CYBER SECURITY (PS02EMCA37)

Unit-3 : Cryptography

- Security Services: Confidentiality, Authentication, Integrity, Non-repudiation, Access Control, Availability
- Symmetric Key Algorithms (DES & AES)
- Asymmetric Key Algorithms (RSA)
- Digital Signature & Message Digest
- Digital Certificate

SECURITY SERVICES

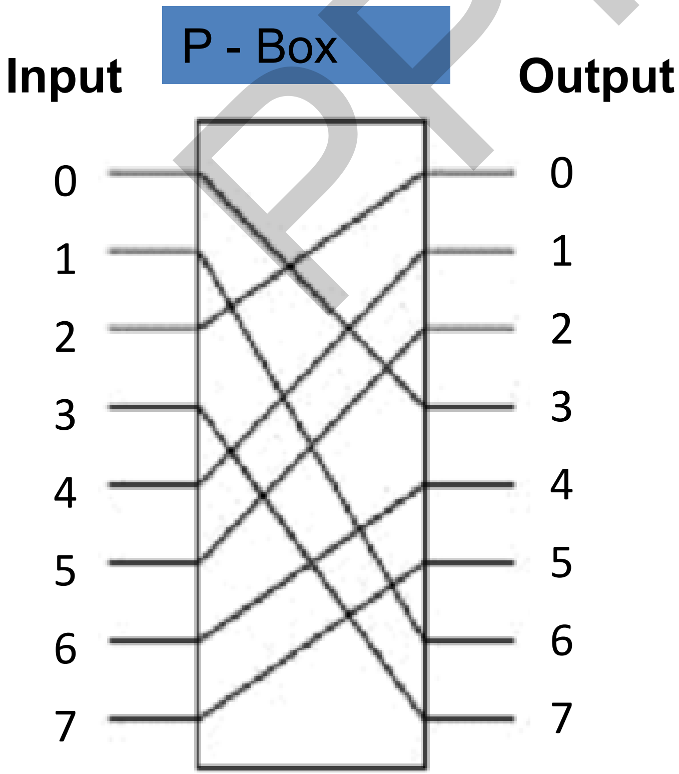
- Secrecy (confidentiality)
  - Keeping information secret from intermediate users.
  - Only the sender and the intended recipient should be able to understand the contents of a message.
- Authentication
  - Determine whom you are communicating before provideing sensitive information.
  - Receiver is sure of the sender's identity
- Nonrepudiation
  - Proof of ownership.
  - Receiver must be able to prove that a message came from a specific sender.
- Integrity control
  - Assuring that a message has not been modified in transit.
  - The data must arrive at the receiver exactly as it was sent

SECURITY SERVICES

- Access Control
  - It is a security technique that controls who or what can view or use resources in a computing environment.
- Availability
  - It is the declaration that a computer system is available or accessible by an authorized user whenever it is needed.

Symmetric-Key Alogrithms

- Use the same secret key to encryption and decryption
- **Block ciphers :** Take an n-bit block of plain text as input and transform with key into n-bit block of cipher text
- Algorithms cab be implemented in hardware (for speed) & software (for flexibility)
- **Device : P-box** used to effect a transposition on 8-bit input.

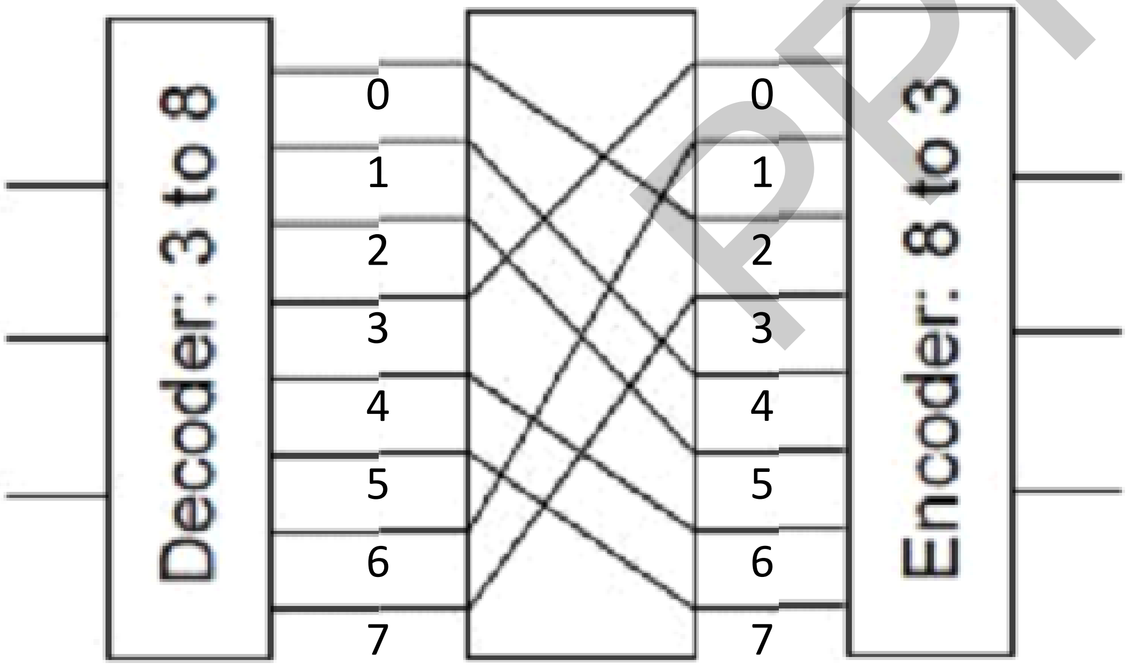


Input:	0	1	2	3	4	5	6	7
Output:	3	6	0	7	1	2	4	5

Symmetric-Key Alogrithms

- **Device : S-box** : It perform substitution.
- 3 bit plain text as input & 3 bit cipher text is output
- 3 bit input selects one of the eight lines exiting from the first stage and set it to 1; all the other lines are 0.
- If input is sequence of octal numbers (0 – 7) find the output.

Input **Output**

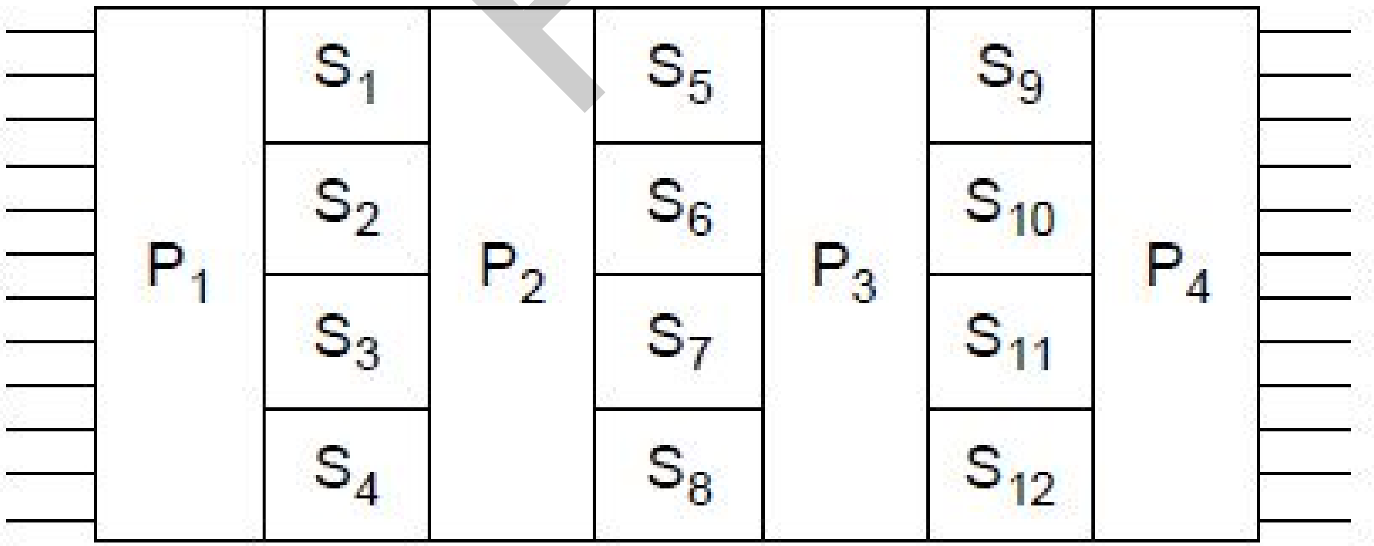


Input (octal values):		0	1	2	3	4	5	6	7
Output	:	2	4	5	0	6	7	1	3

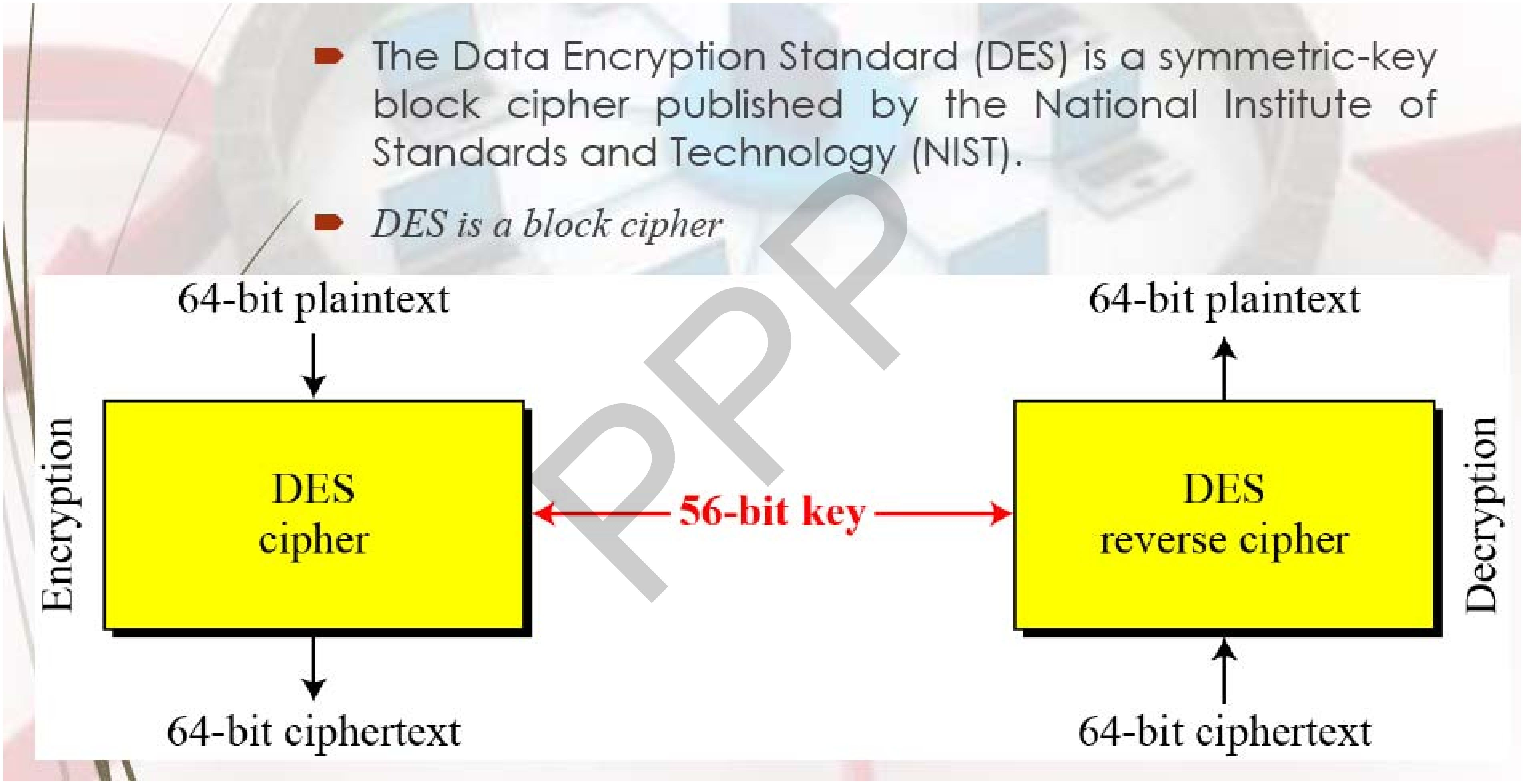
Symmetric-Key Alogrithms

- **Product Cipher:**
  - It is a combination of p-box and s-box
  - Here for example 12 bit input and 12 bit output.
  - Out put of P1 is break up into four groups of 3 bits.
  - Typically, inputs is 64 to 256 bit and 18 physical stages instead of 7 as in below figure.

Product cipher



DES (Data Encryption Standard)



- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).
- *DES is a block cipher*

Encryption

64-bit plaintext

DES
cipher

64-bit ciphertext

56-bit key

Decryption

64-bit plaintext

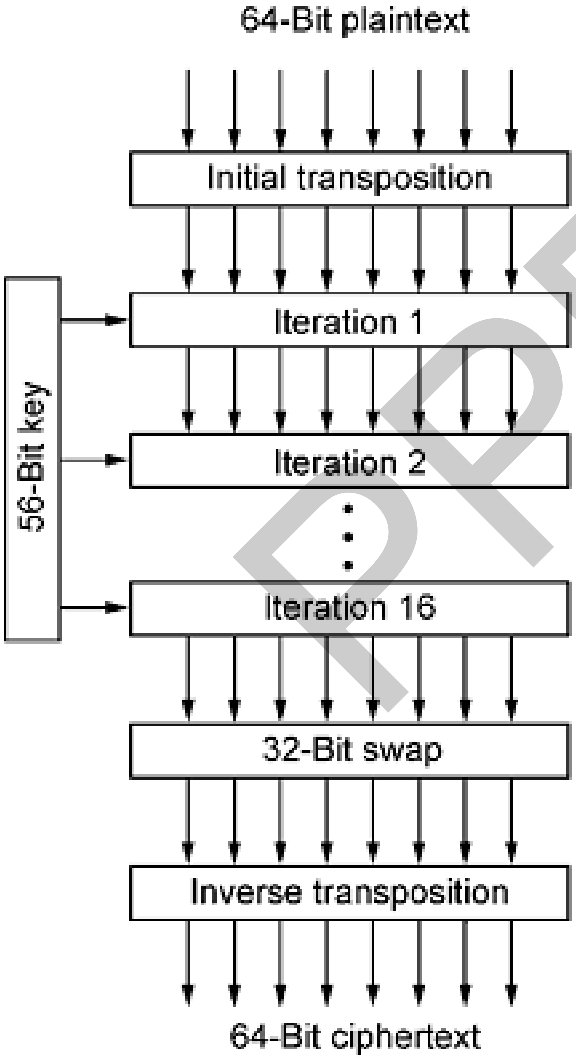
DES
reverse cipher

64-bit ciphertext

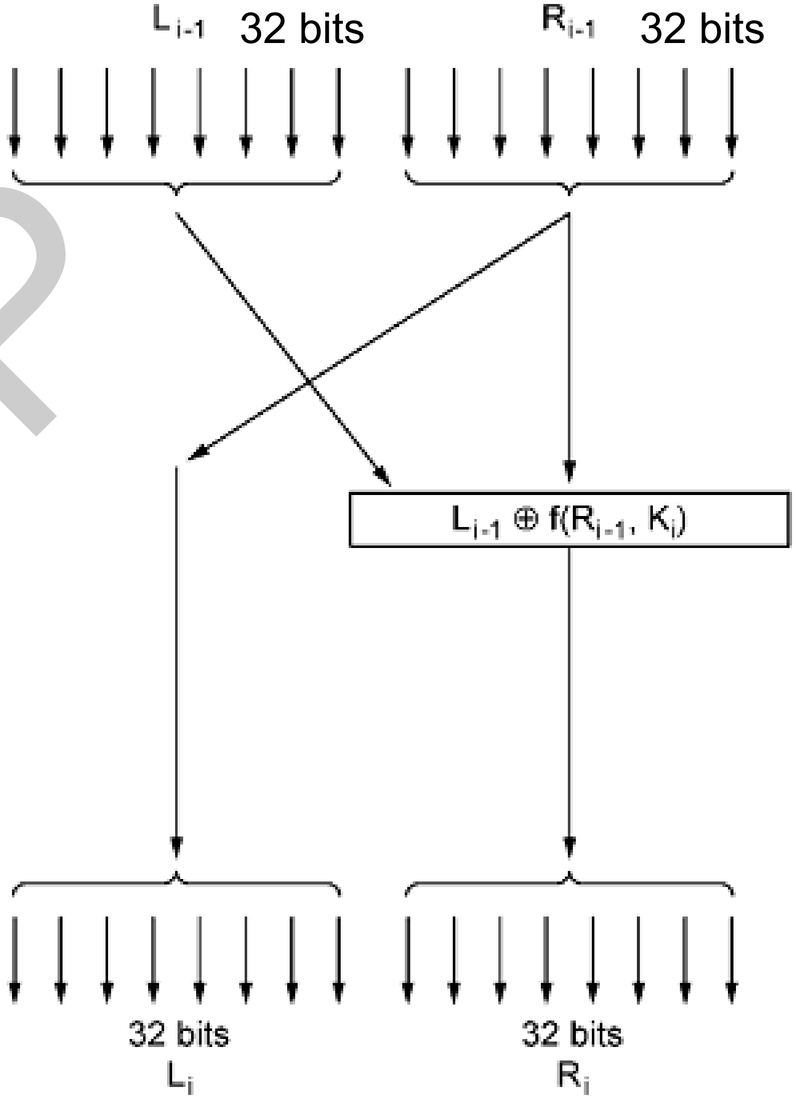
DES (Data Encryption Standard)

- U.S. government adopted a product cipher developed by IBM.
- It was widely adopted by the industry for use in security products.

P. T. size: 64 bit
Key size : 56 bit



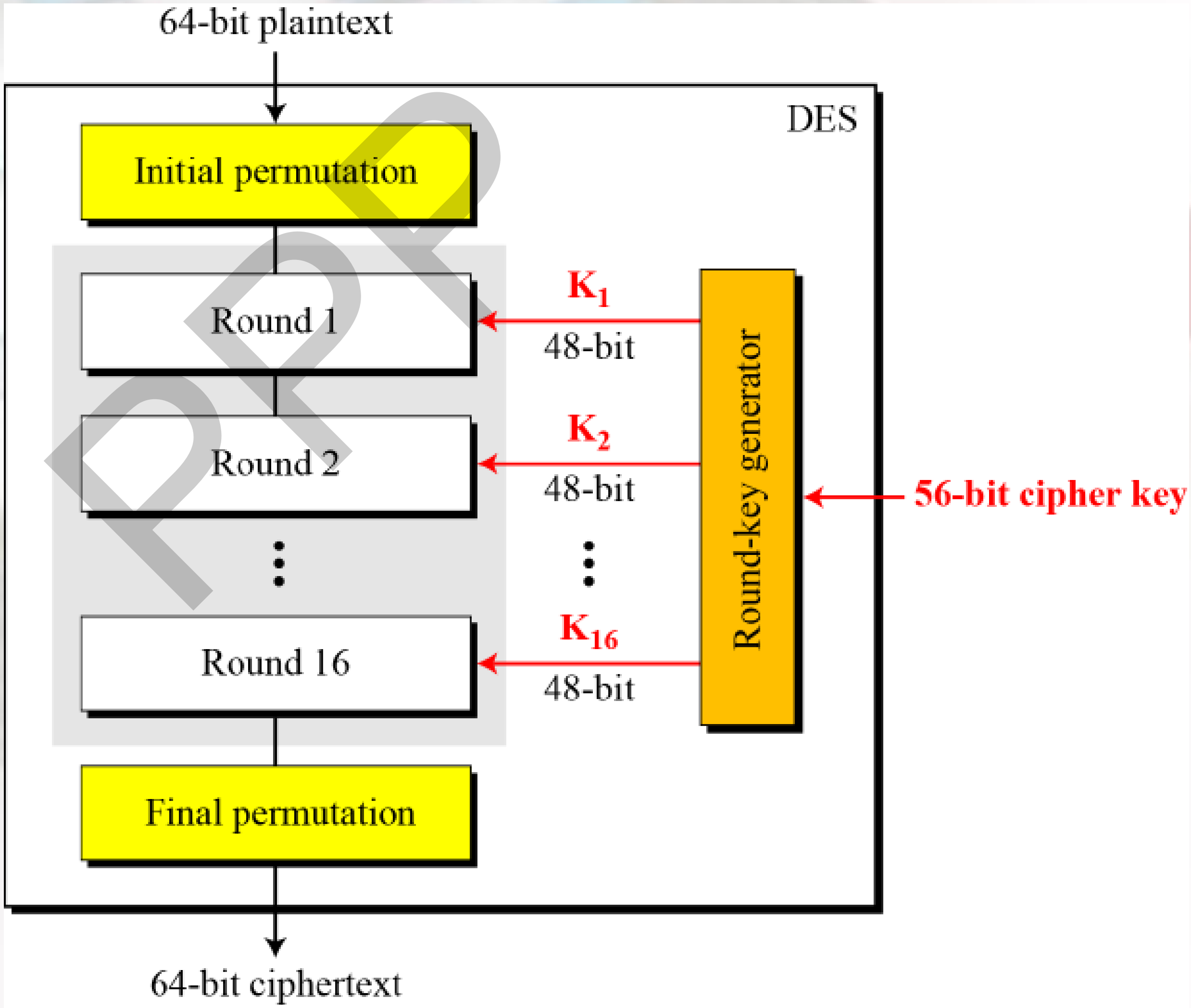
(a)General outline.



(b) Detail of one Iteration

DES Structure

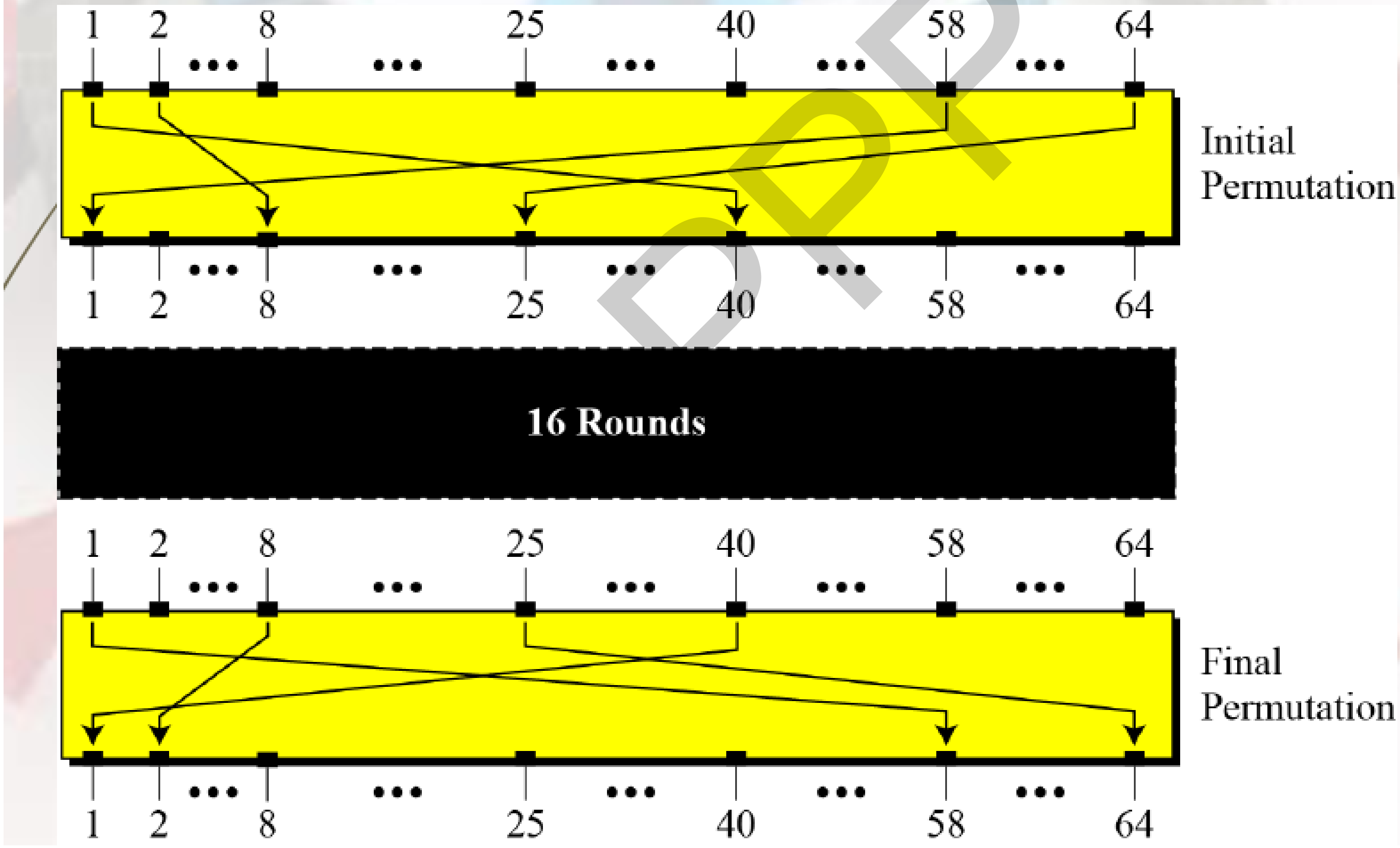
The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



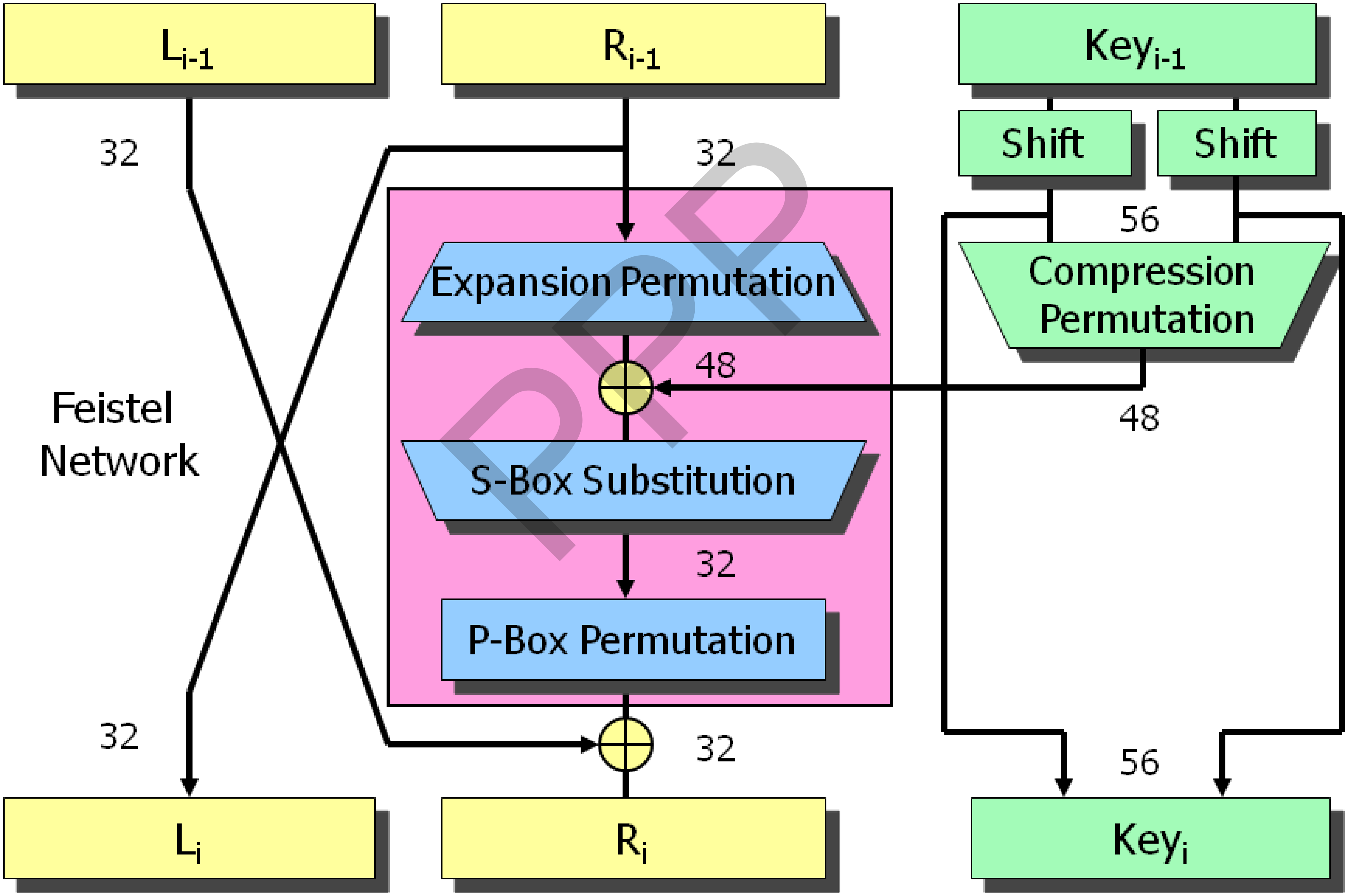
DES

- Initial Permutation and Final Permutation are inverse to each other.
- For Initial Permutation : E.g. A bit at position 1 moves to position 40 & a bit at position 64 moves at position 1 and so on.

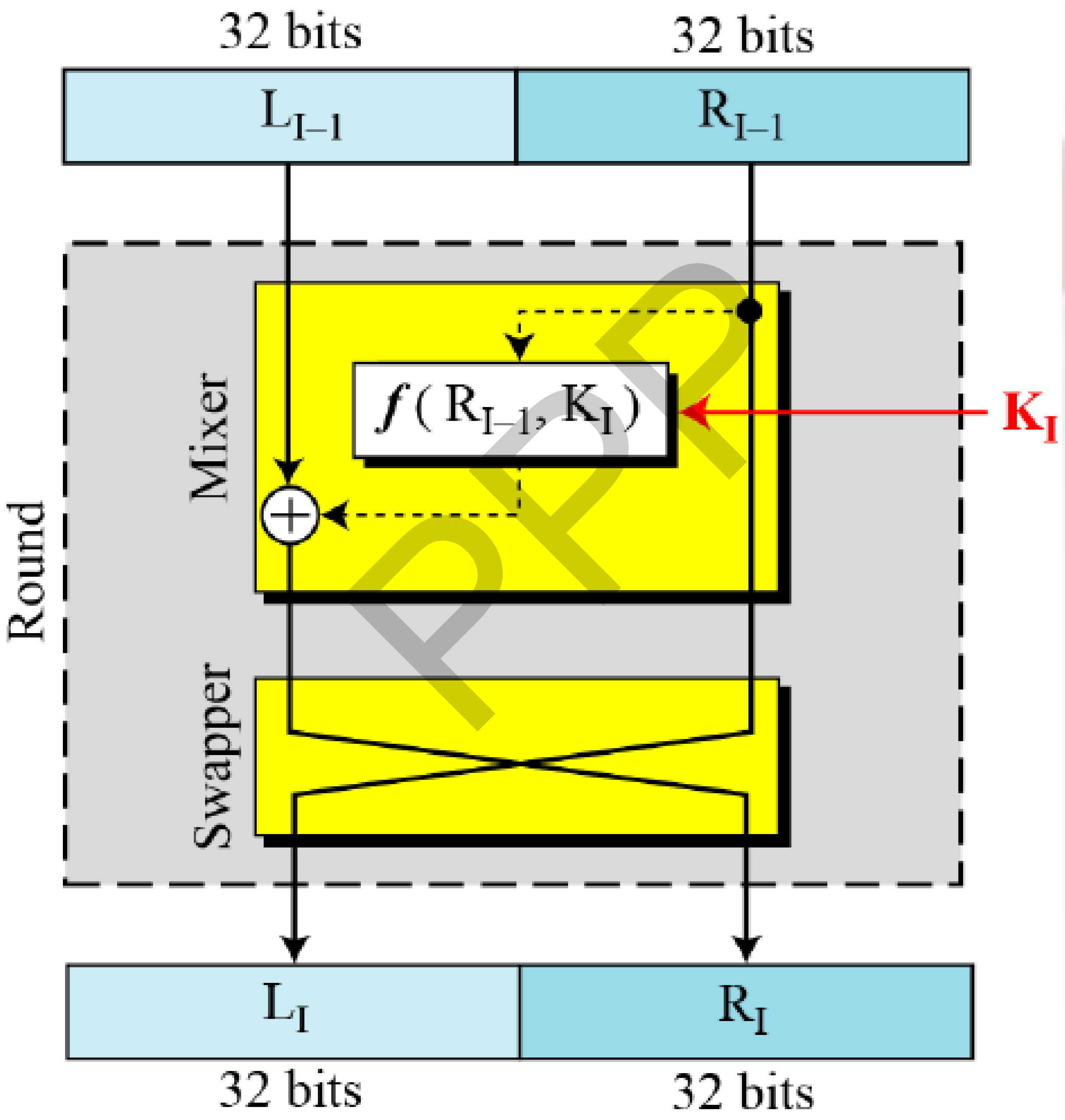
Initial and final permutation steps in DES



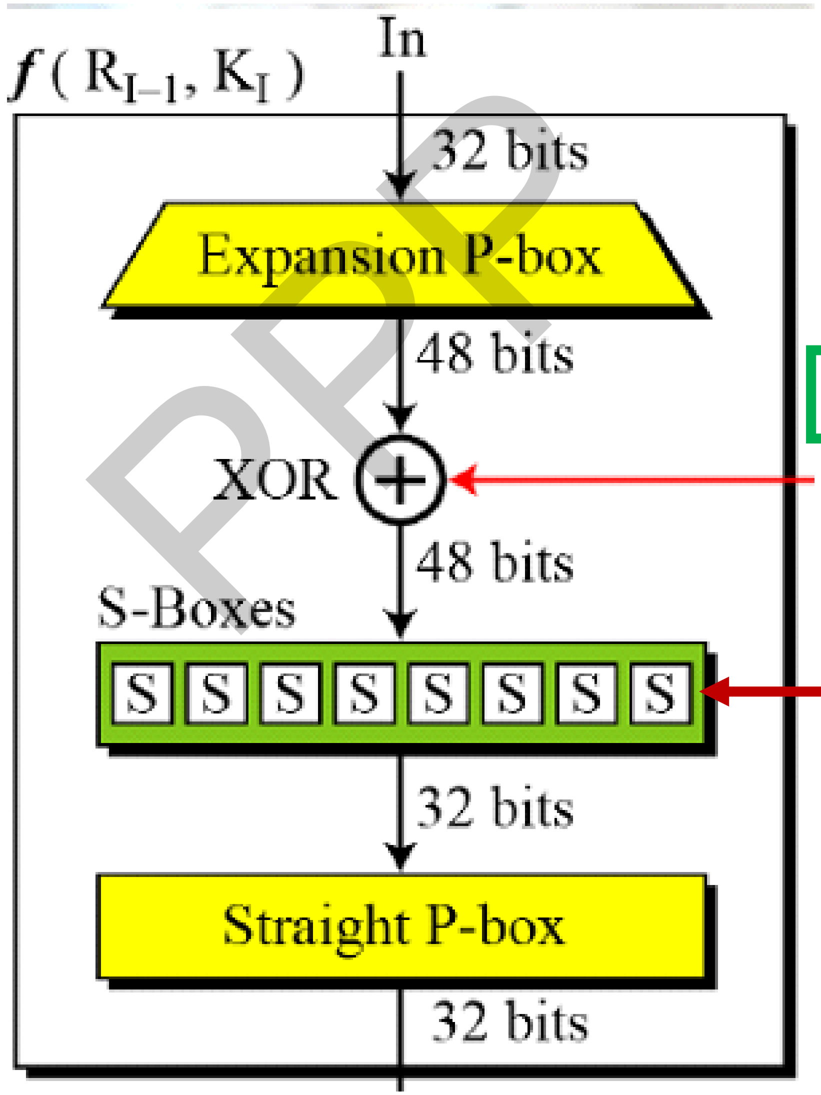
One Round Of DES



DES Round


32 bits
32 bits
L_{I-1}
R_{I-1}
Round
Mixer
$f(R_{I-1}, K_I)$
K_I
Swapper
L_I
R_I
32 bits
32 bits

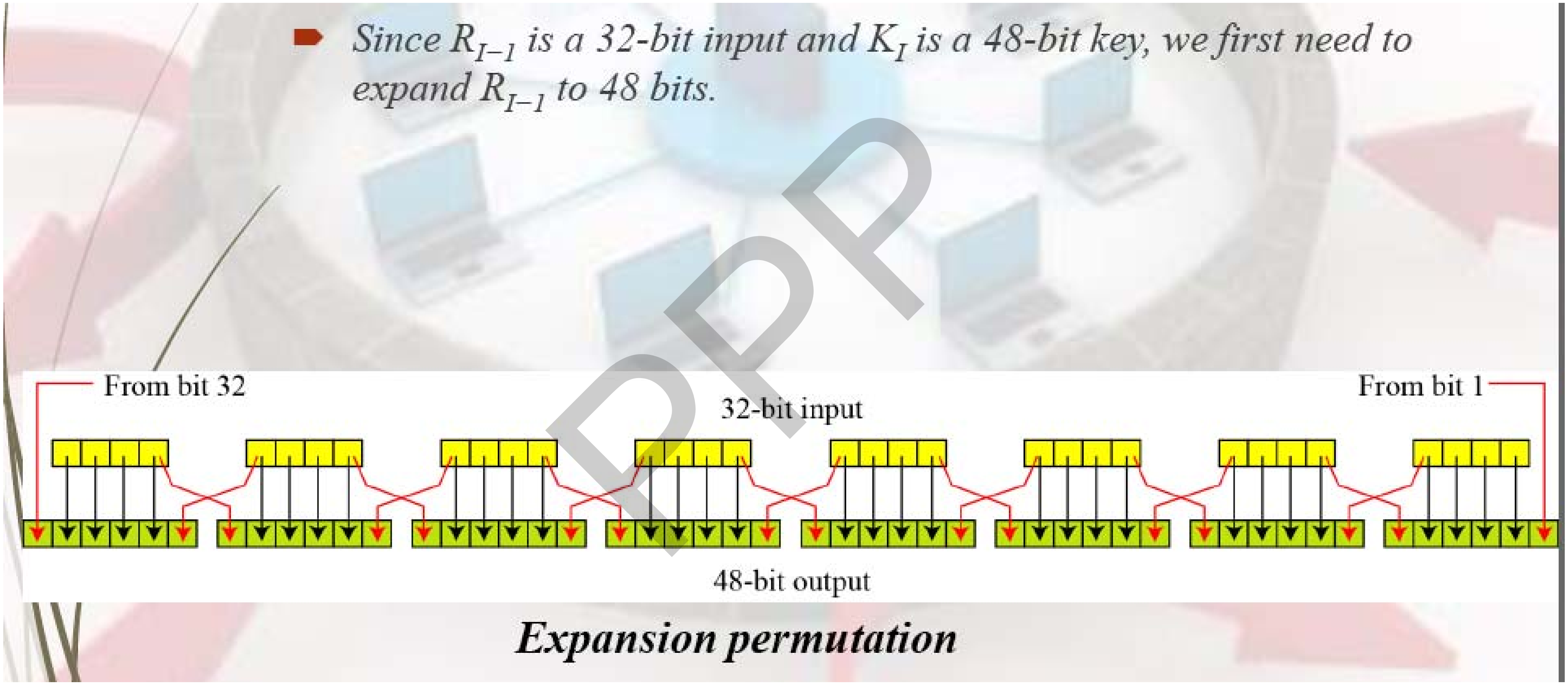
DES : Function In Each Iteration



K_I : 48 Bits

For Each S-Box
Input : 6 Bits
Output : 4 bits

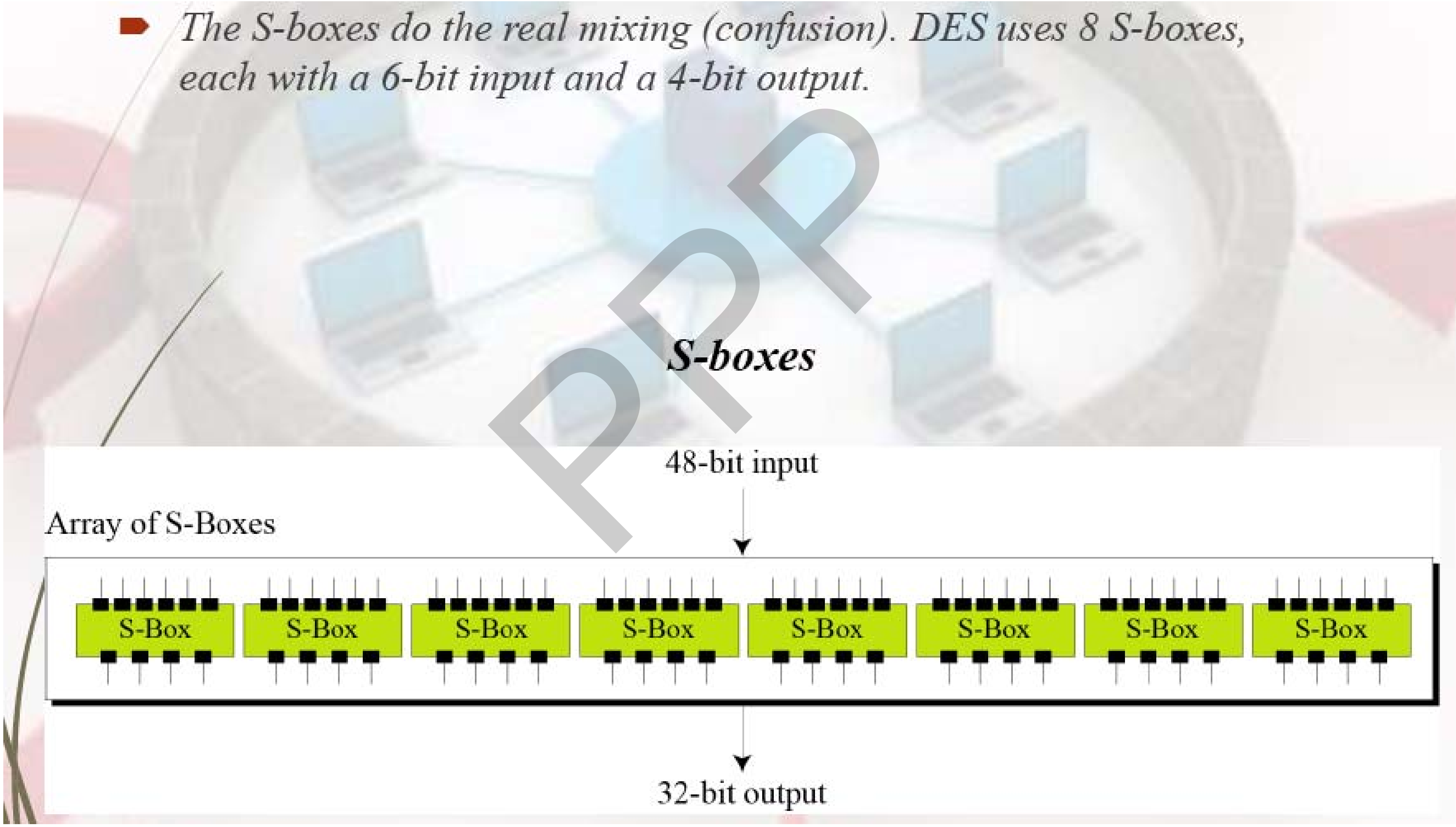
Expansion P-Box



- *Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.*

Expansion permutation

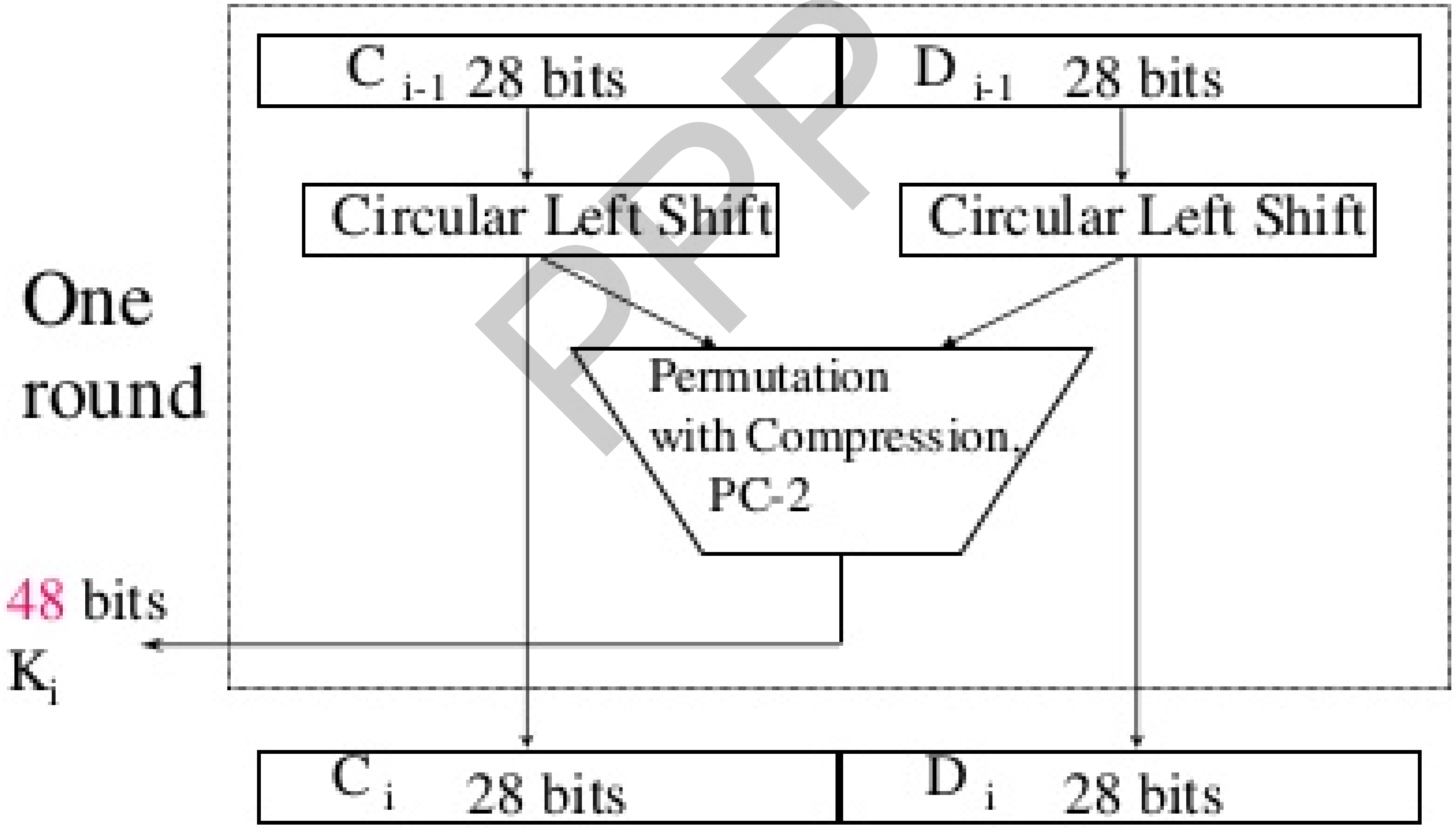
S-Boxes



- *The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.*

S-boxes

Per-Round Key Generation : Convert round key Ki from 56 bit to 48 bit

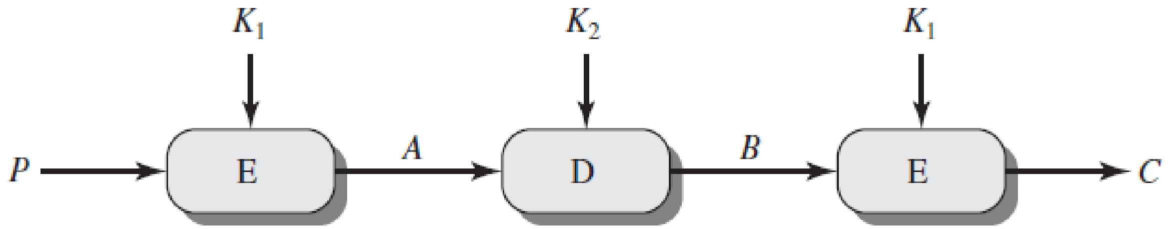


IBM & NSA

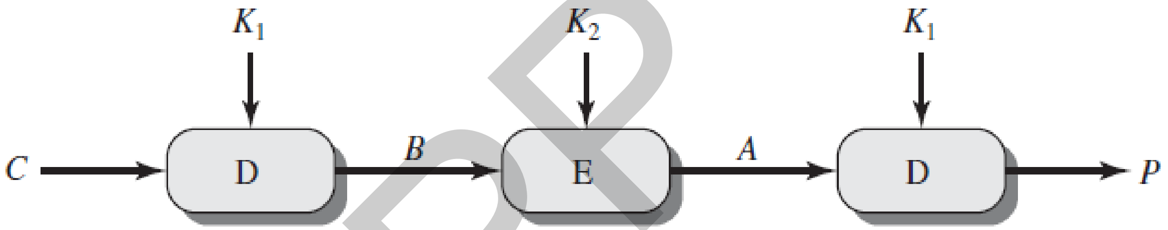
- **NSA : National Security Agency**
- **IBM : International Business Machines**
- IBM’s cipher used 128-bit key but NSA force them to reduce it to 56-bit key & keep secret the process by which DES was designed to make keys.

- In 1977, Stanford cryptography researchers, Diffie and Hellman designed a machine to break DES and estimated that it could be built for 20 million dollars. The machine could creak the 2^{56} possible key in one day.

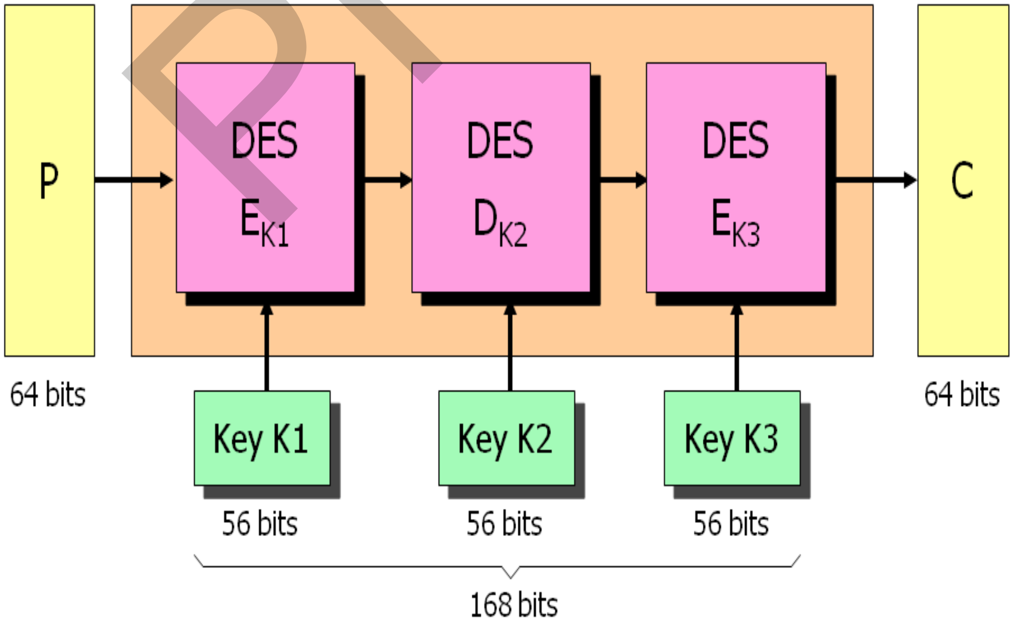
Triple DES



Encryption



Decryption



Triple DES

- Why Only Two keys are used for processing?
  - Key length 112 is enough
- Why Choose EDE instead of EEE?
  - Backward compatibility

AES – The Advanced Encryption Standard

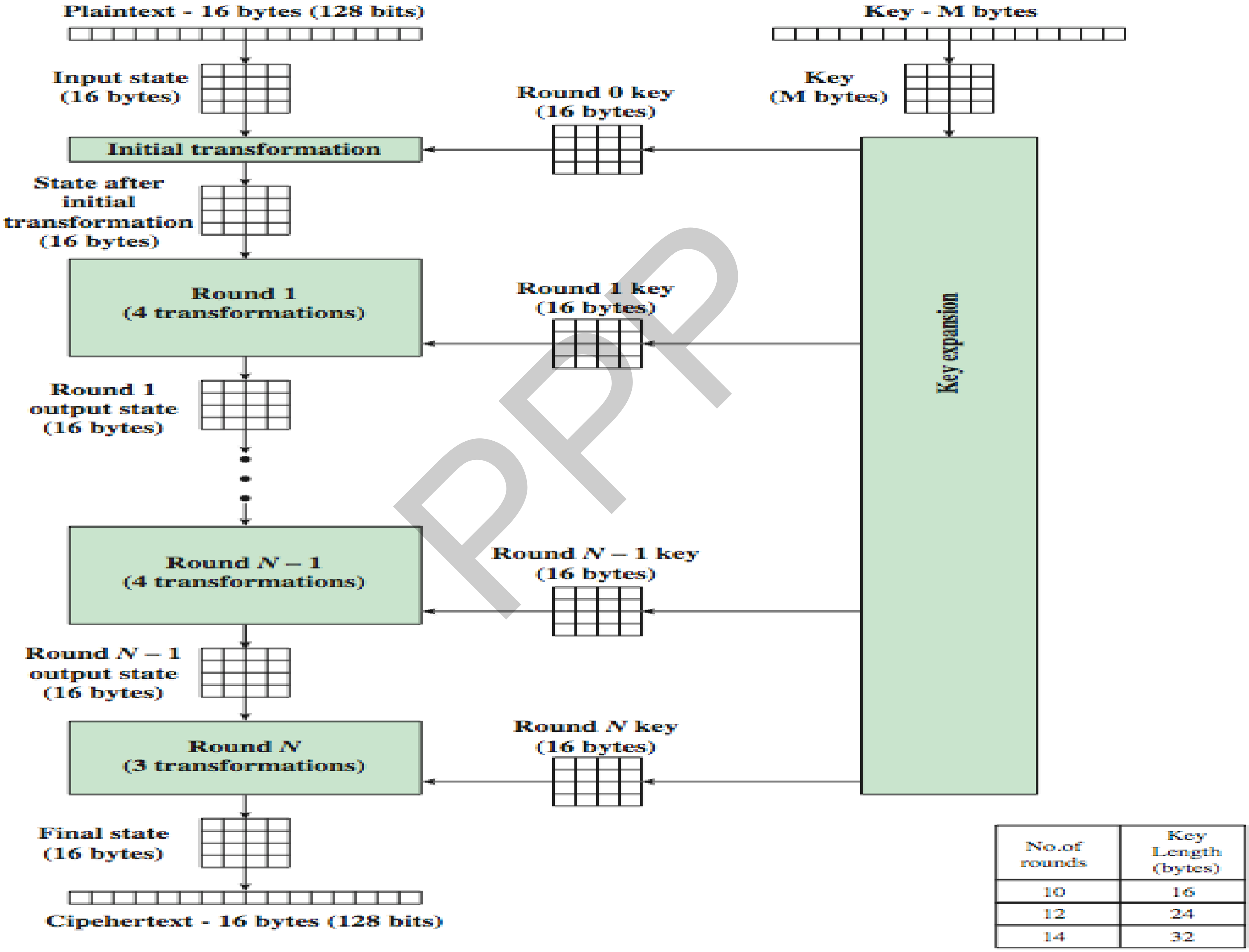
- **Rules for AES proposals**

1. The algorithm must be a symmetric block cipher.
2. The full design must be public.
3. Key lengths of 128, 192, and 256 bits supported.
4. Both software and hardware implementations required
5. The algorithm must be public or licensed on nondiscriminatory terms.

AES – The Advanced Encryption Standard

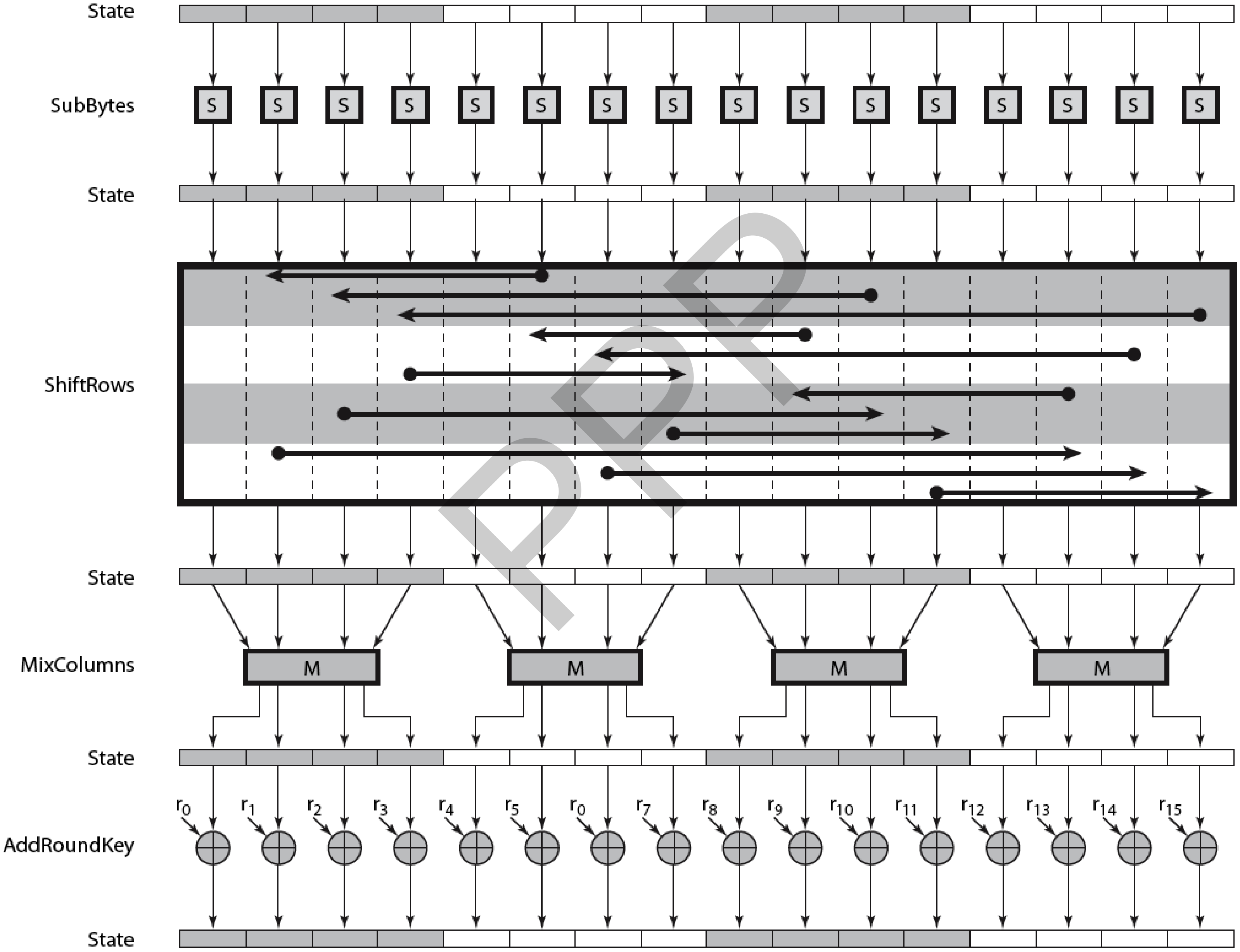
- Designed by Rijmen-Daemen in Belgium
- Key Length : 128/192/256 bit
- Data (Plain Text) : 128 bit
- Designed to have:
  - resistance against known attacks
  - speed and code compactness on many CPUs
  - design simplicity

AES – Encryption Process

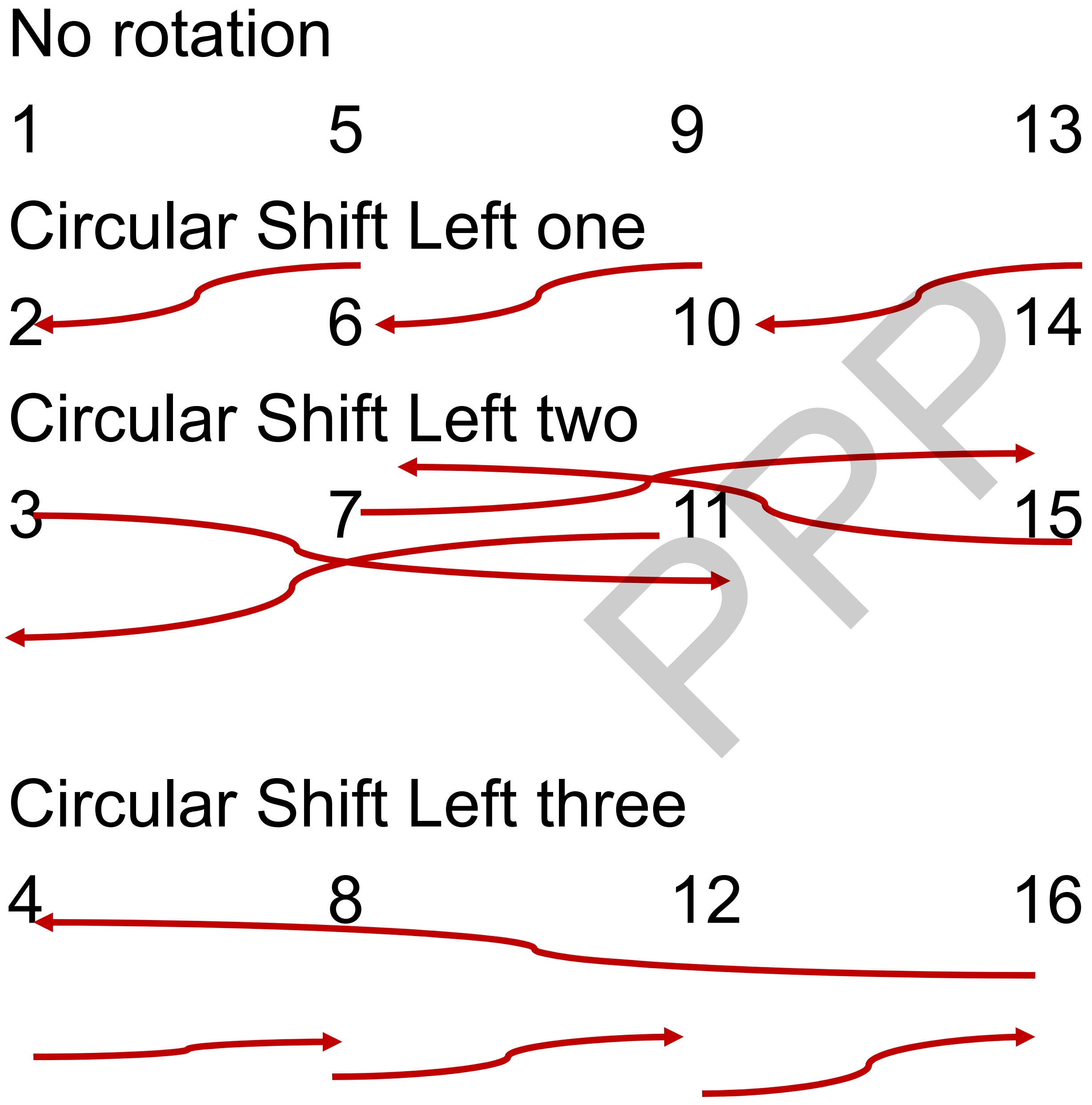

Plaintext - 16 bytes (128 bits)
Key - M bytes
Input state (16 bytes)
Key (M bytes)
Round 0 key (16 bytes)
Initial transformation
State after initial transformation (16 bytes)
Round 1 (4 transformations)
Round 1 key (16 bytes)
Key expansion
Round 1 output state (16 bytes)
Round N – 1 (4 transformations)
Round N – 1 key (16 bytes)
Round N – 1 output state (16 bytes)
Round N (3 transformations)
Round N key (16 bytes)
Final state (16 bytes)
Cipehertext - 16 bytes (128 bits)

No.of rounds	Key Length (bytes)
10	16
12	24
14	32

AES Round


State
SubBytes
S S S S S S S S S S S S S S S S
State
ShiftRows
State
MixColumns
M M M M
State
AddRoundKey
r_0 r_1 r_2 r_3 r_4 r_5 r_0 r_7 r_8 r_9 r_{10} r_{11} r_{12} r_{13} r_{14} r_{15}
State

AES Structure : Rotation Step



No rotation

1 5 9 13

Circular Shift Left one

2 6 10 14

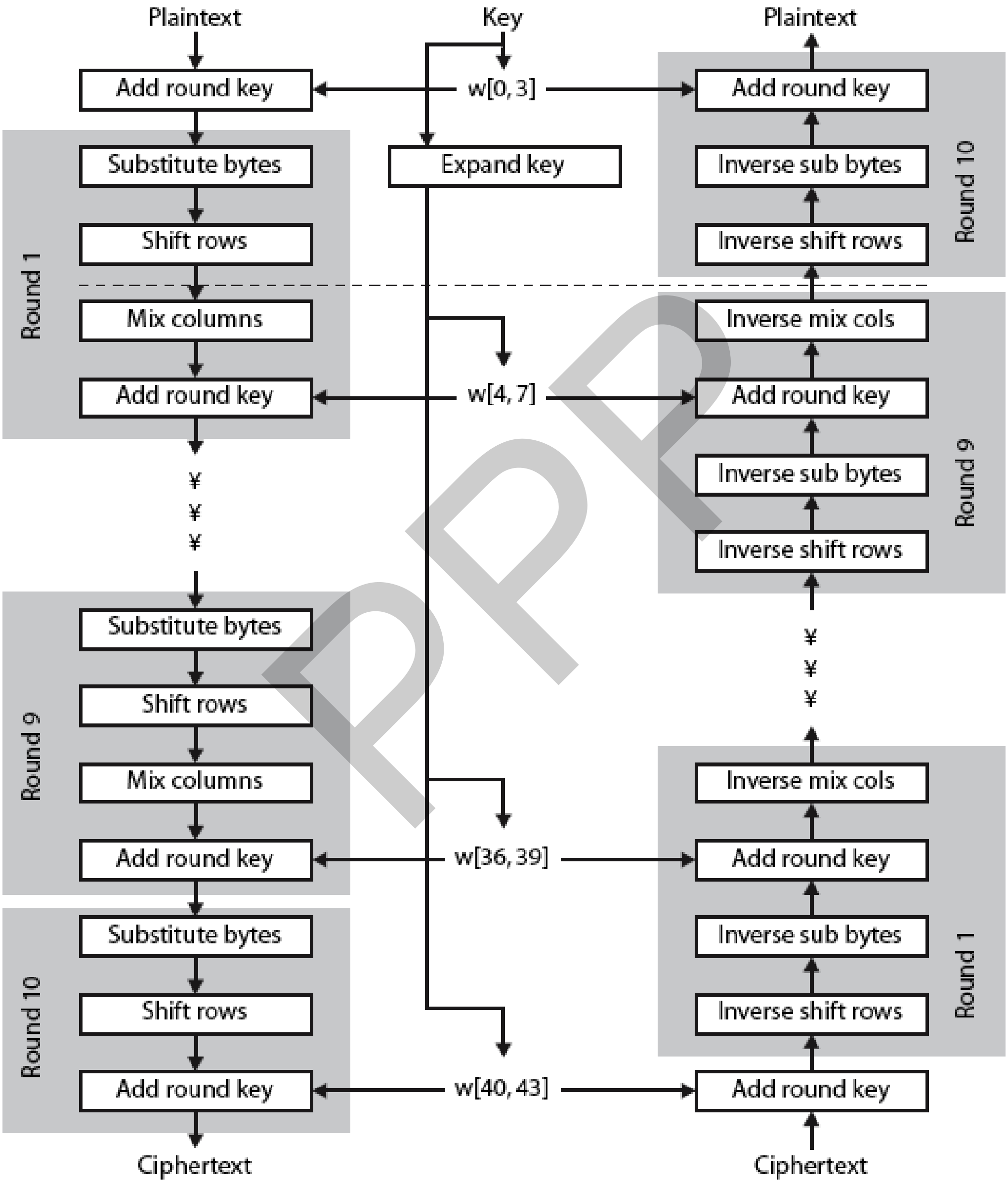
Circular Shift Left two

3 7 11 15

Circular Shift Left three

4 8 12 16

AES Structure



(a) Encryption

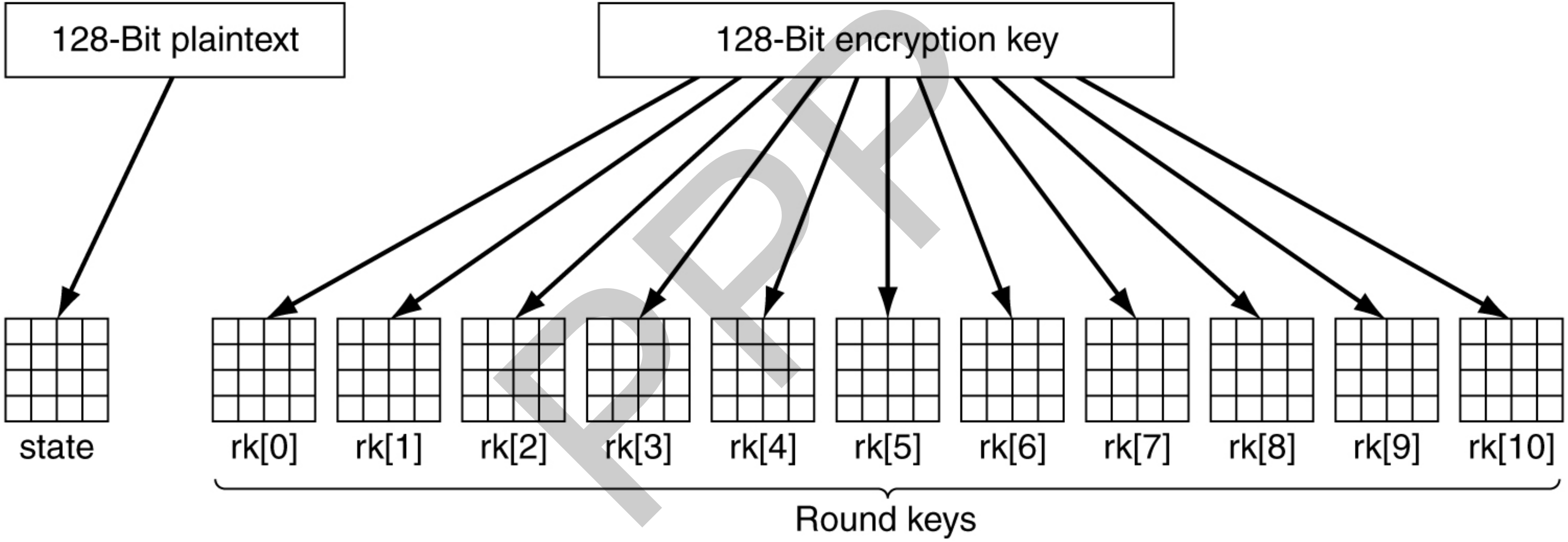
(b) Decryption

AES

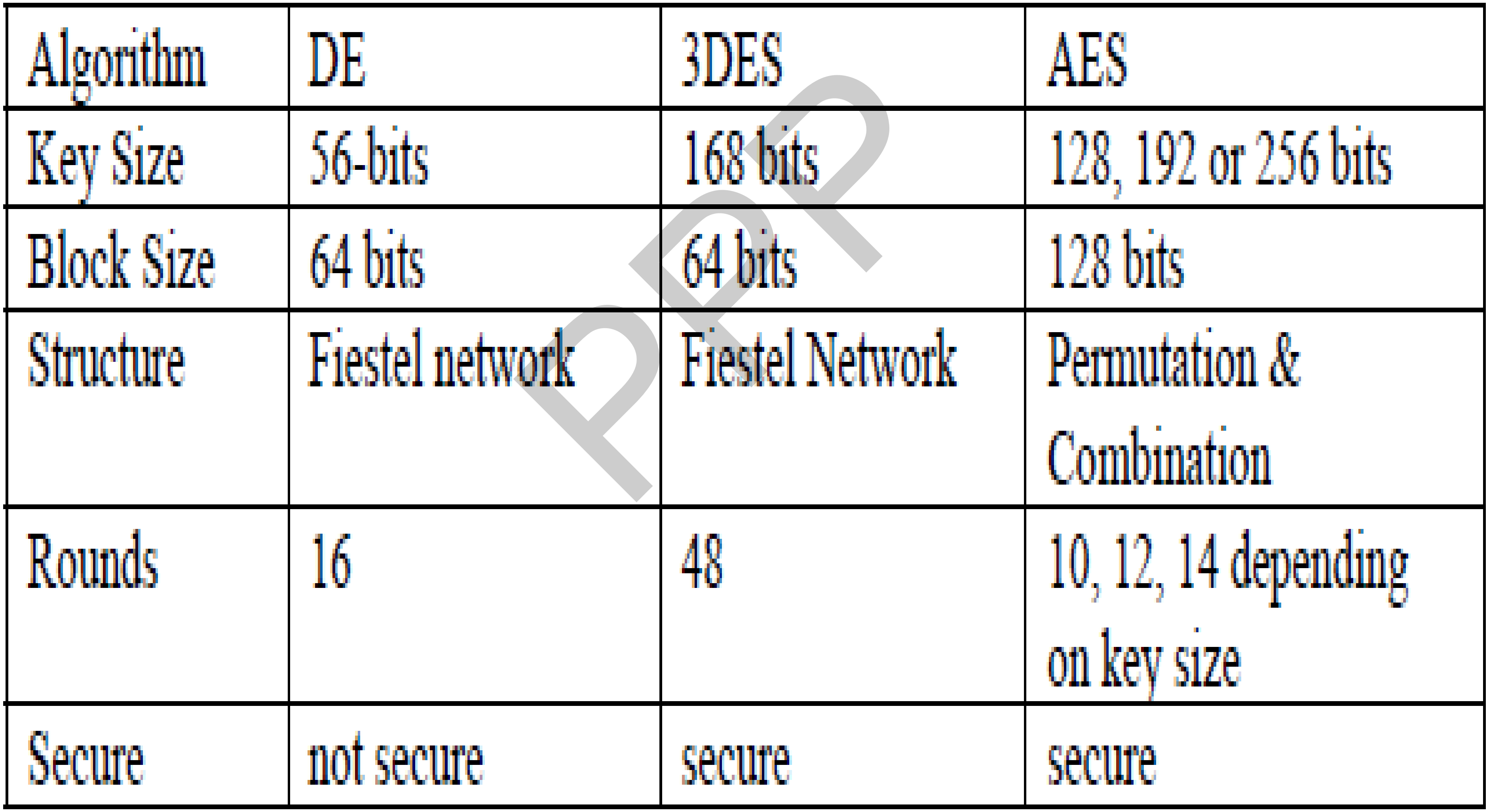
```
#define LENGTH 16                                   /* # bytes in data block or key */
#define NROWS 4                                     /* number of rows in state */
#define NCOLS 4                                     /* number of columns in state */
#define ROUNDS 10                                   /* number of iterations */
typedef unsigned char byte;                         /* unsigned 8-bit integer */

rijndael(byte plaintext[LENGTH], byte ciphertext[LENGTH], byte key[LENGTH])
{
 int r;                                             /* loop index */
 byte state[NROWS][NCOLS];                          /* current state */
 struct {byte k[NROWS][NCOLS];} rk[ROUNDS + 1];     /* round keys */

 expand_key(key, rk);                               /* construct the round keys */
 copy_plaintext_to_state(state, plaintext);         /* init current state */
 xor_roundkey_into_state(state, rk[0]);             /* XOR key into state */

 for (r = 1; r <= ROUNDS; r++) {
      substitute(state);                            /* apply S-box to each byte */
      rotate_rows(state);                           /* rotate row i by i bytes */
      if (r < ROUNDS) mix_columns(state);           /* mix function */
      xor_roundkey_into_state(state, rk[r]);        /* XOR key into state */
 }
 copy_state_to_ciphertext(ciphertext, state);       /* return result */
}
```

AES (3)

- Creating of the *state* and *rk* arrays.



Difference between DES, 3DES and AES



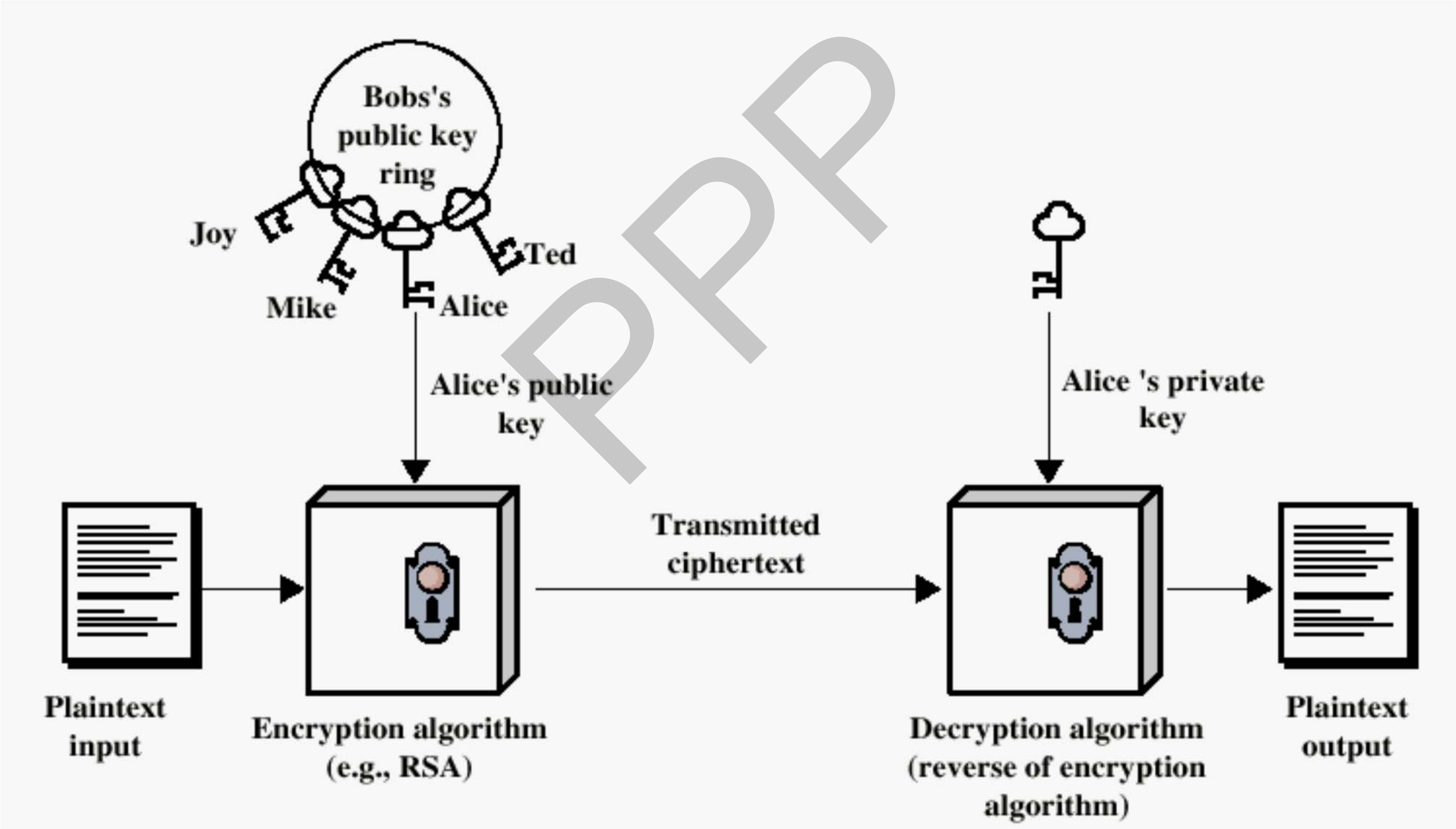
Algorithm	DE	3DES	AES
Key Size	56-bits	168 bits	128, 192 or 256 bits
Block Size	64 bits	64 bits	128 bits
Structure	Fiestel network	Fiestel Network	Permutation & Combination
Rounds	16	48	10, 12, 14 depending on key size
Secure	not secure	secure	secure

Public-Key Cryptography Principles

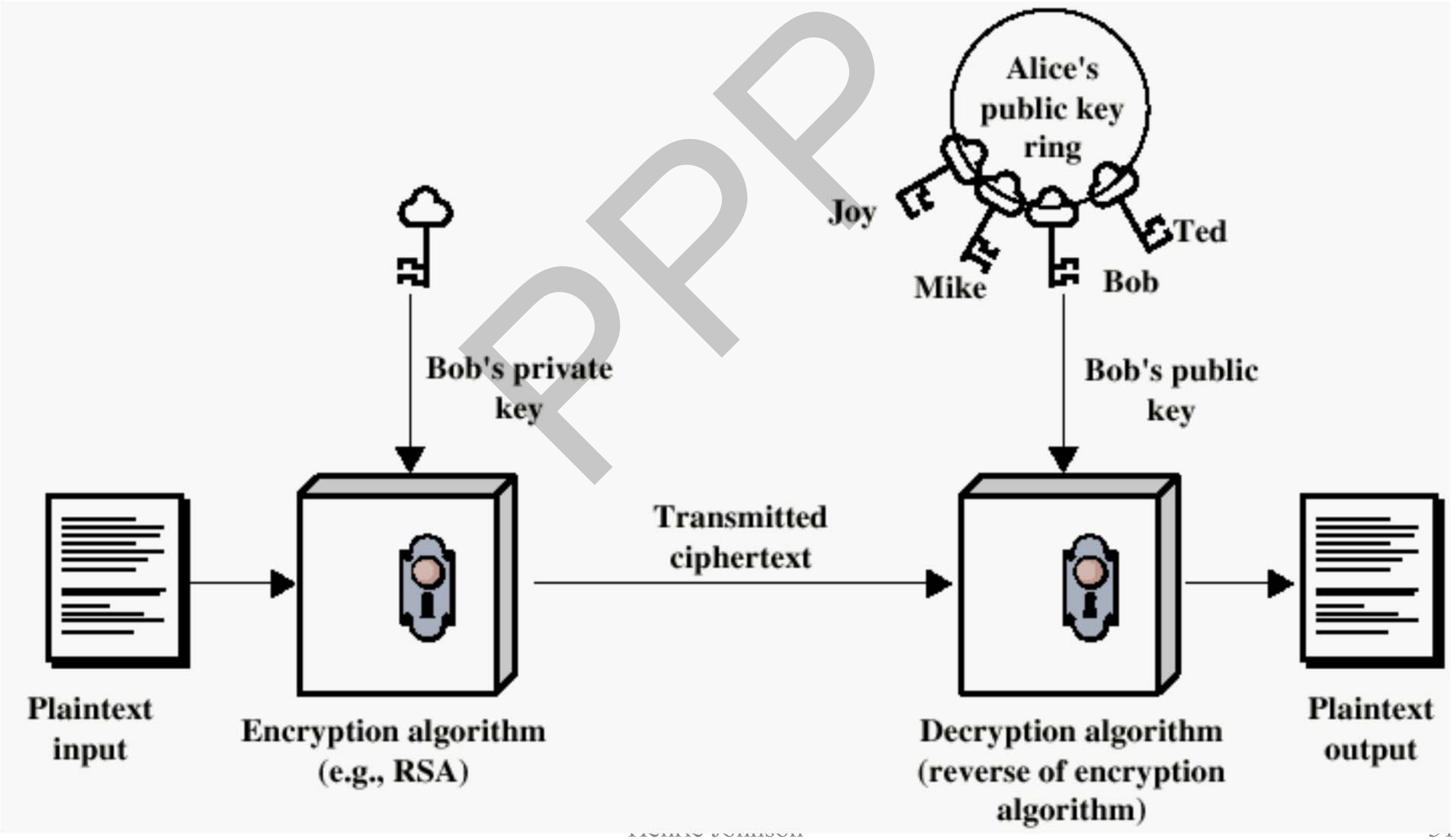
- The use of two keys has consequences in: key distribution, confidentiality and authentication.

- The scheme has six ingredients
  - Plaintext
  - Encryption algorithm
  - Public key
  - Private key
  - Ciphertext
  - Decryption algorithm

Encryption using Public-Key system



Authentication using Public-Key System



Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p-1)(q-1)$	
Select integer e	$\gcd(\phi(n), e) = 1;\ 1 < e < \phi(n)$
Calculate d	$de \bmod \phi(n) = 1$
Public key	$KU = \{e, n\}$
Private key	$KR = \{d, n\}$

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \pmod{n}$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \pmod{n}$

Figure 3.8 The RSA Algorithm

p = 17 q = 11

n = p * q = 187

Ø(n) = (p-1) * (q-1) = 160

e = 7 d = 23

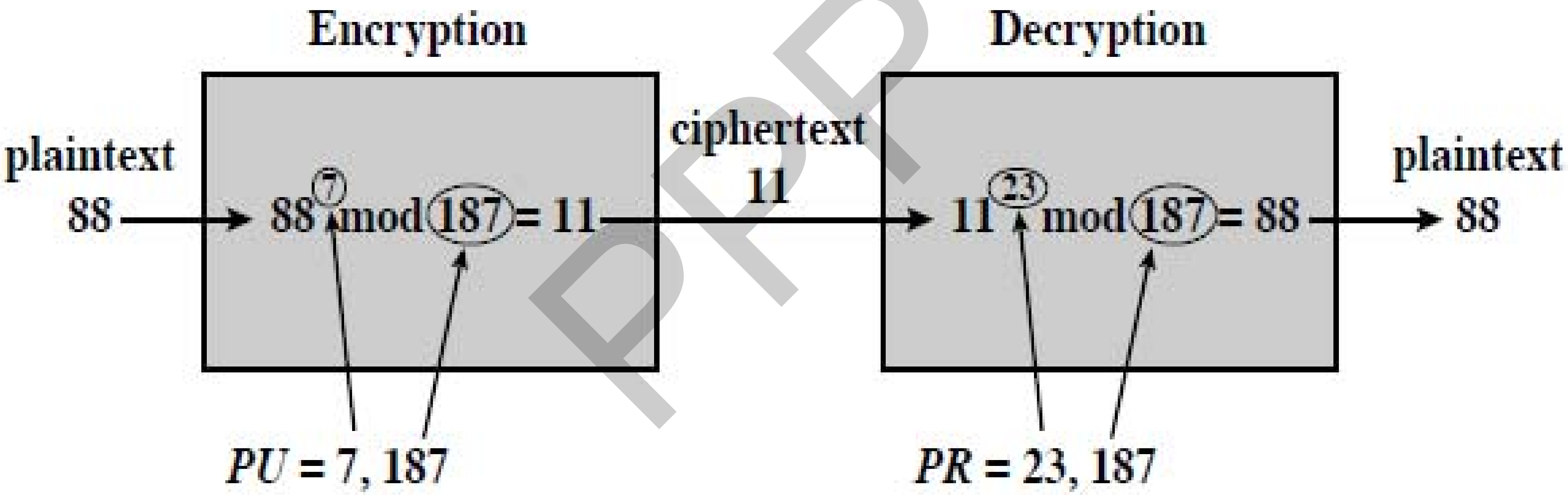


Figure 3.9 Example of RSA Algorithm

P=3 & q =11 ,
n = p * q = 33
Ø(n) = (p – 1) * (q – 1) = 20
d = 7
Select e such that d * e mod Ø(n) = 1 i.e. 7 * e mod 20
e =3
$C = p^{e} \text{ mod } n$
$P = c^{d} \text{ mod } n$

Plaintext (P)			Ciphertext (C)		After decryption	
Symbolic	Numeric	P^3	P^3 (mod 33)	C^7	C^7 (mod 33)	Symbolic
S	19	6859	28	13492928512	19	S
U	21	9261	21	1801088541	21	U
Z	26	17576	20	1280000000	26	Z
A	01	1	1	1	01	A
N	14	2744	5	78125	14	N
N	14	2744	5	78125	14	N
E	05	125	26	8031810176	05	E

Sender's computation

Receiver's computation

An example of the RSA algorithm.

Why RSA Key Unbreakable?

- With latest algorithm with 1 µsec instruction time to find a prime factors of a no. with 500 digits takes 10^{25} years.

- With a million chips running in parallel, each with an instruction time of 1 nsec, it would still takes 10^{16} years.

- If p & q are of 512 bit (64 byte),

  then n is of 1024 bit (128 byte). &

  No. of digits in n is 300. It would still takes 10^{8} years

Digital Signature

To send a signed message from one party to another party the following conditions should be meet.

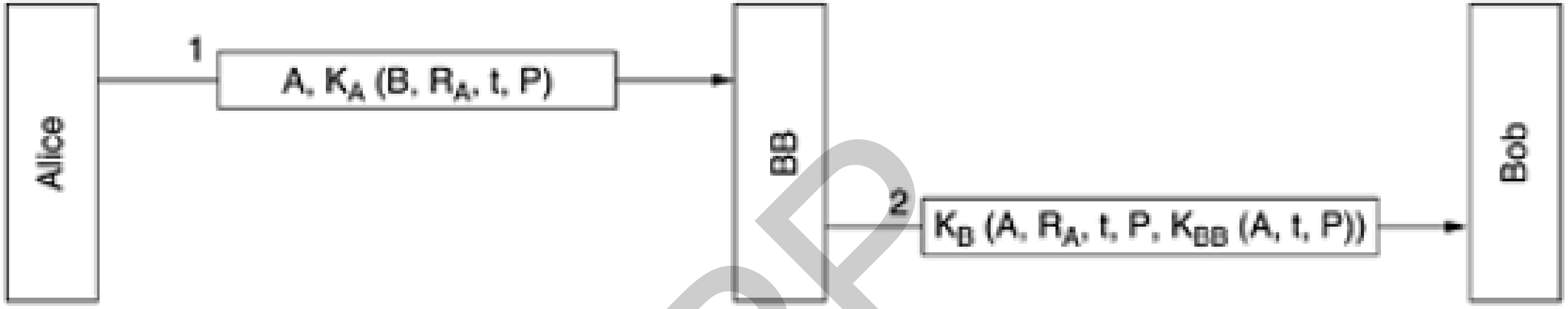
- The receive can verify the claimed identity of the sender.
- The sender cannot later deny the contents of the message.
- The receiver cannot possibly have alter the message himself/herself

Two Types Signatures:

- Symmetric key signature
- Public key signature

Symmetric-Key Signatures

Figure 8-18. Digital signatures with Big Brother.


Alice
1
A, K_A (B, R_A, t, P)
BB
2
K_B (A, R_A, t, P, K_{BB} (A, t, P))
Bob

- A – Sender (Alice) BB – Intermediator (Big Brother) B – Receiver (Bob)
- P – Plain Text Ra – Random Number t – timestamps

- K_A Known by only A & BB
- K_B Known by only B & BB
- K_{BB} is only known to the BB.

•Why K_{BB} (A , t , P) send by BB to Bob?

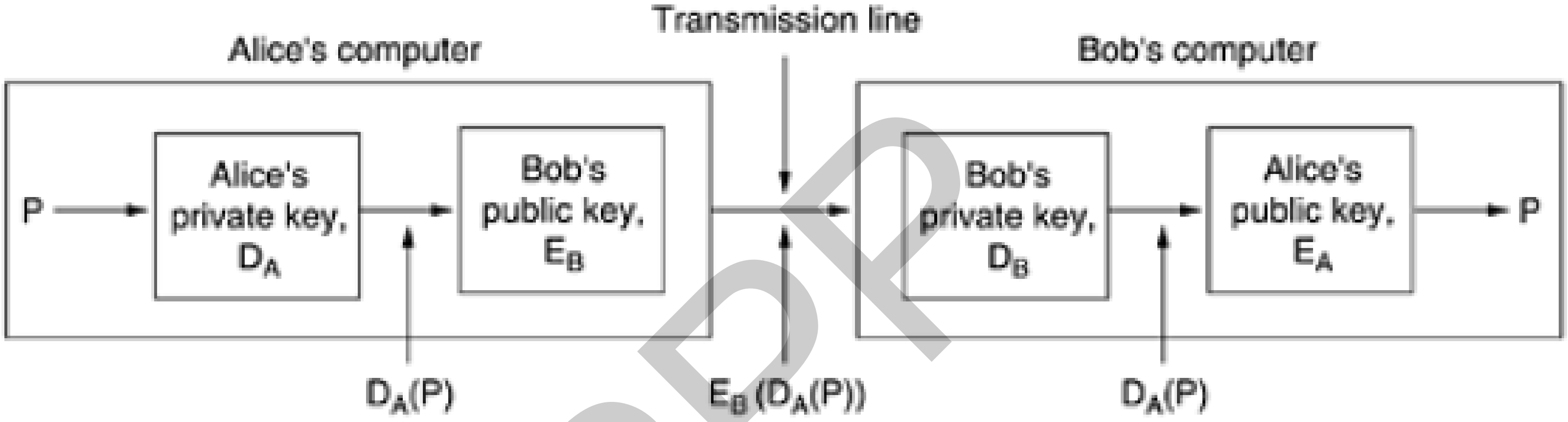
- If any dispute in future than K_{BB} (A , t , P) is used for verification.

•Trudy (Intruder) replaying message, So Ra is used to check freshness of message.

- **Disadvantage :** Everyone has to trust on BB.
  BB gets to read all signed messages.

Public-Key Signatures

Digital signatures using public-key cryptography.



- **Disadvantage :**
  - Alice runs to the police claiming that PC holding her key was stolen.
  - If Alice decides to change her key immediately.
  - Sometimes only authentication is needed but not secrecy.

Message Digest

Message Digest is used to ensure the integrity of a **message** transmitted over an insecure channel.

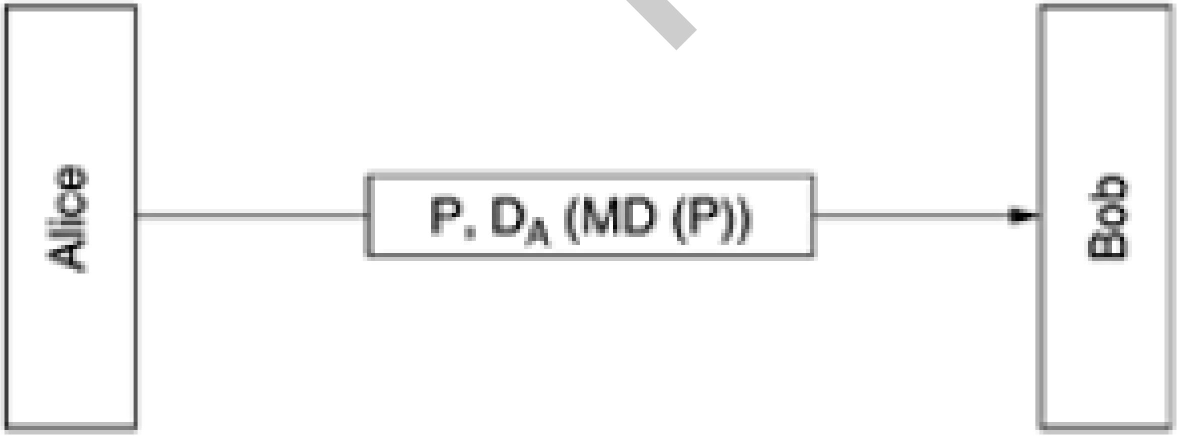
The **message** is passed through a Cryptographic **hash** function. MD5, SHA-1, etc.

This function creates a compressed image of the **message** called **Digest**.

Message Digest Properties:

- Given P, it is easy to compute MD (P)
- Given MD (P) , it is effectively impossible to find P.
- Given P no one can find P’ such that MD (P ‘) = MD (P)
- A change to the input of even 1 bit produces a very different output.

Digital signatures using message digests.



Message Digest Algorithm (SHA)

- Steps for SHA Algorithm
  - Step 1 Append padding bits
  - Step 2 Append length
  - Step 3 Initialize hash buffer
  - Step 4 Process message
  - Step 5 Output

SHA-1 for Message Digest

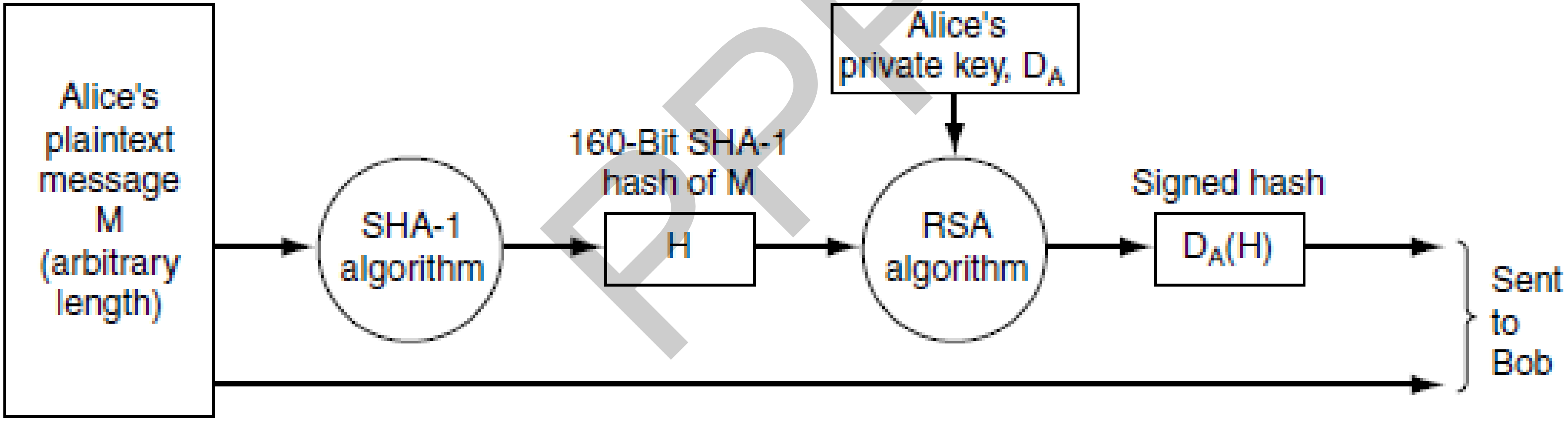


Figure 8-21. Use of SHA-1 and RSA for signing nonsecret messages.

SHA-1 for Message Digest

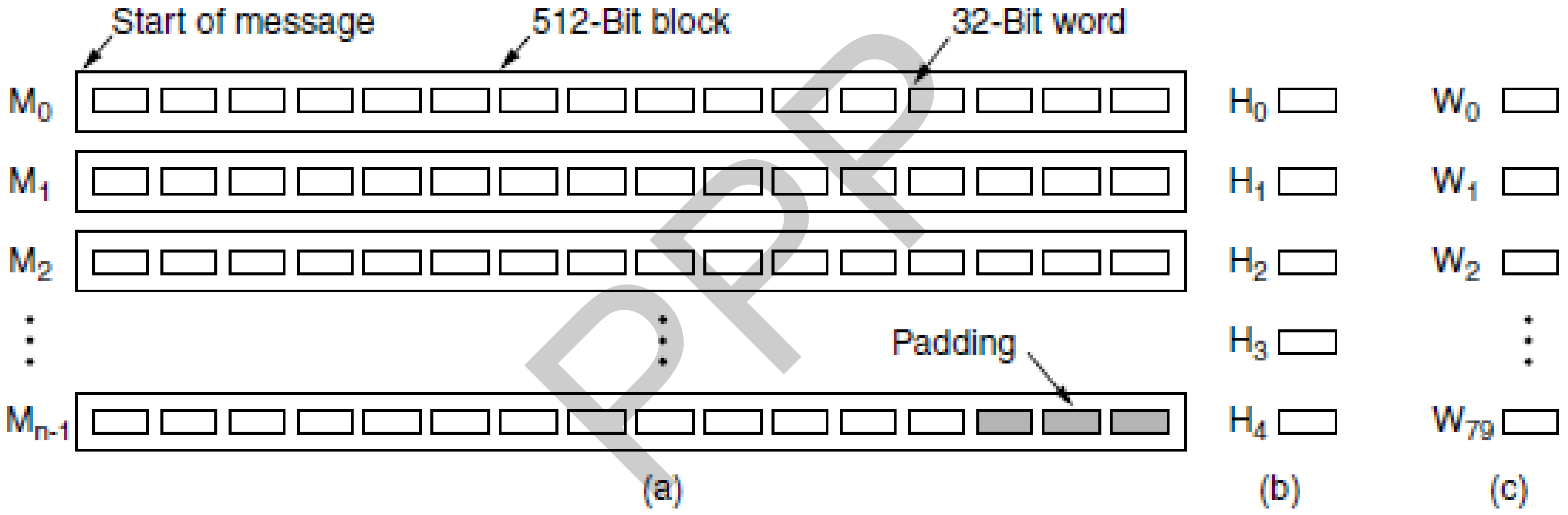


Figure 8-22. (a) A message padded out to a multiple of 512 bits. (b) The output variables. (c) The word array.

Hash Output (160 Bits) : H0 to H4 are five 32 bit variables.
W0 to W79 are eighty 32 bit variables

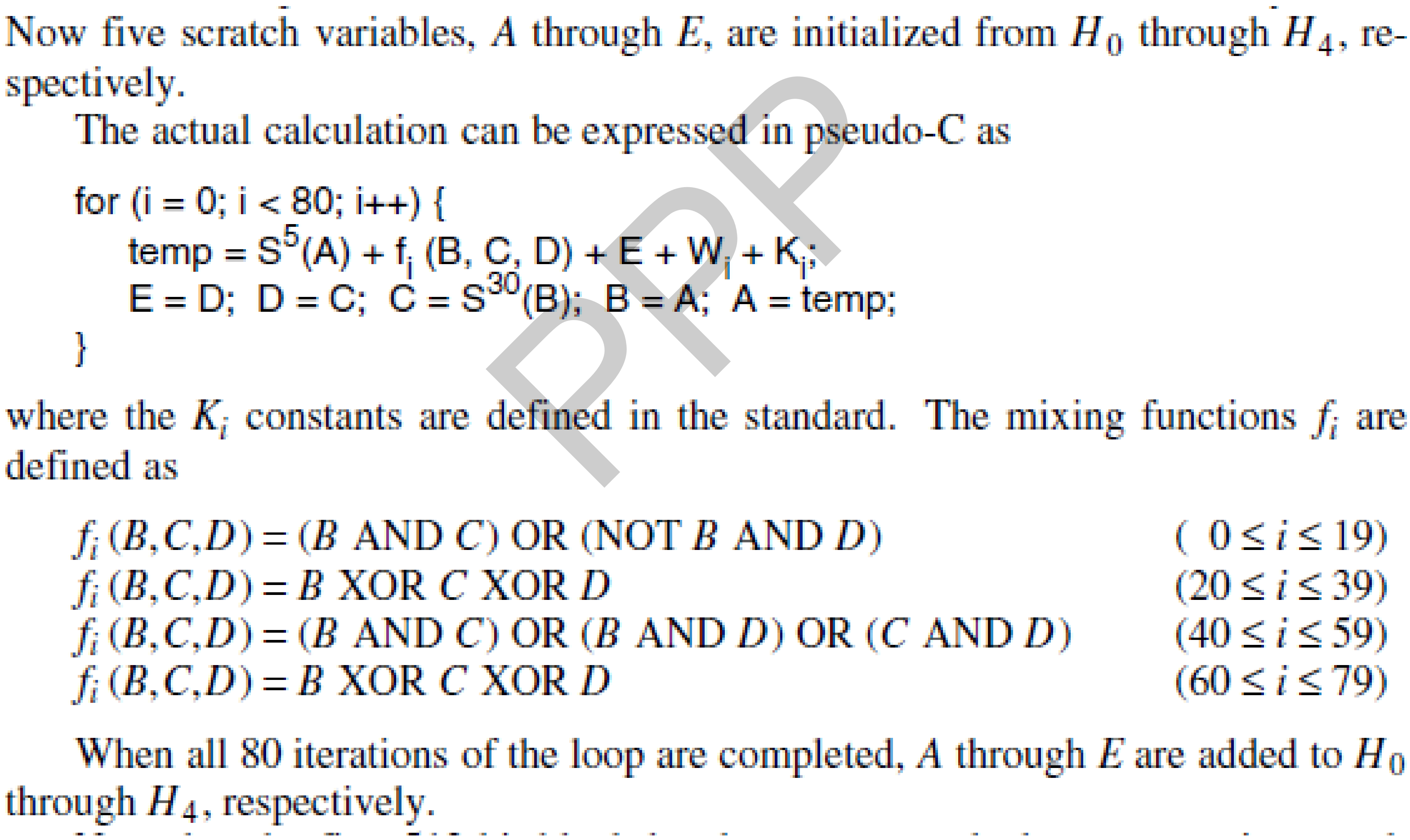
Calculation of W_0…. W_{79}

- ***Mi (512/32 = 16 words) are copied into*** W_0…. W_{15}
- Then the other 64 words in *W (*W_{16}…. W_{79} *)are filled in* using the formula

$$W_i = S^1(W_{i-3} \text{ XOR } W_{i-8} \text{ XOR } W_{i-14} \text{ XOR } W_{i-16}) \quad (16 \le i \le 79)$$

  - Where **S^b*(W)*** *represents the* ***left circular rotation*** *of the 32-bit word, W,* ***by b bits****.*

SHA-1
Calculation For First Block (M_0 – 512 bit)



Now five scratch variables, A through E, are initialized from H_0 through H_4, respectively.

The actual calculation can be expressed in pseudo-C as

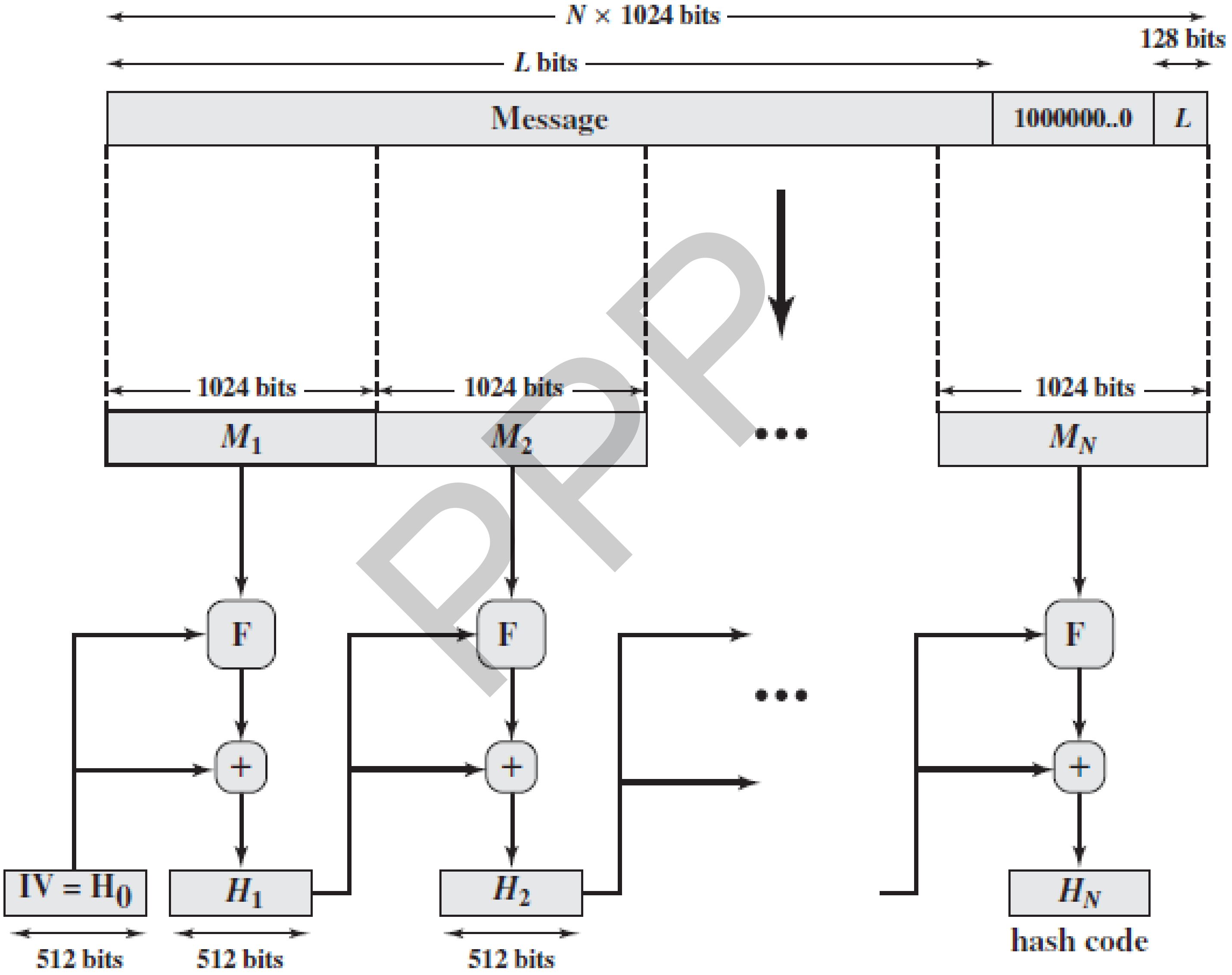
```
for (i = 0; i < 80; i++) {
    temp = $S^5(A) + f_i\ (B, C, D) + E + W_i + K_i$;
    E = D;  D = C;  C = $S^{30}(B)$;  B = A;  A = temp;
}
```

where the K_i constants are defined in the standard. The mixing functions f_i are defined as

$$f_i\,(B,C,D) = (B \text{ AND } C) \text{ OR } (\text{NOT } B \text{ AND } D) \qquad (\ 0 \le i \le 19)$$
$$f_i\,(B,C,D) = B \text{ XOR } C \text{ XOR } D \qquad (20 \le i \le 39)$$
$$f_i\,(B,C,D) = (B \text{ AND } C) \text{ OR } (B \text{ AND } D) \text{ OR } (C \text{ AND } D) \qquad (40 \le i \le 59)$$
$$f_i\,(B,C,D) = B \text{ XOR } C \text{ XOR } D \qquad (60 \le i \le 79)$$

When all 80 iterations of the loop are completed, A through E are added to H_0 through H_4, respectively.

SHA-512



$+$ = word-by-word addition mod 2^{64}

Figure 3.4 Message Digest Generation Using SHA-512

SHA-512

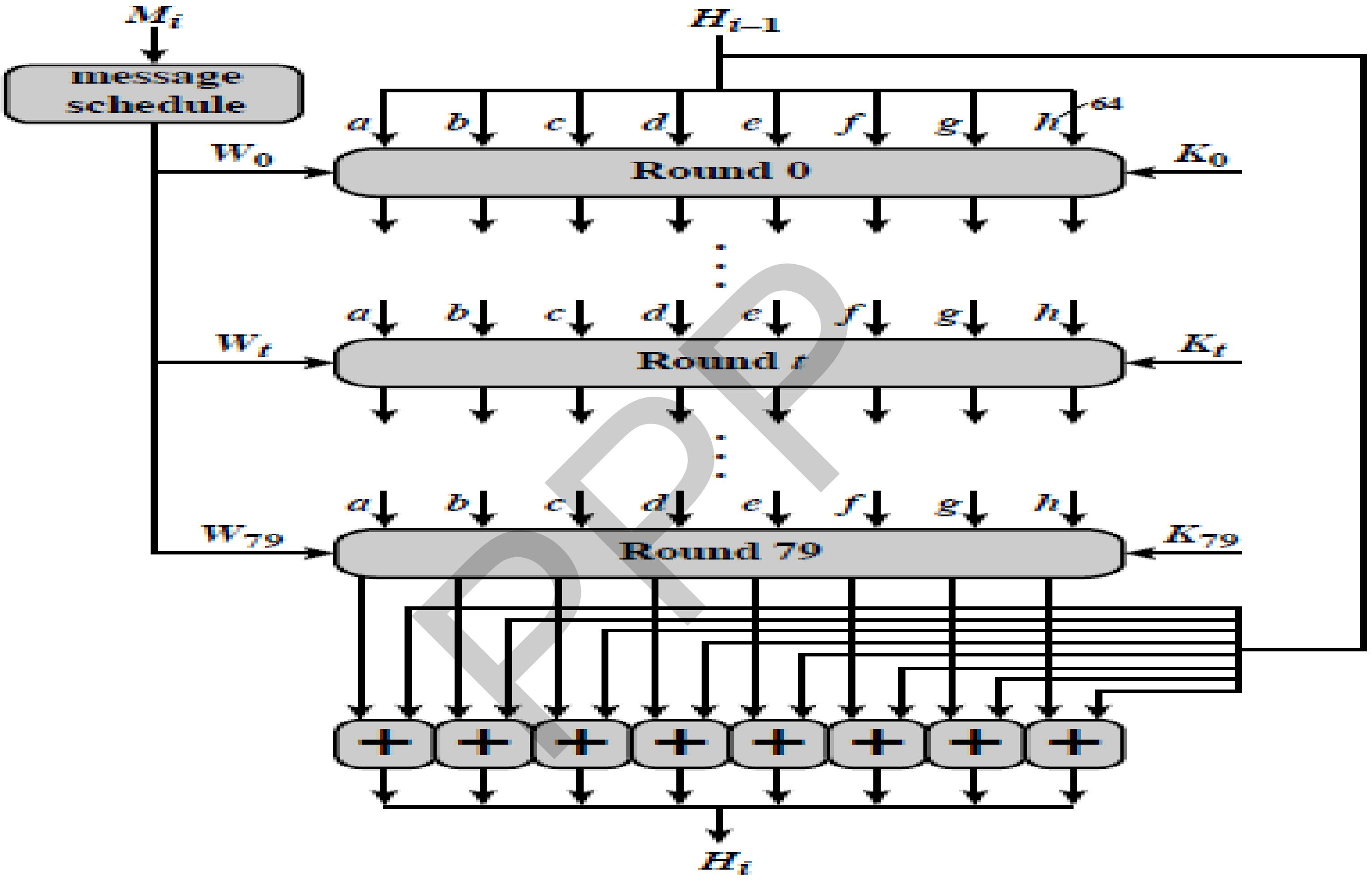


Figure 3.5 SHA-512 Processing of a Single 1024-Bit Block

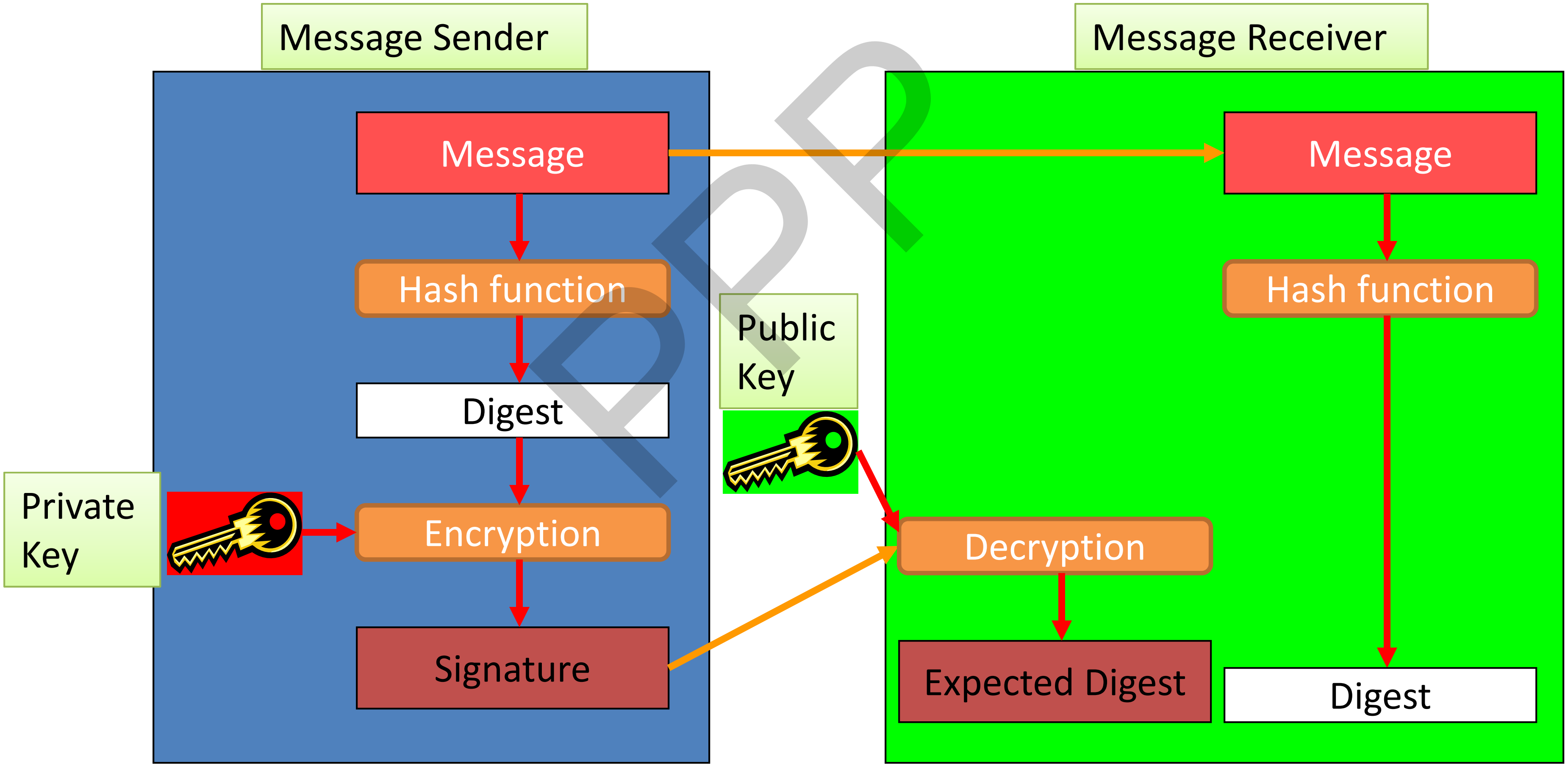
M : 1024 bit
W : 64 bit (Derived from M)
K : 64 bit (Fractional part of cube root of frist 80 prime numbers.)
H : 512 bit (Intermediate and final output)

Comparison of SHA Parameters

	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	64	80	80
Security	80	128	192	256

Notes: 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 $2^{n/2}$.

Digital Signature Generation and Verification



Digital Certificates

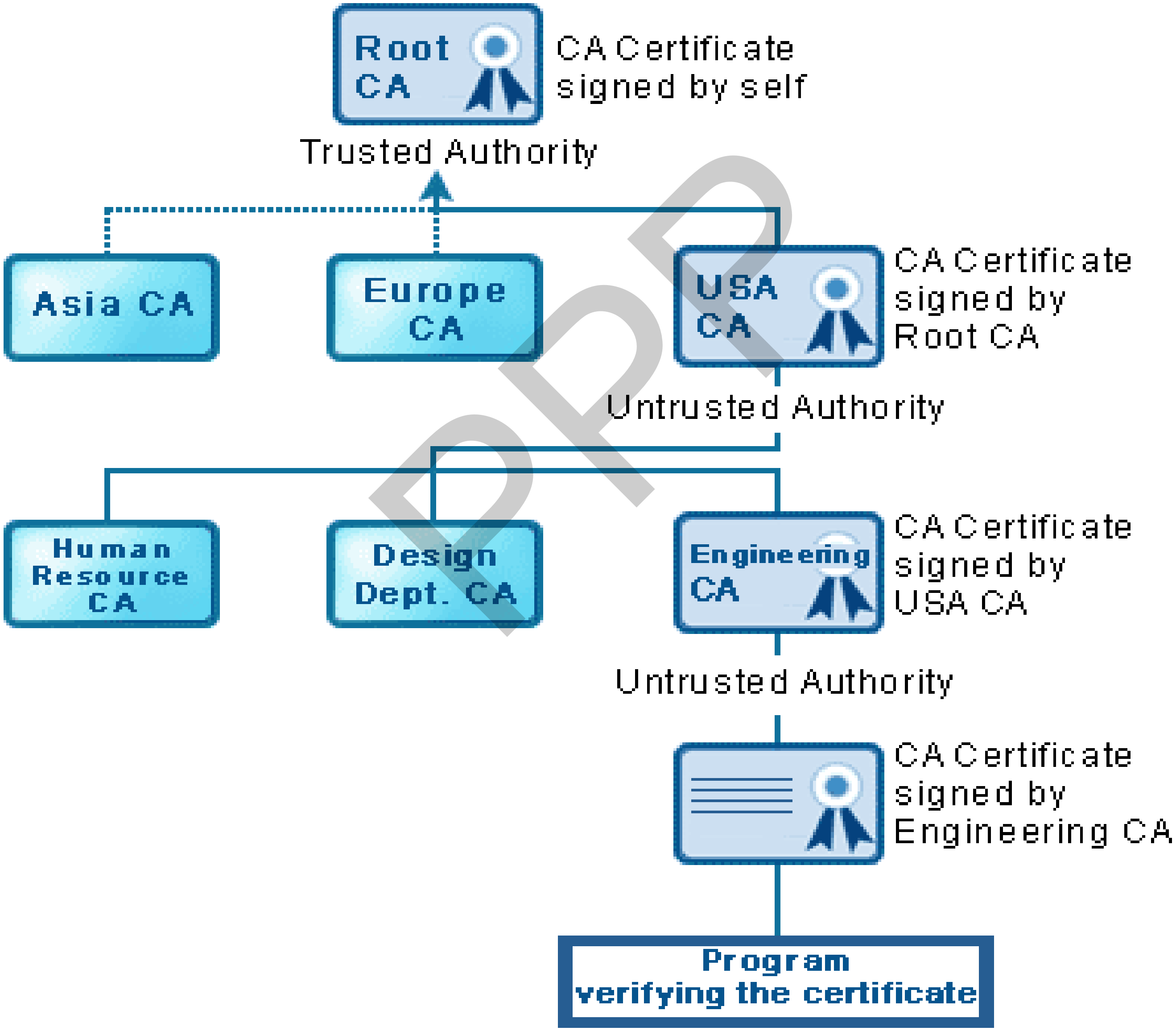
- When you conduct business online whether it's selling products through a Web site or simply using email for company communication the business is not face to face.
- To address this risk, digital certificates were created. If you use the Web to transact business or communicate sensitive information with clients, then digital certificates are a must.
- In a public key environment, it is critical that you are assured that the public key to which you are encrypting data is in fact the public key of the intended recipient and not a fake.
- You could encrypt to those keys, which have been physically handed to you. But suppose you need to exchange information with people you have never met; how can you tell that you have the correct key?
- Digital certificates, simplify the task of establishing whether a public key truly belongs to the purported owner.
- A certificate is a form of credential. Examples might be your driving license, your passport, or your birth certificate.

Digital Certificates

- We need an independent third party to verify the person’s identity (through non-electronic means) and issue a **digital certificate**
- A digital certificate is information included with a person's public key that helps others verify that a key is genuine or valid.
- **Digital Certificate Contents:**
  - Name of holder
  - Public key of holder
  - Name of trusted third party (certificate authority)
  - DIGITAL SIGNATURE OF CERTIFICATE AUTHORITY
  - Data on which hash and public-key algorithms have been used
  - Other business or personal information
- **Certificate Authority:**
  - Digital certificates are generated and themselves digitally signed by organizations known as certificate authorities.
  - It is the job of a certificate authority to verify the identity of the person requesting a digital certificate before issuing one to them.

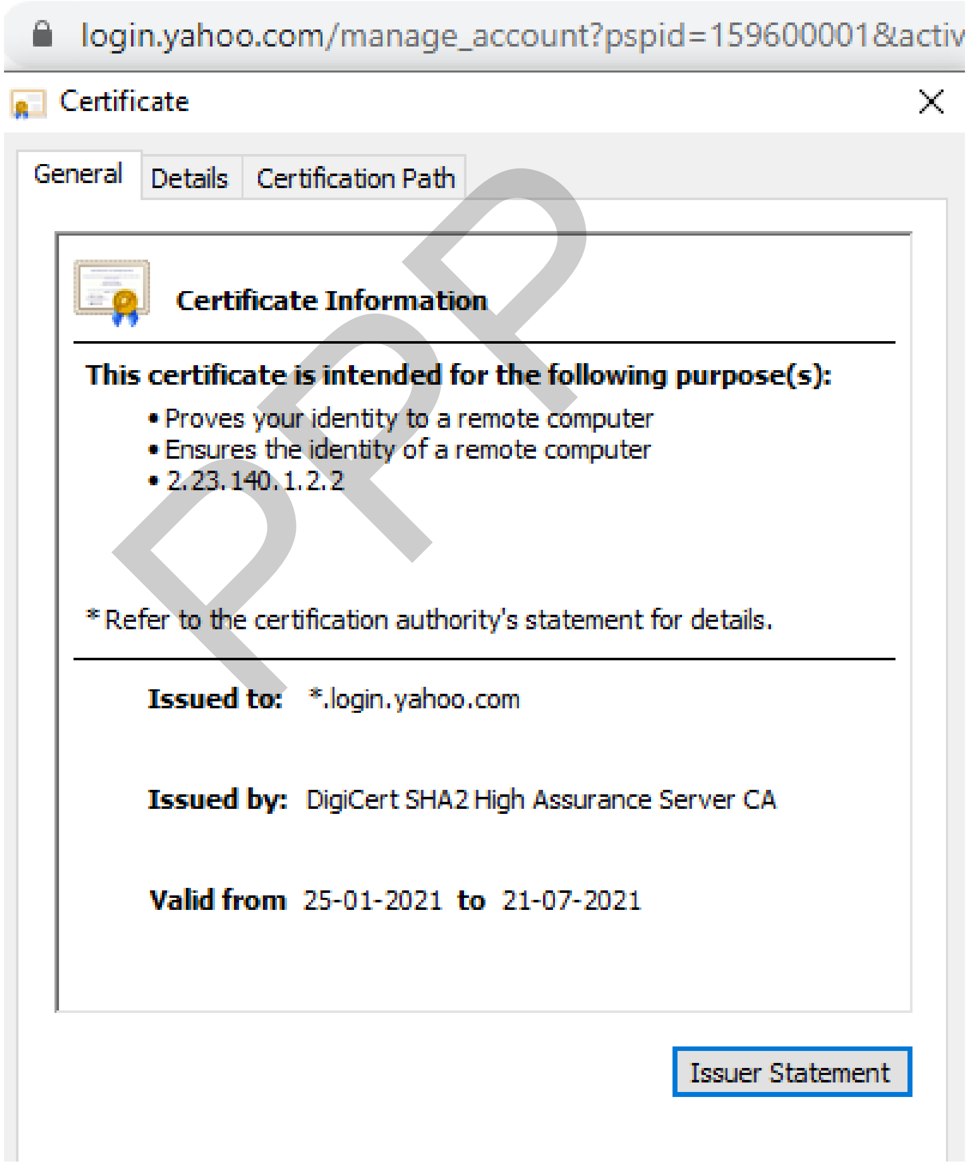
CA (Certificate Authority)

Hierarchies


Root CA
CA Certificate signed by self
Trusted Authority
Asia CA
Europe CA
USA CA
CA Certificate signed by Root CA
Untrusted Authority
Human Resource CA
Design Dept. CA
Engineering CA
CA Certificate signed by USA CA
Untrusted Authority
CA Certificate signed by Engineering CA
Program verifying the certificate

Digital Certificate :

E.g. www.mail.yahoo.com



login.yahoo.com/manage_account?pspid=159600001&activ

Certificate Information

This certificate is intended for the following purpose(s):

- Proves your identity to a remote computer
- Ensures the identity of a remote computer
- 2.23.140.1.2.2

* Refer to the certification authority's statement for details.

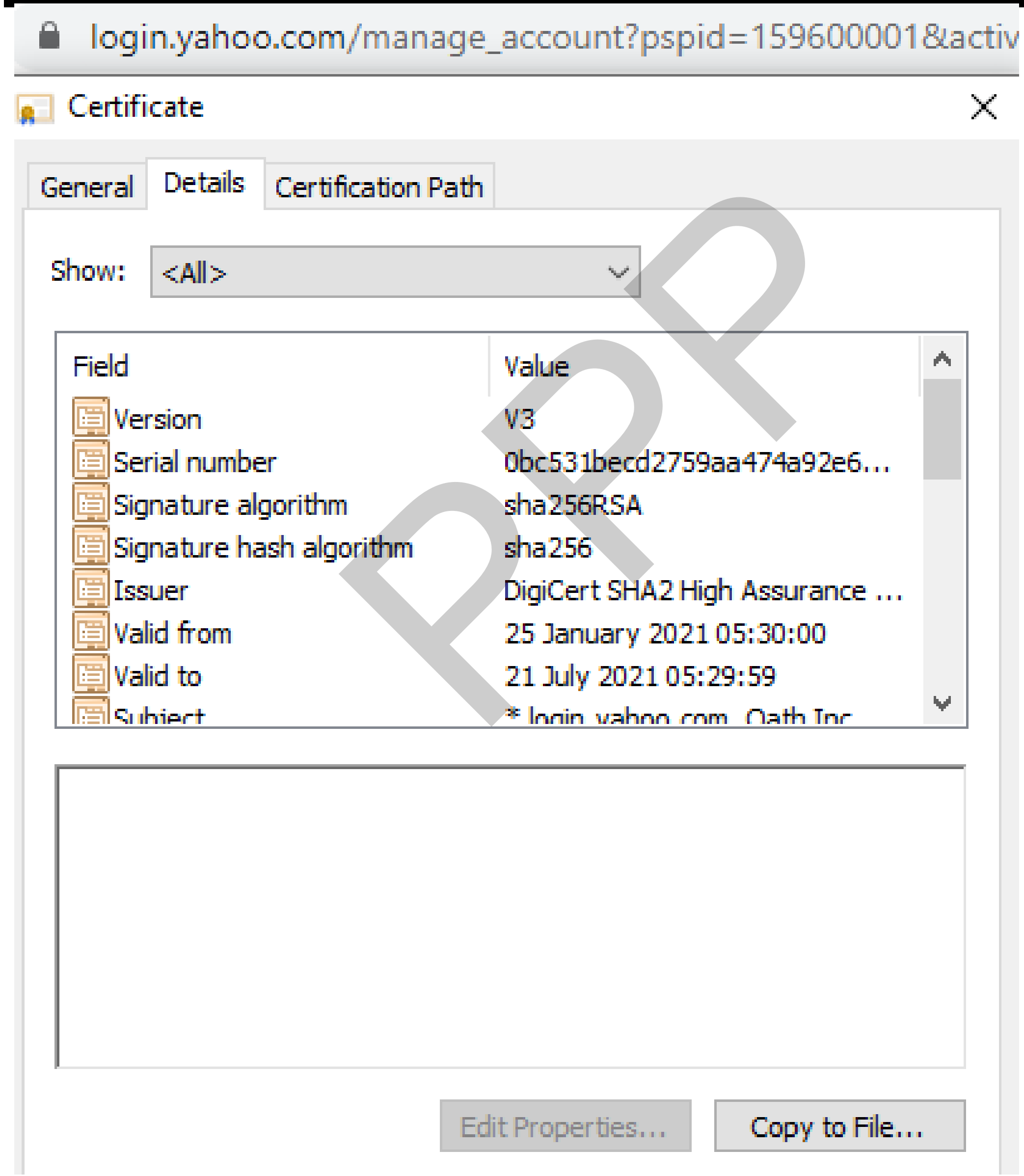
Issued to: *.login.yahoo.com

Issued by: DigiCert SHA2 High Assurance Server CA

Valid from 25-01-2021 **to** 21-07-2021

Digital Certificate :

E.g. www.mail.yahoo.com



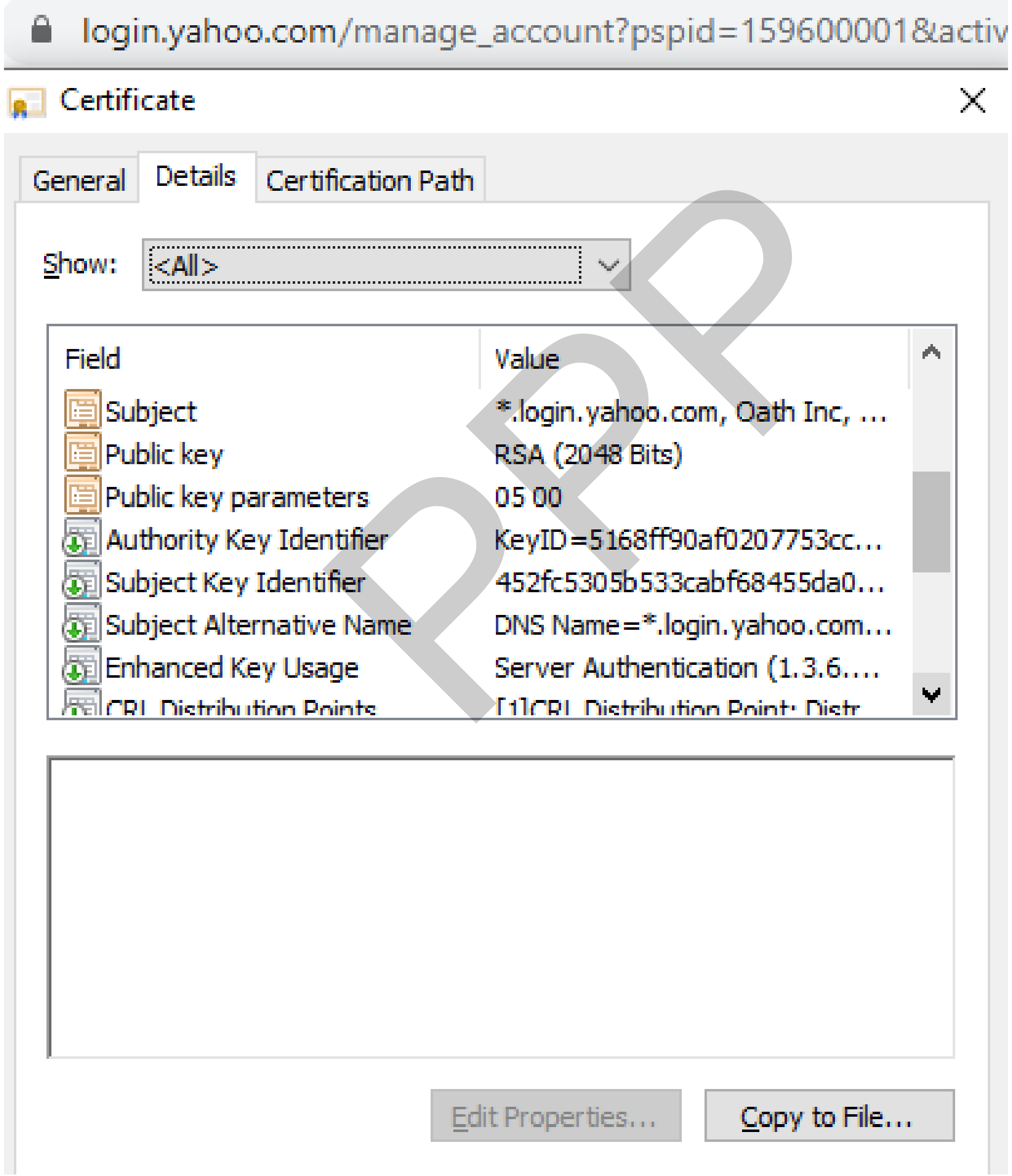
login.yahoo.com/manage_account?pspid=159600001&activ

Show: <All>

Field	Value
Version	V3
Serial number	0bc531becd2759aa474a92e6...
Signature algorithm	sha256RSA
Signature hash algorithm	sha256
Issuer	DigiCert SHA2 High Assurance ...
Valid from	25 January 2021 05:30:00
Valid to	21 July 2021 05:29:59
Subject	*.login.yahoo.com, Oath Inc

Digital Certificate :

E.g. www.mail.yahoo.com



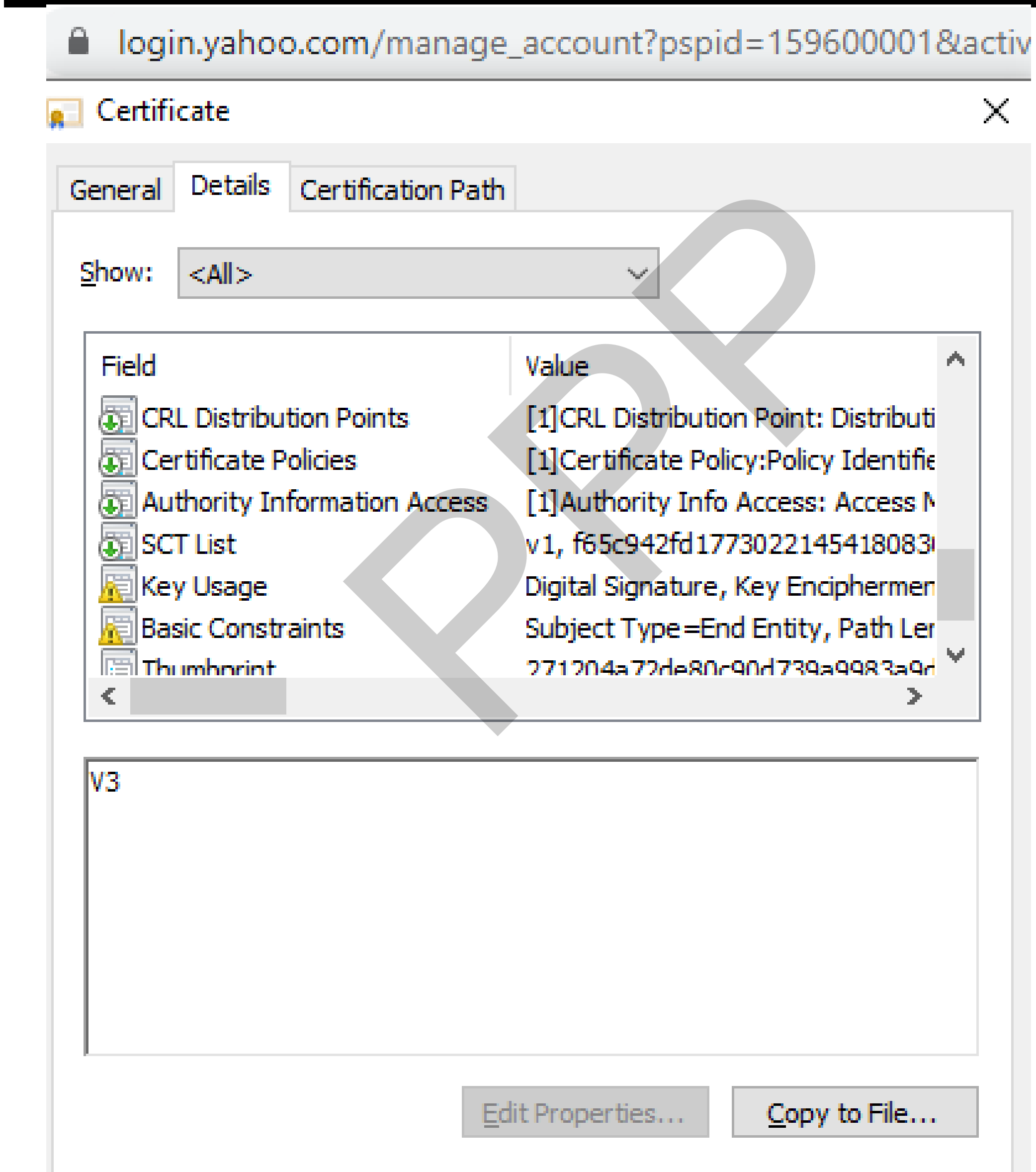
login.yahoo.com/manage_account?pspid=159600001&activ

Show: <All>

Field	Value
Subject	*.login.yahoo.com, Oath Inc, ...
Public key	RSA (2048 Bits)
Public key parameters	05 00
Authority Key Identifier	KeyID=5168ff90af0207753cc...
Subject Key Identifier	452fc5305b533cabf68455da0...
Subject Alternative Name	DNS Name=*.login.yahoo.com...
Enhanced Key Usage	Server Authentication (1.3.6....
CRL Distribution Points	[1]CRL Distribution Point: Distr

Digital Certificate :

E.g. www.mail.yahoo.com



login.yahoo.com/manage_account?pspid=159600001&activ

Certificate

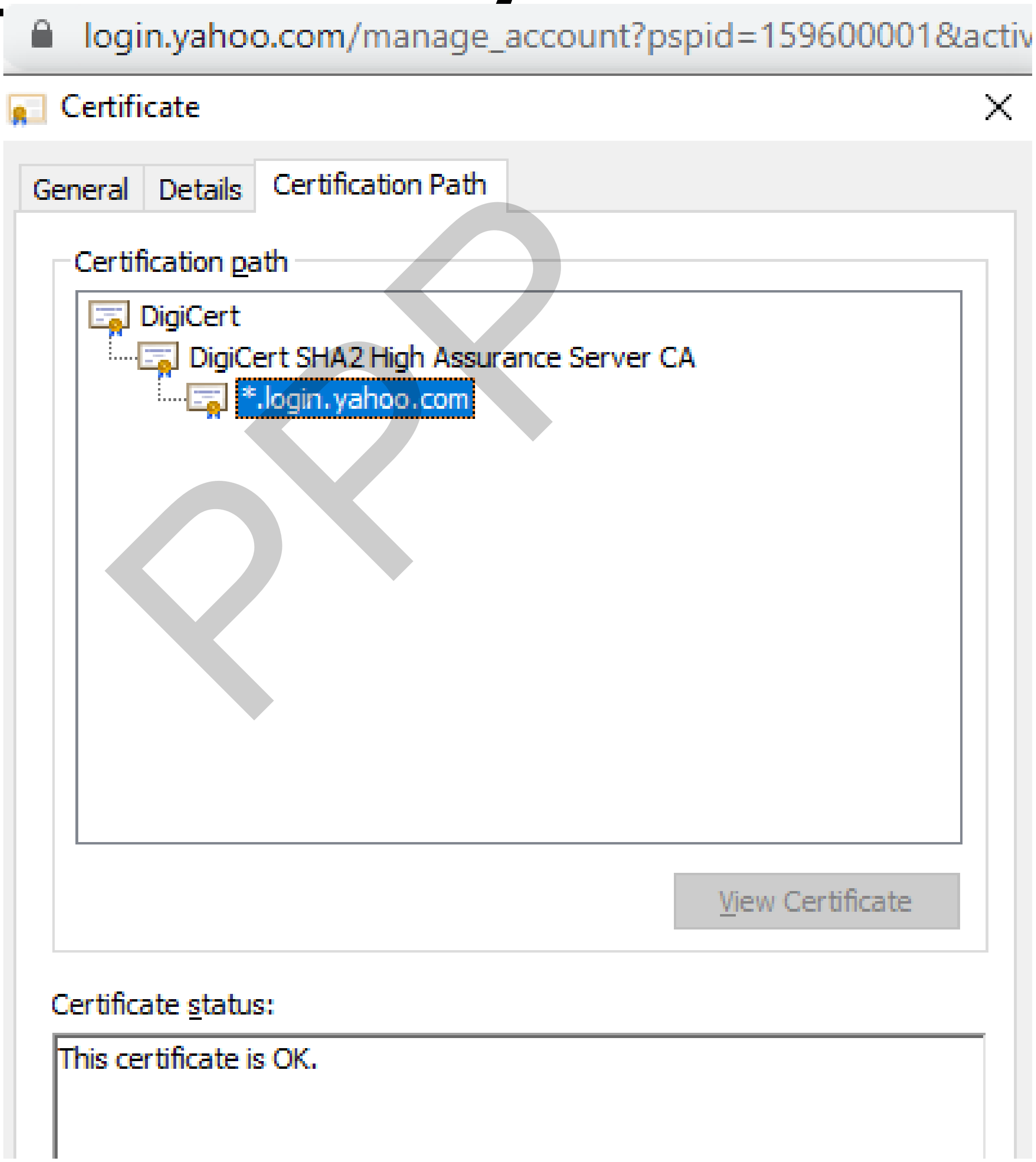
Show: <All>

Field	Value
CRL Distribution Points	[1]CRL Distribution Point: Distributi
Certificate Policies	[1]Certificate Policy:Policy Identifie
Authority Information Access	[1]Authority Info Access: Access M
SCT List	v1, f65c942fd17730221454180831
Key Usage	Digital Signature, Key Enciphermen
Basic Constraints	Subject Type=End Entity, Path Ler
Thumbprint	271204a72de80c90d739a9983a9c

V3

Digital Certificate :

E.g. www.mail.yahoo.com



login.yahoo.com/manage_account?pspid=159600001&activ

Certification path

DigiCert
DigiCert SHA2 High Assurance Server CA
*.login.yahoo.com

Certificate status:

This certificate is OK.