Message Authentication, Revisited

Yevgeniy Dodis* Eike Kiltz† Krzysztof Pietrzak‡ Daniel Wichs§

October 28, 2012

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 example, we obtain elegant DDH-based MACs with much shorter keys than the quadratic-sized key of the Naor-Reingold PRF. We also show that several natural (probabilistic) digital signature schemes, such as those by Boneh-Boyen and Waters, can be significantly optimized when "downgraded" into a MAC, both in terms of their efficiency (e.g., no bilinear pairings) and security assumptions (e.g., standard CDH instead of bilinear CDH).
- 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). In fact, a number of elegant constructions, such as recently constructed MACs based on Learning Parity with Noise (LPN) and some of the new MACs constructed in this work, are uf-cma but not not uf-cmva secure by themselves. We give an *efficient* generic transformation from any uf-cma secure MAC which is "message-hiding" into a uf-cmva secure MAC. Applied to LPN-based MACs, this resolves the main open problem of Kiltz et al. from Eurocrypt'11.
- 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.

*New York University. Partially supported by NSF Grants CNS-1065134, CNS-1065288, CNS-1017471, CNS-0831299 and Google Faculty Award. E-mail: dodis@cs.nyu.edu

†Faculty of Mathematics, Horst Görtz Institute for IT Security, Ruhr-Universität Bochum. Supported by Sofja Kovalevskaja Award of the Alexander von Humboldt Foundation, funded by the German Federal Ministry for Education and Research. E-mail: eike.kiltz@ruhr-uni-bochum.de

‡IST Austria. E-mail: pietrzak@ist.ac.at Supported by the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Starting Grant (259668-PSPC)

§IBM T. J. Watson Research Center. E-mail: wichs@cs.nyu.edu

1 Introduction

Message Authentication Codes (MACs) are one of the most fundamental primitives in cryptography. Historically, a vast majority of MAC constructions are based on pseudorandom functions (PRFs).[1] 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 [5, 8] or HMAC [4, 3]), or using elegant, but slower number-theoretic constructions, such as the Naor-Reingold (NR) PRF [38]. 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 [38]), or a number of exponentiations linear in the security parameter (e.g., tree-type PRFs based on the GGM transform [23] applied to some discrete-log-type pseudorandom generator), or are based on exotic and relatively untested assumptions (e.g., Dodis-Yampolskiy PRF [20] 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 [39]. Alternatively, even when relying on decisional assumptions (e.g. DDH), MAC constructions are allowed to be *probabilistic*.[2] In contrast, building a PRF (and, more generally, deterministic MACs) 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.[3] 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 [37, 36, 11] and, yet, not necessary for most applications of digital signatures. In particular, prominent digital signature schemes in the standard model[4] [17, 11, 44] are all probabilistic. In fact, such signature schemes trivially give MACs. Of course, such MACs are not necessarily as efficient

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

[2] Of course, probabilistic MACs can always be converted to deterministic one using a PRF by replacing the random coins with the output of the PRF (where the key for the PRF is part of the MAC key, and the input to the PRF is the message to be authenticated), but recall that we are trying to design MACs while avoiding PRFs!

[3] 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 [45], and several other follow-up works (e.g., [34, 28, 19]). However, most prior probabilistic MACs were still explicitly based on a PRF or a block cipher.

[4] In fact, even in the random oracle model there are noticeable advantages. E.g., full domain hash (FDH) signatures [9, 14] have worse exact security than *probabilistic* FDH signatures [15], while Fiat-Shamir signatures [21] are inherently probabilistic.

as they could be, since they "unnecessarily" support public verification.[5] 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* [27, 29, 31, 32] (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 [22, 1, 43, 27, 29, 31, 32]. 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 [2], 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 [32].

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 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 [6], 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 [32], 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. [32], who construct uf-cmva MACs from the learning parity with noise (LPN) problem. We remark that [32] 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

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

of uf-cma and suf-cma LPN based MACs from [32] 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 [3] 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 [32].

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 [7]. 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 [18] or many of the subsequent schemes (e.g. [35, 25, 26, 42, 24]). 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 [18, 35, 25], 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, 26, 24]. 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 [40] has a quadratic-size secret key, while our construction based on the Hofheinz-Kiltz [26] 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) [18]. Hash Proof Systems were originally defined [18] 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 [38] 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 [44], 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 [44] 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 5.2 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 [38], 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 [30], which in turn consists of a parallel repetition of the HB^+ protocol of Juels and Weiss [29]. 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 [30, 29]. In fact, we could relax the notion of wPRF slightly to allow for probabilistic computation with approximate correctness, so that the protocol of [30] 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 (uf-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 2.1 ((Selective) unforgeability under chosen message (& verification) attack.) *A* MAC *is* $(t, Q_T, Q_V, \varepsilon)$-uf-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)-uf-cma *secure if it is* $(t, Q_T, 1, \varepsilon)$-uf-cmva*-secure. That is,* uf-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* suf-cma *and* suf-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 [3]. His notion is for deterministic MACs, whereas our notion can only be achieved for probabilistic MACs.

Definition 2.2 (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 $\mathsf{ind\text{-}cma}$ 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 $\mathsf{TAG}_k(0)$.

2.3 Diffie-Hellman Assumptions

We now recall a number of standard Diffie-Hellman assumptions. Let $(F_\lambda)_{\lambda\in\mathbb{N}}$ be a family of groups, where each $\lambda \in \mathbb{N}$ describes a group $\mathbb{G}$ of prime-order p and a generator g of $\mathbb{G}$. We say that the Computational Diffie-Diffie Hellman (CDH) problem is (ϵ, t)-hard if for all adversaries A that run in time t,

$$\Pr_{x,y\leftarrow_R\mathbb{Z}_p}[\mathsf{A}(1^\lambda, g^x, g^y) = g^{xy}] \leq \varepsilon.$$

We say that the Gap Computational Diffie-Diffie Hellman (Gap-CDH) problem is (ϵ, Q, t)-hard if for all adversaries A that run in time t, and make at most Q queries to their oracle, we have

$$\Pr_{x,y\leftarrow_R\mathbb{Z}_p}[\mathsf{A}^{\mathcal{O}_x(\cdot,\cdot)}(1^\lambda, g^x, g^y) = g^{xy}] \leq \varepsilon,$$

where $\mathcal{O}_x(\cdot,\cdot)$ is a DDH oracle with fixed basis g^x, i.e., $\mathcal{O}_x(Y, Z) = 1$ iff $Y^x = Z$. We say Gap-CDH is just (ϵ, t)-hard if it is $(\epsilon, Q = t, t)$-hard. We say that the Decisonal Diffie-Diffie Hellman (DDH) problem is (ϵ, t)-hard if for all adversaries A that run in time t,

$$\left|\Pr_{x,y\leftarrow_R\mathbb{Z}_p}[\mathsf{A}(1^\lambda, g^x, g^y, g^{xy}) = 1] - \Pr_{x,y,z\leftarrow_R\mathbb{Z}_p}[\mathsf{A}(1^\lambda, g^x, g^y, g^z) = 1]\right| \leq \varepsilon.$$

3 Transformations for MACs

In this section we give some general transformations for MACs as discussed in the introduction. We start with a few definitions.

Definition 3.1 (almost universal and pairwise independent hash function) *A keyed function $h : \{0,1\}^\ell \times \{0,1\}^m \to \{0,1\}^n$ is* δ-almost universal *if any two distinct inputs collide with probability at most δ over the choice of the random key, i.e., for all $x \neq x' \in \{0,1\}^m$*

$$\Pr_{k\leftarrow_R\{0,1\}^\ell}[h_k(x) = h_k(x')] \leq \delta.$$

Moreover, h as above is pairwise independent *if it behaves like a uniformly random function on any two inputs, i.e., for all $x \neq x' \in \{0,1\}^m$ and $y, y' \in \{0,1\}^n$*

$$\Pr_{k\leftarrow_R\{0,1\}^\ell}[h_k(x) = y \wedge h_k(x') = y'] = 2^{-2n}.$$

A pairwise independent h as above is 2^{-n}-universal. The reason to consider the weak universality notion is that it's already sufficient for the domain extension shown in Section 3.2, while requiring much shorter keys, especially when the input domain is very large (the key for h is linear in the output, not the input, see the Proposition below.) For our transformation in Section 3.1 we need pairwise independence, but there the domain is quite small (namely the tag length of the underlying MAC.)

Proposition 3.2 *There exists a 2^{-n+1}-universal hash function as above with key length $\ell = 4(n + \log m)$. Furthermore, there exits a pairwise independent hash function with key length $\ell = 2\max\{m, n\}$.*

We next state a simple lemma which we'll use in the proof of Theorem 3.4 below. For some $\mu, Q \in \mathbb{N}$, consider the following game. We first pick a pairwise independent hash function $h : \mathcal{T} \to \{0,1\}^\mu$. Now an adversary A can adaptively choose values $z_1, \ldots, z_Q \in \mathcal{T}$. After choosing z_i we pick a uniformly random $b_i \in \{0,1\}^\mu$ and give her $x_i := b_i \oplus h(z_i)$. Finally, the adversary outputs a pair (z, x) which must be different from all (z_i, x_i). She wins if $b = b_i$ for some i where $b := x \oplus h(z)$.

Lemma 3.3 *The advantage of any (even computationally unbounded) adversary in winning the above game is* $\leq Q/2^\mu$.

Proof. For the proof it is convenient to think of a different sampling procedure (but which gives the same distribution) for the x_i, z_i where h is chosen at the very end. When we get z_i, we answer with a uniformly random x_i. At the end we get (z, x) from the adversary, and only know we sample h which defines the b_i's as $b_i = h(z_i) \oplus x_i$. From the adversaries perspective, this is exactly the same game as before, in both cases the x_i are uniformly random and independent of the z_i and h. The adversary wins if for any i, $b_i \oplus h(z_i) = b \oplus h(z)$. As h is pairwise independent and independent of the z_i's (the h is sampled uniformly, and the z_i's are fixed before h is sampled), we have $\Pr[b_i \oplus h(z_i) = b \oplus h(z)] = 2^{-\mu}$ for any i, taking the union bound over all $i = 1, \ldots, Q$ we get an upper bound of $Q/2^\mu$ as stated in the Lemma. ∎

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

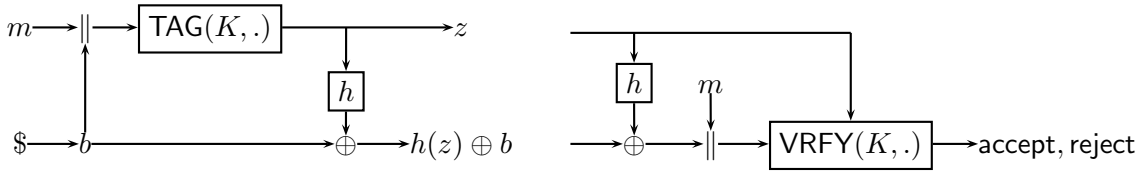

Figure 1: $\overline{\mathsf{TAG}}$ and $\overline{\mathsf{VRFY}}$ with key (k, h), message m and randomness b.

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.
- 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 3.4 (uf-cma + ind-cma ⇒ 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*

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

Proof. Let $\overline{\mathsf{A}}$ be a (t, Q_T, Q_V) adversary who breaks the uf-cmva security of $\overline{\mathsf{MAC}}$ with advantage δ. From this adversary we will construct an adversary A who either

(*) breaks the $(t', Q_T, \delta/2 - 2Q_TQ_V/2^\mu)$-ind-cma security of MAC, or

(**) breaks the $(t', Q_T, \delta/2Q_V)$-uf-cma security of MAC.
This implies the Theorem as follows: assume that $\delta > \varepsilon' = 2Q_V\varepsilon + 2Q_VQ_T/2^\mu$. If (**) holds, then A breaks the uf-cma security of MAC with advantage $\delta/2Q_V > \epsilon$. If (*) holds, A breaks the ind-cma security with advantage $\delta/2 - Q_TQ_V/2^\mu > \frac{2Q_V\varepsilon+2Q_VQ_T/2^\mu}{2} - Q_TQ_V/2^\mu = Q_V\varepsilon \geq \varepsilon$.

Consider the experiment $\mathbf{Exp}_0 = \mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\overline{\mathsf{A}}, \lambda, Q_T, Q_V)$. Below we define two events α_C, α_{FF} for $\mathbf{Exp}_0$ ($\alpha = 1$ denotes that the event occurred). With $\delta_C = \Pr_{\mathbf{Exp}_0}[\alpha_C = 1]$ and $\delta_{FF} = \Pr_{\mathbf{Exp}_0}[\alpha_{FF} = 1]$ we denote the probability of the respective events. Let $Q = Q_T + Q_V$ be the total number of queries, and let $b_1, \ldots, b_Q$ be the randomness associated with this queries (for a $\overline{\mathsf{TAG}}(m)$ query, b is the randomness sampled for this query. For a $\overline{\mathsf{VRFY}}((z, y), m)$ query, we have $b = h(z) \oplus y$.) The events are defined as

α_C: The event α_C (C for **C**ollision) occurs if $\overline{\mathsf{A}}$ finds a forgery $(m, (z, y))$, and this forgery uses randomness b that has already been used. (I.e., if the tth query is the first accepting $\overline{\mathsf{VRFY}}$ query, then $b_t = b_j$ for some $j < t$.)

α_{FF}: The event α_{FF} (FF for **F**resh **F**orgery) occurs if $\overline{\mathsf{A}}$ finds a forgery $(m, (z, y))$, and this forgery is "fresh" in the sense that the b contained in this forgery is different from the b's in all the queries so far.

As the forgery must occur either before or after a collision

$$\delta_C + \delta_{FF} = \delta \tag{1}$$

thus either $\delta_C \geq \delta/2$ or $\delta_{FF} \geq \delta/2$, by the two claims below, the first case implies (*) and the second implies (**).

Claim 3.5 MAC *is not* $(t, Q_T, \delta_C - Q_TQ_V/2^\mu)$-ind-cma *secure.*

Proof. We define an adversary A (who will use $\overline{\mathsf{A}}$ as a black-box) and consider an ind-cma attack on MAC, i.e., $\mathsf{A}^{\mathcal{O}}$ has access to an oracle $\mathcal{O}$ and must tell if it implements $\mathsf{TAG}_k(\cdot)$ or $\mathsf{TAG}_k(0)$ (cf. Definition 2.2).

$\mathsf{A}^{\mathcal{O}}$ first samples h and then invokes $\overline{\mathsf{A}}$. $\mathsf{A}^{\mathcal{O}}$ answers tag queries m by $\overline{\mathsf{A}}$ with $(z, h(z) \oplus b)$ where b is random and $z = \mathcal{O}(m\|b)$. It answers all verification queries with reject. Finally $\mathsf{A}^{\mathcal{O}}$ outputs 1 if any of the Q b's in the experiment collide, and 0 otherwise.

If the oracle implements $\mathcal{O}(\cdot) = \mathsf{TAG}_k(\cdot)$, then $\mathsf{A}^{\mathcal{O}}$ simulated the $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\overline{\mathsf{A}}, \lambda, Q_T, Q_V)$ experiment, except that all verification queries were rejected. This difference does not affect the probability of the event α_C, as at the point where an adversary gets accept as response to a verification query, the outcome of the event α_C is determined. Moreover, if $\alpha_C = 1$, then $\mathsf{A}^{\mathcal{O}}$ outputs 1, so

$$\Pr[\mathsf{A}^{\mathcal{O}} \to 1 \ ; \ \mathcal{O}(\cdot) = \mathsf{TAG}_k(\cdot)] \geq \delta_C. \tag{2}$$

On the other hand, if $\mathcal{O}(\cdot) = \mathsf{TAG}_k(0)$ then

$$\Pr[\mathsf{A}^{\mathcal{O}} \to 1 \ ; \ \mathcal{O}(\cdot) = \mathsf{TAG}_k(0)] \leq Q_TQ_V/2^\mu. \tag{3}$$

To see this, first note that the values that $\overline{\mathsf{A}}$ gets as answer to a tag query m is $(z, h(z) \oplus b)$ where $z = \mathsf{TAG}_k(0)$. In particular, z is independent of b (as the $\mathsf{TAG}_k(0)$ oracle ignores its input $m\|b$.)

Assume that A' makes only one forgery attempt, i.e., $Q_V = 1$. Then the above attack is a special case of the game considered in Lemma 3.3 above, and $\mathsf{A}^{\mathcal{O}}$ outputs 1 iff it wins the game, which by Lemma 3.3 happens with probability $Q_T/2^\mu$. For the case $Q_V > 1$ get an upper bound $Q_VQ_T/2^\mu$ by a simple reduction which uses the fact that $\mathsf{A}^{\mathcal{O}}$ answers all verification queries with reject: assume

$\mathsf{A}^{\mathcal{O}}$ outputs 1 when interacting with A' with probability $> Q_V Q_T/2^\mu$. Let A'' be like A', but where A'' suppresses all but a randomly chosen one of the Q_V verification queries (think of A'' as running A' but only forwarding one of the verification queries, answering the other with reject.) If $\mathsf{A}^{\mathcal{O}}$ interacts with A'', it will output 1 with at least $1/Q_V$ the probability as when interacting with A', that is, $> Q_T/2^\mu$, which again contradicts Lemma 3.3.

Summing up, by eq.(2) and (3) $\mathsf{A}^{\mathcal{O}}$ has advantage $\delta_C - Q_T Q_V/2^\mu$ in the ind-cma experiment. ■

Claim 3.6 MAC *is not* $(t, Q_T, \delta_{FF}/Q_V)$-uf-cma *secure.*

Proof. We construct an adversary A who breaks the uf-cma security of MAC with advantage δ_{FF}/Q_V as follows. $\mathsf{A}^{\mathcal{O}}$ samples a random hash function h and a random index $j \in \{1, \ldots, Q_V\}$. It then invokes $\overline{\mathsf{A}}$, answering all verification queries with reject, and tag queries m with $(z, h(z) \oplus b)$ where b is random and $z = \mathcal{O}(m\|b)$. When $\overline{\mathsf{A}}$ makes the jth verification query $(m, (z, y))$ and $b := y \oplus h(z)$ if fresh, $\mathsf{A}^{\mathcal{O}}$ outputs $(m\|b, z)$ as its forgery attempt. If in this uf-cmva attack the event α_{FF} holds, and the first accepting verification query is the jth one, then this is a valid and fresh forgery for MAC with key k. This event has probability at δ_{FF}/Q_V. ■

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 3.7 (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* $g : \{0,1\}^\ell \times \{0,1\}^m \to \{0,1\}^n$*. (By Proposition 3.2 we can set* $\beta = 2^{-n+1}$ *using only a short key* $\ell = 4(n + \log m)$*.) If* MAC *is*

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

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

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

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

Proof. We begin with point (2), proving $(t', Q, 2\varepsilon)$-ind-cma security of MAC'. Let A' be a (t', Q') adversary against the ind-cma security of MAC'. Firstly, by ind-cma security of MAC, the attacker A' has advantage at most ε in distinguishing oracle access to $\mathsf{TAG}'_{(k,g)}(\cdot) = \mathsf{TAG}_k(g(\cdot))$ from oracle access to $\mathsf{TAG}_k(0)$. Secondly, by another application of the ind-cma security of MAC, the attacker A' has advantage at most ε in distinguishing oracle access to $\mathsf{TAG}_k(0)$ from $\mathsf{TAG}'_{k,g}(0) = \mathsf{TAG}_k(g(0))$ (which returns random tags of the message $g(0)$). Therefore the overall advantage of A' in distinguishing $\mathsf{TAG}'_{(k,g)}(\cdot)$ and $\mathsf{TAG}'_{(k,g)}(0)$ is at most 2ε as we wanted to show.

We will now prove point (1). Consider a (t', Q) uf-cma adversary A' against MAC'. In an uf-cma attack the adversary $\mathsf{A}'^{\mathsf{TAG}'_{(k,g)}(\cdot)}$ makes Q queries $m_1, \ldots, m_Q$ and one forgery attempt (m, σ) (wlog.

we assume m is different from all m_i.) Let δ_C be the probability that the hash of the forgery message m collides with the hash of some m_i, i.e., $\delta_C = \Pr[\exists m_i : \ g(m_i) = g(m)]$, we claim that

Claim 3.8 $\delta_C - Q\beta \leq \varepsilon$.

Proof. [of Claim] To prove the claim, we show that every uf-cma adversary A' as above who after Q queries to $\mathsf{TAG}'_{(k,g)}(\cdot) = \mathsf{TAG}_k(g(\cdot))$ outputs a colliding forgery attempt with probability δ_C (note that we don't care here if this attempt is actually successful or not) can be used to break the ind-cma security of MAC with probability $\delta_C - Q\beta$, by assumption this probability must be $\leq \varepsilon$.

Now, assume for contradiction A' breaks the uf-cma security of MAC' with advantage $\beta > 2\varepsilon + Q\beta$. By the above Claim at least $\beta - \delta_C > \varepsilon$ of the successful forgeries (m, σ) of A' are "fresh", i.e., $g(m) \neq g(m_i)$ for all Q tag-queries m_i made so far. Thus we also get a forgery $(g(m), \sigma)$ for the underlying MAC MAC with advantage $> \varepsilon$, contradicting its (t, Q, ε) ind-cma security. ■

3.3 From Selective to Full Security: suf-cma ⇒ uf-cmva

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

Proof. Assume for contradiction that there exists an adversary A' who breaks the uf-cma security of MAC with advantage $\delta > \varepsilon 2^\mu$. From this adversary we get an adversary A of basically the same size, who breaks the suf-cma security of MAC with advantage $> \varepsilon$ (contradicting the assumed suf-cma security) as follows. $\mathsf{A}^{\mathsf{TAG}_K(\cdot)}$ in the suf-cma experiment first samples a random $m \leftarrow \{0,1\}^\mu$, then $\mathsf{A}^{\mathsf{TAG}_K(\cdot)}$ invokes A', forwarding its queries to the oracle $\mathsf{TAG}_K(\cdot)$. Finally, A' outputs a forgery attempt (m', σ'), if $m = m'$, $\mathsf{A}^{\mathsf{TAG}_K(\cdot)}$ outputs this as its forgery (otherwise A gives up.) A will output a valid forgery if A' does (which happens with probability δ), and moreover the initial guess m on m' was correct (which happens with probability $2^{-\mu}$). So A has advantage $\delta/2^\mu > \varepsilon$. ■

Remark 3.10 (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 artificaily 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)-suf-cma *and* ind-cma *secure into a* (t', Q', ε')-uf-cma *and* (t', Q', ε)-ind-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)-suf-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')-uf-cma *for all polynomial* $t', Q', 1/\epsilon'$.

4 Constructions of MACs

In this section we provide a number of MACs from a variety of underlying primitives such as CCA-secure encryption, hash proof systems [18], homomorphic weak PRFs, and digital signatures. For concreteness, the constructions obtained from Diffie-Hellman type assumptions are summarized in

MAC construction	Secret Key k	Tag σ on m	Security	Assumption
$\mathsf{MAC}_{\mathsf{CS}}$ (§4.1)	$(\omega, x, x', y, k_2) \in \mathbb{Z}_p^4 \times \{0,1\}^\lambda$	$(U, U^\omega, U^{xH(U,V_1,m)+x'}, U^z \cdot k_2) \in \mathbb{G}^4$	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 [32].

Table 1; the constructions we obtain from the LPN assumption are summarized in Table 2. The constructions which are only uf-cma or suf-cma secure can be boosted to full cmva-security using the transformations from Section 3.

4.1 Constructions from CCA-secure Encryption

Definition. A *symmetric-key labeled encryption scheme* $\mathsf{E} = (\mathsf{KG}, \mathsf{ENC}, \mathsf{DEC})$ is a triple of algorithms:

- $k \leftarrow \mathsf{KG}(\lambda)$: generates a key k.
- $c \leftarrow \mathsf{ENC}_k(m, L)$: generates an encryption under key k and label L of the message m.
- $m = \mathsf{DEC}_k(c, L)$: decrypts the ciphertext c under key k and label L.

The *chosen-ciphertext attack (CCA)* security of the scheme E is defined via the following experiment:

Experiment $\mathbf{Exp}^{\mathsf{cca}}_{\mathsf{E}}(\mathsf{A}, \lambda, Q_E, Q_D, b)$
Sample $k \leftarrow \mathsf{KG}(1^\lambda)$.
Learning Stage 1: Run $(L, m_0, m_1, aux) \leftarrow \mathsf{A}^{\mathsf{ENC}_k(\cdot,\cdot),\mathsf{DEC}_k(\cdot,\cdot)}(1^\lambda)$.
Challenge: Sample $c \leftarrow \mathsf{ENC}_k(m_b, L)$.
Learning Stage 2: Run $\tilde{b} \leftarrow \mathsf{A}^{\mathsf{ENC}_k(\cdot,\cdot),\mathsf{DEC}_k(\cdot,\cdot)}(1^\lambda, c, aux)$.
Output: If A queries $\mathsf{DEC}_k(c, L)$ in Stage 2, or makes more than Q_E and Q_D queries in total to ENC and DEC respectively, output $\perp$. Else output $\tilde{b}$.

Definition 4.1 *A* symmetric-key labeled encryption scheme E *is* $(t, Q_E, Q_D, \varepsilon)$*-CCA secure if for all* A *running in time* $t = t(\lambda)$ *we have*

$$|\Pr[\mathbf{Exp}^{\mathsf{cca}}_{\mathsf{E}}(\mathsf{A}, \lambda, Q_E, Q_D, 1) = 1] - \Pr[\mathbf{Exp}^{\mathsf{cca}}_{\mathsf{E}}(\mathsf{A}, \lambda, Q_E, Q_D, 0) = 1]| \leq \varepsilon(\lambda).$$

Our Construction. Let $\mathsf{E} = (\mathsf{KG}_\mathsf{E}, \mathsf{ENC}, \mathsf{DEC})$ be a $(t, Q_E, Q_D, \varepsilon)$-CCA secure labeled encryption scheme 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 4.2 *Assume that* E *is a* $(t, Q_E, Q_D, \varepsilon)$*-CCA secure labeled encryption scheme. Then the construction* MAC *above is* $(t', Q_T, Q_V, \varepsilon')$-uf-cmva *secure with* $t' \approx t$, $Q_T = Q_E$, $Q_V = Q_D$ *and* $\varepsilon' = Q_T \cdot \varepsilon + 2^{-\lambda}$.

Proof. Assume there is some A running in time t' and making Q_T, Q_V queries to $\mathsf{TAG}, \mathsf{VRFY}$ such that $\Pr[\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] \geq \varepsilon'$.

Let us define the experiments $\mathbf{Exp}_i$ similarly to $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}$ with the modification that:

1. The first i queries to $\mathsf{TAG}_{(k_1,k_2)}(m)$ are answered by sampling $c \leftarrow \mathsf{ENC}_{k_1}(0, m)$ with the plaintext 0 instead of k_2. The rest of the queries are answered as before.

2. All queries to $\mathsf{VRFY}_{(k_1,k_2)}(m, \sigma)$ where $\sigma = c$ was the output of a previous TAG query with input m are answered as accept.

Note that $\mathbf{Exp}_0$ is equivalent to $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}$ by the *correctness* of the encryption scheme E, which ensures that modification (2) above does not change any outputs.

We now claim that for all $i \in [Q_T]$, we have

$$|\Pr[\mathbf{Exp}_{i-1}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_i(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq \varepsilon. \tag{4}$$

Otherwise, we can construct an attacker $\mathsf{B}^{\mathsf{ENC}_{k_1}(\cdot,\cdot),\mathsf{DEC}_{k_1}(\cdot,\cdot)}$ with running time $t \approx t'$, and making $Q_E = Q_T, Q_D = Q_V$ encryption/decryption queries and advantage $> \varepsilon$ in the CCA game. The attacker B runs A. It selects $k_2 \leftarrow \{0,1\}^\lambda$ on its own at random, and answers queries for A as follows:

TAG **queries:** For the first $i-1$ queries with message m, call the encryption oracle with $(0, m)$ and output the resulting ciphertext. For the ith query with message m, give the values $(m_0 = 0, m_1 = k_2, L = m)$ to the challenger and get back the challenge ciphertext c^* which is given as the reply to the query. For the remaining queries with messages m, call the encryption oracle with (k_2, m) and output the resulting ciphertext.

VRFY **queries:** On any query (m, σ) if m was previously queried to TAG and the output was σ then output accept. Else use the decryption oracle with $(c = \sigma, L = m)$ and output accept iff the reply is k_2.

If A ever makes a verification query with a fresh m that was never queries to TAG and the reply is accept then output 1, else output 0. It's easy to see that

$$\Pr[\mathbf{Exp}_{i-1}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] = \Pr[\mathbf{Exp}^{\mathsf{cca}}_{\mathsf{E}}(\mathsf{B}, \lambda, Q_T, Q_V, 0) = 1],$$
$$\Pr[\mathbf{Exp}_{i}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] = \Pr[\mathbf{Exp}^{\mathsf{cca}}_{\mathsf{E}}(\mathsf{B}, \lambda, Q_T, Q_V, 1) = 1].$$

which proves equation 4 must hold.

By the hybrid argument, we therefore have $|\Pr[\mathbf{Exp}_0(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_{Q_T}(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq Q_T \cdot \varepsilon$. Moreover, in $\mathbf{Exp}_{Q_T}$ the attacked does not get *any information* about k_2 but only wins

if it produces an encryption of k_2. Therefore $\Pr[\mathbf{Exp}_{Q_T}(\mathsf{A},\lambda,Q_T,Q_V)=1] \le 2^{-\lambda}$. Putting this all together, we obtain

$$\varepsilon' \le \Pr[\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A},\lambda,Q_T,Q_V)=1] \le \Pr[\mathbf{Exp}_{Q_T}(\mathsf{A},\lambda,Q_T,Q_V)=1] + Q_T\cdot\varepsilon \le Q_T\cdot\varepsilon + 2^{-\lambda}.$$

This concludes the proof. ∎

Examples. There exists CCA-secure (public-key) encryption schemes from a variety of assumptions such as DDH [16, 35, 25], Paillier [18], lattices [42], and factoring [26]. In Table 1 we describe $\mathsf{MAC}_{\mathsf{CS}}$, which is $\mathsf{MAC}_{\mathsf{ENC}}$ instantiated with Cramer-Shoup encryption [16].

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 [18]. 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 $\Lambda^{\ell}_k : \mathcal{C}\times\mathcal{L}\to\mathcal{K}$ 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^{ℓ}_k over the subset $\mathcal{V}$. That is, for every $C\in\mathcal{V}$, the value $K=\Lambda^{\ell}_k(C)$ 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 $(C,\ell)\neq(C^*,\ell^*)$, we have:

$$(pk,\Lambda^{\ell^*}_k(C^*),\Lambda^{\ell}_k(C)) = (pk,K,\Lambda^{\ell}_k(C)) \tag{5}$$

(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 $\ell\in\mathcal{L}$,

$$\Lambda^{\ell}_k(C) = K \tag{6}$$

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 $K=\Lambda^{\ell}_k(C)$. The deterministic private evaluation algorithm Priv inputs $k\in\mathcal{SK}$ and returns $\Lambda^{\ell}_k(C)$, 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 $\le t$,

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

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 message authentication code $\mathsf{MAC}_{\mathsf{HPS}} = \{\mathsf{KG}, \mathsf{TAG}, \mathsf{VRFY}\}$ 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 4.3 *Let* HPS *be universal*$_2$ *and extracting. If the subset membership problem is* (t, ε)*-hard, then* $\mathsf{MAC}_{\mathsf{HPS}}$ *is* $(t', \varepsilon', Q_T, Q_V)$*-*$\mathsf{uf\text{-}cmva}$ *secure with* $\varepsilon' = Q_T\varepsilon + O(Q_T Q_V)/|\mathcal{K}|$ *and* $t' \approx t$.

Proof. Assume there is some A running in time t' and making Q_T, Q_V queries to $\mathsf{TAG}, \mathsf{VRFY}$ such that $\Pr[\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] \geq \epsilon'$.

We now define a number of experiments where we change the distribution of the TAG and VRFY oracles. We remark that whenever we change the TAG oracle we also implicitly adapt the VRFY oracle such this is outputs accept on m, σ, whenever σ was the output of a previous TAG query with input m.

Let us define the experiments $\mathbf{Exp}_{i,j}$ similarly to $\mathbf{Exp}^{\mathsf{uf\text{-}cmva}}_{\mathsf{MAC}}$ with the modification that the first $i-1$ queries to the $\mathsf{TAG}_k(m)$ oracle are answered with random $\sigma \leftarrow_R \mathcal{C} \times \mathcal{K}$ and all queries from the $i+1$th query on are answered real (i.e., $C \leftarrow_R \mathcal{V}, K = \Lambda_k^m(C)$), as in $\mathbf{Exp}$. The ith query to the $\mathsf{TAG}_k(m^*)$ oracle and the queries to the $\mathsf{VRFY}_k(m, \sigma)$ oracles are defined as follows.

	ith $\mathsf{TAG}_k(m^*)$	$\mathsf{VRFY}_k(m, \sigma = (C, K))$
$\mathbf{Exp}_{i,0}$	real	real
$\mathbf{Exp}_{i,1}$	$C^* \leftarrow_R \mathcal{C}, K^* = \Lambda_k^{m^*}(C^*)$	real
$\mathbf{Exp}_{i,2}$	$C^* \leftarrow_R \mathcal{C}, K^* = \Lambda_k^{m^*}(C^*)$	reject if $C \in \mathcal{C} \setminus \mathcal{V}$
$\mathbf{Exp}_{i,3}$	$(C^*, K^*) \leftarrow_R \mathcal{C} \times \mathcal{K}$	reject if $C \in \mathcal{C} \setminus \mathcal{V}$
$\mathbf{Exp}_{i,4}$	$(C^*, K^*) \leftarrow_R \mathcal{C} \times \mathcal{K}$	real

Note that $\mathbf{Exp} = \mathbf{Exp}_{1,0}$ and $\mathbf{Exp}_{i,4} = \mathbf{Exp}_{i+1,0}$.

We claim that for all $i \in [Q_T]$, we have

$$|\Pr[\mathbf{Exp}_{i,0}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_{i,1}(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq \epsilon. \tag{7}$$

Otherwise, we can construct an attacker B with running time $t \approx t'$ and advantage $> \epsilon$ in the subset memebership experiment. Note that B can simulate the whole experiment by generating k itself.

We claim that for all $i \in [Q_T]$, we have

$$|\Pr[\mathbf{Exp}_{i,1}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_{i,2}(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq Q_V/|\mathcal{K}|. \tag{8}$$

$$|\Pr[\mathbf{Exp}_{i,3}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_{i,4}(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq Q_V/|\mathcal{K}|. \tag{9}$$

To prove (8), we consider the information A learns about $k \in \mathcal{SK}$. From all consistent tags $\sigma = (C, K)$, A can at most learn the projected key $pk = \mu(k)$. This follows by the existance of the publication evaluation algorithm Pub of HPS. Furthermore, A learns one single inconsistent tag $\sigma^* = (C^*, K^*)$ from the ith query to $\mathsf{TAG}_k(m^*)$ where $C^* \leftarrow_R \mathcal{C} \setminus \mathcal{V}$ and $K^* = \Lambda_k^{m^*}(C^*)$. Now consider an inconsistent $\mathsf{VRFY}_k(m, \sigma = (C, K))$ query with $(C^*, K^*, m^*) \neq (C, K, m)$. By the universal$_2$ property, given the view of A, the probability that A hits the right K that would make

VRFY accept in $\mathbf{Exp}_{i,2}$ is bounded by $1/|\mathcal{K}|$, since K is uniform in $\mathcal{K}$). Now (8) follows by a hybrid argument. The proof of (9) is analogous.

Using the same argument we can duduce from the universal$_2$ property (5) that for all $i \in [Q_T]$, we have

$$|\Pr[\mathbf{Exp}_{i,2}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}_{i,3}(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq 1/|\mathcal{K}|. \tag{10}$$

We define $\mathbf{Exp}^*$ as $\mathbf{Exp}_{Q_T+1,0}$, with the difference that all VRFY queries are answered with reject. We now claim that

$$|\Pr[\mathbf{Exp}_{Q_T+1,0}(\mathsf{A}, \lambda, Q_T, Q_V) = 1] - \Pr[\mathbf{Exp}^*(\mathsf{A}, \lambda, Q_T, Q_V) = 1]| \leq Q_V/|\mathcal{K}|. \tag{11}$$

Until A makes the first VRFY query, no information about k is leaked through to the adversary. Hence, by definition of completeness (6), the probability that a VRFY query accepts the first VRFY query bounded by $1/|\mathcal{K}|$. The claim follows by a hybrid argument over Q_V. ∎

Example. We recall a universal$_2$ HPS by Cramer and Shoup [18], 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

$$\Lambda_k^\ell(C) := c_1^{x_1\ell+y_1} c_2^{x_2\ell+y_2} \ . \tag{12}$$

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 (12) 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$_2$ [18] and can be verified to be extracting.

Applying our construction from Theorem 4.3 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 4.4 *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 of prime order $q = q(\lambda)$ with an efficient group operation written additively, and if for any fixed $x \in \mathcal{X}$ the function $f_k(x)$ is a group isomorphism 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 $f_{a\cdot k_1+b\cdot k_2}(x) = a \cdot f_{k_1}(x) + b \cdot f_{k_2}(x)$, and for $k_1 \neq k_2$ we have $f_{k_1}(x) \neq f_{k_2}(x)$.*

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 4.5 *Assuming that $\{f_k\}$ 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')-suf-cma-MAC (selective unforgeability, no verification queries) with $t' \approx t$ and $\varepsilon' = \varepsilon + 1/q$. It is also (t', Q, ε)-ind-cma.*

Proof. Assume that A runs in time t' and breaks suf-cma security with probability ε'. First, let us consider an experiment $\mathbf{Exp}_0$ which is defined the same way as the suf-cma experiment $\mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathsf{MAC}}$ experiment *except* that, after A specifies the message m^* on which he will forge, all tagging queries during the game with query message m are answered by computing $x, y = f_{(m-m^*)k_1+k_2}(x)$ instead of computing y as specified. Also the final VRFY query on message m^* and tag $\sigma = (x, y)$ is verified by checking $y \stackrel{?}{=} f_{k_2}(x)$ instead of as specified. It is easy to see tat $\mathbf{Exp}_0$ is actually equivalent to $\mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathsf{MAC}}$ just by thinking of the latter game as using $k_2' = (m - m^*)k_1 + k_2$ in place of k_2. Since k_2' and k_2 are both uniformly random and independent of k_1 the experiments are equivalent.

Now, define $\mathbf{Exp}_1$ where the tagging queries for messages $m \neq m^*$ are just answered with values $(x, y) \leftarrow \mathcal{X} \times \mathcal{Y}$ chosen uniformly at random. The final verification query on m^* is still checked by $y \stackrel{?}{=} f_{k_2}(x)$. We claim that $|\Pr[\mathbf{Exp}_1(\mathsf{A}, \lambda, Q) = 1] - \Pr[\mathbf{Exp}_0(\mathsf{A}, \lambda, Q) = 1]| \leq \varepsilon$. To see this we construct a reduction B which gets Q challenge values $(x_1, y_1), \ldots, (x_Q, y_Q)$. It then chooses $k_2 \leftarrow \mathcal{K}$ at random and runs A who initially specifies the forged message m^*. The reduction B answers tagging queries using the challenge values (x_i, y_i) in a clever way: if the ith tagging query is m_i, it answers with (x_i, y_i') where $y_i' = (m_i - m^*) \cdot y_i + f_{k_2}(x_i)$. Lastly, it outputs 1 iff A makes a verification query $m^*, (x^*, y^*)$ with $f_{k_2}(x^*) = y^*$. Note that if the challenge values a pseudo-random with $y_i = f_{k_1}(x_i)$, then this perfectly simulates $\mathbf{Exp}_0$. Else, if x_i, y_i are truly random, this perfectly simulates $\mathbf{Exp}_1$. This proves the claimed indistinguishability of $\mathbf{Exp}_1$ and $\mathbf{Exp}_0$.

Lastly, we claim that $\Pr[\mathbf{Exp}_1(\mathsf{A}, \lambda, Q) = 1] \leq 1/q$. This follows since the attacker does not know anything about k_2 during the course of $\mathbf{Exp}_1$ and therefore the probability of outputting any value (x, y) such that $f_{k_2}(x) = y$ is at most $1/q$. (Recall that for fixed x, the function $f_{k_2}(x)$ is a group homomorphism and so $f_{k_2}(x)$ is uniformly random over the choice of k_2.) Therefore $\varepsilon' = \Pr[\mathbf{Exp}^{\mathsf{suf\text{-}cma}}_{\mathsf{MAC}}(\mathsf{A}, \lambda, Q_T)] \leq 1/q + \varepsilon$ as we wanted to show.

To show that ind-cma security follows from weak PRF security, we build a reduction that chooses $k_1 \leftarrow \mathcal{K}$ and gets Q tuples $(x_1, y_1), \ldots, (x_Q, y_Q)$. It then answers the ith tagging query m_i with $m_i \cdot f_{k_1}(x_i) + y_i$. If the tuples (x_i, y_i) are pseudorandom with $y_i = f_{k_2}(x_i)$ then this corresponds to the correct tagging procedure. Else, if the tuples are truly random, this corresponds to answering tagging queries randomly. Therefore an attacker with advantage ε in the ind-cma game can be used to win the weak PRF game with probability ε. ∎

DDH example. To instantiate the above MAC, we can take some DDH group $\mathbb{G}$ of prime order q. Let $\mathcal{K} = \mathbb{Z}_q$, $\mathcal{X} = \mathbb{G} \setminus \{1\}$, $\mathcal{Y} = \mathbb{G}$ (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 suf-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.

Full security. As an alternative to the transformation from Section 3.3, we now sketch how to use Waters' argument [44] to obtain a (full) uf-cma-secure MAC from any 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 $q = q(\lambda)$. 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 $k_0, \ldots, k_\lambda \leftarrow_R \mathcal{K}$ uniformly 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.

Verification Queries. Since our $\mathsf{suf\text{-}cma}$/$\mathsf{uf\text{-}cma}$-secure MACs are also $\mathsf{ind\text{-}cma}$-secure, we can apply the transformation on Theorem 3.4 to make them $\mathsf{suf\text{-}cmva}$/$\mathsf{uf\text{-}cmva}$-secure at a very small extra cost.

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 group, 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 4.6 *If the gap-CDH assumption is $(t, Q_T + Q_V, \varepsilon)$-hard, then $\mathsf{MAC}_{\mathsf{BB}}$ is $(t', \varepsilon', Q_T, Q_V)$ $\mathsf{suf\text{-}cmva}$ secure with $\varepsilon' = \varepsilon$ and $t' \approx t$.*

Proof. Assume there exists an adversary A that $(t', \varepsilon', Q_T, Q_V)$-breaks the $\mathsf{suf\text{-}cmva}$ security of $\mathsf{MAC}_{\mathsf{BB}}$. We build an adversary B that $(t, Q_T + Q_V, \varepsilon)$-breaks the gap-CDH assumption. Given an instance of the gap-CDH problem g^x, g^y, and the target message m^* obtained by A in the $\mathsf{suf\text{-}cmva}$ experiment, B picks a random $a \in \mathbb{Z}_p$ and implicitly defines the secret key $k = (x, x', y)$, where

$$x' = -m^* x + a.$$

Note that with this definitions of the secret key k, a properly distributed MAC on message m is of the form

$$\sigma = (U, V) = (g^r, g^{xy + r(x(m-m^*)+a)}), \tag{13}$$

for $r \leftarrow_R \mathbb{Z}_p$. Adversary A is executed and its $\mathsf{TAG}_k(\cdot)$ and $\mathsf{VRFY}_k(\cdot)$ queries are answered by B as follows.

- $\mathsf{TAG}_k(m)$ for $m \neq m^*$. B picks random $r' \in \mathbb{Z}_p$ and implicitly defines $r = -y/(m - m^*) + r' \bmod p$. A MAC tag is computed as $\sigma = ((g^y)^{1/(m-m^*)} g^{r'}, (g^x)^{r'(m-m^*)} g^{r'})$, which has the same distribution as (13).

- $\mathsf{VRFY}_k(m, \sigma)$ for $m \neq m^*$. Assume $\sigma = (U, V)$ is of the form $(U, V) = (g^r, g^{xy + r(x(m-m^*)+a)+z})$ which we rewrite as $(g^{r'+y/(m-m^*)}, g^{-ya/(m-m^*) + xr'(m-m^*) + ar' + z})$ by substituting $r = -y/(m - m^*) + r'$. B has to verify if $z = 0$. From $\sigma = (U, V)$, B can compute (using the given values g^x, g^y, g^r, a)

  $$(U', V') = (g^{r'}, g^{xr'+z}),$$

  so $z = 0$ iff $(g, g^x, g^{r'}, g^{xr'+z})$ is a DDH tuple which can be checked with the DDH oracle $\mathcal{O}_x(\cdot, \cdot)$.

- $\mathsf{VRFY}_k(m^*, \sigma)$. A valid tag for message m^* is of the form $\sigma = (U, V) = (g^r, g^{xy+a)})$ from which the wanted Diffie-Hellman key g^{xy} can be computed by B.

This gives a perfect simulation of the suf-cmva experiment and the bounds on ε and t follow. ■

The above construction is only secure under the gap-CDH assumption. We now sketch 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 4.7 *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 $t' \approx t$.*

Proof. (Sketch) We only give a brief idea of the proof, as it is quite similar to the proof of Theorem 4.6. We say that the Gap Twin-CDH problem is (ε, t, Q_G)-hard if for all adversaries A that run in time t,

$$\Pr_{x_1, x_2, y \leftarrow_R \mathbb{Z}_p}[\mathsf{A}^{\mathcal{O}_{x_1, x_2}(\cdot,\cdot)}(1^\lambda, g^{x_1}, g^{x_2}, g^y) = g^{x_1 y}] \leq \varepsilon,$$

where A makes maximal Q_G queries to the Twin DDH oracle $\mathcal{O}$ with fixed basis g^{x_1}, g^{x_2}, i.e., $\mathcal{O}_{x_1,x_2}(R, Z_1, Z_2) = 1$ iff $R^{x_1} = Z_1$ and $R^{x_2} = Z_2$. If the CDH assumption is (t, ε)-hard, then the Twin Gap CDH problem is $(t, \varepsilon + O(Q_G)/p, Q_G)$-hard [13].

Now the proof of suf-cmva security of $\mathsf{MAC}_{\mathsf{TBB}}$ under the Twin Gap CDH assumption is the same as the proof of Theorem 4.6. ■

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 [44]. 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 [32, Section 4]. To both constructions can apply the transformations from Section 3 to obtain efficient uf-cmva-secure MACs.

The $\mathsf{LPN}_{\ell,\tau}$ assumption in dimension $\ell \in \mathbb{N}$ and Bernoulli parameter $0 < \tau \leq 1/2$ holds, if for fixed $x \leftarrow_R \mathbb{Z}_2^\ell$,

$$(\mathbf{r}_i^T, \mathbf{r}_i^T \cdot \mathbf{x} + e_i) \approx_c (\mathbf{r}_i^T, \mathbf{u}_i),$$

where for $1 \leq i \leq \mathrm{poly}(\ell)$, $\mathbf{r}_i \leftarrow_R \mathbb{Z}_2^\ell$ and $e_i \in \mathbb{Z}_2$ is sampled according to the Bernoulli distribution with parameter τ.

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 $\mathcal{K} = \mathbb{Z}_2^{2\ell}$, and tag space $\mathcal{T} = \mathbb{Z}_2^{(\ell+1)n}$. The message space is defined as $\mathcal{M} = \{\mathbf{m} \in \mathbb{Z}_2^{2\ell} : \mathrm{hw}(\mathbf{m}) = \ell\}$, where $\mathrm{hw}(\cdot)$ denotes the *hamming weight*. In other words, the messages must have hamming-weight ℓ. We assume that $\ell > n + \omega(\log(\lambda))$.

- 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: $\mathsf{rank}(R) = n$ and $|\mathbf{R}^T \cdot \mathbf{x}_{\downarrow \mathbf{m}} - \mathbf{z}| \leq \tau' n$.

Concretely, [32, Th. 4] shows (implicitly)[6] that $\mathsf{MAC}_{\mathsf{LPN}}$ has $2^{-O(n)}$ completeness error and is $\mathsf{suf\text{-}cma}$ and $\mathsf{ind\text{-}cma}$-secure under the $\mathsf{LPN}_{\ell,\tau}$ assumption in dimension $\approx \ell$ and Bernoulli paramter τ.

Second construction ($\mathsf{uf\text{-}cma}$). 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 chooses a secret key a matrix $\mathbf{X} \leftarrow_R \mathbb{Z}_2^{\ell \times \lambda}$ and $\mathbf{x}_0 \leftarrow_R \mathbb{Z}_2^{\ell}$ and outputs $(\mathbf{X}, \mathbf{x}_0)$.
- 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 $\sigma = (\mathbf{R}, \mathbf{R}^T \cdot (\mathbf{X} \cdot \mathbf{m} + \mathbf{x}_0) + \mathbf{e})$, 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: $\mathsf{rank}(R) = n$ and $|\mathbf{R}^T \cdot (\mathbf{X} \cdot \mathbf{m} + \mathbf{x}_0) - \mathbf{z}| \leq \tau' n$.

[32, Th. 5] shows that $\mathsf{MAC}_{\mathsf{BilinLPN}}$ is $\mathsf{uf\text{-}cma}$ and $\mathsf{ind\text{-}cma}$-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.

5 Symmetric Authentication Protocols

In this section we show how to directly build symmetric-key authentication protocols from any weak PRF, without relying on an intermediate MAC or PRF. But first we recall the definitions of authentication protocols, and make some definitions supporting our construction.

5.1 Definitions of Authentication Protocols

An authentication protocol Π is an interactive protocol executed between a Tag $\mathcal{T}$ and a reader $\mathcal{R}$, both PPT algorithms. Both hold a secret K (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, $\mathcal{R}$ outputs either accept or reject. We say that the protocol has completeness error α if for all $\lambda \in \mathbb{N}$, all secret keys K generated by $\mathsf{KG}(1^\lambda)$, the honestly executed protocol returns reject with probability at most α. We now define different security notions of an authentication protocol.

[6] [32] give a direct construction of a MAC that is $\mathsf{suf\text{-}cmva}$ secure. $\mathsf{MAC}_{\mathsf{LPN}}$ is the underlying MAC that can be proved only $\mathsf{suf\text{-}cma}$ secure.

MAC	2-round Symmetric authentication
uuf-cma ($\Leftarrow$ suf-cma $\Leftarrow$ uf-cma)	security against active attacks
uuf-cmva ($\Leftarrow$ suf-cmva $\Leftarrow$ uf-cmva)	security against man-in-the-middle attacks

Figure 2: Security relations between MACs and resulting 2-round authentication protocols.

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

ACTIVE ATTACKS. A stronger notion for authentication protocols is security against *active* attacks. Here the adversary A runs in two stages. First, she can interact with the honest tag a polynomial number of times (with concurrent executions allowed). In the second phase A interacts with the reader only, and wins if the reader returns accept. Here we only give the adversary one shot to convince the verifier.[7] An authentication protocol is (t, q, ε)-*secure against active adversaries* if every PPT A, running in time at most t and making q queries to the honest reader, 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 initially interact (concurrently) with any number of tags and – unlike in an active attacks – also readers. The adversary gets to learn the reader's accept/reject decisions.

FROM MAC TO SYMMETRIC AUTHENTICATION. It is well known that a MAC immediately implies a two-round (challenge-and-response) authentication protocol Π as follows. The reader $\mathcal{R}$ first sends a random challenge $c \leftarrow_R \mathcal{M}$ to the tag $\mathcal{T}$. Next, the tag answers with $\sigma = \mathsf{TAG}_k(c)$, which can be verified by $\mathcal{R}$ running $\mathsf{VRFY}_k(c, \sigma)$.

Formally, in order to achieve security against active attacks, the MAC only needs to be *universally unforgeable* against chosen message (and no verification) attacks, denoted uuf-cma. This means that after the end of the chosen message attacks (with no verification queries), it should be hard to produce a valid tag for a *random message* $c \in \mathcal{M}$. Clearly, selective security suf-cma implies uuf-cma security, whenever the message space is exponentially large. Thus, all our MAC constructions (even those which are selectively secure) are enough to yield two-round actively secure authentication protocols.

Moreover, if the MAC is also uuf-cmva secure against verification queries, the resulting authentication protocol is easily seen secure against MIM attacks. Once again, uuf-cmva security is clearly implied by suf-cmva security. Thus, since all our MACs are either directly secure against verification queries, or can be made such via our generic transformation in Theorem 3.4 (by virtue of being ind-cma secure), all our MAC constructions imply two-round MIM secure symmetric authentication protocols.

Table 2 shows the security relations between MACs and the resulting 2-round authentication protocols.

5.2 Three-Round Authentication from Any Weak PRF

In this section we show an elegant construction of actively secure authentication protocols based on *any weak pseudorandom function* (wPRF). Notice, wPRFs are much weaker primitives than regular

[7] 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.

PRFs. In particular, unlike regular PRFs, it is not known how to efficiently build a MAC using a weak PRFs, without using some inefficient tree-based construction [23], or relying on some *additional* properties, such as our MAC construction from *key-homomorphic* weak PRFs in Section 4.3.

Our authentication protocol will use any wPRF family $\mathcal{F}$ and any weak Almost XOR-Universal (wAXU) family $\mathcal{H}$ (whose definition is given below). As we will see shortly, we could set either $\mathcal{H} = \mathcal{F}$, or build $\mathcal{H}$ unconditionally.

Let $\mathcal{K} = \mathcal{K}(\lambda), \mathcal{X} = \mathcal{X}(\lambda), \mathcal{Y} = \mathcal{Y}(\lambda)$ be some spaces. Without loss of generality, we assume that the "range" $\mathcal{Y}$ forms an additive group (e.g., if $\mathcal{Y} = \{0,1\}^v$, we can use the XOR operation, explaining the name).

Definition 5.1 (wAXU Hash Functions) *A family $\mathcal{H}$ of keyed hash functions $\{f_k : \mathcal{X} \mapsto \mathcal{Y}\}_{k \in \mathcal{K}}$ is called (t, ρ)-*wAXU hash family*, if for any attacker $\mathsf{B}(x_1, x_2)$ running in time at most t and outputting some value $\Delta \in \mathcal{Y}$, we have $\Pr_{k \leftarrow \mathcal{K}, x_1, x_2 \leftarrow \mathcal{X}}[h(x_1) - h(x_2) = \mathsf{B}(x_1, x_2)] \leq \rho$. When $t = \infty$, we simply say that $\mathcal{H}$ is ρ-*wAXU.*

*If, instead, we have that for any $x_1 \neq x_2 \in \mathcal{X}$ and any $\Delta \in \mathcal{Y}$, we have $\Pr_{k \leftarrow \mathcal{K}}[h(x_1) - h(x_2) = \Delta] \leq \rho$, then $\mathcal{H}$ is called ρ-*AXU*. When $\rho = 1/|\mathcal{Y}|$, we say that $\mathcal{H}$ is (perfectly)* AXU*.*

We notice that ρ-AXU family is also $(\rho + 2^{-u})$-wAXU family, since the AXU definition allows the (non-uniform) attacker to *select* $x_1 \neq x_2$ and then hardwire Δ, instead of getting random x_1, x_2 (equal with probability 2^{-u}) and then producing Δ in some bounded time t. However, in contrast to regular AXU property, we can see that any (t, q, ε)-wPRF family is also $(t' \approx t, \rho)$-wAXU where $\rho = 1/|\mathcal{X}| + 1/|\mathcal{Y}| + \varepsilon$, since no B can win the wAXU game against a truly random function with probability above $(1/|\mathcal{X}| + 1/|\mathcal{Y}|)$. In particular, having some attacker B contradicting the $(t' \approx t, \rho)$-wAXU of our wPRF family, will immediately imply a distinguisher telling apart evaluations of h_k on two random points x_1, x_2 from evaluations of a random function on these two points, by subtracting the two outputs and testing if the difference is equal to $B(x_1, x_2)$.

In fact, wAXU (and regular AXU) hash functions can be easily constructed unconditionally. For example, when $\mathcal{X} = \{0,1\}^u$ and $\mathcal{Y} = \{0,1\}^v$, where $u \geq v$, the function $h_k(x) = [x \cdot k]_v$ is $(2^{-u} + 2^{-v})$-wAXU, where x and k are interpreted as elements of finite field $GF[2^u]$, $x \cdot y$ is field multiplication, and $[b]_v$ denotes the v least significant bits of b. Indeed, as long as $x_1 \neq x_2$ (which happens with probability 2^{-u}), the value $[(x_1 - x_2)k]_v$ is uniform in $\{0,1\}^v$. While the above family has large key length $|k| = u$, using standard polynomial hash [33], one can also get a $(2^{-u} + (1+\varepsilon)2^{-v})$-wAXU family with key length $O(\log(1/\varepsilon) + v)$.

Our Protocol. 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 $\mathcal{H} = \{h_{k_2} : \mathcal{X}_2 \mapsto \mathcal{Y}\}_{k_2 \in \mathcal{K}_2}$. (Recall, as explained above, we could set either $\mathcal{H} = \mathcal{F}$, or build $\mathcal{H}$ unconditionally.)

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 5.2 *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 $\varepsilon' = \sqrt{2\varepsilon + \frac{1}{|\mathcal{X}|} + \frac{1}{|\mathcal{Y}|}}$.

Proof. Assume that A runs in time t' and breaks the active security of Π with probability ε'. Let $\mathbf{Exp}_0$ be the original active security experiment, and $p_0 = \varepsilon'$ be A's success probability in $\mathbf{Exp}_0$.

We define $\mathbf{Exp}_1$ to be the following experiment. We complete the run of A as in $\mathbf{Exp}_0$. In particular, in the second impersonation stage A sends some message r^* to $\mathcal{R}$, gets random challenge x_1, responds with z_1^*, and wins if $z_1^* \stackrel{?}{=} f_{k_1}(r^*) + h_{k_2}(x_1)$. Namely,

$$p_0 \stackrel{\text{def}}{=} \Pr[z_1^* = f_{k_1}(r^*) + h_{k_2}(x_1)]$$

We now *rewind* A to the point right after it sent r^*, and then send a fresh random challenge x_2 to A. A then responds with z_2^*, and we say that A won if and only if A succeeded in *both runs*: $z_1^* = f_{k_1}(r^*) + h_{k_2}(x_1)$ and $z_2^* = f_{k_1}(r^*) + h_{k_2}(x_2)$. Let

$$p_1 \stackrel{\text{def}}{=} \Pr[z_1^* = f_{k_1}(r^*) + h_{k_2}(x_1) \text{ and } z_2^* = f_{k_1}(r^*) + h_{k_2}(x_2)]$$

be the probability that A won in the new experiment $\mathbf{Exp}_1$. We claim that $p_1 \geq p_0^2 > (\varepsilon')^2$. Indeed, for a given set of coins of A and $\mathcal{T}$ during the first stage, which we denote by α, let $p_0(\alpha)$ be the probability that A wins in $\mathbf{Exp}_0$, taken only over the choice of random challenge x_1. Then p_0 is the expectation of $p_0(\alpha)$ over α. Similarly, let $p_1(\alpha)$ be the probability that A wins in $\mathbf{Exp}_1$, taken only over the choice of random challenges x_1, x_2. Clearly, p_1 is the expectation of $p_1(\alpha)$ over α, but also for any fixed α, we have $p_1(\alpha) = p_0(\alpha)^2$. Hence, $p_1 = \mathbb{E}_\alpha[p_1(\alpha)] = \mathbb{E}_\alpha[p_0(\alpha)^2] \geq (\mathbb{E}_\alpha[p_0(\alpha)])^2 = p_0^2$.

We now define $\mathbf{Exp}_2$ to be the same experiment as $\mathbf{Exp}_1$, except we no longer test that $z_1^* \stackrel{?}{=} f_{k_1}(r^*) + h_{k_2}(x_1)$ and $z_2^* \stackrel{?}{=} f_{k_1}(r^*) + h_{k_2}(x_2)$, but *only* test that the difference between these equations holds: $h_{k_2}(x_1) - h_{k_2}(x_2) \stackrel{?}{=} z_1^* - z_2^*$. Let p_2 be the probability that A wins in $\mathbf{Exp}_2$:

$$p_2 \stackrel{\text{def}}{=} \Pr[h_{k_2}(x_1) - h_{k_2}(x_2) \stackrel{?}{=} z_1^* - z_2^*]$$

Clearly, $p_2 \geq p_1$, since we test an equality implied by the previous two equalities.

Next, we define the experiment $\mathbf{Exp}_3$ to be the same as $\mathbf{Exp}_2$ except A no longer needs to send the value r^* in the first round, since this value is never used by the challenger anymore. Moreover, instead of sending two responses z_1^* and z_2^*, we let A send the value $\Delta \stackrel{\text{def}}{=} z_1^* - z_2^*$. In particular, the new impersonation stage simply corresponds to the challenger sending two random challenges x_1 and x_2, and A responding with the value Δ (derived by running the original A twice to respond to challenges x_1 and x_2 on the same r^*, and subtracting those responses from each other). Let p_3 be the probability that A wins in $\mathbf{Exp}_3$:

$$p_3 \stackrel{\text{def}}{=} \Pr[h_{k_2}(x_1) - h_{k_2}(x_2) = \Delta]$$

Clearly, $p_3 = p_2$, since we simply changed the names of some variables.

Next, we define the experiment $\mathbf{Exp}_4$ to be the same as $\mathbf{Exp}_3$ except in the first "learning" stage the honest tag $\mathcal{T}$ does not use the true value $f_{k_1}(r_i)$ when preparing the i-th response $z_i = f_{k_1}(r_i) + h_{k_2}(x_i^*)$. Instead, $\mathcal{T}$ uses a completely random string $y_i \leftarrow \mathcal{Y}$ by setting $z_i = y_i + h_{k_2}(x_i^*)$. Let p_4 be the probability that A wins in $\mathbf{Exp}_3$. We claim that $p_4 \geq p_3 - \varepsilon$, by the wPRF security of $\mathcal{F}$. Indeed, all we did was replacing the value of f_{k_1} on *random points* r_i with a random string y_i. And since the winning condition check $h_{k_2}(x_1) - h_{k_2}(x_2) \stackrel{?}{=} \Delta$ no longer involves f_{k_1}, we can test it in the wPRF reduction by having the distinguisher pick its own k_2 and check if A won the experiment.

Finally, we notice that A is not getting any useful information from $\mathcal{T}$ in $\mathbf{Exp}_4$, since the values $z_i = y_i + h_{k_2}(x_i^*)$ are completely random and independent of k_2. So A can simply simulate the first stage by itself. Hence, we can define the final experiment $\mathbf{Exp}_5$ where there is only the second stage, where A gets random challenges x_1, x_2 and has to produce a value Δ. As before, A wins this game if

$h_{k_2}(x_1) - h_{k_2}(x_2) = \Delta$. Clearly, $p_5 = p_4$. However, the experiment $\mathbf{Exp}_5$ is precisely the execution of the wAXU experiment with the following attacker B running in time at most $2t' = t$. B simulates the first stage of A by simply returning random values to A, and then runs A twice on the same r^* and two fresh challenges x_1, x_2, and then outputs $\Delta = z_1^* - z_2^*$. By the wAXU security of $\mathcal{H}$, we get that $p_5 \leq \rho$, which means that

$$\rho \geq p_5 = p_4 \geq p_3 - \varepsilon = p_2 - \varepsilon \geq p_1 - \varepsilon \geq p_0^2 - \varepsilon = (\varepsilon')^2 - \varepsilon$$

completing the proof that $\varepsilon' \leq \sqrt{\rho + \varepsilon}$. ■

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 denore $k_1 = a$, $k_2 = b$, $r = g$, 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 [41]. 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.

Acknowledgements

We thank Elette Boyle for valuable comments on an earlier draft, and Abhishek Banerjee, Chris Peikert, and Alon Rosen for finding a mistake in an earlier version of the paper. We also thank Joël Alwen and Pavel Raykov for their valuable comments and for pointing out several minor errors and omissions in an earlier draft.

References

[1] 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

[2] Abhishek Banerjee, Chris Peikert, and Alon Rosen. Pseudorandom functions and lattices. In *EUROCRYPT*, pages 228–245, 2012. 3

[3] 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. 2, 4, 6

[4] 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. 2

[5] 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. 2

[6] 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/. 3

[7] Mihir Bellare and Chanathip Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In Tatsuaki Okamoto, editor, *Advances in Cryptology – ASIACRYPT 2000*, volume 1976 of *Lecture Notes in Computer Science*, pages 531–545. Springer, December 2000. 4

[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. 2

[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. 2

[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. 4

[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. 2, 5, 18

[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. 2

[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. 4, 5, 19

[14] Jean-Sébastien Coron. On the exact security of full domain hash. In Mihir Bellare, editor, *Advances in Cryptology – CRYPTO 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 229–235. Springer, August 2000. 2

[15] Jean-Sébastien Coron. Security proof for partial-domain hash signature schemes. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 613–626. Springer, August 2002. 2

[16] 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. 14

[17] Ronald Cramer and Victor Shoup. Signature schemes based on the strong RSA assumption. In *ACM CCS 99: 6th Conference on Computer and Communications Security*, pages 46–51. ACM Press, November 1999. 2

[18] 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. 4, 11, 14, 16

[19] 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. 2

[20] 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. 2

[21] 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. 2

[22] 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. 3

[23] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, 1986. 2, 22

[24] 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. 4

[25] 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. 4, 14

[26] 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. 4, 14

[27] 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. 3

[28] 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. 2

[29] 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. 3, 5

[30] 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. 5

[31] 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. 3

[32] 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. 3, 4, 12, 19, 20

[33] Hugo Krawczyk. LFSR-based hashing and authentication. In Yvo Desmedt, editor, *Advances in Cryptology – CRYPTO'94*, volume 839 of *Lecture Notes in Computer Science*, pages 129–139. Springer, August 1994. 22

[34] 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. 2

[35] 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. 4, 14

[36] Anna Lysyanskaya. Unique signatures and verifiable random functions from the DH-DDH separation. In Moti Yung, editor, *Advances in Cryptology – CRYPTO 2002*, volume 2442 of *Lecture Notes in Computer Science*, pages 597–612. Springer, August 2002. 2

[37] 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. 2

[38] 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. 2, 4, 5

[39] 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. 2

[40] 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. 4

[41] 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. 24

[42] 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. 4, 14

[43] 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. 3

[44] 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. 2, 5, 17, 19

[45] 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. 2