On the Static Diffie-Hellman Problem on Elliptic Curves over Extension Fields

Robert Granger[*]

Claude Shannon Institute
School of Computing, Dublin City University
Glasnevin, Dublin 9, Ireland
rgranger@computing.dcu.ie

Abstract. We show that for any elliptic curve $E(\mathbb{F}_{q^n})$, if an adversary has access to a Static Diffie-Hellman Problem (Static DHP) oracle, then by making $O(q^{1-\frac{1}{n+1}})$ Static DHP oracle queries during an initial learning phase, for fixed $n > 1$ and $q \to \infty$ the adversary can solve *any* further instance of the Static DHP in *heuristic* time $\tilde{O}(q^{1-\frac{1}{n+1}})$. Our proposal also solves the *Delayed Target DHP* as defined by Freeman, and naturally extends to provide algorithms for solving the *Delayed Target DLP*, the *One-More DHP* and *One-More DLP*, as studied by Koblitz and Menezes in the context of Jacobians of hyperelliptic curves of small genus. We also argue that for *any* group in which index calculus can be effectively applied, the above problems have a natural relationship, and will *always* be easier than the DLP. While practical only for very small n, our algorithm reduces the security provided by the elliptic curves defined over $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^4}$ proposed by Galbraith, Lin and Scott at EUROCRYPT 2009, should they be used in any protocol where a user can be made to act as a proxy Static DHP oracle, or if used in protocols whose security is related to any of the above problems.

1 Introduction

In recent years, there has been a steadily growing appreciation of the existence of an apparent separation, in some cases, between the hardness of breaking discrete logarithms in a particular group, and the hardness of solving in that group certain problems to which the security of a cryptosystem is provably related. This situation can arise when auxiliary information is provided to an attacker in the form of limited access to a particular oracle, either within the game played by the attacker in the security proof, or in practice when a user can be made to act as a proxy oracle by virtue of the nature of the protocol itself.

For example, in 2004 Brown and Gallant studied the *Static Diffie-Hellman Problem* (Static DHP), in which a party reuses the same Diffie-Hellman secret

[*] Research supported by the Claude Shannon Institute, Science Foundation Ireland Grant No. 06/MI/006.

for multiple Diffie-Hellman key agreements [6]. The authors proved that if the associated DLP for the static secret is hard, then so is the Static DHP. However, their reduction naturally becomes an algorithm for solving the DLP if an attacker has access to a Static DHP oracle. In the protocols [15, 17] and [7] for instance, a user can indeed be made to act as a proxy Static DHP oracle, thus rendering such systems vulnerable to this attack. In the best case (from an attacker's perspective), one can compute a static Diffie-Hellman secret in a group of order r with only $O(r^{1/3})$ Static DHP oracle queries and $O(r^{1/3})$ group operations [6]. For cryptographically interesting elliptic curves, i.e., those for which generic attacks are the best known, this result is in stark contrast to the time required to compute discrete logarithms, namely $O(r^{1/2})$. So while solving the Static DHP in this case may still be hard, it has lower complexity than the best DLP algorithms.

Koblitz and Menezes have shown that several other problems exhibit a similar apparant hardness separation[1] from the DLP, in the context of Jacobians of hyperelliptic curves of small genus [38]; namely the *Delayed Target DHP* [18], the *Delayed Target DLP* [38], the *One-More DHP* [3] and the *One-More DLP* [1, 2]. For each of these problems, it is the use of oracle queries that creates these separations. For instance, for the *Delayed Target DHP*, the Brown-Gallant algorithm can be applied immediately, since the game played by the attacker gives him initial access to a Static DHP oracle.

In 2006 Cheon rediscovered the Brown-Gallant algorithm when the requisite information is provided in the guise of the Strong Diffie-Hellman Problem (Strong DHP) [8]. Cheon also extended the attack to utilise divisors of $r+1$ as well as of $r-1$, as with the Brown-Gallant algorithm; indeed both algorithms can be regarded as instances of the well-known reduction from the DLP to the DHP due to den Boer, Maurer, Wolf *et al.,* (see [42] for a survey), but with restricted access to a DHP oracle. Incidently, Cheon's break of the Strong DHP does not in itself reveal any weakness in the protocols that depend upon it, since reductions given in security proofs until that time were in the wrong direction, i.e., they showed that breaking the system enables one to solve the Strong DHP, but not the other way around. Hence, if an algorithm that efficiently solves the Strong DHP is found, then while all security proofs that assume its hardness would no longer provide any security assurance, no actual break of the associated systems would result. In the case of Boneh-Boyen signatures [4], which relies on the Strong DHP in the above manner, Jao and Yoshida have given a reduction in the reverse direction, thus strengthening the proof of security for these signatures, and at the same time providing an attack on the scheme with complexity $O(r^{2/5+\epsilon})$, if $O(r^{1/5+\epsilon})$ signature queries are permitted to be performed [32]. This result suggests that if one can establish an equivalence between a given protocol and a problem that exhibits an apparent hardness separation from the DLP, then in some attack models the security assurances provided by these arguments will likely be lower than that provided by the DLP.

[1] We use the prefix *apparent*, since these separations exist only relative to the current understanding of the respective problems, which could of course change.

Similarly, for the RSA problem, in 2007 Joux, Naccache and Thomé showed that with initial subexponential access to an e-th root oracle an attacker can later compute the e-th root of any element with complexity lower than that required to factor the modulus [36]. This algorithm was then adapted to solve the oracle-assisted Static DHP in finite fields [34], with similar efficiency improvements, demonstrating an apparent hardness separation in this case also.

It is therefore natural to ask whether initial access to a Static DHP oracle can aid in solving later Static DHP instances faster than solving the DLP, in the context of elliptic curves? As previously mentioned, in the best case the Brown-Gallant-Cheon algorithm requires $O(r^{1/3})$ oracle queries and group operations. However, for elliptic curves defined over extension fields $\mathbb{F}_{q^n}$, we present an algorithm which for fixed $n > 1$ and $q \to \infty$ requires $O(q^{1-\frac{1}{n+1}})$ oracle queries and has *heuristic* time complexity $\tilde{O}(q^{1-\frac{1}{n+1}})$. This should be compared with the best known DLP algorithm for these curves which has complexity $\tilde{O}(q^{2-2/n})$ [24], hence our proposal approaches being a square-root faster than the DLP with increasing n. Note that for $n = 2$ our complexity is the same as the best-case Brown-Gallant-Cheon complexity, but applies to all elliptic curves over $\mathbb{F}_{q^2}$ and not just those with appropriate divisors of $r \pm 1$, while for $n > 2$ our result is superior. We also present an *heuristic* subexponential oracle-assisted Static DHP algorithm for elliptic curves over a special family of extension fields.

Our proposal also naturally extends to provide algorithms for solving the four problems studied by Koblitz and Menezes in [38]. In this work it was found that the relationships between the hardness of these problems do not appear to behave as one might expect (cf. [39, 40]). We correct a minor oversight in the analysis and argue that in the context of *any* group in which index calculus is effective, i.e., one in which index calculus provides the best known algorithm to solve the given problems — which includes Jacobians of hyperelliptic curves and elliptic curves over extension fields — the aforementioned problems do indeed have natural relationships, and are *always* easier than the DLP (this statement naturally only applies with respect to the state of the art in index calculus algorithms). However a central conclusion of [38], namely that it is difficult to assess what security assurances are provided by security proofs when the games played are interactive or have complicated inputs, still holds.

Due to the fact that the implicit constant in the complexity of our algorithm grows very quickly with n, it is practical only for small values of n, namely $n = 2, 3$ or 4 (and whenever n is divisible by $2, 3$ or 4). However, based on the results of timing estimates arising from an implementation of the components of the attack, the security provided by the elliptic curves defined over $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^4}$ proposed by Galbraith, Lin and Scott at EUROCRYPT 2009, would be significantly reduced, should these curves be used in any protocol where a user can be made to act as a proxy Static DHP oracle, or if used in protocols whose security is related to any of the problems studied in [38].

Independently of this work, Joux and Vitse [37] have proposed the same basic algorithm as the one presented here for the oracle-assisted Static DHP, namely Heuristic Result 1. Although Joux and Vitse did not consider our factor base reduction method that leads to Heuristic Result 2, their main idea improves upon Gaudry's relation finding technique — which is used in the present work — enabling one to find relations for elliptic curves over degree five extension fields, which is currently impractical with Gaudry's method.

One goal of the present work was to assess the impact the algorithms presented herein had upon the oracle-assisted Static DHP on the Oakley Well-known Groups 3 and 4, which form part of the IPSEC set of protocols [31] (see §4 and §5), and which are defined over degree five extension fields in characteristic two. Noting from our results how much easier it is to find relations in characteristic two than for large prime characteristic, the author contacted the authors of [37] in order to apply their idea to these curves. Using the results of this paper and [37], we recently announced the experimental verification of the feasibility of solving the oracle-assisted Static DHP on the 152-bit Group 3 curve [28] over $\mathbb{F}_{2^{155}}$, which has long been suspected of weakness, but had until now resisted many attacks [49, 25, 22, 44].

The sequel is organised as follows. In §2 we recall the Static DHP. In §3 we motivate our main idea, present our basic algorithm, and analyse asymptotic variants of it. Then in §4 we detail curves in the literature that are vulnerable to our attack. In §5 we give a full account of our experimental implementation at the 128-bit security level for extension degrees $n = 2, 3, 4$ and 5, over large prime and characteristic two fields, assess their impact on the above curves and report on the Oakley Group 3 curve. In §6 we present algorithms for three other problems which arise in cryptographic protocols and analyse their impact, and make some concluding remarks.

2 The Static Diffie-Hellman Problem

Let $\mathbb{G}$ be a cyclic group of prime order r, and let g be a fixed generator of $\mathbb{G}$. The classical Diffie-Hellman problem in $\mathbb{G}$ can be stated as follows [12]:

Problem 1. (DHP): Given g and random g^x and g^y, find g^{xy}.

In Diffie-Hellman (DH) key agreement between two parties, Alice chooses a random secret $x \in \mathbb{Z}/r\mathbb{Z}$ and computes g^x, while Bob chooses a random secret $y \in \mathbb{Z}/r\mathbb{Z}$ and computes g^y, which are then exchanged. Upon receipt each party computes the shared secret g^{xy} by exponentiating the other party's group element by their own secret. A fundamental security requirement of DH key agreement is that the DHP should be hard.

Should Alice for any reason repeatedly reuse the same secret, $x = d$ say, then the resulting set of DHP problem instances forms a tiny subset of all problem instances featured in the DHP, and it is thus not *a priori* clear that these instances should be hard, even if the DHP is hard. This problem is referred to as the Static DHP$_d$, which we state as follows:

Problem 2. (Static DHP$_d$): Given fixed g and g^d, and random g^y, find g^{dy}.

Observe that this situation need not just arise as an efficiency measure during multiple DH key agreements — Alice need only compute g^d once and reuse this value for multiple key agreements — but also arises in text-book El-Gamal encryption [15], Ford-Kaliski key retrieval [17] and Chaum-Van Antwerpen's undeniable signatures [7]. As mentioned in §1, Brown and Gallant have shown that if the associated DLP is hard, then so is the Static DHP$_d$ [6]. However, in the above three protocols, one of the system entities acts as a Static DHP$_d$ oracle, thus turning the Brown-Gallant reduction into an attack.

As in [6] we define an oracle for solving the Static DHP$_d$ as follows:

Definition 1. *(Static DHP$_d$ Oracle). Let $\mathbb{G}$ be a cyclic group of prime order r, written additively. For a fixed base element $P \in \mathbb{G}$ and a fixed element $Q \in \mathbb{G}$ let $d \in \mathbb{Z}/r\mathbb{Z}$ be such that $Q = dP$. Then a Static DHP$_d$ oracle (with respect to $\mathbb{G}$) computes the function $\delta : \mathbb{G} \rightarrow \mathbb{G}$ defined by:*

$$\delta(X) = dX.$$

Likewise a Static DHP$_d$ algorithm is said to be *oracle-assisted* if during an initial learning phase, it can make a number of Static DHP$_d$ queries, after which, given a previously unseen challenge element X, it outputs dX. We now consider how to solve the oracle-assisted Static DHP when $\mathbb{G} = E(\mathbb{F}_{q^n})$.

3 An Oracle-Assisted Static DHP Algorithm for $E(\mathbb{F}_{q^n})$

In this section we motivate and present our algorithm for solving the oracle-assisted Static DHP$_d$ in the present context.

The key observation in [36] is that if one is able to define a suitable 'factor base' in the group under consideration, i.e., a relatively small subset of group elements over which a non-negligible proportion of all group elements can be 'factored' via the group operation, then it is possible to solve the Static DHP$_d$ with input an arbitrary group element, given knowledge of the action of the Static DHP$_d$ oracle on the factor base elements alone. This follows from the simple fact that if in an additively written group $\mathbb{G}$ we have $R = P_1 + \cdots + P_n$, with P_i in some factor base $\mathcal{F}$, then

$$\delta(R) = dR = dP_1 + \cdots + dP_n = \delta(P_1) + \cdots + \delta(P_n).$$

Note that if an arbitrary group element R is not expressible over the factor base, then by adding a random element $Q \in \mathcal{F}$ (or any linear combination thereof) to R and testing expressibility, one can produce an element $R + Q$ which factors over $\mathcal{F}$, thus permitting the Static DHP$_d$ to be solved as before. Therefore a good factor base over which a non-negligible proportion of elements may be expressed, combined with randomisation, enables one to solve the Static DHP$_d$ for arbitrary group elements.

Observe that for the oracle-assisted Static DHP$_d$, one does not ever need to know d in order to compute the action of multiplication by d on an arbitrary element of $\mathbb{G}$, i.e., one can solve the Static DHP$_d$ without solving the DLP. This is because implicit information is 'leaked' via the Static DHP$_d$ oracle queries which enables one to solve the Static DHP$_d$ using the above observations, more readily than one is able to solve the DLP, for which there is no such information. This idea is central to both [34] and [38].

When $\mathbb{G}$ is the multiplicative group of a finite field, the problem of how best to construct a factor base, and how to express arbitrary elements over such a factor base is well studied [35, 33, 34]. For finite fields there exists a natural notion of size for elements, or equivalently a norm function, given by either the absolute value of an element for prime fields, or the degree of an element for extension fields, or a combination of both depending on the algorithm being used to generate multiplicative relations. A norm function imbues a notion of smoothness for a group and those elements of small norm generate more group elements than those elements of larger norm, hence the best choice for a factor base is those elements of norm up to some bound.

In the context of elliptic curves over prime fields, there does not appear to be a utilisable notion of norm that enables the selection of a factor base that generates a higher proportion of group elements than any other, nor a means by which to factor elements over one should one be chosen. It is precisely this issue that has so far precluded the discovery of a successful native index calculus algorithm for computing discrete logarithms on such curves[2], which is why they are so attractive from a security perspective.

For elliptic curves over extension fields, the story is very different. While the 'Weil descent' methodology [19, 25, 30] has proven successful for solving or weakening the DLP in some cases, this involves mapping to a generally larger group, which although possessing a natural factor base, does not allow the requisite Static DHP oracle queries to be made on the preimages of the factor base elements, since in general such preimages will not exist. There does however exist a notion of smoothness for such elliptic curves, as remarkably discovered by Gaudry [24].

3.1 Gaudry's Insight

Developing upon an intriguing idea due to Semaev [47], in 2004[3] Gaudry showed how to define a useful factor base for $E(\mathbb{F}_{q^n})$, over which elements can be 'factored', or more properly, decomposed, which leads to an index calculus algorithm for computing logarithms over these curves [24]. For fixed $n > 1$ and $q \to \infty$, the algorithm has heuristic complexity $\tilde{O}(q^{2-\frac{2}{n}})$, which is much faster than the Pollard rho complexity $\tilde{O}(q^{n/2})$.

We begin by recalling Semaev's *Summation Polynomials* [47].

[2] There are of course attacks that apply to a very small minority of elliptic curves [43, 20, 48, 46, 45], though these are well understood and are easily avoided, or in the case of pairing-based cryptography, which relies on curves which are susceptible to [43, 20], are employed.

[3] Gaudry's algorithm was initially posted to the IACR preprint server in 2004 (paper number 2004/073), but was not published until 2009, in [24].

Definition 2. *For* $\mathrm{char}(\mathbb{F}_q) > 3$ *let* E *be an elliptic curve defined over* $\mathbb{F}_{q^n}$ *by the equation* $y^2 = x^3 + ax + b$*. The summation polynomials* f_n *of* E *are defined by the following recurrence, with initial values for* $n = 2$ *and* 3 *given by* $f_2(X_1, X_2) = X_1 - X_2$*, and*

$$\begin{aligned} f_3(X_1, X_2, X_3) = {} & (X_1 - X_2)^2 X_3^2 - 2((X_1 + X_2)(X_1X_2 + a) + 2b)X_3 \\ & +((X_1X_2 - a)^2 - 4b(X_1 + X_2)), \end{aligned}$$

and for $n \geq 4$ *and* $1 \leq k \leq n - 3$*,*

$$f_n(X_1, \dots, X_n) = Res_X(f_{n-k}(X_1, \dots, X_{n-k-1}, X), f_{k+2}(X_{n-k}, \dots, X_n, X)).$$

While this definition may appear rather mysterious, Semaev derived the above formulae by insisting that f_n satisfies the following property, which relates f_n to the addition law on E.

Theorem 1. *(Semaev [47]). Let* E *be an elliptic curve over a field* k*,* $n \geq 2$ *and* f_n *its* n*-th summation polynomial. Let* $x_1, \dots, x_n$ *be* n *elements of an algebraic closure* $\overline{k}$ *of* k*. Then* $f_n(x_1, \dots, x_n) = 0$ *iff there exists an* n*-tuple* $(y_1, \dots, y_n)$ *of elements in* $\overline{k}$ *such that for all* i*,* $P_i = (x_i, y_i)$ *is a point on* E *and*

$$P_1 + \cdots + P_n = \mathcal{O}.$$

One can therefore see immediately that f_n provides an encoding for all sets of n points on a given curve whose sum is the identity element. For an elliptic curve E over a prime field $\mathbb{F}_p$, Semaev proposed setting the factor base to be the set be all points on E whose abscissa have magnitude less than $p^{1/n}$. Then one computes random multiples of some base point P, say $R_i = r_iP$, and attempts to write each such R_i as a sum of n points in the factor base. To do this one need only solve

$$f_{n+1}(x_1, \dots, x_n, x_{R_i}) = 0. \tag{1}$$

By symmetry, one heuristically expects this to be possible for a proportion $1/n!$ of points R_i, and when $O(p^{1/n})$ points that decompose have been found (the approximate size of the factor base) one can obtain their logarithms with respect to P via a sparse linear algebra elimination, which has complexity $\tilde{O}(p^{2/n})$. Finding the logarithm of an arbitrary group element is then easy. Therefore, if finding small roots of (1) were possible, for fixed $n \geq 5$ and $p \to \infty$ this algorithm would be faster than Pollard rho.

Unfortunately, finding such small roots, at least for more than two variables [9], appears hard. Gaudry's insight was to observe that for elliptic curves over $\mathbb{F}_{q^n}$, if one uses a factor base consisting of points with abscissae in the base field $\mathbb{F}_q$, then assuming the field of definition of the curve is $\mathbb{F}_{q^n}$, the Weil restriction of scalars of equation (1) from $\mathbb{F}_{q^n}$ to $\mathbb{F}_q$ forms an algebraic system of n equations in n indeterminates over $\mathbb{F}_q$, which is nearly always zero-dimensional and which can be solved via elimination theory [24]. Note that the above assumption crucially also ensures that the factor base elements do not form a subgroup. Using a 'double large prime variation' [26] this leads to a DLP algorithm with complexity $\tilde{O}(q^{2-\frac{2}{n}})$. We are now ready to present the basic version of our algorithm, in which we detail how this Weil restriction approach works.

3.2 Basic Oracle-Assisted Static DHP Algorithm

Let E be an elliptic curve whose field of definition is $\mathbb{F}_{q^n}$. We define a factor base $\mathcal{F}$ à la Gaudry [24] as follows:

$$\mathcal{F} = \{P = (x, y) \in E(\mathbb{F}_{q^n}) \mid x \in \mathbb{F}_q\}.$$

On heuristic grounds, one expects $|\mathcal{F}| \approx q$, see [24]. For each $P \in \mathcal{F}$ we make an oracle call to the Static DHP oracle, to give $\delta(P) = dP$.

For an arbitrary point $R \in E(\mathbb{F}_{q^n})$, the goal is to find dR. We attempt write R as a sum of n elements of $\mathcal{F}$, i.e.,

$$R = P_1 + \cdots + P_n.$$

By symmetry, one heuristically expects the proportion of elements expressible in such a way to be approximately $1/n!$. To perform this decomposition one uses Semaev's summation polynomial f_{n+1}, and attempts to solve

$$f_{n+1}(x_1, \ldots, x_n, x_R) = 0 \in \mathbb{F}_{q^n}. \tag{2}$$

Note that the expression on the left of equation (2) involves the defining coefficients of the curve E, and the abscissa x_R, all of which are in $\mathbb{F}_{q^n}$. Fix a polynomial basis $\{1, t, \ldots, t^{n-1}\}$ for the extension $\mathbb{F}_{q^n}/\mathbb{F}_q$. Then each one of the n coefficients of powers of t must be zero. Since each of the n abscissae x_i are in $\mathbb{F}_q$, equation (2) defines a variety with n equations in n indeterminates over $\mathbb{F}_q$, which one solves via a Grobner basis computation, see §3.3.

If there is a solution $(x_1, \ldots, x_n)$ to the system (2), then one needs to compute all 2^n possible combinations $\pm P_1 \pm \cdots \pm P_n$ for the corresponding ordinates in order to find the correct combination which sums to R. Then the solution to the Static DHP for R is immediate:

$$\delta(R) = dR = dP_1 + \cdots + dP_n,$$

where all the terms on the right hand side are already known, due to the oracle queries on $\mathcal{F}$.

As already discussed, if a solution does not exist, then one adds to R a random element $Q \in \mathcal{F}$ (or any linear combination thereof) and attempts to decompose this point once again. One expects this to succeed after approximately $n!$ attempts. When it does we have the following equation:

$$R + Q = P_1 + \cdots + P_n,$$

which implies that

$$\delta(R) = dR = dP_1 + \cdots + dP_n - dQ,$$

where again all the terms on the right hand side are already known. Hence our Static DHP$_d$ instance is solved.

3.3 Discussion

Our first observation is that the above algorithm and this discussion of it are entirely heuristic; however we believe that the algorithm and its complexity can be made completely rigorous using the results of Diem [10, 11], should one choose to do so, see §3.4.

Our second observation — which is fundamental to the complexity of the algorithm — is that in contrast to the DLP, there is no linear algebra elimination, since only a single relation is sought. So once the initial oracle querying phase is complete, the complexity of the algorithm depends only on the problem of computing one relation. We therefore analyse this cost now.

For $n \geq 3$, Semaev's summation polynomials $\{f_n\}$ are symmetric and are of degree 2^{n-2} in each variable. Hence equation (2) is of degree 2^{n-1} each variable. In order to simplify the system greatly, it pays to express f_{n+1} in terms of the elementary symmetric functions $e_1, \ldots, e_n$ in the variables $x_1, \ldots, x_n$. We then have a system of n equations in the n indeterminates $e_1, \ldots, e_n$ each of which again has degree bounded by 2^{n-1} in each variable. In order to solve this system, we perform a Gröbner basis computation.

In practice our experiments (see §5) showed that the Gröbner basis w.r.t the lexicographic ordering always satisfies the so-called shape lemma [27, 41], i.e., it is of the following form:

$$e_1 - g_1(e_n), e_2 - g_2(e_n), \ldots, e_{n-1} - g_{n-1}(e_n), g_n(e_n), \tag{3}$$

where $g_i(e_n)$ is a univariate polynomial in e_n for each i. In general the degree of the univariate polynomial in e_n that we obtain will be $2^{n(n-1)}$ and indeed in our experiments this is borne out. The complexity of Faugère's algorithm F4 [16] to compute this basis is therefore at least

$$\tilde{O}(Poly(2^{n(n-1)})).$$

Since this is doubly exponential in n, this makes the algorithm practical only for very small values of n. However for fixed n and $q \to \infty$, this is polynomial in $\log q$.

To find whether or not the system has roots $e_1, \ldots, e_n \in \mathbb{F}_q$, one extracts the linear factors of the univariate polynomial $g_n(e_n)$ using a gcd computation with $e_n^q - e_n$ followed by Cantor-Zassenhaus and then substitutes each $\mathbb{F}_q$ root e_n into $g_i(e_n)$ to find $e_{n-1}, \ldots, e_1$. For each such vector of $\mathbb{F}_q$ roots $(e_1, \ldots, e_n)$ one tests whether the polynomial

$$p(x) = x^n - e_1 x^{n-1} + e_2 x^{n-2} - \cdots - (-1)^n e_n \tag{4}$$

splits over $\mathbb{F}_q$. If it does then these roots are the abscissae of points in $E(\mathbb{F}_q)$, and there exists a linear combination

$$\epsilon_1 P_1 + \cdots + \epsilon_n P_n \tag{5}$$

with $\epsilon_i \in \{-1, 1\}$ which sums to R. This step is also polynomial in $\log q$.

On average one expects to have to perform $n!$ such decompositions in order to find a relation. Therefore the complexity of the our basic Static DHP algorithm for fixed $n > 1$ and $q \to \infty$ is polynomial in $\log q$. This gives the following heuristic result.

Heuristic Result 1. *For any elliptic curve $E(\mathbb{F}_{q^n})$, by making $O(q)$ queries to a Static DHP$_d$ oracle during an initial learning phase, for fixed $n > 1$ and $q \to \infty$, an adversary can solve any further instance of the Static DHP$_d$ in time $Poly(\log q)$.*

Note that prior to the learning phase, the adversary needs to construct the factor base by testing whether a given abscissa $x \in \mathbb{F}_q$ gives a point lying on E or not. We incorporate this computation into the learning phase, since it has the same complexity of $\tilde{O}(q)$. It is of course possible to balance the cost of the learning and relation-finding phases, which we now consider.

3.4 Balancing the setup and relation-finding costs

To balance the cost of the oracle querying phase and the relation finding phase, one needs to reduce the size of the factor base by some proportion. To this end, Let $|\mathcal{F}| = q^\alpha$, with $0 < \alpha \leq 1$. Then given the decomposition of a random point $R \in E$ as a sum of points whose abscissa are in $\mathbb{F}_q$, the probability that a single abscissa is in $\mathcal{F}$ is $q^{\alpha-1}$. Assuming these events are independent, the probability that all n abscissae are in $\mathcal{F}$ is $q^{n(\alpha-1)}$. Hence in order to obtain one relation, one expects to have to perform $1/q^{n(\alpha-1)} = q^{n(1-\alpha)}$ successful decompositions.

Asymptotically for fixed $n > 1$ and $q \to \infty$ one can regard the cost of a decomposition as unital (modulo some log factors) and hence to balance the two stages α must satisfy:

$$q^\alpha = q^{n(1-\alpha)},$$

and so $\alpha = n/(n+1) = 1 - \frac{1}{n+1}$. This gives the following heuristic result as stated in the abstract.

Heuristic Result 2. *For any elliptic curve $E(\mathbb{F}_{q^n})$, by making $O(q^{1-\frac{1}{n+1}})$ queries to a Static DHP$_d$ oracle during an initial learning phase, for fixed $n > 1$ and $q \to \infty$, an adversary can solve any further instance of the Static DHP$_d$ in time $\tilde{O}(q^{1-\frac{1}{n+1}})$.*

Observe that there is no possibility (nor necessity) for considering so-called large primes, i.e., those with absicissa in $\mathbb{F}_q$ but not lying in $\mathcal{F}$, since there is no linear algebra elimination step on the single relation. If we compare the above complexity to that obtained by Gaudry for the DLP, namely $\tilde{O}(q^{2-\frac{2}{n}})$, which uses a double large-prime variant, we see that our algorithm for solving the Static DHP approaches being a square root faster for increasing n. Intuitively this difference in complexity arises from there not being a linear algebra step in the solution of the Static DHP$_d$.

We note that Diem has given a rigorous algorithm that is essentially equivalent to Gaudry's DLP algorithm above [10], which for fixed $n \geq 2$ solves the

DLP on any elliptic curve over $\mathbb{F}_{q^n}$ in proven expected time $q^{2-2/n}(\log q)^{O(1)}$. We believe his treatment can be adapted *mutatis mutandis* to transform the above two heuristic results into theorems, though since it is not the primary focus of this paper, we have not verified this here.

Observe that in practice the limiting factor is not the decompositions, but the oracle queries, since these would typically be performed on a single server, whereas the former can be easily distributed. One can therefore reduce the number of such queries below the above threshold, at the expense of needing to perform more decompositions. Such a trade-off is easily optimised, based on the amount of computing power available, but will nevertheless require an exponential number of oracle queries, for fixed n and $q \to \infty$. We now consider how and when the number of oracle queries may be made subexponential in the size of the group.

3.5 Subexponential Oracle-Assisted Static DHP algorithm

Diem has also proven the following remarkable result [11]. For $n \to \infty$ and assuming $n = O(\sqrt{\log q})$, the DLP over any elliptic curve $E(\mathbb{F}_{q^n})$ can be solved in expected time $q^{O(1)} = e^{O(\log{(q^n)}^{2/3})}$. Thus for a family of finite fields, any elliptic curve DLP can be solved using a *native* subexponential index calculus algorithm.

While Diem is not precise in his analysis of the exponents in the complexity of the constituent parts of the algorithm, it is clear that since for the oracle-assisted Static DHP$_d$ there is no linear algebra step, one expects a similar improvement over the DLP algorithm in this context to the fixed n case, i.e., nearly square root, and that this also can be rigorously proven. This therefore provides an oracle-assisted Static DHP$_d$ algorithm that requires a subexponential number of oracle queries. We leave it as an open problem to find the precise complexity of Diem's algorithm, and the resulting complexity of our algorithm in this context.

4 Potentially Vulnerable Curves

At EUROCRYPT 2009, Galbraith, Lin and Scott proposed the use of special elliptic curves over $\mathbb{F}_{p^2}$ and $\mathbb{F}_{p^4}$ [21], which possess efficiently computable homomorphisms that permit a particularly efficient version of Gallant-Lambert-Vanstone point multiplciation method [23]. As well as the single bit speed-up of Pollard rho available on these curves, both the GHS attack [25] and Gaudry's attack [24] are considered, and appropriate recommendations are made in light of these. In particular, for curves over $\mathbb{F}_{p^2}$, neither of these attacks is faster than Pollard rho, and so these curves were believed to attain the desired security level. For curves over $\mathbb{F}_{p^4}$, in light of the latter attack the authors recommend that primes of length 80 bits should be used to achieve 128-bit security, rather than of length 64 bits, although it is stated that this is a very conservative choice, since Gaudry's algorithm requires expensive computations, and so potentially smaller primes could be used. Similarly Hankerson, Karabina and Menezes have considered the GLS point multiplication method over binary fields of the form $\mathbb{F}_{q^2}$ [29].

Prior to our attack, the only potential weakness of cryptographically interesting curves over $\mathbb{F}_{p^2}$ would be due to the Brown-Gallant-Cheon attack. In the best case (from an adversary's perspective), should the group order ±1 be divisible by an integer of size $O(p^{2/3})$, then the Static DHP$_d$ secret d can be computed in time $\tilde{O}(p^{2/3})$. Such a condition can be easily avoided should this attack be a concern. For the curves considered in [29], the Weil descent method is analysed and it is shown that the proportion of susceptible curves is negligible and can be provably avoided with a feasible computation. However, regardless of the divisibility properties of the group order ±1, the balanced oracle-assisted Static DHP$_d$ algorithm from §3.4 achieves a complexity of $\tilde{O}(p^{2/3})$ (and similarly for the binary curves). Assuming that point decompositions over the factor base can be computed efficiently, this attack therefore poses a real threat.

For curves over $\mathbb{F}_{p^4}$, our attack has complexity $\tilde{O}(p^{4/5})$, which is much faster than Gaudry's attack on the DLP, which has complexity $\tilde{O}(p^{3/2})$. Again assuming that point decompositions can be performed efficiently, curves over degree 4 extensions are also vulnerable.

Also of interest are the legacy curves which until recently formed part of the Oakley Key Determination Protocol, a part of IPSEC. These are the 'Well Known Groups' 3 and 4 [31] which are elliptic curves defined over the fields $\mathbb{F}_{2^{155}}$ and $\mathbb{F}_{2^{185}}$, and which have been the target of numerous attempted attacks via the Weil descent method [49, 25, 22, 44], since their inception.

5 Experimental Results

We implementated our oracle-assisted Static DHP$_d$ algorithm using the computational algebra system MAGMA [5] (V2.16-5), which was run on an Intel Xeon running at 3.16GHz with 32G of memory. We considered two sets of curves. The first set consisted of four randomly selected curves of prime order, each of which were 256 bits in length, for fields of the form $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^3}$, $\mathbb{F}_{p^4}$ and $\mathbb{F}_{p^5}$, see §5.1 and §5.2. These curves were chosen in order to measure how vulnerable the curves proposed in [21] are to our algorithm. We also provide estimates for solving the DLP on these curves via Pollard rho and the state of the art index calculus algorithms. The second set consisted of four randomly selected curves of order $4 \cdot p$ with p of bitlength 256 over the binary fields $\mathbb{F}_{2^{ln}}$, for $n = 2, 3, 4$ and 5, so that ln was as close to 256 as possible. The reason for implementing the attack on these curves was twofold: firstly to assess the security of the curves proposed in [29]; and secondly to compare the efficiency of the attack with the prime field case, with a view to assessing the difficulty of breaking the oracle-assisted Static DHP$_d$ on the Oakley curves, which we report in §5.3.

While our implementations in MAGMA are clearly sub-optimal, our goal was to provide a proof-of-concept implementation, and to give a reasonable indication of what can be achieved in practice. Indeed our results provide an upper bound for the time required to solve the oracle-assisted Static DHP$_d$ in each case. With a tailored and optimised low-level implementation our attack times can be improved significantly, as exemplified by the result reported in [28].

5.1 Large prime characteristic

For each of $n = 2, 3, 4$ and 5 we used curves of the form

$$E(\mathbb{F}_{p^n}) : y^2 = x^3 + ax + b,$$

for a and b randomly chosen elements of $\mathbb{F}_{p^n}$, such that $\#E(\mathbb{F}_{p^n})$ was a prime of bitlength 256.

For $n = 2, 3$ and 4 we computed the symmetrised summation polynomials f_3, f_4 and f_5 respectively, and all experiments were completed within two hours. For the computation of f_6, we surprisingly ran out of memory, and so instead independently symmetrised the two f_4 polynomials used in the resultant computation to reduce the number of terms, and substituted x_R into this partially symmmetrised version of f_6. One can extract the elementary symmetric polynomials from these two independent sets by appropriately recombining them. However the resulting Gröbner basis computation eventually exhausted the available memory and so the $n = 5$ experiments were unable to be completed. Without an accurate idea of how long the Gröbner basis computation might take were we to have sufficient available memory, we consider finding relations for curves over these fields to be impractical given our resources at the present time. Note however that for prime base fields, we know of no proposals in the literature for the use of degree five extension fields for elliptic curve cryptography. We therefore include results only for $n = 2, 3$ and 4, in Table 1.

Table 1. Data for testing and decomposing points for elliptic curves over extension fields. Times are in seconds.

n	$\log p$	$\#f_{n+1}$	$\#$ symf_{n+1}	T(GB)	T(roots)
2	128	13	5	0.001	0.009
3	85.3	439	43	0.029	0.027
4	64	54777	1100	5363	3.68

The column titles in the table denote respectively: the degree of the extension field; the size of the prime base field in bits; the number of monomials in f_{n+1}; the number of monomials in f_{n+1} once symmetrised; the average time required to perform a Gröbner basis computation; and the average time required to find the points that sum to the point being decomposed respectively.

As per §3.3 the last of these consists of the extraction of the degree one factors of the polynomial $g_n(e_n)$ and then substitutes the roots into the remaining polynomials $g_i(e_n)$ in equation (3). This is followed by the desymmetrisation factorisation (equation (4)) and then computation of the correct linear combination of factor base elements that sum to P (equation (5)).

As one can see, symmetrisation reduces the size of the system greatly. Note that the only setup cost comes from computing f_{n+1} and its symmetrisation; the final two columns give the average decomposition cost per input point, which for $n = 2$ and 3 is over 1000 inputs includes both those that do decompose over $\mathcal{F}$, as well as those that do not.

For $n = 4$, since the computation is significantly more costly, we report the time for one input point only; note that the input system for the Gröbner basis computation always has the same form but with different coefficients, and hence one expects this part of the computation to be very consistent. With regards to the root finding time, the three stages described above took $3.68s, 0.00s$ and $0.04s$ respectively, and so the dominant cost is the initial factorisation, which is necessary whether an input point decomposes or not. Hence we estimate the average time over uniformly chosen input points to be $\approx 3.68 + 0.04/4! \approx 3.68s$, since a point decomposes with probability $1/4!$.

5.2 Upper bounds on attack times

From the data in Table 1 and the time required to compute a scalar multiplication, one can compute an upper bound on the time required to carry out the attack in §3.4. Setting $|\mathcal{F}| = p^{\alpha}$, a minimising α balances the two stages of the attack, namely the oracle calls, and the relation finding stage. We ignore the cost of constructing the factor base since this only involves a handful of field operations and a Legendre symbol computation. A more careful version of the argument of §3.4 leads to the following equation:

$$p^{n(1-\alpha)} \cdot n! \cdot (T(\mathrm{GB}) + T(\mathrm{roots})) = p^{\alpha} \cdot T(\mathrm{scalar}),$$

where $T(\mathrm{scalar})$ denotes the average cost of a scalar multiplication. With our implementation the latter costs approximately $0.008s$, $0.011s$ and $0.012s$ on the curves defined over $\mathbb{F}_{p^2}$, $\mathbb{F}_{p^3}$ and $\mathbb{F}_{p^4}$ respectively.

Table 2 details the resulting values of alpha for $n = 2, 3, 4$ and the corresponding estimated attack times. As stated in §3.4, these estimates assume that each set of q^{α} factor base elements has the same probability of expressing the decomposition a random decomposable point as a linear sum of elements from that set.

Table 2. Attack time estimates for our implementation. Times are in seconds.

n	α	Attack time	Pollard rho
2	0.6701 (2/3)	$2^{79.8}$	$2^{111.3}$
3	0.7645 (3/4)	$2^{59.7}$	$2^{111.4}$
4	0.8730 (4/5)	$2^{50.5}$	$2^{111.4}$

The Pollard rho attack times have been estimated as $\sqrt{\pi \cdot 2^{256}/2}$ group operations, where the cost of a group operation has been estimated using the T(scalar) times above, assuming use of the double and add algorithm. We have incorporated the speed-up afforded by performing random walks on equivalence classes of points [14, 51] when the set of points $\{\pm\psi^i(P) : 0 \leq i < m\}$ for a given point P are deemed to be equivalent, where ψ is the homomorphism from [21]. This results in the three curves have virtually identical security.

Pollard rho however is not the fastest asymptotic DLP algorithm in this context. In the basic index calculus one finds $O(p)$ relations with a linear algebra cost of $O(p^2)$. Assuming the decomposition cost is sufficiently small, one can reduce the size of the factor base to balance the cost of the two stages, to $O(p^{2-\frac{2}{n+1}})$, which is originally due to Harley. In addition, one can also use single and double large prime variations [50, 26], resulting in complexities of $O(p^{2-\frac{2}{n+1/2}})$ and $O(p^{2-\frac{2}{n}})$ respectively.

Our implementation allows one to give upper bounds for the attack times for each of these approaches, and consequently provides information regarding what size of p should be chosen to provide 128 bit security, for each n, subject to our attack implementation. This security level is the length of time required to compute 2^{128} basic group operations. Note that in the double large prime variation, for the most interesting case $n = 4$ the number of relations required is $O(p^{3/2})$. With our decomposition implementation, the time for the relation generation stage is $p^{3/2} \cdot 4! \cdot 5366.68s \approx 2^{113.0}s$, which is comparable to Pollard rho. Hence for this security level, p of length 64 bits would appear to be secure. However, in an optimised implementation the decomposition time could clearly be improved, necessitating increasing p accordingly to compensate. Furthermore, since the relation generation stage is more costly than the linear algebra, to balance the two stages of the algorithm one would need to increase the factor base size marginally. These intricacies mean that although our implementation provides an upper bound for the attack time, how to select an appropriate size p to ensure security for elliptic curves over these extension fields remains an open issue.

5.3 Characteristic two

For each of $n = 2, 3, 4$ and 5 we used curves of the form

$$E(\mathbb{F}_{2^{ln}}) : y^2 + xy = x^3 + b, \tag{6}$$

for b a randomly chosen element of $\mathbb{F}_{2^{ln}}$, such that $\#E(\mathbb{F}_{2^{ln}})$ was a four times a prime of bitlength 256. Note that (6) is the form of the Oakley curves [31]. Also note that the base fields $\mathbb{F}_{2^l}$ in each case are not necessarily of prime extension degree over $\mathbb{F}_2$. Since our focus was to compare the effect of characteristic for fields of a given size with particular small extension degrees, we disregard any possible DLP weaknesses due to Weil descent for these example curves.

For these curves the summation polynomials are surprisingly simple, and very sparse, making their computation easy, in contrast to the prime base field case. Observe that as a result the size of the f_i and their symmetrisation is much smaller than before, faciliting a much faster Gröbner basis computation for $n = 4$.

As was the case for prime base fields, for $n = 5$ we also had insufficient memory to complete a decomposition using Gaudry's method. However, as stated in §1 and announced in [28], by attempting to write a random point on the curve as a sum of four factor base elements as in [37] one is able to find such a decomposition, at the expense of reducing the probability from $1/5!$ to $1/(2^l \cdot 4!)$. As with Gaudry's decomposition method, this method is much faster in characteristic two than in large prime characteristic. Thus for the Oakley Group 3 curve, the oracle-assisted Static DHP$_d$ problem is practical. Whether this can be extended to the Oakley Group 4 curve is worthy of further investigation.

For $n = 2, 3$ and 4 the time for a scalar multiplication is $0.014s$. Table 3 details the results using Gaudry's decomposition technique, with the final row detailing the results from the announcement [28] on the Oakley Group 3 curve.

Table 3. Data for testing and decomposing points for elliptic curves over binary extension fields and attack time estimates. Times are in seconds.

n	l	$\#f_{n+1}$	# symf_{n+1}	Time GB	Time roots	α	Attack time
2	129	5	3	0.000	0.008	0.6672 (2/3)	$2^{80.9}$
3	86	24	6	0.005	0.008	0.7572 (3/4)	$2^{60.0}$
4	65	729	39	247	0.88	0.8575 (4/5)	$2^{50.6}$
5	52	148300	638	N/A	N/A	N/A	N/A
5	31	729	39	0.021	(total time)	30/31	$2^{30.0}$

Note that despite the α values being smaller for binary fields — due to faster decompositions — the attack times are slightly higher, because the fields are 258 and 260 bits in size, as opposed to 256 bits. Due to the scalar multiplication time being very similar to the prime field case (with our implementation), the Pollard rho times are similar and hence the curves in [29] should also be considered vulnerable to our attack.

6 Other Cryptographically Relevant Assumptions

Our proposed oracle-assisted Static DHP$_d$ algorithm also solves the *Delayed Target DHP*, as defined by Freeman [18], which may be phrased as follows: A solver is given initial access to a Static DHP$_d$ oracle for the element $Q = dP \in \mathbb{G}$; when the Static DHP$_d$ oracle is removed, the solver is given a random element $X \in \mathbb{G}$ and must solve the DHP for input (Q, X), namely, output dX.

Koblitz and Menezes studied this problem in the context of Jacobians of hyperelliptic curves of small genus [38], along with several other problems, including the *Delayed Target DLP*, the *One-More DHP* and the *One-More DLP*. In the Delayed Target DLP, rather than given access to a Static DHP$_d$ oracle, the solver is given access to a discrete logarithm oracle but the problem is otherwise identical to the Delayed Target DHP. In the One-More DHP and One-More DLP the solver is supplied with a challenge oracle that outputs random elements of the group, as well as a Static DHP$_d$ oracle or a DLP oracle respectively. This time however the solver chooses an integer t and must solve t instances of the Static DHP$_d$ or the DLP, but is only allowed to use the Static DHP$_d$ or the DLP oracle at most $t - 1$ times.

The One-More DHP was first formulated in [3] while the One-More DLP was first formulated in [1] and [2]. Using Jacobians of hyperelliptic curves of small genus as example groups, Koblitz and Menezes argue that the constituents of each of the two pairs of similar problems — the Delayed Target DHP and Delayed Target DLP, and the One-More DHP and One-More DLP — should each be incomparable to one another. In particular there very probably does not exist a reduction between the Delayed Target DHP and the Delayed Target DLP, since in some groups the former appears to be easier than the latter, while in others the converse is true, and similarly for the One-More problems. However, their analysis of the Delayed Target DHP and One-More DHP contains a minor oversight, since it only considers the impact of the Brown-Gallant-Cheon algorithm and not the index calculus methods they used for studying the corresponding DLP versions. Doing so for Jacobians of hyperelliptic curves of genus ≥ 3, one sees that the complexities for the Delayed Target problems are identical, and similarly for the One-More problem variants.

Indeed, taking the basic Static DHP$_d$ algorithm presented in §3.2, one sees that by changing the Static DHP$_d$ oracle calls to DLP oracle calls, one obtains an otherwise unaltered algorithm and hence the complexities of the two delayed target problems are the same. Similarly any variation in factor base size will give rise to algorithms of the same complexity; the oracle calls themselves are not relevant to the structure of the algorithm, so it should be clear that for any group in which one can identify and use a factor base to generate relations, the Delayed Target DHP and Delayed Target DLP will have identical complexities, *whenever this method provides the most effective means to solve both problems.* Exceptions to this condition arise, for instance, when a faster algorithm applies to just one problem, as with the Brown-Gallant-Cheon algorithm for the Delayed Target DHP, for an elliptic curve over $\mathbb{F}_p$ whose group order ± 1 is divisible by an integer of size $\approx p^{1/3}$.

For the One-More problem variants, in our context we have the following simple algorithm. We choose the same factor base as in §3.2, and perform $|\mathcal{F}|$ Static DHP$_d$ oracle calls on its elements. Then for each of the $|\mathcal{F}| + 1$ challenge elements, we solve the appropriate problem exactly as before. The only difference between the one-more and the delayed target problems is that for the one-more variants we must solve $|\mathcal{F}| + 1$ such challenges, and not just one. If we perform the analysis of §3.4 once more we find that the optimal size of $\mathcal{F}$ is given by $\alpha = 1$, exactly as in §4.5 of [38]. As before either oracle can be applied to a given relation and so the One-More DHP and One-More DLP have the same complexity, and again will do so in any group *for which this method provides the most effective means to solve both problems.*

Interestingly this means that even when one can not find a natural reduction between two problems, the presence of an effective index calculus ensures that in some circumstances the problems have the same complexity. Furthermore the two pairs of problems considered above (as well as oracle-assisted Static DHP_d) are easier to solve than the DLP, for elliptic curves over extension fields, Jacobians of hyperelliptic curves of genus ≥ 3, and indeed for any group for which index calculus provides the best means to solve each of these problems.

Acknowledgements

The author would like to thank both Steven Galbraith and Alfred Menezes for their comments on an earlier draft of this article, and the reviewers for their suggestions.

References

1. M. Bellare, C. Namprempre, D. Pointcheval and M. Semanko, *The one-more-RSA-inversion problems and the security of Chaum's blind signature scheme,* Journal of Cryptology, 16, pp. 185-215, 2003.
2. M. Bellare and A. Palacio, *GQ and Schnorr identification schemes: proofs of security against impersonation under active and concurrent attacks,* Advances in Cryptology - CRYPTO 2002, LNCS 2442, pp. 149-162, Springer-Verlag, 2002.
3. A. Boldyreva, *Efficient threshold signatures, multisignatures and blind signatures based on the gap-Diffie-Hellman-group signature scheme,* PKC 2003, LNCS 2567, pp. 31-46, Springer-Verlag, 2003.
4. D. Boneh and X. Boyen, *Short signatures without random oracles,* Advances in Cryptology - EUROCRYPT 2004, LNCS 3027, pp. 56-73, Springer-Verlag, 2004.
5. Wieb Bosma, John Cannon and Catherine Playoust. *The Magma algebra system I: The user language.* J. Symbolic Comput., 24(3-4):235-265, 1997.
6. D.R.L. Brown and R.P. Gallant, *The Static Diffie-Hellman Problem,* Cryptology ePrint Archive, Report 2004/306, 2004.
7. D. Chaum and H. van Antwerpen, *Undeniable signatures,* Advances in Cryptology - Crypto 1989, LNCS 435, pp. 212-217, Springer-Verlag, 1989.
8. J. Cheon, *Security analysis of the Strong Diffie-Hellman problem,* Advances in Cryptology - EUROCRYPT 2006, LNCS 4004, pp. 1-11, Springer-Verlag, 2006.
9. D. Coppersmith. *Finding a Small Root of a Bivariate Integer Equation; Factoring with High Bits Known*, Advances in Cryptology - EUROCRYPT 1996, LNCS 1070, pp. 178-189, Springer-Verlag, 1996.
10. C. Diem, *On the discrete logarithm problem in class groups of curves,* Mathematics of Computation, to appear.
11. C. Diem, *On the discrete logarithm problem in elliptic curves,* Preprint, 2009.
12. W. Diffie and M.E. Hellman, *New directions in cryptography,* IEEE Trans. Inform. Theory 22 (6), pp. 644-654, 1976.
13. *Digital Signature Standard (DSS)*, FIPS PUB 186-2, 2000.

14. I. Duursma, P. Gaudry and F. Morain. *Speeding up the Discrete Log Computation on Curves with Automorphisms*, Advances in Cryptology - ASIACRYPT99, LNCS 1716, pp. 103-121, Springer-Verlag, 1999.
15. T. El-Gamal, *A public-key cryptosystem and a signature scheme based on discrete logarithms,* Advances in Cryptology - Crypto 1984, LNCS 196, pp. 10-18, Springer-Verlag, 1985.
16. J.C. Faugère, *A new efficient algorithm for computing Gröbner bases (F4),* Journal of Pure and Applied Algebra, 139 (1-3):61–88, 1999.
17. W. Ford and B. Kaliski, *Server-assisted generation of a strong secret from a password,* 9th international workshop on enabling technologies - WET ICE 2000, IEEE Press, 2000.
18. D. Freeman, *Pairing-based identification schemes,* technical report HPL-2005-154, Hewlett-Packard Laboratories, 2005.
19. G. Frey, *How to disguise an elliptic curve,* Talk at Waterloo workshop on the ECDLP, 1998, `http://cacr.math.uwaterloo.ca/conferences/1998/ecc98/slides.html`
20. G. Frey and H.G. Rück, *A remark concerning m-divisibility and the discrete logarithm problem in the divisor class group of curves,* Math. Comp., 62, pp. 865-874, 1994.
21. S.D. Galbraith, X. Lin and M. Scott, *Endomorphisms for Faster Elliptic Curve Cryptography on a Large Class of Curves,* Advances in Cryptology - EUROCRYPT 2009,LNCS 5479, pp. 518-535, Springer-Verlag, 2009.
22. S.D. Galbraith, F. Hess, N.P. Smart, *Extending the GHS Weil Descent Attack,* Advances in Cryptology - EUROCRYPT 2002, LNCS 2332, pp. 29-44, Springer-Verlag, 2004.
23. R.P. Gallant, R.J. Lambert and S.A. Vanstone, Faster Point Multiplication on Elliptic Curves with Efficient Endomorphisms, Advances in Cryptology - Crypto 2001, LNCS 2139, pp. 190-200, Springer-Verlag, 2001.
24. P. Gaudry, *Index calculus for abelian varieties of small dimension and the elliptic curve discrete logarithm problem,* Journal of Symbolic Computation 44, pp. 1690-1702, 2009.
25. P. Gaudry, F. Hess and N.P. Smart, *Constructive and destructive facets of Weil descent on elliptic curves,* Journal of Cryptology, (15), pp. 19-46, 2002.
26. P. Gaudry, E. Thomé, N. Thériault and C. Diem. *A Double Large Prime Variation for Small Genus Hyperelliptic Index Calculus,* Math. Comp. **76**, No. 257 (2007), pp. 475-492.
27. P. Gianni and T. Mora, *Algebraic solution of systems of polynomial equation using Gröbner bases*, Proc. AAECC 5, LNCS 356, 247–257, 1989.
28. R. Granger, A. Joux and V. Vitse. *New timings for oracle-assisted SDHP on the IPSEC Oakley 'Well Known Group' 3 curve.* Web announcement on Number Theory List, 8th July 2010. Available from `http://listserv.nodak.edu/archives/nmbrthry.html`
29. D. Hankerson, K. Karabina and A.J. Menezes, *Analyzing the Galbraith-Lin-Scott point multiplication method for elliptic curves over binary fields,* IEEE Transactions on Computers, 58, 1411-1420, 2009.
30. F. Hess. *The GHS Attack Revisited,* Advances in Cryptology - EUROCRYPT 2003, LNCS 2656, pp. 374-387, Springer-Verlag, 2003.
31. IETF, The Oakley Key Determination Protocol, IETF RFC 2412, November 1998.
32. D. Jao and K. Yoshida, *Boneh-Boyen Signatures and the Strong Diffie-Hellman Problem,* Pairing 2009, LNCS 5671, pp. 1-16, Springer-Verlag, 2009.

33. A. Joux and R. Lercier *The Function Field Sieve in the Medium Prime Case,* Advances in Cryptology - EUROCRYPT 2006, LNCS 4004, pp. 254-270, Springer-Verlag, 2006.
34. A. Joux, R. Lercier, D. Naccache and E. Thomé, *Oracle-Assisted Static Diffie-Hellman Is Easier Than Discrete Logarithms.* 12th IMA International Conference, Cryptography and Coding 2009, LNCS 5921, pp. 351-367, Springer-Verlag, 2009.
35. A. Joux, R. Lercier, N.P. Smart and F. Vercauteren, *The Number Field Sieve in the Medium Prime Case,* Advances in Cryptology - CRYPTO 2006, LNCS 4117, pp. 326-344, Springer-Verlag, 2006.
36. A. Joux, D. Naccache and E. Thomé, *When e-th roots become easier than factoring,* Advances in Cryptology - ASIACRYPT 2007, LNCS 4833, pp. 13-28, Springer-Verlag, 2007.
37. A. Joux and V. Vitse. *Elliptic Curve Discrete Logarithm Problem over Small Degree Extension Fields. Application to the static Diffie-Hellman problem on $E(\mathbb{F}_{q^5})$,* Cryptology ePrint Archive, Report 2010/157, 2010.
38. N. Koblitz and A.J. Menezes, *Another look at non-standard discrete log and Diffie-Hellman problems,* Journal of Mathematical Cryptology, Volume 2, Issue 4, pp. 311–326, December, 2008.
39. N. Koblitz and A.J. Menezes, *Intractable problems in cryptography,* Proceedings of the 9th International Conference on Finite Fields and Their Applications, to appear.
40. N. Koblitz and A.J. Menezes, *The brave new world of bodacious assumptions in cryptography,* Notices of the AMS, 57, 357-365, 2010.
41. Y.N. Lakshman, *On the complexity of computing Gröbner bases for zero-dimensional ideals.* Ph. D. Thesis, RPI, Troy, 1990.
42. U.M. Maurer and S. Wolf. *The Diffie-Hellman Protocol.* Designs, Codes, and Cryptography, volume 19, pp. 147–171, 2000.
43. A.J. Menezes, T. Okamoto and S.A. Vanstone, *Reducing elliptic curve logarithms to a finite field,* IEEE Trans. Info. Theory, 39, pp.1639-1646, 1993.
44. A. Menezes, E. Teske and A. Weng, *Weak Fields for ECC,* Topics in Cryptology – CT-RSA, 2004, LNCS 2964, pp. 366-386, Springer-Verlag, 2004.
45. T. Satoh and K. Araki, *Fermat quotients and the polynomial time discrete log algorithm for anomalous elliptic curves,* Comm. Math. Univ. Sancti Pauli, 47, pp. 81-92, 1998.
46. I.A. Semaev, *Evaluation of discrete logarithms on some elliptic curves,* Math. Comp., 67, pp.353-356, 1998.
47. I. Semaev, *Summation Polynomials and the discrete logarithm problem on elliptic curves,* Cryptology ePrint Archive, Report 2004/031, 2004.
48. N.P. Smart, *The discrete logarithm problem on elliptic curves of trace one,* Journal of Cryptology, 12, pp. 141-151, 1999.
49. N.P. Smart, *How Secure are Elliptic Curves over Composite Extension Fields?,* Advances in Cryptology - EUROCRYPT 2001, LNCS 2045, pp. 30-39, Springer-Verlag, 2001.
50. N. Thériault. *Index calculus attack for hyperelliptic curves of small genus*, Advances in Cryptology - ASIACRYPT 2003, LNCS 2894, pp. 75-92, SpringerVerlag, 2003.
51. M.J. Wiener and R.J. Zuccherato. *Faster Attacks on Elliptic Curve Cryptosystems*, Selected Areas in Cryptography - SAC 1998, LNCS 1556, pp. 190-200, SpringerVerlag, 1998.