ElGamal Cryptosystem

DR. ODELU VANGA

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRI CITY,
CHITTOOR, INDIA

Public Key Cryptography Early History

Diffie & Hellman: "***New Directions in Cryptography***" IEEE Transactions on Information Theory, Nov 1976.

1. Public-key encryption schemes
2. Key distribution systems
   - Diffie-Hellman key agreement protocol
3. Digital signature

Public-key encryption was proposed in 1970 in a classified paper by James Ellis

- paper made public in 1997 by the British Governmental Communications Headquarters

Public Key Encryption Algorithms

Almost all public-key encryption algorithms use either number theory and modular arithmetic, or elliptic curves

RSA

- based on the hardness of factoring large numbers

ElGamal

- Based on the hardness of solving discrete logarithm
- Use the same idea as Diffie-Hellman key agreement

$a, g^a, G = \langle g \rangle$

Given $\underbrace{g, g^a}$ $\not\Rightarrow a$

$\boxed{a = \log g^a}$

Diffie-Hellman Key Agreement Protocol

Not a Public Key Encryption system, but can allow A and B to agree on a shared secret in a public channel (against passive, i.e., eavesdropping only adversaries).

b g g , g^b , g^a , DLP.

Setup: p prime and g generator of Z_p^*, p and g public

A

Pick random, secret a

Compute and send g^a mod p

$K = (g^b \bmod p)^a = g^{ab} \bmod p$

$y^a = K$

$x = g^a \bmod p$ →

← $y = g^b \bmod p$

$x^b = y^a$

B

Pick random, secret b

Compute and send g^b mod p

$K = (g^a \bmod p)^b = g^{ab} \bmod p$

$K = x^b$

Diffie-Hellman : Example

$2^4 = 5 \bmod 11$

$\log_g$

$4 = \log_g 5$

Let p=11, g=2, then

a	1	2	3	4	5	6	7	8	9	10	11
g^a	2	4	8	16	32	64	128	256	512	1024	2048
$g^a \bmod p$	2	4	8	5	10	9	7	3	6	1	2

1. Alice chooses 4 and Bob chooses 3
2. Then, shared secret is **$(2^3)^4 = (2^4)^3 = 2^{12} = 4 \pmod{11}$**
3. Adversaries sees $2^3=8$ and $2^4=5$
4. So, needs to solve one of $\boxed{\mathbf{2^x=8}}$ and $\boxed{\mathbf{2^y=5}}$ to figure out the shared secret.

2^{xy}

Three Problems Believed to be Hard to Solve

Discrete Log (DLG) Problem:

Given <g, h, p>, computes a such that $g^a = h \bmod p$.

Computational Diffie Hellman (CDH) Problem:

Given <g, g^a mod p, g^b mod p> (without a, b) compute g^{ab} mod p.

Decision Diffie Hellman (DDH) Problem:

Distinguish (g^a, g^b, g^{ab}) from (g^a, g^b, g^c), where a, b, c are randomly and independently chosen.

If one can solve the DL problem, one can solve the CDH problem.

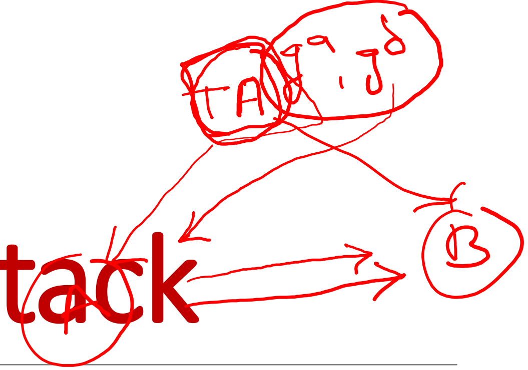
If one can solve CDH, one can solve DDH.

Suppose Alice wants to share files securely to Bob. Alice use the following steps to generate the message and send to Bob.

- Uses Die-Hellman protocol to establish a secret symmetric key *k* with Bob.
- Encrypts the file *f*, which contains (*docName, docContent, userPassword, Nonce*), and *HMAC* of file using the symmetric encryption algorithm with key *k*, that is, $E_k(f, H(k||f))$. Sends the message it to Bob.
- After receiving message from Alice, Bob decrypt it using established secret key k and verify the validity of the le. If all of the checks succeed, the Bob accept the le f sent by Alice.

Q. In the above scenario, how can a network attacker read the *userPassword* ? Justify your answer.

Man-In-The-Middle-Attack



A	Attacker	B
$\langle a, g^a \rangle$	$\langle x, g^x \rangle$	$\langle b, g^b \rangle$
$g^a \longrightarrow$	g^a, g^b	$\longleftarrow g^b$
$\longleftarrow g^x$		$g^x \longrightarrow$

$K_1 = (g^x)^a = g^{xa}$

$K_1 = (g^a)^x$

$K_2 = (g^b)^x$

$K_2 = (g^x)^b = g^{xb}$

CDH assumption ensures that M cannot be fully recovered.

ElGamal Encryption

- Public key **<g, p, h=g^a mod p>** and Private key is **a**

Encryption: chooses random **b** **(one-time use)**,

computes $C=[\mathbf{c_1} = g^b \bmod p,\ \mathbf{c_2} = g^{ab} * M \bmod p]$

- Idea: for each M, sender and receiver establish a shared secret $\mathbf{g^{ab}}$ via the DH protocol.
- The value $\mathbf{g^{ab}}$ hides the message M by multiplying it.

$(g^{ab})^{-1} c_2 = (g^{ab})^{-1} \cdot g^{ab} \cdot M = M$

Decryption: Given $\mathbf{C=[c_1,c_2]}$, computes M where

$$\mathbf{((c_1^{\,a} \bmod p) * M) \bmod p = c_2}$$

- To find M for $\mathbf{x * M \bmod p = c_2}$, compute **z** such that $\mathbf{x*z \bmod p =1}$, then $\mathbf{M = C_2*z \bmod p}$

Real World Usage
Public Key Encryption

- Consider the real-life example where a person pays by credit card and signs a bill; the seller verifies that the signature on the bill is the same with the signature on the card
- Contracts are valid if they are signed.
- Signatures provide non-repudiation.
  - ensuring that a party in a dispute cannot repudiate, or refute the validity of a statement or contract.

private key

$A \rightarrow \{m\}_A \rightarrow$ Bob

$\{m\}_A$

X

1,000,000

C

Optimal Asymmetric Encryption Padding (OAEP)

??

To encrypt M,

- chooses random r
- Encode M as $M' = [X = M \oplus H_1(r)\ ,\ Y = r \oplus H_2(X)\ ]$
  where H_1 and H_2 are cryptographic hash functions
- Then encrypt it as $(M')^{e} \bmod n$
- Note that given $M'=[X,Y]$,

$$r = Y \oplus H_2(X)$$

$$M = X \oplus H_1(r)$$

Non-repudiation

Nonrepudiation is the assurance that someone cannot deny something. Typically, nonrepudiation refers to the ability to ensure that a party to a contract or a communication cannot deny the authenticity of their signature on a document or the sending of a message that they originated.

- Can one deny a signature one has made?
- Does email provide non-repudiation?

Thank You