ANOTHER LOOK AT TIGHTNESS II: PRACTICAL ISSUES IN CRYPTOGRAPHY

SANJIT CHATTERJEE, NEAL KOBLITZ, ALFRED MENEZES, AND PALASH SARKAR

Abstract. How to deal with large tightness gaps in security proofs is a vexing issue in cryptography. Even when analyzing protocols that are of practical importance, leading researchers often fail to treat this question with the seriousness that it deserves. We discuss nontightness in connection with complexity leveraging, HMAC, lattice-based cryptography, identity-based encryption, and hybrid encryption.

1. Introduction

The purpose of this paper is to address practicality issues in cryptography that are related to nontight security reductions. A typical security reduction (often called a "proof of security") for a protocol has the following form: A certain mathematical task $\mathcal{P}$ reduces to the task $\mathcal{Q}$ of successfully mounting a certain class of attacks on the protocol — that is, of being a successful adversary in a certain security model. More precisely, the security reduction is an algorithm $\mathcal{R}$ for solving the mathematical problem $\mathcal{P}$ that has access to a hypothetical oracle for $\mathcal{Q}$. If the oracle takes time at most T and is successful with probability at least ϵ (here T and ϵ are functions of the security parameter k), then $\mathcal{R}$ solves $\mathcal{P}$ in time at most T' with probability at least ϵ' (where again T' and ϵ' are functions of k). We call $(T'\epsilon)/(T\epsilon')$ the *tightness gap*. The reduction $\mathcal{R}$ is said to be *tight* if the tightness gap is 1 (or is small); otherwise it is *nontight*. Usually $T' \approx T$ and $\epsilon' \approx \epsilon$ in a tight reduction.

A tight security reduction is often very useful in establishing confidence in a protocol. As long as one is not worried about attacks that lie outside the security model (such as side-channel attacks, duplicate-signature key selection attacks, or multi-user attacks [59]), one is guaranteed that the adversary's task is at least as hard as solving a certain well-studied mathematical problem (such as integer factorization) or finding a better-than-random way to predict output bits from a standardized primitive (such as AES).

The usefulness of a nontight security reduction is more controversial. If, for example, the tightness gap is 2^{40}, then one is guaranteed that the adversary's task is at least 2^{-40} times as hard as solving the mathematical problem or compromising AES. Opinions about whether nontightness is a cause of concern depend on how much importance one attaches to quantitative guarantees. In his paper [11] explaining practice-oriented provable security, Bellare writes:

> Practice-oriented provable security attempts to explicitly capture the inherently *quantitative* nature of security, via a *concrete* or *exact* treatment

Date: 20 April 2016; updated on 25 August 2016.

> of security.... This enables a protocol designer to know exactly how much security he/she gets. (emphasis in original)

In contrast, some researchers minimize the importance of quantitative security and object strongly when someone criticizes a practice-oriented provable security result for giving a useless concrete security bound. For example, an anonymous reviewer of [57] defended the nonuniform proof in [12], acknowledging that its nonuniformity "reduces the quantitative guarantees" but then stating:

> Many proofs do not yield tight bounds, but they still are powerful qualitative indicators of security.

This reviewer characterized the use of the word "flaw" in [57] in reference to a fallacious analysis and erroneous statement of quantitative guarantees as "misleading" and "offensive," presumably because the "qualitative indicators" in [12] were still valid.

What makes the nontightness question particularly sensitive is that cryptographers are supposed to be cautious and conservative in their recommendations, and sources of uncertainty and vulnerability are not supposed to be swept under the rug. In particular, one should always keep in mind the possibility of what Menezes in [68] calls the *nightmare scenario* — that there actually is an attack on the protocol that is reflected in the tightness gap.

In [27] the authors presented attacks on MAC schemes in the multi-user setting — attacks that are possible because the natural security reduction relating the multi-user setting to the single-user setting is nontight. Similar attacks on protocols in the multi-user setting were given for a network authentication protocol, aggregate MAC schemes, authenticated encryption schemes, disk encryption schemes, and stream ciphers.

In Appendix B we describe the attacks of Zaverucha [83] on hybrid encryption in the multi-user setting. In §5 we describe another situation where the tightness gap reflects the fact that there's an actual attack, in this case due to Pietrzak [75, 40].

A practical issue that is closely related to the nontightness question is the matter of safety margins. There are at least two kinds of safety margins: (1) parameter sizes that give significantly more bits of security than are currently needed, and (2) "optional" features in a protocol that are believed (sometimes because of tradition and "instinct" rather than any rigorous security argument) to help prevent new attacks or attacks that are outside the commonly used security models.

At present it is widely agreed that it is prudent to have at least 128 bits of security.[1] Why not 96? In the near future it is unlikely that anyone (even the NSA) will expend 2^{96} operations to break a protocol. The reason for insisting on 128 bits of security is that one should anticipate incremental improvements in cryptanalytic attacks on the underlying mathematical problem that will knock several bits off the security level. If nontightness has already reduced the security assurance provided by the proof from 128 to 96 bits (and if the parameter sizes have not been increased so as to restore 128 bits of security), then even relatively small advances in attacking the mathematical problem will bring the security assurance further down to a level where a successful attack on the protocol is feasible in principle.

[1] By "k bits of security" we mean that there is good reason to believe that, if a successful attack (of a specified type) takes time T and has success probability ϵ, then $T/\epsilon > 2^k$.

A common explanation of the value of security proofs is that features that are not needed in the proof can be dropped from the protocol. For instance, Katz and Lindell make this point in the introduction to [49]. However, in Appendix B (see also §5 of [59]) we shall find that optional features included in protocols often thwart attacks that would otherwise reduce the true security level considerably.

On the one hand, there is widespread agreement that tight proofs are preferable to nontight ones, many authors have worked hard to replace nontight proofs with tighter proofs when possible, and most published security reductions duly inform the reader when there is a large tightness gap. On the other hand, authors of papers that analyze protocols that are of practical importance almost never suggest larger parameters that compensate for the tightness gap. Presumably the reason is that they would have to sacrifice efficiency. As Bellare says [11],

> A weak reduction means that to get the same level of security in our protocol we must use larger keys for the underlying atomic primitive, and this means slower protocols.

Indeed, many standardized protocols were chosen in part because of security "proofs" involving highly nontight security reductions. Nevertheless, we are not aware of a single protocol that has been standardized or deployed with larger parameters that properly account for the tightness gaps. Thus, acknowledgment of the nontightness problem remains on the level of lip service.

In §§3-7 we discuss nontightness in connection with complexity leveraging, HMAC, lattice-based cryptography, and identity-based encryption; in Appendix B we discuss Zaverucha's results on nontightness in security proofs for hybrid encryption in the multi-user setting. In the case of HMAC, in view of the recent work [57, 40] on the huge tightness gaps in pseudorandomness results, in §5 we recommend that standards bodies reexamine the security of HMAC when used for non-MAC purposes (such as key derivation or passwords) or with MD5 or SHA1.

2. An Important Caveat

In our view, any scientific work that makes ambitious claims of practical importance needs to be examined carefully and critically. One should not be blinded by hype or wishful thinking, or by the authors' impressive credentials. In an interdisciplinary field such as cryptography, where mistakes can be devastating, it is important to welcome the commentary of people with a variety of backgrounds — mathematicians, engineers, and hackers, as well as computer scientists.

However, an important caveat must be made. It is not right to trash work that contains elegant ideas and makes no claim to have practical applications in the foreseeable future. The proof of Fermat's Last Theorem in 1995 was rightly regarded as a major achievement of human thought. Closer to our field, work on the oracle-complexity of factoring, first by Rivest and later by Maurer, was elegant and compelling. It would be anti-intellectual and philistine to ridicule this type of work because it has no known applications outside of theory.[2] (See [54] for a discussion of this type of philistinism.)

[2]In the trip-report [71] about Eurocrypt 1992, the NSA author makes fun of Maurer's results with sarcastic humor.

One of the negative consequences of the anti-intellectualism that is so prevalent in the United States and some other countries is that in grant applications and elsewhere theoretical mathematicians have sometimes exaggerated or even fabricated a connection between their research and cryptography. As Koblitz commented in the *Notices of the American Mathematical Society* [53], "It was sad that some mathematicians seemed to feel pressured into portraying their research as being somehow related to cryptography." Part of the explanation is that people are responding to the common notion in our society that scholarship has to be commercially useful.

* * *

In his well-written and thought-provoking essay [77], Rogaway sharply criticizes what he calls *crypto-for-crypto*, "meaning that it doesn't ostensibly benefit commerce or privacy, and it's quite speculative if it will ever evolve to do either." In particular, he ridicules the entire fields of Fully Homomorphic Encryption (FHE) and indistinguishability Obfuscation (iO) as "speculative, theory-centric directions" that DARPA (the U.S. Defense Advanced Research Projects Agency) is happy to fund precisely because they are so useless. Rogaway's essay suggests that it is immoral (or amoral) to work in such areas. This theory-bashing is regrettable. Both FHE and iO are fields in their infancy, and attacking them in such an extreme way is like trying to strangle them at birth.

Some authors of papers on FHE and iO do perhaps deserve to be criticized for hyping their work and misleading readers about its relation to practice. However, both fields have produced some elegant ideas and constructions. And it's not completely true that none of it is of practical use. For example, Lauter *et al.* [52, 65] have used homomorphic encryption to develop methods of privacy protection for human genome datasets. As far as we can judge, this work is practical and even (in Rogaway's sense) "moral."

In contrast to FHE and iO, which he regards as useless, Rogaway gives a series of recommendations for "moral" cryptography that are based in part on his own work and in part on other work that he favors. It is perfectly reasonable to have strong opinions about desirable areas of research. However, to suggest that those who choose to work in other subfields are "amoral" is uncollegial and a tad arrogant.

3. Complexity Leveraging

"Complexity leveraging" is a general technique for proving that a cryptographic protocol that has been shown to be *selectively secure* is also *adaptively secure*. Here "selectively secure" means that the adversary has to select its target before it is presented with its inputs (e.g., public keys, signing oracles, etc.), whereas "adaptive security" means that the adversary is free to select its target at any time during its attack. The second type of adversary is in general much stronger than the first type. Thus, selective security is in principle a much weaker result than adaptive security, and so is not usually relevant to practice. Because selective security is often easier to prove than adaptive security, researchers devised the method of complexity leveraging to convert any selective security theorem into an adaptive security theorem.

Complexity leveraging has been used to prove the adaptive security of many kinds of cryptographic protocols including identity-based encryption [23], functional encryption [39],

constrained pseudorandom functions [25], and constrained verifiable random functions [35]. In §3.1 we illustrate the problems with complexity leveraging in the context of signature schemes. In §3.2 we consider the case of identity-based encryption.

3.1. **Signature schemes.** The most widely accepted definition of security of a signature scheme is against an *existential forger under chosen-message attack*. This means that the forger is given a user's public key and is allowed $\leq q$ queries, in response to which she is given a valid signature on each queried message. The forger is successful if she then forges a signature for any message M other than one that was queried.

A much weaker property is security against a *selective forger*. In that case the adversary is required to choose the message M for which she will forge a signature before she even knows the user's public key. She cannot modify M in response to the public key or the signature queries, and to be successful she must forge a signature on the original M. Selective security is obviously much weaker than existential security. A theorem that gives only selective security is not generally regarded as satisfactory for practice.

Complexity leveraging works by converting an arbitrary existential forger into a selective forger, as follows. The selective forger Cynthia guesses a message M, which she desperately hopes will be the message on which the existential forger eventually forges a signature. She then runs the existential forger. She is successful if the message forged is M; otherwise she simply tries again with a different guess. Her probability of success in each run is $\epsilon = 2^{-\mathfrak{m}}$, where $\mathfrak{m}$ is the allowed bitlength of messages. The bound $\mathfrak{m}$ on the message length could be large, such as one gigabyte.

Fortunately for Cynthia, in practice messages are normally hashed, say by SHA256, and it is the hash value that is signed. Thus, Cynthia needs to guess the 256-bit hash value of the message on which the existential forger forges a signature, not the message itself. Her probability of success is then 2^{-256}, and so the tightness gap in going from selective to existential security is 2^{256}.

Suppose, for example, that we have an integer-factorization-based signature protocol for which selective security has been shown to be tightly equivalent to factoring. How large does the modulus N have to be so that the corresponding existential security theorem gives us a guarantee of 128 bits of security? If only 3072-bit N is used, then the protocol will have 128 bits of selective security, but complexity leveraging gives us no existential security, because of the 2^{256} tightness gap. In order to have 128 bits of existential security, we need to have $128 + 256 = 384$ bits of security against factoring N, and this means roughly 40,000-bit N. Even though this is what we must do if we want complexity leveraging to give us the desired security, no one would ever seriously recommend deploying 40,000-bit moduli. Thus, from a practical standpoint complexity leveraging gives us nothing useful here.

3.2. **Identity-based encryption.** Boneh and Boyen [23] used bilinear pairings on elliptic curves to design an identity-based encryption scheme. They proved that their scheme is selectively secure in the sense that the adversary has to select the target before she gets the public parameters and access to the appropriate oracles (see §7 for background on identity-based encryption). The highlight of the proof is that it does not invoke the random oracle assumption.

Boneh and Boyen [23, Theorem 7.1] then used complexity leveraging to prove that a generic identity-based encryption scheme that is selectively secure is also adaptively secure.

The proof has a tightness gap of $2^{2\ell}$, where ℓ is the desired security level and 2ℓ is the output length of a collision-resistant hash function (the hash function is applied to the identifiers of parties). Boneh and Boyen remarked that the reductionist proof is "somewhat inefficient" and explained that the desired level of security can be attained by increasing the parameters of the underlying pairing.

Suppose now that one desires 128 bits of security. Suppose also that the proof of selective security for the identity-based encryption scheme is tight. Then one can achieve 128 bits of selective security by using an (asymmetric) bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ derived from a prime-order Barreto-Naehrig (BN) elliptic curve E over a finite field $\mathbb{F}_p$ [10]. Here, p is a 256-bit prime, $\mathbb{G}_1 = E(\mathbb{F}_p)$, $\mathbb{G}_2$ is a certain order-n subgroup of $E(\mathbb{F}_{p^{12}})$, and $\mathbb{G}_T$ is the order-n subgroup of $\mathbb{F}^*_{p^{12}}$, where $n = \#E(\mathbb{F}_p)$. This pairing is ideally suited for the 128-bit security level since the fastest attacks known on the discrete logarithm problems in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ all take time approximately 2^{128}.[3] If resistance to adaptive attacks is desired, then to account for the tightness gap of 2^{256} a pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T$ should be selected so that the fastest attacks known on the discrete logarithm problems in $\mathbb{G}_1$, $\mathbb{G}_2$ and $\mathbb{G}_T$ take time at least 2^{384}. If the protocol is implemented using BN curves, then one now needs $p^{12} \approx 2^{40000}$ and thus $p \approx 2^{3300}$. Consequently, computations in $\mathbb{G}_1$ and $\mathbb{G}_T$ will be over 3300- and 40000-bit fields, instead of 256- and 3072-bit fields had the reduction been tight. Hence, the tightness gap that arises from complexity leveraging has a very large impact on efficiency.

4. Nonuniformity to Achieve Better Tightness

Informally speaking, the difference between a *nonuniform* algorithm to solve a problem $\mathcal{P}$ and the more familiar notion (due to Turing) of a uniform algorithm is that the former is given an "advice string," depending on the input length (and usually assumed to be of polynomial size in the input length). In general, a nonuniform algorithm is more powerful than a uniform one because the advice string may be very helpful in solving $\mathcal{P}$. Several prominent researchers have repeatedly claimed that security theorems that are proved in the nonuniform model of computation are stronger than theorems proved in the uniform model, because they provide assurances against successful attacks by nonuniform as well as uniform adversaries. In their lecture notes for their 2008 course at MIT [42], Bellare and Goldwasser state:

> Clearly, the nonuniform adversary is stronger than the uniform one. Thus to prove that "something" is "secure" even in presence of a nonuniform adversary is a better result than only proving it is secure in presence of a uniform adversary. (p. 254)

In an email explaining why his paper [12] did not inform the reader that the security reduction was being given in the nonuniform model, Bellare wrote [13]:

> I had no idea my paper would be read by anyone not familiar with the fact that concrete security is nonuniform.

[3] We are not accounting for recent progress by Kim and Barbulescu [51] in algorithms for computing discrete logarithms in $\mathbb{G}_T$. This will lead to working with even larger parameters.

What these researchers are failing to take into account is that the use of the nonuniform model makes *the hypothesis as well as the conclusion* of the theorem stronger. Thus, the theorem's assumption that a certain mathematical task is hard or that a certain compression function cannot be distinguished from a random function has to allow nonuniform algorithms. It is usually very difficult to get any idea of the strength of the commonly-used primitives against nonuniform attacks, and in practice they are not designed to withstand such attacks. See [58] for a discussion of the history of confusion about this issue in the literature and a detailed rebuttal of the arguments in favor of the nonuniform model in cryptography.

Whether or not nonuniform algorithms for a problem $\mathcal{P}$ are known that are much faster than uniform ones depends very much on the problem $\mathcal{P}$.

Example 1. (*No known difference between uniform and nonuniform.*) There is no known nonuniform algorithm for the general integer factorization problem that is faster than the fastest known uniform algorithms.

In the next two examples, let $\mathcal{H}_k$ be a fixed family of hash functions, one for each security level k. In both examples, suppose that the input is k written in unary (this is a trick used to allow the input length to be different for different k).

Example 2. (*Trivial in the nonuniform model.*) For a well-constructed family $\mathcal{H}_k$, by definition one knows no efficient uniform algorithm for finding a collision. In contrast, one has a trivial nonuniform algorithm, since the advice string can consist of two messages whose hash values are equal.

Example 3. (*Between these two extremes.*) Consider the problem of distinguishing a hash function $\mathcal{H}_k$ in a family of keyed hash functions from a random function; a function for which this cannot be done with non-negligible success probability is said to have the pseudorandom function property (PRF). More precisely, an attack on the PRF property is an algorithm that queries an oracle that with equal probability is either the hash function with hidden key or else a random function and, based on the responses, can determine which it is with probability $\epsilon + 1/2$ of being correct, where the *advantage* ϵ is significantly greater than 0. For a well-constructed hash function no uniform algorithm is known that is faster than simply guessing the key, and this has advantage roughly $T/2^\ell$, where ℓ is the key-length and T is the time (here we are assuming that each query takes unit time). However, there is a simple nonuniform algorithm that runs in unit time and distinguishes a hash function with hidden key from a random function with advantage roughly $2^{-\ell/2}$ — an advantage that would take the uniform algorithm time $T \approx 2^{\ell/2}$ to achieve. Our advice string is a message M that has a very special property with respect to $\mathcal{H}_k$ when averaged over all possible keys. For example, let M be a message that maximizes the probability that the 29th output bit is 1 rather than 0. The nonuniform algorithm then queries M to the oracle; if the oracle's response has 29th bit equal to 1, it guesses that the oracle is the hash function with hidden key, but if the 29th bit is 0, it guesses that the oracle is a random function. It follows by an easy argument from the theory of random walks that the expected advantage of this nonuniform algorithm is roughly $2^{-\ell/2}$.

As pointed out in [58], almost all security proofs in the literature are valid in the uniform model of complexity, and only a few use what's sometimes called *coin-fixing* to get a proof

that is valid only in the nonuniform model. As far as we are aware, none of the nonuniform theorems in the literature have hypotheses of the sort in Examples 1 and 2; all are like Example 3, that is, the task whose hardness is being assumed is easier in the nonuniform model, but not trivial. The authors' main purpose in using coin-fixing in these cases is to achieve a tighter security reduction than they could have achieved in the uniform model.

Unfortunately, it is easy to get tripped up if one attempts to use coin-fixing to get a stronger result — authors fool themselves (and others) into thinking that their result is much stronger than it actually is. The most important example of a researcher who was led astray by his belief in the nonuniform model is Bellare in his Crypto 2006 paper [12] on HMAC. We will summarize this story and carry it up to the present by discussing some errors in his revised version [14], which was recently published in the *Journal of Cryptology*.

5. The HMAC Saga

HMAC [17, 19] is a popular hash-function-based message authentication code (MAC). The controversy about nonuniform reductions concerns security proofs of the PRF property (see Example 3 of §4) of NMAC, which is a MAC that is closely related to HMAC. We shall discuss NMAC rather than HMAC, because the extension of results from NMAC to HMAC has generated relatively little controversy (see [60] for an analysis of 1-key variants of NMAC).

By a compression function we mean a function $z = f(x, y)$, where $y \in \{0,1\}^b$ and $x, z \in \{0,1\}^c$; typically $b = 512$ and c is equal to either 128 (for MD5), 160 (for SHA1), or 256 (for SHA256).

Given a compression function f, to construct an iterated hash function $\mathcal{H}$ one starts with an *initialization vector* IV, which is a publicly known bitstring of length c that is fixed once and for all. Suppose that $M = (M_1, \ldots, M_m)$ is a message consisting of $m \leq \mathfrak{m}$ b-bit blocks (where $\mathfrak{m}$ is the bound on the block-length of messages; for simplicity we suppose that all message lengths are multiples of b). Then we set $x_0 = \text{IV}$, and for $i = 1, \ldots, m$ we recursively set $x_i = f(x_{i-1}, M_i)$; finally, we set $\mathcal{H}(M) = \mathcal{H}_{\text{IV}}(M) = x_m$, which is the c-bit hash value of M.[4]

Suppose that Alice shares two secret c-bit keys K_1 and K_2 with Bob, and wants to create an NMAC-tag of a message M so that Bob can verify that the message came from Alice. She first uses K_1 as the IV and computes $\mathcal{H}_{K_1}(M)$. She pads this with $b - c$ zeros (denoted by a 0-superscript) and sets her tag $t(M)$ equal to $\mathcal{H}_{K_2}(\mathcal{H}_{K_1}(M)^0)$.

The purpose of finding a security reduction for NMAC is to show that if one has confidence that the compression function f enjoys a certain security property, then one can be sure that NMAC has the same property. Two decades ago HMAC was first proposed by Bellare, Canetti, and Krawczyk [17, 19]. In [17] they proved (assuming weak collision-resistance of $\mathcal{H}$) that if f has the secure-MAC property, then so does NMAC. (The secure-MAC property is analogous to existential security of signatures, see §3.) The proof in [17] was tight. It was also short and well-written; anyone who was considering using HMAC could readily verify that the proof was tight and correct.

[4] In iterated hash functions one also appends a "length block" to the message M before hashing. We are omitting the length block for simplicity.

In 2006 Bellare [12] published a different security reduction for NMAC. First, he dispensed with the collision-resistance assumption on $\mathcal{H}$, which is a relatively strong assumption that has turned out to be incorrect for some real-world iterated hash functions. Second, he replaced the secure-MAC property with the stronger PRF property, that is, he showed that if $f(x, y)$ (with x serving as the hidden key) has the PRF property, then so does NMAC. This was important in order to justify the use of HMAC for purposes other than message authentication — in applications where the PRF property is desired, such as key-derivation protocols [34, 45, 63] and password-systems [70].

Remark 1. A third advantage (not mentioned in [12, 14]) of assuming the PRF property rather than collision-resistance arises if one derives a concrete security assurance using the best known generic attacks on the property that the compression function is assumed to have. As far as we know the best generic attack on the PRF property using classical (i.e., uniform and non-quantum) algorithms has running time $\approx 2^c$ (it amounts to guessing the hidden key), whereas the birthday-paradox attack on collision-resistance only takes time $\approx 2^{c/2}$. Other things being equal, one expects that c must be twice as great if one is assuming collision-resistance than if one is assuming the PRF property.

However, in 2012 Koblitz and Menezes found a flaw in [12]. For Bellare, who along with Rogaway popularized the concept of "practice-oriented provable security" [11], his theorem was not merely a theoretical result, but rather was intended to provide some concrete assurance to practitioners. Thus, it was important for him to determine in real-world terms what guarantee his theorem provided. To do this, Bellare's approach was to take the fastest known generic attack on the PRF property of a compression function, and evaluate what his theorem then implied for the security of NMAC. In his analysis he took the key-guessing attack (see Example 3 of §4) as the best generic attack on f, and concluded that NMAC is secure "up to roughly $2^{c/2}/\mathtt{m}$ queries." For instance, for a bound of $\mathtt{m} = 2^{20}$ on the block-length of messages Bellare was claiming that NMAC-MD5 is secure up to 2^{44} queries and NMAC-SHA1 up to 2^{60} queries. (In 2006, MD5 and SHA1 were common choices for hash functions.)

Bellare failed to account for the fact that, because of his "coin-fixing," i.e., nonuniform security reduction, he was logically required to examine security of f against *nonuniform* attacks, not just uniform attacks. As we saw in §4, there are simple generic nonuniform attacks on the PRF property that have a much higher success probability than the key-guessing attack. If one repeats Bellare's analysis using the nonuniform attack described in §4, one finds that NMAC's security is guaranteed only up to at most $2^{c/4}/\sqrt{\mathtt{m}}$ queries, that is, 2^{22} for NMAC-MD5 and 2^{30} for NMAC-SHA1. That level of security is of little value in practice.

When we say that Bellare's paper had a basic flaw, we have in mind the definition of the f-word that was given by Stern, Pointcheval, Malone-Lee, and Smart [82], who said:

> The use of provable security is more subtle than it appears, and flaws in security proofs themselves might have a devastating effect on the trustworthiness of cryptography. By flaws, we do not mean plain mathematical errors but rather ambiguities or misconceptions in the security model.

* * *

Now let us bring this story up to the present. In an effort to determine what can be said about the relation between the PRF property of the compression function f and the PRF property of NMAC, Koblitz and Menezes [57] gave a uniform security reduction that had tightness gap $\mathfrak{m} \cdot \max(2,\ q^2/(2^c\epsilon))$, where ϵ is a measure of the PRF-security of f and q is a bound on the number of queries. They had to use a stronger version of the PRF property of f (a version that's similar to the property used in [18]); a corollary of their theorem then gave a tightness gap of $2\mathfrak{m}q$ if one assumes only standard PRF-security of f.[5]

The interpretation in [57] of the authors' Theorem 10.1 and Corollary 10.3 on NMAC security is pessimistic. Those results assume the single-user setting and strong properties of f; moreover, they have large tightness gaps. The authors conclude:

> We would not want to go out on a limb and say that our Theorem 10.1 is totally worthless. However, its value as a source of assurance about the real-world security of HMAC is questionable at best.

Specifically, they caution that "In our opinion none of the provable security theorems for HMAC with MD5 or SHA1 [...] by themselves provide a useful guarantee of security." For instance, suppose that the query bound q is 2^{30}, the block-length bound $\mathfrak{m}$ is 2^{25}, and the number of users n is 2^{25}. (As we shall see in Appendix B, the step from single-user to multi-user setting introduces an additional factor of n in the tightness gap.) Then the number of bits of security drops by $30 + 25 + 25 = 80$ due to these tightness gaps. In other words, the guarantees drop to 48 bits and 80 bits in the case of MD5 and SHA1, respectively.

Remark 2. If SHA256 is used in order to have at least 128 bits of HMAC security, then there is such a huge safety margin that even these tightness gaps do not lower the security to an undesirable level, at least if one assumes that there is no attack on the PRF property of the SHA256 compression function that is faster than the generic key-guessing one. This is because key-guessing takes time $\approx 2^{256}$, leaving a safety margin of 128 bits. One reason SHA256 might be used for HMAC even if only 128 bits of security are required is that the user might need SHA256 for other protocols that require collision-resistance and so she cannot allow fewer than 256 bits of hash-output; in the interest of simplicity she might decide to use a single hash function everywhere rather than switching to SHA1 for HMAC.

Remark 3. The above comment about a huge safety margin when SHA256 is used in HMAC applies only if a 256-bit key and 256-bit message tags are used. Not all standards specify this. For example, the NIST standard [33] recommends 128-bit HMAC keys for 128 bits of security and allows 64-bit tags. The recommendations in [33] are supported by an *ad hoc* analysis, but are not supported by any provable security theorem.

Aside from the issue of the tightness gaps, there is another fundamental reason why the theorems in [12, 14, 57] about security of NMAC and HMAC under the PRF assumption offer little practical assurance. To the best of our knowledge, the PRF assumption has never been seriously studied for the compression functions used in MD5, SHA1, or SHA256; in

[5]The early posted versions of [57] contained a serious error that was pointed out to the authors by Pietrzak, namely, the theorem is given assuming only the PRF property rather than the strong PRF property that is needed in the proof. This error was explained and corrected in the posted versions and the published version. Soon after the corrected version was posted, Pietrzak posted a paper [75] containing a different proof of essentially the same result as in Corollary 10.3 of Theorem 10.1 of [57] (see also [40]).

fact, we are not aware of a single paper that treats this question. Moreover, when those compression functions were constructed, the PRF property was not regarded as something that had to be satisfied rather, they were constructed for the purpose of collision-resistance and pre-image resistance. Thus, in the case of the concrete hash functions used in practice, we have no evidence that could rule out attacks on the PRF property that are much better than the generic ones. It would be very worthwhile for people to study how resistant the concrete compression functions are to attacks on the PRF property; in the meantime it would be prudent not to rely heavily on theorems that make the PRF assumption.

Remark 4. The situation was quite different for AES, since a longstanding criterion for a good block cipher has been to have the pseudorandom permutation (PRP) property with respect to the secret (hidden) key. That is, an adversary should not be able to distinguish between the output of a block cipher with hidden key and that of a random permutation. The PRF property is close to the PRP property as formalized by the PRP/PRF switching lemma (see Section 5 of [79]), and so it is reasonable to assume that AES has the PRF property. On the other hand, the criteria for judging hash constructions have been very different from those for judging encryption.

Remark 5. In [15] the authors prove security of a MAC scheme called AMAC, which is a prefix-MAC in which the output of the hash function is truncated so as to thwart the extension attacks to which prefix-MACs are susceptible. As in the case of the HMAC papers discussed above, the authors of [15] assume that the underlying compression function is a PRF. Their proof has the remarkable feature that it does not lose tightness in the multi-user setting. On the other hand, the tightness gap in the single-user setting is much larger than in the above security reductions for HMAC — namely, roughly $q^2\mathtt{m}^2$. With, for instance, $q \approx 2^{30}$ and $\mathtt{m} \approx 2^{25}$ one has a tightness gap of 110 bits. The paper [15] does recommend the use of SHA512, and if one assumes 512 bits of PRF-security for its compression function, then we have such a large safety margin that a 2^{110} tightness gap is not worrisome. Nevertheless, it should be stressed that the PRF assumption is a very strong one that, to the best of our knowledge, has never been studied or tested for the SHA512 compression function.

Remark 6. In [43], Goldwasser and Kalai propose a notion of what it means for a complexity assumption to be reasonable in the context of reductionist security proofs. Among other things, the assumption should be falsifiable and non-interactive. Since the assumption that the compression function in a hash function such as MD5, SHA1, SHA256 or SHA512 has the PRF property is an interactive one, it does not meet the Goldwasser-Kalai standard for a reasonable cryptographic assumption. Rather, in the words of Goldwasser and Kalai, such an assumption "can be harmful to the credibility of our field."

Returning to our narrative, in 2015 Bellare [14] published a revised version of [12] in *J. Cryptology* that, regrettably, just muddied the waters because of errors and unclarities in his abstract and introduction that could easily mislead practitioners. First of all, the first sentence of the abstract states that the 1996 paper [17] proved "HMAC...to be a PRF assuming that (1) the underlying compression function is a PRF, and (2) the iterated hash

function is weakly collision resistant." In fact, only the secure-MAC property, not the PRF property, was proved in [17].[6]

In the second place, in the concluding paragraph of the introduction of [14] Bellare gives the impression that Pietrzak in [75] proved tight bounds for the PRF-security of NMAC:[7] "Tightness estimates [in the present paper] are now based on the blackbox version of our reductions and indicate that our bounds are not as tight as we had thought. The gap has been filled by Pietrzak [75], who gives blackbox reduction proofs for NMAC that he shows via matching attack to be tight."[8] A practitioner who reads the abstract and introduction of [14] but not the technical sections would probably go away believing that PRF-security of NMAC has been proved to be tightly related to PRF-security of the compression function. This is false. In fact, it is the opposite of what Pietrzak proved.

What Pietrzak showed in [75, 40] was that the $\mathfrak{m}q$ tightness gap cannot be reduced in the general case (although the possibility that better tightness might conceivably be achieved for a special class of compression functions wasn't ruled out). He found a simple attack on NMAC that shows this. This is far from reassuring — it's what Menezes in [68] called the "nightmare scenario." To put it another way, Pietrzak's attack shows a huge separation in PRF-security between the compression function and NMAC. The desired interpretation of a security reduction of the sort in [14], [57] or [40] is that it should tell you that the study of a certain security property of a complicated protocol is unnecessary if one studies the corresponding property of a standard primitive. In this case the tightness gap along with Pietrzak's attack show that this is *not* the case.

It is unfortunate that neither of Bellare's papers [12, 14] discuss the practical implications of the large tightness gap. It would be interesting to know why he disagrees with the conclusion of Koblitz–Menezes that the tightness gaps and other weaknesses render the security reductions (proved by them in Theorems 10.1 and Corollary 10.3 of [57]) "questionable at best" as a source of real-world assurance. In view of Pietrzak's recent work, which shows that the tightness gap cannot be removed and reflects an actual attack, it is particularly puzzling that even the revised paper [14] has nothing to say about the practical implications of this weakness in the security reductions for HMAC.

We conclude this section with a recommendation. *Standards bodies should reexamine — taking into account tightness gaps — the security of all standardized protocols that use HMAC for non-MAC purposes such as key derivation or passwords. The same should be done for HMAC-protocols using hash functions such as MD5 or SHA1 that are not believed to have weak collision-resistance in the sense of* [17].

[6]The abstract to [40] also erroneously states that "NMAC was introduced by Bellare, Canetti and Krawczyk [Crypto96], who proved it to be a secure pseudorandom function (PRF), and thus also a MAC, assuming that (1) f is a PRF and (2) the function we get when cascading f is weakly collision-resistant."

[7]In this quotation Bellare uses the word "blackbox" in a non-standard way. Later in his paper he defines a "blackbox" reduction to be one that is constructible and a "non-blackbox" reduction to be one that is non-constructible. However, when comparing a proof as in [12] that uses "coin-fixing" with more recent proofs that do not, the standard terms are nonuniform/uniform rather than non-blackbox/blackbox.

[8]The section "Our Contributions" in [40] starts out: "Our first contribution is a simpler, uniform, and as we will show, basically tight proof for the PRF-security of NMAC^f assuming only that f is a PRF." The authors apparently meant to say that their tightness gap is best possible, i.e., cannot be improved. Their proof is not tight, however — far from it. Their tightness gap is nq, essentially the same as in Corollary 10.3 of [57].

In some cases adjustments should be made, such as mandating a feature that is currently optional (such as a nonce or a randomization) in order to prevent known attacks; in other cases the recommended parameters or choices of hash function may need to be changed in order to account for the tightness gaps. Protocols that use HMAC as a MAC and use a collision-resistant hash function do not have to be reexamined, because in that case [17] has a tight security reduction. (However, in view of the multi-user attacks discussed in Appendix B, the standards for any protocol that is used in a setting with a large number of users should be modified if necessary to account for the multi-user/single-user tightness gap.)

6. Lattice-Based Quantum-Safe Crypto

The reason for intense interest in lattice-based cryptography can be traced back to the early years of public key, when Merkle–Hellman proposed the knapsack public-key encryption system. It aroused a lot of interest both because of its superior efficiency (compared to RSA) and the supposedly high level of confidence in its security, since it was based on an NP-hard problem. Within a few years Shamir, Brickell and others completely broke both the original knapsack and modified versions of it. It turned out that the knapsack was based on an easy subproblem of the NP-hard subset sum problem, not on hard instances. This was a traumatic experience for researchers in the nascent field of public-key cryptography. The lesson learned was that it would be good to base systems on hardness of a problem for which the average case is provably equivalent to the hardest case (possibly of a different problem).

There was a lot of excitement (even in the popular press) when Ajtai–Dwork announced a lattice-based encryption scheme based on such a problem [2, 3]. Since that time much of the motivation for working on lattice-based systems (especially now that standards bodies are looking for quantum-safe cryptographic protocols that have provable security guarantees) is that many of them can be proved to have worst-case/average-case equivalence. (For a comprehensive overview of research on lattice-based cryptography in the last ten years, see [74].) In this section we shall look at the worst-case to average-case reductions from the standpoint of tightness.

First, though, it is important to recognize that equivalence between average and worst cases is not the Holy Grail for cryptographers that some might think. As Dan Bernstein has noted (quoted in [43]), long before Ajtai-Dwork we had discrete-log cryptosystems over characteristic-two fields. For each k the Discrete Log Problem (DLP) in the group $\mathbb{F}_{2^k}^*$ is random self-reducible, meaning that instances can be randomized. This gives a tight equivalence between hardest instances and average instances. However, the DLP in those groups has long been known to be weaker than the DLP in the multiplicative group of prime-order fields [30], and recently it was completely broken [9].

Meanwhile the general DLP in the multiplicative group of prime fields $\mathbb{F}_p^*$ does not have this nice self-reducibility property, since for a given bitlength of p one has vastly different levels of difficulty of the DLP. Yet as far as we know these groups are secure for suitably chosen p of bitlength > 1024.

6.1. Lattices.

A (full rank) *lattice* L in $\mathbb{R}^n$ is the set of all integer linear combinations of n linearly independent vectors $B = \{v_1, v_2, \ldots, v_n\}$. The set B is called a *basis* of L, and the

dimension of L is n. If the v_i are in $\mathbb{Z}^n$, then L is said to be an integer lattice; all lattices in this section are integer lattices. The *length* of a vector is its Euclidean norm. For each $1 \leq i \leq n$, the *ith successive minimum* $\lambda_i(L)$ is the smallest real number r such that L has i linearly independent vectors the longest of which has length r. Thus, $\lambda_1(L)$ is the length of a shortest nonzero vector in L.

6.2. **Lattice problems.** Let L be an n-dimensional lattice. When we say that we are "given a lattice" L, we mean that we are given some arbitrary basis for L.

A well-studied lattice problem is the Shortest Vector Problem (SVP): Given L, find a lattice vector of length $\lambda_1(L)$. The SVP problem is NP-hard (under randomized reductions). The fastest classical algorithms known for solving it have provable running time $2^{n+o(n)}$ [1] and heuristic running time $2^{0.337n+o(n)}$ [64]. The fastest quantum algorithm known for solving SVP has heuristic running time $2^{0.286n+o(n)}$ [64]. More generally, one can consider the Approximate-SVP Problem (SVP_γ), which is the problem of finding a nonzero lattice vector of length at most $\gamma \cdot \lambda_1(L)$. If $\gamma > \sqrt{n}$, then SVP_γ is unlikely to be NP-hard [41]. In fact, if $\gamma > 2^{n \log\log n / \log n}$, then SVP_γ can be solved in polynomial time using the LLL algorithm. For $\gamma = 2^k$, the fastest algorithm known for SVP_γ has running time $2^{\tilde{\Theta}(n/k)}$, where the $\tilde{\Theta}$ term hides a constant factor and a factor of a power of $\log n$ (see [74]).

A related problem to SVP_γ is the Approximate Shortest Independent Vectors Problem (SIVP_γ): Given L, find n linearly independent lattice vectors all of which have length at most $\gamma \cdot \lambda_n(L)$. The hardness of SIVP_γ is similar to that of SVP_γ [21]; in fact, $\text{SIVP}_{\sqrt{n}\gamma}$ polynomial-time reduces to SVP_γ [69].

6.3. **Learning with errors.** The Learning With Errors (LWE) problem was introduced by Regev in 2005 [76]. The LWE problem and the related R-LWE problem (see [67]) have been extensively used to design many cryptographic protocols including public-key encryption, identity-based encryption, and fully homomorphic encryption. Public-key encryption schemes based on LWE (and R-LWE) are also attractive because no quantum algorithms for solving LWE are known that perform better than the fastest known classical algorithms. Thus, LWE-based public-key encryption schemes are viable candidates for post-quantum cryptography.

Let $q = q(n)$ and $m = m(n)$ be integers, and let $\alpha = \alpha(n) \in (0, 1)$ be such that $\alpha q > 2\sqrt{n}$. Let χ be the probability distribution on $\mathbb{Z}_q$ obtained by sampling from a Gaussian distribution with mean 0 and variance $\alpha^2/2\pi$, and then multiplying by q and rounding to the closest integer modulo q; for more details see [76]. Then the (search version of the) LWE problem is the following: Let s be a secret vector selected uniformly at random from $\mathbb{Z}_q^n$. Given m samples $(a_i, a_i \cdot s + e_i)$, where each a_i is selected independently and uniformly at random from $\mathbb{Z}_q^n$, and where each e_i is selected independently from $\mathbb{Z}_q^n$ according to χ, determine s. Intuitively, in LWE you are asked to solve a linear system modulo q, except that the constants on the right of the system are given to you with random errors according to a Gaussian distribution.

The decisional version of LWE, called DLWE, asks us to determine whether we have been given m LWE samples $(a_i, a_i \cdot s + e_i)$ or m random samples (a_i, u_i), where each u_i is selected independently and uniformly at random from $\mathbb{Z}_q$.

6.4. **Regev's reduction.** Regev [76] proved the following remarkable result[9]

Theorem 1. *If there exists an efficient algorithm that solves DLWE (in the average case), then there exists an efficient quantum algorithm that solves SIVP$_\gamma$ in the worst case where $\gamma = \tilde{O}(n/\alpha)$.*

Suppose now that a lattice-based cryptosystem has been designed with a reductionist security proof with respect to the hardness of average-case DLWE. By Theorem 1, this cryptosystem also has a reductionist security proof with respect to the hardness of SIVP$_\gamma$ in the *worst case*. This is widely interpreted as providing ironclad assurance for the security of the cryptosystem since there is compelling evidence that the well-studied SIVP$_\gamma$ problem is hard in the worst case when γ is small.

However, Regev's theorem and similar results are asymptotic. Although results of this type are interesting from a qualitative point of view, it is surprising that in the literature there are virtually no attempts to determine the concrete security assurances that worst-case to average-case results such as Theorem 1 provide for lattice-based cryptosystems. That is, in the lattice-based cryptography literature concerning worst-case/average-case results, practice-oriented provable security in the sense of Bellare-Rogaway (as explained in the quote from [11] in the Introduction) is conspicuous by its absence.

Remark 7. Suppose that one has a polynomial-time reduction of a well-studied worst-case problem Π_1 to an average-case problem Π_2. Then, if one assumes that the worst-case instances of Π_1 are not polytime solvable, then the reduction provides the assurance that no polynomial-time algorithm can solve Π_2 on average. This asymptotic assurance is viewed by some as ruling out "structural weaknesses" in Π_2; for example, see Section 5.1 of [66]. However, in the absence of a concrete analysis, the reduction by itself does not guarantee the hardness of fixed-sized instances of Π_2.

A closer examination of Theorem 1 reveals several obstacles to using it to obtain concrete security assurances for DLWE-based cryptosystems. We list five such difficulties. Whereas the first and second are widely acknowledged in the literature, there is scant mention of the remaining three difficulties.

(1) One needs to assess the hardness of SIVP$_\gamma$ under *quantum* attacks and not just under attacks on classical computers.
(2) For parameters n, q and α that arise in DLWE-based cryptosystems, the SIVP$_\gamma$ problem is likely *not* NP-hard. Thus, the evidence for worst-case hardness of SIVP$_\gamma$ instances that arise in lattice-based cryptography is not as compelling as the evidence for the worst-case hardness of an NP-hard problem.
(3) Very little work has been done on concretely assessing the hardness of SIVP$_\gamma$. As mentioned in §6.2, the fastest attack on SIVP$_\gamma$ where $\gamma = 2^k$ has running time $2^{\tilde{\Theta}(n/k)}$; however this expression for the running time is far from concrete.
(4) The statement of Theorem 1 uses "efficient" to mean "polynomial time in n". However, the exact tightness gap in the reduction of worst-case SIVP$_\gamma$ to average-case DLWE has to the best of our knowledge never been stated.

[9]Regev's theorem can also be stated with the GapSVP$_\gamma$ problem instead of SIVP$_\gamma$. Given an n-dimensional lattice L and a number $r > 0$, GapSVP$_\gamma$ requires that one output "yes" if $\lambda_1(L) \leq r$ and "no" if $\lambda_1(L) > \gamma r$ (either "yes" or "no" is allowed if $r < \lambda_1 \leq \gamma r$).

(5) A more precise formulation of DLWE involves several parameters including the number of available samples and the adversary's advantage in distinguishing between LWE and random samples. In practice, these parameters have to be chosen based on the security needs of the DLWE-based cryptosystem. However, there is little discussion in the literature of concrete values for these parameters in the context of specific protocols. All the reductionist security claims that we examined for DLWE-based cryptosystems are stated in asymptotic terms and make liberal use of the phrases "polynomial time," "polynomial number," and "non-negligible."

§6.5 elaborates on (4) and (5).

6.5. **Analysis of Regev's reduction.** A careful examination of Regev's proof of Theorem 1 (see Appendix A for the details) reveals the following refined statement. For concreteness, we will take $q = n^2$ and $\alpha = 1/(\sqrt{n}\log^2 n)$, whence $\gamma = \tilde{O}(n^{1.5})$; these are the parameters proposed by Regev for his DLWE-based public-key encryption scheme [76]. Suppose that there is an algorithm W_1 that, given $m = n^c$ samples, solves DLWE for a fraction $1/n^{d_1}$ of all $s \in \mathbb{Z}_q^n$ with advantage at least $1/n^{d_2}$. Then there is a polynomial-time algorithm W_2 for solving SIVP_γ that calls the W_1 oracle a total of

$$O(n^{11+c+d_1+2d_2}) \tag{1}$$

times. The tightness gap is thus $O(n^{11+c+d_1+2d_2})$. While this is polynomial in n, it can be massive for concrete values of n, c, d_1 and d_2.

Suppose, for example, that one takes $n = 1024$ ($n = 1024$ is used in [5, 26] for implementations of an R-LWE based cryptosystem). In a DLWE-based encryption scheme such as Regev's [76], the public key is a collection of $m = n^{1+\epsilon}$ LWE samples and the secret key is s; for simplicity we take $m = n$ whence $c = 1$. The encryption scheme is considered to be insecure if an attacker can distinguish between encryptions of 0 and 1 with advantage at least $1/n^d$ for some $d > 0$ depending on the security parameter. This advantage is assessed over choices of public-private key pairs and the randomness in the encryption algorithm. Regev showed that such an adversary can be used to solve DLWE for a fraction $1/4n^d$ of all $s \in \mathbb{Z}_q^n$ with advantage at least $1/8n^d$; thus $d_1 \approx d$ and $d_2 \approx d$. If one is aiming for the 128-bit security level, then a reasonable choice for d might be 12.8. Then, ignoring the hidden constant in the expression (1), the tightness gap is $n^{50.4} \approx 2^{504}$. Thus, if average-case DLWE can be solved in time T, then Theorem 1 shows that SIVP_γ can be solved by a quantum algorithm in time $2^{504}T$. As mentioned above, the fastest quantum algorithm known for solving SVP has running time $2^{0.286n+o(n)}$. If we assume that this is also the fastest quantum algorithm for solving SIVP_γ and ignore the $o(n)$ term in the exponent, then the algorithm has running time approximately $2^{293} \ll 2^{504}T$. Thus, for our choice of parameters Theorem 1 provides no assurances whatsoever for the hardness of average-case DLWE or for the security of the encryption scheme. In other words, even though Theorem 1 is viewed by many as providing "powerful qualitative indicators of security" (in the words of the anonymous reviewer quoted in §1), the quantitative security assurance it provides is vacuous.

Remark 8. The condition $\alpha q > 2\sqrt{n}$ is needed for Regev's proof of Theorem 1 to go through. It was later discovered that this condition is indeed necessary for security. In 2011, Arora and Ge [7] showed that if $\alpha q = n^t$, where $t < 1/2$ is a constant and $q \gg n^{2t}\log^2 n$,

then there is a subexponential $2^{\tilde{O}(n^{2t})}$ algorithm that solves LWE. This attack is touted as a demonstration of the importance of security proofs — Theorem 1 anticipated the Arora-Ge attack which was discovered 6 years after Theorem 1 was proven. In the same vein, one can wonder about the implications of the large tightness gap in Theorem 1 for the concrete hardness of DLWE. One needs to ask: Is the tightness gap anticipating yet-to-be-discovered algorithms for solving DLWE that are considerably faster than the fastest algorithms for solving $\mathrm{SIVP}_{n^{1.5}}$? The answer to this question has major consequences for the security of DLWE-based protocols.

On the other hand, if one were to select a larger value for n while still targeting the 128-bit security level, then the large tightness gap in (1) might not be a concern if there is a very large safety margin — large enough so that the fastest quantum algorithm for solving the corresponding SIVP_γ is believed to have running time 2^k for $k \gg 128$. While this necessitates selecting a larger value of n, the impact on the cryptosystem's performance might not be too large. Thus, there remains the possibility that Theorem 1 can indeed provide meaningful security assurances for DLWE-based cryptosystems in practice. In order for this to occur, the following problems should be further investigated:

(1) Determine concrete lower bounds for the worst-case quantum hardness of SIVP_γ (or GapSVP_γ) in terms of n and γ.
(2) Determine whether the tightness gap in Regev's worst-case to average-case reduction (see the estimate (1)) can be improved. Such improvements might be achieved either through a closer analysis of Regev's reduction, or else by formulating new reductions.
(3) Determine appropriate values of c, d_1 and d_2.
(4) Assess the tightness gap in the reductionist security proof for the cryptosystem (with respect to average-case DLWE).

Similarly, it would be very worthwhile to assess whether the analogue of Theorem 1 for the R-LWE problem provides any meaningful assurances for cryptosystems based on R-LWE using parameters that have been proposed in recent work [72, 4, 26, 5]. We note that the worst-case to average-case reduction for R-LWE [67] is with respect to SVP_γ in so-called ideal lattices (that is, lattices that come from ideals in rings). Deriving concrete bounds on the hardness of SVP_γ for these lattices is more challenging than deriving concrete bounds on the hardness of SIVP_γ for arbitrary lattices.

Remark 9. In preparation for the possible advent of large-scale quantum computers, standards organizations have begun examining candidates for public-key cryptosystems that withstand attacks by quantum computers (see [61]). Public-key cryptosystems based on R-LWE are considered to be one of the leading candidates for these quantum-safe standards. Initial deployment of quantum-safe cryptosystems will likely be for the protection of highly sensitive data whose confidentiality needs to be assured for several decades. For these applications, long-term security guarantees will be more important than short-term concerns of efficiency. Thus, it would be prudent to select parameters for R-LWE cryptosystems in such a way that the worst-case to average-case reductions provide meaningful concrete security guarantees. As mentioned above, the degradation in performance that results from larger lattice parameters might not be of great concern for high-security applications.

Remark 10. NTRU is a lattice-based public-key encryption scheme that was first presented in 1996 (see [47, 46]) and has been standardized by several accredited organizations including ANSI [6] and IEEE [48]. NTRU uses lattices that arise from certain polynomial rings. The algebraic structure of these lattices facilitate implementations that are significantly faster than public-key encryption schemes based on LWE and R-LWE. Despite its longevity, NTRU is routinely disparaged in the theoretical cryptography literature because, unlike the case of public-key encryption schemes based on LWE or R-LWE (including some variants of NTRU that were proposed more recently [81]), there are no worst-case to average-case reductions to support the security of its underlying lattice problems. However, as we have noted, whether or not these asymptotic worst-case to average-case reductions provide meaningful concrete security assurances is far from being understood. Thus, the claim that, because of worst-case/average-case reductions, the more recent lattice-based encryption schemes have better security than classical NTRU rests on a flimsy scientific foundation.

In [73] Peikert describes asymptotic analyses of the security of lattice-based systems, and concludes:

> ...worst-case reductions give a hard-and-fast guarantee that the cryptosystem is at least as hard to break as the *hardest* instances of some underlying problem. This gives a true lower bound on security, and prevents the kind of unexpected weaknesses that have so often been exposed in schemes that lack such reductions.

This would be true in a meaningful sense if the reductions were tight and if the underlying problem were SIVP_γ for a small γ (small enough so that SIVP_γ is NP-hard or so that there is reason to have confidence that there are no efficient algorithms for SIVP_γ). However, neither is the case. When discussing asymptotic results and writing for a broad readership interested in practical cryptography, the use of such terms as "hard-and-fast guarantee" and "true lower bound on security" is inappropriate and misleading, because in real-world cryptography the normal interpretation of these terms is that one has concrete practical security assurances.

7. Tightness in Identity-Based Encryption

By way of counterpoint to the main theme of this paper — the potential dangers in ignoring tightness gaps in security reductions — we now discuss the case of Boneh-Franklin Identity-Based Encryption (IBE), where a large tightness gap is, we believe, of no concern. The evidence for this belief is that an informal (but convincing) argument allows one to reduce to the case where the adversary is not allowed any key-extraction queries.

An identity-based encryption scheme offers the flexibility of using any string — in particular, the identity of an individual or entity — as a public key. There is an authority called the Private Key Generator which publishes its own public parameters, including a public key, and maintains a master secret key. To obtain a decryption key corresponding to her identity, a user in the system applies to the Private Key Generator, which performs appropriate checks (possibly including physical checks) to ascertain the identity. Then the Private Key Generator uses its public parameters and master secret key to generate the decryption key corresponding to the identity. This decryption key is transmitted to the

user through a secure channel. Anybody who wishes to securely send a message uses the identity of the recipient and the public parameters to perform the encryption. The recipient can decrypt using her decryption key.

Security of an IBE scheme is modeled using a game between a simulator and an adversary [24]. The game models security against an attack by a set of colluding users attempting to decrypt a ciphertext intended for a user outside the set.

In the initial phase, the simulator sets up an instance of the scheme based on the security parameter. The simulator generates the public parameters, which are given to the adversary, and the master secret key. The adversary is allowed to adaptively make key-extraction queries to the simulator, who must provide the decryption keys corresponding to identities of the adversary's choosing. At some point, the adversary provides the simulator with an identity $\mathsf{id}^\star$ (called the target identity) and two messages M_0 and M_1 of equal length. The simulator randomly chooses a bit b and provides the adversary with $C^\star$, which is an encryption of M_b for the identity $\mathsf{id}^\star$. The adversary continues making key-extraction queries in an adaptive manner. Finally, the adversary outputs its guess b'; its advantage in winning the game is defined to be $|\mathsf{Pr}[b = b'] - 1/2|$. The adversary may not make more than one key-extraction query for the same id; and of course it must not have queried the simulator for the decryption key of $\mathsf{id}^\star$, as otherwise the game becomes trivial to win. The adversary's resources are measured by the time that it takes and the number of key-extraction queries that it makes.

The model that we have described provides what is called IND-ID-CPA security (indistinguishability for ID-based encryption under key-extraction[10] attack). This model does not allow the adversary to make decryption queries. The model where such queries are also allowed is said to provide IND-ID-CCA (chosen ciphertext) security.

The first efficient IBE construction is due to Boneh and Franklin [24]. Their scheme — and in fact all subsequent efficient IBE constructions — uses bilinear pairings. A (symmetric) bilinear pairing is a map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$, where $\mathbb{G} = \langle P \rangle$ and $\mathbb{G}_T$ are groups of some prime order p, that satisfies the following conditions: $e(aP, bP) = e(P, P)^{ab}$, $e(P, P) \neq 1$, and e is efficiently computable. Practical bilinear pairings are obtained from elliptic curves where $\mathbb{G}$ is a subgroup of points on an appropriately chosen elliptic curve and $\mathbb{G}_T$ is a subgroup of the multiplicative group of a finite field.

Identity-based encryption schemes are proved secure under various computational hardness assumptions. We mention the basic bilinear Diffie-Hellman (BDH) assumption and two of its derivatives. The bilinear Diffie-Hellman (BDH) assumption is that computing $e(P, P)^{abc}$ given (P, aP, bP, cP) is infeasible. The decisional bilinear Diffie-Hellman (DBDH) assumption is that distinguishing between the distributions $(P, aP, bP, cP, e(P, P)^{abc})$ and $(P, aP, bP, cP, e(P, P)^z)$, where a, b, c and z are independent and uniform random choices from $\mathbb{Z}_p$, is infeasible. The gap bilinear Diffie-Hellman (GBDH) assumption is that computing $e(P, P)^{abc}$ given (P, aP, bP, cP) and access to a DBDH oracle is infeasible.

We now briefly describe the basic Boneh-Franklin IBE scheme. The Private Key Generator sets up the scheme by selecting a generator P of the group $\mathbb{G}$; choosing a random s from $\mathbb{Z}_p$ and setting $Q = sP$; and selecting two hash functions $H_1 : \{0, 1\}^* \to \mathbb{G}$, $H_2 : \mathbb{G}_T \to$

[10]In the IBE setting "CP" does not stand for *chosen plaintext* but rather for *clave pedida*, which means "requested key" in Spanish.

$\{0,1\}^n$. The public parameters are (P, Q, H_1, H_2) while the master secret key is s. Given an identity $\mathsf{id} \in \{0,1\}^*$, let $Q_{\mathsf{id}} = H_1(\mathsf{id})$; the decryption key is defined to be $d_{\mathsf{id}} = sQ_{\mathsf{id}}$. Encryption of an n-bit message M for the user with identity id is done by first choosing a random r in $\mathbb{Z}_p$ and then computing the ciphertext $(C_1, C_2) = (rP, M \oplus H_2(e(Q, Q_{\mathsf{id}})^r))$. Decryption is made possible from the relation $e(Q, Q_{\mathsf{id}})^r = e(rP, d_{\mathsf{id}})$.

Note that the basic Boneh-Franklin scheme does not provide chosen-ciphertext security, because the message occurs in the ciphertext only in the last XOR step. This means that a plaintext M can be determined from its ciphertext (C_1, C_2) by asking for the decryption of the ciphertext (C_1, C_2'), where C_2' is C_2 with the first bit flipped. One can, however, obtain IND-ID-CPA security results for the basic Boneh-Franklin scheme under the assumption that H_1 and H_2 are random oracles.

Using the Fujisaki-Okamoto transformation [36], the basic Boneh-Franklin IBE scheme can be converted into a scheme, called FullIdent (see [24]), that provides IND-ID-CCA security. To get FullIdent the basic scheme is modified as follows. First, a random $\rho \in \{0,1\}^n$ is chosen and r is set equal to $H_3(\rho, M)$, where H_3 is a hash function that maps bitstrings to integers mod p; we then define $C_1 = rP$ as before. The second component C_2 of the ciphertext is defined by $C_2 = \rho \oplus H_2(e(Q, Q_{\mathrm{id}})^r)$ (that is, the hash value is XORed with ρ rather than with M), and we also need a third component C_3 defined by $C_3 = M \oplus H_4(\rho)$, where H_4 is a hash function that maps $\{0,1\}^n$ to $\{0,1\}^n$. The decryption proceeds by first computing $\rho = C_2 \oplus H_2(e(C_1, d_{\mathrm{id}}))$ and then $M = C_3 \oplus H_4(\rho)$. But the decryption rejects the ciphertext unless it is validated by checking that $H_3(\rho, M)P = C_1$. This last check is very important, since it prevents an adversary from generating a valid ciphertext for an unknown message M.

Boneh and Franklin [24] argued for the IND-ID-CCA security of their construction using a three stage reduction based on BDH; the reduction turned out to be flawed. Galindo [38] provided a corrected reduction which resulted in a tightness gap of q_H^3, where q_H is the maximum number of queries made to any of the random oracles H_1, H_2, H_3 or H_4. Zhang and Imai [84] provided a direct reduction based on the same BDH assumption with a tightness gap of $q_D \cdot q_E \cdot q_H$, where q_D bounds the number of decryption queries and q_E bounds the number of key-extraction queries made by the adversary.[11] The tightness gap can be reduced to $q_E \cdot q_H$ by making the following change to the simulation of the H_3 random oracle in the proof of Theorem 1 in [84]: when the simulator responds to a query (σ_i, M_i) with r_i, it stores g^{r_i} in addition to (σ_i, M_i, r_i) in its "H_3-list" (here we're using the notation of the proof in [84] rather than our own notation, in which σ would be ρ and g^r would be rP). With this change, the simulator can respond to all q_D decryption queries in time q_D instead of $q_D \cdot q_H$ (we are ignoring the time to sort and search the H_3-list). As a result, the lower bound for the BDH-time now has order equal to the sum of the query bounds $q_D + q_{H_2} + q_{H_4} + q_E$, which is essentially the adversary's running time. In other words, in this way we can remove the tightness gap in the running times, and we're left with the tightness gap $q_E \cdot q_{H_2}$ that comes from the success probabilities in Theorem 1 of [84].

[11] In Table 1 of [84], Zhang and Imai claim that their security reduction has a tightness gap of $q_E \cdot q_H$; this assertion is repeated in Table 4 of [8]. However, they neglected to account for the tightness gap arising from the running times in Theorem 1 of their paper.

As noted in [8], the tightness gap reduces further to q_E if one is willing to base the security on the presumably stronger DBDH or GBDH assumptions. In practice, the hash functions in the IBE constructions are publicly known functions. Thus, the number of queries made to these functions by the adversary can be quite high — q_H could be 2^{64} or even 2^{80} for powerful adversaries. The number of key-extraction queries q_E, on the other hand, will be lower.

An informal argument can be used to show why the tightness gaps in the reductions for Boneh-Franklin IBE are inconsequential for real-world security. Namely, we claim that key-extraction queries give no useful information to the adversary, and so without loss of generality we may take $q_E = 0$; in that case, as mentioned above, there is a tight reduction based on the DBDH or GBDH assumption. Recall that in response to a queried id, the Private Key Generator returns $Q_{\mathsf{id}} = H_1(\mathsf{id})$, where H_1 is a random oracle, and $d_{\mathsf{id}} = sQ_{\mathsf{id}}$. This can be simulated by the adversary itself, who chooses k mod p at random and sets $Q_{\mathsf{id}} = kP$ and $d_{\mathsf{id}} = kQ$. Note that this does not give a valid formal reduction from the case when $q_E > 0$ to the case when $q_E = 0$, because the adversary does not get the "true" key pair of the user, whose public point is produced by the random oracle H_1. However, it is hard to conceive of any difference this could possibly make in the adversary's effectiveness against the IND-ID-CCA security of FullIdent.

Remark 11. In §3.1 of [56] Koblitz and Menezes made an analogous informal argument in order to conclude that the tightness gap in the security reduction for RSA Full Domain Hash should not be a cause of concern. These examples show, as remarked in [55], that "whether or not a cryptographic protocol lends itself to a tight security reduction argument is not necessarily related to the true security of the protocol.... the question of how to interpret a nontight reductionist security argument has no easy answer."

8. Conclusion

Reductionist arguments can contribute to our understanding of the real-world security of a protocol by providing an ironclad guarantee that certain types of attacks are infeasible as long as certain hardness assumptions remain valid. However, even this limited kind of assurance may, as we have seen, turn out to be meaningless in practice if the reduction is nontight and the parameters have not been increased to account for the tightness gap. In order to properly evaluate provable security claims, one needs to study the tightness issue. In this paper we have given examples of the type of analysis of tightness that should be performed, but much work remains to be done. Among the open problems are the following:

(1) Examine all uses of complexity leveraging to see whether or not the concrete adaptive security results are meaningful.
(2) Evaluate the effect on the required parameter sizes of nontightness in security proofs for HMAC and adjust standards accordingly, particularly in applications that require the pseudorandom function property; also study whether or not the commonly used hash compression functions are likely to satisfy the PRF assumption.
(3) Carefully evaluate all lattice-based protocols that have worst-case-to-average-case reductions to see what meaningful concrete bounds, if any, follow from these reductions.

(4) For protocols whose security reductions lose tightness in the multi-user setting or the multi-challenge setting (or both), determine how parameter sizes should be increased to account for this.

Acknowledgments

We wish to thank Greg Zaverucha for extensive help with Appendix B as well as useful comments on the other sections, Michael Naehrig for reviewing and commenting on §6, Somindu C. Ramanna for providing helpful comments on an earlier draft of §7, Ann Hibner Koblitz for editorial suggestions, and Ian Blake, Eike Kiltz, and Chris Peikert for helpful feedback and suggestions. Of course, none of them is responsible for any of the opinions expressed in this article.

References

[1] D. Aggarwal, D. Dadush, O. Regev, and N. Stephens-Davidowitz, Solving the shortest vector problem in 2^n time via discrete Gaussian sampling, *Proc. 47th Annual Symp. Foundations of Computer Science*, 2015, pp. 733-742.

[2] M. Ajtai, Generating hard instances of lattice problems, *Proc. 28th Annual ACM Symp. Theory of Computing*, ACM, 1996, pp. 99-108.

[3] M. Ajtai and C. Dwork. A public-key cryptosystem with worst-case/average-case equivalence, *Proc. 29th Annual ACM Symp. Theory of Computing*, ACM, 1997, pp.284-293.

[4] M. Albrecht, R. Player, and S. Scott, On the concrete hardness of Learning with Errors, *Journal of Mathematical Cryptology*, **9** (2015), pp. 169-203.

[5] E. Alkim, L. Ducas, T.Pöppelmann, and P. Schwabe, Post-quantum key exchange – a new hope, available at http://eprint.iacr.org/2015/1092.

[6] ANSI X9.98, Lattice-Based Polynomial Public Key Establishment Algorithm for the Financial Services Industry, Part 1: Key Establishment, Part 2: Data Encryption, 2010.

[7] S. Arora and R. Ge, New algorithms for learning in presence of errors, *ICALP 2011*, LNCS 6755, Springer-Verlag, 2011, pp. 403-415.

[8] N. Attrapadung, J. Furukawa, T. Gomi, G. Hanaoka, H. Imai, and R. Zhang, Efficient identity-based encryption with tight security reduction, *CANS 2006*, LNCS 4301, Springer-Verlag, 2006, pp. 19-38.

[9] R. Barbulescu, P. Gaudry, A. Joux, and E. Thomé, A heuristic quasi-polynomial algorithm for discrete logarithm in finite fields of small characteristic: Improvements over FFS in small to medium characteristic, *Advances in Cryptology — Eurocrypt 2014*, LNCS 8441, Springer-Verlag, 2014, pp. 1-16.

[10] P. Barreto and M. Naehrig, Pairing-friendly elliptic curves of prime order, *Selected Areas in Cryptography — SAC 2005*, LNCS 3897, Springer-Verlag, 2016, pp. 319-331.

[11] M. Bellare, Practice-oriented provable-security, *Proc. First International Workshop on Information Security (ISW '97)*, LNCS 1396, Springer-Verlag, 1998, pp. 221-231.

[12] M. Bellare, New proofs for NMAC and HMAC: Security without collision-resistance, *Advances in Cryptology — Crypto 2006*, LNCS 4117, Springer-Verlag, 2006, pp. 602-619.

[13] M. Bellare, email to N. Koblitz, 24 February 2012.

[14] M. Bellare, New proofs for NMAC and HMAC: Security without collision-resistance, *J. Cryptology*, **28** (2015), pp. 844-878.

[15] M. Bellare, D. Bernstein, and S. Tessaro, Hash-function based PRFs: AMAC and its multi-user security, *Advances in Cryptology — Eurocrypt 2016*, LNCS 9665, Springer-Verlag, 2016, pp. 566-595.

[16] M. Bellare, A. Boldyreva, and S. Micali, Public-key encryption in a multi-user setting: Security proofs and improvements, *Advances in Cryptology — Eurocrypt 2000*, LNCS 1807, Springer-Verlag, 2000, pp. 259-274; full version available at https://cseweb.ucsd.edu/~mihir/papers/musu.html.

[17] M. Bellare, R. Canetti, and H. Krawczyk, Keying hash functions for message authentication, *Advances in Cryptology — Crypto 1996*, LNCS 1109, Springer-Verlag, 1996, pp. 1-15.

[18] M. Bellare, R. Canetti, and H. Krawczyk, Pseudorandom functions revisited: The cascade construction and its concrete security, *Proc. 37th Annual Symp. Foundations of Computer Science*, 1996, pp. 514-523; extended version available at http://cseweb.ucsd.edu/users/mihir/papers/cascade.pdf.
[19] M. Bellare, R. Canetti, and H. Krawczyk, HMAC: Keyed-hashing for message authentication, Internet RFC 2104, 1997.
[20] D. Bernstein, Multi-user Schnorr security, revisited, available at http://eprint.iacr.org/2015/996.pdf.
[21] J. Blömer and J. Seifert, On the complexity of computing short linearly independent vectors and short bases in a lattice, *Proc. 31st Annual ACM Symp. Theory of Computing*, ACM, 1999, pp. 711-720.
[22] A. Boldyreva, Strengthening security of RSA-OAEP, *Topics in Cryptology — CT-RSA 2009*, LNCS 5473, Springer-Verlag, 2009, pp. 399-413.
[23] D. Boneh and X. Boyen, Efficient selective-ID secure identity based encryption without random oracles, http://eprint.iacr.org/2004/172.pdf.
[24] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *SIAM J. Comput.*, **32** (2003), pp. 586-615.
[25] D. Boneh and B. Waters, Constrained pseudorandom functions and their applications, *Advances in Cryptology — Asiacrypt 2013*, LNCS 8270, Springer-Verlag, 2013, pp. 280-300.
[26] J. Bos, C. Costello, M. Naehrig, and D. Stebila, Post-quantum key exchange for the TLS protocol from the ring learning with errors problem, *Proc. 2015 IEEE Symposium on Security and Privacy*, pp. 553-570.
[27] S. Chatterjee, A. Menezes, and P. Sarkar, Another look at tightness, *Proc. SAC 2011*, LNCS 7118, Springer-Verlag, 2011, pp. 293-319.
[28] L. Chen, Recommendation for key derivation using pseudorandom functions (revised), NIST SP 800-108, 2009.
[29] L. Chen, Recommendation for key derivation through extraction-then-expansion, NIST SP 800-56C, 2011.
[30] D. Coppersmith, Fast evaluation of logarithms in fields of characteristic two, *IEEE Transactions on Information Theory*, **30** (1984), pp. 587-594.
[31] R. Cramer and V. Shoup, A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack, *Advances in Cryptology — Crypto '98*, LNCS 1462, Springer-Verlag, 1998, pp. 13-25.
[32] R. Cramer and V. Shoup. Design and analysis of practical public-key encryption schemes secure against adaptive chosen ciphertext attack, *SIAM Journal on Computing*, **33** (2003), pp. 167-226.
[33] Q. Dang, Recommendation for applications using approved hash algorithms, NIST SP 800-107, 2012.
[34] T. Dierks and C. Allen, The TLS protocol, Internet RFC 2246, 1999.
[35] G. Fuchsbauer, Constrained verifiable random functions, *SCN 2014*, LNCS 8462, Springer-Verlag, 2014, pp. 95-114.
[36] E. Fujisaki and T. Okamoto, Secure integration of asymmetric and symmetric encryption schemes, *Advances in Cryptology — CRYPTO 1999*, LNCS 1666, Springer-Verlag, 1999, pp. 537-554.
[37] S. Galbraith, J. Malone-Lee, and N. Smart, Public key signatures in the multi-user setting, *Inf. Process. Lett.*, **83** (2002), pp. 263-266.
[38] D. Galindo, Boneh-Franklin identity-based encryption revisited, *ICALP 2005*, LNCS 3580, Springer-Verlag, 2005, pp. 791-802.
[39] S. Garg, C. Gentry, S. Halevi, M. Raykova, A. Sahai, and B. Waters, Candidate indistinguishability obfuscation and functional encryption for all circuits, http://eprint.iacr.org/2013/451.pdf.
[40] P. Gaži, K. Pietrzak, and M. Ribár, The exact PRF-security of NMAC and HMAC, *Advances in Cryptology — Crypto 2014*, LNCS 8617, Springer-Verlag, 2014, pp. 113-130.
[41] O. Goldreich and S. Goldwasser, On the limits of nonapproximability of lattice problems, *Journal of Computer and System Sciences*, **60** (2000), pp. 540-563.
[42] S. Goldwasser and M. Bellare, *Lecture Notes on Cryptography*, July 2008, available at http://cseweb.ucsd.edu/mihir/papers/gb.pdf.
[43] S. Goldwasser and Y. Kalai, Cryptographic assumptions: A position paper, available at http://eprint.iacr.org/2015/907.pdf.
[44] S. Goldwasser and S. Micali, Probabilistic encryption, *J. Computer and System Science*, **28** (1984), pp. 270-299.

[45] D. Harkins and D. Carrel, The internet key exchange (IKE), Internet RFC 2409, 1998.
[46] J. Hoffstein, N. Howgrave-Graham, J. Pipher, and W. Whyte, Practical lattice-based cryptography: NTRUEncrypt and NTRUSign, in *The LLL Algorithm*, Springer-Verlag, 2010, pp. 349-390.
[47] J. Hoffstein, J. Pipher, and J. Silverman, NTRU: a ring-based public key cryptosystem, *Algorithm Number Theory*, LNCS 1423, Springer-Verlag, 1998, pp. 267-288.
[48] IEEE 1363.1, Standard Specification for Public Key Cryptographic Techniques Based on Hard Problems over Lattices, 2008.
[49] J. Katz and Y. Lindell, *Introduction to Modern Cryptography*, Chapman and Hall/CRC, 2007.
[50] E. Kiltz, D. Masny, and J. Pan, Optimal security proofs for signatures from identification schemes, *Advances in Cryptology — Crypto 2016*, LNCS 9815, Springer-Verlag, 2016, pp. 33-61.
[51] T. Kim and R. Barbulescu, Extended tower number field sieve: A new complexity for medium prime case, *Advances in Cryptology — Crypto 2016*, LNCS 9814, Springer-Verlag, 2016, pp. 543-571.
[52] M. Kim and K. Lauter, Private genome analysis through homomorphic encryption, available at http://eprint.iacr.org/2015/965.pdf.
[53] N. Koblitz, The uneasy relationship between mathematics and cryptography, *Notices Amer. Math. Soc.*, **54** (2007), pp. 972-979.
[54] N. Koblitz, S. Krantz, and F. Verhulst, Hype! An exchange of views, *The Mathematical Intelligencer*, **36** (2014), No. 3, pp. 8-13.
[55] N. Koblitz and A. Menezes, Another look at 'provable security.' II, *Progress in Cryptology — Indocrypt 2006* LNCS 4329, Springer-Verlag, 2006, pp. 148-175.
[56] N. Koblitz and A. Menezes, Another look at 'provable security,' *J. Cryptology*, **20** (2007), pp. 3-37.
[57] N. Koblitz and A. Menezes, Another look at HMAC, *J. Mathematical Cryptology*, **7** (2013), pp. 225-251.
[58] N. Koblitz and A. Menezes, Another look at non-uniformity, *Groups Complexity Cryptology*, **5** (2013), pp. 117-139.
[59] N. Koblitz and A. Menezes, Another look at security definitions, *Advances in Mathematics of Communications*, **7** (2013), pp. 1-38.
[60] N. Koblitz and A. Menezes, Another look at security theorems for 1-key nested MACs, *Open Problems in Mathematics and Computational Science*, Springer-Verlag, 2014, pp. 69-89.
[61] N. Koblitz and A. Menezes, A riddle wrapped in an enigma, available at http://eprint.iacr.org/2015/1018.
[62] H. Krawczyk and P. Eronen, HMAC-based Extract-and-Expand Key Derivation Function (HKDF), Internet RFC 5869, 2010.
[63] H. Krawczyk, Cryptographic extraction and key derivation: The HKDF scheme, *Advances in Cryptology — Crypto 2010*, LNCS 6223, Springer-Verlag, 2010, pp. 631-648.
[64] T. Laarhoven, M. Mosca, and J. van de Pol, Finding shortest lattice vectors faster using quantum search, *Designs, Codes and Cryptography*, **77** (2015), pp. 375-400.
[65] K. Lauter, A. López-Alt, and M. Naehrig, *Proc. Latincrypt 2014*, LNCS 8895, Springer-Verlag, 2014, pp. 3-27.
[66] V. Lyubashevsky, D. Micciancio, C. Peikert, and A. Rosen, SWIFFT: A modest proposal for FFT hashing, *Proc. FSE 2008*, LNCS 5086, Springer-Verlag, 2008, pp. 54-72.
[67] V. Lyubashevsky, C. Peikert, and O. Regev, On ideal lattices and learning with errors over rings, *Journal of the ACM*, **60** (2013), pp. 43:1-43:35.
[68] A. Menezes, Another look at provable security, Invited talk at Eurocrypt 2012, available at http://www.cs.bris.ac.uk/eurocrypt2012/Program/Weds/Menezes.pdf.
[69] D. Micciancio and S. Goldwasser, *Complexity of Lattice Problems: A Cryptographic Perspective*, Springer, 2002.
[70] D. M'Raihi, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen, HOTP: An HMAC-based one time password algorithm, Internet RFC 4226, 2005.
[71] National Security Agency, *CRYPTOLOG*, 1st issue of 1994, available at http://tinyurl.com/eurocrypt1992.
[72] C. Peikert, Lattice cryptography for the internet, *PQCrypto 2014*, LNCS 8772, Springer-Verlag, 2014, pp. 197-219.
[73] C. Peikert, 19 February 2015 blog posting, http://web.eecs.umich.edu/~cpeikert/soliloquy.html

[74] C. Peikert, A decade of lattice cryptography, available at http://eprint.iacr.org/2015/939.
[75] K. Pietrzak, A closer look at HMAC, available at http://eprint.iacr.org/2013/212.pdf.
[76] O. Regev, On lattices, learning with errors, random linear codes, and cryptography, *Journal of the ACM*, **56** (2009), pp. 34:1-32:40.
[77] P. Rogaway, The moral character of cryptographic work, available at http://eprint.iacr.org/2015/1162.pdf.
[78] C.-P. Schnorr. Efficient identification and signatures for smart cards, *Advances in Cryptology — Crypto '89*, LNCS 435, Springer-Verlag, 1990, pp. 239-252.
[79] V. Shoup, Sequences of games: a tool for taming complexity in security proofs, available at http://eprint.iacr.org/2004/332.pdf.
[80] V. Shoup, ISO/IEC 18033-2:2006, Information Technology — Security Techniques — Encryption Algorithms — Part 2: Asymmetric Ciphers, 2006; final draft available at http://www.shoup.net/iso/std6.pdf.
[81] D. Stehlé and R. Steinfeld, Making NTRU as secure as worst-case problems over ideal lattices, *Advances in Cryptology — Eurocrypt 2011*, LNCS 6632, Springer-Verlag, 2011, pp. 27-47.
[82] J. Stern, D. Pointcheval, J. Malone-Lee, and N. Smart, Flaws in applying proof methodologies to signature schemes, *Advances in Cryptology — Crypto 2002*, LNCS 2442, Springer-Verlag, 2002, pp. 93-110.
[83] G. M. Zaverucha, Hybrid encryption in the multi-user setting, available at http://eprint.iacr.org/2012/159.pdf.
[84] R. Zhang and H. Imai, Improvements on security proofs of some identity based encryption schemes, *CISC 2005*, LNCS 3822, Springer-Verlag, 2005, pp. 28-41.

Appendix A. Concrete analysis of Regev's worst-case/average-case reduction

Let $q = q(n)$ and $m = m(n)$ be integers, and let $\alpha = \alpha(n) \in (0,1)$ be such that $\alpha q > 2\sqrt{n}$. Let χ be the probability distribution on $\mathbb{Z}_q$ obtained by sampling from a Gaussian distribution with mean 0 and variance $\alpha^2/2\pi$, and then multiplying by q and rounding to the closest integer modulo q. Then the (search version of the) LWE problem is the following: Let s be a secret vector selected uniformly at random from $\mathbb{Z}_q^n$. Given m samples $(a_i, a_i \cdot s + e_i)$, where each a_i is selected independently and uniformly at random from $\mathbb{Z}_q^n$, and where each e_i is selected independently from $\mathbb{Z}_q$ according to χ, determine s. The decisional version of LWE, called DLWE, asks us to determine whether we have been given m LWE samples $(a_i, a_i \cdot s + e_i)$ or m random samples (a_i, u_i), where each u_i is selected independently and uniformly at random from $\mathbb{Z}_q$.

Regev [76] proved that the existence of an efficient algorithm that solves DLWE in the average case implies the existence of an efficient quantum algorithm that solves SIVP_γ in the worst case where $\gamma = \tilde{O}(n/\alpha)$. In the remainder of this section we provide justification for the following refinement of Regev's theorem:

Claim 1. *Let $q = n^2$ and $\alpha = 1/(\sqrt{n}\log^2 n)$, whence $\gamma = \tilde{O}(n^{1.5})$. Suppose that there is an algorithm W that, given $m = n^c$ samples, solves DLWE for a fraction $1/n^{d_1}$ of all $s \in \mathbb{Z}_q^n$ with advantage at least $1/n^{d_2}$. Then there is a polynomial-time algorithm W' for solving $SIVP_\gamma$ that calls the W oracle a total of $O(n^{11+c+d_1+2d_2})$ times.*

A.1. **Gaussian distributions.** Recall that the *Gaussian distribution* with mean 0 and variance σ^2 is the distribution on $\mathbb{R}$ given by the probability density function

$$\frac{1}{\sqrt{2\pi}\cdot\sigma}\exp\left(\frac{-x^2}{2\sigma^2}\right).$$

For $x \in \mathbb{R}^n$ and $s > 0$, define the Gaussian function scaled by s:

$$\rho_s(x) = \exp\left(\frac{-\pi\|x\|^2}{s^2}\right).$$

The *Gaussian distribution D_s of parameter s* over $\mathbb{R}^n$ is given by the probability density function

$$D_s(x) = \frac{\rho_s(x)}{s^n}.$$

Note that D_s is indeed a probability distribution since $\int_{x\in\mathbb{R}^n}\rho_s(x)\,dx = s^n$.

If L is a lattice, we can define

$$\rho_s(L) = \sum_{x\in L}\rho_s(x).$$

Then the *discrete Gaussian probability distribution $D_{L,s}$* of width s for $x \in L$ is

$$D_{L,s}(x) = \frac{\rho_s(x)}{\rho_s(L)}.$$

Let L be a (full-rank integer) lattice of dimension n.

A.2. **Concrete analysis.** In this section the tightness gap of a reduction algorithm from problem A to problem B is the number of calls to the oracle for B that are made by the reduction algorithm.

Regev's worst-case/average-case reduction has two main components:

(1) The reduction of (search-)LWE to average-case DLWE (denoted DLWE_{ac}).
(2) The reduction of worst-case SIVP_γ to LWE.

A.2.1. *Reduction of LWE to DLWE_{ac}.* This reduction has three parts.

Part I. Worst-case to average-case. DLWE_{wc} denotes the worst-case DLWE problem. Lemma 4.1 in [76] shows that an algorithm W_1 that solves DLWE_{ac} for a fraction $\frac{1}{n^{d_1}}$ of all $s \in \mathbb{Z}_q^n$ with acceptance probabilities differing by at least $\frac{1}{n^{d_2}}$ can be used to construct an algorithm W_2 that solves DLWE_{wc} with probability essentially 1 for *all* $s \in \mathbb{Z}_q^n$. The algorithm W_2 invokes W_1 a total of $O(n^{d_1+2d_2+2})$ times.

Part II. Search to decision. Lemma 4.2 in [76] shows that an algorithm W_2 which solves DLWE_{wc} for all $s \in \mathbb{Z}_q^n$ with probability essentially 1 can be used to construct an algorithm W_3 that solves (search-)LWE for *all* $s \in \mathbb{Z}_q^n$ with probability essentially 1. Algorithm W_3 invokes W_2 a total of nq times, so this reduction has a tightness gap of nq.

Part III. Continuous to discrete. Lemma 4.3 in [76] shows that an algorithm W_3 that solves LWE can be used to construct an algorithm W_5 that solves $\text{LWE}_{q,\Psi_\alpha}$. (See [76] for the definition of the $\text{LWE}_{q,\Psi_\alpha}$ problem.) This reduction is tight.

A.2.2. *Reduction of SIVP$_\gamma$ to LWE.* This reduction has tightness gap $6n^{6+c}$. The reduction has two parts.

Part I. DGS to LWE. Let $\epsilon = \epsilon(n)$ be some negligible function of n. Theorem 3.1 of [76] shows that an algorithm W_4 that solves $\mathrm{LWE}_{q,\Psi_\alpha}$ given m samples can be used to construct a *quantum* algorithm W_9 for $DGS_{\sqrt{2n}\cdot\eta_\epsilon(L)/\alpha}$. Here, $\eta_\epsilon(L)$ is the "smoothing parameter with accuracy ϵ", and $\mathrm{DGS}_{r'}$ (discrete Gaussian sampling problem) is the problem of computing a sample from the discrete Gaussian probability distribution $D_{L,r'}$ where $r' \geq \sqrt{2n}\cdot\eta_\epsilon(L)/\alpha$.

Let $r = \sqrt{2n}\cdot\eta_\epsilon(L)/\alpha$. Let $r_i = r\cdot(\alpha q/\sqrt{n})^i$ for $i \in [0, 3n]$. Algorithm W_9 begins by producing n^c samples from $D_{L,r_{3n}}$ (Lemma 3.2 in [76]); the W_4 oracle is not used in this step. Next, by repeatedly applying the 'iterative step,' it uses the n^c samples from D_{L,r_i} to produce n^c samples from $D_{L,r_{i-1}}$ for $i = 3n, 3n-1, \ldots, 1$. Since $r_0 = r$, the last step produces the desired sample from $D_{L,r}$.

The iterative step (Lemma 3.3 in [76]) uses n^c samples from D_{L,r_i} to produce one sample from $D_{L,r_{i-1}}$; this step is then repeated to produce n^c samples from $D_{L,r_{i-1}}$. Thus, the iterative step is executed a total of $3n\cdot n^c = 3n^{1+c}$ times.

Each iterative step has two parts.

(1) The first part invokes W_4 a total of n^2 times:
   - Lemma 3.7 in [76] uses W_4 to construct an algorithm W_5 that solves $\mathrm{LWE}_{q,\Psi_\beta}$; W_5 invokes W_4 n times.
   - Lemma 3.11 in [76] uses W_5 and the n^c samples from D_{L,r_i} to construct an algorithm W_6 that solves the $\mathrm{CVP}^{(q)}_{L^*,\alpha q/\sqrt{2}r_i}$ problem. The reduction is tight.
   - Lemma 3.5 in [76] uses W_6 to construct an algorithm W_7 that solves the $\mathrm{CVP}_{L^*,\alpha q/\sqrt{2}r_i}$ problem. Algorithm W_7 invokes W_6 n times.

(2) The second part (Lemma 3.14 in [76]) uses W_7 to construct a *quantum* algorithm W_8 that produces a sample from $D_{L,r_{i-1}}$. This reduction is tight.

Since each iterative step has tightness gap n^2, the total tightness gap for the reduction of DGS to LWE is $3n^{3+c}$.

Part II. SIVP$_\gamma$ to DGS. Lemma 3.17 in [76] uses W_9 to construct an algorithm W_{10} that solves $\mathrm{SIVP}_{2\sqrt{2}n\eta_\epsilon(L)/\alpha}$. Algorithm W_{10} invokes W_9 $2n^3$ times.

Lemma 2.12 in [76] states that $\eta_\epsilon(L) \leq \sqrt{\omega(\log n)}\cdot\lambda_n(L)$ for some negligible function $\epsilon(n)$. Thus

$$\gamma = \frac{2\sqrt{2}n\eta_\epsilon(L)}{\alpha\cdot\lambda_n(L)} = \frac{2\sqrt{2}n\sqrt{\omega(\log n)}}{\alpha} = \tilde{O}\left(\frac{n}{\alpha}\right) = \tilde{O}(n^{1.5}).$$

A.2.3. *Summary.* Regev's reduction of SIVP_γ to DLWE_{ac} has tightness gap

$$n^{d_1+2d_2+2}\cdot nq\cdot 3n^{3+c}\cdot 2n^3 = 6n^{11+c+d_1+2d_2}.$$

Appendix B. Nontightness and Multi-User Attacks

In an important paper that has been all but ignored by the cryptographic research community, Zaverucha [83] showed that "provably secure" hybrid encryption, as described in several standards, is insecure in the multi-user setting if certain permitted (and even recommended) choices are made in the implementation. Because this work should be much better

known than it is, we shall devote this section to explaining and summarizing [83]. We shall focus on hybrid encryption schemes in the comprehensive ISO/IEC 18033-2 standard [80].

We first recall the definition in [16] of IND-CCA security (Indistinguishability under Chosen-Ciphertext Attack) of encryption in the multi-user setting. Suppose there are n users. The adversary is given n public keys, a decryption oracle for each public key, and an LR (left-or-right encryption) oracle for each public key. The adversary can query each decryption oracle up to q_D times and each LR oracle up to q_{LR} times. A decryption query simply asks for a chosen ciphertext to be decrypted under the corresponding public key. An LR query works differently. The n LR-oracles all have a hidden random bit b in common. The adversary chooses two equal-length messages M_0 and M_1 to query to one of the LR-oracles, which then returns an encryption C^* of M_b. The adversary is not permitted to query C^* to the decryption oracle for the same public key. The adversary's task is to guess b with success probability significantly greater than $1/2$.

Remark 12. This "multi-challenge" security model (that is, $q_{\mathrm{LR}} > 1$) can also be used in the single-user setting, but almost never is ([22] is a rare exception); in the standard IND-CCA security model $q_{\mathrm{LR}} = 1$. We shall later give a simple attack that shows that the standard IND-CCA is deficient and should be replaced by the multi-challenge model.

Remark 13. In [16] the authors give a generic reduction with tightness gap $n \cdot q_{\mathrm{LR}}$ between the multi-user and single-user settings. In the full version of [16] they also give a construction that shows that this tightness bound is optimal; that is, they describe a protocol that can be attacked with $n \cdot q_{\mathrm{LR}}$ times the advantage in the multi-user setting than in the single-challenge single-user setting. Their construction is contrived and impractical; later we shall describe a simple attack on hybrid encryption that shows that in practice as well as in theory the generic tightness bound in [16] is best possible. That is, the attack described below reduces security by a factor equal to n times the number of messages sent to each user (see Remark 16). (In specific cases tighter reductions are sometimes possible — for example, the paper [16] contains a reduction with tightness gap q_{LR} in the case of the Cramer–Shoup public-key encryption scheme [31].)

We now recall the setup and terminology of hybrid encryption. The encryption has two stages: a key-encapsulation mechanism (KEM) using a public-key cryptosystem (with the recipient's public/secret key pair denoted PK/SK), and a data-encapsulation mechanism (DEM) using a symmetric-key cryptosystem that encrypts the data by means of the shared key K that is produced by the KEM. The KEM takes PK as input and produces both the key material K by means of a key-derivation function (KDF) and also a ciphertext C_1 that will enable the recipient to compute K; the DEM takes K and the message M as input and produces a ciphertext C_2. The recipient decrypts by first using C_1 and SK to find K and then using C_2 and the symmetric key K to find M.

Among the public-key systems commonly used for KEM are Cramer-Shoup [31] and ECIES (ElGamal encryption using elliptic curves, see [80]); symmetric-key systems commonly used for DEM are AES in cipher block chaining (CBC) mode and XOR-Encrypt using a hash function with a counter. (We will describe this in more detail shortly.) The KDF is a publicly known way to produce key material of a desired length L from a shared secret that's computed using the public-key system.

Suppose, following [80], that we use 128-bit AES in CBC-mode with zero initialization vector for DEM. Let MAC denote a message authentication code that depends on a 128-bit key. Our KDF produces two 128-bit keys $K = (k_1, k_2)$. To send a $128m$-bit message M, we set C_2 equal to a pair (C', t), where C' is the $128m$-bit ciphertext computed below and $t = \mathrm{MAC}_{k_2}(C')$ is its tag. The ciphertext $C' = (C'_1, \ldots, C'_m)$ is given by: $C'_1 = \mathrm{AES}_{k_1}(M_1)$, $C'_i = \mathrm{AES}_{k_1}(C'_{i-1} \oplus M_i)$ for $i = 2, \ldots, m$.

After receiving $(C_1, C_2) = (C_1, C', t)$, the recipient first uses C_1, SK, and the KDF to find (k_1, k_2), and then uses the shared key k_2 to verify that t is in fact the tag of C'; otherwise she rejects the message. Then she decrypts using k_1.

Alternatively, for DEM we could use XOR-Encrypt with a hash function $\mathcal{H}$ as follows. To send a message M consisting of m 256-bit blocks, we have the KDF generate a $256m$-bit key $k_1 = (k_{1,1}, \ldots, k_{1,m})$ by setting $k_{1,i} = \mathcal{H}(z_0 \| i)$, where z_0 is a shared secret produced by KEM, and also a MAC-ing key k_2. The MAC works as before, but now C' is determined by setting $C'_i = M_i \oplus k_{1,i}$. This is the hash function with counter (CTR) mode mentioned above.

In [32] Cramer and Shoup gave a tight proof that hybrid encryption has IND-CCA security under quite weak assumptions. The MAC-scheme need only be "one-time secure" (because it receives a new key k_2 for each message), and the symmetric encryption function need only be one-time secure against passive adversaries — in particular, there is no need for randomization (again the reason is that it gets a new key k_1 for each message). In accordance with the general principle that standards should not require extra features that are not needed in the security reductions, the standards for hybrid encryption [80] do not require randomization in the symmetric encryption; nor do they impose very stringent conditions on the KDF. In addition, in [80] Shoup comments that if KEM is implemented using the Cramer–Shoup construction [31], which has a security proof without random oracles, and if DEM is implemented using AES-CBC, then it is possible to prove a tight security reduction for the hybrid encryption scheme without the random oracle assumption. Thus, anyone who mistrusts random oracle proofs should use AES-CBC rather than XOR-Encrypt. All of these security proofs are given in the single-user setting.

B.1. **Attacks in the multi-user setting.** We now describe some of the attacks of Zaverucha [83] in the multi-user setting, which of course is the most common setting in practice. Let $n = 2^a$ be the number of users. First suppose that the DEM is implemented using AES128 in CBC-mode. Suppose that Bob sends all of the users messages that all have the same first two blocks (M_1, M_2) (that is, they start with the same 256-bit header). The rest of the message blocks may be the same (i.e., broadcast encryption), or they may be different. The adversary Cynthia's goal is to read at least one of the 2^a messages. She guesses a key k that she hopes is the k_1-key for one of the messages. She computes $C''_1 = \mathrm{AES}_k(M_1)$ and $C''_2 = \mathrm{AES}_k(C''_1 \oplus M_2)$ and compares the pair (C''_1, C''_2) with the first two blocks of ciphertext sent to the different users.[12] If there's a match, then it is almost certain that she has guessed the key $k_1 = k$ for the corresponding message. That is because there are 2^{128}

[12]We can suppose that the 2^a ciphertexts are sorted according to their first two blocks (or perhaps stored using a conventional hash function). Then one iteration of the attack takes essentially unit time, since it just requires computing (C''_1, C''_2) and looking for it in the sorted table. Since the expected number of iterations is 2^{128-a}, the running time of the attack is $T = 2^{128-a}$ (and the success probability is essentially 1).

possible keys k_1 and 2^{256} possible pairs (C_1', C_2'), so it is highly unlikely that distinct keys would give the same (C_1', C_2'). Once Cynthia knows k_1 — each guess has a $2^{-(128-a)}$ chance of producing a match — she can quickly compute the rest of the plaintext. This means that even though the hybrid encryption scheme might have a tight security reduction in the single-user setting that proves 128 bits of security, in the multi-user setting it has only $128 - a$ bits of security. Commenting on how dropping randomization in DEM made his attack possible, Zaverucha [83] calls this "an example of a provable security analysis leading to decreased practical security."

Remark 14. In modern cryptography — ever since the seminal Goldwasser-Micali paper [44] — it has been assumed that encryption must always be probabilistic. In [80] this principle is violated in the interest of greater efficiency because the security proof in [32] does not require randomization. This decision was bold, but also rash, as Zaverucha's attack shows.

Remark 15. A time–memory–data tradeoff can be applied to speed up the on-line portion of the attack; see Remark 7 in [27]. Namely, at a cost of precomputation time 2^{128-a} and storage size $2^{2(128-a)/3}$, the secret key k of one of the 2^a users can be determined in time $2^{2(128-a)/3}$.

Remark 16. The above attack can also be carried out in the single-user setting if we suppose that Bob is sending Alice $2^{a'}$ different messages that all have the same header (M_1, M_2). Since different keys are generated for different messages (even to the same user), there is no need for the recipients of the messages to be different. This gives a reduction of the number of bits of security by a'. This attack shows the need for the multi-challenge security model even in the single-user setting. Thus, even in the single-user setting the standard security model for encryption is deficient because it fails to account for the very realistic possibility that Bob uses hybrid encryption as standardized in [80] to send Alice many messages that have the same header.

Remark 17. Note that if the $2^{a'}$ messages are broadcast to 2^a users, then obviously the reduction in security is by $a' + a$ bits. In some circumstances $a' + a$ could be large enough to reduce the security well below acceptable levels. For example, if $a' + a > 32$, it follows that what was thought to have 128 bits of security now has fewer than 96, which, as remarked in §1, is not enough. It should be emphasized that the security is reduced because of actual practical attacks, not because of a tightness gap that could conceivably be removed if one finds a different proof.

We note that the above attack does not in general work if DEM is implemented using XOR-Encrypt. (Of course, someone who does not trust security proofs that use random oracles would not be using XOR-Encrypt, and so would be vulnerable.) But Zaverucha has a different attack on hybrid encryption with XOR-Encrypt that works for certain KDF constructions.

B.2. **Attacks on Extract-then-Expand with XOR-Encrypt.** The most commonly used KDF takes the shared secret z_0 produced in KEM and derives a key of the desired length by concatenating $\mathcal{H}(z_0 \| i)$ for $i = 1, \ldots$. However, at Crypto 2010, as Zaverucha [83] explains,

> Krawczyk argues that cryptographic applications should move to a single, well-studied, rigorously analyzed family of KDFs. To this end, he formally defines security for KDFs, presents a general construction that uses any keyed pseudorandom function (PRF), and proves the security of his construction in the new model. The approach espoused by the construction is called *extract-then-expand*. [...] The HKDF scheme is a concrete instantiation of this general construction when HMAC is used for both extraction and expansion.

The Extract-then-Expand key derivation mechanism was soon standardized [28, 29] and [62]. In particular, RFC 5869 describes HKDF, which instantiates the Extract-then-Expand mechanism with HMAC, and states that HKDF is intended for use in a variety of KDF applications including hybrid encryption.

Extract-then-Expand works in hybrid encryption as follows. Suppose that z_0 is the shared secret produced in KEM. The Extract phase produces a bitstring $z_1 = \text{Extract}(z_0)$, perhaps of only 128 bits, which is much shorter than z_0. (The Extract phase may also depend on a "salt," but this is optional, and we shall omit it.) Then the key material K is obtained by a function that expands z_1, i.e., $K = \text{Expand}(z_1, L)$, where L as before is the bitlength of K. (There is also the option of putting some contextual information inside the Expand-function, but we shall not do this.)

We now describe Zaverucha's attack on hybrid encryption when Extract-then-Expand with 128-bit z_1 values is used as the KDF and XOR-Encrypt is used for message encryption. Suppose that Bob sends messages to 2^a users that all have the same header (M_1, M_2) and the same bitlength L. Cynthia's goal is to recover at least one of the plaintexts. Rather than guessing a key, she now guesses the bitstring z_1. For each guess she computes $K = \text{Expand}(z_1, L)$ and $C_i'' = M_i \oplus k_{1,i}$, $i = 1, 2$. When she gets a match with (C_1', C_2') for one of the users, she can then recover the rest of the plaintext sent to that user: $M_i = C_i' \oplus k_{1,i}$, $i > 2$.

Note that this attack does not work for XOR-Encrypt with the KDF using $\mathcal{H}(z_0 \| i)$ described above. Once again the "provably secure" choice of Extract-then-Expand turns out to be vulnerable, whereas the traditional choice of KDF is not. Zaverucha comments that "In this example, replacing a commonly used KDF in favor of a provably secure one causes a decrease in practical security."

As discussed in [83], Zaverucha's attacks can be avoided in practice by putting in features that are not required in the standard single-user single-challenge security proofs. It would be worthwhile to give proofs of this.

Open problem. Give a tight security reduction for hybrid encryption in the multi-user multi-challenge security model (random oracles are permitted) if DEM uses either: (1) randomized encryption rather than one-time-secure encryption (for example, AES-CBC with random IV that is different for each message and each recipient), (2) XOR-Encrypt using $\mathcal{H}(z_0 \| i)$ for the KDF, (3) XOR-Encrypt using HKDF with a recipient- and message-dependent salt in the Extract phase and/or recipient- and message-dependent contextual information in the Expand phase.

We conclude this section by noting a curious irony. As we remarked in §1, it is very rare for a standards body to pay much attention to tightness gaps in the security reductions that are used to support a proposed standard or to whether those security reductions were proved in the multi-user or single-user setting. However, recently the IETF decided that the standard for Schnorr signatures [78] should require that the public key be included in the hash function. The reason was that Bernstein [20] had found a flaw in the tight reduction from an adversary in the single-user setting to an adversary in the multi-user setting that had been given by Galbraith, Malone-Lee, and Smart [37], and he had proved that a tight security reduction could be restored if the public key is included in the hash function. (Later Kiltz, Masny, and Pan [50] gave a tight security reduction *without* needing to include the public key in the hash function; however, their assumptions are stronger than in [37], and it is not yet clear whether their result will cause the IETF to go back to dropping the public key from the hash input.)

The peculiar thing is that the tightness gap between single-user and multi-user settings is only a small part of the tightness problem for Schnorr signatures.

Lemma 5.7 in [50] gives a security proof in the random oracle model for the Schnorr signature scheme in the single-user setting. The proof has a tightness gap equal to the number of random oracle queries, which can be very large — in particular, much larger than the number of users in the multi-user setting. Even a tight single-user/multi-user equivalence leaves untouched the large tightness gap between Schnorr security and hardness of the underlying Discrete Log Problem. It should also be noted that the IETF was responding to the error Bernstein found in a proof, not to any actual attack that exploited the tightness gap (we now know that such an attack is probably impossible, because of the recent proof in [50] that under a certain reasonable assumption there is no single-user/multi-user tightness gap).

In the meantime, standards bodies have done nothing to address Zaverucha's critique of the standardized version [80] of hybrid encryption, which allows implementations that have far less security than previously thought, as shown by actual attacks.

Department of Computer Science and Automation, Indian Institute of Science, Bangalore, India
E-mail address: sanjit@csa.iisc.ernet.in

Department of Mathematics, Box 354350, University of Washington, Seattle, WA 98195 U.S.A.
E-mail address: koblitz@uw.edu

Department of Combinatorics & Optimization, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
E-mail address: ajmeneze@uwaterloo.ca

Applied Statistics Unit, Indian Statistical Institute, Kolkata, India
E-mail address: palash@isical.ac.in