INTRACTABLE PROBLEMS IN CRYPTOGRAPHY

NEAL KOBLITZ AND ALFRED MENEZES

Abstract. We examine several variants of the Diffie-Hellman and Discrete Log problems that are connected to the security of cryptographic protocols. We discuss the reductions that are known between them and the challenges in trying to assess the true level of difficulty of these problems, particularly if they are interactive or have complicated input.

1. Introduction

No one can give an unconditional proof of the security of a public-key cryptographic protocol. Indeed, anyone who succeeded in giving such a proof would have proved the P≠NP conjecture as a corollary.

Rather, the "provable security" of a protocol is always conditional upon some intractability assumption. The "proof" is then a *reduction* from the problem $\mathcal{P}$ whose intractability is being assumed to the problem $\mathcal{Q}$ of mounting a successful attack of a specified type on the protocol. The reduction shows that breaking the protocol in the specified sense must be at least as hard as solving $\mathcal{P}$.

In this paper we shall focus on problems $\mathcal{P}$ that arise in connection with discrete-log-based protocols. Let $\mathbb{G}$ be a group of prime order p with generator P. We shall use additive notation for the group operation; in many practical protocols $\mathbb{G}$ would be the group of multiples of a point P on an elliptic curve defined over a finite field. In this setting the two classical problems are:

- The Discrete Log (DL) problem for a group $\mathbb{G}$ of order p and generator P has as input a second point $Q \in \mathbb{G}$ and asks for the integer x modulo p such that $Q = xP$.
- The Diffie-Hellman (DH) problem for a group $\mathbb{G}$ of order p and generator P has as input two other points $Q, R \in \mathbb{G}$ and asks for the point $S \in \mathbb{G}$ such that $z \equiv xy \pmod{p}$, where x, y, z are the integers mod p such that $Q = xP$, $R = yP$, $S = zP$.

The DH problem is sometimes called the Computational Diffie-Hellman problem to distinguish it from the Decision Diffie-Hellman (DDH) problem, discussed in §3.

Date: May 14, 2010.

This paper is an updated version of our paper with the same title in the *Proceedings of the 9th International Conference on Finite Fields and Their Applications*. We incorporated recent observations by Granger in §7 of [17]. We also corrected the introduction to §12, which had confused the ordered multi-signature (OMS) and the identity-based sequential aggregate signature (IBSAS) schemes proposed in [5]. The same updates should be applied to the companion paper "The brave new world of bodacious assumptions in cryptography", *Notices of the AMS*, 57 (2010), 357-365.

It is the DL problem that has been most extensively studied and for which many algorithms have been developed. Subexponential-time algorithms have been found for important classes of groups, such as the multiplicative group of a finite field and the jacobian group of a high-genus curve. In other groups, such as the group of points of a suitably chosen elliptic curve, the best algorithms are variants of Pollard-rho, which have running time of order $\sqrt{p}$. Indeed, a result of Shoup [35] shows that in a generic group (see §11) there can be no faster-than-squareroot algorithm for either the DL or DH problem.

Obviously the ability to solve DL implies the ability to solve DH. But it is much more difficult to determine whether the converse implication holds. There is considerable evidence that it does; for a survey of related results see [30]. Thus, the DH as well as DL problems on a suitable group are generally regarded as classical intractable problems.

In §§2–5 we look at different versions of ElGamal encryption in order to highlight some features of security reductions and the mathematical problems that they use.

2. The Naive ElGamal Protocol

We first illustrate the idea of a security "proof" (that is, a reductionist argument) in a very simple setting. Consider the naive ElGamal encryption protocol, in which the message is a point $M \in \mathbb{G}$, Alice's private key is an integer x mod p, and her public key is $Q = xP$. To send her the message M, Bob randomly chooses an integer y mod p, computes $R = yP$ and $C = M + yQ$, and sends the ciphertext (R, C). Alice decrypts by computing $C - xR = M$.

It is clear that an adversary who knows how to solve either the DL or DH problem in the group $\mathbb{G}$ can break the protocol. Conversely, we have a security theorem that says that the ability to decrypt messages implies the ability to solve DH.

Proposition 1. *The DH problem reduces to the problem of decryption of naive ElGamal ciphertexts.*

Sketch of proof. The idea is that if, given any Q and ciphertext (R, C), you can compute M, that means that you can find the solution S to the DH problem by simply subtracting $S = C - M = xyP$.

More formally, we can construct a reduction as follows. Suppose that we have a decryption "oracle" for naive ElGamal. This means a black box that, given the public key and a ciphertext, will return the corresponding plaintext. We must show how, given an instance of DH, we can use the oracle to solve it. So suppose that we are given the points $Q, R \in \mathbb{G}$ and want to find the point S that solves the DH problem. We choose a random point $T \in \mathbb{G}$ and give the decryption oracle the public key Q and ciphertext (R, T). The oracle gives us the decryption M, and we merely have to set $S = T - M$. □

3. The Decision Diffie-Hellman Problem

A somewhat more demanding definition of security of an encryption scheme is to ask that, if m_1 and m_2 are plaintexts and c is the ciphertext coming from one of

them, the adversary not be able to determine which of the two has c as ciphertext. A simple argument similar to the proof of Proposition 1 above shows that naive ElGamal encryption is secure in this sense provided that the following Decision Diffie-Hellman (DDH) problem is hard.

- The DDH problem for a group $\mathbb{G}$ of order p and generator P has as input three points $Q, R, S \in \mathbb{G}$ and asks whether or not $z \equiv xy \pmod{p}$, where x, y, z are the integers mod p such that $Q = xP$, $R = yP$, $S = zP$.

The DDH problem obviously reduces to both the DL and DH problems, but it is unlikely that there is a reduction in the other direction. Indeed, there is an important class of groups in which the DDH is easy and the DH and DL problems are believed to be hard. These are the "Diffie-Hellman gap groups" (first described in [22, 10]) that are used in pairing-based cryptography (see §10).

Remark 1. For a fixed group $\mathbb{G}$ let DL(P), DH(P), DDH(P) denote the Discrete Log, Diffie-Hellman, and the Decision Diffie-Hellman problems for a given generator P. In order to be sure that the choice of P doesn't matter, for a different generator Q we'd like to have reductions between DL(P) and DL(Q), DH(P) and DH(Q), and DDH(P) and DDH(Q). The first two are easy to construct, but finding a reduction from DDH(P) to DDH(Q) is an open problem. Nevertheless, it is hard to believe that there could be a prime-order group in which the difficulty of DDH depends on the choice of generator. Thus, the absence of any known reduction from DDH(P) to DDH(Q) should probably be interpreted as an example of the limitations of reductions rather than as evidence that the two problems DDH(P) and DDH(Q) might actually differ in difficulty.

Even if we use a group in which DDH is intractable — that is, cannot feasibly be solved using current technology — naive ElGamal is not a good encryption protocol, because it fails to pass a more stringent test of security, namely, resistance to *chosen-ciphertext* attacks. We illustrate by describing a hypothetical scenario.

Suppose that Alice is receiving messages that have been encrypted using naive ElGamal; her private key is x and her public key is $Q = xP \in \mathbb{G}$. Cynthia, after intercepting the ciphertext (R, C) that her competitor Bob sent to Alice, wants to know the plaintext M (let's say it was his bid on a job). If Cynthia asks Alice for M directly, Alice won't tell her Bob's bid, because it's against Alice's interests for Cynthia to know that. But suppose that a while 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 (R, C'). Since $C' \neq C$, Alice's computer willingly decrypts this for Cynthia and sends her $M' = C' - xR$. But in reality Cynthia formed C' by choosing a random T and setting $C' = C + T$. After Alice is tricked into sending her M', all Cynthia has to do is to subtract T to get $M = M' - T$.

More generally, in a chosen-ciphertext attack the adversary is assumed to be able to get Alice to decipher any ciphertexts she wants other than the target ciphertext. The system is said to have chosen-ciphertext security if knowledge of all those other plaintexts will not enable Cynthia to find the one she wants.

Under certain conditions a chosen-ciphertext attack on naive ElGamal could have much more damaging consequences than simply revealing a plaintext. Namely, suppose that the only direct attacks known on the DL and DH problems on the group are the squareroot attacks and that the group order p was chosen large enough so that $p^{1/2}$-attacks are not feasible but $p^{1/3}$-attacks would be feasible. According to a striking result of Brown and Gallant [13], if $p-1$ has a factor of order $p^{1/3}$ and if roughly $p^{1/3}$ chosen-ciphertext queries are allowed, then Cynthia can learn Alice's private key x in time of order $p^{1/3}$ — after which Cynthia can, of course, decrypt all previous and future ciphertexts sent to Alice.

Brown and Gallant considered the following *One-sided Diffie-Hellman* problem.

- The One-sided Diffie-Hellman problem for a group $\mathbb{G}$ of order p with generator P and fixed second point Q has as input $R \in \mathbb{G}$ and asks for the point $S \in \mathbb{G}$ such that $z \equiv xy \pmod{p}$, where x, y, z are the integers mod p such that $Q = xP$, $R = yP$, $S = zP$.

This problem differs from the DH problem only because Q is fixed. It is easy to see that in a chosen-ciphertext attack on naive ElGamal a decryption query is equivalent to a call to a One-sided Diffie-Hellman oracle, that is, a black box that for fixed $(\mathbb{G}, p, P, Q)$ takes input $R \in \mathbb{G}$ and returns the solution S to One-sided Diffie-Hellman. The algorithm in [13] shows how to find the discrete log of Q in time roughly $p^{1/3}$ using $p^{1/3}$ calls to a One-sided Diffie-Hellman oracle (provided that $p-1$ has a factor of order $p^{1/3}$). We shall return to this result again in a different context in §11.

4. Chosen-Ciphertext Secure ElGamal

In this version of ElGamal one uses the technique of naive ElGamal to establish a key for use with a symmetric-key encryption scheme, which is then used for the actual encryption. Let E_k be such an encryption function — that is, each key k for each t determines a permutation of the set of strings of t bits. We assume that E_k and its inverse are easy to compute once k is known, but that it is impossible to encrypt a plaintext or decrypt a ciphertext without knowing k. We also suppose that we have a hash function H that takes input of arbitrary length and produces a value k that will serve as the key for the encryption.

As in naive ElGamal, Alice's private key is $x \bmod p$, and her public key is $Q = xP \in \mathbb{G}$. When Bob wants to send her a t-bit message m, he first chooses a random $y \bmod p$ and computes $R = yP$, $S = yQ$, and then $k = H(R, S)$. The ciphertext is the pair $(R, E_k(m))$. Alice decrypts by setting $S = xR$ and then finding the key $k = H(R, S)$.

In order to get a chosen-ciphertext security theorem for this version of ElGamal, one needs to introduce the following variant of the Computational Diffie-Hellman problem:

- The One-sided Gap-DH problem[1] has the same input and the same desired output as the DH problem, but the solver is also allowed to use a One-sided Decision Diffie-Hellman Oracle. In other words, in a group $\mathbb{G}$ of order p with generator P, given DH input (Q, R), the solver is supplied with an oracle that for any $R', S' \in \mathbb{G}$ correctly answers the question: Is $z' \equiv xy' \pmod{p}$, where x, y', z' are the integers mod p such that $Q = xP$, $R' = y'P$, $S' = z'P$?

It is easy to see that an adversary who knows how to solve the One-sided Gap-DH problem can defeat the chosen-ciphertext security of this version of ElGamal. The next proposition establishes that the converse holds *in the random oracle model*, that is, the hash function H is assumed to be modeled by an oracle that when asked for $H(R, S)$ gives a random value (with the only condition being that if the same query is made a second time, the same random value must be returned).

Proposition 2. *If One-sided Gap-DH is intractable, then the above version of ElGamal is chosen-ciphertext secure under the random oracle assumption for the hash function H.*

Sketch of proof. In a fixed group $\mathbb{G}$ with generator P we are given a DH input (Q, R), where $Q = xP$ and $R = yP$ for unknown x and y. We have a One-sided DDH oracle and a second oracle $\mathcal{A}$ (the adversary) with which we interact. We present the oracle $\mathcal{A}$ with the public key Q and a target ciphertext (R, c) to decipher. In the interaction we must simulate two oracles for $\mathcal{A}$ — a hash function oracle that returns a random value in response to a hash query and a decryption oracle that answers queries for any chosen-ciphertext $(R', c') \neq (R, c)$. The proof must show that we can interact with $\mathcal{A}$ in such a way as to obtain the solution $S = xyP$ of the DH problem.

We give the adversary $\mathcal{A}$ the public key Q, and we choose (R, c), where we pick c at random, as the target ciphertext. In response to any hash query from $\mathcal{A}$ we give a random value, keeping a record of the queries and our responses. Of course, if the same query is made twice, we give the same value in response. Whenever a query for $H(R', S')$ is made, we use our One-sided DDH oracle to determine whether or not (Q, R', S') is a Diffie-Hellman triple (that is, whether or not $S' = xR'$). If it is, then we note the value k' we give for $H(R', S')$, since it must be used to decrypt any queried ciphertext of the form (R', c').

Whenever $\mathcal{A}$ makes a ciphertext query for (R', c') we first check whether we have already given a value $k' = H(R', S')$ where (Q, R', S') is a Diffie-Hellman triple; if not, we choose an arbitrary random value for k' (but keep a record of it, because the same k' must be returned in the event of a later query for $H(R', S')$). We then compute the decryption $m' = E_{k'}^{-1}(c')$.

Eventually $\mathcal{A}$ gives the decryption of (R, c). By assumption in order to do that $\mathcal{A}$ must know the key k, and this means that $\mathcal{A}$ at some point must have queried the hash value $H(R, S)$. As soon as we receive that query, we determine through

[1]This problem was first defined by Abdalla, Bellare and Rogaway [1], who called it the "Strong DH" problem. We have chosen a different name for the problem in order to avoid confusion with the "Strong DH" problem considered in §11. The problem should also not be confused with the Gap-DH problem introduced in [32] where the solver is given access to an oracle for the full Decision Diffie-Hellman problem.

our One-sided DDH oracle that (Q, R, S) is a Diffie-Hellman triple, at which point we have the solution of the DH problem, as desired. □

5. The Twin Diffie-Hellman Problem

In [14] Cash, Kiltz, and Shoup constructed an ElGamal type encryption scheme that is slightly more complicated than the one in §4. Its advantage is that their proof of chosen-ciphertext security assumes intractability of a problem that — although at first appearing to be just a contrived and complicated variant of DH — in fact turns out to be equivalent to it. Thus, the same intractability assumption that's used to prove security of naive ElGamal in a very weak sense can be used to prove security of the scheme in [14] in a much stronger sense. Note that the somewhat unnatural One-sided Gap-DH assumption needed to establish chosen-ciphertext security of the protocol in §4 is not required for the version in [14].

We first describe the "twin" version of ElGamal encryption in [14]; after that we define the Twin Diffie-Hellman (TDH) problem upon which its security proof is based. We shall omit the proof, which is very similar to that of Proposition 2 in the previous section. Then we give the reductions that show the equivalence of TDH with DH.

Twin ElGamal encryption. As before, $\mathbb{G}$ is a group of order p with generator P. Alice's private key is a pair (x_1, x_2) of integers mod p, and her public key is the pair $Q_1 = x_1P$, $Q_2 = x_2P \in \mathbb{G}$. When Bob wants to send her a message m, he first chooses a random y mod p and sets $R = yP$, $S_1 = yQ_1$, $S_2 = yQ_2$. The hash-value $k = H(R, S_1, S_2)$ is the symmetric key, and the ciphertext is the pair $(R, E_k(m))$. Alice decrypts by setting $S_1 = x_1R$ and $S_2 = x_2R$, from which she can find the key k.

In [14] this scheme is shown to have chosen-ciphertext security if the following problem is hard.

- The Twin Diffie-Hellman (TDH) problem for a group $\mathbb{G}$ of prime order p with generator P has as input three points $Q_1, Q_2, R \in \mathbb{G}$. You are given a Decision Twin Diffie-Hellman oracle (for fixed Q_1, Q_2), that is, an oracle that for any $R^*, S_1, S_2 \in \mathbb{G}$ correctly answers the question: Is it the case that both $S_1 = y^*Q_1$ and $S_2 = y^*Q_2$, where y^* is the integer mod p for which $R^* = y^*P$? You must find x_1R, where x_1 is the integer mod p such that $Q_1 = x_1P$.

Remark 2. The definition of TDH in [14] also requires that the solver find x_2R, where x_2 is the integer mod p such that $Q_2 = x_2P$. However, in the application to the twin version of ElGamal encryption the second output can be omitted from the definition.

Proposition 3 ([14])**.** *The Twin Diffie-Hellman (TDH) problem is equivalent to the Computational Diffie-Hellman (DH) problem.*

Sketch of proof. The implication is trivial in one direction: someone who can solve DH can obviously solve TDH (without even needing the oracle). We outline the proof of the reverse implication.

To do this we suppose that we have a TDH oracle and show how we can use it to solve an instance of DH. So suppose we are given $(\mathbb{G}, p, P)$ and two points $Q, R \in \mathbb{G}$. We must find the point $S \in \mathbb{G}$ such that $S = xR$, where x is the (unknown to us) discrete log of Q to the base P.

We give the TDH oracle the input $(\mathbb{G}, p, P)$ and three points $Q_1, Q_2, R \in \mathbb{G}$, where we set $Q_1 = Q$, take R to be the same R as in our instance of DH, and then choose two random integers r and s mod p and set $Q_2 = sP - rQ$. That is, $x_2 = s - rx_1 \mod p$, where x_i is the discrete log of Q_i to the base P. Note that the output of the TDH oracle will be the solution of our instance of DH. However, in order to use the TDH oracle we have to be able to supply it with accurate answers to its Decision Twin Diffie-Hellman queries.

So suppose that the TDH oracle asks us to answer such a query with input $R^*, S_1, S_2 \in \mathbb{G}$. We compute $rS_1 + S_2$ and sR^*. We answer "yes" if these are equal and "no" otherwise. We show that the probability that this answer is incorrect is only of order p^{-1} (where the probability is taken over the set of all query inputs R^*, S_1, S_2). This means that the probability that the oracle will fail to give us the desired answer is small (and if that happens, we can repeat the process with a different r and s).

If "yes" is the correct answer to the query, then it is easy to see that we will always answer correctly, since in that case $S_1 = x_1R^*$, $S_2 = x_2R^*$, and $rS_1 + S_2 = (rx_1 + x_2)R^* = sR^*$.

If "no" is the correct answer to the query, then $S_1 = x_1'R^*$, $S_2 = x_2'R^*$, where at least one of the two x_i' is not equal to the corresponding x_i. If only one of the two x_i' is not equal to x_i, then we check that always $rS_1 + S_2 \neq sR^*$, so in that case our answer to the query is correct. That leaves the case when $x_1' \neq x_1$ and $x_2' \neq x_2$. Our answer will be incorrect if $rS_1 + S_2 = sR^*$, that is, if $rx_1' + x_2' \equiv s \equiv rx_1 + x_2 \pmod p$. For fixed x_1, x_2, r, this happens if and only if $x_2' = x_2 - r(x_1' - x_1) \mod p$. For variable S_1, S_2, R^* the chance of this equation holding mod p is of order p^{-1}, as claimed. This completes the outline of the proof. □

Remark 3. This proposition from [14] is a nice result. The authors designed a modified ElGamal encryption protocol in such a way that its chosen-ciphertext security could be proved using a rather unnatural-looking interactive problem, namely, Twin Diffie-Hellman. This by itself is not particularly impressive, since, as we shall see, the invention of contrived, exotic problems and protocols whose security is related to them has become a cottage industry. What is unusual is that in this case the authors were able to choose a problem — TDH — that they could prove to be equivalent to the classical, much-studied DH problem. It is their proof of the above proposition that makes their version of ElGamal worthwhile.

Remark 4. Speaking very informally, what makes their proof work — that is, what makes it possible to dispense with the Decision Diffie-Hellman oracle that's needed in the previously discussed version of ElGamal — is the addition of a second dimension (corresponding to the second fixed input Q_2 in the twin version of DDH). This gives the DH solver enough flexibility so that she can successfully simulate the oracle. This technique is reminiscent of the method used in [16] (see also the

discussion in §2 of [24]) to develop a discrete-log based encryption scheme that has a reductionist security proof using only a "standard" assumption (rather than the random oracle assumption) for the hash function.

Remark 5. The One-sided Gap-DH problem and the version of ElGamal encryption discussed in §4 can also be considered in a group $\mathbb{G}$ of composite order. Hofheinz and Kiltz [18] studied the One-sided Gap-DH problem in the group of so-called *signed quadratic residues* modulo a composite integer N that is the product of two distinct primes (satisfying some additional constraints). They proved that the problem is at least as hard as that of factoring N, thereby obtaining a reductionist argument for the security of the version of ElGamal encryption in §4 under the assumption that factoring is intractable and the random oracle assumption for the hash function.

6. A Strange Relationship Between Two Variants of the Diffie-Hellman Problem

In this section we examine two problems that have a curious relation to one another. Reductions are known in one direction for the search versions and in the opposite direction for the decision versions of the problems.

6.1. **The Square Diffie-Hellman problem.**

The following variant of the Diffie-Hellman problem was first presented in [29].

- The Square Diffie-Hellman (SqDH) problem for a group $\mathbb{G}$ of prime order p with generator P has as input another point $Q = xP \in \mathbb{G}$ (with unknown x mod p) and asks for the point $R \in \mathbb{G}$ for which $R = x^2P$.

The decision version of this problem is also of interest:

- The Decision Square Diffie-Hellman (DSqDH) problem for a group $\mathbb{G}$ of prime order p with generator P has as input $Q, R \in \mathbb{G}$ and asks whether or not $R = x^2P$, where x mod p is such that $Q = xP$.

Proposition 4 ([29])**.** *SqDH is equivalent to DH and DSqDH reduces to DDH.*

Proof. The reductions of SqDH to DH and of DSqDH to DDH are obvious, since the square-version of Diffie-Hellman is just a special case of the general version. The only nontrivial claim in the proposition is that DH reduces to SqDH. So suppose that we have an oracle for SqDH, and we are given DH input $(\mathbb{G}, p, P, Q, R)$, where $Q = xP$ and $R = yP$ for unknown x and y. For fixed $\mathbb{G}, p, P$ we apply the SqDH oracle three times to the points (i) Q, (ii) R, and (iii) $Q + R$. Let S_1, S_2, S_3 be the answers the oracle gives. We then compute

$$\frac{p+1}{2}(S_3 - S_1 - S_2) \;=\; \frac{p+1}{2}\big((x+y)^2 - x^2 - y^2\big)P \;=\; xyP,$$

as desired. □

Remark 6. No reduction from DDH to DSqDH is known. Thus, we write

$$\text{SqDH} \;\approx\; \text{DH} \qquad \text{but} \qquad \text{DSqDH} \;\leq\; \text{DDH}.$$

6.2. **The Tripartite Diffie-Hellman problem.**

- The Tripartite Diffie-Hellman (TriDH) problem for a group $\mathbb{G}$ of prime order p with generator P has as input a sextuple of points $(xP, yP, zP, xyP, xzP, yzP)$ (with unknown x, y, z) and asks for the point $xyzP \in \mathbb{G}$.
- The Decision Tripartite Diffie-Hellman (DTriDH) problem has the same input as the TriDH along with another point $S \in \mathbb{G}$ and asks whether or not $S = xyzP$.

These problems originated in the tripartite Diffie-Hellman key exchange, which works as follows. The three parties Alice (with private/public key pair (x, xP)), Bob (with key pair (y, yP)) and Cathy (with key pair (z, zP)) exchange the following information: Alice sends Bob $x(zP)$, Bob sends Cathy $y(xP)$, and Cathy sends Alice $z(yP)$, after which all three of them can compute the shared key $xyzP$.

6.3. **Mongrel dogs and transitivity of decision problem reductions.** Until now we have used a naive definition of a decision problem oracle, namely, a black box that always gives us the correct yes-or-no answer. When giving informal proofs and high-level overviews of arguments, usually nothing is lost by using this definition. However, the result we describe next — the reduction from DDH to DTriDH — simply cannot be accomplished (so far as we know) using the naive definition. Thus, regrettably we have to give a more technical definition of the task of a decision problem oracle. Readers who find the notation in this section burdensome should feel no guilt at all about simply skipping it — none of the sequel depends upon this section — and just taking our word for it that DDH $\approx$ DTriDH.

On the other hand, the reader who is willing to tolerate a reduction argument that occupies two pages will get to see a nice example of a type of proof called a *hybrid argument*. The basic idea is quite simple. The word "hybrid" should bring to mind the following situation. Suppose that a dog's parents are of two different breeds and differ from one another by $> \epsilon$. Then the dog must differ from at least one of its parents by $> \epsilon/2$. That is the hybrid argument.

Definition 1. *If X_1 and X_2 are two distributions on the same space, we say that X_1 and X_2 are (t, ϵ)-*indistinguishable *if for all algorithms $\mathcal{A}$ that upon input from the space return either 0 or 1 in time $\leq t$ one has $|p_1 - p_2| < \epsilon$, where p_i denotes the probability that $\mathcal{A}$ returns 1 when it is given input from X_i.*

Definition 2. *Let DP and DP′ be two decision problems; let X_1 (resp. X_2) denote the distribution that is uniform on the set of inputs for which the answer to DP is "yes" (resp. "no") and zero elsewhere, and let X_1' and X_2' be the analogous distributions for DP′. We say that DP reduces to DP′ if, given an oracle $\mathcal{A}'$ that (t', ϵ')-distinguishes X_1' from X_2', one can construct an algorithm $\mathcal{A}$ that (t, ϵ)-distinguishes X_1 from X_2, where $t \leq ct'$ and $\epsilon \geq \epsilon'/c$ for some constant c.*

Thus, to prove that DP reduces to DP′ one shows that (t, ϵ)-indistinguishability of X_1 and X_2 implies (t', ϵ')-indistinguishability of X_1' and X_2' for some $t' \geq t/c$, $\epsilon' \leq c\epsilon$.

Proposition 5 ([37])**.** DDH $\approx$ DTriDH.

Sketch of proof. First we define six distributions that will be used in the next two lemmas.

(1) X_1 is the uniform distribution on all of $\mathbb{G} \times \mathbb{G} \times \mathbb{G}$ (which we denote $\mathbb{G}^3$);
(2) X_2 is the distribution that is uniform on the subset
$$\Big\{(xP, yP, xyP) \;\Big|\; x, y \bmod p\Big\} \subset \mathbb{G}^3$$
and zero elsewhere;
(3) X_3 is the distribution that is uniform on the subset
$$\Big\{(xP, zP, xzP), (yP, zP, yzP), (zP, xyP, rP) \;\Big|\; x, y, z, r \bmod p\Big\} \subset \mathbb{G}^9$$
and zero elsewhere;
(4) X_4 is the distribution that is uniform on the subset
$$\Big\{(xP, zP, xzP), (yP, zP, yzP), (zP, dP, rP) \;\Big|\; x, y, z, d, r \bmod p\Big\} \subset \mathbb{G}^9$$
and zero elsewhere;
(5) X_5 is the distribution that is uniform on the subset
$$\Big\{(xP, zP, xzP), (yP, zP, yzP), (zP, dP, dzP) \;\Big|\; x, y, z, d \bmod p\Big\} \subset \mathbb{G}^9$$
and zero elsewhere;
(6) X_6 is the distribution that is uniform on the subset
$$\Big\{(xP, zP, xzP), (yP, zP, yzP), (zP, xyP, xyzP) \;\Big|\; x, y, z \bmod p\Big\} \subset \mathbb{G}^9$$
and zero elsewhere.

(Note that the successive X_i, $3 \le i \le 6$, differ only in one or two of the last three components.) Then DDH is the problem of distinguishing X_1 from X_2, and DTriDH is the problem of distinguishing X_3 from X_6. The intermediate problems X_4 and X_5 are needed for the hybrid argument; they should be regarded as two mongrels with parents X_3 and X_6 of different breeds. We warm up by proving a lemma that gives the reduction in the easy direction.

Lemma 1. *If X_3 and X_6 are (t, ϵ)-indistinguishable, then so are X_1 and X_2.*

Proof. Suppose that there were an algorithm $\mathcal{A}'$ with running time $\le t$ for which $|p'_1 - p'_2| \ge \epsilon$, where p'_i denotes the probability that $\mathcal{A}'$ returns 1 when the input is taken from X_i. We then construct an algorithm $\mathcal{A}$ with running time $\le t$ for which $|p_3 - p_6| \ge \epsilon$, where p_i denotes the probability that $\mathcal{A}$ returns 1 when given input from X_i. This will prove the lemma.

Given an element $(Q_1, \ldots, Q_9) \in \mathbb{G}^9$, $\mathcal{A}$ simply applies $\mathcal{A}'$ to the triple (Q_1, Q_6, Q_9). One checks that $p_3 = p'_1$ and $p_6 = p'_2$, in other words, $\mathcal{A}'$ has the same effect in its efforts to distinguish X_3 from X_6 as $\mathcal{A}$ does in trying to distinguish X_1 from X_2. □

Now we give the reduction in the hard direction, where a hybrid argument is needed.

Lemma 2. *If X_1 and X_2 are $(t, \epsilon/3)$-indistinguishable, then X_3 and X_6 are (t, ϵ)-indistinguishable.*

Proof. The hybrid argument uses the following transitivity property: Suppose X, X', X'' are distributions on the same space. If X and X' are (t, ϵ_1)-indistinguishable and X' and X'' are (t, ϵ_2)-indistinguishable, then X and X'' are $(t, \epsilon_1+\epsilon_2)$-indistinguishable. We prove this claim by contradiction. Suppose that there were an algorithm $\mathcal{A}$ that $(t, \epsilon_1 + \epsilon_2)$-distinguishes between X and X''; that is, given any input from the space, $\mathcal{A}$ returns 0 or 1 in time $\leq t$, and $|p - p''| \geq \epsilon_1 + \epsilon_2$, where we let p (resp. p', p'') denote the probability that $\mathcal{A}$ returns 1 when the input is from X (resp. X', X''). It follows that either $|p - p'| \geq \epsilon_1$ or $|p' - p''| \geq \epsilon_2$, and this contradicts the assumed indistinguishability of X and X' and of X' and X''.

Hence, to prove the lemma it suffices to show that if X_1 and X_2 are $(t, \epsilon/3)$-indistinguishable, then so are (a) X_3 and X_4; (b) X_4 and X_5; and (c) X_5 and X_6. We prove (a); the other two parts are similar.

Suppose that there were an algorithm $\mathcal{A}'$ with running time $\leq t$ for which $|p'_3 - p'_4| \geq \epsilon/3$. Given an element $(Q_1, Q_2, Q_3) \in \mathbb{G}^3$, we construct an algorithm $\mathcal{A}$ by choosing random z and r and running $\mathcal{A}'$ on input

$$(Q_1, zP, zQ_1), (Q_2, zP, zQ_2), (zP, Q_3, rP) \in \mathbb{G}^9.$$

Clearly $p_1 = p'_3$ and $p_2 = p'_4$, so this shows that if X_3 could be $(t, \epsilon/3)$-distinguished from X_4, then X_1 could be $(t, \epsilon/3)$-distinguished from X_2. (Note: In reality the running time of $\mathcal{A}$ is slightly more than that of $\mathcal{A}'$ because of the need to generate random z and r and compute rP, zP, zQ_1, zQ_2. However, for simplicity we have ignored this minor detail in the statement and proof of the lemma.) □

The two lemmas together imply equivalence of DDH and DTriDH. □

6.4. **The relation between Square and Tripartite Diffie-Hellman.** We now summarize what is known about reductions between the Diffie-Hellman, Square Diffie-Hellman, and Tripartite Diffie-Hellman problems and between the corresponding decision problems. As before, the notation $\mathcal{P} \approx \mathcal{Q}$ means that there are efficient reductions both from $\mathcal{P}$ to $\mathcal{Q}$ and from $\mathcal{Q}$ to $\mathcal{P}$, and we write $\mathcal{P} \leq \mathcal{Q}$ to mean that there is an efficient reduction from $\mathcal{P}$ to $\mathcal{Q}$ but none is known from $\mathcal{Q}$ to $\mathcal{P}$.

We note that reductions are not known from DH to TriDH or from DDH to DSqDH. Thus, we have

$$\text{TriDH} \leq \text{DH} \approx \text{SqDH},$$

while for the corresponding decision problems

$$\text{DTriDH} \approx \text{DDH} \geq \text{DSqDH}.$$

Thus,

$$\text{TriDH} \leq \text{SqDH} \qquad \text{but} \qquad \text{DTriDH} \geq \text{DSqDH}.$$

If we interpret the absence of known reductions to mean that one problem might be strictly harder than the other, then our conclusion would be that perhaps the search problem TriDH is strictly easier than SqDH, whereas the decision problem DTriDH is strictly harder than DSqDH. In other words, it would be easier to find a solution to TriDH than to SqDH, but it would be harder to say whether a candidate solution to TriDH is correct than to say whether a candidate solution to DSqDH is correct. Such a discrepancy between search and decision problems would certainly

defy intuition! Thus, a more reasonable interpretation of the above two inequalities is that they once again show the limitations of reductions as a way of gauging the true relative difficulty of two problems.

7. The One-More-Discrete-Log and One-More-Diffie-Hellman Problems

Because of the nature of chosen-ciphertext security for encryption (or chosen-message security for signatures) 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. On occasion such a problem, despite its unnatural appearance, might be used carefully and to good effect (as we discussed in §3). 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 (1MDL) problem as first formulated in [2] and [3]. 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 (1MDH) problem as first formulated (in a slightly different version) in [4]. 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 [26] 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 1MDL is harder than 1MDH for $g = 1, 2$, whereas Granger [17] recently observed that the two problems are of roughly equal difficulty for $g \geq 3$; and it is only for non-hyperelliptic curves of genus 3 that the two problems are no easier than the DL and DH problems. Note that reductions are not known from 1MDH to 1MDL or from 1MDL to 1MDH. Our conclusion was that it is often unclear how to gauge the true level of difficulty of an interactive problem or one with complicated input. Even though attacks on these problems do not necessarily lead to attacks on the corresponding encryption and signature schemes, it nevertheless seems a little risky to rely upon such problems for assurances about the security of protocols.

Remark 7. It follows from a general result of Brown [12] that no subexponential time reduction (for arbitrary $\mathbb{G}$) can exist either from DL to 1MDL or from DH to 1MDH. (See §8, where we discuss a similar nonexistence result in the case of the One-Prime-Not-p DL problem.) This theoretical result complements our empirical analysis and adds to the evidence that 1MDL and 1MDH are strictly easier than DL and DH.

8. The All-Primes-But-p Discrete Log Problem

In [27] the authors introduce a certain type of number-theoretic assumption to achieve a goal related to composition of secure computations. They summarize their assumption intuitively as follows: "it says that the [intractability of the DL problem] holds even in the presence of oracles breaking the [DL problem] for other groups."

In [27] the intractability assumption is given in the setting of the subgroup of order p of the multiplicative group of a prime field of $2p + 1$ elements ($2p + 1$ is called a "safe" prime). They point out that "our assumptions and protocols could be considered over other groups," and they invite further work:

> ...we believe [that] our new assumption is worth studying independently of the current context, and is likely to find other cryptographic applications.

We shall restate (and rename) the problem whose intractability they assume, along with a closely related one, in the context of general groups $\mathbb{G}$ of order p. We then give evidence that in this setting their problem is strictly easier than the DL problem. Thus, assumptions that the DL problems in different groups are independent of one another should be used with caution.

- The All-Primes-But-p Discrete Log Problem. You are given a t-bit prime p, a group $\mathbb{G}$ of order p, and two elements $P, Q \in \mathbb{G}$. You are also given an oracle that, acting as a black box, returns the solution to any discrete log problem in any group of order q, where q is any prime of at most t bits other than p. You must find the discrete log of Q to the base P.
- The One-Prime-Not-p Discrete Log Problem. You are given a t-bit prime p, a group $\mathbb{G}$ of order p, and two elements $P, Q \in \mathbb{G}$. For a single prime q of your choice that has at most t bits and is not equal to p you are given an oracle that returns the solution to any discrete log problem in any group of order q. You must find the discrete log of Q to the base P.

Remark 8. If one thinks of an oracle as an algorithm, then the notion that these problems (or the ones defined in [27]) might be hard seems implausible, because it is inconceivable that an algorithm would work for groups of all prime orders except one. However, the assumption makes perfect sense if one thinks of an oracle in the correct sense, which is as nothing but a black box that gives correct answers.

Remark 9. In a similar way at first glance it might seem to make no sense to ask about the hardness of the DL problem if one has an oracle that gives DL solutions in any group of order p'. Namely, Shoup [35] proved that any algorithm that finds discrete logs in a generic group of order p' must take time at least $\sqrt{p'}$. In roughly the

same length of time a generic algorithm could also find discrete logs in the original group $\mathbb{G}$ of order $p \approx p'$. However, once again the assumption makes sense because an oracle is not an algorithm, but rather just a black box that gives answers. A reader who is bothered by this on philosophical grounds and thinks that it sounds too much like making a provably false assumption should feel free to replace the words "any group" by "any algebraic group" in the two problems above — then the lower bound in [35] will not apply.

The result we prove below depends upon a conjecture about the distribution of numbers that are "almost smooth" in the sense of having at most one large prime factor.

Conjecture 1. *There exist non-negative constants k and ℓ such that, if $N(x, \ell)$ denotes the number of integers in the interval $[x, x + 4\sqrt{x}]$ that either are a prime or prime power or else have the property that their second largest prime factor is at most $\log^{\ell}(x)$, then we have*

$$N(x, \ell) > \sqrt{x}/\log^{k}(x)$$

for x sufficiently large.

It seems that a proof of Conjecture 1 is beyond the reach of current techniques of analytic number theory. However, from a heuristic standpoint it is extremely plausible; indeed, the Prime Number Theorem suggests that Conjecture 1 is true with $k = 1$ and $\ell = 0$.

Proposition 6. *Under Conjecture 1, for a group $\mathbb{G}$ of prime order p the One-Prime-Not-p Discrete Log Problem is polytime reducible to Diffie-Hellman (DH).*

Proof. Given an instance of the DL problem in $\mathbb{G}$, we first randomly select elliptic curves E over the field of p elements until we find a curve E whose group order has at most one prime divisor greater than $\log^{\ell}(x)$ (where $x = p+1-2\sqrt{p}$ is the beginning of the Hasse interval and ℓ is the constant in Conjecture 1). By Conjecture 1, this can be done in polynomial time. (Note that the order of E can also be factored in polynomial time.) We let p' denote the largest prime divisor of the order of E.

We suppose that we have a DH oracle for the group $\mathbb{G}$ and a DL oracle for any group of order p'. We must show that the DL problem on $\mathbb{G}$ can be solved in polynomial time. Now the DL problem on E can be solved in polynomial time using the One-Prime-Not-p DL oracle along with Pohlig-Hellman. We can now use the technique of Maurer (see [28, 30]) to see that DL on $\mathbb{G}$ does in fact reduce in polynomial time to DL on E.

We recall the main part of Maurer's argument. Suppose that x is the unknown discrete log of Q to the base P in our DL instance on $\mathbb{G}$. After finding a curve E over $\mathbb{F}_p$ on which the DL problem is easy, he constructs a point $A \in E$ whose coordinates are given explicitly and a point B whose x-coordinate is the unknown x and whose y-coordinate is found implicitly — that is, the element $yP \in \mathbb{G}$ is found — using the DH oracle in $\mathbb{G}$. Such a point is represented not as (x, y) but rather as (Q, R), where $Q = xP$, $R = yP$. After solving DL on E to get the discrete log of

B to the base A, he finds the point (x, y) explicitly and in that way determines x. This concludes the proof. □

Remark 10. As mentioned above, the notion that the One-Prime-Not-p and even the All-Primes-But-p versions of the DL problem should be as hard as DL is essentially a way of saying that groups of different prime orders have completely independent Discrete Log Problems. Proposition 6 can be viewed as a *relativized* result (in the sense that Shoup uses the term in [36]). Namely, it says that, relative to a Diffie-Hellman oracle, this conjectured independence seems to fail. In the presence of a DH oracle One-Prime-Not-p DL seems easier than DL. The former can be solved in polynomial time by Proposition 6, whereas the best results for the latter (see [9]) are subexponential but very far from polytime.

The following corollary follows immediately from Proposition 6 by the transitivity of reductions.

Corollary 1. *Under Conjecture 1, the Discrete Log problem is not polytime reducible to the One-Prime-Not-p Discrete Log Problem unless DL is polytime reducible to DH.*

In [12] Brown proved a general result that implies that this corollary is moot: in fact, there can be no subexponential time reduction from DL to One-Prime-Not-p DL. Brown showed that for any problem $\mathcal{P}$ with the *random self-reducibility* property, if there were a reduction from the basic problem $\mathcal{P}$ to a version of $\mathcal{P}$ that gives the solver an oracle for other instances of $\mathcal{P}$, then $\mathcal{P}$ itself is easy. In particular, if DL polytime reduces to One-Prime-Not-p DL, then discrete logs can be found in polynomial time. The same statement holds with "polytime" replaced by "subexponential time". Namely, we have (see [12])

Proposition 7. *If there is a subexponential time reduction $\mathcal{R}$ from the Discrete Log problem in groups $\mathbb{G}$ of prime order p to the One-Prime-Not-p Discrete Log problem in $\mathbb{G}$, then DL can be solved in subexponential time.*[2]

Sketch of proof. Suppose we are given an instance of DL in a group $\mathbb{G}''$ of order p'', where p'' is a t-bit prime. We choose an arbitrary t-bit prime $p \neq p''$ and let $\mathbb{G}$ be a generic group of order p.[3] This means that when $\mathcal{R}$ works with $\mathbb{G}$ we simulate an oracle that gives $\mathcal{R}$ the (random) labels of elements and the results of the group operation.

Because of Shoup's lower bound [35], it follows that $\mathcal{R}$ cannot solve the DL instance in subexponential time without the use of the One-Prime-Not-p oracle. When it calls upon that oracle, we simulate a One-Prime-Not-p solver, and $\mathcal{R}$ must give correct answers to the queries that the solver is allowed to make. We choose $p' = p''$, and simply demand that $\mathcal{R}$ give us the answer to our original instance of DL. It happily does so, and we're done. □

Corollary 2. *No subexponential time reduction (for arbitrary groups of prime order) exists from DL to One-Prime-Not-p DL.*

[2] In essentially the same time as required by $\mathcal{R}$.

[3] In [12] a different argument is given using random self-reducibility rather than a generic group to force $\mathcal{R}$ to use the One-Prime-Not-p oracle that's available to it.

Namely, such a reduction would lead to a subexponential time algorithm for the DL on generic groups, and this is impossible by [35].

9. 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-value, 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 — that is, if 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 this word in cryptography. For example, a proof of security that uses weaker assumptions about the hash function than the random oracle assumption is commonly said to be a proof under a standard assumption. 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 non-standard problem. To avoid this misunderstanding, we would much prefer that researchers use the term "concrete" rather than "standard" when they want to emphasize that they are not using the random oracle assumption.

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 non-standard. The over-use of the word "standard" in connection with assumptions that are anything but standard provides another instance of *narrative inversion* in cryptography (see §12 of [23]).

10. 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, 33]), 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 non-degenerate 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 $p \mid q^k - 1$. 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 $p \mid q+1$ and $q+1 \mid q^2-1$, the embedding degree is $k=2$ and pairings can be computed quickly.

If $\mathbb{G}$ has a readily computable pairing, then the DDH on $\mathbb{G}$ is easy. Namely, given $P, Q, R, S \in \mathbb{G}$ with $Q = xP$, $R = yP$, and $S = zP$ for unknown integers x, y, z mod p, by the bilinear property of pairings we have

$$e(Q,R) = e(P,P)^{xy}, \qquad e(P,S) = e(P,P)^z.$$

Hence, the DDH has a "yes" answer if and only if $e(Q,R) = e(P,S)$.

Remark 11. If the Diffie-Hellman problem is hard on a group $\mathbb{G}$ with an easily computable pairing, then $\mathbb{G}$ is a DH gap group (see §3). In fact, groups with pairings are the only known examples of gap groups. For all other groups we solve DDH simply by finding discrete logs; there is no known way to solve DDH that is faster than that.

One of the first uses of pairing-based cryptography was the elegant solution by Boneh and Franklin [8] to an old question of Shamir [34], 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 e-mail address). Such a system is called *identity-based encryption*. Another early application was the Boneh-Lynn-Shacham signature scheme (see [10]), where the signatures had the advantage of being more compact than in most other protocols.

By the time pairing-based cryptography arose, it had become *de rigeur* 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.

11. The Strong Diffie-Hellman Problem

In [6, 7], 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 non-degenerate 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 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 that had never been studied before. It was because of nervousness about the ℓ-SDH assumption that the authors of [6] 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 [31] and Shoup [35]. 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 [35] Shoup proved that neither the Discrete Log problem nor the Diffie-Hellman problem can be solved in fewer than $\sqrt{p}$ steps in a generic group of prime order p.

In §5 of [6] 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 DL or DH 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 DL and DH. However, at Eurocrypt 2006 Cheon [15], using the same attack that had been described earlier in a different setting by Brown and Gallant [13] (see §3), showed that ℓ-SDH can be solved — and in fact the discrete logarithm x can be found — in $\sqrt{p/\ell_0}$ operations if $\ell_0 \leq \ell$ 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 DH problem 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 [15], one can forge signatures in roughly $p^{2/5}$ operations (with roughly $p^{1/5}$ signature queries) under certain conditions.

On the one hand, the attack on the Boneh-Boyen scheme in [20] is not practical, because it takes time $p^{2/5}$ and because it can be avoided simply by a condition on the divisors of $p-1$ (and in view of a closely related attack in [15] one also needs a condition on the divisors of $p+1$). On the other hand, no such restriction on p was thought to be necessary when the protocol was proposed, and the attack arising from [13, 15, 20] came as a surprise. Thus, it is reasonable to have doubts about the true security of Boneh-Boyen signatures.

For short signatures using pairings, probably the best advice is to stick with the Boneh-Lynn-Shacham scheme [10]. As we remarked in [25], in our opinion it is not a good idea to switch away from BLS signatures simply because its reductionist security argument uses the random oracle assumption. In this case the devil we know (the random oracle model) seems to be more benign than the devil we don't know (vulnerability of ℓ-SDH).

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 [11]:

- 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 see something very positive in the unusual assortment of intractability assumptions. In a paper presented at the Pairing 2008 conference [11], 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 touts the advantages of 21-st 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.

12. Sequential Aggregate Signatures

In 2007 Boldyreva, Gentry, O'Neill, and Yum [5] constructed a new sequential aggregate signature scheme. 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 [5] is to secure routing of messages through a network.

The authors of [5] describe the advantages of their signature scheme. In the first place, it is identity-based; in other words, there are no public keys other than the signers' email addresses; this "permits savings on bandwidth and storage..." 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 identity-based sequential aggregate signature scheme (IBSAS) in [5] has "improved security."

The construction in [5] used groups $\mathbb{G}$ with bilinear pairings, and their 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 non-degenerate bilinear pairing $e : \mathbb{G} \times \mathbb{G} \longrightarrow \mu_p$, fixed non-identity 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 [6], the authors of [5] 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 [5] appeared, Hwang, Lee, and Yung [19] made a startling discovery: the "provably secure" IBSAS scheme in [5] 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

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

(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, that is, it works in any group of order p.

But Theorem 5.1 of [5], 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.

What went wrong? The 4-page single-spaced argument purporting to prove Theorem 5.1 is hard to read because of its cumbersome notation and turgid formalism.

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 the attacker is free to do arithmetic in that field and even solve an equation or two.

13. Conclusions

(1) If proponents of a protocol prove a tight reduction linking security to a certain mathematical problem and also prove that this problem is equivalent to a classical, much-studied problem — as was done in [14] — then the reductionist argument is a useful contribution to the analysis of the security of the protocol.

(2) If, on the other hand, a provable security theorem assumes intractability of a contrived, ornate, and poorly understood mathematical problem, then the "proof" is of little value in assessing the actual security of the protocol. Readers should be wary of protocol designers who try to put a positive spin on the "rich palette" of such problems (as in [11]) or who over-use the word "standard" (as in [5]), since the reality is often the direct opposite of what such language suggests. This is an example of *narrative inversion* (see [23]).

(3) There are many problems that empirically seem to be equivalent to one another but for which this equivalence is unlikely to be provable by reductions in both directions. In such cases the relation $\mathcal{P} \leq \mathcal{Q}$ may be interpreted either as a warning that $\mathcal{P}$ might turn out to be strictly easier than $\mathcal{Q}$ or else as an indication of the limitations of reductions. It is possible that for certain problems $\mathcal{Q}$ such as the DL problem there are a vast number of problems $\mathcal{P}$ that reduce to $\mathcal{Q}$ and cannot be solved more efficiently than by first solving $\mathcal{Q}$ but for which there is no reduction from $\mathcal{Q}$ to $\mathcal{P}$. In other words, the reduction approach does not necessarily provide a reliable guide to the actual relative difficulty of two problems.

Acknowledgments

We wish to thank Michael Fellows, Ann Hibner Koblitz, Igor Shparlinski, and the anonymous referee for helpful comments and discussions.

References

[1] M. Abdalla, M. Bellare, and P. Rogaway, The oracle Diffie-Hellman assumptions and an analysis of DHIES, *Topics in Cryptology — CT-RSA 2001*, LNCS 2020, Springer-Verlag, 2001, pp. 143-158.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] D. Boneh and X. Boyen, Short signatures without random oracles and the SDH assumption in bilinear groups, *J. Cryptology*, **21** (2008), pp. 149-177.

[8] 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.

[9] D. Boneh and R. Lipton, Algorithms for black-box fields and their application to cryptography, *Advances in Cryptology — CRYPTO '96*, LNCS 1109, Springer-Verlag, 1996, pp. 283-297.

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

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

[12] D. Brown, Irreducibility to the one-more evaluation problems: More may be less, available at http://eprint.iacr.org/2007/435.

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

[14] D. Cash, E. Kiltz, and V. Shoup, The twin Diffie-Hellman problem and applications, *J. Cryptology*, **22** (2009), pp. 470-504.

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

[16] 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.

[17] R. Granger, On the static Diffie-Hellman problem on elliptic curves over extension fields, available at http://eprint.iacr.org/2010/177.

[18] D. Hofheinz and E. Kiltz, The group of signed quadratic residues and applications, *Advances in Cryptology — CRYPTO 2009*, LNCS 5677, Springer-Verlag, 2009, pp. 637-653.

[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*, pp. 157-160.

[20] D. Jao and K. Yoshida, Boneh-Boyen signatures and the Strong Diffie-Hellman problem, *Pairing-Based Cryptography — Pairing 2009*, Lecture Notes in Computer Science, 5671 (2009), pp. 1-16.

[21] A. Joux, A one round protocol for tripartite Diffie-Hellman, *Algorithmic Number Theory: Fourth International Symposium*, Lecture Notes in Computer Science, 1838 (2000), pp. 385-393.

[22] A. Joux and K. Nguyen, Separating decision Diffie-Hellman from computational Diffie-Hellman in cryptographic groups, *J. Cryptology*, **16** (2003), pp. 239-247.

[23] A. H. Koblitz, N. Koblitz, and A. Menezes, Elliptic curve cryptography: The serpentine course of a paradigm shift, to appear in *J. Number Theory*, available at http//eprint.iacr.org/2008/390.

[24] N. Koblitz and A. Menezes, Another look at "provable security," *J. Cryptology*, **20** (2007), pp. 3-37.

[25] N. Koblitz and A. Menezes, Another look at generic groups, *Advances in Mathematics of Communications*, **1** (2007), pp. 13-28.

[26] N. Koblitz and A. Menezes, Another look at non-standard discrete log and Diffie-Hellman problems, *J. Math. Cryptology*, **2** (2008), pp. 311-326.
[27] T. Malkin, R. Moriarty, and N. Yakovenko, Generalized environmental security from number theoretic assumptions, *Theory of Cryptography — TCC 2006*, LNCS 3876, Springer-Verlag, 2006, pp. 343-359.
[28] U. Maurer, Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete logarithms, *Advances in Cryptology — CRYPTO '94*, LNCS 839, Springer-Verlag, 1994, pp. 271-281.
[29] U. Maurer and S. Wolf, The relationship between breaking the Diffie-Hellman protocol and computing discrete logarithms, *SIAM J. Computing*, **28** (1999), pp. 1689-1721.
[30] U. Maurer and S. Wolf, The Diffie-Hellman protocol, *Designs, Codes and Cryptography*, **19** (2000), pp. 147-171.
[31] V. I. Nechaev, Complexity of a deterministic algorithm for the discrete logarithm, *Mathematical Notes*, **55** (2) (1994), pp. 165-172.
[32] T. Okamoto and D. Pointcheval, The gap-problem: a new class of problems for the security of cryptographic schemes, *Public Key Cryptography — PKC 2001*, LNCS 1992, Springer-Verlag, 2001, pp. 104-118.
[33] R. Sakai, K. Ohgishi, and M. Kasahara, Cryptosystems based on pairings, *Proceedings of the 2000 Symposium on Cryptography and Information Security*, Okinawa, 2000.
[34] A. Shamir, Identity-based cryptosystems and signature schemes, *Advances in Cryptology — CRYPTO '84*, LNCS 196, Springer-Verlag, 1985, pp. 277-296.
[35] V. Shoup, Lower bounds for discrete logarithms and related problems, *Advances in Cryptology — Eurocrypt '97*, LNCS 1233, Springer-Verlag, 1997, pp. 256-266.
[36] V. Shoup, OAEP reconsidered, *Advances in Cryptology — CRYPTO 2001*, LNCS 2139, Springer-Verlag, 2001, pp. 239-259.
[37] M. Steiner, G. Tsudik, and M. Waidner, Diffie-Hellman key distribution extended to group communication, *Proc. 3rd ACM Conference on Computer and Communications Security*, 1996, pp. 31-37.

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

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