The Brave New World of Bodacious Assumptions in Cryptography

Neal Koblitz and Alfred Menezes

There is a lot at stake in public-key cryptography. It is, after all, a crucial component in efforts to reduce identity theft, online fraud, and other forms of cybercrime. Traditionally, the security of a public-key system rests upon the assumed difficulty of a certain mathematical problem. Hence, newcomers to the field would logically expect that the problems that are used in security proofs come from a small set of extensively studied, natural problems. But they are in for an unpleasant surprise. What they encounter instead is a menagerie of ornate and bizarre mathematical problems whose presumed intractability is a basic assumption in the theorems about the security of many of the cryptographic protocols that have been proposed in the literature.

What Does Security Mean?

Suppose that someone is using public-key cryptography to encrypt credit card numbers during online purchases, sign a message digitally, or verify the route that a set of data followed in going from the source to her computer. How can she be sure that the system is secure? What type of evidence would convince her that a malicious adversary could not somehow compromise the security of the system?

At first glance it seems that this question has a straightforward answer. At the heart of any public-key cryptosystem is a *one-way function*—a function $y = f(x)$ that is easy to evaluate but for which it is computationally infeasible (one hopes) to find the inverse $x = f^{-1}(y)$. The two most important examples of such functions are the following:

- The Rivest-Shamir-Adleman (RSA) type of cryptography [28] is based on the assumed intractability of inverting the function $(p, q) \mapsto N = pq$, where (p, q) is a pair of randomly generated primes of roughly the same magnitude. The task of inverting this function is the famous integer factorization problem (the most difficult cases of which are believed to have the form $N = pq$ of an RSA modulus).
- Elliptic curve cryptography (ECC) is based on the assumed difficulty of inverting the function $x \mapsto xP$, where P is a point of large prime order p on an elliptic curve E defined over the field $\mathbb{F}_q$ of q elements and x is an integer mod p. The task of inverting this function is the Elliptic Curve Discrete Logarithm Problem (ECDLP).

Indeed, a large proportion of all of the mathematical research in public-key cryptography is concerned with algorithms for inverting the most important one-way functions. Hundreds of papers in mathematics as well as cryptography journals have been devoted to index calculus methods for factoring integers, to improved Pollard-ρ algorithms [33] and Weil descent methods [18] for finding discrete logarithms on elliptic curves, and to searches for weak parameters, i.e., RSA moduli N that are a little easier to factor than most, finite fields over which the ECDLP is slightly easier to solve, and so on. Traditionally, many mathematicians working in cryptography have tended to regard the question of security of a type of public-key system as equivalent to hardness of inverting the underlying one-way function.

Neal Koblitz is professor of mathematics at the University of Washington, Seattle. His email address is koblitz@math.washington.edu.

Alfred Menezes is professor of combinatorics and optimization at the University of Waterloo. His email address is ajmeneze@uwaterloo.ca.

However, this answer to the security question is woefully inadequate. In the first place, the implication goes only one way: if the underlying problem is efficiently solvable, then the system is insecure; but if it is intractable, the system is not necessarily secure. In other words, intractability is a necessary but not sufficient condition.

In the case of RSA, for example, the encryption function is exponentiation modulo N: $C = P^e \bmod N$, where e is a fixed integer prime to $\phi(N) = (p-1)(q-1)$, P is a block of plaintext (regarded as an integer less than N), and C is the scrambled text, called *ciphertext*. The decryption function $P = C^d \bmod N$ (where d is an inverse of the exponent e modulo $\phi(N)$) can be computed if one knows the factorization of N. But it has never been proved that knowledge of that factorization is *necessary* in order to decrypt. In fact, in a paper titled "Breaking RSA may not be equivalent to factoring" [12], Boneh and Venkatesan gave evidence that the above e-th root problem modulo N might be strictly easier than factoring N.

Moreover, there might be indirect ways to exploit the particular implementation of RSA that in certain cases would allow someone (Cynthia) other than the intended recipient (Alice) to learn the secret plaintext. For example,

• Suppose that Alice is receiving messages that have been encrypted using RSA; her public key is (N, e). Cynthia, after intercepting the ciphertext C that her competitor Bob sent to Alice, wants to know the plaintext P (let's say it was his bid on a job). If Cynthia asks Alice for P directly, Alice won't tell her Bob's bid, because it's against Alice's interests for Cynthia to know that. But suppose that awhile back, before Bob muscled in on her territory, Cynthia had extensive correspondence with Alice, and she now sends a message to Alice saying (falsely) that she lost one of her messages to Alice, she needs it for her records, and all she has is the ciphertext C'. Alice's computer willingly decrypts C' for Cynthia and sends her $P' = C'^d \bmod N$. But in reality Cynthia formed C' by choosing a random R and setting $C' = CR^e \bmod N$. After Alice is tricked into sending her P', all Cynthia has to do is divide it by R modulo N in order to learn P. This is called a *chosen-ciphertext attack.*

More precisely, in such an attack the adversary is assumed to be able to get Alice to decipher any ciphertext C' she wants other than the target ciphertext C. The system is said to have *chosen-ciphertext security* if knowledge of all those other plaintexts P' will not enable Cynthia to decrypt C.

In RSA the simplest way to prevent a chosen-ciphertext attack is to "pad" a message with a block of random bits before encryption (see, for example, [3]); then when Alice reveals only the subset of bits of P' that are in the message part of C'^d, Cynthia is stymied.

• Again suppose that Alice is receiving messages that have been encrypted using RSA. The plaintext messages have to adhere to a certain format, and if a decrypted message is not in that form, Alice's computer transmits an error message to the sender. This seems innocuous enough. However, Bleichenbacher [5] showed that the error messages sometimes might compromise security.

Bleichenbacher's idea can be illustrated if we consider a simplified version of the form of RSA that he attacked in [5]. Suppose that we are using RSA with a 1024-bit modulus N to send a 128-bit secret key m (for use in symmetric encryption). We decide to pad m by putting a random number r in front of it, but since this doesn't take up the full 1024 bits, we just fill in zero-bits to the left of r and m. When Alice receives our ciphertext, she decrypts it, checks that it has the right form with zero-bits at the left end—if not, she informs us that there was an error and asks us to resend—and then deletes the zero-bits and r to obtain m. In that case Bleichenbacher can break the system—in the sense of finding the plaintext message—by sending a series of carefully chosen ciphertexts (certain "perturbations" of the ciphertext he wants to decipher) and keeping a record of which ones are rejected because their e-th root modulo N is not of the proper form; that is, does not have the prescribed number of zero-bits.

Notice that the particular way that RSA is being used plays a crucial role. Thus, when discussing security, one must specify not only the type of cryptography and choice of parameters but also the instructions that will be followed. The sequence of steps the users of the system go through is called a *protocol.* A protocol description might take the form, "First Alice sends Bob the elements…; then Bob responds with…; then Alice answers with…; and so on."

Also notice that both of the above types of attacks can be avoided if a protocol is used that has chosen-ciphertext security, that is, if it can withstand a chosen-ciphertext attack. Ideally, what this means is that there is an efficient reduction from $\mathcal{P}$ to $\mathcal{Q}$, where $\mathcal{Q}$ is the problem of making a successful chosen-ciphertext attack and $\mathcal{P}$ is a mathematical problem (such as integer factorization) that is widely believed to be very difficult (provided that one chooses the parameters suitably). Such a reduction implies that $\mathcal{Q}$ is at least as hard as $\mathcal{P}$. What a "security proof"—or, as we prefer to say, a *reductionist security argument* [23]—does is show that an adversary cannot succeed in mounting a certain category of attack unless a certain underlying mathematical problem is tractable.

Rabin-Williams

In 1979 Rabin [27] proposed an encryption function that could be *proved* to be invertible only by someone who could factor N. His system is similar to RSA, except that the exponent is 2 rather than an integer e prime to $\varphi(N)$. For N a product of two primes the squaring map is 4-to-1 rather than 1-to-1 on the residues modulo N, so Rabin finds all four square roots of a ciphertext C (and in practice chooses the plaintext that makes sense to the message recipient).

Reductionist Security Claim. Someone who can find messages P from the ciphertext C must also be able to factor N.

Argument. Informally, the reason is that finding P means being able to find all four square roots of C, because any one of them could be the true plaintext P. Those square roots are $\pm P$ and $\pm \epsilon P$, where ϵ is a residue mod N that is $\equiv 1 \pmod{p}$ and $\equiv -1 \pmod{q}$. That means that someone who can find messages must know the value of ϵ, in which case N can be factored quickly using the Euclidean algorithm, since $\gcd(N, \epsilon - 1) = p$.

A more formal reduction would go as follows. We suppose that there exists an adversary that takes N and C as input and produces one of the square roots of C modulo N. We think of the adversary as a computer program, and we show how someone (Cynthia) who has that program could use it to quickly factor N.

What Cynthia does is the following. She chooses a random residue x, sets $C = x^2 \bmod N$, and inputs that value of C to the adversary. The adversary outputs a square root P of C mod N. With probability $1/2$ the root P is $\pm \epsilon x$, and in that case Cynthia can immediately compute $\epsilon = \pm x/P$ and then factor N. If, on the other hand, $P = \pm x$, then the value of P won't help her factor N, and she tries again, starting with a new value of x. There is only a $1/2^k$ chance that she will fail to factor N in k or fewer tries. We say that this argument reduces factoring N to breaking Rabin encryption mod N (where "breaking" means recovering plaintext messages). Rabin's scheme was the first public-key system to be proposed that was accompanied with a reductionist security argument. Users of Rabin encryption could be certain that no one could recover plaintexts unless they knew the factorization of N.

Soon after Rabin proposed his encryption scheme, Rivest pointed out that, ironically, the very feature that gives it an extra measure of security would also lead to total collapse if it were confronted with a chosen-ciphertext attacker. Namely, suppose that the adversary could somehow fool Alice into decrypting a ciphertext of its own choosing. The adversary could then follow the same procedure that Cynthia used in the previous paragraph to factor N. An adversary who could trick Alice into deciphering k chosen ciphertexts would have a $1 - 2^{-k}$ probability of factoring N.

However, at about the same time that Rivest made this observation, Williams [34] developed a variant in which the mapping is 1-to-1 that is especially useful for digital signatures. The resulting Rabin-Williams signature scheme appears to have significant efficiency and security advantages over traditional RSA. Recently, Bernstein [4] was able to show that even without random padding of messages, Rabin-Williams signatures are safe from chosen-message attack unless the adversary can factor N.[1] Unlike many proofs of security in the literature, Bernstein's paper is well written, logical, and lucid. In fact, after reading it, the obvious reaction is to ask: Why doesn't everyone switch to Rabin-Williams signatures?

In the real world, however, it is too late for that. Because of progress in factoring, for the highest security it is now recommended that 15360-bit N be used for any factorization-based cryptosystem. Meanwhile, the very highest security level with ECC requires q of 571 bits. Thus, users of RSA who are willing to change their software to accommodate a different system are going to switch to ECC, not to Rabin-Williams. If [4] had been published twenty years earlier, the history of digital signatures might have been very different.

The neglect of exponent 2 in RSA is a typical example of how historical happenstance and sociological factors, rather than intrinsic technical merit, can often determine what technology is widely used (see [22] for more discussion of this phenomenon).

The One-More-Discrete-Log Problem

Just as integer factorization is not exactly the problem one has to solve to invert the RSA encryption function, similarly, in systems using elliptic curves and other algebraic groups, the discrete log problem (DLP) is not the problem that is most immediately related to the task of the adversary Cynthia. Take, for example, the simplest ECC protocol, namely, the basic Diffie-Hellman key exchange [17] between two users, Alice and Bob. Let $\mathbb{G}$ be the group that is generated by a point $P \in E(\mathbb{F}_q)$ of prime order p. Suppose that Alice's public key is $Q_A = xP$ and her secret key is the integer x mod p; and Bob's public key is $Q_B = yP$ and his secret key is y. Then the shared key is simply xyP, which Alice computes as xQ_B and Bob as yQ_A.

[1] *Resistance to chosen-message attacks, in which the adversary can obtain signatures of messages of her choice and then has to sign a different message, is the commonly accepted standard of security of digital signatures; it is closely analogous to chosen-ciphertext security for encryption.*

The task of Cynthia, who knows P, xP, and yP, but neither of the secret keys, is to determine xyP from that triple of points. This is called the Diffie-Hellman Problem (DHP) in the group $\mathbb{G}$. Someone who can find discrete logs in $\mathbb{G}$ can obviously solve the DHP. The converse is a much more difficult question. However, in contrast to the situation with RSA, where there is doubt about the equivalence of the e-th root problem mod N and integer factorization, in ECC there is considerable evidence that the DHP and the DLP are of equivalent difficulty. The results showing this equivalence in many cases are surveyed in [25].

Because of the nature of chosen-ciphertext (or chosen-message) security and because many cryptographers want to have formal reduction arguments, they have had to greatly enlarge the types of mathematical problems that are used in their security analyses. Often the problems whose intractability is linked to the security of the protocols have lengthy, elaborate input or are interactive. In an interactive problem the solver is permitted to request additional information by making a bounded number of queries to an *oracle,* that is, a black box whose only function is to give correct answers to a certain type of question. On occasion, an interactive problem or one with input and output that appear unnatural might be used carefully and to good effect (see, for example, [15]). But in other cases the use of this type of problem raises more questions than it answers about the true security of the protocol.

Here are some examples of such problems that arose in connection with protocols that use elliptic curves or other algebraic groups:

- *The One-More-Discrete-Log Problem (1MDLP)* as first formulated in [1] and [2]. The solver is supplied with a challenge oracle that produces a random group element $Y_i \in \mathbb{G}$ when queried and a discrete log oracle. After ℓ queries to the challenge oracle (where ℓ is chosen by the solver) and at most $\ell - 1$ queries to the discrete log oracle, the solver must find the discrete logs of all ℓ elements Y_i.
- *The One-More-Diffie-Hellman Problem (1MDHP)* as first formulated (in a slightly different version) in [6]. The solver is given an element $X \in \mathbb{G}$, an oracle that can solve the Diffie-Hellman problem for the given X and arbitrary $Y \in \mathbb{G}$, and a challenge oracle that produces random group elements Y_i. After ℓ queries to the challenge oracle (where ℓ is chosen by the solver) and at most $\ell - 1$ queries to the Diffie-Hellman oracle, the solver must find all ℓ solutions $Z_i = xy_iP$ (where $X = xP$ and $Y_i = y_iP$).

At first it might seem that these problems should be equivalent in difficulty to the problem of finding the discrete log of a single random element or finding the Diffie-Hellman element Z for fixed X and a single random Y. However, it turns out that this depends very much on what groups are used. In [24] we studied these problems and several others in the setting of the jacobian group of a genus-g curve. Assuming that one uses current state-of-the-art algorithms, we found that 1MDLP is harder than 1MDHP for $g = 1, 2$, whereas it is strictly easier than 1MDHP for $g \geq 4$; the two problems are of roughly equal difficulty for $g = 3$; and it is only for nonhyperelliptic curves of genus 3 that the two problems are no easier than the DLP and DHP. Our conclusion is that it is often unclear how to gauge the true level of difficulty of an interactive problem or one with complicated input.

Reduction Theorems That Do Not Say Much

Suppose that the designers of a cryptographic protocol claim to have proved its security by constructing a reduction from $\mathcal{P}$ to $\mathcal{Q}$, where $\mathcal{Q}$ is the problem of mounting a successful attack (of a prescribed type) on the protocol and $\mathcal{P}$ is a mathematical problem that they believe to be intractable. Often a close examination of the two problems $\mathcal{P}$ and $\mathcal{Q}$ will show that they are trivially equivalent, in which case the theorem that supposedly establishes security is really assuming what one wants to prove. In that case the problem $\mathcal{P}$ has been tailored to make the proof work, and, in fact, the main difference between $\mathcal{P}$ and $\mathcal{Q}$ is simply that in the former the extraneous elements and cryptographic terminology have been removed.

For example, in most signature schemes the actual messages being signed are extraneous to an analysis of the scheme, because the first thing one does to a message is to compute its hash-function value (fingerprint), which is used instead of the message itself in all subsequent steps. If the security theorem is assuming that the hash-values are indistinguishable from random numbers—in which case one says that the proof is in the random-oracle model—then the set of messages can be replaced by a set of random numbers. If $\mathcal{P}$ has been constructed by removing this sort of irrelevant feature from $\mathcal{Q}$, then the equivalence of the two problems will be a tautology, and the reduction theorem does not provide any meaningful assurance that the protocol is secure.

Even if the reduction from $\mathcal{P}$ to $\mathcal{Q}$ is not trivial, one has to wonder about the value of the theorem whenever $\mathcal{P}$ is complicated and contrived. One should be especially skeptical if the protocol designers refer to $\mathcal{P}$ as a "standard" problem, because there is a long history of misleading uses of the word "standard" in cryptography. For example, a proof of security that uses weaker assumptions about the hash function than the random-oracle assumption (see above) is commonly said to be a proof in the standard model. The reader might not notice that, in order to work in the

standard rather than the random-oracle model, the authors had to invent a new nonstandard problem.

There is another questionable use of the word "standard" that is frequently encountered in the literature. After a complicated interactive problem $\mathcal{P}$ has been used in a couple of papers, subsequent papers refer to it as a standard problem. The casual reader is likely to think that something that is standard has withstood the test of time and that there's a consensus among researchers that the assumption or problem is a reasonable one to rely upon—although neither conclusion is warranted in such cases. The terminology obfuscates the fact that the new problem is highly nonstandard.

Pairing-Based Cryptography

Starting in 2001, pairing-based cryptosystems were proposed by Dan Boneh, Matt Franklin, and others. Although some of the ideas had been around for a couple of years (see, for example, [21, 29]), their tremendous potential had not been realized before.

The basic idea is that the Weil or Tate pairing on elliptic curves allows certain cryptographic goals to be achieved that no one knows how to achieve with conventional techniques. In some other cases, pairings give more efficient or conceptually simpler solutions.

Let

$$e: \mathbb{G} \times \mathbb{G} \longrightarrow \mu_p \subset \mathbb{F}_{q^k}$$

be a nondegenerate bilinear pairing on the group $\mathbb{G} \subset E(\mathbb{F}_q)$ generated by a point P of prime order p with values in the p-th roots of unity of the degree-k extension of $\mathbb{F}_q$, where k (called the *embedding degree*) is the smallest positive integer such that $p|q^k - 1$. The feasibility of computing pairings depends on how big k is. For example, if $\mathbb{F}_q$ is a prime field and E has $q + 1$ points (such a curve is called *supersingular*), then since $p|q + 1$ and $q + 1|q^2 - 1$, the embedding degree is $k = 2$, and pairings can be computed quickly.

One of the first uses of pairing-based cryptography was the elegant solution by Boneh and Franklin [10] to an old question of Shamir [30], who had asked whether an efficient encryption scheme could be devised in which a user's public key would be just her identity (e.g., her email address). Such a system is called *identity-based encryption*. Another early application (see [11]) was to obtain short signatures.

By the time pairing-based cryptography arose, it had become *de rigueur* when proposing a cryptographic protocol always to give a "proof of security", that is, a reduction from a supposedly intractable mathematical problem $\mathcal{P}$ to a successful attack (of a specified type) on the protocol. A peculiar feature of many pairing-based cryptosystems is that $\mathcal{P}$ has often been very contrived—the sort of problem that hardly any mathematician would recognize as natural, let alone want to study. Nevertheless, it has become customary to regard a conditional result of the form "if $\mathcal{P}$ is hard, then my protocol is safe from chosen-ciphertext attacks" as a type of guarantee of security.

The Strong Diffie-Hellman Problem

In [8, 9], Boneh and Boyen proposed a new digital signature that works as follows. As before, let $\mathbb{G}$ be the group generated by a point $P \in E(\mathbb{F}_q)$ of prime order p, and let $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mu_p$ be a nondegenerate bilinear pairing with values in the p-th roots of unity in a (not too big) field extension of $\mathbb{F}_q$.

In the Boneh-Boyen protocol, to sign a message m, which is regarded as an integer mod p, Alice uses her secret key (x, y), which is a pair of integers mod p. Her public key, which the recipient (Bob) will use to verify her signature, consists of the two points $X = xP$ and $Y = yP$. Alice picks a random r mod p and sets $Q = (x + yr + m)^{-1}P$ (where the reciprocal of $x + yr + m$ is computed mod p). Her signature consists of the pair (Q, r).

After receiving m and (Q, r), Bob verifies her signature by checking that

$$e(Q, X + rY + mP) = e(P, P);$$

if equality holds, as it should because of the bilinearity of e, he is confident that Alice was truly the signer — that is, only someone who knows the discrete logs of X and Y could have computed the point Q that makes the above equality hold.

Boneh and Boyen give a reductionist security argument that basically shows that a chosen-message attacker cannot forge a signature provided that the following Strong Diffie-Hellman (SDH) problem is hard. This problem is parameterized by an integer ℓ (which is a bound on the number of signature queries the attacker is allowed to make) and is denoted ℓ-SDH:

- The ℓ-SDH problem in the group $\mathbb{G} \subset E(\mathbb{F}_q)$ generated by a point P of prime order p is the problem, given points $P, xP, x^2P, \ldots, x^\ell P$ (where x is an unknown integer mod p), of constructing a pair (c, H) such that $(x + c)H = P$ (where c is an integer mod p and $H \in \mathbb{G}$).

The difficulty of this problem can be shown to be less than or equal to that of the classical Diffie-Hellman problem (which requires the construction of xyP given P, xP, and yP). But the problem is an odd one—the "S" in SDH should really have stood for "strange"—that had never been studied before. It was because of nervousness about the ℓ-SDH assumption that the authors of [8] felt the need to give evidence that it really is hard. What they did was derive an exponential-time lower bound for the amount of time it takes to solve ℓ-SDH in the *generic group model*.

The notion of a generic group in cryptography was first formalized by Nechaev [26] and Shoup [31]. The generic group assumption essentially means that the group has no special properties that could

be exploited to help solve the problem. Rather, the only things that a solver can do with group elements are performing the group operation, checking whether two elements are equal, and (in the case of pairing-based cryptography) computing the pairing value for two elements. A lower bound on solving $\mathcal{P}$ in the generic group model means that, in order to solve $\mathcal{P}$ in a specific group such as $E(\mathbb{F}_q)$ in time less than that bound, one would have to somehow exploit special features of the elliptic curve. In [31] Shoup proved that neither the discrete log problem (DLP) nor the Diffie-Hellman problem (DHP) can be solved in fewer than $\sqrt{p}$ steps in a generic group of prime order p.

In §5 of [8] Boneh and Boyen proved that ℓ-SDH in a generic group with a pairing cannot be solved in fewer than (roughly) $\sqrt{p/\ell}$ operations.

Note that this lower bound $\sqrt{p/\ell}$ for the difficulty of ℓ-SDH is weaker by a factor of $\sqrt{\ell}$ than the lower bound $\sqrt{p}$ for the difficulty of the DLP or the DHP in the generic group model. At first it seemed that the factor $\sqrt{\ell}$ was an artifact of the proof and not a cause for concern and that the true difficulty of the ℓ-SDH problem was probably $\sqrt{p}$ as in the case of the DLP and DHP. However, at Eurocrypt 2006 Cheon [16], using the same attack that had been described earlier in a different setting by Brown and Gallant [14], showed that ℓ-SDH can be solved—and in fact the discrete logarithm x can be found—in $\sqrt{p/\ell_0}$ operations if $\ell_0 \le \ell$ divides $p-1$ and $\ell_0 < p^{1/3}$. Thus in some cases ℓ-SDH can be solved in $p^{1/3}$ operations. This means that, to get the same security guarantee (if one can call it that) that signatures based on the DHP have with group order of a certain bitlength, Boneh-Boyen signatures should use a group whose order has 50% greater bitlength. It should also be noted that, even though solving ℓ-SDH does not immediately imply the ability to forge Boneh-Boyen signatures, recently Jao and Yoshida [20] showed how, using the solution to ℓ-SDH in [16], one can forge signatures in roughly $p^{2/5}$ operations (with roughly $p^{1/5}$ signature queries) under certain conditions.

Some of the other supposedly intractable problems that arise in security reductions for pairing-based protocols are even more ornate and contrived than the ℓ-SDH. Several such problems, such as the following Hidden Strong Diffie-Hellman (HSDH), are listed in [13]:

• In ℓ-HSDH one is given $P, xP, yP \in \mathbb{G}$ and $\ell - 1$ triples

$$(w_j P, (x + w_j)^{-1} P, y w_j P), \qquad j = 1, \ldots, \ell - 1,$$

and is required to find one more triple of the form $(wP, (x+w)^{-1}P, ywP)$ that is distinct from any of the $\ell - 1$ triples in the problem's input.

When readers encounter the bewildering array of problems whose presumed difficulty is linked to the security of important cryptographic protocols, a common reaction is dismay. However, some people who work in pairing-based cryptography prefer to put a positive spin on the unusual assortment of intractability assumptions. In a paper presented at the Pairing 2008 conference [13], Boyen said:

> The newcomer to this particular branch of cryptography will therefore most likely be astonished by the sheer number, and sometimes creativity, of those assumptions. The contrast with the more traditional branches of algebraic cryptography is quite stark indeed… the much younger "Pairing" branch…is already teeming with dozens of plausible assumptions, whose distinctive features make them uniquely and narrowly suited to specific types of constructions and security reductions.
>
> Far from being a collective whim, this haphazard state of affair [sic] stems from the very power of the bilinear pairing…in comparison to the admittedly quite simpler algebraic structures of twentieth-century public-key cryptography…[T]he new "bilinear" groups offer a much richer palette of cryptographically useful trapdoors than their "unidimensional" counterparts.

Boyen eloquently expresses a youthful optimism about the advantages of twenty-first-century cryptography—with its "rich palette" of exotic intractability assumptions—over the "unidimensional" RSA and ECC that were invented in the 1970s and 1980s. However, some recent experiences with these "plausible assumptions" suggest a need to temper this exuberance.

In the next section we describe a particularly dramatic example of how things can go wrong.

Sequential Aggregate Signatures

In 2007, Boldyreva, Gentry, O'Neill, and Yum [7] constructed a new type of digital signature called an Ordered Multi-Signature (OMS). This means a single compact signature produced by several people acting in sequence. It has fixed length independent of the number of signers—even though the different signers may be attesting to different messages. The main application discussed in [7] is to secure routing of messages through a network.

The authors of [7] describe the advantages of their OMS. In the first place, it is identity-based, i.e., there are no public keys other than the signers' email addresses; this "permits savings on bandwidth and storage…Our OMS construction substantially improves computational efficiency and scalability over any existing scheme with suitable functionality." Moreover, the authors write,

> In contrast to the only prior scheme to provide this functionality, ours offers improved security that does not rely on synchronized clocks or a trusted first signer. We provide formal security definitions and support the proposed scheme with security proofs under appropriate computational assumptions.

That is, the OMS in [7] is not only more efficient but also has "improved security".

The construction in [7] used groups $\mathbb{G}$ with bilinear pairings, and the proof of security assumed that the following Modified Lysyanskaya-Rivest-Sahai-Wolf (M-LRSW) problem is intractable:

• Given a group $\mathbb{G}$ of prime order p, a nondegenerate bilinear pairing $e: \mathbb{G} \times \mathbb{G} \longrightarrow \mu_p$, fixed nonidentity elements $P, U, V \in \mathbb{G}$ that are known to the solver, and fixed exponents $a, b \bmod p$ with aP and bP but not a or b known to the solver, the M-LRSW problem assumes that the solver is given an oracle that, when queried with an integer $m \bmod p$, chooses a random $r \bmod p$ and gives the solver the triple (X, Y, Z) of elements of $\mathbb{G}$ such that

$$X = mrU + abP, \qquad Y = rV + abP, \qquad Z = rP.$$

The solver must then produce some m' not equal to any of the m that were queried and one more triple (X', Y', Z') such that for some integer x

$$X' = m'xU + abP, \qquad Y' = xV + abP, \qquad Z' = xP.$$

Just as Boneh and Boyen did in [8], the authors of [7] argue that this problem is truly hard by giving an exponential lower bound for the time needed to solve M-LRSW in a generic group. They emphasize that:

> This has become a standard way of building confidence in the hardness of computational problems in groups equipped with bilinear maps.

Just about a year after [7] appeared, Hwang, Lee, and Yung [19] made a startling discovery: the "provably secure" protocol in [7] can very easily be broken, and the supposedly intractable M-LRSW problem can very easily be solved! Here is the fast and simple solution to M-LRSW that they found. Choose any m_1, m_2, and m' that are distinct and nonzero modulo p. Choose β_1, β_2 to be solutions in $\mathbb{F}_p$ to the two relations $\beta_2 = 1 - \beta_1$ and

$$\frac{\beta_1}{m_1} + \frac{\beta_2}{m_2} = \frac{1}{m'}.$$

(The solutions are $\beta_i = \frac{m_i(m_{3-i} - m')}{m'(m_{3-i} - m_i)}$, $i = 1, 2$.) Then make two queries to the oracle with inputs m_1 and m_2; let (X_i, Y_i, Z_i), $i = 1, 2$, denote the oracle's responses, and let r_i, $i = 1, 2$, denote the random r used by the oracle to produce (X_i, Y_i, Z_i). One then easily checks that, for m' the triple

$$X' = m'\left((\beta_1/m_1)X_1 + (\beta_2/m_2)X_2\right),$$
$$Y' = \beta_1 Y_1 + \beta_2 Y_2, \; Z' = \beta_1 Z_1 + \beta_2 Z_2$$

(where the coefficients of the X_i are computed in $\mathbb{F}_p$) is a solution of M-LRSW (with $x = \beta_1 r_1 + \beta_2 r_2$). Notice that this algorithm is generic, i.e., it works in any group of order p.

But Theorem 5.1 of [7], which is proved in Appendix D of the full version of the paper, gives an exponential lower bound (essentially of order $\sqrt{p}$) for the time needed to solve M-LRSW. The above Huang-Lee-Yung algorithm shows that Theorem 5.1 is dramatically false.

Oops!

What went wrong? The 4-page single-spaced argument purporting to prove Theorem 5.1 is presented in a style that is distressingly common in the provable security literature, with cumbersome notation and turgid formalism that make it unreadable to nonspecialists (and even to some specialists). To a mathematician reader, Appendix D of [7] does not resemble what we would normally recognize as a proof of a theorem. If one tries to wade through it, one sees that the authors are essentially assuming that all an attacker can do is make queries of the oracle and some rudimentary hit-or-miss computations and wait for two group elements to coincide. They are forgetting that the exponent space is a publicly known prime field and that the attacker is free to do arithmetic in that field and even solve an equation or two.

Conclusion

What are the implications of all this confusion? Should we be worried about the true security of the protocols that are deployed in the real world? Should we cut up our credit cards and stop making online purchases?

No, that's not the conclusion to draw from these examples. In the first place, fallacies found in proofs of security do not necessarily lead to an actual breach. Rather, the flaw in the proof simply means that the advertised guarantee disappears. Similarly, even if we are bewildered and unimpressed by the mathematical problem whose intractability is being assumed in a security proof, we might still have confidence—based on other criteria besides the reductionist proof—that the protocol is secure.

In the second place, cryptographic protocols are not developed and marketed in the real world unless they have been approved by certain industrial-standards bodies. Most cryptosystems proposed in academic papers never get used commercially, and the ones that do have a long lag—sometimes decades—between the initial proposal and actual deployment. Protocols that are based on dubious

assumptions or fallacious proofs are not likely to survive this process.

In reality the mathematical sciences have only a limited role to play in evaluating the true security of a cryptographic protocol. Admittedly it is tempting to hype up the centrality of mathematics in cryptography and use cryptographic applications as a marketing tool for mathematics, saying things like: "Number theory can provide the foundation for information security in an electronic world." The first author pleads guilty to having made this statement to an audience of several thousand security specialists at the 2009 RSA Conference. In so doing he violated his own belief that scientists should show self-restraint and refrain from BS-ing[2] the public.

Perhaps the main lesson to learn from the unreliability of so many "proofs of security" of cryptosystems is that mathematicians (and computer scientists) should be a bit more modest about our role in determining whether or not a system can be relied upon. Such an evaluation needs to incorporate many other disciplines and involve people with hands-on experience and not just theoretical knowledge. A discussion of the nonmathematical side of this problem would be out of place in the *AMS Notices*. For the interested reader a good place to start would be the short article [32] by Brian Snow, the Technical Director of Research (now retired) at the U.S. National Security Agency.

References

[1] M. Bellare, C. Namprempre, D. Pointcheval, and M. Semanko, The one-more-RSA inversion problems and the security of Chaum's blind signature scheme, *J. Cryptology* **16** (2003), pp. 185–215.

[2] M. Bellare and A. Palacio, GQ and Schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks, *Advances in Cryptology—Crypto 2002*, LNCS 2442, Springer-Verlag, 2002, pp. 149–162.

[3] M. Bellare and P. Rogaway, Optimal asymmetric encryption—how to encrypt with RSA, *Advances in Cryptology—Eurocrypt '94*, LNCS 950, Springer-Verlag, 1994, pp. 92–111.

[4] D. Bernstein, Proving tight security for Rabin-Williams signatures, *Advances in Cryptology—Eurocrypt 2008*, LNCS 4965, Springer-Verlag, 2008, pp. 70–87.

[5] D. Bleichenbacher, A chosen ciphertext attack against protocols based on the RSA encryption standard PKCS #1, *Advances in Cryptology—Crypto '98*, LNCS 1462, Springer-Verlag, 1998, pp. 1–12.

[6] A. Boldyreva, Efficient threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme, *Proc. Public Key Cryptography 2003*, LNCS 2567, Springer-Verlag, 2003, pp. 31–46.

[7] A. Boldyreva, C. Gentry, A. O'Neill, and D. H. Yum, Ordered multisignatures and identity-based sequential aggregate signatures, with applications to secure routing, *Proc. 14th ACM Conference on Computer and Communications Security, CCS 2007*, ACM Press, 2007, pp. 276–285; full version available at `http://eprint.iacr.org/2007/438`.

[8] D. Boneh and X. Boyen, Short signatures without random oracles, *Advances in Cryptology—Eurocrypt 2004*, LNCS 3027, Springer-Verlag, 2004, pp. 56–73.

[9] ———, Short signatures without random oracles and the SDH assumption in bilinear groups, *J. Cryptology* **21** (2008), pp. 149–177.

[10] D. Boneh and M. Franklin, Identity-based encryption from the Weil pairing, *Advances in Cryptology—Crypto 2001*, LNCS 2139, Springer-Verlag, 2001, pp. 213–229; *SIAM J. Computing* **32** (4) (2003), pp. 586–615.

[11] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the Weil pairing, *J. Cryptology* **17** (2004), pp. 297–319.

[12] D. Boneh and R. Venkatesan, Breaking RSA may not be equivalent to factoring, *Advances in Cryptology—Eurocrypt '98*, LNCS 1233, Springer-Verlag, 1998, pp. 59–71.

[13] X. Boyen, The uber-assumption family: A unified complexity framework for bilinear groups, *Pairing-Based Cryptography—Pairing 2008*, LNCS 5209, Springer-Verlag, 2008, pp. 39–56.

[14] D. Brown and R. Gallant, The static Diffie-Hellman problem, available at `http://eprint.iacr.org/2004/306`.

[15] D. Cash, E. Kiltz, and V. Shoup, The twin Diffie-Hellman problem and applications, *Advances in Cryptology—Eurocrypt 2008*, LNCS 4965, Springer-Verlag, 2008, pp. 127–145.

[16] J. Cheon, Security analysis of the Strong Diffie-Hellman problem, *Advances in Cryptology—Eurocrypt 2006*, LNCS 4004, Springer-Verlag, 2006, pp. 1–11.

[17] W. Diffie and M. Hellman, New directions in cryptography, *IEEE Trans. Inf. Theory, IT-22*, 1976, pp. 644–654.

[18] F. Hess, Weil descent attacks, in *Advances in Elliptic Curve Cryptography*, ed. by I. Blake, G. Seroussi, and N. Smart, Cambridge University Press, 2005, pp. 151–182.

[19] J. Y. Hwang, D. H. Lee, and M. Yung, Universal forgery of the Identity-Based Sequential Aggregate Signature Scheme, *ACM Symposium on Information, Computer & Communication Security, ASIACCS 2009*.

[20] D. Jao and K. Yoshida, Boneh-Boyen signatures and the Strong Diffie-Hellman problem, *Pairing-Based Cryptography—Pairing 2009*, LNCS 5671, Springer-Verlag, 2009, pp. 1–16.

[21] A. Joux, A one round protocol for tripartite Diffie-Hellman, *Algorithmic Number Theory: Fourth International Symposium*, LNCS 1838, Springer-Verlag, 2000, pp. 385–393.

[22] A. H. Koblitz, N. Koblitz, and A. Menezes, Elliptic curve cryptography: The serpentine course of a paradigm shift, to appear in

[2] *With apologies to the* Notices *editor, who asked us to be sure to write out all acronyms.*

J. Number Theory, available at http://eprint.iacr.org/2008/390.

[23] N. KOBLITZ and A. MENEZES, Another look at "provable security", *J. Cryptology* **20** (2007), pp. 3–37.

[24] N. KOBLITZ and A. MENEZES, Another look at non-standard discrete log and Diffie-Hellman problems, *J. Math. Cryptology* **2** (2008), pp. 311–326.

[25] U. MAURER and S. WOLF, The Diffie-Hellman protocol, *Designs, Codes and Cryptography* **19** (2000), pp. 147–171.

[26] V. I. NECHAEV, Complexity of a deterministic algorithm for the discrete logarithm, *Mathematical Notes* **55** (2) (1994), pp. 165–172.

[27] M. RABIN, Digitalized signatures and public-key functions as intractable as factorization, MIT Lab. for Computer Science Technical Report LCS/TR-212, 1979.

[28] R. RIVEST, A. SHAMIR, and L. ADLEMAN, A method for obtaining digital signatures and public key cryptosystems, *Communications of the ACM* **21** (2) (1978), pp. 120–126.

[29] R. SAKAI, K. OHGISHI, and M. KASAHARA, Cryptosystems based on pairings, *Proc. 2000 Symposium on Cryptography and Information Security*, Okinawa, 2000.

[30] A. SHAMIR, Identity-based cryptosystems and signature schemes, *Advances in Cryptology—Crypto '84*, LNCS 196, Springer-Verlag, 1985, pp. 277–296.

[31] V. SHOUP, Lower bounds for discrete logarithms and related problems, *Advances in Cryptology—Eurocrypt '97*, LNCS 1233, Springer-Verlag, 1997, pp. 256–266.

[32] B. SNOW, We need assurance!, *Proc. 21st Annual Computer Security Applications Conference*, IEEE Computer Society, 2005, pp. 3–10.

[33] E. TESKE, Square-root algorithms for the discrete log problem (a survey), in *Public-Key Cryptography and Computational Number Theory*, Walter de Gruyter, 2001, pp. 283–301.

[34] H. WILLIAMS, A modification of the RSA public-key encryption procedure, *IEEE Trans. Inf. Theory, IT-26*, 1980, pp. 726–729.

KOÇ UNIVERSITY

DEAN • College of Sciences

Koç University invites applications and nominations for the position of the Dean of the College of Sciences for an appointment to be effective 1 June 2010. Koç University is a private, non-profit institution of higher education, founded in 1993. Koç University is on a state-of-the-art campus at Rumeli Feneri, overlooking the Black Sea, close to the city of Istanbul. The University is committed to the pursuit of excellence in both research and teaching. Its aim is to provide world class education and research opportunities to a high quality group of students by distinguished faculty. The medium of instruction is English. Currently, Koç University offers degrees in the Colleges of Administrative Sciences and Economics, Sciences, Social Sciences and Humanities, Engineering, and Law as well as the School of Nursing. The new School of Medicine will admit its first class in September, 2010. More detailed information about Koç University can be found at the web site: http://www.ku.edu.tr

The College of Sciences offers B.S. degree programs in its Departments of Chemistry, Physics, Mathematics and Molecular Biology and Genetics. In addition the Graduate School offers M.S. and Ph.D. degrees.

The new Dean is expected to have a distinguished research and publication record in one of the programs covered by the college or related fields. The candidate must provide leadership in teaching and research programs of the college, and foster relations with the academic community and business world. In addition the Dean is expected to have demonstrated ability for organizational and interpersonal skills.

The compensation package is competitive. All information on candidates will be kept confidential. Review of applications will start 1 February 2010 and continue until the position is filled. Candidates should submit their letter, a list of references, curriculum vitae and further inquiries to:

Professor Tekin Dereli
Department of Physics
Chair of Dean Search Committee
Koç University
Rumelifeneri Yolu, 34450 Sarıyer, Istanbul, Turkey
Phone: 90-212-338 1510 E-mail: tdereli@ku.edu.tr