Security – Keys, Digital Signatures and Certificates I

CS3524 Distributed Systems and Security

Lecture 19

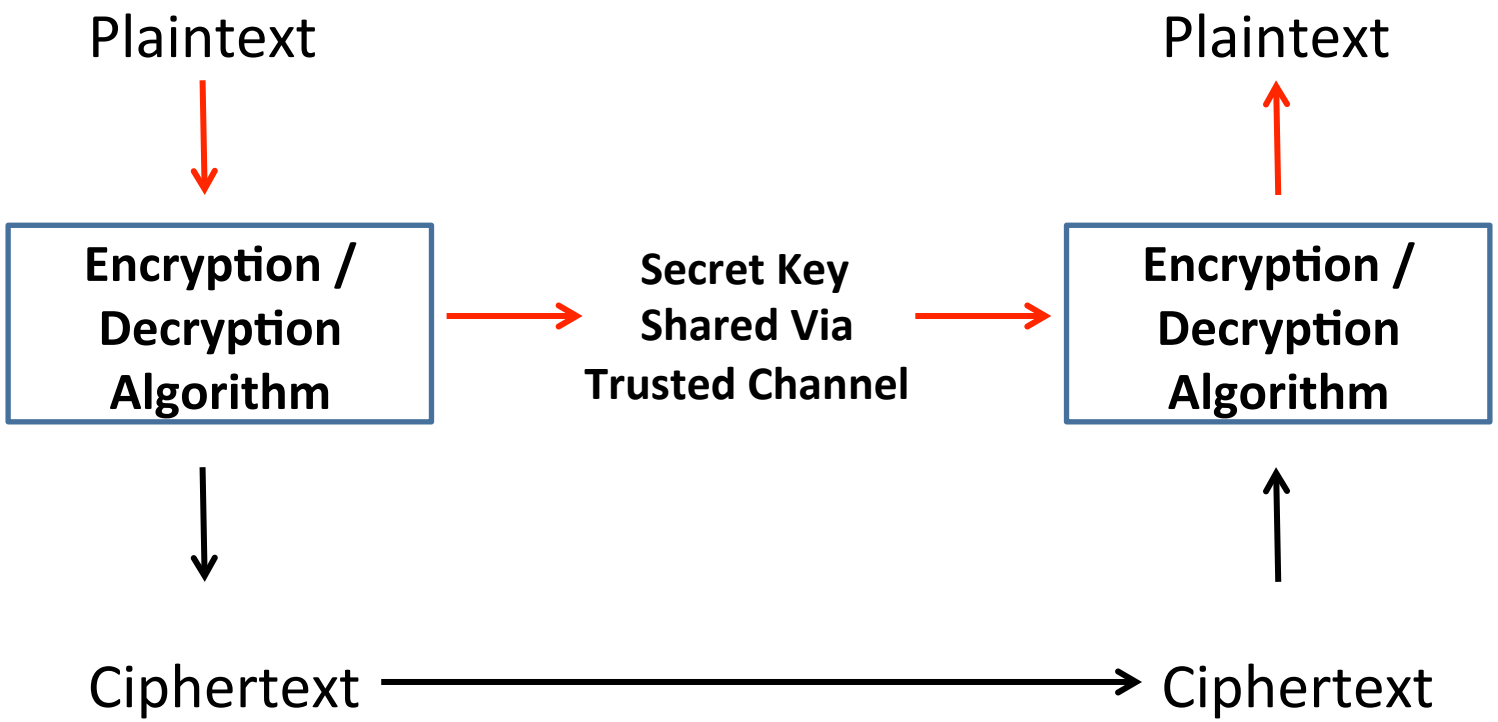
Modern Cryptography

- Relates to
  - Communication: encrypt / decrypt with key
    - Symmetric: use a secret key for both encryption and decryption
    - Asymmetric: use different keys for encryption and decryption, usually a public key and a private key
  - Authentication, Data Integrity: encrypt for creating a unique “fingerprint” or “message digest” for a digital object (e.g. a message or files)
    - digital signatures, certificates

Modern Cryptography

- Classic ciphers encrypted written text messages
- Modern ciphers operate on the bit sequences representing digital objects to be transmitted ("plain text") to produce an encoded result ("cipher text")
  - Block ciphers: encoding the plain text (sequence of bits) takes place in a block-wise fashion
  - Stream ciphers: bit-wise encoding of a stream of plain text with a key
- Goal for a high-quality cipher
  - Fast and resource-efficient in encoding
  - Breaking the cipher would require an effort many magnitudes larger , making cryptanalysis impractical

Symmetric-Key Encryption
(Secret Key Encryption)



- Same key is used by sender and receiver, has to be shared via some trusted channel

Secret Key Encryption Concepts

- Plaintext:
  - Original message or data, input to encryption algorithm
- Encryption algorithm:
  - Performs **substitutions** and **transformations** on the plaintext
- Secret key:
  - The exact substitutions and transformations performed depend on the key, is also an input to the encryption algorithm
- Ciphertext:
  - Encrypted message produced as output, depends on the plaintext and the key
- Decryption algorithm
  - Encryption algorithm run in reverse, takes ciphertext and key as input, produces plaintext as output

Secret Key Cryptography

- Relies on a symmetric key for encryption and decryption
- Encryption algorithm should be hard to break
  - Attackers should be unable to decrypt ciphertext or discover the key (even if they have a set of corresponding cipher / plain texts)
- The larger the key size, the harder to attack
- Sender and receiver must obtain copies of the secret key in a secure fashion
- As long as the key is kept secret, the cryptographic procedure does not have to be secret

Block Ciphers

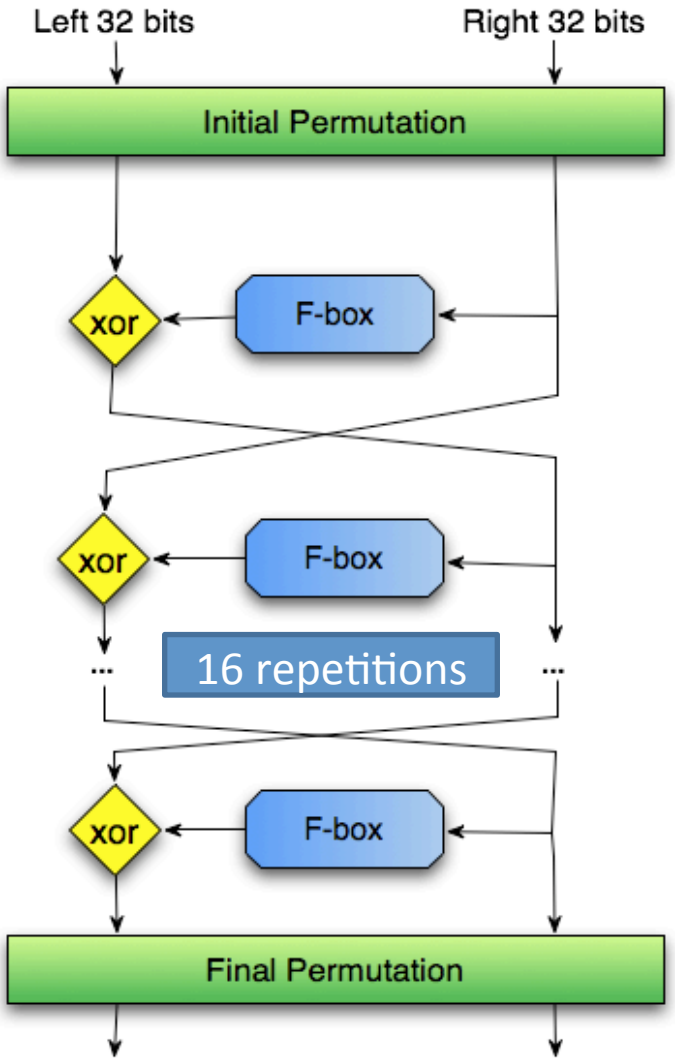
- Message is broken into blocks (usually 16 or 32bit words), each block is encrypted separately
- Operate with a fixed transformation procedure on large blocks of plaintext data
- Block Ciphers
  - Feistel Cipher
    - Is the first block cipher, which inspired subsequent cipher methods such as DES, etc.
  - Substitution-Permutation (SP) network

Symmetric-key Encryption

- Encryption methods
  - DES (Data Encryption Standard)
    - Was highly influential in modern cryptography
    - Developed by IBM
    - Is now considered unsecure due to its key size (56 bits)
  - 3DES
    - Is a form of DES that is regarded as secure, uses three keys and three executions of the DES algorithm
  - AES (Advanced Encryption Standard)
    - Has superseded DES
  - Cast-128
    - Uses fixed, carefully designed S-boxes for subkey generation
  - RC5
    - Fast, variable number of rounds, variable key length
  - IDEA
    - Uses a 128-bit key, does not use S-boxes, but a combination of three operations: XOR, binary addition of 16-bit integer, binary multiplication of 16-bit integers; IDEA is used in PGP (Pretty Good Privacy)
  - Blowfish
    - High execution speed, small footprint, variable keys up to 448 bits, uses S-boxes and XOR, S-boxes are not fixed but dynamically created from key
    - Very strong crypto-algorithm, no practical weaknesses found so far

DES Data Encryption Standard

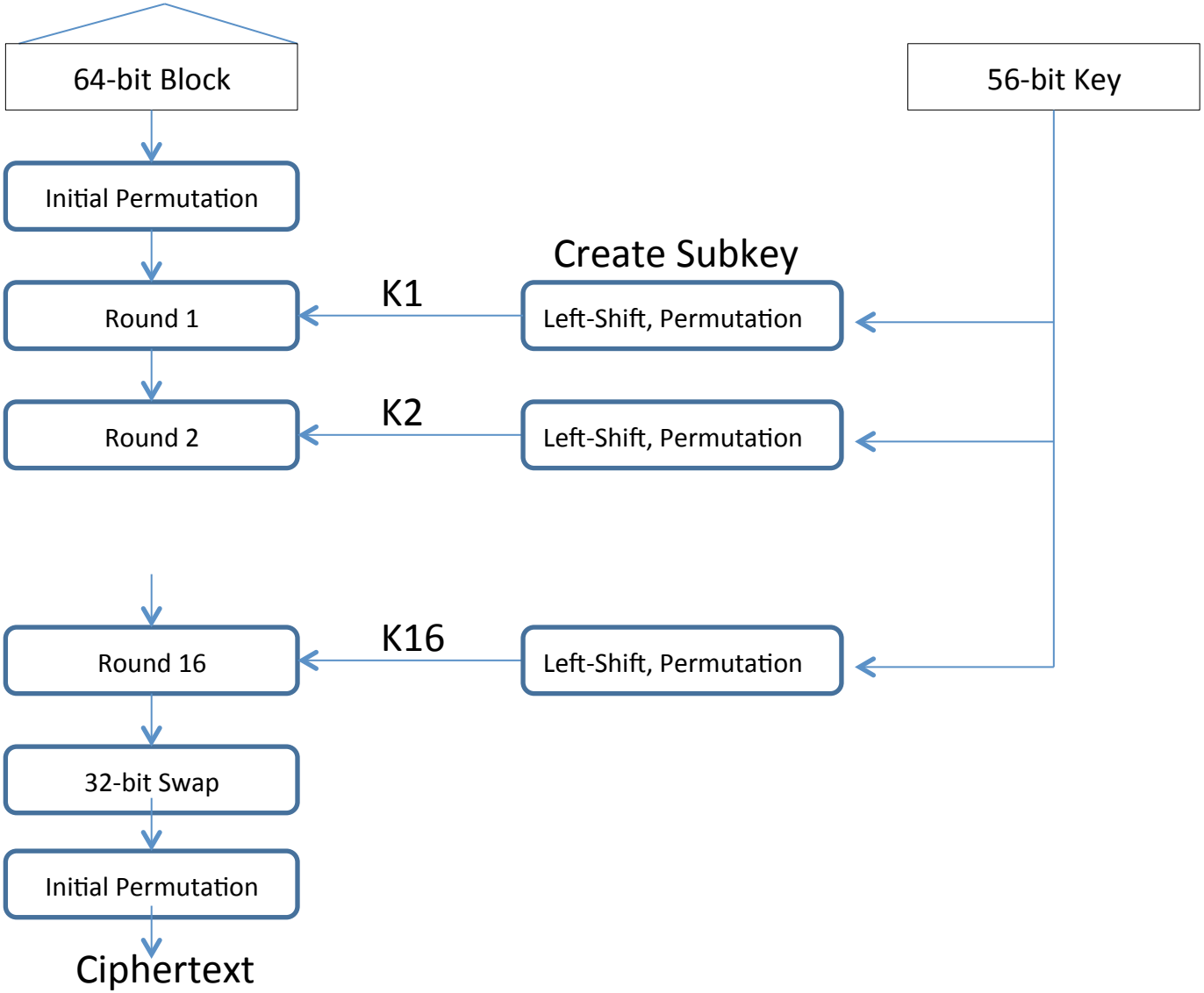
Plaintext 0100101100111010101000101110011001 . . .



Ciphertext

DES Data Encryption Standard

Plaintext 0100101100111010101000101110011001 . . .



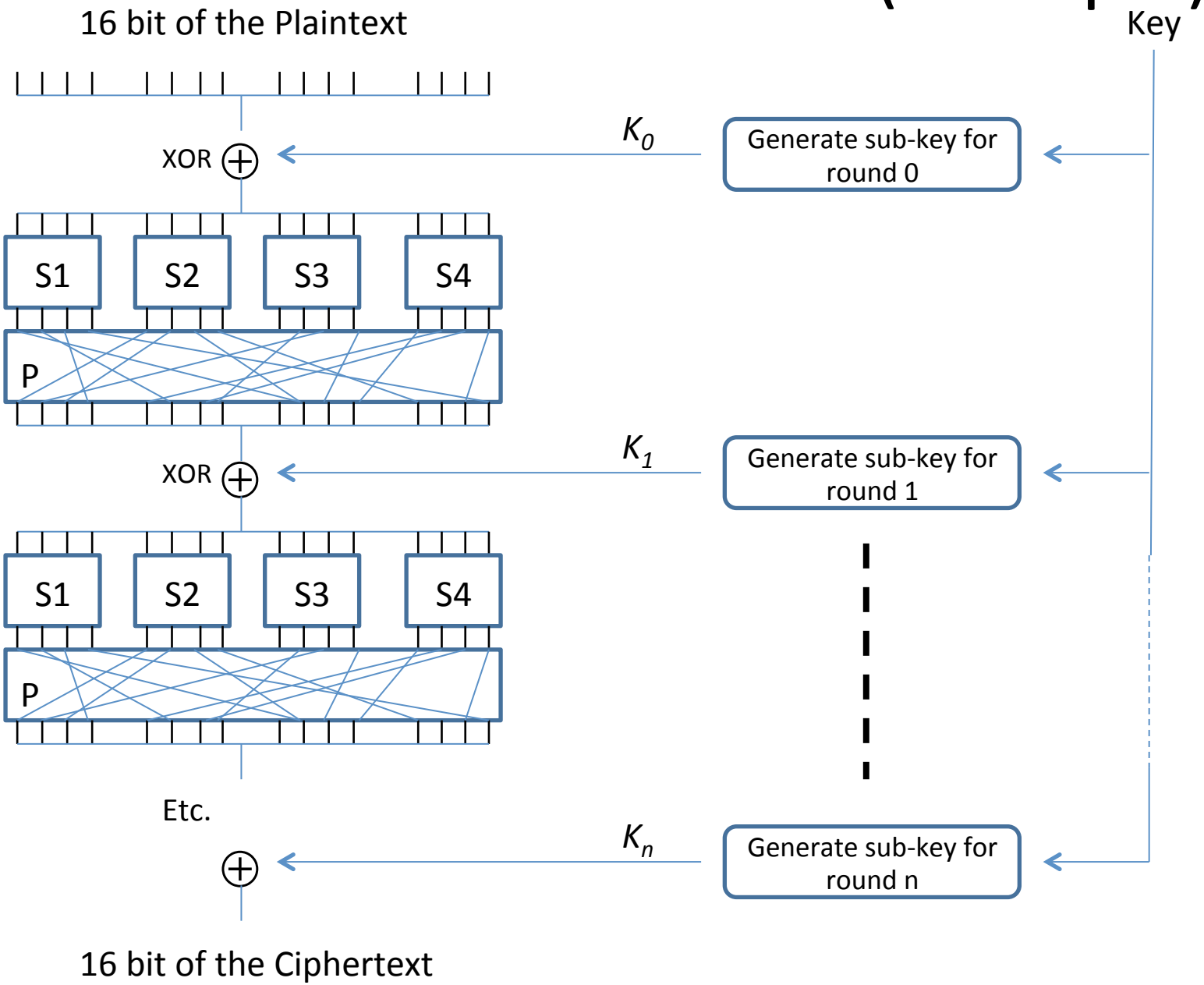
Shannon's Principles for Cryptography

- Shannon's principles of "confusion" and "diffusion"
  - Problems: non-uniformity of individual letters in plain text (e.g. Letter "E" occurs most often in English text) should not be reflected in ciphertext or hard to derive
- Confusion:
  - Make the relationship between plaintext and ciphertext (their statistical features in terms of non-uniformity) too complex to be exploited by an attacker
- Diffusion:
  - Output bits in ciphertext should depend on input bits in plaintext in a very complicated way – a change of one input bit should change at least half of all output bits an unpredictable / pseudorandom manner , (and not just one bit of it)

Substitution-Permutation Network

- SP-Networks (SPN) describe a series of substitution and permutation operations to be applied on plain text
  - The plaintext is separated into blocks (16bit words)
- The encoding operates over a sequence of rounds (“layers”), reapplying substitution and permutation operations over and over again to the output of a previous round

SP-Network (Example)

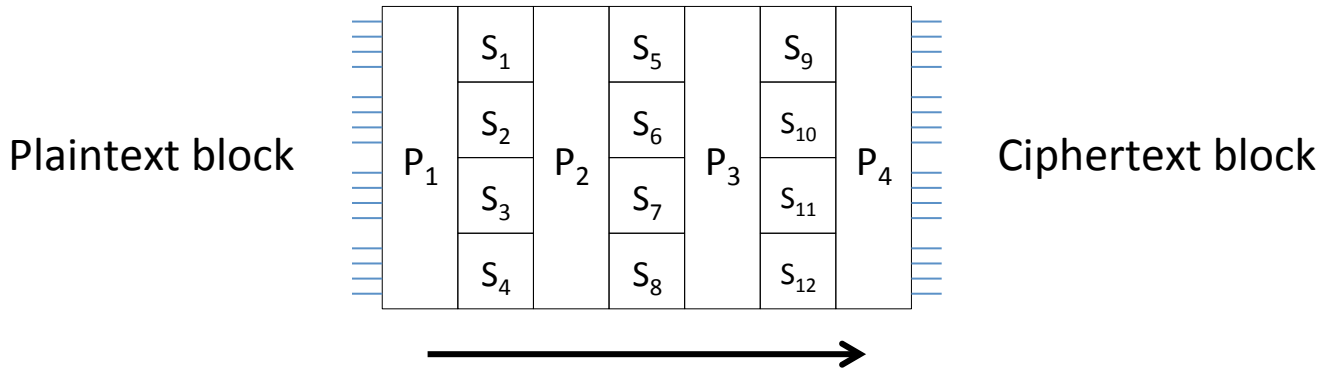


Substitution and Permutation

- Substitution in an SP-Network is performed with a so-called "substitution box" or S-box:
  - Are used to obscure or "confuse" the relationship between key and ciphertext
  - Takes as input *m* bits, produces a corresponding output of *n* bits
  - Implemented as an *m* x *n* lookup table
  - Substitution table is carefully designed to reduce vulnerability (Shannon: a change of one input bit should change at least half of all output bits)
- Permutation is performed with a so-called "permutation box" or P-box:
  - Takes the output of all substitution boxes as its input
  - Re-orders (permutes) bits to produce output

Substitution and Permutation

- Substitution and Permutation are called the "mixing transformations" by Shannon
  - S-box: provides "confusion" of input bits
  - P-box: provides "diffusion" across S-box inputs



AES Advanced Encryption Standard

- Based on the Rijndael algorithm
- Uses a SP-network
  - Fixed block size of 128 bits
  - Supports key sizes of 128, 192 and 256 bits
- Is fast in encryption / decryption both in software and hardware implementations
- Is the default encryption standard for the US government
- Probably cannot be cracked with current technology

Discussion of Secret-Key Cryptography

- Advantages
  - High rates of data throughput, with hardware solutions up to hundreds of megabytes per second
  - Key length is relatively short
  - Symmetric-key ciphers can also be combined to produce stronger ciphers
- Disadvantage
  - The key must remain secret at both ends
  - Cryptographic practice leads to frequent key change
  - In large networks, many key pairs have to be managed

Robustness

- Average time required for exhaustive key search:

Key Size (in bits)	Number of Alternative Keys	Time required at 10^6 Decryptions/μs
32	$2^{32} = 4.3 \times 10^9$	2.15 milliseconds
56	$2^{56} = 7.2 \times 10^{16}$	10 hours
128	$2^{128} = 3.4 \times 10^{38}$	5.4×10^{18} years
168	$2^{168} = 3.7 \times 10^{50}$	5.8×10^{30} years

Key Distribution

- With any symmetric algorithm, the key must be agreed upon by sender and receiver in a secure way
- Before 1976, key exchange was by far the biggest problem in secure communications
- Possible Strategies:
  - A key could be selected by A and physically delivered to B
  - A third party could select the key and physically deliver it to A and B
  - If A and B have previously used a key, one party could transmit the new key by encrypting it with the old key
  - If both A and B have an encrypted connection with a third party C, C could deliver a key on the encrypted links to A and B

Diffie-Hellman Key Exchange

- Developed in 1976, is a key exchange method where two parties exchange information that allows them to derive the same key, but never actually exchange the key
- Method:
  - Two parties, Alice and Bob, agree on a large prime number *p* and a small integer *g*; these two numbers are public
  - Alice picks a secret large random integer *a*, and calculates a number A:

$$A = g^a \bmod p$$

  - A becomes a public key, Alice transmits A to Bob
  - Bob picks a secret large random integer *b*, and calculates a number B:

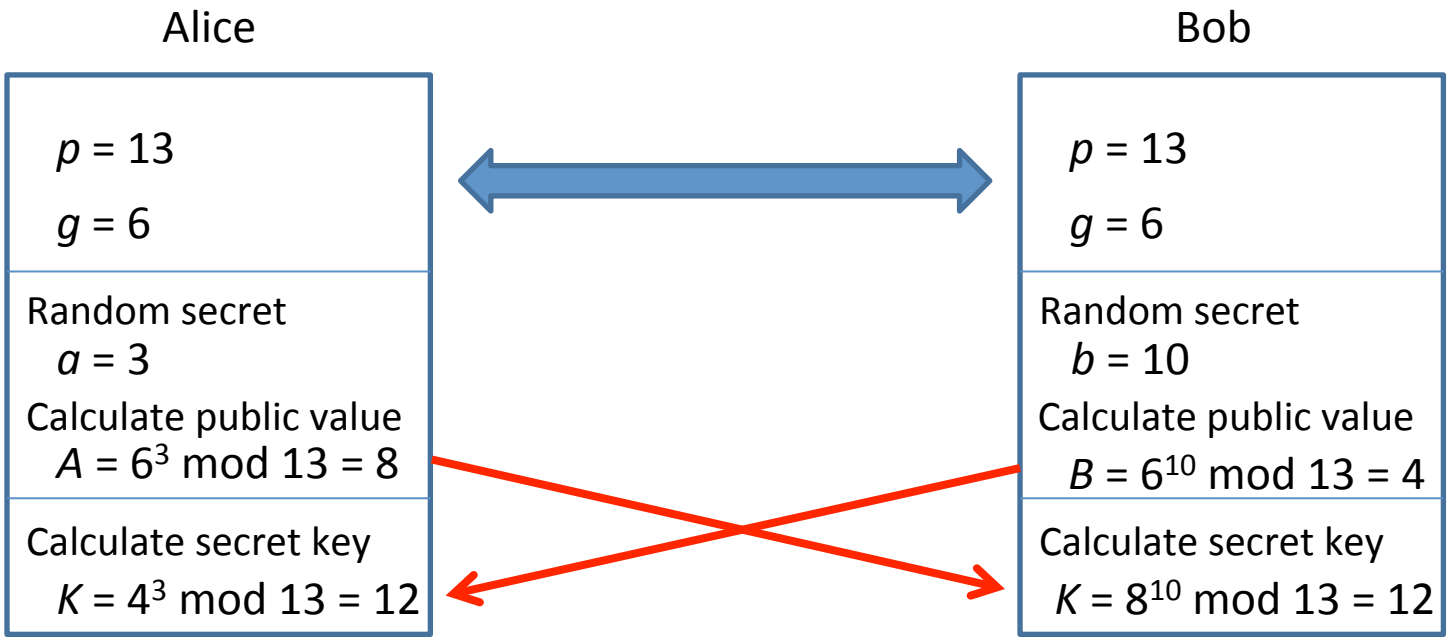
$$B = g^b \bmod p$$

  - B becomes a public key, Bob transmits B to Alice
  - Alice computes the secret key: $K_{Alice} = B^a \bmod p$
  - Bob computes the secret key: $K_{Bob} = A^b \bmod p$

- Rules
  - p must be a prime number, $\boldsymbol{p > 2}$
  - g must be a small integer, $\boldsymbol{g < p}$
  - a and b are large random integers, $\boldsymbol{a < p-1}$, $\boldsymbol{b < p-1}$

Diffie-Hellman Key Exchange

- Example



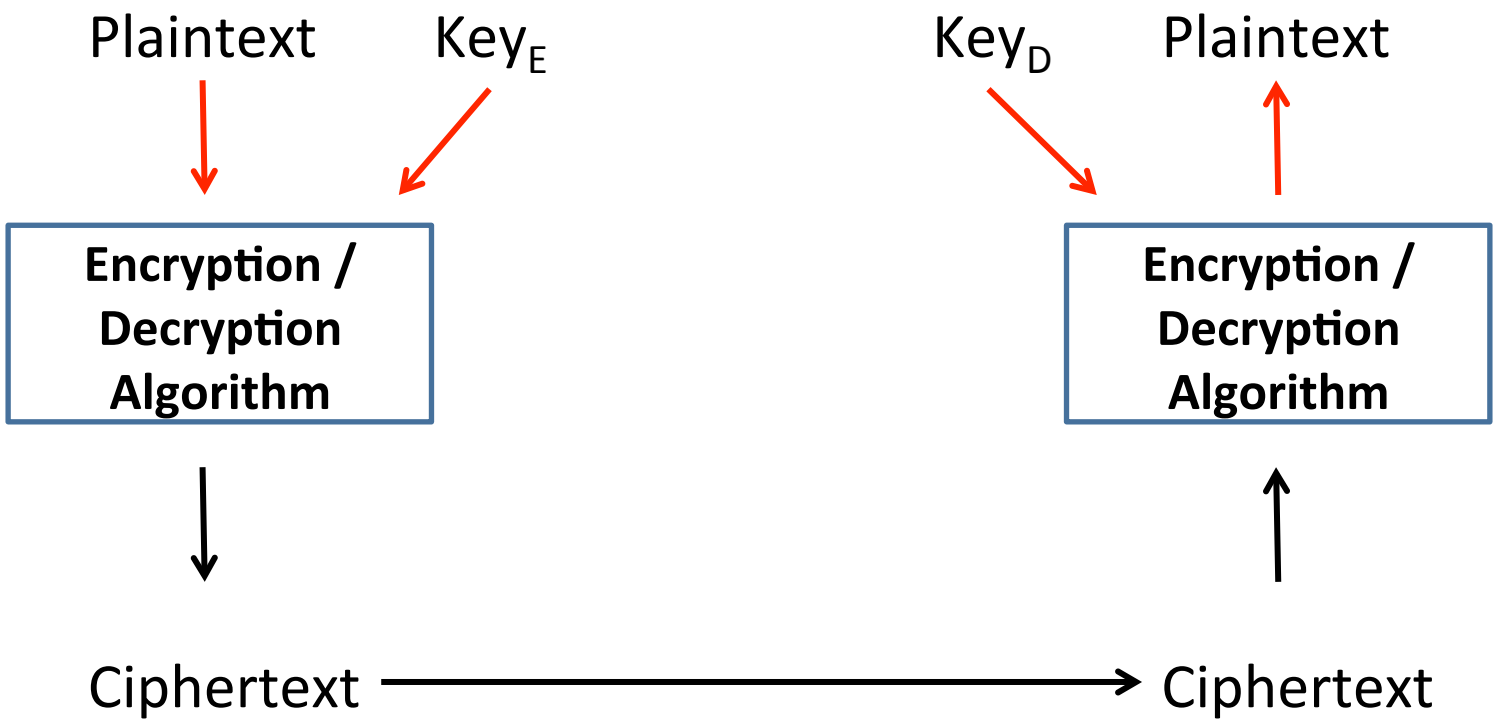
Key Distribution

- Session key
  - Data encryption with a one-time session key, is destroyed at the conclusion of a session
- Permanent key
  - Used between entities for the purpose of distributing session keys

Public Key Cryptography

- Using one secret (private) key poses a security risk
- A solution to this problem is “Public Key Cryptography”
- Two keys: public and private (secret) key
- The keys are matched so that
  - A message encrypted with the public key can be decrypted using the private key
  - A message encrypted with the private key can be decrypted using the public key

Public-Key Encryption
(Asymmetric Key Encryption)



- Two different keys: public key, private key

Applications for Public Key Cryptography

- Encryption / Decryption
  - The sender encrypts the message with the recipients public key
- Digital Signature
  - The sender “signs” a message with its private key. For this, a cryptographic algorithm is applied to the whole message or to a small block of data that is a function of the message (a “fingerprint” of the message, called a message “digest”)
- Key Exchange
  - Exchange key information using the private key of one or both parties

Public-Key Cryptography

- Privacy:
  - Encryption with public key, decryption with private key
    - Anyone can send a message, using the public key of the receiver
      - no one else can read the message, because only the private key can decrypt the message
    - Only the owner of the private key can decrypt the message

Public-Key Cryptography

- Authenticity
  - Encryption with the private key, decryption with the public key
    - Receivers of a message can verify who sent the message with the sender's public key
    - Only the owner of the private key can have generated such an encrypted message

Public-Key Cryptography-Requirements

- Easy to generate a public key / private key pair
- Easy for a sender to generate ciphertext using the public key
- Easy for the receiver to decrypt ciphertext using the private key
- Computationally infeasible to determine the private key, knowing the public key
- Computationally infeasible to recover the message without the private key, knowing the public key and ciphertext
- Either of the two keys can be used for encryption, with the other used for decryption

RSA Public-Key Cryptosystem

- RSA (Rivest, Shamnir, Adleman, 1977): best known, regarded as the most practical public-key scheme
  - Used for encrypting messages, key exchange and creating digital signatures
  - Is a block cipher
  - Plaintext and Ciphertext are represented as integers in the range of [0 .. n-1] for some n

RSA

- Encryption:
  - A ciphertext block *C* is the result of encryption of a plaintext block *M*, using the publicly known numbers *e* and *n*

$$C = M^e \bmod n$$

- Decryption
  - A plaintext block *M* is the result of decryption of a ciphertext block *C*, using the secret number *d*

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

RSA Key Generation

- To do:
  - Public Key: both sender and receiver must know the values of *n* and *e*
    - Calculate number *n* (maximum possible value of a plaintext / ciphertext block)
    - Calculate number *e* (a value needed for *e*ncryption)
  - Private Key
    - Calculate number *d* (a value needed for *d*ecryption)
    - only the receiver knows the value of *d*

RSA Key Generation

- Calculate *n*
  - Select two large prime numbers *p* and *q*, these are secret
  - Calculate: $n = p \times q$
- Calculate the public *e*
  - *e* is "relatively prime" to the Euler Totient $\phi(n)$
  - $e < \phi(n)$
- *What is*
  - "relatively prime" ?
  - Euler Totient $\phi(n)$?

RSA

- Relatively prime numbers:
  - Two integers n and m are relatively prime, if their greatest common divisor is 1: gcd(n,m) = 1
    - n and m do not share any common positive prime factors (divisors) except 1
- Euler Totient $\phi(n)$
  - Is the ***number*** of positive integers that are ≤ *n* and that are relatively prime to *n*
    - E.g.: n = 10, {1,3,7,9} is the set of positive integers relative prime to 10, therefore: $\phi(n) = 4$
  - If *n* is the product of two prime numbers, *p* and *q*, then $\phi(n) = (p-1)(q-1)$
    - E.g.:
      - p = 3, q = 5, n = p x q = 3 x 5 = 15, therefore: $\phi(n) = (p-1)(q-1) = 2 \times 4 = 8$
      - n = 15, {1,2,4,6,7,8,11,13}

RSA Key Generation

- Calculate *n*
  - Calculate: $n = p \times q$, *p* and *q* are two large prime numbers
- Calculate the public *e*
  - We know:
    - If $n = p \times q$, *p* and q are prime numbers, then $\phi(n) = (p-1)(q-1)$
  - Choose *e*:
    - e is relatively prime to (p-1)(q-1) and $1 < e < (p-1)(q-1)$
- Calculate the private key *d*
  - $d = e^{-1} \bmod (p-1)(q-1)$
- Result
  - Public key $K_{PUB} = \{e,n\}$
  - Private key $K_{PRIV} = \{d,n\}$

- *Encryption: encrypt a plaintext M to generate a ciphertext C via* $\boldsymbol{C = M^e \bmod n}$
- *Decryption: decrypt a ciphertext C to generate a plaintext M via* $\boldsymbol{M = C^d \bmod n}$

RSA Example

- Select two prime numbers, $p = 11$, $q = 3$
- Calculate $n = pq = 11 \times 3 = 33$
- Calculate $(p-1)(q-1) = 10 \times 2 = 20$
- Select e such that e is relatively prime to $(p-1)(q-1) = 20$ and $e < 20$
  - We select $e = 3$
- Calculate d such that $de = 1 \bmod 20$ and $d < 20$
  - That means: $3d = 1 \bmod 20$
  - Result: $d = 7$, because $(7 \times 3)/20$ has the remainder 1 (that's the same as: $77x5 = 21 = 2 \times 20 + 1$, or $21 \equiv 1 \bmod 20$)
- Keys
  - Public Key = {3, 33}
  - Private Key = {7, 33}
- Encrypt
  - Plaintext $M = 5$
  - Calculate ciphertext: $C = 5^3 \bmod 33 = 125 \bmod 33 = 26$
- Decrypt
  - Ciphertext = 26
  - Calculate plaintext $M = 26^7 \bmod 33 = 5$

RSA Cryptanalysis

- Brute force
  - Attack: try all possible private keys
  - Defence: the larger the number of bits for encoding *e* and *d*, the more secure this form of encoding will be
  - Problem: large keys slow down calculations
- Factoring *n*:
  - Vulnerability comes from the number *n*: If the number *n* is calculated from two large prime numbers *p* and *q*, finding these two secret prime factors by factoring *n* would allow us to calculate all elements of the RSA
  - Factoring *n* is a hard problem
  - We can factor a 512-bit number with 100s of CPU years (available to NSA, not random hacker)
  - We cannot factor 1024-bit numbers with present technology and math