Historic perspective on computational encryption and the lessons we learned

Computational cryptography

- Reuse the same key to encrypt multiple messages
- Since it is not longer perfectly secure, what we want is the amount of effort it requires to break is larger than all the computational power in the universe

Historical perspective

- Caesar cipher
- Reusing one-time pad
- Substitution cipher

Caesar cipher

- Keygen
  - Select a random key $\mathrm{k} \in [1,26]$
- Encryption
  - Convert each letter to a number
    - $(a, \dots, z) \rightarrow (0, \dots, 25)$
  - Encrypt each number letter using the shift
    - $c_i \leftarrow m_i + k \ (mod\ 26)$
  - Convert the result back to number
    - $(0, \dots, 25) \rightarrow (a, z)$
- Example
  - $\mathrm{k} = 4$
  - $\mathrm{m} = Caesar$
  - $c = geiwev$

Very easy to break

- Trivial attack: Only 26 keys.
- Lesson: **A good computational encryption scheme needs long keys.**
- How large must a key be?
  - It should require at least 2^{60} operations to break the key

Exercise

- Decrypt

OVDTHUFWVZZPISLRLFZHYLAOLYL

- Encrypt a random message with a random key
- Give someone else your encrypted message and ask them to decrypt them (without)

Useful for writing spoilers on the internet

- Use caesar cipher (mod 13) to write spoilers on the internet

- Game of throne spoiler
  - Tnzr bs guebar fcbvyre vf jung vf jevggra

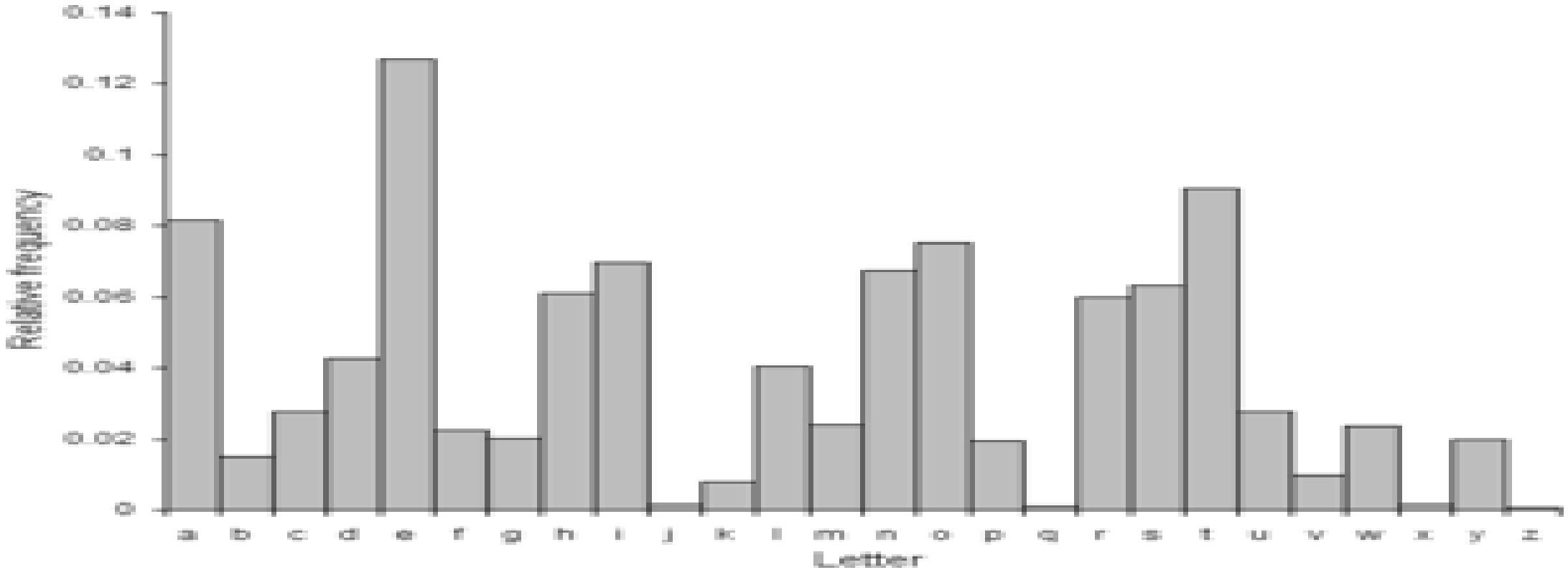
Obscurity does not help

- Trying to hide the encryption algorithm is a bad idea
  - Enigma fell into the hands of allies
  - Untested assumptions
  - Insider attacks
  - Difficulty of evaluating parameters

Permutation cipher

- Key generation
  - Sample permutation π: $[a, \dots, z] \rightarrow [a, \dots, z]$
- Encryption
  - $c_i \leftarrow \pi(m_i)$
- Decryption
  - $m_i \leftarrow \pi^{-1}(c_i)$

Frequency of letters in the English language



Most common two letter pairs

Sequence	Frequency (per 10,000 chars)
th	330
he	302
an	181
in	179
er	169
nd	146
re	133
ed	126
es	115
ou	115
to	115
ha	114
en	111
ea	110
st	109
nt	106
on	106
at	104
hi	97
as	95
it	93
ng	92
is	86
or	84
et	83
of	80
ti	76

Sequence	Frequency (per 10,000 chars)
ar	75
te	75
se	74
me	68
sa	67
ne	66
wa	66
ve	65
le	64
no	60
ta	59
al	57
de	57
ot	57
so	57
dt	56
ll	56
tt	56
el	55
ro	55
ad	52
di	50
ew	50
ra	50
ri	50
sh	50

Other attack vectors

- Most common words.
- Most common expressions
- Knowledge about the plaintext

Which princess are they plotting to kidnap?

WMMW

Anna

Elsa

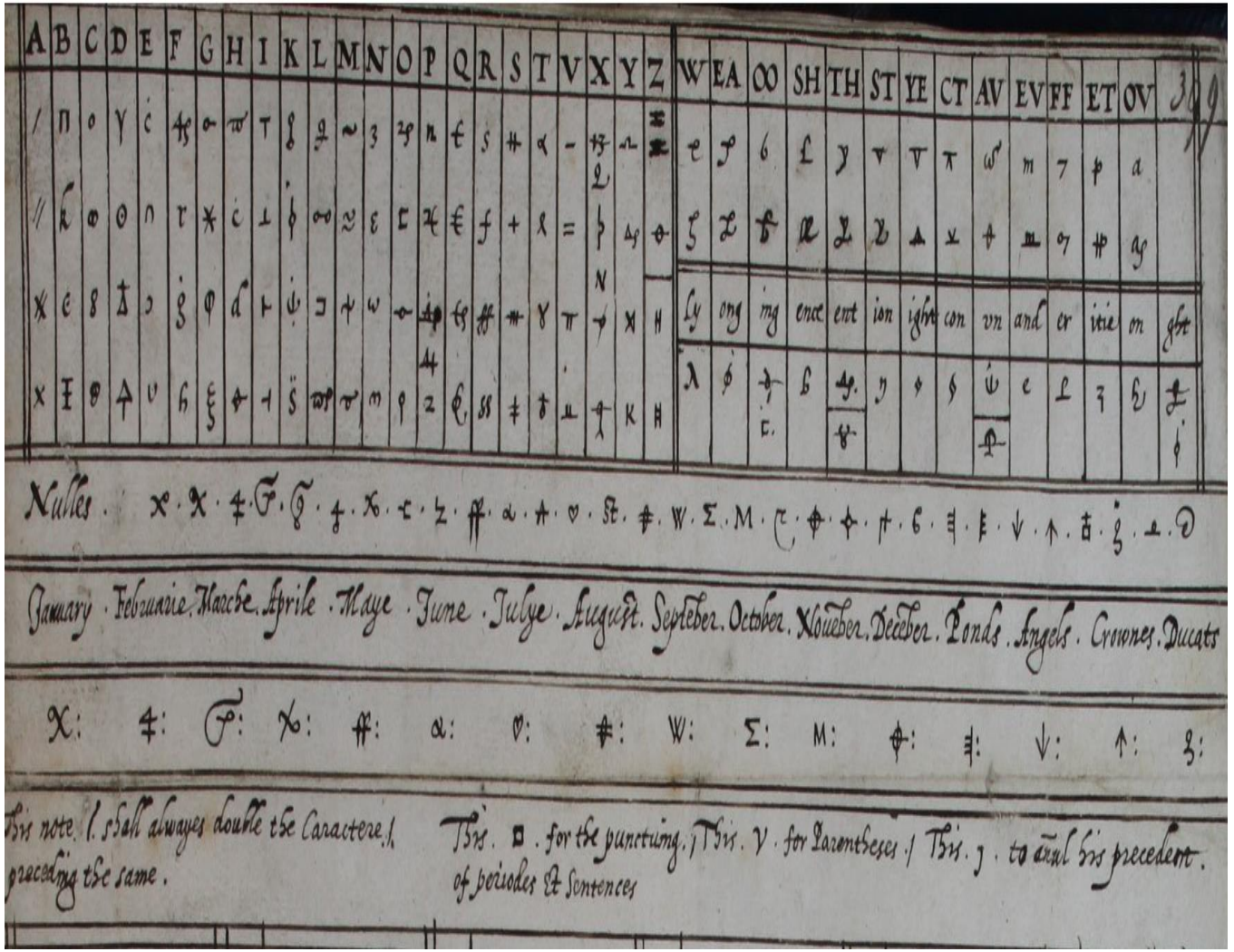
Security of permutation cipher

- # Keys = 26! = 2^88

- Lesson: **Encryption scheme should break correlations**

Mary Queen of Scots

- Planned to depose the queen
- Sent secret messages using
- the cipher on the right



Cryptanalysis of Mary's cipher and ultimate faith of Mary

Thomas Phelippes

Broke the cipher using statistical analysis.

Problem with substitution cipher

- Map each symbol (letter, place or thing) to a set of symbols.
  - Caesar cipher
  - Permutation cipher
  - Using special symbols for places, locations and things

Midway islands

- Site of one of the most important battles of world war two

- Americans cryptographers were able to predict the attack ahead of time.

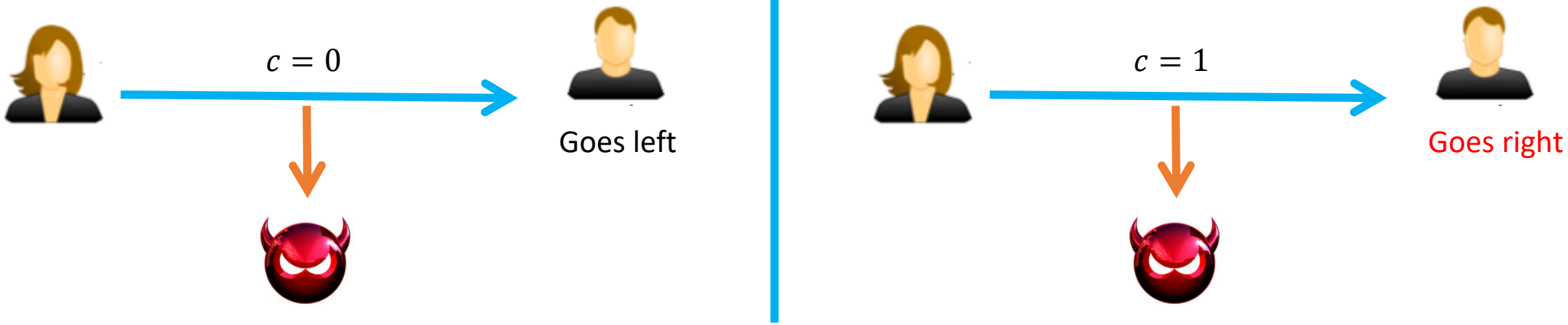
Midway islands

- American cryptanalysts: Does $* = midway\ islands$?
- Americans sent: “Midway is low on water”
- Japanese sent an encrypted message with “*” in it.
- Confirmation: $* = midway\ islands$?
- Lesson: Adversaries can influence the message.

Reusing the one-time pad

- Keygen
  - $k \in_R \{0,1\}^{\ell}$
- Enc
  - $m = m_1 \dots m_{\ell} \in \{0,1\}^{m \cdot d}$
  - $c \leftarrow m_1 \oplus k, \dots, m_d \oplus k$
- Breaking reusing the one-time pad
  - Try it for every length.

Problem with reusing the one-time pad



Enigma machine

- Consisted of rotors

- Each time a letter was presented
  - Letter substituted by other letter
    - Based on rotor position
  - Rotors configuration changed.

Lessons from enigma

- Weakness: a letter can never be encrypted back to itself
  - Misused encryption scheme
- Polish cryptanalysts were able to reverse engineer enigma’s settings
  - Hiding your method of Encryption is foolish
- Lessons: Encryption scheme should not exclude any possibility

Which princess are they plotting to kidnap?

UVND

Anna

Elsa

Kerckhoff's principle

- Lesson: **A cryptosystem should be secure even if it is known by the adversary**

- Why security through obscurity fails
  - Analysis
  - Stealing of machines (as occurred with enigma)
  - Spies

- Modern extension
  - Security of a cryptosystem should rely only on the secrecy of the key
  - NSA assumed that their machines would fall into Soviet hands

Justifications for Kerckhoff's principle

- Easier to change a short key than to change an algorithm
- Algorithm can be leaked by insider

Justifications for Kerckhoff's principle

- If you have n parties, which of these is easier?
  - One secret encryption algorithm for each pair of party
  - A secret key between each party

Application of Kerckhoff's

- **Lesson: Only public (well-studied) algorithms should be used**

- Secret algorithms could
  - Contain flaws
  - Weak parameters
  - Vulnerable to reverse-engineering
  - Backdoors

Principals of modern cryptography

- Principal #1:

The first problem in solving a cryptographic problem is the formulation of a rigorous and precise definition of security

Principals of modern cryptography

- Principal #2:

When the security of a cryptographic scheme relies on an unproven assumption, the assumption should be made explicit

- Ideally the assumption should be simple.

Comparing encryption schemes

- Efficiency
- Level of security
- Simplicity
- Tradeoff

Simplicity

- Why do we care if a cryptographic algorithm is simple?

- The more simpler a crypto algorithm
  - The easier it is to avoid bugs
  - The easier it is to protect against side-channel attacks

Trying to define encryption

- Attempt #1: An encryption scheme is secure if no adversary can learn the key

- Why it fails: In the princess example, you knew which princess to capture without knowing the key.

Trying to define encryption

- Attempt #2: An encryption scheme is secure if no adversary can find the plaintext that corresponds to the ciphertext.

- Why it fails: In the midway example, you just needed to learn which island they were referring to and didn’t really care about what the rest of the message said.

Trying to define encryption

- Attempt #3: An encryption scheme is secure if no adversary can determine any character of the plaintext that corresponds to the ciphertext.

- Why it fails: In the princess example with enigma, we were able to learn which message it was by excluding a character

A valid definition of security for encryption

- An encryption scheme is secure if no adversary can derive any function of the plaintext from the ciphertext.

- Lesson: Definition of security should apply in every context

Cryptographic assumption

- Assumptions where that a problem is hard to solve (conjecture)
  - Finding the prime factors of a number
  - Solving the discrete logarithm

Reductionist approach to security

- To show that a protocol π securely realizes a primitive P under some assumption Q
  1. Take an adversary A' which breaks π
  2. Construct an adversary A which uses A' to break Q

- Example: We will build a protocol for public-key encryption and prove that any adversary which breaks the

Preferences in term of assumptions

- We use Q_1, Q_2 to denote assumptions

- We prefer Q_1 over Q_2 when
  - Q_1 is hard implies Q_2 is hard
  - Q_1 is more studied than Q_2
  - Q_1 is simpler than Q_2
  - Q_1 has post-quantum security and Q_2 doesn't