COL7160 : Quantum Computing

Lecture 15: Breaking Diffie–Hellman Key Exchange

Instructor: Rajendra Kumar **Scribe:** Vulasala Jayanth

1 Diffie–Hellman Key Exchange

Idea: The Diffie–Hellman key exchange allows two parties to establish a shared secret over a public channel.
Public Parameters: A cyclic group G of prime order p with generator g.
Private Inputs: Alice chooses a secret $a \in \mathbb{Z}_p$, and Bob chooses a secret $b \in \mathbb{Z}_p$.
Exchange:

- Alice computes $A = g^a \bmod p$ and sends A to Bob.
- Bob computes $B = g^b \bmod p$ and sends B to Alice.

Shared Key Computation:

- Alice computes $K = B^a = g^{ba} \bmod p$.
- Bob computes $K = A^b = g^{ab} \bmod p$.

Result: Both parties obtain the same shared key

$$K = g^{ab} \bmod p.$$

Security: An adversary observing (g, A, B) cannot efficiently compute K due to the hardness of the discrete logarithm problem.

2 Discrete Logarithm Problem

Input: A prime p, a generator g of the multiplicative group $\mathbb{Z}_p^*$, and an element $h \in \mathbb{Z}_p^*$.
Goal: Find $x \in \mathbb{Z}_{p-1}$ such that

$$g^x \equiv h \pmod{p}.$$

Assumption: There is no efficient classical algorithm to solve this problem for large p.

3 Breaking Diffie–Hellman Key Exchange

Input: A cyclic group G with generator g, and public values $A = g^a$, $B = g^b$.
Goal: Compute the shared key

$$K = g^{ab}.$$

Definition: The problem of computing g^{ab} from (g, g^a, g^b) is known as the Computational Diffie–Hellman (CDH) problem.

Remark 1. Shor [Sho94] also gave a quantum algorithm that solves the discrete logarithm problem in polynomial time. Consequently, cryptographic schemes such as RSA and Diffie–Hellman key exchange are not quantum-safe.

4 Reducing Discrete Logarithm to Order Finding

Problem: Given $h = g^x$ in $\mathbb{Z}_p^*$, find x.
Setup: Let g be a generator of $\mathbb{Z}_p^*$, so that $\mathrm{ord}(g) = p - 1$.

4.1 Order Relation

If $h = g^x$, then

$$\mathrm{ord}(h) = \frac{p-1}{\gcd(p-1, x)}.$$

Let $\delta = \mathrm{ord}(h)$. Then

$$x \cdot \delta \equiv 0 \pmod{p-1},$$

which implies

$$x \equiv 0 \pmod{\frac{p-1}{\delta}}.$$

If δ is small, we can try all possible candidates for x by computing g^y, where y is an integer multiple of $\frac{p-1}{\delta}$.

4.2 Using Randomization

Choose a uniformly random $y \in \mathbb{Z}_p^*$, and $a = g^y$. Then

$$h \cdot a = g^{x+y}.$$

Let $\delta' = \mathrm{ord}(h \cdot a)$. Then

$$x + y \equiv 0 \pmod{\frac{p-1}{\delta'}},$$

which implies

$$x \equiv -y \pmod{\frac{p-1}{\delta'}}.$$

4.3 Combining Information

Repeating the above process yields congruences:

$$x \equiv -y_i \pmod{\frac{p-1}{\delta_i}}.$$

These can be combined to recover x modulo

$$\mathrm{lcm}\left(\frac{p-1}{\delta_1}, \frac{p-1}{\delta_2}, \dots\right).$$

The process of combining information from multiple moduli relies on the structure of the group. For example, if we consider $\mathbb{Z}_N^*$, it is isomorphic to $\mathbb{Z}_{p_1}^* \times \mathbb{Z}_{p_2}^*$ via the Chinese Remainder Theorem, allowing us to recover x by gathering information about its residues in these sub-structures.

Algorithm 1 Discrete Logarithm via Randomization (Classical Sketch)

Input: Generator g, target $h = g^x \in \mathbb{Z}_p^*$
Output: Secret x
$\mathcal{S} \leftarrow \emptyset$ ▷ Set of congruences
repeat
    Pick random $y \in \mathbb{Z}_{p-1}$
    Compute $a = g^y$ and $\delta' = \mathrm{ord}(h \cdot a)$
    Add $x \equiv -y \pmod{\frac{p-1}{\delta'}}$ to $\mathcal{S}$
    $M \leftarrow \mathrm{lcm}(\text{moduli in } \mathcal{S})$
until $M = p - 1$
return $x \pmod{p-1}$

4.4 Case Analysis Based on Order

Let $r = \mathrm{ord}(h)$.
Case 1: r is small.
Then $\frac{p-1}{r}$ is large, and we obtain

$$x \equiv 0 \pmod{\frac{p-1}{r}},$$

which gives strong information about x. In this case, x can be determined efficiently.
Case 2: r is large.
Then $\frac{p-1}{r}$ is small, yielding weak information about x. However, by repeating the process with random elements $a = g^y$, we obtain relations of the form

$$x \equiv -y \pmod{\frac{p-1}{r'}}.$$

Combining multiple such congruences allows recovery of x modulo a larger modulus.

4.5 Limitations of the Reduction: The Case $p - 1 = 2q$

While the reduction of the Discrete Logarithm Problem to Order Finding is theoretically sound, its efficiency depends heavily on the arithmetic structure of the group order $p - 1$. A particularly difficult case occurs when $p - 1 = 2q$, where q is a large prime (often referred to as a "safe prime" structure).
In such a group, by Lagrange's Theorem , the possible orders for any element $h \in \mathbb{Z}_p^*$ are $\{1, 2, q, 2q\}$. If we attempt to recover x using the relation $x \equiv -y \pmod{\frac{p-1}{\delta}}$, we analyze the information gain based on the observed order δ:

1. **Case $\delta = q$:** The modulus becomes $\frac{p-1}{q} = \frac{2q}{q} = 2$. This yields the congruence:

   $$x \equiv -y \pmod 2$$

   This provides only **one bit** of information (the parity of x).

2. **Case $\delta = 2q$:** The modulus becomes $\frac{p-1}{2q} = 1$. This yields the trivial congruence:

   $$x \equiv -y \pmod 1$$

   This provides **zero bits** of information, as every integer satisfies a congruence modulo 1.

Remark 2. Since the majority of elements in $\mathbb{Z}_p^*$ have order q or $2q$, randomizing the input $h \cdot g^y$ will almost always result in a modulus of 1 or 2. To recover the full value of x (which is $\approx \log_2 p$ bits long), we would require an exponential number of iterations to gather enough information.

Conclusion: For groups with large prime factors in their order, the simple reduction to order-finding via modular congruences is classically inefficient.

5 The Hidden Subgroup Problem and Group Theory Background

Before defining the Hidden Subgroup Problem, we establish the necessary group-theoretic foundations regarding subgroups and cosets.

5.1 Mathematical Background

Definition 3 (Subgroup)**.** A subset $H \subseteq G$ is called a *subgroup* (denoted $H \leq G$) if H is itself a group under the operation of G.

Lemma 4 (Subgroup Test)**.** *A nonempty subset $H \subseteq G$ is a subgroup if and only if $\forall a, b \in H$, $ab^{-1} \in H$.*

Proof. Let $a \in H$. Then $aa^{-1} = e \in H$, so the identity exists. Since $e, a \in H$, then $ea^{-1} = a^{-1} \in H$, so inverses exist. Closure follows since for $a, b \in H$, we have $b^{-1} \in H$, and thus $a(b^{-1})^{-1} = ab \in H$. □

Definition 5 (Cosets)**.** Let $H \leq G$ and $g \in G$. The *left coset* of H represented by g is defined as:

$$gH = \{gh : h \in H\}.$$

Theorem 6 (Disjointness of Cosets)**.** *For any $g, g' \in G$, the left cosets gH and $g'H$ are either identical or completely disjoint. That is, $gH = g'H$ or $gH \cap g'H = \emptyset$.*

Proof. Suppose $gH \cap g'H \neq \emptyset$. Then there exists some x such that $x = gh_1 = g'h_2$ for $h_1, h_2 \in H$. This implies $g^{-1}g' = h_1 h_2^{-1} \in H$. Let $g^{-1}g' = h \in H$. Then $g' = gh$. For any element $g'h' \in g'H$, we have $g'h' = (gh)h' = g(hh') \in gH$, proving $g'H \subseteq gH$. By symmetry, $gH \subseteq g'H$, so $gH = g'H$. □

Remark 7 (Lagrange's Theorem). The cosets of H partition the group G. A consequence of this is Lagrange's Theorem, which states that $|H|$ must divide $|G|$. Thus, the size of any subgroup is always a factor of the size of the parent group.

5.2 Problem Formulation: Hidden Subgroup Problem

Input: A finite group G and a function $f : G \to X$ (where X is a finite set).
Promise: There exists a subgroup $H \leq G$ such that the function f *hides* H. This means:

$$f(g) = f(g') \iff g^{-1}g' \in H \iff gH = g'H.$$

Goal: Determine the subgroup H efficiently.

5.3 Properties and Generating Sets

By the promise above, the function f exhibits specific behavior:

1. **Constant on cosets:** $f(g) = f(gh)$ for all $h \in H$.
2. **Distinct on different cosets:** If $g_1 H \neq g_2 H$, then $f(g_1) \neq f(g_2)$.

To "determine" H efficiently, we do not list all its elements. Instead, we look for a **generating set** S such that $H = \langle S \rangle$.
We will see the definition of generating set in next lecture.

References

[Sho94] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.