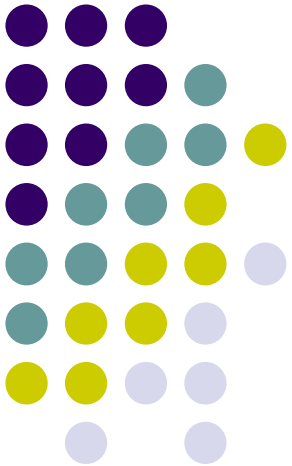
CRYPTOGRAPHY

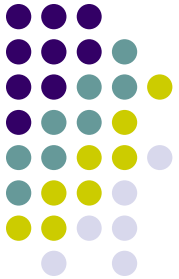
Mr.Jantapong Boodluck

Electronic Computer Technology

King Mongkut's University of Technology North Bangkok

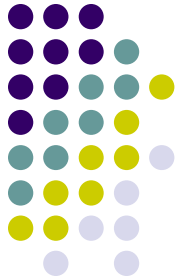


Outline



- Intro
- Encryption and Decryption
- Purpose
- Encryption Model
  - Classical Encryption
  - Modern Encryption
  - Hash Encryption

Intro

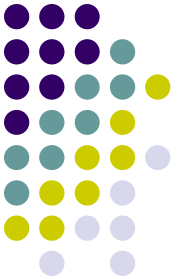


หากกล่าวถึงการเข้ารหัสและถอดรหัสมักจะได้ยินคำว่า “Cryptography” คำว่า Cryptography นี้มาจากคำว่า Crypto ที่แปลว่า “การซ่อน” ผนวกกับคำว่า Graph ที่แปลว่า “การเขียน” ดังนั้น Cryptography จึงมีความหมายว่า “การเขียนเพื่อซ่อนข้อมูล”

กระบวนการของ Cryptography มี 2 อย่างคือ Data Encryption และ Data Decryption ซึ่งหมายถึงการเข้ารหัสข้อมูล และการถอดรหัสข้อมูลตามลำดับ

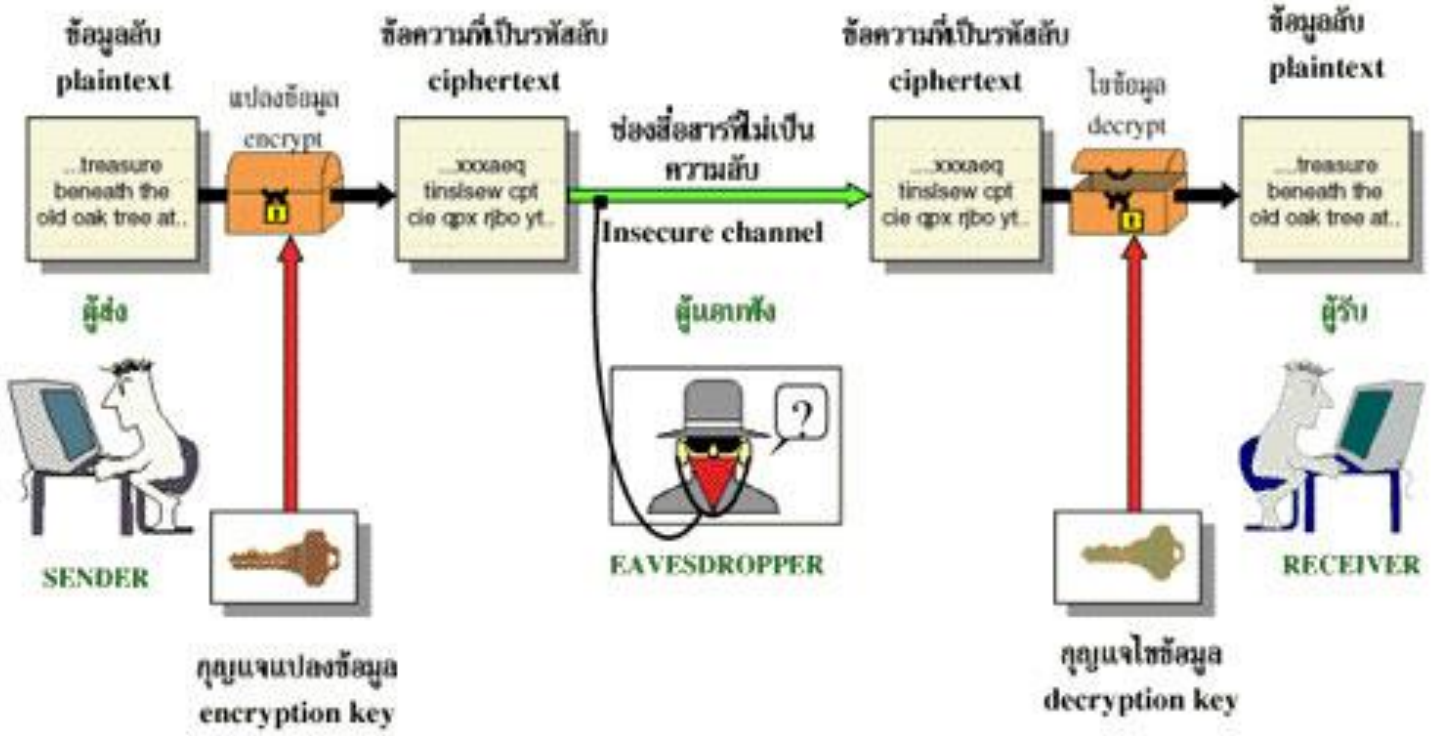
ส่วนประโยชน์ของ Cryptography คือ การรักษาความลับของข้อมูล

Encryption and Decryption



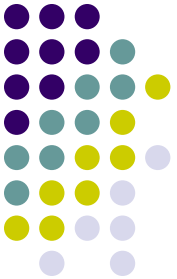
- การเข้ารหัสข้อมูลโดยพื้นฐานแล้วจะเกี่ยวข้องกับวิธีการทางคณิตศาสตร์เพื่อใช้ในการป้องกันข้อมูลหรือข้อความตั้งต้นที่ต้องการส่งไปถึงผู้รับ ข้อมูลตั้งต้นจะถูกแปรเปลี่ยนไปสู่ข้อมูลหรือข้อความอีกรูปแบบหนึ่งที่ไม่สามารถอ่านเข้าใจได้โดยใครก็ตามที่ไม่มีกุญแจสำหรับเปิดดูข้อมูลนั้น
- เราเรียกกระบวนการในการแปรรูปของข้อมูลตั้งต้นว่า **“การเข้ารหัสข้อมูล”** (Encryption)
- กระบวนการในการแปลงข้อความที่ไม่สามารถอ่านและทำความเข้าใจให้กลับไปสู่ข้อความดั้งเดิมว่า **“การถอดรหัสข้อมูล”** (Decryption)

Encryption and Decryption



- Plain Text คือข้อความตั้งต้นหรือข้อความที่ยังไม่ได้เข้ารหัส
- Cipher Text คือข้อความตั้งต้นที่เข้ารหัสแล้ว ซึ่งไม่สามารถอ่านหรือทำความเข้าใจได้

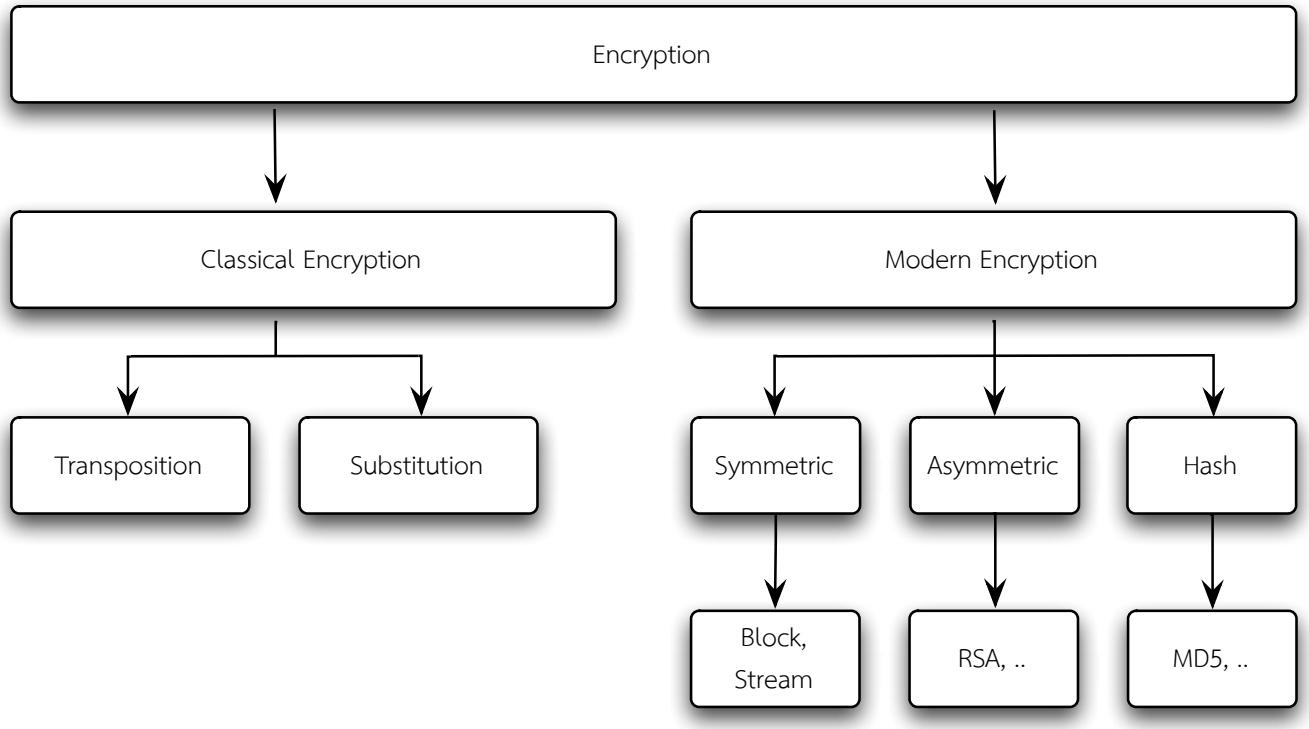
Purpose



- ผู้ที่ไม่ได้รับอนุญาตจะไม่สามารถเข้าถึงข้อมูลได้ หรือ เข้าถึงข้อมูลได้ยาก
- ให้เพียงแค่บุคคลที่อนุญาตให้เข้าถึงข้อมูลเข้าถึงข้อมูลได้เท่านั้น
- ผู้ที่ไม่ได้รับอนุญาตอาจเข้าถึงข้อมูลได้แต่ไม่สามารถเข้าใจความหมายของข้อมูลนั้นนั้นได้

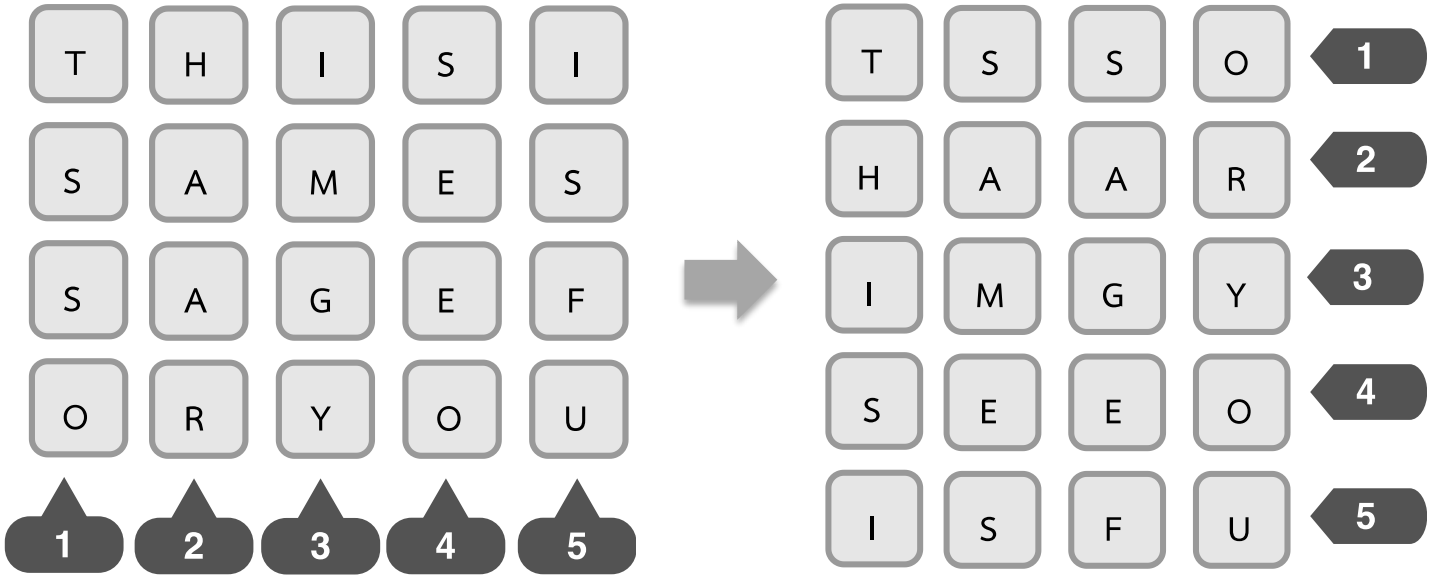
Encryption Model

หากแบ่ง Cryptography ตามยุคสมัยแล้วเราสามารถที่จะแบ่งได้เป็น 2 ยุคคือ ยุคประวัติศาสตร์ (หรือที่เรียกว่ายุค Classic) และยุคปัจจุบัน (Modern)



Classical Encryption : Transposition

- **แบบไม่ใช้ Key** : ไม่มีการสลับลำดับของหลัก
- **ตัวอย่าง Plain Text** : THIS IS A MESSAGE FOR YOU และใช้ 5 หลัก



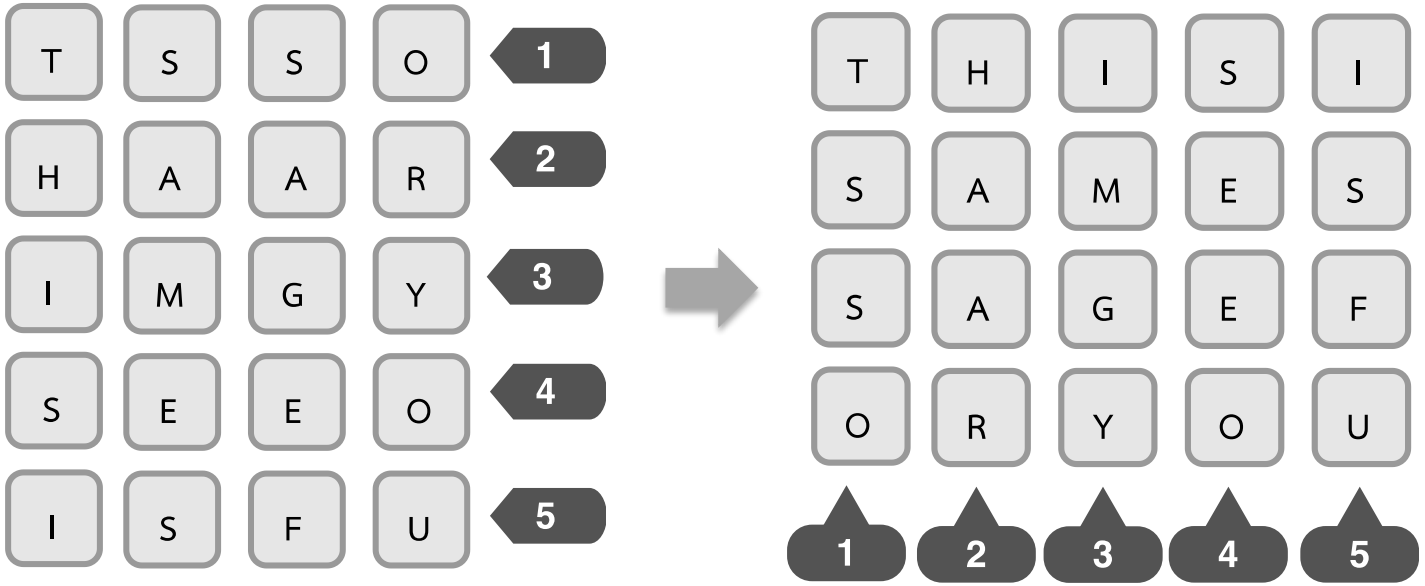
Transposition : เปลี่ยนหลักให้เป็นแถว

Cipher Text: TSSOHAARIMGYSEEOISFU

Classical Encryption : Transposition

- Transposition (Decryption)

Cipher Text: TSSOHAARIMGYSEEOISFU



Plain Text : THIS IS A MESSAGE FOR YOU

Classical Encryption : Transposition

- **แบบใช้ Key ช่วย** : มีการจัดลำดับหลักตาม key เช่น MEGABUCK
- **ตัวอย่าง Plain Text** : THIS IS A MESSAGE FOR YOU

M	E	G	A	B	U	C	K
7	4	5	1	2	8	3	6
T	H	I	S	I	S	A	M
E	S	S	A	G	E	F	O
R	Y	O	U	X	X	X	X

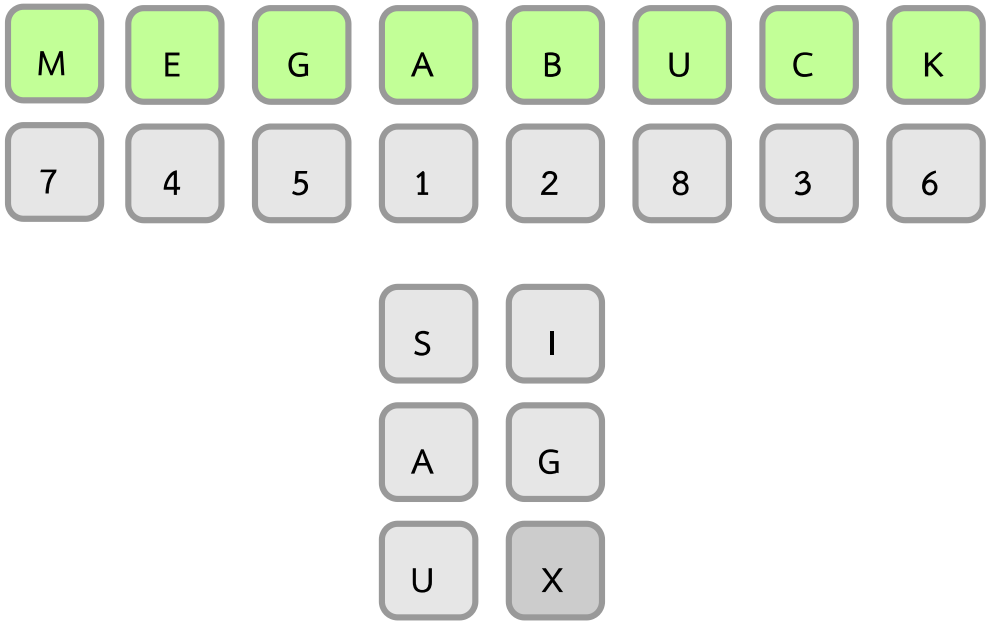
Pad Character Ex. XX

Cipher Text: : SAUIGXAFXHSYISOMOXTERSEX

Classical Encryption : Transposition

- Transposition (Decryption)

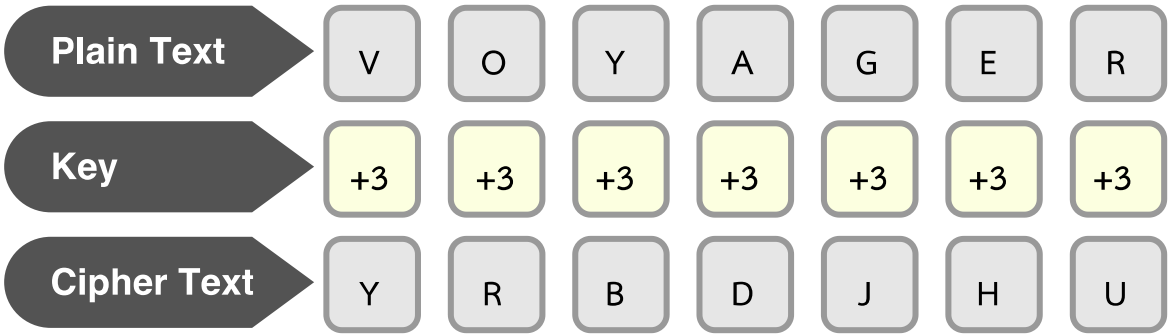
Cipher Text: SAUIGXAFXHSYISOMOXTERSEX



Plain Text : THIS IS A MESSAGE FOR YOU

Classical Encryption : Substitution

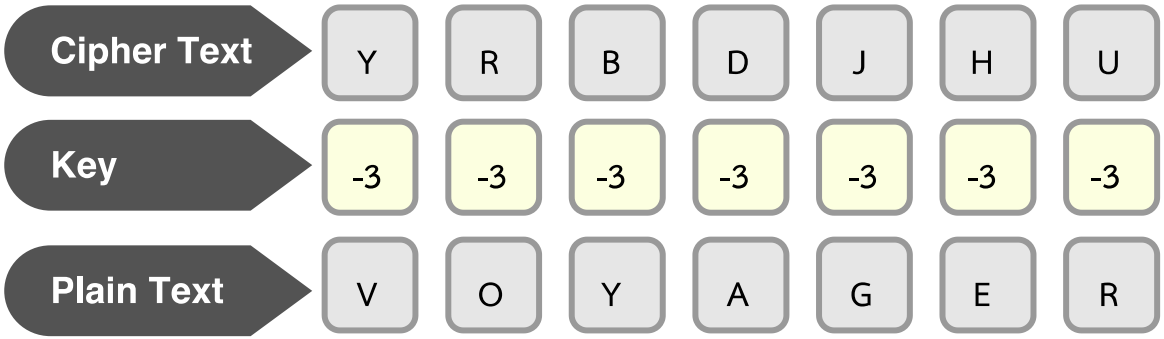
- **Caesar** (Encryption)
  - เป็นการแทนค่าแต่ละตัวอักษรด้วยสัญลักษณ์เพียงตัวเดียว เป็นวิธีที่ง่ายที่สุด ใช้มาตั้งแต่สมัยจูเลียส ซีซาร์ ในการเข้ารหัสเนื้อความจดหมายส่งไปให้ทัพทหารระหว่างการรบ



Plain Text : VOYAGER → **Cipher Text :** YRBDJHU

Classical Encryption : Substitution

- **Caesar** (Decryption)



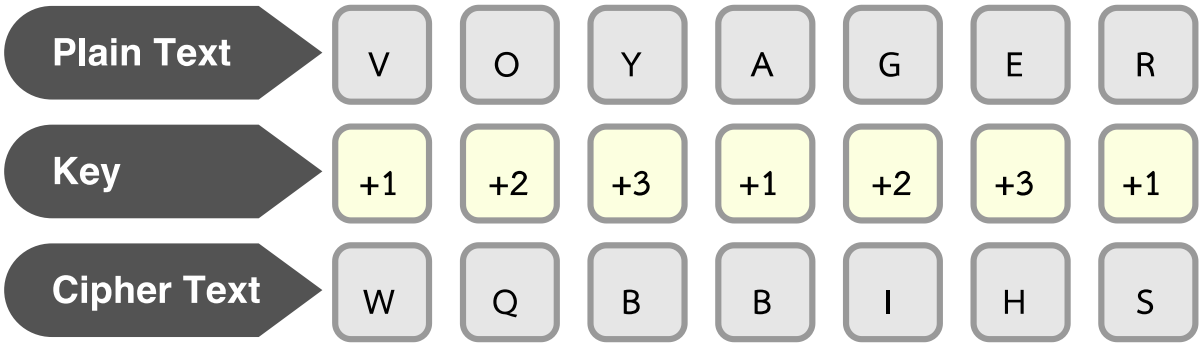
เลื่อนกลับ 1 ตำแหน่ง : .....

เลื่อนกลับ 2 ตำแหน่ง : .....

เลื่อนกลับ 3 ตำแหน่ง VOYAGER ◊ จะเจอคำที่สามารถอ่านได้

Classical Encryption : Substitution

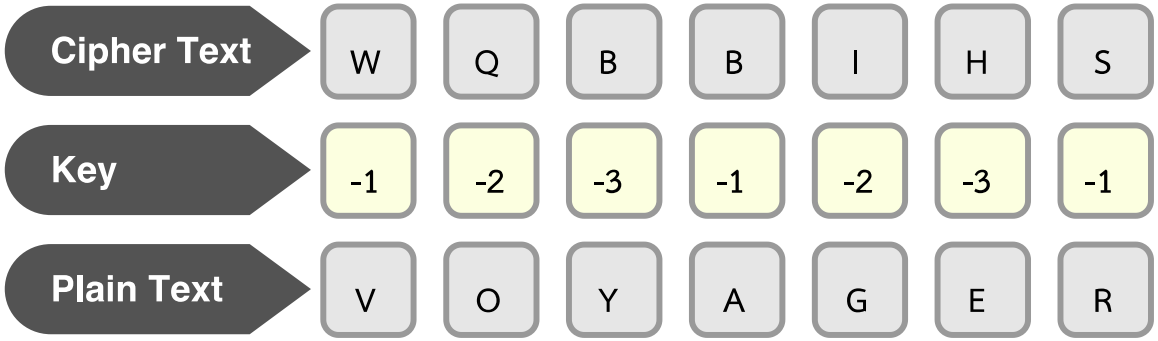
- **Caesar** (Encryption) with a key of “123”



Plain Text : VOYAGER ➡ **Cipher Text :** WQBBIHS

Classical Encryption : Substitution

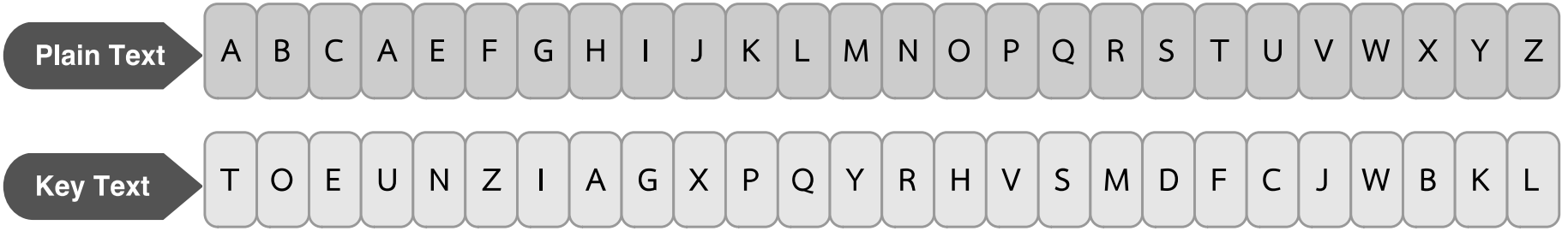
- Caesar with key (Decryption)



Cipher Text : WQBBIHS ➡ **Plain Text** : VOYAGER

Classical Encryption : Substitution

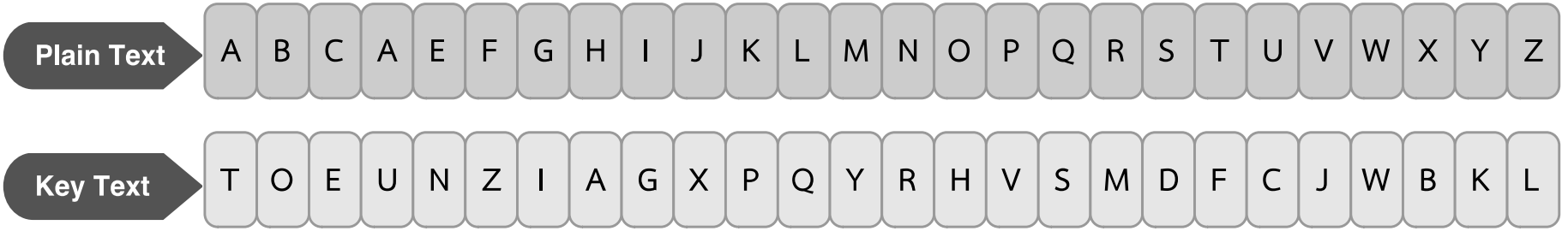
- **MonoAlphabetic** (Encryption)



Plain Text : VOYAGER ➡ **Cipher Text :** JHKTXNM

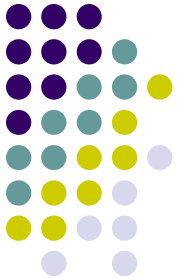
Classical Encryption : Substitution

- **MonoAlphabetic** (Decryption)



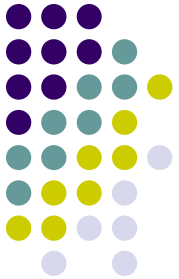
Cipher Text : JHKTXNM → **Plain Text** : VOYAGER

Classical Encryption : Vigenere



- Vigenere Cipher เป็นการเข้ารหัสแบบซีเคร็ทคีย์ (Secret Key) หรือ Symmetric Key Cryptography ที่อาศัยพื้นฐานเดียวกันกับ Caesar
- หลักการของ Vigenere Cipher คือ จะใช้ Key ที่เป็นคำมาเรียงต่อๆกัน แล้วเข้ารหัสโดยสร้าง Caesar Cipher จากตัวอักษรที่ปรากฏอยู่ใน Key

Classical Encryption : Vigenere



- **Vigenere** (Encryption) **ตัวอย่างเช่น**
  - เรามี **Plaintext : ATTACK** และเลือกใช้ Keyword : LEMON

    นำ Plaintext มาเรียงคู่กับ Keyword ให้ได้ความยาวเท่ากันดังนี้

    Plain Text : ATTACK

    Key : LEMONL

    Cipher Text : LXFOPV
  - ตัวอักษรตัวแรก - A จะถูกเข้ารหัสด้วย Caesar Cipher Key L
  - ตัวอักษรตัวที่ 2 - T จะถูกเข้ารหัสด้วย Caesar Cipher Key E
  - ตัวอักษรตัวที่ 3 - T จะถูกเข้ารหัสด้วย Caesar Cipher Key M

    และเรียงต่อไปเรื่อยๆ จนกว่าจะครบประโยค

Classical Encryption : Vigenere

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

Plain Text	A	T	T	A	C	K
Key	L	E	M	O	N	L
Cipher Text	L	X	F	O	P	V

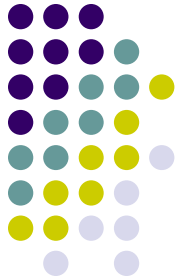
ตัวอย่างเช่น ต้องการเข้ารหัสตัว “A”

Cipher Text = P + K ; P=Plain Text, K=Key

Cipher Text = 0 + 11 = 11 ; กรณีบวกกันเกิน 25 ให้เริ่มนับที่ A = 26, B = 27,...

Cipher Text = L

Classical Encryption : Vigenere



- Vigenere (Decryption) กระบวนการย้อนกลับเหมือนกับของ Caesar แต่ต้องรู้ Keyword
  - **Ciphertext : LXFOPV และ Keyword : LEMON**

    นำ Ciphertext มาเรียงคู่กับ Keyword ให้ได้ความยาวเท่ากันดังนี้

    Ciphertext : LXFOPV

    Key : LEMONL

    Plaintext : ATTACK
  - ตัวอักษรตัวแรก - L จะถูกถอดรหัสด้วย Caesar Cipher Key L
  - ตัวอักษรตัวที่ 2 - X จะถูกถอดรหัสด้วย Caesar Cipher Key E
  - ตัวอักษรตัวที่ 3 - F จะถูกถอดรหัสด้วย Caesar Cipher Key M

    และเรียงต่อไปเรื่อยๆ จนกว่าจะครบประโยค

Classical Encryption : Vigenere

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

Cipher Text	L	X	F	O	P	V
Key	L	E	M	O	N	L
Plain Text	A	T	T	A	C	K

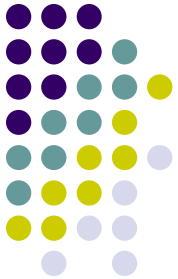
ตัวอย่างเช่น ต้องการถอดรหัสตัว “L”

Plain Text = K – C ; K=Key, C=Cipher Text

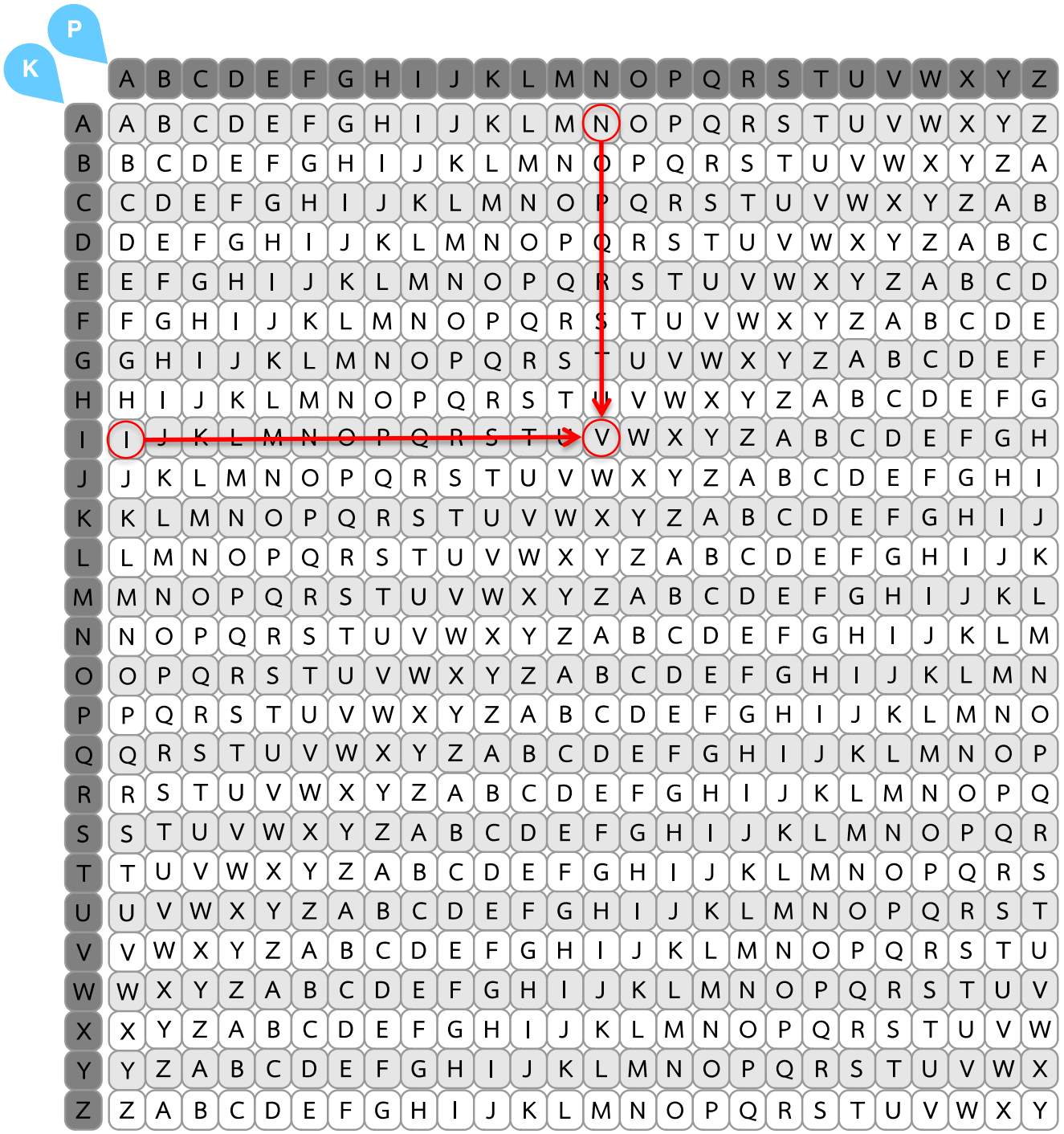
Plain Text = 11 - 11 = 0 ; ถอดรหัส F จะได้ 12-31= -19

Plain Text = A

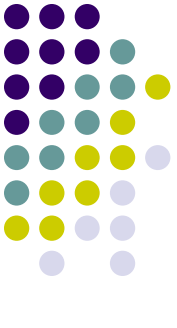
Vigenere Encryption & Decryption (Vigenere Table)



- จากตัวอย่างข้างต้นหากมีการเข้ารหัสแล้วมีค่ามากกว่า 25 อาจเป็นผลให้เกิดข้อผิดพลาดในการถอดรหัส คือได้ Plain Text ที่ไม่ถูกต้อง
- Vigenere Square Or Vigenere Table เป็นตารางที่จะนำมาทำการ Encrypt และ Decrypt อีกวิธีหนึ่งที่สามารถทำได้อย่างถูกต้อง โดยไม่สนใจว่า ค่าตำแหน่งจะมากกว่า 25 หรือเป็นค่าเท่าไร เพราะวิธีนี้จะใช้การเปรียบเทียบจากตาราง



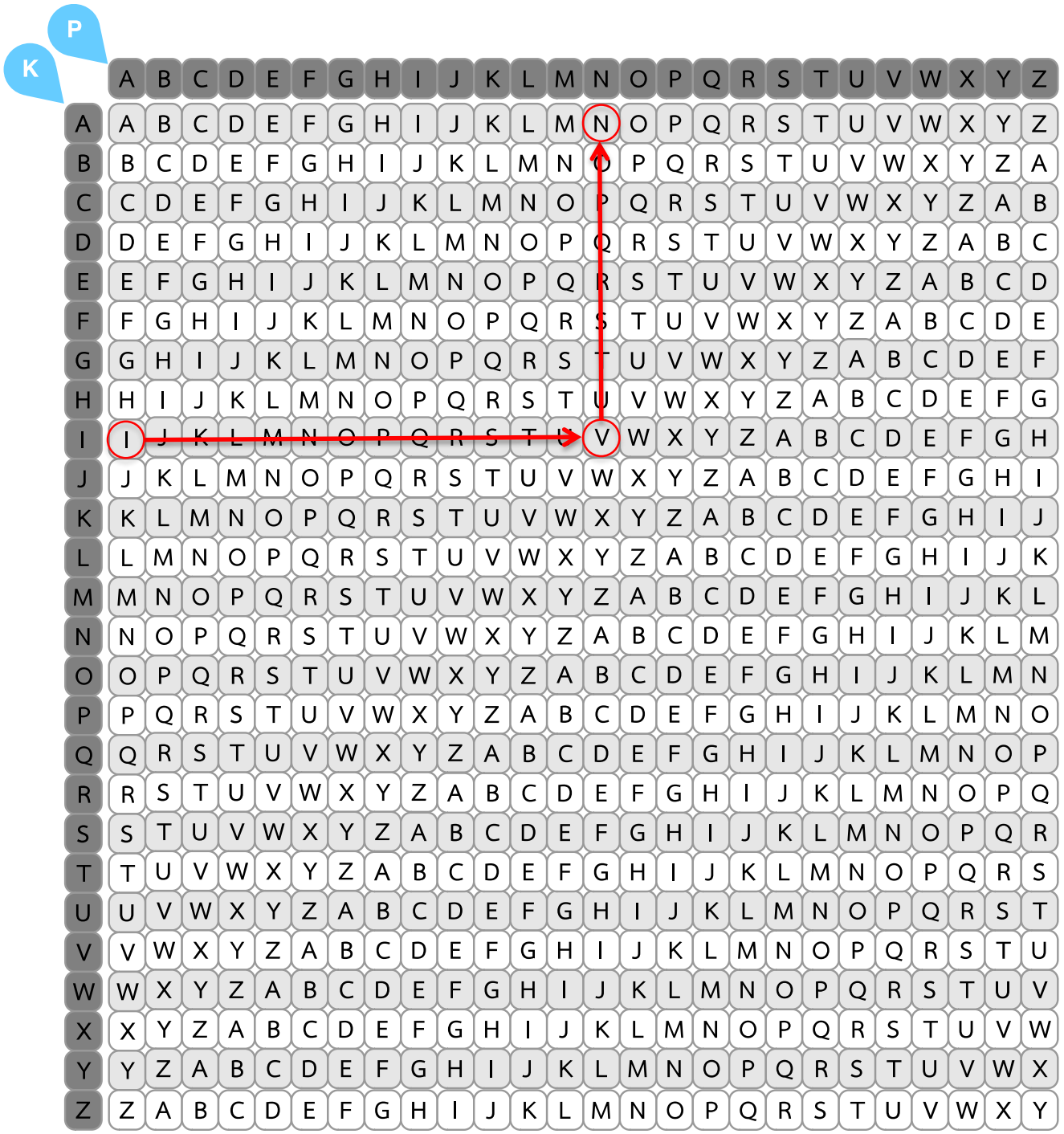
P=Plain Text
K=Key Text



Plain Text: NETWORK

Key: IPSECIP

Cipher Text: VTLBQZZ



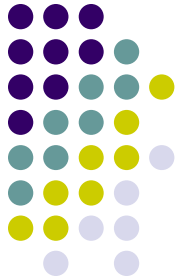
P=Plain Text
K=Key Text

Cipher Text: VTLBQZZ

Key: IPSECIP

Plain Text: NETWORK

Modern Encryption

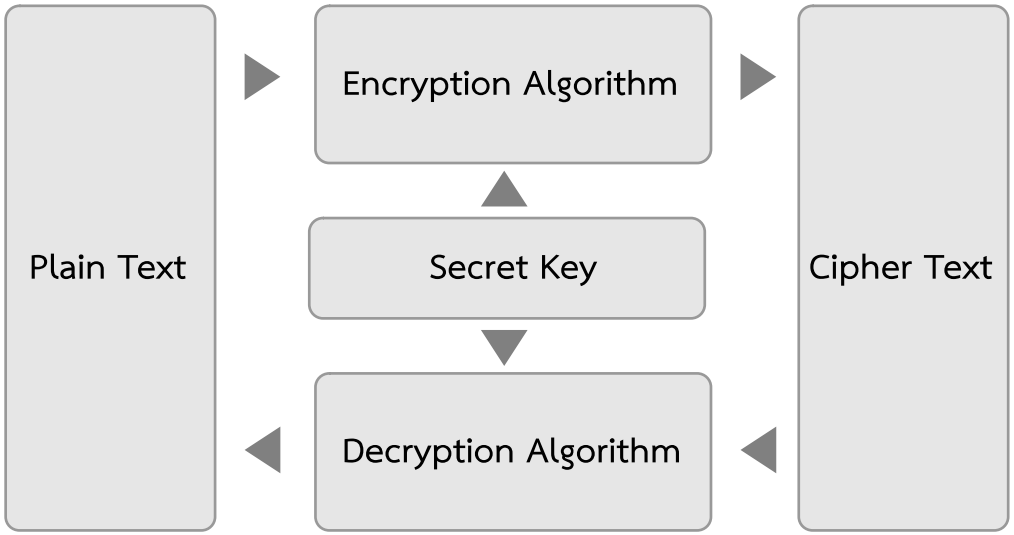


- การเข้ารหัสแบบสมมาตร (Symmetric Key Algorithms)
- การเข้ารหัสแบบอสมมาตร (Asymmetric Key Algorithms)
- การเข้ารหัสแบบ Hash (Cryptography Hash)

Modern Encryption :

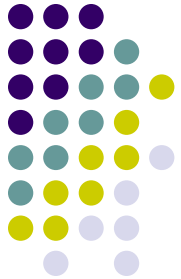
Symmetric Key Algorithms

- การเข้ารหัสแบบสมมาตร (Symmetric Key Algorithms)
  - การเข้ารหัสแบบสมมาตร จะใช้กุญแจลับในการติดต่อกันระหว่าง 2 คน อันเดียวกันทั้งในการเข้ารหัสและถอดรหัส
  - ก่อนที่จะส่งข้อมูลที่ถูกเข้ารหัสแล้วผ่านระบบเน็กเวิร์คทั้ง 2 กลุ่มต้องมีกุญแจและอัลกอริทึมที่ตกลงร่วมกัน เพื่อใช้ในการเข้ารหัสและถอดรหัส



Modern Encryption :

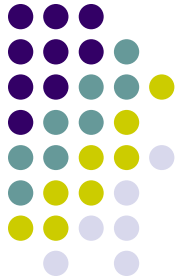
Symmetric Key Algorithms



- ปัญหาของการเข้ารหัสแบบสมมาตร
  - ปัญหาที่เกิดขึ้นในการใช้กุญแจลับคือ การส่งกุญแจให้อีกกลุ่มหนึ่ง แล้วโดนดักลักลอบเอากุญแจไปโดยผู้ไม่ประสงค์ดี
  - ถ้า นาย ก และ นาย ข ใช้การส่งข้อมูลโดยใช้ กุญแจลับ และ นาย ค ดัก กุญแจลับระหว่างการส่งกุญแจจะทำให้ นาย ค สามารถอ่านข้อมูลลับ ที่ส่งกันระหว่าง นาย ก กับ นาย ข
  - ไม่เพียงแค่นั้น นาย ค อาจจะสามารถสร้างข้อมูลหลอกนาย ก ว่าเป็น นาย ข ได้
  - ถ้าติดต่อกับหลายกลุ่ม ต้องใช้กุญแจจำนวนมาก

Modern Encryption :

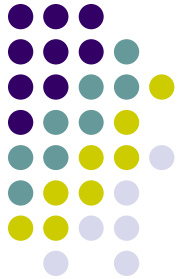
Symmetric Key Algorithms



- Symmetric Encryption
  - การเข้ารหัสข้อมูลแบบทีละตัว (Stream Cipher)
    - Caesar, Vigenere
  - การเข้ารหัสข้อมูลแบบเป็นกลุ่ม (Block Cipher)
    - Transposition

Modern Encryption :

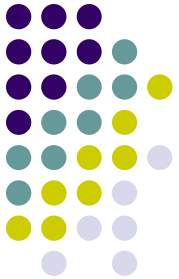
Symmetric Key Algorithms



- การเข้ารหัสข้อมูลแบบทีละตัว (Stream Cipher)
  - การเข้ารหัสแบบนี้เป็นการเปลี่ยนตัวอักษรของข้อความไปเป็นสัญลักษณ์ Cipher ทีละตัวเช่น การเข้ารหัสแบบ Caesar, Vigenere
  - การเข้ารหัสแบบนี้จะทำการเข้ารหัสของข้อความทีละตัว

Modern Encryption :

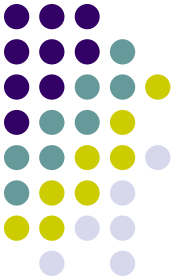
Symmetric Key Algorithms



- ข้อดีของการเข้ารหัสทีละตัว (Stream Cipher)
  - ความเร็ว (Speed of Transformation) เนื่องจากการเข้ารหัสแบบนี้ทำทีละตัวโดยที่ขั้นตอนในการเข้ารหัสนั้นไม่ต้องรอหรือเกี่ยวข้องกับตัวอักษรอื่น ๆ
  - ความผิดพลาดต่ำ (Low Error Propagation) เนื่องจากความผิดพลาดที่เกิดขึ้นกับตัวอักษรตัวใดตัวหนึ่งในระหว่างการทำการเข้ารหัสนั้นจะไม่มีผลต่อกระบวนการเข้ารหัสของตัวอักษรอื่น ๆ เลย

Modern Encryption :

Symmetric Key Algorithms

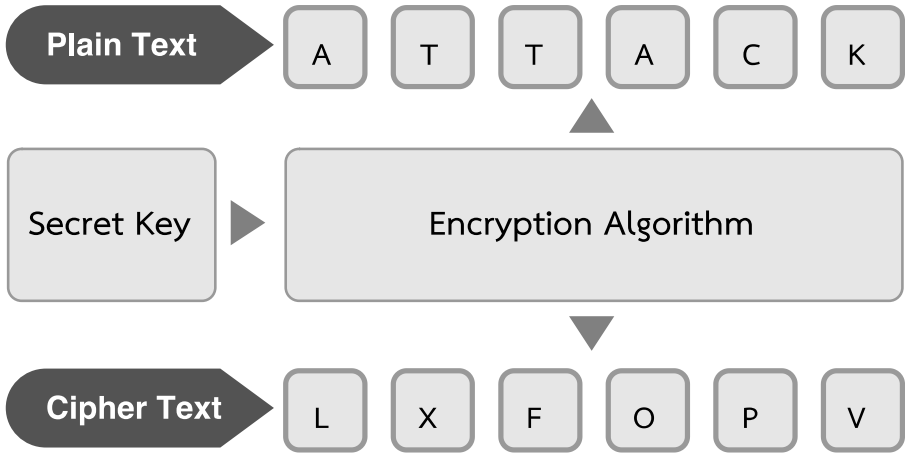


- ข้อเสียของการเข้ารหัสทีละตัว (Stream Cipher)
  - มีความสามารถต่ำในการกระจายความซ้ำกันของตัวอักษร (Low-Diffusion) เนื่องจากการเข้ารหัสแบบนี้ทำทีละตัวโดยไม่คำนึงถึงตัวอักษรอื่น ดังนั้นการกระจายความซ้ำกันจึงไม่อาจทำได้ และก่อให้เกิดสิ่งที่เรียกว่า Pattern ของตัวอักษรขึ้นซึ่งจะทำให้ง่ายต่อการวิเคราะห์ของนักเจาะรหัส
  - ง่ายต่อการดัดแปลงแก้ไข (Susceptibility to malicious insertion and modifications) เพราะว่าการเข้ารหัสแบบนั้นไม่ต้องอาศัยความถูกต้องในการเข้ารหัสของตัวอักษรอื่นๆเลย ดังนั้นความเป็นไปได้ประการหนึ่งก็คือหากมีผู้เจาะรหัสที่สามารถล้วงความลับของข้อมูลได้แล้วก็อาจทำการเปลี่ยนแปลงแก้ไขข้อมูลนั้นๆ หรือทำการใส่ข้อความเพิ่มเติมเข้าไปได้โดยที่ผู้ทำการถอดรหัสไม่อาจล่วงรู้ได้ว่ามีการทุจริตกับข้อความที่ได้รับมา

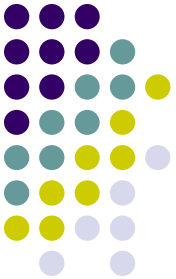
Modern Encryption :

Symmetric Key Algorithms

- การเข้ารหัสข้อมูลแบบเป็นกลุ่ม (Block Cipher)
  - การเข้ารหัสข้อมูลแบบนี้จะทำเป็นกลุ่ม (Group) หรือเป็น Block ข้อมูล แทนที่จะทำการเข้ารหัสทีละตัว เช่น การเข้ารหัสแบบ Transposition
  - การเข้ารหัสแบบนี้มีข้อดีหลายประการที่เหนือกว่าการเข้ารหัสแบบทีละตัว

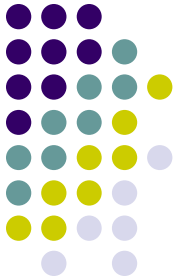


Modern Encryption : Symmetric



- ข้อดีของการเข้ารหัสแบบ Block Cipher
  - การกระจายความซ้ำกันของตัวอักษร (Diffusion) การทำการเข้ารหัสแบบนี้จะทำให้ตัวอักษรของข้อความที่อ่านได้ (Plain Text) แต่ละตัวนั้นถูกแทนค่าด้วยตัวอักษรหลายตัวในข้อความที่เข้ารหัสแล้ว (Cipher Text) ดังนั้นการวิเคราะห์หาข้อความเดิมจึงทำได้ยากกว่า
  - สามารถต่อต้านการดัดแปลงต่อเติมข้อมูลได้ (Insertion Resistant) เนื่องจากวิธีนี้ข้อมูลจะถูกเข้ารหัสทีละกลุ่ม ดังนั้นการขโมยข้อมูลนี้ไปต่อเติม จะไม่สามารถทำได้เนื่องจากว่าความยาวของข้อมูลใน Block จะเปลี่ยนไปทำให้ไม่สามารถทำการถอดรหัสได้อย่างถูกต้อง

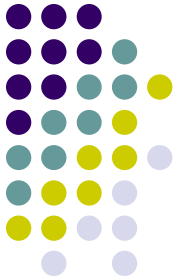
Modern Encryption : Symmetric



- ข้อเสียของการเข้ารหัสแบบ Block Cipher
  - ใช้เวลานาน (Slowness of Encryption) การเข้ารหัสแบบนี้ไม่สามารถที่จะทำได้หากข้อมูลที่จะทำการเข้ารหัสไม่ครบ หรือไม่เต็ม Block ดังนั้นก่อนที่จะทำการเข้ารหัสจำเป็นต้องรอข้อมูลให้เต็มเสียก่อน
  - มีความผิดพลาดต่อเนื่อง (Error Propagation) การเข้ารหัสแบบนี้ก่อให้เกิดความผิดพลาดของตัวอักษรที่สามารถส่งต่อความผิดพลาดไปยังตัวอักษรอื่นๆ ได้

Modern Encryption :

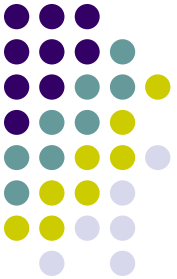
Asymmetric key algorithms

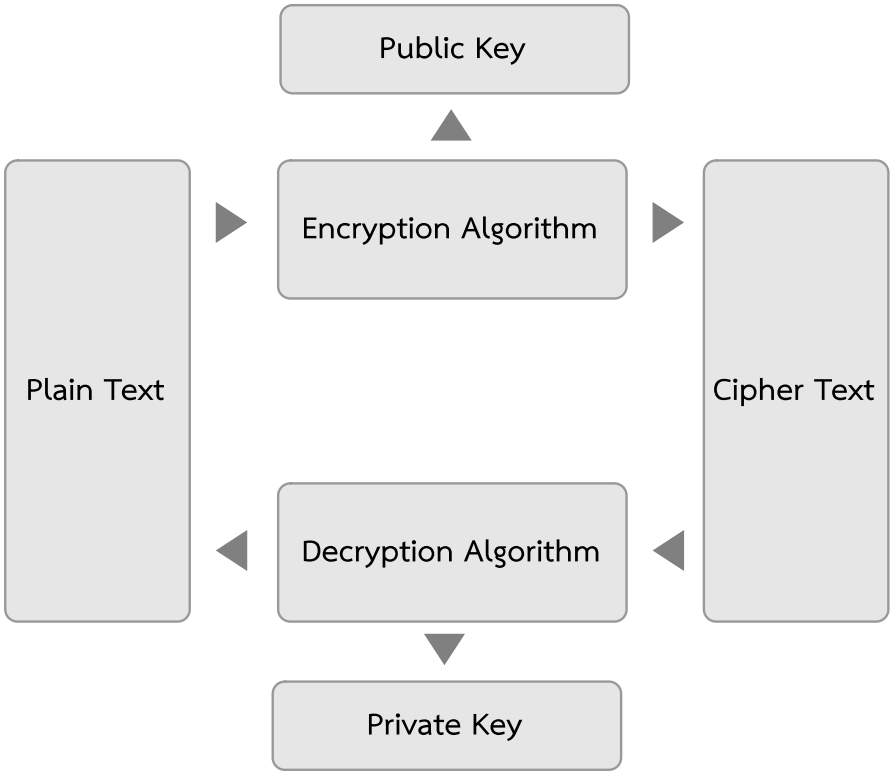


- การเข้ารหัสแบบอสมมาตร **(Asymmetric key algorithms)**
  - การเข้ารหัสแบบอสมมาตร เป็นการแก้ปัญหาโดยใช้หลักการของ กุญแจสาธารณะและกุญแจส่วนตัว
  - โดยที่กุญแจสาธารณะนั้นเปิดเผย ในระบบเน็กเวิร์ดได้ ส่วนกุญแจส่วนตัวนั้นเก็บไว้เฉพาะบุคคลเท่านั้น
  - การใช้กุญแจสาธารณะและกุญแจส่วนตัวในการเข้ารหัสนั้นเป็นแบบตรงข้ามกัน คือมีกุญแจเป็นคู่ 2 อันคือ ใช้กุญแจอันหนึ่งเข้ารหัส ต้องใช้อีกกุญแจ เพื่อทำการถอดรหัสเท่านั้น

Modern Encryption :

Asymmetric Key Algorithms

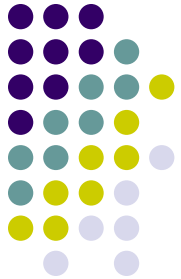




กุญแจเป็นคู่ 2 อันคือ ใช้ กุญแจ อันหนึ่ง เข้ารหัส ต้องใช้อีกกุญแจ เพื่อทำการ ถอดรหัส เท่านั้น

Modern Encryption :

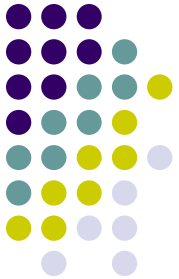
Asymmetric Key Algorithms



- ตัวอย่าง
  - นาย ข ต้องการส่ง ข้อมูลลับ ให้กับ นาย ก
  - ในการเข้ารหัส นาย ก มีคู่ของกุญแจคือ กุญแจสาธารณะและกุญแจส่วนตัว
  - จากนั้น นาย ก ได้ส่ง กุญแจสาธารณะของ นาย ก ไปให้ นาย ข และเก็บกุญแจส่วนตัว ไว้กับตัวเอง
  - เมื่อ นาย ข ได้ กุญแจสาธารณะของ นาย ก แล้วจึงทำการเข้ารหัสข้อมูลด้วย กุญแจสาธารณะของ นาย ก แล้วทำการส่งข้อมูลลับให้ นาย ก
  - เมื่อ นาย ก ได้รับแล้วทำการถอดรหัส ด้วย กุญแจส่วนตัวของ นาย ก จะได้ข้อมูลที่ นาย ข ต้องการส่งให้ นาย ก

Modern Encryption :

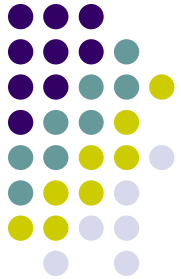
Asymmetric Key Algorithms



- ตัวอย่าง (ต่อ)
  - ถ้า นาย ก เข้ารหัสข้อมูลด้วยกุญแจส่วนตัวของ นาย ก และส่ง ข้อมูลลับไปให้ นาย ข นาย ข จะสามารถแน่ใจได้เลยว่าข้อมูลลับ นี้มาจาก นาย ก
  - ถ้า นาย ข สามารถถอดรหัสข้อมูลลับได้ด้วยกุญแจสาธารณะของ นาย ก แสดงว่าข้อมูลลับถูกเข้ารหัสมาด้วย กุญแจส่วนตัวของ นาย ก
  - นาย ก เท่านั้นที่มี กุญแจส่วนตัว ของ นาย ก นาย ข จึงแน่ใจได้ว่าข้อมูลนี้มาจาก นาย ก

Modern Encryption :

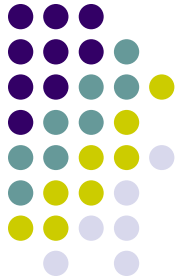
Asymmetric Key Algorithms



- **ปัญหาของการเข้ารหัสแบบอสมมาตร**
  - ในกรณีนี้คนที่มี กุญแจสาธารณะของ นาย ก สามารถอ่านข้อมูลลับนี้ได้ เพราะกุญแจสาธารณะของ นาย ก อนุญาตให้คนทั่วไปมีได้
  - การเข้ารหัสแบบอสมมาตรใช้การคำนวณที่ค่อนข้างซับซ้อนทำให้เสียเวลาในการเข้ารหัส และ ถอดรหัส ข้อมูลมักเหมาะกับการเข้ารหัสข้อมูล ที่ค่อนข้างเล็ก เช่น กุญแจลับ ไม่ค่อยนิยม เข้ารหัส และ ถอดรหัสตัวข้อมูลทั้งหมด

Modern Encryption :

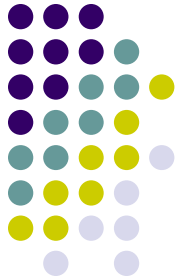
Asymmetric Key Algorithms



- RSA
  - RSA เป็นอัลกอริทึมในการเข้าหัสแบบอสมมาตร ถูกสร้างขึ้นมาเมื่อปี 1978 โดย Ron **R**ivest, Adi **S**hamir และ Leonard **A**dleman ตั้งแต่คิดค้นมายังไม่มีใครสามารถเบรคอัลกอริทึมนี้ได้ และ RSA ได้ถูกนำมาใช้อย่างแพร่หลายในด้าน E-Commerce
  - อัลกอริทึมนี้สามารถใช้ในการเข้ารหัสข้อมูลรวมทั้งการลงลายมือชื่ออิเล็กทรอนิกส์ด้วย

Modern Encryption :

Asymmetric Key Algorithms



- **วิธีการทำงานและการคำนวณของ RSA**

  (1) เลือก p และ q ซึ่งเป็นจำนวนเฉพาะที่มีค่าต่างกัน

  (2) ให้ n = pq

  (3) ให้ m = (p-1)(q-1)

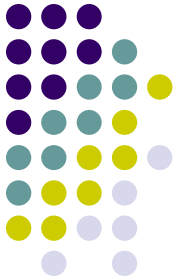
  (4) เลือกค่า e ที่ 1 < e < m ซึ่งหารร่วมมากของ m กับ e มีค่าเป็น 1

  (สามารถหาค่า e ได้โดยการสุ่มค่าจำนวนเต็มบวกพร้อมกับทดสอบว่าหารร่วมมากของ m กับ e มีค่าเป็น 1)

  (5) หาค่า d ที่ทำให้ ed mod m = 1

Modern Encryption :

Asymmetric Key Algorithms



(6) Public key คือ (n,e)

(7) Private key คือ (n,d)

(8) ให้ M คือข้อความที่ยังไม่ถูกเข้ารหัส (ในรูปแบบของตัวเลข) M < n

(9) การเข้ารหัส => C = M^e mod n

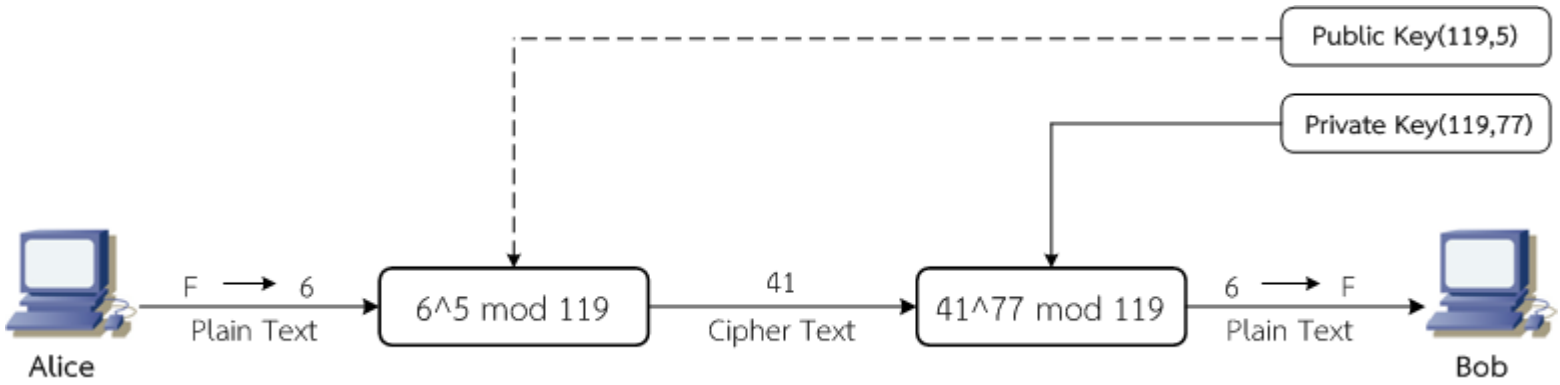
(10) การถอดรหัส => M = C^d mod n

Modern Encryption :

Asymmetric Key Algorithms

การเข้ารหัส => C = M^e mod n

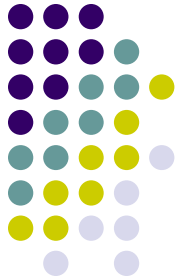
การถอดรหัส => M = C^d mod n



RSA Encrypt & Decrypt

Modern Encryption :

Asymmetric Key Algorithms



- **ตัวอย่างการเข้ารหัสและถอดรหัสด้วย RSA**

(1) เลือก p และ q ซึ่งเป็นจำนวนเฉพาะที่มีค่าต่างกัน

p = 7

q = 17

(2) ให้ n = pq

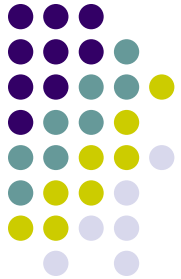
ดังนั้น n = 7*17 = 119

(3) ให้ m = (p-1)(q-1)

ดังนั้น m = 6*16 = 96

Modern Encryption :

Asymmetric Key Algorithms



(4) เลือกค่า e ที่ 1 < e < m ซึ่งหารร่วมมากของ m กับ e มีค่าเป็น 1

(สามารถหาค่า e ได้โดยการส่มค่าจำนวนเต็มบวกพร้อมกับทดสอบว่าหารร่วมมากของ m กับ e มีค่าเป็น 1)

เลือก e = 5 และทดสอบหารร่วมมากของ 96 กับ 5 แล้วได้ 1

(5) หาค่า d ที่ทำให้ ed mod m = 1

ได้ค่า d = 77 เพราะ 5*77 mod 96 ได้ 1

(6) Public key คือ (n,e)

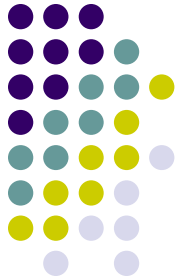
ดังนั้น Public key คือ (119,5)

(7) Private key คือ (n,d)

ดังนั้น Private key คือ (119,77)

Modern Encryption :

Asymmetric Key Algorithms



(8) ให้ M คือข้อความที่ยังไม่เข้ารหัส (ในรูปแบบของตัวเลข) M < n

ให้ข้อความที่ยังไม่เข้ารหัส M = 19

(9) การเข้ารหัส => C = M^e mod n

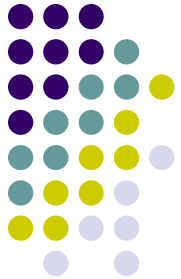
ได้ C = 19^5 mod 119 = 66

(10) การถอดรหัส => M = C^d mod n

ได้ M = 66^77 mod 119 = 19

Modern Encryption :

Asymmetric Key Algorithms



- ตัวอย่าง 2
  - ถ้า p=5, q=7, C=17 M=?

1. หาค่า n=5X7=35

2. หาค่า m=(5-1)(7-1)=24

3. หาค่า e = 1 < e < 24 ** e เป็นจำนวนจำเพาะที่ไม่สามารถหาร 24 ลงตัว เลือกตั้งแต่ 2 ขึ้นไป หรือหารร่วมมากของ m กับ e คือ 1

24 = 1*2*2*2*3

5 = 1*5

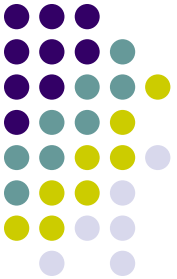
จะได้ e = 5 **เพราะหารร่วมมากของ m กับ e คือ 1

4. หาค่า d โดย e.d mod m = 1 ได้ d=5

ดังนั้น M=17^5 mod 35 = 12

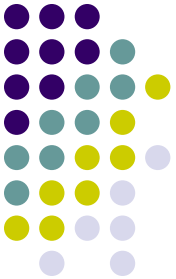
Modern Encryption :

Compare Symmetric & Asymmetric key



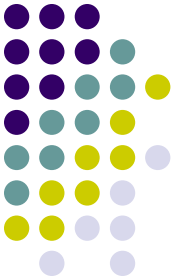
- การเข้ารหัสแบบสมมาตร ข้อมูลทำการเข้ารหัสและถอดรหัสได้รวดเร็ว แต่ก่อนจะต้องมีการตกลงกุญแจกันก่อน และ มีปัญหาในการแลกเปลี่ยนกุญแจที่ไม่มีความปลอดภัย
- การเข้ารหัสแบบอสมมาตรการแลกเปลี่ยนกุญแจนั้นไม่มีปัญหา เพราะกุญแจสาธารณะไม่เป็นความลับแต่อัลกอริทึมในการเข้ารหัสและถอดรหัสนั้นเสียเวลามากทำให้ช้า
- การเข้ารหัสแบบสมมาตร ถ้าต้องการติดต่อกับกลุ่มหลายกลุ่มต้องใช้กุญแจลับหลายกุญแจ แต่แบบอสมมาตรจะใช้แค่กุญแจสารธารณะและกุญแจส่วนตัวเท่านั้น
- ระบบกุญแจสาธารณะต้องใช้เวลาในการคำนวณการเข้าและถอดรหัส เมื่อเทียบกับระบบกุญแจสมมาตร และอาจใช้เวลาเป็นพันเท่าของเวลาที่ใช้โดยระบบกุญแจสมมาตร

Modern Encryption : Cryptography Hash



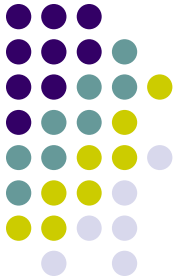
- การเข้ารหัสแบบ Hash (Cryptography hash) หมายถึง การแปลงรูปแบบของข้อมูลที่รับเข้ามาให้เป็นข้อมูลที่ถูกย่อย (Message Digest) ไม่ว่าข้อมูลต้นฉบับจะมีขนาดเล็กหรือใหญ่เท่าใดก็ตามก็จะถูกย่อยให้อยู่ในรูปแบบที่มีขนาดคงที่
- ดังนั้นจึงไม่สามารถทำกระบวนการย้อนกลับเพื่อให้กลายเป็นข้อมูลต้นฉบับได้ จะทำได้เพียงแค่ตรวจสอบว่าข้อมูลที่ให้มาแต่ละครั้งเหมือนกันหรือไม่
- ฟังก์ชั่น Hash ที่สำคัญ ๆ ได้แก่ MD4, MD5, SHA-1 และ SHA-2

Modern Encryption : Cryptography Hash



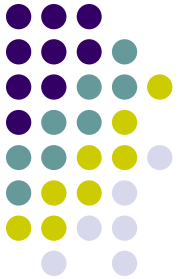
- คุณสมบัติที่สำคัญ
  - ทุกๆ บิตของไดเจสต์จะขึ้นอยู่กับทุกบิตของข้อความตั้งต้น
  - การเปลี่ยนแปลงแก้ไขข้อความตั้งต้นโดยผู้ไม่ประสงค์ดีแม้ว่าอาจแก้ไขเพียงเล็กน้อยก็ตาม ก็จะส่งผลให้ผู้รับข้อความทราบว่าข้อความที่ตนได้รับไม่ใช่ข้อความตั้งต้น
  - โอกาสที่ข้อความตั้งต้น 2 ข้อความใดๆ ที่มีความแตกต่างกัน จะสามารถคำนวณได้ค่าไดเจสต์เดียวกันมีโอกาสน้อยมาก
  - คุณสมบัติข้อนี้ทำให้แน่ใจได้ว่า เมื่อผู้ไม่ประสงค์ดีทำการแก้ไขข้อความตั้งต้น ผู้รับข้อความที่ถูกแก้ไขไปแล้วนั้นจะสามารถตรวจพบได้ถึงความผิดปกติที่เกิดขึ้นอย่างแน่นอน
  - อย่างไรก็ตามในทางทฤษฎีแล้ว มีโอกาสที่ข้อความ 2 ข้อความที่แตกต่างกันจะสามารถคำนวณแล้วได้ค่าไดเจสต์เดียวกัน *ปัญหานี้เรียกกันว่าการชนกันของไดเจสต์(Collision)*
  - อัลกอริทึมสำหรับสร้างไดเจสต์ที่ดีควรจะมีโอกาสน้อยมากๆ ที่จะก่อให้เกิดปัญหาการชนกันของไดเจสต์

Modern Encryption : Cryptography Hash



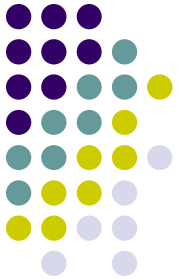
- ตัวอย่างเช่น
  - ผู้ใช้ User1 ตั้งรหัสผ่านเป็นคำว่า abc123 หากเก็บรหัสผ่านลงบน Database โดยตรง จะทำให้ผู้ใดก็ตามที่เข้าถึงฐานข้อมูลได้ ทราบรหัสผ่านที่เก็บ เช่น ผู้ดูแลระบบ ผู้ดูแลฐานข้อมูลและแฮกเกอร์ที่เจาะเข้ามา(SQL Injection)
  - หากทำการย่อยรหัสผ่านด้วยฟังก์ชั่น Hash เช่นใช้ MD5 ย่อยรหัสผ่าน abc123 ได้เป็น e99a18c428cb38d5f260853678922e03 แล้วจึงเก็บค่าแฮชนั้นลงใน Database จะทำให้การเปิดดูรหัสผ่านใน Database โดยตรง ไม่พบรหัสผ่าน abc123 แต่จะพบเพียงค่าแฮช (e99a18c428cb38d5f260853678922e03)

Modern Encryption : Cryptography Hash



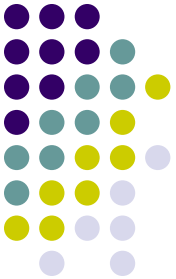
- ซึ่งเป็นการป้องกันการเปิดเผยรหัสผ่านและไม่สามารถใช้ค่าแฮชเพื่อคำนวณย้อนกลับไปเป็นรหัสผ่านได้ ในการตรวจสอบสิทธิ์ผู้ใช้แต่ละครั้งสำหรับการล็อกอินก็สามารถทำได้โดยนำรหัสผ่านที่ผู้ใช้ส่งผ่านฟอร์มล็อกอินเข้ามา แล้วนำไปผ่านฟังก์ชั่น Hash เช่น MD5 จากนั้นก็นำค่าแฮชที่ได้มาเทียบกับค่าแฮชที่เก็บไว้ใน Database หากมีค่าตรงกันก็แสดงว่ารหัสผ่านถูกต้อง

Modern Encryption : Cryptography Hash



- Md2,Md3,Md4,Md5
  - ผู้พัฒนาคือ Ronald Rivest อัลกอริทึมนี้เชื่อกันว่ามีความแข็งแกร่งที่สุดในบรรดาอัลกอริทึมต่างๆ ที่ Rivest พัฒนาขึ้นมา
  - แม้จะเป็นที่นิยมใช้งานกันอย่างแพร่หลาย ทว่าในปี 1996 ก็มีผู้พบจุดบกพร่องของ MD5 จึงทำให้ความนิยมเริ่มลดลง MD5 ผลิตไดเจสต์ที่มีขนาด 128 บิต
  - MD5 ก็ถูกเบรคได้โดยนักคณิตศาสตร์หญิงชาวจีน (Professor Dr. Xiaoyun Wang) ในปี 2004 โดยใช้เครื่องซูเปอร์คอมพิวเตอร์ IBM P690 และใช้เวลาแคร็ก 1 ชั่วโมงก็สามารถเบรคได้

Modern Encryption : Cryptography Hash



- SHA และ SHA1 ได้ถูกพัฒนาให้มีความแข็งแรงกว่า MD5 โดยได้พัฒนาจาก MD5 เดิมให้ Output มีความเป็น Random สูงกว่า และมี Collision น้อยกว่าเพื่อลดโอกาสในการถูกแคร็กได้
- SHA และ SHA1 ก็ถูกเบรคได้โดยนักคณิตศาสตร์หญิงชาวจีน (Professor Dr. Xiaoyun Wang) คนเดียวกันกับที่เคยเบรค MD5 ได้ ดังนั้นปัจจุบันนี้ความหวังจึงอยู่ที่ SHA2 ซึ่งยังไม่มีใครเบรคได้ อัลกอริทึมของ SHA2

Resources

- Computer Security Division Computer Security Resource Center / National Institute of Standard and Technology (CSRC/NIST)
  - csrc.nist.gov
- Computer Emergency Response Team(CERT)
  - www.cert.org
- อาจารย์ ธนัญชัย ตรีภาค สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
- Bruce Schneier, Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons Inc, December 1995
- Security Focus
  - www.securityfocus.com