PUBLIC-KEY CRYPTOGRAPHY
(Introduction)

Ratna Dutta

Department of Mathematics

Indian Institute of Technology, Kharagpur

Autumn, 2019

Outline

Some basic public key cryptosystems

- Encryption - Rabin, RSA, Merkle-Hellman, Paillier, Goldwasher-Micali, ElGamal, Generalised ElGamal
- Signature - RSA, ElGamal, DSA
- Key agreement - Diffie-Hellman, Burmester-Desmedt

Public Key Encryption

- *Setup*: Generates system parameters and for each entity a pair of encryption and decryption key.
  - The system parameters and the encryption key are public.
  - The decryption key corresponding to the encryption key is kept private to the corresponding entity.
- *Encrypt*: Encrypts message using the public encryption key.
- *Decrypt*: Decrypts the message using the private decryption key of the corresponding public encryption key.

Rabin's Cryptosystem

- *Setup*: $n = pq$, p, q large primes, both $p, q \equiv 3 \pmod 4$,

$$\mathsf{PK} = n, \mathsf{SK} = (p, q)$$

- *Encrypt*: Message $x \in Z_n^*$, public key PK

$$y = x^2 \mod n$$

- *Decrypt*: Ciphertext y, secret key SK

$$x = \sqrt{y} \mod n$$

(requires knowledge of p, q to extract square roots modulo $n = pq$ using CRT)

Chinese Remainder Theorem (CRT)

- $m_1, \ldots, m_r$ pairwise relatively prime, $a_1, \ldots, a_r \in Z$
- System of congruences:

$$
\begin{aligned}
x &= a_1 \mod m_1 \\
x &= a_2 \mod m_2 \\
&\vdots \\
x &= a_r \mod m_r
\end{aligned}
$$

- Unique solution modulo $M = m_1 m_2 \ldots m_r$

$$x = \sum_{i=1}^{r} a_i M_i y_i \mod M$$

where $M_i = \frac{M}{m_i}$, $y_i = M_i^{-1} \mod m_i$ for $1 \le i \le r$

- **Quadratic residue modulo p:** Let p be an odd prime and $x \in Z_p^*$. x is defined to be a *quadratic residue* or *square* modulo p if the congruence
$$y^2 \equiv x \pmod{p}$$
has a solution $y \in Z_p$.
- *Example:* $\mathsf{QR}_{11} = \{1, 3, 4, 5, 9\}$.
- **Euler's Criterion:** x is a quadratic residue modulo p if and only if $x^{\frac{p-1}{2}} \equiv 1 \pmod{p}$
- Suppose z is a quadratic residue and $p \equiv 3 \pmod{4}$. Then, the two square roots of z modulo p are $\pm z^{\frac{p+1}{4}} \mod p$.

Correctness

- both $p, q \equiv 3 \pmod 4$ - this restriction simplifies computation of square roots modulo n

$$\left(\pm y^{\frac{p+1}{4}}\right)^2 = y^{\frac{p+1}{2}} = y\ y^{\frac{p-1}{2}} \equiv y\ \left(\frac{y}{p}\right) \equiv y \pmod p$$

(Legendre symbol $\left(\frac{y}{p}\right) \equiv y^{\frac{p-1}{2}} \equiv 1 \pmod p$ by Euler's criterion if $y \in \mathsf{QR}_p$)

- $\pm y^{\frac{p+1}{4}}$ are the square roots of $y \mod p$
- $\pm y^{\frac{q+1}{4}}$ are the square roots of $y \mod q$
- find 4 square roots modulo $n = pq$ using CRT

- **Security:** Hardness of integer factorzation
- **Disadvantage:** Decryption ambiguous - 4 possible plaintexts as 4 square roots modulo a valid ciphertext y modulo n
- if $p \equiv 1 \pmod 4$, no polynomial time *deterministic* algorithm to compute the square roots of QR_p
- restriction on p, q can be omitted at the expense of computation cost

RSA Cryptosystem

- *Setup*: $n = pq$, p, q are large primes
  - choose e, with $\mathsf{gcd}(\phi(n), e) = 1; 1 < e < \phi(n)$
  - set $d \equiv e^{-1} \pmod{\phi(n)}$
  - $\mathsf{PK} = (n, e)$, $\mathsf{SK} = (d)$
- *Encrypt*: Message $M \in Z_n^*$, PK

$$C \equiv M^e \pmod{n}$$

- *Decrypt*: Ciphertext C, SK

$$M \equiv C^d \pmod{n}$$

- **Correctness:**
  - $ed = k\phi(n) + 1$, $k \in Z$ as $d \equiv e^{-1} \pmod{\phi(n)}$
  - (Euler's theorem) $M^{\phi(n)} \equiv 1 \pmod{n}$ as $M \in Z_n^*$
  - $C^d \equiv (M^e)^d \equiv (M)^{k\phi(n)+1} \equiv M\ (M^{\phi(n)})^k$
    $\equiv M \pmod{n}$
- **Computational Aspects:** primality test, **gcd** computation, modular inverse computation, fast exponentiation
- **Security:** Hardness of facorization problem.
  - if one knows n (always public) and $\phi(n)$, then the factorization $n = pq$ can be readily obtained

Merkle-Hellman Knapsack Cryptosystem

- Based on the **Subset Sum (Knapsack) problem**
  - **Instance:** $I = (\vec{a} = (a_1, a_2, \ldots, a_n), S)$, where $a_1, \ldots, a_n$ and S are positive integers. The a_i's are called sizes and S is called the target sum.
  - **Question:** Is there a 0-1 vector $\vec{x} = (x_1, x_2, \ldots, x_n)$ such that
  $$\vec{a} \cdot \vec{x} = \sum_{i=1}^{n} a_i x_i = S?$$
- Subset sum problem is NP-complete problem, *i.e.*, there is no polynomial-time algorithm that solves it.

- However, certain special cases can be solved in polynomial time.
- **Superincreasing sequence:** A list of sizes $(a_1, \ldots, a_n)$ is superincreasing if

$$a_j > \sum_{i=1}^{j-1} a_i$$

for $2 \leq j \leq n$.
- If the list of sizes in superincreasing, then Subset sum search problem can be solved in time $O(n)$, and a solution $\vec{x}$ (if it exists) must be unique.

- **Algorithm for the subset sum problem**

  1. **for** $i = n$ downto 1 **do**
  2.   **if** $S \geq a_i$ **then**
  3. $S = S - a_i$
  4. $x_i = 1$
  5.   **else**
  6. $x_i = 0$
  7. **end do**
  8. **if** $S = 0$ **then** $X = (x_1, \ldots, x_n)$ is the solution

  10. **else** there is no solution

- *Setup*: $\mathsf{PK} = (\vec{a}, m)$, $\mathsf{SK} = (w)$ generated as follows:
  - choose $\vec{a}' = (a'_1, \ldots, a'_n)$, a superincreasing list of integers
  - choose m and w, two positive integers such that

$$m > \sum_{i=1}^{n} a'_i \text{ and } \mathsf{gcd}(w, m) = 1$$

  - set $\vec{a} = (a_1, \ldots, a_n) = w\vec{a}'$, *i.e.* $a_i = wa'_i \mod m$
- *Encrypt*: Message $\vec{x} = \{x_1, \ldots, x_n\} \in \{0, 1\}^n$, $\mathsf{PK} = (\vec{a}, m)$

$$C = \vec{a} \cdot \vec{x} = \sum_{i=1}^{n} x_i a_i \mod m$$

- *Decrypt*: Ciphertext $C = \vec{a} \cdot \vec{x} \in Z_m$, $\mathsf{SK} = (w)$
  - compute $\vec{a}' = w^{-1}\vec{a}$
  - set $S = w^{-1}C \mod m$
  - solve the subset sum problem $(\vec{a}' = (a'_1, \ldots, a'_n), S)$ and recover the message $\vec{x} = (x_1, \ldots, x_n)$
- **Correctness:** As $\vec{a}'$ is superincreasing and

$$S = w^{-1}C = w^{-1}(\vec{a} \cdot \vec{x}) = (w^{-1}\vec{a}) \cdot \vec{x} = \vec{a}' \cdot \vec{x} \mod m,$$

the subset sum problem $(\vec{a}' = (a'_1, \ldots, a'_n), S)$ can be solved and the message $\vec{x} = (x_1, \ldots, x_n)$ can be recovered

Probabilistic Encryption

- many possible encryption of each plaintext
- not feasible to test whether a given ciphertext is an encryption of a particular plaintext
- no information about the plaintext should be computable from the ciphertext (in polynomial time)

Goldwasser-Micali Cryptosystem

- *Setup*: $n = pq$, p, q are large primes
  - pseudosquare $m \in_R \widetilde{\mathsf{QR}}_n$, *i.e.* $\left(\frac{m}{p}\right) = \left(\frac{m}{q}\right) = -1$
  - $\mathsf{PK} = (m, n)$, $\mathsf{SK} = (p, q)$
- *Encrypt*: Message $x \in \{0, 1\}$, PK, $r \in_R Z_n^*$

$$y = m^x r^2 \mod n$$

- *Decrypt*: Ciphertext $y \in \mathsf{QR}_n \cup \widetilde{\mathsf{QR}}_n$, SK

$$x = \begin{cases} 0 & \text{if } y \in \mathsf{QR}_n \\ 1 & \text{if } y \notin \mathsf{QR}_n \end{cases}$$

(requires p, q to decide whether $y \in \mathsf{QR}_n$ or not)

Correctness

- Jacobi symbol $\left(\frac{y}{n}\right) = 1$ as $y \in \mathsf{QR}_n \cup \widetilde{\mathsf{QR}}_n$. Then

$$y \in \mathsf{QR}_n \Longleftrightarrow \left(\frac{y}{p}\right) = 1$$

  $(i.e.$ No need to check $\left(\frac{y}{q}\right) = 1)$

- Compute $\left(\frac{y}{p}\right) \equiv y^{\frac{p-1}{2}} \pmod{p}$ by Euler's criterion, check whether $\left(\frac{y}{p}\right) = 1$. If so, conclude $y \in \mathsf{QR}_n$ and recover the message as $x = 0$

- Thus explicit knowledge of p (or q) is required for correct decryption

- **Security:** Simantically secure assuming Decisional Quadratic Residuosity (DQR) problem is hard.
- **DQR problem:** Given $n = pq$ and $\left(\frac{y}{n}\right) = 1$, where p, q large *unknown* primes, decide whether $y \in \mathsf{QR}_n$ or not.
- **Homomorphic property:** If y_0, y_1 are encryptions of x_0, x_1 respectively, then $y_0 y_1$ is the encryption of $x_0 \oplus x_1$.

ElGamal Cryptosystem in Z_p^*

- *Setup*: $Z_p^* = \langle \alpha \rangle$, p is a large primes so that DLP in Z_p is intractable
  - set $\beta \equiv \alpha^a \pmod{p}$, where $0 \leq a \leq p-2$
  - $\mathsf{PK} = (p, \alpha, \beta)$, $\mathsf{SK} = (a)$
- *Encrypt*: Message $x \in Z_p^*$, PK, $k \in_R Z_{p-1}^*$

$$C = (y_1, y_2), y_1 = \alpha^k \bmod p, y_2 = x\ \beta^k \bmod p$$

- *Decrypt*: Ciphertext $C = (y_1, y_2) \in Z_p^* \times Z_p^*$, SK

$$x = y_2 (y_1^a)^{-1} \bmod p$$

- **Correctness:**
  $$y_2(y_1^a)^{-1} \equiv (x\beta^k)\{(\alpha^k)^a\}^{-1} \equiv x\{(\alpha)^a\}^k(\alpha^{ka})^{-1}$$
  $$\equiv x \pmod p$$

- **Security:** Simantically secure assuming Discrete Logarithm Problem (DLP) is hard in Z_p.

- **DLP problem:** Given $I = (p, \alpha, \beta)$, where p is prime, $\alpha \in Z_p$ is a primitive element, and $\beta \in Z_p^*$, find the unique integer $a, 0 \leq a \leq p-2$, such that
  $$\alpha^a \equiv \beta \pmod p$$
  (This integer a is denoted by $\log_\alpha \beta$)

Generalized ElGamal Encryption

- *Setup*: Let G be a finite group with group operation $\circ$
  - choose $H = \langle \alpha \rangle$, a subgroup of G, DLP is hard in H
  - choose $a \in_R Z_{|H|}$ and set $\beta = \alpha^a = \alpha \circ \alpha \circ \ldots \circ \alpha$
  - $\mathsf{PK} = (p, \alpha, \beta)$, $\mathsf{SK} = (a)$
- *Encrypt*: Message $x \in G$, $\mathsf{PK} = (p, \alpha, \beta)$, $k \in_R Z_{|H|}$

$$C = (y_1, y_2), y_1 = \alpha^k, y_2 = x \circ \beta^k$$

- *Decrypt*: Ciphertext $C = (y_1, y_2) \in G \times G$, $\mathsf{SK} = (a)$

$$x = y_2 \circ (y_1^a)^{-1}$$

- **Correctness:**
  $y_2 \circ (y_1^a)^{-1} = (x \circ \beta^k) \circ \{(\alpha^k)^a\}^{-1} = x \circ \{(\alpha)^a\}^k \circ (\alpha^{ka})^{-1} = x$
- **Security:** Simantically secure assuming Discrete Logarithm Problem (DLP) is hard in $(G, \circ)$.
- **DLP problem in $(G, \circ)$:** Given $\langle G, \alpha \in G, \beta \in H \rangle$, where $H = \langle \alpha \rangle$ is a subgroup of G, find the unique integer $a, 0 \leq a \leq |H| - 1$, such that $\alpha^a = \beta$.
- **Homomorphic property:** If C_0, C_1 are encryptions of x_0, x_1 respectively, then $C_0 \circ C_1$ is the encryption of $x_0 \circ x_1$.

Paillier Cryptosystem

- *Setup*: $n = pq$, p, q are large primes and $\gcd(n, \phi(n)) = 1$
  - $\lambda = \mathsf{lcm}(p-1, q-1)$
  - choose $g \in Z^*_{n^2}$ such that $n | \mathsf{ord}_{n^2}(g)$ to ensure

$$\mathsf{gcd}(L(g^\lambda \mod n^2), n) = 1, L(u) = \frac{u-1}{n}$$

  - $\mathsf{PK} = (n, g)$, $\mathsf{SK} = (\lambda)$
- *Encrypt*: Message $m \in Z^*_n$, PK, $r \in Z^*_n$

$$c = g^m \cdot r^n \bmod n^2$$

- *Decrypt*: Ciphertext $c \in Z^*_{n^2}$, $\mathsf{SK} = (\lambda)$

$$m = \frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} \mod n$$

- **Correctness:** (choosing $g = (n+1)^t \in Z^*_{n^2}$)

$$L(c^\lambda \mod n^2) = \frac{c^\lambda - 1}{n} = \frac{(1 + tnm\lambda) - 1}{n} = tm\lambda$$

$$L(g^\lambda \mod n^2) = \frac{(n+1)^t\lambda - 1}{n} = \frac{1 + nt\lambda - 1}{n} = t\lambda$$

$$\frac{L(c^\lambda \mod n^2)}{L(g^\lambda \mod n^2)} = \frac{mt\lambda}{t\lambda} = m \mod n$$

- **Security:** Simantically secure assuming Decisional Composite Residuosity (DCR) problem is hard.
- **DCR assumption:** Given $n = pq$ and an integer z, where p, q large *unknown* primes, decide whether z is an n-residue modulo n^2 or not,
  i.e., whether there exists y such that
  $$z \equiv y^n \mod n^2$$
- Given the Paillier encryptions of two messages, there is no known way to compute an encryption of the product of these messages without knowing the private key.

Homomorphic property

- Homomorphic addition of plaintexts:

$$\mathcal{D}(\mathcal{E}(m_1, r_1) \cdot \mathcal{E}(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n$$

$$\mathcal{D}(\mathcal{E}(m_1, r_1) \cdot g^{m_2} \bmod n^2) = m_1 + m_2 \bmod n$$

- Homomorphic multiplication of plaintexts

$$\mathcal{D}(\mathcal{E}(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n$$

$$\mathcal{D}(\mathcal{E}(m_2, r_2)^{m_1} \bmod n^2) = m_1 m_2 \bmod n$$

- More generally,

$$\mathcal{D}(\mathcal{E}(m_1, r_1)^k \bmod n^2) = k m_1 \bmod n$$

- Useful property to design *advanced* cryptographic primitives for outsourcing of private computations, for instance, in the context of cloud computing
- **Partially homomorphic cryptosystems** - RSA, ElGamal, GM, Paillier
- **Fully homomorphic encryption** - supports arbitrary computation on ciphertexts (lattice-based cryptography)

Other Important PKC

- **McEliece:** This is based on algebraic coding theory (uses Goppa codes) and the security is based on the problem of decoding a linear code (which is NP-complete).
- **Elliptic Curve:** Elliptic Curve Crytosystems (ECC) work in the domain of elliptic curves rather than finite fields. The ECC appears to remain secure for smaller keys than other public-key crytosystems.

Digital Signature Schemes

- A standard digital signature scheme $\mathsf{DSig} = (Setup, Sign, Verify)$ consists of three algorithms.

  1. $Setup$: generates randomly public system parameters params and public/secret key pair PK, SK of a signer.
  2. $Sign$: generates a signature on a given message m using the secret key SK of a signer.
  3. $Verify$: checks the validity of a signature on a given message using the public key of a signer.

RSA signature

- *Setup*: $n = pq$, p, q are large primes
  - choose a, $1 < a < \phi(n)$ with $\mathsf{gcd}(\phi(n), a) = 1$
  - set $b \equiv a^{-1} \pmod{\phi(n)}$
  - $\mathsf{PK} = (n, b)$, $\mathsf{SK} = (a)$
- *Sign*: Message $m \in Z_n^*$, signing key $\mathsf{SK} = (a)$

$$\sigma \equiv m^a \pmod{n}$$

- *Verify*: Message m, signature σ, verification key $\mathsf{PK} = (n, b)$, verify

$$m \equiv \sigma^b \pmod{n}$$

ElGamal signature

- *Setup*: $Z_p^* = \langle \alpha \rangle$, p is a large primes so that DLP in Z_p is intractable
  - set $\beta \equiv \alpha^a \pmod{p}$, where $0 \leq a \leq p-2$
  - $\mathsf{PK} = (p, \alpha, \beta)$, $\mathsf{SK} = (a)$
- *Sign*: Message $x \in Z_p^*$, signing key $\mathsf{SK} = (a)$, $k \in_R Z_{p-1}^*$, signature $\sigma = (\gamma, \delta) \in Z_p^* \times Z_{p-1}$ where

$$\gamma = \alpha^k \bmod p$$

$$\delta = (x - a\gamma)k^{-1} \mod (p-1)$$

- *Verify*: Message m, signature $\sigma = (\gamma, \delta)$, verification key $\mathsf{PK} = (p, \alpha, \beta)$, verify

$$\beta^{\gamma}\gamma^{\delta} \equiv \alpha^{x} \pmod{p}$$

- **Correctness:**

$$\beta^{\gamma}\gamma^{\delta} \equiv (\alpha^{a})^{\gamma}\ (\alpha^{k})^{\delta} \equiv \alpha^{a\gamma + k(x - a\gamma)k^{-1}} \equiv \alpha^{x} \pmod{p}$$

Digital Signature Algorithm (DSA)

- *Setup*: $\mathsf{PK} = (p, q, \alpha, \beta)$, $\mathsf{SK} = (a)$ where
  - p , a 512-bit prime such that DLP in Z_p is intractable
  - $q|(p-1)$, a 160-bit prime
  - $\alpha \in Z_p^*$, a q-th root of unity modulo p ($i.e.$ α generates a subgroup of Z_p^* of order q)
  - set $\beta \equiv \alpha^a \pmod{p}$, where $0 \le a \le q$

- *Sign*: Message $x \in Z_p^*$, signing key $\mathsf{SK} = (a)$,
  - choose k, $1 \leq k \leq q-1$ and set signature $\sigma = (\gamma, \delta) \in Z_q \times Z_q$ where

$$\gamma = (\alpha^k \bmod p) \mod q$$

$$\delta = (x + a\gamma)k^{-1} \mod q$$

- *Verify*: Message m, signature $\sigma = (\gamma, \delta)$, verification key $\mathsf{PK} = (p, q, \alpha, \beta)$, verify

$$(\alpha^{e_1}\beta^{e_2} \mod p) \mod q = \gamma$$

where

$$e_1 = x\delta^{-1} \mod q$$
$$e_2 = \gamma\delta^{-1} \mod q$$

- **Correctness:**

$$\alpha^{e_1}\beta^{e_2} = \alpha^{x\delta^{-1}}\left(\alpha^{a}\right)^{\gamma\delta^{-1}} = \alpha^{(x+a\gamma)\delta^{-1}}$$

$$= \alpha^{k} = \gamma \mod p \mod q$$

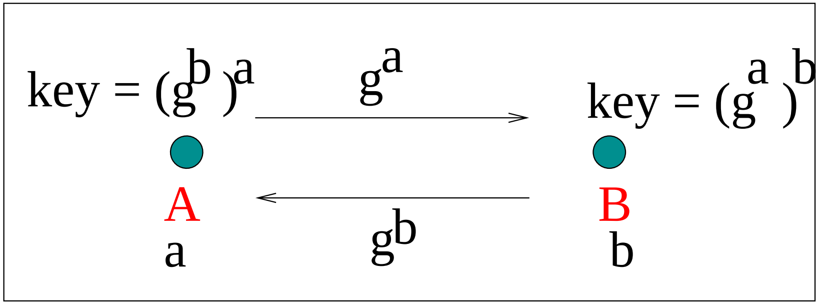
as

$$\gamma = (\alpha^k \bmod p) \mod q$$

$$\delta = (x + a\gamma)k^{-1} \mod q$$

Two-Party Key Agreement

(Diffie-Hellman, IEEE-IT, 1976)

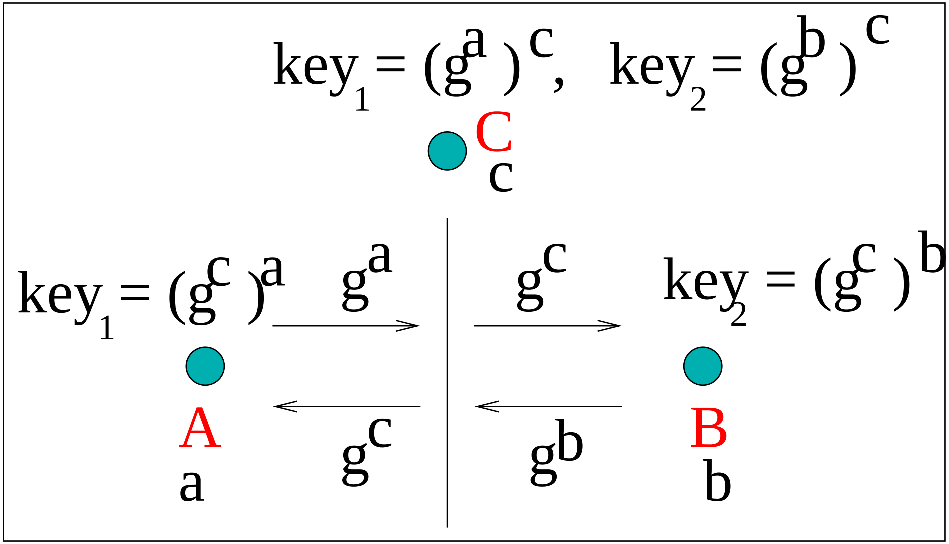


- $G = \langle g \rangle$, order of G is q, a large prime

- security: hardness of DDH problem.
- DDH (Decision Diffie-Hellman) Problem in $G = \langle g \rangle$:

  given $\langle g, g^a, g^b, g^c \rangle$ for some $a, b, c \in Z_q^*$, decide whether $c = ab \bmod q$.
- unauthenticated

Man-in-the-middle Attack



- authentication - digital signature

Burmester-Desmedt Group Key Agreement (variant)

Consider participants $U_1, \ldots, U_n$ are on a *virtual* ring, $G = \langle g \rangle$ be a multiplicative group of some large prime order q and $\mathcal{H} : \{0,1\}^* \to Z_q^*$ be a hash function

- Round 1: U_i picks $x_i \in_R Z_q^*$ and broadcasts $X_i = g^{x_i}$
- Round 2: U_i on receiving X_{i-1} and X_{i+1} computes
  - $K_i^L = X_{i-1}^{x_i}$, its left key
  - $K_i^R = X_{i+1}^{x_i}$, its right key

and broadcasts $Y_i = \frac{K_i^R}{K_i^L}$

U_1	U_2	$\cdots$	U_{n-1}	U_n	
x_1	x_2	$\cdots$	x_{n-1}	x_n	
g^{x_1}	g^{x_2}	$\cdots$	$g^{x_{n-1}}$	g^{x_n}	: round 1
$\frac{K_1^R}{K_1^L}$	$\frac{K_2^R}{K_2^L}$	$\cdots$	$\frac{K_{n-1}^R}{K_{n-1}^L}$	$\frac{K_n^R}{K_n^L}$	: round 2

- Key Computation:
  - U_i computes

$$K_{i+1}^R = Y_{i+1}K_i^R (= \frac{K_{i+1}^R}{K_{i+1}^L} K_i^R)$$

$$K_{i+2}^R = Y_{i+2}K_{i+1}^R (= \frac{K_{i+2}^R}{K_{i+2}^L} K_{i+1}^R)$$

$$\vdots$$

$$K_{i+(n-1)}^R = Y_{i+(n-1)}K_{i+(n-2)}^R (= \frac{K_{i+(n-1)}^R}{K_{i+(n-1)}^L} K_{i+(n-2)}^R)$$

– U_i then verifies if $K^R_{i+(n-1)} = K^L_i (= K^R_{i+(n-1)})$

– if verification fails, U_i aborts

– else U_i has correct right keys of all the users

– U_i computes the session key

$$\mathsf{sk} = K^R_1 K^R_2 \ldots K^R_n = g^{x_1x_2+x_2x_3+\cdots+x_nx_1}$$

– security: hardness of DDH problem.

– unauthenticated