log n)$.

Proof of Theorem 18.

$$\mathsf{Adv}_{\mathcal{A}}^{\mathsf{HB}^{\mathsf{N}}} \leq |\mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}} - \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{0,0}}| + \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}} \tag{42}$$

$$\leq 2^{k-n} + k2^{-m} + 2m \cdot \varepsilon_{\mathsf{LPN}} + \mathsf{Adv}_{\mathcal{A}}^{\mathrm{A}_{4,m-1}} \tag{43}$$

$$\leq \left(2^{k-n} + k2^{-m} + 2^{-k\left(\frac{1-4\varepsilon+4\varepsilon^2}{1+4\varepsilon-4\varepsilon^2}\right)^2} \right) + 2m \cdot \varepsilon_{\mathsf{LPN}} \tag{44}$$

$$= \mathsf{negl} \tag{45}$$

Equation 42 follows from the triangle inequality. Equation 43 follows from Theorem 19. Equation 44 follows from Equation 41. Equation 45 follows from the LPN assumption and from setting $m = \omega(\log n)$, $k = \omega(\log n)$, $k = n - \omega(\log n)$, and $\varepsilon = \theta(1)$. □

6 Conclusion

We have introduced HB^{N}, a bilinear version of HB, and proven its security in the $\mathsf{Man\text{-}in\text{-}the\text{-}Middle}$ model. Along the way, we have introduced a new notation the simplifies working with random variables over $\mathbb{F}_2$, assembled a useful collection of lemmas for working with LPN, and introduced the LSN problem. Additionally, we have designed a new probabilistic verification procedure which is in this case symmetric to the probabilistic prover procedure. We hope that these technical tools will be useful for future work.

We are grateful to Eike Kiltz and David Cash for pointing out a gap in the proof of the version of Corollary 22 in a previous version of this work. We would also like to thank Krzysztof Pietrzak for useful discussions about LPN, and Miaomiao Zhang for helpful remarks on earlier drafts of this work.

References

[AL87] Dana Angluin and Philip D Laird. Learning from Noisy Examples. *Machine Learning*, 2(4):343–370, 1987.

[BC08] Julien Bringer and Herve Chabanne. Trusted-HB: a low-cost version of HB secure against man-in-the-middle attacks. *arXiv*, 2008.

[BCD06] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a lightweight authentication protocol secure against some attacks. In *Second International Workshop on Security, Privacy and Trust in Pervasive and Ubiquitous Computing (SecPerU 2006)*, pages 28–33. IEEE Computer Society, 2006.

[DK08] D Duc and Kwangjo Kim. Securing HB against GRS man-in-the-middle attack. *caislab.icu.ac.kr*, 2008.

[FS09] Dmitry Frumkin and Adi Shamir. Un-Trusted-HB: Security vulnerabilities of Trusted-HB. *EPrint*, 2009.

[GRS05] Henri Gilbert, Matthew Robshaw, and Herve Sibert. Active attack against HB^{+}: a provably secure lightweight authentication protocol. *Electronics Letters*, 2005.

[GRS08a] Henri Gilbert, Matthew Robshaw, and Yannick Seurin. Good variants of HB^{+} are hard to find. In *Proc. Financial Cryptography and Data Security*, pages 156–170, 2008.

[GRS08b] Henri Gilbert, Matthew Robshaw, and Yannick Seurin. $\mathrm{HB}^{\#}$: Increasing the security and efficiency of HB. In *Proc. EUROCRYPT*, volume 4965, pages 361–378, 2008.

[HB01] Nicholas Hopper and Manuel Blum. Secure human identification protocols. In *Proc. ASIACRYPT*, 2001.

[JW05] Ari Juels and Stephen Weis. Authenticating pervasive devices with human protocols. In *Proc. CRYPTO*, pages 293–308, 2005.

[Kea93] M. Kearns. Efficient noise-tolerant learning from statistical queries. In *Proceedings of the 25th ACM Symposium on Theory of Computing*, pages 392–401. ACM, 1993.

[KPC+11] Eike Kiltz, Krzystof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient Authentication from Hard Learning Problems. In *Proc. Eurocrypt*, pages 7–26, 2011.

[KSS10] Jonathan Katz, Ji Sun Shin, and Adam Smith. Parallel and concurrent security of the HB and HB^{+} protocols. *Journal of Cryptology*, 23(3):402–421, 2010.

[LMM08] X Leng, K Mayes, and K Markantonakis. HB-MP+ protocol: An improvement on the HB-MP protocol. *2008 IEEE International Conference on RFID*, 2008.

[MP07] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 2007.

[OOV08] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB$^{\#}$ against a man-in-the-middle attack. *Proc. ASIACRYPT*, 2008.

[Pie10] Krzystof Pietrzak. Subspace LWE, 2010. Manuscript available at `http://homepages.cwi.nl/~pietrzak/publications/SLWE.pdf`.

A Modeling the Active Security Game

The adversary $\mathcal{A}$ can be defined as two algorithms $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}_2)$. In Phase I, $\mathcal{A}_1$ has access to the Phase I oracles $\mathcal{T}, \mathcal{R}$, and outputs its state s for input to $\mathcal{A}_2$. $\mathcal{A}_2$ submits $\mathbf{b}^{*(i)}$ to the Phase II challenger $\mathcal{R}^*$ (either in parallel or in serial) and receives $\mathbf{a}^{*(i)}$ in exchange, as shown in Figure 8. From the model, we see that the reason HB^{N} can be used in either two or three rounds is precisely because the computation of $\mathbf{b}^{(i)}$ does not depend on $\mathbf{a}'^{(i)}$, and $\mathbf{a}^{(i)}$ does not depend on $\mathbf{b}'^{(i)}$.

Phase I	$s \overset{\$}{\leftarrow} \mathcal{A}_1^{\mathcal{T},\mathcal{R}}$,
Phase II	$\mathbf{b}^{*(i)} \overset{\$}{\leftarrow} \mathcal{A}_2(s)$,
In serial or parallel	$\mathbf{a}^{*(i)} \overset{\$}{\leftarrow} \mathcal{R}^*(\mathbf{b}^{*(i)})$,
Phase II: Final	$\mathbf{z}_i^* \overset{\$}{\leftarrow} \mathcal{A}_2(\mathbf{b}^{*(i)}, \mathbf{a}^{*(i)}, s)$

Two round	Three Round
$\mathbf{b}^{(i)} \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{b}}(\mathbf{a}'^{(i)})$	$\mathbf{b}^{(i)} \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{b}}()$
$\mathbf{a}^{(i)} \overset{\$}{\leftarrow} \mathcal{R}^{\mathbf{a}}()$	$\mathbf{a}^{(i)} \overset{\$}{\leftarrow} \mathcal{R}^{\mathbf{a}}(\mathbf{b}'^{(i)})$
$\mathbf{z}_i \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{z}}(\mathbf{a}'^{(i)}, \mathbf{b}^{(i)})$	$\mathbf{z}_i \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{z}}(\mathbf{a}'^{(i)}, \mathbf{b}^{(i)})$
$\mathbf{w}_i \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{w}}(\mathbf{z}'_i, \mathbf{a}^{(i)}, \mathbf{b}'^{(i)})$	$\mathbf{w}_i \overset{\$}{\leftarrow} \mathcal{T}^{\mathbf{w}}(\mathbf{z}'_i, \mathbf{a}^{(i)}, \mathbf{b}'^{(i)})$

Figure 8: Modeling the Oracles in Two and Three Rounds