Data and Network Security

Jordan Pyott

2021-05-06 17:02

Contents

Course Information

Lecturers Details

- Lecturer: Dr. Clementine Gritti
  - Office: Erskine 304
  - Email: clementine.gritti@canterbury.ac.nz
- Tutor: Ryan Beaumont
  - Email: rbe72@uclive.ac.nz

Weekly Lab Times

- Tuesday, 16:00–18:00, Jack Erskine 136 Lab 4.
- Wednesday, 14:00–16:00, Jack Erskine 136 Lab 1.
- Friday, 16:00–18:00, Jack Erskine 136 Lab 4.

Other Information

- Labs and Quiz's will be available on learn
- Textbooks
  - Cryptography and network security : principles and practice, William Stallings, 5th edition
    * This course is inspired from this book as the bulk of the course is founded in cryptography.
    * The exam will only be on content in the slides not from the book
  - Computer security : principles and practice, William Stallings and Lawrie Brown, 3rd edition

2.1 Term 1 Plan

Week starting date	Course week	Monday lecture	Thursday lecture	Lab
19/07/2021	1	L1: Course introduction	L2: Course overview	no lab
26/07/2021	2	L3: Discrete mathematics	L4: CrypTool (at home)	Lab 1: Introduction
02/08/2021	3	L5: Classical encryption part 1	L6: Classical encryption part 2	Lab 2: Discrete maths exercises
09/08/2021	4	L7: Block ciphers	L8: Block cipher modes	Lab 3: CrypTool part 1
16/08/2021	5	L9: Stream ciphers	L10: Number theory	Lab 4: CrypTool part 2
23/08/2021	6	L11: Hash functions and MACs	no lecture	Lab 5: Number theory exercises

2.2 Term 2 Plan

Week starting date	Course week	Monday lecture	Thursday lecture	Lab
13/09/2021	7	L12: Public key crypto part 1	L13: Public key crypto part 2	Lab 6: Hash functions and MACs exercises
20/09/2021	8	L14: Digital signatures	L15: PKI and certificates	Lab 7: CrypTool part 3
27/09/2021	9	L16: Key establishment	L17: TLS part 1	Lab 8: PKI and certificates
04/10/2021	10	L18: TLS part 2	L19: IPSec and VPN	Lab 9: Digital signatures and key establishment exercises
11/10/2021	11	L20: Email security	L21: Malware and attacks	Lab 10: TLS
18/10/2021	12	L22: Recap lecture	no lecture	Lab 11: IPSec and email security exercises

Figure 1: Timetable

Assessment

1. Labs (10%) - attendance and participation:

- Labs are done individually but you are encouraged to discuss and share with your peers (you are allowed to see each other during labs).
- Attending one lab each week over the semester automatically gives you full mark: – The tutor will assess your attendance.
- If you cannot attend one lab session, then a report (along with a justification of student absence) will be required and assessed:
  - The report needs to be submitted by one week after the missed session.
  - Example: if you miss Tuesday lab on Week X then you are asked to submit a report by Tuesday of Week X+1.
  - The report needs to be sent to *both* the lecturer and the tutor.

2. Weekly quizzes (20%):

- They can be found and done on LEARN.
- 9 quizzes in total.
- Each quiz contains 10 questions. Each question contains 4 choices such that only one choice is correct.

- 2 attempts per quiz, such that the highest grade is taken into account.
- A quiz is given on Friday of Week X, and should be done before Friday of Week X+1 (except for the one released just before the break):

3. Assignment (20%):

- *Deadline*: 17 September 2021.
- Small exercises on what has been covered so far.
- The assignment will be released on LEARN on 20 August 2021.
- Your report should be uploaded to LEARN.

4. Final exam (50%)

- 3-hours duration
- 25 multiple-choice questions
- 5 open questions, such that if additional information is needed to solve the problem then it will be provided.
- Covers all content from all lectures study *definitions, mechanisms, processes*
    - Not expected to remember the code of each standard (e.g. RFC1234)

Lectures

Lecture One - Course Introduction

- All materials will be found on learn, including lectures, labs, quizzes and assignments.
- Course outline available
- Labs must be done in person or a report *will not get full marks if do not attend*
- Labs start next week
- Weekly quizzes go over two lectures each *multi-choice*
- Midterm and final will all be entirely open questions

Why do we need cyber security

- Privacy
- Security
- Risk management

Famous recent attacks

- Dark hotel attack
    - Targeted phishing attacks using spy-ware

    - Infiltrating guests computers through WIFI networks at hotels
    - Loss of confidentiality
- POODLE attack
    - man in the middle exploit
    - Communications can be decryped and exploited
- EncroChat
    - A communications network and service provider
    - Infiltrated by police in 2020
- WannaCry
    - Loss of availability
    - stolen government hacking tools
    - Worm encrypting files on computers hard drive
        * Was a form of ransom-ware
- Botnet
    - Botnet attacking IoT devices with default admin credentials
    - DDos
    - Loss of availability

Because of these attacks, some users have lost confidence in the service provided not storing/selling data.

Course Focus

- Cryptography as a foundation for information security
- Applications of cryptography
- History of cryptography
- Modern cryptography
    - Block ciphers, stream ciphers
    - public key crypto
    - Hashing and MAC
- Some mathematics
    - Modular arithmetic
    - Number theory
    - Elliptic curves

- Using all of the cryptography
  - Public key infrastructure
  - Secure email
  - TLS (HTTPS)

Lecture Two - Course Overview

What is cyber security?

Definition from the NIST computer security handbook:

- The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability and confidentiality of information system resources.
  - Some literature might differentiate between *computer security and cyber security*

Definitions

- A threat
  - Represents a potential security harm to an asset
- An attack
  - is a threat that is carried out
- The threat agent
  - carrying out the attack is referred to as an attacker
- A countermeasure
  - Any means taken to deal with an attack
- A residual level of risk to the assets
  - represented by vulnerabilities possibly exploited by threat agents
- **Assets**
  - Computer systems and other data processing, storage and data communication devices
  - OS, system utilities and applications
  - Files and databases and further data
  - Local and wide area network links
- **Vulnerabilities**

    - A computer system can be:
        * Leaky
            · meaning gives access to information through the network, violates confidentiality
        * Corrupted
            · meaning that it does the wrong thing, violates integrity principle
        * Unavailable
            · meaning that it becomes impossible or impractical to use, violates availability

- Passive attacks
    - Interception
    - Traffic analysis
        * Spoofing, finding information and observing traffic

- Active attacks
    - Altering information and system resources
    - May be hard to prevent but easy to detect
    - Masquerade
        * the attacker claims to be someone else *authorized*
    - Falsification (Man in middle)
        * the attacker changes messages during transmission
    - Misappropriation (DDOS)
        * the attacker prevents legitimate users from accessing resources

- Inside attacks
    - initiated by an entity INSIDE the security perimeter
    - authorization to access system resources but use of them in a malicious way
    - Exposure
        * the attacker intentionally releases sensitive information to an outsider.
    - Falsification
        * the attacker alters or replaces valid data or introduces false data into a file or database.

- Outside attacks
    - initiated from OUTSIDE the perimeter, by an unauthorised or illegitimate user of the system
    - Obstruction
        * the attacker disables communication links or alters communication control information.

  - Intrusion
    * the attacker gains unauthorised access to sensitive data by overcoming the access control protections.

Security functional requirements

- Information security management requires to:
  1. Identify threats
  2. Classify all threats according to liklihood and severity
  3. Apply security controls based on cost benefit analysis
- Countermeasures to vulnerabilities and threats comprise:
  1. Computer security technical measures
  - access control, authentication and system protection
  2. Management measures
  - awareness and training
  3. Both
  - configuration management

Defining Information Security

Definition from the NIST computer security handbook:

- The term security is used in the sense of minimizing the vulnerabilities of assets and resources. An asset is anything of value. A vulnerability is any weakness that could be exploited to violate a system or the information it contains. A threat is a potential violation of security.

The CIA Triad

- Confidentiality
  - Preventing unauthorised disclosure of information
- Integrity
  - Preventing modification or destruction of information
- Availability
  - ensuring resources are accessible when required

Information Security Definitions

- Security Service
  - a processing or communication service to give a specific kind of protection to system resources.
  - Types of security services [Lecture Two - Slide 20/27]:
    * Peer entity authentication
    * Data origin authentication
    * Access control
    * Data confidentiality
    * Traffic flow confidentiality
    * Data integrity
    * Non-repudiation
    * Availability
- Security Mechanism
  - a method of implementing one or more security services.
  - Types of security mechanisms
    * Encipherment
    * Digital signature
    * Access control
      · access control lists, password or tokens which may be used to indicate access rights
    * Data integrity
    * Authentication exchange
    * Traffic padding
    * Routing control
    * Notarization

Risk Management

A key tool in information security management:

1. Identify threats
2. Classify all treats according to likelihood and severity
3. Apply security controls based on cost benefit analysis

Lecture Three - Number Theory and Finite Fields

Factorisation

The set of all integers is denoted by $\mathbb{Z} = ..., -3, -2, -1, 0, 1, 2, 3, ...$, given $a, b \in \mathbb{Z}$, a divides b if there exists $k \in \mathbb{Z} \quad s.t \quad ak = b$.

- This means that a is a factor of b
- $a|b$

We use p to denote a prime, an integer $p \geq 1$ is a *prime* if its divisors are $(1, p)$.

- Testing a prime number p by trial numbers up to the square root of p
  - There are more efficient ways to check for primality *later in the course*.
- **Division algorithm**
  - given $a, b \in \mathbb{Z}, \quad s.t \quad a > b$, then there exists $q, r \in mathbbZ \quad s.t \quad a = bq + r$
  - $a = bq + r$ and $0 \leq r \; leqb$, we can use this to show $r < \frac{a}{2}$.
- **Greatest common divisor (GCD)**
  - $gcd(a, b) = d$ if $d|a$ and $d|b$
  - if $c|a$ and $c|b$ then $c|d$
  - a and b are *relatively prime / co-prime* when $gcd(a, b) = 1$
- **Euclidean Algorithm**
  - Find $d = gcd(a, b)$

$$a = bq * 1 + r_1 \quad for \quad 0 < r_1 < b$$

$$b = r_1 q_1 + r_2 \quad for \quad 0 < r_2 < r_1$$

$$r_1 = r_2 q_1 + r_3 \quad for \quad 0 < r_3 < r_2$$

$$...$$

$$f * k - 3 = r * k - 2q * k - 1 + r * k - 1 \quad for \quad 0 < f * k - 1 < f * k - 2$$

$$f * k - 2 = r * k - 1q * k + r * k \quad for \quad 0 < f * k < f * k - 1$$

$$f * k - 1 = r * kq * k + 1 + r * k + 1 \quad with \quad r * k + 1 = 0$$

- Hence $d = r_k = gcd(a, b)$
- **Back Substitution - Extending Euclidean Algorithm**
  - Finding x, y in $ax + ay = d = r_k$
  - This is essentially reversing the Euclidean algorithm

- **Modular Arithmetic:**
- Given $a \equiv b$ (mod n) and $c \equiv d$ (mod n), then the following conditions hold:
  - $a + b \equiv c + d$ (mod n)
  - $ac \equiv bd$ (mod n)
  - $ka \equiv kb$ (mod n)
- **Groups**
  - A group $\mathbb{G}$ is a set with *binary operation* and:
    * Closure: $a \cdot b \in \mathbb{G}$ for $a, b \in \mathbb{G}$
    * Identity: there is an element 1, s.t. $a \cdot 1 = 1 \cdot a = a$ for $a \in \mathbb{G}$
    * Inverse: there is an element b s.t. $a \cdot b = 1$ for $a \in \mathbb{G}$
    * Associativity: $(a \cdot b) \cdot c = a \cdot (b \cdot c)$ for $a, b, c \in \mathbb{G}$
    * Commutativity: $a \cdot b = b \cdot a$ for $a, b \in \mathbb{G}$
      · If this condition holds, the group is said to be *abelian*.
- **Cyclic Groups**
  - the order $|\mathbb{G}|$ of a group $\mathbb{G}$ is the number of elements in $\mathbb{G}$
  - g^k denote the repeated application of $g \in \mathbb{G}$, using the group operation
  - The order $|g|$ of $g \in \mathbb{G}$ is the smallest integer k s.t. $g^k = 1$
  - g is a generator of $\mathbb{G}$
  - A group is said to be *cyclic* if it has a generator
- **Finding inverse**
  - Use extended Euclidean algorithm if GCD is 1.
- **Fields**
  - A field $\mathbb{F}$ is a set with binary operations:
  - $\mathbb{F}$ is an *abelian group* under the operation $+$ with identity element of 0.
- **Finite Fields**
  - Setting up secure communications requires fields with a finite number of elements.
  - Notation is $GF(p) = \mathbb{Z}\_p$
  - Addition modulo 2: XOR LOGIC

Lecture Four - CrypTool

What is cryptool?

- Open source program that focuses on the free e-learning software, illustrating cryptographic concepts.
- We will be using CrypTool 2
- Video to go back and watch

Lecture Five - Classic Encryption

Goals:

- Study historical ciphers
- Establish basic notation and terminology
- Introduce basic cryptographic operations
- Explore typical attacks and adversary capabilities

Terminology:

- Cryptography: The study of designing crypto-systems, key components are:
  - Confidentiality: A key is needed to *read* the message
  - Authentication: A key is needed to *write* the message
- Cryptanalysis: The study of breaking crypto-systems
- Stenography: The study of concealing information
- Crypto-system
  - A set of plain-texts
  - A set of cipher-texts
  - A set of keys
  - A function called *encryption* which transforms plain text to cipher text
  - An inverse function called *decryption*, which reverses the encrypted message to plain-text
- Symmetric and Asymmetric Cryptography
  - Symmetric key cipher:
    * Encryption and decryption keys are known to the sender and receiver
    * Secure channel for transmission of the keys
  - Asymmetric key cipher
    * Each participant has a public key and a private key
    * Possibly working for both encryption of messages and the creation of digital signatures

Symmetric Encryption Notation for Symmetric Encryption Algorithms:

- Encryption function: E
- Decryption function: D
- Message or plain-text: M
- Cipher-text: C
- Shared secret key: K

Encryption is denoted as $C = E(M, K)$
Decryption is denoted as $M = D(C, K)$

Methods to break Symmetric encryption:

An adversary has access to many methods to break a cryptosystem, however such methods depend on some conditions and known knowledge, such as:

- What are the resources available to the adversary?
  - Examples: *computational capability, inputs/outputs, knowledge of the crypto-system*
- What does the adversary want?
  - Do we want to know the secret key, distinguish two messages, **what are we trying to achieve?**.

Exhaustive Key Search:

This is a brute force attack, that checks all combinations of possible keys, note there is no way to prevent such an attack, our best option is just to raise complexity to where it is to combinationally difficult to compute, however this does not stop people from getting lucky when breaking your system.

- We may be able to break the system without exhaustive search.
- We may be able to break the system without finding the key.

Attack classifications:

These are ordered from the least powerful to the most dangerous information an attacker could have.

1. Cipher-text Only Attack - the attacker only has access to intercepted encrypted messages
2. Known Plain-text Attack - The attacker knows some of the plain text
3. Chosen Plain-text Attack - The attacker can obtain the cypher-text from some plain-text that has been selected
4. Chosen Cipher-text Attack - The attacker can obtain the plain-text from some cipher-text that it has selected

Which of these should be prevented?

- A crypto-system is seen as very insecure if it can be practically attacked using only intercepted cypher texts
- A crypto-system should be secure against chosen plain-text and chosen cipher-text attacks (*this is the modern standard*)
- History shows that chosen cipher-text attacks are practical to set up for an attacker

Kerchhoffs' Principle: We assume an attacker has complete knowledge of the cipher, the only item that is unknown about the crypto-system is the private key.

- Using a secret, non-standard algorithm can cause severe problems
  - This would be an example of *security through obscurity* (note this is a bad habit)

Statistical Attacks:

- Depends on using the redundancy of the alphabet
- Information from distribution of letters
- Recognising patterns to guess information about the crypto-system

Basic cipher operations:

Historical ciphers combine two basic operations:

- Transposition: Characters on the plain-text are mixed up with each other
  - Permutating characters in a fixed period d and a permutation f
  - Plain-text is seen as a matrix of rows of length d
  - Key is (d, f)
  - Each block of d characters is re-ordered using permutation f
  - Complexity is $d!$
  - We will go through how to do this by hand and a formulation of how to automate this process using Crypt Tool
- Substitution: Characters are replaced by a different character (or set of characters)
  - Each character in the plain-text alphabet replaced by a character in the cipher-text alphabet, following a cypher-text table.
  - *Caesar cipher* is a simple example of a substitution cipher
    * Functions for encryption $C_i = (M_i + j) mod n$
    * Functions for decryption $M_i = (C_i + j) mod n$
    * To break this we can do a frequency analysis in order to find common characters (*example, use ' ' as most common letter*)

Lecture Six - Classic Encryption: Block ciphers

Defining Polyalphabetic Substitution

- Using multiple mappings from plaintext to ciphertext
- The effect with multiple alphabets is to smooth frequency distrabution
  - Direct frequency analysis should no longer be effective
- Typical polyalphabetic ciphers are periodic substitution ciphers based on period d
- Given d ciphertext alphabets $C_0, C_1, ..., C_{d-1}$ let $f_i : A \rightarrow C_i$ be a mapping from the plaintext alphabet A to the i_{th} ciphertext alphabet $C_i \forall 0 \leq i \leq d-1$
- Encryption Process:
  - Plaintext: $M = M_0...M_{d-1}M_d...M_{2d-1}M_{2d}...$
  - is encrypted to:
    - $E(K, M) = f_0((M_0))...f_{d-1}(M_{d-1})f_0(M_d)...f_{d-1}(M_{2d-1})f_0(M_{2d})...$
  - Special case with $d = 1$: the cipher is monoalphabetic (simple substitution cipher)
- Key generation:
  - Select a block d
  - Generate d random simple substitution table
- Encryption:
  - Encrypting the character by using the substitution table number j such that $i \equiv j(modd)$
- Decryption
  - Using the same substitution table as in encryption in order to reverse the simple substitution

Let $d = 3$, thus there are 3 ciphertext alphabets.

Pltxt char.	ABC	DEF	GHI	JKL	MNO
C_1	UWY	SX∇	TVZ	CEI	AFG
C_2	QLM	PJO	RKN	∇XS	YUW
C_3	MLQ	RNQ	GFA	ZVT	YWU

Pltxt char.	PQR	STU	VWX	YZ∇	
C_1	BDH	KNR	JOP	LMQ	
C_2	ZVT	FGA	HDB	EIC	
C_3	POJ	HDB	IEC	∇XS	

If the plaintext is IT∇IS∇A∇BEAUTIFUL∇DAY then the ciphertext is ZGSZFSUCLXQBNNKRSSSQ∇.

Figure 2: Polyalphabetic example

Vigenere Cipher

- Popular form of periodic substitution ciphers based on *shifted* alphabets
- The key K is a sequence of characters
  - $K = K_0K_1...K_{d-1}$

Message M	AT∇T	HE∇T	IME∇
Key K	LOCK	LOCK	LOCK
$E(K, M)$	LGBC	SSBC	T∇GJ

- Numbering the alphabet:
  $A = 0, B = 1, \cdots, Z = 25, \nabla = 26.$
- In particular, $L = 11, O = 14, C = 2, K = 10$:
  - the 1st character of each 4-character group is shifted by 11,
  - the 2nd character is shifted by 14,
  - the 3rd character is shifted by 2,
  - the 4th character is shifted by 10.
- Shifting is computed modulo 27 (the alphabet "wraps around").

Figure 3: Vigenere example

Crypto-analysis:

- Identify the period length
  - Kasiski method
  - Cryptool uses autocorrelation to estimate the period
- Attack separately d substitution tables
  - Each substitution is just a shift
    * If there is sufficient ciphertext then it is trivial

Autocorrelation method

- Method used to find the period length d of any periodic polyalphabetic cipher
- Given ciphertext C, computing the correlation between C and its shift C_i for all values of i, of the period
- Seeing peaks in the value will help to find i

Hill Cipher

- THe americal mathematition Lester S. Hill published his cipher in 1929

- *Polygram cipher*
  - Simple substitution cipher on an extended alphabet consisting of multiple characters
  - Example: Diagram substatution in which the alphabet consists of all pairs of characters
- Major weakness: its linearity, hence known plaintext attacks are easy
- Let $d = 2$ so encryption takes diagrams as input and output blocks
- Each plaintext pair is written as a column vector, letters are encoded as numbers

Performing a linear transformation on d plaintext characters to get d ciphertext characters:

- Encryption involves multiplying a $d \times d$ matrix K by the block of plaintext M.
  - $C = KM$
  - tutorial: 27:00
- Decryption involves multiplying the matrix K^{-1} by the block of ciphertext C
  - $M = K^{-1}C$

$$d = 2, K = \begin{pmatrix} 4 & 6 \\ 1 & 7 \end{pmatrix}, K^{-1} = \begin{pmatrix} 4 & 12 \\ 11 & 10 \end{pmatrix}$$

$$\text{One can check that } KK^{-1} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

$$\text{Plaintext: } M = (\text{EG}) = \begin{pmatrix} 4 \\ 6 \end{pmatrix}$$

$$\text{Encryption: } C = KM = \begin{pmatrix} 4 & 6 \\ 1 & 7 \end{pmatrix}\begin{pmatrix} 4 \\ 6 \end{pmatrix} = \begin{pmatrix} 25 \\ 19 \end{pmatrix} = (\text{ZT})$$

$$\text{Decryption: } M = K^{-1}C = \begin{pmatrix} 4 & 12 \\ 11 & 10 \end{pmatrix}\begin{pmatrix} 25 \\ 19 \end{pmatrix} = \begin{pmatrix} 4 \\ 6 \end{pmatrix} = (\text{EG})$$

Figure 4: encryption example

Crypto analysis:

- Known plaintext attacks possible given d
- Given blocks m_i and C_i for $0 \leq i \leq d - 1$
- example: 29:00

Block Cipher

- Block ciphers are the main bulk encryption algorithm in commercial applications
- AES and legacy cipher DES are widely used
- Symmetric key ciphers where each block is encrypted with the same key
- A block is a set of plaintext symbols of a fixed size
- length of plaintext and ciphertext is the same

Product cipher:

- Cryptosystem where encryption is formed by applying several sub-encryption functions
- Most block ciphers are a composition of simple functions

Iterated cipher:

Most modern block ciphers are special product ciphers

- encryption is divided into r smaller rounds
- sub-encryption functions are all same function, called a *round function*
- key K_i is derived from overall master key K
- tutorial: 38:40
- Decrypt by taking the inverse

Types of iterated ciphers:

- Substitution-Permutation network
  - Block length n mst allow each block to be split into m sub-blocks
  - we have two operations:
    * Substitution: $\Pi_S : 0, 1^{'} \rightarrow 0, 1^{'}$
    * Permutation: $\Pi_P : 1, ..., n \rightarrow 1, ..., n$
    * tutorial: 45:00

Feistel Cipher:

- Round function swaps two blocks and forms a new right hand half
- Can be mapped to a network where two halves plaintext travels through
- tutorial: 50:00

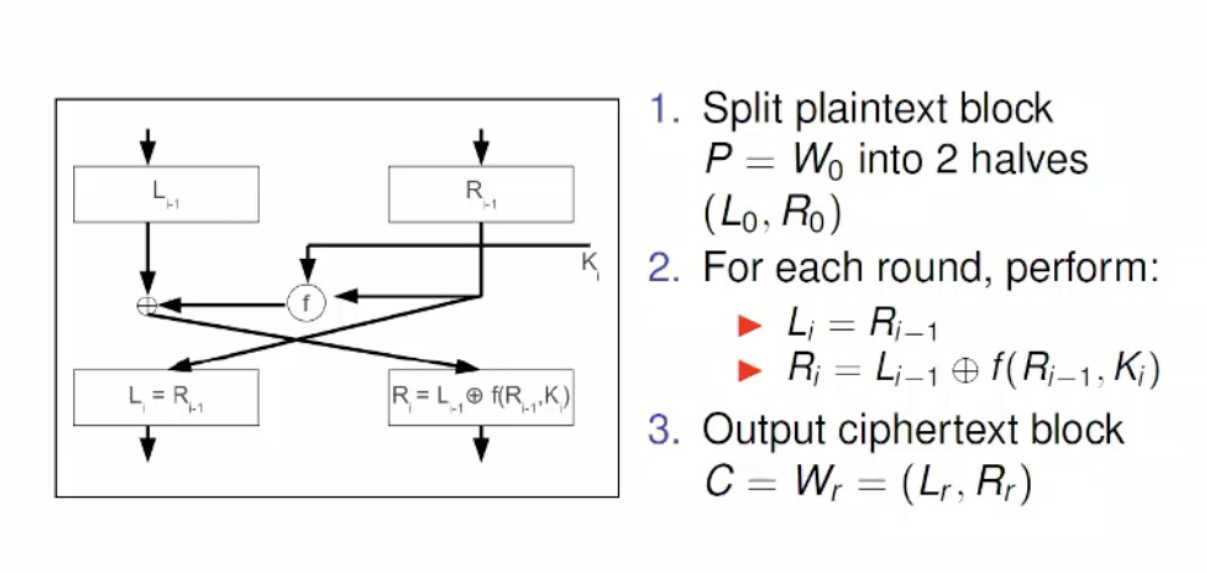


Figure 5: Feistel graphic

Differential Crypto-analysis

- Chosen plaintext attack
- Based on the idea that the difference between two input plaintext can be correlated to the difference between the output text

Linear Crypto-analysis

- Known plaintext attack
- Theoretically used to break DES

Lecture Seven: Block ciphers continued

- Encryption and decryption definitions are public property
- Security resides in the difficulty of decryption without the key
- Encryption steps: 08:00

Row No.	Column No.															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

- Let input block W be $x_1x_2x_3x_4x_5x_6$
- Digits x_1 and x_6 define row number between 0 and 3
- Digits x_2, x_3, x_4, x_5 define column number between 0 and 15
- Example: $W = 100101$
  - $x_1 = 1$ and $x_6 = 1$, thus 11, that is 3
  - $x_2 = 0, x_3 = 0, x_4 = 1, x_5 = 0$, thus 0010, that is 2

Figure 6: S-box Example

Key schedule

- Each of the 16 rounds involves 48 bits of the 56-bit key
- Each 48-bit subkey is defined by a series of permutations

Breaking DES

- Testing all possible 2^k keys in order to find the key K
- Key identified by using small number of cipher-text blocks or by looking for low entropy in decrypted plain-text
- 2^{56} DES keys to test
- Short DES key size was criticised from the start
  - Realistically brute force is a good method to break a 56-bit key
  - In 2017 we could find the keys in < 25 seconds
- To deal with the lack of security they created a double encryption
  - Having two keys K_1, K_2 be two block cipher keys

MITM attack

Let (P, C) be a single plaintext-ciphertext pair:

1. For each key K store $C' = E(P, K)$ in memory

2. Check if $D(C, K') = C'$ for any key k'
3. Check if any key values in 2. work for other (P, C) pairs

> we need to store plain-text, single encryption keys and decryption key

> We can just add more layers of encryption to get even better security (triple encryption), this will protect against MITM attacks, it is just too expensive and time restricted to be feasible

> this is only used in legacy code now, other block ciphers are better

AES, Advanced Encryption Standard

- Designed in an open competition due to DES being bad
- Process over several years with much public debate
- 128-bit data block (fixed)
- Can have any sized master key (128, 192, 256)
- Byte-based design
- uses a state matrix
- Information on AES: 33:00

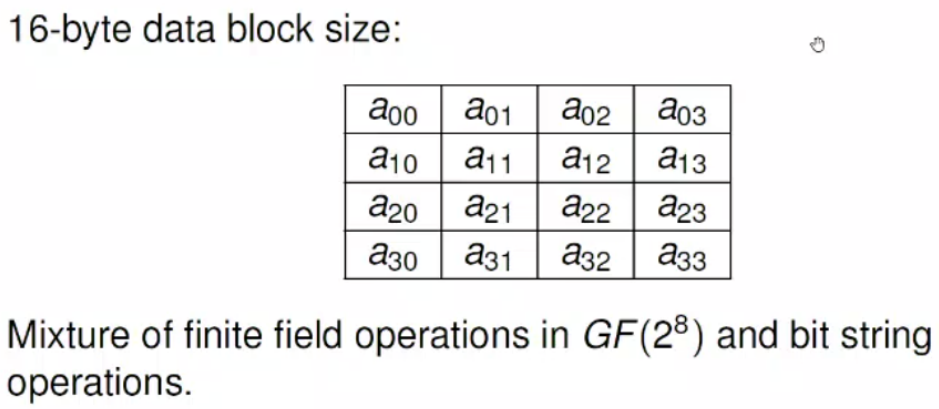


Figure 7: State matrix AES

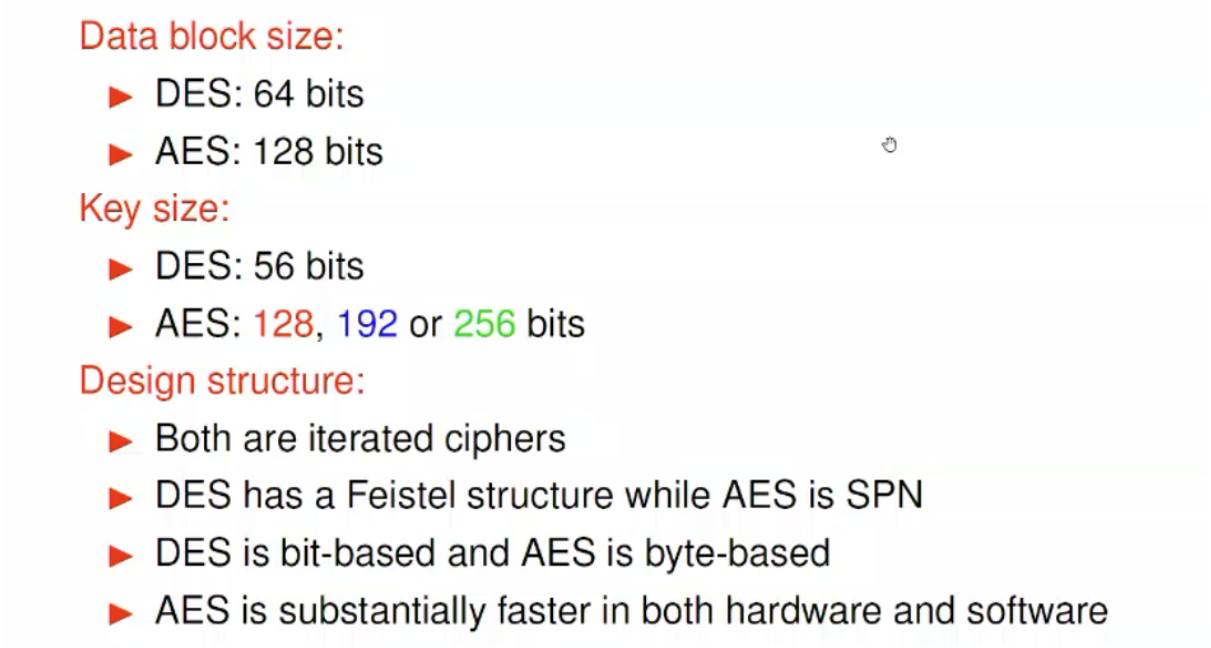


Figure 8: AES vs DES

- Block ciphers are the workhorses of secure communication
- AES is the current choice, Triple DES is still used
- Block ciphers are used as building blocks fore confidentiality and authentication

Lecture Eight: Block cipher's Modes of Operation

Why different modes?

Designed to provide confidentiality for data OR authentication for data OR both

Different modes have:

- Different efficiency properties
- Different communication properties

Problem with randomised encryption:

- The same plaintext block is encrypted to the same ciphertext block
- Prevention:
  - By using an initialisation vector which propagates through the entire ciphertext (IV)
  - IV may require to be either unique or random

- Features impacting on efficiency:
  - Parallel processing
    * Multiple plaintext blocks are encrypted in parallel
    * Multiple ciphertext blocks are decrypted in parallel
  - Error propagation
    * Bit errors can be an issue
- Padding:
  - Requiring the plaintext to complete of complete blocks
  - NIST suggests a padding method
  - Padding bits removed unabiguously if usage of this padding method is known
    * Remove all trailing `0` bits after the last `1` bit
    * Remove the single `1` bit
- Notations:
  - Plaintext message P
  - $t-th$ plaintext block P_t
  - Ciphertext $C\, t-th$ ciphertext block C_t
  - Key K
  - initialisation vector IV

Confidentiality Modes

- ECB Mode encryption
  - Basic mode of a block cipher
  - Encryption: $C_t = E(P_t, K)$
  - Decryption: $P_t = E(C_t, K)$
  - Padding required
  - Not randomised
  - No IV
  - Allows for parallel encryption and decryption
- CBC Mode Encryption
  - Encryption: $C_t = E(P_t \bigoplus C_{t-1})$ s.t $P_0 = IV$
  - Decryption: $P_t = E(C_t \bigoplus C_{t-1})$ s.t $C_0 = IV$
  - Randomised
  - Padding required

  - IV must be random
  - Parallel decryption allowed
  - Common choice for channel protection in TLS

- CTR Mode:

  - Synchronous stream cipher mode
  - A counter and a nonce are used initialised using a randomly selected value N
  - Encryption: $C_t = O_t \bigoplus P_t$
  - Plaintext block P_t is XOR'd with O_t
  - Decryption: $P_t = O_t \bigoplus C_t$
  - Don't require padding
  - randomised
  - error propagation: occurs in specific bits of current block

Authentication Modes

> NOTE: I got bored of typing here, should probably go back over this content

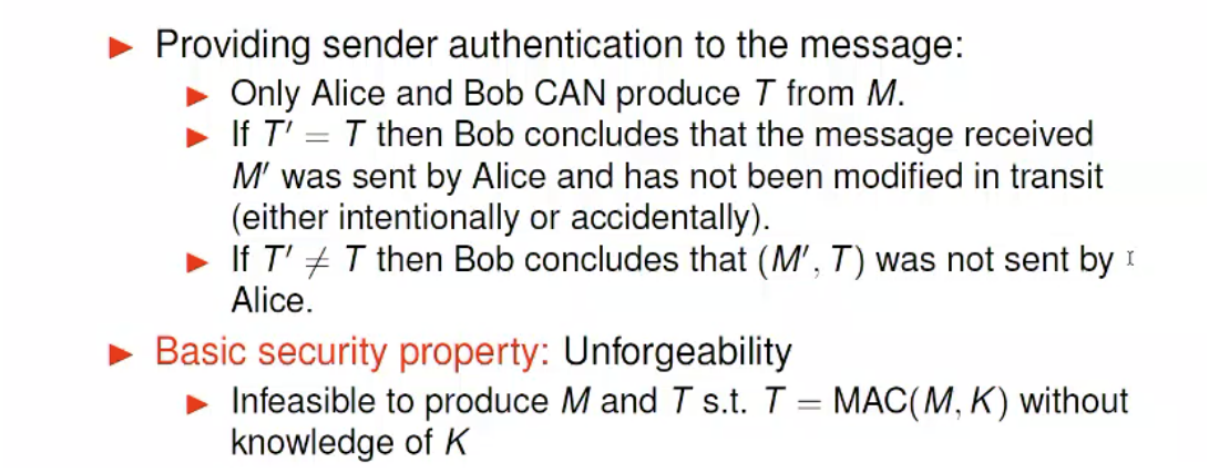


Figure 9: MAC Properties

- Using a block cipher to create a MAC providing message integrity, not confidentiality
- IV must be fixed and public, can be set to all 0's
- P is the message

- $T = CBC - MAC(P, K)$
- Unforgeable as long as the message length is fixed

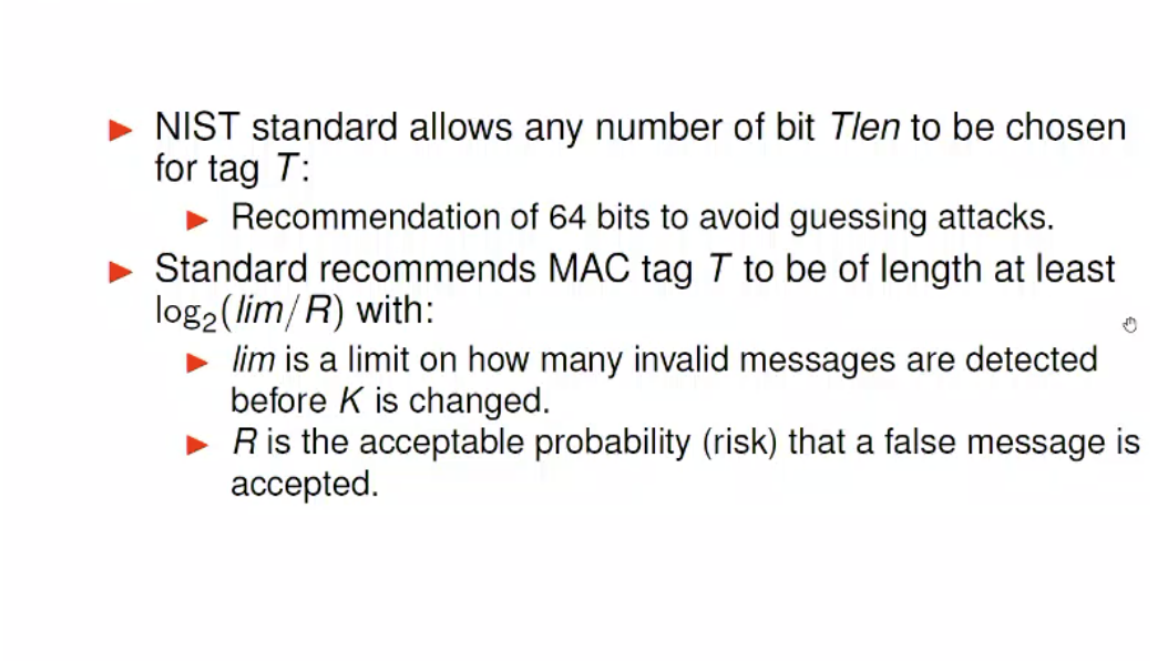


Figure 10: CMAC

Lecture Nine: Stream ciphers and Randomization

Randomness

- Defining randomness is difficult
- We want any specific string of bits to be as random as any other string
- Generators of random strings are used
- Using a TRNG to provide a seed for a PRNG

True Random number generator (TRNG)

- The entropy source includes:
  - A physical noise source
  - A digitalization process
  - Post-processing stages
- The output of the entropy source is any requested number of bits
- Periodic health test to ensure continuing reliable operation

- Intel introduces TRNG to Ivy Bridge processors in 2012

Pseudo random Number generator (PRNG)

- Each generator takes a seed as input
- It outputs a bit string before its state

These random number generators are made using a set of functions *outlined in lecture slides*

CTR_DRBG

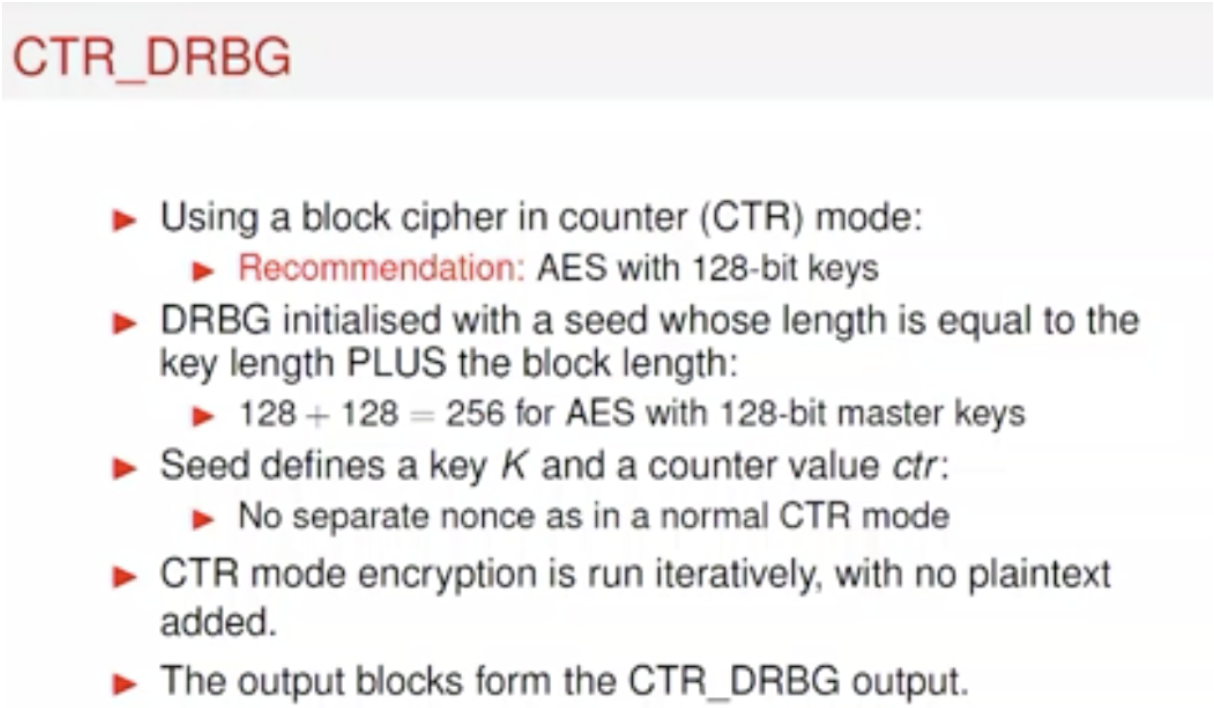


Figure 11: Example Implementation: CTR_DRBG

Update function:

- Each request DRBG generates up to 2^{19} bits
- From the function generate
- Updating provides backtracking resistance
- Restriction: up to 2^{48} seed bits
- Each re-seed provides forward prediction and backtracking resistance

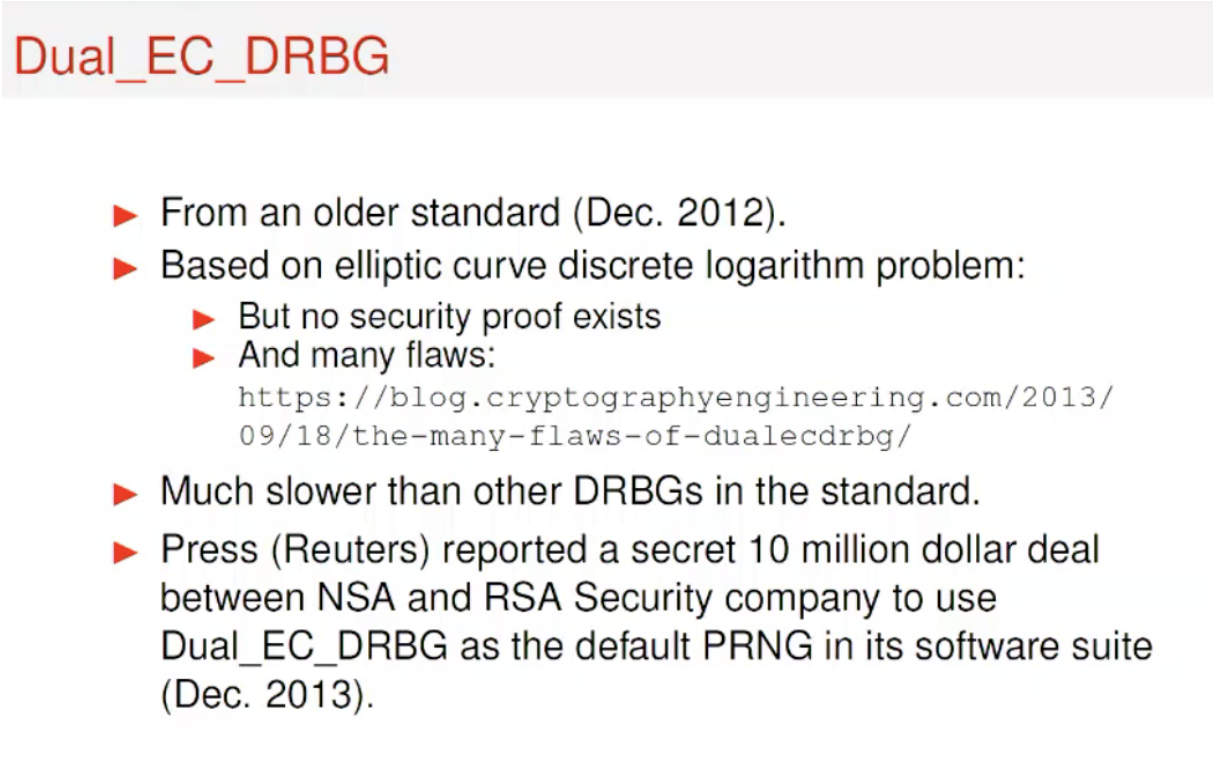


Figure 12: Duel_EC_DRBG

Stream Ciphers

- Characterised by the generation of a keystream using a short key and an initialisation value IV
- Synchronous stream ciphers
  - The keystream is generated independently of the plaintext
  - Both sender and receiver need to generate the same keystream and sync its usage
  - Vigenere cipher seen as a periodic synchronous stream sipher where each shift is defined by a key letter
  - CTR mode of operation for a block cipher is one method of generating a keystream
- Binary synchronous stream ciphers
  - Encryption: $C(t) = p(t) \bigoplus s(t)$
  - Decryption: $P(t) = C(t) \bigoplus s(t)$

One time pad

- Key is random sequence of characters, all of them are independently generated
- Each char in the key is used once

- Relies on perfect secrecy
- Is the only unbreakable cipher
- Practical usage is possible for pre-assigned communications between fixed parties
- How to deal with key management of completely random keys

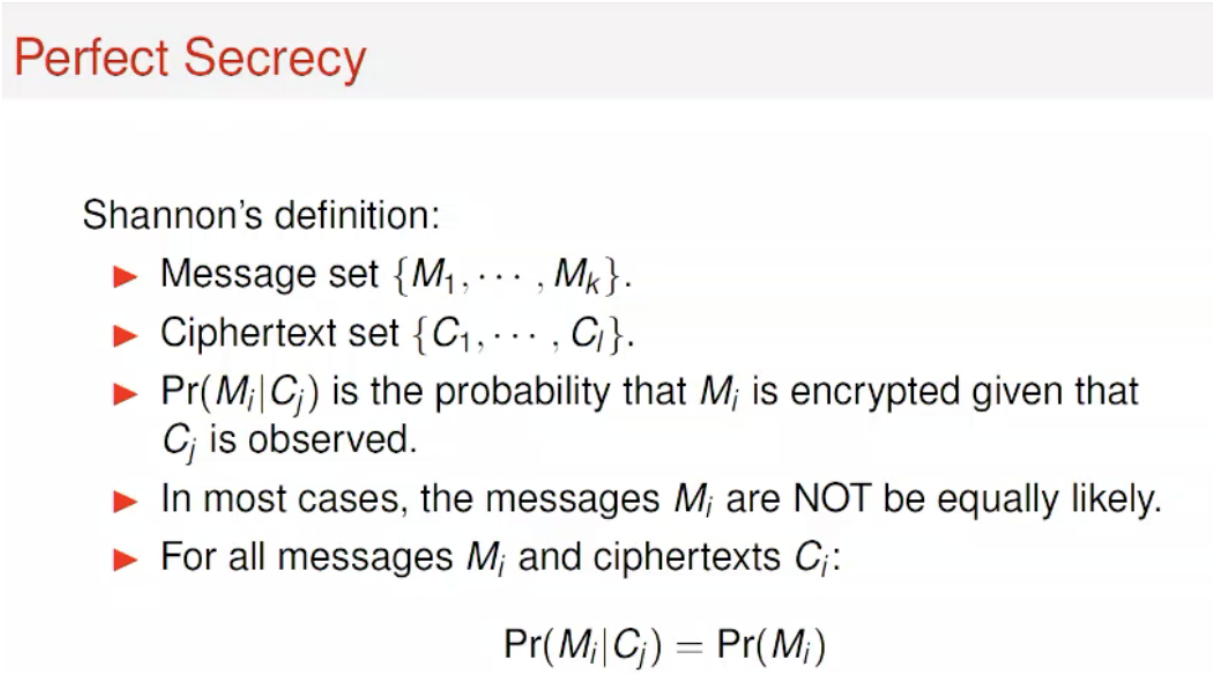
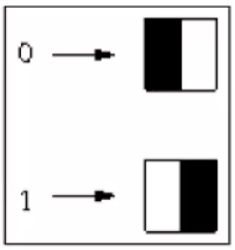


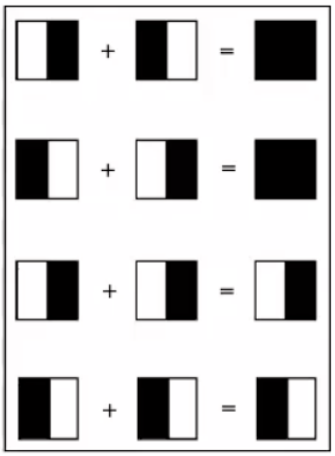
Figure 13: Perfect Secrecy

Visual Cryptography

- Application of one time pad: visual splits an image into two shares
- Decryption works by overlaying the two shared images
- Many generalisations are possible
- Each pixel is shared in a random way, similar to splitting a bit in the one time pad



- Generate a one time pad P (random bit string) with length equal to the number of pixels for the image I
- Generate a share $S_{I,1}$ by replacing each bit in P using the sub-pixel patterns shown on the left
- Generate the other share $S_{I,2}$ s.t.:
  - the same as $S_{I,1}$ for all the white pixels of I
  - the opposite of $S_{I,1}$ for all black pixels of I



- To reveal the hidden image I, $S_{I,1}$ and $S_{I,2}$ are overlayed
- Each black pixel of I is black in the overlay
- Each white pixel of I is half white in the overlay

###Lecture Twelve: Public Key Cryptography (Part One)

Motivation

- Public key cryptography (PKC) has features that symmetric key cryptography does not.
- Applied for key management in protocols such as TLS and IPsec.
- RSA is one of the best known public key cryptosystems, widely used.
- Alternatives include discrete log based ciphers, also widely deployed and standardised.

One-way Functions

- A function f is a one-way function if $f(x) = y$ is easily computed given x, but if $f^{-1}(y) = x$ is computationally difficult to compute given y
- Open problem: *Do one-way functions actually exist?*
- Example of functions believed to be one-way
  - Multiplication of large primes:

Trapdoor One-way Functions

- A trapdoor one-way function f is a one-way function s.t $f^{-1}(y)$ is easily computed given an additional piece of information called a *trapdoor*.

Asymmetric Cryptography

- Asymmetry: encryption and decryption keys are different
- Encryption key is public
- Decryption key is private
- Advantage: Key management is simple, can share public key to anyone, secret key still needs to be handled with care

RSA Algorithms

- Public key cryptosystem and digital signature scheme
  - Anyone can check if the signature is valid using public encryption key
- Based on the `integer factorisation problem`
- RSA patent expired in 2000

Key Generation:

- Randomly choose two distinct primes p, q from the set of all primes of a certain size
- Compute $n = pq$
- Randomly choose e s.t. $gcd(e, \phi(n)) = 1$
  - Here, $\phi(n) = \phi(pq) = (p-1)(q-1)$
- Compute $d = e^{-1} \mod \phi(n)$
- Set the public key K_E ad (n, e)
- Set the private key K_D as (p, q, d)

Encryption:

- Public encryption key $K_E = (n, e)$

- Input is a value M s.t. $0 \leq M \leq n$
- Compute $C = Enc(M, K_E)$

Decryption:

- Private decryption key is $K_D = (p, q, d)$
- Compute $Dec(C, K_D) = C^d \mod n$

Any message requires to be pre-processes to become M

- Coding it as a number
- Adding randomness

> This lecture is basically a recap of RSA, see last years lectures for more information

RSA Security

Attacks:

- Most of existing attacks avoided by using standardised padding mechanisms
  - Factorisation of the modulus n
  - Finding d from n and e
- An attacker factorises n into its prime factors p,q and thus recover d
- Breaking RSA is shown to be as hard as the RSA problem
  - It is unknown if RSA problem is as hard as factorising n
- **Quantum computing**:
  - If we get commercial quantum computing, by Shor's algorithm, we can break RSA in polynomial time

Lecture Thirteen: Public Key Cryptography (Part Two)

Diffie-Hellman Key Exchange

- Two users, Alice and bob, share a secret using only public communication.
- Public elements:
  - Large prime p
  - Generator $g \in \mathbb{Z}_p^*$
- Alice and bob each selects random values a and b respectively

- Alice and Bob both compue the secret key $Z = g^{ab}$

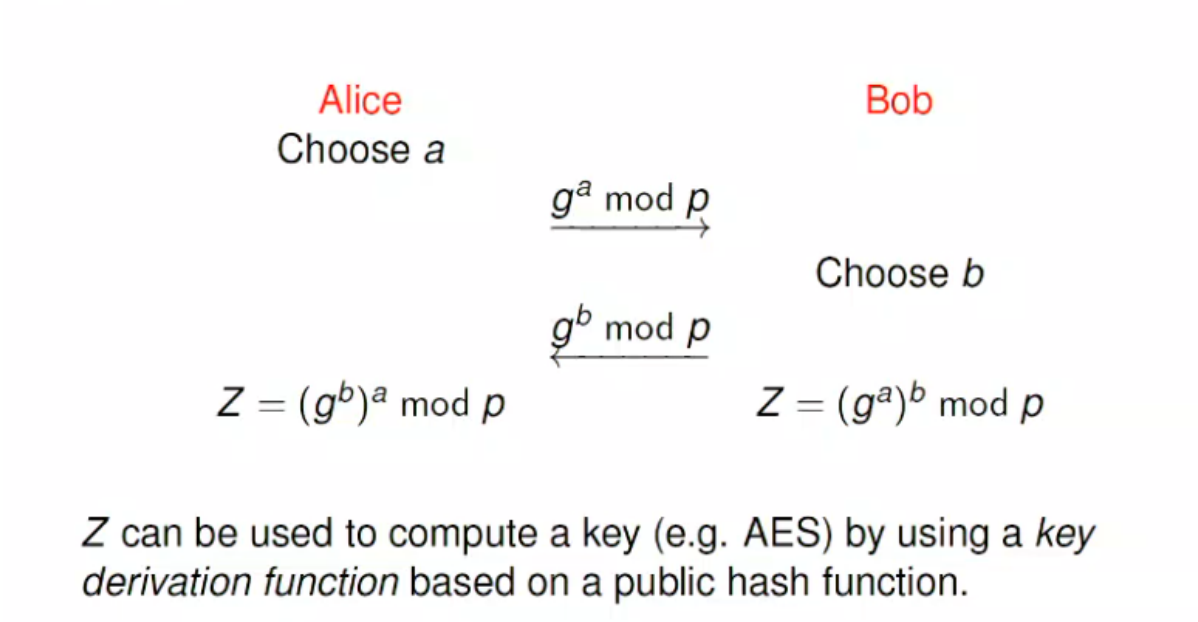
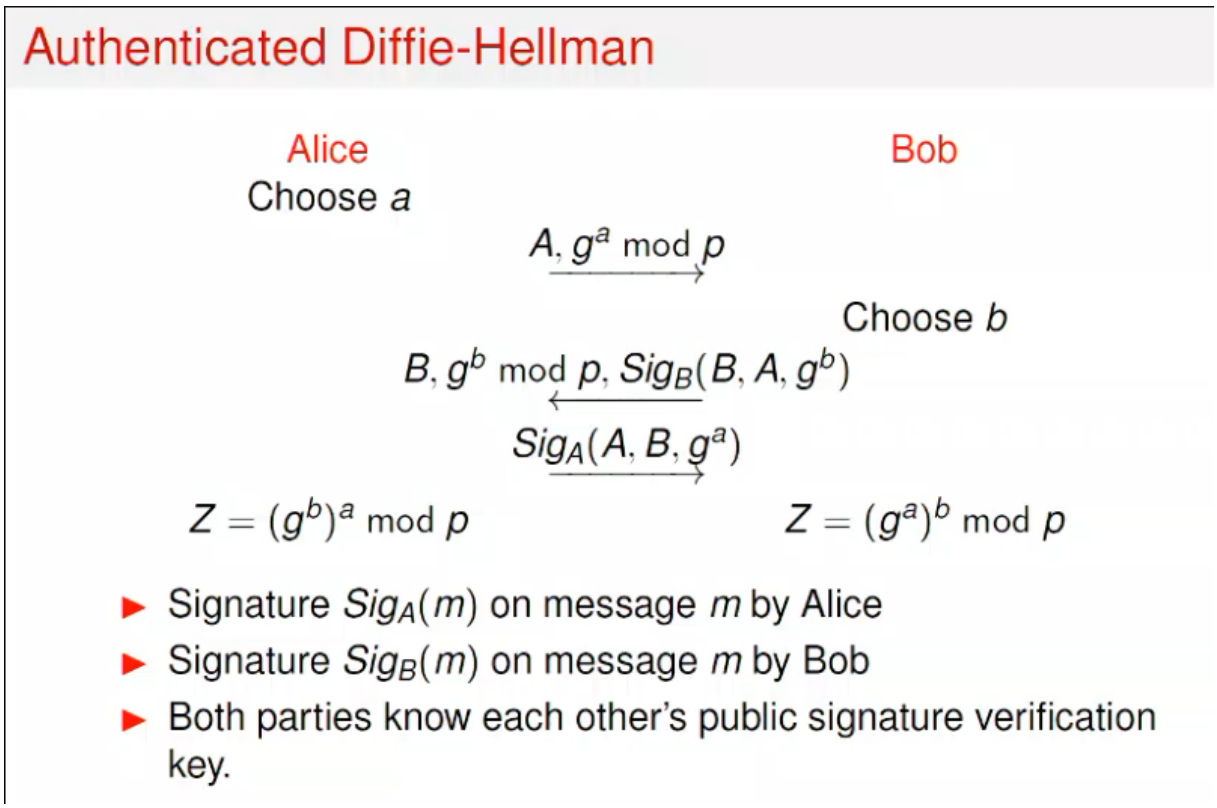


Figure 14: Diffie-Hellman Protocol

Here is the protocol using signatures to authenticate Diffie-Hellman:



Static and Ephemeral Diffie-Hellman

- The above protocol uses *ephemeral keys*
  - Key used once and then discarded
- In the *static* Protocol:
  - Alice chooses a long term private key X_A and a public key $Y_A = g^{X_B} \mod p$
  - Bob chooses a long term private key X_B and a public key $Y_B = g^{X_A} \mod p$
- Alice and bob find a shared secret key $S = g^{X_A X_B} \mod p$ that is static:
  - S stays the same until Alice and Bob change their public keys

Elgamal Cryptosystem

- Diffie-Hellman turned into a cryptosystem
- Key generation:
  - Select a prime p and a generator $g \in \mathbb{Z}_p^*$
  - Select a long term private key $K_D = x$ where $1 < x < p$
  - Compute $y = g^x \mod p$

- Set the long term public key as $K_E = (p, g, y)$

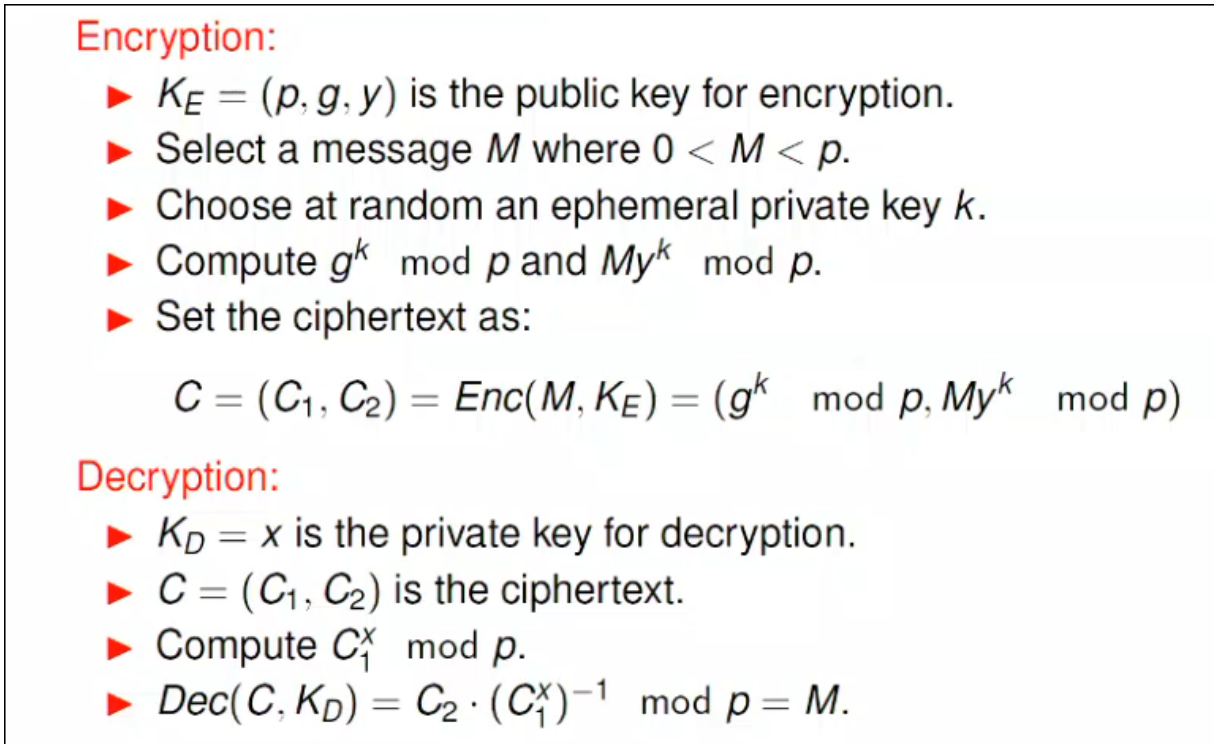


Figure 15: Elgamal Encryption and Decryption

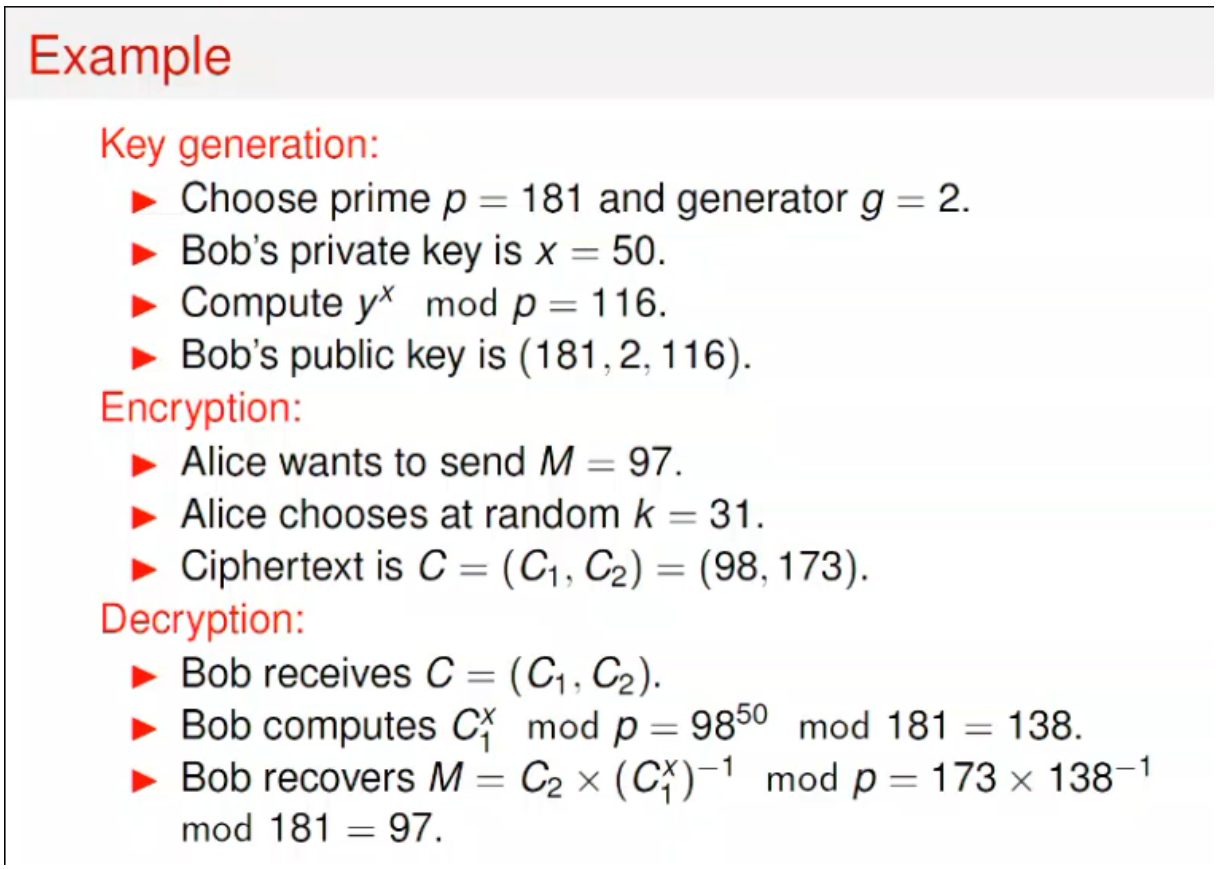


Figure 16: Worked Elgamal Example

Identity-Based Cryptography

- Extensively researched in the 2000's, with the use of elliptic curve *pairings*
- Public keys and certificates are not needed
  - Identity of the key owner replaces the public key
  - Message encryption using public parameters and recipients identity
- Limitation: Need of trusted key generation process
- Generalisation with functional cryptography
  - General access policies used to define who may decrypt the cipher text

Lecture Fourteen: Digital Signatures

Motivation behind signatures

- Digital signatures are one of them main benefits of public key cryptosystems

- In some countries, digital signatures are legally binding in the same way as handwritten signatures

Confidentiality and Authentication

- Message authentication codes (MAC) only allow for entity with shared secret to generate a valid tag:
  - Providing data integrity and data authentication
- Digital signatures use public key cryptography to provide properties of a MAC and more

Algorithms

- Key generation
  - Outputs two keys
    * Private *signing key* K_S
    * Public *verification key* K_V
- Signature Generation
- Signature Validation

Signature Generation Algorithm

- Inputs:
  - Alice private signing key K_S
  - Message M
- Output:
  - Signature $s = Sig(M, K_S)$
- Only alice the owner of K_S should be able to generate a valid signature
- The message should be any bit string
- The set of all signatures is usually a set of fixed size

Signature Validation

- Inputs:
  - Alice's public verification key K_V
  - Message M
  - Claimed signature s
- Outputs:

  - $Ver(M, s, K_V) = Boolean$

- Anyone should be able to verify the signature

Discrete Logarithm Signatures

- Security relying on difficulty of discrete logarithm property
- Three versions:
  1. Original Elgamal signatures
  2. Digital signature algorithm (DSA)
  3. DSA based on elliptic curve groups, known as ECDSA

Elgamal

Signature generation:

1. Alice selects a random k s.t. $gcd(k, p-1) = 1$ and computes

$$r = g^k \mod p$$

2. Alice solves $M = xr + ks \mod (p-1)$ for s by computing

$$s = k^{-1}(M - xr) \mod (p-1)$$

3. Alice outputs the tuple (M,r,s).

Signature verification:

- Bob checks if $g^M \equiv y^r r^s \mod p = ((g^x)^r (g^k)^s)$

Digital Signature Algorithm (DSA)

- Prime p is chosen s.t. $p-1$ has a prime divisor q of much smaller size (~256 bits)
- A generator g used in Elgamal signatures replaced by $g = h^{\frac{p-1}{q}} \mod p$
  - g has order q since $g^q \mod p = 1$
- Differences with Elgamal signatures:
  - Message is hashed using SHA hash algorithm
  - g is chosen to be the order of q

Elliptic Curve DSA (ECDSA)

- Parameters chosen from NIST approved curves

- Signature generation and verification are the same, except that:
  - q becomes the order of the elliptic curve group
  - multiplication mod p is replaced by the elliptic curve group operation
  - After operations on group elements, only the x coordinate is kept from the pair
- Signatures are generally not shorter than DSA at the same security level
  - Size varies with underlying curve

Lecture Fifteen: PKI and Certificates

Motivation

- Public key infrastructure implies the use of public digital certificates
- Digital signatures provide these certificates
- X.509 certificates are standardized and used in most network security applications

What is a public key infrastructure (PKI)

- A public key infrastructure is the key management environment for public key information of a public key cryptosystem - `NIST`
- Key management concerned with *lifecycle* of cryptographic keys
  - Generation, distribution, storage and destruction of keys

Digital certificates

- How to be confident of the correct binding between a public key and its owner?
  - When using a public key to encrypt a message or to verify a digital signature
- Achieved through the use of *digital certificates*
  - They contain the public key and the owner identity
  - There is other information such as signature algorithm and validity period
- Certificate digitally signed by a certification authority (CA):
  - CA should be trusted by the certificate verifier
- Certificates play a central role in key management for PKI's

Using a certificate

- Verifying a certificate

  - By checking that the CA's signature is valid
  - Check that any conditions set in the certificate are correct

- In order to verify the certificate:
  - The user of the certificate must have the correct public key of the CA
- It does not matter who obtains the certificate
- Public directories may store certificates

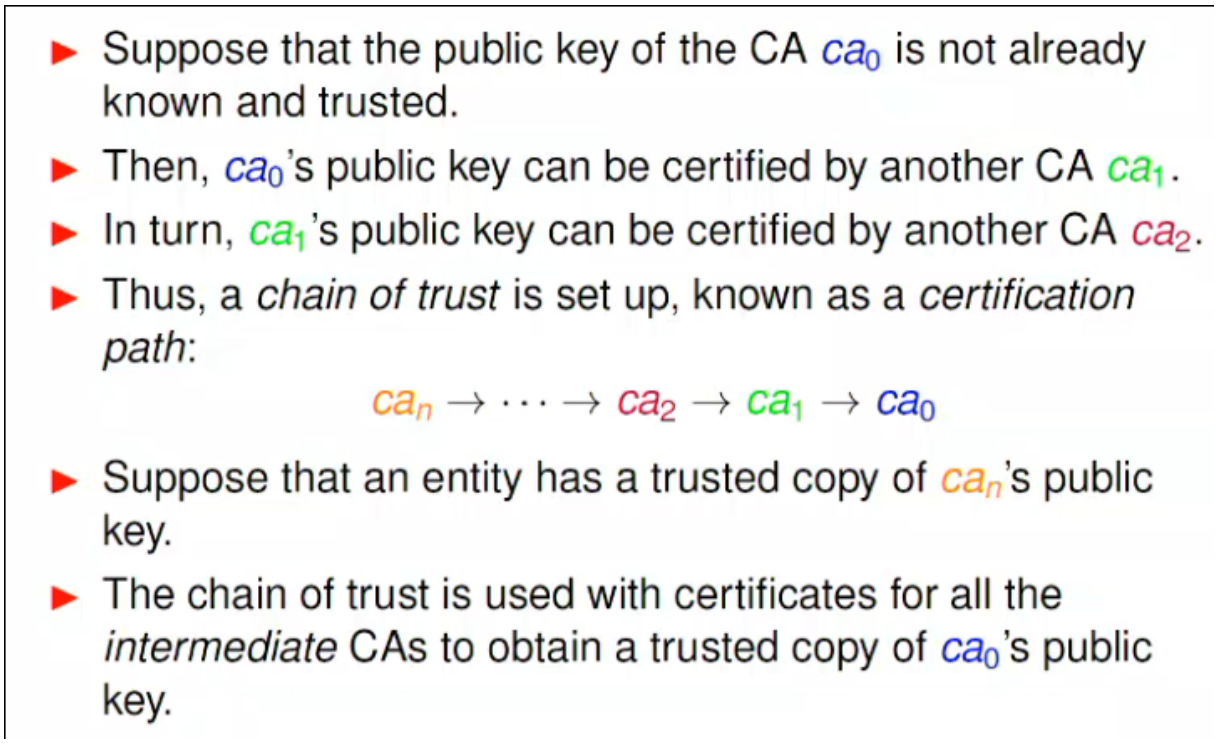


Figure 17: certification paths

Phishing Attack on Certificates

- The victim connects securely to a bogus site with the wrong certificate
- The attacker makes the URL similar and the interface identical to the genuine site
- Not always easy to tell if a certificate is one for a genuine site
  - Especially for the average user
- Certificates can be revoked from a website, and this is logged on the `Certificate Revocation List (CRL)`

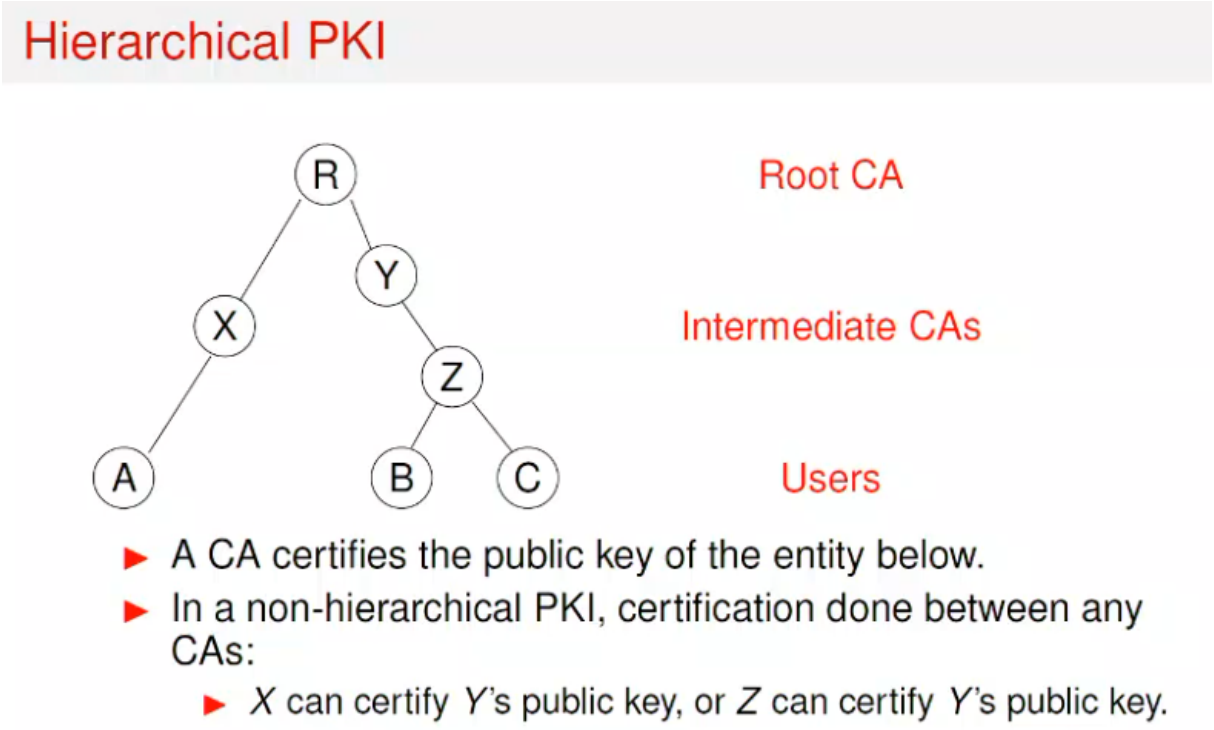


Figure 18: Heirarchical PKI

Browser PKI

- Multiple hierarchies with preloaded public keys as root CA's
- Intermediate CA's can be added
- Users can also add their own certificates
- Most servers send their public key and certificate to the browser at the start of a secure communication using TLS protocol

OpenPGP PKI

- Used in PGP email security
- Certificate includes ID, Public key, validity period and *self-signature*
- There is NO certification authorities
- Various key servers store keys
- Often known as *web of trust*

Lecture Sixteen: Key Establishment

Motivation

- Distribution of cryptographic keys to protect subsequent communication sessions
- Key establishment in TLS uses public keys to allow clients and servers to share a new communication key

Key management

- Critical aspect of any cryptographic system
- Phases:
  - Key generation: keys should be generated such that they are equally likely to occur
  - Key distribution: Keys should be distributed in a secure fashion
  - Key protection: Keys should be accessible for use in relevant algorithms, but not accessible to unauthorised parties
  - Key destruction: once a ley has performed its function, it should be destroyed s.t. it is of no value to an attacker

Key types

Keys are often organized in a hierarchy:

- **Long term keys**
  - Also called static keys
  - Intended to be used for a long time
  - depending upon the application, from a few hours to a few years
  - Used to protect distribution of session keys
- **Short term keys**
  - Also called session keys
  - Intended to be used for a short time
  - depending upon the application, from a few seconds to a few hours
  - Used to protect communications in a session

Key Distribution Security

- In practice, session keys are symmetric keys used with ciphers
- Long term keys can be either symmetric or asymmetric keys depending on how they are used
- How to establish secret session keys among communicating parties using the long term keys
  - Common approaches:
    * Key pre-distribution
    * Using an online server with symmetric long term keys
    * Using asymmetric long term keys

- **Goals of key distribution**
  - Authentication
  - Confidentiality

Mutual and Unilateral Authentication

- If both parties achieve the authentication goal, then the protocol provides *mutual authentication*
- If only one party achieves it, then the protocol provides *unilateral authentication*
- Many real world key establishment protocols achieve only unilateral authentication
  - Typically, clients can authenticate servers.
  - Client authentication often happens later, protected with the establishment key

Adversary Capabilities

Let a strong adversary know the details of the cryptosystem algorithms involved and be able to:

- Eavesdrop on all messages sent in a protocol
- Alter all messages sent in a protocol using any information available to him/her
- Re-route any messages to any other party
- Obtain the value of previous session keys used to run the protocol

Key distribution using symmetric keys

- Key distribution through an online server
- The TA shares a long-term shared key with each user
- An online TA generates and distributes session keys to users when requested

Key distribution using asymmetric keys

- No online TA is required
- Public keys used for authentication
- Public keys managed by PKI
- Users are trusted to generate good session keys

Forward secrecy

What happens when a long term key is compromised?

- The attacker can now act as the owner of the long term key
- Previous session keys may also be compromised
  - This can be the case with key transport
  - This can be prevented with key agreement

A protocol provides *(perfect) forward secrecy* if compromise of long term secret keys does not reveal session keys previously agreed using those long-term keys

Key transport

- User chooses key material and sends it encrypted to another party
  - Sometimes, the message is also signed by the sender
- TLS includes options for key transport
- Not providing *forward secrecy*

Examples of session key distribution Timestamp (24:11)