DD2448 Foundations of Cryptography
Lecture 12

Douglas Wikström
KTH Royal Institute of Technology
dog@kth.se

April 28, 2020

Discrete Logarithm (1/2)

Definition. Let G be a cyclic group of order q and let g be a generator G. The **discrete logarithm** of $y \in G$ in the basis g (written $\log_g y$) is defined as the unique $x \in \{0, 1, \ldots, q-1\}$ such that

$$y = g^x \ .$$

Compare with a "normal" logarithm! ($\ln y = x$ iff $y = e^x$)

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$?

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? $(9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9)$

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? ($9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9$)

Example. 7 is a generator of $\mathbb{Z}_{13}^*$.

What is $\log_7 9$?

Discrete Logarithm (2/2)

Example. 7 is a generator of $\mathbb{Z}_{12}$ additively, since $\gcd(7, 12) = 1$.

What is $\log_7 3$? $(9 \cdot 7 = 63 = 3 \bmod 12$, so $\log_7 3 = 9)$

Example. 7 is a generator of $\mathbb{Z}_{13}^*$.

What is $\log_7 9$? $(7^4 = 9 \bmod 13$, so $\log_7 9 = 4)$

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n \in \mathbb{N}}$ by G.

Definition. The **Discrete Logarithm (DL) Assumption** in G states that if generators g_n and y_n of G_{q_n} are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g_n, y_n) = \log_{g_n} y_n\right]$$

is negligible.

Discrete Logarithm Assumption

Let G_{q_n} be a cyclic group of prime order q_n such that $\lfloor \log_2 q_n \rfloor = n$ for $n = 2, 3, 4, \ldots$, and denote the family $\{G_{q_n}\}_{n\in\mathbb{N}}$ by G.

Definition. The **Discrete Logarithm (DL) Assumption** in G states that if generators g and y of G are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g, y) = \log_g y\right]$$

is negligible.

We usually remove the indices from our notation!

Definition. Let g be a generator of G. The **Diffie-Hellman (DH) Assumption** in G states that if $a, b \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\Pr\left[A(g^a, g^b) = g^{ab}\right]$$

is negligible.

Decision Diffie-Hellman Assumption

Definition. Let g be a generator of G. The **Decision Diffie-Hellman (DDH) Assumption** in G states that if $a, b, c \in \mathbb{Z}_q$ are randomly chosen, then for every polynomial time algorithm A

$$\left|\Pr\left[A(g^a, g^b, g^{ab}) = 1\right] - \Pr\left[A(g^a, g^b, g^c) = 1\right]\right|$$

is negligible.

Relating DL Assumptions

- Computing discrete logarithms is at least as hard as computing a Diffie-Hellman element g^{ab} from g^a and g^b.
- Computing a Diffie-Hellman element g^{ab} from g^a and g^b is at least as hard as distinguishing a Diffie-Hellman triple (g^a, g^b, g^{ab}) from a random triple (g^a, g^b, g^c).
- In most groups where the DL assumption is conjectured, DH and DDH assumptions are conjectured as well.
- There exists special elliptic curves where DDH problem is easy, but DH assumption is conjectured!

Security of El Gamal

- Finding the secret key is equivalent to DL problem.
- Finding the plaintext from the ciphertext and the public key and is equivalent to DH problem.
- The CPA security of El Gamal is equivalent to DDH problem.

Brute Force and Shank's

Let G be a cyclic group of order q and g a generator. We wish to compute $\log_g y$.

- **Brute Force.** $O(q)$
- **Shanks.** Time and **Space** $O\left(\sqrt{q}\right)$.
  1. Set $z = g^m$ (think of m as $m = \sqrt{q}$).
  2. Compute z^i for $0 \leq i \leq q/m$.
  3. Find $0 \leq j \leq m$ and $0 \leq i \leq q/m$ such that $yg^j = z^i$ and output $x = mi - j$.

Birthday Paradox

Lemma. Let $q_0, \ldots, q_k$ be randomly chosen in a set S. Then

1. the probability that $q_i = q_j$ for some $i \neq j$ is approximately $1 - e^{-\frac{k^2}{2s}}$, where $s = |S|$, and
2. with $k \approx \sqrt{-2s \ln(1-\delta)}$ we have a collision-probability of δ.

Proof.

$$\left(\frac{s-1}{s}\right)\left(\frac{s-2}{s}\right) \cdot \ldots \cdot \left(\frac{s-k}{s}\right) \approx \prod_{i=1}^{k} e^{-\frac{i}{s}} \approx e^{-\frac{k^2}{2s}} \quad .$$

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1} g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1} y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!

Pollard-ρ (1/2)

Partition G into S_1, S_2, and S_3 "randomly".

- Generate "random" sequence $\alpha_0, \alpha_1, \alpha_2 \ldots$

$$\alpha_0 = g$$

$$\alpha_i = \begin{cases} \alpha_{i-1}g & \text{if } \alpha_{i-1} \in S_1 \\ \alpha_{i-1}^2 & \text{if } \alpha_{i-1} \in S_2 \\ \alpha_{i-1}y & \text{if } \alpha_{i-1} \in S_3 \end{cases}$$

- Each $\alpha_i = g^{a_i} y^{b_i}$, where $a_i, b_i \in \mathbb{Z}_q$ are known!
- If $\alpha_i = \alpha_j$ and $(a_i, b_i) \neq (a_j, b_j)$ then $y = g^{(a_i - a_j)(b_j - b_i)^{-1}}$.

Pollard-ρ (2/2)

- If $\alpha_i = \alpha_j$, then $\alpha_{i+1} = \alpha_{j+1}$.
- The sequence $(a_0, b_0), (a_1, b_1), \ldots$ is "essentially random".
- The Birthday bound implies that the (heuristic) expected running time is $O(\sqrt{q})$.
- We use "double runners" to reduce memory.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
  1. Choose $s_j \in \mathbb{Z}_q$ randomly and attempt to factor $g^{s_j} = \prod_i p_i^{e_{j,i}}$ as an **integer**.
  2. If g^{s_j} factored in $\mathcal{B}$ and $e_j = (e_{j,1}, \ldots, e_{j,B})$ is linearly independent of $e_1, \ldots, e_{j-1}$, then $j \leftarrow j + 1$.
  3. If $j < B$, then go to (1)

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.

Index Calculus

- Let $\mathcal{B} = \{p_1, \ldots, p_B\}$ be a set of small prime **integers**.
- Compute $a_i = \log_g p_i$ for all $p_i \in \mathcal{B}$.
- Repeat:
  1. Choose $s \in \mathbb{Z}_q$ randomly.
  2. Attempt to factor $yg^s = \prod_i p_i^{e_i}$ as an **integer**.
  3. If a factorization is found, then output $\left(\sum_i a_i e_i - s\right) \bmod q$.

Excercise: Why doesn't this work for any cyclic group?