ANOTHER LOOK AT NON-STANDARD DISCRETE LOG AND DIFFIE-HELLMAN PROBLEMS

NEAL KOBLITZ AND ALFRED MENEZES

ABSTRACT. We examine several versions of the one-more-discrete-log and one-more-Diffie-Hellman problems. In attempting to evaluate their intractability, we find conflicting evidence of the relative hardness of the different problems. Much of this evidence comes from natural families of groups associated with curves of genus 2, 3, 4, 5, and 6. This leads to questions about how to interpret reductionist security arguments that rely on these non-standard problems.

1. INTRODUCTION

In [25] we raised the issue of the use of non-standard versions of discrete logarithm and Diffie-Hellman problems in order to give reductionist security proofs or in order to give such proofs without random oracles. In this paper we look more closely at the "one-more" versions of these problems. Our purpose is to show their subtlety and give evidence that versions that at first glance seem to be equally hard might in the real world turn out to have vastly different levels of difficulty. This evidence comes from certain natural families of groups — the jacobians of curves of small genus.

We shall distinguish between several different formulations, which we argue are not likely to be equivalent to one another. The exact statement of a "hard" problem makes a big difference, although sometimes researchers tend to lump different variants together. For example, in the context of the RSA e-th root problem, Joux, Naccache, and Thomé [22] attributed their one-more version of the problem to Bellare *et al* [4], although in reality the problem defined in [4] (which in the context of discrete logs we label 1MDLP and 1MDHP below) is quite different from the version in [22] (the analogous problems for discrete log and Diffie-Hellman are labeled DTDLP and DTDHP below).

In §2 we state eight variants of the problems and briefly discuss what inequalities are known for their levels of difficulty. In §3 we describe some protocols whose security is related to these problems. In some cases the one-more-Diffie-Hellman problems are equivalent to the adversary's task in attacking the protocol, but we know of no protocol for which this is true of

Date: December 10, 2007; revised on April 22, 2008.

Key words and phrases. Public Key Cryptography, discrete log problem, Diffie-Hellman problem, jacobian groups, index calculus, provable security.

the one-more-discrete-log problems. In §4 we give an overview of algorithms for discrete-log type problems on the jacobian of a low-genus curve. We show that state-of-the-art algorithms for such groups suggest that several of the problems in §2 are incomparable to one another. In §5 we draw some conclusions.

2. Discrete Log and Diffie-Hellman Problems

2.1. **The problems.** Let G be a group of prime order n, whose group operation will be written multiplicatively (although in the case of the jacobian groups we shall be considering later it is traditional to write it additively). Group elements are described by binary strings of length $O(\log n)$, and hence $\log n$ is the complexity parameter. Let g be a generator, i.e., a non-identity element. In four of the eight problems we consider we also need a "challenge oracle," which, when queried by the solver, gives a random group element for which the solver must find either the discrete log or else the solution to one-sided Diffie-Hellman. This oracle models the situation where the solver is allowed to decide at any point how many group elements the input contains.

(1) The Discrete Logarithm Problem (DLP). Given $Y \in G$, find an integer y mod n such that $Y = g^y$.
(2) The Diffie-Hellman Problem (DHP). Given $X, Y \in G$, find $Z \in G$ such that $z \equiv xy \pmod{n}$, where $X = g^x$, $Y = g^y$, and $Z = g^z$.
(3) The One-More Discrete Log Problem (1MDLP) as first formulated in [4] and [6]. The solver is supplied with a challenge oracle that produces a random group element Y_i when queried and a discrete log oracle. After t queries to the challenge oracle (where t is chosen by the solver) and at most $t-1$ queries to the discrete log oracle, the solver must find the discrete logs of all t elements Y_i.
(4) The ("static" or "one-sided") One-More Diffie-Hellman Problem (1MDHP) as first formulated by Boldyreva [8] (her version was slightly different, see Remark 1 below). The solver is given an element $X \in G$, an oracle that can solve the DHP for the given X and arbitrary $Y \in G$, and a challenge oracle that produces random group elements Y_i. After t queries to the challenge oracle (where t is chosen by the solver) and at most $t-1$ queries to the DHP oracle, the solver must find the solutions Z_i of all DHP instances with input X, Y_i, $i = 1, \ldots, t$.
(5) The "Delayed Target" One-More Discrete Log Problem in the sense of Joux-Naccache-Thomé (DTDLP). The solver is supplied with a discrete log oracle and must find the discrete log of a random group element Y that is given to the solver only after all the queries have been made.

(6) The "Delayed Target" One-More Diffie-Hellman Problem as defined by Freeman [18] (DTDHP). The solver is given $X \in G$ and a one-sided Diffie-Hellman oracle, and must solve the DHP with input X, Y, where Y is a random group element that is given to the solver only after all the queries have been made.

(7) The DLP1 problem (by analogy with the RSA1 problem in [24]). The solver is supplied with a challenge oracle that produces random group elements Y_i and an oracle that will give the discrete logs of any of the Y_i. (The number of discrete log queries must be strictly less than the number of challenge queries.) The solver must find the discrete log of one of the Y_i that was not queried.

(8) The DHP1 problem (see [19]). The solver is given $X \in G$, a challenge oracle that produces random group elements Y_i, and an oracle that will solve the DHP with input X, Y_i for any of the Y_i. (The number of DHP queries must be strictly less than the number of challenge queries.) The solver must solve the DHP with input X, Y_i for one of the Y_i that was not queried.

2.2. **Relative difficulty.** Because seven of these problems are related to the security of cryptographic protocols and all of them are useful in gaining insight into security reductions, we should try to understand their relative difficulty (see Figure 1). First, a DLP algorithm trivially solves any of the remaining seven problems. Similarly, a DHP algorithm will immediately solve 1MDHP, DTDHP, or DHP1. It is also easy to see that 1MDLP efficiently reduces to DLP1, and 1MDHP efficiently reduces to DHP1.

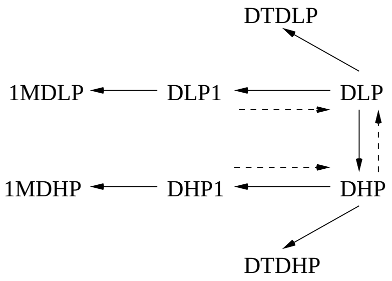

FIGURE 1. Relative difficulty of discrete log and Diffie-Hellman problems. $A \leftarrow B$ means that there is an efficient reduction from A to B. The dashed arrows from DLP1 to DLP and from DHP1 to DHP denote non-tight reductions. The dashed arrow from DHP to DLP means that a reduction is known under certain conditions on the problem instances.

In addition, in the 1990's work by den Boer, Maurer, Wolf, Boneh, Lipton and others (see [29] for a survey) developed strong evidence for the equivalence of the Diffie–Hellman and Discrete Log Problems in all prime-order groups.

Informally speaking, it seems clear that DLP1 is equivalent to DLP and DHP1 is equivalent to DHP, even though, so far as we know, there are no tight reductions from DLP to DLP1 or from DHP to DHP1. The argument is the same as the one we gave in [24] in the closely analogous case of the RSA problem and the problem that we called RSA1. Let's consider, for example, DLP1. The discrete log queries will give us a randomly distributed set of pairs (Y_i, y_i) that is indistinguishable from the set that an attacker could generate for herself starting with random y_i and setting $Y_i = g^{y_i}$. Thus, the queries can't help, and we can assume a passive attacker. In that case there is a simple reduction from DLP to the problem of finding one out of t discrete logs: given an input Y to DLP, one creates an input to the latter problem by setting $Y_i = g^{r_i}Y$ for random r_i, and we see that the two problems are equivalent.

Beyond this, however, it is not easy to prove (or even make convincing informal arguments for) inequalities between the levels of difficulty of these eight problems. For example, in practice DTDLP seems to be easier than 1MDLP, because in the former problem only a single discrete log must be found. Moreover, the feature of 1MDLP that potentially could make it easier than DTDLP in some cases — namely, that the solver in 1MDLP can make discrete log queries after knowing some or all of the group elements Y_i for which she must find the discrete logs, whereas in DTDLP the target is revealed only after she makes the queries — doesn't seem to help in any of the known algorithms. Thus, we might be tempted to write DTDLP $\leq$ 1MDLP, although it seems unlikely that there is a tight reduction from DTDLP to 1MDLP, and we do not even know of an informal argument in support of this inequality.

Although DHP is trivially easier than (or equivalent to) DLP, it is by no means clear that the same is true of 1MDHP and 1MDLP or of DTDHP and DTDLP. Although on the DLP side the solver's task is harder, she has a more powerful oracle (that is, a DLP oracle rather than a DHP oracle) to help her out. Thus, it is not obvious in which direction the inequality should go between the "one-more" versions of DLP and DHP. One of our purposes in this paper is to give evidence that the two problems are incomparable. In other words, there are natural families of groups in which 1MDHP is strictly easier than 1MDLP in practice, and there are natural families of groups in which the reverse is true; and the same holds for the comparison between DTDHP and DTDLP. In fact, we can conclude that in the unlikely event that someone constructs a tight reduction in either direction, that would immediately give an improved real-world algorithm in certain groups.

3. Protocols Based on These Problems

3.1. **Protocols equivalent to the problems.** One of the earliest and most elegant pairing-based protocols was the Boneh-Lynn-Shacham signature scheme [10]. If x is the private key and $X = g^x$ is the public key of the signer, and if H is the hash value of the message to be signed, then the signature is simply $S = H^x$. The verification consists in checking that the two pairing values (g, S) and (X, H) are equal. In [19] Galindo noticed that the security of BLS signatures against chosen-message attack is precisely equivalent to DHP1.

In [18] Freeman gave the following identification protocol based on BLS signatures [10]. Suppose that the Prover wants to convince the Verifier that she knows the discrete log x of $X = g^x$. The Verifier randomly chooses y, sets $Y = g^y$, and sends Y as a challenge to the Prover, who must respond with $Z = Y^x$. The Verifier checks that $Z = X^y$.

The security of this scheme means that a Verifier who makes a bounded number of challenges must not then be able to convince another verifier that he possesses the discrete log x. It is easy to see that the adversary's task here is equivalent to DTDHP.

In [8] Boldyreva proposed the following blind signature scheme (also based on BLS signatures [10]). Suppose that Alice wants a signer S to help her sign a message m. She first hashes the message, chooses a random r, and sends $H(m)g^r$ to S. The signer has public key $X \in G$ and private key x, i.e, $X = g^x$. After receiving $H(m)g^r$ from Alice, S raises it to the x-th power and sends that to Alice, who divides by X^r. The result is $(H(m)g^r)^x/X^r = H(m)^x$, which is Alice's BLS signature for m.

The accepted definition of security of a blind signature scheme is that an adversary that makes at most $t - 1$ queries to S (where t is chosen by the adversary) cannot feasibly produce signatures for t messages of its choice. Under the random oracle assumption it is easy to check that the adversary's task is equivalent to 1MDHP (more precisely, to the "chosen-target" version described in Remark 1).

Remark 1. The definition of 1MDHP in [8] is slightly different from the one in §2. Namely, the solver is required to give the solution to the one-sided DHP only for a subset of the values produced by the challenge oracle that is greater in number than the number of queries to the DHP oracle. (This version models what happens in a chosen-message attack on a signature scheme, with the challenge oracle corresponding to the hash function and the DHP oracle modeling signature queries.) This "chosen-target" version is closely analogous to a similar RSA e-th root problem that was studied in [4]. This version is clearly no harder than the version of 1MDHP in §2, which is sometimes called the "known-target" version. Moreover, the chosen-target problem can be shown to be equivalent to the known-target problem in the following strong sense: An algorithm that solves the chosen-target version in a given group G with fewer than b queries to the DHP oracle can be used to solve the known-target version with fewer than b queries to the DHP oracle. We sketch an outline of how this reduction works. Following the idea in [4] (where the analogous equivalence was proved for the RSA e-th root problem), we suppose that we have such an algorithm for the chosen-target version and want to use it to solve an instance of the known-target version. In the latter problem we are given a group element X, a challenge oracle, and a one-sided DHP oracle. We make b queries to the challenge oracle to get $Y_1, \ldots, Y_b$. Now we run the algorithm for the chosen-target version of 1MDHP with input X. We answer its DHP queries (of which there are at most $t-1 < b$) with our DHP oracle, and we answer the i-th challenge query by choosing random exponents a_{ij}, $j = 1, \ldots, b$, and setting

$$Y'_i = \prod_{j=1}^{b} Y_j^{a_{ij}}. \tag{1}$$

The algorithm eventually outputs $Z'_i = (Y'_i)^x$ (where x is the discrete log of X) for some subset of t of the Y'_i. If $t < b$, then we randomly choose $b-t$ of the remaining Y'_i and find the corresponding Z'_i by querying our DHP oracle. (Note that a total of fewer than b queries to this oracle have been made.) Renumbering the Y'_i, we may suppose that we have the DHP solutions $Z'_1, \ldots, Z'_b$ for $Y'_1, \ldots, Y'_b$. From (1) it follows that $Z'_i = \prod_{j=1}^{b} Z_j^{a_{ij}}$, where the $Z_j = Y_j^x$ are the (not yet known) DHP solutions for the Y_j. If the $b \times b$ matrix $\{a_{ij}\}$ is invertible modulo the group order n with inverse matrix $\{a'_{ij}\}$, then we can solve the known-target version of 1MDHP by computing this inverse matrix and setting $Z_i = \prod_{j=1}^{b} (Z'_j)^{a'_{ij}}$; if it is not invertible, we repeat with different random a_{ij}.

3.2. **Protocols with reduction but not equivalence.** There have been a series of papers giving reductions from either the 1MDLP (see [5, 6, 7, 21]) or the 1MDHP (see [5, 3, 27]) to successful attacks on identification and signature schemes of Schnorr and others [32]. There have also been a few cases where the security of a pairing-based scheme has been shown to be implied by hardness of the 1MDLP (see [2]). In all these papers, however, what was shown was not that the purportedly hard one-more DLP/DHP problem is equivalent to breaking the protocol, but rather that this problem is no more difficult than the adversary's task. In the case of 1MDLP it seems hard even to imagine a protocol whose security is *equivalent* to the problem.

We now briefly describe the Schnorr identification scheme in order to discuss why it is unlikely that the reduction of 1MDLP to breaking the scheme can be reversed (i.e., it is unclear how to break the scheme if 1MDLP is solved). Suppose that the Prover wants to prove to the Verifier that she knows the discrete log x of her public key X. As before, we are working in a group G of prime order n with fixed generator g, capital letters denote group elements, and the corresponding small letters denote their discrete logs, which are regarded as integers mod n.

The Prover chooses random y and sends $Y = g^y$ to the Verifier. He chooses a random challenge c, which he sends to the Prover, who computes

$z = y + cx \bmod n$, which she sends to the Verifier. He verifies that $g^z = YX^c$, and accepts her identity.

In [6] Bellare and Palacio reduce 1MDLP to breaking this identification scheme. On the other hand, it is unlikely that a reduction can be constructed in the other direction. In order to do so one would have to show how to use an attacker on Schnorr to answer the discrete log oracle queries in 1MDLP. But an oracle that responds to a challenge c by giving the discrete log $z = y + cx$ of YX^c cannot in any obvious way be used as an oracle that gives the discrete log of an arbitrary Z. In order to determine a challenge c that produces such a result, one would have to find c such that $X^c = Z/Y$, in other words, one would have to find the discrete log of Z/Y to the base X; and presumably this is no easier than finding discrete logs to the base g. Thus, it seems unlikely that the security of the Schnorr identification scheme is provably equivalent to intractability of 1MDLP.

In [25] we commented on the daunting task of assessing the security assurance provided by a reduction from a non-standard mathematical problem. If the only way to obtain a “proof of security” (or obtain a proof without random oracles) is to concoct an interactive math problem that people have no desire to study, it seems to us that the resulting guarantee is not very convincing.

From the standpoint of someone who’s thinking about studying a non-standard problem such as the various interactive versions of the DLP, there is a disincentive to do so if the reduction goes only one way, that is, if the “hard” problem might be strictly easier than a successful attack on the protocol. The drawback is clear if we consider an analogous situation. Suppose that a protocol is advertised with certain parameters and, after much effort, someone breaks it. The cryptanalyst’s nightmare is that the proponents of the cryptosystem will learn of this and quickly replace the old parameters with new ones in all their postings, thereby blunting the impact of the analyst’s work. In like manner, suppose that a cryptanalyst works hard to develop a faster-than-expected algorithm for a non-standard DLP-type problem used in a security proof, and that this does not break the protocol, but only calls into question the assurance given by one particular proof. The danger (from the point of view of the analyst) is that the promoters of the protocol will point out that their protocol has not been broken, and then quickly give a new “proof of security” based on a slightly different problem — in which case the researcher’s algorithm no longer has any relevance. If the non-standard problem was of little interest except for its appearance in the earlier “proof of security” (now superceded), then the cryptanalyst might justifiably feel resentment about having wasted time developing an attack on the problem. So if one wants to encourage intensive research on an underlying “hard” problem, it’s best if the problem is *equivalent* to a successful attack.

Bellare and Palacio [6] justify their use of a non-standard DLP problem as follows:

> Although the assumption is relatively novel and strong, our result reduces the security of the Schnorr identification scheme to a question about the hardness of a number-theoretic problem, thereby freeing a cryptanalyst from consideration of attacks related to the identification problem itself.

In other words, they have removed extraneous features from the protocol and reduced the security issue to a clean, clearcut math problem.

We would find this point of view more persuasive if the security of the Schnorr scheme had been shown to be *equivalent* to 1MDLP, and of course this is not the case. In fact, in §4 we give evidence that in certain groups 1MDLP is probably strictly easier than breaking Schnorr.

Our view is that if a reductionist security argument is to be of much value, it should satisfy at least one of two conditions: (1) the problem that is shown to reduce to successful attacks on the scheme is of independent interest; or (2) the problem, while somewhat contrived and unnatural, is at least *equivalent* to security of the protocol, i.e., it is a faithful reflection of the true level of difficulty of successful attacks on the scheme.

Remark 2. For an example of an interactive version of the Decision Diffie-Hellman Problem that was used in a security reduction and soon after turned out to be easy to solve, see [1] and [34]. However, Szydlo's successful attack in [34] on the non-standard problems in [1] did not imply that the corresponding protocols were broken, but only that "the level of security assurance provided by this scheme is an open question."

Similarly, when Cheon [13] (using an algorithm first developed by Brown-Gallant [12]) showed that the "m-Strong Diffie-Hellman Problem" in [9] was easier than expected (i.e., his algorithm in certain cases was much faster than the best available DHP algorithms), that did not give an attack on the Boneh-Boyen scheme, because the reduction between the Strong DHP and chosen-message attacks on the signature scheme goes in only one direction.

Remark 3. It seems difficult to design a protocol whose security against active adversaries is equivalent to 1MDLP or DTDLP. The obstruction is that the DLP does not have any obvious "trapdoor", and so it is unclear how to get a component of a protocol to behave as a DLP oracle. Many years ago Maurer and Yacobi [30] used a trapdoor version of DLP to set up an ID-based encryption system in which the trusted authority returned the discrete log of Alice's identification as her secret key. However, they had to assume that the group order remained secret, and for this reason the security of their system would disappear if an attacker were allowed to query the trusted authority with any number of her choice (rather than with an actual identification number verified by a supporting document). In fact, if an attacker is allowed to query a DLP oracle (even a small number of times), the group order will not remain secret. Namely, an attacker can set $X_j = g^j$ for a few large values of j and get the discrete logs x_j, after which she takes the g.c.d. of the $x_j - j$ to find the group order (which divides each of these differences).

Recently, Teske [35] developed a more promising way — based on Weil descent and isogenies of elliptic curves — to construct a trapdoor for the discrete log problem. Unlike Maurer and Yacobi, she does not assume that the group order is secret.

Open Problem 1. Design a protocol whose security against active attacks is equivalent to either 1MDLP or DTDLP.

Remark 4. In [14] the authors propose an on-line/off-line threshold signature scheme whose security against chosen-message attack is related to a problem that they call "one-more-r." It is easy to see that this problem as they describe it (Definition 9 of [14]) is equivalent to 1MDLP. However, a closer examination of their protocol reveals that the problem whose intractability is necessary for its security is actually DLP1, not 1MDLP. Thus, their protocol is a nice example of one whose security is dependent upon hardness of DLP1, but it does not provide a solution to the above open problem.

In this connection recall that DLP1 can be shown to be equivalent to DLP by an informal argument and by a non-tight reduction, whereas 1MDLP is probably an easier problem in some settings (see Table 1 below). Because Definition 9 of [14] gave the incorrect impression that the protocol in [14] is dependent upon hardness of 1MDLP, it appeared that it was still an open problem to design a DLP-based on-line/off-line threshold signature scheme. An unfortunate result of this misunderstanding was that the authors of [11] expended considerable effort in order to fill this apparent gap — although in reality no such gap existed.

4. DLP and DHP Algorithms

4.1. **Index calculus.** In index calculus algorithms to find discrete logarithms what is most time-consuming is, first, the generation of relations by finding elements that are smooth with respect to a factor base and, second, the determination of the discrete logs of the elements of the factor base by linear algebra. The factor base is chosen to have a size that optimizes the algorithm, essentially by equating the running times of these two phases.

4.2. **The DLP in $\mathbb{F}_p^*$.** We first show that in the multiplicative group of a prime field $\mathbb{F}_p$ the DTDLP is easier to solve than the DLP if one is using naive (i.e., pre-number-field-sieve) index calculus. The general idea of using the oracle in a "one-more" problem to speed up index calculus is due to Joux, Naccache, and Thomé [22]; in fact, it was their paper that prompted us to think that the jacobian groups of low-genus curves might allow us to separate the difficulty levels of some of the eight problems in §2.

Following [31], we give a brief summary of the naive index-calculus algorithm in $\mathbb{F}_p^*$ and a rough estimate of its running time. Let g be a generator of $\mathbb{F}_p^*$, and let $L(p)$ denote $\exp(\sqrt{\ln p \ln\ln p})$. The factor base consists of the first $m = L(p)^c$ primes, where c will be chosen later. To generate relations, one chooses random j and hopes that the least nonnegative residue of g^j is p_m-smooth (that is, has no prime factor $> p_m$), where p_m is the m-th prime. The standard estimate for the number of j's that will have to be tried before one expects to find a p_m-smooth value of g^j is u^u, where $u = \ln p/\ln p_m \approx \ln p/\ln m$ (since $\ln p_m$ and $\ln m$ differ only by a term of order $\ln\ln p$, and in this discussion we ignore all lower-order terms). We have

$$\begin{aligned} u^u =& \exp(u\ln u) \approx \exp\left(\frac{\ln p}{\ln m}(\ln\ln p - \ln\ln m)\right) \\ \approx& \exp\left(\frac{\ln p}{c\sqrt{\ln p \ln\ln p}}(\ln\ln p - \ln\sqrt{\ln p})\right) \\ =& L(p)^{1/(2c)}. \end{aligned}$$

For each value g^j we can use Lenstra's elliptic curve factorization algorithm (see [28]) to quickly test it for p_m-smoothness. Because the running time of that factorization method is roughly $L(q)$, where q is the prime factor that is found by an iteration of the algorithm, it follows that this time will be of a lower order of magnitude than $L(p)$ and can be neglected. Thus, we may suppose that it takes $u^u \approx L(p)^{1/(2c)}$ operations to find a relation.

The number of operations required to generate slightly more than m relations is then $\approx L(p)^{c+\frac{1}{2c}}$, after which the sparse linear algebra stage requires $\approx m^2 \approx L(p)^{2c}$ operations. Equating these two running times gives $c = \frac{1}{2}\sqrt{2}$, which also happens to be the value of c that minimizes the exponent $c + \frac{1}{2c}$ in the running time for the first phase. Thus, this value $c = \frac{1}{2}\sqrt{2}$ is optimal for two reasons. The resulting time estimate for the whole algorithm is then $L(p)^{\sqrt{2}}$.

4.3. **The DTDLP in $\mathbb{F}_p^*$.** In this case we can find the discrete logs of the elements of the factor base through oracle queries that each take unit time; the total time in this phase is thus $m \approx L(p)^c$. We are then given an element Y and must find its discrete logarithm y. To do this we need only find a single relation expressing g^jY for some j in terms of the factor base. The total running time is then of order $L(p)^c + L(p)^{1/(2c)}$, which is optimal when $c = \frac{1}{2}\sqrt{2}$. This means that the DTDLP can be solved in time $L(p)^{\sqrt{2}/2}$ — in other words, the number of operations needed to solve the DLP is roughly the square of the number needed to solve the DTDLP.

Of course, it is the number field sieve, and not naive index-calculus, that gives the best algorithm available for the DLP in $\mathbb{F}_p^*$. It would be useful to make the comparison between the DTDLP and the DLP using the number field sieve, just as Joux-Naccache-Thomé did for the RSA e-th root problem in [22]. Most likely the result would be the same as in [22], namely, the exponent in the expression for the running time for the DTDLP would be lower than that for the DLP by a factor of $\sqrt[3]{2}$.

Open Problem 2. Compare the optimized number field sieve for the DLP and the DTDLP in finite fields, and compare implementations with realistic parameters, as was done in [22] for the RSA e-th root problem.

We have seen that in finite fields we are able to separate the DTDLP from the DLP. However, we do not know of any way to get an index-calculus algorithm for the 1MDLP in a finite field that is faster than for the DLP. We leave this as an open problem.

Open Problem 3. Find an algorithm for the 1MDLP in some family of finite fields that is faster than any available algorithm for the DLP in those fields.

4.4. **The DLP in a jacobian group.** For the remainder of this section the group G will be a subgroup of prime order $n \approx q^g$ of the jacobian group of a genus-g hyperelliptic curve over the field of q elements. Elements of G can be uniquely represented as so-called "reduced divisors" (a, b), where a is a monic polynomial of degree at most g and b is a polynomial of degree less than $\deg(a)$ satisfying a certain equation. In the case $g = 1$ we can identify monic linear polynomials with field elements a, and in this case a reduced divisor (a, b) is simply a point on the elliptic curve in the usual sense. In this paper g will be small, say $1 \leq g \leq 6$.

We now give an overview of index-calculus algorithms to solve the Discrete Log Problem (DLP) on G. We first choose a factor base FB consisting of a subset of $|\mathrm{FB}| \approx q^\alpha$ degree-one divisors (that is, divisors (a, b) for which a has degree one). Here $\alpha \leq 1$ is chosen later so as to optimize the running time. Let "Prob" denote the probability that a random element $(a, b) \in G$ is FB-smooth, that is, a splits over the ground field into a product of linear factors from FB. (If a is such a product, then it is easy to write the divisor (a, b) in terms of the factor base.) Up to a constant (that depends on g) one has $\mathrm{Prob} \approx q^{g\alpha}/n$, and so the reciprocal of this probability is $\approx n^{1-\alpha} \approx q^{g(1-\alpha)}$.

The relations-generation phase of index-calculus for the DLP requires us to find approximately $|\mathrm{FB}|$ random elements of G that are FB-smooth, and this takes time roughly

$$\frac{|\mathrm{FB}|}{\mathrm{Prob}} \approx q^{g-\alpha(g-1)}. \tag{2}$$

Once the relations are found, the other time-consuming part is the sparse linear algebra phase that finds the discrete logs of the factor base. That takes time roughly $|\mathrm{FB}|^2 \approx q^{2\alpha}$. To optimize the running time, we equate the running times $|\mathrm{FB}|^2$ and $|\mathrm{FB}|/\mathrm{Prob}$ of the two major components and solve for α, obtaining $\alpha = g/(g+1) = 1 - \frac{1}{g+1}$. This leads to a running time of roughly $q^{2-2/(g+1)}$ for the entire DLP algorithm. (This algorithm is due to Pierrick Gaudry [20] and to unpublished work of Robert Harley.)

This is close to the fastest running time in the literature. However, for $g \geq 2$ the state-of-the-art algorithm uses a "double large prime variant" [16] which lowers the running time to $q^{2-2/g}$. We shall use this as our estimate of the time required to find discrete logs, except in the case $g = 1$, where there are no known index-calculus methods that are as fast as Pollard-rho, which takes time of order $q^{1/2}$.

Remark 5. In [15] (see also [17]) Diem gives a DLP algorithm with running time roughly $q^{2-2/(g-1)}$ for "sufficiently general" non-hyperelliptic curves of genus $g \geq 3$. In the present paper we are focusing on hyperelliptic curves, where much more work has been done than for non-hyperelliptic curves. An examination of the best available algorithms in the latter case would give a similar comparison to that in Table 1 below. Indeed, for fixed genus g in the non-hyperelliptic case all of the entries in the corresponding row of Table 1 would simply have to be multiplied by the factor

$$\frac{2-\frac{2}{g}}{2-\frac{2}{g-1}} = 1 + \frac{1}{g(g-2)}.$$

This means that the comparisons given below between the different problems in §2 carry over to non-hyperelliptic curves as well.

In Table 1 the non-hyperelliptic case is included for genus $g = 3$, because Benjamin Smith [33] developed a method using isogenies (that apparently works only for $g = 3$) that for many hyperelliptic curves allows one to transfer the DLP to a non-hyperelliptic curve. Thus, for $g = 3$ the algorithm in [15] is the fastest one for a substantial set of hyperelliptic curves.

4.5. **Index calculus for non-standard DLP on a jacobian group.** We next consider the DTDLP. In this case one finds the discrete logs of the elements of FB through oracle queries that each take unit time, after which one must find the discrete log of some element Y. By adding to Y random known multiples of the generator one finds an FB-smooth element in time 1/Prob, after which one immediately gets the discrete log of Y. Thus, we optimize by setting $|\mathrm{FB}| = 1/\mathrm{Prob}$, which is algebraically the same as what we did in §4.1, i.e., once again $\alpha = 1 - \frac{1}{g+1}$. But now — thanks to not having to do linear algebra — the running time is much less than it was for the DLP; it is of order $q^{\alpha} = q^{g/(g+1)} \approx n^{1/(g+1)}$.

Finally, we consider the 1MDLP. We now must find the discrete logs of $|\mathrm{FB}|+1$ elements Y_i after making $|\mathrm{FB}|$ discrete log queries. The running time is dominated by the time it takes to add random multiples of the generator to each of the Y_i until an FB-smooth element is found. This is roughly $|\mathrm{FB}|/\mathrm{Prob} \approx q^{g-\alpha(g-1)}$ by (2), and this is optimal when $\alpha = 1$. Thus, for the 1MDLP the running time is of order $q \approx n^{1/g}$.

4.6. **Brown–Gallant–Cheon for non-standard DHP on a jacobian group.** Leaving the realm of index-calculus algorithms, we also have need of an algorithm due to Brown-Gallant [12] and Cheon [13] that, given a one-sided Diffie-Hellman oracle of the type in 1MDHP and DTDHP, finds the discrete logarithm of X (and hence solves either of the two problems) in time $\approx n^{1/3}$. It should be noted that this running time applies only in the case when $n-1$ has a factor u of order $n^{1/3}$ (of which there is a significant probability for random primes n) and when at least u queries are made.

4.7. **Relative running times.** The approximate running times of these algorithms are summarized in Table 1, where we have chosen q and $n \approx q^g$ so that the DLP requires time $\approx 2^{80}$ using the best available algorithms.

genus	$\log_2 q$	$\log_2 n$	$\log_2 n^{1/3}$	$\log_2 n^{1/(g+1)}$
	(1MDLP)		(1MDHP or DTDHP)	(DTDLP)
1	160 (80)	160	53	80
2	80	160	53	53
3	60	180	60	45
3 non-hyper	80	240	80	60
4	54	216	72	43
5	50	250	83 (80)	42
6	48	288	96 (80)	41

TABLE 1. Estimated running times of algorithms in jacobian groups having 80 bits of DLP security.

Notes: (i) The running time estimates in Table 1 ignore multiplicative constants. These constants do not appear to be large enough to affect the general conclusions we draw from Table 1; nevertheless, extensive experimentation is needed before these conclusions can be accepted with certainty. (ii) For $g = 1$ the value of q is determined by setting $q^{1/2} \approx 2^{80}$; for non-hyperelliptic genus-3 curves we set $q \approx 2^{80}$; and in all other cases we set $q^{2-2/g} \approx 2^{80}$. (iii) The 80 in parentheses indicates that the DLP algorithm, requiring time $\approx 2^{80}$, would be faster than the one described above for 1MDLP, 1MDHP, or DTDHP whose running time is given in the column.

From the table we see that using state-of-the-art algorithms the difficulty levels of the problems satisfy the following inequalities:

$$\text{1MDLP} > \text{1MDHP or DTDHP for } g = 1, 2$$

$$\text{1MDHP or DTDHP} > \text{1MDLP for } g = 4, 5, 6$$

$$\text{DTDLP} > \text{1MDHP or DTDHP for } g = 1$$

$$\text{1MDHP or DTDHP} > \text{DTDLP for } g = 3, 4, 5, 6$$

$$\text{1MDLP} > \text{DTDLP for } g = 2, 3, 4, 5, 6.$$

Remark 6. To the best of our knowledge, the DTDLP is the only DLP-type problem known to be easier on genus-2 than on genus-1 curves using current algorithms.

Remark 7. In Table 1 we're assuming that the prime order n of the subgroup G of the jacobian group is $\approx q^g$. However, in a DSA-type application one chooses n just to be large enough so that generic DLP algorithms on G require time at least 2^{80}; in other words, one chooses $n \approx 2^{160}$. If this is done in genus $g \geq 3$, then the prime n is $\ll q^g$. In that case the running times for 1MDLP and DTDLP, which depend on q and g but not n, are unaffected, whereas the log of the running time for 1MDHP and DTDHP, which depends directly on n, is always $\log_2 n^{1/3} \approx 53$. Thus, in the DSA setting we have 1MDLP > 1MDHP/DTDHP > DTDLP for genus-3 hyperelliptic curves, and 1MDLP > DTDLP > 1MDHP/DTDHP in the non-hyperelliptic genus-3 case. Furthermore, for $g = 4, 5, 6$ we see that 1MDLP is of comparable difficulty to 1MDHP or DTDHP, whereas the latter two problems are harder than DTDLP.

Remark 8. In order to separate the difficulty of DHP from that of the Decision Diffie-Hellman Problem one needs to look at the so-called "gap groups," which are extremely rare among elliptic curve or jacobian groups. Indeed, in recent years much effort has been devoted to constructing examples of such groups. In contrast, the groups we are using in order to show the likely incomparability of some of the problems in §2 are vast families with no restrictions on the ground field or on the coefficients of the defining equations.

5. Conclusion

There are many issues that arise in interpreting reductionist "proofs" of security of a protocol, some of which we discussed in [24, 25, 26]. Of all those questions the most obvious one is: What is the true level of difficulty of the underlying problem that the reduction connects to the adversary's task? Without at least a partial analysis of this question, the security reduction tells us very little. Our results in this paper should be viewed as only a small part of the work in this area that needs to be done.

It seems likely that DLP1 is equivalent in difficulty to DLP in practice, but based on our examination of low-genus jacobian groups we believe that both 1MDLP and DTDLP are strictly easier than DLP. Similarly, we conjecture that DHP1 is equivalent in difficulty to DHP, but that both 1MDHP and DTDHP are easier than DHP. We further conjecture that IMDLP is incomparable to 1MDHP, and that DTDLP is incomparable to DTDHP.

We also believe that security proofs that reduce these non-standard versions to successful attacks should be regarded as giving a weaker assurance about security than a reduction of DLP or DHP itself, and that this is still the case even when the latter reduction uses the random oracle assumption and the reduction from the non-standard problem does not.

References

[1] M. Abdalla and D. Pointcheval, Interactive Diffie-Hellman assumptions with applications to password-based authentication, *Financial Cryptology '05*, LNCS 3570, Springer-Verlag, 2005, pp. 341-356.

[2] J. Baek, R. Safavi-Naini, and W. Susilo, Universal designated verifier signature proof (or How to efficiently prove knowledge of a signature), *Progress in Cryptology – Asiacrypt 2005*, LNCS 3788, Springer-Verlag, 2005, pp. 644-661.

[3] M. Bellare, C. Namprempre, and G. Neven, Security proofs for identity-based identification and signature schemes, *Advances in Cryptology – Eurocrypt 2004*, LNCS 3027, Springer-Verlag, 2004, pp. 268-286.

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

[5] M. Bellare and G. Neven, Transitive signatures: new schemes and proofs, *IEEE Transactions on Information Theory*, **51** (2005), pp. 2133-2151.

[6] 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.

[7] M. Bellare and S. Shoup, Two-tier signatures, strongly unforgeable signatures, and Fiat-Shamir without random oracles, *Public Key Cryptography – PKC 2007*, LNCS 4450, Springer-Verlag, 2007, pp. 201-216.

[8] 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.

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

[10] D. Boneh, B. Lynn, and H. Shacham, Short signatures from the Weil pairing, *Advances in Cryptology – Asiacrypt 2001*, LNCS 2248, Springer-Verlag, 2001, pp. 514-532.

[11] E. Bresson, D. Catalano, and R. Gennaro, Improved on-line/off-line threshold signatures, *Public Key Cryptography – PKC 2007*, LNCS 4450, Springer-Verlag, 2007, pp. 217-232.

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

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

[14] C. Crutchfield, D. Molnar, D. Turner, and D. Wagner, Generic on-line/off-line threshold signatures, *Public Key Cryptography – PKC 2006*, LNCS 3958, Springer-Verlag, 2006, pp. 58-74.

[15] C. Diem, An index calculus algorithm for plane curves of small degree, *Algorithmic Number Theory – ANTS VII*, LNCS 4076, Springer-Verlag, 2006, pp. 543-557.

[16] C. Diem, P. Gaudry, N. Thériault, and E. Thomé, A double large prime variation for small genus hyperelliptic index calculus, *Mathematics of Computation*, **76** (2007), pp. 475-492.

[17] C. Diem and E. Thomé, Index calculus in class groups of non-hyperelliptic curves of genus three, *Journal of Cryptology*, to appear.

[18] D. Freeman, Pairing-based identification schemes, technical report HPL-2005-154, Hewlett-Packard Laboratories, 2005.

[19] D. Galindo, Practice-oriented provable security – the case of pairing based cryptographic schemes, presentation at the International Workshop on Pairings in Cryptography, Dublin, Ireland, 2005. Slides available from http://pic.computing.dcu.ie/talks/DavidPic2005.pdf.

[20] P. Gaudry, An algorithm for solving the discrete log problem on hyperelliptic curves, *Advances in Cryptology – Eurocrypt 2000*, LNCS 1807, Springer-Verlag, 2000, pp. 19-34.

[21] R. Gennaro, D. Leigh, R. Sundaram, and W. Yerazunis, Batching Schnorr identification scheme with applications to privacy-preserving authorization and low-bandwidth communication devices, *Progress in Cryptology – Asiacrypt 2004*, LNCS 3329, Springer-Verlag, 2004, pp. 276-292.

[22] A. Joux, D. Naccache, and Thomé, When e-th roots become easier than factoring, *Progress in Cryptology – Asiacrypt 2007*, LNCS 4833, Springer-Verlag, 2007, pp. 13-28.

[23] A. Joux and K. Nguyen, Separating Decision Diffie–Hellman from Computational Diffie–Hellman in cryptographic groups, *Journal of Cryptology*, **16** (2003), pp. 239-247.

[24] N. Koblitz and A. Menezes, Another look at "provable security," *Journal of 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 'provable security.' II, *Progress in Cryptology – Indocrypt 2006*, LNCS 4329, Springer-Verlag, 2006, pp. 148-175.

[27] K. Kurosawa and S.-H. Heng, From digital signature to ID-based identification/signature, *Public Key Cryptography – PKC 2004*, LNCS 2947, Springer-Verlag, 2004, pp. 248-261.

[28] H. W. Lenstra, Jr., Factoring integers with elliptic curves, *Annals of Math.*, **126** (2) (1987), pp. 649-673.

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

[30] U. Maurer and Y. Yacobi, Non-interactive public-key cryptography, *Advances in Cryptology – Eurocrypt '91*, LNCS 547, Springer-Verlag, 1991, pp. 498-507.

[31] K. McCurley, The discrete logarithm problem, in C. Pomerance, ed., *Cryptology and Computational Number Theory*, Amer. Math. Soc., 1990, pp. 49-74.

[32] C. Schnorr, Efficient signature generation for smart cards, *Journal of Cryptology*, **4** (1991), pp. 161-174.

[33] B. Smith, Isogenies and the discrete logarithm problem in jacobians of genus 3 hyperelliptic curves, *Advances in Cryptology – Eurocrypt 2008*, LNCS 4965, Springer-Verlag, 2008, pp. 163-180.

[34] M. Szydlo, A note on chosen-basis decisional Diffie-Hellman assumptions, *Financial Cryptology '06*, LNCS 4107, Springer-Verlag, 2006, pp. 166-170.

[35] E. Teske, An elliptic curve trapdoor system, *Journal of Cryptology*, **19** (2006), pp. 115-133.

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`