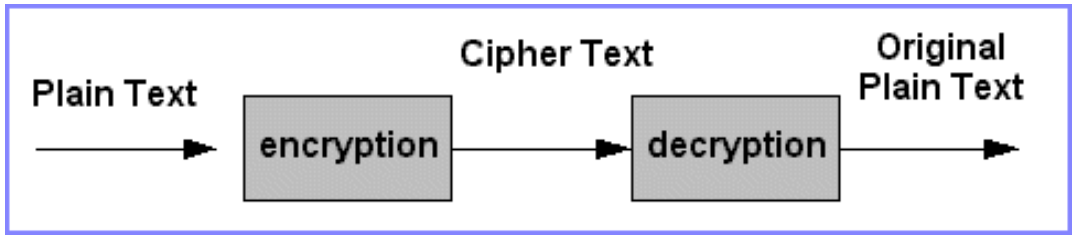
Secrete key Cryptography and Public Key Cryptography

บทที่ 4 (ต่อ)

Why Encrypt?

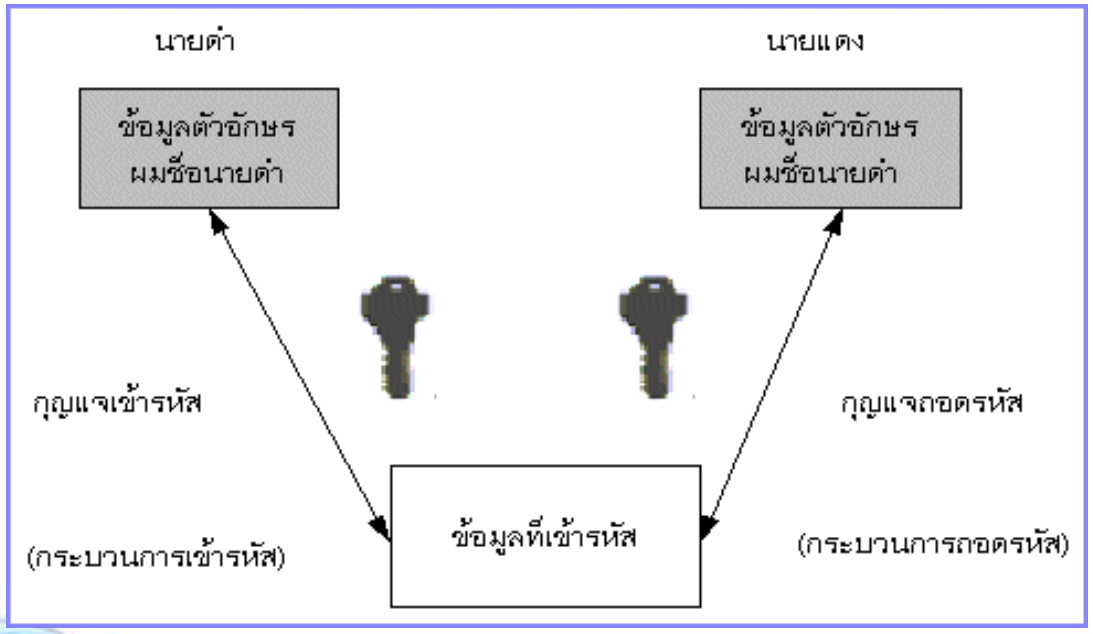
ทบทวน

- การเข้ารหัสข้อมูลมีจุดประสงค์เพื่อรักษาความลับของข้อมูล ข้อมูลนั้นจะถูกเปิดอ่านโดยบุคคลที่ได้รับอนุญาตเท่านั้น หลักการของการเข้ารหัสข้อมูลคือแปลงข้อมูล (encrypt) ไปอยู่ในรูปของข้อมูลที่ไม่สามารถอ่านได้โดยตรง ข้อมูลจะถูกถอดกลับด้วยกระบวนการถอดรหัส (decryption)



Symmetric Cryptography (Secret key)

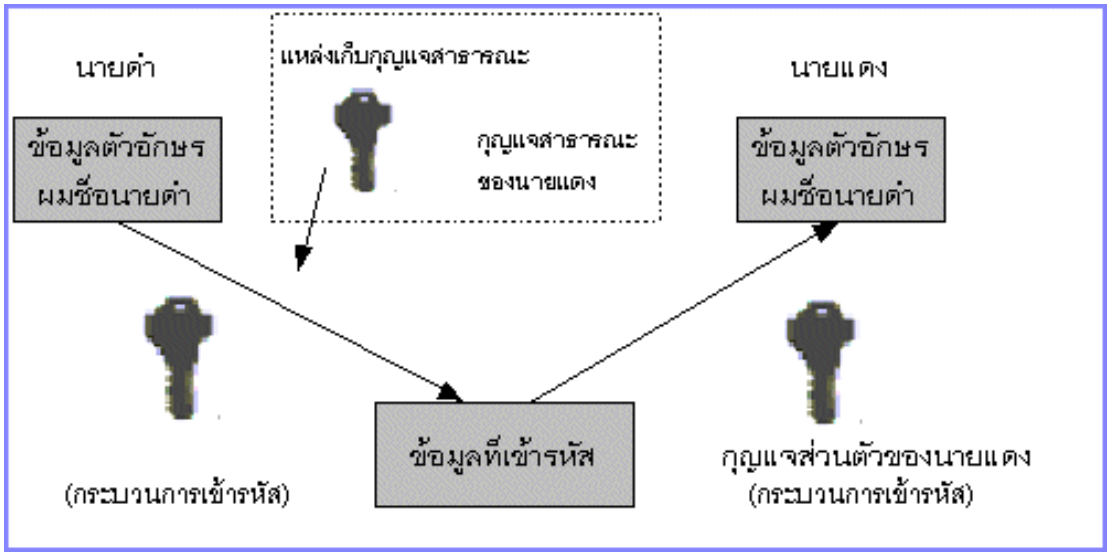
- บางทีอาจเรียกว่า Single-key algorithm หรือ one-key algorithm คือ การเข้ารหัสและถอดรหัสโดยใช้กุญแจรหัสตัวเดียวกัน คือ ผู้ส่งและผู้รับจะต้องมีกุญแจรหัสที่เหมือนกันเพื่อใช้ในการเข้ารหัสและถอดรหัส



ที่มา มหาวิทยาลัยเกษตรศาสตร์

Asymmetric Cryptography (Public key)

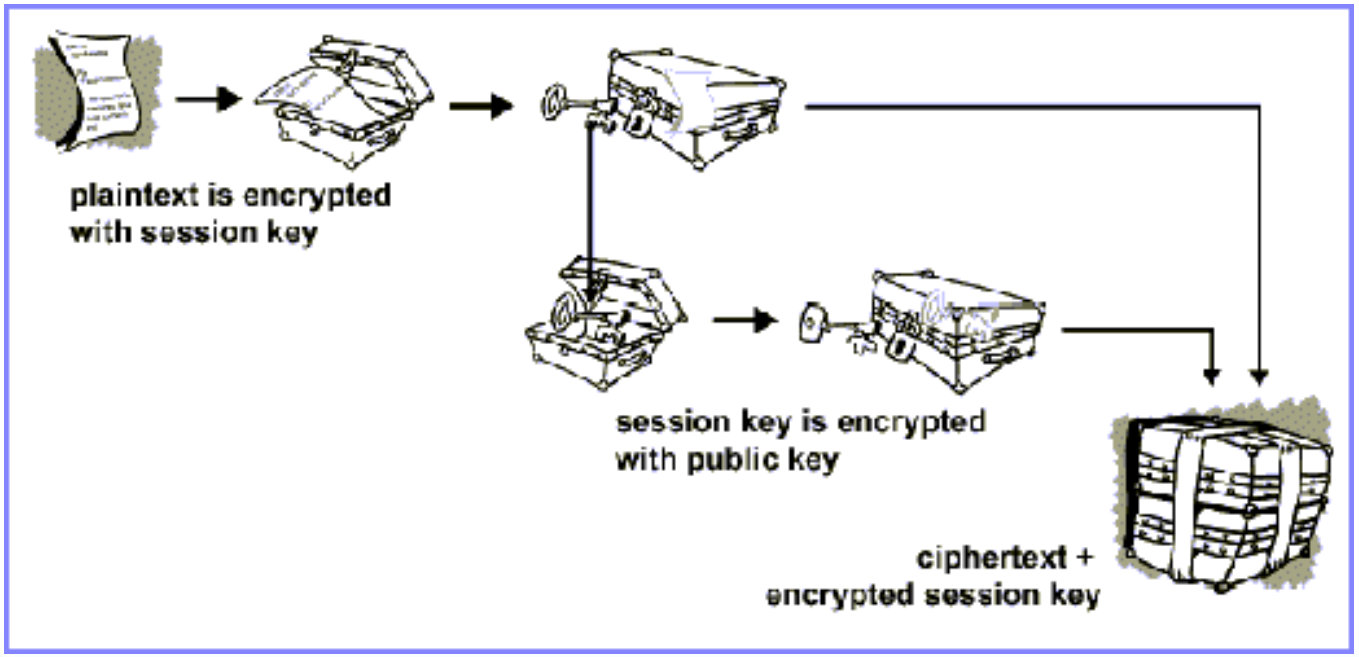
- การเข้ารหัสและถอดรหัสโดยใช้กุญแจรหัสคนละตัวกัน การส่งจะมีกุญแจรหัสตัวหนึ่งในการเข้ารหัส และผู้รับก็จะมีกุญแจรหัสอีกตัวหนึ่งเพื่อใช้ในการถอดรหัส ผู้ใช้รายหนึ่งๆจึงมีกุญแจรหัส 2 ค่าเสมอคือ **กุญแจสาธารณะ (public key)** และ **กุญแจส่วนตัว (private key)** ผู้ใช้จะประกาศให้ผู้อื่นทราบถึงกุญแจสาธารณะของตนเอง เพื่อให้นำไปใช้ในการเข้ารหัสและส่งข้อมูลที่เข้ารหัสแล้วมาให้ ข้อมูลที่เข้ารหัสดังกล่าวจะถูกถอดออกได้โดยกุญแจส่วนตัวเท่านั้น



ที่มา มหาวิทยาลัยเกษตรศาสตร์

Public Key Implementation

- ในทางปฏิบัติแล้วมักมีการใช้การเข้ารหัสทั้งสองระบบร่วมกันเช่นในระบบ PGP (Pretty Good Privacy) ซึ่งใช้ในการเข้ารหัส E-mail จะใช้วิธีสร้าง session key ซึ่งเป็นรหัสลับตามแบบ secret key) เมื่อข้อมูลถูกเข้ารหัสด้วย session key แล้ว จากนั้น session key จะถูกเข้ารหัสโดยใช้กุญแจสาธารณะของผู้รับ และถูกส่งไปกับข้อมูลที่เข้ารหัสแล้ว



ที่มา มหาวิทยาลัยเกษตรศาสตร์

Symmetric VS Asymmetrics

กุญแจ สมมาตร	กุญแจ อสมมาตร
ข้อดี • มีความรวดเร็วเพราะใช้การคำนวณที่น้อยกว่า • สามารถสร้างได้ง่ายโดยใช้ฮาร์ดแวร์	ข้อดี • การบริหารจัดการกุญแจทำได้ง่ายกว่า เพราะใช้กุญแจในการเข้ารหัส และถอดรหัสต่างกัน • สามารถระบุผู้ใช้โดยการใช้ร่วมกับลายมือชื่ออิเล็กทรอนิกส์
ข้อเสีย • การบริหารจัดการกุญแจทำได้ยาก เพราะกุญแจในการเข้ารหัสและถอด รหัสเหมือนกัน	ข้อเสีย • ใช้เวลาในการเข้าและถอดรหัสค่อนข้างนาน เพราะต้องใช้การคำนวณ

Asymmetric Key Encryption

RSA

- เป็นอัลกอริทึมในการเข้าหัสแบบอสมมาตร ถูกสร้างขึ้นมาเมื่อปี1978 โดย Ron Rivest, Adi Shamir และ Leonard Adleman ตั้งแต่คิดค้นมายังไม่มีใครสามารถเบรคอัลกอริทึมนี้ได้ และ RSA ได้ถูกนำมาใช้อย่างแพร่หลายในด้าน e-commerce
- RSA ยากที่จะทำการเบรกได้ เพราะ แม้จะทราบ Public Key ทราบค่า Message และทราบค่า Cipher ก็ตาม แต่ก็ยากที่จะทำการคำนวณย้อนกลับเพื่อหาค่าของ Private Key ได้

Asymmetric Key Encryption

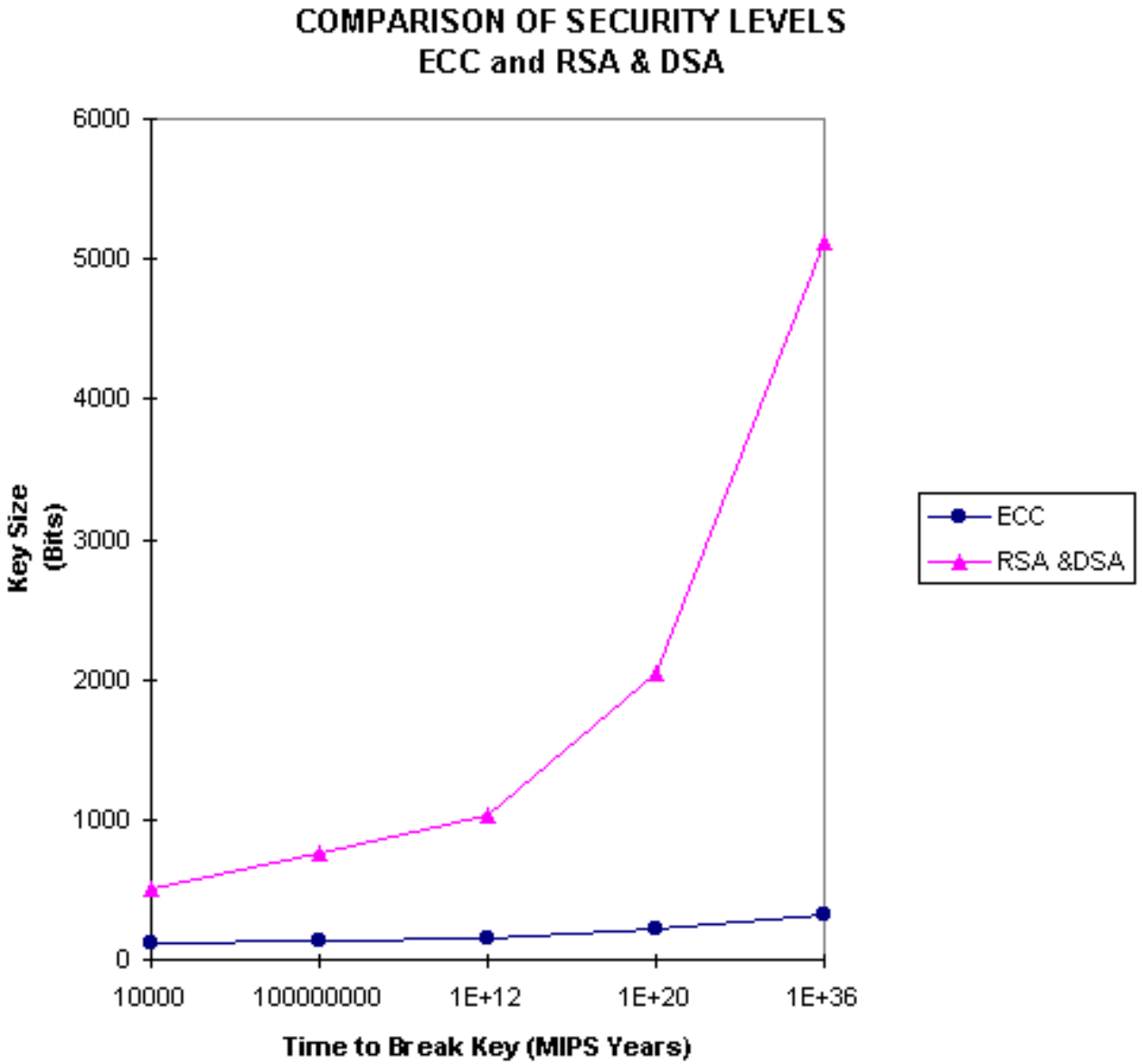
ECC

- ECC ย่อมาจาก Elliptic Curves Cryptography ได้รับการนำเสนอโดย Neal Koblitz และ Victor S. Miller ในปี 1985 โดยอัลกอริทึมการเข้ารหัส ECC นี้ได้รับการพัฒนาจากสมการของเส้นโค้งของวงรี

$$y^2= x^3 + ax + b$$

จุดเด่น ECC

- ECC มีข้อดีที่เหนือกว่า RSA คือ จะใช้คีย์ที่สั้นกว่าแต่สามารถให้ความปลอดภัยเท่ากับ RSA ได้ หรือ หากใช้คีย์ที่ยาวเท่ากับคีย์ของ RSA จะมีความปลอดภัยสูงกว่า
- เนื่องจาก ECC ใช้ Key ที่มีขนาดเล็กกว่า RSA มาก และมีความสามารถในการคำนวณที่รวดเร็ว ใช้พลังงานต่ำและใช้หน่วยความจำน้อย ดังนั้น ECC จึงเหมาะสำหรับการใช้งานในอุปกรณ์เคลื่อนที่ขนาดเล็ก เช่น โทรศัพท์มือถือ Pocket PC และ PDA เป็นต้น



เปรียบเทียบเวลาที่ใช้ในการแกะรหัสระหว่าง RSA และ ECC

HASH

การเข้ารหัสแบบ Hash (Cryptographic hash) หมายถึง การแปลงรูปแบบของข้อมูลที่รับเข้ามาให้เป็นข้อมูลที่ถูกย่อย (Message Digest) ไม่ว่าข้อมูลต้นฉบับจะมีขนาดเล็กหรือใหญ่เท่าใดก็ตามก็จะถูกย่อยให้อยู่ในรูปแบบที่มีขนาดคงที่ ดังนั้นจึงไม่สามารถทำกระบวนการย้อนกลับเพื่อให้กลายเป็นข้อมูลต้นฉบับได้ จะทำได้เพียงแค่ตรวจสอบว่าข้อมูลที่ให้มาแต่ละครั้งเหมือนกันหรือไม่

ฟังก์ชั่น Hash ที่สำคัญ ๆ ได้แก่ MD4, MD5, SHA-1 และ SHA-2

ตัวอย่างการใช้งาน

- **ในระบบจัดการฐานข้อมูลอย่างชนิด จะทำการย่อยรหัสผ่านด้วยฟังก์ชั่น Hash เช่นใช้ MD5 ย่อยรหัสผ่าน abc123 ได้เป็น**

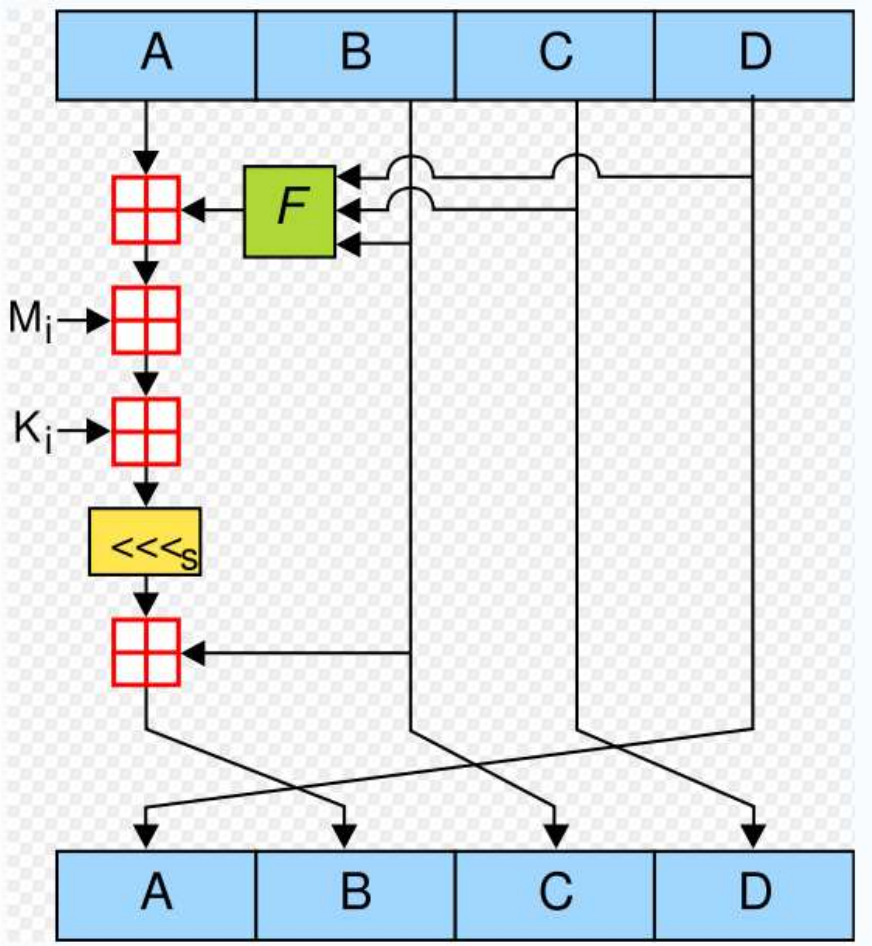
  [e99a18c428cb38d5f260853678922e03]

  แล้วจึงเก็บค่าแฮชนั้นลงใน Database จะทำให้การเปิดดูรหัสผ่านใน Database โดยตรง ไม่พบรหัสผ่าน abc123 แต่จะพบเพียงค่าแฮช (e99a18c428cb38d5f260853678922e03)

- ไฟล์รหัสผ่านของ Linux (/etc/shadow) ก็แฮชรหัสผ่านด้วย MD5 เช่นกัน นอกจากนั้นก็ยังพบเห็นการประยุกต์ใช้การแฮชรหัสผ่านใน Web Application ต่าง ๆ เช่น Moodle และ Mambo

MD5

อัลกอริทึม MD5 คิดค้นโดย Ron Rivest ซึ่งเป็น 1 ใน 3 คนที่คิดค้น RSA

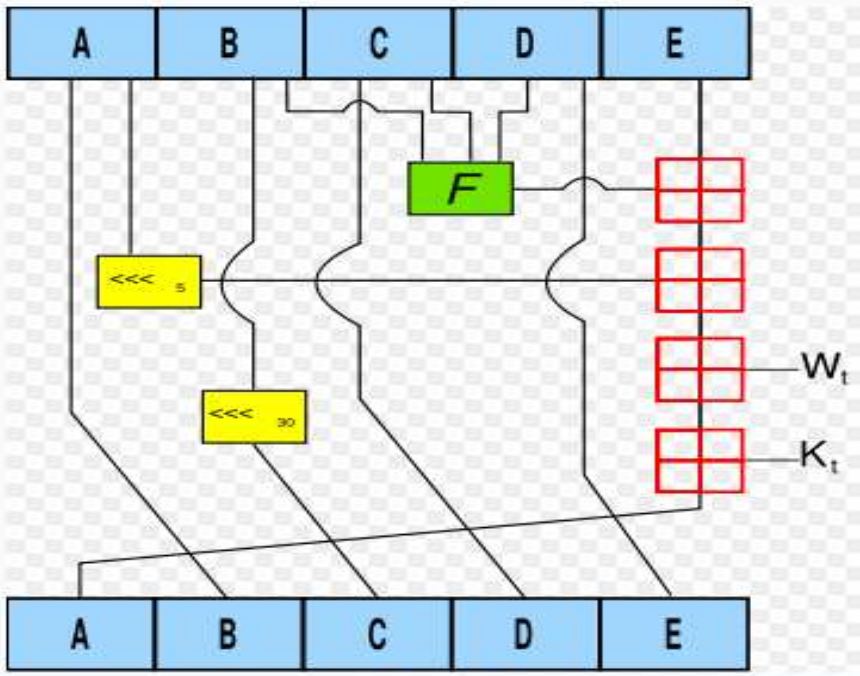


เบรก MD5

ถึงแม้ MD5 จะได้รับความนิยมอย่างมาก และได้มีการนำมาใช้แพร่หลายเช่นนำมาใช้สร้าง Digital Signature ในระบบ e-commerce อย่างไรก็ตาม MD5 ก็ถูกเบรคได้โดยนักคณิตศาสตร์หญิงชาวจีน (Professor Dr. Xiaoyun Wang) ในปี 2004 โดยใช้เครื่องซูเปอร์คอมพิวเตอร์ IBM P690 และใช้เวลาแคร็ก 1 ชั่วโมง ก็สามารถเบรคได้ หลังจากนั้นก็มีคนอ้างว่าสามารถใช้เครื่องคอมพิวเตอร์ Notebook ความเร็ว 1.6 GHz เบรค MD5 ได้ภายในเวลา 8 ชั่วโมง

SHA

- SHA0 และ SHA1 ได้ถูกพัฒนาให้มีความแข็งแรงกว่า MD5 โดยได้พัฒนาจาก MD5 เดิมให้ output มีความเป็น Random สูงกว่า และมี Collision น้อยกว่าเพื่อลดโอกาสในการถูกแคร็กได้
- อัลกอริทึมของ SHA1



เบรก SHA

- **SHA0 และ SHA1 ก็ถูกเบรคได้โดยนักคณิตศาสตร์หญิงชาวจีน (Professor Dr. Xiaoyun Wang) คนเดียวกันกับที่เคยเบรค MD5 ได้ ดังนั้นปัจจุบันนี้ความหวังจึงอยู่ที่ SHA2 ซึ่งยังไม่มีใครเบรคได้ อัลกอริทึมของ SHA2**

Public Key Cryptography

- การเข้ารหัสเดิมใช้ Key เดียวในการเข้าและถอดรหัสเรียกว่า Secret Key
- Secret Key ต้องเก็บเป็นความลับให้มากที่สุด
- ส่วน public key cryptography จะมีกุญแจสองดอก คือ
  - Private key ซึ่งจะเก็บเป็นความลับมีเจ้าของคนเดียวเท่านั้นที่รู้
  - Public key ซึ่งไม่จำเป็นต้องเก็บเป็นความลับ
- ทั้งสองตัวจะใช้งานต่างกันคือ ถ้าใช้กุญแจอันหนึ่งเข้ารหัส จะต้องใช้กุญแจอีกตัวหนึ่งที่เข้าคู่กันในการถอดรหัส
- การที่มีกุญแจสองแบบทำให้ public key cryptography ได้เปรียบ secret key cryptography ตรงที่ผู้รับกับผู้ส่งใช้กุญแจคนละตัวกัน

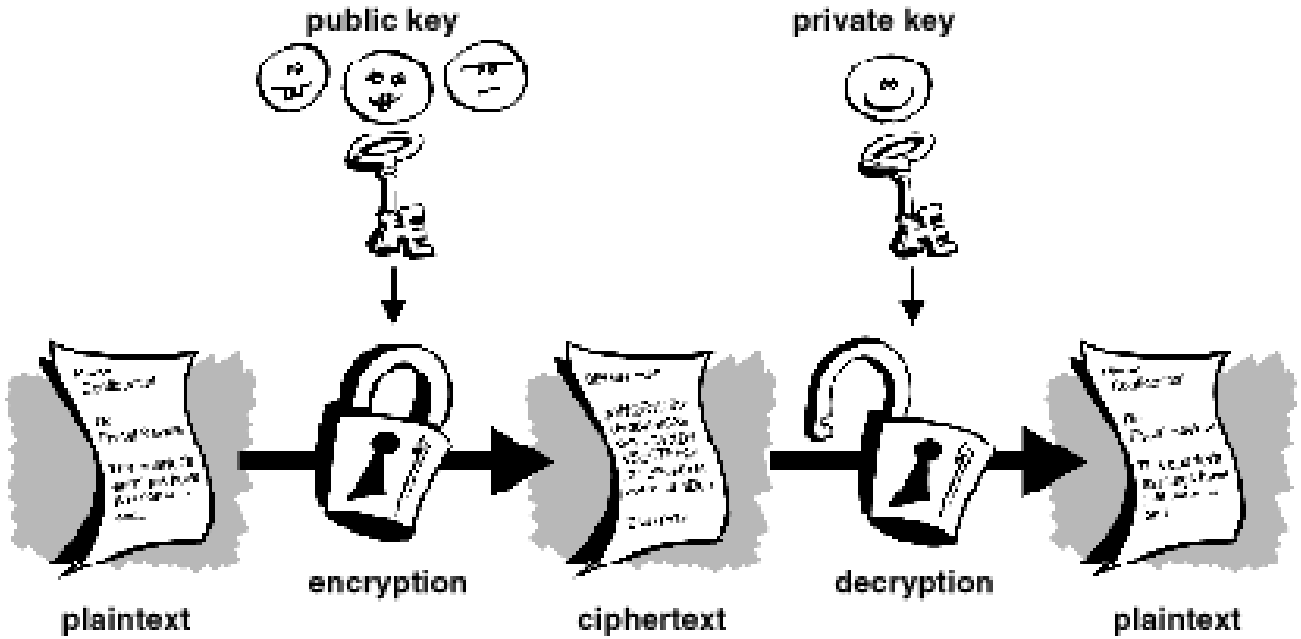
แนวทาง Public Key

- สมมติว่ามี public key A กับ private key B เป็นกุญแจที่เข้าคู่กัน
- ถ้านำ A เข้ารหัส จะมีแต่ B เท่านั้นที่ถอดรหัสนั้นออก และในทางกลับกันถ้าใช้ B เป็นตัวเข้ารหัสก็จะมีแต่ A เท่านั้นที่จะถอดรหัสได้

ที่มา Public Key

- คนแรกที่คิดเรื่องนี่คือ วิธฟีลด์ ดิฟฟี่ (Whitfield Diffie) และ มาร์ติน เฮลแมน (Martin Hellman)
- Public key cryptography เกิดจากหลักคณิตศาสตร์ที่เรียกว่า ฟังก์ชันทางเดียว (one-way function)
- กลุ่มของฟังก์ชันทางเดียวส่วนหนึ่งมีความเกี่ยวข้องกับเลขจำนวนเฉพาะ (prime number) เลขที่หารได้เฉพาะ 1 และตัวมันเอง
- เกิด RSA cryptosystem ซึ่งตีพิมพ์โดย รอน ริเวสต์ (Ron Rivest), อาดิ ชาร์เมียร์ (Adi Shamir), และ เลียวนาร์ด เอเดิลแมน (Leonard Adleman)

การทำงาน Public key



การใช้งาน Public key

- Public key cryptography นำมาประยุกต์ใช้ค่อนข้างกว้าง จนอาจจะแก้ปัญหาของระบบความปลอดภัยได้ครบทั้ง 4 ประการ คือ
  - **Authentication**
  - **Integrity**
  - **Confidentialty**
  - **Non-repudiation**
- สามารถนำมาประยุกต์ใช้เป็น digital signature ได้

การใช้งาน Public Key(ต่อ)

Secure Shell (SSH)

- SSH เป็น protocol ที่แก้ปัญหาเรื่องความไม่ปลอดภัยในการ remote login เข้าใช้งาน service ต่างๆ

SSL (Secure Socket Layer)

- SSL เริ่มมาจากบริษัท Netscape ที่พัฒนา browser
- เป็น security protocol ก็คือเป็นตัวให้บริการความปลอดภัยในการสื่อสารข้อมูล
- SSL เป็น protocol ที่ทำงานแทรกอยู่ระหว่าง application กับ transport layer (TCP) ปัจจุบันมี service มากมายที่ทำงานกับ SSL เช่น http, ftp, telnet, pop3, smtp หรือแม้แต่ VPN

การใช้งาน Public Key(ต่อ)

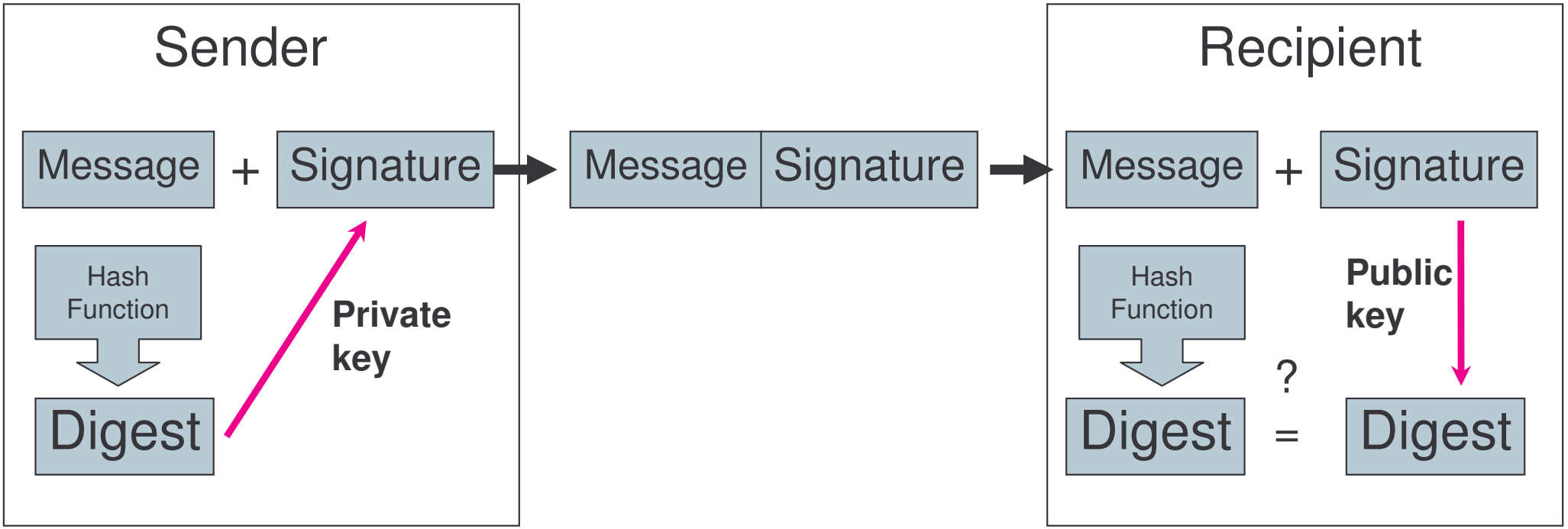
PGP (Pretty-Good Privacy)

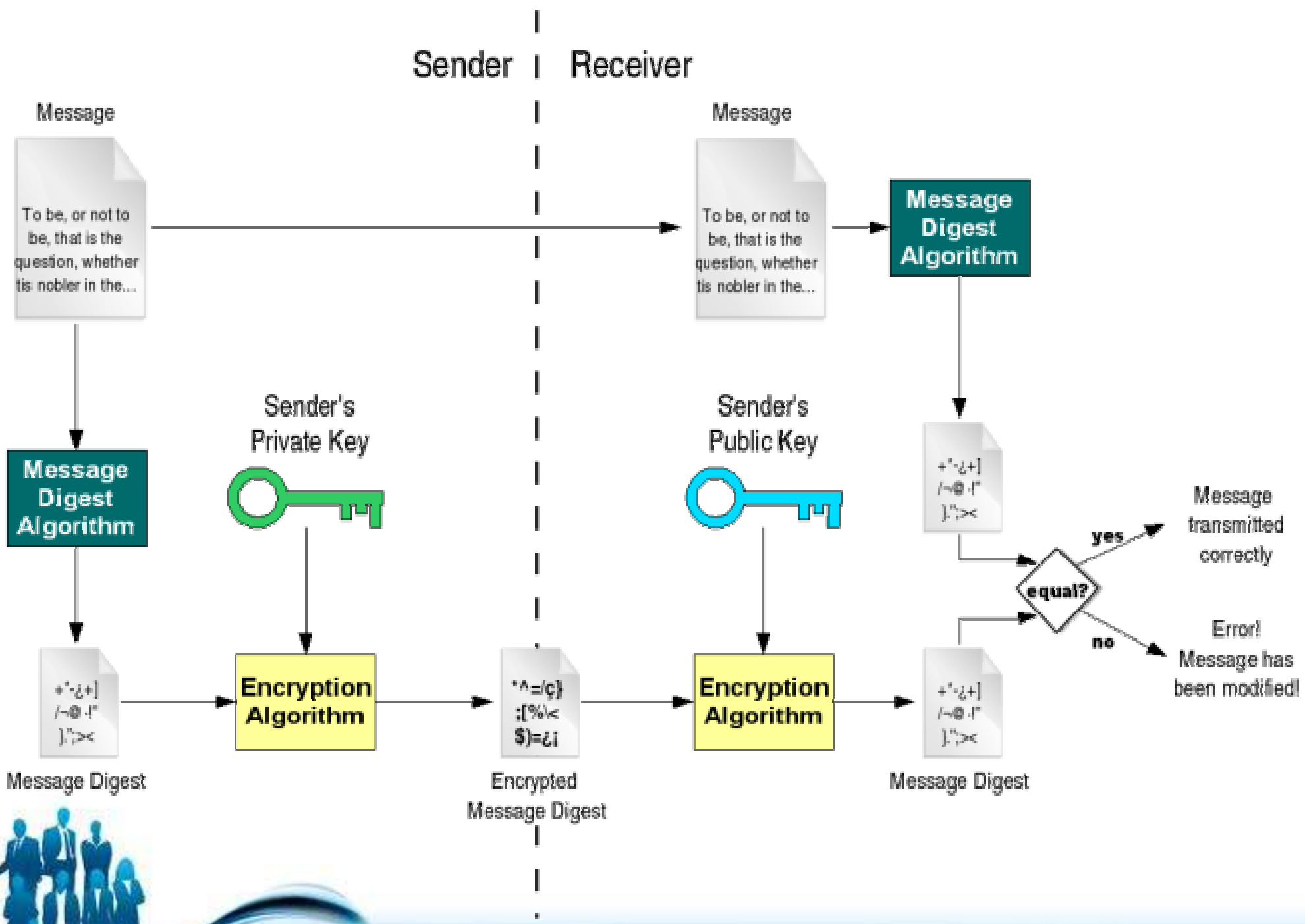
- เป็น public-domain program ใช้ IDEA (International Data Encryption Algorithm)
- IDE เป็น algorithm สำหรับ encryption ใช้ RSA สำหรับจัดการ key และใช้ MD5 (Message Digest v.5) สำหรับสร้าง hash
- วิธีการสร้าง key ของ PGP จะใช้ latency ในการพิมพ์ keyboard มาเป็นตัวหาเลขสุ่ม แล้วจึงเอาเลขสุ่มนี้ไปหา key อีกที

Digital Signature

- การทำธุรกรรมอิเล็คทรอนิกส์จำเป็นที่จะต้องมีการยืนยันเอกสารหรือข้อมูลที่ส่งว่าถูกส่งมาจากผู้ส่งจริง เพื่อการป้องกันการปฏิเสธความรับผิดชอบ (Non-repudiation) และเป็นการพิสูจน์ทราบตัวตน (Authentication)
- ในการส่งข้อมูลที่ "ไม่เป็นความลับ" และเป็นข้อมูลมีขนาดใหญ่ หากต้องการที่จะยืนยันผู้ส่งด้วยวิธีการข้างต้นเราจะต้องทำการเข้ารหัสข้อมูลทั้งหมดด้วย Private Key ของผู้ส่ง (เพื่อเป็นการยืนยันตัวตนผู้ส่ง) นั้น จะมีข้อเสียคือจะต้องมีการเข้ารหัสข้อมูลขนาดใหญ่ทั้งหมด และผู้รับจะต้องทำการถอดรหัสข้อมูลทั้งหมดเช่นกัน ซึ่งทำให้เปลือง CPU และเปลืองเวลาในการประมวลผล แต่ก็สามารถทำให้ CPU เปลืองเวลาน้อยลงได้โดยใช้ฟังก์ชั่น Hash

หลักการทำงานของลายเซ็นดิจิตอล




Sender
Receiver
Message
To be, or not to be, that is the question, whether tis nobler in the...
Message
To be, or not to be, that is the question, whether tis nobler in the...
Message Digest Algorithm
Sender's Private Key
Sender's Public Key
Message Digest Algorithm
Encryption Algorithm
Encryption Algorithm
equal?
yes
no
Message transmitted correctly
Error! Message has been modified!
Message Digest
Encrypted Message Digest
Message Digest

ประโยชน์ของลายเซ็นดิจิตอล

- ใช้เพื่อเพิ่มความปลอดภัย ช่วยยืนยันตัวจดหมายว่าส่งมาจากผู้ส่งนั้นจริง
- ใช้หลักการในการเปลี่ยนข้อความทั้งหมดให้เหลือเพียงข้อความสั้น ๆ เรียกว่า "Message digest" ซึ่งจะถูกสร้างขึ้นด้วยกระบวนการเข้ารหัสยอดนิยมที่เรียกว่า One-way hash function
- จะใช้ message digest นี้ในการเข้ารหัสเพื่อเป็นลายเซ็นดิจิตอล(Digital Signature) โดยจะแจก Public key ไปยังผู้ที่ต้องการติดต่อ

End.