CS579: Foundations of Cryptography

Spring 2023

Introduction

Instructor: **Nikos Triandopoulos**

Introduction to modern cryptography

Cryptography / cryptology

- Etymology
  - two parts: “crypto” + “graphy” / “logy”
  - original meaning: κρυπτός + γράφω / λόγος (**in Greek**)
  - English translation: secret + write / speech, logic
  - meaning: secret writing / the study of secrets
- Historically developed/studied for secrecy in communications
  - i.e., message encryption in the symmetric-key setting
  - main application area: use by military and governments

Classical cryptography Vs. modern cryptography

antiquity – ~70s

- *"the art or writing and solving codes"*
- approach
  - ad-hoc design
  - trial & error methods
  - empirically evaluated

~80s – today

- *"the study of **mathematical techniques** for **securing** digital information, systems, and distributed computations again **adversarial attacks**"*
- approach
  - systematic development & analysis
  - formal notions of security / adversary
  - rigorous proofs of security (or insecurity)

Modern cryptography

Formal treatment

- **fundamental notions** underlying the **design & evaluation** of crypto primitives

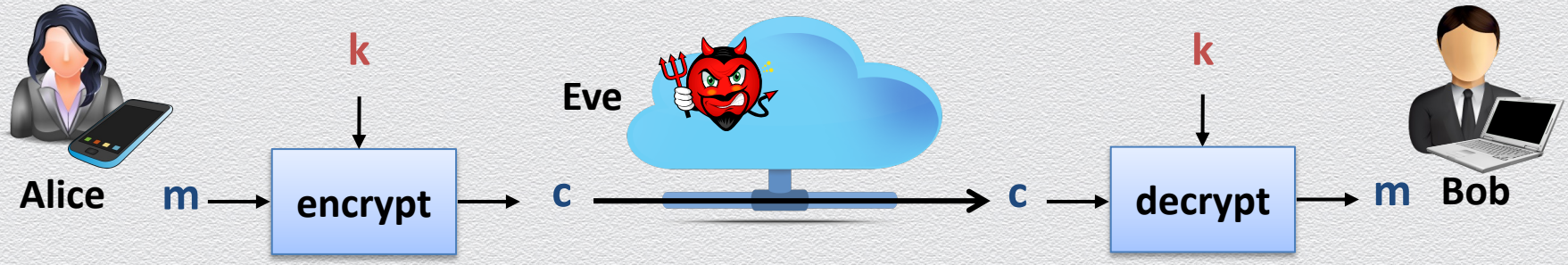
Systematic process

- (A) security goals (what it means for a crypto primitive to be “secure”?)
  - abstracted into **security definitions** amenable to mathematical treatment
- (B) threat model (which forms of attacks are allowed – and which aren’t?)
  - specified by **adversarial settings** & **computational assumptions**
- (C) security analysis (why a candidate instantiation is indeed secure – or not)?
  - expressed by **proofs of security**, **inherent limitations** & **characterizations**

Symmetric-key encryption

The setting of “symmetric-key encryption”

- Motivation: **Secret communication** amongst two parties
  - Alice (sender/source) wants to send a message m to Bob (recipient/destination)
  - Eve (attacker/adversary) can eavesdrop sent messages (i.e., unprotected channel)
- Solution concept: **Symmetric-key encryption scheme**
  - Alice **encrypts** her message m to **ciphertext c**, which is sent instead of **plaintext m**
  - Bob **decrypts** received message c to original message m; Eve “**cannot learn**” m from c
  - a **secret key k** (shared by Alice & Bob) is used by both message transformations



Symmetric-key encryption scheme

Abstract crypto primitive

- defined by **message space** $\mathcal{M}$ & triplet of algorithms **(Gen, Enc, Dec)**
  - Gen: randomized algorithm, outputs a uniformly random key k from key space $\mathcal{K}$
  - Enc: probabilistic algorithm, on input plaintext m and key k, outputs ciphertext c
  - Dec: deterministic algorithm, on input c and key k, outputs a plaintext m
- satisfying desired properties
  - **efficiency**: key generation & message transformations "are fast"
  - **correctness**: for all m, k it holds that Dec(Enc(m, k) , k) = m
  - **security**: one "cannot learn" plaintext m from ciphertext c

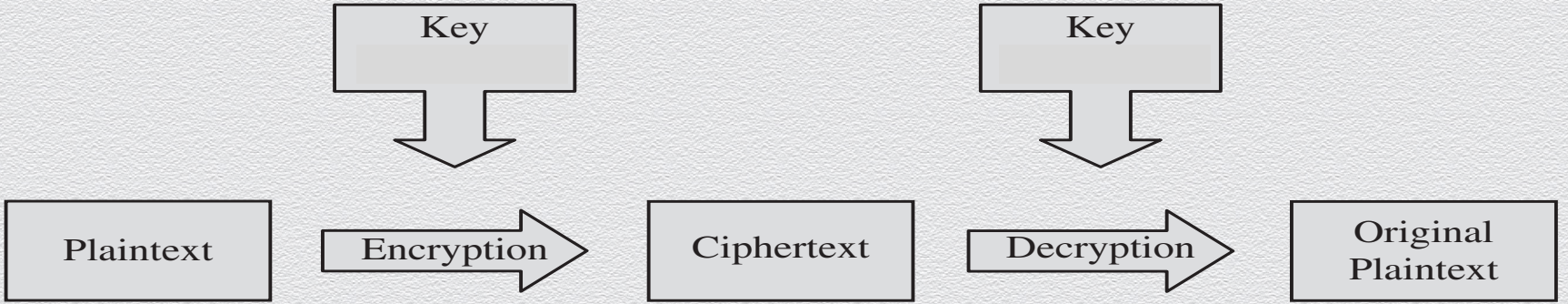
Kerckhoff's principle

"The cipher method must not be required to be secret, and it must be able to fall into the hands of the enemy without inconvenience."

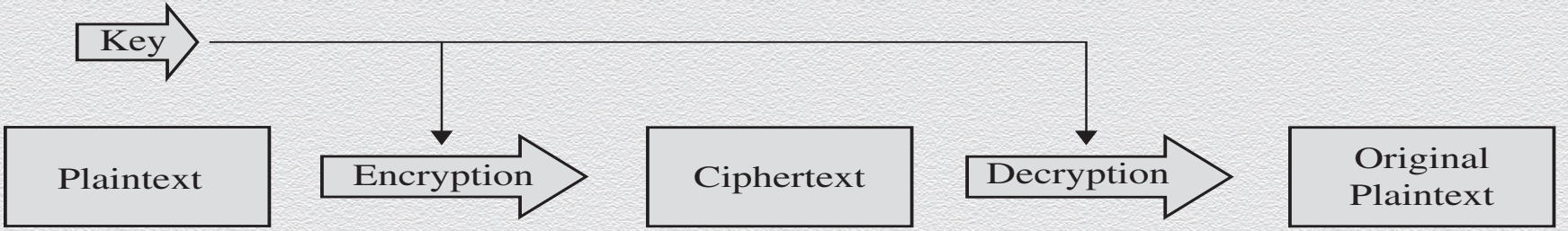
Reasoning

- due to security & correctness, Alice & Bob must share some secret info
- if no shared key captures this secret info, it must be captured by Enc, Dec
- but keeping Enc, Dec secret is problematic
  - harder to keep secret an algorithm than a short key (e.g., after user revocation)
  - harder to change an algorithm than a short key (e.g., after secret info is exposed)
  - riskier to rely on custom/ad-hoc schemes than publicly scrutinized/standardized ones

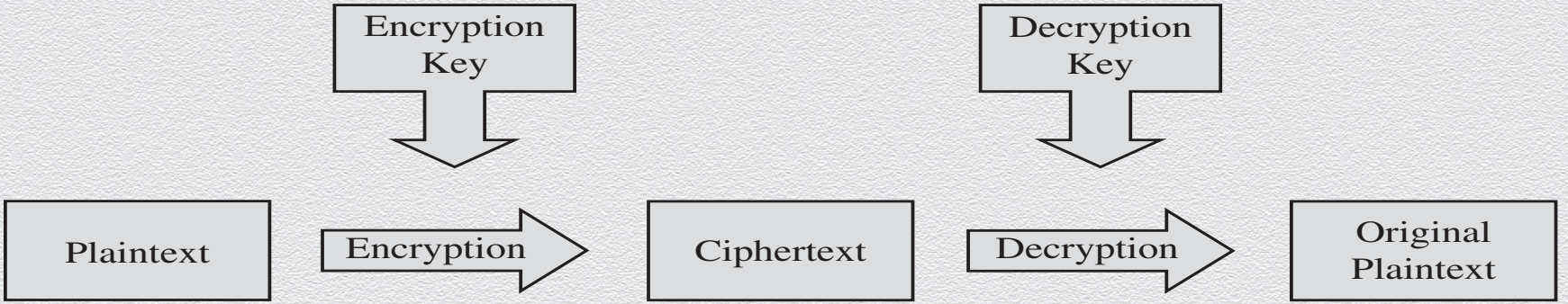
Symmetric-key encryption



Symmetric Vs. Asymmetric encryption



(a) Symmetric Cryptosystem



(b) Asymmetric Cryptosystem

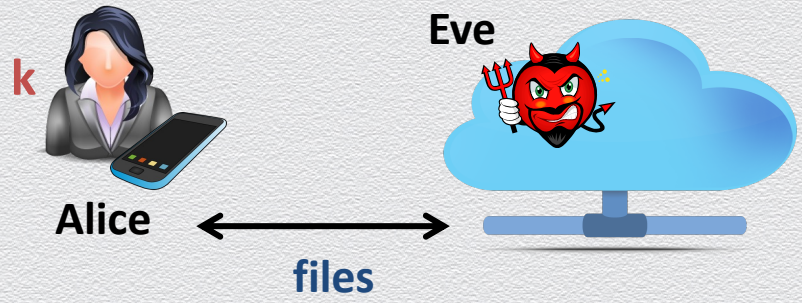
Main application areas

Secure communication

- **encrypt messages** sent among parties
- assumption
  - Alice and Bob **securely generate, distribute and store shared key k**
  - attacker does not learn key k

Secure storage

- **encrypt files** outsourced to the cloud
- assumption
  - Alice **securely generates and stores key k**
  - attacker does not learn key k



Brute-force attack

Generic attack

- given a captured ciphertext c and known key space $\mathcal{K}$, Dec
- strategy is an **exhaustive search**
  - try all possible keys k in $\mathcal{K}$ and determine if Dec (c,k) is a likely plaintext m
- **requires some knowledge on the message space $\mathcal{M}$**
  - i.e., structure of the plaintext (e.g., PDF file or email message)

Countermeasure

- key should be a **random** value from a **sufficiently large key space $\mathcal{K}$** to make exhaustive search attacks **infeasible**

Perfect secrecy

A formal, mathematic view of symmetric encryption

A symmetric-key encryption scheme is defined by

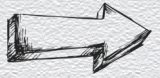
- a **message space** $\mathcal{M}$, $|\mathcal{M}| > 1$, and a triple (**Gen**, **Enc**, **Dec**)
- **Gen**: probabilistic key-generation algorithm, defines **key space** $\mathcal{K}$
  - Gen $\rightarrow$ k $\in \mathcal{K}$
- **Enc**: probabilistic encryption algorithm, defines **ciphertext space** $\mathcal{C}$
  - Enc: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, Enc(k, m) = $\text{Enc}_k(m) \rightarrow c \in \mathcal{C}$
- **Dec**: deterministic encryption algorithm
  - Dec: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$, Dec(k, c) = $\text{Dec}_k(c) := m \in \mathcal{M}$

A view through the lens of probability

Symmetric-key encryption scheme: **message space** $\mathcal{M}$ & triple (**Gen**, **Enc**, Dec)

- **Gen** defines **key space** $\mathcal{K}$ i.e., Gen $\rightarrow$ k $\in \mathcal{K}$
- **Enc** defines **ciphertext space** C i.e., $\text{Enc}_k(m) \rightarrow c \in C$

Assumption

- **messages** & **keys** are chosen **independently** according to prob. distributions $\mathcal{D}_{\mathcal{M}}$, $\mathcal{D}_{\mathcal{K}}$
- if M, K are random variable denoting the chosen message and key respectively
  - for any m $\in \mathcal{M}$, $\mathcal{D}_{\mathcal{M}}$ defines Pr[M = m]  *a priori* probability that m is sent
  - for any k $\in \mathcal{K}$, $\mathcal{D}_{\mathcal{K}}$ defines Pr[K = k]  typically uniform

Fact

- given $\mathcal{D}_{\mathcal{M}}$, $\mathcal{D}_{\mathcal{K}}$ & **internally used randomness**, Enc imposes a prob. distr. $\mathcal{D}_C$ (over C)
- if C denotes the sent ciphertext, then for any c $\in C$, $\mathcal{D}_C$ defines Pr[C = c]

Perfect correctness

For any $k \in \mathcal{K}$, $m \in \mathcal{M}$ and any ciphertext c output of $Enc_k(m)$,

it holds that

$$\Pr[\, Dec_k(c) = m \,] = 1$$

Towards defining perfect security

- defining security for an encryption scheme is not trivial
  - e.g., what we mean by << Eve “cannot learn” m (from c) >> ?
- our setting so far is a random experiment
  - a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
  - a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
  - $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

how to define security?

Attempt 1: Protect the key k!

- a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

- Security means that

the adversary should **not** be able to **compute the key k**

- Intuition
  - it'd better be the case that the key is protected!...

necessary condition

- Problem
  - this definition fails to exclude clearly insecure schemes
  - e.g., the key is never used, such as when $\text{Enc}_k(m) := m$

but not
sufficient condition!

Attempt 2: Don't learn m!

- a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

- Security means that

the adversary should **not** be able to **compute the message m**

- Intuition
  - it'd better be the case that the message m is not learned...
- Problem
  - this definition fails to exclude clearly undesirable schemes
  - e.g., those that protect m partially, i.e., they reveal the least significant bit of m

Attempt 3: Learn nothing!

- a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

- Security means that

the adversary should **not** be able to **learn any information about m**

- Intuition
  - it seems close to what we should aim for perfect secrecy…
- Problem
  - this definition fails to ignore the adversary's prior knowledge on $\mathcal{M}$
  - e.g., distribution $\mathcal{D}_{\mathcal{M}}$ may be known or estimated
    - m is a valid text message, or one of "`attack`", "`no attack`" is to be sent

Attempt 4: Learn nothing more!

- a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

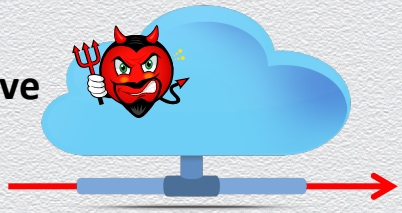
- Security means that

the adversary should **not** be able to **learn any <u>additional</u> information on m**

- How can we formalize this?

Alice **m**

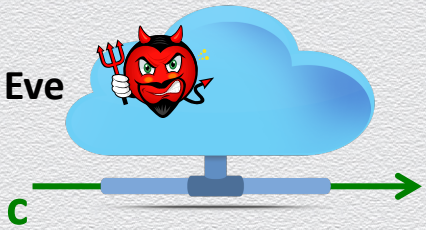
$\text{Enc}_k(m) \rightarrow c$



m =

attack	w/ prob. 0.8
no attack	w/ prob. 0.2

Eve's view remains the same!



c

m =

attack	w/ prob. 0.8
no attack	w/ prob. 0.2

Perfect secrecy (or information-theoretic security)

Definition 1

A symmetric-key encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$, is **perfectly secret** if for every $\mathcal{D}_{\mathcal{M}}$, every message m ∈ $\mathcal{M}$ and every ciphertext c ∈ C for which Pr [C = c] > 0, it holds that

$$\Pr[M = m \mid C = c] = \Pr[M = m]$$

- intuitively
  - the *a posteriori* probability that any given message m was actually sent is the **same** as the *a priori* probability that m would have been sent
  - observing the ciphertext reveals **nothing** about the underlying plaintext

Alternative view of perfect secrecy

Definition 2

A symmetric-key encryption scheme (Gen, Enc, Dec) with message space $\mathcal{M}$, is **perfectly secret** if for every messages m, m' ∈ $\mathcal{M}$ and every c ∈ C, it holds that

$$\Pr[\ \mathrm{Enc}_K(m) = c\ ] = \Pr[\ \mathrm{Enc}_K(m') = c\ ]$$

- intuitively
  - the probability distribution $\mathcal{D}_C$ **does not depend** on the plaintext
  - i.e., M and C are **independent** random variables
  - the ciphertext contains "**no information**" about the plaintext
  - "**impossible to distinguish**" an encryption of **m** from an encryption of **m'**

Definitions 1 and 2 are equivalent (Def. 1 ⇒ Def. 2)

Fix any **m**, **m'** ∈ $\mathcal{M}$ and c ∈ C for which Pr [C = c] > 0

- we know

  Pr[M = **m** | C = c] = Pr [M = **m**] = p_1 — by assumption (1)

  Pr[M = **m'** | C = c] = Pr [M = **m'**] = p_2 — by assumption (2)

  $Pr[Enc_K(\mathbf{m^*}) = c] = Pr[C = c \mid M = \mathbf{m^*}]$, for any **m*** ∈ $\mathcal{M}$, c ∈ C — by definition (3)

- then (4)

  Pr[M = **m** | C = c] = Pr[C = c | M = **m**] Pr[M = **m**] / Pr[C = c] — Baye's Theorem

  p_1 = $Pr[Enc_K(\mathbf{m}) = c]\ p_1$ / Pr[C = c] — (1), (3)

  $Pr [Enc_K(\mathbf{m}) = c]$ = Pr[C = c] — for **m**

- thus: $Pr [Enc_K(\mathbf{m}) = c] = Pr[C = c] = Pr [Enc_K(\mathbf{m'}) = c]$ — (2), (4) for **m'**

Definitions 1 and 2 are equivalent (Def. 1 ⇐ Def. 2)

Fix any $\mathcal{D}_{\mathcal{M}}$, m ∈ $\mathcal{M}$ and c ∈ $\mathcal{C}$ for which Pr [C = c] > 0

- case 1: if Pr[M = m] = 0, then clearly Pr[M = m | C = c] = 0 = Pr[M = m]
- case 2: otherwise, recall that
  - Pr[Enc_K(**m***) = c] = Pr[C = c | M = **m***], for any **m*** ∈ $\mathcal{M}$, c ∈ $\mathcal{C}$ (by definition)
  - Pr[Enc_K(**m***) = c] = δ_c (by assumption that Def. 2 holds)
  - Pr[A|B] = Pr[B|A] Pr[A] / Pr[B] (for Pr[B] > 0) (Baye's Theorem)
  - Pr[A] = $\Sigma_{i=1}^{t}$ Pr[A|E_i] Pr[E_i], for any partition {E_i | i = 1,…,t} of the probability space
  - thus: Pr[M = m | C = c] = Pr[C = c | M = m] Pr[M = m] / Pr[C = c]
    = δ_c Pr[M = m] / $\Sigma_{m^* \in \mathcal{M}}$ Pr[C = c | M = m*] Pr[M = m*]
    = δ_c Pr[M = m] / δ_c $\Sigma_{m^* \in \mathcal{M}}$ Pr[M = m*]
    = Pr[M = m]

The one-time pad

The one-time pad

A type of “substitution” cipher that is absolutely unbreakable

- invented in 1917 Gilbert Vernam and Joseph Mauborgne
- to encrypt a plaintext m of length t, it uses **a block of “shift keys” of size n** ($k_1, k_2, \ldots, k_t$), with each shift key being chosen **uniformly at random**

It is **perfectly secure**

- since each shift is random, every ciphertext is equally likely for any plaintext

Formal definition of the one-time pad cipher

Fix t to be any positive integer; set $\mathcal{M} = C = \mathcal{K} = \{0,1\}^t$

- **Gen**: choose t bits uniformly at random (each bit independently w/ prob. .5)
  - Gen $\rightarrow \{0,1\}^t$
- **Enc**: given a key and a message of equal lengths, compute the bit-wise XOR
  - $\text{Enc}(k, m) = \text{Enc}_k(m) \rightarrow k \oplus m$ (i.e., mask the message with the key)
- **Dec**: compute the bit-wise XOR of the key and the ciphertext
  - $\text{Dec}(k, c) = \text{Dec}_k(c) := k \oplus c$
- Correctness
  - trivially, $k \oplus c = k \oplus k \oplus m = 0 \oplus m = m$

Recall: Perfect secrecy

a posteriori = a priori ~ **C is independent of M**

For every $\mathcal{D}_{\mathcal{M}}$, m ∈ $\mathcal{M}$ and c ∈ $\mathcal{C}$, for which Pr [C = c] > 0, it holds that

Pr[M = m | C = c] = Pr[M = m]

For every m, m' ∈ $\mathcal{M}$ and c ∈ $\mathcal{C}$, it holds that

Pr[Enc_K(m) = c] = Pr[Enc_K(m') = c]

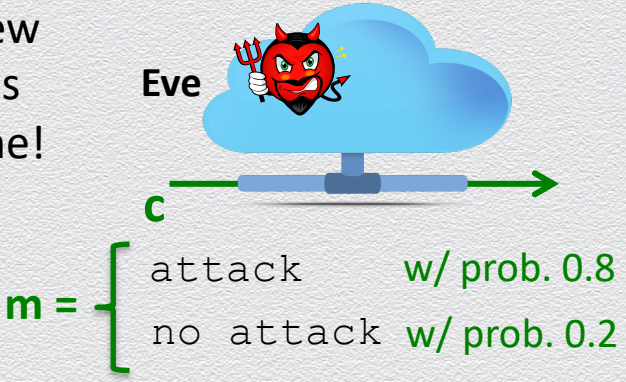
random experiment

$\mathcal{D}_{\mathcal{M}} \rightarrow$ **m = M**

$\mathcal{D}_{\mathcal{K}} \rightarrow$ k

Enc_k(m) → c = C

Eve's view remains the same!



Perfect security (using Def. 2)

For all t-bit long messages m_1 and m_2 and ciphertexts c, it holds that

$$\Pr[\, E_K(m_1) = c \,] \quad = \quad \Pr[\, E_K(m_2) = c],$$

where probabilities are measured over the possible keys chosen by Gen.

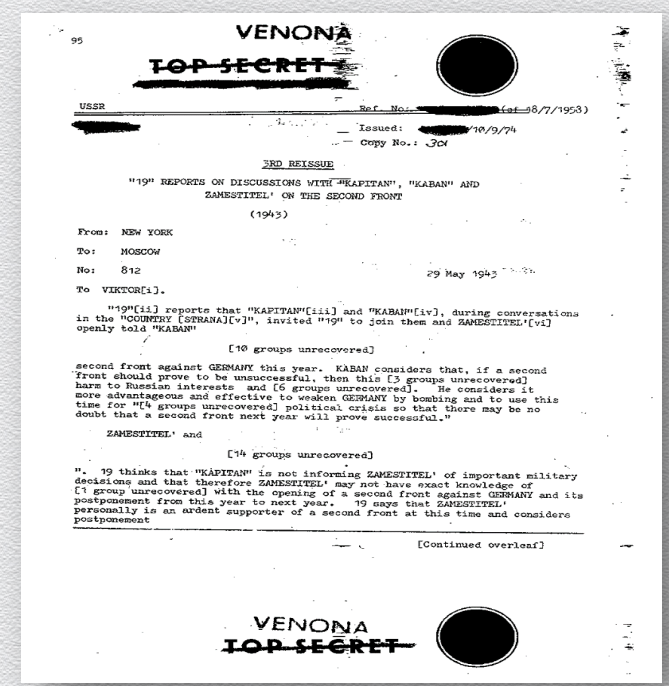
Proof

- the event "$Enc_K(m) = c$" is the event "$m \oplus K = c$" or the event "$K = m \oplus c$"
- K is chosen at random, irrespectively of m, with probability 2^{-t}
- namely ciphertext does not reveal anything about the plaintext

But…

In spite of their perfect security, one-time pads have two notable weaknesses

- the key has to be **as long as** the plaintext
- keys **can never be reused**
  - if they are reused, the XOR of plaintext messages is leaked
  - repeated use of one-time pads compromised communications during the cold war
    - NSA decrypted Soviet messages that were transmitted in the 1940s
    - that was possible because the Soviets reused the keys in the one-time pad scheme



95

VENONA

~~TOP SECRET~~

USSR

Ref. No.: [redacted] (of 28/7/1953)

[redacted]

Issued: [redacted] 10/9/74

Copy No.: 301

3RD REISSUE

"19" REPORTS ON DISCUSSIONS WITH "KAPITAN", "KABAN" AND ZAMESTITEL' ON THE SECOND FRONT

(1943)

From: NEW YORK

To: MOSCOW

No: 812

29 May 1943

To VIKTOR[i].

"19"[ii] reports that "KAPITAN"[iii] and "KABAN"[iv], during conversations in the "COUNTRY [STRANA][v]", invited "19" to join them and ZAMESTITEL'[vi] openly told "KABAN"

[10 groups unrecovered]

second front against GERMANY this year. KABAN considers that, if a second front should prove to be unsuccessful, then this [3 groups unrecovered] harm to Russian interests and [6 groups unrecovered]. He considers it more advantageous and effective to weaken GERMANY by bombing and to use this time for "[4 groups unrecovered] political crisis so that there may be no doubt that a second front next year will prove successful."

ZAMESTITEL' and

[14 groups unrecovered]

". 19 thinks that "KAPITAN" is not informing ZAMESTITEL' of important military decisions and that therefore ZAMESTITEL' may not have exact knowledge of [1 group unrecovered] with the opening of a second front against GERMANY and its postponement from this year to next year. 19 says that ZAMESTITEL' personally is an ardent supporter of a second front at this time and considers postponement

[Continued overleaf]

VENONA

~~TOP SECRET~~

One-time pad (OTP): Limitations are inherent

Theorem (Shannon, 1949)

OTP is **optimal** in the class of **perfectly secret** symmetric encryption schemes

(1) For any perfect cipher $\mathcal{M}$, (Gen, Enc, Dec), with key space $\mathcal{K}$ it holds that $|\mathcal{K}| \geq |\mathcal{M}|$

- thus, OTP keys must be at least as large as the message length
- proof: perfect secrecy $\Rightarrow$ C = $\text{Enc}_K(m)$ independent of M

  $\Rightarrow \forall m, m' \in \mathcal{M}, c \in \mathcal{C}$: $\Pr[\text{Enc}_K(m) = c] = \Pr[\text{Enc}_K(m') = c]$

  $\Rightarrow \forall m, m' \in \mathcal{M}, c \in \mathcal{C}$: $\{\text{Enc}_k(m)\}_{k \in \mathcal{K}} = \{\text{Enc}_k(m')\}_{k \in \mathcal{K}} = \mathcal{C}'$

  $\Rightarrow |\mathcal{K}| \geq |\mathcal{C}'|$

  correctness $\Rightarrow |\mathcal{C}'| \geq |\mathcal{M}|$

(2) Similarly, one-time key usage is inherent

Shannon's theorem

Let Π={$\mathcal{M}$, (Gen, Enc, Dec)} be an encryption scheme with message space M, for which $|\mathcal{M}|=|\mathcal{K}|=|C|$. Then Π is perfectly secure if and only if:

1. Every key k∈$\mathcal{K}$ is chosen with equal probability $1/|\mathcal{K}|$ by algorithm Gen
2. For every m∈$\mathcal{M}$ and every c∈ C, there exists a unique key k∈$\mathcal{K}$ such that $Enc_k(m)$ outputs c

Overall: Characteristics of OTP cipher

A "substitution" cipher

- encrypt an n-symbol m using n uniformly random "shift keys" $k_1, k_2, \ldots, k_n$

2 equivalent views

- $\mathcal{K} = \mathcal{M} = C$
- "shift" method

view 1 $\{0,1\}^n$

bit-wise XOR ($m \oplus k$)

or

view 2 G, (G,+) is a group

addition/subtraction (m +/- k)

Perfect secrecy

- since each shift is random, every ciphertext is equally likely for any plaintext

Limitations

- "shift keys" (1) are **as long as messages** & (2) **cannot be reused**

Classical ciphers

Recall: Classical cryptography Vs. modern cryptography

antiquity – ~70s

- *"the art or writing and solving codes"*
- approach
  - ad-hoc design
  - trial & error methods
  - empirically evaluated

~80s – today

- *"the study of **mathematical techniques** for **securing** digital information, systems, and distributed computations again **adversarial attacks**"*
- approach
  - systematic development & analysis
  - formal notions of security / adversary
  - rigorous proofs of security (or insecurity)

Recall: Approach in modern cryptography

Formal treatment

- **fundamental notions** underlying the **design & evaluation** of crypto primitives

Systematic process

- (A) **formal definitions** (what it means for a crypto primitive to be “secure”?)
- (B) **precise assumptions** (which forms of attacks are allowed – and which aren’t?)
- (C) **provable security** (why a candidate instantiation is indeed secure – or not)?

Recall: Perfect secrecy

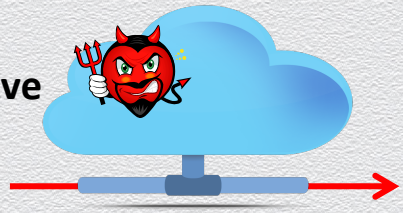
- a message m is chosen according to $\mathcal{D}_{\mathcal{M}}$
- a key k is chosen according to $\mathcal{D}_{\mathcal{K}}$
- $\text{Enc}_k(m) \rightarrow c$ is given to the adversary

- Perfect security means that

the adversary should **not** be able to **learn any <u>additional</u> information on m**

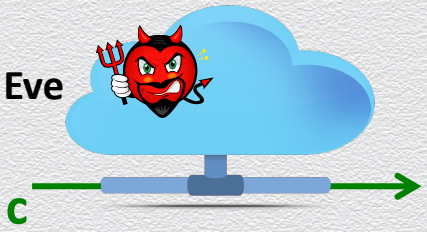
Alice m

$\text{Enc}_k(m) \rightarrow c$



m = { attack w/ prob. 0.8 ; no attack w/ prob. 0.2 }

Eve's view remains the same!



m = { attack w/ prob. 0.8 ; no attack w/ prob. 0.2 }

How well did classical cryptography capture this security notion?

Classical ciphers – general structure

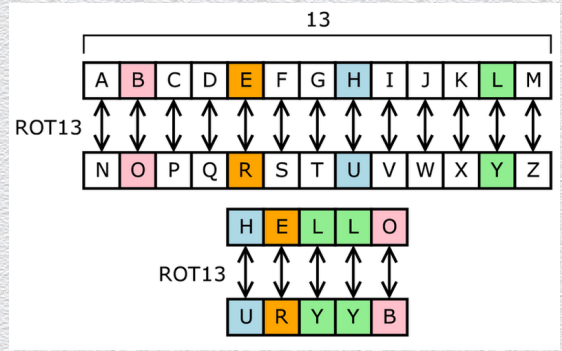
Class of ciphers based on letter substitution

- message space M is "**valid words**" from a given alphabet
  - e.g., English text without spaces, punctuation or numerals
  - characters can be represented as numbers in [0:25]
- encryption
  - ciphertext is produced by **mapping each plaintext character into another character**
  - a character mapping is typically defined as a "**shift**" of a plaintext character **by a number of positions** in a canonical ordering of the characters in the alphabet
  - character shifting occurs with "**wrap-around**" (using mod 25 addition)
- decryption
  - **undo shifting** of characters with "wrap-around" (using mod 25 subtraction)

Substitution ciphers

- Each letter is uniquely replaced by another
- There are 26! possible substitution ciphers

- One popular substitution “cipher” for some Internet posts is ROT13



Frequency analysis

- Letters in a natural language, like English, are not uniformly distributed
- Knowledge of letter frequencies, including pairs and triples can be used in cryptologic attacks against substitution ciphers

a:	8.05%	b:	1.67%	c:	2.23%	d:	5.10%
e:	12.22%	f:	2.14%	g:	2.30%	h:	6.62%
i:	6.28%	j:	0.19%	k:	0.95%	l:	4.08%
m:	2.33%	n:	6.95%	o:	7.63%	p:	1.66%
q:	0.06%	r:	5.29%	s:	6.02%	t:	9.67%
u:	2.92%	v:	0.82%	w:	2.60%	x:	0.11%
y:	2.04%	z:	0.06%				

Letter frequencies in the book *The Adventures of Tom Sawyer*, by Twain.

Classical ciphers – examples

Caesar's cipher

- shift each character in the message by 3 positions
  - or by 13 position in ROT-13
- cryptanalysis
  - **no secret key is used** – based on “security by obscurity”
  - thus the code is trivially insecure once knows Enc (or Dec)

Classical ciphers – examples (II)

Shift cipher

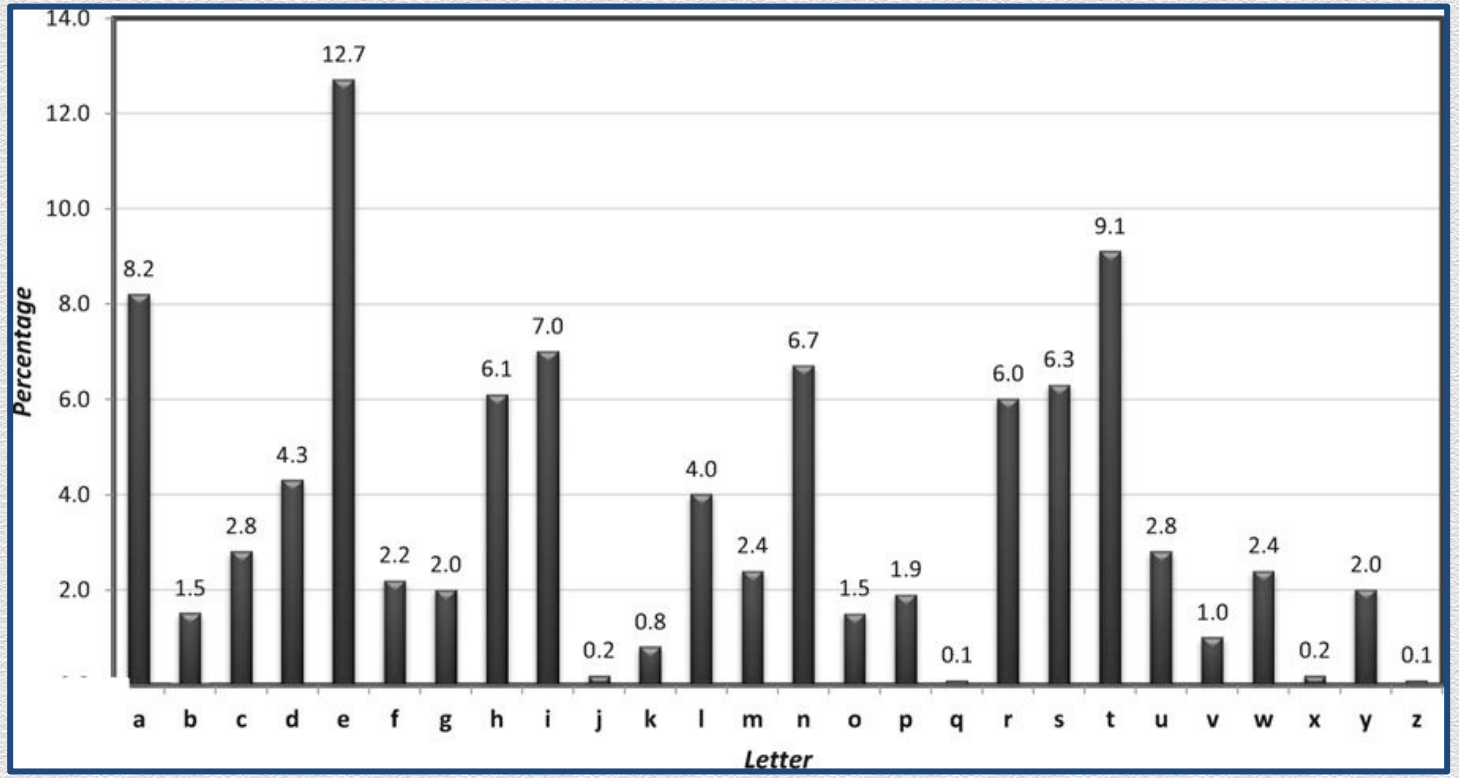
- **keyed extension** of Caesar's cipher
- randomly set key k in [0:25]
  - shift each character in the message by k positions
- cryptanalysis
  - **brute-force attacks** are effective given that
    - **key space is small** (26 possibilities or, actually, 25 as 0 should be avoided)
    - message space M is **restricted to "valid words"**
      - e.g., corresponding to valid English text

Classical ciphers – examples (III)

Mono-alphabetic substitution cipher

- **generalization** of shift cipher
- key space defines **permutation** on alphabet
  - use a **1-1 mapping between characters** in the alphabet to produce ciphertext
  - i.e., shift each **distinct** character in the plaintext (by some appropriate number of positions defined by the key) to get a **distinct** character in the ciphertext
- cryptanalysis
  - key space is large (of the order of 26! or ~2^{88}) but cipher is vulnerable to attacks
  - character mapping is **fixed** by key so **plaintext & ciphertext exhibit same statistics**

Letter frequency in (sufficiently large) English text



Alternative attack against “shift cipher”

- brute-force attack + inspection if English “make sense” is quite **manual**
- a better **automated** attack is based on statistics
  - if character i (in [0:25]) in the alphabet has frequency p_i (in [0..1]), then
    - from known statistics, we know that $\Sigma_i\ p_i^2 \approx 0.065$, so
    - since character i (in plaintext) is mapped to character i + k (in ciphertext)
      - if $L_j = \Sigma_i\ p_i\ q_{i+j}$, then we expect that $L_k \approx 0.065$
- thus, a brute-force attack can **test** all possible keys w.r.t. the **above criterion**
  - the search space **remains the same**
  - yet, the condition to finish the search **becomes much simpler**

Classical ciphers – examples (IV)

The Vigenère cipher (or poly-alphabetic shift cipher)

- **generalization** of mono-alphabetic substitution cipher
- key space defines **fixed (shift) mapping** that is applied on **blocks of characters**
  - a key k is a **string of letters** of length t, defining the shift for blocks of size t
    - period t
    - e.g., if k is “back” or (2, 1, 3, 11), then each block (i.e., every 4 characters in the plaintext) are shifted respectively by 2, 1, 3 and 11 positions
    - i.e., the plaintext-to-ciphertext mapping is **many-to-many**
      - depending on block-location (in plaintext) and character-location (in block)

Cryptanalysis of Vigenère cipher – case I

If key length t is **known**

- problem is **reduced to attacking the shift cipher**
  - consider **t-streams**, subsequences of ciphertext characters that are t-positions apart
    - i.e., subsequences of the form $c_j, c_{j+t}, c_{j+2t}, \ldots$, where j is in [1:t]
  - no text will "make sense" + brute-force attacks may be infeasible (of the order of 26^t)
  - yet, attacks based on statistics (e.g., frequencies or "squares" test) can be used

Cryptanalysis of Vigenère cipher – case II

If key length t is **unknown**

- repeat above attack for guessed values of t (if an upper bound T is known); or find t
- Kasiski's method
  - identify repeated patterns of length 2 or 3 in the ciphertext
  - they're likely to correspond to common bigrams/trigrams in the plaintext (e.g., "the")
  - period t can be deduced by locations of these patterns in the text; or
- index of coincidence method (using character frequencies over t-streams)
  - when character i is shifted by j, we expect $p_i \approx q_{i+j}$ and, thus, $\sum_i p_i^2 \approx \sum_i p_i^2 \approx 0.065$
  - compute $S_\tau = \sum_i q_i^2$ and stop when $S_\tau \approx 0.065$; then τ is (a multiple of) t