Contents

What is Network Security?

- Confidentiality
  - Only sender, intended receiver should "understand" message contents
    * Sender encrypts message
    * Receiver decrypts message
- Authentication
- Message Integrity

Friends and Enemies

- Bob and Alice want to communicate "securely"

Could be

- Real life Bob and Alice
- Web browser/server for electronic transactions
- On-line banking client/server
- DNS servers
- Routers exchanging routing table updates

What can bad guys do?

- Passive attack
- Active attack
  - Actively insert messages into connection
- Impersonation
  - Fake (spoof) source address in packet (or any field in packet)
- Hijacking
  - "Take over" ongoing connection by removing sender or receiver, inserting himself in place

Cryptography

- Original data to be transferred is called Plaintext or Cleartext
  - Encrypted version is called Ciphertext
- Plaintext is denoted `P`, whereas Ciphertext is denoted `C`
  - Encryption function `E` operates on `P` to produce `C`
- In the reverse process
  - Decryption function `D` operates on `C` to produce `P`
- $D(E(P)) = P$ must also hold true for the cryptosystem to function correctly

Cryptographic Keys

- All modern encryption algorithms use a key denoted by `K`
- The key can take on many possible values

The encryption and decryption functions now become $\mathrm{E_K}$ and $\mathrm{D_K}$.

Substitution Ciphers

- Each letter of a group of letters is replaced by another letter or group of letters to disguise it
- Caesar Cipher - Mono-alphabetical Substitution

```
a b c d e f g h i j k l m n o p q r s t u v w x y z
D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

- To send a secret message
  - Letters of the message are taken one by one and the letters appearing below are written instead
- The message "send spears" would be enciphered as "VHQG VSHDUV"

Attacks

- Identify commonly occurring characters
- Commonly occurring bigrams/digrams
- Domain specific buzz words

Substitution ciphers preserve the order of the text symbols but disguise them

The Vigenère Cipher

- Some protection from the above can be gained by using a poly-alphabetical cipher

A: `A B C D E F G H I J K L M N O P Q R S T U V W X Y Z`
B: `B C D E F G H I J K L M N O P Q R S T U V W X Y Z A`
C: `C D E F G H I J K L M N O P Q R S T U V W X Y Z A B`
⋮
Z: `Z A B C D E F G H I J K L M N O P Q R S T U V W X Y`

- Pick a key and repeat it for the length of the plaintext
- To encode a letter, go to its row and across by the index of the current letter in the key

Example

- Plaintext: ATTACKATDAWN
- Key: LEMON → LEMONLEMONLE
- Ciphertext: LXFOPVEFRNHR

Transposition Ciphers

- Transposition ciphers reorder the symbols
- Plaintext is written horizontally in rows
- Ciphertext is read out in columns
  - Starting with the column whose key is the lowest

Example:

M (7)	E (4)	G (5)	A (1)	B (2)	U (8)	C (3)	K (6)
p	l	e	a	s	e	t	r
a	n	s	f	e	r	o	n
e	m	i	l	l	i	o	n
d	o	l	l	a	r	s	t
o	m	y	s	w	i	s	s
b	a	n	k	a	c	c	o
u	n	t	s	i	x	t	w
o	t	w	o	a	b	c	d

- Plaintext: pleasetransferonemilliondollarstomyswissbankaccountsixtwotwo
- Ciphertext: AFFLSKSOSELAWAIATOOSSCTCLNMOMANTESILYN-TWRNNTSOWDPAEDOBUOERIRICXB

Symmetric-Key Encryption

- Based on the sender and the receiver of a message knowing and using the key
- Sender uses the secret key to encrypt the message
  - Receiver uses the same secret key to decrypt the message

Key Management

- Main problem is getting the sender and receiver to agree on a secret key without anyone else finding out
- Key management is one of the fundamental issues that have to be addressed in symmetric key cryptosystems
- Examples of symmetric key algorithms are
  - DES, Triple DES, IDEA, AES

Data Encryption Standard (DES)

- In January 1977 a standard encryption method was adopted by the U.S gov
  - Origins lie in an internal IBM project codenamed Lucifer
- Though the algorithm used is complex
  - It is easily implemented in hardware
  - Software implementations are also widely available
- DES is a Block Cipher
  - Operates on a single chunk of data at a time
    - 64 bits (8 bytes)
- The key length is 56 bits
  - Often expressed as a 8 character string with the extra bits used as a parity check
- Algorithm as 19 distinct stages
- First stage re-orders the bits of the 64 bit input block by applying a fixed permutation (P-box)

- Last stage is the exact inverse of this permutation
- Penultimate stage exchanges the leftmost 32 bits with the rightmost 32 bits
- Remaining 16 stages are called Rounds
  - functionally identical but take as an input a quantity computed from the key and the round number

Cracking DES

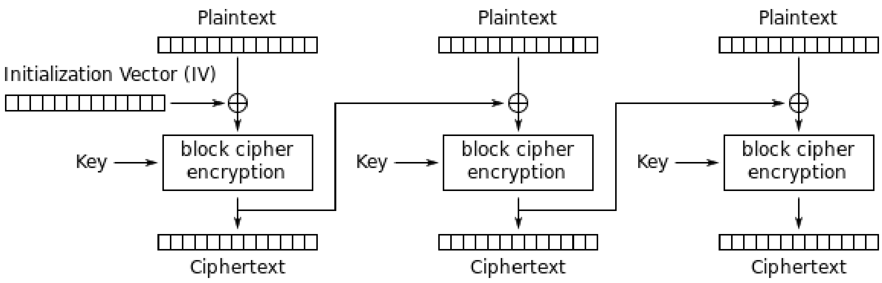
- 56 bits is a short key
- Brute force attack (try every key)
  - Special chips can check 4 million keys/sec
  - $1 million DES cracking machine could break it in a few hours
- Jan '99 Challenge 111 won in 22 hrs and 15 mins using a supercomputer and 100,000 Internet nodes
  - Tested 245 billion keys per second
- Improvement - Triple DES or 3DES
  - Uses EDE or DED mode

Modes of Operation for Block Ciphers

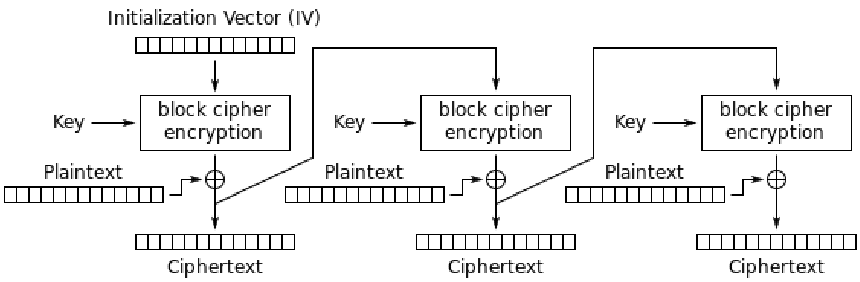
- In the previous discussion of DES known as Electronic Code Book (ECB) mode
  - Each 64-bit block is encoded independently
- Still allows for a passive intruder to replicate the information
  - e.g. Repeat request for withdrawal of $1bn
  - Person knows last block of encrypted spreadsheet is bonus
  - Can swap blocks around

Cipher Block Chaining Mode (CBC)

- In CBC mode each block of plaintext is XORed with previous ciphertext block before being encrypted
- Each ciphertext block depends on all plaintext blocks processed up to that point



Cipher Block Chaining (CBC) mode encryption

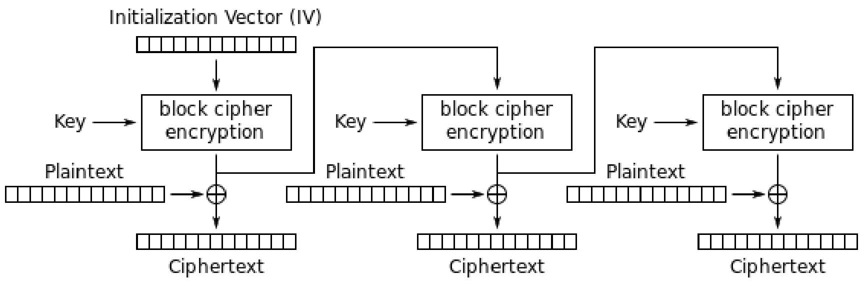


Cipher Feedback (CFB) mode encryption

Cipher Feedback Mode (CFB)

- The CFB mode is a close relative of CBC
  - Makes a block cipher into a self-synchronising "Stream Cipher"
- Used for encrypted characters at a time instead of whole blocks

Output Feedback Mode (OFB)



Output Feedback (OFB) mode encryption

- OFB mode generates a keystream of blocks which are then XORed with the plaintext blocks to get the ciphertext

The Varnam Cipher

- Simplest and most secure stream cipher is called the One-time Pad
- Chooses a key stream (k) that is randomly chosen for each encipherment - makes use of XOR operator
  - `k >= p`

Advanced Encryption Standard (AES)

- In 1997 NIST announced a call for proposals to develop a new Advanced Encryption Standard
  - After a long vetting process five algorithms were short listed
  - Rijndael (Rhine-doll) was eventually chosen as the new AES
- Symmetric cipher with variable key and block sizes of 128, 192 and 256 bits
  - Most common mode is 128 bit key and block size

  - Support for fast encryption and decryption in s/w - 700 Mbps
    * Can be implemented efficiently in smartcards

- A device that could have a 10^{18} AES keys/s would in theory require about 3×10^{51} years to exhaust the 256-bit key space
- Brute force decryption taking 1 sec on DES, takes 149 trillion years for AES
- The cipher consists of between 10 or 14 rounds (N_r)
  - Depending on the key length ($\mathtt{N}_k$) and the block length ($\mathtt{N}_b$)
- A plaintext block `X` undergoes `n` rounds of operations to produce an output block `Y`
  - Each operation is based on the value of the $\mathtt{n}^{th}$ round key
- Round keys are derived from the Cipher key by first expanding the key
  - Then selecting parts of te expanded key for each round

Asymmetric-Key Cryptosystems

- Public key cryptography was discovered by Diffie and Hellman in 1976
  - Solves the key management problem associated with symmetric key cryptosystems
- In public key cryptography each person generates a pair of keys
  - The *public key* and the *private key*
- Public key is published and widely distributed
  - While the private key is kept secret
- Examples of public-key cryptographic algorithms are
  - RSA, Diffie-Hellman, ElGamal, ECC

Properties of Asymmetric-Key Cryptosystems

- Must be computationally easy to encipher or decipher a message given the appropriate key
- Must be computationally infeasible to derive the private key from the public key
- Need for exchanging secret keys is eliminated
  - All secure communications now only involves public keys

Public-Key Cryptography

- Each user in a public-key system selects their own private key (K^-) and their own public key (K^+)
- Alice wants to send an encrypted message to Bob
  - Encrypts message with Bob's public key
    * She looks up his public key ($K^+{}_b$) in a public directory
  - Sent over an open channel but can only be decrypted with the private key
  - Bob decrypts message with the private key

RSA

- De-facto standard algorithm for implementing asymmetric-key cryptography (although moving towards ECC)
  - Named after Rivest, Shamir and Adleman who developed it in 1978 at MIT
- Its security is based on the difficulty of factoring very large numbers
  - One-way "Trapdoor Function"
- Example: prime factoring
  - Easy to calculate product of 2 large prime numbers
  - Difficult to calculate the prime factors from product

Modular Arithmetic

- Most number sets we are used to are infinite, e.g. real numbers
  - However most cryptographic algorithms are based on arithmetic fine a *finite* set of number
- Consider the hours of a clock
  - 1h, 2h, 3h, …, 11h, 12h, 1h, 2h, 3h, …, 11h, 12h, 1h, 2h, 3h, …
- **Modulo Operation**
  - Let $a, r, m \in \mathbb{Z}$ (where $\mathbb{Z}$ is the set of all integers) and $m > 0$
    * $a \equiv r \bmod m$
  - a is said to be congruent to r mod m if m divides $a - r$
    * m is called the modulus and r is called the remainder
- It is always possible to write $a \in \mathbb{Z}$ such that

- $a = q \times m + R, 0 \leq r < m$
- Since $a - r = q \times m$ (m divides $a - r$) we can now write
  - $a \equiv r \bmod m$
- Example: Let $a = 42, m = 9$
  - $42 = 4 \times 9 + 6$
- Q: $-11 \equiv x \pmod 7$
  - $-11 \equiv 3 \pmod 7$

Multiplicative Inverse

- The integers modulo n, denoted $\mathbb{Z}_n$ is the set of integers $\{0, 1, 2 \ldots, n-1\}$
  - Addition, subtraction and multiplcation are performed modulo n
    * $\mathbb{Z}_n$ is referred to as an *Integer Ring*
- $\mathbb{Z}_{25} = \{0, 1, 2, \ldots, 24\}$
  - $13 + 16 = 4$
  - $29 \equiv 4 \pmod{25}$
- The multiplicative inverse of a modulo n is an integer $x \in \mathbb{Z}_n$ such that
  - $ax \equiv 1$ (mod n)
- The multiplicative inverse only exists for an element $a \in \mathbb{Z}_n$ iff
  - $\gcd(a, n) = 1$
- Q: Does the multiplicative inverse of 15 exist in $\mathbb{Z}_{26}$?
  - $\gcd(15, 26) = 1 \implies$ *yes*

Extended Euclidean Algorithm (EEA)

- Division of a by b modulo n is a product of a and b^{-1} modulo n
  - $b \mid a$ is equivalent to $a \times b^{-1}$ (mod n)
- Q: What is 4^{-1} modulo 11?
  - $x \equiv 4^{-1} \pmod{11}$
  - $x \times 4 \equiv 1 \pmod{11}$
- The modular multiplicative inverse of a modulo m can be found with the Extended Euclidean Algorithm
  - $a \times r_0 + t \times r_1 = \gcd(r_0, r_1)$
  - $s \times r_0 + t \times r_1 = 1$

- $t = r_1^{-1} \bmod r_0$

- $\gcd(11, 4) = 1$
  - $11 = 2 \times 4 + 3$
  - $4 = 1 \times 3 + 1$
  - $3 = 3 \times 1 + 0$
- Back substitution
  - $1 = 4 - 1 \times 3$
  - $1 = 4 - 1 \times (11 - 2 \times 4)$
  - $1 = 3 \times 4 - 1 \times 11$
  - $1 = (-1) \times 11 + (3) \times 4$
  - $t = 3$
    - $4 \times 3 \equiv 1 \pmod{11}$
- Exercise: Compute $15^{-1} \bmod 26$?
  - $t = 7$

Euler's Totient function $\phi(n)$

- Number of positive integers less than n and relatively prime to n
- Example: $\phi(10) = 4$
  - $1, 3, 7, 9$ are relatively prime to 10
    - Relatively prime means $\gcd(a, b) = 1$
- Example: $\phi(21) = 12$
  - $1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20$ are relatively prime to 21
- Q: What is $\phi(7)$ and $\phi(11)$?
  - $\phi(m) = m - 1$ when m is prime!

Euler's Phi Function

- In general let m have the following canonical factorization

$$m = p_1^{e_1} \times p_2^{e_2} \times \cdots \times p_n^{e_n}$$

- Where p_i are distinct primes and e_i are positive integers

$$\phi(n) = \prod_{i=1}^{n} (p_i^{e_i} - p_i^{e_i - 1})$$

- Exercise: Calculate $\phi(240)$
  - Hint: $m = 240 = 16 \times 15 = 2^4 \times 3 \times 5$
  - $(2^4 - 2^3) \times (3^1 \times 3^0) \times (5^1 - 5^0) = 14$

Fermat's Little Theorem :)

- Let a be an integer and p be a prime, then
  - $a^p \equiv a$ (mod p)
  - $a^{p-1} \equiv 1$ (mod p)
  - $a \times a^{p-2} \equiv 1$ (mod p)
  - $a^{-1} \equiv a^{p-2}$ (mod p)
- Thus we have a way for inverting an integer a modulo a prime
- Compute 4^{-1} mod 11
  - $a = 4$ and $p = 11$;
  - $4^9 = 262144 \equiv 3 \pmod{11}$
- Can also be used for basic primality testing

RSA Algorithm

- Choose two large dictinct primes p and q
- Compute the product (modulus)
  - $n = p \times q$
- Randomly choose an encryption key e, less than n that has no common factors with $\phi(n)$
  - e and $\phi(n)$ are relatively prime
- Finally compute the decryption key, d such that
  - $e \times d \equiv 1 \pmod{\phi(n)}$
  - $d \equiv e^{-1} \pmod{\phi(n)}$
  - $d \equiv e^{-1} \pmod{(p-1)(q-1)}$

RSA Usage

- The numbers e and n are the public key
  - The number d is the private key
- Break the plaintext message into a number of blocks
  - Represent each block as an integer

- Encryption
  - $CipherTextBlock = (PlaintextBlock)^e$ (mod n)
- Decryption
  - $PlaintextBlock = (CiphertextBlock)^d$ (mod n)
- Key Sizes
  - 1024, 2048, 3072, 7680 bits

RSA with Workable Numbers

- Let $p = 3$ and $q = 11$
- Using $n = p \times q = 33$
  - $\phi(n) = (p - 1) \times (q - 1) = 20$
- Choose $e = 3$ (e and $\phi(n)$ have no common factors)
- Solving $e \times d = 1$ mod 20 and $d < 20$
  - $d \equiv e^{-1}$ mod 20

RSA Example

Symbol	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbol
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

An Important Property of RSA

The following property will be very useful later

$$K_B^-(K_B^+(m)) = K_B^+(K_B^-(m))$$

Use public key first, followed by private key = Use private key first, followed by public key

Hybrid Schemes

- Asymmetric-key algorithms are not a replacement for symmetric-key algorithms such as DES or AES
  - Rather than supplement DES or any other fast bulk encryption cipher

1. Encrypt plaintext with key
2. Encrypt key with private key
3. Decrypt key with public key
4. Decrypt ciphertext with key

We can use a public key algorithm to securely transfer a session key

Message Authentication and Integrity

- Bob receives a message from Alice, wants to ensure
  - Message originally came from Alice - *authentication*
  - Message not changed since sent by Alice - *integrity*

Message Digest/Cryptographic Hash

- A message digest is a strong digital fingerprint of a message
- Take input m, produces fixed length value, $H(m)$
  - 128/160/256 bits
- Computationally infeasible to find two different messages, x and y such that $H(x) = H(y)$
- Examples of message digest algorithms are
  - MD2, MD4, MD5, SHA-1 and SHA-2

Digital Signatures

- Cryptographic technique analogous to hand-written signatures
- Sender (Bob) digitally signs the document establishing he is the document owner/creator
- Recipient (Alice) can prove to someone that Bob, and no one else (including Alice) must have signed document

Key Management

- When Alice obtains Bob's public-key (from a website, e-mail, etc.) how does she know it's Bob's public key, not Trudy's?
- Public key cryptography is based on the idea that
  - An individual will generate a key pair
  - Keep one component secret and public the other component (public key)
- Other users on the network
  - Must be able to retrieve this public key and associate the user's identity with it
- One way to form this association is to apply to a 'certificate authority' for a digital certificate

X.509 Certificates

- The TTP will construct a message referred to as a Certificate
- The cert contains a number of fields
  1. Subject (Identity of User)
  2. Public Key
  3. Validity Period
  4. Issuer (Identity of TTP)
  5. Other fields
  6. Signature of TTP
- Assumes that every user in the system is equipped with the public-key of the TTP
  - Allows one to verify the digital signature on the certificate
    * Guaranteeing that the public key is associated with the named user

Certificate Hierarchy

- TTPs (Trusted Third Party) that issue certificates are referred to as certification authorities (CAs)
- The root CA issues certificates only to other CAs

Diffie-Hellman Key Exchange (DHKE)

- A protocol that allows strangers to establish a shared symmetric key without them having to meet
  - Without the need for a cryptosystem to be in place!
- The basic idea behind DHKE is that exponetiation in $\mathbb{Z}_p^*$ (p is a prime) is a one-way function and that exponentation is commutative
- The value $k \equiv (a^x)^y \equiv (a^y)^x \mod p$ is a "joint secret"
  - Can be used as a session key between the two parties
- Securely choose the *domain parameters*
  - Choose a large prime p
  - Choose an integer $\alpha \in \{2, 3, \ldots, p-2\}$
  - Public p and α

Security of DHKE

- The domain parameters p should have a length of 1024 bits (308 digits) or longer
- α should be a *primitive element* or *generator* of the group (G)
  - One whose powers modulo p generate all the integers from 1 to $p-1$

Man-in-the-Middle Attack (MITM)

- Alice computes the key $g^{xz} \mod n$
  - So does Trudy (for messages to Alice)
- Bob computes $g^{yz} \mod n$
  - So does Trudy (for messages to Bob)
- Alice thinks she is talking to Bob and so establishes a session key (with Trudy) and so does Bob
  - Also known as the bucket brigade attack

Groups

- The group operation $\circ$ is closed, i.e. for all $a, b \in G$, it holds that $a \circ b = c \in G$
- The group operation is associative, i.e. $a \circ (b \circ c) = (a \circ b) \circ c$ for all $a, b, c \in G$

- There is an element $1 \in G$, called the neutral element (or identity element), such that $a \circ 1 = 1 \circ a = a$ for all $a \in G$
- For each $a \in G$ there exists an element $a^{-1} \in G$, called the inverse of a, such that $a \circ a^{-1} = a^{-1} \circ a = 1$
- A group G is abelian (or commutative) if, furthermore, $a \circ b = b \circ a$ for all $a, b \in G$
- Example $(\mathbb{Z}, +)$ is a group
  - The set of integers $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$ together with the usual addition forms an abelian group
  - Where $e = 0$ is the identity element and $-a$ is the inverse of an element $a \in \mathbb{Z}$

Field

- In cryptography, we are almost always interested in fields with a *finite number of elements*
  - Which we call finite fields or Galois fields
- A field F is a set of elements with the following properties
  - All elements of F form an *additive group* with the group operation "+" and the neutral element 0
  - All elements of F except 0 form a *multiplicative group* with the group operations "*" and the neutral element 1
  - When the two group operations are mixed, the distributivity law holds
    * i.e., for all $a, b, c \in F : a * (b + c) = (a * b) + (a * c)$

Prime Fields

- The set $\mathbb{Z}_p$ (where p is a prime) is denoted as $GF(p)$ and is referred to as a prime field - aka Galois field with a prime number of elements
- Elements of the field $GF(p)$ can be represented by integers $0, 1, \dots, p-1$
  - All nonzero elements of $GF(p)$ have an inverse
  - Arithmetic in $GF(p)$ is done modulo p
- Two operations of the field are *integer addition* and *integer multiplication*, both *modulo p*
  - e.g. $GF(5) = \{0, 1, 2, 3, 4\}$

Finite and Cyclic Groups

- A group $(G, \circ)$ is finite if it has a finite number of elements
- We denote the cardinality or *order of the group G* by $\mid G \mid$
- The order of an element a of a group $(G, \circ)$ is the smallest positive integer such that
  - $a^k = a \circ a \circ \cdots \circ a = 1$

Example: We try to determine the order of $a = 3$ in the group $\mathbb{Z}_{11}^*$. For this, we keep computing powers of a until we obtain the identity element 1.

$a^1 = 3$

$a^2 = a * a = 3 * 3 = 9$

$a^3 = a^2 * a = 9 * 3 = 27 \equiv 5 \mod 11$

$a^4 = a^3 * a = 5 * 3 = 15 \equiv 4 \mod 11$

$a^5 = a^4 * a = 4 * 3 = 12 \equiv 1 \mod 11$

From the last line it follows that $ord(3) = 5$

Cyclic Groups

- A group G which contains an element a with maximum order $ord(a) = \mid G \mid$ is said to be *cyclic*
- Elements with maximum order are called *primitive elements* or *generators*
- $ord(2) = 4 = \mid \mathbb{Z}_5^* \mid$
  - This implies that $a = 2$ is a primitive element and $\mid \mathbb{Z}_5^* \mid$ is cyclic

Discrete Logarithm Problem (DLP) in $\mathbb{Z}_p^*$

- Given a finite cyclic group $\mathbb{Z}_p^*$ of order $p-1$ and a primitive element $\alpha \in \mathbb{Z}_p^*$ and another element $\beta \in \mathbb{Z}_p^*$
- The DLP is the problem if determining an integer $1 \leq x \leq p-1$ such that $\alpha^x = \beta$

The Elgamal Encryption Scheme

- The Elgamal encryption scheme can be viewed as an extension of the DHKE protocol
- Its security is also based on the intractability of the discree logarithm problem

- We consider the Elgamal encryption scheme over the group $\mathbb{Z}_p^*$, where p is a prime
- If Alice wants to send an encrypted massage x to Bob, both parties first perform a DHKE to derive a shared key kM
- The new idea is that Alice uses kM as a multiplicative mask to encrypt x

Principle of Elgamal Encryption

- Bob computes his private key d and public key β
- Alice, however, has to generate a new public-private key pair for the encryption of every message
  - Her private key is denoted by i and her public key by kE
- The joint key is denoted by kM and is used for masking the plaintext

The Elgamal Protocol

- The set-up phase is executed once by the party who issues the public key and who will receive the message
- The encryption phase and the decryption phase are executed every time a message is being sent
- In contrast to the DHKE, no trusted party is needed to choose a prime and primitive element

Elliptic Curve Cryptography

- Elliptic Curve Cryptography (ECC) is based on the generalised discrete logarithm problem
- ECC is more efficient
- An elliptic "curve" over $\mathbb{Z}_p, p > 3$ is the set of all pairs $(x, y) \in \mathbb{Z}$ which fulfill

$$y^2 = x^3 + ax + b \mod p$$

- Together with an imaginary point O, where $a, b \in \mathbb{Z}_p$ and the condition $4 \times a^3 + 27 \times b^2 \neq 0 \mod p$
  - The curve is non-singular, i.e. has no self-intersections or vertices

Elliptic Curve Properties

- An elliptic curve has several interesting properties
- One of these is horizontal symmetry
  - Any point on the curve can be reflected over the x-axis and remain the same curve

Group Operations on Elliptic Curves

- Given two points $P = (x_1, y_1)$ and $Q = (x_2, y_2)$ we compute the coordinates of the third point R such that

$$P + Q = R$$

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

Point Addition $P + Q$:

- Compute $R = P + Q$ where $P \neq Q$
- Draw a line through P and Q and obtain a third point of intersection between the elliptic curve and the line

Point Doubling $P + P$

- Compute $R = P + Q$ where $P = Q$
- Draw the tangent line through P
  - Obtain a second point of intersection between this line and the elliptic curve
- Mirror the point of the second intersection along the x-axis to obtain R

Computations on Elliptic Curves

$$x_3 = s_2 - x_1 - x_2 \mod p$$

and

$$y_3 = s(x_1 - x_3) - y_1 \mod p$$

where

$$s = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} \mod p & \text{if } P \neq Q \text{ (point addition)} \\ \frac{3x_1^2 - a}{2y_1} \mod p & \text{if } P = Q \text{ (point doubling)} \end{cases}$$

Example: Given $E : y^2 = x^3 + 2x + 2 \mod 17$ and point $P = (5, 1)$ compute $2P$?

$2P = P + P = (5,1) + (5,1) = (x_3, y_3)$

$s = \frac{3x_1^2 + a}{2y_1}$

$s = (2 \times 1)^{-1} \times (3 \times 5^2 + 2) = 2^{-1} \times 9 \equiv 9 \times 9 \equiv 13 \mod 17$

$x_3 = s^2 - x_1 - x_2 = 13^2 - 5 - 5 = 159 \equiv 6 \mod 17$

$y_3 = s(x_1 - x_3) - y_1 = 13(5 - 6) - 1 = -14 \equiv 3 \mod 17$

$2P = (5,1) + (5,1) = (6,3)$

- The points on an elliptic curve and the point at infinity O form cyclic subgroups
  - $2P = (5,1) + (5,1) = (6,3)$
  - $3P = 2P + P = (10,6)$
  - $4P = (3,1)$
  - $5P = (9,16)$
  - $6P = (16,13)$
  - $7P = (0,6)$
  - $8P = (13,7)$
  - $9P = (7,6)$
  - $10P = (7,11)$
  - $11P = (13,10)$
  - $12P = (0,11)$
  - $13P = (16,4)$
  - $14P = (9,1)$
  - $15P = (3,16)$
  - $16P = (10,11)$
  - $17P = (6,14)$
  - $18P = (5,16)$
  - $19P = O$
- This elliptic curve has order $\#E =\mid E \mid= 19$ since it contains 19 points in its cyclic group

Number of Points on an Elliptic Curve

- Determining the point count on elliptic curves in general is hard
- Hasse's theorem bounds the number of points to a restricted interval

  Hasse's theorem: Given an elliptic curve E modulo p, the number of points on the curve is denoted by $\#E$ and is bounded by

  $$p + 1 - 2\sqrt{p} \leq \#E \leq p + 1 + 2\sqrt{p}$$

- The number of points is "close to" the prime p

Elliptic Curve Discrete Logarithm Problem

- ECC cryptosystems are based on the idea that d is large, kept secret, and attackers cannot compute it easily.

  Elliptic Curve Discrete Logarithm Problem (ECDLP): Given an elliptic curve E. We consider a primitive element P and another element T. The DL problem is finding the integer d, where $1 \leq d \leq \#E$, such that
  $$P + P + \cdots + P = dP = T$$

- If d is known, an efficient method to compute the point multiplication dP is required to create a reasonable cryptosystem

Double-and-Add Algorithm for Point Multiplication

- Point multiplication is analog to exponentation in multiplicative groups
- In order to do it efficiently, we can directly adopt the square-and-multiply algorithm (RSA)

Algorithm

Input: Elliptic curve E, and elliptic curve point P and a scalar d with bits d_i

Output: $T = dP$

Initialisation: $T = P$

```
FOR i=t-1 DOWNTO 0
    T=T+T mod n
    IF di = 1
        T=T+P mod n
    RETURN(T)
```

Example

- We consider the scalar multiplication $26P$, which has the following binary representation:
  $$26P = (11010_2)P = (d_4 d_3 d_2 d_1 d_0)_2 P$$
- The algorithm scans the scalar bits starting on the left with d_4 and ending with the rightmost bit d_0

0. $P = 1_2P$
1. $P + P = 2P = 10_2P$
2. $2P + P = 3P = 10_2P + 1_2P = 11_2P$
3. $3P + 3P = 6P = 2(11_2P) = 110_2P$
4. $6P + 6P = 12P = 2(110_2P) = 1100_2P$
5. $12P + P = 13P = 1100_2P + 1_2P = 1101_2P$
6. $13P + 13+ = 26P = 2(1101_2P) = 11010_2P$

Elliptic Curve Diffie-Hellman Key Exchange (ECDH)

- Given a prime p, a suitable elliptic curve E and a point $P = (x_p, y_p)$
  - The Elliptic Curve Diffie-Hellman Key Exchange is defined by the following protocol

1. Alice:
   - Choose $k_{PrA} = a \in \{2, 3, \ldots, \#E - 1\}$
   - Compute $k_{PubA} = A = aP = (x_A, y_A)$
2. Bob:
   - Choose $k_{PrB} = b \in \{2, 3, \ldots, \#E - 1\}$
   - Compute $k_{PubB} = B = bP = (x_B, y_B)$
3. Alice:
   - Compute $aB = T_{AB}$
4. Bob:
   - Compute $bA = T_{AB}$

- Joint secret between Alice and Bob: $T_{AB} = (x_{AB}, y_{AB})$

Key Lengths and Efficiency

Algorithm Family	Cryptosystems	80 bit	128 bit	192 bit	256 bit
Integer factorisation	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete Logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Symmetric-Key	AES, 3DES	80 bit	128 bit	192 bit	256 bit

256-bit ECC key provides the same security as a 3072-bit RSA key

Authentication Protocols

- Authentication is the technique by which a process verifies that a communicating partner is who it is supposed to be and not an impostor
- Authentication based on a shared secret key

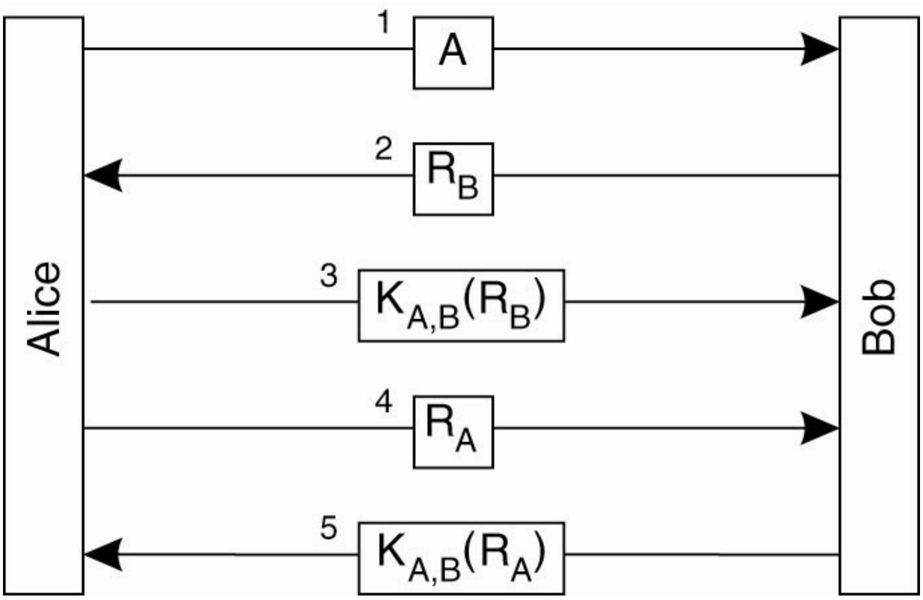


Figure 1: Challenge-Response Protocol

The Reflection Attack

- Taking shortcuts
  1. Alice sends A and R_A together
  2. Bob sends R_B and $K_{AB}(R_A)$ together
  3. Alice sends $K_{AB}(R_B)$
- Trudy can break this if it is possible to open multiple sessions with Bob
  - e.g. if Bob is a bank and is prepared to accept many connections from teller machines at once

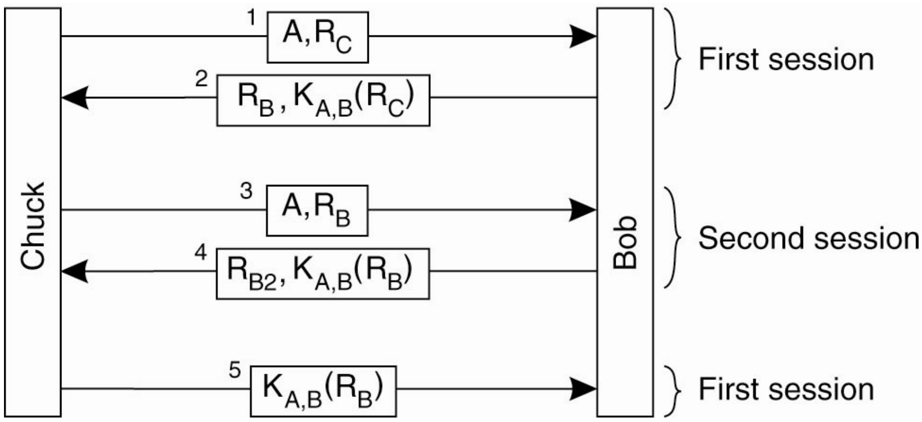


Figure 2: Reflection Attack

Message Authentication Code from Hash Functions (HMAC)

- Calculated using a cryptographic hash function in combination with a secret key
- Like digital signatures they can be used to **quickly** verify *data integrity* and *authenticity* of a message

Authentication Using a KDC

- To talk to n people using a shared secret would require setting up $n \times (n-1)/2$ keys between them
- Alternative is to introduce a Key Distribution Centre (KDC)
- Each user has a single shared key with the KDC

Needham-Schroder Authentication Protocol

- Alice tells the KDC that she wants to talk to Bob
  - Sends a message containing a random number R_A as a nonce
- The KDC sends back message 2 containing R_A, a session key and a ticket that she can send to Bob

Kerberos

- Kerberos V5 is a widely used protocol (e.g. Windows)

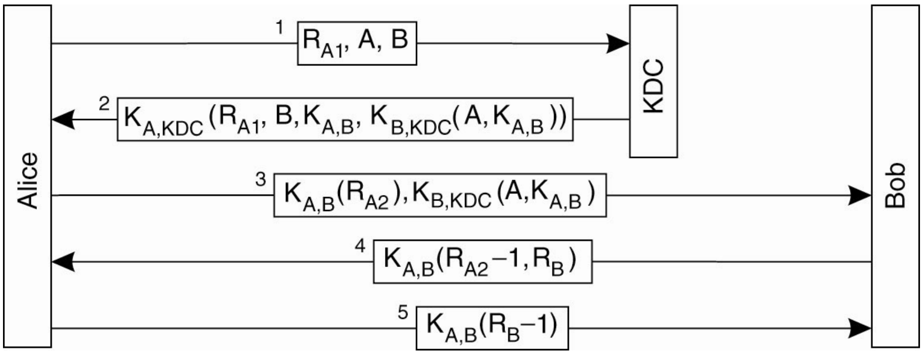


Figure 3: Needham-Schroder

- Authentication includes a Ticket Granting Server (TGS)

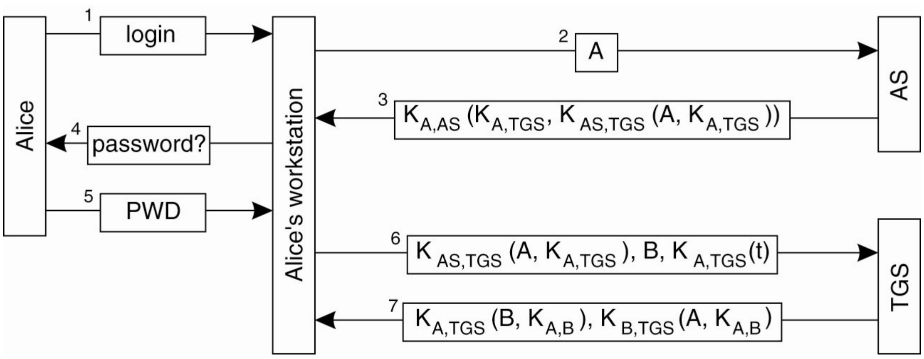


Figure 4: Kerberos

Secure Socket Layer (SSL)

- Provides transport layer security to any TCP-based application using SSL services
  - e.g. Between Web browsers and servers for E-commerce
    * In practice TransportLayer Security (TLS) v1.2 should be used

Application with SSL
Application
SSL

Application with SSL
TCP IP

Toy SSL: A Simple Secure Channel

- Handshake
  - Alice and Bob use their certificates, private keys to authenticate each other and exchange shared secret
- Key Derivation
  - Alice and Bob use shared secret to derive set of keys
- Data Transfer
  - Data to be transferred is broken up into series of records
- Connection Closure
  - Special messages to securely close connection

Key Derivation

- Considered bad to use same key for more than one cryptographic operation
- Four keys
  - K_C: Encryption key for data sent from client to server
  - M_C: MAC key for data sent from client to server
  - K_S: Encryption key for data sent from server to client
  - M_S: MAC key for data sent from server to client
- Keys derived from key derivation function (KDF)
  - Takes master secret and (possibly) some additional random data and creates the keys

Data Records

- Why not encrypt data in constant stream as we write it to TCP?
- Instead, break stream in series of records
  - Each records carries a MAC
- Receiver can act on each record as it arrives
  - Need to distinguish MAC from data
    * Want to use variable-length records

Control Information

- Problem: Attacker can capture and replay, record or re-order records
- Sol: Put sequence number into MAC
  - MAC = MAC(M_X, sequence || data)
    * Note: no sequence number field
- Problem: Attack could replay all records
- Problem: Truncation Attack
  - Attacker forges TCP connection close segment
    * One or both sides thinks there is less data than there actually is
      Sol: Record Types, with one type for closure
  - Type 0 for data, Type 1 for closure
  - Mac = Mac(M_X, sequence || type || data)

Summary and Outstanding Issues

- How long are fields?
- Which encryption protocols?
- Want negotiation?
  - Allow client and server to support different encryption algorithms
  - Allow client and server to choose together specific algorithm before data transfer

SSL Cipher Suite

- Cipher Suite
  - Public-key algorithm
  - Symmetric encryption algorithm
  - MAC algorithm
- SSL supports several cipher suits
- Negotiation
  - Client, server agree on cipher suite
  - Client offers choice
    * Server picks one
- Common SSL Symmetric Ciphers
  - DES: Data Encryption Standard (block)
  - 3DES: Triple Strength (block)
  - RC2: Rivest Cipher 2 (block)

- RC4: Rivest Cipher 4 (stream)

- SSL Public Key Encryption
  - RSA

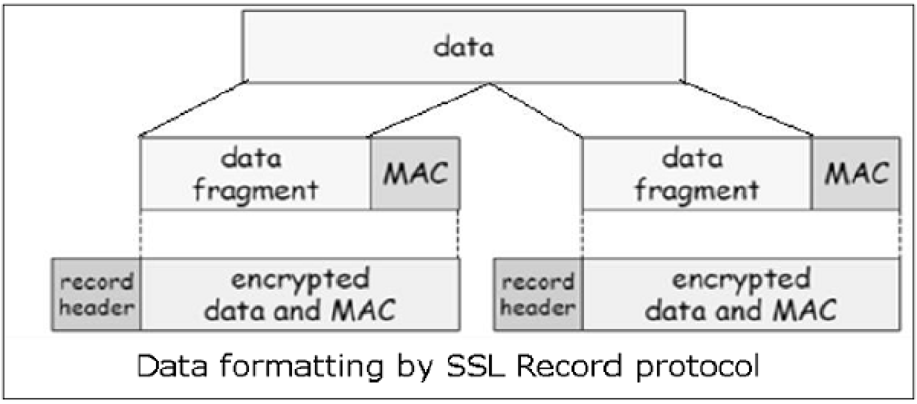
Real SSL

- Server Authentication
- Negotiation: Agree on crypto algorithms
- Establish Keys
- Client Authentication (optional)

Handshake

- Client sends list of algorithms it supports, along with client nonce
- Server chooses algorithms from list; sends back: choice + certificate + server nonce
- Client verifies certificate, extracts server's public key, generates `pre_master_secret`, encrypts with server's public key, sends to server
- Client and server independently compute encryption and MAC keys from `pre_master_secret` and nonces
- Client sends a MAC of all the handshake messages
- Server sends a MAC of all the handshake messages
- Last 2 steps protect handshake from tampering
  - Client typically offers range of algorithms, some strong, some weak
- MITM could delete stronger algorithms from list
  - Last 2 steps prevent this
- Why two random nonces?
  - Suppose Trudy sniffs all messages between Alice and Bob
- Next day, Trudy sets up TCP connection with Bob (Amazon)
  - Bob thinks Alice made two separate orders of the same thing
- Sol: Bob sends different random nonce for each connection. This causes encryption keys to be different on the two days

SSL Record Protocol



- *Record Header*: content type; version; length
- *MAC*: includes sequence number; MAX key M_X
- *Fragment*: each SSL fragment 2^{14} bytes (~16 Kbytes)

SSL Key Derivation

- Client nonce, server nonce, and pre-master secret key input into pseudo random-number generator
  - Produces Master Secret (MS)
- Master Secret and nonces input into another random-number generator
  - To produce the "key block"
- Key block sliced and diced
  - Client MAC key
  - Server MAC key
  - Client encryption key
  - Server encryption key
  - Client initialisation vector (IV)
  - Server initialisation vector (IV)

Virtual Private Networks (VPNs)

- Institutions often want private networks for security
  - Costly! Separate routers, links, DNS infastructure
- With a VPN, institution's inter-office traffic is sent over public Internet instead

IPsec

Transport vs Tunnel Mode

- Transport Mode
  - IPsec datagram emitted and received by end-system
- Tunnel Mode
  - End routers are IPsec aware

Network Layer Security

- Network-layer authentication
  - Destination host can authenticate source IP address
- Network-layer secrecy
  - Sending host encrypts the data in IP datagram
    * TCP and UDP segments, ICMP and SNMP messages
- Two principal protocols
  - Authentication header (AH) protocol
  - Encapsulation security payload (ESP) protocol
- Most common
- Source and destination perform a handshake
  - Create network-layer logical channel called a security association (SA)
- SA at R1
  - 32-bit identifier for SA: Security Parameter Index (SPI)
  - Origin interface of the SA (200.168.1.100)
  - Destination interface of the SA (193.68.2.23)
  - Type of encryption to be used (e.g. 3DES with CBC)
  - Encryption key
  - Type of integrity check (e.g. HMAC with MD5)
  - Authentication key

ESP with Tunnel Mode

- Appends to back of original datagram (which includes original header fields!) an “ESP trailer” field

  - Encrypts result using algorithm & key specified by SA

- Appends to front of this encrypted quantity the ESP header
- Creates authentication MAC over the four fields, using algorithm and key specified in SA
  - Appends MAC to the end forming payload
- Creates brand new IP header, with all the classic IPv4 header fields, which it appends before payload

DNS Security Extension (DNSSEC)

- Plain old DNS does not allow you to check the *authenticity* or *integrity* of a message
- MITM attack
  - A resolver has no way to verify the authenticity and integrity of the data sent by name servers
- Packet sniffing
  - DNS sends an entire query or response in a single unsigned and unencrypted UDP packet
    - Attacker can mount an active attack and change the contents
- Transaction ID guessing
  - An attacker can respond with false answers to a predicted query
    - On the client there are 2^{32} possible combinations of ID (2^{16}) and UDP ports (2^{16})

DNSSEC provides origin authentication and integrity assurance services for DNS data.

DNSSEC has two perspectives

- Domain owners sign their zone and publish the signed zone on their authoritative name servers
- Querying hosts validate the digital signatures they receive in answers, along a chain of trust

DNSSEC RRs

DNSSEC adds a number of new resource record types:

- RRSIG: Contains a cryptographic signature

- DNSKEY: Contains a public signing key
- DS: Contains a hash of a DNSKEY record
- Others…

RRsets

- First step towards securing a zone with DNSSEC is to group all the records with the same type into a resource record set (RRset)
  - e.g. If you have three AAAA records in your zone, they would all be bundled into a single AAAA Rset

Zone-Signing Keys

- Each zone in DNSSEC has a zone-signing key pair (ZSK)
- A zone operator creates digital signatures of each RRset using the private ZSK
- Zone operators also need to make their public ZSK available by adding it to their name server in a DNSKEY record
- When a DNSSEC resolver requests a particular record type (e.g. AAAA), the name server also returns the corresponding RRSIG

Key-Signing Keys

- The KSK validates the DNSKEY record
- It signs the public ZSK (which is stored in a DNSKEY record)

DNSSEC Validation

- Requests the desired RRset
  - Returns the corresponding RRSIG record
- Request the DNSKEY records containing the public ZSK and public KSK
  - Returns the RRSIG for the DNSKEY RRset
- Verify the RRSIG of the requested RRset with the public ZSK
- Verify the RRSIG of the DNSKEY RRset with the public KSK

Delegation Signer Records

- DNSSEC introduces a delegation signer (DS0 record to allow the transfer of trust from a parent zone to a child zone
- A zone operator hashes the DNSKEY record containing the public KSK
- Every time a resolver is referred to a child zone, the parent zone also provides a DS record
  - To check the validity of the child zone's public KSK