Introduction to Cryptography

Cryptography (lecture portion)

Andreas Unterweger

School of ITS
Salzburg UAS

Winter term 2024/25

Overview

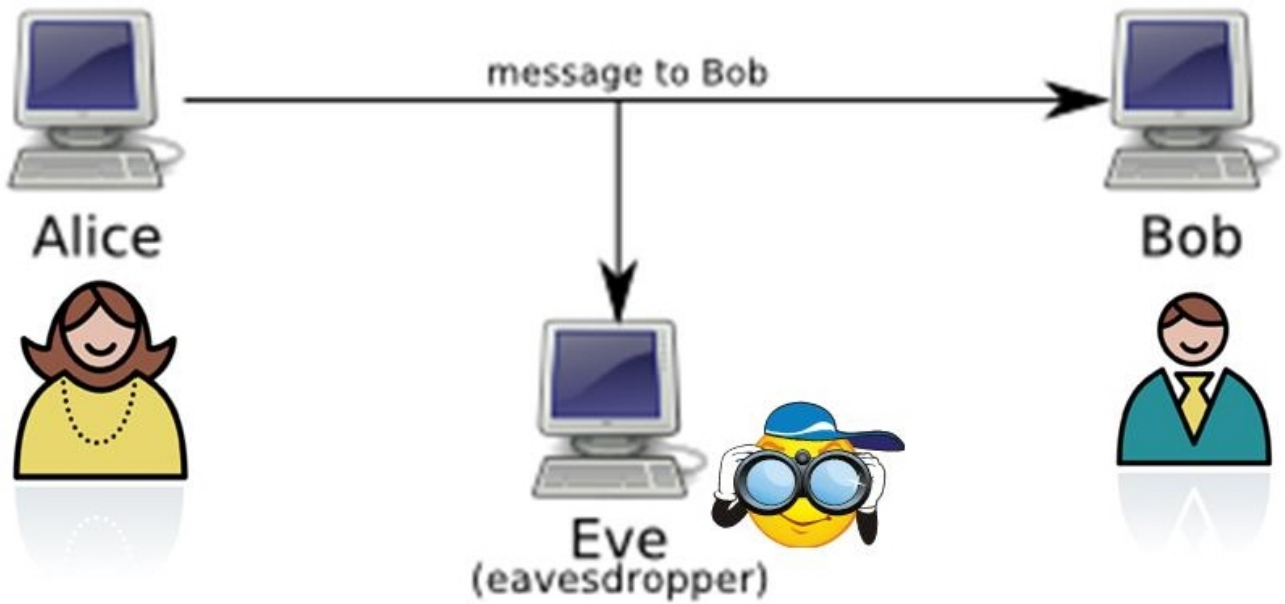
- What is cryptography?
  - Dictionary definition[1]: "secret writing" *(from Greek kryptós, meaning "hidden" or "secret," and graphein, meaning "to write.")*
  - Historically: "[T]he art of writing and solving codes"
  - Nowadays (much broader): "[T]he scientific study of techniques for securing digital information [..]"
- Goals
  - Historically: Secret communication (develop and break ciphers)
  - Authentication (e.g., digital signatures)
  - Secret/key management
  - Proof(s) of security

...

[1] `https://www.merriam-webster.com/dictionary/cryptography` (accessed on August 16, 2022)

Actors in classical cryptography

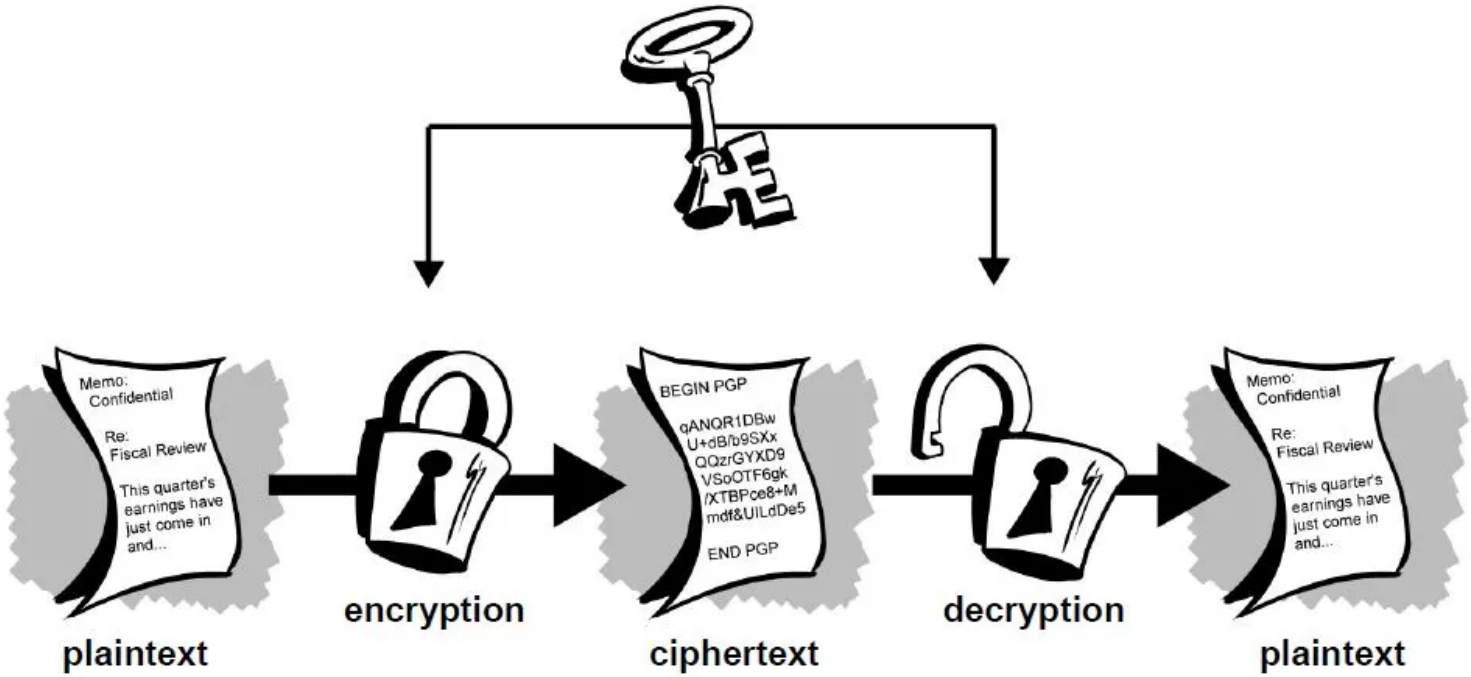
- Setting: Secret communication
- Traditional naming of actors
  - Alice: Message/Secret sender
  - Bob: Message/Secret receiver
  - Eve: Eavesdropper (wants to listen/intercept)



Source: CS110 staff: Encryption and Security. `https://cs.wellesley.edu/~cs110/reading/cryptography-files/handoutPrint.html` (accessed on August 16, 2022), 2014.

Terms in classical cryptography I

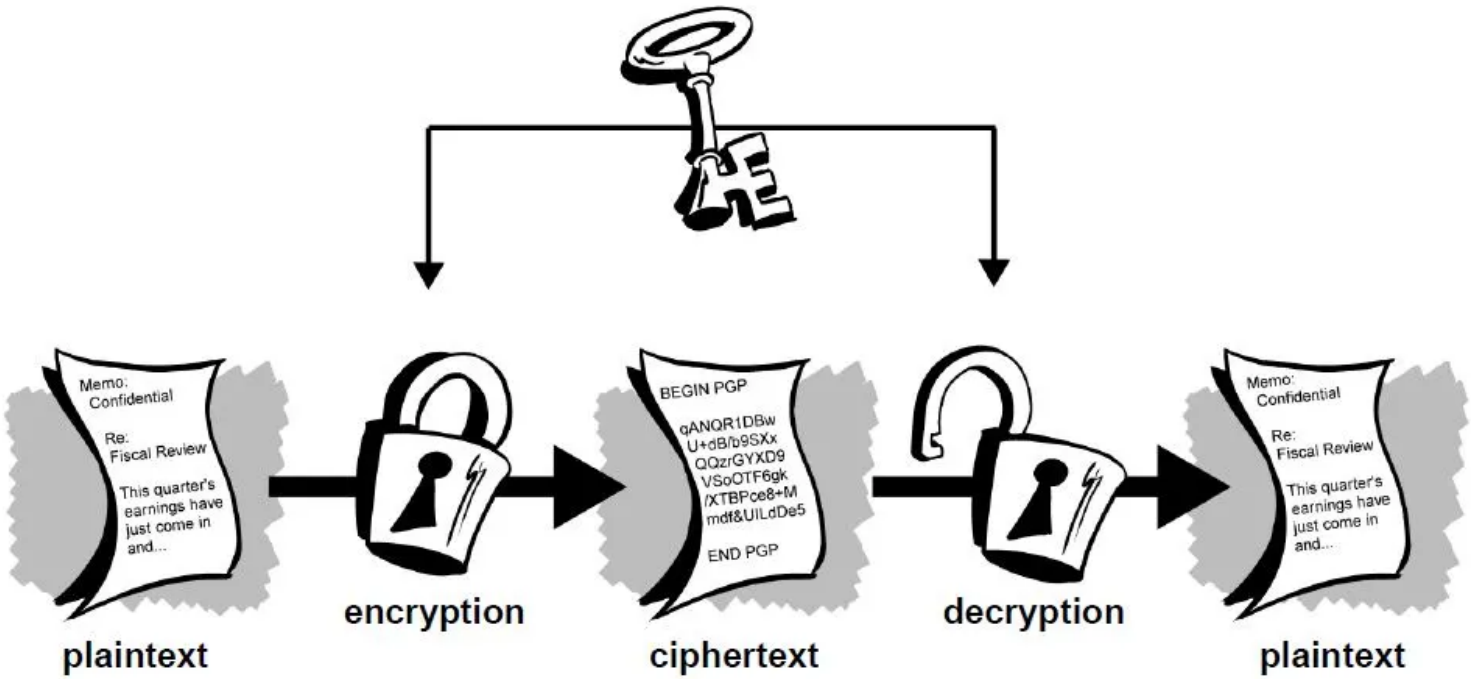
- Plaintext (message) m: Original message
- Ciphertext c: Encrypted message
- Key k: Secret information



Source: Pacia, C.: Beginners' Guide To Off-the-Record Messaging. `https://www.bitcoinnotbombs.com/beginners-guide-to-off-the-record-messaging/` (accessed on August 16, 2022), 2014.

Terms in classical cryptography II

- Encryption (function) $E(k, m)$: Converts a (plaintext) message into a ciphertext, i.e., $c := E(k, m)$
- Decryption (function) $D(k, c)$: Converts a ciphertext back into its plaintext, i.e., $D(k, c) \stackrel{!}{=} m$



Source: Pacia, C.: Beginners' Guide To Off-the-Record Messaging. `https://www.bitcoinnotbombs.com/beginners-guide-to-off-the-record-messaging/` (accessed on August 16, 2022), 2014.

Types of ciphers

- Key symmetry
  - Symmetric (secret key) cryptography: Uses the same key for encryption and decryption
  - Asymmetric (public key) cryptography: Uses different keys for encryption and decryption
- Input/output
  - Block cipher: Processes data in blocks of, e.g., multiple bytes
  - Stream cipher: Processes data character by character (or bit by bit)
- Additional categorization, e.g., substitution ciphers (replace characters by other characters in an invertible way)
  - Mono-alphabetic: Characters are replaced using a single alphabet
  - Poly-alphabetic: Characters are replaced using multiple alphabets

[1] Robshaw, M. J. B.: Stream Ciphers – RSA Laboratories Technical Report TR-701. `http://www.networkdls.com/Articles/tr-701.pdf` (accessed on August 18, 2022), 1995.

Security assumptions

- Perfect security does not exist in practice
  - Some people always need to have access (inherent risk)
  - Human factor is a risk in addition to technical aspects
  - Any system is only as strong as its weakest link
  - Threat model: Security only against certain attacks $\rightarrow$ How powerful is the attacker (assumed to be)?
- Principles of modern cryptography
  - Precise security definition (formalized)
  - If an assumption (e.g., some operation is difficult) cannot be proven, it must be stated, well studied, and as minimal as possible
  - Rigorous proofs (out of scope for this course): Reduce the security of a cipher or cryptographic system to an assumption being true

Kerckhoffs' principle

- The security of a cipher
  - only depends on the secrecy of the key
  - must **not** rely on the cipher/algorithm being secret
- Reasoning
  - (Short) keys are easier to keep secret than (complex) algorithms
  - Exposed keys are easier to replace than broken ciphers
  - Different people using different keys is easier than different people using different ciphers
- Consequence: Modern ciphers/algorithms are public
  - Simplified standardization
  - Experts can assess weaknesses

  $\rightarrow$ Higher confidence and more likely reporting of flaws

The Vernam cipher I

- Also called one-time pad (OTP)
- Inputs/outputs constant-length binary strings
- $E(k, m) := k \oplus m$, with $\oplus$ denoting bit-wise exclusive-or
- $D(k, c) := k \oplus c$ (same as encryption)

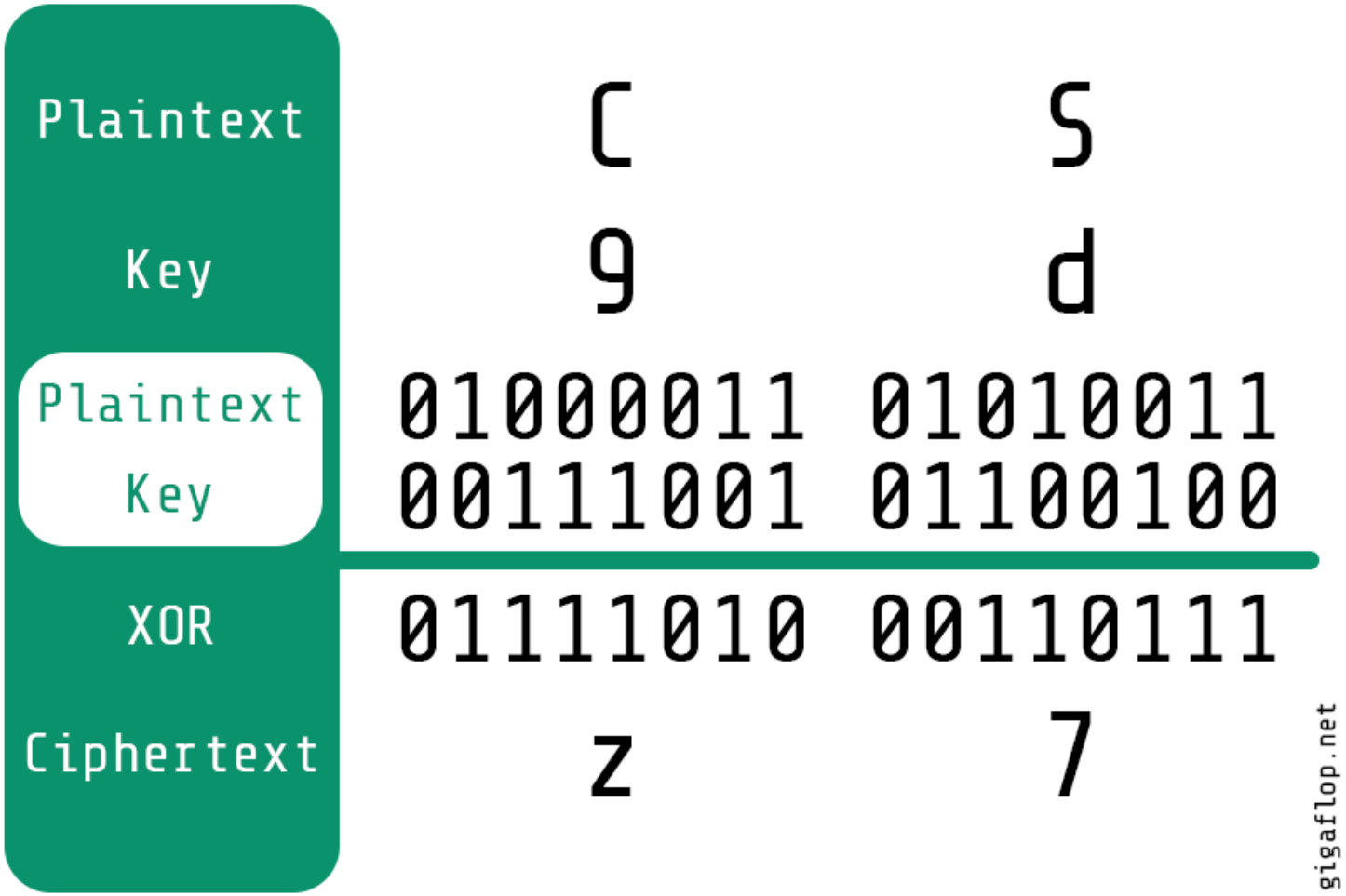
$\rightarrow D(k, E(k, m)) = k \oplus (k \oplus m) = (k \oplus k) \oplus m = 0 \oplus m \stackrel{!}{=} m$

- Provably secure if
  - message m and key k have **exactly** the same length
  - a key is **never** reused
  - the key bit string is uniformly distributed (practical limitations are out of scope for this course)

$\rightarrow$ otherwise the attacker can break perfect secrecy

- Ciphertext does not reveal anything about plaintext since all plaintexts are equally likely under a random key; formal proof in literature

The Vernam cipher II



Source: Gigaflop: Encryption. https://gigaflop.net/chapter/data-representation/encryption/ (accessed on August 16, 2022), 2022.

Overview of classical ciphers

- Classical cryptographic ciphers
  - Used in the past for secret communication
  - Susceptible to various kinds of attacks
  - $\rightarrow$ No longer secure (do not use them)
- Examples of classical ciphers
  - Caesar cipher
  - Vigenère cipher
  - Enigma
- Simplified (message) alphabet
  - Character set: `A`..`Z` (only upper case, no lower case, no digits or special characters)
  - For calculations, `A` corresponds to 0, `B` to 1, ..., `Z` to 25

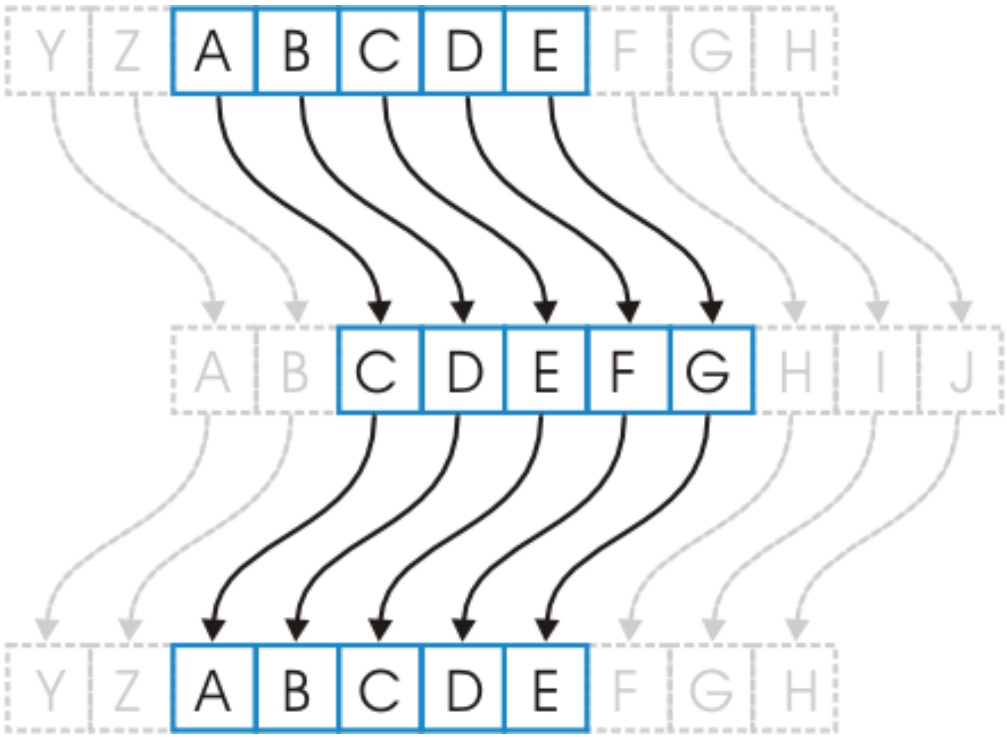
The Caesar cipher I

- General idea: Shifting the alphabet by a constant offset (with rollover)

→ Encryption: Shift every character by the offset, e.g., 2

→ Decryption: Shift back every character by the same offset, e.g., -2

- Offset is the (cipher) key, e.g., $k = 2$ (C)



Source: Smith, J.: A functional implementation of the Caesar cipher in Swift. `https://github.com/ijoshsmith/swift-caesar-cipher` (accessed on August 16, 2022), 2015.

The Caesar cipher II

- $E(k, m) := (m + k) \mod 26$
- $D(k, c) := (c - k) \mod 26$
- Two examples with $k = 2 \hat{=} \texttt{C}$
- Example 1 (no rollover):
  - $m = 1 \hat{=} \texttt{B}$
  - $c := E(k, m) = (m + k) \mod 26 = (1 + 2) \mod 26 = 3 \hat{=} \texttt{D}$
  - $m \stackrel{!}{=} D(k, c) = (c - k) \mod 26 = (3 - 2) \mod 26 = 1 \hat{=} \texttt{B}$
- Example 2 (rollover):
  - $m = 25 \hat{=} \texttt{Z}$
  - $c := E(k = 2, m = 25) = (25 + 2) \mod 26 = 27 \mod 26 = 1 \hat{=} \texttt{B}$
  - $m \stackrel{!}{=} D(k = 2, c = 1) = (1 - 2) \mod 26 = -1 \mod 26 = 25 \hat{=} \texttt{Z}$

The Caesar cipher III

- Mono-alphabetic substitution cipher (character by character)
- Encryption and decryption of messages with multiple characters m_i:
  - $c_i := E(k, m_i) := (m_i + k) \mod 26$ for all i
  - $m_i \overset{!}{=} D(k, c_i) = (c_i - k) \mod 26$ for all i
- Example: Encrypting the message ZOO with $k = 2 \overset{\wedge}{=}$ C
  - $m_0 = 25 \overset{\wedge}{=}$ Z
  - $m_1 = m_2 = 14 \overset{\wedge}{=}$ O
  - $c_0 := E(k = 2, m_0 = 25) = (25 + 2) \mod 26 = 27 \mod 26 = 1 \overset{\wedge}{=}$ B
  - $c_1 = c_2 := E(k = 2, m_2 = 14) = (14 + 2) \mod 26 = 16 \overset{\wedge}{=}$ Q

  $\rightarrow$ Encrypted message (ciphertext) BQQ

The Vigenère cipher I

- General idea: Shifting the alphabet by a position-dependent offset (with rollover), repeated in regular intervals if necessary
- For individual characters: Caesar cipher with position-dependent key

→ Poly-alphabetic substitution cipher

- Key length determines how many different shifts/position-dependent keys exist; example code table for $k \triangleq \texttt{CODE}$:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D

Source: Ibmathsresources.com: Crypto Analysis to Crack Vigenere Ciphers. https://schoolcodebreaking.com/2015/06/18/crypto-analysis-to-crack-vigenere-ciphers/ (accessed on August 16, 2022), 2015.

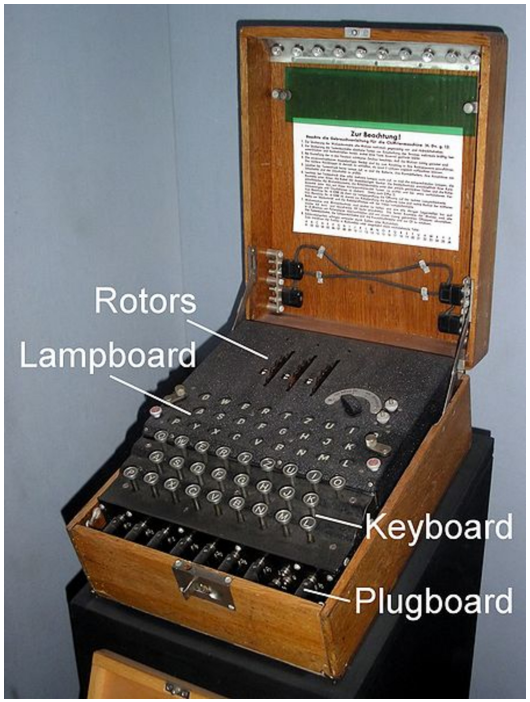
The Vigenère cipher II

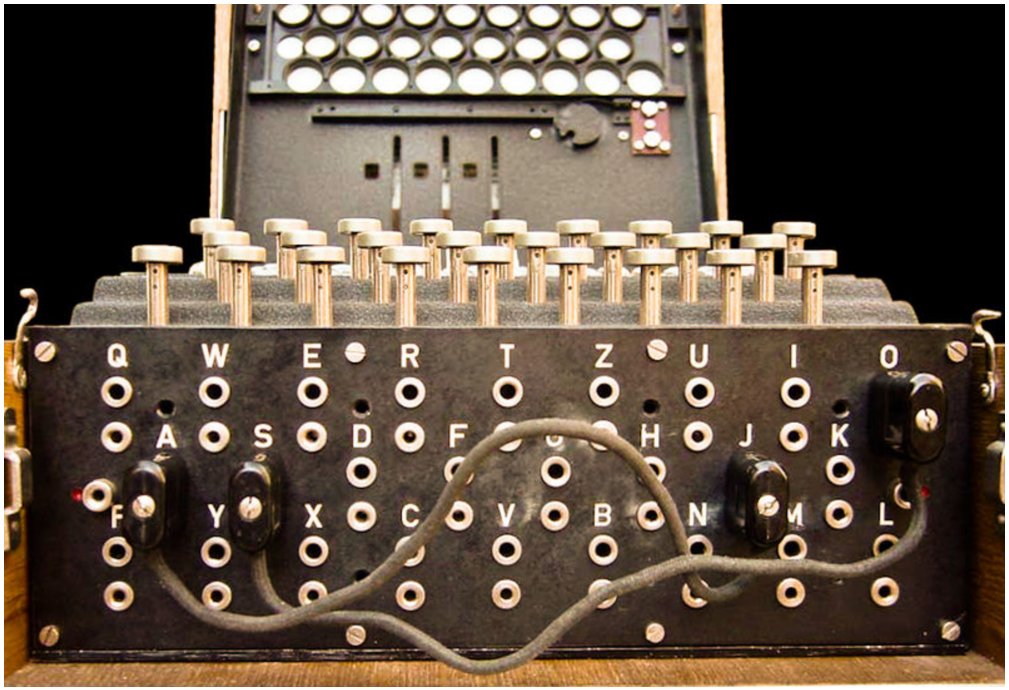
- $Vigenere(k, m_i) = Caesar(k_{i \mod ||k||}, m_i)$, where $||\cdot||$ denotes length
- $c_i := (m_i + k_{i \mod ||k||}) \mod 26$
- $m_i \stackrel{!}{=} (c_i - k_{i \mod ||k||}) \mod 26$
- Example: Encrypting the message TESTING with $k \stackrel{\wedge}{=}$ CODE
  - $m_0 \stackrel{\wedge}{=}$ T , $k_{0 \mod 4} = \text{CODE}_0 \stackrel{\wedge}{=}$ C $\rightarrow c_0 \stackrel{\wedge}{=}$ V
  - $m_1 \stackrel{\wedge}{=}$ E , $k_{1 \mod 4} = \text{CODE}_1 \stackrel{\wedge}{=}$ O $\rightarrow c_1 \stackrel{\wedge}{=}$ S
  - $m_2 \stackrel{\wedge}{=}$ S , $k_{2 \mod 4} = \text{CODE}_2 \stackrel{\wedge}{=}$ D $\rightarrow c_2 \stackrel{\wedge}{=}$ V
  - $m_3 \stackrel{\wedge}{=}$ T , $k_{3 \mod 4} = \text{CODE}_3 \stackrel{\wedge}{=}$ E $\rightarrow c_3 \stackrel{\wedge}{=}$ X
  - $m_4 \stackrel{\wedge}{=}$ I , $k_{4 \mod 4} = \text{CODE}_0 \stackrel{\wedge}{=}$ C $\rightarrow c_4 \stackrel{\wedge}{=}$ K
  - $m_5 \stackrel{\wedge}{=}$ N , $k_{5 \mod 4} = \text{CODE}_1 \stackrel{\wedge}{=}$ O $\rightarrow c_5 \stackrel{\wedge}{=}$ B
  - $m_6 \stackrel{\wedge}{=}$ G , $k_{6 \mod 4} = \text{CODE}_2 \stackrel{\wedge}{=}$ D $\rightarrow c_6 \stackrel{\wedge}{=}$ J

$\rightarrow$ Ciphertext $c \stackrel{\wedge}{=}$ VSVXKBJ

Enigma I

- Complex substitution cipher with multiple variants
- Machine-assisted encryption and decryption with code books





Sources: Moore, K. et al.: Enigma Machine. https://brilliant.org/wiki/enigma-machine/ (accessed on August 16, 2022), 2022.

[2] Moore, K. et al.: Enigma Machine. https://brilliant.org/wiki/enigma-machine/ (accessed on August 16, 2022), 2022.

Enigma II

- Simplified encryption process:

1. Letter swapping: Plugged letters get swapped before actual encryption (e.g., `A` to `J` and vice versa)
   - Changed daily based on code book
   - Fatal flaw: Letters could never be mapped to themselves in the ciphertext $\rightarrow$ Message recovery by (character) exclusion
2. Sequential multi-rotor substitution: Each of three rotors performs a different substitution
   - Rotor order can be chosen, five rotors to choose from in total
   - Starting position of each rotor can be set

- Decryption requires identical machine setup (swap board, rotor order, rotor starting positions) as for encryption

[2] Moore, K. et al.: Enigma Machine. `https://brilliant.org/wiki/enigma-machine/` (accessed on August 16, 2022), 2022.

Overview of cryptanalysis

- Dictionary definition[2]: "the solving of [ciphers] or cryptographic systems"
- Old threat model: The cipher/algorithm is not known (e.g., Enigma)

→ Potentially need to find decryption algorithm first

- Modern threat model: The cipher is known (Kerckhoffs' principle)
- Recap: Security only against certain attacks → How powerful is the attacker?
- Different attack scenarios
- Examples of simple cryptanalysis
  - Frequency analysis
  - Attacks on the Vigenère cipher

[2] `https://www.merriam-webster.com/dictionary/cryptanalysis` (accessed on August 16, 2022)

Attack scenarios I

- Different potential goals of an attacker
  - Recover/decrypt plaintext from a particular ciphertext
  - Recover key to recover plaintexts from arbitrary ciphertexts (stronger attacker who can read all past and future messages, if successful)

- **Basic attacks**
- Exhaustive/**brute-force search attack**
- Timing attacks
- Side-channel attacks

...

Attack scenarios II

Four basic attacks (from weak to strong, for the cipher):

- Ciphertext-only attack: Decrypt a single message based only on the ciphertext (no other capabilities)
- Known-plaintext attack: Collect matching plaintext/ciphertext pairs from the same key and derive key to decrypt other ciphertexts
- Chosen-plaintext attack: Select/influence plaintext which is encrypted into ciphertext and try to reconstruct plaintexts from other ciphertexts
- Chosen-ciphertext attack: Decrypt arbitrary of ciphertexts into plaintexts and try to derive key to decrypt other ciphertexts

Attack scenarios III

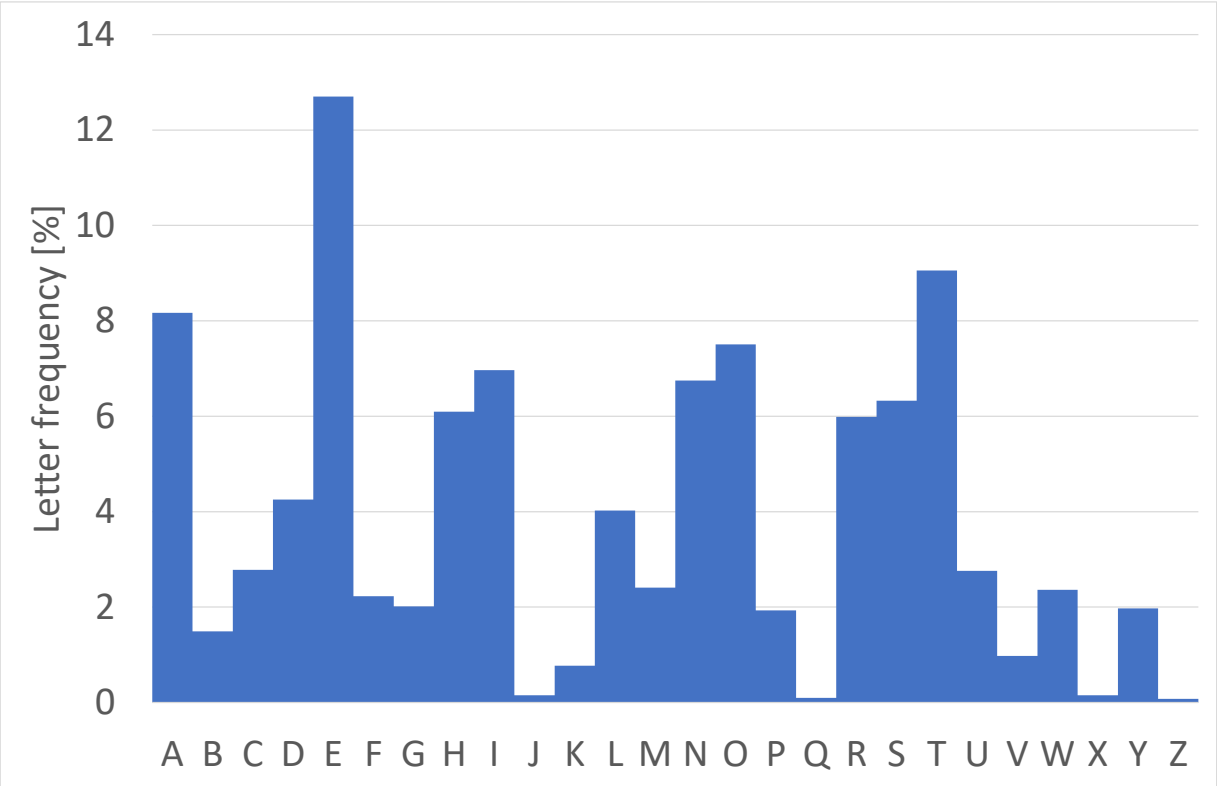
- Brute-force search attack
  - Try all possible keys until the correct one is found
  - $\rightarrow$ More work (processing power) for longer keys
  - Key space: The set of all possible keys

- Key space size examples:
  - Caesar cipher: $26 \approx 10^1$ keys
  - Vigenère cipher with a 3-digit key: $26^3 = 17,576 \approx 10^4$ keys
  - Vernam cipher with a 10-bit key/message: $2^{10} = 1,024 \approx 10^3$ keys
  - Vernam cipher with an n-bit key: $2^n \approx 10^{\frac{n}{3}}$ decryptions

- Practical goal: Infeasible key space sizes for attackers (e.g., 256 bits)

Simple cryptanalysis example: Frequency analysis I

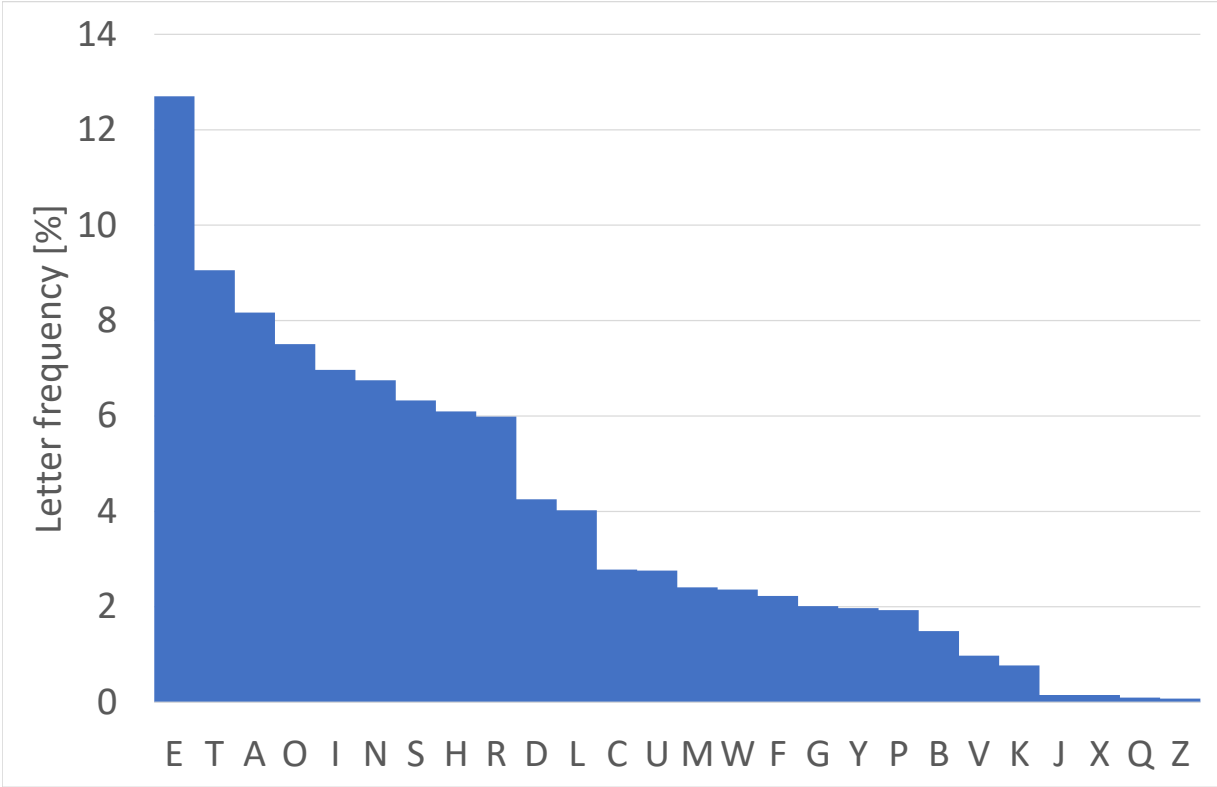
- Assumption: Plaintext is a message in natural language, e.g., English
- The letter frequencies in natural-language texts are known



Data source: Neckář, J.: Letter frequency (English). `http://en.algoritmy.net/article/40379/Letter-frequency-English` (accessed on August 16, 2022), 2016.

Simple cryptanalysis example: Frequency analysis II

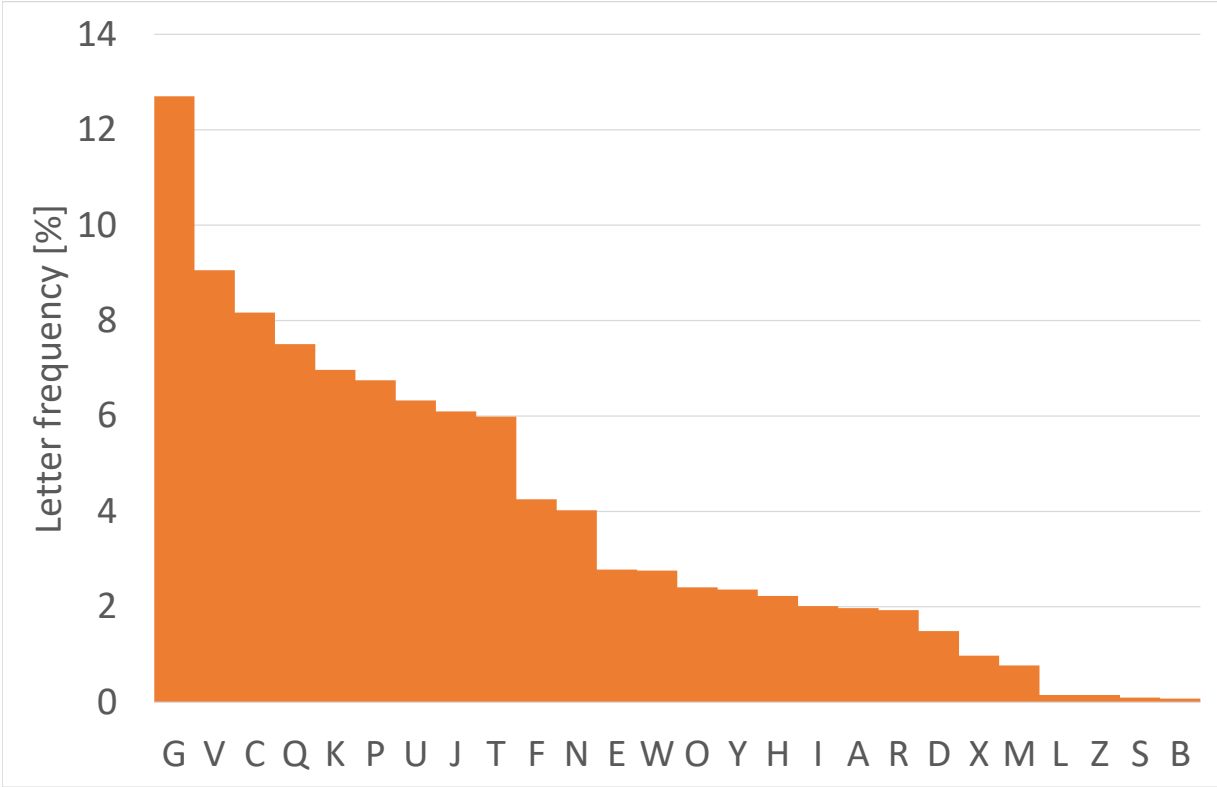
- Frequent characters in the English language are E, T, A, ...



Data source: Neckář, J.: Letter frequency (English). http://en.algoritmy.net/article/40379/Letter-frequency-English (accessed on August 16, 2022), 2016.

Simple cryptanalysis example: Frequency analysis III

- Frequency analysis of ciphertext shows ciphertext character frequency



Original data source: Neckář, J.: Letter frequency (English).
`http://en.algoritmy.net/article/40379/Letter-frequency-English` (accessed on August 16, 2022), 2016.

Simple cryptanalysis example: Frequency analysis IV

- For mono-alphabetic substitution ciphers (e.g., Caesar), frequencies are expected to be similar, with corresponding letters being shifted
- Consistent shift (in the alphabet) between similarly frequent characters allows offset/key recovery (some margin of error is useful!)

- (Ideal) example:
  - Most frequent letter (`G` vs. `E`) $\rightarrow$ Difference of $+2$ (`C`)
  - Second-most frequent letter (`V` vs. `T`) $\rightarrow$ Difference of $+2$ (`C`)

  ...

  $\rightarrow$ Key is `C`

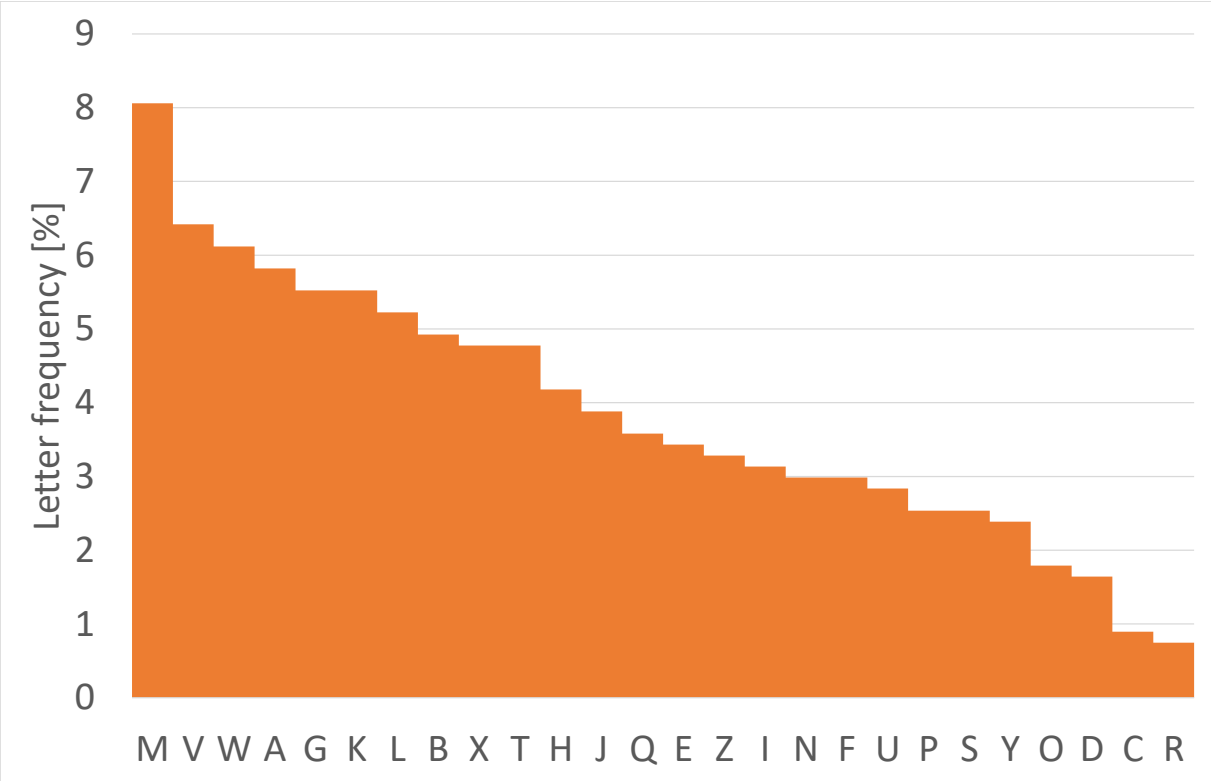
- Caveat: Longer text required
  - Character frequencies only converge for longer, language-typical texts $\rightarrow$ the longer the text, the better; foreign words harm analysis

  $\rightarrow$ Short text may be insufficient for frequency analysis

Attacks on the Vigenère cipher: Freq. analysis attempt

- For poly-alphabetic substitution ciphers like Vigenère, frequency analysis on the whole ciphertext does not work
  - Same plaintext letter m_i mapped to different c_i at different positions

  $\rightarrow$ Character frequencies are "smeared" (more for longer keys)



Attacks on the Vigenère cipher: Modified freq. analysis

- Remember
  - Vigenère cipher is Caesar cipher with position-dependent key → slide 15
  - → Key repeats every $||k||$ positions:
    $Vigenere(k, m_i) = Caesar(k_{i \mod ||k||}, m_i)$ → slide 16
  - → A Vigenère ciphertext is a merge of $||k||$ Caesar ciphertexts:
    $C_0 = \{c_i : i \mod ||k|| = 0\}$,
    $C_1 = \{c_i : i \mod ||k|| = 1\}$,
    $\dots$,
    $C_{||k||-1} = \{c_i : i \mod ||k|| = ||k|| - 1\}$
- **If** the key length $||k||$ is known
  - Perform frequency analysis on each Caesar ciphertext C_n individually
  - Determine shift/key for each Caesar ciphertext
  - → Combine keys of Caesar ciphertexts into Vigènere key
- How to find the key length?
  - Kasiski's method
  - Auto-correlation

Attacks on the Vigenère cipher: Kasiski's method I

- Remember: Key repeats every $||k||$ positions

$\rightarrow$ Same plaintext words results in same ciphertext words if they are an integer multiple of $||k||$ apart

```
K QRQG BR QQFH XQ ADOG MRY NOXKJ: HKMPUV RQK, WLCH EICF D AGWJLVM
DRF O VITWRYU PUSY, GDH, JWJL, CBG AQFNMPU, IYNZ RJ UHDXG OQH
YCH, WWQK RQPOI UQHRGG DW FFDA VVH IAS WS HZRA, YS QSY DUIUSQX.
VVRWG HKEV QDR RWWC, JSUI OOB, MH HKIA HKMPY LX YSOP, NSW JCZO E
VSDV; VVH WWPMIEH ZMNZ GIUSUZG WW. WWQK EU ULZG HKIKF PSPSB SWH
RJ JCSI VVHC OOB FGZLIXS, PEA VHVG TLRF HUYVV WSQ. HKSUS WLCH
FSOS WS USH SPZB E UVRA QF WAQ, OQH UC DKTSH XJS SPCM PEA DDWU,
WI XJSB FG GWMNZ DRF KLPNWQK, K'ZO YPRHVVONI OOB WGS DACM WLGWU
WJWOPKBJ VKQKPA WQ XYC VLQFW LQIUW. QBOC VVHC VVDX ECPI VC KICF D
QGFUC DOZHA DOEA, O QSKGH SH HDVISWW, QF WS USH E HSOPQK LR C
ZRRI ARXNSB GQOW KWOUHGR ZMVV BINZRA, YWOP DS GIESLZGR; IST,
UHRVZH LGOUITG, NRQK, WS TOQO QIU GJCVIP HUYVV ZMVV VYEV D WJCZ
[..]
```

Attacks on the Vigenère cipher: Kasiski's method II

- Remember: Key repeats every $||k||$ positions

$\rightarrow$ Same plaintext words results in same ciphertext words if they are an integer multiple of $||k||$ apart

```
K QRQG BR QQFH XQ ADOG MRY NOXKJ: HKMPUV RQK, WLCH EICF D AGWJLVM
DRF O VITWRYU PUSY, GDH, JWJL, CBG AQFNMPU, IYNZ RJ UHDXG OQH
YCH, WWQK RQPOI UQHRGG DW FFDA VVH IAS WS HZRA, YS QSY DUIUSQX.
VVRWG HKEV QDR RWWC, JSUI OOB, MH HKIA HKMPY LX YSOP, NSW JCZO E
VSDV; VVH WWPMIEH ZMNZ GIUSUZG WW. WWQK EU ULZG HKIKF PSPSB SWH
RJ JCSI VVHC OOB FGZLIXS, PEA VHVG TLRF HUYVV WSQ. HKSUS WLCH
FSOS WS USH SPZB E UVRA QF WAQ, OQH UC DKTSH XJS SPCM PEA DDWU,
WI XJSB FG GWMNZ DRF KLPNWQK, K'ZO YPRHVVONI OOB WGS DACM WLGWU
WJWOPKBJ VKQKPA WQ XYC VLQFW LQIUW. QBOC VVHC VVDX ECPI VC KICF D
QGFUC DOZHA DOEA, O QSKGH SH HDVISWW, QF WS USH E HSOPQK LR C
ZRRI ARXNSB GQOW KWOUHGR ZMVV BINZRA, YWOP DS GIESLZGR; IST,
UHRVZH LGOUITG, NRQK, WS TOQO QIU GJCVIP HUYVV ZMVV VYEV D WJCZ
[..]
```

Attacks on the Vigenère cipher: Kasiski's method III

- Terminology:
  - Bigram[3]: *a group of two successive letters* → slide 29
  - Trigram[4]: *a cluster of three successive letters* → slide 30
- In long texts, the same bigram/trigram/etc. appears at positions which are an integer multiple of $||k||$

→ Differences in positions are very likely multiples of the key length

→ Greatest common divisor (*gcd*) of the position differences is very likely the key length or a multiple thereof

[3] `https://www.merriam-webster.com/dictionary/bigram` (accessed on August 18, 2022) and `https://www.merriam-webster.com/dictionary/digraph` (accessed on August 18, 2022)

[4] `https://www.merriam-webster.com/dictionary/trigram` (accessed on August 18, 2022) and `https://www.merriam-webster.com/dictionary/trigraph` (accessed on August 18, 2022)

Attacks on the Vigenère cipher: Kasiski's method IV

- `WS` bigram from example ($\rightarrow$ slide 29):
  - At positions 126, 294, 470 and 550
  - $\rightarrow$ Position differences 168, 80 and 80
- `OQH` trigram from example ($\rightarrow$ slide 30):
  - At positions 93 and 313
  - $\rightarrow$ Position difference 220
- $gcd(168, 80, 80, 220) = 4$

$\rightarrow$ Key length is likely 4 or a multiple of 4

- Alternative approach: Perform frequency analysis of the factors of the position differences (e.g., of all bigrams)

$\rightarrow$ More/most frequent factors are very likely factors of the key length

Attacks on the Vigenère cipher: Auto-correlation I

- Remember: Key repeats every $||k||$ positions

$\rightarrow$ Ciphertext characters which are an integer multiple of the key length apart are more likely to be the same

$\rightarrow$ Compute incidence of identical characters (coincidence) at ciphertext character positions $i + l$ for all ciphertext characters c_i with any plausible key length l

$\rightarrow$ Compute incidence I_l for all plausible key lengths $l \in \mathbb{N}^+$

$\rightarrow$ Key length $\arg\max_l I_l$ with highest incidence is very likely the key length or a multiple thereof

Attacks on the Vigenère cipher: Auto-correlation II

- Illustration of character coincidence (reference: R at position 2):

K QRQG BR QQFH XQ ADOG MRY NOXKJ: HKMPUV RQK, WLCH EICF D AGWJLVM DRF O VITWRYU PUSY, [..]

  - (Correctly) assuming key length of $l = 4$

→ 3 out of 15 matches (20%)

K QRQG BR QQFH XQ ADOG MRY NOXKJ: HKMPUV RQK, WLCH EICF D AGWJLVM DRF O VITWRYU PUSY, [..]

  - (Incorrectly) assuming key length of $l = 3$

→ 0 out of 20 matches (0%)

- Expected coincidence: $\frac{1}{26} \approx 4\%$ (in longer texts)

Thank you for your attention!

Questions?