Cryptology – introduction

Martin Stanek

Department of Computer Science
Comenius University
stanek@dcs.fmph.uniba.sk

Cryptology 1 (2021/22)

Content

Introduction

- Cryptology = cryptography & cryptanalysis
- security in the presence of an adversary
  - security means …/ adversary means …?
  - the answers take you to different topics in cryptology
- cryptography: constructions (algorithms, schemes, protocols) for various security requirements:
  - confidentiality (encryption)
  - integrity/authenticity (hash functions, message authentication codes, digital signatures)
  - authentication (protocols)
  - non-repudiation (digital signatures)
  - privacy, anonymity, etc.
- cryptanalysis: breaking or finding weaknesses in cryptographic constructions

Security requirements/goals

- **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

Cryptology vs. information security

cryptography $\subsetneq$ information security

- cryptography is not answer to all security needs:
  - availability (redundancy)
  - secure software (software engineering, security testing), etc.
- cryptography is usually not the entire answer to security needs:
  - cryptography is useless without other security measures
  - key management, access control, risk assessment, personnel security, information classification, etc.

Constructions

- Commonly used cryptographic constructions:
  - encryption, digital signatures, authentication and key distribution protocols, message authentication codes
- Less common constructions:
  - secret sharing, onion routing, voting protocols, electronic money etc.
- "Exotic" constructions:
  - private information retrieval, fully homomorphic encryption, etc.

Encryption

- traditional use of cryptography
- intuitively, we know what encryption is
  - fiction ∼ E.A. Poe, A.C. Doyle, J. Verne …
- encryption provides confidentiality of
  - communicated data, e.g. SSL/TLS, WPA2, S/MIME – preventing data compromise when an attacker eavesdrops
  - stored data, e.g. BitLocker, VeraCrypt, FileVault 2 – preventing data compromise after the attacker, for example, stole a disk
- informally:
  encryption + decryption ∼ encryption scheme ∼ cipher

Encryption – terminology 1

- original data = plaintext
- data after encryption = ciphertext
- P, C, K – finite sets of all plaintexts, ciphertexts, keys
- symmetric/secret key encryption scheme:
  - key generation (usually random bit string)
  - encryption: $E : K \times P \rightarrow C$
    sometimes randomized $E : K \times R \times P \rightarrow C$
  - decryption: $D : K \times C \rightarrow P$

Encryption – terminology 2

- correctness: $\forall k \in K \; \forall p \in P : D_k(E_k(p)) = p$
  for randomized encryption: $\forall k \in K \; \forall r \in R \; \forall p \in P : D_k(E_k(r, p)) = p$
- sometimes more complicated by using various modes of encryption, …
- efficiency: no one wants to wait for en/de-cryption
- security is much more difficult
  - usually "resistance to all known attacks"
  - e.g. identity is correct and efficient but insecure

Example – Shift cipher (Caesar cipher) 1

- alphabet $A = \{\mathrm{A}, \mathrm{B}, \dots, \mathrm{Z}\}$
- natural mapping between characters and numbers:
  A $\leftrightarrow$ 0, B $\leftrightarrow$ 1, …, Z $\leftrightarrow$ 25
- $P = C = A$, $K = \mathbb{Z}_{26}$
- 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

Example – Shift cipher (Caesar cipher) 2

- 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
- security: none (when encrypting natural language text of reasonable length)
  - small key space, only 26 keys (all keys can be tested)
  - How easy is to recognize a plaintext?

Example – Simple substitution cipher

- alphabet $A = \{\mathrm{A}, \mathrm{B}, \dots, \mathrm{Z}\}$
- $P = C = A$, $K = \{\pi \mid \pi \text{ is a permutation on } A\}$
- encryption: $E_\pi(p) = \pi(p)$
- decryption: $D_\pi(c) = \pi^{-1}(c)$
- trivially correct, long plaintext – each character encrypted individually
- large number of keys: $|K| = 26! \approx 2^{88.38}$
- easily broken by frequency and/or pattern analysis (for details see the lecture on cryptanalysis of classical ciphers)
- E.A. Poe: The Gold-Bug (1843)
  - steganography (invisible ink, revealed by heating)
  - includes SSC and a detailed description of its cryptanalysis
- real-world example: Chilean drug traffickers (2010)
- many variants: polyalphabetic subst. (multiple substitutions), homophonic subst. (e.g. frequent letters to multiple targets), …

Example – Simple substitution cipher

- alphabet $A = \{A, B, \dots, Z\}$
- $P = C = A$, $K = \{\pi \mid \pi \text{ is a permutation on } A\}$
- encryption: $E_\pi(p) = \pi(p)$
- decryption: $D_\pi(c) = \pi^{-1}(c)$
- trivially correct, long plaintext – each character encrypted individually
- large number of keys: $|K| = 26! \approx 2^{88.38}$
- easily broken by frequency and/or pattern analysis (for details see the lecture on cryptanalysis of classical ciphers)

- E.A. Poe: The Gold-Bug (1843)
  - steganography (invisible ink, revealed by heating)
  - includes SSC and a detailed description of its cryptanalysis
- real-world example: Chilean drug traffickers (2010)
- many variants: polyalphabetic subst. (multiple substitutions), homophonic subst. (e.g. frequent letters to multiple targets), …

Example – Permutation cipher

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

Example – Fleissner/Cardano Grille

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

Example – Fleissner/Cardano Grille 2

- German army in WWI (withdrawn after 4 months)
- J. Verne: Mathias Sandorf (1885)

	H		A		T
				E	
		M			
	U			S	
					T
			M		

		A			K
			E		
A					M
		A		N	
	P				R

		O			
D					
	U			C	
			T		
	I				
V		E		O	

T				H	
	E		R		
W					I
		S			
E			O		

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 encryption scheme

- How to define the security of encryption scheme?
  What is the goal of an attacker?
  - find the key …what about identity?
  - find plaintext from ciphertext …what about half of the plaintext?
  - find at least one bit/character of the plaintext from ciphertext …
  - compute any nontrivial function of plaintext?
- robust security definitions are nontrivial tasks

Perfect secrecy (intuitively)

- (a priori) probability distribution of plaintexts
  - e.g. “tomorrow” is more probable than “mjuuwerq”
- (a posteriori) probability distribution of plaintexts (knowing a ciphertext)
- def: encryption scheme satisfies perfect secrecy if

  ‘a priori’ distribution of plaintexts is equal to ‘a posteriori’ distribution of plaintext (i.e. knowing a ciphertext does not change the probability distribution of the plaintexts)
- alternative:

  the probability of obtaining particular ciphertext c for arbitrary plaintext is constant
- eavesdropper learns *nothing* from the ciphertext
- single ciphertext; what about the length of the plaintext?

Vernam cipher (one-time pad) 1

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

Vernam cipher (one-time pad) 2

- keys with nonuniform distribution:
  - change the probability distribution of plaintexts (after observing ciphertext)
- reused 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 easily be solved for natural (and other) language texts
    …two time pad problem
- disadvantage: $|\text{key}| = |\text{plaintext}|$
  - distribution in advance (e.g. DVD)
  - shorter key $\Rightarrow$ sacrifice of perfect secrecy

Modern symmetric ciphers

- designed for efficient hardware and software implementations
- cannot have the perfect secrecy property (usually $|\text{key}| < |\text{plaintext}|$)
- operate on bit vectors
- block ciphers:
  - encryption/decryption algorithm defined over bit vectors
  - $E, D : \{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$
  - AES (block size: 128 bits, key size: 128/192/256 bits),
    3DES (block size: 64 bits, key size: 112/168 bits)
- stream ciphers:
  - key and an initialization vector
  - finite state deterministic generator producing (pseudo-random) keystream
  - examples: Snow 3G (LTE networks), ChaCha20 (option in SSL/TLS)
  - block ciphers in specific modes of operation

Asymmetric encryption schemes

- each user generates his/her own instance
- based on intractable mathematical problems
  - factoring, discrete logarithm, 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 key for decryption (only the owner can decrypt)
- correctness:

$$\forall (\mathrm{pk}, \mathrm{sk}) \leftarrow \mathrm{Gen}()\ \forall m : \mathrm{Dec}_{\mathrm{sk}}(\mathrm{Enc}_{\mathrm{pk}}(m)) = m$$

Kerckhoffs's principle

A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.

- don't rely on secret algorithms
- replacing (HW or SW) implementation is costly/impossible
- some known failures:
  - RC4 stream cipher …source code leaked on Internet
  - A5/1 …algorithm reverse engineered
  - CSS (Content Scramble System) encryption of DVDs …reverse engineered
- protecting the design of 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 plaintext-ciphertext pairs
  - known headers, data structures, 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 $\Rightarrow$ *use the strongest possible scheme (with respect to an attack scenario).*

Attack scenarios vs. simple ciphers

- Simple substitution cipher
- COA: frequency/patterns analysis
- KPA: reveals π values for all symbols appearing in the plaintext
- CPA: chosen plaintext “abc…xyz”
- CCA: similar to CPA (the attack cannot be improved)
- similarly for shift cipher …

Key length (symmetric schemes)

- generic attack: exhaustive search of key space (brute-force)
- large key space size: necessary but not sufficient requirement
- example of brute-force attack (key space covered):

time	key length [bits]
1 minute	33.7
1 hour	39.6
1 day	44.1
1 month	49.1
1 year	52.7

i7-2600, 4 cores, HT, HW accelerated AES, 225 mil. AES-128 decryptions/s

Data authenticity

- ensuring that received message is genuine (authentic)
- encryption alone does not guarantee authenticity
  - see one-time pad, …
  - but: authenticated encryption
- MAC (message authentication code)
  - symmetric construction, MAC sent together with a message
  - verification: recomputing and comparing MAC
  - without non-repudiation
- digital signature scheme
  - asymmetric construction (public/private key)
  - anyone can verify a signature
  - only the owner of the private key can sign
- how to define security: goal and capabilities of an attacker?

Modern cryptology

1. formal security definitions
2. precise formulation of assumptions (attacker's capabilities, assumptions)
3. security proofs

How cryptography fails

- real-world security problems
- NIST's National Vulnerability Database (NVD)
- usual problems related to cryptography:
  - bad randomness source for keys generation
  - insufficient checking of public-key certificates
  - incorrect implementation of cryptographic algorithms/protocols
  - fixed passwords of service accounts or passwords derived from public information

NVD – statistics for 2020 (selected CWEs)

Category	Count
CWE-311 Missing Encryption of Sensitive Data	33
CWE-326 Inadequate Encryption Strength	40
CWE-327 Use of a Broken or Risky Cryptographic Algorithm	75
CWE-338 Use of Cryptographically Weak Pseudo-Random Number Generator (PRNG)	5
CWE-347 Improper Verification of Cryptographic Signature	64
CWE-798 Use of Hard-coded Credentials	179
CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	2098
CWE-20 Improper Input Validation	792
CWE-200 Exposure of Sensitive Information to an Unauthorized Actor	316
NVD-CWE-noinfo Insufficient Information	3512