SHARE

SHARE in Pittsburgh 2014 August 3-8

MAINFRAME CRYPTO

Cryptographic Basics

Greg Boyd

www.mainframecrypto.com

Copyrights and Trademarks

- Presentation based on material copyrighted by IBM, and developed by myself, as well as many others that I worked with over the past 10 years

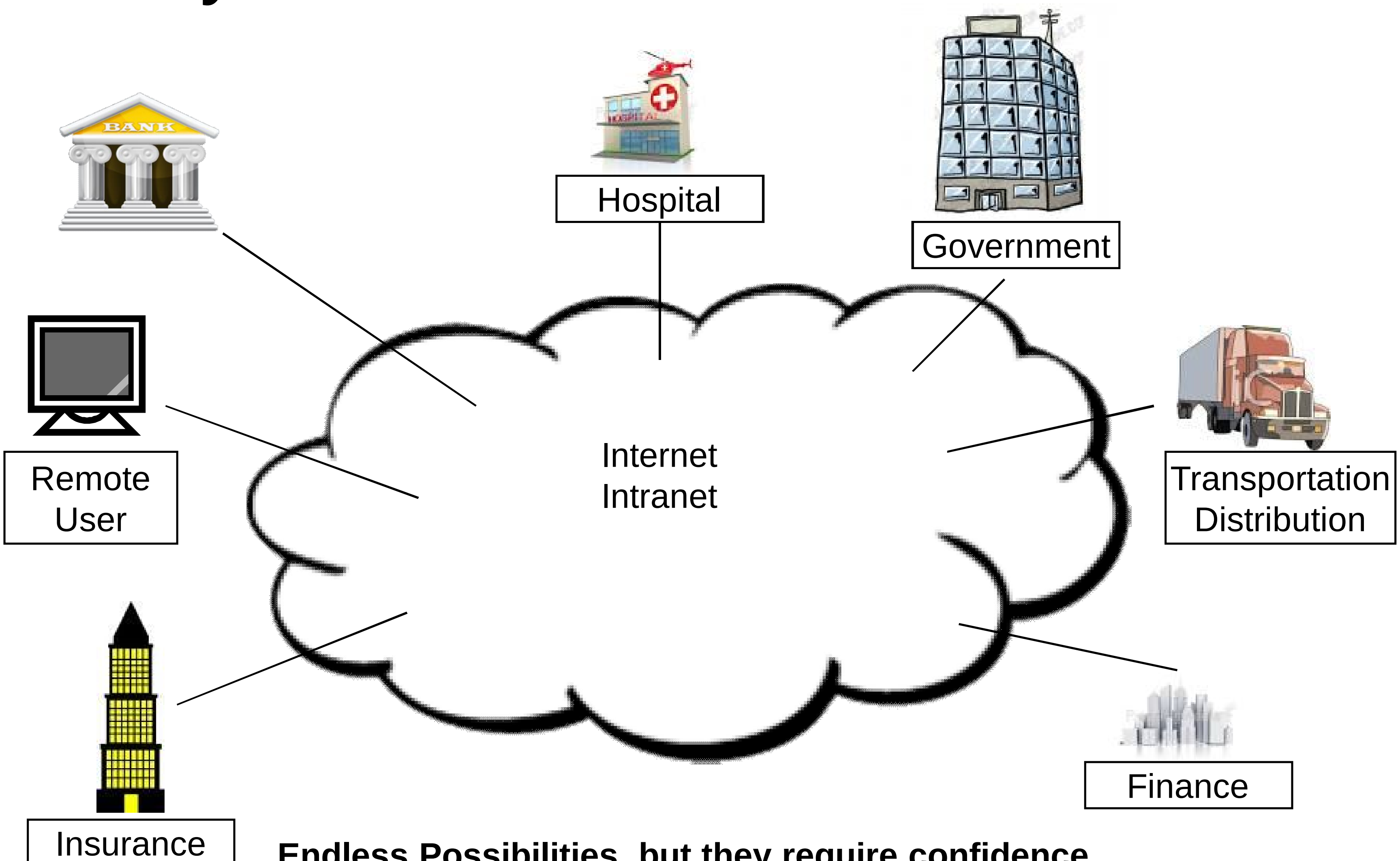
QR Code

- Share #15659

What we're going to cover

- Some context
- Cryptographic Functions
  - Symmetric Algorithms
  - Asymmetric Algorithms
  - Hashing
  - Digital Signatures and Digital Certificates
  - Financial

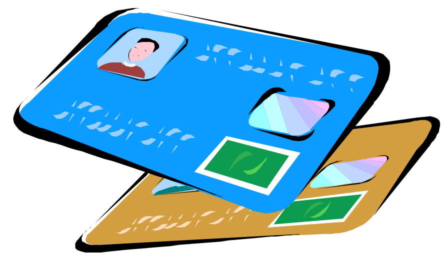
Today's Business Environment

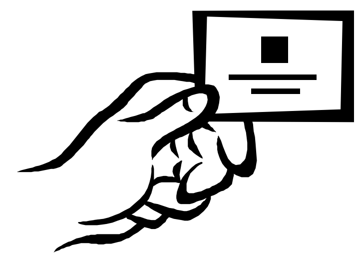


Endless Possibilities, but they require confidence

Business Requirements

- Trust
- Confidentiality
  - Trade Secrets
  - Business transactions
- Privacy
  - Personal Information
- Accountability/ Auditability





Industry Pressures: Addressing Regulations

Privacy Regulations

1999
Gramm-Leach-Bliley Act (GLBA)
US

2000
PIPEDA
Canada

2000
COPPA and CIPA
US

2003
California Individual Privacy (SB1386)
California

2008
PCI DSS v1.2
Industry

1987
Computer Security Act US

1995
EU Data Protection Directive
EU

1996
HIPAA
US

1997
Personal Health Information Act
Canada

1998
Data Protection Act
UK

Financial Integrity and Solvency Regulations

2005
8th Company Law Directive (Euro SOX)
EU

2006
Financial Instruments and Exchange Law (J-SOX)
Japan

2012
Solvency II
EU

2002
Sarbanes-Oxley Act
US

2002
Corporate Law Economic Reform Program
Australia

2003
Basel II
EU

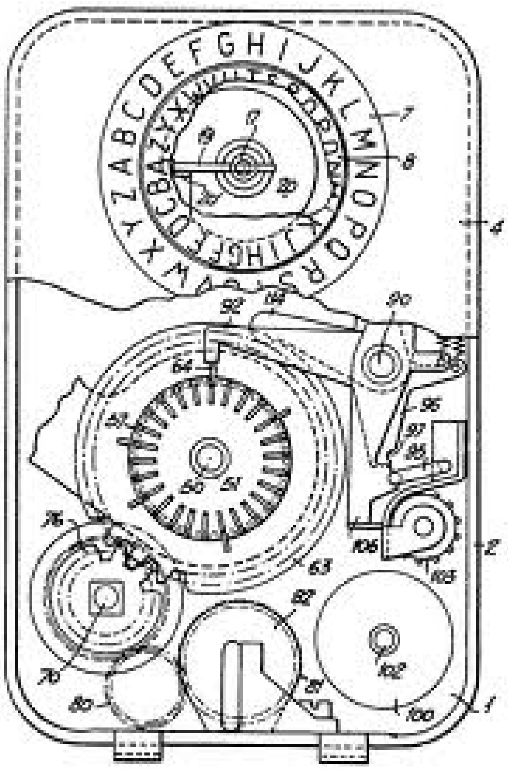
Other Regulations

2006
Federal Rules of Evidence
US

2001
USA PATRIOT Act
US

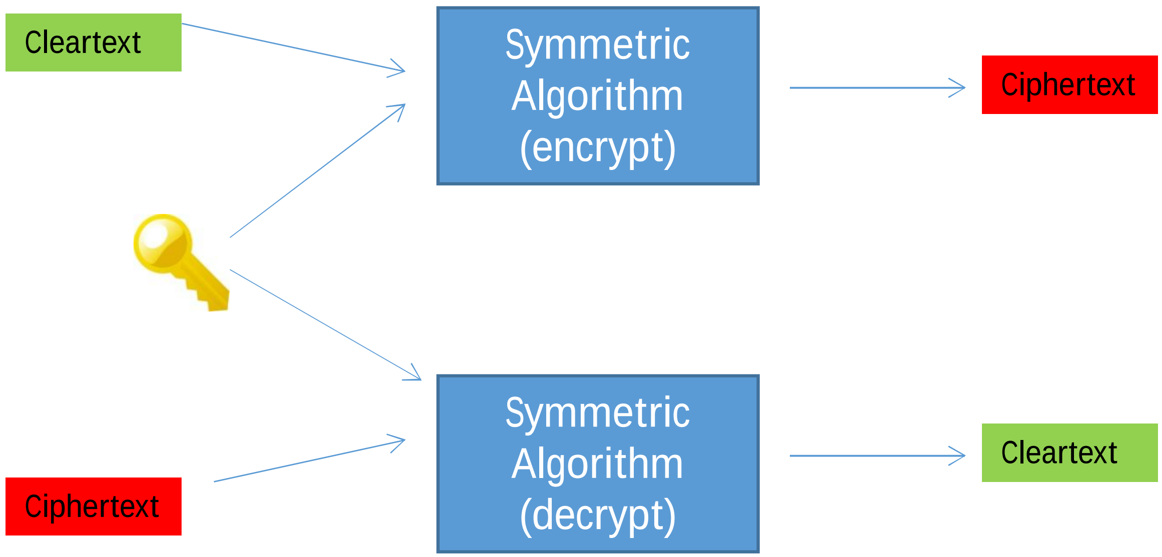
Cryptography

- Secrecy
- Integrity
- Financial Authentication
- Key Protection



Secrecy Algorithms - Symmetric

- Symmetric - One key shared by both parties

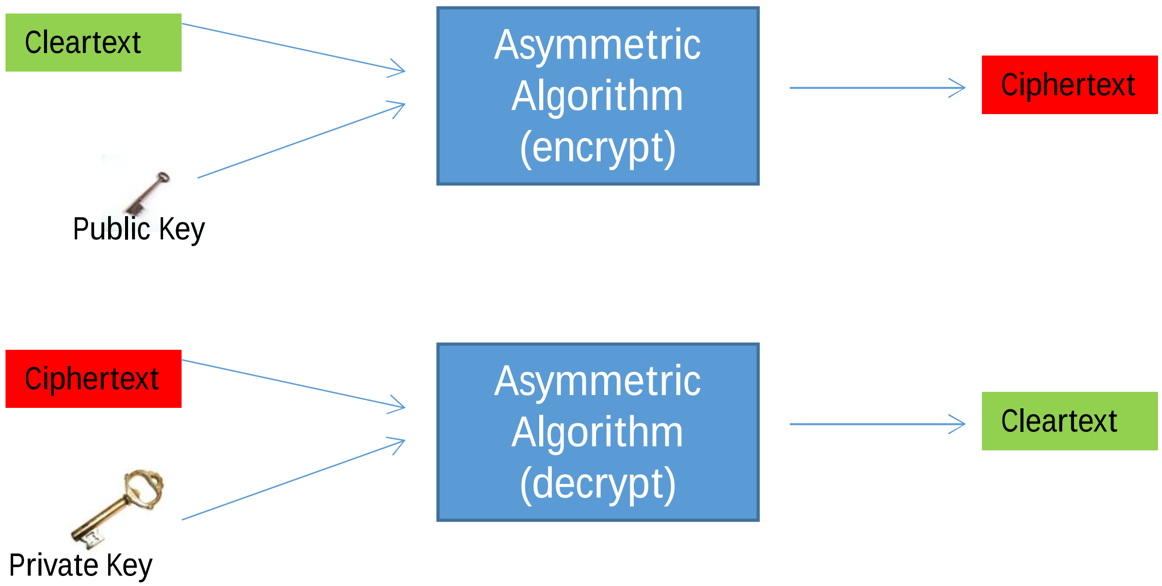


+ Speed (compared to asymmetric)

- Key Distribution

Secrecy Algorithms - Asymmetric

- Asymmetric – two different, but mathematically related keys (public and private)



\+ Key Distribution

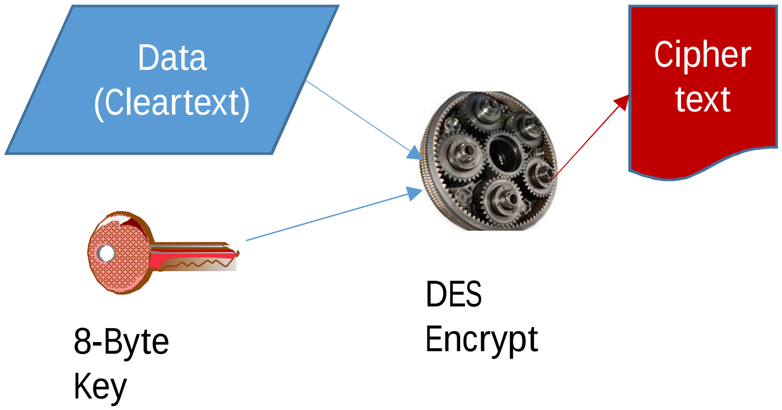
\- Speed (compared to symmetric)

Algorithms

- Symmetric
  - DES/TDES*
  - AES*
  - Blowfish / Twofish
  - Serpent
  - IDEA
  - RC2 / RC4
  - Skipjack
  - ....
- Asymmetric
  - RSA*
  - Diffie-Hellman*
  - ECC*

*Supported on IBM Hardware

DES Algorithm - Encrypt



DES

One round

- One of the 16 iteration steps
- $L \equiv$ left half of 64-bit message
  $R \equiv$ left half of 64-bit message
  - Each is 32 bits
- Key is 56 bits
  - $K_1 - K_{16}$ are 16 permutations on the master key
- Use lookup tables for the permutations and substitutions

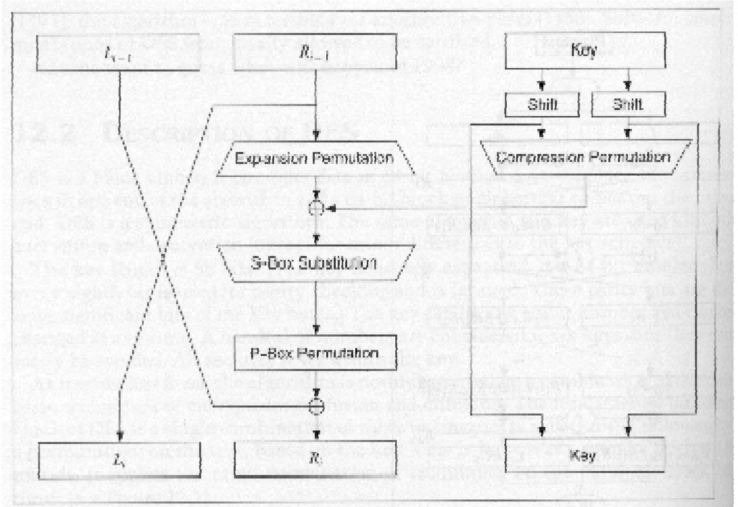
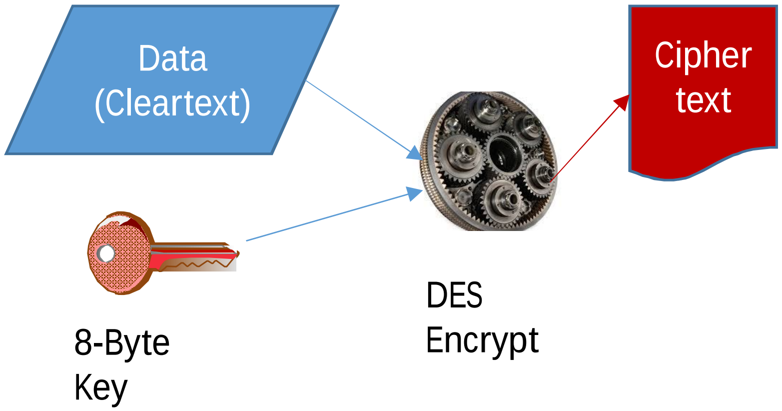


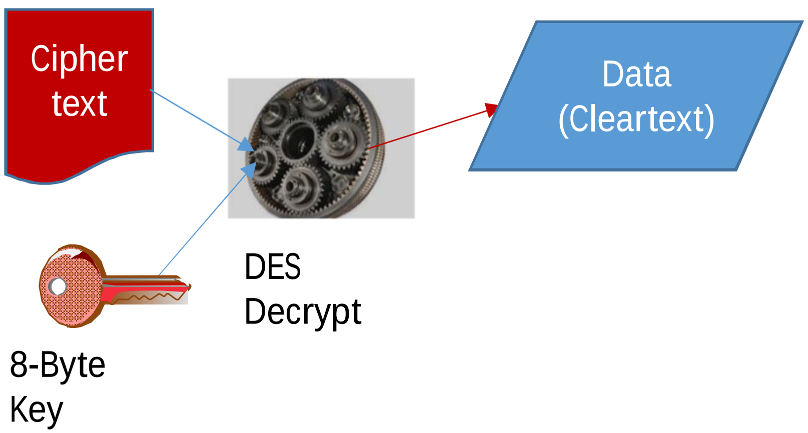
Figure 12.2 One round of DES.

C. Diorio, Lecture 16: DES primer 6

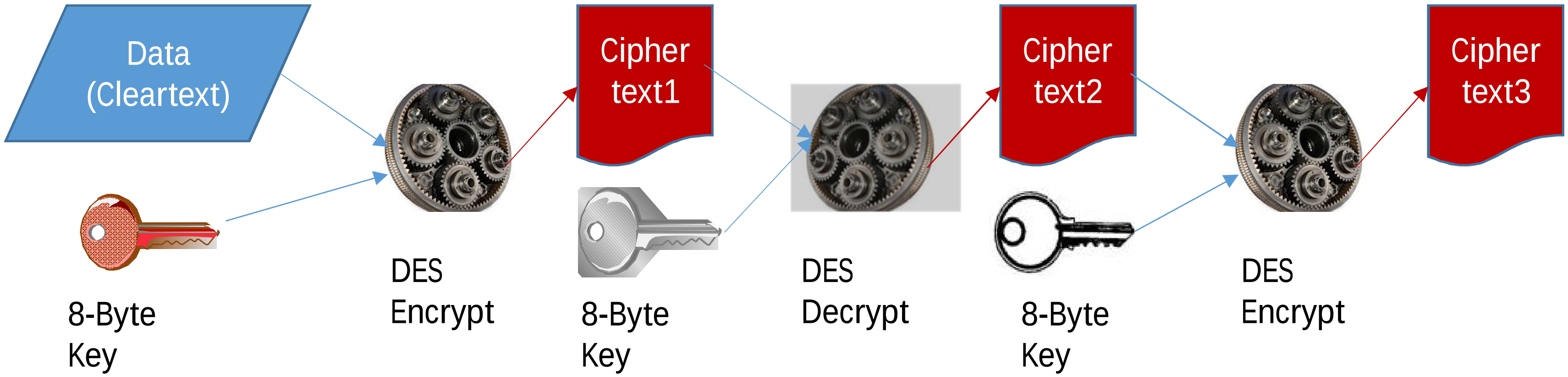
http://courses.cs.washington.edu/courses/cse467/99au/admin/Slides/Week6Lecture1/sld006.htm

DES Algorithm – Encrypt & Decrypt

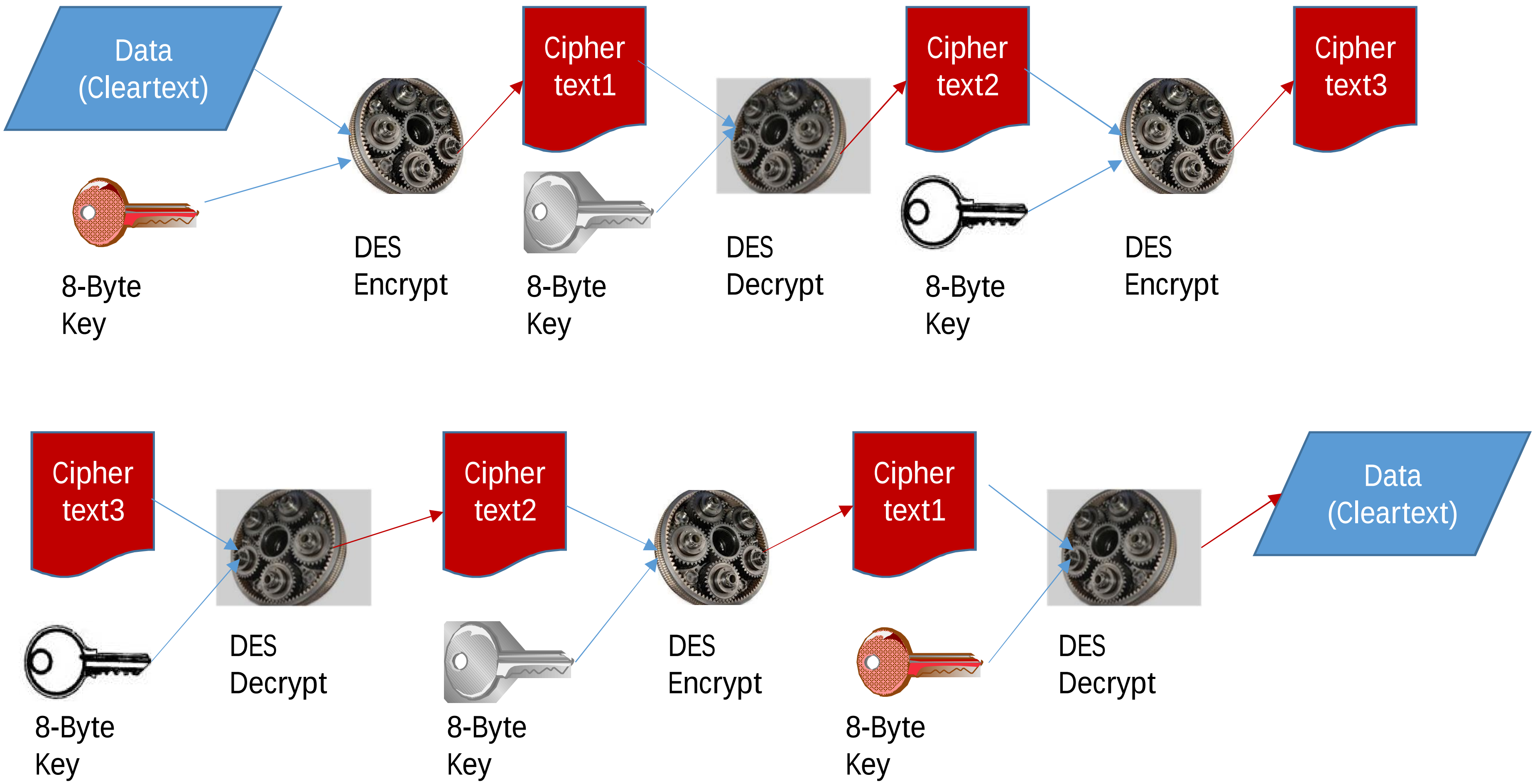




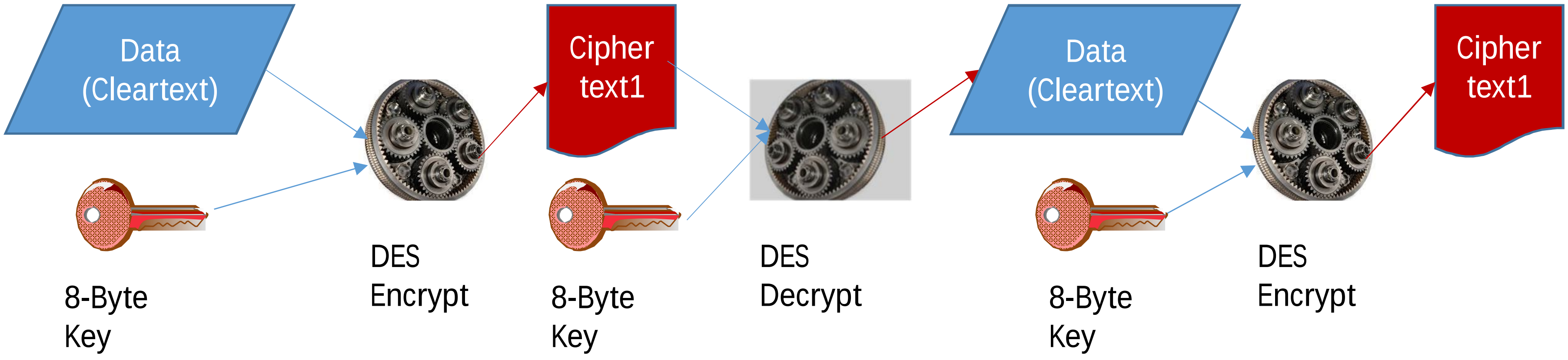
TDES Algorithm - Enrcrypt



TDES Algorithm – Encrypt & Decrypt

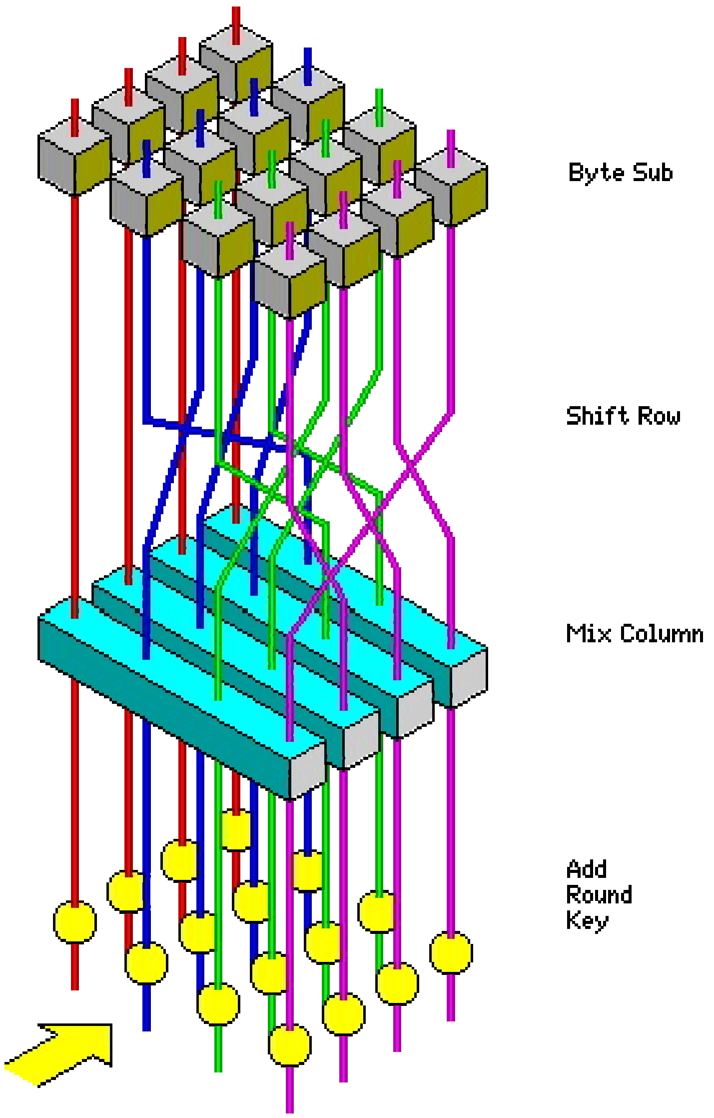


TDES Algorithm – DES Compatibility



AES Algorithm

- Multiple Rounds
  - 128-bit keys, 10 cycles
  - 192-bit keys, 12 cycles
  - 256-bit keys, 14 cycles



Asymmetric Algorithms

- RSA
- Diffie-Hellman
- Elliptic Curve

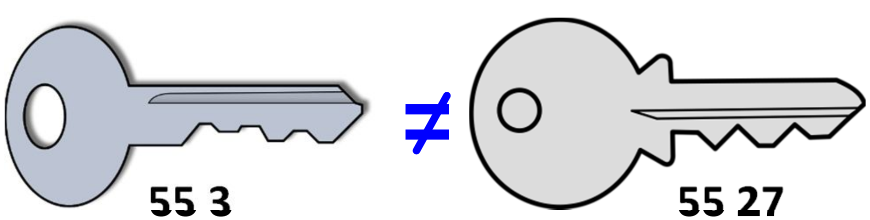
Generating RSA Keys

- RSA Keys consists of two parts, a modulus (N) and an exponent (E for the public key; D for the private key)
  - Public Key => N E
  - Private Key => N D
- The modulus is calculated by multiplying two prime numbers (P & Q) together
  - P = 5 Q = 11 (in reality, these should be very large prime numbers, 100s of digits long)
  - N = P x Q => 5 x 11 = 55
- Next, select an odd number, E, that will be the exponent for the public key
  - Good values include 3 or 65537 (64K+1)

    Public Key => N E => **55 3**
- Finally, calculate the exponent for the private key, D, where

  1 = (D * E) MOD ((P-1)(Q-1))
  - In our example, solve for 1 = (D * 3) MOD 40 => D = 27!

    Private Key => N D => **55 27**



Encipher the Message ‘GPB’

Public Key (N E) => **55 3**

Private Key (N D) => **55 27**

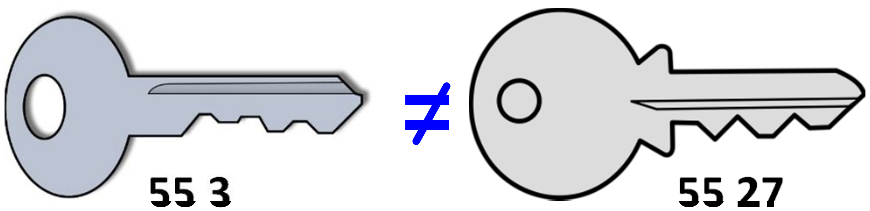
Convert characters to numeric (a=1, b=2, c=3, etc.)

‘G’ = 7; ‘P’ = 16; ‘B’ = 2;

ciphertext = (cleartext**E) Mod N

- For ‘G’ (7**3) MOD 55 => 343 MOD 55 = 13
- For ‘P’ (16**3) MOD 55 => 4096 MOD 55 = 26
- For ‘B’ (2**3) MOD 55 => 8 MOD 55 = 8

Ciphertext is 13 26 8



Decipher the message 13 26 8

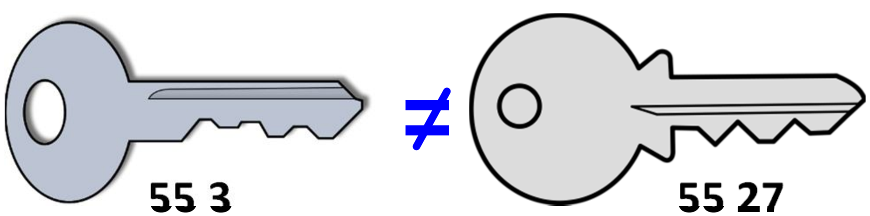
Public Key (N E) => **55 3**

Private Key (N D) => **55 27**

Cleartext = (ciphertext**D) MOD N

- For 13 13**27 MOD 55 = 7
  (13**27 = 1192533292512492016559195008117)
- For 26 26**27 MOD 55 = 16
  (26**27 = 1.6005910908538609008071353149841e+38)
- For 8 8**27 mod 55 = 2
  (8**27 = 2417851639229258349412352)

- My decrypted message is 7 16 2 => “G” “P” “B”



ECC Algorithm

Effective Key Size		
Symmetric	RSA	ECC
80	1024	163
112	2048	224
128	3072	256
192	7680	384
256	15360	512
From NIST SP 800-57 Part 1 (Table 2) at www.nist.gov		

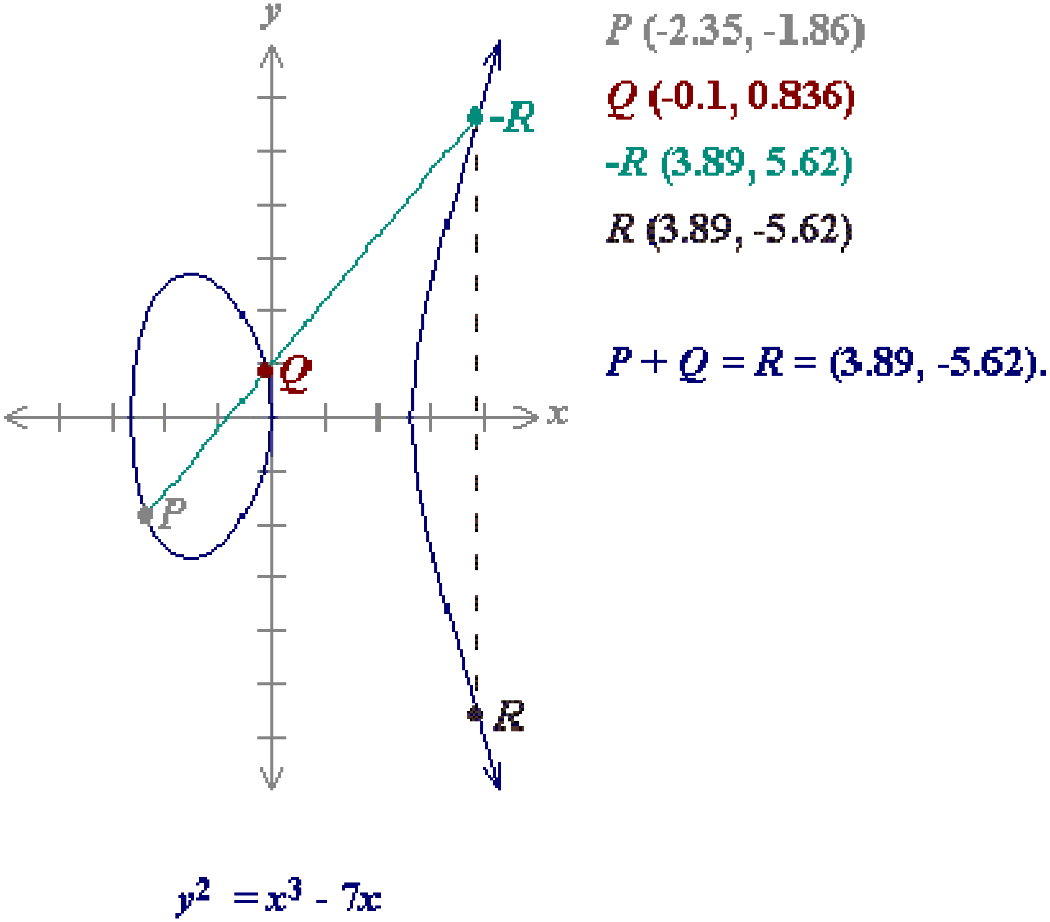
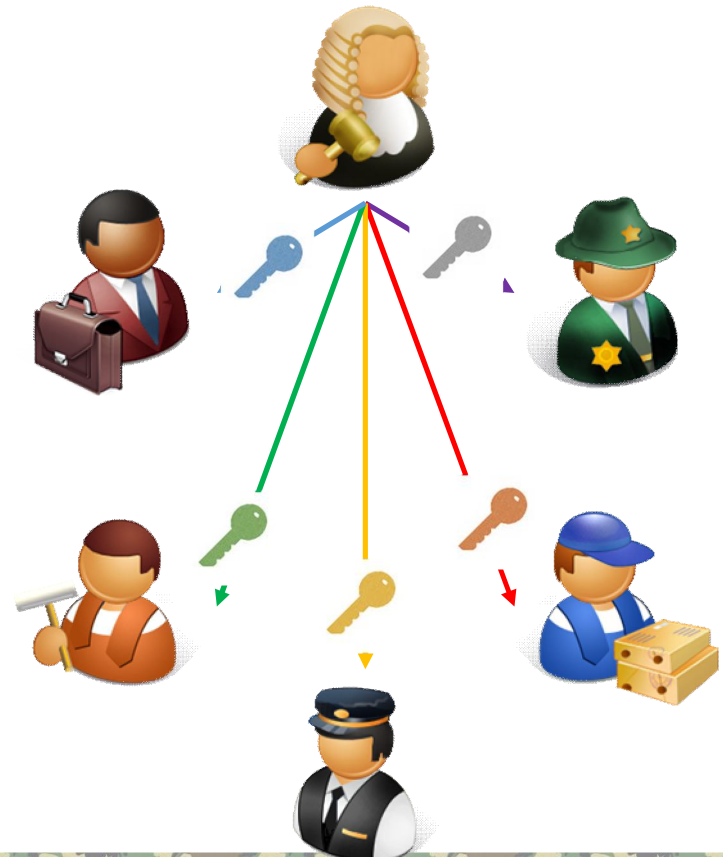
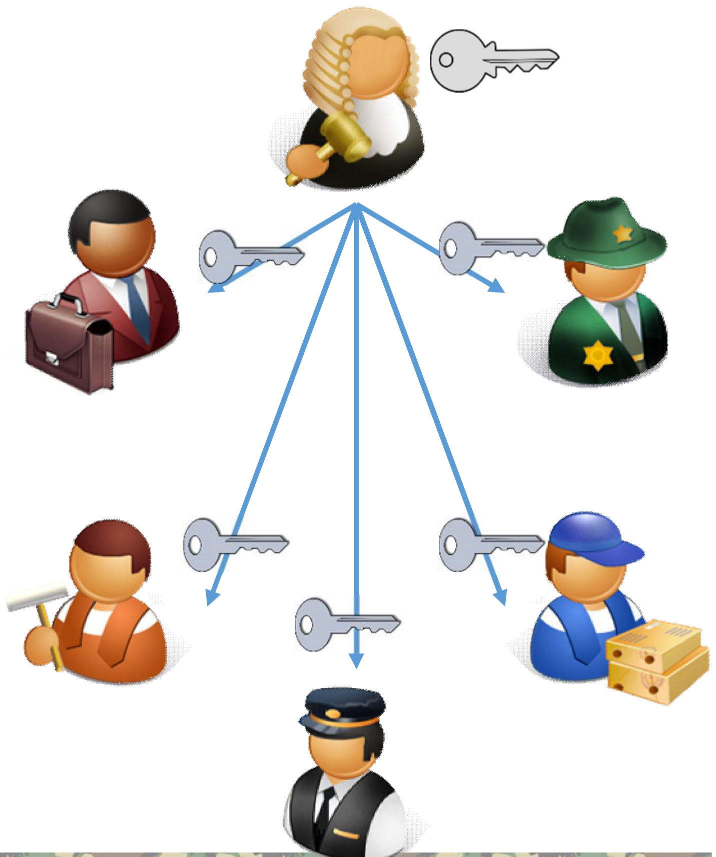


Image from crypto.stackexchange.com

Complimentary algorithms

- Symmetric
  - Fast for large messages, but key distribution is a problem
- Asymmetric
  - Expensive, so only use for small messages, but easy to distribute keys



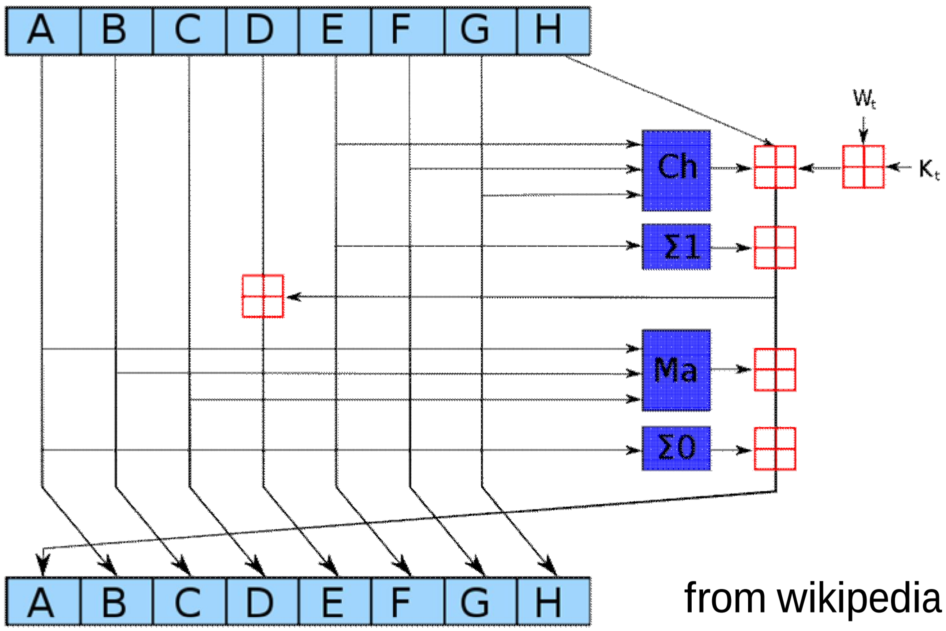


Hashing

- One iteration in a SHA-2 family compression function. The blue components perform the following operations:

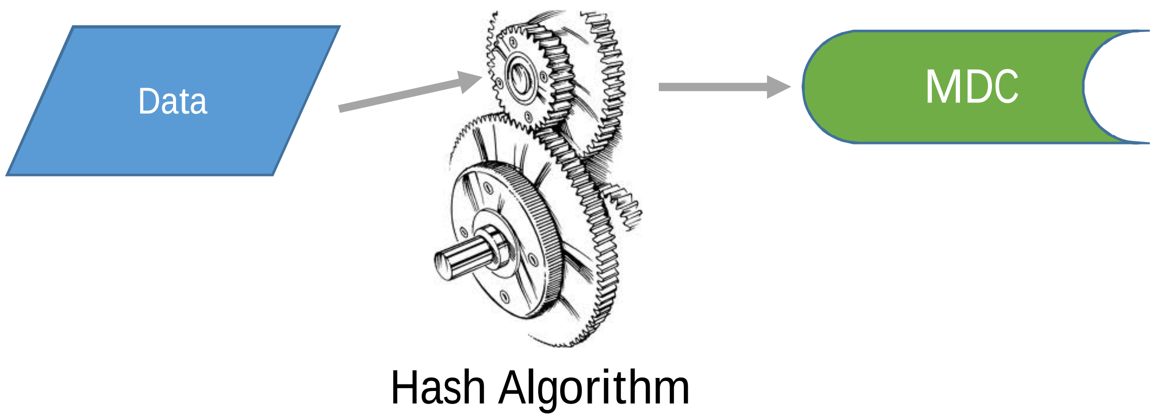
$$\mathrm{Ch}(E,F,G) = (E \wedge F) \oplus (\neg E \wedge G)$$
$$\mathrm{Ma}(A,B,C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$$
$$\Sigma_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$$
$$\Sigma_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$$

- The bitwise rotation uses different constants for SHA-512. The given numbers are for SHA-256. The red ⊞ is addition modulo 2^{32}.



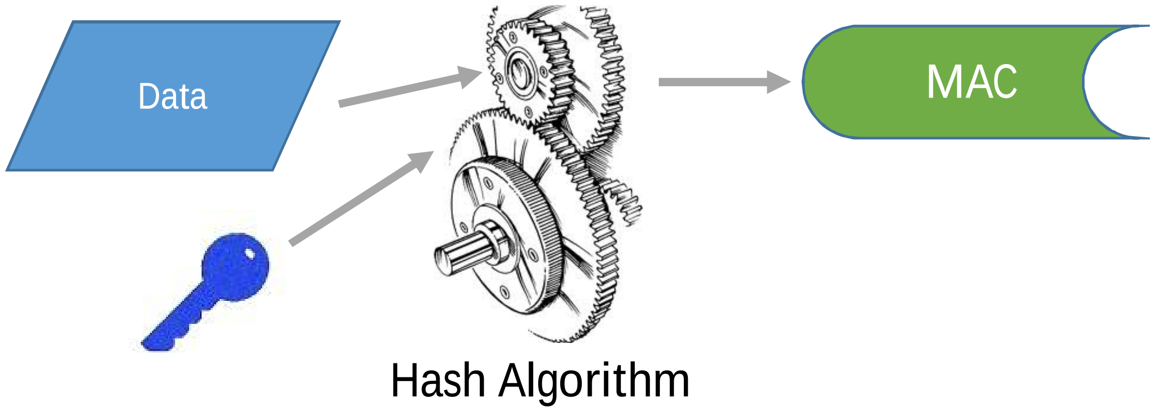
from wikipedia

Hashing – Modification Detection Code



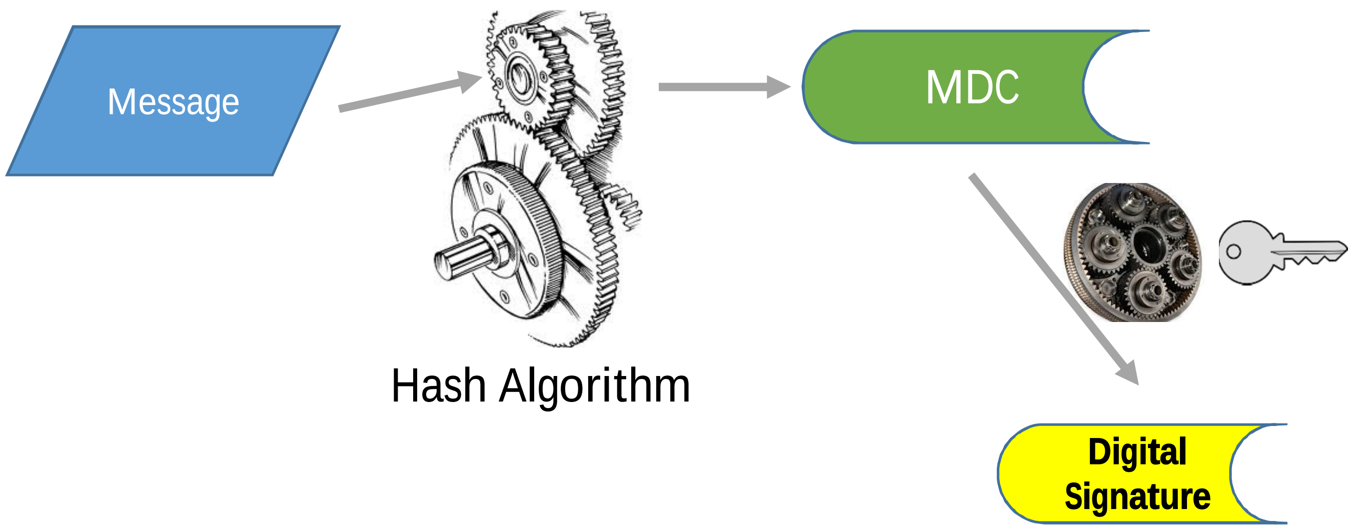
- Characteristics of a good hash algorithm
  - One-way – can't recover the data from the hash
  - Hard to find collisions
  - The result does not reveal information about the input

Hashing – Message Authentication Code

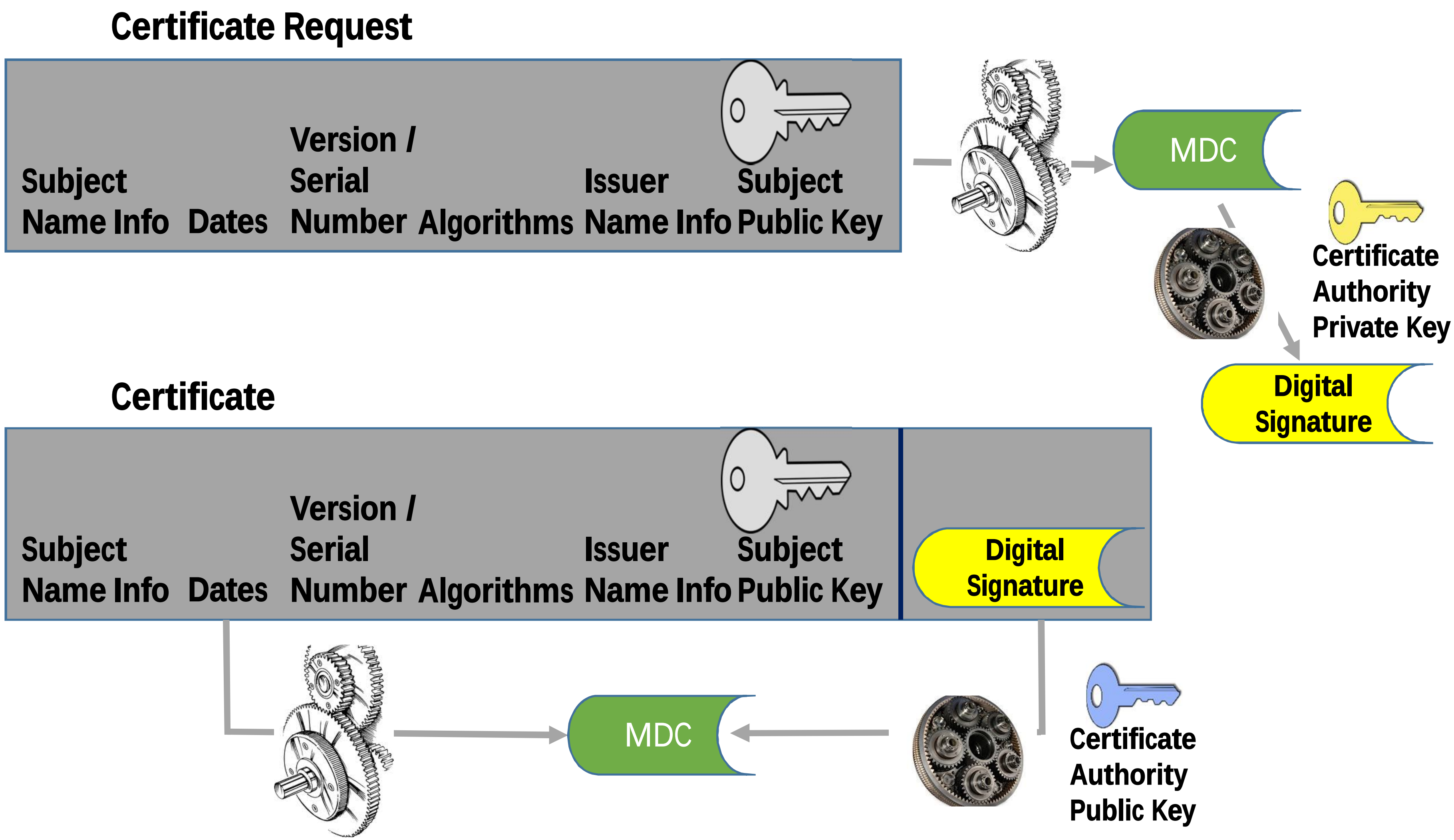


- Add a secret key into the calculation

Digital Signatures



Digital Certificate



Financial Authentication

Routing Number: 12345678
Account Number: 9876543210
Sequence Number: 1

1159

Offset:
1234

0925

PIN Block:
4567898765432101

8A092F6E7D637B25

Decimalization Table			
0 -> 0	1 -> 1	2 -> 2	3 -> 3
4 -> 4	5 -> 5	6 -> 6	7 -> 7
8 -> 8	9 -> 9	A -> 0	B -> 1
C -> 2	D -> 3	E -> 4	F -> 5

Pin Block Formats
ECI-2, ECI-3, ISO-0, ISO-1, ISO-2, ISO-3, VISA-2, VISA-3, VISA-4, 3621, 3624, 4704-EPP

References

- Cryptography Books
  - Bruce Schneier, 'Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in "C"', John Wiley & Sons, Inc. 1996
  - Simon Singh, 'The Code Book', Anchor Books, 1999
  - Niels Ferguson, Bruce Schneier, 'Practical Cryptography', Wiley Publishing, Inc. 2003
- Free Stuff
  - www.schneier.com – Bruce Schneier website, with monthly newsletter Crypto-gram
- Standards
  - csrc.nist.gov – Computer Security Resource Center of NIST
  - www.emc.com/emc-plus/rsalabs - RSA Labs

Questions?

gregboyd@mainframecrypto.com

QR Code

- Share #15659