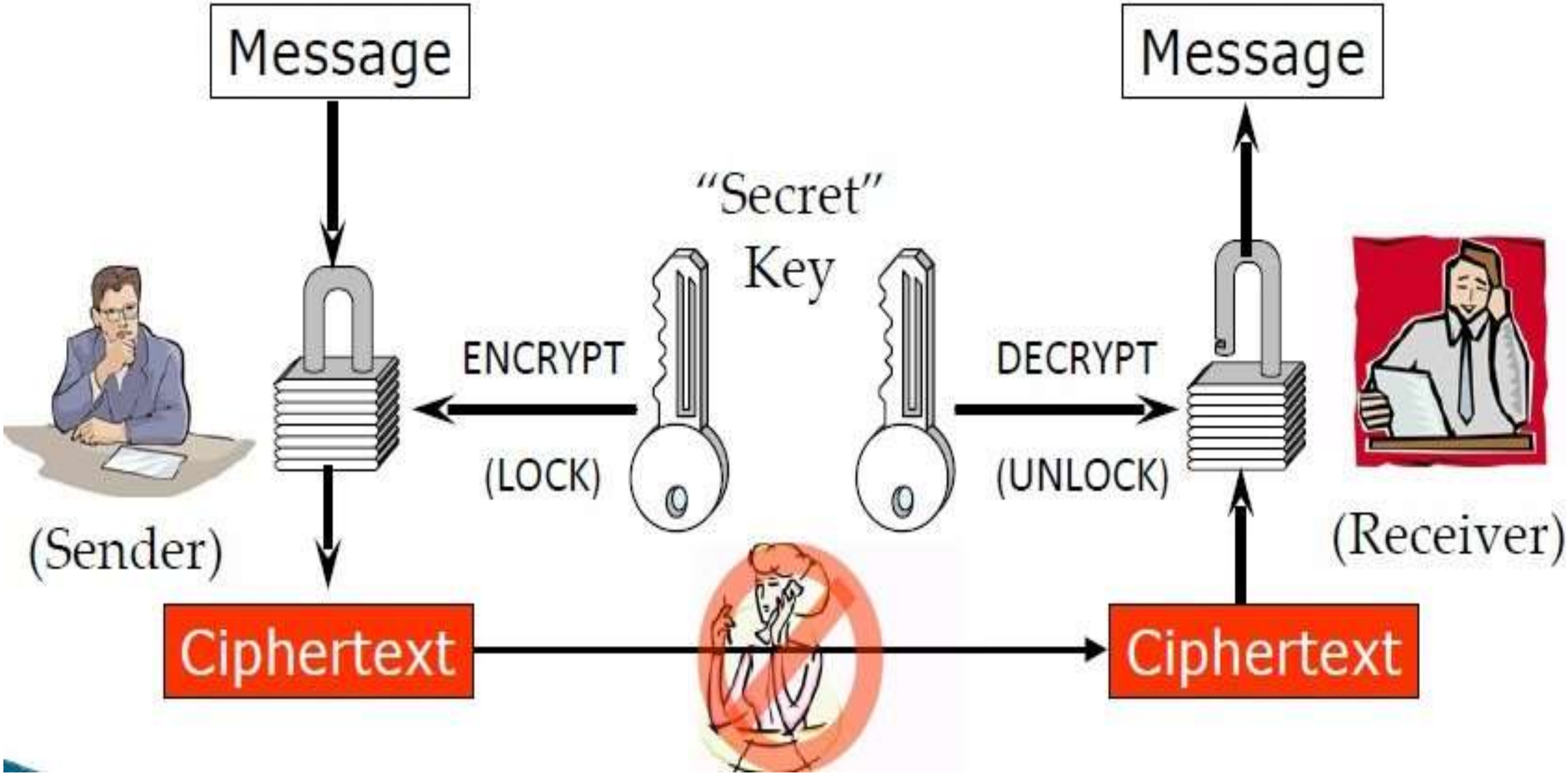
Digital Signature

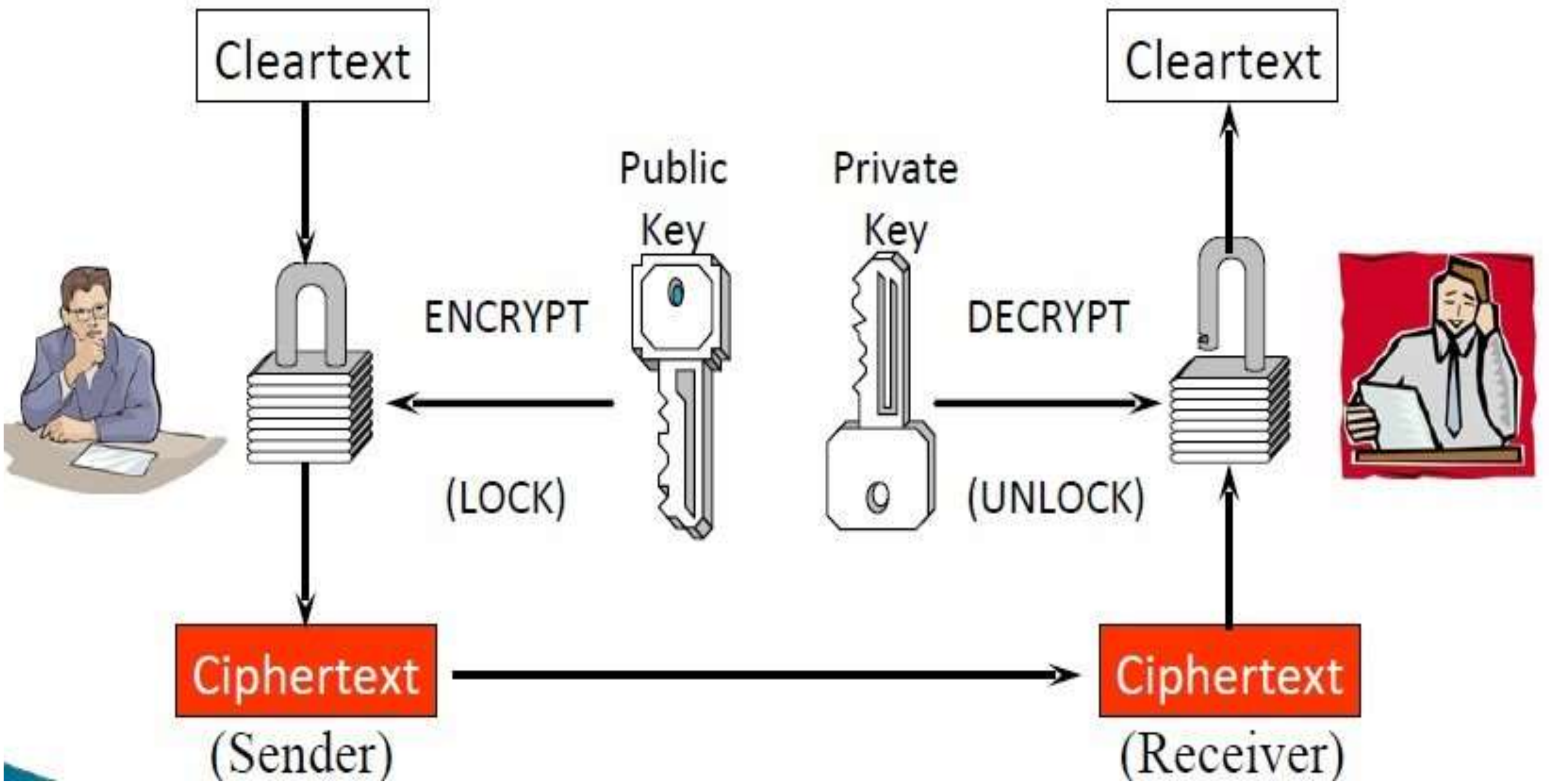
Agenda

- What is Digital Signature?
- How DS work?
- General Framework of DS
- Key Requirements
- Private Key Protection
- Benefits of Digital Signature
- Drawbacks of Digital Signature
- Applications

Symmetric Key Cryptography



Asymmetric Key Cryptography

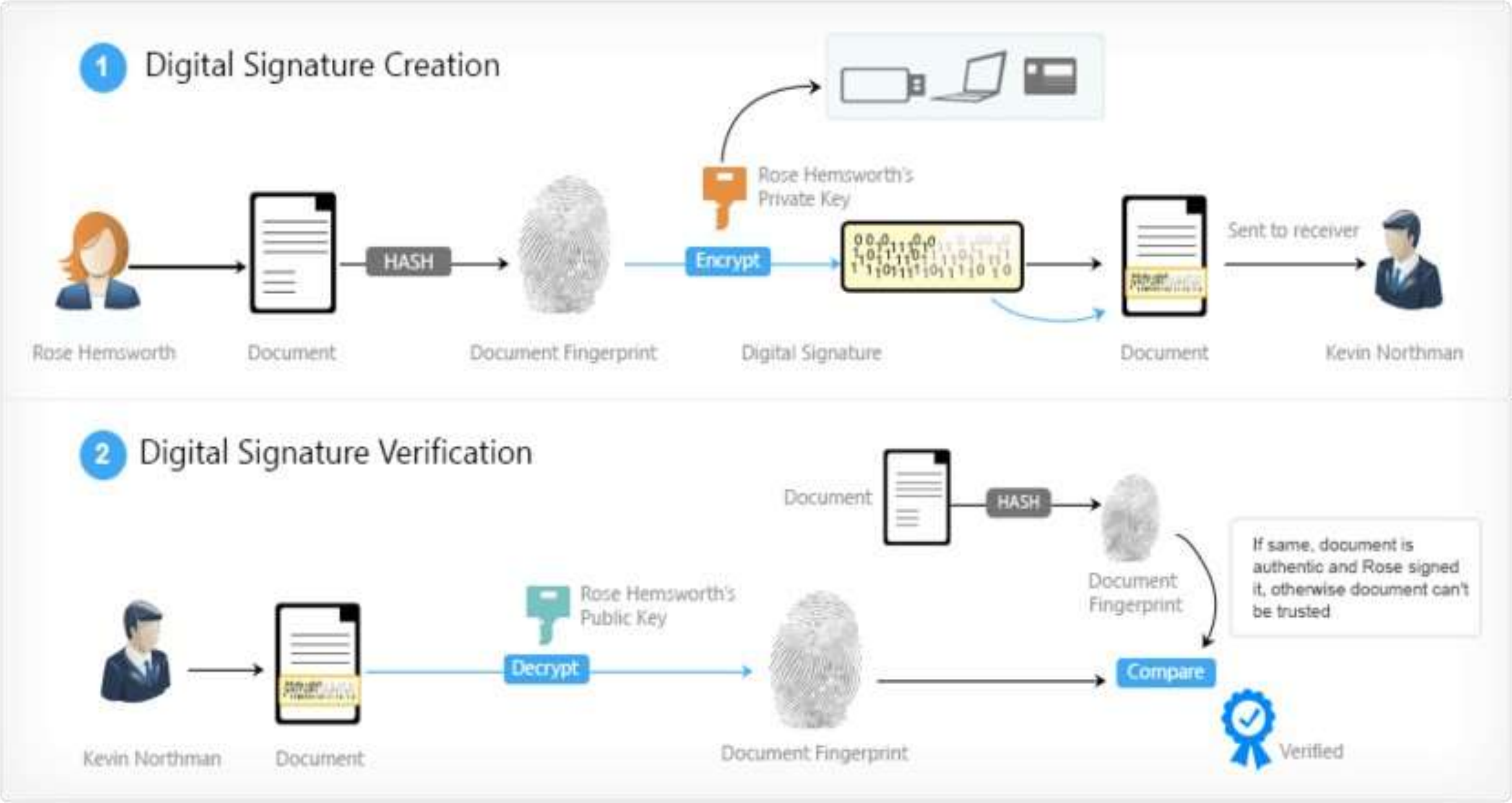


What is Digital Signature?

- **Digital Signature** is a type of asymmetric cryptography used to simulate the security properties of a signature in digital, rather than written, form.

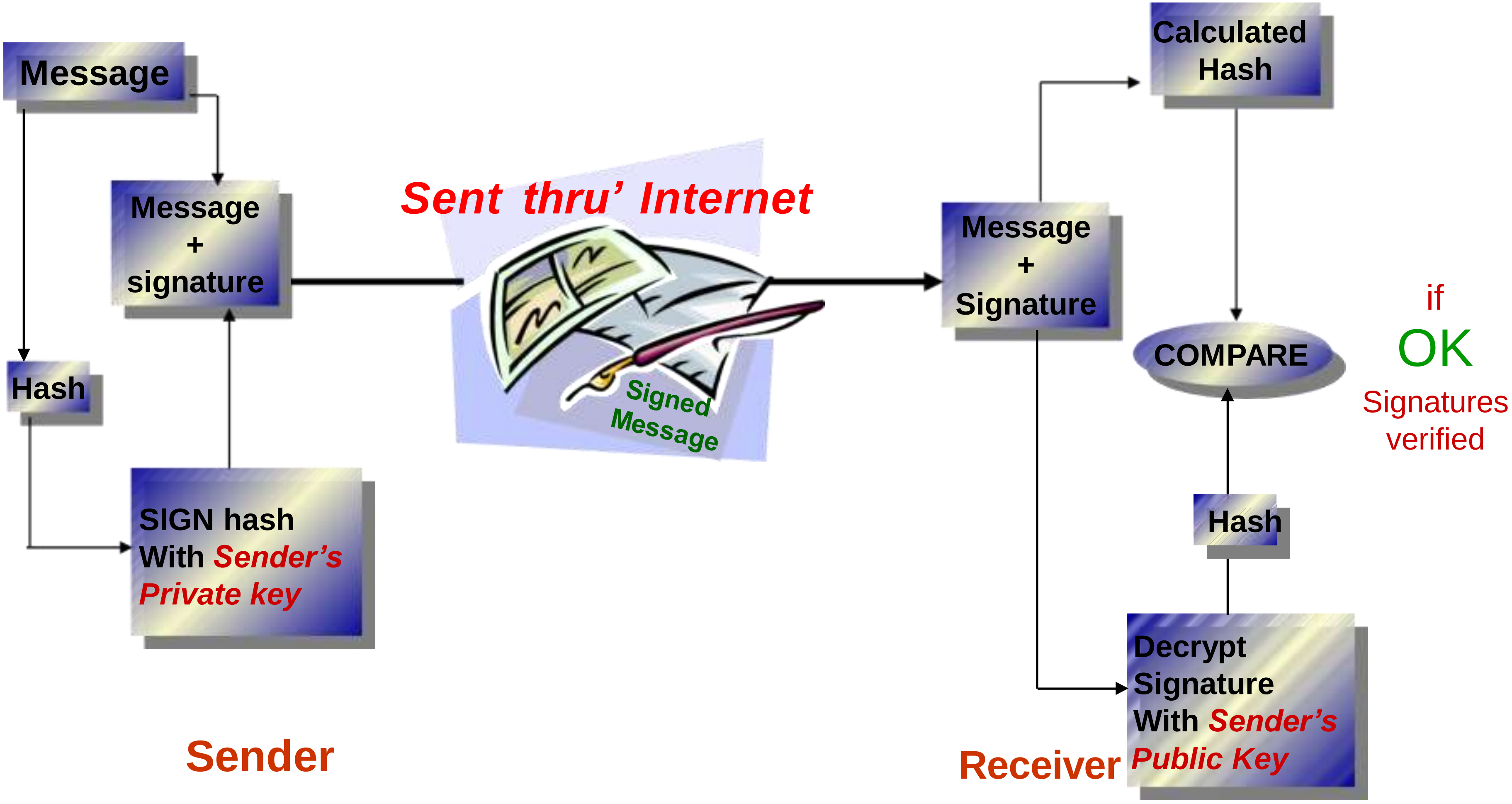
- **Digital Signature** is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document, and possibly to ensure that the original content of the message or document that has been sent is unchanged.

How it work ?



If same, document is authentic and Rose signed it, otherwise document can't be trusted

How it work?



Message

Hash

Message + signature

SIGN hash With ***Sender's Private key***

Sender

Sent thru' Internet

Signed Message

Message + Signature

Calculated Hash

COMPARE

Hash

Decrypt Signature With ***Sender's Public Key***

Receiver

if

OK

Signatures verified

How it work ?

- The use of digital signatures usually involves two processes, one performed by the signer and the other by the receiver of the digital signature:
  - **Digital Signature Creation**
    - **The** process is performed by the sender of the message.
  - **Digital Signature Verification**
    - The process is performed by the receiver of the message.

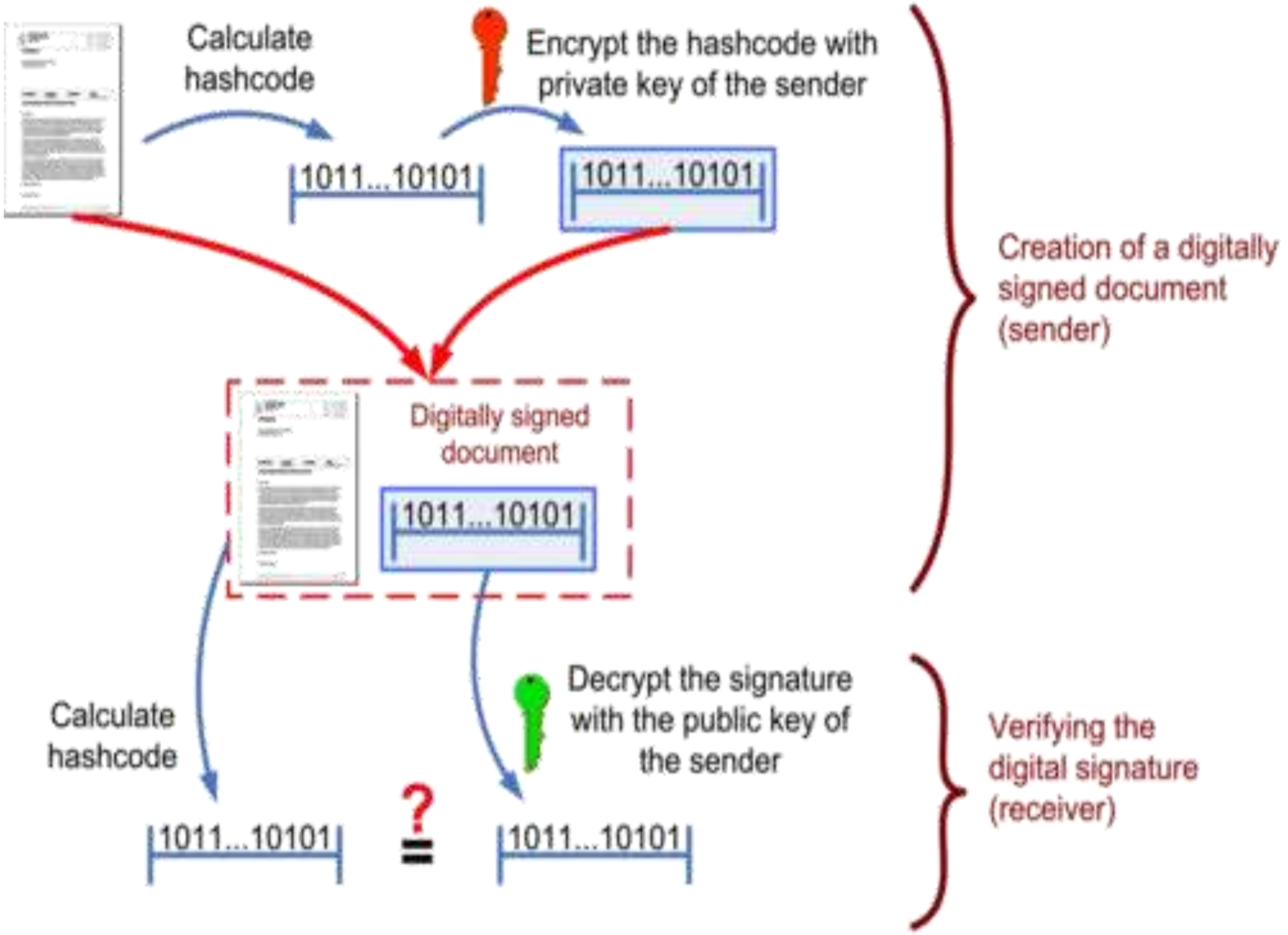
How it work ?

- **Digital Signature Creation:**
  - Uses a hash result derived from and unique to both the signed message and a given private key. For the hash result to be secure, there must be only a negligible possibility that the same digital signature could be created by the combination of any other message or private key.

- **Digital Signature Verification:**
  - is the process of checking the digital signature by reference to the original message and a given public key, thereby determining whether the digital signature was created for that same message using the private key that corresponds to the referenced public key.

Digital Signature Framework

Creating and verifying a digital signature


Calculate hashcode
Encrypt the hashcode with private key of the sender
1011...10101
1011...10101
Creation of a digitally signed document (sender)
Digitally signed document
1011...10101
Calculate hashcode
Decrypt the signature with the public key of the sender
Verifying the digital signature (receiver)
1011...10101
?
=
1011...10101

If the calculated hashcode does not match the result of the decrypted signature, either the document was changed after signing, or the signature was not generated with the private key of the alleged sender.

Key Requirements

- Each individual generates his own key pair, private and public keys.

- Private key:
  - Only known by the owner/sender
  - Used to create the digital signature

- Public key
  - It is known to everyone.
  - Used to verify the digital signature.

Key Requirements

- Digital Certificate:
  - Digital Identity that establishes your credentials when doing business or other transactions on the Web
  - Issued by a Certifying Authority (CA)
  - Contains your name, serial number, expiration dates, public key, signature of CA.

- Certificate Authority:
  - Trusted Third Party.
  - An organization which issues public key certificates.
  - Assures the identity of the parties to whom it issues certificates.
  - Maintains online access to the public key certificates issued.

Private Key Protection

Soft Token

Hardware tokens

Smart card

Benefits of Digital Signature

1. **Authentication**

   - Digital signatures can be used to authenticate the source of messages. When ownership of a digital signature secret key is bound to a specific user, a valid signature shows that the message was sent by that user.

   - The importance of high confidence in sender authenticity is especially obvious in a financial context.

Benefits of Digital Signature

2. Integrity:

- In many scenarios, the sender and receiver of a message may have a need for confidence that the message has not been altered during transmission.
- Although encryption hides the contents of a message, it may be possible to change an encrypted message without understanding it.
- If a message is digitally signed, any change in the message will invalidate the signature.
- Furthermore, there is no efficient way to modify a message and its signature to produce a new message with a valid signature, because this is still considered to be computationally infeasible by most cryptographic hash functions.

Paper Signature vs. Digital Signature

Parameter	Paper	Electronic
Authenticity	May be forged	Can not be copied
Integrity	Signature independent of the document	Signature depends on the contents of the document
Non-repudiation	a. Handwriting expert needed b. Error prone	a. Any computer user b. Error free

Drawbacks of Digital Signature

- The private key must be kept in a secure manner.
- The process of generation and verification of digital signature requires considerable amount of time.
- For using the digital signature the user has to obtain private and public key, the receiver has to obtain the digital signature certificate also.

Applications

- Electronic Mail
- Data Storage
- Electronic Funds Transfer
- Software Distribution
- Smart Cards
- Blind Signatures
- Time Stamped Signature

Secure Hash Algorithm

Secure Hash Algorithm

- SHA originally designed by NIST & NSA in 1993
- was revised in 1995 as SHA-1
- US standard for use with DSA signature scheme
  - standard is FIPS 180-1 1995, also Internet RFC3174
  - nb. the algorithm is SHA, the standard is SHS
- based on design of MD5 with key differences
- produces 160-bit hash values
- recent 2005 results on security of SHA-1 have raised concerns on its use in future applications

How SHA Works?

- Digest Length=***160 bit***
- I/P Text=512 bit
- Sub Block size=32bit
- 512/32=16 total Sub blocks
- No. Of Rounds=4
- Iteration per round=***20***
- Chaining Variable = 5*32=160
- K[t] constant= ***Where t=0 to 79***
- O/P-> four 32 bit blocks

SHA Overview

1. ***Padding:*** Length of the message is 64 bits short of multiple of 512 after padding.
2. ***Append*** a 64-bit ***length*** value of original message is taken.
3. ***Divide the input into 512-bit blocks***
4. ***Initialise CV*** 5-word (160-bit) buffer (A,B,C,D,E) to

(***A***=01 23 45 67,

B=89 AB CD EF,

C=FE DC BA 98,

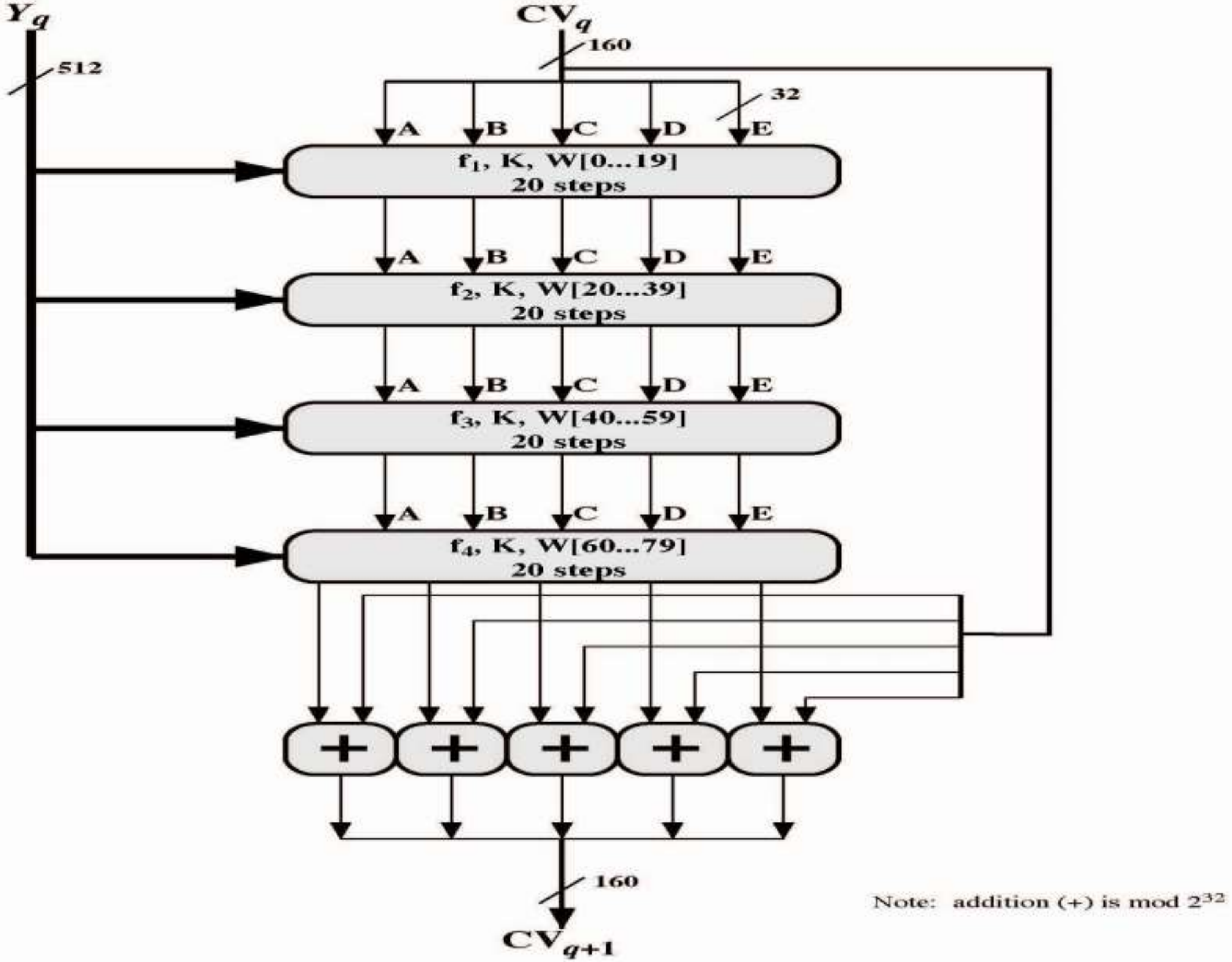
D=76 54 32 10,

E=C3 D2 E1 F0)

Continue…

5. ***Process Blocks*** now the actual algorithm begins. message in 16-word (512-bit) chunks:
   - Copy CV into single register for storing temporary intermediate as well as the final results.
   - Divide the current 512-bit blocks into 16 sub-blocks, each consisting of 32 bits.
   - □ Has No. Of Rounds=4, each round consisting of 20 ***bit /step iteration*** operations on message block & buffer
   - □ expand 16 words into 80 words(20*4) by mixing & shifting.K[t] constant= ***Where t=0 to 79***
   - □ Form new buffer value by adding output to input.

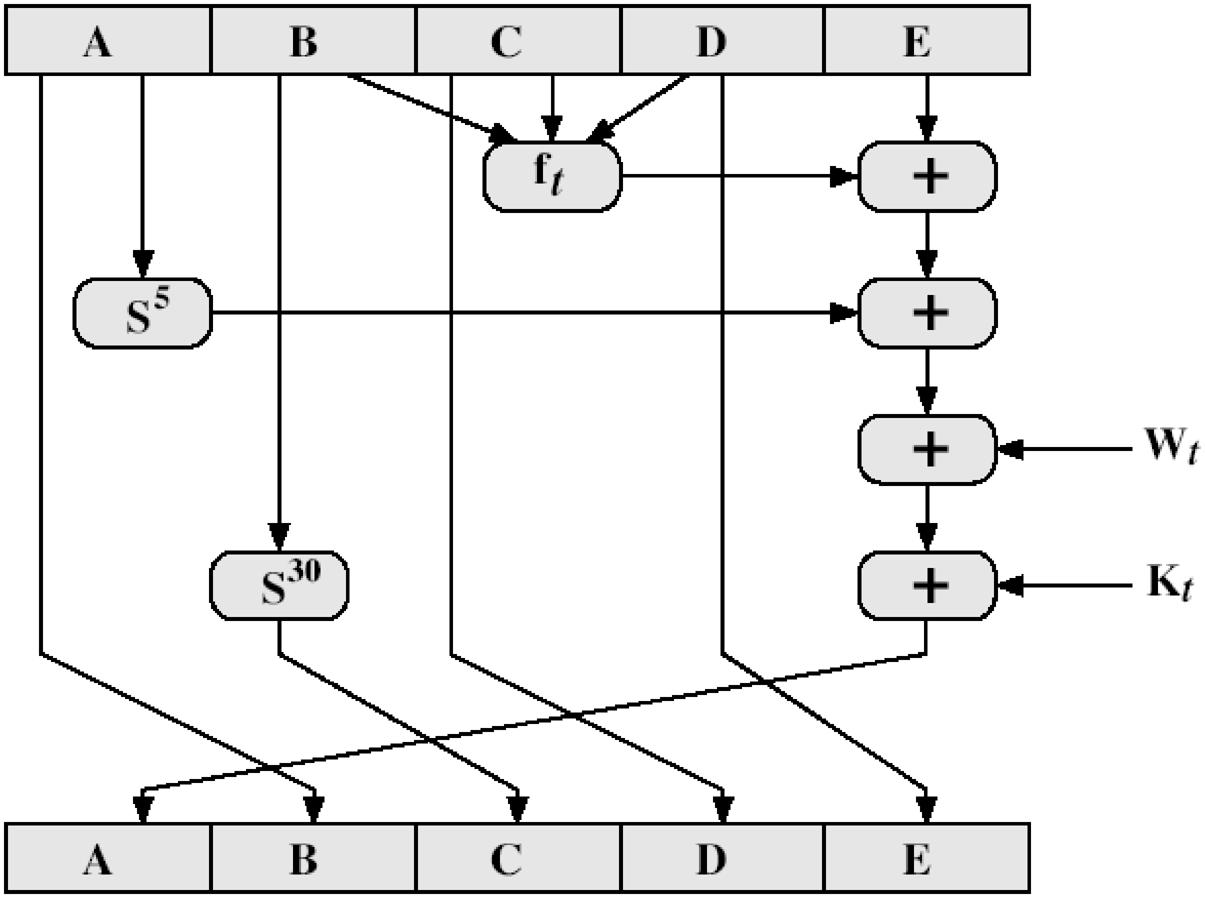
6. output hash value is the final buffer value



Note: addition (+) is mod 2^{32}

Figure 12.5 SHA-1 Processing of a Single 512-bit Block (SHA-1 Compression Function)

SHA-1 Compression Function



ABCDE=(F[t]+E+S5(A)+W[t]+K[t]),>>>Shift right by 1 bit for next iteration

SHA-1 Compression Function terms

- each round has 20 steps which replaces the 5 buffer words thus:

  ```
  (A,B,C,D,E) <-
    (E+f(t,B,C,D)+(A<<5)+Wt+Kt),A,(B<<30),C,D)
  ```

- ABCDE refer to the 5 words of the buffer
- t is the step number
- `f(t,B,C,D)` is nonlinear function for round
- W_t is derived from the message block
- K_t is a constant value
- `S^t` circular left shift of 32 bit sub-block by t bits

Process F(t) in each SHA-1 round

□where g can be expressed as:

ROUND 1: (b AND c) OR ((NOT b) AND (d)) same as MD5

ROUND 2: b XOR c XOR d

ROUND 3: (b AND c) OR (b AND d) OR (c AND d)

ROUND 4: b XOR c XOR d

Creation of 80-word input W_t

- Adds redundancy and interdependence among message blocks

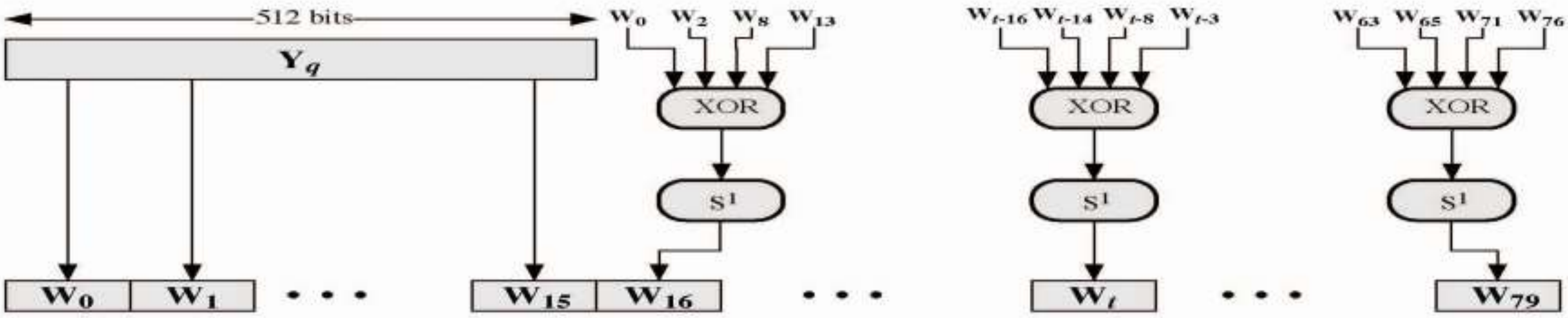

512 bits
Y_q
W_0 W_2 W_8 W_{13}
W_{t-16} W_{t-14} W_{t-8} W_{t-3}
W_{63} W_{65} W_{71} W_{76}
XOR
XOR
XOR
S^1
S^1
S^1
W_0
W_1
• • •
W_{15}
W_{16}
• • •
W_t
• • •
W_{79}

Figure 12.7 Creation of 80-word Input Sequence for SHA-1 Processing of Single Block

SHA-1 verses MD5

- brute force attack is harder (160 vs 128 bits for MD5)
- not vulnerable to any known attacks (compared to MD4/5)
- a little slower than MD5 (80 vs 64 steps)
- both designed as simple and compact

Revised Secure Hash Standard

- NIST issued revision FIPS 180-2 in 2002
- adds 3 additional versions of SHA
  - SHA-256, SHA-384, SHA-512
  - Different lengths of Message Digest in bits
- designed for compatibility with increased security provided by the AES cipher
- structure & detail is similar to SHA-1
- hence analysis should be similar
- but security levels are rather higher

Table 12.3 Comparison of SHA Properties

	SHA-1	SHA-256	SHA-384	SHA-512
Iessage digest size	160	256	384	512
Iessage size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
lock size	512	512	1024	1024
Vord size	32	32	64	64
umber of steps	80	80	80	80
ecurity	80	128	192	256

Iotes: 1. All sizes are measured in bits.
2. Security refers to the fact that a birthday attack on a message digest of size n produces a collision with a workfactor of approximately $2^{n/2}$.