Message Authentication, Revisited

Yevgeniy Dodis[1], Eike Kiltz[2⋆], Krzysztof Pietrzak[2⋆⋆], and Daniel Wichs[4]

[1] New York University
[2] Ruhr-Universität Bochum
[3] IST Austria
[4] IBM T. J. Watson Research Center

Abstract. Traditionally, symmetric-key message authentication codes (MACs) are easily built from pseudorandom functions (PRFs). In this work we propose a wide variety of other approaches to building efficient MACs, without going through a PRF first. In particular, unlike deterministic PRF-based MACs, where each message has a unique valid tag, we give a number of *probabilistic* MAC constructions from various other primitives/assumptions. Our main results are summarized as follows:

- We show several new probabilistic MAC constructions from a variety of general assumptions, including CCA-secure encryption, Hash Proof Systems and key-homomorphic weak PRFs. By instantiating these frameworks under concrete number theoretic assumptions, we get several schemes which are more efficient than just using a state-of-the-art PRF instantiation under the corresponding assumption.
- For probabilistic MACs, unlike deterministic ones, unforgeability against a chosen message attack (uf-cma) alone does not imply security if the adversary can additionally make verification queries (uf-cmva). We give an *efficient* generic transformation from any uf-cma secure MAC which is "message-hiding" into a uf-cmva secure MAC. This resolves the main open problem of Kiltz et al. from Eurocrypt'11; By using our transformation on their constructions, we get the first efficient MACs from the LPN assumption.
- While all our new MAC constructions immediately give efficient actively secure, two-round symmetric-key identification schemes, we also show a very simple, three-round actively secure identification protocol from *any weak PRF*. In particular, the resulting protocol is much more efficient than the trivial approach of building a regular PRF from a weak PRF.

1 Introduction

Message Authentication Codes (MACs) are one of the most fundamental primitives in cryptography. Historically, a vast majority of MAC constructions are

⋆ Supported by a Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation, funded by the German Federal Ministry for Education and Research.
⋆⋆ Supported by the European Research Council/ERC Starting Grant 259668-PSPC.

based on pseudorandom functions (PRFs).[5] In particular, since a PRF with large output domain is also a MAC, most research on symmetric-key authentication concentrated on designing and improving various PRF constructions. This is done either using very fast heuristic constructions, such as block-cipher based PRFs (e.g., CBC-MAC [6, 8] or HMAC [5, 4]), or using elegant, but slower number-theoretic constructions, such as the Naor-Reingold (NR) PRF [33]. The former have the speed advantage, but cannot be reduced to simple number-theoretic hardness assumptions (such as the DDH assumption for NR-PRF), and are not friendly to efficient zero-knowledge proofs about authenticated messages and/or their tags, which are needed in some important applications, such as compact e-cash [12]. On the other hand, the latter are comparably inefficient, due to their reliance on number theory. Somewhat surprisingly, the inefficiency of existing number-theoretic PRFs goes beyond what one would expect by the mere fact that "symmetric-key" operations are replaced by the more expensive "public-key" operations. For example, when building a PRF based on discrete-log-type of assumptions, such as DDH, one would naturally expect that the secret key would contain a constant number of group elements/exponents, and the PRF evaluation should cost at most a constant number of exponentiations. In contrast, state-of-the art discrete-log-type PRFs either require a key of quadratic size in the security parameter (e.g. the NR PRF [33]), or a number of exponentiations linear in the security parameter (e.g., tree-type PRFs based on the GGM transform [20] applied to some discrete-log-type pseudorandom generator), or are based on exotic and relatively untested assumptions (e.g., Dodis-Yampolskiy PRF [17] based on the so called "q-DDHI" assumption). In particular, to the best of our knowledge, prior to this work it was unknown how to build a MAC (let alone a PRF) based on the classical DDH assumption, where the secret key consists of a constant number of group elements / exponents and the MAC evaluation only require a constant number of exponentiations.

Of course, one way to improve such deficiencies of existing "algebraic MACs" would be to improve the corresponding "algebraic PRF" constructions. However, as the starting point of our work, we observe that there might exist alternative approaches to building efficient MACs, *without going through a PRF first*. For example, MACs only need to be unpredictable, so we might be able to build efficient MACs from *computational assumptions* (e.g., CDH rather than DDH), without expensive transformations from unpredictability-to-pseudorandomness [34]. Alternatively, even when relying on decisional assumptions (e.g. DDH), MAC constructions are allowed to be *probabilistic*. In contrast, building a PRF effectively forces one to design a MAC where there is only one valid tag for each message, which turns out to be a serious limitation for algebraic constructions.[6]

[5] Or block ciphers, which, for the purposes of analysis, are anyway treated as length-preserving PRFs.

[6] The observation that probabilistic MAC might have advantages over the folklore "PRF-is-a-MAC" paradigm is not new, and goes back to at least Wegman and Carter [40], and several other follow-up works (e.g., [30, 25, 16]). However, most prior probabilistic MACs were still explicitly based on a PRF or a block cipher.

For example, it is instructive to look at the corresponding "public-key domain" of digital signatures, where forcing the scheme to have a unique valid signature appears to be very hard [32, 11] and, yet, not necessary for most applications of digital signatures. In particular, prominent digital signature schemes in the standard model[7] [11, 39] are all probabilistic. In fact, such signature schemes trivially give MACs. Of course, such MACs are not necessarily as efficient as they could be, since they "unnecessarily" support public verification.[8] However, the point is that such trivial signature-based constructions already give a way to build relatively efficient "algebraic MACs" *without building an "algebraic PRF" first.*

Yet another motivation to building probabilistic MAC comes from the desire of building efficient MACs (and, more generally, symmetric-key authentication protocols) from the *Learning Parity with Noise* [24, 26, 28, 29] (LPN) assumption. This very simple assumption states that one cannot recover a random vector x from any polynomial number of noisy parities $(a, \langle a, x\rangle + e)$, where a is a random vector and e is small *random* noise, and typically leads to very simple and efficient schemes [19, 2, 38, 24, 26, 28, 29]. However, the critical dependence on random errors makes it very hard to design deterministic primitives, such as PRFs, from the LPN assumption. Interestingly, this ambitious challenge was very recently overcome for a more complicated *Learning With Errors* (LWE) assumption by [3], who build a PRF based on a new (but natural) variant of the LWE assumption. However, the resulting PRF has the same deficiencies (e.g., large secret key) as the NR-PRF, and is *much* less efficient than the direct probabilistic MAC constructions from LPN/LWE assumptions recently obtained by [29].

1.1 Our Results

Motivated by the above considerations, in this work we initiate a systematic study of different methods for building efficient probabilistic MACs from a variety assumptions, both general and specific, without going through the PRF route. Our results can be summarized as follows:

Dealing with Verification Queries and Other Transformations. The desired notion of security for probabilistic MACs is called "unforgeability against chosen message and verification attack" uf-cmva, where an attacker can arbitrarily interleave tagging queries (also called signing queries) and verification queries. For deterministic MACs, where every message corresponds to exactly one possible tag, this notion is equivalent to just considering a weaker notion called uf-cma (unforgeability under chosen message attack) where the attacker can only make

[7] In fact, even in the random oracle model there are noticeable advantages. E.g., full domain hash (FDH) signatures [9] have worse exact security than *probabilistic* FDH signatures, while Fiat-Shamir signatures [18] are inherently probabilistic.

[8] Indeed, one of our results, described shortly, will be about "optimizing" such signature-based constructions.

tagging queries but *no* verification queries. This is because, in the deterministic case, the answers to verification queries are completely predictable to an attacker: for any message for which a tagging query was already made the attacker knows the unique tag on which the verification oracle will answer affirmatively, and for any new message finding such a tag would be equivalent to breaking security without the help of the verification oracle. Unfortunately, as discussed by [7], the situation is more complicated for the case of probabilistic MACs where the attacker might potentially get additional information by modifying a valid tag of some message and seeing if this modified tag is still valid for the same message. In fact, some important MAC constructions, such as the already mentioned "basic" LPN-based construction of [29], suffer from such attacks and are only uf-cma, but not uf-cmva secure.

In Section 3 we give several general transformations for probabilistic MACs. The most important one, illustrated in Figure 1, *efficiently* turns a uf-cma secure (i.e. unforgeable without verification queries) MAC which is "message hiding" (a property we call ind-cma) into a uf-cmva secure (i.e. unforgeable with verification queries) MAC. This transformation is very efficient, requiring just a small amount of extra randomness and one invocation of a pairwise independent hash function with fairly short output.

This transformation solves the main open problem left in Kiltz et al. [29], who construct uf-cmva MACs from the learning parity with noise (LPN) problem. We remark that [29] already implicitly give an uf-cma to uf-cmva transformation, but it is quite inefficient, requiring the evaluation of a pairwise-independent *permutation* over the entire tag of a uf-cma secure MAC. We list the two constructions of uf-cma and suf-cma LPN based MACs from [29] in Section 4.5. Using our transformations, we get uf-cmva secure MACs with basically the same efficiency as these constructions.

Our second transformation extends the domain of an ind-cma secure MAC. A well known technique to extend the domain of PRFs is the "hash then encrypt" approach where one applies an almost universal hash function to the (long) input before applying the PRF. This approach fails for MACs, but we show that it works if the MAC is ind-cma secure. A similar observation has been already made by Bellare [4] for "privacy preserving" MACs.

The last transformation, which actually does nothing except possibly restricting the message domain, states that a MAC which is only selectively secure is also fully secure, albeit with quite a large loss in security. Such a transformation was already proposed in the context of identity based encryption [10], and used implicitly in the construction of LPN based MACs in [29].

New Constructions of Probabilistic MACs. In Section 4, we present a wide variety of new MAC constructions.

First, we show how to build an efficient MAC from any chosen ciphertext attack (CCA) secure (symmetric- or public-key) encryption. At first glance, using CCA-secure encryption seems like a complete "overkill" for building MACs. In fact, in the symmetric-key setting most CCA-secure encryption schemes are actually built *from MACs*; e.g., via the encrypt-then-MAC paradigm. However,

if we are interested in obtaining number-theoretic/algebraic MACs using this approach, we would start with *public-key* CCA-secure encryption, such as Cramer-Shoup encryption [15] or many of the subsequent schemes (e.g. [31, 22, 23, 37, 21]). Quite remarkably, CCA-secure encryption has received so much attention lately, and the state-of-the-art constructions are so optimized by now, that the MACs resulting from our simple transformation appear to be *better*, at least in certain criteria, than the existing PRF constructions from the same assumptions. For example, by using any state-of-the-art DDH-based scheme, such as those by [15, 31, 22], we immediately obtain a probabilistic DDH-based MAC where both the secret key and the tag are of constant size, and the tagging/verification each take a constant number of exponentiations. As we mentioned, no such DDH-based MAC was known prior to our work. In fact, several recent constructions built efficient CCA-secure encryption schemes from *computational assumptions*, such as CDH and factoring [13, 23, 21]. Although those schemes are less efficient than the corresponding schemes based on decisional assumptions, they appear to be more efficient than (or at least comparable with) the best known PRF constructions from the same assumption. For example, the best factoring-based PRF of [35] has a quadratic-size secret key, while our construction based on the Hofheinz-Kiltz [23] CCA-encryption from factoring would have a linear-size (constant number of group elements) secret key.

Second, we give an efficient MAC construction from any *Hash Proof Systems* (HPS) [15]. Hash Proof Systems were originally defined [15] for the purpose of building CCA-secure public-key encryption schemes, but have found many other applications since. Here we continue this trend and give a direct MAC construction from HPS, which is more optimized than building a CCA-secure encryption from HPS, and then applying our prior transformation above.

Third, we give a simple construction of probabilistic MACs from any *key-homomorphic weak PRF* (hwPRF). Recall, a weak PRF [33] is a weakening of a regular PRF, where the attacker can only see the PRF value at random points. This weakening might result in much more efficient instantiations for a variety of number-theoretic assumptions. For example, under the DDH assumption, the basic modulo exponentiation $f_k(m) = m^k$ is already a weak PRF, while the regular NR-PRF from DDH is much less efficient. We say that such a weak PRF $f_k(m)$ is *key homomorphic* (over appropriate algebraic domain and range) if $f_{ak_1+bk_2}(m) = a \cdot f_{k_1}(m) + b \cdot f_{k_2}(m)$. (For example, the DDH-based weak PRF above clearly has this property.) We actually give two probabilistic MACs from any hwPRF. Our basic MAC is very simple and efficient, but only achieves so called *selective* security, meaning that the attacker has to commit to the message to be forged before the start of the attack. It is somewhat reminiscent (in terms of its design and proof technique, but not in any formal way) to the Boneh-Boyen selectively-secure signature scheme [11]. In contrast, our second construction borrows the ideas from (fully secure) Waters signature scheme [39], and builds a less efficient standard MAC from any hwPRF. Interestingly, both constructions are only uf-cma secure, but do not appear to be uf-cmva-secure. Luckily, our MACs are easily seen to be "message-hiding" (i.e., ind-cma-secure),

so we can apply our efficient generic transformation to argue full uf-cmva security for both resulting constructions.

Our final MAC constructions are from signature schemes. Recall, any signature scheme trivially gives a MAC which "unnecessarily" supports public verification. This suggests that such constructions might be subject to significant optimizations when "downgraded" into a MAC, both in terms of efficiency and the underlying security assumption. Indeed, we show that this is true for the (selectively-secure) Boneh-Boyen [11] signature scheme, and the (fully-secure) Waters [39] signature schemes. For example, as signatures, both schemes require a bilinear group with a pairing, and are based on the CDH assumption in such a group. We make a simple observation that when public verification is no longer required, no pairing computations are needed, and standard (non-bilinear) groups can be used. However, in doing so we can only prove (selective or full) security under the *gap-Diffie-Hellman assumption*, which states that CDH is still hard even given the DDH oracle. Luckily, we show how to apply the "twinning" technique of Cash et al. [13] to get efficient MAC variants of both schemes which can be proven secure under the standard CDH assumption.

Symmetric-Key Authentication Protocols. While all our new MAC constructions immediately give efficient actively secure, two-round symmetric-key identification schemes, in Section 4.6 we also show a very simple, three-round actively secure identification protocol from *any* weak PRF (wPRF). In particular, the resulting protocol is much more efficient than the trivial approach of building a regular PRF from a weak PRF [33], and then doing the standard PRF-based authentication. Given that all our prior MAC constructions required some algebraic structure (which was indeed one of our motivations), we find a general (and very efficient) construction of actively secure authentication protocols from any wPRF to be very interesting.

Our protocol could be viewed as an abstraction of the LPN-based actively secure authentication protocol of Katz and Shin [27], which in turn consists of a parallel repetition of the HB^+ protocol of Juels and Weiss [26]. Although the LPN based setting introduces some complications due to handling of the errors, the high level of our protocol and the security proof abstracts away the corresponding proofs from [27, 26]. In fact, we could relax the notion of wPRF slightly to allow for probabilistic computation with approximate correctness, so that the protocol of [27] will become a special case of our wPRF-based protocol.

2 Definitions

2.1 Notation

We denote the set of integers modulo an integer $q \geq 1$ by $\mathbb{Z}_q$. For a positive integer k, $[k]$ denotes the set $\{1, \ldots, k\}$; $[0]$ is the empty set. For a set $\mathcal{X}$, $x \leftarrow_R \mathcal{X}$ denotes sampling x from $\mathcal{X}$ according to the uniform distribution.

2.2 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 $k \leftarrow \mathsf{KG}(1^\lambda)$ 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 $\sigma \leftarrow \mathsf{TAG}_k(m)$ takes as input a secret key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$ and outputs an authentication tag $\sigma \in \mathcal{T}$.
- Verification. The deterministic verification algorithm $\mathsf{VRFY}_k(m, \sigma)$ takes as input a secret key $k \in \mathcal{K}$, a message $m \in \mathcal{M}$ and a tag $\sigma \in \mathcal{T}$ and outputs a decision: $\{\mathsf{accept}, \mathsf{reject}\}$.

If the TAG algorithm is deterministic one does not have to explicitly define VRFY, since it is already defined by the TAG algorithm as $\mathsf{VRFY}_k(m, \sigma) = \mathsf{accept}$ iff $\mathsf{TAG}_k(m) = \sigma$. We say that MAC has completeness error α if for all $m \in \mathcal{M}$ and $\lambda \in \mathbb{N}$,

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

Security. The standard security notion for a *randomized* MAC is unforgeability under chosen message and chosen verification queries attack ($\mathsf{uf\text{-}cmva}$). We denote by $\mathsf{Adv}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T, Q_V)$, the advantage of the adversary A in forging a message for a random key $k \leftarrow \mathsf{KG}(1^\lambda)$, where A can make Q_T queries to $\mathsf{TAG}_k(\cdot)$ and Q_V queries to $\mathsf{VRFY}_k(\cdot, \cdot)$. Formally this is the probability that the following experiment outputs 1.

Experiment $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T, Q_V)$
$k \leftarrow \mathsf{KG}(1^\lambda)$
Invoke $\mathsf{A}^{\mathsf{TAG}_k(\cdot), \mathsf{VRFY}_k(\cdot,\cdot)}$ who can make up to Q_T queries to $\mathsf{TAG}_k(\cdot)$ and Q_V queries to $\mathsf{VRFY}_k(\cdot, \cdot)$.
Output 1 if A made a query (m^*, σ^*) to $\mathsf{VRFY}_k(\cdot, \cdot)$ where
1. $\mathsf{VRFY}_k(m^*, \sigma) = \mathsf{accept}$
2. A did not already make the query m^* to $\mathsf{TAG}_k(\cdot)$
Output 0 otherwise.

We also define a weaker notion of *selective security*, captured by the experiment $\mathbf{Exp}^{\mathsf{suf\text{-}cmva}}_{\mathsf{MAC}}$, which is defined in the same way as above with the only difference that A has to specify to the target message m^* (that causes the experiment to output 1) ahead of time, before making any queries to its oracles.

Definition 1 ((Selective) unforgeability under chosen message (& verification) attack.). *A* MAC *is* $(t, Q_T, Q_V, \varepsilon)$-$\mathsf{uf\text{-}cmva}$ *secure if for any* A *running in time* t *we have* $\Pr[\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] \leq \varepsilon$. *It is* (t, Q_T, ε)-$\mathsf{uf\text{-}cma}$ *secure if it is* $(t, Q_T, 1, \varepsilon)$-$\mathsf{uf\text{-}cmva}$-*secure. That is,* $\mathsf{uf\text{-}cma}$ *security does not allow the adversary to make any verification queries except for the one forgery attempt. We also define the* selective security *notions* $\mathsf{suf\text{-}cma}$ *and* $\mathsf{suf\text{-}cmva}$ *security analogously by considering the experiment* $\mathbf{Exp}^{\mathsf{suf\text{-}cmva}}(\mathsf{MAC})$.

In the next section we show a simple generic transformation which turns any uf-cma-secure MAC into a uf-cmva-secure $\overline{\mathsf{MAC}}$. For this transformation to work, we need one extra non-standard property for MAC to hold, namely that tags computationally "hide" the message. A similar notion called "privacy preserving MACs" was considered by Bellare [4]. His notion is for deterministic MACs, whereas our notion can only be achieved for probabilistic MACs.

Definition 2 (ind-cma: indistinguishability under chosen message attack). *A* MAC *is* (t, Q_T, ε)-ind-cma *secure if no adversary* A *running in time* t *can distinguish tags for chosen messages from tags for a fixed message, say* 0*, i.e.*

$$\left| \Pr_{k \leftarrow \mathsf{KG}(1^\lambda)}[\mathsf{A}^{\mathsf{TAG}_k(\cdot)}(1^\lambda) = 1] - \Pr_{k \leftarrow \mathsf{KG}(1^\lambda)}[\mathsf{A}^{\mathsf{TAG}_k(0)}(1^\lambda) = 1] \right| \leq \varepsilon \ .$$

Here $\mathsf{TAG}_k(0)$ is an oracle which ignores its input, and outputs a tag for some fixed message 0 using key K. Note that a MAC that is secure against ind-cma adversaries must be probabilistic, otherwise A can trivially distinguish by queries on two different messages $m \neq m'$, and checking if the tags she receives are identical, which will be the case iff the oracle implements $\mathsf{TAG}_k(0)$.

3 Transformations for MACs

In this section we give some general transformations for MACs as discussed in the introduction.

3.1 From One to Multiple Verification Queries: uf-cma + ind-cma ⇒ uf-cmva

Let $\mu = \mu(\lambda)$ denote a statistical security parameter and let $\mathcal{H}$ be a family of pairwise independent hash functions $h : \mathcal{T} \to \{0,1\}^\mu$. From $\mathsf{MAC} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with key space $\mathcal{K}$, message space $\mathcal{M} \times \{0,1\}^\mu$, and tag space $\mathcal{T}$ we construct $\overline{\mathsf{MAC}} = \{\overline{\mathsf{KG}}, \overline{\mathsf{TAG}}, \overline{\mathsf{VRFY}}\}$ with key space $\mathcal{K} \times \mathcal{H}$, message space $\mathcal{M}$, and tag space $\mathcal{T} \times \{0,1\}^\mu$ as follows.

- Key Generation. Algorithm $\overline{\mathsf{KG}}(1^\lambda)$ runs $k \leftarrow \mathsf{KG}(1^\lambda)$ and samples a pairwise independent hash function $h \leftarrow \mathcal{H}$ with $h : \mathcal{T} \to \{0,1\}^\mu$. It outputs (k, h) as the secret key.
- Tagging. The tagging algorithm $\overline{\mathsf{TAG}}_{(k,h)}(m)$ samples $b \leftarrow_R \{0,1\}^\mu$ and runs $z \leftarrow \mathsf{TAG}_k(m \| b)$. It returns $(z, h(z) \oplus b)$ as the tag.

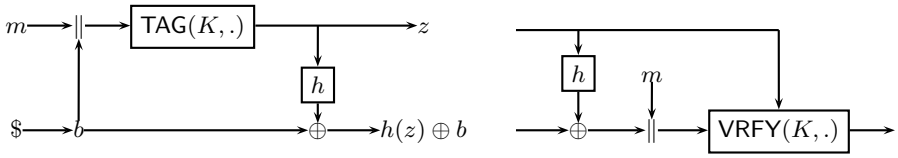

Fig. 1. $\overline{\mathsf{TAG}}$ and $\overline{\mathsf{VRFY}}$ with key (k, h), message m and randomness b.

– Verification. The verification algorithm $\overline{\mathsf{VRFY}}_{(k,h)}(m,(z,y))$ computes $b = y \oplus h(z)$ and outputs $\mathsf{VRFY}_k(m\|b,z)$.

Theorem 1 (uf-cma + ind-cma $\Rightarrow$ uf-cmva)**.** *For any $t, Q_T, Q_V \in \mathbb{N}$, $\varepsilon > 0$, if* MAC *is*
– (t, Q_T, ε)-uf-cma *secure (unforgeable with no verification queries)*
– (t, Q_T, ε)-ind-cma *secure (indistinguishable)*
then $\overline{\mathsf{MAC}}$ *is* $(t', Q_T, Q_V, \varepsilon')$-uf-cmva *secure (unforgeable with verification queries) where*

$$t' \approx t \qquad \varepsilon' = 2Q_V\varepsilon + 2Q_VQ_T/2^\mu.$$

The proof of Theorem 1 can be found in the full version of this paper.

3.2 Domain Extension for ind-cma MACs

A simple way to extend the domain of a pseudorandom function from n to $m > n$ bits is the "hash then encrypt" paradigm, where one first hashes the m bit input down to n bits using an ϵ-universal function, before applying the PRF. Unfortunately this simple trick does not work for (deterministic or probabilistic) MACs. Informally, the reason is that the output of a MAC does not "hide" its input, and thus an adversary can potentially learn the key of the hash function used (once she knows the key, she can find collisions for g which allows to break the MAC.) Below we show that, not surprisingly, for MACs where we explicitly require that they hide their input, as captured by the ind-cma notion, extending the domain using a universal hash function is safe.

Proposition 1 (Domain Extension for ind-cma **Secure MACs).** *Consider* $\mathsf{MAC} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ *with (small) message space* $\mathcal{M} = \{0,1\}^n$*, and let* $\mathsf{MAC}' = \{\mathsf{KG}', \mathsf{TAG}', \mathsf{VRFY}'\}$ *for large message space* $\{0,1\}^m$ *be derived from* MAC *by first hashing the message using an β-universal hash function $g : \{0,1\}^\ell \times \{0,1\}^m \to \{0,1\}^n$. (Using existing constructions we can set $\beta = 2^{-n+1}, \ell = 4(n + \log m)$, see the full version of the paper for details.) If* MAC *is*

$$(t, Q, \varepsilon) - \text{uf-cma} \quad \textit{secure and } (t, Q, \varepsilon) - \text{ind-cma} \quad \textit{secure}$$

then, for any $Q' \leq Q$, MAC' *is*

$$\textit{(1) } (t', Q', 2\varepsilon + Q'\beta) - \text{uf-cma} \quad \textit{secure and (2) } (t', Q', \varepsilon) - \text{ind-cma } \textit{secure}$$

where $t' \approx t$ can be derived from the proof.

The proof of Proposition 1 can be found in the full version of this paper.

3.3 From Selective to Full Security: suf-cma $\Rightarrow$ uf-cma

In this section we make the simple observation, that every *selectively* chosen-message secure MAC is also a chosen-message secure MAC, as we can simply guess the forgery. This guessing will loose a factor 2^μ in security if the domain is $\{0,1\}^\mu$.

Proposition 2 (From selective to full security). *Consider a MAC* $\mathsf{MAC} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ *with domain* $\{0,1\}^\mu$. *If* MAC *is* $(t, Q, \varepsilon) - \mathsf{suf\text{-}cma}$ *secure, then it is* $(t, Q, \varepsilon 2^\mu) - \mathsf{uf\text{-}cma}$ *secure.*

The proof of Proposition 2 can be found in the full version of this paper.

Remark 1 (Security Loss and Domain Extension). The security loss from the above transformation is 2^μ for MACs with message space $\{0,1\}^\mu$. In order to keep the security loss small, we are better off if we start with a MAC that has a small domain, or if we artificially restrict its domain to the first μ bits. Once we get a fully secure MAC on a small domain, we can always apply the domain-extension trick from Section 3.2 (using $\beta = 2^{-\mu+1}$) to expand this domain back up. Using both transformations together, we can turn any MAC that is (t, Q, ε)-$\mathsf{suf\text{-}cma}$ and $\mathsf{ind\text{-}cma}$ secure into a (t', Q', ε')-$\mathsf{uf\text{-}cma}$ and (t', Q', ε)-$\mathsf{ind\text{-}cma}$ secure MAC with the same-size (or arbitrarily larger) domain and where $t' \approx t$, and ε' depends on our arbitrary choice of μ as $\varepsilon' = \varepsilon 2^{\mu+1} + Q'/2^{\mu-1}$. In particular, if for some super-polynomial t, Q we assume a known corresponding negligible value ε such that the original MAC is (t, Q, ε)-$\mathsf{suf\text{-}cma}$, we can set $\mu = \log(1/\varepsilon)/2$ and the resulting MAC will be secure in the standard asymptotic sense - i.e. (t', Q', ϵ')-$\mathsf{uf\text{-}cma}$ for all polynomial $t', Q', 1/\epsilon'$.

4 Constructions of Authentication Protocols

In this section we provide a number of MACs from a variety of underlying primitives such as CCA-secure encryption, hash proof systems [15], homomorphic weak PRFs, and digital signatures. For concreteness, the constructions obtained from Diffie-Hellman type assumptions are summarized in Table 1; the constructions we obtain from the LPN assumption are summarized in Table 2. The constructions which are only $\mathsf{uf\text{-}cma}$ or $\mathsf{suf\text{-}cma}$ secure can be boosted to full cmva-security using the transformations from Section 3.

MAC construction	Secret Key k	Tag σ on m	Security	Assumption
$\mathsf{MAC}_{\mathsf{CS}}$ (§4.1)	$(\omega, x, x', y, k_2) \in \mathbb{Z}_p^4 \times \mathbb{G}$	$(U, U^\omega, U^{xH(U,V_1,m)+x'}, U^z \cdot k_2) \in \mathbb{G}^4$	uf-cmva	DDH
$\mathsf{MAC}_{\mathsf{HPS}}$ (§4.2)	$(\omega, x, x') \in \mathbb{Z}_p^3$	$(U, U^\omega, U^{xH(U,V_1,m)+x'}) \in \mathbb{G}^3$	uf-cmva	DDH
$\mathsf{MAC}_{\mathsf{hwPRF}}$ (§4.3)	$(x, x') \in \mathbb{Z}_p^2$	$(U, U^{xm+x'}) \in \mathbb{G}^2$	suf-cma	DDH
$\mathsf{MAC}_{\mathsf{WhwPRF}}$ (§4.3)	$(x, x'_0, \ldots, x'_\lambda) \in \mathbb{Z}_p^{\lambda+2}$	$(U, U^{x+\sum x'_i m_i}) \in \mathbb{G}^2$	uf-cma	DDH
$\mathsf{MAC}_{\mathsf{BB}}$ (§4.4)	$(x, x', y) \in \mathbb{Z}_p^3$	$(U, g^{xy} \cdot U^{xm+x'}) \in \mathbb{G}^2$	suf-cmva	gap-CDH
$\mathsf{MAC}_{\mathsf{TBB}}$ (§4.4)	$(x_1, x_2, x'_1, x'_2, y) \in \mathbb{Z}_p^5$	$(U, g^{x_1 y} U^{x_1 m + x'_1}, g^{x_2 y} U^{x_2 m + x'_2}) \in \mathbb{G}^3$	suf-cmva	CDH
$\mathsf{MAC}_{\mathsf{Waters}}$ (§4.4)	$(x, y, x'_1, \ldots, x'_\lambda) \in \mathbb{Z}_p^{\lambda+2}$	$(U, g^{xy} \cdot U^{x+\sum x'_i m_i}) \in \mathbb{G}^2$	uf-cmva	gap-CDH

Table 1. Overview of MAC constructions over prime-order groups. In all protocols, $\mathsf{TAG}_k(m)$ first generates $U \leftarrow_R \mathbb{G}$ and derives the rest of σ deterministically from U and k.

MAC construction	Key size	Tag size	Security	Assumption
$\mathsf{MAC}_{\mathsf{LPN}}$ (§4.5)	$\mathbb{Z}_2^{2\ell}$	$\mathbb{Z}_2^{(\ell+1)\times n}$	suf-cma & ind-cma	LPN
$\mathsf{MAC}_{\mathsf{BilinLPN}}$ (§4.5)	$\mathbb{Z}_2^{\ell\times\lambda}$	$\mathbb{Z}_2^{(\ell+1)\times n}$	uf-cma & ind-cma	LPN

Table 2. Overview of MAC constructions from the LPN problem from [29].

4.1 Constructions from CCA-secure Encryption

Let $\mathsf{E} = (\mathsf{KG}_{\mathsf{E}}, \mathsf{ENC}, \mathsf{DEC})$ be a (t, Q_E, Q_D, ϵ)-CCA secure labeled encryption scheme (see the full version of the paper for a formal definition.) Define $\mathsf{MAC} = (\mathsf{KG}_{\mathsf{MAC}}, \mathsf{TAG}, \mathsf{VRFY})$ as follows.

- Key Generation. $k = (k_1, k_2) \leftarrow \mathsf{KG}_{\mathsf{MAC}}(1^\lambda)$ samples $k_1 \leftarrow \mathsf{KG}_{\mathsf{E}}(1^\lambda)$ and $k_2 \leftarrow_R \{0,1\}^\lambda$.
- Tagging. $\mathsf{TAG}_{(k_1,k_2)}(m)$ samples $\sigma \leftarrow \mathsf{ENC}_{k_1}(k_2, m)$, i.e., it encrypts the plaintext k_2 using m as a label.
- Verification. $\mathsf{VRFY}_{(k_1,k_2)}(m, \sigma)$ output accept iff $\mathsf{DEC}_{k_1}(c, m) \stackrel{?}{=} k_2$.

Theorem 2. *Assume that* E *is a* (t, Q_E, Q_D, ϵ)*-CCA secure labeled encryption scheme. Then the construction* MAC *above is* $(t', Q_T, Q_V, \epsilon')$-uf-cmva *secure with* $t' \approx t$, $Q_T = Q_E$, $Q_V = Q_D$ *and* $\epsilon' = Q_T \cdot \epsilon + 2^{-\lambda}$.

The proof of Theorem 2 can be found in the full version of this paper.

Examples. There exists CCA-secure (public-key) encryption schemes from a variety of assumptions such as DDH [14, 31, 22], Paillier [15], lattices [37], and factoring [23]. In Table 1 we describe $\mathsf{MAC}_{\mathsf{CS}}$, which is $\mathsf{MAC}_{\mathsf{ENC}}$ instantiated with Cramer-Shoup encryption.

4.2 Constructions from Hash proof Systems

We now give a more direct construction of a MAC from any hash proof system. We recall the notion of (labeled) hash proof systems as introduced by Cramer and Shoup [15]. Let $\mathcal{C}, \mathcal{K}$ be sets and $\mathcal{V} \subset \mathcal{C}$ a language. In the context of public-key encryption (and viewing a hash proof system as a labeled key encapsulation mechanism (KEM) with "special algebraic properties") one may think of $\mathcal{C}$ as the set of all *ciphertexts*, $\mathcal{V} \subset \mathcal{C}$ as the set of all *valid (consistent) ciphertexts*, and $\mathcal{K}$ as the set of all *symmetric keys*. Let $\Lambda_k^\ell : \mathcal{C} \times \mathcal{L} \to \mathcal{K}$ be a labeled hash function indexed with $k \in \mathcal{SK}$ and label $\ell \in \mathcal{L}$, where $\mathcal{SK}$ and $\mathcal{L}$ are sets. A hash function Λ_k is *projective* if there exists a projection $\mu : \mathcal{SK} \to \mathcal{PK}$ such that $\mu(k) \in \mathcal{PK}$ defines the action of Λ_k^ℓ over the subset $\mathcal{V}$. That is, for every $C \in \mathcal{V}$, the value $K = \Lambda_k^\ell(C)$ is uniquely determined by $\mu(k)$, C. In contrast, nothing is guaranteed for $C \in \mathcal{C} \setminus \mathcal{V}$, and it may not be possible to compute $\Lambda_k(C)$ from $\mu(k)$ and C. A projective hash function is universal$_2$ if for all $C, C^* \in \mathcal{C} \setminus \mathcal{V}$, $\ell, \ell^* \in \mathcal{L}$ with $\ell \neq \ell^*$,

$$(pk, \Lambda_k^{\ell^*}(C^*), \Lambda_k^\ell(C)) = (pk, K, \Lambda_k^\ell(C)) \tag{1}$$

(as joint distributions) where in the above $pk = \mu(k)$ for $k \leftarrow_R \mathcal{SK}$ and $K \leftarrow_R \mathcal{K}$. It is extracting if for all $C \in \mathcal{C}$ (including valid ones) and $\ell \in \mathcal{L}$,

$$\Lambda_k^\ell(C) = K \tag{2}$$

where in the above $k \leftarrow_R \mathcal{SK}$ and $K \leftarrow_R \mathcal{K}$.

A labeled hash proof system $\mathsf{HPS} = (\mathsf{Param}, \mathsf{Pub}, \mathsf{Priv})$ consists of three algorithms. The randomized algorithm $\mathsf{Param}(1^k)$ generates parametrized instances of $params = (group, \mathcal{K}, \mathcal{C}, \mathcal{V}, \mathcal{PK}, \mathcal{SK}, \Lambda_{(\cdot)} : \mathcal{C} \to \mathcal{K}, \mu : \mathcal{SK} \to \mathcal{PK})$, where *group* may contain some additional structural parameters. The deterministic public evaluation algorithm Pub inputs the projection key $pk = \mu(k)$, $C \in \mathcal{V}$, a witness r of the fact that $C \in \mathcal{V}$, and a label $\ell \in \mathcal{L}$, and returns $K = \Lambda_k^\ell(C)$. The deterministic private evaluation algorithm Priv inputs $k \in \mathcal{SK}$ and returns $\Lambda_k^\ell(C)$, without knowing a witness. We further assume that μ is efficiently computable and that there are efficient algorithms given for sampling $k \in \mathcal{SK}$, sampling $C \in \mathcal{V}$ uniformly (or negligibly close to) together with a witness r, sampling $C \in \mathcal{C}$ uniformly, and for checking membership in $\mathcal{C}$.

As computational problem we require that the *subset membership problem* is (ϵ, t)-hard in HPS which means that for all adversaries B that run in time $\leq t$,

$$\left| \Pr[\mathsf{B}(\mathcal{C}, \mathcal{V}, C_1) = 1] - \Pr[\mathsf{B}(\mathcal{C}, \mathcal{V}, C_0) = 1] \right| \leq \epsilon$$

where $\mathcal{C}$ is taken from the output of $\mathsf{Param}(1^k)$, $C_1 \leftarrow_R \mathcal{C}$ and $C_0 \leftarrow_R \mathcal{C} \setminus \mathcal{V}$.

Construction. We define a MAC $\mathsf{MAC}_{\mathsf{HPS}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathcal{SK}$, message space $\mathcal{M} = \mathcal{L}$, and tag space $\mathcal{T} = \mathcal{C} \times \mathcal{K}$ as follows.

- Key Generation. The key-generation algorithm KG samples $k \leftarrow_R \mathcal{SK}$ and outputs k.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_k(m)$ picks $C \leftarrow_R \mathcal{V}$. It computes $K = \Lambda_k^m(C) \in \mathcal{K}$ and outputs $\sigma = (C, K)$.
- Verification. The verification algorithm $\mathsf{VRFY}_k(m, \sigma)$ parses $\sigma = (C, K)$ and outputs accept iff $K = \Lambda_k^m(C)$.

Note that the construction does not use the public evaluation algorithm Pub of HPS. Both tagging and verification only use the private evaluation algorithm Priv.

Theorem 3. *Let* HPS *be universal*$_2$ *and extracting. If the subset membership problem is* (t, ε)*-hard, then* $\mathsf{MAC}_{\mathsf{HPS}}$ *is* $(t', \varepsilon', Q_T, Q_V)$-uf-cmva *secure with* $\varepsilon' = Q_T\varepsilon + O(Q_T Q_V)/|\mathcal{K}|$ *and* $t' \approx t$.

The proof of Theorem 3 can be found in the full version of this paper.

Example. We recall a universal$_2$ HPS by Cramer and Shoup [15], whose hard subset membership problem is based on the DDH assumption. Let $\mathbb{G}$ be a group of prime-order p and let g_1, g_2 be two independent generators of $\mathbb{G}$. Define $\mathcal{L} = \mathbb{Z}_p$, $\mathcal{C} = \mathbb{G}^2$ and $\mathcal{V} = \{(g_1^r, g_2^r) \subset \mathbb{G}^2 \ : \ r \in \mathbb{Z}_p\}$. The value $r \in \mathbb{Z}_p$ is a witness of $C \in \mathcal{V}$. Let $\mathcal{SK} = \mathbb{Z}_p^4$, $\mathcal{PK} = \mathbb{G}^2$, and $\mathcal{K} = \mathbb{G}$. For $k = (x_1, x_2, y_1, y_2) \in \mathbb{Z}_p^4$,

define $\mu(k) = (g_1^{x_1} g_2^{x_2}, g_1^{y_1} g_2^{y_2})$. This defines the output of $\mathsf{Param}(1^k)$. For $C = (c_1, c_2) \in \mathcal{C}$ and $\ell \in \mathcal{L}$, define

$$\Lambda_k^\ell(C) := c_1^{x_1 \ell + y_1} c_2^{x_2 \ell + y_2} \ . \tag{3}$$

This defines $\mathsf{Priv}(k, C)$. Given $pk = \mu(k) = (X_1, X_2)$, $C \in \mathcal{V}$ and a witness $r \in \mathbb{Z}_p$ such that $C = (g_1^r, g_2^r)$ public evaluation $\mathsf{Pub}(pk, C, r)$ computes $K = \Lambda_k(C)$ as $K = (X_1^\ell X_2)^r$. Correctness follows by (3) and the definition of μ. This completes the description of HPS. Clearly, under the DDH assumption, the subset membership problem is hard in HPS. Moreover, this HPS is known to be universal$_2$ [15] and can be verified to be extracting.

Applying our construction from Theorem 3 we get the following MAC which we give in its equivalent (but more efficient) "explicit rejection" variant. Let $\mathbb{G}$ be a group of prime order p and g be a random generator of $\mathbb{G}$. Let $H : \mathbb{G}^2 \times \mathcal{M} \to \mathbb{Z}_p$ be a (target) collision resistant hash function. We define a message authentication code $\mathsf{MAC}_{\mathsf{HPS}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathbb{Z}_p^3$, message space $\mathcal{M}$, and tag space $\mathcal{T} = \mathbb{G}^3$ as follows.

- Key Generation. The key-generation algorithm KG outputs a secret key $k = (\omega, x, x') \leftarrow_R \mathbb{Z}_p^3$.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_k(m)$ samples $U \leftarrow_R \mathbb{G}$ and outputs an authentication tag $\sigma = (U, V_1, V_2) = (U, U^\omega, U^{x\ell + x'}) \in \mathbb{G}^3$, where $\ell = H(U, V_1, m)$.
- Verification. The verification algorithm $\mathsf{VRFY}_k(m, \sigma)$ parses $\sigma = (U, V_1, V_2) \in \mathbb{G}^3$ and outputs accept iff $A^\omega = V_1$ and $U^{x\ell + x'} = V_2$, where $\ell = H(U, V_1, m)$.

4.3 Construction from Key-Homomorphic Weak-PRFs

Definition 3. *Let $\mathcal{K} = \mathcal{K}(\lambda), \mathcal{X} = \mathcal{X}(\lambda), \mathcal{Y} = \mathcal{Y}(\lambda)$ and $\{f_k : \mathcal{X} \mapsto \mathcal{Y}\}_{k \in \mathcal{K}}$ be a weak PRF. We say that $\{f_k\}$ is* key-homomorphic *weak PRF if $\mathcal{K}, \mathcal{Y}$ are groups with an efficient group operation (written additively) of prime order $q = q(\lambda)$ and if for any fixed $x \in \mathcal{X}$ the function $f_k(x)$ is a group homomorphism of $\mathcal{K} \mapsto \mathcal{Y}$. In particular, for any $k_1, k_2 \in \mathcal{K}$ and $a, b \in \mathbb{Z}_q$, we have $f_{a \cdot k_1 + b \cdot k_2}(x) = a \cdot f_{k_1}(x) + b \cdot f_{k_2}(x)$.*

Construction. Let $\{f_k : \mathcal{X} \mapsto \mathcal{Y}\}_{k \in \mathcal{K}}$ be a key-homomorphic weak PRF where $\mathcal{K}, \mathcal{Y}$ are of prime order $q = q(\lambda)$. Define $\mathsf{MAC} = (\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY})$ with key-space $\mathcal{K} \times \mathcal{K}$ and message-space $\mathbb{Z}_q$ via:

- Key Generation. $\mathsf{KG}(1^\lambda)$ chooses $k_1, k_2 \leftarrow_R \mathcal{K}$ uniformly at random and outputs $k = (k_1, k_2)$.
- Tagging. $\mathsf{TAG}_{(k_1,k_2)}(m)$ chooses $x \leftarrow \mathcal{X}$ uniformly at random and sets $y = f_{m \cdot k_1 + k_2}(x)$. Output $\sigma = (x, y)$.
- Verification. $\mathsf{VRFY}_{(k_1,k_2)}(m, \sigma)$ parses $\sigma = (x, y)$ and outputs accept iff $f_{m \cdot k_1 + k_2}(x) \stackrel{?}{=} y$.

Theorem 4. *If $\{f_k\}$ is a (t, Q, ϵ)-weak PRF which is key-homomorphic over groups $\mathcal{K}, \mathcal{Y}$ of prime order $q = q(\lambda)$. Then the above construction is a (t', Q, ϵ')-*$\mathsf{suf\text{-}cma}$*-MAC (selective unforgeability, no verification queries) with $t' \approx t$ and $\epsilon' = \epsilon + 1/q$. It is also (t', Q, ϵ)-*$\mathsf{ind\text{-}cma}$.

The proof of Theorem 4 can be found in the full version of this paper.

DDH example. To instantiate the above MAC, we can take some DDH group $\mathbb{G}$ of prime order q. Let $\mathcal{K} = \mathbb{Z}_q$, $\mathcal{X} = \mathbb{G}$, $\mathcal{Y} = \mathbb{G}$ (which we now write multiplicatively) and define $f_k(x) = x^k$. This is a weak PRF by the DDH assumption. Furthermore, it is key-homomorphic with $f_{a \cdot k_1 + b \cdot k_2}(x) = (f_{k_1}(x))^a (f_{k_2}(x))^b$. Therefore, the above construction gives us the $\mathsf{suf\text{-}cma}$ MAC $\mathsf{MAC}_{\mathsf{hwPRF}}$ for messages $m \in \mathbb{Z}_q$, defined by $\mathsf{TAG}_{k_1,k_2}(m) := (g, h)$ with $g \leftarrow \mathbb{G}$ and $h := g^{k_1 \cdot m + k_2}$. See Table 1.

LWE example. To obtain another instantiation from the learning with erros problem, we use a recent construction of a weak PRF implicitly given in [3]. For integers $p < q$ and $x \in \mathbb{Z}_q$, define $\lceil x \rfloor_p = \lceil (p/q) \cdot x \rfloor \bmod p$. For a vector $\mathbf{x} \in \mathbb{Z}_q^m$ we extend this notion component wise to $\lceil \mathbf{x} \rfloor_p \in \mathbb{Z}_p^m$.

We let $\mathcal{K} = \mathbb{Z}_q^{m \times n}$, $\mathcal{X} = \mathbb{Z}_q^n$, $\mathcal{Y} = \mathbb{Z}_p^m$ (written additively) and define $f_{\mathbf{K}}(\mathbf{x}) = \lfloor \mathbf{K} \cdot \mathbf{x} \rceil_p$. This is a weak PRF under the Learning with Rounding (LWR) assumption of [3]. If p, q are integers such that q/p and the inverse LWE error rate $1/\alpha$ are super-polynomial in n, then the LWE_α assumption implies the LWR assumption [3]. Furthermore, it is key-homomorphic with $f_{a \cdot \mathbf{K}_1 + b \cdot \mathbf{K}_2}(x) = a f_{\mathbf{K}_1}(x) + b f_{\mathbf{K}_2}(x)$ for almost all inputs $\mathbf{x} \in \mathcal{X}$. (This is sufficient for our generic construction.) Therefore, the above construction gives us the $\mathsf{suf\text{-}cma}$ and $\mathsf{ind\text{-}cma}$ secure MAC for messages $m \in \mathbb{Z}_q$, defined by $\mathsf{TAG}_{\mathbf{K}_1,\mathbf{K}_2}(m) = (\mathbf{x}, \mathbf{y})$ with $\mathbf{x} \leftarrow \mathbb{Z}_q^n$ and $\mathbf{y} = \lfloor (m\mathbf{K}_1 + \mathbf{K}_2)\mathbf{x} \rceil_p$. (The message space can be extended to $\mathbb{Z}_q^n$ by encoding $\mathbf{m} \in \mathbb{Z}_q^n$ into a matrix $\mathbf{M} \in \mathbb{Z}_q^{n \times n}$ using a full-rank-difference encoding [1, 29].)

Full security. As an alternative to the transformation from Section 3.3, we sketch how to use Waters' argument [39] to obtain a (full) $\mathsf{uf\text{-}cma}$-secure MAC from a homomorphic weak PRF. Let $\{f_k : \mathcal{X} \mapsto \mathcal{Y}\}_{k \in \mathcal{K}}$ be a key-homomorphic weak PRF where $\mathcal{K}, \mathcal{Y}$ are of prime order $q = q(\lambda)$. Now define $\mathsf{MAC}_{\mathsf{WhwPRF}} = (\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY})$ with key-space $\mathcal{K}^{\lambda+1}$ and message-space $\{0,1\}^\lambda$ via:

- $\mathsf{KG}(1^\lambda)$: Choose $k_0 \ldots k_\lambda \leftarrow_R \mathcal{K}$ at random, output $k = (k_0, \ldots, k_\lambda)$.
- $\mathsf{TAG}_k(m)$: Choose $x \leftarrow_R \mathcal{X}$ uniformly at random and set $y = f_{k_0 + \sum k_i m_i}(x)$. Output $\sigma = (x, y)$.
- $\mathsf{VRFY}_k(m, \sigma)$: Parse $\sigma = (x, y)$ and outpt accept iff $f_{k_0 + \sum k_i m_i}(x) \stackrel{?}{=} y$.

The resulting $\mathsf{MAC}_{\mathsf{WhwPRF}}$ can be proved to be $\mathsf{uf\text{-}cma}$ and $\mathsf{ind\text{-}cma}$-secure. A DDH-based example instantiation is contained in Table 1.

4.4 Constructions from Signatures

Clearly, an $\mathsf{uf\text{-}cma}$-secure digital signature scheme directly implies an $\mathsf{uf\text{-}cmva}$-secure MAC. In certain cases we can obtain improved efficiency, as we demonstrate with a MAC derived from Boneh-Boyen signatures [11]. Concretely, we

can instantiate the MAC in any prime-order groups, no bilinear maps are needed. We define a message authentication code $\mathsf{MAC}_{\mathsf{BB}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathbb{G} \times \mathbb{Z}_p^2$, message space $\mathcal{M} = \mathbb{Z}_p$, and tag space $\mathcal{T} = \mathbb{G}^2$ as follows.

- Key Generation. The key-generation algorithm KG outputs a secret key $k = (x, x', y) \leftarrow_R \mathbb{Z}_p^3$.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_k(m)$ samples $U \leftarrow_R \mathbb{G}$ and outputs an authentication tag $\sigma = (U, g^{xy} \cdot U^{xm+x'}) \in \mathbb{G}^2$.
- Verification. The verification algorithm $\mathsf{VRFY}_k(m, \sigma)$ parses $\sigma = (U, V) \in \mathbb{G}^2$ and outputs accept iff $g^{xy} \cdot U^{xm+x'} = V$.

Theorem 5. *If the gap-CDH assumption is $(t, Q_T + Q_V, \varepsilon)$-hard, then $\mathsf{MAC}_{\mathsf{BB}}$ is $(t', \varepsilon', Q_T, Q_V)$ suf-cmva secure with $\varepsilon' = \varepsilon$ and $t' \approx t$.*

The proof of Theorem 5 can be found in the full version of this paper. The above construction is only secure under the gap-CDH assumption. We now show how to apply the twinning technique [13] to obtain a MAC secure under the standard CDH assumption. We define a message authentication code $\mathsf{MAC}_{\mathsf{TBB}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathbb{Z}_p^5$, message space $\mathcal{M} = \mathbb{Z}_p$, and tag space $\mathcal{T} = \mathbb{G}^3$ as follows.

- Key Generation. The key-generation algorithm KG outputs a secret key $k = (x_1, x_1', x_2, x_2', y) \leftarrow_R \mathbb{Z}_p^5$.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_k(m)$ picks $U \leftarrow_R \mathbb{G}$ and outputs an authentication tag $\sigma = (U, V_1 = g^{x_1 y} U^{x_1 m + x_1'}, V_2 = g^{x_2 y} U^{x_2 m + x_2'}) \in \mathbb{G}^3$.
- Verification. The verification algorithm $\mathsf{VRFY}_k(m, \sigma)$ parses $\sigma = (U, V_1, V_2)$ and outputs accept iff $g^{x_1 y} U^{x_1 m + x_1'} = V_1$ and $g^{x_2 y} U^{x_2 m + x_2'} = V_2$.

Theorem 6. *If the CDH problem is (t, ε)-hard, then MAC is $(t', \varepsilon', Q_T, Q_V)$ suf-cmva secure with $\varepsilon' = \varepsilon + O((Q_T + Q_V)/p)$ and $t' \approx t$.*

The proof of Theorem 6 can be found in the full version of this paper. We remark that $\mathsf{MAC}_{\mathsf{BB}}$ and $\mathsf{MAC}_{\mathsf{TBB}}$ are only selectively secure (suf-cmva) MACs. Even though this is sufficient for obtaining man-in-the-middle secure authentication protocols, to obtain a fully secure MAC $\mathsf{MAC}_{\mathsf{Waters}}$, one can update the constructions using Waters' hash function [39]. The drawback is that the secret key then contains λ many elements in $\mathbb{Z}_p$ and that the security reduction is not tight anymore. We remark that it is also possible to build slightly more efficient suf-cmva-secure MACs from the (Gap) q-Diffie-Hellman inversion problems.

4.5 Constructions from the LPN assumption

In this section we review the suf-cma and uf-cma-secure MACs constructions implicitly given in [29, Section 4]. To both constructions can apply the transformations from Section 3 to obtain efficient uf-cmva-secure MACs.

First construction (suf-cma). Let n denote the number of repetitions, τ the parameter of the Bernoulli distribution, and $\tau' := 1/4 + \tau/2$ controls the correctness error.

We define a message authentication code $\mathsf{MAC}_{\mathsf{LPN}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathbb{Z}_2^{2\ell}$, message space $\mathcal{M} = \{\mathbf{m} \in \mathbb{Z}_2^{2\ell} : \mathrm{hw}(\mathbf{m}) = \ell\}$, and tag space $\mathcal{T} = \mathbb{Z}_2^{(\ell+1)n}$ as follows.

- Key Generation. The key-generation algorithm KG outputs a secret key a vector $\mathbf{x} \leftarrow_R \mathbb{Z}_2^{2\ell}$.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_{\mathbf{x}}(\mathbf{m})$ samples $\mathbf{R} \leftarrow_R \mathbb{Z}_2^{\ell \times n}$ and outputs an authentication tag $\sigma = (\mathbf{R}, \mathbf{R}^T \cdot \mathbf{x}_{\downarrow \mathbf{m}} + \mathbf{e})$, where $\mathbf{e} \in \mathbb{Z}_2^n$ is sampled according the Bernoulli distribution with parameter τ and $\mathbf{x}_{\downarrow \mathbf{m}} \in \mathbb{Z}_2^{\ell}$ is the vector obtained from $\mathbf{x}$ by deleting all entries where $\mathbf{m}_i = 0$.
- Verification. The verification algorithm $\mathsf{VRFY}_{\mathbf{x}}(\mathbf{m}, \sigma)$ parses $\sigma = (\mathbf{R}, \mathbf{z}) \in \mathbb{Z}_2^{\ell \times n} \times \mathbb{Z}_2^n$ and outputs accept iff $|\mathbf{R}^T \cdot \mathbf{x}_{\downarrow \mathbf{m}} - \mathbf{z}| \leq \tau' n$.

Concretely, [29, Th. 4] shows (implicitly)[9] that $\mathsf{MAC}_{\mathsf{LPN}}$ has $2^{-O(n)}$ completeness error and is suf-cma and ind-cma-secure under the $\mathsf{LPN}_{\ell,\tau}$ assumption in dimension $\approx \ell$ and Bernoulli parameter τ.

Second construction (uf-cma). We define a message authentication code $\mathsf{MAC}_{\mathsf{BilinLPN}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ with associated key space $\mathcal{K} = \mathbb{Z}_2^{\ell \times \lambda}$, message space $\mathcal{M} = \mathbb{Z}_2^{\lambda}$, and tag space $\mathcal{T} = \mathbb{Z}_2^{(\ell+1)n}$ as follows.

- Key Generation. The key-generation algorithm KG outputs a secret key a matrix $\mathbf{X} \leftarrow_R \mathbb{Z}_2^{\ell \times \mu}$.
- Tagging. The probabilistic authentication algorithm $\mathsf{TAG}_{\mathbf{X}}(\mathbf{m})$ samples $\mathbf{R} \leftarrow_R \mathbb{Z}_2^{\ell \times n}$ and outputs an authentication tag $\sigma = (\mathbf{R}, \mathbf{R}^T \cdot \mathbf{X} \cdot \mathbf{m} + \mathbf{e})$, where $\mathbf{e} \in \mathbb{Z}_2^n$ is sampled according the Bernoulli distribution with parameter τ.
- Verification. The verification algorithm $\mathsf{VRFY}_{\mathbf{X}}(\mathbf{m}, \sigma)$ parses $\sigma = (\mathbf{R}, \mathbf{z}) \in \mathbb{Z}_2^{\ell \times n} \times \mathbb{Z}_2^n$ and outputs accept iff $|\mathbf{R}^T \cdot \mathbf{X} \cdot \mathbf{m} - \mathbf{z}| \leq \tau' n$.

[29, Th. 5] shows that $\mathsf{MAC}_{\mathsf{BilinLPN}}$ is uf-cma and ind-cma-secure under the $\mathsf{LPN}_{\ell,\tau}$ assumption. We remark that $\mathsf{MAC}_{\mathsf{BilinLPN}}$ can also be viewed as an instantiation of $\mathsf{MAC}_{\mathsf{WhwPRF}}$ of Section 4.3 when generalizing the construction to *randomized* weak PRFs and using $f_{\mathbf{x}}(\mathbf{R}) = \mathbf{R}^T\mathbf{x} + \mathbf{e}$ which is a randomized weak PRF under LPN.

4.6 Three-Round Authentication from Any Weak PRF

We now state our authentication protocol Π using any wPRF family $\mathcal{F} = \{f_{k_1} : \mathcal{X}_1 \mapsto \mathcal{Y}\}_{k_1 \in \mathcal{K}_1}$ and any weak Almost XOR-Universal (wAXU) family $\mathcal{H} = \{h_{k_2} : \mathcal{X}_2 \mapsto \mathcal{Y}\}_{k_2 \in \mathcal{K}_2}$ (see the full version of the paper for more details on how $\mathcal{H}$ can be instantiated.)

[9] [29] give a direct construction of a MAC that is suf-cmva secure. $\mathsf{MAC}_{\mathsf{LPN}}$ is the underlying MAC that can be proved only suf-cma secure.

The key generation algorithm $\mathsf{KG}(1^\lambda)$ selects random $k_1 \leftarrow \mathcal{K}_1, k_2 \leftarrow \mathcal{K}_2$ and outputs $k = (k_1, k_2)$. Following this, the three round protocol between a Tag $\mathcal{T}(k)$ and a reader $\mathcal{R}(k)$ is defined below:

- $\mathcal{T} \to \mathcal{R}$: choose random $r \in \mathcal{X}_1$ and send r to $\mathcal{R}$.
- $\mathcal{R} \to \mathcal{T}$: choose random $x \in \mathcal{X}_2$ and send x to $\mathcal{T}$.
- $\mathcal{T} \to \mathcal{R}$: compute $z = f_{k_1}(r) + h_{k_2}(x)$ and send z to $\mathcal{R}$.
- $\mathcal{R}$: accept if and only if $z \stackrel{?}{=} f_{k_1}(r) + h_{k_2}(x)$.

Theorem 7. *Assuming that $\mathcal{F} = \{f_{k_1}\}$ is a (t, Q, ε)-wPRF and $\mathcal{H} = \{h_{k_2}\}$ is (t, ρ)-wAXU. Then the above authentication protocol is (t', Q, ε')-secure against active adversaries, with $t' = t/2$ and $\varepsilon' = \sqrt{\varepsilon + \rho}$.*

In particular, setting $\mathcal{F} = \mathcal{H}$ and $\mathcal{X}_1 = \mathcal{X}_2 = \mathcal{X}$, we get $\varepsilon' = \sqrt{2\varepsilon + \frac{1}{|\mathcal{X}|} + \frac{1}{|\mathcal{Y}|}}$.

The proof of Theroem 7 can be found in the full version of this paper.

Example. To instantiate the above authentication protocol, we can take some DDH group $\mathbb{G}$ of prime order q. Let $\mathcal{K} = \mathcal{K}_1 = \mathcal{K}_2 = \mathbb{Z}_q$, $\mathcal{X} = \mathcal{X}_1 = \mathcal{X}_2 = \mathbb{G}$, $\mathcal{Y} = \mathbb{G}$ (which we now write multiplicatively). For notational convenience, let us denote $k_1 = a$, $k_2 = b$, $r = g$, and define $f_a(g) := g^a$, $h_b(x) := x^b$ so that $\mathcal{F}$ is a wPRF by DDH, and $\mathcal{H} = \mathcal{F}$ is wAXU by DDH as well. We get the following very simple DDH-based protocol with secret key $k = (a, b)$.

- $\mathcal{T} \to \mathcal{R}$: choose random $g \in \mathbb{G}$ and send g to $\mathcal{R}$.
- $\mathcal{R} \to \mathcal{T}$: choose random $x \in \mathbb{G}$ and send x to $\mathcal{T}$.
- $\mathcal{T} \to \mathcal{R}$: compute $z = g^a x^b \in G$ and send z to $\mathcal{R}$.
- $\mathcal{R}$: accept if and only if $z \stackrel{?}{=} g^a x^b$.

It is interesting to compare the above actively secure authentication protocol with Okamoto's public-key authentication protocol based on the discrete log assumption [36]. On the one hand, Okamoto's scheme is based on a weaker assumption and works in the public-key setting. On the other hand, our DDH-based protocol is more efficient. Our verifier only has to perform two exponentiations, while Okamoto's verifier needs to do three exponentiations. Also, our last flow z contains one group element, while Okamoto's protocol contains two exponents, which is likely going to be longer.

References

1. Shweta Agrawal, Dan Boneh, and Xavier Boyen. Efficient lattice (H)IBE in the standard model. In Henri Gilbert, editor, *Advances in Cryptology – EUROCRYPT 2010*, volume 6110 of *Lecture Notes in Computer Science*, pages 553–572. Springer, May 2010.
2. Benny Applebaum, David Cash, Chris Peikert, and Amit Sahai. Fast cryptographic primitives and circular-secure encryption based on hard learning problems. In Shai Halevi, editor, *Advances in Cryptology – CRYPTO 2009*, volume 5677 of *Lecture Notes in Computer Science*, pages 595–618. Springer, August 2009.
3. Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. Cryptology ePrint Archive, Report 2011/401, 2011. http://eprint.iacr.org/.

4. Mihir Bellare. New proofs for NMAC and HMAC: Security without collision-resistance. In Cynthia Dwork, editor, *Advances in Cryptology – CRYPTO 2006*, volume 4117 of *Lecture Notes in Computer Science*, pages 602–619. Springer, August 2006.
5. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Keying hash functions for message authentication. In Neal Koblitz, editor, *Advances in Cryptology – CRYPTO'96*, volume 1109 of *Lecture Notes in Computer Science*, pages 1–15. Springer, August 1996.
6. Mihir Bellare, Ran Canetti, and Hugo Krawczyk. Pseudorandom functions revisited: The cascade construction and its concrete security. In *37th Annual Symposium on Foundations of Computer Science*, pages 514–523. IEEE Computer Society Press, October 1996.
7. Mihir Bellare, Oded Goldreich, and Anton Mityagin. The power of verification queries in message authentication and authenticated encryption. Cryptology ePrint Archive, Report 2004/309, 2004. http://eprint.iacr.org/.
8. Mihir Bellare, Krzysztof Pietrzak, and Phillip Rogaway. Improved security analyses for CBC MACs. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 527–545. Springer, August 2005.
9. Mihir Bellare and Phillip Rogaway. The exact security of digital signatures: How to sign with RSA and Rabin. In Ueli M. Maurer, editor, *Advances in Cryptology – EUROCRYPT'96*, volume 1070 of *Lecture Notes in Computer Science*, pages 399–416. Springer, May 1996.
10. Dan Boneh and Xavier Boyen. Efficient selective-ID secure identity based encryption without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 223–238. Springer, May 2004.
11. Dan Boneh and Xavier Boyen. Short signatures without random oracles. In Christian Cachin and Jan Camenisch, editors, *Advances in Cryptology – EUROCRYPT 2004*, volume 3027 of *Lecture Notes in Computer Science*, pages 56–73. Springer, May 2004.
12. Jan Camenisch, Susan Hohenberger, and Anna Lysyanskaya. Compact e-cash. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 302–321. Springer, May 2005.
13. David Cash, Eike Kiltz, and Victor Shoup. The twin Diffie-Hellman problem and applications. In Nigel P. Smart, editor, *Advances in Cryptology – EUROCRYPT 2008*, volume 4965 of *Lecture Notes in Computer Science*, pages 127–145. Springer, April 2008.
14. Ronald Cramer and Victor Shoup. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack. In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 13–25. Springer, August 1998.
15. Ronald Cramer and Victor Shoup. Universal hash proofs and a paradigm for adaptive chosen ciphertext secure public-key encryption. In Lars R. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Computer Science*, pages 45–64. Springer, April / May 2002.
16. Yevgeniy Dodis and Krzysztof Pietrzak. Improving the security of MACs via randomized message preprocessing. In Alex Biryukov, editor, *Fast Software Encryption – FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 414–433. Springer, March 2007.

17. Yevgeniy Dodis and Aleksandr Yampolskiy. A verifiable random function with short proofs and keys. In Serge Vaudenay, editor, *PKC 2005: 8th International Workshop on Theory and Practice in Public Key Cryptography*, volume 3386 of *Lecture Notes in Computer Science*, pages 416–431. Springer, January 2005.
18. Amos Fiat and Adi Shamir. How to prove yourself: Practical solutions to identification and signature problems. In Andrew M. Odlyzko, editor, *Advances in Cryptology – CRYPTO'86*, volume 263 of *Lecture Notes in Computer Science*, pages 186–194. Springer, August 1987.
19. Henri Gilbert, Matthew J. B. Robshaw, and Yannick Seurin. How to encrypt with the LPN problem. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *ICALP 2008: 35th International Colloquium on Automata, Languages and Programming, Part II*, volume 5126 of *Lecture Notes in Computer Science*, pages 679–690. Springer, July 2008.
20. Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986.
21. Kristiyan Haralambiev, Tibor Jager, Eike Kiltz, and Victor Shoup. Simple and efficient public-key encryption from computational Diffie-Hellman in the standard model. In Phong Q. Nguyen and David Pointcheval, editors, *PKC 2010: 13th International Conference on Theory and Practice of Public Key Cryptography*, volume 6056 of *Lecture Notes in Computer Science*, pages 1–18. Springer, May 2010.
22. Dennis Hofheinz and Eike Kiltz. Secure hybrid encryption from weakened key encapsulation. In Alfred Menezes, editor, *Advances in Cryptology – CRYPTO 2007*, volume 4622 of *Lecture Notes in Computer Science*, pages 553–571. Springer, August 2007.
23. Dennis Hofheinz and Eike Kiltz. Practical chosen ciphertext secure encryption from factoring. In Antoine Joux, editor, *Advances in Cryptology – EUROCRYPT 2009*, volume 5479 of *Lecture Notes in Computer Science*, pages 313–332. Springer, April 2009.
24. Nicholas J. Hopper and Manuel Blum. Secure human identification protocols. In Colin Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Computer Science*, pages 52–66. Springer, December 2001.
25. Éliane Jaulmes, Antoine Joux, and Frédéric Valette. On the security of randomized CBC-MAC beyond the birthday paradox limit: A new construction. In Joan Daemen and Vincent Rijmen, editors, *Fast Software Encryption – FSE 2002*, volume 2365 of *Lecture Notes in Computer Science*, pages 237–251. Springer, February 2002.
26. Ari Juels and Stephen A. Weis. Authenticating pervasive devices with human protocols. In Victor Shoup, editor, *Advances in Cryptology – CRYPTO 2005*, volume 3621 of *Lecture Notes in Computer Science*, pages 293–308. Springer, August 2005.
27. Jonathan Katz and Ji Sun Shin. Parallel and concurrent security of the HB and HB+ protocols. In Serge Vaudenay, editor, *Advances in Cryptology – EUROCRYPT 2006*, volume 4004 of *Lecture Notes in Computer Science*, pages 73–87. Springer, May / June 2006.
28. 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.
29. Eike Kiltz, Krzysztof Pietrzak, David Cash, Abhishek Jain, and Daniele Venturi. Efficient authentication from hard learning problems. In Kenneth G. Paterson, editor, *Advances in Cryptology – EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 7–26. Springer, May 2011.

30. Hugo Krawczyk. New hash functions for message authentication. In Louis C. Guillou and Jean-Jacques Quisquater, editors, *Advances in Cryptology – EUROCRYPT'95*, volume 921 of *Lecture Notes in Computer Science*, pages 301–310. Springer, May 1995.
31. Kaoru Kurosawa and Yvo Desmedt. A new paradigm of hybrid encryption scheme. In Matthew Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, volume 3152 of *Lecture Notes in Computer Science*, pages 426–442. Springer, August 2004.
32. Silvio Micali, Michael O. Rabin, and Salil P. Vadhan. Verifiable random functions. In *40th Annual Symposium on Foundations of Computer Science*, pages 120–130. IEEE Computer Society Press, October 1999.
33. Moni Naor and Omer Reingold. Number-theoretic constructions of efficient pseudo-random functions. In *38th Annual Symposium on Foundations of Computer Science*, pages 458–467. IEEE Computer Society Press, October 1997.
34. Moni Naor and Omer Reingold. From unpredictability to indistinguishability: A simple construction of pseudo-random functions from MACs (extended abstract). In Hugo Krawczyk, editor, *Advances in Cryptology – CRYPTO'98*, volume 1462 of *Lecture Notes in Computer Science*, pages 267–282. Springer, August 1998.
35. Moni Naor, Omer Reingold, and Alon Rosen. Pseudo-random functions and factoring (extended abstract). In *32nd Annual ACM Symposium on Theory of Computing*, pages 11–20. ACM Press, May 2000.
36. Tatsuaki Okamoto. Provably secure and practical identification schemes and corresponding signature schemes. In Ernest F. Brickell, editor, *Advances in Cryptology – CRYPTO'92*, volume 740 of *Lecture Notes in Computer Science*, pages 31–53. Springer, August 1993.
37. Chris Peikert. Public-key cryptosystems from the worst-case shortest vector problem: extended abstract. In Michael Mitzenmacher, editor, *41st Annual ACM Symposium on Theory of Computing*, pages 333–342. ACM Press, May / June 2009.
38. Jacques Stern. A new identification scheme based on syndrome decoding. In Douglas R. Stinson, editor, *Advances in Cryptology – CRYPTO'93*, volume 773 of *Lecture Notes in Computer Science*, pages 13–21. Springer, August 1994.
39. Brent R. Waters. Efficient identity-based encryption without random oracles. In Ronald Cramer, editor, *Advances in Cryptology – EUROCRYPT 2005*, volume 3494 of *Lecture Notes in Computer Science*, pages 114–127. Springer, May 2005.
40. Mark N. Wegman and Larry Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22:265–279, 1981.