Norwegian University of Science and Technology

DISCRETE LOGARITHMS, DIFFIE-HELLMAN PROBLEMS AND THE MAURER REDUCTION

Trial Lecture

Jonathan Komada Eriksen

23.08.2024

Contents

Contents

Diffie-Hellman

A Cornerstone of Modern Cryptography

In 1976, Diffie and Hellman came up with a way for two parties to arrive at a shared secret, only communicating over a public channel.

Setup

Fix a cyclic group $G = \langle g \rangle$ of order N.

Alice		Bob
$a \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$		
	$\xrightarrow{g^a}$	
		$b \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z}$
	$\xleftarrow{g^b}$	
$\text{key} = (g^b)^a$		$\text{key} = (g^a)^b$

Discrete logarithm

Setup

Fix a cyclic group $G = \langle g \rangle$ of order N.

$$
\begin{array}{ccc}
\text{Alice} & & \text{Bob} \\
\hline
a \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z} & & \\
 & \xrightarrow{\quad g^a \quad} & \\
 & & b \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z} \\
 & \xleftarrow{\quad g^b \quad} & \\
\text{key} = (g^b)^a & & \text{key} = (g^a)^b
\end{array}
$$

The discrete logarithm problem

Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Discrete logarithm

Setup

Fix a cyclic group $G = \langle g \rangle$ of order N.

$$
\begin{array}{ccc}
\text{Alice} & & \text{Bob} \\
\hline
a \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z} & & \\
 & \xrightarrow{\quad g^a \quad} & \\
 & & b \xleftarrow{\$} \mathbb{Z}/N\mathbb{Z} \\
 & \xleftarrow{\quad g^b \quad} & \\
\text{key} = (g^b)^a & & \text{key} = (g^a)^b
\end{array}
$$

The discrete logarithm problem

Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

The computational Diffie-Hellman problem

Given $g, g^a, g^b \in G$, compute $g^{ab} \in G$.

Reductions

Relating problems

Given Problem 1 and Problem 2, how do we prove which one is harder?

[1] Something which takes an instance of Problem 1 and spits out an answer in polynomial time. Importantly, we do *not* care how.

Reductions

Relating problems

Given Problem 1 and Problem 2, how do we prove which one is harder?

Oracles and reductions!

Assume we are given an oracle[1] $\mathcal{O}$ for Problem 1, we say that Problem 2 *reduces to* Problem 1, if we can use $\mathcal{O}$ to solve Problem 2 (in polynomial time).
Intuitively, Problem 2 can not be *harder* than Problem 1.

[1] Something which takes an instance of Problem 1 and spits out an answer in polynomial time. Importantly, we do *not* care how.

Reductions

Relating problems

Given Problem 1 and Problem 2, how do we prove which one is harder?

Oracles and reductions!

Assume we are given an oracle[1] $\mathcal{O}$ for Problem 1, we say that Problem 2 *reduces to* Problem 1, if we can use $\mathcal{O}$ to solve Problem 2 (in polynomial time).
Intuitively, Problem 2 can not be *harder* than Problem 1.

Equivalent problems

If Problem 1 reduces to Problem 2 AND Problem 2 reduces to Problem 1, we say that these problems are *equivalent*.

[1] Something which takes an instance of Problem 1 and spits out an answer in polynomial time. Importantly, we do *not* care how.

A Trivial Reduction

The discrete logarithm problem (DLOG)

Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

The computational Diffie-Hellman (CDH) problem

Given $g, g^a, g^b \in G$, compute $g^{ab} \in G$.

Observation

CDH reduces to DLOG.

A Trivial Reduction

The discrete logarithm problem (DLOG)

Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

The computational Diffie-Hellman (CDH) problem

Given $g, g^a, g^b \in G$, compute $g^{ab} \in G$.

Observation

CDH reduces to DLOG.

Proof.

Given an instance $g, g^a, g^b \in G$ of CDH, and a oracle $\mathcal{O}$ for DLOG, get $a \leftarrow \mathcal{O}(g, g^a)$, and output $(g^b)^a$. $\square$

Summary

Goal of lecture

- So far, we have that CDH reduces to DLOG.
- This does not really say much about the security of Diffie-Hellman...
- What we really want is a reduction in the OTHER direction.

Contents

Pohlig-Hellman

The discrete logarithm (DLOG) problem

Let $G = \langle g \rangle$, of order $N = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$, where p_i are prime powers. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Pohlig-Hellman

Reduces this to computing DLOGs in groups of order p_i.

- Solving discrete logs in groups of prime power order.
- Combining results using the chinese remainder theorem.

Pohlig-Hellman - Prime power case

DLOG - Special case

Let $G = \langle g \rangle$, of order $N = p^e$, where p is prime. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Algorithm

Solving the above reduces to solving DLOG in groups of order p, by iteratively computing coefficients in the p-adic expansion of x.

- Write $x = x_0 + px_1 + \cdots + p^{e-1}x_{e-1}$ in base p.
  - i.e. $X = g^{x_0 + px_1 + \cdots + p^{e-1}x_{e-1}}$.

Pohlig-Hellman - Prime power case

DLOG - Special case

Let $G = \langle g \rangle$, of order $N = p^e$, where p is prime. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Algorithm

Solving the above reduces to solving DLOG in groups of order p, by iteratively computing coefficients in the p-adic expansion of x.

- Write $x = x_0 + px_1 + \cdots + p^{e-1}x_{e-1}$ in base p.
  - i.e. $X = g^{x_0 + px_1 + \cdots + p^{e-1}x_{e-1}}$.
- Let $y = x_0 + px_1 + \cdots + p^{n-1}x_{n-1}$ be a partial solution.

Pohlig-Hellman - Prime power case

DLOG - Special case

Let $G = \langle g \rangle$, of order $N = p^e$, where p is prime. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Algorithm

Solving the above reduces to solving DLOG in groups of order p, by iteratively computing coefficients in the p-adic expansion of x.

- Write $x = x_0 + px_1 + \cdots + p^{e-1}x_{e-1}$ in base p.
  - i.e. $X = g^{x_0 + px_1 + \cdots + p^{e-1}x_{e-1}}$.
- Let $y = x_0 + px_1 + \cdots + p^{n-1}x_{n-1}$ be a partial solution.
- Notice $g^{-y}X = g^{x_n p^n + \cdots + p^{e-1}x_{e-1}}$.

Pohlig-Hellman - Prime power case

DLOG - Special case

Let $G = \langle g \rangle$, of order $N = p^e$, where p is prime. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Algorithm

Solving the above reduces to solving DLOG in groups of order p, by iteratively computing coefficients in the p-adic expansion of x.

- Write $x = x_0 + px_1 + \cdots + p^{e-1}x_{e-1}$ in base p.
  - i.e. $X = g^{x_0 + px_1 + \cdots + p^{e-1}x_{e-1}}$.
- Let $y = x_0 + px_1 + \cdots + p^{n-1}x_{n-1}$ be a partial solution.
- Notice $g^{-y}X = g^{x_n p^n + \cdots + p^{e-1}x_{e-1}}$.
- Then $(g^{p^{e-1}})^{x_n} = (g^{-y}X)^{p^{e-1-n}}$.

Pohlig-Hellman - Full algorithm

Back to the general case, where G has order $N = p_1^{e_1} p_2^{e_2} \cdots p_n^{e_n}$.

The Chinese Remainder Theorem

Since $G \simeq \mathbb{Z}/p_1^{e_1}\mathbb{Z} \times \mathbb{Z}/p_2^{e_2}\mathbb{Z} \times \cdots \times \mathbb{Z}/p_n^{e_n}\mathbb{Z}$, simply project onto each summand.
Solving each prime power case, we get a system of congruences

$$\begin{aligned} x &\equiv x_1 \pmod{p_1^{e_1}}, \\ &\vdots \\ x &\equiv x_n \pmod{p_n^{e_n}}, \end{aligned}$$

which recovers $x \mod N$.

Baby Step - Giant Step

Due to Pohlig-Hellman, the following is sufficient:

The discrete logarithm (DLOG) problem

Let $G = \langle g \rangle$, of prime order p. Given an element $X \in G$, compute $x \in \mathbb{Z}$ such that $g^x = X$.

Baby step-Giant step

Solves in $O(\sqrt{p})$ time and memory.

- Based on a simple time-memory trade-off.

Baby step-Giant step algorithm

Basic idea

Write the solution $x = am + b$ for $m = \lceil\sqrt{p}\rceil$, i.e. $g^{am+b} = X$.

1. Set $m = \lceil\sqrt{p}\rceil$.
2. For each $0 \leq b < m$:
   2.1 Compute and save the pair (b, g^b) in a table.
3. compute $Y = g^{-m}$.
4. For each $0 \leq a < m$:
   4.1 Compute and check if XY^a is in the table, say for b.
   4.2 If so, return $am + b$.

Generic Groups vs. Actual Groups

Can we do better?

The above is essentially optimal for *generic groups*. However, for *actual* groups, there may be better algorithms.

Generic Groups vs. Actual Groups

Can we do better?

The above is essentially optimal for *generic groups*. However, for *actual* groups, there may be better algorithms.

- In $(\mathbb{Z}/N\mathbb{Z}, +)$ DLOG is poly time; division modulo N.

Generic Groups vs. Actual Groups

Can we do better?

The above is essentially optimal for *generic groups*. However, for *actual* groups, there may be better algorithms.

- In $(\mathbb{Z}/N\mathbb{Z}, +)$ DLOG is poly time; division modulo N.
- In $(\mathbb{Z}/N\mathbb{Z})^\times$ DLOG is sub-exponential.

Generic Groups vs. Actual Groups

Can we do better?

The above is essentially optimal for *generic groups*. However, for *actual* groups, there may be better algorithms.

- In $(\mathbb{Z}/N\mathbb{Z}, +)$ DLOG is poly time; division modulo N.
- In $(\mathbb{Z}/N\mathbb{Z})^\times$ DLOG is sub-exponential.
- In $E(\mathbb{F}_q)$ we do not know any better algorithms.

Contents

Elliptic Curves - Very short intro

Elliptic Curves

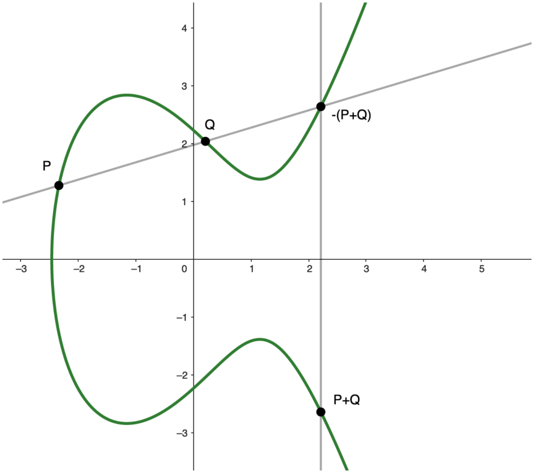
Let $A, B \in \mathbb{F}_q$. Then we can think of an elliptic curve $E/\mathbb{F}_q$ defined by A, B as the set

$$E = \{(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q \mid y^2 = x^3 + Ax + B\} \cup \{\infty\}$$

Incredible fact:

The set above can be given a group structure, where $P + Q$ can be computed from rational functions in $x(P), y(P), x(Q), y(Q)$.

Elliptic Curves - Very short intro



Elliptic Curves - Very short intro

Elliptic Curves

Let $A, B \in \mathbb{F}_q$. Then we can think of an elliptic curve $E/\mathbb{F}_q$ defined by A, B as the set

$$E = \{(x, y) \in \overline{\mathbb{F}}_q \times \overline{\mathbb{F}}_q \mid y^2 = x^3 + Ax + B\} \cup \{\infty\}$$

Incredible fact:

The set above can be given a group structure, where $P + Q$ can be computed from rational functions in $x(P), y(P), x(Q), y(Q)$.

Rational points

For any field $\mathbb{F}_q$ where E is defined, $E(\mathbb{F}_q)$ denotes the *subgroup* of $\mathbb{F}_q$-rational points on E (i.e. points P where $x(P), y(P) \in \mathbb{F}_q$).

The Hasse Interval

Theorem (Hasse)

Let $E/\mathbb{F}_q$ be an elliptic curve. Then

$$q + 1 - 2\sqrt{q} \leq \#E(\mathbb{F}_q) \leq q + 1 + 2\sqrt{q}$$

In fact, a pretty strong converse to this theorem also holds.
We need the following:

Theorem (Waterhouse/Deuring/Rück)

Let $N \in [p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$ be an integer. Then there exists an elliptic curve $E/\mathbb{F}_p$ with $E(\mathbb{F}_p) = \langle P \rangle$ cyclic of order N.

Contents

Den Boer

Setup

Assume that $G = \langle g \rangle$ is a group of prime order p, and that $p-1$ is (polynomially) smooth.

Further, let $\mathcal{O}$ be a CDH oracle, i.e. something which on input (g, g^a, g^b) returns g^{ab} in polynomial time.

Theorem (Den Boer)

Let G be as above, and assume we have access to $\mathcal{O}$. Then there exists a polynomial time algorithm for solving DLOG in G.

Intuitively, in these special cases, DLOG is equivalent to CDH.

Black box field arithmetic

Definition

Let $G = \langle g \rangle$ be a group of prime order p. We define the black-box field $(\boxed{\mathbb{F}_p}, +, \cdot)$ as:

- $\boxed{\mathbb{F}_p} = \{g^a \mid a \in \mathbb{Z}\}$ as sets.
- Addition: $g^a + g^b := g^a g^b$.
- Multiplication: $g^a \cdot g^b := g^{ab}$.

Black box field arithmetic

Definition

Let $G = \langle g \rangle$ be a group of prime order p. We define the black-box field $(\boxed{\mathbb{F}_p}, +, \cdot)$ as:

- $\boxed{\mathbb{F}_p} = \{g^a \mid a \in \mathbb{Z}\}$ as sets.
- Addition: $g^a + g^b := g^a g^b$.
- Multiplication: $g^a \cdot g^b := g^{ab}$.

Lemma

Let $\mathbb{F}_p$ denote the finite field $\mathbb{Z}/p\mathbb{Z}$. Then

$$\boxed{} : \mathbb{F}_p \to \boxed{\mathbb{F}_p}$$
$$\boxed{a} = g^a$$

is an isomorphism of fields.

Computing operations

Almost everything is easy to compute in $\boxed{\mathbb{F}_p}$.

- Computing $\boxed{a} + \boxed{b} := g^a g^b$ is efficient.
- Computing $\boxed{a} \cdot \boxed{b}$ requires computing g^{ab} from g^a and g^b. Precisely what $\mathcal{O}$ does!
- Computing $\boxed{a}$ from $a \in \mathbb{Z}/p\mathbb{Z}$ is simply computing g^a.
- Computing a from $\boxed{a}$ however is hard. In fact, this is precicely solving the DLOG instance (g, g^a).

Computing operations

Almost everything is easy to compute in $\boxed{\mathbb{F}_p}$.

- Computing $\boxed{a} + \boxed{b} := g^a g^b$ is efficient.
- Computing $\boxed{a} \cdot \boxed{b}$ requires computing g^{ab} from g^a and g^b. Precisely what $\mathcal{O}$ does!
- Computing $\boxed{a}$ from $a \in \mathbb{Z}/p\mathbb{Z}$ is simply computing g^a.
- Computing a from $\boxed{a}$ however is hard. In fact, this is precicely solving the DLOG instance (g, g^a).

Our Dlog instance...

We were given the Dlog instance (g, g^x). These objects of G can also be seen as elements of $\boxed{\mathbb{F}_p}$, namely $\boxed{1}$ and $\boxed{x}$. Magic: Using some algebraic relation on $\boxed{x}$, we can recover x.

Proof of Den Boer's Theorem

- Fix any generator r of $(\mathbb{F}_p)^\times$.

Proof of Den Boer's Theorem

- Fix any generator r of $(\mathbb{F}_p)^\times$.
- Compute $\boxed{r}$.

Proof of Den Boer's Theorem

- Fix any generator r of $(\mathbb{F}_p)^\times$.
- Compute $\boxed{r}$.
- Use generic group algorithms to solve the Dlog instance $(\boxed{r}, \boxed{x})$ in $(\boxed{\mathbb{F}_p})^\times$.
  - This crucially requires $\mathcal{O}$ (for multiplication in $\boxed{\mathbb{F}_p}$) and the fact that $p-1$ is smooth (Pohlig-Hellman in $(\boxed{\mathbb{F}_p})^\times$).

Proof of Den Boer's Theorem

- Fix any generator r of $(\mathbb{F}_p)^\times$.
- Compute $\boxed{r}$.
- Use generic group algorithms to solve the Dlog instance $(\boxed{r}, \boxed{x})$ in $(\boxed{\mathbb{F}_p})^\times$.
  - This crucially requires $\mathcal{O}$ (for multiplication in $\boxed{\mathbb{F}_p}$) and the fact that $p-1$ is smooth (Pohlig-Hellman in $(\boxed{\mathbb{F}_p})^\times$).
- Let y be the solution (i.e. $\boxed{r}^y = \boxed{x}$). Recover x as r^y

A fantastic trick

Limitations

The requirement that $p-1$ is smooth in Den Boer's reduction almost certainly not hold for random primes. The requirement came from the fact that we needed DLOG to be easy in $(\mathbb{F}_p)^\times$.

A fantastic trick

Limitations

The requirement that $p-1$ is smooth in Den Boer's reduction almost certainly not hold for random primes. The requirement came from the fact that we needed DLOG to be easy in $(\mathbb{F}_p)^\times$.

Brilliant idea!

Replace $\mathbb{F}_p^\times$ with some other algebraic group over $\mathbb{F}_p$!!

A fantastic trick

Limitations

The requirement that $p-1$ is smooth in Den Boer's reduction almost certainly not hold for random primes. The requirement came from the fact that we needed DLOG to be easy in $(\mathbb{F}_p)^\times$.

Brilliant idea!

Replace $\mathbb{F}_p^\times$ with some other algebraic group over $\mathbb{F}_p$!!

Theorem (Maurer)

Let G be a group of prime order p, and assume we have access to $\mathcal{O}$. Assume further that we are given an elliptic curve E with $\#E(\mathbb{F}_p)$ smooth. Then there exists a polynomial time algorithm for solving DLOG in G.

Intuitively, as soon as we know a smooth ordered algebraic group over $\mathbb{F}_p$ (e.g. $E(\mathbb{F}_p)$), DLOG is equivalent to CDH in G.

Black box curve

Corollary

The map

$$\boxed{P} : E(\mathbb{F}_p) \to E(\boxed{\mathbb{F}_p})$$
$$\boxed{P} = (\boxed{x(P)}, \boxed{y(P)})$$

is an isomorphism of groups.

Proof of Maurer's Theorem

Assume, for simplicity that $E(\mathbb{F}_p)$ is cyclic.

- Fix any generator P of $E(\mathbb{F}_p)$.
- Compute $\boxed{P} \in E(\boxed{\mathbb{F}_p})$.
- Compute the point $\boxed{Q} = (\boxed{x}, -)$.[2]
- Use generic group algorithms to solve the Dlog instance $(\boxed{P}, \boxed{Q})$ in $E(\boxed{\mathbb{F}_p})$.
- Let y be a solution (i.e. $[y]\boxed{P} = \boxed{Q}$). Recover x as the x-coordinate of $[y]P$.

[2]When x does not define a point on the curve, we can simply replace x by $x + d$ for any d that we know, and proceed as usual.

Are we done?

Limitations

Formally, this does NOT prove that CDH and DLOG are equivalent. But it comes *very* close.

- The existance of a polynomially smooth number in the Hasse interval $[p + 1 - 2\sqrt{p}, p + 1 + 2\sqrt{p}]$ is only conjectural.
- Bigger problem: Finding a curve of given order over $\mathbb{F}_p$ is generally hard.
- In practice, such curves are known for widely used p.
  - See May, Schneider (2023): Dlog is Practically as Hard (or Easy) as DH – Solving Dlogs via DH Oracles on EC Standards

Questions?