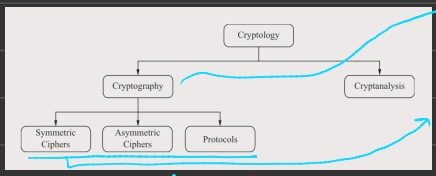
Chapter 1. Intro to Cryptography and Data Security

eg Ancient Egypt, Greece, Romans, Enigma Machine

→ Cryptography involved in the protection of digital info against misuse (Cybersecurity) → Cryptographic algorithms are needed for secure digital systems

→ Field of Cryptography.



→ Cryptography: Science of securing comm. against adversary and for other security goals such as the integrity and authenticity of messages
↳ Main Goal: Hide meaning of a message

→ Cryptoanalysis: Science/Art of breaking cryptosystems respectfully → Tests security

→ Symmetric Ciphers: 2 parties have an encrypt. and decrypt. method they share a secret key

→ Anti-Symmetric/Public-Key Ciphers: 2 keys exist. User keeps a symmetric and public key
↳ can be used for digital signatures

→ Cryptographic Protocols: Realize more complex security functions through crypt. alg (e.g. Transport Layer Security (TLS) on WEB)

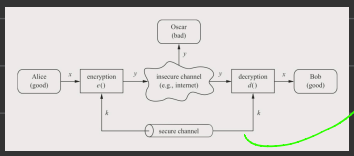
→ Hybrid Schemes: Symmetric + Anti-Symmetric + Hash Functions

→ Insecure Channel: Internet, comm media, Air Drop

Symmetric Cryptography (a.k.a. Symmetric-key, Secret-key, single-key)

→ Intro Ex: Alice and Bob want to comm. over insecure channel. Oscar (Bad Guy) hacks channel to eavesdrop.

→ Symmetric Cryptography Sol'n: Alice encrypts her message x using symmetric algorithm yielding ciphertext y. Bob receives y and decrypts y to read x.



* k = key → Key space: Set of all possible keys

Secure Channel: In-Person, Encrypted DM Apps, LAN

→ IMPORTANT: Encryption + Decryption Algs. are publically known → only way to see if they're good

× bypasses Oscar if alg is good.

Simple Symmetric Cipher: Substitution Cipher

→ Goal: Encryption of text by subst. each letter w/ another character (Assume random)
↳ NOT SECURE → size of Key space is about 2^{88} or exactly 26!

Key of Subst. Cipher

PlainText	Cipher
A	K
B	d
C	U
⋮	⋮

→ Types of Attacks for Subst Cipher: Brute Force, Letter Freq.

Brute-Force / Exhaustive Key Search

→ Treat cipher as black box → Oscar has short piece of plaintext and decrypts first piece of ciphertext w/ all possible keys → if plaintext = resulting plain ⇒ correct key

→ Basic BFA/EKS: Let (x,y) denote pair of plain and cipher text. $k = \{k_1, \dots, k_K\}$ is key space.
BFA checks $\exists k_i \in K \; (d_{k_i}(y) = x)$

→ Complicated, slow but always possible.

Letter Frequency Analysis:

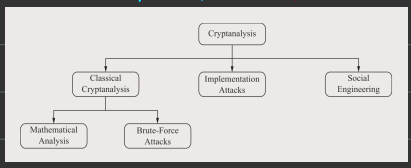
→ Flaw of Subst Cipher: Mapping is the same for all symbols → Easy to see patterns through letter frequency

1. Determine frequency of every ciphertext character → Find patterns and encrypt ciphertext

Cryptoanalysis:

→ For a secure cryptosystem, it is important to use sound cryptographic algorithms/protocols and their correct implementations

→ Classical Cryptoanalysis: Attempts to break ciphers by analyzing inputs and outputs



→ Main Classes of Attacks:
↳ Ciphertext-only Attack: Adversary has only access to ciphertext
↳ Known-plaintext Attack: Adversary has ciphertext and some pieces of plaintext
↳ Chosen-plaintext Attack: Adversary can choose plaintext being encrypted and corresponding ciphertext when they have access to decryption device → Goal: Find out key
↳ Chosen-ciphertext Attack: Adversary chooses ciphertext and also obtains corresponding plaintext

→ Implementation Attacks: Side-channel attacks used to extract secret key by observing behaviour of cryptographic implementation (integrated circuit or piece of sw)
↳ Ex: Heat Analysis, Electromagnetic power consumption, timing measurement, cache access patterns

→ Social Engineering Attacks: Social Forceful tactics like bribe, blackmail to get key → can get violent and scammy

→ IMPORTANT: Attacker always looks for weakest link in system. Always choose strong alg. and make sure types of implementation & Social Engineering Attacks are not feasible → adhere to Kerckhoff's principle

→ Kerckhoffs' Principle: Cryptosystem should be secure even if adversary knows all details (except for secret key) a.k.a encryption & decryption algorithms

→ 2 aspects to think bout when determining length of key for ciphers:
1. Large key space doesn't help at all if there is an analytical attack that works
2. Key lengths for SC and ASC is different

Table 1.2 Estimated time for successful brute-force attacks on symmetric cipher with different key lengths

Key length	Security estimation
56–64 bits	short term: a few hours or days
112–128 bits	long term: several decades in the absence of quantum computers
256 bits	long term: several decades, even with quantum computers that run the currently known quantum computing algorithms

Modular Arithmetic and Historic Ciphers:

Ex. Consider hours on a clock. $1h, 2h, 3h, \ldots, 12h, 1h, 2h, 3h, \ldots$

→ Almost all cryptographic algorithms are based on arithmetic within finite number of elements ← Ex. $A = \{0, 1, 2, \ldots, 9\}$ We can do any arithmetic as long as result is in set

→ Modulo Operation: Let $a, r \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$ $\quad a \equiv r \bmod m$ if $m \mid a - r$ for $0 \le r < m$ $\quad a = q \cdot m + r$ for some $q \in \mathbb{Z}$

→ Remainder is not unique: For any given $a \in \mathbb{Z}$ and $m \in \mathbb{Z}^+$, there are infinitely many valid remainders.

Ex. Reduce $12 \,\%\, 9$

1. $12 \equiv 3 \bmod 9$ because $9 \mid 12 - 3$
2. $12 \equiv 21 \bmod 9$ because $9 \mid 12 - 21$
3. $12 \equiv -6 \bmod 9$ because $9 \mid 12 + 6$

→ Equivalence Class for mod 9: $\{\ldots, -24, -15, -6, 3, 12, 21, 30, \ldots\}$

There are 9 total equivalence classes for mod 9

$$\{\ldots, -27, -18, -9, 0, 9, 18, 27, \ldots\}$$
$$\{\ldots, -26, -17, -8, 1, 10, 19, 28, \ldots\}$$
$$\vdots$$
$$\{\ldots, -19, -10, -1, 8, 17, 26, 35, \ldots\}$$

→ For given modulus m, it doesn't matter which element from a class we choose for a computation

↳ Fixed modulus → Choose class element resulting in easiest computation

Ex: Core operation in many asymmetric schemes is an exponentiation of form $x^e \bmod m$ where x, e, m are large $\mathbb{Z}$, say 2048 bits

Demonstrate 2 ways of modular exponentiation.

$3^8 \bmod 7$

Method 1: Naive Method: $3^8 = 6561 \equiv 2 \bmod 7$ since $6561 = 937 \cdot 7 + 2$

Smarter Method: Perform 2 partial exponentiations. $3^8 = 3^4 \cdot 3^4 = 81 \cdot 81$

We can now replace 81 by another member of same equivalence class $\quad 81 \equiv 4 \bmod 7$

$\therefore 3^8 = 81 \cdot 81 \equiv 4 \cdot 4 = \boxed{16 \equiv 2 \bmod 7}$

IMPORTANT: Always of computational advantage to apply modulo reduction as soon as possible.

As long as $r \in [0, m)$ it doesn't matter which equivalence class we choose

→ Integer Ring Def^n: The Integer Ring $\mathbb{Z}_m$ consists of

1. Set $\mathbb{Z}_m = \{0, 1, 2, \ldots, m-1\}$
2. 2 operations "$\cdot$" and "$+$" $\forall a, b \in \mathbb{Z}_m$ s.t.:

$a + b \equiv c \bmod m \quad (c \in \mathbb{Z}_m)$

$ab \equiv c \bmod m \quad (c \in \mathbb{Z}_m)$

Ex: $m = 9 \Rightarrow \mathbb{Z}_m = \{0, 1, \ldots, 8\}$

$6 + 8 = 14 \equiv 5 \bmod 9$

$6 \cdot 8 = 48 \equiv 3 \bmod 9$

→ Properties of Integer Rings:

1. Closed under addition and multiplication
2. Addition and Multiplication are associative: $\forall a, b, c \in \mathbb{Z}_m \; (a + (b + c) = (a + b) + c \;\wedge\; a(bc) = (ab)c)$
3. Addition is commutative: $\forall a, b \in \mathbb{Z}_m \; (a + b = b + a)$
4. 0 is the neutral element w.r.t. addition: $\forall a \in \mathbb{Z}_m \; (a + 0 \equiv a \bmod m)$
5. For any $a \in \mathbb{Z}_m$, there is always a negative $-a$ s.t. $a + (-a) \equiv 0 \bmod m$ (Additive Inverse)
6. 1 is the neutral element w.r.t. multiplication $\forall a \in \mathbb{Z}_m \; (a \cdot 1 \equiv a \bmod m)$
7. Multiplicative inverse only exists for some elements: Let $a \in \mathbb{Z}$. a^{-1} defined s.t. $a \cdot a^{-1} \equiv 1 \bmod m$

If inverse exists, we can divide a since $b/a \equiv b \cdot a^{-1} \bmod m$

8. Distributive Law: $\forall a, b, c \in \mathbb{Z}_m \; (a \cdot (b + c) = ab + ac)$

→ Element $a \in \mathbb{Z}_m$ has inverse $a^{-1} \equiv \gcd(a, m) = 1 \equiv a$ and m are coprime

Shift / Caesar Cipher:

→ Case of Subst. Cipher: Shift every plaintext letter by fixed # of positions in the alphabet ← can also be a rotation cipher

Both plaintext and ciphertext letters are part of $\mathbb{Z}_{26}$

Table 1.3 Encoding of letters for the shift cipher

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

→ Shift Cipher Def^n: Let $x, y, k \in \mathbb{Z}_{26}$

Encryption: $e_k(x) \equiv x + k \bmod 26$

Decryption: $d_k(x) \equiv x - k \bmod 26$

→ 2 types of attacks: BFA and LFA

Affine Cipher:

→ Improvement of Shift Cipher: Encrypts by multiplying plaintext by 1 part of key followed by addition of another part of key

→ Affine Cipher Def^n: Let $x, y, a, b \in \mathbb{Z}_{26}$

Encryption: $e_k(x) = y \equiv a \cdot x + b \bmod 26$

Decryption: $d_k(y) = x \equiv a^{-1} \cdot (y - 1) \bmod 26$

Key: $k = (a, b)$ which has restriction $\gcd(a, 26) = 1$

Can be broken w/ LFA