Introduction: Context and Basic Notions

Cryptology (1)

Martin Stanek

2025

KI FMFI UK Bratislava

Introduction

- Cryptology = cryptography & cryptanalysis
  - cryptography: constructing algorithms, schemes, and protocols
  - cryptanalysis: attacking these construction, analyzing their security
- security in the presence of an adversary
  - security means … (requirements in particular context/application)
  - adversary means … (capabilities of an attacker)
- some requirements and related cryptographic constructions
  - confidentiality $\mapsto$ encryption
  - integrity/authenticity $\mapsto$ hash functions, MAC, digital signatures
  - authentication $\mapsto$ protocols
  - non-repudiation $\mapsto$ digital signatures
  - other requirements: privacy, anonymity, etc.

- **confidentiality** – Preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- **integrity** – Guarding against improper information modification or destruction, and includes ensuring information non-repudiation and authenticity.
- **authenticity** – The property of being genuine and being able to be verified and trusted; confidence in the validity of a transmission, message, or message originator.
- **non-repudiation** – Protection against an individual who falsely denies having performed a certain action and provides the capability to determine whether an individual took a certain action, such as creating information, sending a message, approving information, or receiving a message.

source: NIST SP 800-53 Rev. 5, 2020

Relation to Cybersecurity

- cryptography $\subset$ cybersecurity $\subset$ information security
- important part of cybersecurity, provides essential tools and techniques
- cryptography is not an answer to all security needs:
  - availability (redundancy),
  - secure software (software engineering, security testing), etc.
- cryptography is often useless without other security measures
  - key management, access control, risk assessment, personnel security, information classification, etc.

This course: cryptographic constructions and their security

Encryption

- traditional cryptographic technique
- intuitively, we know what encryption is
  - transforming data so that an unauthorized subject is unable to read it
  - probably using some sort of secret key
- encryption provides confidentiality (prevents data compromise) for
  - communicated data – when an attacker eavesdrops
    SSL/TLS, WPA2/WPA3, S/MIME, …
  - stored data – when an attacker gets access to storage media
    BitLocker, VeraCrypt, FileVault, …
- informally: encryption + decryption ~ encryption scheme ~ cipher

Encryption – basic terminology

- original data ∼ plaintext
- data after encryption ∼ ciphertext
- finite sets of all plaintexts P, ciphertexts C, and keys K
- symmetric (secret key) encryption scheme:
  - key generation; usually a random bit string in modern ciphers
  - encryption: $E : K \times P \to C$ (might be probabilistic)
  - decryption: $D : K \times C \to P$
- sometimes more complicated by using various modes of encryption, randomization, ...

Encryption – what we want

- correctness: $\forall k \in K\ \forall p \in P : D_k(E_k(p)) = p$
  - probabilistic encryption: $\forall k \in K\ \forall p \in P\ \forall c \leftarrow E_k(p) : D_k(c) = p$
- efficiency: encryption and decryption should as fast as possible
  - reasonable speed depends on application, computational resources, etc.
- security – difficult to define precisely
  - usually “resistance to all known attacks”
- identity is correct and efficient but completely insecure
- security vs. efficiency trade off

Example 1 – Shift cipher (Caesar cipher)

- alphabet $A = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$
- natural mapping between characters and numbers: A $\leftrightarrow$ 0, B $\leftrightarrow$ 1, …, Z $\leftrightarrow$ 25
- plaintexts and ciphertexts: $P = C = A$
- keys: $K = \mathbb{Z}_{26}$
- encryption: $E_k(p) = (p + k) \bmod 26$
- decryption: $D_k(c) = (c - k) \bmod 26$
- correctness follows from using inverse operation in decryption; $(\mathbb{Z}_{26}, +)$ is a group:

$$D_k(E_k(p)) = ((p + k) - k) \bmod 26 = p, \quad \text{for any } p, k \in \mathbb{Z}_{26}$$

Example 1 – Shift cipher (Caesar cipher) – remarks

- plaintext longer than single character?
  - using cipher in a *mode*, e.g., encrypt each individual character separately
- Julius Caesar used $k = -3$ in his private correspondence
  - regardless of cipher security, fixed key is a security risk

Security

- none in any reasonable context
  - reasonable: encrypting natural text of nontrivial length
- the main problem: small key space, only 26 keys
  - all keys can be tested (brute force attack)
  - How easy is to recognize a plaintext?

Brute force attack
$\Rightarrow |K|$ must be large !

Example 2 – Simple substitution cipher

- alphabet $A = \{\mathrm{A}, \mathrm{B}, \ldots, \mathrm{Z}\}$
- plaintexts and ciphertexts: $P = C = A$
- keys: $K = \{\pi \mid \pi \text{ is a permutation on } A\}$
- encryption: $E_\pi(p) = \pi(p)$
- decryption: $D_\pi(c) = \pi^{-1}(c)$
- trivially correct
- long plaintext – encrypt each character individually

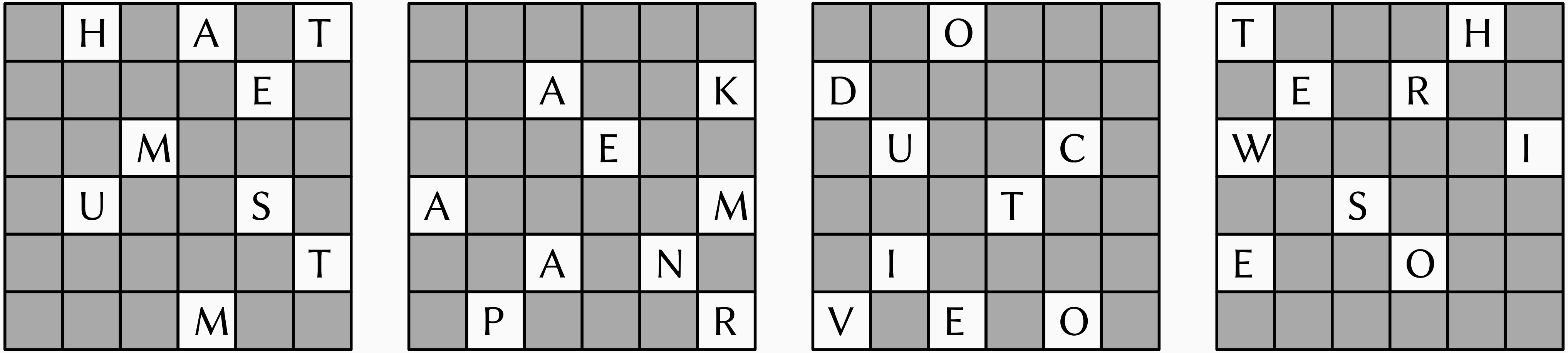
- large number of keys: $|K| = 26! \approx 2^{88.38}$
  - brute force does not work
- easily broken by frequency and/or pattern analysis
  - see the next lecture
  - E.A. Poe: The Gold-Bug (1843)
- various variants/improvements exist
  - multiple (polyalphabetic) substitutions
  - frequent letters to multiple targets, … homophonic substitutions

Example 3 – Permutation cipher

- $P = C = A^n, K = \{\pi \mid \pi \text{ is a permutation on } \mathbb{Z}_n\}$
- encryption: $E_\pi(p_0 p_1 \ldots p_{n-1}) = p_{\pi(0)} p_{\pi(1)} \ldots p_{\pi(n-1)}$
- decryption: $D_\pi(c_0 c_1 \ldots c_{n-1}) = c_{\pi^{-1}(0)} c_{\pi^{-1}(1)} \ldots c_{\pi^{-1}(n-1)}$
- trivially correct
- long plaintext can be divided into separate blocks of length n
- key space size: $|K| = n!$
- cryptanalysis
  - frequency analysis of digrams/trigrams for various key lengths and parts of π
- various variants of permutation cipher exist

Example 4 – Fleissner/Cardano Grille

- $2n \times 2n$ square with n^2 perforations
  - exactly one position chosen for perforation from each quadruple of rotational-symmetric positions
  - key: positions of perforations, i.e., the key space size is 4^{n^2}
- encryption: using perforations to write the plaintext
  - rotating the square by 90° when needed, fill unused space with suitable text
- decryption: rotate the square and read the text
- long plaintext divided into blocks of length $4n^2$



Hate must make a man productive.
Otherwise one might as well love.

Karl Kraus

T	H	O	A	H	T
D	E	A	R	E	K
W	U	M	E	C	I
A	U	S	T	S	M
E	I	A	O	N	T
V	P	E	M	O	R

Security of an encryption scheme

- robust security definition is a nontrivial task
- What is the goal of an attacker?
  - Find the key … what about identity?
  - Find the plaintext from the ciphertext … what about half of the plaintext?
  - Find at least one bit/character of the plaintext from the ciphertext … function?
  - Compute any nontrivial function of the plaintext?
- What capabilities are available to the attacker?
  - attack scenarios … *see later in this lecture*

Perfect secrecy

- plaintext, ciphertext, and key as random variables ($\boldsymbol{P}, \boldsymbol{C}, \boldsymbol{K}$)
  - $\boldsymbol{P}$: (a priori) probability distribution of plaintexts
    - e.g. *tomorrow* is more probable than *mjuuwerq*
    - we don't need to know the distribution of $\boldsymbol{P}$
  - $\boldsymbol{K}$ depends on key generation algorithm (often uniform)
  - $\boldsymbol{C}$ depends on encryption algorithm, $\boldsymbol{P}$, and $\boldsymbol{K}$

(Shannon) An encryption scheme is **perfectly secure**, if for any $p \in P$ and $c \in C$ such that $\Pr[\boldsymbol{C} = c] > 0$: $\Pr[\boldsymbol{P} = p \mid \boldsymbol{C} = c] = \Pr[\boldsymbol{P} = p]$.

- knowing a ciphertext does not change the probability distribution of the plaintexts
- an eavesdropper learns nothing from the ciphertext

Perfect secrecy (2)

- observation: $|K| \geq |P|$ for any perfectly secure encryption scheme
- information-theoretic security (arbitrary strong attacker)
- limitations:
  - single ciphertext attack
  - no additional information about the plaintext
- an equivalent definition (encryptions of plaintexts are indistinguishable):

An encryption scheme is **perfectly secure**, if for all $p_0, p_1 \in P$ and any $c \in C$ such that $\Pr[\boldsymbol{C} = c] > 0$: $\Pr[\boldsymbol{C} = c \mid \boldsymbol{P} = p_0] = \Pr[\boldsymbol{C} = c \mid \boldsymbol{P} = p_1]$.

Vernam cipher (one-time pad)

- $P = C = K = \{0,1\}^n$, for $n \in \mathbb{N}$
- encryption: $E_k(p) = p \oplus k$, where $\oplus$ denotes a bitwise XOR
- decryption: $D_k(c) = c \oplus k$
- correctness: $D_k(E_k(p)) = (p \oplus k) \oplus k = p \oplus (k \oplus k) = p$
- perfectly secure if
  1. keys are random with uniform distribution
  2. keys are not reused (new key is generated for each plaintext)
- intuition: given a ciphertext c, can some p' be the corresponding plaintext?
  - sure, if $k' = c \oplus p'$ is used as the key

Perfect secrecy of one-time pad

– for any $p \in P$ and $c \in C$ (where $\Pr[\boldsymbol{C} = c] > 0$):

$$\begin{aligned}
\Pr[\boldsymbol{P} = p \mid \boldsymbol{C} = c] &= \frac{\Pr[\boldsymbol{P} = p \cap \boldsymbol{C} = c]}{\Pr[\boldsymbol{C} = c]} = \frac{\Pr[\boldsymbol{C} = c \mid \boldsymbol{P} = p] \cdot \Pr[\boldsymbol{P} = p]}{\Pr[\boldsymbol{C} = c]} \\
&= \frac{\Pr[\boldsymbol{K} = (p \oplus c)] \cdot \Pr[\boldsymbol{P} = p]}{\sum_{k \in K} \Pr[\boldsymbol{K} = k] \cdot \Pr[\boldsymbol{C} = c \mid \boldsymbol{K} = k]} \\
&= \frac{2^{-n} \cdot \Pr[\boldsymbol{P} = p]}{2^{-n} \cdot \sum_{k \in K} \Pr[\boldsymbol{C} = c \mid \boldsymbol{K} = k]} \\
&= \frac{\Pr[\boldsymbol{P} = p]}{\sum_{k \in K} \Pr[\boldsymbol{P} = (c \oplus k)]} = \frac{\Pr[\boldsymbol{P} = p]}{\sum_{p' \in P} \Pr[\boldsymbol{P} = p']} = \Pr[\boldsymbol{P} = p]
\end{aligned}$$

Vernam cipher (one-time pad) – remarks

- keys with nonuniform distribution:
  - change the probability distribution of plaintexts (after observing the ciphertext)
- reusing keys:
  - let $c_1 = p_1 \oplus k, c_2 = p_2 \oplus k$
  - then $c_1 \oplus c_2 = p_1 \oplus p_2$ (XOR of two plaintexts)
  - can be solved for common texts (languages) … two time pad problem
- disadvantage: |key| = |plaintext|
  - consider key distribution in advance (e.g. physical storage media)
  - shorter key ⇒ sacrifice of perfect secrecy

Modern symmetric ciphers

- designed for efficient hardware and software implementations
  - operate on bit vectors
- cannot have the perfect secrecy property, since $|\text{key}| < |\text{plaintext}|$
- block ciphers: $E, D : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$
  - encryption and decryption algorithms are defined over bit vectors of fixed length
  - AES (block size: 128 bits, key length: 128/192/256 bits)
- stream ciphers:
  - key and an initialization vector (nonce)
  - finite state deterministic generator producing (pseudo-random) keystream
  - ChaCha20 (key length: 256 bits, nonce length: 96 bits)
  - block ciphers in specific modes of operation

Asymmetric (public key) encryption schemes

- each user generates his/her own instance
- based on intractable mathematical problems
  - factoring, discrete logarithm, learning with errors, etc.
- three algorithms (Gen, Enc, Dec):
  - Gen: public key pk, secret (private) key sk
  - encryption: $\mathrm{Enc}_{\mathrm{pk}}(m) = c$
  - decryption: $\mathrm{Dec}_{\mathrm{sk}}(c) = m$
- public key for encryption (everyone can encrypt)
- secret (private) key for decryption, only the owner can decrypt
- correctness: $\forall(\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Gen}() \;\; \forall m : \mathrm{Dec}_{\mathrm{sk}}\big(\mathrm{Enc}_{\mathrm{pk}}(m)\big) = m$

Kerckhoffs's principle

Auguste Kerckhoffs: "*A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.*" (19th century)

- the security should not rely on secret algorithms
- replacing (HW or SW) implementation is costly/impossible
- a recent failure: TETRA:BURST
  - ETSI TETRA (European Telecommunications Standards Institute)
  - Terrestrial Trunked Radio public standard, some secret cryptography (20+ years)
  - widely used by police, military, and intelligence
  - reverse engineered, several vulnerabilities found (Midnight Blue, 2023)
- protecting the design of a cryptosystem is sometimes used
  - but again, the security should not depend on it

Attack scenarios – ciphers

- COA – Ciphertext only attack
  - attacker gets some ciphertexts
  - eavesdropping, theft, …
- KPA – Known plaintext attack
  - attacker knows some plaintext and ciphertext pairs
  - headers in files, data structures, opening/closing sentences, …
- CPA – Chosen plaintext attack
  - attacker can (adaptively) choose plaintexts and obtain their encryption
  - always possible with asymmetric schemes
- CCA – Chosen ciphertext attack
  - attacker can (adaptively) choose ciphertexts and obtain their decryption

- We know neither the environment nor the operational conditions of an encryption scheme ⇒ use the strongest possible scheme (with respect to an attack scenario).

Example – Attack scenarios vs. Simple substitution cipher

- COA: frequency/patterns analysis
- KPA: reveals values in π for all symbols appearing in the plaintext
- CPA: chosen plaintext “`ABCDE...XYZ`”
- CCA: similar to CPA (the attack cannot be improved further)
- similarly for shift cipher, and other simple ciphers

Key length (symmetric schemes)

- generic attack: exhaustive search of the key space (brute-force)
- large key space: necessary but not sufficient requirement for security
- example of a brute-force attack (what key space is covered):

time key	length (bits)
1 minute	34.4
1 hour	40.3
1 day	44.9
1 month	49.8
1 year	53.4

- ≈ 380 mil. AES-128 operations/s (Intel Core Ultra 5 125U, HW accelerated AES)
- `/usr/bin/openssl speed -multi 8 -bytes 16 -evp aes-128-ecb`
- better CPUs, GPUs, ASICs, and more parallelism improve results (but not much), 2^{128} is infeasible

Modern cryptology

- emphasis on formal security definitions and proofs
- precise formulation of assumptions
  - attacker’s capabilities
  - hardness of computational problems
  - properties of underlying primitives

How cryptography fails

- common real-world security problems related to cryptography:
  - bad randomness source for generation of keys
  - insufficient checking of public-key certificates
  - incorrect implementation of cryptographic algorithms/protocols
  - fixed passwords of service accounts or passwords derived from public information
  - sending sensitive data in plaintext (no encryption)
  - using weak/obsolete cryptographic algorithms
- examples can be found in NIST's National Vulnerability Database (NVD)

Exercises

1. *(Double Encryption) Apply two consecutive encryptions with independent keys:* $c = E_{k_2}\big(E_{k_1}(p)\big)$. *If used for the Simple substitution cipher, does this make the resulting cipher weaker, stronger, or equally strong as the original cipher?*
2. *Show that Shift cipher used for a single character plaintext is perfectly secure.*
3. *Show that* $|K| \geq |P|$ *for any perfectly secure encryption scheme.*
4. *We test for an unknown password. Let's assume it is in some set of leaked passwords. Compare the expected number of tests needed when*
   a) *the passwords are uniformly distributed,*
   b) *the distribution of passwords is "skewed".*

   For realistic data, use any leaked database with counts, such as `phpbb-withcount.txt`.