# One-way/Trapdoor Functions

- ★ For public key cryptography, we need functions
  - ▶ Easy to calculate, given the secret
  - ▶ Hard to invert if you do not know the secrets

- ★ We have few candidates
  - ▶ Discrete Logarithm
  - ▶ Factorization
  - ▶ Linear decoding

# Discrete Logarithm Related Problems

- ★ Let $G$ be a cyclic group where $|G| = m$
- ★ Let $g \in G$ be a generator of G
- ★ Discrete Logarithm function $DLog_{G,g} : G \mapsto Z_m$

$$DLog_{G,g}(a) = i : \text{ if } g^i = a$$

$\mathcal{O}(m)$

$g^0, \ldots, g^m$

| Problem | Given | Figure out |
|---|---|---|
| Discrete logarithm (DL) | $g^x$ | $x$ |
| Computational Diffie-Hellman (CDH) | $g^x, g^y$ | $g^{xy}$ |
| Decisional Diffie-Hellman (DDH) | $g^x, g^y, g^z$ | Is $z \equiv xy \pmod{\|G\|}$? |

$Z_p^*$ lo poly ( log(m) )

# Discrete Logarithm Problem

- ★ Let $G$ be a cyclic group where $|G| = m$
- ★ Let $g \in G$ be a generator of G
- ★ Let $A$ be an algorithm that returns $i \in Z_m$
- ★ We consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{dl}}_{G,g}(A)$

$x \xleftarrow{\$} \mathbf{Z}_m \,;\, X \leftarrow g^x$

$\overline{x} \leftarrow A(X)$

If $g^{\overline{x}} = X$ then return 1 else return 0

The *dl-advantage* of $A$ is defined as

$$\mathbf{Adv}^{\mathrm{dl}}_{G,g}(A) = \Pr\left[\mathbf{Exp}^{\mathrm{dl}}_{G,g}(A) = 1\right] . \quad \blacksquare$$

# Diffie-Hellman Key Exchange

★ Let $G$ be a cyclic group where $|G| = m$

★ Let $g \in G$ be a generator of G

private

★ Alice announces $X = g^x$ for random $x \in Z_m$

public

★ Bob announces $Y = g^y$ for random $y \in Z_m$

private

★ Alice and Bob set $g^{xy}$ as the joint key

Note $X^y = \underline{Y^x} = \underline{g^{xy}}$

★ Diffie-Hellman assumption: $g^x$ $g^y$

▶ Hard to calculate $\underline{g^{xy}}$ from $X$ and $Y$

# Computational Diffie-Hellman

- ★ Let $G$ be a cyclic group where $|G| = m$
- ★ Let $g \in G$ be a generator of G
- ★ Let $A$ be an algorithm that returns $b \in G$
- ★ We consider the following experiment:

Experiment $\mathbf{Exp}^{\mathrm{cdh}}_{G,g}(A)$

$x \xleftarrow{\$} \mathbf{Z}_m$ ; $y \xleftarrow{\$} \mathbf{Z}_m$

$X \leftarrow g^x$ ; $Y \leftarrow g^y$

$Z \leftarrow A(X, Y)$

If $Z = g^{xy}$ then return 1 else return 0

The *cdh-advantage* of $A$ is defined as

$$\mathbf{Adv}^{\mathrm{cdh}}_{G,g}(A) \quad = \quad \Pr\left[\mathbf{Exp}^{\mathrm{cdh}}_{G,g}(A) = 1\right] . \quad \blacksquare$$

# Decisional Diffie-Hellman problem

- Let $G$ be a cyclic group where $|G| = m$
- Let $g \in G$ be a generator of G
- Adversary is given $X = g^x$, $Y = g^y$ for random $x, y \in Z_m$ and $Z$
- In world 0:
  - $Z = g^z$ for random $z \in Z_m$
- In world 1:
  - $Z = g^{xy}$

$g^x, g^y \Rightarrow g^{xy}$

$(g^x, g^y, g^{xy})$ ← world 1

$(g^x, g^y, g^z)$ world 0

random

# Decisional Diffie-Hellman Problem

- ★ Let $G$ be a cyclic group where $|G| = m$
- ★ Let $g \in G$ be a generator of G
- ★ $A$ returns a bit $b \in \{0, 1\}$
- ★ We consider the following experiment:

| Experiment $\mathbf{Exp}^{\text{ddh-1}}_{G,g}(A)$ | Experiment $\mathbf{Exp}^{\text{ddh-0}}_{G,g}(A)$ |
|---|---|
| $x \xleftarrow{\$} \mathbf{Z}_m$ | $x \xleftarrow{\$} \mathbf{Z}_m$ |
| $y \xleftarrow{\$} \mathbf{Z}_m$ | $y \xleftarrow{\$} \mathbf{Z}_m$ |
| $z \leftarrow xy \bmod m$ | $z \xleftarrow{\$} \mathbf{Z}_m$ |
| $X \leftarrow g^x \,;\, Y \leftarrow g^y \,;\, Z \leftarrow g^z$ | $X \leftarrow g^x \,;\, Y \leftarrow g^y \,;\, Z \leftarrow g^z$ |
| $d \leftarrow A(X, Y, Z)$ $(z \leftarrow g^{xy})$ | $d \leftarrow A(X, Y, Z)$ |
| Return $d$ | Return $d$ |

The *ddh-advantage* of $A$ is defined as

$$\mathbf{Adv}^{\text{ddh}}_{G,g}(A) \;=\; \Pr\left[\mathbf{Exp}^{\text{ddh-1}}_{G,g}(A) = 1\right] - \Pr\left[\mathbf{Exp}^{\text{ddh-0}}_{G,g}(A) = 1\right] .$$

# Relationships between Problems

- Let $G$ be a cyclic group where $|G| = m$
- Let $g \in G$ be a generator of G

$g^x \Rightarrow x$

- Let $A_{dl}$ be an adversary against DL problem
- Let $A_{cdh}$ be an adversary against CDH problem
- Let $A_{ddh}$ be an adversary against DDH problem
- Proposition 7.4:

$g^x, g^y \Rightarrow g^{xy}$

$(g^x, g^y, g^{xy})$ $(g^x, g^y, g^z) \Rightarrow \{0,1\}$

$$\mathrm{Adv}^{dl}_{G,g}(A_{dl}) \leq Adv^{cdh}_{G,g}(A_{cdh}) \leq Adv^{ddh}_{G,g}(A_{ddh}) + \frac{1}{|G|}$$

UTD

# Proof of Proposition 7.4

- ★ Define $A_{cdh}$ given $A_{dl}$

  Adversary $A_{cdh}(X, Y)$
  
  $\bar{x} \leftarrow A_{dl}(X)$
  
  Return $\cancel{Z^{\bar{x}}}$ $Y^{\bar{x}}$

- ★ If $A_{dl}$ is successful then $Y^{\bar{x}} = Y^{x} = (g^{y})^{x} = g^{xy}$
- ★ Define $A_{ddh}$ given $A_{cdh}$

  Adversary $A_{ddh}(X, Y, Z)$
  
  $\bar{Z} \leftarrow A_{cdh}(X, Y)$
  
  Return $(Z = \bar{Z})$

- ★ Claim:

$$\Pr\left[\mathbf{Exp}^{\text{ddh-1}}_{G,g}(A_{\text{ddh}}) = 1\right] = \mathbf{Adv}^{\text{cdh}}_{G,g}(A_{\text{cdh}})$$

$$\Pr\left[\mathbf{Exp}^{\text{ddh-0}}_{G,g}(A_{\text{ddh}}) = 1\right] = \frac{1}{|G|},$$

$\mathrm{Adv}^{ddh}_{G,g}(A_{ddh}) = \mathrm{Adv}(A_{cdh}) - \frac{1}{G}$

# The Choice of the Group

- For any reasonable $G$, an algorithm:
  - Finds the Discrete Logarithm in $O(|G|^{\frac{1}{2}})$ $= O(\sqrt{|G|})$

Trivial runs $O(|G|)$

- Two important algorithms for general groups
  - Pollards algorithm
  - Shanks baby-step giant-step algorithm
- We will explore Shanks algorithm as an example

Algorithm $A_{\mathrm{bsgs}}(X)$
$n \leftarrow \lceil \sqrt{m} \rceil$ ; $N \leftarrow g^n$
For $b = 0, \ldots, n$ do $B[X g^{-b}] \leftarrow b$
For $a = 0, \ldots, n$ do
$Y \leftarrow N^a$
If $B[Y]$ is defined then $x_0 \leftarrow B[Y]$ ; $x_1 \leftarrow a$
Return $a x_1 + x_0$

$B[Y] = B[X g^{-b}]$

$\Rightarrow$

$Y = X g^{-b}$

$N^a = X g^{-b}$

![UTD logo]

# Shank's DL Algorithm

- ★ Given $|G| = m$ and $n \leftarrow \lceil \sqrt{(m)} \rceil$

- ★ Let $N \leftarrow g^n$

  $X = g^{x}$

- ★ Note that for any $x \in Z_m$,
  $x = nx_1 + x_0$ for $0 \le x_0, x_1 \le n$

  $x_0, x_1 \Rightarrow X$

- ★ $\underline{g^x = g^{nx_1+x_0} = X}$ implies $\boxed{Xg^{-x_0} = g^{nx_1}}$

- ★ Shanks Algorithms Idea:
  - ▶ Find $a, b$ s.t. $Xg^{-b} = \cancel{g^{na}}$ $(g^n)^a$

- ★ Running time is $O(|G|^{\frac{1}{2}})$

# Integer modulo a prime

- Let $G = Z_p^*$ and $g$ is a generator of $G$
- Solving DDH is easy in $Z_p^*$
- For any $p \geq 3$, there exists $A$ attacking DDH problem s.t. $A$ has
  - running time $O(|p|^3)$
  - $Adv_{G,g}^{ddh}(A) = \frac{1}{2}$
- Currently best known solution for CDH is through solving DL
- There may be other solutions for CDH without solving DL
- General Number Field Sieve finds DL in Not poly(log(p))

$$O(e^{(C+o(1)).\ln(p)^{1/3}.(\ln(\ln(p)))^{1/3}})$$

# Integer modulo a prime

- ★ If the factorization of $p-1$ has all small factors then DL is easy to solve
- ★ In Practice, make sure that $p-1$ has a large prime divisor
- ★ Common choice:
  - ▶ $p = sq + 1$ for $s \geq 2$ and $q$ is prime
- ★ Constants are important in practice
- ★ Parallel and distributed implementations can decrease running time
- ★ 1024 bit $p$ are needed/recommended in commercial applications

# The RSA System

★ Let $N = pq$ for primes $p$ and $q$

★ Let $ed = 1 \bmod \phi(N)$

★ $RSA_{N,e} : Z_N^* \mapsto Z_N^*$ s.t $RSA_{N,e}(m) = m^e \bmod N$

★ Note that

$$
\begin{aligned}
RSA_{N,d}(RSA_{N,e}(x)) &= (x^e)^d \bmod N \\
&= x^{ed} \bmod N \\
&= x^{k\phi(n)+1} \bmod N \\
&= \cancel{1}\ x
\end{aligned}
$$

# The RSA System

- RSA assumption:
  - Given $e, N, RSA_{N,e}(m)$, it is hard to find $m$
    $e, N, m^e \bmod N$ '' '' $m$
- Note that given $e$ and $\phi(N)$, it is easy to find $d$
- In practice, we need efficient ways to find
  - $k$ bit long primes $p$ and $q$

# Miller-Rabin Primality Test

- ★ Primality test can be done in deterministic polynomial time
- ★ Deterministic primality test is slow in practice
- ★ Miller-Rabin Test is a randomized test
- ★ Note that for prime $p$ and $p-1 = 2^s m$ and $a \in Z_p^*$
  - ▶ $a^m = 1 \bmod p$
  - ▶ or $a^{2^j m} = -1 \bmod p$ for $0 \leq j \leq s-1$

# Miller-Rabin Primality Test

- ★ $N$ is odd composite number where $N-1=2^s r$
- ★ Let $a \in \{0, N-1\}$

  $\Pr[a \text{ passes test} \mid N \text{ is prime}] = 1$

  $\Pr[a \text{ '' } \text{ '' } \mid N \text{ is composite}] \leq \frac{1}{4}$
- ★ $a$ is strong witness if
  - ▶ $a^r \neq 1$
  - ▶ ~~or~~ and $a^{2^j r} \neq -1 \bmod N$ for $0 \leq j \leq s-1$
- ★ $a$ is a strong liar if it is not a strong witness
- ★ For composite $N$, there are at most $N/4$ strong liars

# Miller-Rabin Primality Test

MILLER-RABIN($n,t$)

INPUT: an odd integer $n \geq 3$ and security parameter $t \geq 1$.

OUTPUT: an answer "prime" or "composite" to the question: "Is $n$ prime?"

1. Write $\boxed{n-1=2^s r}$ such that $r$ is odd.
2. For $i$ from 1 to $t$ do the following:
   - 2.1 Choose a random integer $a$, $2 \leq a \leq n-2$.
   - 2.2 Compute $\underline{y = a^r \bmod n}$ using Algorithm 2.143.
   - 2.3 If $\underline{y \neq 1}$ and $\underline{y \neq n-1}$ then do the following:
     - $j \leftarrow 1$.
     - While $j \leq s-1$ and $y \neq n-1$ do the following:
       - Compute $y \leftarrow y^2 \bmod n$.
       - If $y = 1$ then return("composite").
       - $j \leftarrow j+1$.
     - If $y \neq n-1$ then return ("composite").
3. Return("prime").

$\frac{1}{4}$

★ For any $n$ composite, the error probability of Miller-Rabin is less than $O(\frac{1}{4}^t)$

$2^{-100}$