Public Key Cryptography

Rana Barua

IAI, TCG CREST
Kolkata

June 24, 2025

Public Key Cryptography

- Before 1977, all cryptosystems used or proposed were symmetric key cryptosystems in which both the sender and receiver used the **same** key for both encryption and decryption.
- In 1976, in their seminal paper *New directions in cryptography*, IEEE Transactions on Information Theory 22 (1976), 644-654; Diffie-Hellman asked whether it is possible to have a cryptosystem where each user would have two keys, a private key and a public key that would be available to all
- In 1977, Rivest, Shamir and Adelman proposed the first feasible Public Key Cryotosystem, now known as **RSA**, using elementary number theory.

Public Key Cryptosystems:Textbook RSA

Key-Generation: Let $N = pq$ be the product of two large primes.
Choose e, d s.t. $ed \equiv 1 \mod \phi(N)$
Public key: (N, e) Secret Key (N, p, q, d)
Encryption: To encrypt a message $M \in \mathbb{Z}_N^*$, compute

$$y = M^e \mod N.$$

Decryption: Given ciphertext $y \in \mathbb{Z}_N^*$, compute

$$M = y^d \mod N.$$

Public Key Cryptosystems :RSA

Correctness: Suppose $y \equiv M^e \mod N$. Since $ed \equiv 1 \mod \phi(N)$ we have $ed = t\phi(N) + 1$. Assume $M \in \mathbb{Z}_N^*$. Then

$$y^d \equiv M^{ed} \equiv (M^{\phi(N)})^t.M \equiv 1.M \mod N.$$

Remark: If factorization of N is known or if $\phi(N)$ is known then RSA is completely broken

Algorithms for Factoring

Several algorithms for factoring exists viz

- Pollard's $p - 1$ algorithm
- Pollard's ρ algorithm
- Number field sieve
- Quadratic sieve
- and many others.

None of these are poly-time algorithms (assuming quantum computers do not exist).

Pollard's $p-1$ Algorithm

INPUT: An integer N
OUTPUT: A non-trivial factor of N.

1. Choose B such that $p-1|B$ but $q-1$ does not divide B
2. $x \stackrel{R}{\leftarrow} \mathbb{Z}_N^*$
3. $y := (x^B - 1) \bmod N$
4. $p := GCD(y, N)$
5. **if** $p \notin \{1, N\}$ **then**
6. **return** p

Probability of Success

We know that $\mathbb{Z}_N^* \leftrightarrow \mathbb{Z}_P^* \times \mathbb{Z}_q^*$. Hence

$$x^B - 1 \bmod N \leftrightarrow (x^B - 1 \bmod p, x^B - 1 \bmod q) = (0, x^B - 1 \bmod q).$$

Note that $x^B \bmod q \neq 1$ if $x \bmod q$ is a generator of $\mathbb{Z}_q^*$. Now, $\mathbb{Z}_q^*$ has exactly $\phi(q-1)$ generators and $x \bmod q$ is a random element of $\mathbb{Z}_q^*$ Hence the probability that $x \bmod q$ is a generator is $\frac{\phi(q-1)}{q-1} = \Omega(1/n)$

Public Key Cryptosystem: Rabin Cryptosystem

Setup: Let $n = pq$ where p, q are primes and $p, q \equiv 3 \mod 4$
PK is n and the secret key is (n, p, q)
Encrypt Given a message $x \in Z_n^*$ compute

$$c = x^2 \mod n.$$

Decrypt: Given ciphertext c find the square-roots of c modulo n.

ElGamal Public Key Cryptosystem, 1984

- **Key Generation:**
  1. Choose a cyclic $G =< g >$ of prime order p
  2. choose $x_A \in_R Z_p$ and compute $y_A = g^{x_A}$
  3. Public key is (g, y_A) and secret key is x_A.
- **Encryption:** Given message $m \in G$,
  1. choose $r \in_R Z_p$ and compute $h = g^r$
  2. send ciphertext $(h, y_A^r.m)$
- **Decryption:** On receiving ciphertext (h, z), compute

$$m = (h^{x_A})^{-1}.z$$

- **Correctness**: $h^{x_A} = (g^r)^{x_A} = y_A^r$

Security of ElGamal

- Discrete Logarithm Problem.
- Diffie-Hellman Problem.

Discrete Logarithm:

- **Instance**: A multiplicative group $(G, .)$, an element $\alpha \in G$ of order n, and an element $\beta \in <\alpha>$, the cyclic group generated by α.
- **Question**: Find the unique integer $a, 0 \leq a \leq n-1$, s.t. $\alpha^a = \beta$.
  The integer a is called the discrete log of β to base α and is denoted by $\log_\alpha \beta$.
- Computing the discrete log is probably difficult in suitable groups.

- **Instance**: A multiplicative group $(G, .)$, an element $\alpha \in G$ of order n, and elements $\alpha^a, \alpha^b \in <\alpha>$, the cyclic group generated by α.
- **Question**: Compute α^{ab}.
- Diffie-Hellman Problem is stronger than the DLP

Decisional Diffie-Hellman(DDH) Problem

- **Instance**: A multiplicative group $(G, .)$, an element $\alpha \in G$ of order n, and a triplet $(\alpha^a, \alpha^b, h) \in <\alpha>^3$ from the cyclic group generated by α.
- Decide whether $h = \alpha^{ab}$ or h is random.
- Decisional Diffie-Hellman Problem is stronger than the CDH

Security Against Chosen Ciphertext Attack (IND-CCA)

- **INIT:** Challenger runs the key generation algorithm and gives the public key to adversary $\mathcal{A}$
- **Phase 1**: Adversary $\mathcal{A}$ makes(adaptively) a finite number of queries to the decryption-oracle $\mathcal{O}_d$. It returns the resulting plaintext or **null** if the ciphertext cannot be decrypted.
- Challenge: When $\mathcal{A}$ decides that Phase 1 is over, it chooses two equal length messagesm_0, m_1 and [pass these to $\mathcal{C}$ The challenger chooses uniformly at random a bit $b \in \{0, 1\}$ and obtains a ciphertext C^* corresponding to m_b , It returns C^* as the challenge ciphertext to $\mathcal{A}$.

- Phase 2: $\mathcal{A}$ now issues additional queries just like Phase 1, with the (obvious) restriction . The challenger responds as in Phase 1.
- Guess: $\mathcal{A}$ outputs a guess $\bar{b}$ of b . The **advantage** of the adversary $\mathcal{A}$ in attacking the PKE scheme H is defined as:

$$Adv_{\mathcal{A}} = |Pr[(b = \bar{b})] - 1/2|.$$

A PKE scheme is said to be IND-CCA secure if for any (poly-time) adversary $\mathcal{A}$ that makes at most polynomial decryption queries, $Adv_{\mathcal{A}}$ is negligible.

CPA-Security

Security of El Gamal Encryption

Theorem

If the DDH problem is hard relative to G, then the El Gamal encryption scheme is CPA-secure.

Questions

What groups G should be chosen for ElGamal Cryptosystems?

- Obvious choice is Z_p^*, for large primes p
  p should be carefully chosen to avoid known algorithms for DLP.
  e.g. $p - 1$ should contain at least one large prime factor.
- Elliptic Curves
- Hyperelliptic curves
- Others

Diffie-Hellman Key Exchange

The Diffie-Hellman problem gives rise to a key exchange protocol. Let $G =< g >$ be a cyclic group of large prime order p for which the CDH problem is hard.

- Alice chooses a random $a \in \mathbb{Z}_p$ and sends it to Bob.
- Bob chooses a random $b \in \mathbb{Z}_p$ and sends it to Alice. (These two acts can be done simultaneously)
- Alice computes $(g^b)^a$ while Bob computes $(g^a)^b$ Thus both Alice and Bob compute a common key g^{ab}.

Algorithms for Discrete Log

Several algorithms for algorithms for Discrete Log exists viz

- Pollard's ρ algorithm
- Shanks' algorithm
- Index Calculus Algorithm
- Pohlig-Hellman Algorithm
- and many others.

None of these are poly-time algorithms (assuming quantum computers do not exist).

Shanks' Algorithm

INPUT: An element $h \in G =< g >$, the cyclic group of order q generated by g.
OUTPUT: $\log_g h$

1. $t := \lfloor \sqrt{q} \rfloor$
2. **for** $i = 0$ **to** $\lfloor q/t \rfloor$
3. **compute** $g_i := g^{it}$
4. **sort** (i, g_i) by their second component
5. **for** $i = 1$ **to** t
6. **compute** $h_i := h.g^i$
7. **if** $h_i = g_k$ for some k,
8. **return** $(kt - i) \bmod q$

Time/Space Complexity

Correctness $h_i = g_k$ implies $hg^i = g^{kt}$ i.e. $g^{x+i} = g^{kt}$.
Hence $x = (kt - i) \mod q$

- **Time complexity**: $O(\sqrt{q} polylog(q))$
- **Space complexity:** $O(\sqrt{q})$

Public Key Signature Scheme

A signature scheme is given by following algorithms:

- **Setup(1^k):** A PPT algorithm which takes a security parameter as input and outputs public parameters *Params*.
- **KG(*Params*):** A PPT algorithm which takes *Params* as input and outputs a public-private key pair (PK, SK).
- **SIG($m, SK, Params$):** A PPT algorithm which takes a message m, a secret key SK and *Params* as input and outputs a signature σ.
- **VER($m, \sigma, PK, Params$):** A deterministic polynomial time algorithm which takes a message m, a signature σ, a public key PK and *Params* as input and outputs **1** if σ is a valid signature on message m, else it returns **0**.

Security Notion of Signature Scheme

A signature scheme is said to be **EUF-CMA (existentially unforgeable against chosen message attack)** secure if no probabilistic polynomial time algorithm has a non-negligible advantage in the following game.

Game$^{EUF-CMA}_{SIG,\mathcal{A}}(1^k)$

- $L \leftarrow \phi$
- $Params \leftarrow \text{Setup}(1^k)$
- $(PK, SK) \leftarrow \text{KG}(Params)$
- $(m, \sigma) \leftarrow \mathcal{A}^{\mathcal{O}}(SK, Params)$
- $x \leftarrow VER(m, \sigma, PK, Params)$

Advantage of $\mathcal{A}$ is defined as $\mathcal{A}dv(\mathcal{A}) = \Pr(x = true \wedge m \notin L)$

RSA Signature Scheme

- $N = pq$, for some large primes p, q and $ed \equiv 1 \mod \phi(N)$. Alice's public key is e and her secret key is d.
- To sugn a message $m \in \mathbb{Z}_p$, Alice computes its signature as $\sigma = m^d \mod N$
- To verify if σ is a valid signature on m, Bob checks if

$$m = \sigma^e \mod N$$

If true, then Bob outputs 1, ekse he outputs 0.

Insecurity of RSA Signature

No message attack: One can obtain a forgery using only the public key $(N.e)$. Choose a random $\sigma \in \mathbb{Z}_N^*$ and compute $m = \sigma^e \bmod N$. Then clearly, σ is a valid signature on m since

$$m^d = \sigma^{ed} = \sigma \bmod N.$$

So (m, σ) is a forgery.
Remark: By using a secure hash function, one can obtain a secure signature. (RSA-FDH)