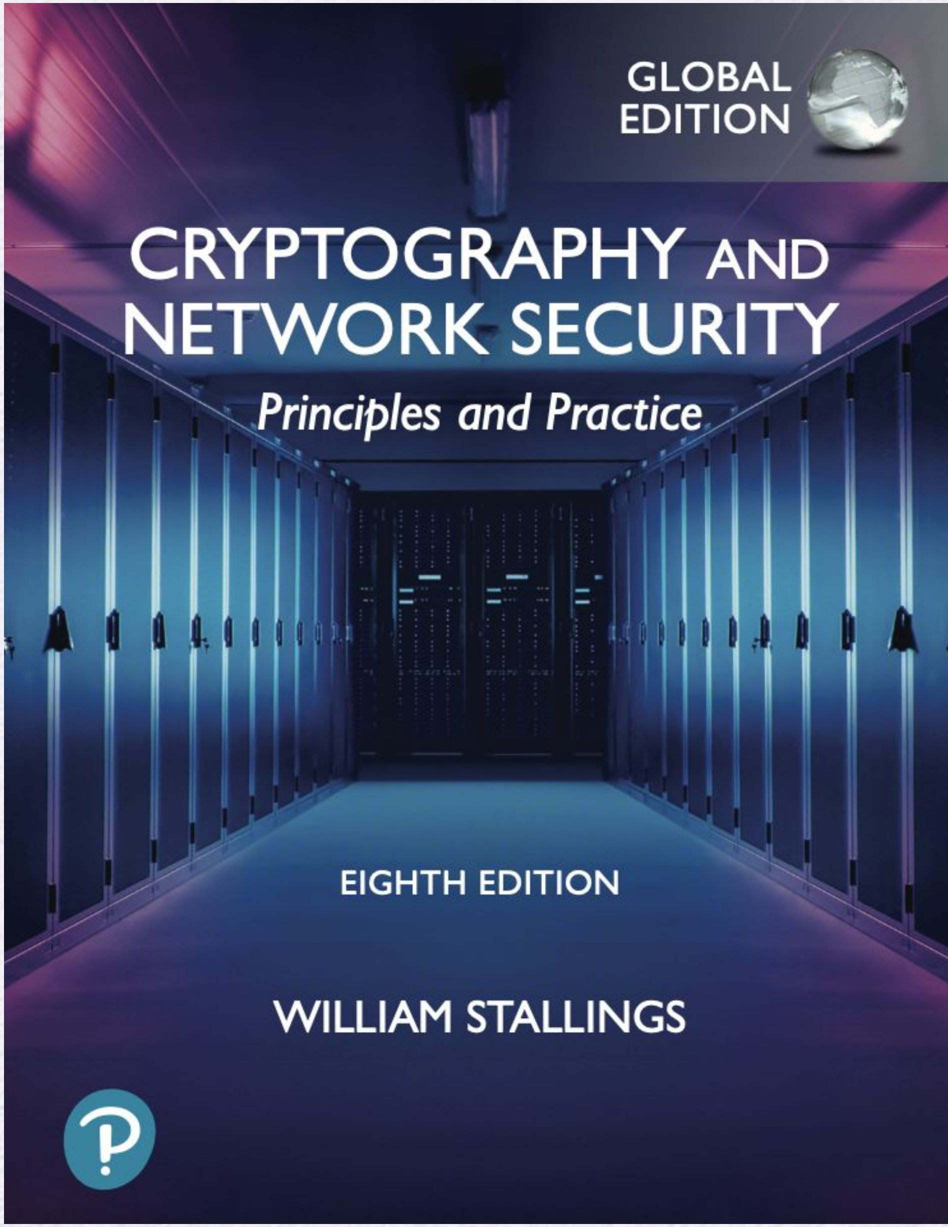
University of Nevada – Reno
Computer Science & Engineering Department

CS454/654 Reliability and Security of Computing Systems - Fall 2024

Lecture 16

Dr. Batyr Charyyev
bcharyyev.com



CHAPTER 10

Other Public-Key Cryptosystems

Diffie-Hellman Key Exchange

- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- Purpose is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages
- The algorithm itself is limited to the exchange of secret values
- Its effectiveness depends on the difficulty of computing discrete logarithms

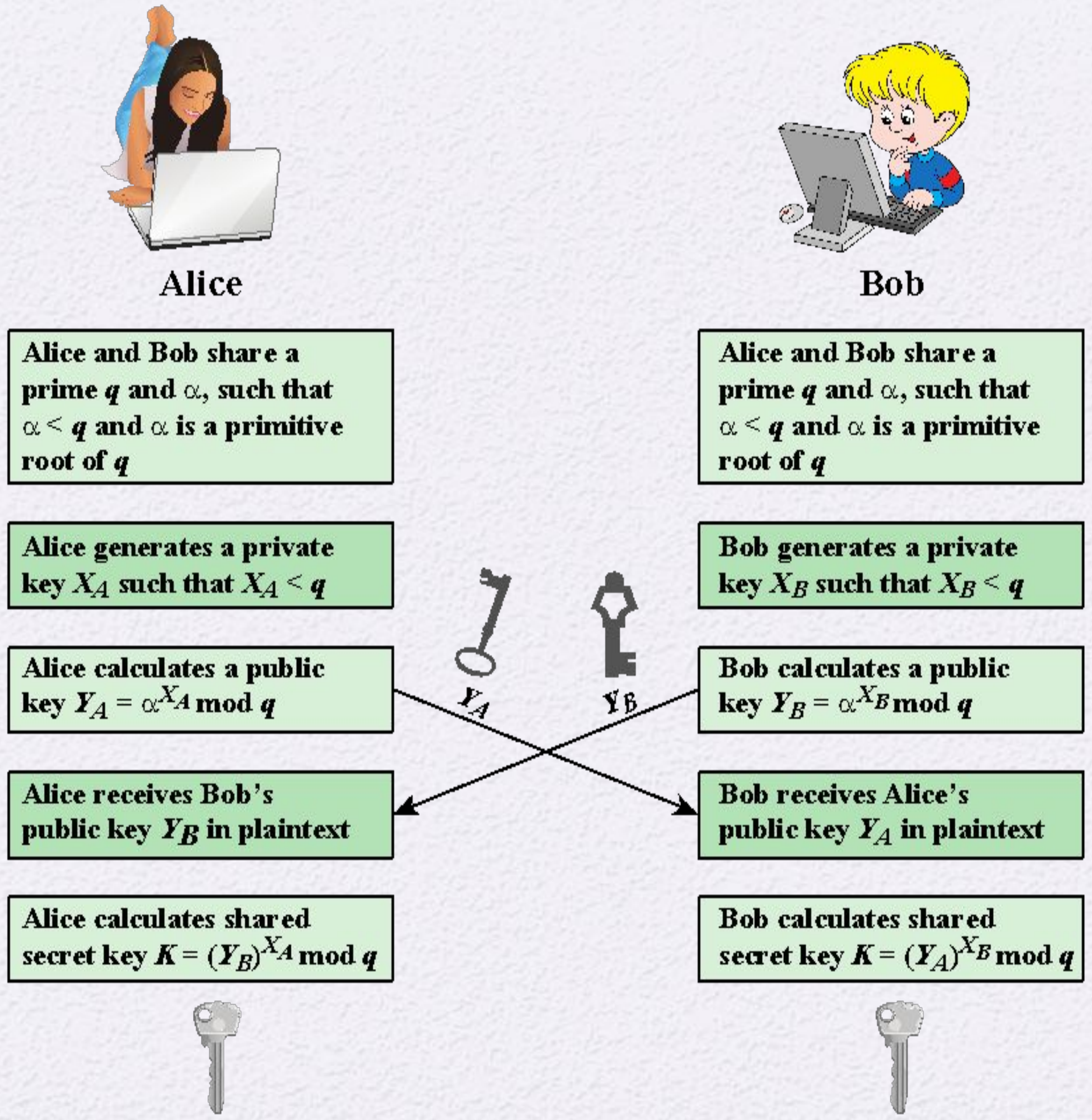


Figure 10.1 Diffie-Hellman Key Exchange

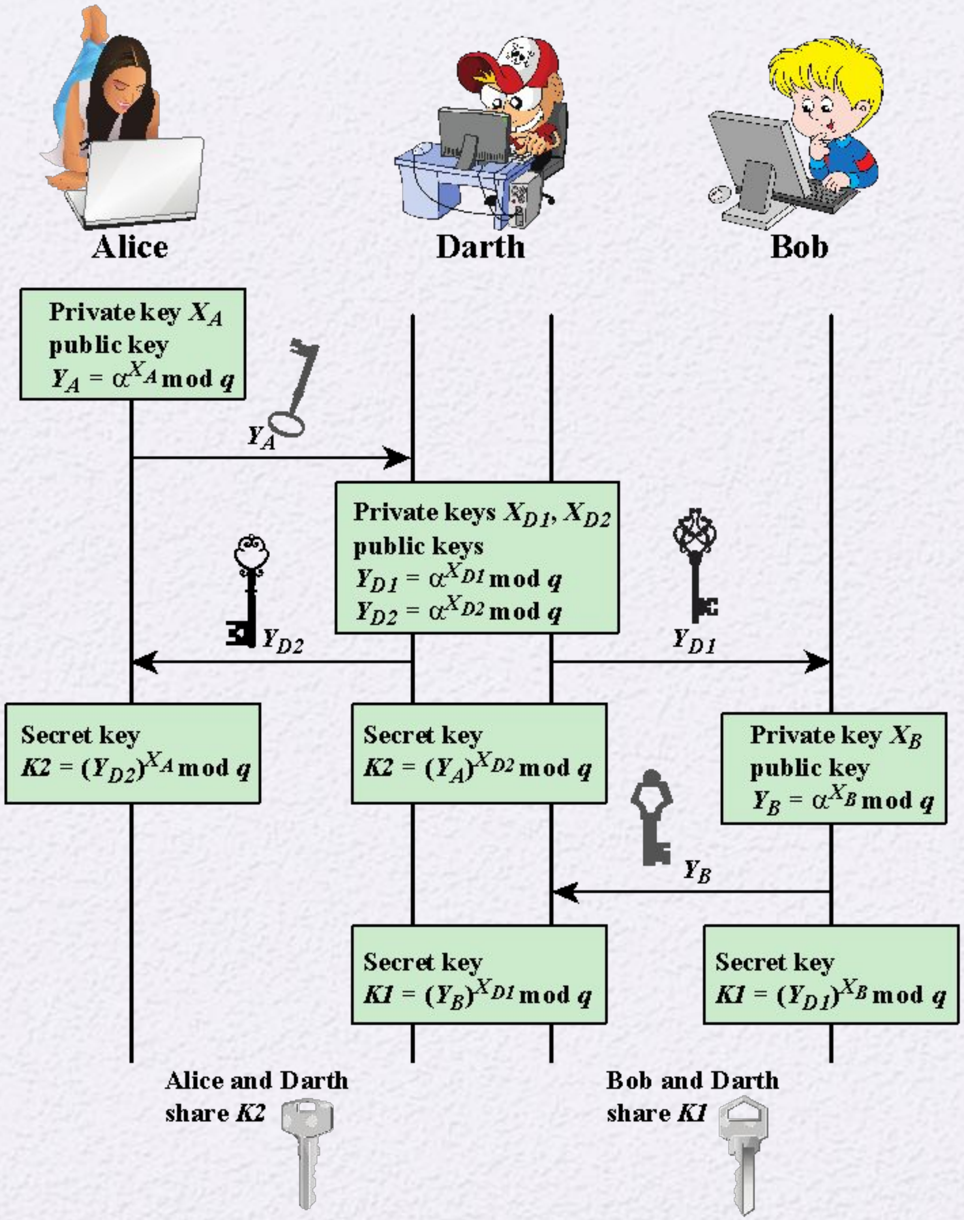


Figure 10.2 Man-in-the-Middle Attack

ElGamal Cryptography

Announced in 1984 by T. Elgamal

Public-key scheme based on discrete logarithms closely related to the Diffie-Hellman technique

Used in the digital signature standard (DSS) and the S/MIME e-mail standard

Global elements are a prime number q and a which is a primitive root of q

Security is based on the difficulty of computing discrete logarithms

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

Elliptic Curve Arithmetic

- Most of the products and standards that use public-key cryptography for encryption and digital signatures use RSA
  - The key length for secure RSA use has increased over recent years and this has put a heavier processing load on applications using RSA
- Elliptic curve cryptography (ECC) is showing up in standardization efforts including the IEEE P1363 Standard for Public-Key Cryptography
- Principal attraction of ECC is that it appears to offer equal security for a far smaller key size

Abelian Group

- A set of elements with a binary operation, denoted by $\cdot$, that associates to each ordered pair (a, b) of elements in G an element $(a \cdot b)$ in G, such that the following axioms are obeyed:

(A1) Closure: If a and b belong to G, then $a \cdot b$ is also in G

(A2) Associative: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ for all a, b, c in G

(A3) Identity element: There is an element e in G such that $a \cdot e = e \cdot a = a$ for all a in G

(A4) Inverse element: For each a in G there is an element a' in G such that $a \cdot a' = a' \cdot a = e$

(A5) Commutative: $a \cdot b = b \cdot a$ for all a, b in G

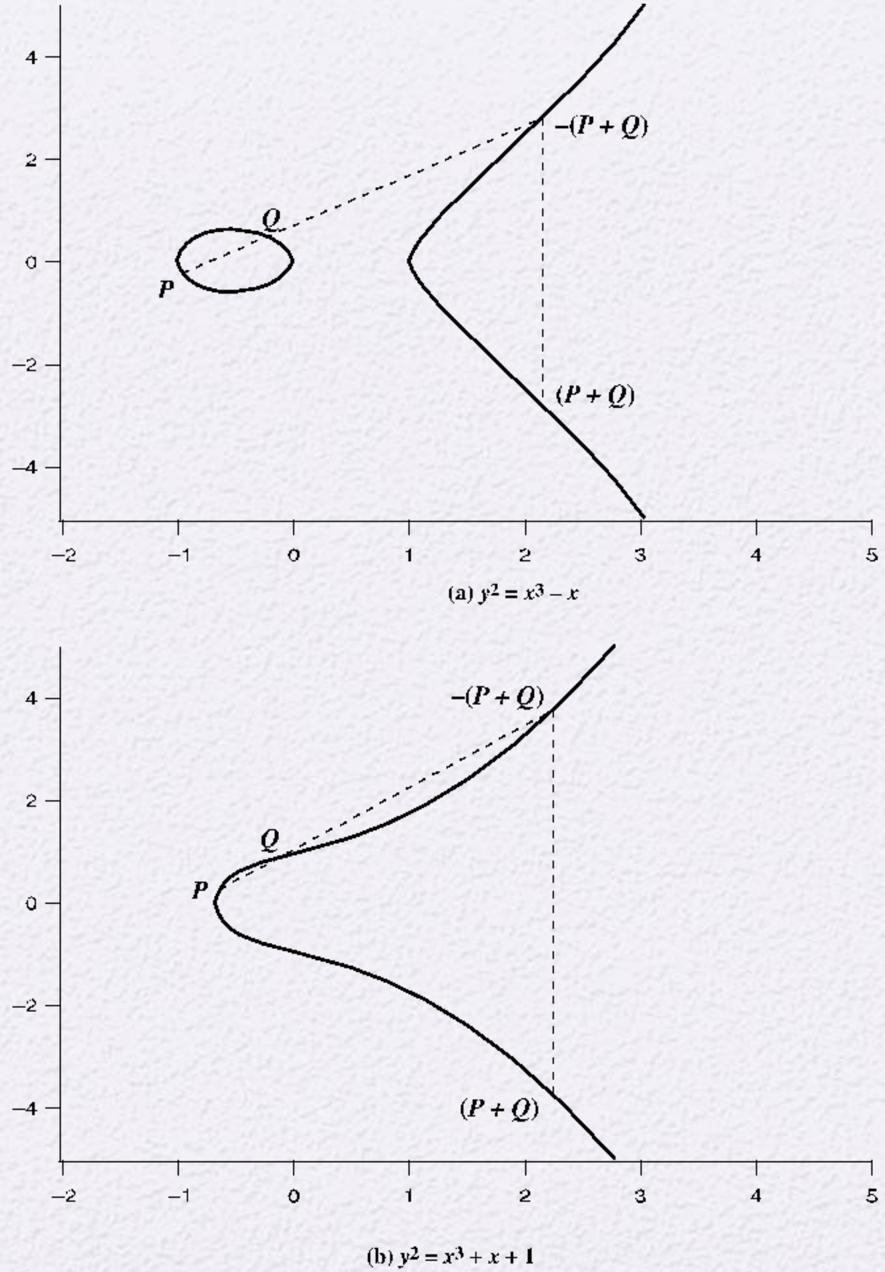
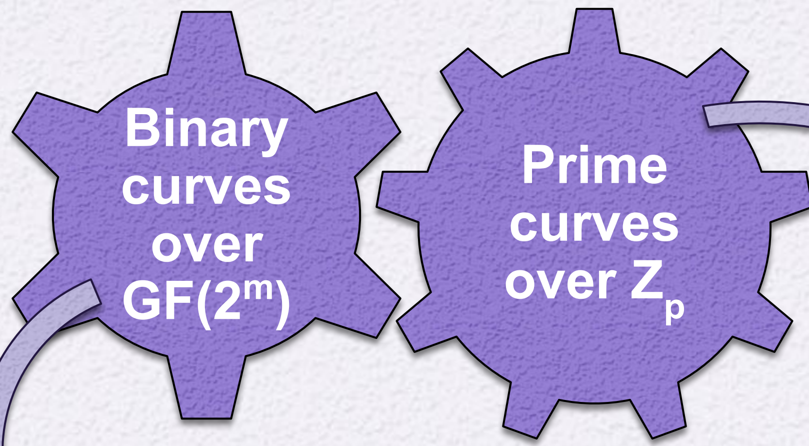


Figure 10.4 Example of Elliptic Curves

Elliptic Curves Over Z_p

- Elliptic curve cryptography uses curves whose variables and coefficients are finite
- Two families of elliptic curves are used in cryptographic applications:



- **Variables and coefficients all take on values in GF(2^m) and in calculations are performed over GF(2^m)**
- **Best for hardware applications**

- **Use a cubic equation in which the variables and coefficients all take on values in the set of integers from 0 through p-1 and in which calculations are performed modulo p**
- **Best for software applications**

Table 10.1

Points (other than O) on the Elliptic Curve $E_{23}(1, 1)$

(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)

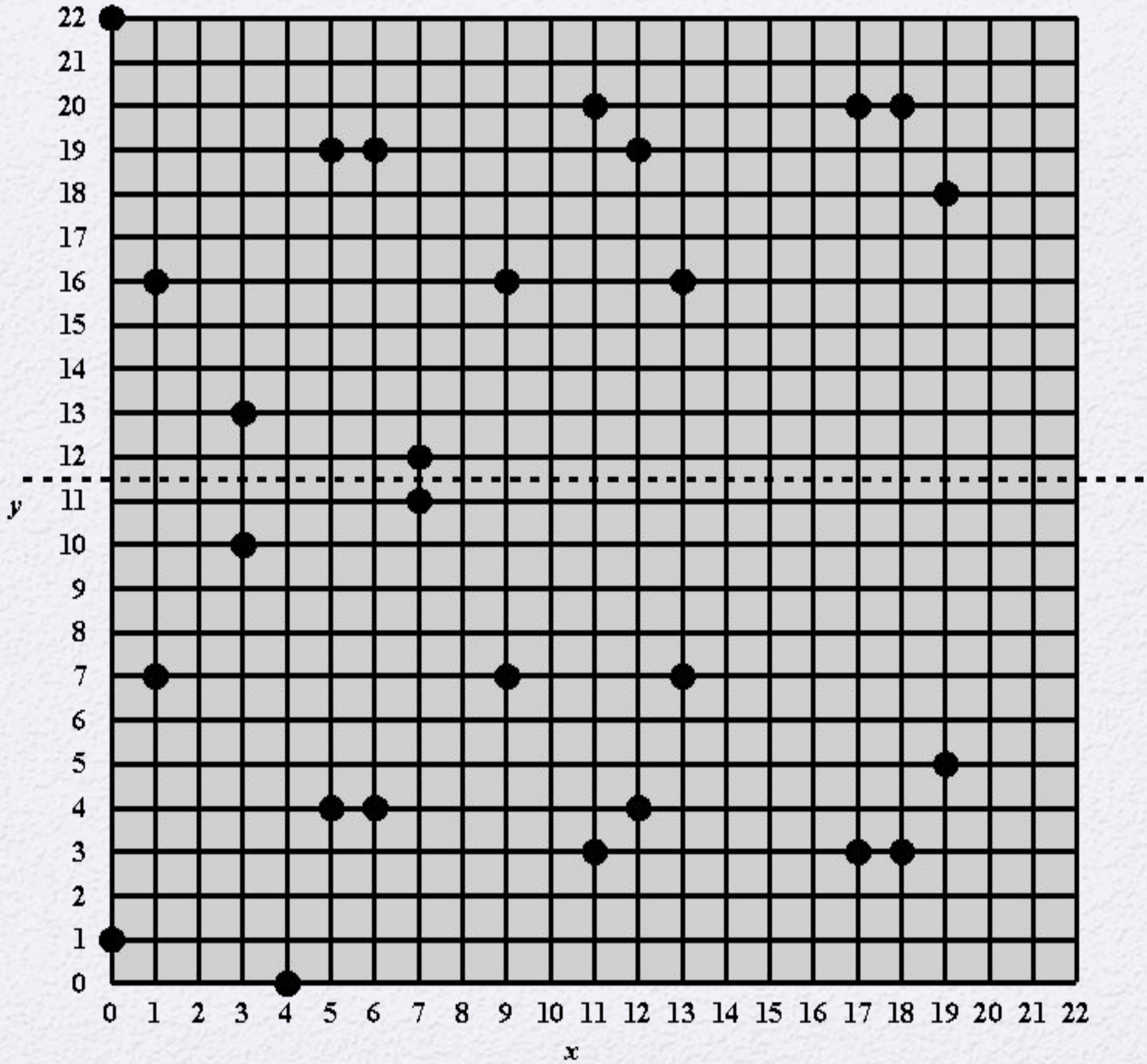


Figure 10.5 The Elliptic Curve $E_{23}(1,1)$

Elliptic Curves Over GF(2^m)

- Use a cubic equation in which the variables and coefficients all take on values in GF(2^m) for some number *m*

- Calculations are performed using the rules of arithmetic in GF(2^m)

- The form of cubic equation appropriate for cryptographic applications for elliptic curves is somewhat different for GF(2^m) than for Z_p
  - It is understood that the variables *x* and *y* and the coefficients *a* and *b* are elements of GF(2^m) and that calculations are performed in GF(2^m)

Table 10.2

Points (other than O) on the Elliptic Curve $E_{2^4}(g^4, 1)$

$(0, 1)$	(g^5, g^3)	(g^9, g^{13})
$(1, g^6)$	(g^5, g^{11})	(g^{10}, g)
$(1, g^{13})$	(g^6, g^8)	(g^{10}, g^8)
(g^3, g^8)	(g^6, g^{14})	$(g^{12}, 0)$
(g^3, g^{13})	(g^9, g^{10})	(g^{12}, g^{12})

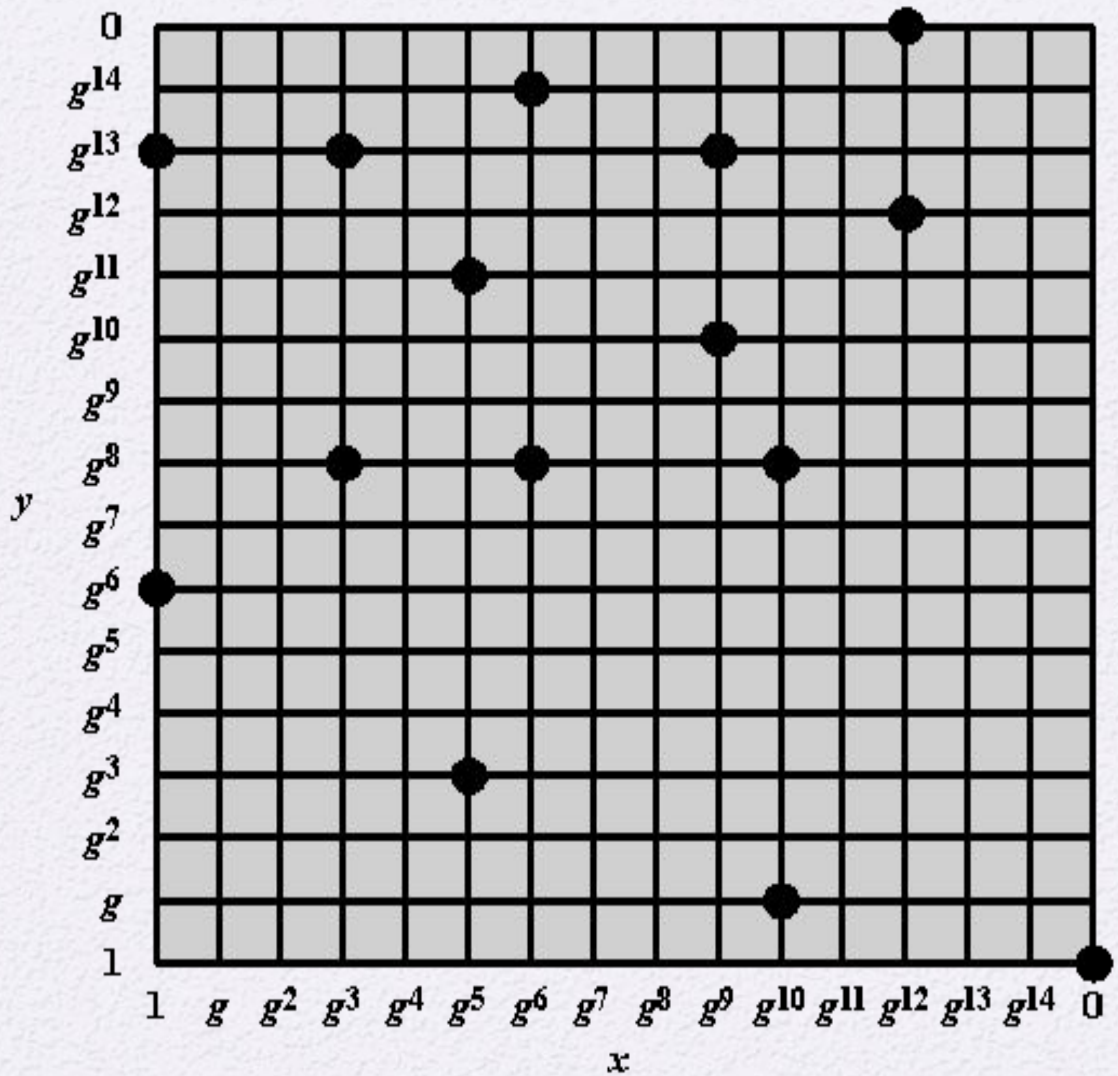


Figure 10.6 The Elliptic Curve $E_{2^4}(g^4, 1)$

Elliptic Curve Cryptography (ECC)

- Addition operation in ECC is the counterpart of modular multiplication in RSA
- Multiple addition is the counterpart of modular exponentiation

To form a cryptographic system using elliptic curves, we need to find a "hard problem" corresponding to factoring the product of two primes or taking the discrete logarithm

- $Q=kP$, where Q, P belong to a prime curve
- Is "easy" to compute Q given k and P
- But "hard" to find k given Q, and P
- Known as the elliptic curve logarithm problem

Global Public Elements

$E_q(a, b)$	elliptic curve with parameters a, b, and q, where q is a prime or an integer of the form 2^m
G	point on elliptic curve whose order is large value n

User A Key Generation

Select private n_A	$n_A < n$
Calculate public P_A	$P_A = n_A \times G$

User B Key Generation

Select private n_B	$n_B < n$
Calculate public P_B	$P_B = n_B \times G$

Calculation of Secret Key by User A

$K = n_A \times P_B$

Calculation of Secret Key by User B

$K = n_B \times P_A$

Figure 10.7 ECC Diffie-Hellman Key Exchange

Security of Elliptic Curve Cryptography

- Depends on the difficulty of the elliptic curve logarithm problem
- Fastest known technique is “Pollard rho method”
- Compared to factoring, can use much smaller key sizes than with RSA
- For equivalent key lengths computations are roughly equivalent
- Hence, for similar security ECC offers significant computational advantages

Table 10.3

Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis (NIST SP-800-57)

Symmetric key algorithms	Diffie-Hellman, Digital Signature Algorithm	RSA (size of n in bits)	ECC (modulus size in bits)
80	$L = 1024$ $N = 160$	1024	160–223
112	$L = 2048$ $N = 224$	2048	224–255
128	$L = 3072$ $N = 256$	3072	256–383
192	$L = 7680$ $N = 384$	7680	384–511
256	$L = 15,360$ $N = 512$	15,360	512+

Note: L = size of public key, N = size of private key

Public key cryptography fundamentally relies on the concept of a trapdoor function. It operates like a sophisticated trap: a rat (representing the data) gets caught using the public key, but it cannot escape on its own. However, with the knowledge of a specific method—the private key—the rat can be freed effortlessly. This elegantly demonstrates the power and precision of mathematics, ensuring that while anyone can set the trap (encrypt), only the intended recipient can unlock it (decrypt) 😊 .

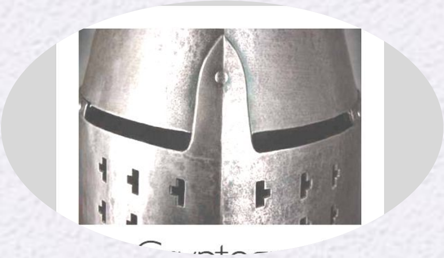
#PublicKeyCryptography

#Security

#RSA

Summary

- Define Diffie-Hellman Key Exchange
- Understand the Man-in-the-middle attack
- Present an overview of the Elgamal cryptographic system



- Understand Elliptic curve arithmetic
- Present an overview of elliptic curve cryptography
- Present two techniques for generating pseudorandom numbers using an asymmetric cipher