Equivalence between DLP and DHP

A. Muzereau
N.P. Smart
F. Vercauteren

Notation

Let G denote a finite abelian group of prime order p

Let G be generated by g.

Discrete Logarithm Problem (DLP)

- Given $h = g^{\alpha}$, compute α.

Diffie–Hellman Problem (DHP)

- Given g^{α} and g^{β} compute $g^{\alpha\beta}$.

Significance

The Diffie–Hellman problem is important in cryptography for a number of reasons

- Clearly Diffie–Hellman protocol
- Various ElGamal like encryption schemes
  - e.g. ECIES
- Higher level protocols
  - e.g. voting, etc.

Questions

How hard is the Diffie–Hellman problem ?

If we could solve the Diffie–Hellman problem could we solve the Discrete Logarithm problem ?

If the Discrete Logarithm problem is as hard as X, what does this say about the Diffie–Hellman problem ?

Difficulties

For finite fields we know the DLP is a sub-exponential problem, hence the best we can hope for is that DHP is a sub-exponential problem.

For elliptic curves we expect the DLP to be an exponential problem, would therefore like the DHP to also be exponential in behaviour.

Would also like an exact security result to link the difficulty of the DHP in terms of the DLP for any given instance

- Since DLP is well studied
- But DHP is actually used in practice

DHP $\leq$ DLP

Clearly if we can solve the DLP in a group G then we can solve the DHP.

- Given g^{α} and g^{β}
- Apply the DL oracle to find α
- Then compute
  - DH secret is $(g^{\beta})^{\alpha}$.

Major question is whether we can solve the DLP using an oracle for the DHP.

- In “most cases” we can using an algorithm of Maurer and Wolf.

Implicit vs Explicit Representation

Supppose we have elements x, y of $\mathbb{F}_p$.

The explicit representation of x and y is simply their usual integer representation.

- We can clearly add and multiply elements in explicit representation.

The implicit representation of x and y is the values g^x and g^y in G.

- Recall G is a group of order p generated by g

Implicit vs Explicit Representation

Supppose we have elements x, y of $\mathbb{F}_p$.

The explicit representation of x and y is simply their usual integer representation.

- We can clearly add and multiply elements in explicit representation.

The implicit representation of x and y is the values g^x and g^y in G.

- Recall G is a group of order p generated by g

To add elements in implicit representation, we multiply their implicit representations

- $g^{x+y} = g^x \cdot g^y$

Implicit vs Explicit Representation

Supppose we have elements x, y of $\mathbb{F}_p$.

The explicit representation of x and y is simply their usual integer representation.

- We can clearly add and multiply elements in explicit representation.

The implicit representation of x and y is the values g^x and g^y in G.

- Recall G is a group of order p generated by g

To multiply elements in implicit representation, we require a Diffie–Hellman oracle.

- $g^{xy} = DH(g^x, g^y)$

Algorithm Overview

In the following slides we describe the reduction of Maurer–Wolf:

Step 1:

- Compute valid x-coordinate in implicit representation, related to the DL α.

Step 2:

- Compute corresponding y in implicit representation.

Step 3:

- Find DL of Q with respect to P, $Q = [k]P$.

Step 4:

- Compute $[k]P$ (for real) and hence recover x, and thus α.

Input

The algorithm requires as input

- A group G of order p and a DLP problem $h = g^{\alpha}$.
- A DH oracle for G.
- An elliptic curve E over $\mathbb{F}_p$,
  - Of "smooth order" so that computing DLs is easy.

$$\#E(\mathbb{F}_p) = \prod_{j=1}^{s} q_j$$

A known point P which is a generator of E.

Step 1:

Compute a valid x-coordinate in implicit representation, related to the DL α.

- Set $x = \alpha$.
- Compute $g^{x^3+ax+b} = g^z$
- Test quadratic residuosity of $z \bmod p$
  - Compute $g^{z^{(p-1)/2}}$ and g and compare them.
  - On equality, go to Step 2,
  - Else replace x by $\alpha + d$ for a random d and repeat.

Step 2:

Compute g^y from $g^z = g^{y^2}$ using the algorithm of Tonelli and Shanks (but in implicit representation).

Write $p - 1 = 2^e \cdot w$ with w odd.

Set $g^s \leftarrow g$, $r \leftarrow e$, $g^y \leftarrow g^{z^{(w-1)/2}}$, $g^b \leftarrow g^{zy^2}$, $g^y \leftarrow g^{zy}$.

If $g^b \equiv 1 \bmod p$, output g^y and go to Step 3.

Otherwise, find the smallest $m \geq 1$ such that $g^{(b^{2^m})} \equiv 1 \bmod p$.

Set $g^t \leftarrow g^{(s^{2^{r-m-1}})}$, $g^s \leftarrow g^{t^2}$, $r \leftarrow m$, $g^y \leftarrow g^{yt}$, $g^b \leftarrow g^{bs}$ and repeat.

Note $Q = (x, y)$ is a point on E, however we only know the implicit representation (g^x, g^y).

Step 3:

Find DL of Q with respect to P, $Q = [k]P$.

For j from 1 to s, do the following:

- Compute (g^{u_j}, g^{v_j}) such that $(u_j, v_j) = \frac{|E|}{q_j} \cdot Q$.
- For i from 0 to $q_j - 1$, do the following:
  - Compute $(u_{ji}, v_{ji}) = i \cdot \frac{|E|}{q_j} \cdot P$.
  - Compute $(g^{u_{ji}}, g^{v_{ji}})$.
  - Compare $(g^{u_{ji}}, g^{v_{ji}})$ with (g^{u_i}, g^{v_i}).
  - On equality let $k_j = i$ and go to next iteration in j

Compute $k \bmod |E|$ such that $\forall j \in \{1, \ldots, s\}, \quad k \equiv k_j \bmod q_j$.

Step 4:

Compute $[k]P$ (for real) and hence recover x, and thus α.

We know $Q = [k]P$.

We know k and P explicitly.

We only know Q implicitly.

But we now compute $[k]P$ and so compute $Q = (x, y)$ explicitly.

But x and α have a known relationship.

- Hence we compute α.

Notes

In the above algorithm sketch in Step 3 we have used a naive exhaustive search for finding Dlogs in subgroups of order q_j.

- In "traditional" use of Maurer–Wolf result this is replaced by the Baby-Step/Giant-Step method

Using standard addition formulae for an elliptic curve we can estimate precisely the average number operations needed to compute the discrete logarithm α.

Need to count both

- Multiplications in $\mathbb{F}_p$.
- Inversions in $\mathbb{F}_p$.
- Calls to the DH oracle.
- Calls to a DH-inversion oracle (computes $g^{1/x}$ given g^x).

Simplifications

We assume

- All exponentiations/point multiplications are performed using the simple binary method.
  - Only give average costs from now on.

- A field inversion costs about ten field multiplications.
  - Usually true in practice for cryptographic sizes of p.

And note:

- A DH-inversion oracle can be implements using on average $\frac{3}{2}\log_2 p$ calls to the DH oracle, since

$$g^{1/x} = g^{x^{p-2}}.$$

Complexity : I

$$\#E(\mathbb{F}_p) = \prod_{j=1}^{s} q_j$$

with $q_j \leq B$.

Let the number of $\mathbb{F}_p$ multiplications be N_M and the number of DH oracle calls be N_O.

Let the cost of solving the DLP be C_{DLP} and the cost of solving the DHP be C_{DHP}.

Using the Baby-Step/Giant-Step method in Step 3 of the algorithm:

- The number of $\mathbb{F}_p$-multiplications and DH oracle calls is roughly balanced at around

$$N_M \approx N_O \approx O\left(\sqrt{B}(\log p)^3\right).$$

Implications : I

Assuming an $\mathbb{F}_p$ multiplication takes unit time we have

$$N_M + N_O \cdot C_{DHP} = C_{DLP}.$$

If we use BSGS in Step 3 and E is polynomially smooth

- i.e. B is polynomial in $\log p$.

Then

- DHP and DLP are polynomial time equivalent.
  - i.e. If DLP is sub-exponential then so is DHP.

This is the traditional usage of the algorithm of Maurer–Wolf.

- Trouble is such polynomially smooth curves are exponentially hard to find, so the result is (almost) useless.

Implications : II

Would rather have a precise security guarantee for the DHP, given some lower bound on the difficulty of the DLP.

$$N_M + N_O \cdot C_{DHP} = C_{DLP}.$$

Now if N_M is smaller than C_{DLP} then

$$C_{DHP} \approx \frac{C_{DLP}}{N_O}.$$

Hence, in our complexity estimate we want to minimize N_O,

- At the possible expense of increasing N_M.
- As long as N_M is less than C_{DLP}.
- i.e. no longer require N_M and N_O to be balanced.

Can achieve this by not using the BSGS in Step 3 and using naive exhaustive search instead.

Complexity : II

Given

$$\#E(\mathbb{F}_p) = \prod_{j=1}^{s} q_j$$

Now have, on average, using naive search in Step 3

$$N_M = (65s - 50)\log_2 p + (5 + \frac{3}{2}\log_2 p)\sum_{j=1}^{s} q_j.$$

and

$$N_O = \left(\frac{11}{2}s - 1 + \frac{9}{4}(s-1)\log_2 p\right) \cdot \log_2 p$$

Implications : III

Assume G is an elliptic curve group

- So that $C_{DLP} \approx \sqrt{p}$.

Now, we do not need $\#E(\mathbb{F}_p)$ to be very smooth at all.

- Easier to find such curves

Best result obtained when $s = 3$ and $q_j \approx p^{1/3}$.

- $N_M = O(p^{1/3})$.
- $N_O = O((\log_2 p)^2)$.

Then $N_M \leq C_{DLP}$ and the exact security of the DHP is given by

$$C_{DHP} \approx \frac{\sqrt{p}}{N_O}$$

This is the minimum number of operations needed to solve C_{DHP} assuming the best algorithm for DLP takes time $\sqrt{p}$.

Results

We calculated the exact values for the expected values of N_O and N_M for various elliptic curves in the standards given our best choice of auxillary curve E.

Assuming N_M was small enough this enabled us to give the an lower bound on the difficulty of DHP, assuming DLP was as hard as conjectured.

E.G.

SECP192R1

Any algorithm to solve the DHP would require at least 2^{77} operations.

- Compare to 2^{96} which is the expected number of operations needed to solve the DLP on this curve.

Meaning

If we found an algorithm for the DHP on the curve SECP192R1 which required less than 2^{77} operations then

- Major advance in ECDLP algorithms
- Conclude no-such DH algorithm for SECP192R1 can exist

This does not mean that a DH algorithm exists which runs in 2^{77} operations

- This is the minimum number of operations required
- We know that we can solve the DH problem in 2^{96} operations
  - By solving the ECDLP

We expect that the actual complexity for the DH problem on SECP192R1 is 2^{96} operations.

Conclusion

The Maurer–Wolf algorithm gives a polynomial time equivalence between DLP and DHP only when a suitably smooth elliptic curve exists.

- It is exponentially hard to find such a curve.

When DLP has expected exponential complexity...

- If we use the Maurer–Wolf with naive search in Step 3
  - Do not need such smooth auxillary elliptic curves
  - Obtain an exact security result for DH in elliptic curves groups