Efficient Authentication from Hard Learning Problems*

Eike Kiltz[1**], Krzysztof Pietrzak[2***],
David Cash[3†], Abhishek Jain[4†], and Daniele Venturi[5†]

[1] RU Bochum
[2] CWI Amsterdam
[3] UC San Diego
[4] UC Los Angeles
[5] Sapienza University of Rome

Abstract. We construct efficient authentication protocols and message-authentication codes (MACs) whose security can be reduced to the learning parity with noise (LPN) problem.
Despite a large body of work – starting with the HB protocol of Hopper and Blum in 2001 – until now it was not even known how to construct an efficient authentication protocol from LPN which is secure against man-in-the-middle (MIM) attacks. A MAC implies such a (two-round) protocol.

1 Introduction

Authentication is among the most basic and important cryptographic tasks. In the present paper we construct efficient (secret-key) authentication schemes from the *learning parity with noise* (LPN) problem. We construct the first efficient message authentication codes (MACs) from LPN, but also simpler and more efficient two-round authentication protocols that achieve a notion called active security. Prior to our work, the only known way to construct an LPN-based MAC was via a relatively inefficient generic transformation [16] (that works with any pseudorandom generator), and all interactive LPN-based protocols with security properties similar to our new protocol required at least three rounds and had a loose security reduction. Our constructions and techniques diverge significantly from prior work in the area and will hopefully be of independent interest.

The pursuit of LPN-based authentication is motivated by two disjoint concerns, one theoretical and one practical. On the theoretical side, the LPN problem provides an attractive basis for provable security [2, 3, 5, 21, 17, 26]. It is

* This is a preliminary full version of a Eurocrypt 2011 paper.
** Funded by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation and the German Federal Ministry for Education and Research.
*** Supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Starting Grant (259668-PSPC)
† Research done while visiting CWI Amsterdam

closely related to the well-studied problem of decoding random linear codes, and unlike most number-theoretic problems used in cryptography, the LPN problem does not succumb to known quantum algorithms. On the practical side, LPN-based authentication schemes are strikingly efficient, requiring relatively few bit-level operations. Indeed, in their original proposal, Hopper and Blum [17] suggested that humans could perform the computation in their provably-secure scheme, even with realistic parameters. The efficiency of LPN-based schemes also makes them suitable for weak devices like RFID tags, where even evaluating a blockcipher may be prohibitive.

Each of our theoretical and practical motivations, on its own, would be sufficiently interesting for investigation, but together the combination is particularly compelling. LPN-based authentication is able to provide a theoretical improvement in terms of provable security *in addition to* providing better efficiency than approaches based on more classical symmetric techniques that are not related to hard problems. Usually we trade one benefit for the other, but here we hope to get the best of both worlds.

Before describing our contributions in more detail, we start by recalling authentication protocols, the LPN problem, and some of the prior work on which we build.

Authentication protocols. An authentication protocol is a (shared-key) protocol where a prover $\mathcal{P}$ authenticates itself to a verifier $\mathcal{V}$ (in the context of RFID implementations, we think of $\mathcal{P}$ as the "tag" and $\mathcal{V}$ as the "reader"). We recall some of the common definitions for security against impersonation attacks. A *passive attack* proceeds in two phases, where in the first phase the adversary eavesdrops on several interactions between $\mathcal{P}$ and $\mathcal{V}$, and then attempts to cause $\mathcal{V}$ to accept in the second phase (where $\mathcal{P}$ is no longer available). In an *active attack*, the adversary is additionally allowed to interact with $\mathcal{P}$ in the first phase. The strongest and most realistic attack model is a *man-in-the-middle attack* (MIM), where the adversary can arbitrarily interact with $\mathcal{P}$ and $\mathcal{V}$ (with polynomially many concurrent executions allowed) in the first phase.

The LPN problem. Briefly stated, the LPN problem is to distinguish from random several "noisy inner products" of random binary vectors with a random secret vector.

More formally, for $\tau < 1/2$ and a vector $\mathbf{x} \in \mathbb{Z}_2^\ell$, define the distribution $\Lambda_{\tau,\ell}(\mathbf{x})$ on $\mathbb{Z}_2^\ell \times \mathbb{Z}_2$ by $(\mathbf{r}, \mathbf{r}^\mathsf{T}\mathbf{x} \oplus e)$, where $\mathbf{r} \in \mathbb{Z}_2^\ell$ is uniformly random and $e \in \mathbb{Z}_2$ is selected according to Ber_τ, the Bernoulli distribution over $\mathbb{Z}_2$ with parameter τ (i.e. $\Pr[e = 1] = \tau$). The $\mathsf{LPN}_{\tau,\ell}$ problem is to distinguish an oracle returning samples from $\Lambda_{\tau,\ell}(\mathbf{x})$, where $\mathbf{x} \in \mathbb{Z}_2^\ell$ is random and fixed, from an oracle returning uniform samples. It was shown by Blum et al. [3] that this is equivalent to the search version of LPN, where one needs to compute $\mathbf{x}$ given oracle access to $\Lambda_{\tau,\ell}(\mathbf{x})$ (cf. [20, Thm.2] for precise bounds). We note that the search and decision variants are solvable with a linear in ℓ number of samples when there is no noise, i.e. when $\tau = 0$, and the best algorithms take time $2^{\ell/\log \ell}$ when $\tau > 0$ is treated as a constant [4, 5, 22].

Authentication protocols from LPN. Starting with the work of Hopper and Blum [17], several authentication protocols based on the LPN problem have been proposed. Their original elegant protocol is simple enough for us to recall right away. The shared secret key is a binary vector $\mathbf{s} \in \mathbb{Z}_2^\ell$. The interaction consists of two messages. First $\mathcal{V}$ sends a random challenge $\mathbf{r} \in \mathbb{Z}_2^\ell$, and then $\mathcal{P}$ answers with the bit $z = \mathbf{r}^\mathsf{T}\mathbf{s} \oplus e$, where $e \in \mathbb{Z}_2$ is sampled according to Ber_τ. Finally, the verifier accepts if $z = \mathbf{r}^\mathsf{T}\mathbf{s}$.

This basic protocol has a large completeness error τ (as $\mathcal{V}$ will reject if $e = 1$) and soundness error $1/2$ (as a random $\mathbf{r}$, z satisfies $\mathbf{r}^\mathsf{T} \cdot \mathbf{s} = z$ with probability $1/2$). This can be reduced via sequential or parallel composition. The parallel variant, denoted HB, is illustrated in Figure 1 (we represent several $\mathbf{r}$ with a matrix $\mathbf{R}$ and the noise bits are now arranged in a vector $\mathbf{e}$). The verifier accepts if at least a τ' fraction (where $\tau < \tau' < 1/2$) of the n basic authentication steps are correct.

The 2-round HB protocol is provably secure against passive attacks, but efficient active attacks are known against it. This is unsatisfying because in several scenarios, and especially in RFID applications, an adversary *will* be able to mount an active attack. Subsequently, Juels and Weis [18] proposed an efficient 3 round variant of HB, called HB^+, and proved it secure against active attacks. Again the error can be reduced by sequential repetition, and as shown by Katz, Shin and Smith via a non-trivial analysis, parallel repetition works as well [19, 20]. The protocol (in its parallel repetition variant) is illustrated in Figure 2.

Despite a large body of subsequent work[6] no improvements in terms of round complexity, security or tightness of the reduction over HB^+ were achieved: 3 round protocols achieving active security $\sqrt{\varepsilon}$ (assuming LPN is ε-hard) are the state of the art. In particular, Gilbert et al. [13] showed that HB^+ can be broken by a MIM attack. Several variants HB^{++} [8], HB^* [10], $\mathsf{HB\text{-}MP}$ [23] were proposed to prevent the particular attack from [13], but all of them were later shown to be insecure [14]. In [15], a variant $\mathsf{HB}^\#$ was presented which provably resists the particular attack from [13], but was shown susceptible to a more general MIM attack [24].

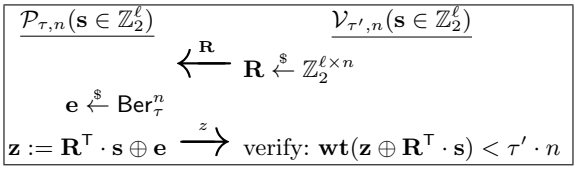

Fig. 1. The HB protocol, secure against passive attacks.

[6] cf. http://www.ecrypt.eu.org/lightweight/index.php/HB for an incomplete list of relevant papers.

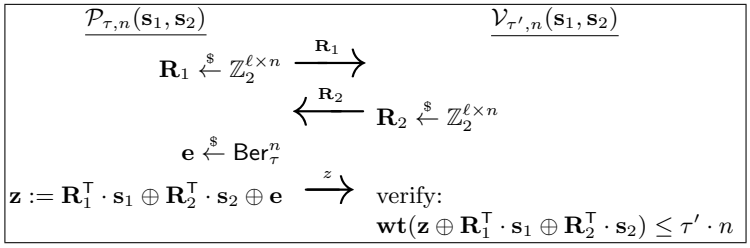

Fig. 2. The HB^+ protocol, secure against active attacks.

1.1 Our Contribution

We provide new constructions of authentication protocols and even MACs from LPN. Our first contribution is a *two*-round authentication protocol secure against active adversaries (this is mentioned as an open problem in [18]) which moreover has a tight security reduction (an open problem mentioned in [20]). As a second contribution, we build two efficient MACs, and thus also get two-round authentication protocols secure against MIM attacks, from the LPN assumption. Unlike previous proposals, our constructions are not ad-hoc, and we give a reduction to the LPN problem. Our authentication protocol is roughly as efficient as the HB^+ protocol but has twice the key length. Our MACs perform roughly the same computation as the authentication protocol plus one evaluation of a pairwise independent permutation of an $\approx 2\ell$ bit domain, where ℓ is the length of an LPN secret.

2-Round Authentication with Active Security. Our first contribution is a two-round authentication protocol which we prove secure against *active* attacks assuming the hardness of the LPN problem. Our protocol diverges considerably from all previous HB-type protocols [17, 18, 20, 15], and runs counter to the intuition that the only way to efficiently embed the LPN problem into a two-round protocol is via an HB-type construction.

We now sketch our protocol. In HB and its two-round variants, the prover must compute LPN samples of the form $\mathbf{R}^\mathsf{T} \cdot \mathbf{s} \oplus \mathbf{e}$, where $\mathbf{R}$ is the challenge chosen by the verifier in the first message. We take a different approach. Instead of sending $\mathbf{R}$, we now let the verifier choose a random subset of the bits of $\mathbf{s}$ to act as the "session-key" for this interaction. It represents this subset by sending a binary vector $\mathbf{v} \in \mathbb{Z}_2^\ell$ that acts as a "bit selector" of the secret $\mathbf{s}$, and we write $\mathbf{s}_{\downarrow \mathbf{v}}$ for the sub-vector of $\mathbf{s}$ which is obtained by deleting all bits from $\mathbf{s}$ where $\mathbf{v}$ is 0. (E.g. if $\mathbf{s} = 111000, \mathbf{v} = 011100$ then $\mathbf{s}_{\downarrow \mathbf{v}} = 110$.) The prover then picks $\mathbf{R}$ by *itself* and computes noisy inner products of the form $\mathbf{R}^\mathsf{T} \cdot \mathbf{s}_{\downarrow \mathbf{v}} \oplus \mathbf{e}$. Curiously, allowing the verifier to choose which bits of $\mathbf{s}$ to use in each session is sufficient to prevent active attacks. We only need to add a few sanity-checks that no pathological $\mathbf{v}$ or $\mathbf{R}$ were sent by an active adversary.

Our proof relies on the recently introduced *subspace LPN problem* [25]. In contrast to the active-attack security proof of HB^+ [20], our proof does not use

any rewinding techniques. Avoiding rewinding has at least two advantages. First, the security reduction becomes tight. Second, the proofs also works in a quantum setting: our protocol is secure against quantum adversaries assuming LPN is secure against such adversaries. As first observed by van de Graaf [28], classical proofs using rewinding in general do not translate to the quantum setting (cf. [30] for a more recent discussion). Let us emphasise that this only means that there is no security proof for HB^+ in the quantum setting, but we do not know if a quantum attack actually exists.

MAC & Man-In-The-Middle Security. In Section 4, we give two constructions of message authentication codes (MACs) that are secure (formally, unforgeable under chosen message attacks) assuming that the LPN problem is hard. Note that a MAC implies a two-round MIM-secure authentication protocol: the verifier chooses a random message as challenge, and the prover returns the MAC on the message.

As a first attempt, let us try to view our authentication protocol as a MAC. That is, a MAC tag is of the form $\phi = (\mathbf{R}, \mathbf{z} = \mathbf{R}^\mathsf{T} \cdot f_\mathbf{s}(\mathbf{m}) \oplus \mathbf{e})$, where the secret key derivation function $f_\mathbf{s}(\mathbf{m}) \in \mathbb{Z}_2^\ell$ first uniquely encodes the message $\mathbf{m}$ into $\mathbf{v} \in \mathbb{Z}_2^{2\ell}$ of weight ℓ and then returns $\mathbf{s}_{\downarrow\mathbf{v}}$ by selecting ℓ bits from secret $\mathbf{s}$, according to $\mathbf{v}$. However, this MAC is not secure: given a MAC tag $\phi = (\mathbf{R}, \mathbf{z})$ an adversary can ask verification queries where it sets individual rows of $\mathbf{R}$ to zero until verification fails: if the last row set to zero was the ith, then the ith bit of $f_\mathbf{s}(\mathbf{m})$ must be 1. (In fact, the main technical difficulty to build a secure MAC from LPN is to make sure the secret $\mathbf{s}$ does not leak from verification queries.) Our solution is to randomize the mapping f, i.e. use $f_\mathbf{s}(\mathbf{m}, \mathbf{b})$ for some randomness $\mathbf{b}$ and compute the tag as $\phi = \pi(\mathbf{R}, \mathbf{R}^\mathsf{T} \cdot f_\mathbf{s}(\mathbf{m}, \mathbf{b}) \oplus \mathbf{e}, \mathbf{b})$, where π is a pairwise independent permutation (contained in the secret key). We can prove that if LPN is hard then this construction yields a secure MAC. (The key argument is that, with high probability, all non-trivial verification queries are inconsistent and hence lead to reject.) However, the security reduction to the LPN problem is quite loose since it has to guess the value $\mathbf{v}$ from the adversary's forgery. (In the context of identity-based encryption (IBE) a similar idea has been used to go from selective-ID to full security using "complexity leveraging" [6].) In our case, however, this still leads to a polynomial security reduction when one commits to the hardness of the LPN problem at the time of the construction. (See the first paragraph of §4 for a discussion.)

To get a strictly polynomial security reduction (without having to commit to the hardness of the LPN problem), in our second construction we adapt a technique originally used by Waters [29] in the context of IBE schemes that has been applied to lattice based signature [7] and encryption schemes [1]. Concretely, we instantiate the above MAC construction with a different secret key derivation function $f_\mathbf{s}(\mathbf{m}, \mathbf{b}) = \mathbf{s}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{s}_i$ (where $\mathbf{v} = h(\mathbf{m}, \mathbf{b})$ and $h(\cdot)$ is a pairwise independent hash). The drawback of our second construction is the larger keysize. Our security reduction uses a technique from [7, 1] based on encodings with full-rank differences (FRD) by Cramer and Damgard [9].

1.2 Efficiency

Construction	Security	Complexity		Key-size	Reduction
		Communication	Computation		
HB [17]	passive (2 rnd)	$\ell \cdot n/c$	$\Theta(\ell \cdot n)$	$\ell \cdot c$	ε (tight)
HB^+ [18]	active (3 rnd)	$\ell \cdot n \cdot 2/$ c	$\Theta(\ell \cdot n)$	$\ell \cdot 2 \cdot c$	$\sqrt{\varepsilon}$
AUTH § 3	active (2 rnd)	$\ell \cdot n \cdot 2.1/c$	$\Theta(\ell \cdot n)$	$\ell \cdot 4.2 \cdot c$	ε (tight)
MAC_1 § 4.1	MAC $\rightarrow$ MIM (2 rnd)	$\ell \cdot n \cdot 2.1/c$	$\Theta(\ell \cdot n) + \mathsf{PIP}$	$\ell \cdot 12.6 \cdot c$	$\sqrt{\varepsilon} \cdot Q$ $(\star)$
MAC_2 § 4.2	MAC $\rightarrow$ MIM (2 rnd)	$\ell \cdot n \cdot 1.1/c$	$\Theta(\ell \cdot n) + \mathsf{PIP}$	$\ell \cdot \lambda \cdot c$	$\varepsilon \cdot Q$
GGM [16]	PRF $\rightarrow$ MIM (2 rnd)	λ	$\Theta(\ell^2 \cdot \lambda)$	$\Theta(\ell)$	$\varepsilon \cdot \lambda$

Fig. 3. A comparison of our new authentication protocol and MACs with the HB, HB^+ protocols and the classical GGM construction. The trade-off parameter $c, 1 \leq c \leq n$ and the term PIP will be explained in the "Communication vs. Key-Size" paragraph below. $(\star)$ See discussion in §4.

Figure 3 gives a rough comparison of our new protocol and MACs with the $\mathsf{HB}, \mathsf{HB}^+$ protocols and, as a reference, also the classical tree-based GGM construction [16]. The second row in the table specifies the security notion that is (provably) achieved under the $\mathsf{LPN}_{\tau,\ell}$ assumption. λ is a security parameter and n denotes the number of "repetitions". Typical parameters can be $\ell = 500, \lambda = 80, n = 250$. Computation complexity counts the number of binary operations over $\mathbb{F}_2$. Communication complexity counts the total length of all exchanged messages.[7] The last row in the table states the tightness of the security reduction, i.e. what exact security is achieved (ignoring constants and higher order terms) assuming the $\mathsf{LPN}_{\tau,\ell}$ problem is ε-hard.

The prover and verifier in the $\mathsf{HB}, \mathsf{HB}^+$ and our new protocols have to perform $\Theta(\ell \cdot n)$ basic binary operations, assuming the $\mathsf{LPN}_{\tau,\ell}$ problem (i.e., LPN with secrets of length ℓ) is hard. This seems optimal, as $\Theta(\ell)$ operations are necessary to compute the inner product which generates a single pseudorandom bit. We will thus consider an authentication protocol or MAC *efficient*, if it requires $O(\ell \cdot n)$ binary operations. Let us mention that one gets a length-doubling PRG under the $\mathsf{LPN}_{\tau,\ell}$ assumption with $\Theta(\ell^2)$ binary operations [11]. Via the classical GGM construction [16], we obtain a PRF and hence a MAC. This PRF, however, requires $\Theta(\ell^2 \cdot \lambda)$ operations per invocation (where λ is the size of the domain of the PRF) which is not very practical. (Recall that $\ell \approx 500$.)

Communication vs. Key-Size. As we will discuss in Appendix A.1, For all constructions except GGM, there is a natural trade-off between communication and key-size, where for any constant c $(1 \leq c \leq n)$, we can decrease communication by a factor of c and increase key-size by the factor c. For the first

[7] For MACs, we consider the communication one incurs by constructing a MIM secure 2-round protocol from the MAC by having the prover compute the tag on a random challenge message.

three protocols in the table, the choice of c does not affect the computational efficiency, but it does so for our MACs: to compute or verify a tag one has to evaluate a pairwise independent permutation (PIP) on the entire tag of length $m := \Theta(\ell \cdot n/c)$.

The standard way to construct a PIP π over $\mathbb{Z}_{2^m}$ is to define $\pi(x) := a \cdot x + b \in \mathbb{F}_{2^m}$ for random $a, b \in \mathbb{F}_{2^m}$. Thus the computational cost of evaluating the PIP is one multiplication of two m bits values: the PIP term in the table accounts for this complexity. Asymptotically, such a multiplication takes only $O(m \log m \log\log m)$ time [27, 12], but for small m (like in our scheme) this will not be faster than using schoolbook multiplication, which takes $\Theta(m^2)$ time. For parameters $\ell = 500, n = 250$ and trade-off $c = n$ (which minimizes the tag-length m) we get $m \approx 1200$ for MAC_1 (i.e., $1200 = 2\ell$ plus some statistical security parameters) and $m \approx 600$ for MAC_2. Hence, depending on the parameters, the evaluation of the PIP may be the computational bottleneck of our MACs.

2 Definitions

2.1 Notation

We denote the set of integers modulo an integer $q \geq 1$ by $\mathbb{Z}_q$. We will use normal, bold and capital bold letters like x, $\mathbf{x}$, $\mathbf{X}$ to denote single elements, vectors and matrices over $\mathbb{Z}_q$, respectively. For a positive integer k, $[k]$ denotes the set $\{1, \ldots, k\}$; $[0]$ is the empty set. For $a, b \in \mathbb{R}$, $]a, b[= \{x \in \mathbb{R} \ ; \ a < x < b\}$. For a vector $\mathbf{x} \in \mathbb{Z}_q^m$, $|\mathbf{x}| = m$ denotes the length of $\mathbf{x}$; $\mathbf{wt}(\mathbf{x})$ denotes the Hamming weight of the vector $\mathbf{x}$, i.e. the number of indices $i \in \{1, \ldots, |\mathbf{x}|\}$ where $\mathbf{x}[i] \neq 0$. The bit-wise XOR of two binary vectors $\mathbf{x}$ and $\mathbf{y}$ is represented as $\mathbf{z} = \mathbf{x} \oplus \mathbf{y}$, where $\mathbf{z}[i] = \mathbf{x}[i] \oplus \mathbf{y}[i]$. For $\mathbf{v} \in \mathbb{Z}_2^m$ we denote by $\overline{\mathbf{v}}$ its inverse, i.e. $\overline{\mathbf{v}}[i] = 1 - \mathbf{v}[i]$ for all i. For two vectors $\mathbf{v} \in \mathbb{Z}_2^\ell$ and $\mathbf{x} \in \mathbb{Z}_q^\ell$, we denote by $\mathbf{x}_{\downarrow \mathbf{v}}$ the vector (of length $\mathbf{wt}(\mathbf{v})$) which is derived from $\mathbf{x}$ by deleting all the bits $\mathbf{x}[i]$ where $\mathbf{v}[i] = 0$. If $\mathbf{X} \in \mathbb{Z}_2^{\ell \times m}$ is a matrix, then $\mathbf{X}_{\downarrow \mathbf{v}}$ denotes the submatrix we get by deleting the ith row if $\mathbf{v}[i] = 0$. A function in λ is *negligible*, written $\mathrm{negl}(\lambda)$, if it vanishes faster than the inverse of any polynomial in λ. An algorithm $\mathcal{A}$ is *probabilistic polynomial time* (PPT) if $\mathcal{A}$ uses some randomness as part of its logic (i.e. $\mathcal{A}$ is probabilistic) and for any input $\mathbf{x} \in \{0,1\}^*$ the computation of $\mathcal{A}(\mathbf{x})$ terminates in at most $\mathrm{poly}(|\mathbf{x}|)$ steps.

2.2 Authentication Protocols

An authentication protocol is an interactive protocol executed between a prover $\mathcal{P}$ and a verifier $\mathcal{V}$, both PPT algorithms. Both hold a secret $\mathbf{x}$ (generated using a key-generation algorithm KG executed on the security parameter λ in unary) that has been shared in an initial phase. After the execution of the authentication protocol, $\mathcal{V}$ outputs either accept or reject. We say that the protocol has completeness error α if for all secret keys $\mathbf{x}$ generated by $\mathsf{KG}(1^\lambda)$, the honestly executed protocol returns reject with probability at most α.

PASSIVE ATTACKS. An authentication protocol is secure against *passive* attacks, if there exists no PPT adversary $\mathcal{A}$ that can make the verifier return accept with non-negligible probability after (passively) observing any number of interactions between the verifier and prover.

ACTIVE ATTACKS. A stronger notion for authentication protocols is security against *active* attacks. Here the adversary $\mathcal{A}$ runs in two stages. First, she can interact with the honest prover a polynomial number of times (with concurrent executions allowed). In the second phase $\mathcal{A}$ interacts with the verifier only, and wins if the verifier returns accept. Here we only give the adversary one shot to convince the verifier.[8] An authentication protocol is (t, Q, ε)-*secure against active adversaries* if every PPT $\mathcal{A}$, running in time at most t and making Q queries to the honest prover, has probability at most ε to win the above game.

MAN-IN-THE-MIDDLE ATTACKS. The strongest standard security notion for authentication protocols is security against man-in-the-middle (MIM) attacks. Here the adversary can intially interact (concurrently) with any number of provers and – unlike in an active attacks – also verifiers. The adversary gets to learn the verifiers $\mathsf{accept}/\mathsf{reject}$ decisions. One can construct two-round authentication schemes which are secure against MIM attacks from basic cryptographic primitives like MACs, which we define next.

2.3 Message Authentication Codes

A message authentication code $\mathsf{MAC} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ is a triple of algorithms with associated key space $\mathcal{K}$, message space $\mathcal{M}$, and tag space $\mathcal{T}$.

- Key Generation. The probabilistic key-generation algorithm KG takes as input a security parameter $\lambda \in \mathbb{N}$ (in unary) and outputs a secret key $K \in \mathcal{K}$.
- Tagging. The probabilistic authentication algorithm TAG takes as input a secret key $K \in \mathcal{K}$ and a message $\mathbf{m} \in \mathcal{M}$ and outputs an authentication tag $\phi \in \mathcal{T}$.
- Verification. The deterministic verification algorithm VRFY takes as input a secret key $K \in \mathcal{K}$, a message $\mathbf{m} \in \mathcal{M}$ and a tag $\phi \in \mathcal{T}$ and outputs $\{\mathsf{accept}, \mathsf{reject}\}$.

If the TAG algorithm is deterministic one does not have to explicitly define VRFY, since it is already defined by the TAG algorithm as $\mathsf{VRFY}(K, \mathbf{m}, \phi) = \mathsf{accept}$ iff $\mathsf{TAG}(K, \mathbf{m}) = \phi$.

COMPLETENESS. We say that MAC has completeness error α if for all $\mathbf{m} \in \mathcal{M}$ and $\lambda \in \mathbb{N}$

$$\Pr[\mathsf{VRFY}(K, \mathbf{m}, \phi) = \mathsf{reject}\ ; K \leftarrow \mathsf{KG}(1^\lambda)\ ,\ \phi \leftarrow \mathsf{TAG}(K, \mathbf{m})] \leq \alpha.$$

[8] By using a hybrid argument one can show that this implies security even if the adversary can interact in $k \geq 1$ independent instances concurrently (and wins if the verifier accepts in at least one instance). The use of the hybrid argument looses a factor of k in the security reduction.

SECURITY. The standard security notion for a MAC is unforgeability under a chosen message attack (uf-cma). We denote by $\mathsf{Adv}^{\mathrm{uf-cma}}_{\mathsf{MAC}}(\mathcal{A}, \lambda, Q)$, the advantage of the adversary $\mathcal{A}$ in forging a message under a chosen message attack for MAC when used with security parameter λ. Formally this is the probability that the following experiment outputs 1.

Experiment $\mathbf{Exp}^{\mathrm{uf-cma}}_{\mathsf{MAC}}(\mathcal{A}, \lambda, Q)$
$K \leftarrow \mathsf{KG}(1^\lambda)$
Invoke $\mathcal{A}^{\mathsf{TAG}(K,\cdot),\mathsf{VRFY}(K,\cdot,\cdot)}$ who can make up to Q queries to $\mathsf{TAG}(K,\cdot)$ and $\mathsf{VRFY}(K,\cdot,\cdot)$.
Output 1 if $\mathcal{A}$ made a query $(\mathbf{m}, \phi)$ to $\mathsf{VRFY}(K,\cdot,\cdot)$ where
1. $\mathsf{VRFY}(K, \mathbf{m}, \phi) = \mathsf{accept}$
2. $\mathcal{A}$ did not already make the query $\mathbf{m}$ to $\mathsf{TAG}(K,\cdot)$
Output 0 otherwise.

We say that MAC is (t, Q, ε)-secure against uf-cma adversaries if for any $\mathcal{A}$ running in time t in the experiment above, we have $\mathsf{Adv}^{\mathrm{uf-cma}}_{\mathsf{MAC}}(\mathcal{A}, \lambda, Q) \leq \varepsilon$.

2.4 Hard Learning Problems

Let Ber_τ be the Bernoulli distribution over $\mathbb{Z}_2$ with parameter (bias) $\tau \in\ ]0, 1/2[$ (i.e., $\Pr[x = 1] = \tau$ if $x \leftarrow \mathsf{Ber}_\tau$). For $\ell \geq 1$, Ber^ℓ_τ denotes the distribution over $\mathbb{Z}_2^\ell$ where each vector consists of ℓ independent samples drawn from Ber_τ. Given a secret $\mathbf{x} \in \mathbb{Z}_2^\ell$ and $\tau \in]0, \frac{1}{2}[$, we write $\Lambda_{\tau,\ell}(\mathbf{x})$ for the distribution over $\mathbb{Z}_2^\ell \times \mathbb{Z}_2$ whose samples are obtained by choosing a vector $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_2^\ell$ and outputting $(\mathbf{r}, \mathbf{r}^\mathsf{T} \cdot \mathbf{x} \oplus e)$ with $e \xleftarrow{\$} \mathsf{Ber}_\tau$.

The LPN assumption, formally defined below, states that it is hard to distinguish $\Lambda_{\tau,\ell}(\mathbf{x})$ (with a random secret $\mathbf{x} \in \mathbb{Z}_2^\ell$) from the uniform distribution.

Definition 1 (Learning Parity with Noise). *The (decisional)* $\mathsf{LPN}_{\tau,\ell}$ *problem is* (t, Q, ε)*-hard if for every distinguisher* D *running in time* t *and making* Q *queries,*

$$\left|\Pr\left[\mathbf{x} \xleftarrow{\$} \mathbb{Z}_2^\ell \ :\ D^{\Lambda_{\tau,\ell}(\mathbf{x})} = 1\right] - \Pr\left[\ D^{U_{\ell+1}} = 1\right]\right| \leq \varepsilon.$$

Below we define the (seemingly) stronger *subspace* LPN assumption (SLPN for short) recently introduced in [25]. Here the adversary can ask for inner products not only with the secret $\mathbf{x}$, but even with $\mathbf{A} \cdot \mathbf{x} \oplus \mathbf{b}$ where $\mathbf{A}$ and $\mathbf{b}$ can be adaptively chosen, but $\mathbf{A}$ must have sufficiently large rank. For minimal dimension $d \leq \ell$, a secret $\mathbf{x} \in \mathbb{Z}_2^\ell$ and $\mathbf{A} \in \mathbb{Z}_2^{\ell \times \ell}$, $\mathbf{b} \in \mathbb{Z}_2^\ell$, we define the distribution

$$\Gamma_{\tau,\ell,d}(\mathbf{x}, \mathbf{A}, \mathbf{b}) = \begin{cases} \perp & \text{if } \mathsf{rank}(\mathbf{A}) < d \\ \Lambda_{\tau,\ell}(\mathbf{A} \cdot \mathbf{x} \oplus \mathbf{b}) & \text{otherwise} \end{cases}$$

and let $\Gamma_{\tau,\ell,d}(\mathbf{x}, \cdot, \cdot)$ denote the oracle which on input $\mathbf{A}, \mathbf{b}$ outputs a sample from $\Gamma_{\tau,\ell,d}(\mathbf{x}, \mathbf{A}, \mathbf{b})$.

Definition 2 (Subspace LPN). *Let $\ell, d \in \mathbb{Z}$ where $d \leq \ell$. The (decisional) $\mathsf{SLPN}_{\tau,\ell,d}$ problem is (t, Q, ε)-hard if for every distinguisher D running in time t and making Q queries,*

$$\left|\Pr\left[\mathbf{x} \xleftarrow{\$} \mathbb{Z}_2^\ell \ : \ D^{\Gamma_{\tau,\ell,d}(\mathbf{x},\cdot,\cdot)} = 1\right] - \Pr\left[\ D^{U_{\ell+1}(\cdot,\cdot)} = 1\right]\right| \leq \varepsilon,$$

where $U_{\ell+1}(\cdot,\cdot)$ on input $(\mathbf{A}, \mathbf{b})$ outputs a sample of $U_{\ell+1}$ if $\mathsf{rank}(\mathbf{A}) \geq d$ and $\perp$ otherwise.

The following proposition states that the subspace LPN problem mapping to dimension $d + g$ is almost as hard as the standard LPN problem with secrets of length d. The hardness gap is exponentially small in g.

Proposition 1 (From [25]). *For any $\ell, d, g \in \mathbb{Z}$ (where $\ell \geq d + g$), if the $\mathsf{LPN}_{\tau,d}$ problem is (t, Q, ε)-hard then the $\mathsf{SLPN}_{\tau,\ell,d+g}$ problem is (t', Q, ε')-hard where*

$$t' = t - \mathrm{poly}(\ell, Q) \qquad \varepsilon' = \varepsilon + 2Q/2^{g+1}.$$

For some of our constructions, we will only need a weaker version of the $\mathsf{SLPN}_{\tau,\ell,d}$ problem that we call sub**set** LPN. As the name suggests, here the adversary does not ask for inner products with $\mathbf{A} \cdot \mathbf{x} \oplus \mathbf{b}$ for any $\mathbf{A}$ (of rank $\geq d$), but only with *subsets* of $\mathbf{x}$ (of size $\geq d$). It will be convienient to explicitly define this special case. For $\mathbf{x}, \mathbf{v} \in \mathbb{Z}_2^\ell$, let $\mathsf{diag}(\mathbf{v}) \in \mathbb{Z}_2^{\ell \times \ell}$ denote the zero matrix with $\mathbf{v}$ in the diagonal, and let

$$\Gamma^*_{\tau,\ell,d}(\mathbf{x}, \mathbf{v}) := \Gamma_{\tau,\ell,d}(\mathbf{x}, \mathsf{diag}(\mathbf{v}), 0^\ell) = \begin{cases} \perp & \text{if } \mathbf{wt}(\mathbf{v}) < d \\ \Lambda_{\tau,\ell}(\mathbf{x} \wedge \mathbf{v}) & \text{otherwise.} \end{cases}$$

Definition 3 (Subset LPN). *Let $\ell, d \in \mathbb{Z}$ where $d \leq \ell$. The $\mathsf{SLPN}^*_{\tau,\ell,d}$ problem is (t, Q, ε)-hard if for every distinguisher D running in time t and making Q queries,*

$$\left|\Pr\left[\mathbf{x} \xleftarrow{\$} \mathbb{Z}_2^\ell \ : \ D^{\Gamma^*_{\tau,\ell,d}(\mathbf{x},\cdot)} = 1\right] - \Pr\left[\ D^{U_{\ell+1}(\cdot)} = 1\right]\right| \leq \varepsilon,$$

where $U_{\ell+1}(\cdot)$ on input $\mathbf{v}$ (where $\mathbf{wt}(\mathbf{v}) \geq d$) outputs a sample of $U_{\ell+1}$ and $\perp$ otherwise.

Remark 1. $\Gamma^*_{\tau,\ell,d}(\mathbf{x}, \mathbf{v})$ samples are of the form $(\mathbf{r}, \mathbf{r}^\mathsf{T}_{\downarrow \mathbf{v}} \cdot \mathbf{x}_{\downarrow \mathbf{v}} \oplus e) \in \mathbb{Z}_2^{\ell+1}$, where $e \xleftarrow{\$} \mathsf{Ber}_\tau$. To compute the inner product only $\mathbf{r}_{\downarrow \mathbf{v}} \in \mathbb{Z}_2^{\mathbf{wt}(\mathbf{v})}$ is needed, the remaining bits $\mathbf{r}_{\downarrow \overline{\mathbf{v}}} \in \mathbb{Z}_2^{\ell - \mathbf{wt}(\mathbf{v})}$ are irrelevant. We use this observation to improve the communication complexity (for protocols) or tag length (for MACs), by using "compressed" samples of the form $(\mathbf{r}_{\downarrow \mathbf{v}}, \mathbf{r}^\mathsf{T}_{\downarrow \mathbf{v}} \cdot \mathbf{x}_{\downarrow \mathbf{v}} \oplus e) \in \mathbb{Z}_2^{\mathbf{wt}(\mathbf{v})+1}$.

3 Two-Round Authentication with Active Security

In this section we describe our new 2-round authentication protocol and prove its active security under the hardness of the $\mathsf{SLPN}^*_{\tau,2\ell,d}$ problem, where $d = \ell/(2+\gamma)$ for some constant $\gamma > 0$. (Concretely, $\gamma = 0.1$ should do for all practical purposes.)

– Public parameters. The authentication protocol has the following public parameters, where τ, τ' are constants and ℓ, n depend on the security parameter λ.

$\ell \in \mathbb{N}$	length of the secret key $\mathbf{s} \in \mathbb{Z}_2^{2\ell}$
$\tau \in]0, 1/2[$	parameter of the Bernoulli error distribution Ber_τ
$\tau' = 1/4 + \tau/2$	acceptance threshold
$n \in \mathbb{N}$	number of parallel repetitions (we require $n \leq \ell/2$)

– Key Generation. Algorithm $\mathsf{KG}(1^\lambda)$ samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^{2\ell}$ and returns $\mathbf{s}$ as the secret key.

– Authentication Protocol. The 2-round authentication protocol with prover $\mathcal{P}_{\tau,n}$ and verifier $\mathcal{V}_{\tau',n}$ is given in Figure 4.

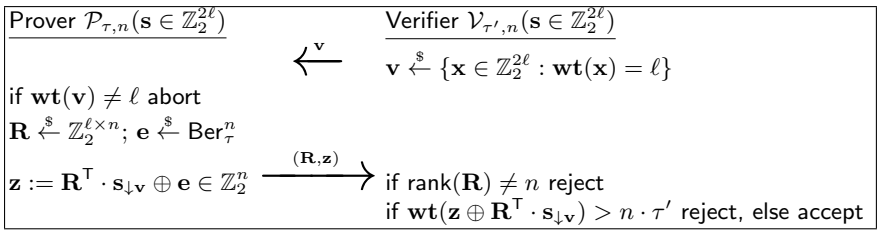

Fig. 4. Two-round authentication protocol AUTH with active security from the LPN assumption.

Theorem 1. *For any constant $\gamma > 0$, let $d = \ell/(2+\gamma)$. If the $\mathsf{SLPN}^*_{\tau,2\ell,d}$ problem is (t, nQ, ε)-hard then the authentication protocol from Figure 4 is (t', Q, ε')-secure against active adversaries, where for constants $c_\gamma, c_\tau > 0$ that depend only on γ and τ respectively,*

$$t' = t - \mathrm{poly}(Q, \ell) \qquad \varepsilon' = \varepsilon + Q \cdot 2^{-c_\gamma \cdot \ell} + 2^{-c_\tau \cdot n} = \varepsilon + 2^{-\Theta(n)} \ .$$

The protocol has completeness error $2^{-c'_\tau \cdot n}$ where $c'_\tau > 0$ depends only on τ.

3.1 Proof of completeness

For any $n \in \mathbb{N}$, $\tau \in]0, 1/2[$, let

$$\alpha_{\tau,n} := \Pr[\mathbf{wt}(\mathbf{e}) > n \cdot \tau' : \ \mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\tau^n] = 2^{-c''_\tau \cdot n} \tag{3.1}$$

denote the probability that n independent Bernoulli samples with bias τ contain more than a $\tau' := 1/4 + \tau/2$ fraction of 1's. The last equality in eq.(3.1) follows from the Hoeffding bound, where the constant $c''_\tau > 0$ depends only on τ.

We now prove that the authentication protocol has completeness error $\alpha \leq 2^{-\ell+n} + \alpha_{\tau,n}$. The verifier performs the following two checks. In the first verification step, the verifier rejects if the random matrix $\mathbf{R}$ does not have full rank.

By Lemma 2 (in Appendix **??**) the probability of this event is $\leq 2^{-n}$. Now, let $\mathbf{e} := \mathbf{z} \oplus \mathbf{R}^{\mathsf{T}} \cdot \mathbf{s}_{\downarrow \mathbf{v}}$ denote the noise added by $\mathcal{P}_{\tau,n}$. Then, in the second verification step, the verifier rejects if $\mathbf{wt}(\mathbf{e}) > n \cdot \tau'$. From equation 3.1, we have that this happens with probability $\alpha_{\tau,n}$. This completes the proof of completeness.

3.2 Proof of security

We first define some terms that will be used later in the security proof. For a constant $\gamma > 0$, let $d = \ell/(2+\gamma)$ (as in Theorem 1). Let $\alpha'_{\ell,d}$ denote the probability that a random substring of length ℓ chosen from a string of length 2ℓ with Hamming weight ℓ, has a Hamming weight less than d. Using the fact that the expected Hamming weight is $\ell/2 = d(1+\gamma/2) = d(1+\Theta(1))$, one can show that there exists a constant $c_\gamma > 0$ (only depending on γ), such that

$$\alpha'_{\ell,d} := \frac{\sum_{i=0}^{d-1} \binom{\ell}{i}\binom{\ell}{\ell-i}}{\binom{2\ell}{\ell}} \leq 2^{-c_\gamma \cdot \ell}. \tag{3.2}$$

For $\tau' = 1/4 + \tau/2$, let $\alpha''_{\tau',n}$ denote the probability that a random bitstring $\mathbf{y} \in \mathbb{Z}_2^n$ has Hamming weight $\mathbf{wt}(\mathbf{y}) \leq n \cdot \tau'$. From the Hoeffding bound, it follows that there exists a constant $c_\tau > 0$ (only depending on τ), such that

$$\alpha''_{\tau',n} := 2^{-n} \cdot \sum_{i=0}^{\lfloor n \cdot \tau' \rfloor} \binom{n}{i} \leq 2^{-c_\tau \cdot n}. \tag{3.3}$$

We now prove security of the authentication protocol. Consider an oracle $\mathcal{O}$ which is either the subset LPN oracle $\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)$ or $U_{2\ell+1}(\cdot)$, as defined in Definition 3. We will construct an adversary $\mathcal{B}^{\mathcal{O}}$ that uses $\mathcal{A}$ (who breaks the active security of AUTH with advantage ε') in a black-box way such that:

$$\Pr[\mathcal{B}^{\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)} \to 1] \geq \varepsilon' - Q \cdot \alpha'_{\ell,d} \qquad \text{and} \qquad \Pr[\mathcal{B}^{U_{2\ell+1}(\cdot)} \to 1] \leq \alpha''_{\tau',n}\ .$$

Thus $\mathcal{B}^{\mathcal{O}}$ can distinguish between the two oracles with advantage $\varepsilon := \varepsilon' - Q \cdot \alpha'_{\ell,d} - \alpha''_{\tau',n}$ as claimed in the statement of the Theorem. Below we define $\mathcal{B}^{\mathcal{O}}$.

Setup. Initially, $\mathcal{B}^{\mathcal{O}}$ samples

$$\mathbf{x}^* \xleftarrow{\$} \mathbb{Z}_2^{2\ell}, \qquad \mathbf{v}^* \xleftarrow{\$} \{\mathbf{y} \in \mathbb{Z}_2^{2\ell} : \mathbf{wt}(\mathbf{y}) = \ell\}.$$

The intuition of our simulation below is as follows. Let us first assume $\mathcal{O}$ is a subset LPN oracle $\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)$ with secret $\mathbf{x}$. In the first phase we have to produce answers $(\mathbf{R},\mathbf{z})$ to a query $\mathbf{v} \in \{\mathbf{y} \in \mathbb{Z}_2^{2\ell} : \mathbf{wt}(\mathbf{y}) = \ell\}$ by $\mathcal{A}$. The simulated answers have exactly the same distribution as the answers of an honest prover $\mathcal{P}_{\tau,n}(\mathbf{s} \in \mathbb{Z}_2^{2\ell})$ where

$$\mathbf{s} = (\mathbf{x}^* \wedge \mathbf{v}^*) \oplus (\mathbf{x} \wedge \overline{\mathbf{v}}^*) \tag{3.4}$$

Thus one part of $\mathbf{s}$'s bits come from $\mathbf{x}^*$, and the other part is from the unknown secret $\mathbf{x}$ (for which we use the oracle $\mathcal{O}$). In the second phase we

give $\mathcal{A}$ the challenge $\mathbf{v}^*$. As $\mathbf{s}_{\downarrow \mathbf{v}^*} = (\mathbf{x}^* \wedge \mathbf{v}^*)_{\downarrow \mathbf{v}^*}$ is known, we will be able to verify if $\mathcal{A}$ outputs a valid forgery.

If $\mathcal{O}$ is the random oracle $U_{2\ell+1}(\cdot)$, then after the first phase $\mathbf{x}^* \wedge \mathbf{v}^*$ is information theoretically hidden, and thus $\mathcal{A}$ cannot come up with a valid forgery but with exponentially small probability.

First phase. In the first phase $\mathcal{B}^{\mathcal{O}}$ invokes $\mathcal{A}$ who expects access to $\mathcal{P}_{\tau,n}(\mathbf{s} \in \mathbb{Z}_2^{2\ell})$. We now specify how $\mathcal{B}^{\mathcal{O}}$ samples the answer $(\mathbf{R}, \mathbf{z})$ to a query $\mathbf{v} \in \{\mathbf{y} \in \mathbb{Z}_2^{2\ell} : \mathbf{wt}(\mathbf{y}) = \ell\}$ made by $\mathcal{A}$. Let

$$\mathbf{u}^* := \mathbf{v} \wedge \mathbf{v}^* \qquad \mathbf{u} := \mathbf{v} \wedge \overline{\mathbf{v}}^*$$

1. $\mathcal{B}^{\mathcal{O}}$ queries its oracle n times on the input $\mathbf{u}$. If the oracle's output is $\perp$ (which happens iff $\mathbf{wt}(\mathbf{u}) < d$), $\mathcal{B}^{\mathcal{O}}$ outputs 0 and stops. Otherwise let $\hat{\mathbf{R}}_1 \in \mathbb{Z}_2^{2\ell \times n}, \mathbf{z}_1 \in \mathbb{Z}_2^n$ denote the n outputs of the oracle.
2. Sample $\hat{\mathbf{R}}_0 \xleftarrow{\$} \mathbb{Z}_2^{2\ell \times n}$ and set $\mathbf{z}_0 = \hat{\mathbf{R}}_0^{\mathsf{T}}(\mathbf{x}^* \wedge \mathbf{u}^*)$.
3. Return $(\mathbf{R} = \hat{\mathbf{R}}_{\downarrow \mathbf{v}} \in \mathbb{Z}_2^{\ell \times n}, \mathbf{z} = \mathbf{z}_0 \oplus \mathbf{z}_1 \in \mathbb{Z}_2^n)$, where $\hat{\mathbf{R}}$ is uniquely determined by requiring $\hat{\mathbf{R}}_{\downarrow \mathbf{v}^*} = \hat{\mathbf{R}}_0$ and $\hat{\mathbf{R}}_{\downarrow \overline{\mathbf{v}}^*} = \hat{\mathbf{R}}_1$.

Second phase. Eventually, $\mathcal{A}$ enters the second phase of the active attack, expecting a challenge from $\mathcal{V}_{\tau',n}(\mathbf{s} \in \mathbb{Z}_2^{2\ell})$.

1. $\mathcal{B}^{\mathcal{O}}$ forwards $\mathbf{v}^*$ as the challenge to $\mathcal{A}$.
2. $\mathcal{A}$ answers with some $(\mathbf{R}^*, \mathbf{z}^*)$.
3. $\mathcal{B}^{\mathcal{O}}$ checks if

$$\mathsf{rank}(\mathbf{R}^*) = n \quad \text{and} \quad \mathbf{wt}(\mathbf{z}^* \oplus {\mathbf{R}^*}^{\mathsf{T}} \cdot \mathbf{x}^*_{\downarrow \mathbf{v}^*}) \leq n \cdot \tau'. \tag{3.5}$$

The output is 1 if both checks succeed and 0 otherwise.

Claim 2 $\Pr[\mathcal{B}^{U_{2\ell+1}(\cdot)} \to 1] \;\leq\; \alpha''_{\tau',n}$.

Proof (of Claim). If $\mathbf{R}^*$ does not have full rank then $\mathcal{B}$ outputs 0 by definition. Therefore, we now consider the case where $\mathsf{rank}(\mathbf{R}^*) = n$.

The answers $(\mathbf{R}, \mathbf{z})$ that the adversary $\mathcal{A}$ obtains from $\mathcal{B}^{U_{2\ell+1}(\cdot)}$ are independent of $\mathbf{x}^*$ (i.e., $\mathbf{z} = \mathbf{z}_0 \oplus \mathbf{z}_1$ is uniform as $\mathbf{z}_1$ is uniform). Since $\mathbf{x}^*_{\downarrow \mathbf{v}^*}$ is uniformly random and $\mathbf{R}^*$ has full rank, the vector

$$\mathbf{y} := {\mathbf{R}^*}^{\mathsf{T}} \cdot \mathbf{x}^*_{\downarrow \mathbf{v}^*} \oplus \mathbf{z}^*$$

is uniformly random over $\mathbb{Z}_2^n$. Thus the probability that the second verification in eq. (3.5) does not fail is $\Pr[\mathbf{wt}(\mathbf{y}) \leq n \cdot \tau'] = \alpha''_{\tau',n}$.

Claim 3 $\Pr[\mathcal{B}^{\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)} \to 1] \;\geq\; \varepsilon' - Q \cdot \alpha'_{\ell,d}$.

Proof (of Claim). We split the proof in two parts. First we show that $\mathcal{B}$ outputs 1 with probability $\geq \varepsilon'$ if the subset LPN oracle accepts subsets of arbitrary small size (and does not simply output $\perp$ on inputs $\mathbf{v}$ where $\mathbf{wt}(\mathbf{v}) < d$), i.e.,

$$\Pr[\mathcal{B}^{\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)} \to 1] \geq \varepsilon'. \tag{3.6}$$

Then we'll upper bound the gap between the probability that $\mathcal{B}$ outputs 1 in the above case and the probability that $\mathcal{B}$ outputs 1 when given access to the oracle that we are interested in as:

$$\left|\Pr[\mathcal{B}^{\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)} \to 1] - \Pr[\mathcal{B}^{\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)} \to 1]\right| \leq Q \cdot \alpha'_{\ell,d}. \tag{3.7}$$

The claim then follows by the triangle inequality from the two equations above.

Eq. (3.6) holds as:

- The answers $(\mathbf{R}, \mathbf{z})$ that $\mathcal{B}^{\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)}$ gives to $\mathcal{A}$'s queries in the first phase of the attack have *exactly* the same distribution as what $\mathcal{A}$ would get when interacting with an honest prover $\mathcal{P}_{\tau,n}(\mathbf{s} \in \mathbb{Z}_2^{2\ell})$ where the "simulated" secret $\mathbf{s}$ is defined in eq.(3.4).

  To see this, recall that on a query $\mathbf{v}$ from $\mathcal{A}$, $\mathcal{B}^{\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)}$ must answer with $(\mathbf{R}, \mathbf{z} = \mathbf{R}^\mathsf{T}\mathbf{s}_{\downarrow\mathbf{v}} \oplus \mathbf{e})$, where $\mathbf{R} = \hat{\mathbf{R}}_{\downarrow\mathbf{v}}$ is the compressed form of $\hat{\mathbf{R}}$ (cf. Remark 1). In the first step, $\mathcal{B}$ queries its oracle with $\mathbf{u} = \mathbf{v} \wedge \overline{\mathbf{v}}^*$ and obtains noisy inner products $(\hat{\mathbf{R}}_1, \mathbf{z}_1)$ with the part of $\mathbf{s}_{\downarrow\mathbf{v}}$ that contains only bits from $\mathbf{x}$, i.e.,

  $$\mathbf{z}_1 = \hat{\mathbf{R}}_1^\mathsf{T} \cdot (\mathbf{x} \wedge \mathbf{u}) \oplus \mathbf{e} = \hat{\mathbf{R}}_1 \cdot (\mathbf{s} \wedge \mathbf{u}) \oplus \mathbf{e}.$$

  In the second step, $\mathcal{B}$ samples n inner products $(\hat{\mathbf{R}}_0, \mathbf{z}_0)$ (with no noise) with the part of $\mathbf{s}_{\downarrow\mathbf{v}}$ that contains only bits from the known $\mathbf{x}^*$, i.e.,

  $$\mathbf{z}_0 = \hat{\mathbf{R}}_0 \cdot (\mathbf{x}^* \wedge \mathbf{u}^*) = \hat{\mathbf{R}}_0 \cdot (\mathbf{s} \wedge \mathbf{u}^*).$$

  In the third step, $\mathcal{B}$ then generates $(\hat{\mathbf{R}}, \hat{\mathbf{R}} \cdot (\mathbf{s} \wedge \mathbf{v}) \oplus \mathbf{e})$ from the previous values where $\hat{\mathbf{R}}$ is defined by $\hat{\mathbf{R}}_{\downarrow\mathbf{v}^*} = \hat{\mathbf{R}}_0$ and $\hat{\mathbf{R}}_{\downarrow\overline{\mathbf{v}}^*} = \hat{\mathbf{R}}_1$. Using $\mathbf{v} = \mathbf{u} \oplus \mathbf{u}^*$, we get

  $$\begin{aligned}\mathbf{z} &= \mathbf{z}_0 \oplus \mathbf{z}_1 \\ &= \hat{\mathbf{R}}_0 \cdot (\mathbf{s} \wedge \mathbf{u}^*) \oplus \hat{\mathbf{R}}_1 \cdot (\mathbf{s} \wedge \mathbf{u}) \oplus \mathbf{e} \\ &= \hat{\mathbf{R}} \cdot (\mathbf{s} \wedge \mathbf{v}) \oplus \mathbf{e} \\ &= \mathbf{R} \cdot \mathbf{s}_{\downarrow\mathbf{v}} \oplus \mathbf{e}.\end{aligned}$$

- The challenge $\mathbf{v}^*$ sent to $\mathcal{A}$ in the second phase of the active attack is uniformly random (even given the entire view so far), and therefore has the same distribution as a challenge in an active attack.
- $\mathcal{B}^{\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)}$ outputs 1 if eq.(3.5) holds, which is exactly the case when $\mathcal{A}$'s response to the challenge was valid. By assumption this probability is at least ε'.

It remains to prove eq.(3.7). Note that $\Gamma^*_{\tau,2\ell,0}(\mathbf{x},\cdot)$ behaves exactly like $\Gamma^*_{\tau,2\ell,d}(\mathbf{x},\cdot)$ as long as one never makes a query $\mathbf{v}$ where $\mathbf{wt}(\mathbf{v} \wedge \overline{\mathbf{v}}^*) < d$.

Since $\mathbf{v}^* \xleftarrow{\$} \{\mathbf{y} \in \mathbb{Z}_2^{2\ell} : \mathbf{wt}(\mathbf{y}) = \ell\}$, for any $\mathbf{v}$, the probability that $\mathbf{wt}(\mathbf{v} \wedge \overline{\mathbf{v}}^*) < d$ is (by definition) $\alpha'_{\ell,d}$ as defined in eq.(3.2). Using the union bound, we can upper bound the probability that $\mathbf{wt}(\mathbf{v} \wedge \overline{\mathbf{v}}^*) < d$ for any of the Q different $\mathbf{v}$'s chosen by the adversary as $Q \cdot \alpha'_{\ell,d}$.

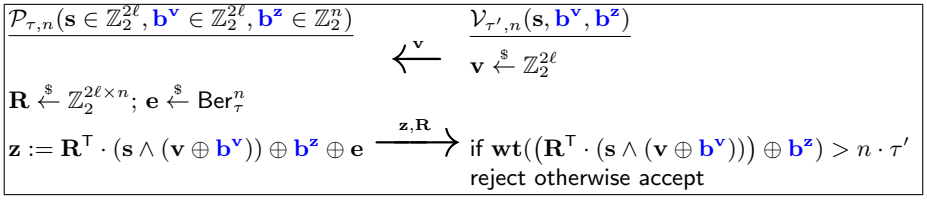

Fig. 5. By blinding the values $\mathbf{v}, \mathbf{z}$ with secret random vectors $\mathbf{b^v}, \mathbf{b^z}$ we can avoid checking whether $\mathbf{wt}(\mathbf{v}) = \ell$ and $\mathsf{rank}(\mathbf{R}) = n$ as in the protocol from Figure 4.

3.3 Avoid Checking

One disadvantage of the protocol in Figure 4, compared to HB style protocols, is the necessity to check whether the messages exchanged have the right from: the prover checks if $\mathbf{v}$ has weight ℓ, while the verifier must make the even more expensive check whether $\mathbf{R}$ has full rank. Eliminating such verification procedures can be particularly useful if for example the prover is an RFID chip where even the simple verification that a vector has large weight is expensive. We note that it is possible to eliminate these checks by blinding the exchanged messages $\mathbf{v}$ and $\mathbf{z}$ using random vectors $\mathbf{b^v} \in \mathbb{Z}_2^{2\ell}$ and $\mathbf{b^z} \in \mathbb{Z}_2^n$ respectively, as shown in Figure 5. The security and completeness of this protocol is basically the same as for the protocol in Figure 5. The security proof is also very similar and is therefore omitted.

4 Message Authentication Codes

In this section, we construct two message authentication codes whose security can be reduced to the LPN assumption. Our first construction is based on the 2-round authentication protocol from Section 3. We prove that if the LPN problem is ε-hard, then no adversary making Q queries can forge a MAC with probability more than $\Theta(\sqrt{\varepsilon} \cdot Q)$. However, the construction has the disadvantage that one needs to fix the hardness of the LPN problem at the time of the construction, c.f. Remark 2. Our second construction has no such issues and achieves better security $\Theta(\varepsilon \cdot Q)$. The efficiency of this construction is similar to that of the first construction, but a larger key is required.

4.1 First construction

Recall the 2-round authentication protocol from Section 3. In the protocol the verifier chooses a random challenge subset $\mathbf{v}$. To turn this interactive protocol into a MAC, we will compute this $\mathbf{v}$ from the message $\mathbf{m}$ to be authenticated as $\mathbf{v} = \mathsf{C}(h(\mathbf{m}, \mathbf{b}))$, where h is a pairwise independent hash function, $\mathbf{b} \in \mathbb{Z}_2^\nu$ is some fresh randomness and C is some encoding scheme. The code C is fixed and public, while the function h is part of the secret key. The authentication tag ϕ

is computed in the same manner as the prover's answer in the authentication protocol. That is, we sample a random matrix $\mathbf{R} \in \mathbb{Z}_2^{\ell \times n}$ and compute a noisy inner product $\mathbf{z} := \mathbf{R}^\mathsf{T} \cdot \mathbf{s}_{\downarrow \mathbf{v}} \oplus \mathbf{e}$, where $\mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\tau^n$. We note that using $(\mathbf{R}, \mathbf{z})$ as an authentication tag would not be secure, and we need to blind these values. This is done by applying an (almost) pairwise independent permutation (PIP) π – which is part of the secret key – to $(\mathbf{R}, \mathbf{z}, \mathbf{b}) \in \mathbb{Z}_2^{\ell \times n + n + \nu}$.

Construction. The message authentication code $\mathsf{MAC}_1 = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated message space $\mathcal{M}$ is defined as follows.

- Public parameters. MAC_1 has the following public parameters.

ℓ, τ, τ', n	as in the authentication protocol from Section 3
$\mu \in \mathbb{N}$	output length of the hash function
$\nu \in \mathbb{N}$	length of the randomness
$\mathsf{C} : \mathbb{Z}_2^\mu \to \mathbb{Z}_2^{2\ell}$	encoding, where $\forall\ \mathbf{x} \neq \mathbf{x}' \in \mathbb{Z}_2^\mu$ we have $\mathbf{wt}(\mathsf{C}(\mathbf{x})) = \ell$ and $\mathbf{wt}(\mathsf{C}(\mathbf{x}) \oplus \mathsf{C}(\mathbf{x}')) \geq 0.9\ell$.

- Key generation. Algorithm $\mathsf{KG}(1^\lambda)$ samples $\mathbf{s} \xleftarrow{\$} \mathbb{Z}_2^{2\ell}$, an (almost) pairwise independent hash function $h : \mathcal{M} \times \mathbb{Z}_2^\nu \to \mathbb{Z}_2^\mu$ and a pairwise independent permutation π over $\mathbb{Z}_2^{\ell \times n + n + \nu}$. It returns $K = (\mathbf{s}, h, \pi)$ as the secret key.
- Tagging. Given secret key $K = (\mathbf{s}, h, \pi)$ and message $\mathbf{m} \in \mathcal{M}$, algorithm TAG proceeds as follows.
  1. $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$, $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^\nu$, $\mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\tau^n$
  2. $\mathbf{v} := \mathsf{C}(h(\mathbf{m}, \mathbf{b})) \in \mathbb{Z}_2^{2\ell}$
  3. Return $\phi := \pi(\mathbf{R}, \mathbf{R}^\mathsf{T} \cdot \mathbf{s}_{\downarrow \mathbf{v}} \oplus \mathbf{e}, \mathbf{b})$
- Verification. On input a secret-key $K = (\mathbf{s}, h, \pi)$, message $\mathbf{m} \in \mathcal{M}$ and tag ϕ, algorithm VRFY proceeds as follows.
  1. Parse $\pi^{-1}(\phi)$ as $(\mathbf{R} \in \mathbb{Z}_2^{\ell \times n}, \mathbf{z} \in \mathbb{Z}_2^n, \mathbf{b} \in \mathbb{Z}_2^\nu)$. If $\mathsf{rank}(\mathbf{R}) \neq n$, then return reject
  2. $\mathbf{v} := \mathsf{C}(h(\mathbf{m}, \mathbf{b}))$
  3. If $\mathbf{wt}(\mathbf{z} \oplus \mathbf{R}^\mathsf{T} \cdot \mathbf{s}_{\downarrow \mathbf{v}}) > n \cdot \tau'$ return reject, otherwise return accept

Theorem 4. *For $\mu = \nu \in \mathbb{N}$, a constant $\gamma > 0$ and $d := \ell/(2+\gamma)$, if the $\mathsf{SLPN}^*_{\tau, 2\ell, d}$ problem is (t, nQ, ε)-hard then MAC_1 is (t', Q, ε')-secure against uf-cma adversaries, where*

$$t' \approx t, \qquad \varepsilon = \min\left\{\varepsilon'/2 - \frac{Q^2}{2^{\mu-2}}\ ,\ \frac{\varepsilon'}{2^{\mu+1}} - 2^{-\Theta(n)}\right\}\ .$$

MAC_1 has completeness error $2^{-c_\tau \cdot n}$ where $c_\tau > 0$ depends only on τ.

Corollary 1. *Choosing μ s.t. $2^\mu = \frac{Q^2 \cdot 2^4}{\varepsilon'}$ in the above theorem, we get $\varepsilon = \min\{\varepsilon'/4, (\varepsilon')^2/(2^5 Q^2) - 2^{-\Theta(n)}\}$. The 2nd term is the minimum here, and solving for ε' gives*

$$\varepsilon' := \sqrt{32} \cdot Q \cdot \sqrt{\varepsilon + 2^{-\Theta(n)}}. \tag{4.1}$$

Remark 2 (about μ). Note that to get security as claimed in the above corollary, we need to choose μ as a function of Q and ε such that $2^\mu \approx Q^2 \cdot 2^4/\varepsilon'$ for ε'

as in eq.(4.1). Of course we can just fix Q (as an upper bound to the number of queries made by the adversary) and ε (as our guess on the actual hardness of $\mathsf{SLPN}^*_{\tau,2\ell,d}$). But a too conservative guess on μ (i.e. choosing μ too small) will result in a construction whose security is worse than what is claimed in the above corallary. A too generous guess on the other hand will make the security reduction meaningless (we don't have any actual attacks on the MAC for large μ though).

We now give an intuition for the proof of Theorem 4. A formal proof will be given in Appendix B.1. Every query $(\mathbf{m}, \phi)$ to VRFY and query $\mathbf{m}$ to TAG defines a subset $\mathbf{v}$ (as computed in the second step in the definitions of both VRFY and TAG). We say that a forgery $(\mathbf{m}, \phi)$ is "fresh" if the $\mathbf{v}$ contained in $(\mathbf{m}, \phi)$ is different from all $\mathbf{v}$'s contained in all the previous VRFY and TAG queries. The proof makes a case distinction and uses a different reduction for the two cases where the forgery found by the adversary is more likely to be fresh, or more likely to be non-fresh. In both cases we consider a reduction $\mathcal{B}^{\mathcal{O}}$ which has access to either a uniform oracle $\mathcal{O} = U$ or a subset LPN oracle $\mathcal{O} = \Gamma^*$. $\mathcal{B}^{\mathcal{O}}$ uses an adversary $\mathcal{A}$ who can find forgeries for the MAC to distinguish those cases and thus break the subset LPN assumption. In the first case, where the first forgery is likely to be **non-fresh**, we can show (using the fact that a pairwise independent permutation is used to blind the tag) that if $\mathcal{B}^{\mathcal{O}}$'s oracle is $\mathcal{O} = U$, even a computationally unbounded $\mathcal{A}$ cannot come up with a message/tag pair $(\mathbf{m}, \phi)$ that contains a non-fresh $\mathbf{v}$. Thus we can distinguish the cases $\mathcal{O} = U$ and $\mathcal{O} = \Gamma^*$ by just observing if $\mathcal{A}$ ever makes a VRFY query $(\mathbf{m}, \phi)$ that contains a non-fresh $\mathbf{v}$ (even without being able to tell if $(\mathbf{m}, \phi)$ is valid or not).

If the forgery found by $\mathcal{A}$ is more likely to be **fresh**, we can use a similar argument as in the proof of our authentication protocol in the last section. An additional difficulty here is that the reduction has to guess the fresh $\mathbf{v} \in \mathbb{Z}_2^\mu$ contained in the first forgery and cannot choose it as in the protocol. This is the reason why the reduction looses a factor 2^μ.

4.2 Second construction

We now give the construction of another MAC based on the hardness of the LPN problem. The main difference to MAC_1 from the last subsection is the way we generate the values $\mathbf{s}(\mathbf{v})$. In the new construction, we define $\mathbf{s}(\mathbf{v}) := \mathbf{s}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{s}_i$, where each $\mathbf{s}_i$ is a part of the secret key. The construction uses ideas from Waters' IBE scheme [29], and parts of the security reduction use simulation tricks from [7, 1] that we need to adapt to the binary case.

Construction. The message authentication code $\mathsf{MAC}_2 = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated message space $\mathcal{M}$ is defined as follows.

- Public parameters. MAC_2 has the following public parameters.
  ℓ, τ, τ', n as in the authentication protocol from Section 3
  $\mu \in \mathbb{N}$ output length of the hash function
  $\nu \in \mathbb{N}$ length of the randomness

– Key generation. Algorithm $\mathsf{KG}(1^\lambda)$ samples $\mathbf{s}_i \xleftarrow{\$} \mathbb{Z}_2^\ell$ (for $0 \le i \le \mu$) and chooses a pairwise independent hash function $h : \mathcal{M} \times \mathbb{Z}_2^\nu \to \mathbb{Z}_2^\mu$, as well as a pairwise independent permutation π over $\mathbb{Z}_2^{\ell \times n+n+\nu}$. It returns $K = (\mathbf{s}_0, \ldots, \mathbf{s}_\mu, h, \pi)$ as the secret key.
– Tagging. Given secret key $K = (\mathbf{s}_0, \ldots, \mathbf{s}_\mu, h, \pi)$ and message $\mathbf{m} \in \mathcal{M}$, algorithm $\overline{\mathsf{TAG}}$ proceeds as follows.
  1. $\mathbf{R} \xleftarrow{\$} \mathbb{Z}_2^{\ell \times n}$, $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^\nu$, $\mathbf{e} \xleftarrow{\$} \mathsf{Ber}_\tau^n$
  2. $\mathbf{v} := h(\mathbf{m}, \mathbf{b})$
  3. $\mathbf{s}(\mathbf{v}) := \mathbf{s}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{s}_i$
  4. Return $\phi := \pi(\mathbf{R}, \mathbf{R}^\mathsf{T} \cdot \mathbf{s}(\mathbf{v}) \oplus \mathbf{e}, \mathbf{b})$
– Verification. On input a secret-key $K = (\mathbf{s}_0, \ldots, \mathbf{s}_\mu, h, \pi)$, message $\mathbf{m} \in \mathcal{M}$ and tag ϕ, algorithm VRFY proceeds as follows.
  1. Parse $\pi^{-1}(\phi)$ as $(\mathbf{R} \in \mathbb{Z}_2^{\ell \times n}, \mathbf{z} \in \mathbb{Z}_2^n, \mathbf{b} \in \mathbb{Z}_2^\nu)$. If $\mathsf{rank}(\mathbf{R}) \neq n$, then return reject
  2. $\mathbf{v} := h(m, \mathbf{b})$
  3. $\mathbf{s}(\mathbf{v}) := \mathbf{s}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{s}_i$
  4. If $\mathbf{wt}(\mathbf{z} \oplus \mathbf{R}^\mathsf{T} \cdot \mathbf{s}(\mathbf{v})) > n \cdot \tau'$ return reject, otherwise return accept

Theorem 5. *If the* $\mathsf{SLPN}_{\tau,\ell,\ell}$ *problem is* (t, nQ, ε)*-hard, then* MAC_2 *is* (t', Q, ε')*-secure against uf-cma adversaries, where*

$$t' \approx t \qquad \varepsilon \ = \min\left\{\varepsilon'/2 - \frac{Q^2}{2^{\mu-2}}, \frac{\varepsilon'}{4Q} - 2^{-\Theta(n)}\right\}.$$

MAC_2 *has completeness error* $2^{-c_\tau \cdot n}$ *where* c_τ *only depends on* τ*.*

We now give an intuition for the proof of Theorem 5. A formal proof will be given in Appendix B.2. Similar to the proof of Theorem 4, we distinguish fresh and non-fresh forgeries. Here the new and interesting case is the fresh forgery. The idea is that in the reduction to the SLPN problem we define the function $\mathbf{s}(\mathbf{v}) = \mathbf{A}(\mathbf{v}) \cdot \mathbf{s} \oplus \mathbf{b}(\mathbf{v})$ (where $\mathbf{s}$ is the LPN secret) such that the following holds with non-negligible probability: (i) for each $\mathbf{v}_i$ from the TAG queries, $\mathbf{A}(\mathbf{v}_i)$ has full rank ℓ and hence the tags can be simulated using the provided $\Gamma_{\tau,\ell,\ell}(\mathbf{s}, \cdot, \cdot)$ oracle; (ii) for the first fresh forgery we have $\mathbf{A}(\mathbf{v}) = 0$ such that $\mathbf{s}(\mathbf{v})$ is independent of $\mathbf{s}$ and the reduction can check the forgery's correctness. The above two properties allow to maintain the simulation. The setup of the function $\mathbf{s}(\cdot)$ is the crucial step and here we adapt a technique recently introduced by Boyen [7] based on homomorphic encodings with full-rank differences that allows us to arbitrarily control the probability that the above simulation works.

Acknowledgements

Krzysztof would like to thank Vadim Lyubashevsky for many interesting discussions on LPN while being in Tel Aviv (and Eyjafjallajökull for making this stay possible.)

References

[1] Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *EUROCRYPT 2010*, volume 6110 of *LNCS*, pages 553–572. Springer, May 2010.

[2] E. Berlekamp, R. McEliece, and H. van Tilborg. On the inherent intractability of certain coding problems. *IEEE Transactions on Information Theory*, 24(3):384–386, 1978.

[3] Avrim Blum, Merrick L. Furst, Michael J. Kearns, and Richard J. Lipton. Cryptographic primitives based on hard learning problems. In Douglas R. Stinson, editor, *CRYPTO'93*, volume 773 of *LNCS*, pages 278–291. Springer, August 1994.

[4] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. In *32nd ACM STOC*, pages 435–440. ACM Press, May 2000.

[5] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[6] Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *EUROCRYPT 2004*, volume 3027 of *LNCS*, pages 223–238. Springer, May 2004.

[7] Xavier Boyen. Lattice mixing and vanishing trapdoors: A framework for fully secure short signatures and more. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010*, volume 6056 of *LNCS*, pages 499–517. Springer, May 2010.

[8] Julien Bringer, Hervé Chabanne, and Emmanuelle Dottax. HB^{++}: a lightweight authentication protocol secure against some attacks. In *SecPerU*, pages 28–33, 2006.

[9] Ronald Cramer and Ivan Damgard. On the amortized complexity of zero-knowledge protocols. In Shai Halevi, editor, *CRYPTO 2009*, volume 5677 of *LNCS*, pages 177–191. Springer, August 2009.

[10] D.N. Duc and K. Kim. Securing HB+ against GRS man-in-the-middle attack. In *SCIS*, 2007.

[11] Jean-Bernard Fischer and Jacques Stern. An efficient pseudo-random generator provably as secure as syndrome decoding. In Ueli M. Maurer, editor, *EUROCRYPT'96*, volume 1070 of *LNCS*, pages 245–255. Springer, May 1996.

[12] Martin Fürer. Faster integer multiplication. *SIAM J. Comput.*, 39(3):979–1005, 2009.

[13] Henri Gilbert, Matt Robshaw, and Herve Sibert. An active attack against HB+ - a provably secure lightweight authentication protocol. Cryptology ePrint Archive, Report 2005/237, 2005. http://eprint.iacr.org/.

[14] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. Good variants of HB+ are hard to find. In Gene Tsudik, editor, *FC 2008*, volume 5143 of *LNCS*, pages 156–170. Springer, January 2008.

[15] Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. $HB^{\sharp}$: Increasing the security and efficiency of HB^{+}. In Nigel P. Smart, editor, *EUROCRYPT 2008*, volume 4965 of *LNCS*, pages 361–378. Springer, April 2008.

[16] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33:792–807, 1986.

[17] Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *ASIACRYPT 2001*, volume 2248 of *LNCS*, pages 52–66. Springer, December 2001.

[18] Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *CRYPTO 2005*, volume 3621 of *LNCS*, pages 293–308. Springer, August 2005.
[19] Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the HB and HB+ protocols. In Serge Vaudenay, editor, *EUROCRYPT 2006*, volume 4004 of *LNCS*, pages 73–87. Springer, May / June 2006.
[20] 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, July 2010.
[21] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.
[22] Éric Levieil and Pierre-Alain Fouque. An improved LPN algorithm. In Roberto De Prisco and Moti Yung, editors, *SCN 06*, volume 4116 of *LNCS*, pages 348–359. Springer, September 2006.
[23] Jorge Munilla and Alberto Peinado. HB-MP: A further step in the HB-family of lightweight authentication protocols. *Computer Networks*, 51(9):2262–2267, 2007.
[24] Khaled Ouafi, Raphael Overbeck, and Serge Vaudenay. On the security of HB# against a man-in-the-middle attack. In Josef Pieprzyk, editor, *ASIACRYPT 2008*, volume 5350 of *LNCS*, pages 108–124. Springer, December 2008.
[25] Krzysztof Pietrzak. Subspace LWE, 2010. Manuscript available at http://homepages.cwi.nl/∼pietrzak/publications/SLWE.pdf.
[26] Oded Regev. On lattices, learning with errors, random linear codes, and cryptography. In Harold N. Gabow and Ronald Fagin, editors, *37th ACM STOC*, pages 84–93. ACM Press, May 2005.
[27] Schönhage and V. Strassen. Schnelle Multiplikation grosser Zahlen. *Computing*, 7, 1971.
[28] Jeroen Van De Graaf. *Towards a formal definition of security for quantum protocols*. PhD thesis, Monreal, P.Q., Canada, Canada, 1998. AAINQ35648.
[29] Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *EUROCRYPT 2005*, volume 3494 of *LNCS*, pages 114–127. Springer, May 2005.
[30] John Watrous. Zero-knowledge against quantum attacks. *SIAM J. Comput.*, 39(1):25–58, 2009.

A Extensions

In this section we discuss some extensions of the protocols we presented in Section 3 and Section 4.

A.1 Trading Key-Size for Communication Complexity

A disadvantage of the schemes proposed in this paper is their large communication complexity. For example, in the authentication protocol from Section 3 the prover has to send the entire $\ell \times n$ matrix $\mathbf{R}$ to the verifier. Similarly, in the MACs from Section 4, the tag is computed by permuting a string of the form $(\mathbf{R}, \mathbf{R}^{\mathsf{T}} \cdot \mathbf{s}(\mathbf{m}) \oplus \mathbf{e}, \mathbf{b})$, where again $\mathbf{R}$ is an $\ell \times n$ matrix.

We now explain a simple tradeoff that is originally due to Gilbert et al. [15]. Consider the authentication protocol from Section 3. Let $1 \leq c \leq n$ be an integer parameter and let $n_s := c$ and $n_r := n/c$. The idea is to use a larger secret matrix

$\mathbf{S} \in \mathbb{Z}_2^{2\ell \times n_s}$ (instead of just one vector $\mathbf{s}$) and a smaller random matrix $\mathbf{R} \in \mathbb{Z}_2^{\ell \times n_r}$ (instead of $\mathbf{R} \in \mathbb{Z}_2^{\ell \times n}$). The resulting protocol is illustrated in Figure 6. Similar extensions can be easily derived for the MACs of Section 4, where the tradeoff is more important due to the pairwise independent permutation π which is the computational bottleneck of the protocol. See Figure 3 for a comparison of the resulting complexities. The proof of Theorem 1, Theorem 4 and Theorem 5 can be adapted to show the same security and completeness results.

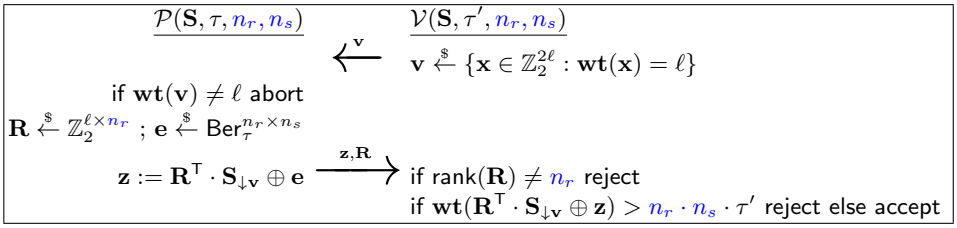

Fig. 6. A generalization of the protocol from Figure 5 where we trade a larger key (which now is a matrix $\mathbf{S} \in \mathbb{Z}_2^{2\ell \times n_s}$) for lower communication and randomness complexity. The protocol is as secure as the protocol from Figure 5 (wich is the special case where $n_r = n$ and $n_s = 1$) with $n = n_r \cdot n_s$.

A.2 Generalization to LWE

All the protocols presented in this paper are based on the hardness of the LPN problem. A natural generalization of this problem is the learning with errors (LWE) problem [26]. The most appealing characteristic of this problem is that it enjoys for certain parameters a worst-case hardness guarantee [26, **?**]. We informally recall the LWE problem below. Let $q \geq 2$ be a prime and denote with $\mathsf{Gau}_{q,\tau}$ the so called "discretized normal error" distribution parameterized by some $\tau \in]0, 1[$. This distribution is obtained by drawing $x \in \mathbb{R}$ from the Gaussian distribution of width τ (i.e., x is chosen with probability $\frac{1}{\tau}\exp(-\pi x^2/\tau^2)$) and outputting $\lfloor q \cdot x \rceil \bmod q$. For a random secret $\mathbf{s} \in \mathbb{Z}_q^\ell$, the (decisional) $\mathsf{LWE}_{q,\tau,\ell}$ problem is to distinguish samples of the form $(\mathbf{r}, \mathbf{r}^\mathsf{T} \cdot \mathbf{s} + e)$ from uniformly random samples in $\mathbb{Z}_q^\ell \times \mathbb{Z}_q$, where $\mathbf{r} \xleftarrow{\$} \mathbb{Z}_q^\ell$, $e \xleftarrow{\$} \mathsf{Gau}_{q,\tau}$ and all the operations are performed modulo q. The subspace/subset version of the LWE problem can be defined exactly in the same fashion as for LPN (cf. Definition 2). It was showed in [25] that the subspace/subset LWE problems are equivalent to the LWE problem.

All the protocols in this paper can be generalized to $\mathbb{Z}_q$ and proven secure under the hardness of the subset LWE assumption (and hence the standard LWE assumption). This requires us to sample all the elements from $\mathbb{Z}_q$ (instead of $\mathbb{Z}_2$), replace Ber_τ with $\mathsf{Gau}_{q,\tau}$ and perform all the operations involved modulo

q. We need also to specify how to replace the verification steps involving the computation of Hamming weights $\mathbf{wt}(\cdot)$. Given a vector $\mathbf{e} \in \mathbb{Z}_q^n$ sampled from $\mathsf{Gau}_{q,\tau}^n$ (where $\mathbf{e}$ has the form $\mathbf{z} - \mathbf{R}^\mathsf{T} \cdot \mathbf{s}_{\downarrow \mathbf{v}} \bmod q$ for an honest execution of the protocol from Section 3 or $\mathbf{z} - \mathbf{R}^\mathsf{T} \cdot \mathbf{s}(\mathbf{v}) \bmod q$ for the schemes from Section 4), this can be done by checking that the (squared) Euclidean norm of $\mathbf{e}$, i.e., the quantity $\|\mathbf{e}\|^2 := \sum_{i=1}^n |\mathbf{e}[i]|^2$, does not exceed $n \left\lfloor \frac{q}{2} \right\rfloor \cdot \tau'$ (which will happen with overwhelming probability by the standard tail bound on Gaussians).

Thus the change of domain from $\mathbb{Z}_2$ to $\mathbb{Z}_q$ buys us security based on a different assumption, which is known to be equivalent (for a proper choice of parameters) to the hardness of well-studied (worst-case) lattice problems. This comes at the price of a higher computational complexity, which may be a problem in the context of resource bounded devices.

B Omitted Proofs

B.1 Proof of Theorem 4

The proof of completeness is essentially the same (and we get exactly the same quantitative bound) as the proof of completeness for the protocol in Figure 4 as claimed in Theorem 1.

We now prove security. As in the theorem statement, we set $\mu = \nu$ (but for clarity we will keep the different letters μ for the range of h and ν for the length of the randomness). Let $\mathcal{A}$ be an adversary that (t', Q, ε')-breaks the uf-cma security of MAC_1. Let Q_{tag} and Q_{vrfy} denote the number of queries that $\mathcal{A}$ makes to the oracles $\mathsf{TAG}(K, \cdot)$ and $\mathsf{VRFY}(K, \cdot, \cdot)$ respectively, such that $Q = Q_{\mathsf{tag}} + Q_{\mathsf{vrfy}}$. We assume that $\mathcal{A}$ never makes the same VRFY query twice (since VRFY is deterministic, repeating queries gives no additional information to $\mathcal{A}$) and also that she never makes a VRFY query $(\mathbf{m}, \phi)$ where ϕ was received as the output from TAG on input $\mathbf{m}$. Since the completeness error of MAC_1 is $2^{-\Theta(n)}$, this is basically without loss of generality (as the answer would almost certainly be accept). Every query $(\mathbf{m}, \phi)$ to VRFY and query $\mathbf{m}$ to TAG defines a subset $\mathbf{v}$ (as computed in step 2. in the definitions of both VRFY and TAG).

By definition, in the uf-cma experiment, with probability ε' the adversary $\mathcal{A}$ at some point makes a VRFY query $(\mathbf{m}, \phi)$ where: (i) ϕ was not received as output on a TAG query $\mathbf{m}$, and (ii) $\mathsf{VRFY}(K, \mathbf{m}, \phi) = \mathsf{accept}$. We say that such a forgery $(\mathbf{m}, \phi)$ is "fresh" if the $\mathbf{v}$ defined by $(\mathbf{m}, \phi)$ is different from all $\mathbf{v}$'s defined by all the previous VRFY and TAG queries. Let E_{fresh} denote the event that $\mathcal{A}$ finds a fresh forgery. As $\mathcal{A}$ finds a forgery with probability ε' and every forgery must be either fresh or not, we have that:

$$\Pr[E_{\mathsf{fresh}}] + \Pr[\neg E_{\mathsf{fresh}}] = \varepsilon'.$$

We will consider the two cases where $\Pr[E_{\mathsf{fresh}}] > \varepsilon'/2$ and $\Pr[E_{\mathsf{fresh}}] \leq \varepsilon'/2$ separately.

The case $\Pr[E_{\mathsf{fresh}}] \leq \varepsilon'/2$. Given $\mathcal{A}$, we will construct an adversary $\mathcal{B}_1^{\mathcal{O}}$ who can distinguish $\mathcal{O} = \Gamma^*_{\tau,2\ell,d}(\mathbf{s}, \cdot)$ from $\mathcal{O} = U_{2\ell+1}(\cdot)$ (as in Definition 3) with

advantage[9]

$$\varepsilon'/2 - \frac{Q^2}{2^{\mu-2}}. \tag{B.1}$$

$\mathcal{B}_1^{\mathcal{O}}$ samples π, h (but not $\mathbf{s}$) as defined by KG. Next, it invokes $\mathcal{A}$ (who expects to attack MAC_1 with a key $(\mathbf{s}, h, \pi)$) answering its queries as follows:

- If $\mathcal{A}$ makes a TAG query $\mathbf{m}$, then $\mathcal{B}_1^{\mathcal{O}}$ does the following:
  1. Sample $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^\nu$ and compute $\mathbf{v} := \mathsf{C}(h(\mathbf{m}, \mathbf{b}))$.
  2. Query the oracle $\mathcal{O}$ for n times on input $\mathbf{v}$: for $i = 1, \ldots, n$ let $(\mathbf{R}[i], \mathbf{z}[i]) \xleftarrow{\$} \mathcal{O}(\mathbf{v})$.
  3. Return $\phi := \pi(\mathbf{R}, \mathbf{z}, \mathbf{b})$ where $\mathbf{R} = [\mathbf{R}[1], \ldots, \mathbf{R}[n]]$ and $\mathbf{z} = [\mathbf{z}[1], \ldots, \mathbf{z}[n]]$ to $\mathcal{A}$.
- If $\mathcal{A}$ makes a VRFY query $(\mathbf{m}, \phi)$, $\mathcal{B}_1^{\mathcal{O}}$ simply answers with reject.

If any TAG or VRFY query contains a $\mathbf{v}$ which has appeared in a previous query, $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 and 0 otherwise. (Note that $\mathcal{B}_1^{\mathcal{O}}$ can compute the value $\mathbf{v}$ in a VRFY query as it knows π, h.)

Claim 6 *If $\mathcal{O} = \Gamma^*_{\tau,2\ell,d}(\mathbf{s}, \cdot)$, then $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 with probability $\geq \varepsilon'/2$.*

Proof (Proof of Claim). The answers to the TAG queries of $\mathcal{A}$ computed by $\mathcal{B}_1^{\mathcal{O}}$ have exactly the same distribution as in the uf-cma experiment (where the secret key is $(\mathbf{s}, h, \pi)$). The answers to the VRFY queries (which are always reject) are correct as long as $\mathcal{A}$ does not query a valid forgery. From our assumption, the probability that $\mathcal{A}$ finds a valid forgery that is *not* fresh is $> \varepsilon'/2$, which is thus a lower bound on the probability that $\mathcal{B}_1^{\mathcal{O}}$ outputs 1.

Claim 7 *If $\mathcal{O} = U_{2\ell+1}(\cdot)$, then $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 with probability $< Q^2/2^{\mu-2}$.*

Proof (Proof of Claim). The answers that $\mathcal{A}$ obtains on a TAG query $\mathbf{m}$ from $\mathcal{B}_1^{U_{2\ell+1}(\cdot)}$ (i.e., $\pi(\mathbf{R}, \mathbf{z}, \mathbf{b})$ where $\mathbf{R}, \mathbf{z}, \mathbf{b}$ are sampled uniformly) are uniformly random, and in particular independent of h or π. The answers to VRFY queries are always reject, and thus contain no information about h, π either. Then, we have that $\mathbf{v}_i = \mathbf{v}_j$ (where $\mathbf{v}_i = \mathsf{C}(h(\mathbf{m}_i, \mathbf{b}_i))$ is defined by the ith TAG or VRFY query) iff $h(\mathbf{m}_i, \mathbf{b}_i) = h(\mathbf{m}_j, \mathbf{b}_j)$.

$\mathcal{A}$ makes a total of Q queries. Assume that up to the $(i-1)$th query, all the $\mathbf{v}$'s were distinct. If the ith query is a TAG query, a fresh $\mathbf{b}_i$ is sampled which will be distinct from all previous $\mathbf{b}_j$ (for any $j < i$) with prob $1 - (i-1)/2^\nu$. Assuming this is the case, the probability that $h(\mathbf{m}_i, \mathbf{b}_i) = h(\mathbf{m}_j, \mathbf{b}_j)$ for any $j < i$ can be upper bounded by $i/2^\mu$ (here we use the fact that the answers that $\mathcal{A}$ gets from $\mathcal{B}_1^{U_{2\ell+1}(\cdot)}$ are uniformly random, and thus $\mathcal{A}$ has no information about h).

If the ith query is a VRFY query $(\mathbf{m}_i, \phi_i)$, then using the fact that π is a pairwise independent permutation (and $\mathcal{A}$ has no information about it) we can show that the probability that ϕ_i contains a $\mathbf{b}_i$ which is equal to some $\mathbf{b}_j$ (s.t.

[9] In this case where $\Pr[E_{\mathsf{fresh}}] \leq \varepsilon'/2$, we can even distinguish a $\mathsf{SLPN}^*_{\tau,2\ell,\ell}$ oracle from $U_{2\ell+1}(\cdot)$.

$\phi_j \neq \phi_i$) is $\leq i/2^{\nu+1}$. If this is the case then $(\mathbf{m}_i, \mathbf{b}_i) \neq (\mathbf{m}_j, \mathbf{b}_j)$ for all $j < i$ with overwhelming probability.[10] As in the previous case, we can then upper bound the probability that $h(\mathbf{m}_i, \mathbf{b}_i) = h(\mathbf{m}_j, \mathbf{b}_j)$ for any $j < i$ by $i/2^{\mu}$.

Using the union bound over all $i, 1 \leq i \leq Q$ we get the bound $Q^2/2^{\nu-2} = Q^2/2^{\mu-2}$ (recall that $\mu = \nu$) as claimed.

The case $\Pr[E_{\mathsf{fresh}}] > \varepsilon'/2$. In this case, $\mathcal{A}$ will make $\mathsf{TAG}, \mathsf{VRFY}$ queries, where with probability $> \varepsilon'/2$, at some point she will make an accepting VRFY query $(\mathbf{m}, \phi)$ that defines a fresh $\mathbf{v}$. We now construct an adversary $\mathcal{B}_2^{\mathcal{O}}$ that uses $\mathcal{A}$ as a black-box, and can distinguish $\mathcal{O} = \Gamma^*_{\tau,2\ell,d}(\mathbf{s}, \cdot)$ from $\mathcal{O} = U_{2\ell+1}(\cdot)$ (as in Definition 3) with advantage

$$\frac{\varepsilon'}{2^{\mu+1}} - Q_{\mathsf{tag}} \cdot \alpha'_{\ell,d} - Q_{\mathsf{vrfy}} \cdot \alpha''_{\tau',n}. \tag{B.2}$$

The construction of $\mathcal{B}_2^{\mathcal{O}}$ is very similar to the adversary $\mathcal{B}$ that we constructed in the proof of Theorem 1 (where we proved that the authentication protocol in Figure 4 is secure against active attacks). The queries to the prover in the first phase of an active attack directly correspond to TAG queries. However, we now have to additionally answer VRFY queries (we will always answer reject). Furthermore, we cannot choose the challenge $\mathbf{v}^*$ (as in the 2nd phase of an active attack). Instead, we will simply hope that (in the case where $\mathcal{O} = \Gamma^*_{\tau,2\ell,d}(\mathbf{s}, \cdot)$) the $\mathbf{v}$ contained in the first valid VRFY query (i.e. forgery) that $\mathcal{A}$ makes is fresh (which by assumption happens with probability $\varepsilon'/2$). Moreover, we will hope that it is the unique $\mathbf{v}^*$ (out of 2^{μ} possible ones) for which $\mathcal{B}_2^{\mathcal{O}}$ can verify this. This gives us a distinguishing advantage of nearly $\varepsilon'/2^{\mu+1}$ as stated in eq.(B.2). We do loose an additional additive term $Q_{\mathsf{tag}} \cdot \alpha'_{\ell,d}$ as there is an exponentially small probability that the transformation of subspace LPN samples to TAG queries will fail, and moreover an exponentially small term $Q_{\mathsf{vrfy}} \cdot \alpha''_{\tau',n}$ which accounts for the probability that $\mathcal{A}$ correctly guesses an accepting tag even in the case where $\mathcal{O} = U_{2\ell+1}(\cdot)$.

$\mathcal{B}_2^{\mathcal{O}}$ samples π, h (but not $\mathbf{s}$) as defined by KG, and $\mathbf{y}^* \xleftarrow{\$} \mathbb{Z}_2^{\mu}$, $\mathbf{s}^* \xleftarrow{\$} \mathbb{Z}_2^{2\ell}$. Let $\mathbf{v}^* := \mathsf{C}(\mathbf{y}^*)$. Next, $\mathcal{B}_2^{\mathcal{O}}$ invokes $\mathcal{A}$ and answers its queries as follows (the intuition for the sampling below is given in the proof of Claim 9).

- $\mathsf{TAG}(K, \cdot)$. The answer ϕ to a TAG query $\mathbf{m} \in \mathcal{M}$ is computed by $\mathcal{B}_2^{\mathcal{O}}$ as follows:
  1. Sample $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^{\nu}$ and compute $\mathbf{v} := \mathsf{C}(h(\mathbf{m}, \mathbf{b}))$. If $\mathbf{v} = \mathbf{v}^*$, output 0 and stop.
     Let $\mathbf{u} := \mathbf{v} \wedge \overline{\mathbf{v}}^*$ and $\mathbf{u}^* := \mathbf{v} \wedge \mathbf{v}^*$.
  2. For $i = 1, \ldots, n$, let $(\mathbf{R}'[i], \mathbf{z}'[i]) \xleftarrow{\$} \mathcal{O}(\mathbf{u})$, $\mathbf{R}''[i] \xleftarrow{\$} \mathbb{Z}_2^{2\ell}$ and $\mathbf{z}''[i] := \langle \mathbf{R}''[i], \mathbf{s}^* \wedge \mathbf{u}^* \rangle$.
     Let $\mathbf{R} = [\mathbf{R}[1], \mathbf{R}[2], \ldots, \mathbf{R}[n]]$ and $\mathbf{z} = [\mathbf{z}[1], \ldots, \mathbf{z}[n]]$ where $\mathbf{R}[i] := (\mathbf{R}'[i] \wedge \mathbf{u} \oplus \mathbf{R}''[i] \wedge \mathbf{u}^*)_{\downarrow \mathbf{v}}$ and $\mathbf{z}[i] := \mathbf{z}'[i] \oplus \mathbf{z}''[i]$.

[10] Note that for $j < i$ where $\phi_i = \phi_j$ we must have that $\mathbf{m}_i \neq \mathbf{m}_j$ since we assume that $\mathcal{A}$ does not repeat queries and does not ask VRFY queries $(\mathbf{m}, \phi)$ if ϕ was the output of a TAG query $\mathbf{m}$.

3. Return $\phi := \pi(\mathbf{R}, \mathbf{z}, \mathbf{b})$ to $\mathcal{A}$.

– $\mathsf{VRFY}(K, \cdot, \cdot)$. If $\mathcal{A}$ makes a VRFY query $(\phi, \mathbf{m})$, then $\mathcal{B}_2^{\mathcal{O}}$ always answers reject, but also makes the following check:
   1. Parse $\mathbf{y} := \pi^{-1}(\phi)$ as $[\mathbf{R} \in \mathbb{Z}_2^{\ell \times n}, \mathbf{z} \in \mathbb{Z}_2^n, \mathbf{b} \in \mathbb{Z}_2^\nu]$ and compute $\mathbf{v} := \mathsf{C}(h(\mathbf{m}, \mathbf{b}))$.
   2. If $\mathbf{v} \neq \mathbf{v}^*$, processing this query is over, otherwise go to the next step.
   3. If $\mathsf{rank}(\mathbf{R}) = n$ and $\mathbf{wt}(\mathbf{R}^\mathsf{T} \cdot \mathbf{s}^*_{\downarrow \mathbf{v}^*} \oplus \mathbf{z}) \leq n \cdot \tau'$ (i.e. we have a forgery) output 1 and stop.

If $\mathcal{A}$ has finished its queries, $\mathcal{B}_2^{\mathcal{O}}$ stops with output 0.

Claim 8 *If $\mathcal{O} = U_{2\ell+1}(\cdot)$, then $\mathcal{B}_2^{\mathcal{O}}$ outputs 1 with probability $\leq Q_{\mathsf{vrfy}} \cdot \alpha''_{\tau',n}$.*

Proof (Proof of Claim). The proof of this claim is almost identical to the proof of Claim 2, except that here we have an additional factor Q_{vrfy} as we have to take the union bound over all Q_{vrfy} queries, whereas in Claim 2 the adversary was (by definition of an active attack) only allowed one guess.

Claim 9 *If $\mathcal{O} = \Gamma^*_{\tau,2\ell,d}(\mathbf{s}, \cdot)$, then $\mathcal{B}_2^{\mathcal{O}}$ outputs 1 with probability $\geq \frac{\varepsilon'}{2^{\mu+1}}$.*

Proof. The proof of this claim is similar to the proof of Claim 3. $\mathcal{B}_2^{\Gamma^*_{\tau,2\ell,d}(\mathbf{s},\cdot)}$ perfectly simulates access to $\mathsf{TAG}(K, \cdot), \mathsf{VRFY}(K, \cdot, \cdot)$ oracles with key $K = (\mathbf{s}', h, \pi)$ where $\mathbf{s}' := (\mathbf{s}^* \wedge \mathbf{v}^*) \oplus (\mathbf{s} \wedge \overline{\mathbf{v}}^*)$ and h, π are sampled by $\mathcal{B}_2^{\mathcal{O}}$. By assumption, in this case, $\mathcal{A}$ outputs a valid fresh forgery with probability $\varepsilon'/2$. Conditioned on this, with probability $2^{-\mu}$, this fresh $\mathbf{v}$ will be $\mathbf{v}^*$ and therefore $\mathcal{B}_2^{\mathcal{O}}$ will output 1.

Summing up, using $\mathcal{A}$ we can break the subset LPN assumption with advantage which is given either by eq.(B.1) or eq.(B.2), i.e.

$$\varepsilon = \min \left\{ \varepsilon'/2 - \frac{Q^2}{2^{\mu-2}} \ , \ \frac{\varepsilon'}{2^{\mu+1}} - Q_{\mathsf{tag}} \cdot \alpha'_{\ell,d} - Q_{\mathsf{vrfy}} \cdot \alpha''_{\tau',n} \right\}$$

.

B.2 Proof of Theorem 5

The proof of the completeness error is similar to the schemes before and is omitted. As for security, let $\mathcal{A}$ be an adversary that successfully forges in the uf-cma experiment with probability ε'. We make the same conventions and the definition of freshness as in the proof of Theorem 4 and split the forging probability as $\Pr[E_{\mathsf{fresh}}] + \Pr[\neg E_{\mathsf{fresh}}] = \varepsilon'$.

The case $\Pr[E_{\mathsf{fresh}}] \leq \varepsilon'/2$. We now give the description of $\mathcal{B}_1^{\mathcal{O}}$ attacking the standard LPN problem, i.e. $\mathcal{B}_1^{\mathcal{O}}$ can distinguish $\mathcal{O} = \Lambda_{\tau,\ell}(\mathbf{s})$ from $\mathcal{O} = U_{\ell+1}$ with advantage

$$\varepsilon'/2 - \frac{Q^2}{2^{\mu-2}}. \tag{B.3}$$

Adversary $\mathcal{B}_1^{\mathcal{O}}$ samples π, h (but not $\mathbf{s}$) as defined by KG and $\mathbf{b}_i \xleftarrow{\$} \mathbb{Z}_2^\ell$, for $0 \leq i \leq \mu$. Next, it implicitly defines $\mathbf{s}_0 := \mathbf{s} \oplus \mathbf{b}_0$ (where $\mathbf{s}$ is unknown) and

$\mathbf{s}_i := \mathbf{b}_i$. It is easy to see that with this setup of $K = (\mathbf{s}_0, \ldots, \mathbf{s}_\mu, h, \pi)$ we have that for each $\mathbf{v} \in \mathbb{Z}_2^\mu$,

$$\mathbf{s}(\mathbf{v}) = \mathbf{s} \oplus \mathbf{b}(\mathbf{v}), \text{ where } \mathbf{b}(\mathbf{v}) := \mathbf{b}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{b}_i. \tag{B.4}$$

Note that adversary $\mathcal{B}_1^{\mathcal{O}}$ cannot evaluate $\mathbf{s}(\mathbf{v})$ but looking ahead, it will use its oracle $\mathcal{O}$ to answer $\mathcal{A}$'s queries as follows.

- If $\mathcal{A}$ makes a $\mathsf{TAG}(K, \cdot)$ query for message $\mathbf{m} \in \mathcal{M}$, then $\mathcal{B}_1^{\mathcal{O}}$ does the following:
  1. Samples $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^\nu$ and compute $\mathbf{v} := h(\mathbf{m}, \mathbf{b})$. Compute $\mathbf{b}(\mathbf{v})$ as in equation (B.4)
  2. Query the oracle $\mathcal{O}$ for n times: for $i = 1, \ldots, n$ let $(\mathbf{R}[i], \mathbf{z}'[i]) \xleftarrow{\$} \mathcal{O}$
  3. Return $\phi := \pi(\mathbf{R}, \mathbf{z}, \mathbf{b})$, where $\mathbf{z} = \mathbf{z}' \oplus \mathbf{R}^\mathsf{T} \cdot \mathbf{b}(\mathbf{v})$.
- If $\mathcal{A}$ makes a $\mathsf{VRFY}(K, \cdot, \cdot)$ query $(\mathbf{m}, \phi)$, $\mathcal{B}_1^{\mathcal{O}}$ simply answers with reject

Finally, if any TAG or VRFY query contains a $\mathbf{v}$ which has appeared in a previous query, $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 and stops. Otherwise, it outputs 0. Note that if $\mathcal{O} = \Lambda_{\tau,\ell}(\mathbf{s})$, then $\mathcal{B}_1^{\mathcal{O}}$ perfectly simulates the $\mathsf{TAG}(K, \cdot)$ oracle.

The following two claims are the analogs of Claims 6 and 7, respectively. Their proofs are essentially the same and are therefore omitted.

Claim 10 *If $\mathcal{O} = \Lambda_{\tau,\ell}(\mathbf{s})$, then $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 with probability $\geq \varepsilon'/2$.*

Claim 11 *If $\mathcal{O} = U_{\ell+1}$, then $\mathcal{B}_1^{\mathcal{O}}$ outputs 1 with probability $< \frac{Q^2}{2^{\mu-2}}$.*

Before we start dealing with the case $\Pr[E_{\mathsf{fresh}}] > \varepsilon'/2$, we recall the concept of encodings with full-rank differences over general finite fields $\mathbb{F}$. (For our proof we will only be interested in the case $\mathbb{F} = \mathbb{Z}_2$.)

Definition 4. *Let $k \geq 1$ be an integer and $\mathbb{F}$ be a finite field. A mapping $\varphi : \mathbb{F}^k \to \mathbb{F}^{k \times k}$ is an encoding with full-rank differences (FRD) over $\mathbb{F}$, if*

- *φ is computable in polynomial time (in k);*
- *for all distinct vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}^k$, $\varphi(\mathbf{a}) - \varphi(\mathbf{b})$ is an invertible matrix.*

Further, φ is homomorphic *if for all vectors $\mathbf{a}, \mathbf{b} \in \mathbb{F}^k$, $\varphi(\mathbf{a}) + \varphi(\mathbf{b}) = \varphi(\mathbf{a} + \mathbf{b})$.*

Cramer and Damgard [9] provided the following construction of an encoding with FRD. Let f be some polynomial of degree k in $\mathbb{F}[X]$ that is irreducible. Recall that if h is a polynomial over $\mathbb{F}[X]$, the polynomial $h \bmod f$ has degree less than k and therefore $\mathrm{coefs}(h \bmod f)$ can be viewed as a vector in $\mathbb{F}^k$. For an input $\mathbf{a} = (\mathbf{a}[0], \ldots, \mathbf{a}[k-1]) \in \mathbb{F}^k$ define the polynomial $g_{\mathbf{a}}(X) = \sum_{i=0}^{k-1} \mathbf{a}[i] X^i \in \mathbb{F}[X]$. Define

$$\varphi(\mathbf{a}) = \begin{pmatrix} \mathrm{coefs}(g_{\mathbf{a}}) \\ \mathrm{coefs}(X \cdot g_{\mathbf{a}} \bmod f) \\ \mathrm{coefs}(X^2 \cdot g_{\mathbf{a}} \bmod f) \\ \vdots \\ \mathrm{coefs}(X^{k-1} \cdot g_{\mathbf{a}} \bmod f) \end{pmatrix} \in \mathbb{F}^{k \times k}$$

As proved in [9], this is an encoding with FRD. Furthermore, it is also homomorphic since $\mathrm{coefs}(X^i \cdot g_{\mathbf{a}+\mathbf{b}} \bmod f) = \mathrm{coefs}(X^i \cdot g_{\mathbf{a}} \bmod f) + \mathrm{coefs}(X^i \cdot g_{\mathbf{b}} \bmod f)$.

Lemma 1. *For any prime $q \geq 2$ and integer $k \geq 1$, there exists an homomorphic encoding with full-rank differences over $\mathbb{Z}_q$.*

The case $\Pr[E_{\mathsf{fresh}}] > \varepsilon'/2$. We will use games, denoting by X_i the output of the experiment in the ith game.

Game 0: The uf-cma security experiment with the modification that it only returns 1 in case E_{fresh} happens. We have that $\Pr[X_0 \to 1] \geq \varepsilon'/2$.

Game 1: Let $k = \lceil \log_2(2Q) \rceil$. The experiment is the same as in Game 0 with the following differences.

- Key Generation. Algorithm KG additionally picks $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_2^k$ $(0 \leq i \leq \mu)$ and defines
$$\mathbf{a}(\mathbf{v}) := \mathbf{a}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{a}_i \in \mathbb{Z}_2^k. \tag{B.5}$$
- Tagging. For a query $\mathbf{m} \in \mathcal{M}$, algorithm TAG proceeds as follows. As in the last game, it computes tag ϕ and all the intermediate values. If $\mathbf{a}(\mathbf{v}) = 0^k \in \mathbb{Z}_2^k$ then the experiment stops and outputs 0. Otherwise, it outputs tag ϕ.
- Verification. For a query $(\mathbf{m}^*, \phi^*)$, algorithm VRFY proceeds as follows. As in the last game, it first computes the output d of the real VRFY algorithm, together with all intermediate values. If $d = \mathsf{accept}$ and $\mathbf{a}(\mathbf{v}^*) \neq 0^k \in \mathbb{Z}_2^k$, then it stops and outputs 0. Otherwise, it returns d.

Claim 12 $\Pr[X_0 \to 1] = 2Q \Pr[X_1 \to 1]$.

Proof. Let E_{fail} be the event that the execution in Game 1 stops, but not in Game 0, i.e.,

$$E_{\text{fail}} := \mathbf{a}(\mathbf{v}^*) = 0^k \wedge \mathbf{a}(\mathbf{v}_1) \neq 0^k \wedge \ldots \wedge \mathbf{a}(\mathbf{v}_Q) \neq 0^k,$$

where $\mathbf{v}_i$ is the $\mathbf{v}$ value appearing in the ith TAG query and $\mathbf{v}^*$ is the $\mathbf{v}$ value of the first fresh VRFY query. We use a variant of [7, Lemma 27] to show that

$$\frac{1}{2^k}\left(1 - \frac{Q}{2^k}\right) \leq \Pr_{\mathbf{a}}[E_{\text{fail}}] \leq \frac{1}{2^k}, \tag{B.6}$$

where the probability is taken over $\mathbf{a} = (\mathbf{a}_0, \ldots, \mathbf{a}_\mu) \xleftarrow{\$} (\mathbb{Z}_2^k)^{\mu+1}$. Note that $\mathbf{a}(\cdot)$ from (B.5) is essentially pairwise independent over $\mathbb{Z}_2^k$, i.e. for each $\mathbf{v}_i \neq \mathbf{v}^* \in \mathbb{Z}_2^\mu$, we have $\Pr[\mathbf{a}(\mathbf{v}^*) = 0^k] = 1/2^k$ and $\Pr[\mathbf{a}(\mathbf{v}_i) = 0^k \mid \mathbf{a}(\mathbf{v}^*) = 0^k] = 1/2^k$. Then (B.6) follows by applying the union bound. The claim now follows by the definition of $k = \lceil \log_2(2Q) \rceil$.

Game 2: Oracle $\mathsf{TAG}(K, \cdot)$ now internally uses uniform $(\mathbf{R}, \mathbf{z}) \in \mathbb{Z}_2^{\ell \times n} \times \mathbb{Z}_2^n$ to generate tag ϕ on message $\mathbf{m}$.

Claim 13 $\Pr[X_1 = 1] - \Pr[X_2 = 1] \leq \varepsilon$.

Proof. Assume there exists an adversary $\mathcal{A}$ that can distinguish between the two games. We now describe adversary $\mathcal{B}_2^{\mathcal{O}(\cdot,\cdot)}$ that (t, nQ, ε)-breaks the $\mathsf{SLPN}_{\tau,\ell,\ell}$ problem.

– KG. Let $k = \lceil \log_2(2Q) \rceil$ and $\varphi : \mathbb{Z}_2^k \to \mathbb{Z}_2^{k\times k}$ be an homomorphic encoding with full-rank differences over $\mathbb{Z}_2$ from Lemma 1. $\mathcal{B}_2^{\mathcal{O}}$ picks $\mathbf{a}_i \xleftarrow{\$} \mathbb{Z}_2^k$, $\mathbf{b}_i \xleftarrow{\$} \mathbb{Z}_2^\ell$ (for $i = 0, \ldots, \mu$) and (implicitly) defines $\mathbf{s}_i = \mathbf{A}_i \cdot \mathbf{s} \oplus \mathbf{b}_i$, where $\mathbf{s}$ is the (unknown) secret from the oracle $\mathit{\Gamma}_{\tau,\ell,\ell}(\mathbf{s},\cdot,\cdot)$ and

$$\mathbf{A}_i := \begin{pmatrix} \underbrace{\boxed{\varphi(\mathbf{a}_i)}}_{\in \mathbb{Z}_2^{k\times k}} & 0 & \cdots & 0 \\ 0 & \boxed{\varphi(\mathbf{a}_i)} & & \vdots \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & & \boxed{\varphi(\mathbf{a}_i)} \end{pmatrix} \in \mathbb{Z}_2^{\ell\times\ell}.$$

(Here we assume that $k|\ell$. If that is not the case, the matrix $\mathbf{A}_i$ is truncated.) This way we have that $\mathbf{s}(\mathbf{v}) = \mathbf{A}(\mathbf{v})\cdot\mathbf{s}\oplus\mathbf{b}(\mathbf{v})$, where $\mathbf{A}(\mathbf{v}) = \mathbf{A}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{A}_i$ and $\mathbf{b}(\mathbf{v}) = \mathbf{b}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{b}_i$. Define $\mathbf{a}(\mathbf{v}) := \mathbf{a}_0 \oplus \bigoplus_{i:\mathbf{v}[i]=1} \mathbf{a}_i$ as in (B.5). By the properties of the homomorphic encoding with FRD we have that

$$\mathbf{A}(\mathbf{v}) = \begin{cases} 0^{\ell\times\ell} & \text{if } \mathbf{a}(\mathbf{v}) = 0^k \\ \text{full-rank matrix} & \text{if } \mathbf{a}(\mathbf{v}) \neq 0^k \end{cases} \tag{B.7}$$

– Next, $\mathcal{B}_2^{\mathcal{O}}$ invokes $\mathcal{A}$. The answer $(\mathbf{R}, \mathbf{z})$ to a $\mathsf{TAG}(K,\cdot)$ query $\mathbf{m} \in \mathcal{M}$ by $\mathcal{A}$ is computed by $\mathcal{B}_2^{\mathcal{O}}$ as follows.
  1. Sample $\mathbf{b} \xleftarrow{\$} \mathbb{Z}_2^\nu$ and compute $\mathbf{v} := h(\mathbf{m}, \mathbf{b})$.
  2. Compute $\mathbf{a}(\mathbf{v})$, $\mathbf{b}(\mathbf{v})$ and $\mathbf{A}$. If $\mathbf{a}(\mathbf{v}) = 0^k$, then stop and output 0. If $\mathbf{a}(\mathbf{v}) \neq 0^k$ we have that $\mathbf{A}(\mathbf{v})$ is a full-rank matrix by (B.7)
  3. Query $\mathcal{O}(\cdot,\cdot)$ for n times on input $(\mathbf{A}(\mathbf{v}), \mathbf{b}(\mathbf{v}))$: for $i = 1, \ldots, n$ let $(\mathbf{R}[i], \mathbf{z}[i]) \xleftarrow{\$} \mathcal{O}(\mathbf{A}(\mathbf{v}), \mathbf{b}(\mathbf{v}))$
  4. Return $\phi = \pi(\mathbf{R}, \mathbf{z}, \mathbf{b})$ to $\mathcal{A}$.

– If $\mathcal{A}$ makes a verification query consisting of $\mathbf{m}^*$ and a tag $\phi^* = \pi(\mathbf{R}^*, \mathbf{z}^*, \mathbf{b}^*)$, then $\mathcal{B}_2^{\mathcal{O}}$ computes $\mathbf{v}^* = h(\mathbf{m}^*, \mathbf{b}^*)$. If $\mathbf{a}(\mathbf{v}^*) \neq 0^k$ then $\mathcal{B}_2^{\mathcal{O}}$ returns reject. Otherwise, we have that $\mathbf{s}(\mathbf{v}^*) = \mathbf{b}(\mathbf{v}^*)$ does not depend on the unknown secret $\mathbf{s}$ anymore and $\mathcal{B}_2^{\mathcal{O}}$ can perform the real check which is

$$\mathsf{rank}(\mathbf{R}^*) = n \quad \text{and} \quad \mathbf{wt}(\mathbf{z}^* \oplus \mathbf{R}^{*\mathsf{T}} \cdot \mathbf{b}(\mathbf{v}^*)) \leq n \cdot \tau'. \tag{B.8}$$

The output is 1 if both checks succeed and 0 otherwise.

We now analyze $\mathcal{B}_2^{\mathcal{O}}$. If $\mathcal{O}(\cdot,\cdot) = \mathit{\Gamma}_{\tau,\ell,\ell}(\mathbf{s},\cdot,\cdot)$ is the real subspace LPN oracle, then the answers that $\mathcal{B}_2^{\mathcal{O}}$ gives to $\mathcal{A}$'s queries in the first phase of the attack have *exactly* the same distribution as what $\mathcal{A}$ would get in Game 1. (That is, the $\mathbf{z}[i]$ from the oracle are of the form $\mathbf{z}[i] = \mathbf{R}[i]^\mathsf{T}(\mathbf{A}(\mathbf{v})\cdot\mathbf{s}\oplus\mathbf{b}(\mathbf{v}))\oplus\mathbf{e} = \mathbf{R}[i]^\mathsf{T}\cdot\mathbf{s}(\mathbf{v})\oplus\mathbf{e}$, as in Game 1.) Hence $\Pr[\mathcal{B}_2^{\mathit{\Gamma}_{\tau,\ell,\ell}(\mathbf{s},\cdot,\cdot)} \to 1] = \Pr[X_1 \to 1]$. If $\mathcal{O} = U_{\ell+1}(\cdot,\cdot)$ is the uniform oracle, then all the outputs made by the TAG oracle are uniformly random. Hence $\Pr[\mathcal{B}_2^{U_{\ell+1}(\cdot,\cdot)} \to 1] = \Pr[X_2 \to 1]$.

Claim 14 $\Pr[X_2 = 1] \leq \alpha''_{\tau',n} = 2^{-\Theta(n)}$.

Proof (Proof of Claim). If $\mathbf{R}^*$ does not have full rank then the experiment outputs 0 by definition. So from now we only consider the case where $\mathsf{rank}(\mathbf{R}^*) = n$. In Game 2, the answers $(\mathbf{R}, \mathbf{z})$ adversary $\mathcal{A}$ obtains from the TAG oracle are independent of the secrets $\mathbf{s}_0, \ldots, \mathbf{s}_\mu$. Since $\mathbf{s}(\mathbf{v}^*)$ is uniformly random and $\mathbf{R}^*$ has full rank, the vector $\mathbf{x} := \mathbf{R}^{*\mathsf{T}} \cdot \mathbf{s}(\mathbf{v}^*) \oplus \mathbf{z}^*$ is uniformly random over $\mathbb{Z}_2^n$. Thus the probability that the second verification $\mathbf{wt}(\mathbf{z}^* \oplus \mathbf{R}^{*\mathsf{T}} \cdot \mathbf{s}(\mathbf{v}^*)) \leq n \cdot \tau'$ fails is $\Pr[\mathbf{wt}(\mathbf{x}) \leq n \cdot \tau'] = \alpha''_{\tau',n} = 2^{-\Theta(n)}$.

To sum up, in the case $\Pr[E_{\mathsf{fresh}}] > \varepsilon'/2$ (putting all together the terms in Claim 12—14), we can use $\mathcal{A}$ to break the subspace LPN assumption with advantage $\frac{\varepsilon'}{4Q} - 2^{-\Theta(n)}$. On the other hand in the case $\Pr[E_{\mathsf{fresh}}] \leq \varepsilon'/2$, we have an advantage as given in eq. (B.3). Thus $\epsilon = \min\left\{\varepsilon'/2 - \frac{Q^2}{2^{\mu-2}}, \frac{\varepsilon'}{4Q} - 2^{-\Theta(n)}\right\}$.

C A Technical Lemma

The following lemma was used in Section 3.1.

Lemma 2. *For $n, t \in \mathbb{Z}$, let $\Delta(n, d)$ denote the probability that a random matrix in $\mathbb{Z}_2^{n+d\times n}$ has rank less than n, then*

$$\Delta(n, d) < 2^{-d} \ .$$

Proof. Assume we sample the n columns of a matrix $M \in \mathbb{Z}_2^{n+d\times n}$ one by one. For $i = 1, \ldots, n$ let E_i denote the event that the first i columns are linearly independent, then

$$\Pr[\neg E_i | E_{i-1}] = \frac{2^{i-1}}{2^{n+d}} = 2^{i-1-n-d}$$

as $\neg E_i$ happens iff the ith column (sampled uniformly from a space of size 2^{n+d}) falls into the space (of size 2^{i-1}) spanned by the first $i-1$ columns. We get further

$$\Delta(n, d) = \Pr[\neg E_n] = \sum_{i=1}^{n} \Pr[\neg E_i | E_{i-1}] = \sum_{i=1}^{n} 2^{i-1-n-d} \leq 2^{-d}.$$