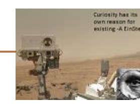
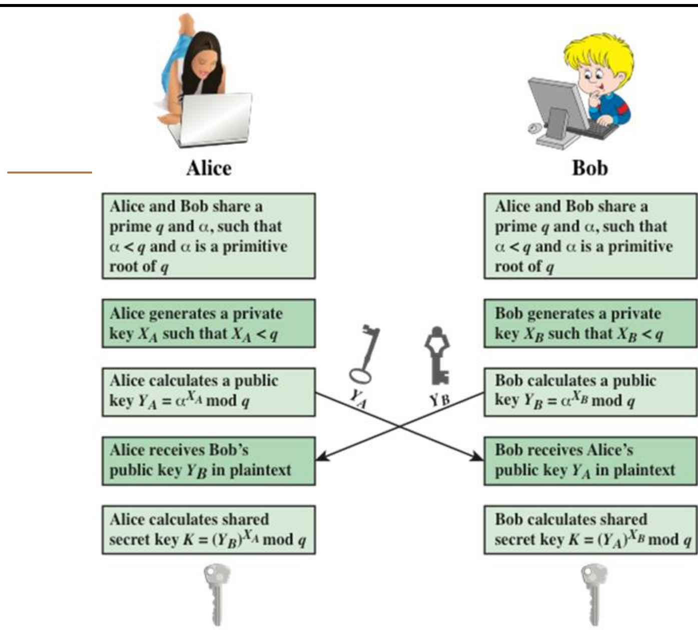
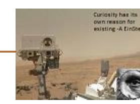
Chapter 10: Other Public-Key Cryptosystems



Diffie-Hellman Key Exchange

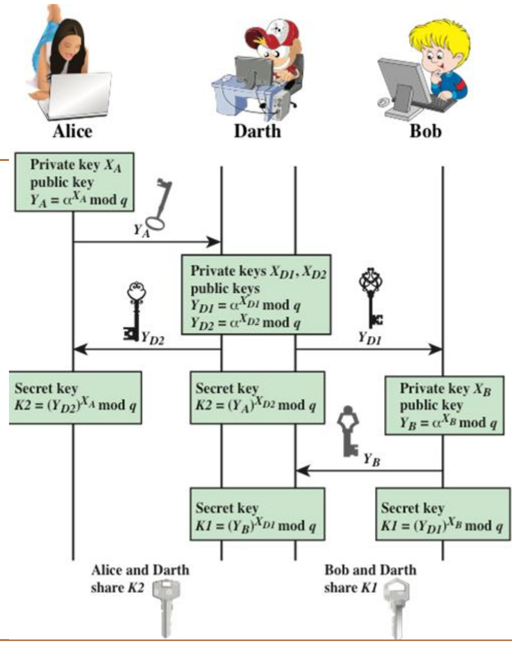
- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- Purpose is to enable two users to securely exchange a key that can then be used for subsequent symmetric encryption of messages
- The algorithm itself is limited to the exchange of secret values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

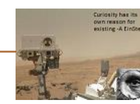




Key Exchange Protocols

- Users could create random private/public Diffie-Hellman keys each time they communicate
- Users could create a known private/public Diffie-Hellman key and publish in a directory, then consulted and used to securely communicate with them
- Vulnerable to Man-in-the-Middle-Attack
- Authentication of the keys is needed





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:	$M = (C_2K^{-1}) \bmod q$

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
- Confidence level in ECC is not yet as high as that in RSA

Abelian Group

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

  (A1) Closure: If *a* and *b* belong to *G*, then *a* • *b* is also in *G*

  (A2) Associative: *a* • (*b* • *c*) = (*a* • *b*) • *c* for all *a*, *b*, *c* in *G*

  (A3) Identity element: There is an element *e* in *G* such that *a* • *e* = *e* • *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* • *a′* = *a′* • *a* = *e*

  (A5) Commutative: *a* • *b* = *b* • *a* for all *a*, *b* in *G*

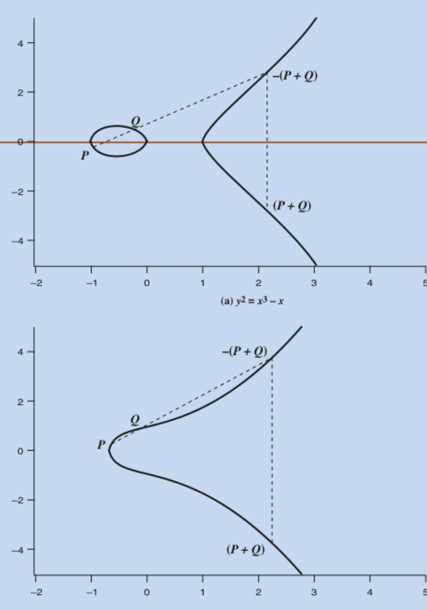


Figure 10.4 Example of Elliptic Curves

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:

Binary curves over GF(2^m)

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

Prime curves over Z_p

- 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

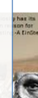
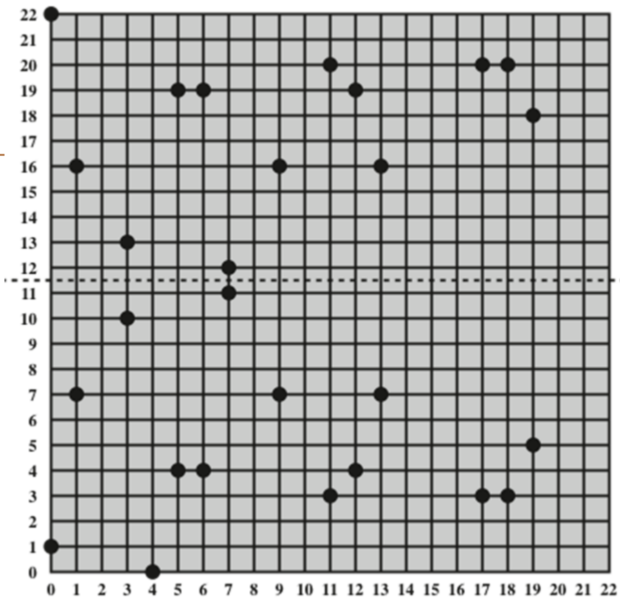
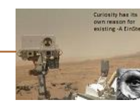


Table 10.1

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

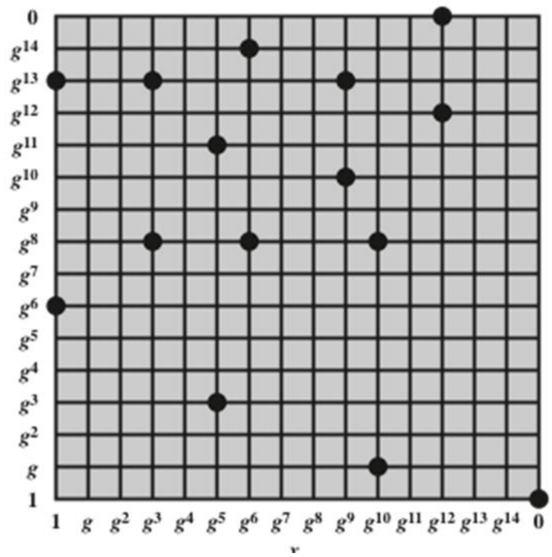
(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)





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)



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

- Certicom example: $E_{23}(9,17)$

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

ECC Encryption/Decryption

- Several approaches using elliptic curves have been analyzed
- Must first encode any message *m* as a point on the elliptic curve P_m
- Select suitable curve and point *G* as in Diffie-Hellman
- Each user chooses a private key n_A and generates a public key $P_A = n_A * G$
- To encrypt and send message P_m to B, A chooses a random positive integer *k* and produces the ciphertext C_m *consisting of the pair of points:*

$$C_m = \{kG, P_m + kP_B\}$$

- To decrypt the ciphertext, B multiplies the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG) = P_m$$

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

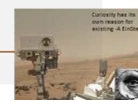


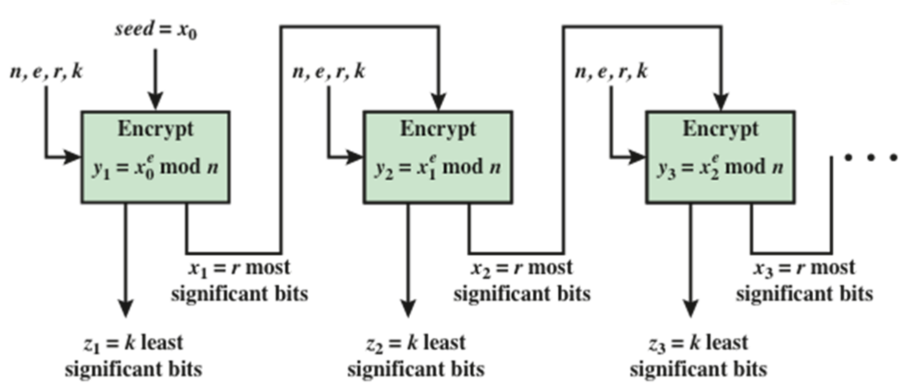
Table 10.3
Comparable Key Sizes in Terms of Computational Effort for Cryptanalysis (NIST SP-800-57)

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

Pseudorandom Number Generation (PRNG) Based on Asymmetric Cipher

- An asymmetric encryption algorithm produces apparently ransom output and can be used to build a PRNG
- Much slower than symmetric algorithms so they're not used to generate open-ended PRNG bit streams
- Useful for creating a pseudorandom function (PRF) for generating a short pseudorandom bit sequence



PRNG Based on Elliptic Curve Cryptography

- Developed by the U.S. National Security Agency (NSA)
- Known as dual elliptic curve PRNG (DEC PRNG)
- Recommended in NIST SP 800-90, the ANSI standard X9.82, and the ISO standard 18031
- Has been some controversy regarding both the security and efficiency of this algorithm compared to other alternatives
  - The only motivation for its use would be that it is used in a system that already implements ECC but does not implement any other symmetric, asymmetric, or hash cryptographic algorithm that could be used to build a PRNG

Summary

- Diffie-Hellman Key Exchange
  - The algorithm
  - Key exchange protocols
  - Man-in-the-middle attack
- Elgamal cryptographic system
- Elliptic curve cryptography
  - Analog of Diffie-Hellman key exchange
  - Elliptic curve encryption/decryption
  - Security of elliptic curve cryptography
- Elliptic curve arithmetic
  - Abelian groups
  - Elliptic curves over real numbers
  - Elliptic curves over Z_p
  - Elliptic curves over $GF(2^m)$
- Pseudorandom number generation based on an asymmetric cipher
  - PRNG based on RSA
  - PRNG based on elliptic curve cryptography