Cryptography

Lecture 12: Elliptic Curves

January 14, 2025

Overview

Contents

1. Classical cryptography
   (Shift & Vigenère cipher, one-time pad, perfect secrecy)
2. Security definitions & threat models
   (Computational security, CPA & CCA)
3. Private-key cryptography
   (Message authentication, hash functions, primitives, relevant ciphers)
4. Public-key cryptography
   (Assumptions, key management, digital signatures, relevant ciphers)

RSA Cipher

Definition (§10.11 Plain RSA)

Let f be RSA key generator, for which RSA problem is hard.
Public-key encryption scheme $\mathrm{RSA} = (\mathrm{gen}, \mathrm{enc}, \mathrm{dec})$ given by

- $\mathrm{gen}(1^n)$ runs $(\ell, e, d) \leftarrow f(1^n)$ and returns $\langle (\ell, e), (\ell, d) \rangle$
- $\mathrm{enc}_{\langle \ell, e \rangle}(m) = m^e \bmod \ell$ for all $m \in \mathbb{Z}_\ell^*$
- $\mathrm{dec}_{\langle \ell, d \rangle}(c) = c^d \bmod \ell$ for all $c \in \mathbb{Z}_\ell^*$

RSA Cipher

Limits of RSA assumption

- Requires uniformly chosen ciphertexts c (and thus messages)
  (typically not fulfilled in applications)
- Makes it hard to compute $\sqrt[e]{c}$ exactly
  (CPA-secure requires no advantage on even partial information)

Notes

- CPA-secure for single message bit with random padding
  (Section 11.5.3 of [Katz & Lindell])
- Expected to be CPA-secure with sufficiently long random padding
  (RSA PKCS #1 v1.5 padding too short; provably not CPA-secure)
- RSA-OAEP (RSA PKCS #1 v2.0) expected to be CCA-secure
  (provably CCA-secure using random oracle model)
- Many attacks known; recommended key length at least 2,048 bits

§10.12 Definition (Cyclic group)

Let $\mathbb{G} = (G, \cdot, 1)$ be finite commutative group. For every $g \in G$ we let

$$\langle g \rangle_{\mathbb{G}} = \{g^i \mid i \in \mathbb{N}\}$$

be subgroup generated by generator g

- Order of $g \in G$ is $\mathrm{ord}_{\mathbb{G}}(g) = |\langle g \rangle_{\mathbb{G}}|$
- $\mathbb{G}$ is cyclic if there exists $g \in G$ with $\langle g \rangle_{\mathbb{G}} = G$

Notes

- Lagrange's theorem shows $\mathrm{ord}_{\mathbb{G}}(g) \mid \ell$ with $\ell = |G|$ for every $g \in G$ (i.e. order of g divides order of $\mathbb{G}$)
- If $|G|$ prime, then $\mathbb{G}$ is cyclic and each $g \in G \setminus \{1\}$ generates G
- All finite cyclic groups of same order are isomorphic

§11.1 Theorem

1. $\mathbb{Z}_p^*$ is cyclic for every prime $p \in \mathbb{N}$
2. Any commutative group $\mathbb{G}$ with prime $|\mathbb{G}|$ is cyclic
   (in this case every non-unit is generator)

Proof

Nontrivial exercise □

Example

- Let $\mathbb{Z}_7^* = \{1, 2, \ldots, 6\}$ which does not have prime order (order 6)
- $\{2^i \mid i \in \mathbb{N}\} = \{1, 2, 4\}$, so 2 is no generator of $\mathbb{Z}_7^*$
- $\{3^i \mid i \in \mathbb{N}\} = \{1, 3, 2, 6, 4, 5\} = \mathbb{Z}_7^*$, so 3 is generator of $\mathbb{Z}_7^*$

§11.2 Definition (Discrete logarithm)

Let $\mathbb{G} = (G, \cdot, 1)$ be finite cyclic group with generator $g \in G$.
For every $a \in G$ the unique $0 \leq i \leq |G| - 1$ such that $g^i = a$ is called discrete logarithm of a (for base g) and denoted by $\log_g(a)$.

Notes

- Easy to compute g^i given g and $i \in \mathbb{N}$
- Seems hard to compute $\log_g(a)$ given g and $a \in G$
  (at least in some groups)
- Can be simple: consider cyclic group $(\mathbb{Z}_7, +, 0)$ of prime order 7
  and any generator g (any $g \in \{1, \ldots, 6\}$)
  $g^i = \underbrace{g + \cdots + g}_{i \text{ summands}} = ig$, so $\log_g(a) = g^{-1}a$ which is simple to compute

§11.3 Definition (Cyclic group generator)

Cyclic group generator $\mathcal{G}$ is deterministic polynomial-time algorithm that given security parameter $n \in \mathbb{N}$ returns $\mathcal{G}(1^n) = (\mathbb{G}, q, g)$ such that

- $\mathbb{G}$ is finite cyclic group of order q with $2^{n-1} < q < 2^n$
  (suitably & efficiently represented)
- generator $g \in \mathbb{G}$ with $\mathrm{ord}_{\mathbb{G}}(g) = |\mathbb{G}|$

Typical implementation

- Output of $\mathcal{G}$ not secret (as opposed to RSA key generator)
- NIST suggests suitable groups & generators
- Problem difficulty depends crucially on group
- Prime order groups strongly prefered
  (Careful: $\mathbb{Z}_p^*$ with p prime does typically not have prime order $p - 1$)

Standard group choices

1. Prime order subgroups of $\mathbb{Z}_p^*$ with p prime
   - Let $p = tq + 1$ with q prime (compute prime factor q of $p - 1$)
   - Then $\{a^t \bmod p \mid a \in \mathbb{Z}_p^*\}$ commutative subgroup of order q
   - Since it has prime order, it is cyclic by Theorem §11.1
     (actually all subgroups of cyclic groups are themselves cyclic)

   NIST recommendation: at least $2,048$ bit p and 224 bit order q

2. Elliptic curves

   NIST recommendation: at least 224 bit order q

Elliptic Curves

§11.4 Definition (Elliptic curve)

Let $p \in \mathbb{N}$ be prime and all operations are performed in field $\mathbb{Z}_p$.
Additionally, let $a, b \in \mathbb{Z}_p$ be such that $4a^3 + 27b^2 \neq 0 \bmod p$.
Elliptic curve given by a and b is

$$\mathfrak{E}_p(a, b) = \{\infty\} \cup \{(x, y) \in \mathbb{Z}_p \times \mathbb{Z}_p \mid y^2 = x^3 + ax + b \bmod p\}$$

Notes

- Essentially curve $f(x) = x^3 + ax + b$ over rationals $\mathbb{Q}$,
  but only at points $f(x)$ which are squares
- Symmetric to x-axis (because each nonzero square has two roots)

Computing squares in $\mathbb{Z}_{11}$

- Squares are $\{0, 1, 4, 9, 5, 3\}$ with following roots

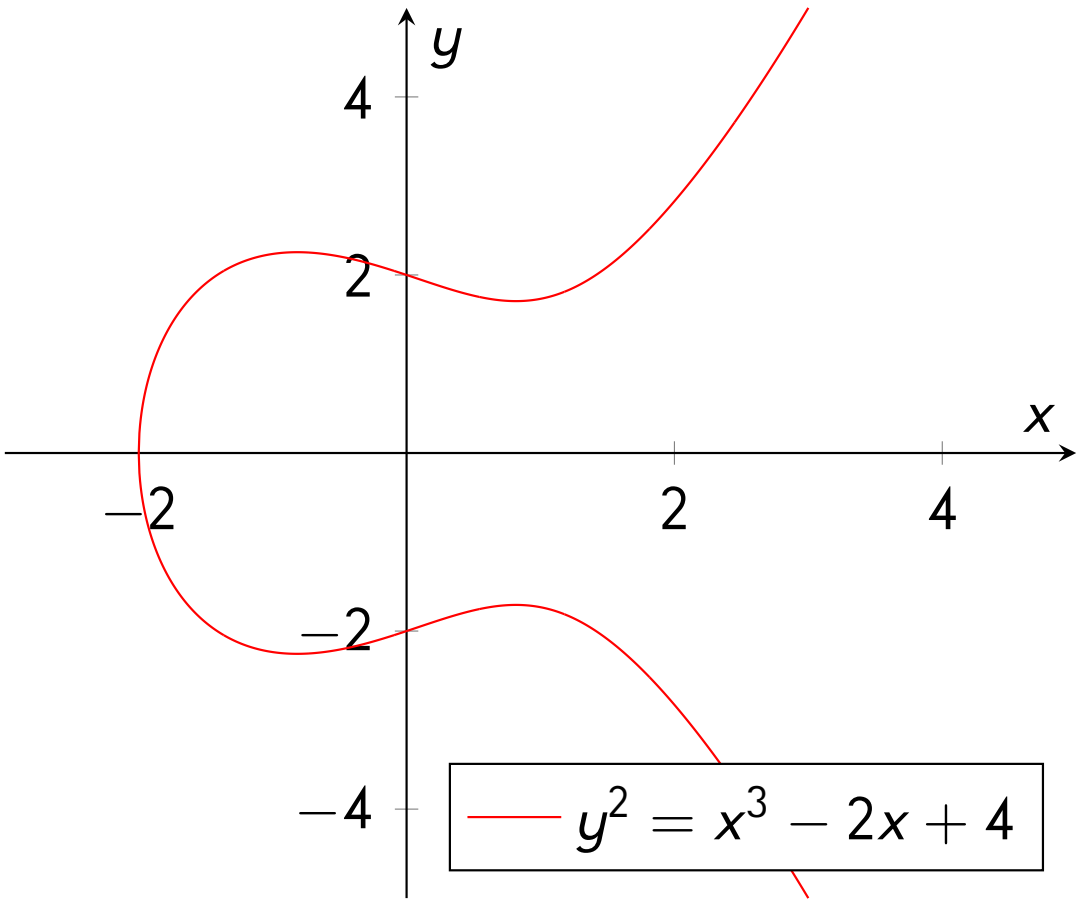
$$
\begin{aligned}
0^2 &= 0 \\
1^2 = 1 &= (-1)^2 = 10^2 \bmod 11 \\
2^2 = 4 &= (-2)^2 = 9^2 \bmod 11 \\
3^2 = 9 &= (-3)^2 = 8^2 \bmod 11 \\
4^2 = 5 &= (-4)^2 = 7^2 \bmod 11 \\
5^2 = 3 &= (-5)^2 = 6^2 \bmod 11
\end{aligned}
$$

- Compute points on elliptic curve $y^2 = f(x) = x^3 - 2x + 4$

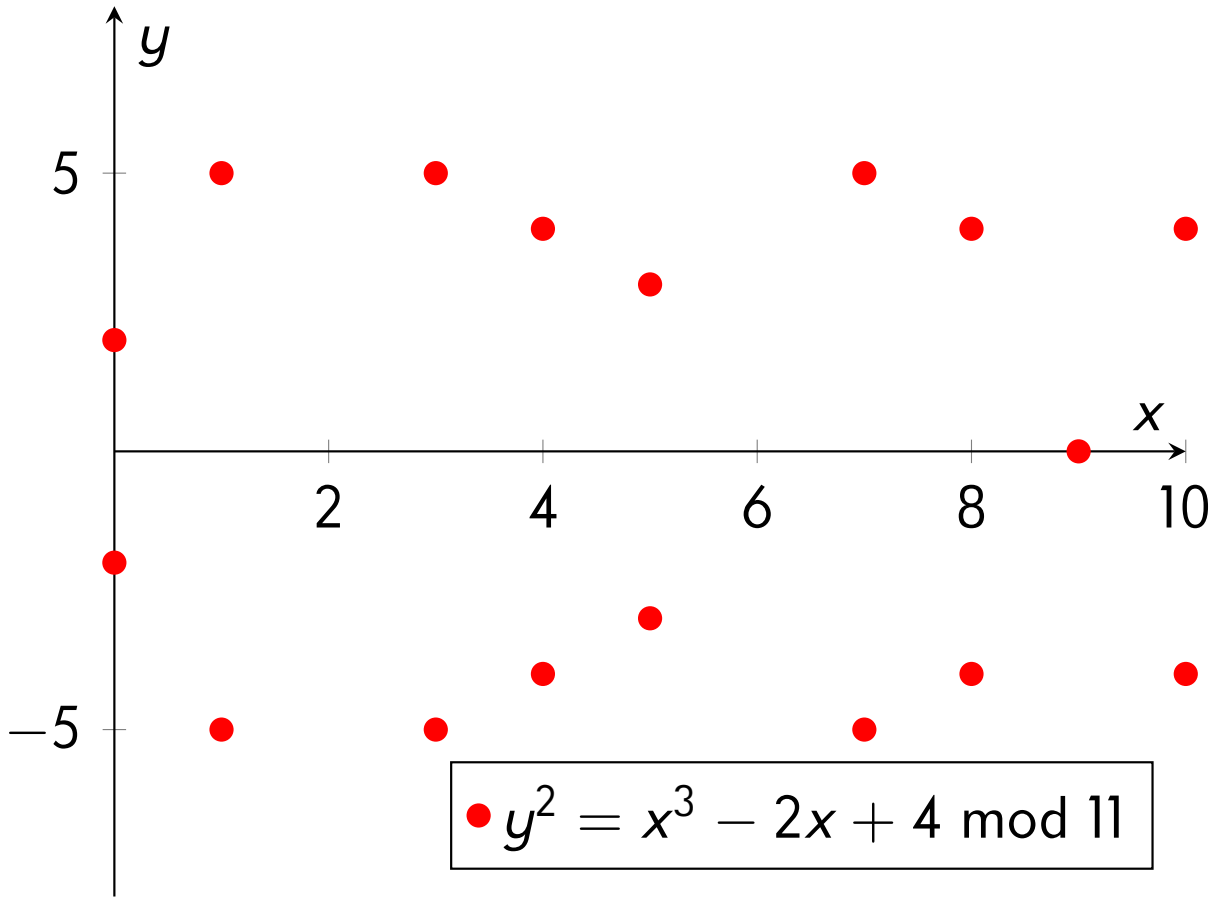
$f(0) = 4$ is square, so $\{(0, 2), (0, 9)\} \subseteq \mathfrak{E}_{11}(9, 4)$

$f(2) = 8$ not square, so $(2, b) \notin \mathfrak{E}_{11}(9, 4) \quad \forall b \in \mathbb{Z}_{11}$

Over the reals $\mathbb{R}$



Over the finite field $\mathbb{Z}_{11}$



Elliptic Curves

All computations are inside $\mathbb{Z}_p$

§11.5 Definition (Elliptic curve group)

Let $p \geq 5$ be prime and $\Pi = \mathfrak{E}_p(a, b)$ be elliptic curve. Define $\cdot : \Pi^2 \to \Pi$

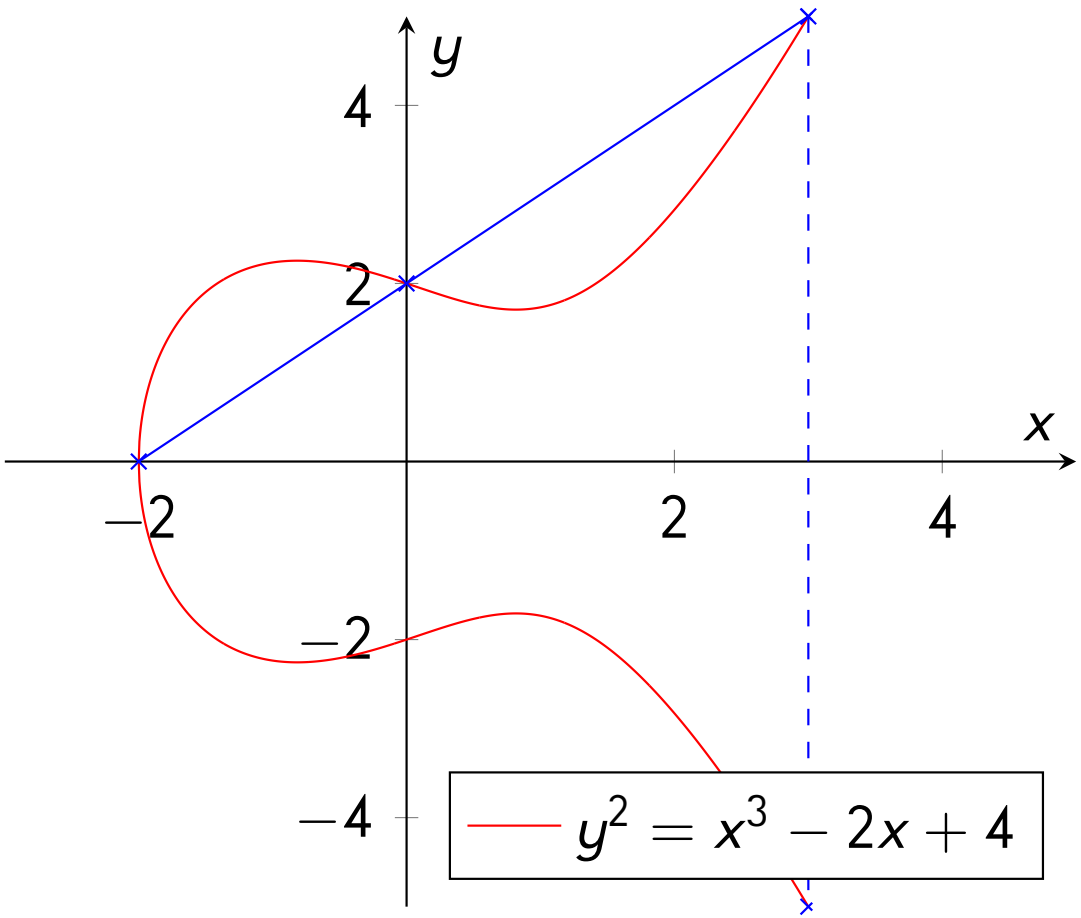
- $\infty \cdot \pi = \pi = \pi \cdot \infty$ for all $\pi \in \Pi$ (∞ neutral)
- $(x, 0) \cdot (x, 0) = \infty$ for all $(x, 0) \in \Pi$
- $(x, y) \cdot (x, y) = (x', y')$ for all $(x, y) \in \Pi$ with $y \neq 0$

$$x' = m^2 - 2x \qquad y' = m(x - x') - y \qquad m = \frac{3x^2 + a}{2y}$$

- $(x, y_1) \cdot (x, y_2) = \infty$ for all $(x, y_1), (x, y_2) \in \Pi$ with $y_1 \neq y_2$
- $(x_1, y_1) \cdot (x_2, y_2) = (x, y)$ for all $(x_1, y_1), (x_2, y_2) \in \Pi$ with $x_1 \neq x_2$

$$x = m^2 - x_1 - x_2 \qquad y = m(x_1 - x) - y_1 \qquad m = \frac{y_2 - y_1}{x_2 - x_1}$$

Over the reals $\mathbb{R}$



Product is x-flip of third intersection of line through points with curve

$(-2, 0) \cdot (0, 2) = (3, -5)$

Elliptic Curves

Derivation for last case: Let $(x_1, y_1), (x_2, y_2) \in \Pi$ with $x_1 \neq x_2$

- Determine slope m of line going through (x_1, y_1) and (x_2, y_2)

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

- Line $y = m(x - x_1) + y_1$ runs through (x_1, y_1) and (x_2, y_2)
- Intersect line with elliptic curve

$$\begin{aligned}
\left(m(x - x_1) + y_1\right)^2 &= x^3 + ax + b \\
m^2(x - x_1)^2 + 2my_1(x - x_1) + y_1^2 &= x^3 + ax + b \\
m^2(x^2 - 2x_1x + x_1^2) + 2my_1x - 2my_1x_1 + y_1^2 &= x^3 + ax + b
\end{aligned}$$

$$x^3 - m^2x^2 + (a + 2m^2x_1 - 2my_1)x + (b - m^2x_1^2 + 2my_1x_1 - y_1^2) = 0$$

Derivation for last case (cont'd): Let $(x_1, y_1), (x_2, y_2) \in \Pi$ with $x_1 \neq x_2$

- Remove known root x_1

$$\frac{x^3 - m^2x^2 + (a + 2m^2x_1 - 2my_1)x + (b - m^2x_1^2 + 2my_1x_1 - y_1^2)}{(x - x_1)}$$
$$= x^2 + (x_1 - m^2)x + (a + 3m^2x_1 - 2my_1 - x_1^2)$$

- Remove known root x_2

$$\frac{x^2 + (x_1 - m^2)x + (a + 3m^2x_1 - 2my_1 - x_1^2)}{(x - x_2)} = x + (x_1 + x_2 - m^2)$$

- Final intersection point $x_3 = m^2 - x_1 - x_2$ with $y_3 = m(x_3 - x_1) + y_1$
- Flip on x-axis, so $x = m^2 - x_1 - x_2$ with $y = m(x_1 - x_3) - y_1$

Elliptic Curves

§11.6 Theorem (Elliptic curve group)

$(\mathfrak{E}_p(a,b), \cdot, \infty)$ is commutative group using the notions of Definition §11.5.

Proof

Manageable exercise, but associativity is very tedious □

Notes

- Efficient algorithms to compute order of $\mathfrak{E}_p(a,b)$ known

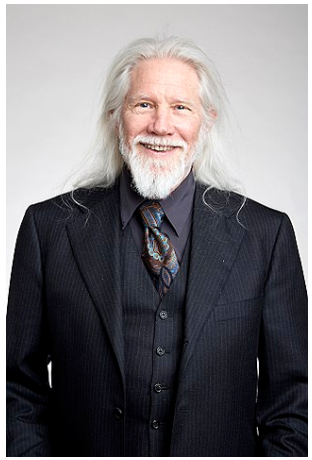
$$p+1-2\sqrt{p} \leq |\mathfrak{E}_p(a,b)| \leq p+1+2\sqrt{p} \qquad \text{(Hasse bound)}$$

- Find parameters a, b with prime $|\mathfrak{E}_p(a,b)| \notin \{p, p+1\}$
- Any $\pi \in \mathfrak{E}_p(a,b) \setminus \{\infty\}$ is generator
- Best to just use standardized curves (published by NIST)

Diffie-Hellman Assumptions

Bailey Whitfield Diffie (* 1944)

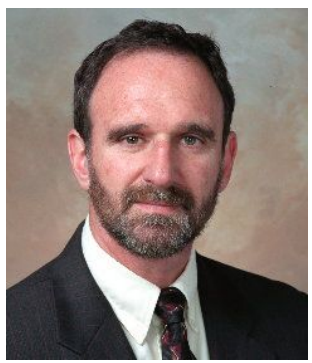
- US cryptographer
- Professor at Royal Holloway University of London & Zhejiang University, China
- Turing award 2015; was hired by Hellman



© The Royal Society

Martin Hellman (* 1945)

- US cryptographer
- Professor at Stanford University
- Met Feistel at IBM (1968–1969)



© Alexander Sigachov

Diffie-Hellman Assumptions

Cyclic group generator $\mathcal{G}$ (public access; everybody has access)

Alice

$(\mathbb{G}, q, g) = \mathcal{G}(1^n)$

uniform $i, i' < q$

$a = g^i$; $b = g^{i'}$

Eve

Diffie-Hellman Assumptions

Cyclic group generator $\mathcal{G}$ (public access; everybody has access)

Alice

$(\mathbb{G}, q, g) = \mathcal{G}(1^n)$

uniform $i, i' < q$

$a = g^i$; $b = g^{i'}$

Challenge $(1^n, a, b)$

Eve

Diffie-Hellman Assumptions

Cyclic group generator $\mathcal{G}$ (public access; everybody has access)

Alice

$(\mathbb{G}, q, g) = \mathcal{G}(1^n)$

uniform $i, i' < q$

$a = g^i$; $b = g^{i'}$

Challenge $(1^n, a, b)$

Candidate c

Eve

Determines c

Diffie-Hellman Assumptions

Cyclic group generator $\mathcal{G}$ (public access; everybody has access)

Alice

$(\mathbb{G}, q, g) = \mathcal{G}(1^n)$

uniform $i, i' < q$

$a = g^i$; $b = g^{i'}$

$c' = b^i$

Return $c \stackrel{?}{=} c'$

Challenge $(1^n, a, b)$

Candidate c

Eve

Determines c

Diffie-Hellman Assumptions

§11.7 Definition (Computational Diffie-Hellman game)

Let $n \in \mathbb{N}$, $\mathcal{G}$ cyclic group generator and $\mathcal{A}$ stateful algorithm.
Computational Diffie-Hellman (CDH) game $\mathrm{CDH}_{\mathcal{G},\mathcal{A}}(n)$ is

1. $(\mathbb{G}, q, g) = \mathcal{G}(1^n)$
2. Select $0 \leq i, i' < q$ uniformly and send $(1^n, g^i, g^{i'})$ to adversary $\mathcal{A}$
3. $\mathcal{A}$ selects & returns element $c \in \mathbb{G}$
4. $c' = (g^i)^{i'}$
5. Return 1 if $c = c'$ ($\mathcal{A}$ wins) and return 0 otherwise

Notes

- $\mathcal{A}$ receives $(1^n, a, b)$ and is expected to compute $a^{\log_g(b)}$ in $\mathbb{G}$
  (it can run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$)
- Verify $c \stackrel{?}{=} (g^i)^{i'} = a^{i'} = a^{\log_g(b)}$

Diffie-Hellman Assumptions

§11.8 Definition (CDH & DDH problem hardness)

Let $\mathcal{G}$ be cyclic group generator.

1. CDH problem is hard for $\mathcal{G}$ if for every PPT algorithm $\mathcal{A}$

$$\mathrm{P}\left[\mathrm{CDH}_{\mathcal{G},\mathcal{A}}(n)\right] \simeq 0$$

2. DDH problem is hard for $\mathcal{G}$ (DDH = Decisional Diffie-Hellman)
if for all PPT algorithms $\mathcal{A}\colon R \times \{1\}^* \times (\{0,1\}^*)^3 \to \{0,1\}$

$$\mathrm{E}\left[\mathrm{P}\left(\mathcal{A}(1^n, g_n^x, g_n^y, g_n^z)\right)\right]_{\substack{x\leftarrow U'_n \\ y\leftarrow U'_n \\ z\leftarrow U'_n}} \simeq \mathrm{E}\left[\mathrm{P}\left(\mathcal{A}(1^n, g_n^x, g_n^y, g_n^{xy})\right)\right]_{\substack{x\leftarrow U'_n \\ y\leftarrow U'_n}}$$

where $\mathcal{G}(1^n) = (\mathbb{G}_n, q_n, g_n)$ and $U'_n\colon \{0,\ldots,q_n-1\} \to [0,1]$ given by $U'_n(i) = q_n^{-1}$ for all $i \in \{0,\ldots,q_n-1\}$
(U'_n = uniform distribution over $\{0,\ldots,q_n-1\}$)

Diffie-Hellman Assumptions

Summary

- Hardness of CDH problem requires hard discrete logarithms
  (otherwise involved discrete logarithms can simply be computed)
- Hardness of DDH problem requires hardness of CDH problem
  (simple exercise)

- CDH problem might be simpler than discrete logarithm
- DDH problem expected simpler than CDH problem & discrete logarithm
  (groups with simple DDH, but expected hard CDH exist)
- Discrete logarithm easy in some finite cyclic groups
  (find further examples)

Summary

- Cyclic groups & Diffie-Hellman assumption
- Cyclic groups, generators, & discrete logarithm
- Elliptic curves
- CDH & DDH assumptions