Manchester Metropolitan University

Public Key Cryptography

Killian O’Brien

6G6Z0024 Applied Cryptography 2024/25

Lecture Week 08 – Wed 20 November 2024

Introduction

- Reading: Stallings, Chapter 10, Sections 10.1, 10.2, *Other Public-key cryptosystems*

In this lecture we shall look at

- Diffie-Hellman key exchange
- the ElGamal system

Key Exchange

- In practice, public key cryptography works hand in hand with symmetric encryption, such as AES
- A common approach is for public key encryption to be used to exchange a private key, that is then used to commence communication under symmetric encryption, such as AES.

DH key exchange

- Diffie-Hellman key exchange (1976) is a method that allows two parties to agree on a shared secret (number) by the exchange unencrypted messages.
- This shared secret can then be used to generate a shared secret key to enable secure, encrypted communication.
- DH key exchange uses a tool from modular arithmetic called *discrete logarithms*.

Reminder: What are logarithms?

- **Exponentials** – Given numbers a, b, a number of the form $c = a^b$, is called an *exponential* (or power), with *base* a, and *exponent* b.
    - we usually have $a > 0$
    - for positive integers b, a^b is given by repeated multiplication, i.e.

      $$a^b = a \cdot a \cdot \dots \cdot a,$$

      the product of b factors of a.
    - exponentials with negative exponents are defined using

      $$a^{-b} = \frac{1}{a^b}$$

- **Logarithms** – These are the *inverses* of exponentials.
    - So if $c = a^b$ then
    - $\log_a(c) = b$.
    - We say b is the *logarithm*, with base a, of c.
- You might see a function $\log$ mentioned without a base. Its meaning is usually determined by context/discipline in which it is used.
    - Computer scientists would usually mean $\log_2$,
    - Mathematicians would usually mean $\log_e$, the *natural logarithm*,
    - Engineers might usually mean $\log 10$.

What are discrete logarithms?

- *Discrete logarithms* is the term for logarithms within modular arithmetic.
- Working modulo a prime p, if $c \equiv a^b \pmod{p}$ then
  - b is the discrete logarithm of c, to the base a, modulo p.
  - Stallings introduces the notation

$$b = \operatorname{dlog}_{a,p}(c).$$

- Usually the base used is a *primitive root* modulo p, i.e. a number a whose powers generate **ALL** the non-zero elements modulo p, i.e. the residues

$$(a \bmod p),\ (a^2 \bmod p),\ (a^3 \bmod p), \ldots, (a^{p-1} \bmod p),$$

  are all distinct and consist of the integers

$$1, 2, \ldots, p-1,$$

  (though probably not in that order).
- Security arises from the facts that
  - computing modular exponentials is fast (using modular reduction, Euler's theorem and repeated squaring)
  - there is no known fast algorithm for computing discrete logarithms. So for suitable large p, it cannot be done in any practical way.

Diffie-Hellman key exchange algorithm – How Alice and Bob can agree on a shared secret

- Alice and Bob agree on a (large) prime q, and a primitive root α, modulo q.
- Alice and Bob generate their own **private** keys $X_A, X_B < q$.
- Alice and Bob then calculate their **public keys**

$$Y_A = (\alpha^{X_A} \bmod q), \quad Y_B = (\alpha^{X_B} \bmod q).$$

- Alice and Bob exchange their public keys Y_A and Y_B. Note that the private keys X_A and X_B are kept private and not exposed.
- Alice calculates

$$K = ((Y_B)^{X_A} \bmod q)$$

and Bob calculates

$$K = ((Y_A)^{X_B} \bmod q)$$

- Note that

$$K = \left((Y_B)^{X_A} = (\alpha^{X_B})^{X_A} = \alpha^{X_B \cdot X_A} = (\alpha^{X_A})^{X_B} = (Y_A)^{X_B} \bmod q\right)$$

- Alice and Bob both know the shared secret K. But nobody else does. Even if they've eavesdropped on all these setup communications.

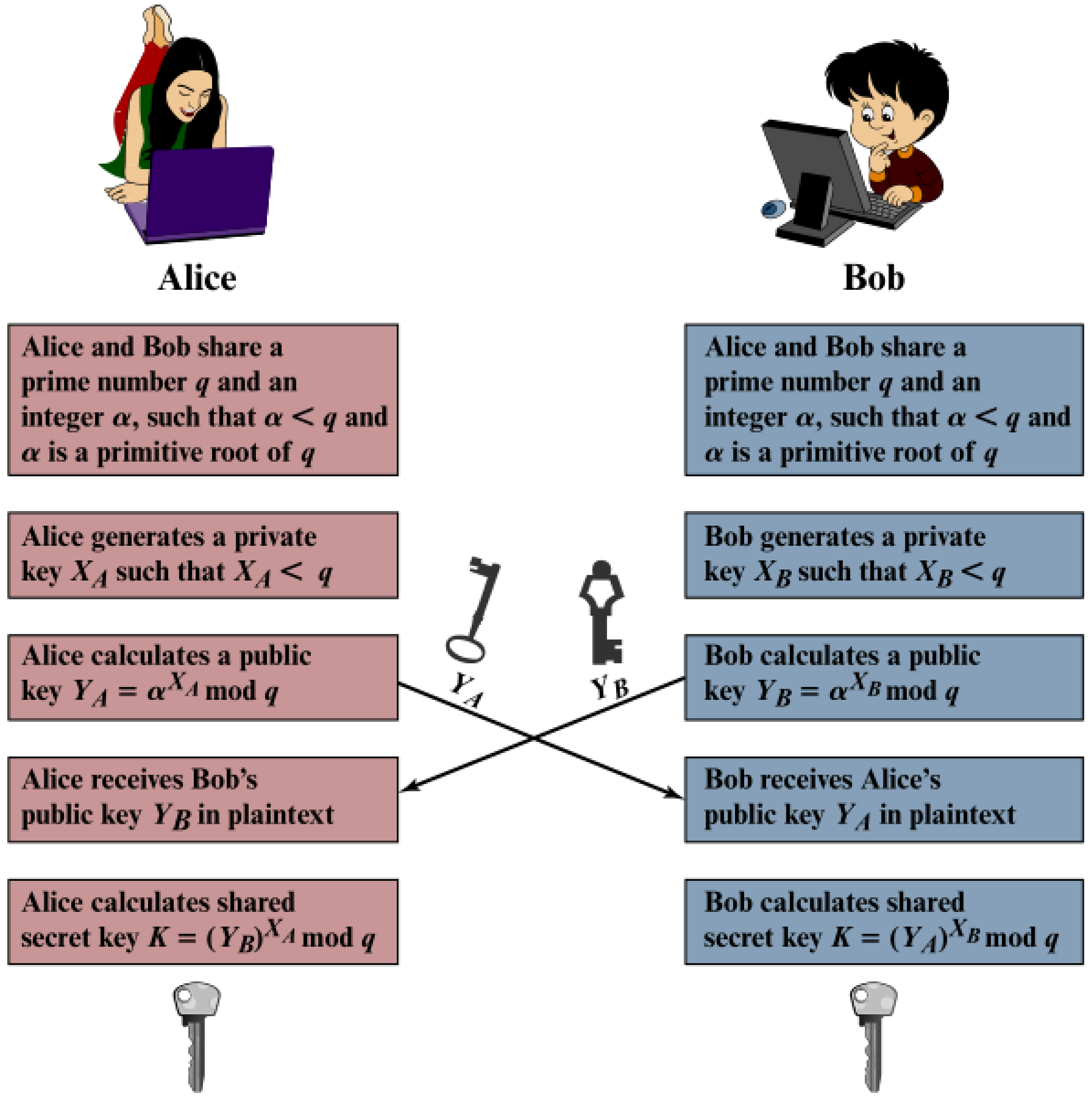

Figure 10.1 The Diffie–Hellman Key Exchange

ElGamal cryptosystem

- From 1984, provides an enhancement of DH key exchange that introduces an element of authentication into the exchanges.
- Stallings' figure to the right shows what's required for Bob to send encrypted communication to Alice.
- Alice generates and publishes her public key $(q, \alpha, Y_A = \alpha^{X_A})$ as before, still retaining her private key X_A.
- Bob prepares plaintext message block M and an element k that is used to caluclate a temporary secret key $K = (Y_A^k \bmod q)$.
- Bob calculates

$$C_1 = (\alpha^k \bmod q),\ C_2 = (KM \bmod q),$$

  and sends the pair (C_1, C_2) to Alice.
- Alice can recover the secret key K by computing

$$K = (C_1^{X_A} \bmod q),$$

  and decrypt the message M by computing

$$M = (C_2 K^{-1} \bmod q).$$

Global Public Elements	
q	prime number
α	$\alpha < q$ and α a primitive root of q

Key Generation by Alice	
Select private X_A	$X_A < q - 1$
Calculate Y_A	$Y_A = \alpha^{X_A} \bmod q$
Public key	$\{q, \alpha, Y_A\}$
Private key	X_A

Encryption by Bob with Alice's Public Key	
Plaintext:	$M < q$
Select random integer k	$k < q$
Calculate K	$K = (Y_A)^k \bmod q$
Calculate C_1	$C_1 = \alpha^k \bmod q$
Calculate C_2	$C_2 = KM \bmod q$
Ciphertext:	(C_1, C_2)

Decryption by Alice with Alice's Private Key	
Ciphertext:	(C_1, C_2)
Calculate K	$K = (C_1)^{X_A} \bmod q$
Plaintext:	$M = (C_2 K^{-1}) \bmod q$

Figure 10.3 The ElGamal Cryptosystem