Introduction and Classical Cryptography

ENCS4320 - Applied Cryptography

Dr. Ahmed I. A. Shawahna

Electrical and Computer Engineering Department

Birzeit University

Presentation Outline

- **Introduction to Cryptography**
- The Setting of Symmetric/Asymmetric Encryption
- Historical Ciphers and Their Cryptanalysis
- Principles of Modern Cryptography

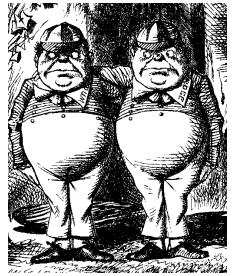
The Cast of Characters

- Alice and Bob are the **good guys**

Alice

Bob

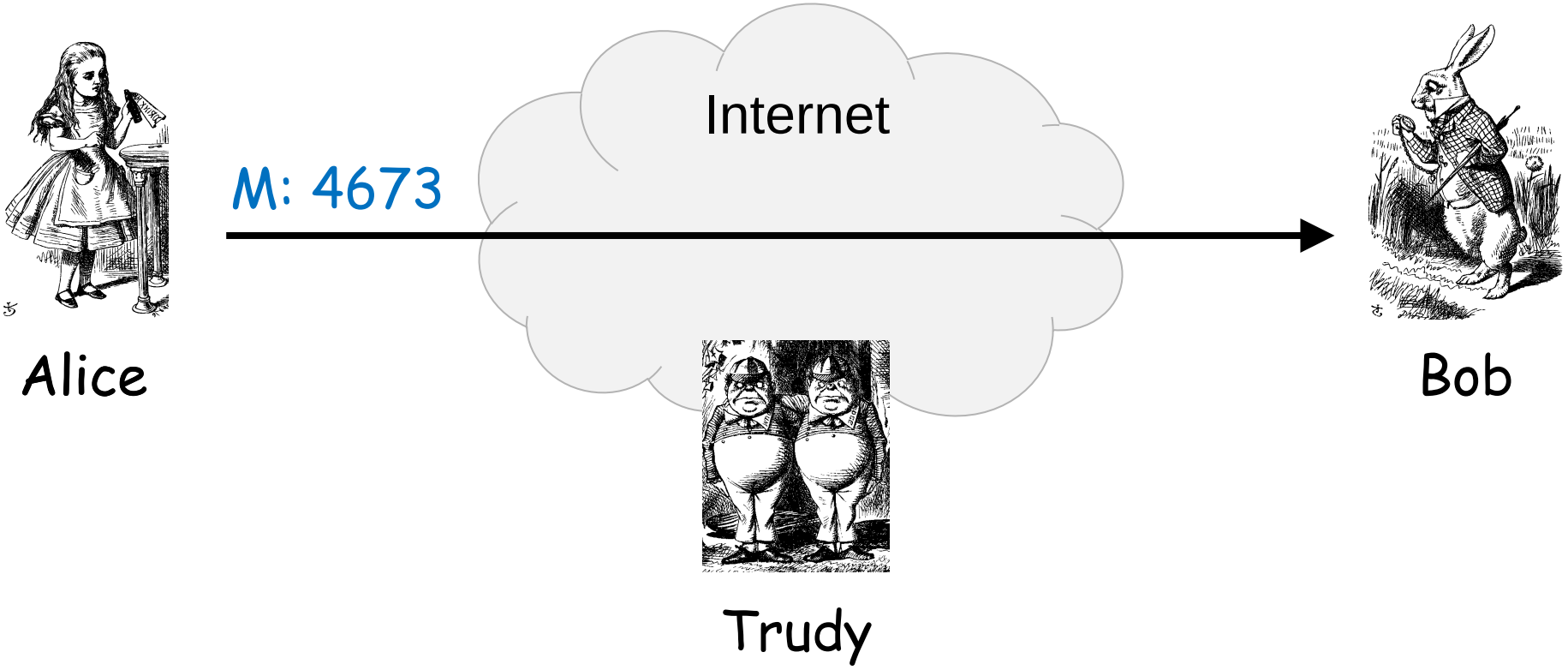
- Trudy is the **bad** "**guy**"



Trudy

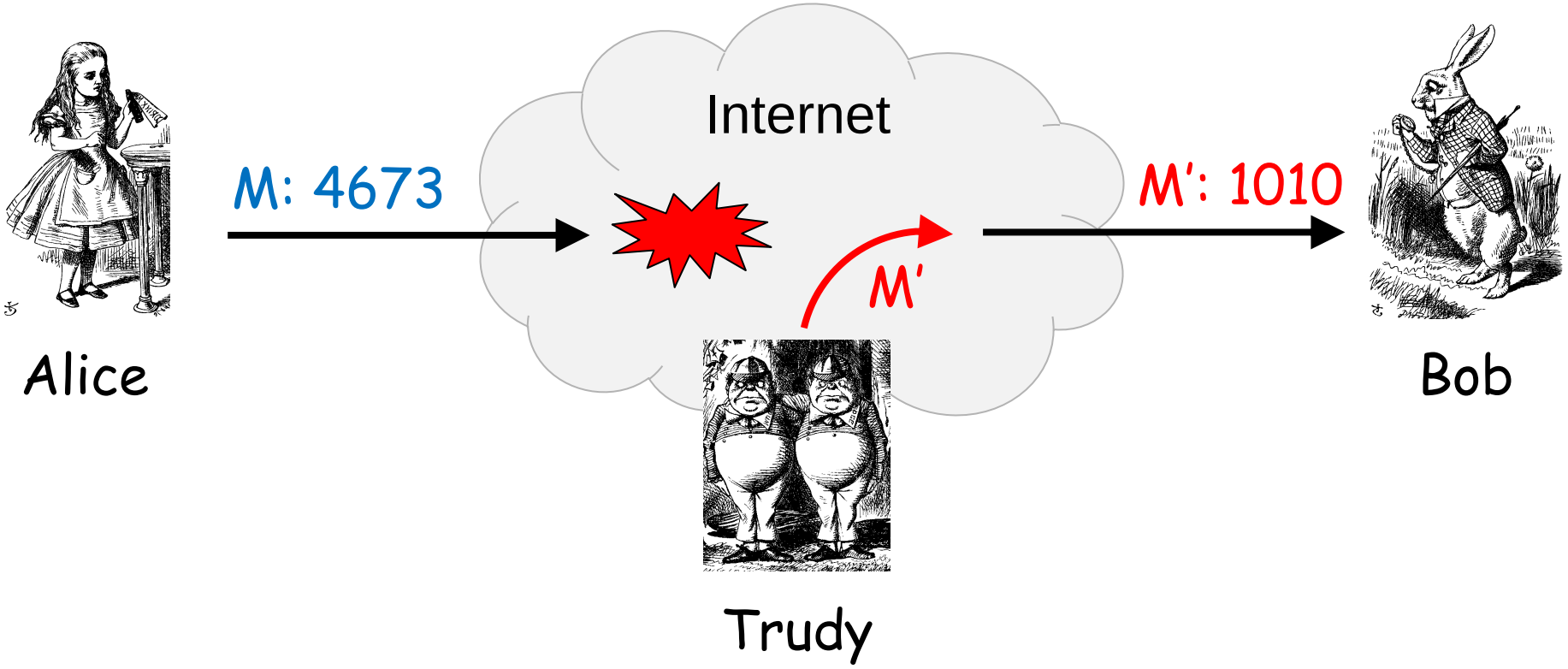
- Trudy is our generic "intruder/adversary"

What Is Cryptography?



- Trudy should not be able to read message M

What Is Cryptography?



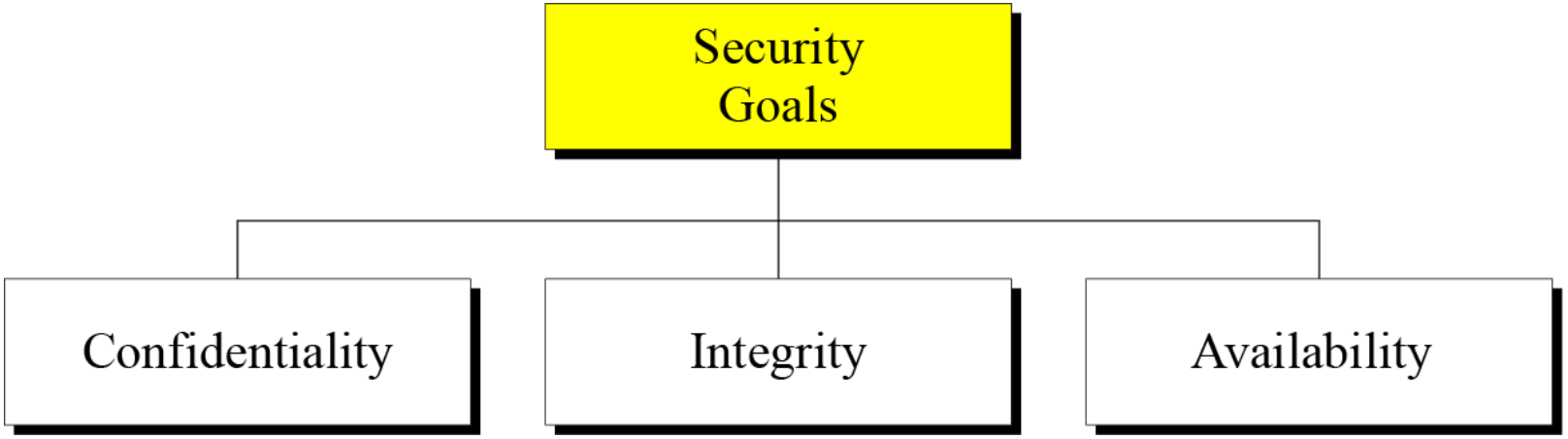
- Trudy should not be able to modify message M
- Message M should be actually originated from Alice

Alice's Online Bank

- Alice opens Alice's Online Bank (AOB)
- What are Alice's security concerns?
- If Bob is a customer of AOB, what are his security concerns?
- How are Alice's and Bob's concerns similar? How are they different?
- How does Trudy view the situation?

Security Goals – CIA

- **CIA** == **C**onfidentiality, **I**ntegrity, and **A**vailability



- Additional security goal:
  - Authenticity

CIA – Confidentiality and Integrity

- AOB must prevent Trudy from learning Bob's account balance
- **Confidentiality (privacy):** prevent unauthorized *reading* of information
  - Cryptography used for confidentiality

- Trudy must not be able to change Bob's account balance
- Trudy must not be able to improperly change her own account balance
- **Integrity:** detect unauthorized *writing* of information (i.e., modification of data)
  - Cryptography used for integrity

CIA – Availability (and Authenticity)

- AOB's information must be available whenever it's needed
- Bob must be able to make transaction
  - If not, he'll take his business elsewhere
- **Availability:** Data is available in a timely manner when needed
- Availability is a "new" security concern
  - Denial of service (DoS) attacks

- **Authenticity:** Data is indeed what it claims to be or what it is claimed to be

Beyond CIA: Crypto

- How does Bob's computer know that "Bob" is really Bob and not Trudy?
- Bob's password must be verified
  - This requires some clever **cryptography**
- What are security concerns of passwords?
- Are there alternatives to passwords?

Beyond CIA: Protocols

- When Bob logs into AOB, how does AOB know that "Bob" is really Bob?
- As before, Bob's password is verified
- Unlike the previous case, **network** security issues arise
- How do we secure network transactions?
  - **Protocols** are critically important
  - Crypto plays critical role in security protocols

Beyond CIA: Access Control

- Once Bob is *authenticated* by AOB, then AOB must restrict actions of Bob
  - Bob can't view Charlie's account info
  - Bob can't install new accounting software, etc.
- Enforcing these restrictions: *authorization*
- **Access control** includes both authentication and authorization

The People Problem

- People often break security
  - Both intentionally and unintentionally
  - Here, we consider the unintentional
- For example, suppose you want to buy something online
  - To make it concrete, suppose you want to buy *Introduction to Modern Cryptography, 3rd edition* from amazon.com
- To buy from amazon.com…
  - Your Web browser uses secure sockets layer (SSL) protocol
  - SSL relies on cryptography
  - Access control issues arise
  - All security mechanisms are in software

The People Problem

- Suppose all of this security stuff works perfectly
  - Then you would be safe, right?
- What could go wrong?
- Trudy tries man-in-the-middle attack
  - SSL is secure, so attack doesn't "work"
  - But, Web browser issues a warning
  - What do you, the user, do?
- If user ignores warning, attack works!
  - None of the security mechanisms failed
  - But user *unintentionally* broke security

Think Like Trudy

- In the past, no respectable sources talked about “hacking” in detail
  - After all, such info might help Trudy
- Recently, this has changed
  - Lots of books on network hacking, evil software, how to hack software, etc.
  - Classes teach virus writing, software reverse engineering (SRE), etc.
- Good guys must think like bad guys!
  - Find the ***weak link*** **before** Trudy does
- A police detective …
  - … must study and understand criminals

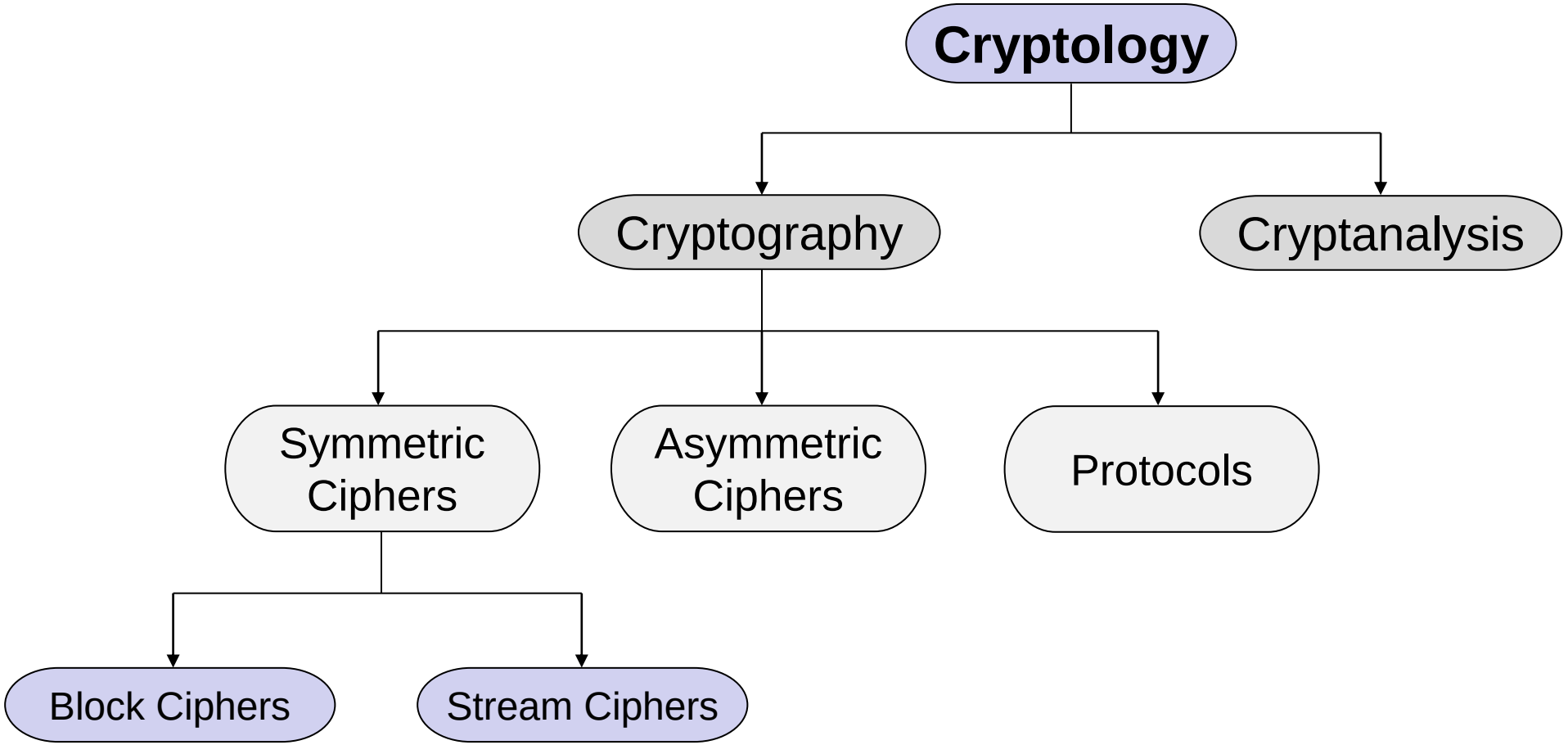
Think Like Trudy

- In information security
  - We want to understand Trudy's methods
  - Might think about Trudy's motives
  - We'll often pretend to be Trudy
- We must try to think like Trudy
- We must study Trudy's methods
- We can admire Trudy's cleverness
- Often, we can't help but laugh at Alice's and/or Bob's stupidity
- But, we **cannot** act like Trudy
  - Except in this class …
  - But don't do anything illegal!

Security Jargons

- **Cryptology** — The art and science of making and breaking "secret codes"
  - Codes: encryption schemes
  - Secret: a key
- **Cryptography** — making "secret codes"
  - Involves the study of mathematical techniques for securing digital information, systems, and distributed computations against adversarial attacks
- **Cryptanalysis** — breaking "secret codes"
- **Crypto** — all of the above (and more)

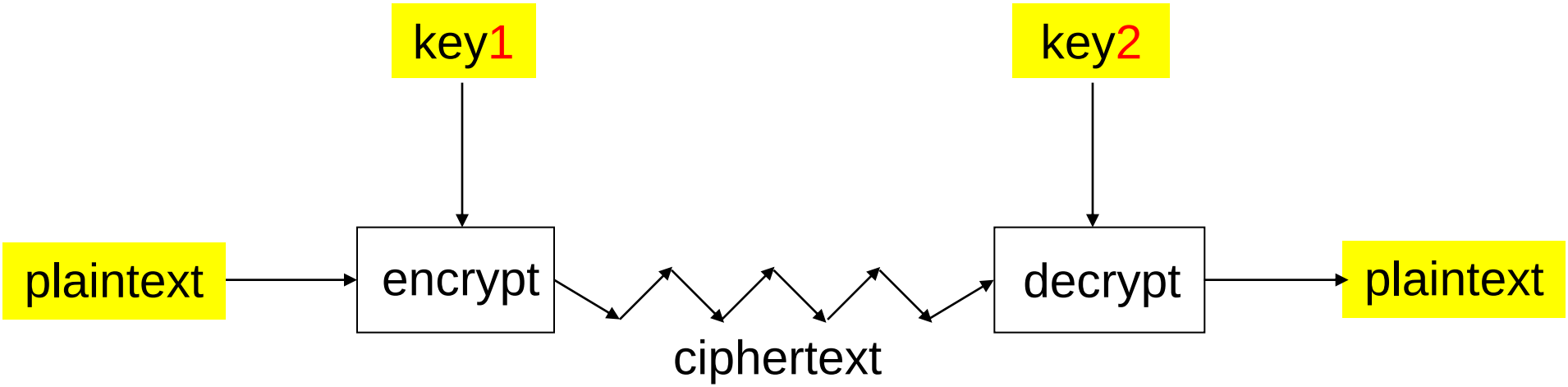
Cryptology



How to Speak Crypto

- A *cipher* or *cryptosystem* is used to *encrypt* the *plaintext*
- The result of encryption is *ciphertext*
- We *decrypt* ciphertext to recover plaintext
- A *key* is used to configure a cryptosystem
- A ***symmetric key*** *(private key)* cryptosystem uses the *same key* to encrypt as to decrypt
  - Key must be kept secret!
  - The two communication parties are assumed to have been able to securely share a key in advance of their communication
- An ***asymmetric key*** *(public key)* cryptosystem uses a *public key* to encrypt and a *private key* to decrypt
  - Both keys must be kept secret?

Cryptographic Schemes



Crypto	Keys
Symmetric Key	key1 = key2
Public Key	key1 ≠ key2

How do keys get distributed? Magic, for now!

Crypto as Black Box

- Our concerns:
  - How to define security goals?
  - How to design encryption and decryption algorithms?
  - How to gain confidence that encryption and decryption algorithms achieve our goals?

Basic Goals of Cryptography

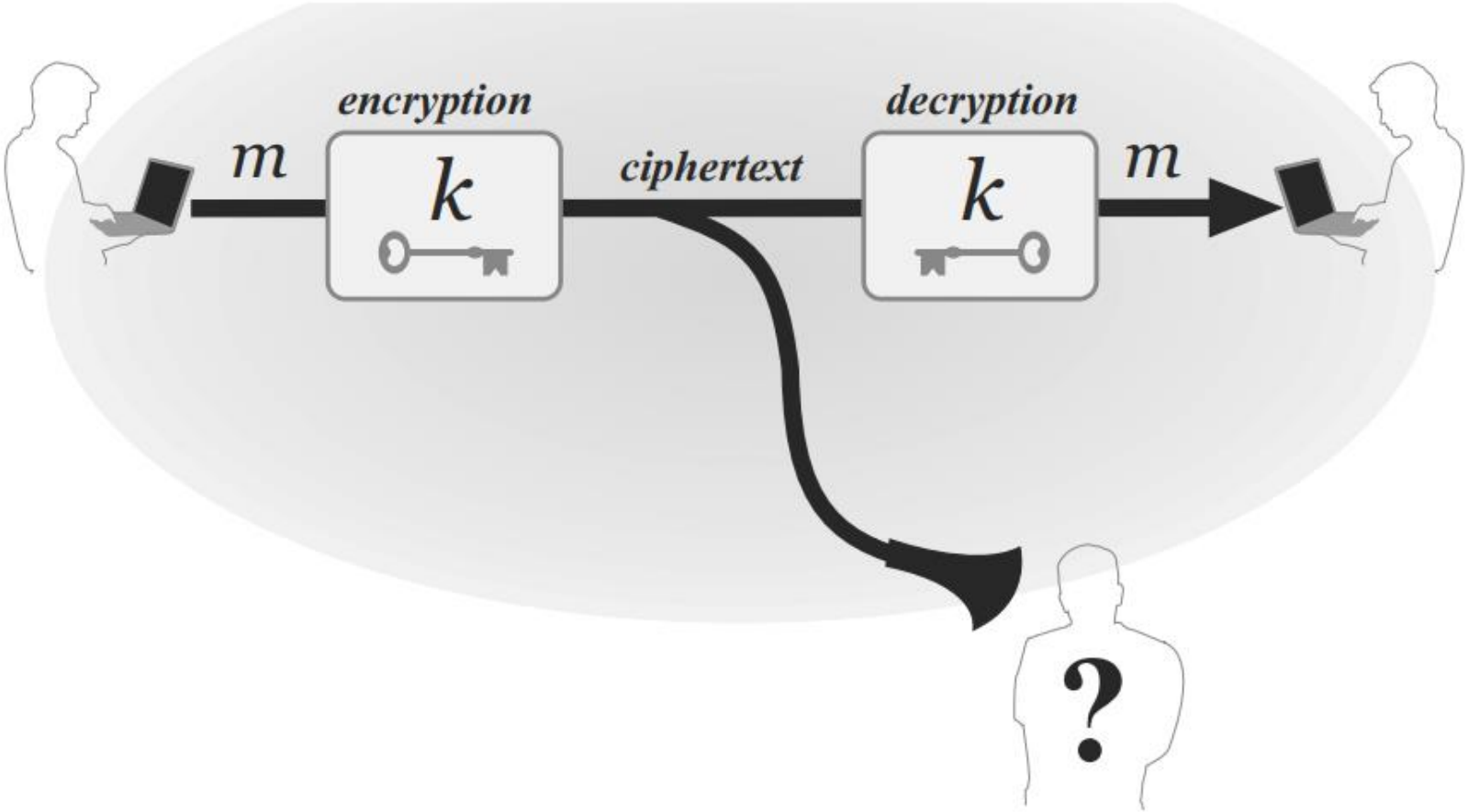
	Message Privacy	Message Integrity / Authentication
Symmetric Keys	Symmetric Encryption (private-key encryption)	Message Authentication Codes (MAC)
Asymmetric Keys	Asymmetric Encryption (public-key encryption)	Digital Signatures

Presentation Outline

- Introduction to Cryptography
- **The Setting of Symmetric/Asymmetric Encryption**
- Historical Ciphers and Their Cryptanalysis
- Principles of Modern Cryptography

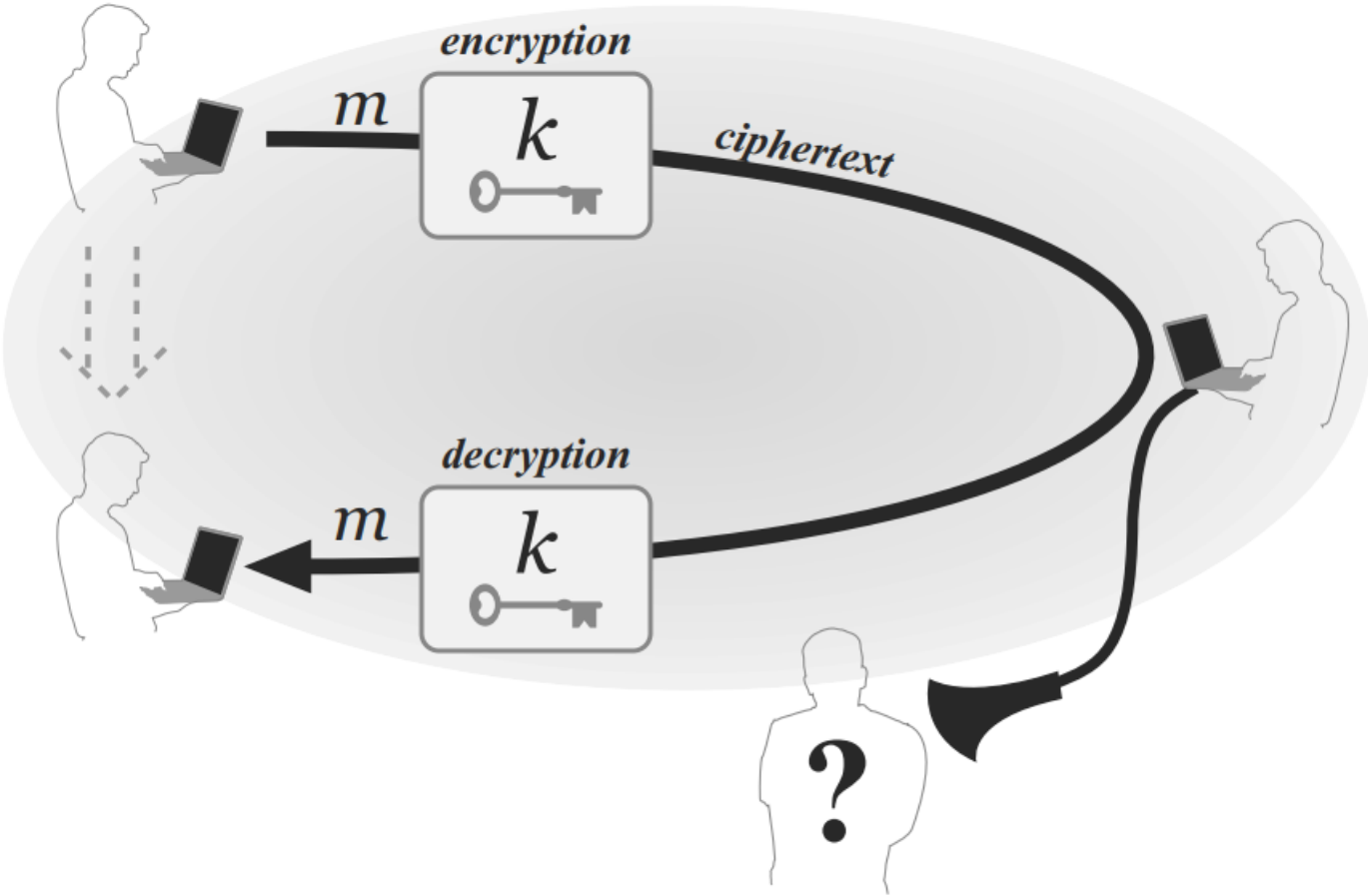
Applications of Private-Key Cryptography

- Two parties share a key that they use to communicate securely
  - For example, a worker in Ramallah communicating with her colleague in Jenin



Applications of Private-Key Cryptography

- Same party communicating with itself over time
  - For example, disk encryption, where a user encrypts some plaintext and stores the resulting ciphertext on his hard drive



Crypto

- Basic assumptions
  - The system is completely known to the attacker
    - Encryption and decryption algorithms are standardized, implemented, and public!
  - Only the key is secret (and, obviously, the plaintext!)
  - That is, crypto algorithms are **not** secret!
- This is known as **Kerckhoffs'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."*
- Why do we make these assumptions?
  - It is significantly easier to maintain secrecy of a short key than to keep secret a (more complicated) encryption scheme
  - Experience has shown that secret algorithms are weak when exposed
  - Secret algorithms never remain secret
  - Better to find weaknesses beforehand

Notation

- ∈ : "element in"
  - $3 \in \{1,2,3,4,5\}$
  - $7 \notin \{1,2,3,4,5\}$
- ∀ : "for all"
  - "$\forall X \in \{0,1\}^4$ …" = "for all bitstrings of length 4 …"
- ∃ : "there exists"
  - "$\exists X \in \{0,1,2, \ldots\}$ such that $X > 13$"
- $X \leftarrow 5$: "assign value 5 to X"
- $\mathcal{X} \times \mathcal{Y}$: set of pairs (X, Y) with $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$
- $X \xleftarrow{\$} \mathcal{X}$: "assign X a random value from set $\mathcal{X}$"

Notation

- **$\{0,1\}^n$** : set of all bitstrings of length n
  - $011 \in \{0,1\}^3$
  - $011 \notin \{0,1\}^5$
- **$\{0,1\}^*$** : set of all bitstrings of finite length
  - $1, 1001, 10, 10001101000001 \in \{0,1\}^*$
- **$F : \mathcal{X} \rightarrow \mathcal{Y}$** : function from set $\mathcal{X}$ to set $\mathcal{Y}$
  - $F : \{0,1\}^5 \rightarrow \{0,1\}^3$
  - G : $\{A,B,C,D\} \rightarrow \{0,1,2,\ldots\}$

Notation

- Symmetric Key Cryptography (Symmetric Encryption):
  - The *key-generation algorithm* (Gen): a probabilistic algorithm that outputs a key k, $k \in \mathcal{K}$, chosen according to some distribution. $\mathcal{K}$ is the set of all possible keys that can be output by Gen
  - The *encryption algorithm* (Enc): encrypt message m, $m \in \mathcal{M}$, using the key k

    $\mathsf{Enc} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$, $\mathsf{Enc}(k, m) = \mathsf{Enc}_k(m) = c$

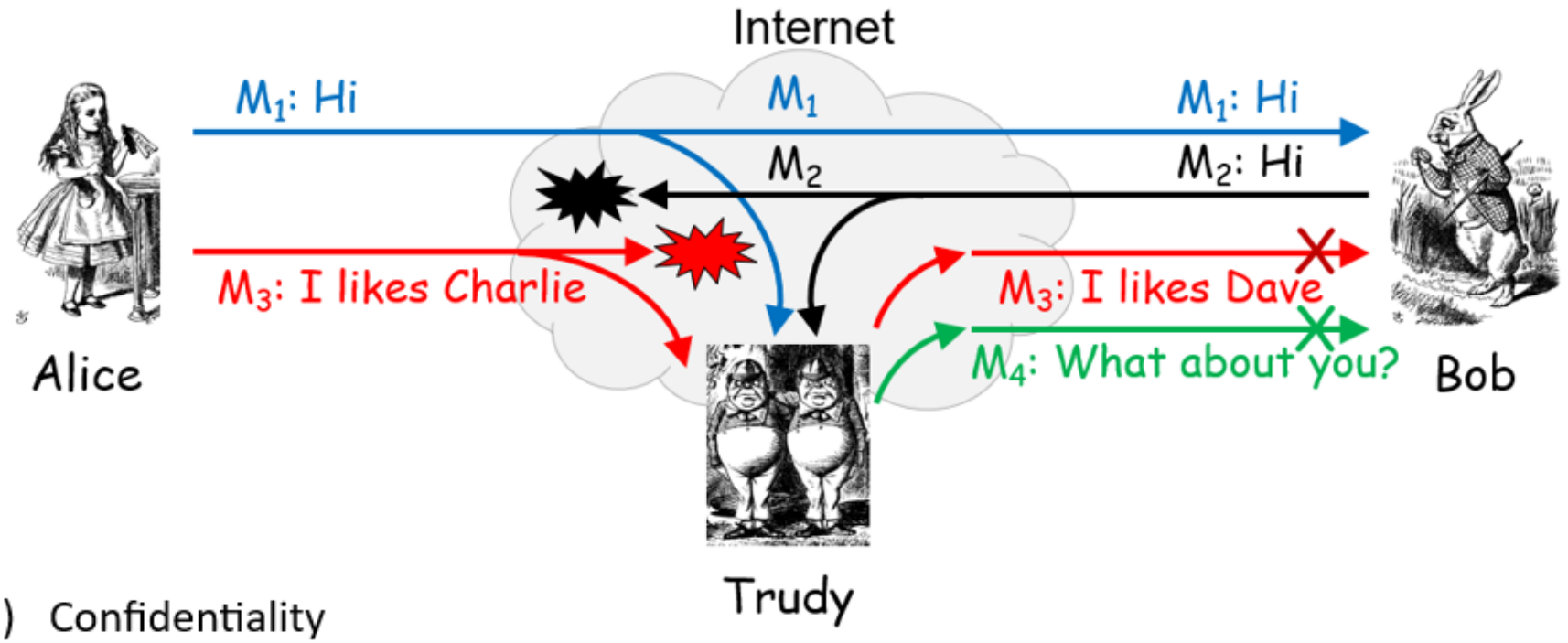
    where, $\mathcal{C}$ denote the set of all possible ciphertexts that can be output by $\mathsf{Enc}_k(m)$
  - The *decryption algorithm* (Dec): decrypt ciphertext c, $c \in \mathcal{C}$, using the key k

    $\mathsf{Dec} : \mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$, $\mathsf{Dec}(k, c) = \mathsf{Dec}_k(c) = m$

Notation

- Correctness requirement:
  - For every key k output by Gen(.), $\forall k \in \mathcal{K}$, and every message $m \in \mathcal{M}$ (the set of "legal" messages, i.e., those supported by the scheme), it holds that $\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$
- Examples:
  - $\mathcal{K} = \{0,1\}^{128}$ $\mathcal{M} = \{0,1\}^*$ $\mathcal{C} = \{0,1\}^*$
  - $\mathcal{K} = \{0,1\}^{128}$ $\mathcal{M} = \{A, B, \dots, Z\}$ $\mathcal{C} = \{A, B, \dots, Z\}$
  - $\mathcal{K} = \{0,1\}^{128}$ $\mathcal{M} = \{\mathrm{YES}, \mathrm{NO}\}$ $\mathcal{C} = \{0,1\}^*$
  - $\mathcal{K} = \{1, \dots, p\}$ $\mathcal{M} = \{A, B, \dots, Z\}$ $\mathcal{C} = \{0,1\}^*$
- Public Key Cryptography (Asymmetric Encryption):
  - Sign message M with Alice's **private key** $[M]_{Alice}$
  - Encrypt message M with Bob's **public key** $\{M\}_{Bob}$

As shown in the figure below, we would like to design a communication system that allows Alice and Bob to exchange messages. Alice and Bob *do not mind* if the system allows Trudy to *read* their original messages, they *do not even mind* if the system *does not work consistently*. But the system *must prevent* Trudy from *modifying* their messages or *creating* messages in their names. Which of these security goals are required when designing this system? **(Select all that apply)**



a) Confidentiality
b) Integrity
c) Authenticity
d) Availability
e) None

In symmetric key cryptography, which of the following should be secret based on Kerckhoffs's Principle?

a) Ciphertext
b) Encryption/Decryption key
c) Encryption algorithm
d) Decryption algorithm
e) Both encryption and decryption algorithms, and the encryption/decryption key

Presentation Outline

- Introduction to Cryptography
- The Setting of Symmetric/Asymmetric Encryption
- **Historical Ciphers and Their Cryptanalysis**
- Principles of Modern Cryptography

Caesar's Cipher

- Plaintext is encrypted by shifting the letters of the alphabet 3 places forward

- Plaintext: begin the attack now
- Key:

Plaintext	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
Ciphertext	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

- Ciphertext: EHJLQWKHDWWDFNQRZ

Caesar's Cipher Decryption

- Suppose we know a Caesar's cipher is being used:

Plaintext	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
Ciphertext	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

- Given ciphertext: EHJLQWKHDWWDFNQRZ
- Plaintext: begintheattacknow

- An immediate problem with Caesar's Cipher is that the encryption method is fixed; there is no key
  - Thus, anyone learning how Caesar encrypted his messages would be able to decrypt effortlessly

Caesar's Cipher: Encryption/Decryption

❖ Assign each letter a numerical value:

Plaintext	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
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

❖ Encryption: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

$C \leftarrow M + 3 \pmod{26}$

❖ Decryption: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

$M \leftarrow C - 3 \pmod{26}$

$\mathcal{K} = \{\,\}$

$\mathcal{M} = \{0,1,2,\ldots,25\}^*$

$\mathcal{C} = \{0,1,2,\ldots,25\}^*$

Math Basics - Modular Arithmetic

- For integers x and n, "x mod n" is the remainder when we compute $x \div n$
  - We can also say "x modulo n"

Definition: $x \bmod n$

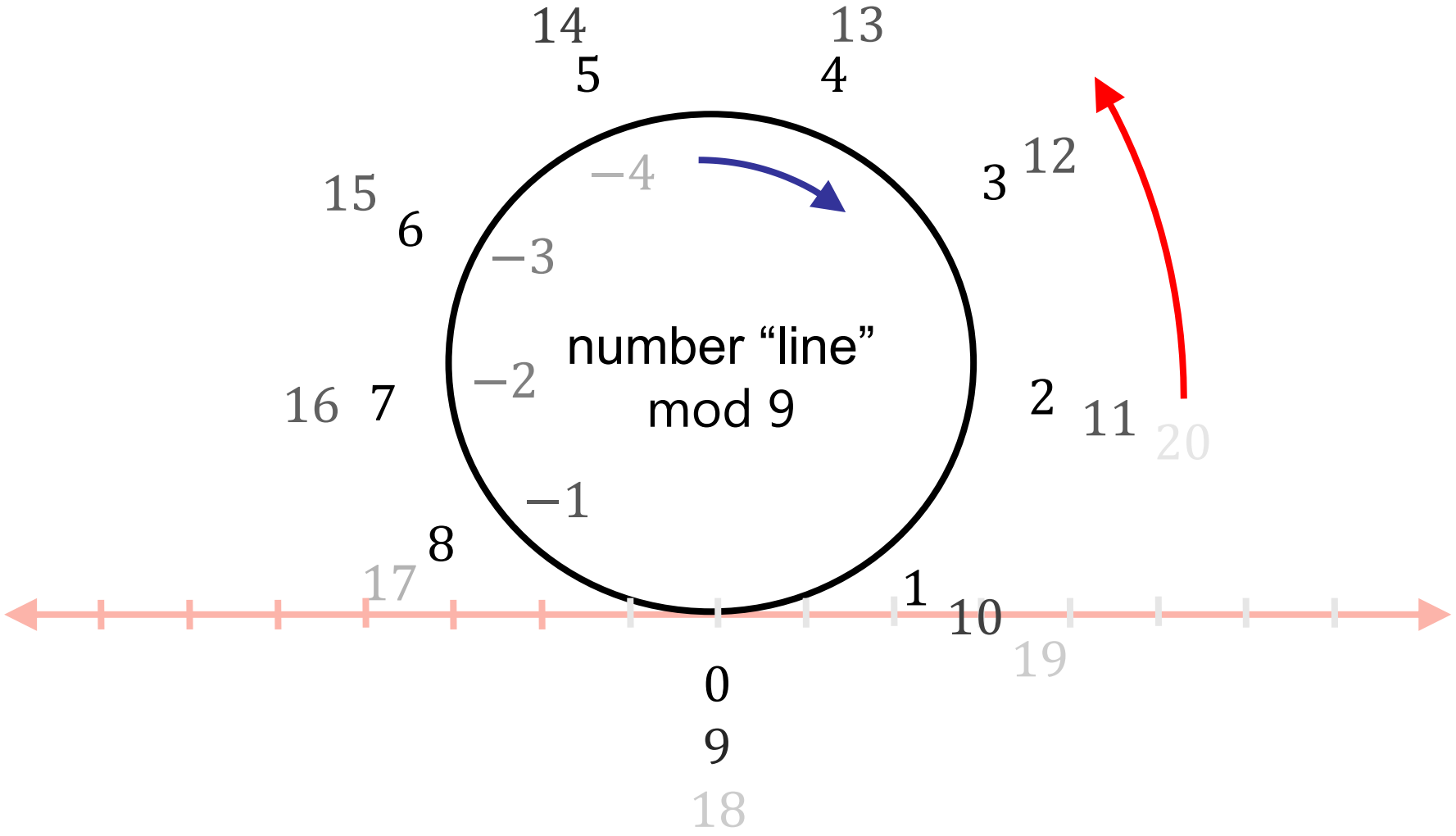
is the *unique* integer $0 \leq r < n$ such that

$$x = q \cdot n + r$$

- Examples
  - $7 \bmod 6 \equiv 1$, $33 \bmod 5 \equiv 3$
  - $17 \bmod 6 \equiv 5$, $33 \bmod 6 \equiv 3$
  - $51 \bmod 17 \equiv 0$, $947 \bmod 6018 \equiv 947$

Math Basics - Modular Arithmetic

Clock Arithmetic



Math Basics - Modular Arithmetic

- Modular Addition - Notation and facts
  - ((a mod n) + (b mod n)) mod n = (a + b) mod n

- Examples
  - $3 + 5 \equiv 2 \bmod 6$
  - $2 + 4 \equiv 0 \bmod 6$
  - $3 + 3 \equiv 0 \bmod 6$
  - $(7 + 12) \bmod 6 = 19 \bmod 6 \equiv 1 \bmod 6$
  - $(7 + 12) \bmod 6 \equiv ([7 \bmod 6] + [12 \bmod 6]) \bmod 6$
    $\equiv (1 + 0) \bmod 6 = 1 \bmod 6$

Math Basics - Modular Arithmetic

- Modular Multiplication - Notation and facts
  - $((a \bmod n) \cdot (b \bmod n)) \bmod n = a \cdot b \bmod n$

- Examples
  - $3 \cdot 4 = 12 \equiv 0 \pmod 6$
  - $2 \cdot 4 = 8 \equiv 2 \pmod 6$
  - $5 \cdot 5 = 25 \equiv 1 \pmod 6$
  - $(7 \cdot 4) \bmod 6 = 28 \bmod 6 \equiv 4 \bmod 6$
  - $28 \bmod 6 = (7 \cdot 4) \bmod 6 \equiv ([7 \bmod 6] \cdot [4 \bmod 6]) \bmod 6$
    $\equiv (1 \cdot 4) \bmod 6 = 4 \bmod 6$

Math Basics - Modular Arithmetic

- Additive inverse of $x \bmod n$, denoted $-x \bmod n$, is the number that must be added to x to get $0 \bmod n$
  - $-2 \bmod 6 \equiv 4$, since $2 + 4 \equiv 0 \bmod 6$

- **Modular Arithmetic Quiz**
- Q: What is $-3 \bmod 6$?
  - 3
- Q: What is $-14 \bmod 6$?
  - 4

More details:
https://en.wikibooks.org/wiki/High_School_Mathematics_Extensions/Primes/Modular_Arithmetic

Which of the following is congruent to **(-25) mod 7** ?

a) ([(20) mod 7] + [(-10) mod 7]) mod 7
b) (-5) mod 7
c) (20) mod 7 + (-10) mod 7
d) ([(14) mod 7] · [(-4) mod 7]) mod 7
e) None

ROT-13

- A variant of Caesar's Cipher where the shift is 13 places instead of 3
- Key:

Plaintext	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
Ciphertext	N	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

- Plaintext:

  in the far distance a helicopter skimmed down between the roofs, hovered for an instant like a bluebottle, and darted away again with a curving flight. It was the police patrol, snooping into people's windows

- Ciphertext:

  VA GUR SNE QVFGNAPR N URYVPBCGRE FXVZZRQ QBJA ORGJRRA GUR EBBSF, UBIRERQ SBE NA VAFGNAG YVXR N OYHROBGGYR, NAQ QNEGRQ NJNL NTNVA JVGU N PHEIVAT SYVTUG. VG JNF GUR CBYVPR CNGEBY, FABBCVAT VAGB CRBCYR'F JVAQBJF

ROT-13: Encryption/Decryption

❖ Assign each letter a numerical value:

Plaintext	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
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

❖ Encryption: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

$C \leftarrow M + 13 \pmod{26}$

$\mathcal{K} = \{\}$

$\mathcal{M} = \{0,1,2,\ldots,25\}^*$

$\mathcal{C} = \{0,1,2,\ldots,25\}^*$

❖ Decryption: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

$M \leftarrow C - 13 \pmod{26}$

The Shift Cipher (ROT-k)

- A keyed variant of Caesar's cipher, letters are shifted as in Caesar's cipher, but now by k places
- Algorithm *Gen* outputs a uniform key $k \in \{0, 1, 2, \ldots, 25\}$
- Example: key $k = 7$

Plaintext	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
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Ciphertext	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G

ROT-k: Encryption/Decryption

- Encryption: $\mathcal{K} \times \mathcal{M} \rightarrow \mathcal{C}$

  $C \leftarrow M + K \pmod{26}$

  - $Enc_k(m_1 \cdots m_L) = c_1 \cdots c_L$, where $c_i = [(m_i + k) \bmod 26]$

- Decryption: $\mathcal{K} \times \mathcal{C} \rightarrow \mathcal{M}$

  $M \leftarrow C - K \pmod{26}$

  - $Dec_k(c_1 \cdots c_L) = m_1 \cdots m_L$, where $m_i = [(c_i - k) \bmod 26]$

$$\mathcal{K} = \{0,1,2,\ldots,25\}$$

$$\mathcal{M} = \{0,1,2,\ldots,25\}^*$$

$$\mathcal{C} = \{0,1,2,\ldots,25\}^*$$

ROT-k: Cryptanalysis I - Try Them All

- Is the shift cipher secure?
  - Decrypt the following ciphertext that was generated using the *shift cipher* and a secret key k
  - Given ciphertext: **OVDTHUFWVZZPISLRLFZHYLAOLYL**
- Is it possible to recover the message without knowing k?
  - It is trivial!
  - Only 26 possible keys — try them all!
  - The correct plaintext will likely be the only candidate on the list of 26 candidate plaintexts that "*makes sense*"
- **Exhaustive-search (or brute-force) attack**
  - An attack that involves trying every possible key
- Solution: key is k = 7
  - Plaintext: **howmanypossiblekeysarethere**

ROT-k: Cryptanalysis I - Try Them All

- Ciphertext: **VA GUR SNE QVFGNAPR N URYVPBCGRE …**

K	M
0	va gur sne qvfgnapr n uryvpbcgre …
1	uz ftq rmd puefmzoq m tqxuoabfqd …
2	ty esp qlc otdelynp l spwtnzaepc …
3	sx dro pkb nscdkxmo k rovsmyzdob …
⋮	
12	jo uif gbs ejtubodf b ifmjdpqufs …
13	**in the far distance a helicopter …**
14	hm sgd ezq chrszmbd z gdkhbnosdq …
⋮	
25	wb hvs tof rwghobqs o vszwqcdhsf …

ROT-k: Cryptanalysis I

- **Conclusion:** key space must be large enough!
- **Sufficient key-space principle:**
  - *Any secure encryption scheme must have a key space that is sufficiently large to make an exhaustive-search attack infeasible*
- What amount of effort makes a task "*infeasible*"?
  - The resources of a potential attacker
  - The length of time for which the sender and receiver want to ensure secrecy of their communication
  - For example, the key space size must be at least 2^{80} due to the use of supercomputers, thousands of cloud servers, or GPUs
- The sufficient key-space principle gives a *necessary* condition for security, but *not* a *sufficient* one

Exercise

Assume an attacker knows that a user's password is either **abcd** or **bedg**. Say the user encrypts his password using the shift cipher, and the attacker sees the resulting ciphertext. Show how the attacker can determine the user's password, or explain why this is not possible.

Solution:

In the shift cipher, the relative shift between characters is preserved. Thus, an encryption of ***abcd*** will always be a ciphertext containing 4 consecutive characters (e.g., ***hijk***), whereas ***bedg*** will not.

Assume an adversary knows that a user's password is either **abcd**, **imko**, **xbzc**, or **bedg**. Say the user encrypts his password using the shift cipher (ROT-K), and the adversary sees the resulting ciphertext. If the adversary sees **CGEH**, can he determine the user's password? If that is possible, what is the encrypted password?

Letter	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
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

a) The adversary cannot determine the user's password
b) abcd
c) imko
d) xbzc
e) bedg

Mono-Alphabetic Substitution Cipher

- In general, simple substitution key can be any **permutation** of letters
  - Not necessarily a shift of the alphabet
- For example:

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

- Given message: tell him about me
- Encrypted message: MXWWYVBJITPMBX
- How many possible keys?
  - 26! ≈ 10^{26} ≈ 2^{88} possible keys!

$\Sigma = \{a, b, c, \ldots, z\}$

$\mathcal{M} = \Sigma^*$

$\mathcal{C} = \Sigma^*$

$\mathcal{K}$ = all permutations on Σ

Permutation Definition

- Let S be a set
- A permutation of S is an ordered list of the elements of S
  - Each element of S appears exactly once
- Suppose $S = \{0,1,2,\ldots,n-1\}$
  - Then the number of perms is …
  - $n*(n-1)*(n-2)* \ldots *(2)*(1) = n!$
- Example: Let $S = \{0,1,2,3\}$
  - Then, there are 24 perms of S
  - For example,
    - $(3,1,2,0)$ is a perm of S
    - $(0,2,3,1)$ is a perm of S, etc.
- Perms are important in cryptography

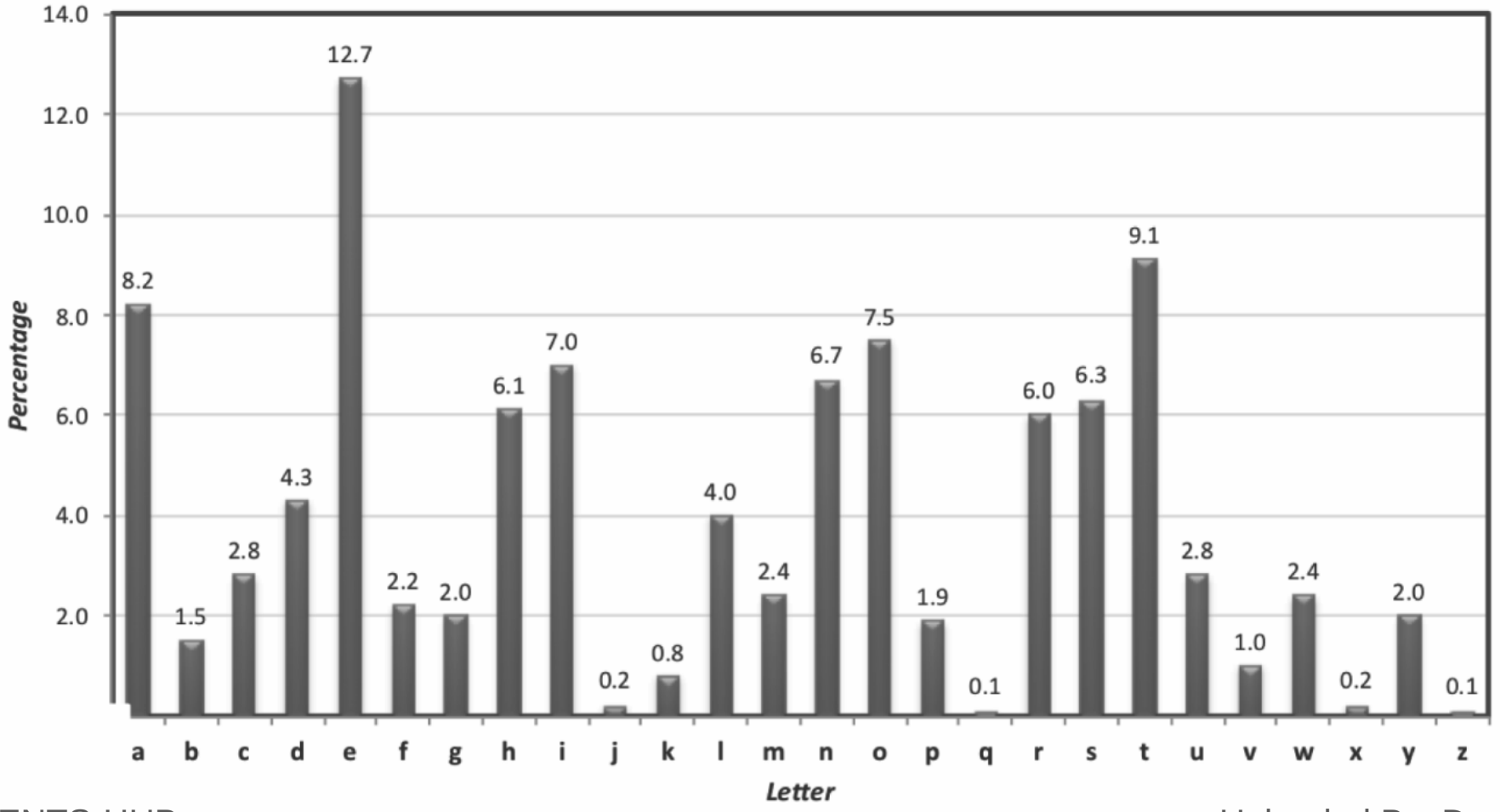
Cryptanalysis II: Be Clever

- We know that a simple substitution is used
- But not necessarily a shift by k
- Find the key given the ciphertext:

jg umw qsn yjtusgdw s mwijdzbuwn tojllwy yzag xwuawwg umw nzzqt, mzpwnwy qzn sg jgtusgu ijow s xicwxzuuiw, sgy ysnuwy sasv sfsjg ajum s dcnpjgf qijfmu. ju ast umw bzijdw bsunzi, tgzzbjgf jguz bwzbiw't ajgyzat

Cryptanalysis II

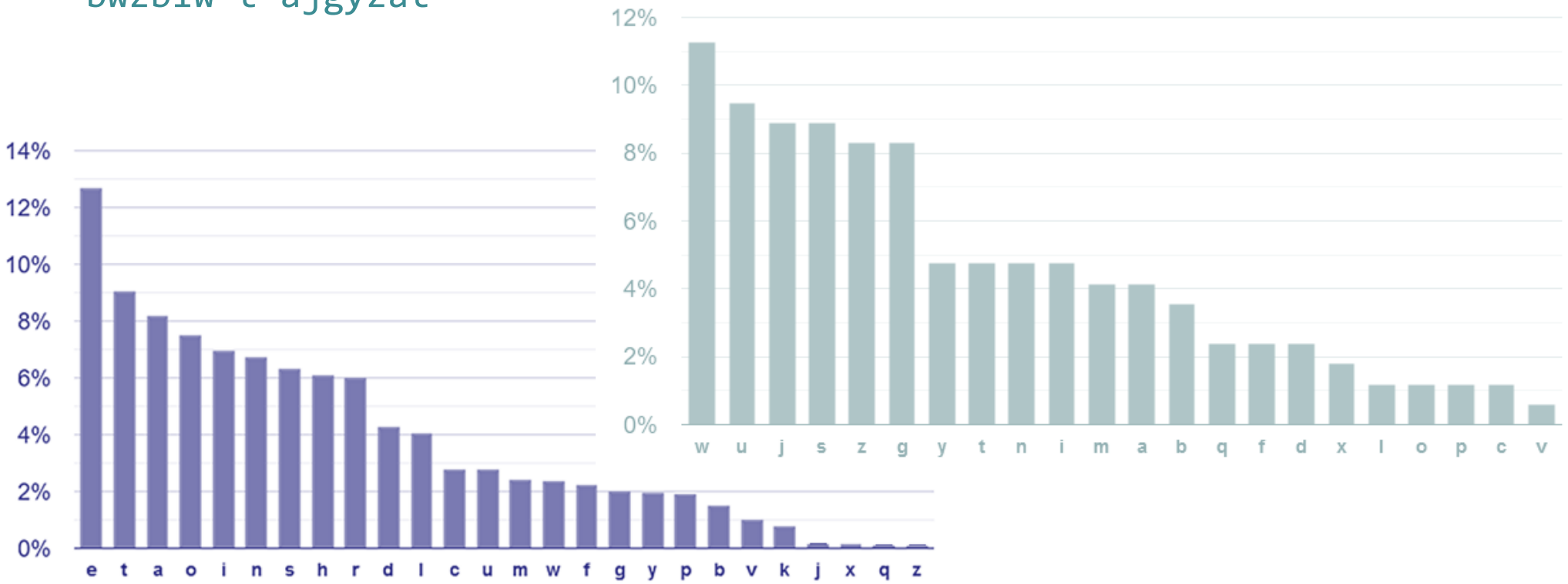
- Cannot try all 2^{88} simple substitution keys
- Can we be more clever? English letter frequency counts…



Cryptanalysis II

- Ciphertext:

```
jg umw qsn yjtusgdw s mwijdzbuwn tojllwy yzag xwuawwg umw nzzqt,
mzpwnwy qzn sg jgtusgu ijow s xicwxzuuiw, sgy ysnuwy sasv sfsjg
ajum s dcnpjgf qijfmu. ju ast umw bzijdw bsunzi, tgzzbjgf jguz
bwzbiw't ajgyzat
```

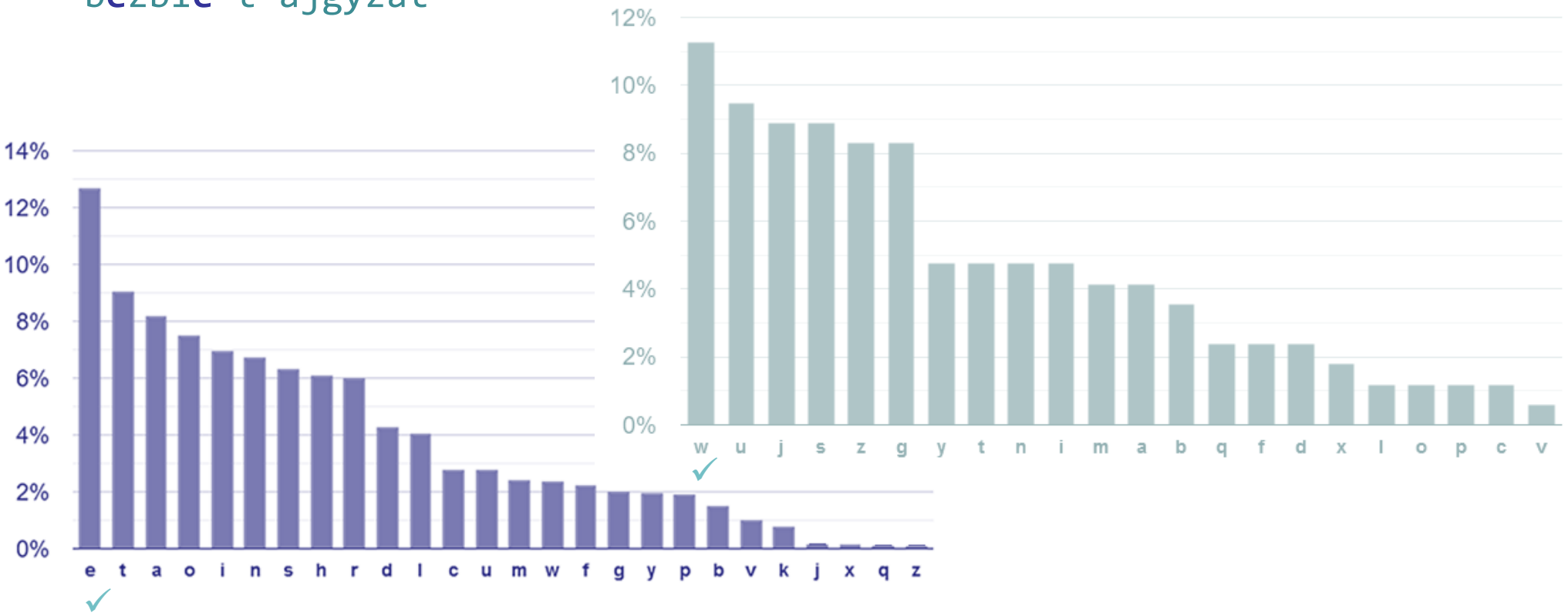


English letter frequency

Cryptanalysis II

- Ciphertext:

```
jg ume qsn yjtusgde s meijdzbuen tojlley yzag xeuaeeg ume nzzqt,
mzpeney qzn sg jgtusgu ijoe s xicexzuuie, sgy ysnuey sasv sfsjg
ajum s dcnpjgf qijfmu. ju ast ume bzijde bsunzi, tgzzbjgf jguz
bezbie't ajgyzat
```

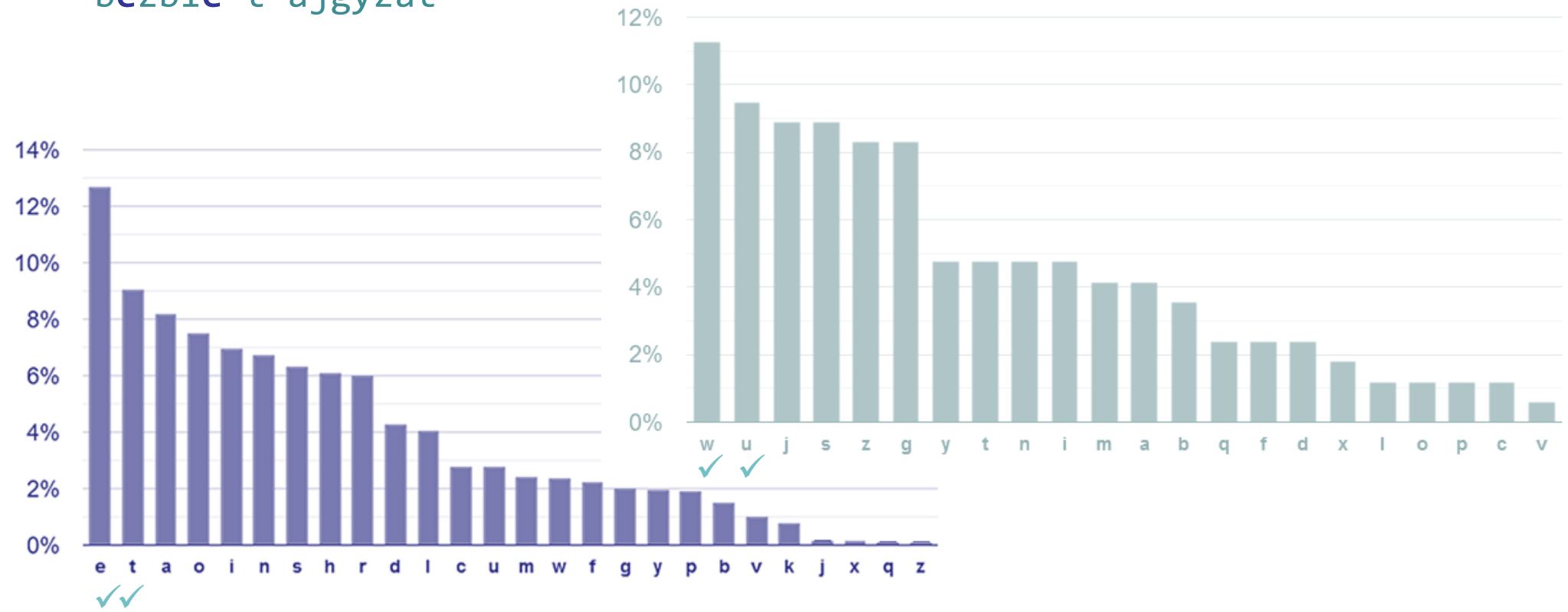


English letter frequency

Cryptanalysis II

- Ciphertext:

```
jg tme qsn yjttsgde s meijdzbten tojlley yzag xetaeeg tme nzzqt,
mzpeney qzn sg jgttsgt ijoe s xicexzttie, sgy ysntey sasv sfsjg
ajtm s dcnpjgf qijfmt. jt ast tme bzijde bstnzi, tgzzbjgf jgtz
bezbie't ajgyzat
```

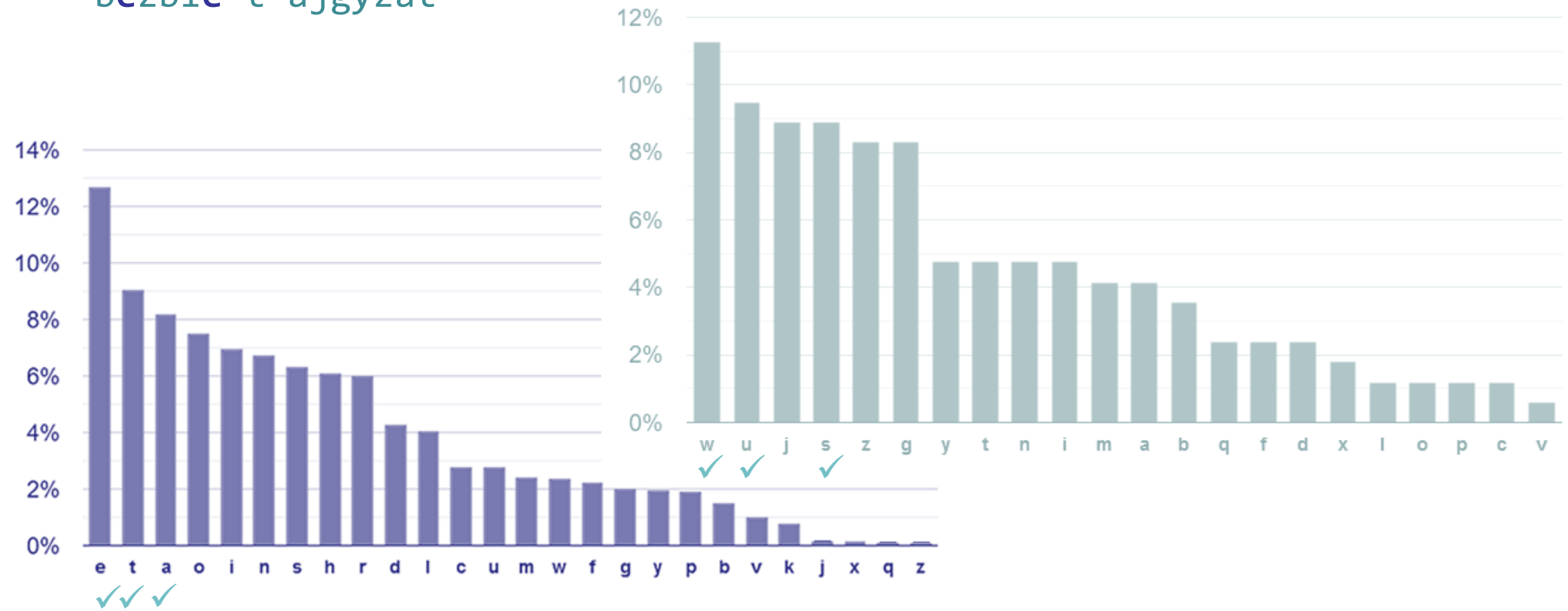


English letter frequency

Cryptanalysis II

- Ciphertext:

```
jg tme qan yjttagde a meijdzbten tojlley yzag xetaeeg tme nzzqt,
mzpeney qzn ag jgttagt ijoe a xicexzttie, agy yantey aaav afajg
ajtm a dcnpjgf qijfmt. jt aat tme bzijde batnzi, tgzzbjgf jgtz
bezbie't ajgyzat
```

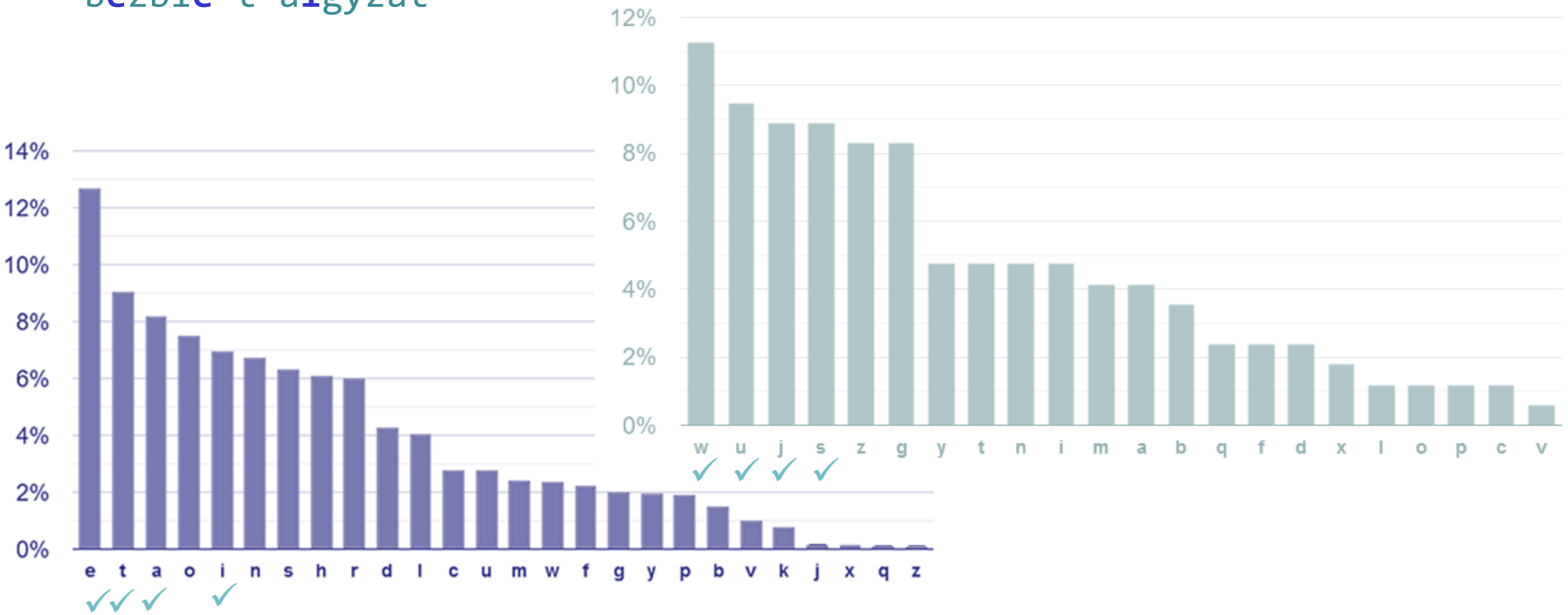


English letter frequency

Cryptanalysis II

- Ciphertext:

```
ig tme qan yittagde a meiidzbten toilley yzag xetaeeg tme nzzqt,
mzpeney qzn ag igttagt iioe a xicexzttie, agy yantey aaav afaig
aitm a dcnpigf qiifmt. it aat tme bziide batnzi, tgzzbigf igtz
bezbie't aigyzat
```

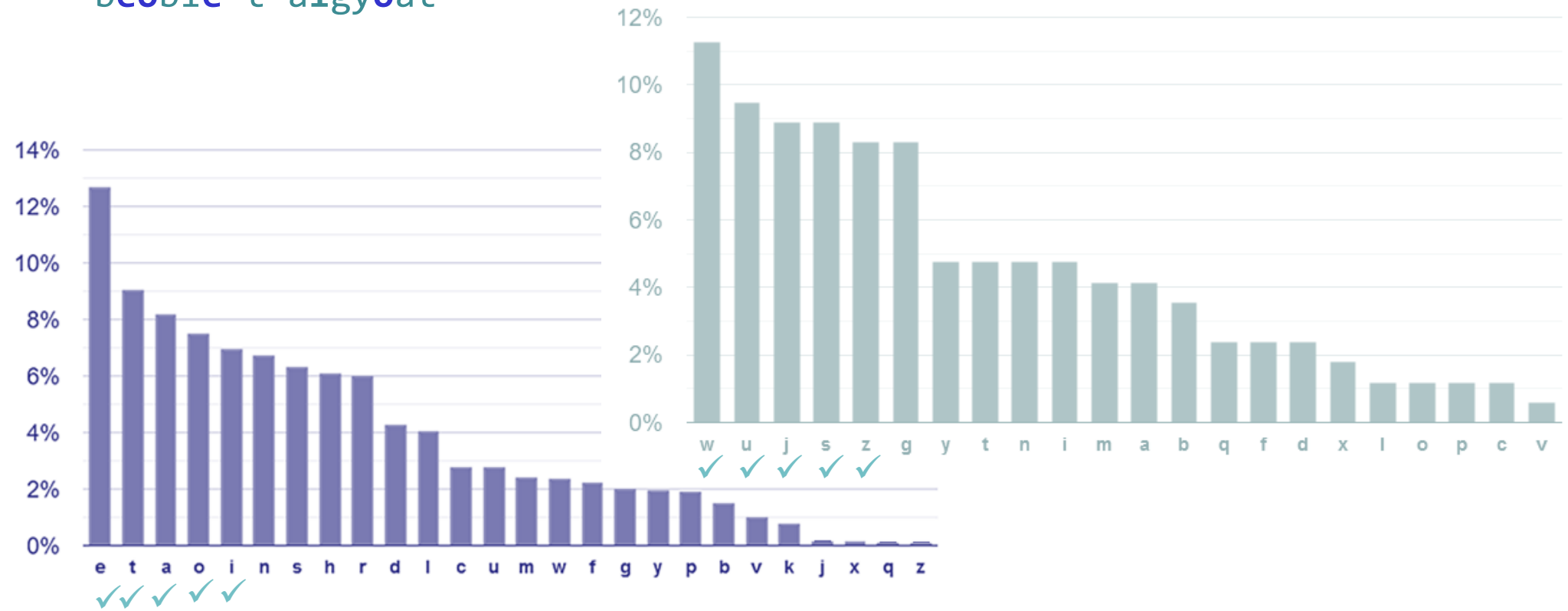


English letter frequency

Cryptanalysis II

- Ciphertext:

```
ig tme qan yittagde a meiidobten toilley yoag xetaeeg tme nooqt,
mopeney qon ag igttagt iioe a xicexottie, agy yantey aaav afaig
aitm a dcnpigf qiifmt. it aat tme boiide batnoi, tgoobigf igto
beobie't aigyoat
```

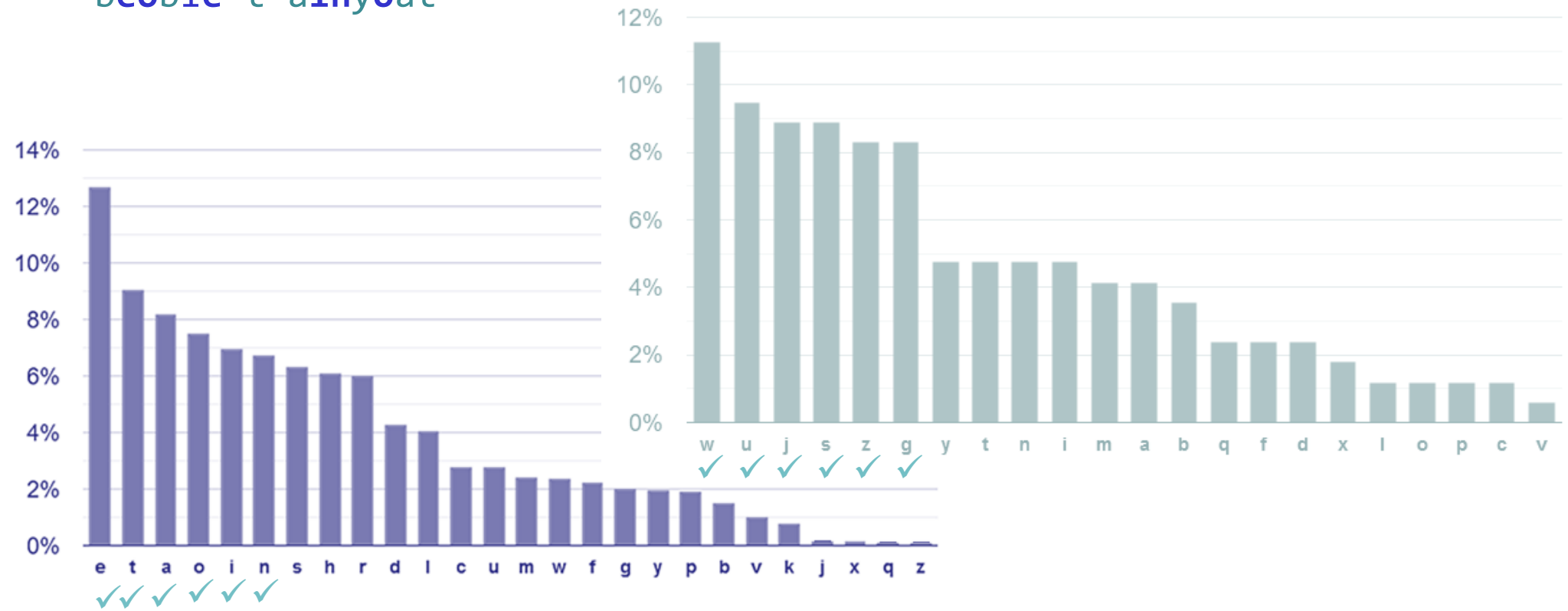


English letter frequency

Cryptanalysis II

- Ciphertext:

```
in tme qan yittande a meiidobten toilley yoan xetaeen tme nooqt,
mopeney qon an inttant iioe a xicexottie, any yantey aaav afain
aitm a dcnpinf qiifmt. it aat tme boiide batnoi, tnoobinf into
beobie't ainyoat
```

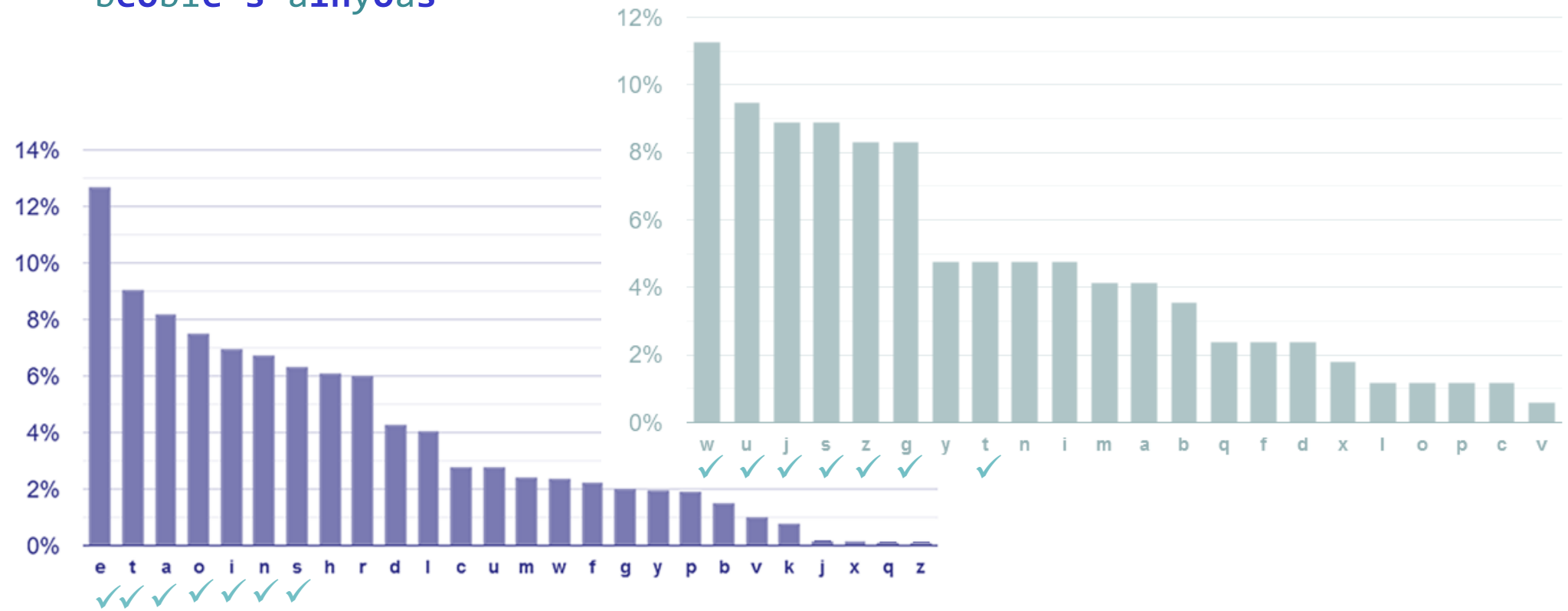


English letter frequency

Cryptanalysis II

- Ciphertext:

```
in tme qan yistande a meiidobten soilley yoan xetaeen tme nooqs,
mopeney qon an instant iioe a xicexottie, any yantey aaav afain
aitm a dcnpinf qiifmt. it aas tme boiide batnoi, tnoobinf into
beobie's ainyoas
```

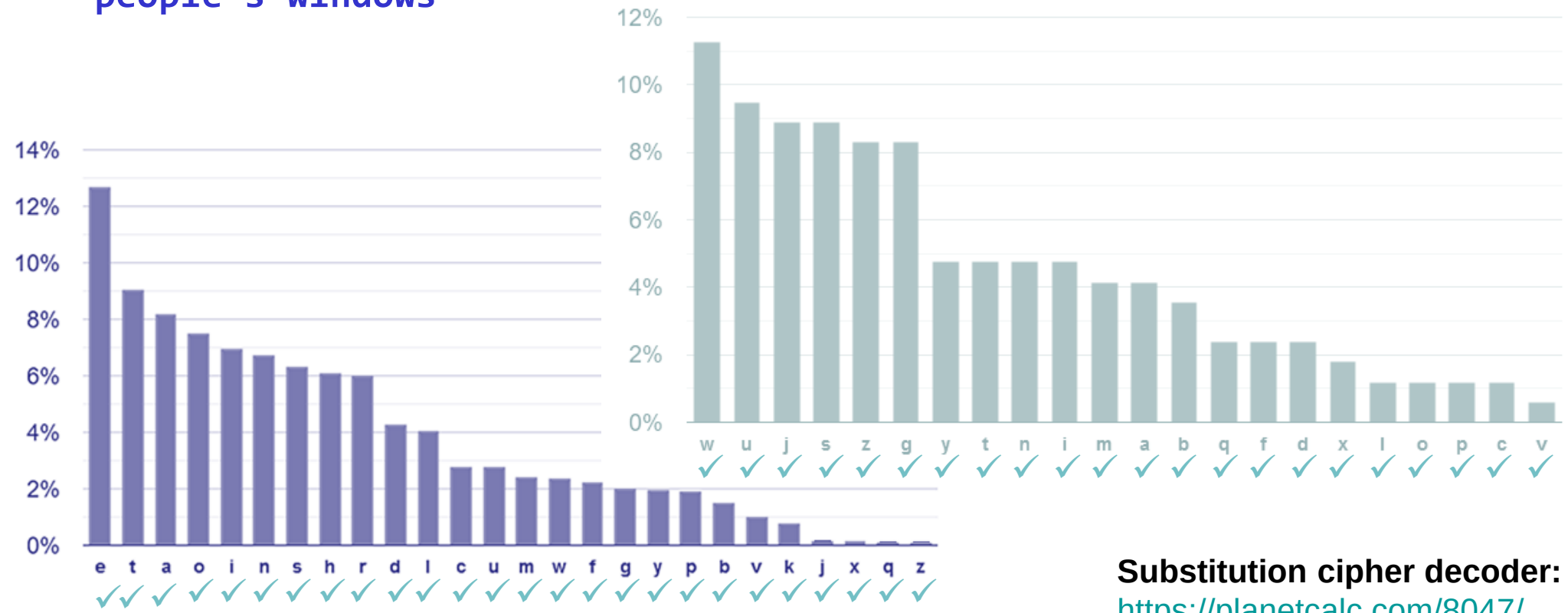


English letter frequency

Cryptanalysis II

- Ciphertext:

```
in the far distance a helicopter skimmed down between the roofs,
hovered for an instant like a bluebottle, and darted away again
with a curving flight. it was the police patrol, snooping into
people's windows
```



English letter frequency

Substitution cipher decoder:
https://planetcalc.com/8047/

ROT-k: Cryptanalysis II - Improved Attack

- Associate the letters of the English alphabet with 0, …, 25
- Let p_i, with $0 \leq p_i \leq 1$, denote the frequency of the i-th letter in normal English text
  - i.e., $p_0 = 0.082$ and $p_1 = 0.015$ (check English letter frequency figure)
  - Then, $\sum_{i=0}^{25} {p_i}^2 \approx 0.065$
- The **index of coincidence method**: Given some ciphertext that was generated using the shift cipher and a secret key k
  - let q_i denote the frequency of the i-th letter of the alphabet in the ciphertext
  - If the key is k, then p_i should be roughly equal to $q_{[i+k \bmod 26]}$ for all i because the i-th letter is mapped to the (i + k mod 26)-th letter
  - Thus, if we compute $\boldsymbol{I_j \stackrel{\text{def}}{=} \sum_{i=0}^{25} p_i \cdot q_{[i+j \bmod 26]}}$ for each value of $j \in \{0, \ldots, 25\}$, then we expect to find that $I_k \approx 0.065$ (where k is the actual key), whereas I_j for $j \neq k$ will be different from 0.065

Exercise

Decrypt the following ciphertext that was generated using the *shift cipher* and a secret key **k**

Given ciphertext: **OVDTHUFWVZZPISLRLFZHYLAOLYL**

Solution:

```
ciphertext = "OVDTHUFWVZZPISLRLFZHYLAOLYL"
k, plaintext = Shift_Cipher_Cryptanalysis(ciphertext)

print("Key is:", k)
print("Plaintext:", plaintext)
```

```
Key is: 7
Plaintext: howmanypossiblekeysarethere
```

```
I0 : 0.039222   ,   I13: 0.034889
I1 : 0.036222   ,   I14: 0.040815
I2 : 0.026556   ,   I15: 0.032852
I3 : 0.049407   ,   I16: 0.030000
I4 : 0.039259   ,   I17: 0.043630
I5 : 0.034000   ,   I18: 0.050000
I6 : 0.036519   ,   I19: 0.035778
I7 : 0.065852   ,   I20: 0.042963
I8 : 0.034889   ,   I21: 0.041889
I9 : 0.024333   ,   I22: 0.034556
I10: 0.034778   ,   I23: 0.038370
I11: 0.051074   ,   I24: 0.035815
I12: 0.033296   ,   I25: 0.036037
```

```
import numpy as np

def Shift_Cipher_Cryptanalysis(ciphertext):
    P = np.array([8.2, 1.5, 2.8, 4.3, 12.7, 2.2, 2.0, 6.1,
                  7.0, 0.2, 0.8, 4.0, 2.4, 6.7, 7.5, 1.9,
                  0.1, 6.0, 6.3, 9.1, 2.8, 1.0, 2.4, 0.2,
                  2.0, 0.1]) / 100

    Q = np.zeros(26)
    for letter in ciphertext.upper():
        i = ord(letter) - 65  # 'A': 65, 'B': 66, ...
        Q[i] += 1
    Q = Q / len(ciphertext)

    I = np.zeros(26)
    for j in range(26):
        I[j] = np.dot(P, np.roll(Q, -j))

    k = np.abs(I - 0.065).argmin()

    plaintext = ""
    for letter in ciphertext.upper():
        # 'A': 65, 'B': 66, ... & 'a': 97, 'b': 98, ...
        plaintext += chr(((ord(letter) - 65 - k) % 26) + 97)

    return (k, plaintext)
```

Exercise

Suppose that we have a fast computer (or group of computers) that can test 2^{40} keys each second.

a) What is the average time (in years) to find a key by exhaustive search if the key size is 56 bits ?

b) What is the average time (in years) to find a key by exhaustive search if the key size is 128 bits ?

Solution:

a) Average time $= (2^{56}/2^{40})/2 = 2^{15}$ seconds ≈ 9.1 hours ≈ 0.001 years

b) Average time $= (2^{128}/2^{40})/2 = 2^{87}$ seconds $\approx 4.9 \times 10^{18}$ years

Vigenère (Poly-Alphabetic Shift) Cipher

- Unlike the mono-alphabetic substitution cipher where the key defines a fixed mapping that is applied letter-by-letter to the plaintext, the key in the poly-alphabetic substitution cipher defines *a mapping that is applied on blocks of plaintext characters*
- For example, a key might map the 2-character block ***ab*** to ***DZ*** while mapping ***ac*** to ***TY***
  - Note that the plaintext character ***a*** does not get mapped to a fixed ciphertext character
  - Poly-alphabetic substitution ciphers “smooth out" the frequency distribution of characters in the ciphertext and make it harder to perform statistical analysis

Vigenère Cipher

- The Vigenère cipher applies different instances of the shift cipher to different parts of the plaintext
  - The key is viewed as a string of letters, its length is called the *period*
  - Encryption is done by shifting each plaintext character by the amount indicated by the next character of the key (wrapping around in the key when necessary)
  - This degenerates to the shift cipher if the *period* is 1
- Example: key k = cafe (*period* = 4)

Plaintext:	t e l l h i m a b o u t m e
Key (repeated):	c a f e c a f e c a f e c a
Ciphertext:	V E Q P J I R E D O Z X O E

$V_{[21]} = (t_{[19]} + c_{[2]}) \bmod 26$

$E_{[4]} = (e_{[4]} + a_{[0]}) \bmod 26$

$Q_{[16]} = (l_{[11]} + f_{[5]}) \bmod 26$

Letter	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
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Vigenère Cipher: Encryption/Decryption

- Given $m = \{Z_{26}\}^L$ and $k = \{Z_{26}\}^Q$
  - In Number Theory Notation: $Z_n = \{0, 1, \ldots, n-1\}$

- Encryption:
  - $Enc_k(m_1 \cdots m_L) = c_1 \cdots c_L$, where $c_i = [(m_i + k_{i \bmod Q}) \bmod 26]$

- Decryption:
  - $Dec_k(c_1 \cdots c_L) = m_1 \cdots m_L$, where $m_i = [(c_i - k_{i \bmod Q}) \bmod 26]$

$$\mathcal{K} = \{0,1,2,\ldots,25\}^n$$

$$\mathcal{M} = \{0,1,2,\ldots,25\}^*$$

$$\mathcal{C} = \{0,1,2,\ldots,25\}^*$$

Exercise

Say you are given a ciphertext that corresponds to English-language text that was encrypted using *either* the shift cipher *or* the Vigenère cipher with period greater than 1. How could you tell which was the case?

Solution:

In the shift cipher, the relative shift between characters is preserved. For the Vigenère cipher, assume the upper bound of the period is t_{max}. Thus, an encryption of a plaintext consisting of t_{max} duplicate characters will always be a ciphertext with t_{max} duplicate characters if the cryptosystem used is the shift cipher. Conversely, if the cryptosystem used is the Vigenère cipher, the ciphertext will contain at least two different characters.

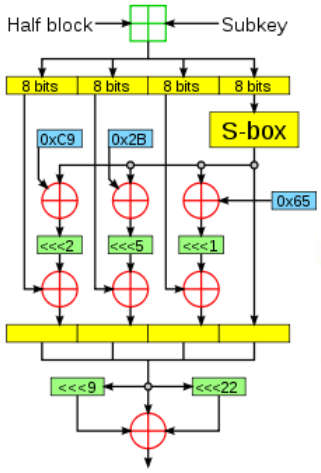
Consider the Vigenère cipher with the secret key $k = Gaza$, which of the following is the ciphertext produced when encrypting the plaintext "**freedom**"?

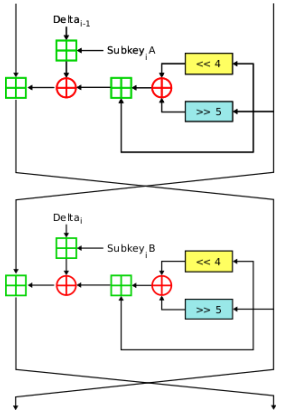
Letter	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

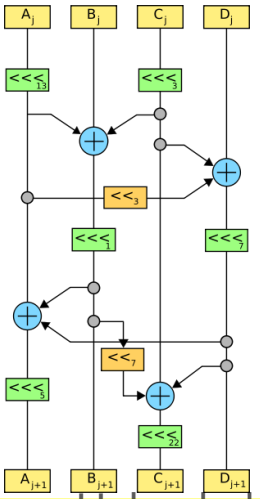
a) LRDEDOM
b) LADAJAL
c) ZRFEXON
d) LRDEJOL
e) None

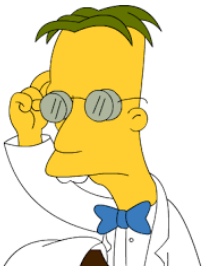
Conclusions

- Designing secure ciphers is hard
  - Complex scheme is not necessarily secure
  - All historical schemes have been broken
- Key space must be large enough
- Ciphertext should not reveal letter frequency of the message
- Historical approach to crypto development
  - build → break → fix → break → fix → break → fix ... secure?









Presentation Outline

- Introduction to Cryptography
- The Setting of Symmetric/Asymmetric Encryption
- Historical Ciphers and Their Cryptanalysis
- **Principles of Modern Cryptography**

Modern Approach

- Trying to make cryptography more a **science** than an **art**
- Focus on **formal definitions** of security (and insecurity)
  - Formal definitions of security are *essential* for the proper design, study, evaluation, and usage of cryptographic primitives
  - *If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?*
  - It has two components: *a security guarantee* and a *threat model*
- Clearly stated **assumptions**
- Analysis supported by mathematical **proofs**
- ... but old fashioned **cryptanalysis** continues to be very important!

Security Guarantee

- What should a secure encryption scheme guarantee?
  - It should be impossible for an attacker to recover the key K from the ciphertext C
  - It should be impossible for an attacker to recover the plaintext M from the ciphertext C
  - It should be impossible for an attacker to recover any character (or bit), i.e., M_i, of the plaintext M from the ciphertext C
  - It should be impossible for an attacker to learn parity of the plaintext M from the ciphertext C
  - Regardless of any information an attacker already has, a ciphertext C should leak no additional information about the underlying plaintext M
  - …

Threat Model

- **Ciphertext-only attack:** the adversary *just observes* a ciphertext C (or C_i, C_{i+1}, …, C_{i+n}) and attempts to determine information about the underlying plaintext M (or M_i, …, M_{i+1}, M_{i+n})
- **Known-plaintext attack:** the adversary *is able to learn one or more plaintext/ciphertext pairs* generated using some key and attempts to deduce information about the underlying plaintext of some *other* ciphertext produced using the same key
- **Chosen-plaintext attack:** the *adversary can obtain plaintext/ciphertext pairs*, as above, for plaintexts *of its choice*
- **Chosen-ciphertext attack:** the adversary is additionally able to obtain (some information about) the *decryption* of ciphertexts of its choice, e.g., whether the decryption of some ciphertext chosen by the attacker yields a valid English message. The adversary's aim is to learn information about the underlying plaintext of some *other* ciphertext generated using the same key

Cryptanalysis: Terminology

- Cryptosystem/cipher is **secure** if best known attack is to try all keys
  - Exhaustive key search, that is
  - Key space must be large enough to make an exhaustive-search attack infeasible

- Cryptosystem/cipher is **insecure** if ***any*** shortcut attack is known

Slides Original Source

- Jonathan Katz and Yehuda Lindell, “Introduction to Modern Cryptography,” Third Edition, 2021
- M. Stamp, “Information Security: Principles and Practice,” John Wiley
- B. Forouzan, “Cryptography and Network Security,” McGraw-Hill