Introduction to Information Security

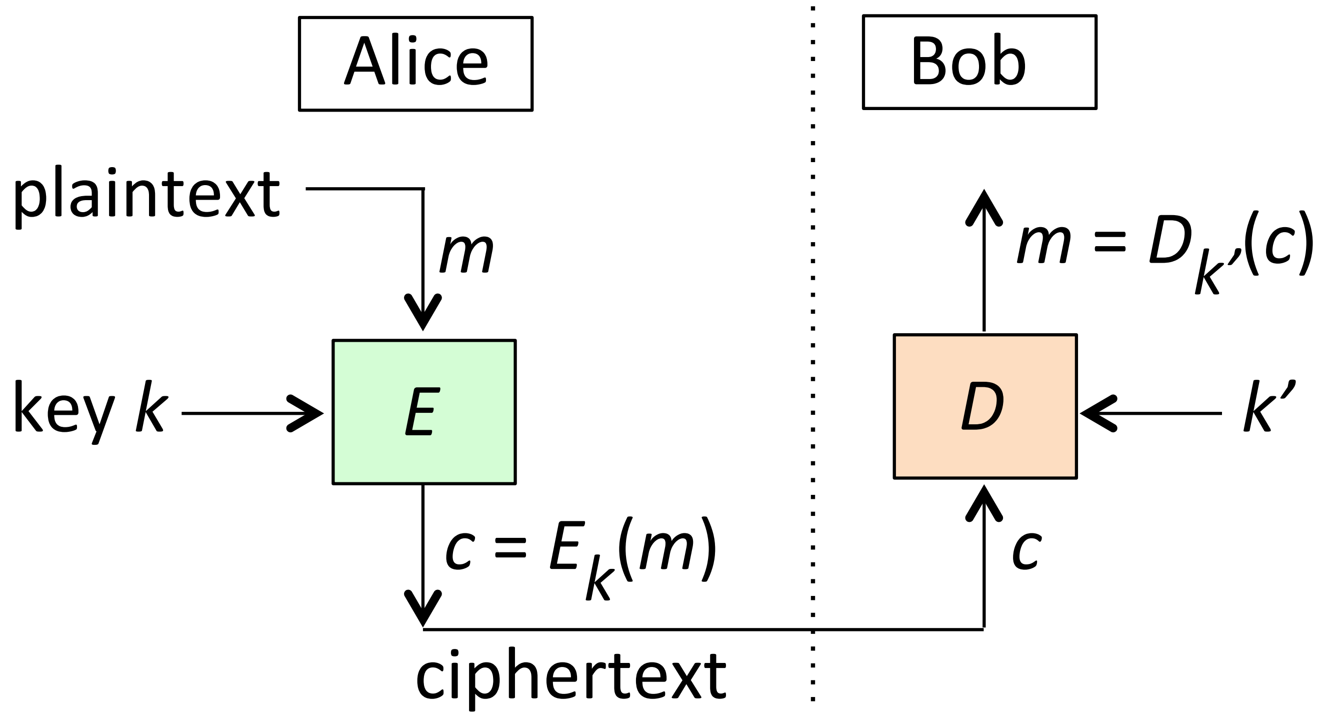
3. Classical Cryptography

Kihong Heo

KAIST

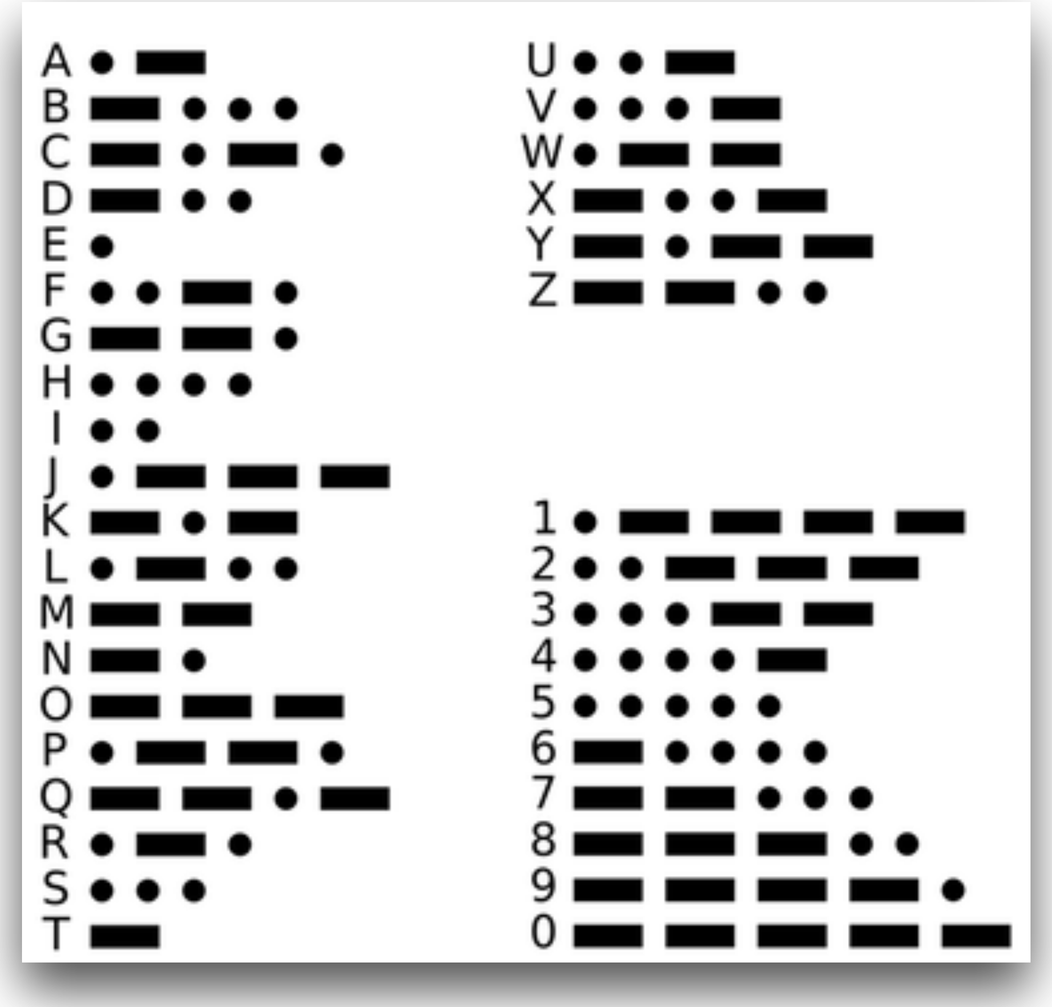
Cryptography

- “Secret writing” in Greek
- Goal: protect your (sensitive) messages/data from eavesdropping
- The most basic building block of computer security
- Two functions: encryption (E_k) and decryption (D_k) parameterized by a cryptographic key
    - Key: a large secret number



Classical vs Modern

- Cryptography: “The **art** of writing or solving **codes**” (Oxford English Dictionary)
- Codes
  - For secret communications: confidentiality
  - Modern cryptography includes more: integrity, secret key exchange, etc
- Art
  - Little theory but ad-hoc designs
  - Modern cryptography: science and math (i.e., democratization!)



Classical Cryptography

- CAUTION: DO NOT use this classical cryptography for any practical uses
- Why do we study classical ones?
  - To highlight the weakness of ad-hoc approaches
  - To demonstrate that simple approaches are unlikely to succeed
- In this lecture,
  - Caesar's cipher
  - Substitution cipher
  - Vigenere cipher

Caesar Cipher

- Encryption: shift each plaintext character 3 places forward
- Example:
    - Plaintext: `helloworld`
    - Ciphertext: `KHOORZRUOG`
- How about *k* places?
- Problem?
    - What is the key?
    - How many other keys could be chosen?

Problem: Exhaustive Key Search

- Key: a number between 0 and 25
- Given a cipher text: `OVDTHUFWVZZPISLRLFZHYLAOLYL`
- Can you find the plaintext? How?

Key Value	Possible Plain Text
1	nucsgtevuyyohrkqkeygxkznkxk
2	mtbrfsdutxxngqjpjdxfwjymjwj
3	lsaqerctswwmfpioicwevixlivi
…	…
7	howmanypossiblekeysarethere
…	…

How to make it more robust?

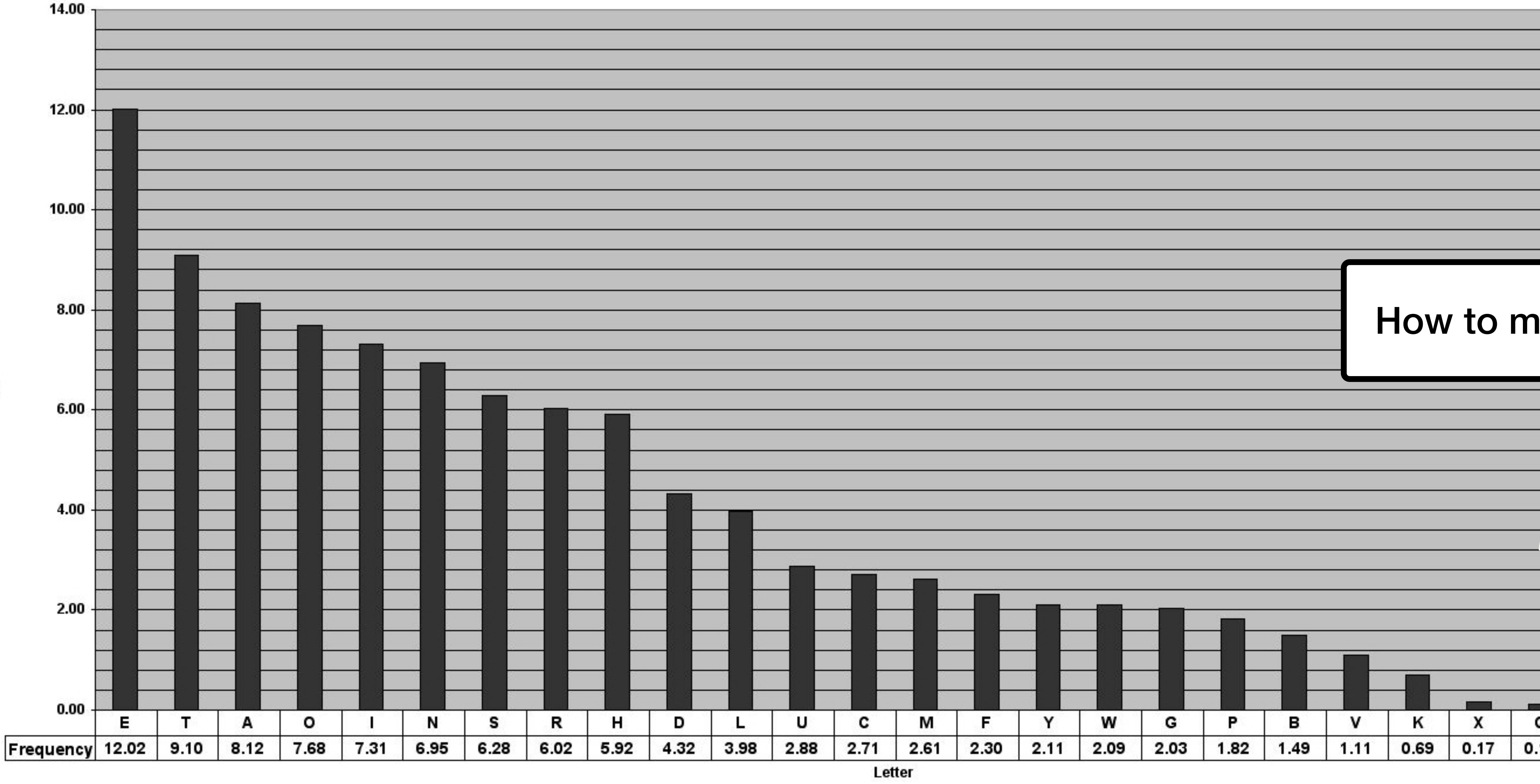
Substitution Cipher

- One-to-one mapping (bijection or permutation)
- 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	Q	W	E	R	T	Y	U	I	O	P	A	S	D	F	G	H	J	K	L	Z	X	C	V	B	N	M

- Key space?
  - $26! \sim 2^{88} \sim 4 \text{ x } 10^{26}$
- Robust enough?

Problem: Letter Frequency Analysis



Frequency	E	T	A	O	I	N	S	R	H	D	L	U	C	M	F	Y	W	G	P	B	V	K	X	Q	[illegible]
Frequency	12.02	9.10	8.12	7.68	7.31	6.95	6.28	6.02	5.92	4.32	3.98	2.88	2.71	2.61	2.30	2.11	2.09	2.03	1.82	1.49	1.11	0.69	0.17	0.11	0.10

How to make it more robust?

Vigenere Cipher

- Encryption: poly-alphabetic shift
- Example:
  - Plaintext: `tellhimaboutme`
  - Key (repeated): `cafecafecafeca`
  - Ciphertext: `VEQPJIREDOZXOE`
- Letters are mapped to different ciphertexts: smooth out the frequency distribution in ciphertext
- Invented in 16th century and had been unbreakable for hundreds of years
- Problem?

Cracking Vigenere Cipher

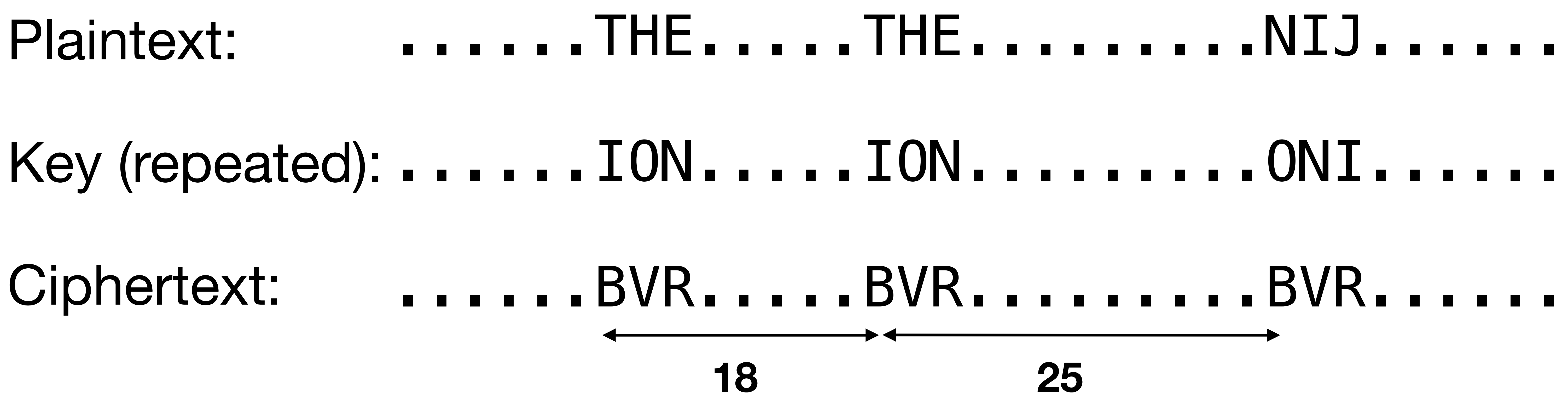
- When the length (*t*) of the key is known:
  - Divide ciphertext into *t* parts and perform statistical analysis for each part

Plaintext:	`tellhimaboutme`
Key (repeated):	`cafecafecafeca`
Ciphertext:	`VEQPJIREDOZXOE`

- When the length of the key is unknown but the max length *T* is known:
  - Repeat the above *T* times
- What if the length is unknown?

Kasiski’s Method

- What if there is a repeated substring in plaintext?
  - A repeated substring **may** exist in the ciphertext
  - The distance of the two occurrences **may** be a multiple of the key length
- Example



```
Plaintext:       ......THE.....THE.........NIJ......
Key (repeated):  ......ION.....ION.........ONI......
Ciphertext:      ......BVR.....BVR.........BVR......
```

Example

```
LFWKI MJCLP SISWK HJOGL KMVGU RAGKM KMXMA MJCVX WUYLG GIISW
ALXAE YCXMF KMKBQ BDCLA EFLFW KIMJC GUZUG SKECZ GBWYM OACFV
MQKYF WXTWM LAIDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEFL FWKIM
JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LKWML AVGKY EDEMJ XHUXD
AVYXL
```

Example

LFWKI MJCLP SISWK HJOGL KMVGU RAGKM KMXMA MJCVX WUYLG GIISW
ALXAE YCXMF KMKBQ BDCLA EF**LFW KIMJC** GUZUG SKECZ GBWYM OACFV
MQKYF WXTWM LAIDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEF**L FWKIM**
JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LKWML AVGKY EDEMJ XHUXD
AVYXL

Example

LFWKI MJCLP SISWK HJOGL KMVGU RAGKM KMXMA MJCVX WUYLG GIISW
ALXAE YCXMF KMKBQ BDCLA EF**LFW KIMJC** GUZUG SKECZ GBWYM OACFV
MQKYF WXT**WM LA**IDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEF**L FWKIM**
JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LK**WML A**VGKY EDEMJ XHUXD
AVYXL

Example

LFWKI **MJC**LP S**ISW**K HJOGL KMVGU RAGKM KMXMA MJCVX WUYLG GI**ISW**
ALXAE YCXMF KMKBQ BDCLA EF**LFW** **KIMJC** GUZUG SKECZ GBWYM OACFV
MQKYF WXT**WM** **LA**IDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEF**L** **FWKIM**
JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LK**WML** **A**VGKY EDEMJ XHUXD
AVYXL

…

Analysis

Substring	Length	Distance	Factors
LFWKIMJC	8	72	2 3 4 6 8 9 12 18 24 36 72
WMLA	4	74	2 37 74
MJC	3	66	2 3 6 11 22 33 66
ISW	3	36	2 3 4 6 9 12 18 36
VMQ	3	32	2 4 8 16 32
DAV	3	30	2 3 5 6 10 15

Candidate Keyword Length

	Factors																		
	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
74	O																		
72	O	O	O		O		O	O			O						O		
66	O	O			O					O									
36	O	O	O		O			O			O						O		
32	O		O				O								O				
30	O	O		O	O				O					O					
Total	6	4	3	1	4	0	2	2	1	1	2	0	0	1	1	0	2	0	0

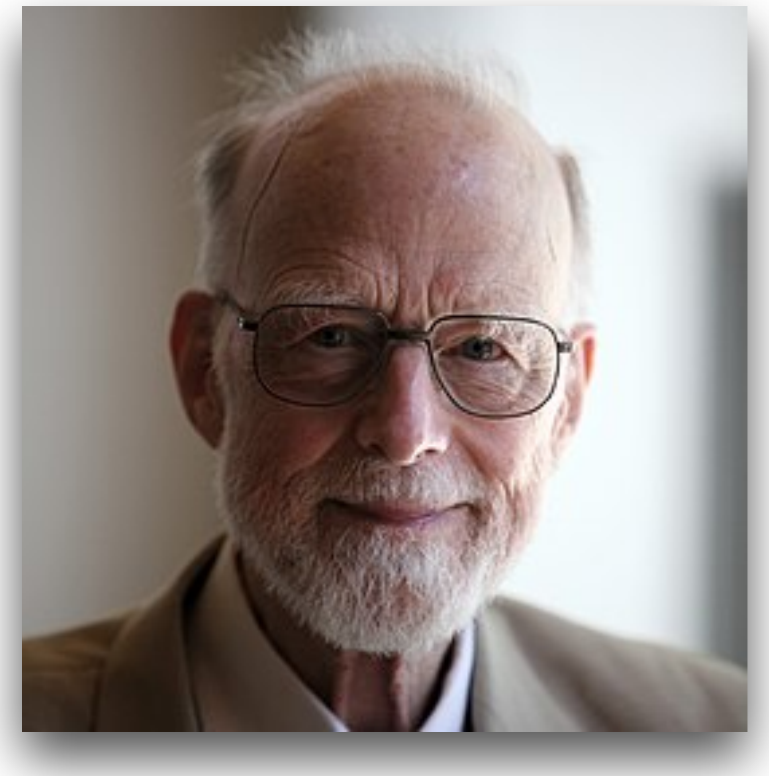
Result

LFWKI **MJC**LP S**ISW**K HJOGL KMVGU RAGKM KMXMA MJCVX WUYLG GI**ISW**
ALXAE YCXMF KMKBQ BDCLA EF**LFW** **KIMJC** GUZUG SKECZ GBWYM OACFV
MQKYF WXT**WM** **LA**IDO YQBWF GKSDI ULQGV SYHJA VEFWB LAEF**L** **FWKIM**
JCFHS NNGGN WPWDA VMQFA AXWFZ CXBVE LK**WML** **A**VGKY EDEMJ XHUXD
AVYXL

THERE **ARE**TW O**WAY**S OFCON STRUC TINGA SOFTW AREDE SIGNO NE**WAY**
ISTOM AKEIT SOSIM PLETH AT**THE** **REARE** OBVIO USLYN ODEFI CIENC
IESAN DTH**EO** **TH**ERW AYIST OMAKE ITSOC OMPLI CATED THAT**T** **HEREA**
RENOO BVIOU SDEFI CIENC IESTH EFIRS TM**ETH** **O**DISF ARMOR EDIFF
ICULT

Pop-up Lesson

> “There are two ways of constructing a software design:
> One way is to make it so simple that there are obviously
> no deficiencies, and the other way is to make it so complicated
> that there are no obvious deficiencies.
> The first method is far more difficult.”
>
> - T. Hoare, ACM Turing Award winner (1980)



Properties of Kasiski's method

Object	Property
Long ciphertext	
Short plaintext with a long key	
Long repeated substring in a ciphertext	
Short repeated substring in a ciphertext	

Principles of Modern Cryptography

- Rigorous vs Ad-hoc approaches to security: Science vs Art
- What we need for science
  - Formal (i.e., rigorous and precise) definitions of security
  - Precise assumptions
  - Proofs of security

Formal Definition

- Can you formally define what you mean by “security”?
- Security definition is a tuple
  - Security guarantee: “what the scheme is intended to prevent the attack from doing”
  - Adversary assumptions: “power (or capabilities) of the adversary”
- Example
  - Assume: attacker obtains plaintext/ciphertext pairs for plaintext of its choice
  - Guarantee: attacker cannot decrypt agiven ciphertext

Security Guarantees

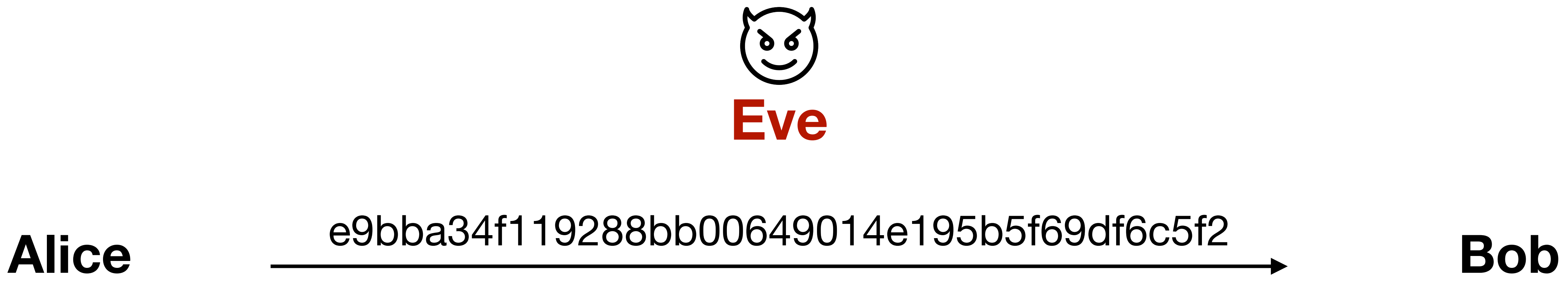
- Example: What are the desired security guarantees for secure encryption?
- Impossible for an attacker
    - To recover the key? Enough?
    - To recover the entire plaintext from the ciphertext? Enough?
    - To recover any character of the plain text from the ciphertext? Enough?
    - To derive any meaningful information about the plaintext from the ciphertext? Enough?
    - To compute any function of the plaintext from the ciphertext (i.e., semantic security)

Adversary Assumptions

- Example: what are the adversary capabilities?
- Attacker capabilities (in order of increasing attacker power)
  - Ciphertext-only attack: most basic attack
  - Known-plaintext attack: attacker obtains certain plaintext/ciphertext pairs
  - Chosen-plaintext attack: attacker obtains plaintext/ciphertext pairs for plaintext of its choice
  - Chosen-ciphertext attack: attacker obtains plaintext/ciphertext pairs for ciphertext of its choice

Ciphertext-Only Attack (COA)

- Most basic attack
- The attacker is assumed to have access only to ciphertexts
- Can the attacker compute any function of the plaintext from the ciphertext?



Eve

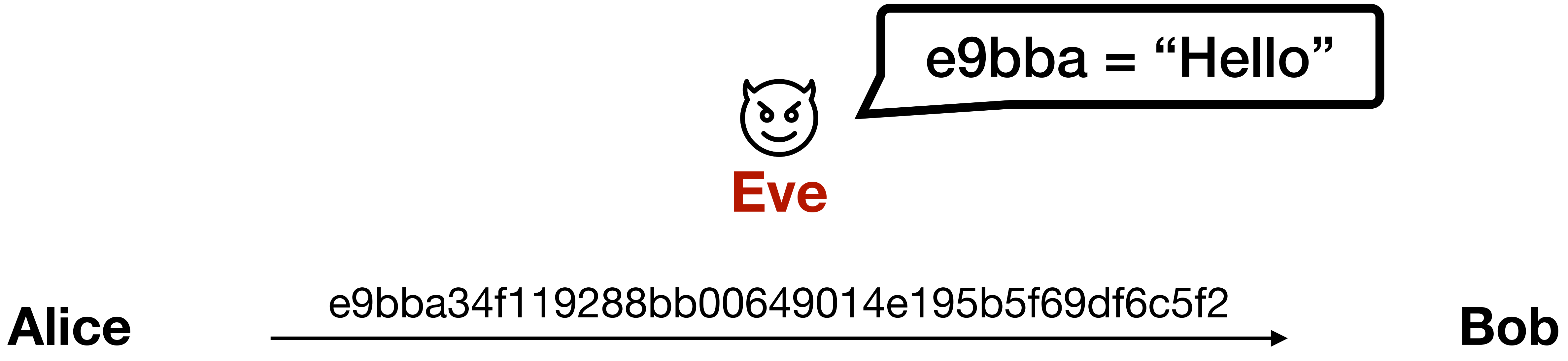
Alice

e9bba34f119288bb00649014e195b5f69df6c5f2

Bob

Known-Plaintext Attack (KPA)

- The attacker is assumed to have access to some plaintext and its corresponding ciphertext
- Can the attacker compute any function of the plaintext from the ciphertext?
- Example: “hello” message



e9bba = “Hello”

Eve

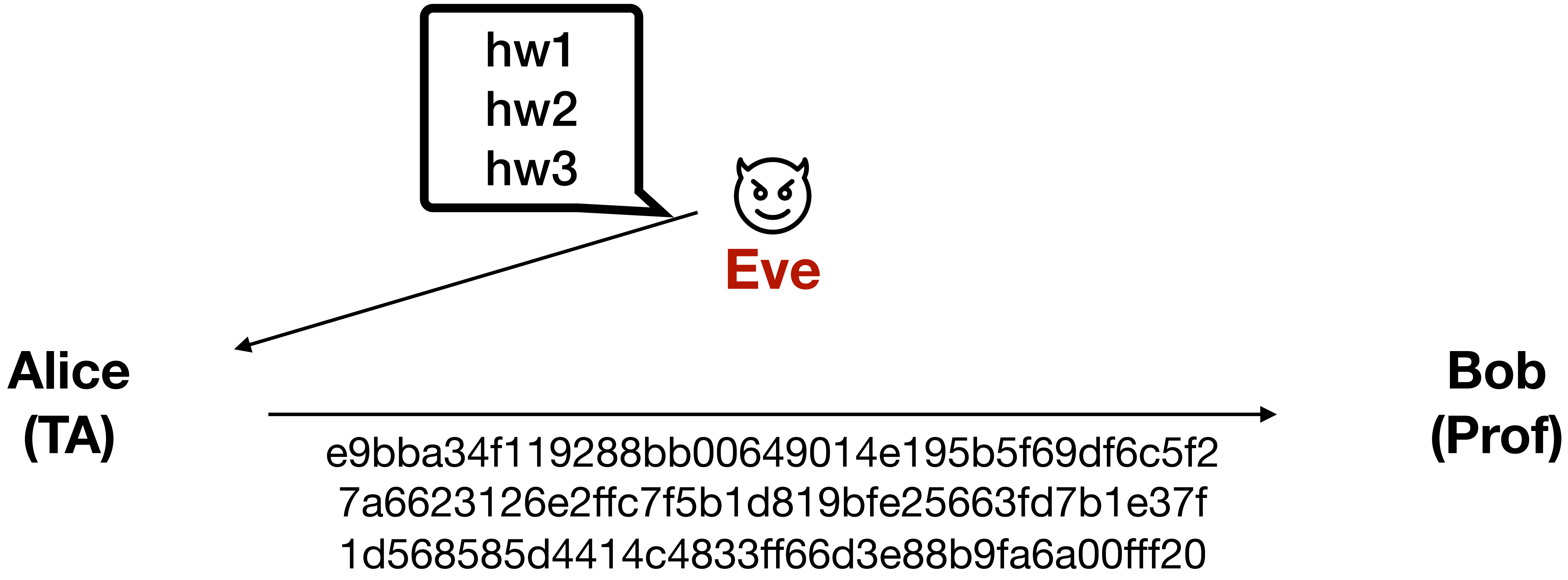
Alice

e9bba34f119288bb00649014e195b5f69df6c5f2

Bob

Chosen-Plaintext Attack (CPA)

- The attacker is assumed to have access to the ciphertexts for arbitrary plaintexts
- Can the attacker compute any function of the plaintext from the ciphertext?
- Example:



hw1
hw2
hw3

Eve

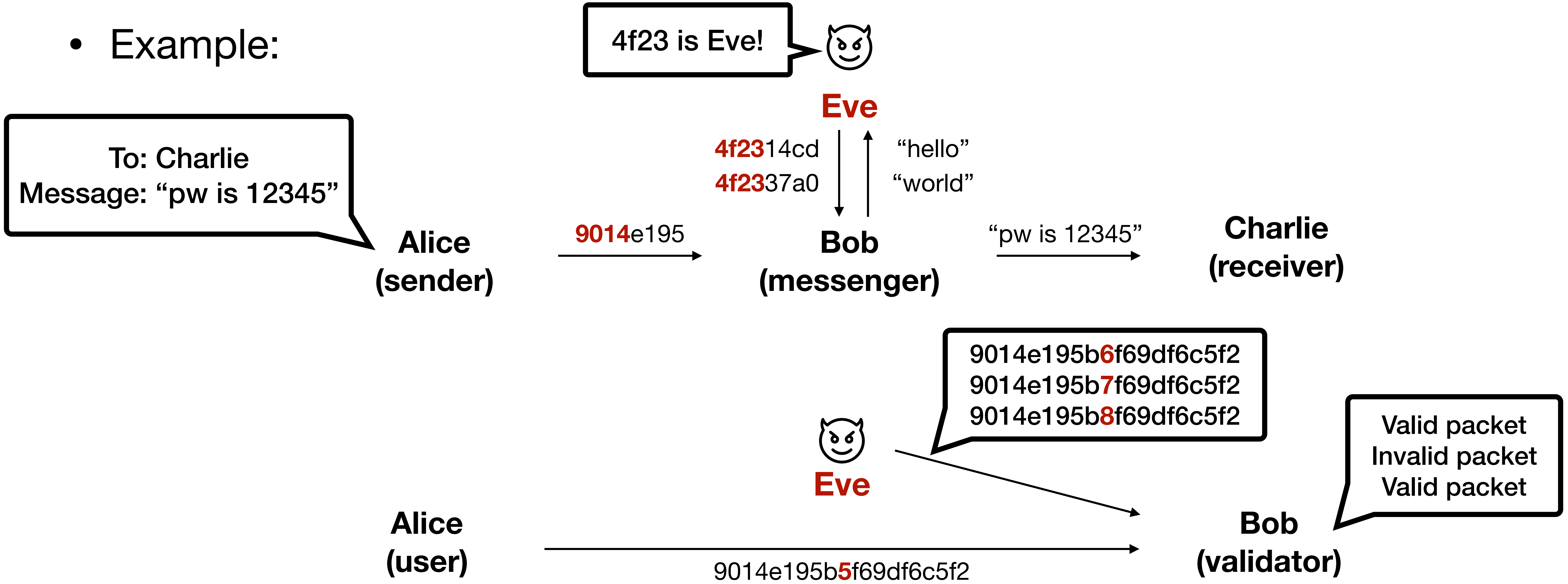
Alice
(TA)

Bob
(Prof)

e9bba34f119288bb00649014e195b5f69df6c5f2
7a6623126e2ffc7f5b1d819bfe25663fd7b1e37f
1d568585d4414c4833ff66d3e88b9fa6a00fff20

Chosen-Ciphertext Attack (CCA)

- The attacker is assumed to have access to the plaintexts for all ciphertexts other than the target
- Can the attacker compute any function of the plaintext from the ciphertext?
- Example:



Precise Assumptions

- Do we have any assumptions in classic cryptography?
- Most security schemes rely on some assumptions conjectured to be true
  - E.g., prime factorization of large numbers for RSA
- Why should we have clear assumptions?
  - Mathematical proofs
  - Validation
  - Comparison
  - Understanding

Summary

- Classical cryptography: ad-hoc design & informal proof
  - Caesar’s cipher, Substitution cipher, Vigenere cipher
- Modern cryptography: rigorous design & formal proof
  - Security guarantee
  - Attack model: COA, KPA, CPA, CCA