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

Robert Granger

rgranger@computing.dcu.ie
Claude Shannon Institute, UCD and DCU, Ireland

ASIACRYPT, 8th December 2010

Outline

Diffie-Hellman Key Agreement

Let $\mathbb{G}$ be a cyclic group of prime order r with generator g.

Diffie-Hellman Key Agreement

Let $\mathbb{G}$ be a cyclic group of prime order r with generator g.

- Alice chooses $x \xleftarrow{R} \mathbb{Z}_r$, computes g^x and sends to Bob
- Bob chooses $y \xleftarrow{R} \mathbb{Z}_r$, computes g^y and sends to Alice
- Alice computes $(g^y)^x$, Bob computes $(g^x)^y$ to give shared secret g^{xy}

Diffie-Hellman Key Agreement

Let $\mathbb{G}$ be a cyclic group of prime order r with generator g.

- Alice chooses $x \stackrel{R}{\longleftarrow} \mathbb{Z}_r$, computes g^x and sends to Bob
- Bob chooses $y \stackrel{R}{\longleftarrow} \mathbb{Z}_r$, computes g^y and sends to Alice
- Alice computes $(g^y)^x$, Bob computes $(g^x)^y$ to give shared secret g^{xy}

A fundamental security requirement of DH Key Agreement is that the *Computational Diffie-Hellman* problem should be hard:

Definition

(CDH): Given g and random g^x and g^y, find g^{xy}

The Static Diffie-Hellman Problem (Static DHP)

Suppose to minimise her exponentiation cost in multiple DH key agreements Alice repeatedly reuses $x = d$.

The Static Diffie-Hellman Problem (Static DHP)

Suppose to minimise her exponentiation cost in multiple DH key agreements Alice repeatedly reuses $x = d$.

- This set of problem instances is a tiny subset of all CDH problem instances

The Static Diffie-Hellman Problem (Static DHP)

Suppose to minimise her exponentiation cost in multiple DH key agreements Alice repeatedly reuses $x = d$.

- This set of problem instances is a tiny subset of all CDH problem instances
- Not *a priori* clear that these instances should be hard, even if CDH is hard

The Static Diffie-Hellman Problem (Static DHP)

Suppose to minimise her exponentiation cost in multiple DH key agreements Alice repeatedly reuses $x = d$.

- This set of problem instances is a tiny subset of all CDH problem instances
- Not *a priori* clear that these instances should be hard, even if CDH is hard

Definition

(Static DHP_d): Given fixed g and g^d, and random g^y, find g^{dy}

The Static DHP - inception and first result

Introduced by Brown and Gallant in 2004, who gave a reduction from the DLP for d to the Static DHP$_d$

The Static DHP - inception and first result

Introduced by Brown and Gallant in 2004, who gave a reduction from the DLP for d to the Static DHP$_d$

- Hence if the DLP for d is hard, then so is the Static DHP$_d$

The Static DHP - inception and first result

Introduced by Brown and Gallant in 2004, who gave a reduction from the DLP for d to the Static DHP$_d$

- Hence if the DLP for d is hard, then so is the Static DHP$_d$
- Equivalently, given access to a Static DHP$_d$ oracle, one can find the associated DLP d

The Static DHP - inception and first result

Introduced by Brown and Gallant in 2004, who gave a reduction from the DLP for d to the Static DHP$_d$

- Hence if the DLP for d is hard, then so is the Static DHP$_d$
- Equivalently, given access to a Static DHP$_d$ oracle, one can find the associated DLP d

Definition

(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 $d \in \mathbb{Z}_r$ be such that $Q = dP$. Then a Static DHP$_d$ oracle (w.r.t. $(\mathbb{G}, P, Q)$) computes the function $\delta : \mathbb{G} \rightarrow \mathbb{G}$ where

$$\delta(X) = dX$$

Oracle-assisted Static DHP$_d$ algorithm

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.

Oracle-assisted Static DHP_d algorithm

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.

Theorem

Let $r = uv + 1$. Then d can be found with u calls to a Static DHP_d oracle, and off-line computational work of $O(\sqrt{u} + \sqrt{v})$ group operations.

DLP to Static DHP$_d$ reduction

- The complexity of the attack is minimised when $u \approx r^{1/3}$
- Depending on the factorisation of $r - 1$, can lead to a real attack which is quicker than solving the DLP

DLP to Static DHP_d reduction

- The complexity of the attack is minimised when $u \approx r^{1/3}$
- Depending on the factorisation of $r - 1$, can lead to a real attack which is quicker than solving the DLP

Brown and Gallant showed that a system entity acts as a Static DHP_d oracle, transforming their reduction into a DLP solver, for the following protocols:

- textbook El Gamal encryption
- Ford-Kaliski key retrieval
- Chaum-Van Antwerpen's undeniable signatures

Results of Koblitz and Menezes

In ‘Another look at non-standard discrete log and Diffie-Hellman problems’ [07], Koblitz and Menezes studied a set of problems in the Jacobian of small genus hyperelliptic curves

- *Delayed Target* DLP/DHP, *One-More* DLP/DHP, and DLP1/DHP1
- Using ‘Index Calculus’ or Brown-Gallant show that some are easier than DLP - hardness separation
- Argue that problems which are either interactive or have complicated inputs can produce weaknesses
- Conclude that security assurances provided by such assumptions should be reassessed/are difficult to assess

An oracle-assisted Static DHP algorithm

Assuming index calculus methodology applies, KM implied the following algorithm (cf. Joux-Naccache-Thomé [07]):

- Construct a factor base $\mathcal{F}$ over which a non-negligible proportion of group elements factor
- Call the Static DHP$_d$ oracle δ on all $P_i \in \mathcal{F}$
- For a target element X attempt to write random multiples aX as a sum of elements of $\mathcal{F}$, i.e., $aX = P_{i_1} + \cdots + P_{i_n}$
- Then $dX = (a^{-1} \bmod r)(\delta(P_{i_1}) + \cdots + \delta(P_{i_n}))$

Applied algorithm to finite fields and small genus hyperelliptic curves — resulting in a hardness separation from DLP

Example (KM): Hyperelliptic Curves

For the DLP, there are four basic variants:

- Gaudry (2000): basic index calculus — $O(q^2)$
- Harley (2000): reduce factor base — $O(q^{2-2/(g+1)})$
- Thériault (2003): large-prime variation — $O(q^{2-2/(g+1/2)})$
- GTTD (2007): double large-prime variation — $O(q^{2-2/g})$

Example (KM): Hyperelliptic Curves

For the DLP, there are four basic variants:

- Gaudry (2000): basic index calculus — $O(q^2)$
- Harley (2000): reduce factor base — $O(q^{2-2/(g+1)})$
- Thériault (2003): large-prime variation — $O(q^{2-2/(g+1/2)})$
- GTTD (2007): double large-prime variation — $O(q^{2-2/g})$

The oracle-assisted Static DHP algorithm is $O(q^{1-1/(g+1)})$ — the square root of Harley's algorithm:

Example (KM): Hyperelliptic Curves

For the DLP, there are four basic variants:

- Gaudry (2000): basic index calculus — $O(q^2)$
- Harley (2000): reduce factor base — $O(q^{2-2/(g+1)})$
- Thériault (2003): large-prime variation — $O(q^{2-2/(g+1/2)})$
- GTTD (2007): double large-prime variation — $O(q^{2-2/g})$

The oracle-assisted Static DHP algorithm is $O(q^{1-1/(g+1)})$ — the square root of Harley's algorithm:

- No linear algebra

Example (KM): Hyperelliptic Curves

For the DLP, there are four basic variants:

- Gaudry (2000): basic index calculus — $O(q^2)$
- Harley (2000): reduce factor base — $O(q^{2-2/(g+1)})$
- Thériault (2003): large-prime variation — $O(q^{2-2/(g+1/2)})$
- GTTD (2007): double large-prime variation — $O(q^{2-2/g})$

The oracle-assisted Static DHP algorithm is $O(q^{1-1/(g+1)})$ — the square root of Harley's algorithm:

- No linear algebra
- Only one relation needed so no large-prime elimination

Example (KM): Hyperelliptic Curves

For the DLP, there are four basic variants:

- Gaudry (2000): basic index calculus — $O(q^2)$
- Harley (2000): reduce factor base — $O(q^{2-2/(g+1)})$
- Thériault (2003): large-prime variation — $O(q^{2-2/(g+1/2)})$
- GTTD (2007): double large-prime variation — $O(q^{2-2/g})$

The oracle-assisted Static DHP algorithm is $O(q^{1-1/(g+1)})$ — the square root of Harley's algorithm:

- No linear algebra
- Only one relation needed so no large-prime elimination

Question: For $g = 1$ have $O(q^{1/2})$, so can one do better?

Oracle-assisted Static DHP for elliptic curves?

- Problem is that one needs a factor base to beat the Brown-Gallant complexity

Oracle-assisted Static DHP for elliptic curves?

- Problem is that one needs a factor base to beat the Brown-Gallant complexity
- For ECs over $\mathbb{F}_p$, currently no known useful factor base

Oracle-assisted Static DHP for elliptic curves?

- Problem is that one needs a factor base to beat the Brown-Gallant complexity
- For ECs over $\mathbb{F}_p$, currently no known useful factor base
- Basic insight is that for ECs over extension fields, one already has a native factorisation via Gaudry-Semaev ECDLP algorithm $\Longrightarrow$ can use the KM methodology

Oracle-assisted Static DHP for elliptic curves?

- Problem is that one needs a factor base to beat the Brown-Gallant complexity
- For ECs over $\mathbb{F}_p$, currently no known useful factor base
- Basic insight is that for ECs over extension fields, one already has a native factorisation via Gaudry-Semaev ECDLP algorithm $\Longrightarrow$ can use the KM methodology
- Basic observation made independently by Joux-Vitse [10]

Semaev's summation polynomials

Let $E : Y^2 = X^3 + aX + b$, over a field $\mathbb{F}_q$ with $\mathrm{char}(\mathbb{F}_q) > 3$.

For $m \geq 2$ define $f_m = f_m(X_1, \ldots, X_m) \in \mathbb{F}_q[X_1, \ldots, X_m]$ by the following property:

- For $x_1, \ldots, x_m \in \overline{\mathbb{F}}_q$, $f_m(x_1, \ldots, x_m) = 0$ is equivalent to $\exists y_1, \ldots, y_m \in \overline{\mathbb{F}}_q$ such that $(x_i, y_i) \in E(\overline{\mathbb{F}}_q)$ and

$$(x_1, y_1) + \cdots + (x_m, y_m) = \mathcal{O} \in E(\overline{\mathbb{F}}_q)$$

- This means that in order to write $R = P_{i_1} + \cdots + P_{i_m}$ over some $\mathcal{F}$ one needs only solve

$$f_{m+1}(x_1, \ldots, x_m, x_R) = 0 \in \mathbb{F}_q$$

Gaudry's insight

Assume that E is defined over a degree n extension $\mathbb{F}_{q^n}$.

- Fix a poly basis $\{t^{n-1}, \ldots, t, 1\}$ for $\mathbb{F}_{q^n}/\mathbb{F}_q$
- Define $\mathcal{F} = \{P = (x, y) \in E(\mathbb{F}_{q^n})\ s.t.\ x \in \mathbb{F}_q\}$
- Note $|\mathcal{F}| \approx q$
- Observe that $f_{n+1}(x_1, \ldots, x_n, x_R) = 0$ has n components via Weil restriction to $\mathbb{F}_q$:

$$f_{n+1,0} + f_{n+1,1}t + \cdots + f_{n+1,n-1}t^{n-1} = 0 \in \mathbb{F}_{q^n}$$

- System of n equations over $\mathbb{F}_q$ in n variables in $\mathbb{F}_q$
- Solved via resultants or a Grobner basis computation

ECDLP complexity with Gaudry-Semaev

- Decomposition complexity $\widetilde{O}(Poly(2^{n(n-1)}))$
- Decomposition probability is $1/n!$
- For fixed n, $q \to \infty$, complexity is $\widetilde{O}(q^2)$, rho is $\widetilde{O}(q^{n/2})$
- Using double large-prime variation reduces to $\widetilde{O}(q^{2-2/n})$
- Computationally *far* more intensive than the Gaudry-Hess-Smart attack
- Works for *all* curves defined over any extension field
- Subexponential attack for a large class of fields (Diem)

$$e^{O((\log q^n)^{2/3})}$$

Algorithm complexity

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* $Poly(\log q)$.

Algorithm complexity

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$(\log q)$.*

- Can reduce the factor base à la Harley:

Algorithm complexity

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 $Poly(\log q)$.*

- Can reduce the factor base à la Harley:

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 $\widetilde{O}(q^{1-\frac{1}{n+1}})$.*

Algorithm complexity

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 $Poly(\log q)$.*

- Can reduce the factor base à la Harley:

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 $\widetilde{O}(q^{1-\frac{1}{n+1}})$.*

- Can also obtain subexponential algorithm à la Diem

The Galbraith-Lin-Scott Curves

At EUROCRYPT 2009 the use of curves defined over extension fields with degree a power of 2 were proposed.

- GLS curves possess an efficiently computable endomorphism $\Longrightarrow$ GLV fast point multiplication method
- Over $\mathbb{F}_{p^2}$ method takes between 0.70 and 0.83 the time of the previous best methods
- Performance over $\mathbb{F}_{p^4}$ currently uninvestigated, but subject to Gaudry's ECDLP attack
- GLS technique investigated for binary curves by Hankerson-Karabina-Menezes [08]

The Oakley key determination protocol curves

'Well-Known Group' 3

Group 3 is defined over the field $\mathbb{F}_{2^{155}} = \mathbb{F}_2[\omega]/(\omega^{155} + \omega^{62} + 1)$, by the equation

$$Y^2 + XY = X^3 + \beta,$$

where

$$\beta = \omega^{18} + \omega^{17} + \omega^{16} + \omega^{13} + \omega^{12} + \omega^9 + \omega^8 + \omega^7 + \omega^3 + \omega^2 + \omega + 1.$$

- $\#E(\mathbb{F}_{2^{155}}) = 12 \cdot r$, with $r =$

  3805993847215893016155463826195386266397436443

- Several unsuccessful DLP attacks via Weil descent: Jacobson-Menezes-Stein [01], Gaudry-Hess-Smart [00], Galbraith-Hess-Smart [02], Hess [03]

The Oakley key determination protocol curves

'Well-Known Group' 4

Group 4 is defined over the field $\mathbb{F}_{2^{185}} = \mathbb{F}_2[\omega]/(\omega^{185} + \omega^{69} + 1)$, by the equation

$$Y^2 + XY = X^3 + \beta,$$

where

$$\beta = \omega^{12} + \omega^{11} + \omega^{10} + \omega^9 + \omega^7 + \omega^6 + \omega^5 + \omega^3 + 1.$$

- $\#E(\mathbb{F}_{2^{185}}) = 4 \cdot r$, with $r =$

$$12259964326927110866866776214413170562013096\backslash$$
$$250261263279$$

- DLP studied by Maurer-Menezes-Teske [01] and Menezes-Teske-Weng [04], the latter concluding that the fields $\mathbb{F}_{2^{5l}}$ for $l > 37$ are 'weak' while the security of ECs over $\mathbb{F}_{2^{185}}$ is questionable

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.

- Implemented in MAGMA (V2.16-5) run on a 3.16 GHz Intel Xeon with 32G RAM

Data for testing and decomposing points for elliptic curves over extension fields (times in s):

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

Large prime characteristic

Upper bounds on attack time

Given data, compute α such that:

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

Large prime characteristic

Upper bounds on attack time

Given data, compute α such that:

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

Attack time estimates for our implementation (times in s):

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)	$2^{50.5}$	$2^{111.4}$

Characteristic two

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

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

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.

Characteristic two

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

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

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.

Data for testing and decomposing points for elliptic curves over binary extension fields and attack time estimates (times in s):

n	$\#f_{n+1}$	$\#$ symf_{n+1}	Time GB	α	Attack time
2	5	3	0.000	0.6672	$2^{80.9}$
3	24	6	0.005	0.7572	$2^{60.0}$
4	729	39	247	0.8575	$2^{50.6}$
5	148300	638	N/A	N/A	N/A

All is not lost however...

Joux-Vitse variant $\Longrightarrow n = 5$ systems are solvable, but with much smaller probability.

- See "New timings for oracle-assisted SDHP on the IPSEC Oakley 'Well Known Group' 3 curve" on NTL, July 2010 [G.,Joux,Vitse]
- Can solve oracle-assisted Static DHP (excluding $\approx 2^{30}$ oracle queries) in ≈ 37.5 years
- Estimated time for 'Well-Known Group' 4 (excluding $\approx 2^{36}$ oracle queries) is $\approx 3.4 \times 10^3$ years

All is not lost however...

Joux-Vitse variant $\Longrightarrow n = 5$ systems are solvable, but with much smaller probability.

- See "New timings for oracle-assisted SDHP on the IPSEC Oakley 'Well Known Group' 3 curve" on NTL, July 2010 [G.,Joux,Vitse]
- Can solve oracle-assisted Static DHP (excluding $\approx 2^{30}$ oracle queries) in ≈ 37.5 years
- Estimated time for 'Well-Known Group' 4 (excluding $\approx 2^{36}$ oracle queries) is $\approx 3.4 \times 10^3$ years

New Result [G.] - in preparation:

- For curves over $\mathbb{F}_{2^{ln}}$ can solve the oracle-assisted Static DHP without using a native factorisation method
- Better complexity than the above and faster for $n = 5$ as soon as $q > 2^{35}$

Conclusions

- Elliptic curves defined over extension fields may be unsuitable in some cryptographic scenarios

Conclusions

- Elliptic curves defined over extension fields may be unsuitable in some cryptographic scenarios
- Practical attack(s) on Oakley 'Well-Known Groups' 3 and 4

Conclusions

- Elliptic curves defined over extension fields may be unsuitable in some cryptographic scenarios
- Practical attack(s) on Oakley 'Well-Known Groups' 3 and 4
- Some problems occurring in security proofs are easier than the DLP - up to nearly square-root faster when index calculus applies