ISOM 5280: Cryptography

Prof. Weiyin Hong

Department of ISOM, HKUST Business School

Fall 2024

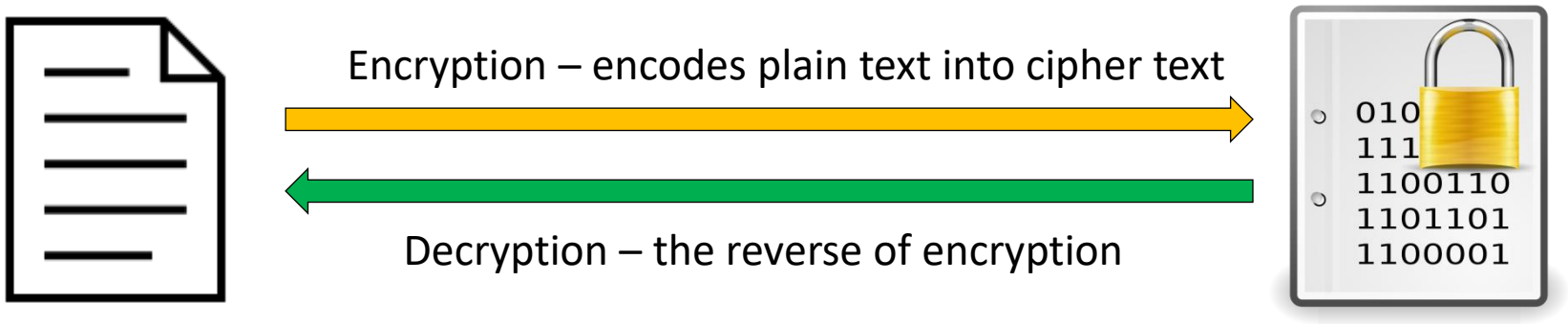
Reading

- [WM] Chapter 10

Table of Contents

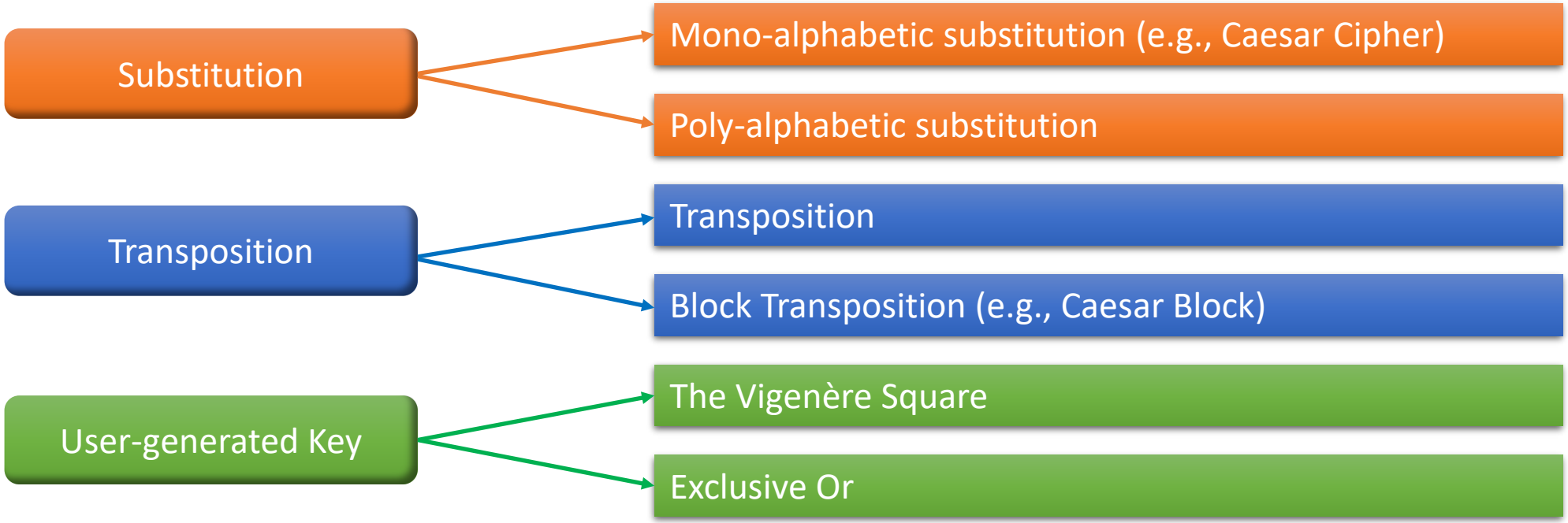
Cryptology

- Cryptology – the science of encryption
  - Cryptography – the process to keep a message secret from unintended audiences
  - Cryptanalysis – the process to obtain original text from encrypted message without knowing the methods/keys



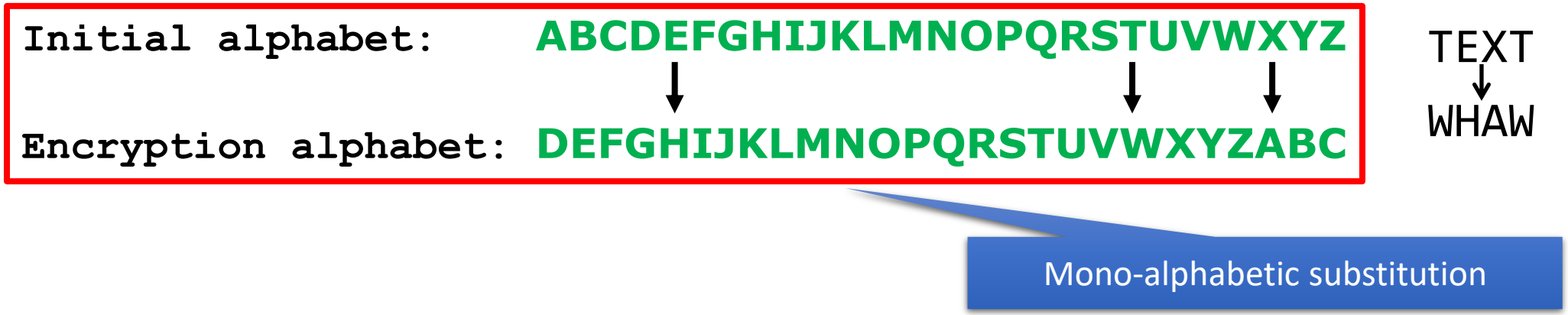
Cipher Methods

Cipher Methods Overview



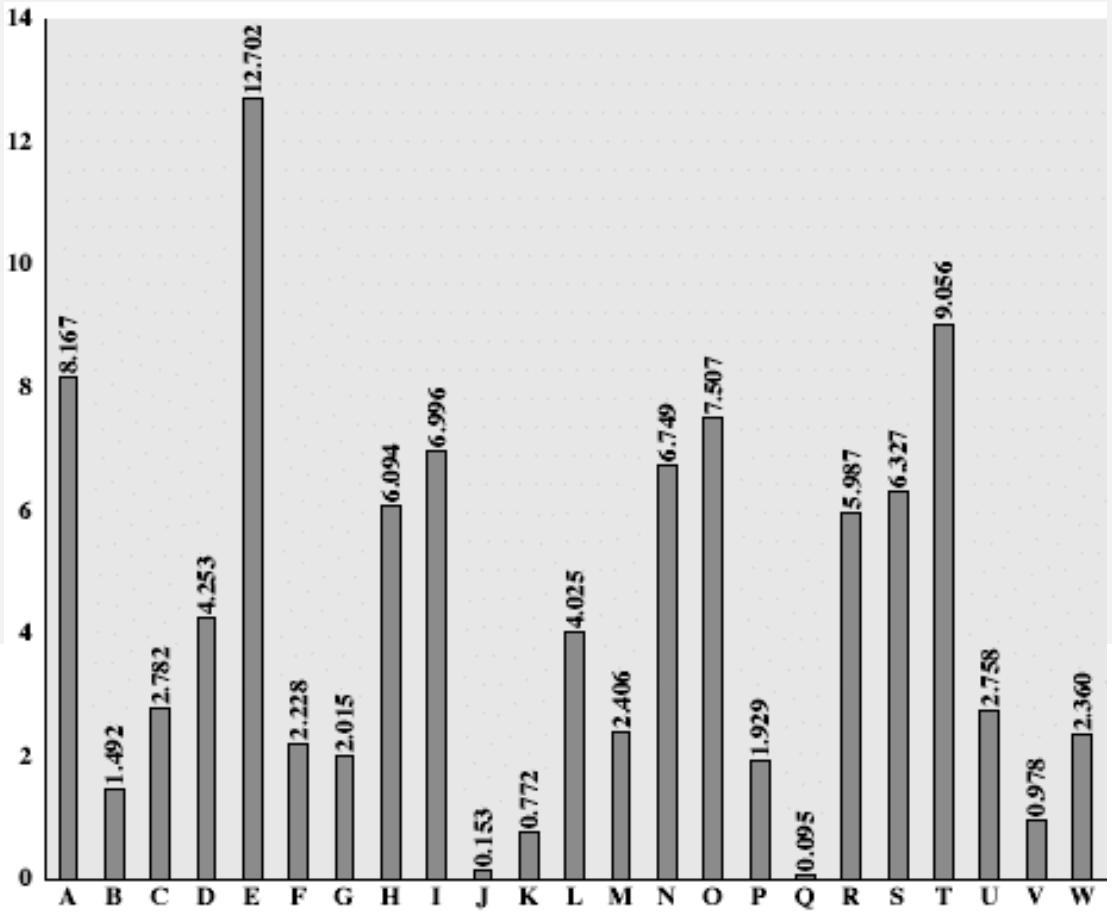
Cipher Methods (1)

- By substitution
  - e.g., Caesar cipher (shift each character position by 3 places to the right)



  - Key: positions to shift (K = 3)
  - Mono-alphabetic vs. poly-alphabetic (one vs. multiple alphabet sets)

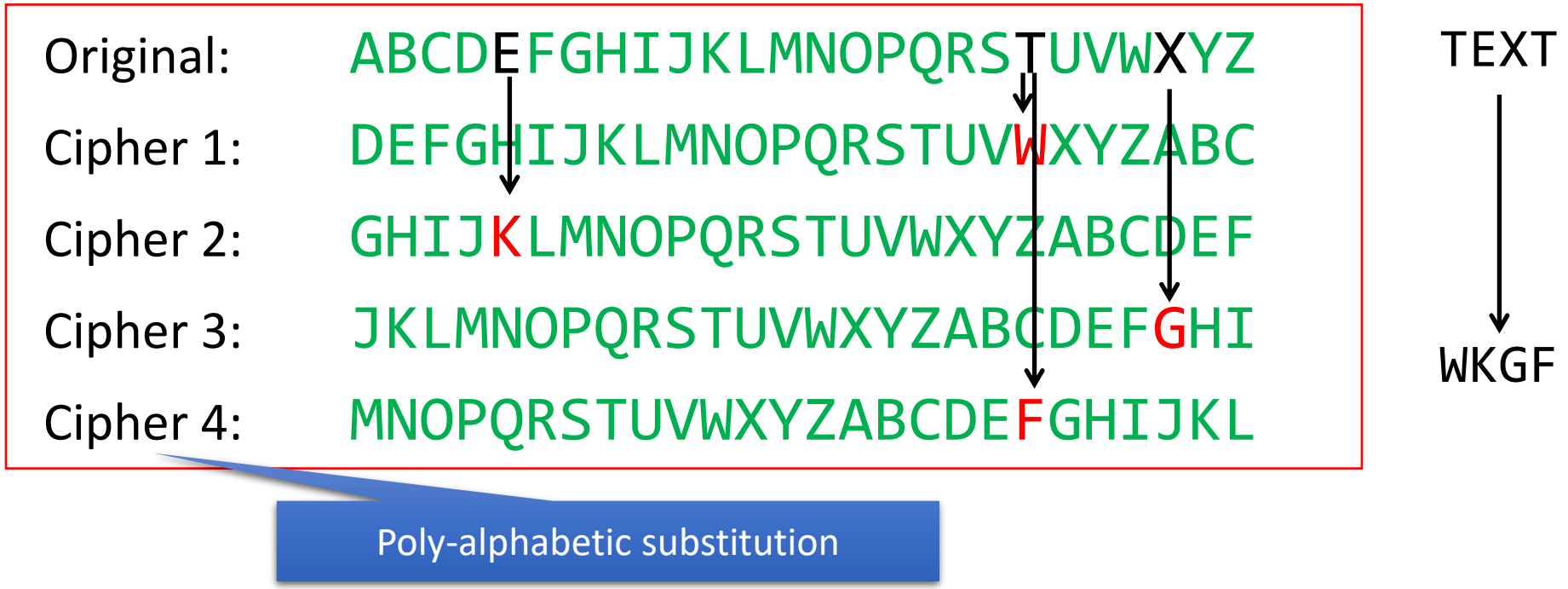
How to crack?



- Use frequency chart to improve efficiency of decryption

Cipher Methods (2)

- By substitution
  - Poly-alphabetic



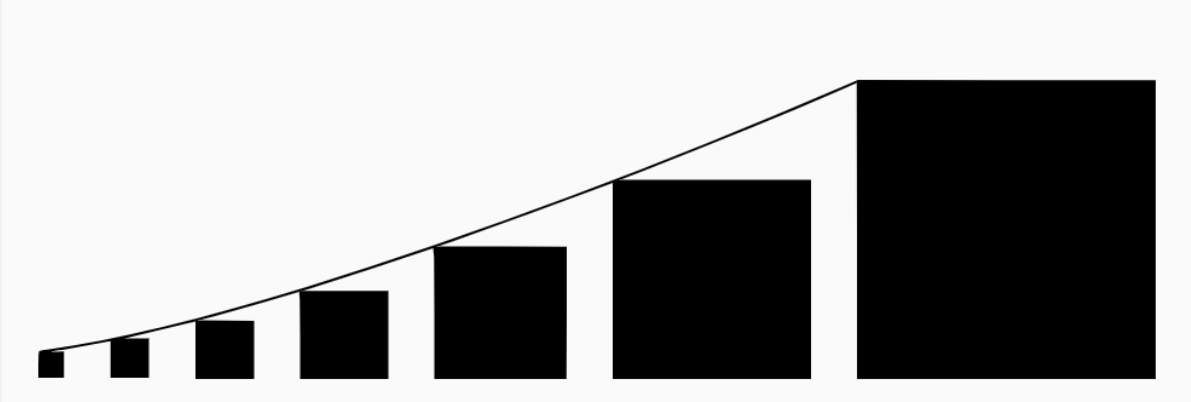
Cipher Methods (3)

- By transposition
  - Caesar block: fit the text to a number square (e.g., 5 by 5)

Plaintext:	SACK GAUL SPARE NO ONE
Caesar block:	S G S _ N
	A A P N E
	C U A O _
	K L R _ _
	_ _ E O _
Ciphered:	SGS_NAAPNECUAO_KLR____EO

How to crack?

(Are bigger squares harder to crack?)



- Block size does not directly affect the security of the cipher. However, if block size is too small, it is not effective.

ASCII Code (Binary Code)

Character	Binary Code	Character	Binary Code	Character	Binary Code	Character	Binary Code	Character	Binary Code
A	01000001	Q	01010001	g	01100111	w	01110111	-	00101101
B	01000010	R	01010010	h	01101000	x	01111000	.	00101110
C	01000011	S	01010011	i	01101001	y	01111001	/	00101111
D	01000100	T	01010100	j	01101010	z	01111010	0	00110000
E	01000101	U	01010101	k	01101011	!	00100001	1	00110001
F	01000110	V	01010110	l	01101100	"	00100010	2	00110010
G	01000111	W	01010111	m	01101101	#	00100011	3	00110011
H	01001000	X	01011000	n	01101110	$	00100100	4	00110100
I	01001001	Y	01011001	o	01101111	%	00100101	5	00110101
J	01001010	Z	01011010	p	01110000	&	00100110	6	00110110
K	01001011	a	01100001	q	01110001	'	00100111	7	00110111
L	01001100	b	01100010	r	01110010	(	00101000	8	00111000
M	01001101	c	01100011	s	01110011	)	00101001	9	00111001
N	01001110	d	01100100	t	01110100	*	00101010	?	00111111
O	01001111	e	01100101	u	01110101	+	00101011	@	01000000
P	01010000	f	01100110	v	01110110	,	00101100	_	01011111

Cipher Methods (4)

- By transposition
  - Shift the values within a block of text (with specified lengths, say, 8-bit blocks) to create the cipher text

E.g., key: 1→4, 2→8, 3→1, 4→5, 5→7, 6→2, 7→6, 8→3

```
Bit locations:  87654321 | 87654321 | 87654321
Plaintext:    % 00100101 | 01101011 | 10010101
Ciphertext:   k 00001011 | 10111010 | 01001101
```

Do you think 8-bit blocks are strong enough? - No

Cipher Methods (5)

- The Vigenère Square
  - Powerful when combined with a user-chosen encryption key

Original:	SACK GAUL SPARE NO ONE
Original:	SACKGAULSPARENOONE
Key :	ITALYITALYITALYITA
Ciphered:	ATCVEINLDNIKEYMWGE

Which letter shall not be used as part of the key?
–A

Original

Key

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

Table 8-2 The Vigenère Square

Cipher Methods (6)

- Exclusive OR (XOR)
  - Substitution by the XOR function with a user-chosen key
  - The key can be of different lengths

```
Bit stream: 011100000110010110000000
Key:        100001011000010110000101
Cipher:     111101011110000000000101
```

First bit	Second bit	Result
0	0	0
0	1	1
1	0	1
1	1	0

Review Questions

Mentimeter

Cipher Methods Review

- Review questions:
  - What's the key for each method?
    - For mono-alphabetic, the key is …
    - For poly-alphabetic, the key is …
    - For Caesar block, the key is …
    - For transposition, the key is …
    - For the other two methods, the keys are …
  - What do you need to pass on to the recipient in order for him or her to decipher?

Fun Time…

1. Find a partner in your class.
2. Use the one of the cipher method learned today to encrypt a short secret message (e.g., “iloveu”) into a ciphered text.
3. Give the other party:
   a. The ciphered text (not the original!!)
   b. The cipher method (i.e., the algorithm)
   c. The key
4. The other party shall try to decipher and find the original text.
5. After decipher, please exchange answers to see if your answer is correct!

Cipher Methods Review

Encryption Algorithm

Key Size

- Which is more important and why? - Size

Table 10-5 Encryption Key Power

It is estimated that to crack an encryption key using a brute force attack, a computer needs to perform a maximum of 2^*k* operations (2^k guesses), where *k* is the number of bits in the key. The average estimated time to crack is approximately half that time.

Key Length (Bits)	Maximum Number of Operations (Guesses)	Maximum Time to Crack	Estimated Average Time to Crack
16	65,536	0.000000112 seconds	0.000000056 seconds
24	16,777,216	0.0000287 seconds	0.0000143 seconds
32	4,294,967,296	0.00734 seconds	0.00367 seconds
56	72,057,594,037,927,900	34.2 hours	17.1 hours
64	18,446,744,073,709,600,000	364.7 days	182.35 days
128	3.40E+38	18,431,695,314,143,700,000 years	9,215,847,657,071,860,000 years
256	1.16E+77	6,271,980,907,862,400,000, 000,000,000,000,000,000,000, 000,000,000,000,000,000 years	3,135,990,453,931,200,000, 000,000,000,000,000,000, 000,000,000,000,000,000,000 years
512	1.34E+154	7.26E+134 years	3.63E+134 years

Note: Estimated Time to Crack is based on a 2020-era Intel i9-10900X 10 Core CPU performing 585 Dhrystone GFLOPS (giga/billion floating point operations per second) at 5.2 GHz (overclocked). Modern workstations are capable of using multiple CPUs, further decreasing time to crack, or simply splitting the workload among multiple systems.

Note: The authors acknowledge that this benchmark is based on a very specific application test and that the results are not generalizable. However, these calculations are shown to illustrate the relative difference between key length and resulting strength rather than to accurately depict time to crack.

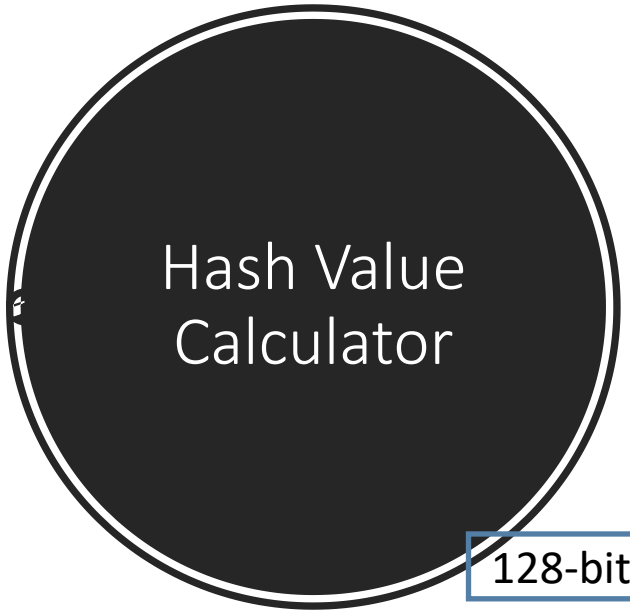
Robustness

What does encryption ensure?

Hash

- Hash functions are mathematical algorithms used to confirm the identity of a specific message and confirm that the content has not been changed.
- It does not create cipher text, instead generates a hash value or message digest.

Hash Calculator Online



128-bit

160-bit

224-bit

256-bit

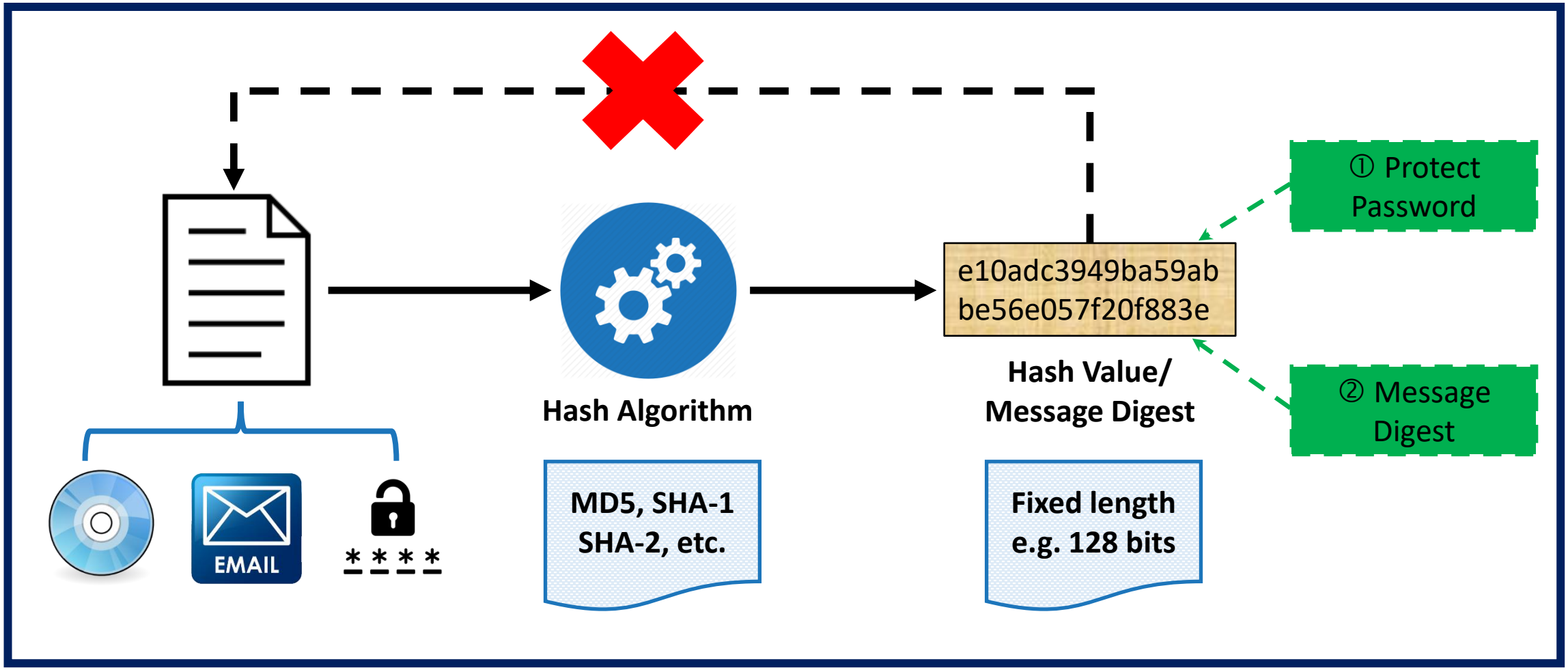
Hash string

butterfly across the ocean

Calculated hashes for 26 bytes

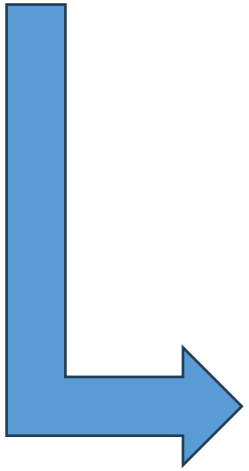
Name	Length	Hash
md2	16	4EEFADA39DBD6DD3B80C05550F5B3862
md4	16	203E1D4B59EA404CE9534201A12B0C67
md5	16	A38143BC28FDF31EE635BC1DC20791F8
sha1	20	D1D0C0736427FD3CDF75697511D830B96388D6FC
sha224	28	43FC8D520F558B55C287617D46E635BEB4FDFA066E64E0EAE3(
sha256	32	5EB4FBE94549F092D9A7E79A9506ACD584AA87F3AC92E2CE62:

Hash Function



Which one looks like a hash to you?

Plain Text



A long time ago, in a galaxy far, far away... It is a dark time for the Rebellion. Although the Death Star has been destroyed, Imperial troops have driven the Rebel forces from their hidden base and pursued them across the galaxy.

C SDJA ZLUK CAD, LJ C ACSCGH OCX, OCX CPCH... LZ LW C QCXT ZLUK ODX ZVK XKIKSSLDJ. CSZVDMAV ZVK QKCZV WZCX VCW IKKJ QKWZXDHKQ, LUEKXLCS ZXDDEW VCRK QXLRKJ ZVK XKIKS ODXNKW OXDU ZVKLX VLQQKJ ICWK CJQ EMXWMKQ ZVKU CNXDWW ZVK ACSCGH.

a446277f2bebe3a799b653485d63d2fc556d602b

Hash characteristics

1. One way and irreversible
2. Same data -> Same hash!
3. A small change in data -> A big change in hash values
4. Uniqueness -> impossible to find two messages with the same hash values (hash collision)

Collision Attack

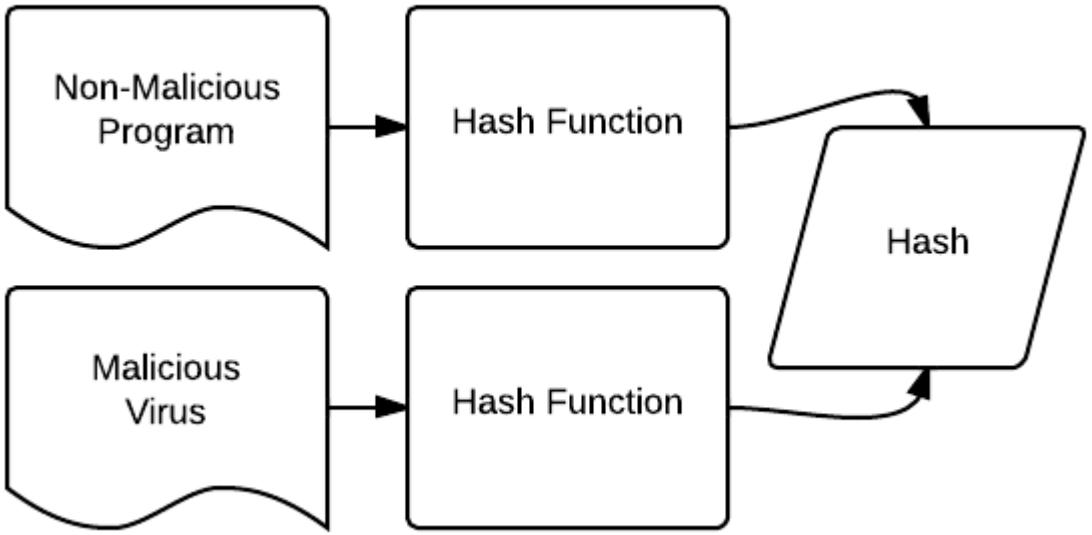


Image Source: PrivacyCanada

For example, if the attacker was offering a file download and showed the hash to prove the file's integrity, he could switch out the file download for a different file that had the same hash, and the person downloading it would be unable to know the difference. The file would appear valid as it has the same hash as the supposed real file.

SHA1 Hash Results

A long time ago, in a galaxy far, far away... It is a dark time for the Rebellion. Although the Death Star has been destroyed, Imperial troops have driven the Rebel forces from their hidden base and pursued them across the galaxy.

a446277f2bebe3a799b653485d63d2fc556d602b

B long time ... far, far away... It is ... the Rebellion. ... ath Star has been d... ial troops have drive... es from their hidde... ued them across the ...

23a8c00c702d89849e5185111d9c9eff232fedb3

Is it possible to reverse engineer hash value to pain text? - No

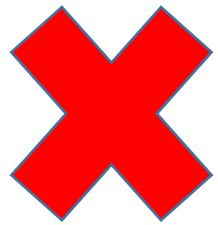
a446277f2bebe3a799b653485d63d2fc556d602b

Hash Collision

Calculated hashes for 26 bytes

Name	Length	Hash
md2	16	4EEFADA39DBD6DD3B80C05550F5B3862
md4	16	203E1D4B59EA404CE9534201A12B0C67
md5	16	A38143BC28FDF31EE635BC1DC20791F8
sha1	20	D1D0C0736427FD3CDF75697511D830B96388D6FC
sha224	28	43FC8D520F558B55C287617D46E635BEB4FDFA066E64E0EAE3(
sha256	32	5EB4FBE94549F092D9A7E79A9506ACD584AA87F3AC92E2CE62:

Hexadecimal (0~9, A~E): **16 ^ 32 = 3.402823669209387e+38** different values



(1) Hash for Password Protection

- Instead of storing pain-text passwords, their hash values are stored on Web servers. So intruders can't see the original password or reverse it.

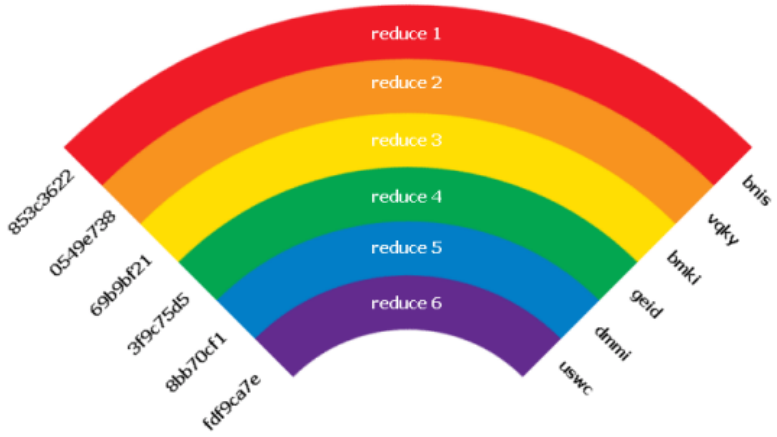
User	Password
Stephen	auhsoJ
Lisa	hsifdrowS
James	1010NO1Z
Harry	sinocarD tupaC
Sarah	auhsoJ

User	Password Hash
Stephen	39e717cd3f5c4be78d97090c69f4e655
Lisa	f567c40623df407ba980bfad6dff5982
James	711f1f88006a48859616c3a5cbcc0377
Harry	fb74376102a049b9a7c5529784763c53
Sarah	39e717cd3f5c4be78d97090c69f4e655

- Inputted password's hash value will be compared with the stored hash value to verify.

- Would longer or more complex pw still be useful? – not from a pure technical perspective, but still helpful as it will cost the hackers more to include longer passwords in a lookup table.

Rainbow Table



- Rainbow Tables are a compromise between a lookup table and low memory usage. The magic in Rainbow Tables is a basically a reduction function.

Hash with salt: The salt is typically stored right next to the salted and hashed password. Additionally, the salt should be unique per password.

```
$pwd=hash(hash($password) + salt)
```

(2) Hash as Message Digest

Alice

Hash

e10adc3949ba59ab
be56e057f20f883e

Message Digest

Compare ✓ ✗

Bob

Hash

e10adc3949ba59ab
be56e057f20f883e

- Does this process ensure confidentiality and integrity? – Only integrity.

- Subject to what types of attack? – sniffing, spoofing and man-in-the-middle.