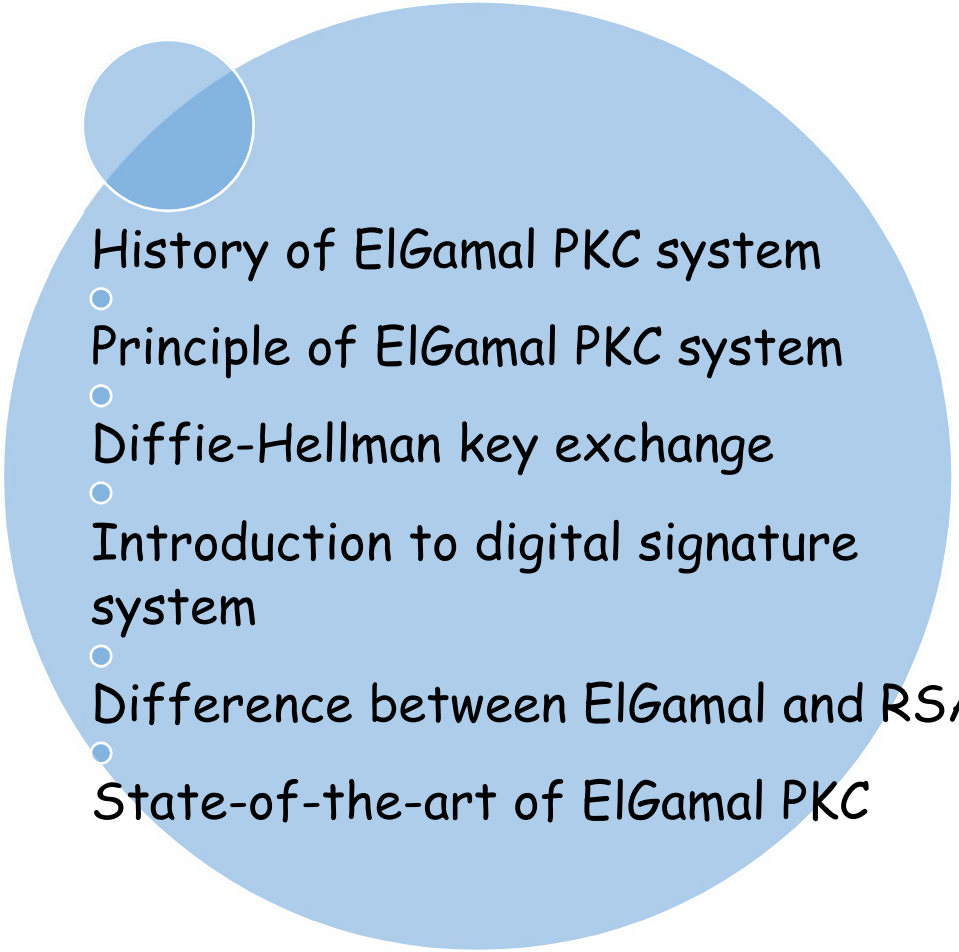
# Introduction to ElGamal Public Key Cryptosystem

Hu Xiong
School of Information and Software Engineering
xionghu.uestc@gmail.com

# History of ElGamal PKC system



# History of ElGamal PKC system

IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. IT-31, NO. 4, JULY 1985



## A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms

TAHER ELGAMAL, MEMBER, IEEE

*Abstract*—A new signature scheme is proposed, together with an implementation of the Diffie–Hellman key distribution scheme that achieves a public key cryptosystem. The security of both systems relies on the difficulty of computing discrete logarithms over finite fields.

### I. Introduction

In 1976, Diffie and Hellman [3] introduced the concept of public key cryptography. Since then, several attempts have been made to find practical public key systems (see, for example, [6], [7], [9]) depending on the difficulty of solving some problems. For example, the Rives–Shamir–Adleman (RSA) system [9] depends on the difficulty of factoring large integers. This paper presents systems that rely on the difficulty of computing logarithms over finite fields.

Section II shows a way to implement the public key distribution scheme introduced by Diffie and Hellman [3] to encrypt and decrypt messages. The security of this system is equivalent to that of the distribution scheme. Section III introduces a new digital signature scheme that depends on the difficulty of computing discrete logarithms over finite fields. It is not yet proved that breaking the system is equivalent to computing discrete logarithms. Section IV develops some attacks on the signature scheme, none of which seems to break it. Section V gives some properties of the system. Section VI contains a conclusion and some remarks.

Hence both $A$ and $B$ are able to compute $K_{AB}$. But, for an intruder, computing $K_{AB}$ appears to be difficult. It is not yet proved that breaking the system is equivalent to computing discrete logarithms. For more details refer to [3].

In any of the cryptographic systems based on discrete logarithms, $p$ must be chosen such that $p-1$ has at least one large prime factor. If $p-1$ has only small prime factors, then computing discrete logarithms is easy (see [8]).

Now suppose that $A$ wants to send $B$ a message $m$, where $0 \le m \le p-1$. First $A$ chooses a number $k$ uniformly between 0 and $p-1$. Note that $k$ will serve as the secret $x_A$ in the key distribution scheme. Then $A$ computes the "key"

$$K \equiv y_B^k \bmod p, \tag{1}$$

where $y_B \equiv \alpha^{x_B} \bmod p$ is either in a public file or is sent by $B$. The encrypted message (or ciphertext) is then the pair $(c_1, c_2)$, where

$$c_1 = \alpha^k \bmod p \qquad c_2 \equiv Km \bmod p \tag{2}$$

and $K$ is computed in (1).

Note that the size of the ciphertext is double the size of the message. Also note that the multiplication operation in (2) can be replaced by any other invertible operation such as addition mod $p$.

The decryption operation splits into two parts. The first step is recovering $K$, which is easy for $B$ since $K \equiv (\alpha^k)^{x_B}$ $= \alpha^{x_B} \bmod p$, and $x_B$ is known to $B$ only. The second step

# Principle of ElGamal PKC system

- History of ElGamal PKC system
- Principle of ElGamal PKC system
- Diffie-Hellman key exchange
- Introduction to digital signature system
- Difference between ElGamal and RSA
- State-of-the-art of ElGamal PKC

# Principle of ElGamal PKC system

**KeyGen:**

- $p$, a large prime
- $g$, a generator of $Z_p^*$
- $\alpha \in Z_{p-1}, \beta = g^{\alpha} \bmod p$
- $\{p, g, \beta\}$ public; $\alpha$ private

**Enc/Dec:**

- generate random, secret $k \in Z_{p-1}$ .
- $\mathrm{Enc}(x, k) = (r, s)$ , where

$$r = g^k \bmod p$$

$$s = x\beta^k \bmod p$$

- $\mathrm{Dec}(r, s) = s(r^{\alpha})^{-1} \bmod p = xg^{\alpha k}g^{-\alpha k} \bmod p = x$

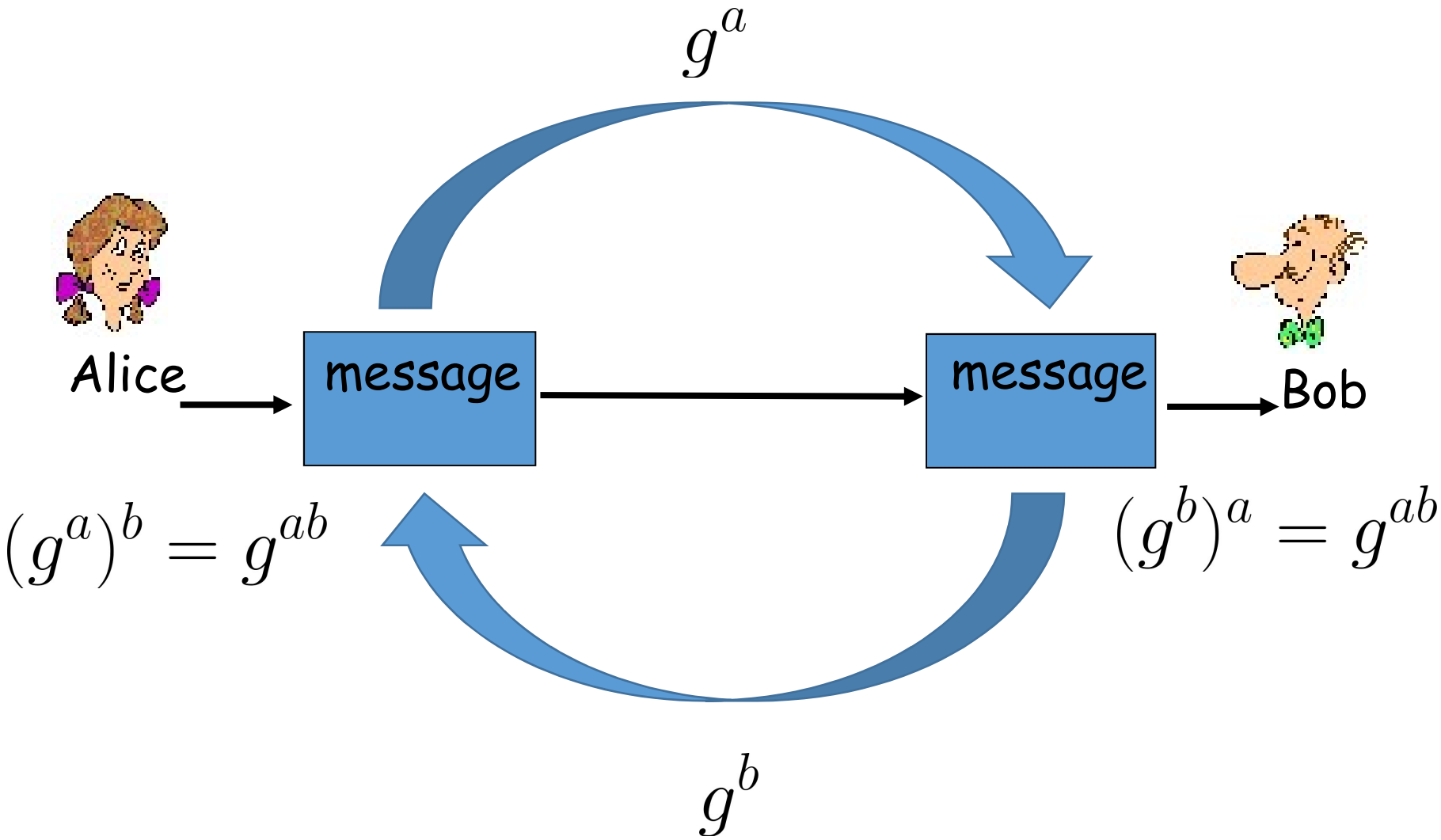
# Principle of ElGamal PKC system

- Plaintext $x$ is masked by a random factor, $g^{\alpha k} \bmod p$ .
- **DH problem**: Given $g^{\alpha}$ , $g^{k} \bmod p$ , what is $g^{\alpha k} \bmod p$ ?
- $p, g$ can be common. Then $g^{k} \bmod p$ can be computed in advance.
- Same $k$ should not be used repeatedly.
- **Performance**:
  - encryption: two exponentiations
  - decryption: one exponentiation, one inversion
- **Size**: Ciphertext twice as large as plaintext.

# Diffie-Hellman key exchange

- History of ElGamal PKC system
- Principle of ElGamal PKC system
- Diffie-Hellman key exchange
- Introduction to digital signature system
- Difference between ElGamal and RSA
- State-of-the-art of ElGamal PKC

# Diffie-Hellman key exchange



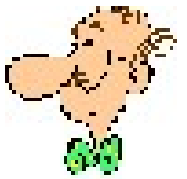
# Diffie-Hellman key exchange

Alice

message

Man-in-the-middle attack

message



Bob

$g^a$

$g^b$

$(g^a)^b = g^{ab}$

$(g^b)^a = g^{ab}$

# Introduction to digital signature system

- History of ElGamal PKC system
- Principle of ElGamal PKC system
- Diffie-Hellman key exchange
- Introduction to digital signature system
- Difference between ElGamal and RSA
- State-of-the-art of ElGamal PKC

# ElGamal – Signature

**KeyGen**: The same as encryption.

**Sign**:

- generate random, secret $k \in Z_{p-1}^*$ .
- $\mathrm{Sign}(m,k) = (r,s)$ , where

$$r = g^k \bmod p$$
$$s = (m - r\alpha)k^{-1} \bmod (p-1)$$

(i.e., $m = r\alpha + sk$ )

**Verify**:

- Is $\beta^r r^s \equiv g^m (\bmod\ p)$ ?
- $\beta^r r^s = g^{\alpha r} g^{k(m-r\alpha)k^{-1}} = g^{\alpha r + (m-r\alpha)} = g^m \bmod p$

# ElGamal – Signature

**Security:**

- Only one who knows $\alpha$ can sign; can be verified by $\beta$ .
- Solving $\alpha$ from $\beta$, or $s$ from $r, m, \beta$ , is discrete log.
- Other ways of forgery? Unknown.
- Same $k$ should not be used repeatedly.

**Variations:**

- Many variants, by changing the "signing equation",
  $$m = r\alpha + sk \ .$$
- E.g., the DSA way:
  $$m = -r\alpha + sk$$
  with verification: $\beta^r g^m \equiv r^s (\mathrm{mod}\ p)?(\equiv g^{m+r\alpha})$

# Schnorr - Signature

- Let $q|(p-1)$ be prime, and $g \in Z_p^*$ be of order $q$ .

- **Schnorr group**: The subgroup in $Z_P^*$ generated by $g$, of prime order $q$.

$$< g >= \{1, g, g^2, \cdots, g^{q-1}\}$$

- **Fact:** $q$ can be much shorter than $p$ (e.g. 160 vs. 1024 bits), and the hardness of DLP in $< g >$ remains the same.

# Schnorr - Signature

**Parameters:** prime $p$, prime $q|(p-1)$, and $g \in Z_p^*$ of order $q$. Hash fnc. $H : \{0,1\}^* \rightarrow Z_q$.

**Keys:** $\alpha \in Z_q$ is private; $\beta = (g^\alpha \bmod p)$ is public.

**Signature:** $(r, s)$ where

$$v = g^k \bmod p$$

$$r = H(M||v)$$

$$s = (k - r\alpha) \bmod q$$

**Verification:**

$$v' = g^s \beta^r \bmod p$$

$$r = H(M||v')?$$

**Advantage:** Reduced size & complexity

# Digital Signature Algorithm (DSA)

- US government standard, by NSA.
- Based on ElGamal & Schnorr:
  - patent-free (ElGamal)
  - can't be used for encryption
- Objections:
  - ElGamal was not analyzed as much as RSA
  - slower verification
  - industry had already invested in RSA
  - closed-door design

# Digital Signature Algorithm (DSA)

**Parameters:** The same as Schnorr's.

**Signature:** $(r, s)$ where

$$v = g^k \bmod p$$

$$r = v \bmod q$$

$$s = (H(M) + r\alpha)k^{-1} \bmod q$$

**Verification:**

$$v' = g^{H(M)S^{-1}} \beta^{rs^{-1}} \bmod p$$

$$r = v' \bmod q$$

(compared to Schnorr?)

# Difference between ElGamal and RSA

- History of ElGamal PKC system
- Principle of ElGamal PKC system
- Diffie-Hellman key exchange
- Introduction to digital signature system
- Difference between ElGamal and RSA
- State-of-the-art of ElGamal PKC

# Difference between ElGamal and RSA

- Mathematical hard problems
- Public parameter settings

# State-of-the-art of ElGamal PKC

- History of ElGamal PKC system
- Principle of ElGamal PKC system
- Diffie-Hellman key exchange
- Introduction to digital signature system
- Difference between ElGamal and RSA
- State-of-the-art of ElGamal PKC

# Elliptic Curve Cryptosystems

**Generalized Discrete Log Problem:**

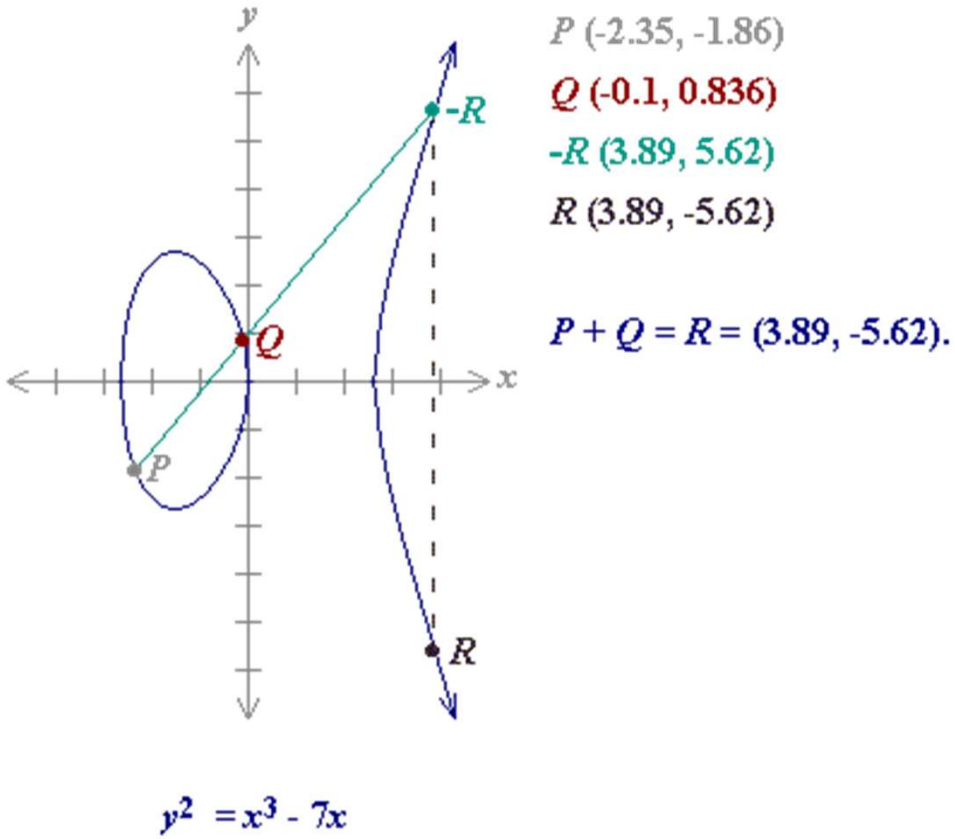
- For any group $(G, \cdot)$ , for $x \in G$ , define
$$x^n = x \cdot x \cdots x \quad (n \text{ times})$$
- DLP: For $y = x^n$, given $x, y$ , what is $n$ ?

**Elliptic curves over $Zp$ :**

- Set of points $(x, y) \in Z_p \times Z_p$ that satisfy
$$y^2 \equiv x^3 + ax + b (\text{mod } p)$$
and an additional point of infinity, 0.
- Group operation: $P \cdot Q$ is the inverse of where the line thru $P\&Q$ intersects the curve. (inverse of $P = (x, y)$ is defined as $P^{-1} = (x, -y)$ ).
- Well-defined, provided that $4a^3 \neq -27b^2 \bmod p$ .

# Elliptic Curve Cryptosystems

EC example over $R^2$ :



# Elliptic Curve Cryptosystems

- Facts for an EC over a finite field:
  - Exponentiation is efficient.
  - DLP is hard. In fact, harder than in $Z_p$. (no sub-exponential algorithm is known)
- Hence, DH, ElGamal, etc. can be used with smaller key sizes over ECs. (160-bit EC ~ 1024-bit RSA)
- Popular for constrained devices (e.g., smart cards)
- Advantages over RSA:
  - smaller key size
  - compact in hardware
  - faster (for private key operations)
- Licensed by NSA.

# Thanks !