COURSE NAME: CRYPTOGRAPHY AND NETWORK SECURITY
COURSE CODE: R1641051
COURSE INSTRUCTOR: MADHU BABU JANJANAM, ASSOC. PROF, CSE
UNIT: 4

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

By the end of this unit…

- Explain different Integrity, Authentication and Key Management techniques and algorithms.

By the end of this session…

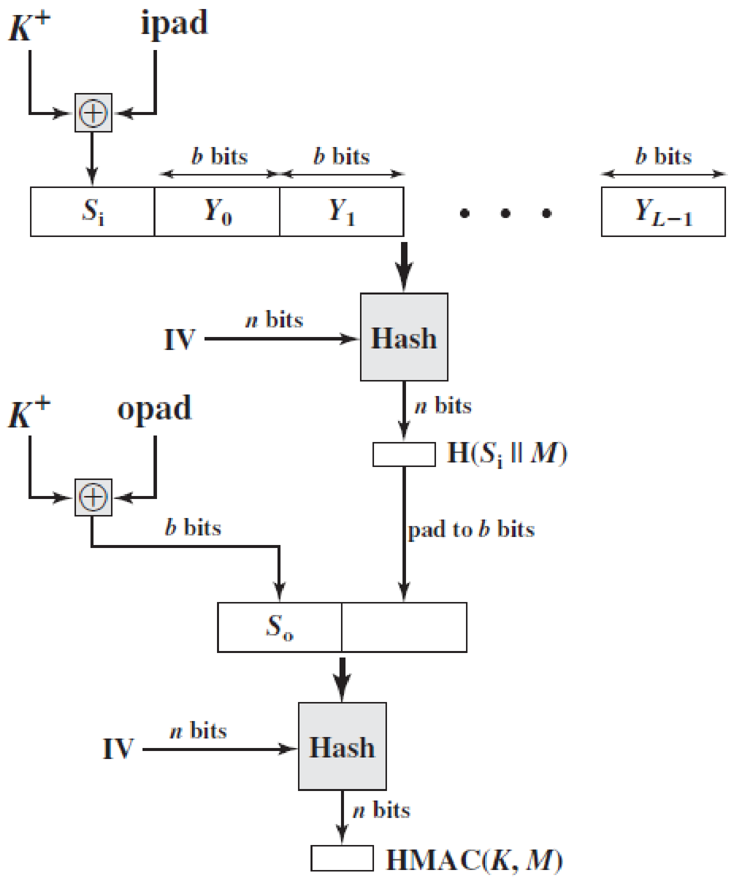
- Describe how message integrity and message authentication is maintained

Start

By the end of this session…

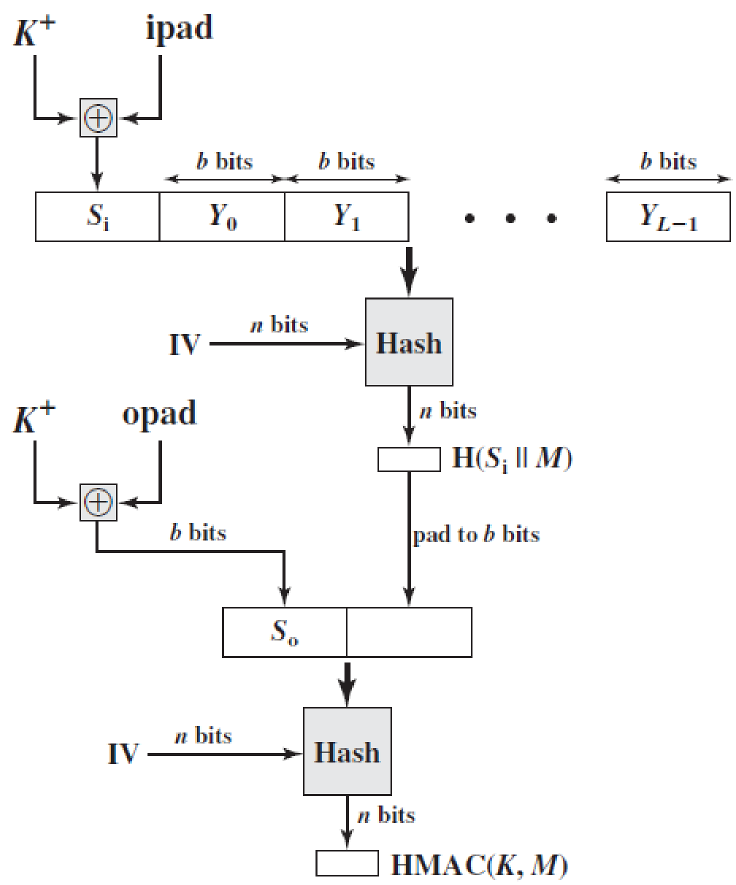
- Differentiate between message integrity and message authentication.
- Describe the working of Nested MAC (HMAC)

HMAC



- NIST (FIPS – 198) standard for nested MAC.
- Often referred as HMAC (Hashed MAC).
- Algorithm:
  1. The message is divided into L blocks, each block of size b bits.
  2. The secret key is left padded with 0's to create a b-bit key. The size of secret key before padding is recommended as n bits.
  3. The result of step 2 is xor with a constant called ipad. The value of ipad is 00110110 (36 in hex).
  4. The result is added as the first block and the number of blocks is now L+1.

Contd...



5. The result of step 4 is hashed to create an n-bit digest. We call the digest the intermediate HMAC.

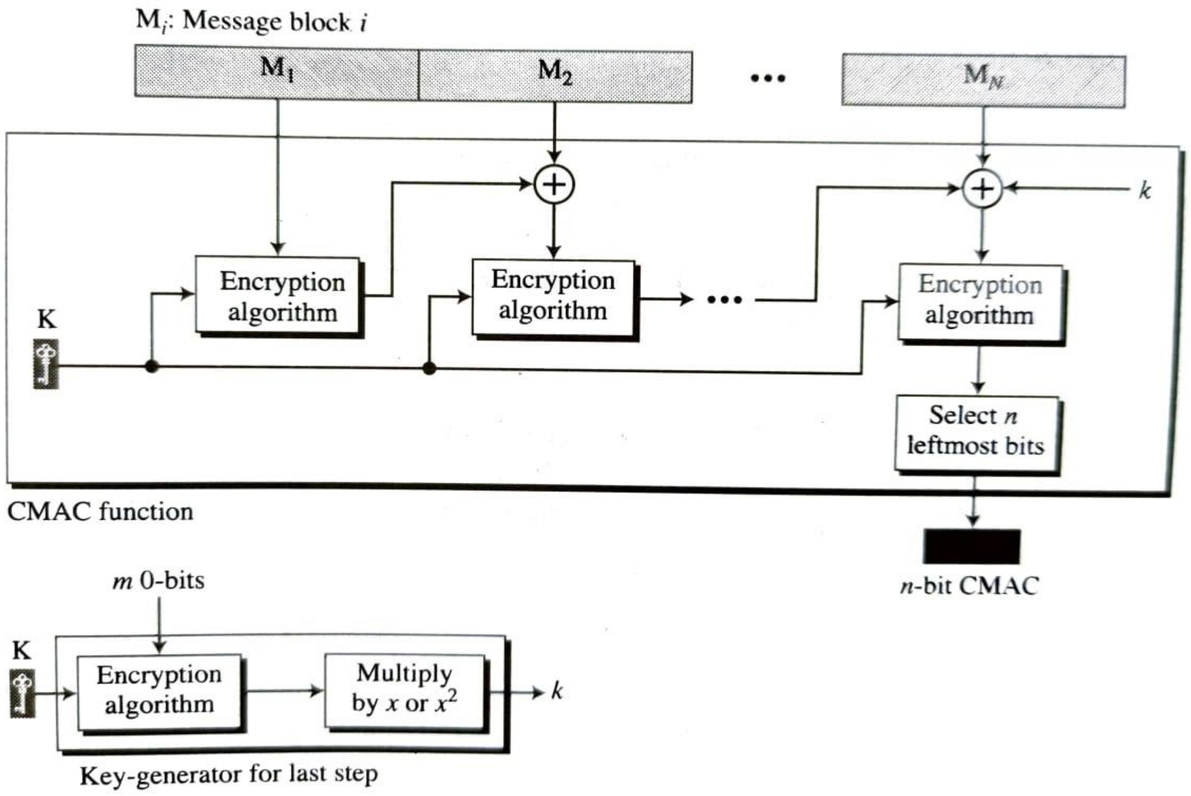
6. The intermediate n-bit HMAC is left padded with 0s to make a b-bit block.

7. The padded key is now xor with opad. The value of opad is 01011100 (5C in hex).

8. The step 4 is repeated to make the result in step 7 as first block to intermediate HMAC.

9. The result of step 8 is hashed with the same hashing algorithm to create the final n-bit HMAC.

CMAC



- NIST standard FIPS 113.
- Named as Data Authentication Algorithm or CMAC or CBCMAC.
- Operating Symmetric encryption algorithm in Cipher block chaining (CBC) mode to form MAC.
- The idea is to create one block of MAC from N blocks of plaintext using a symmetric key cipher N times.

Random Oracle Model

Message	Message Digest
4523AB1352CDEF45126	13AB
723BAE38F2AB3457AC	02CA
AB45CD10348372GAAA	A38B

- Introduced in 1993 by Bellare and Rogaway.
- Function based on ROM behaves as follows:
  - When a new message is given, the oracle generates a fixed length message digest as a random string of 0s and 1s. The oracle records the message and message digest.
  - When a message is given for which digest exists, the oracle simply gives the message digest.
  - The digest for a new message needs to be chosen independently from previous digests.
- The message digest of a message cannot be generated using a formula but generated as a random function.

Pigeonhole Principle

Message	Message Digest
000	00
001	01
010	10
011	11
100	00
101	01
110	10
111	11

- If n pigeonholes are occupied by n+1 pigeons, then atleast one pigeonhole is occupied by two pigeons.
- If n pigeonholes are occupied by kn+1 pigeons, then at least one pigeonhole is occupied by k+1 pigeons.
- There are some digests that correspond to morethan one message because the digest should be shorter than the message.
- Ex: if the message length is 6 bits long and digest size is 4 bits long then there are 16 digests possible, but there are 64 messages.

Birthday Problems

- There are four different birthday problems:
  - Problem 1: What is the minimum number of, k, of students in a classroom such that it is likely that at least one student has a predefined birthday?
  - Problem 2: What is the minimum number of k, of students in a classroom such that it is likely that at least one student has the same birthday as the student selected by the professor?
  - Problem 3: What is the minimum number of k, of students in a classroom such that it is likely that at least two students have the same birthday?
  - Problem 4: We have two classes, each with k students. What is the minimum value of k so that it is likely that at least one student from the first classroom has the same birthday as a student from the second classroom.

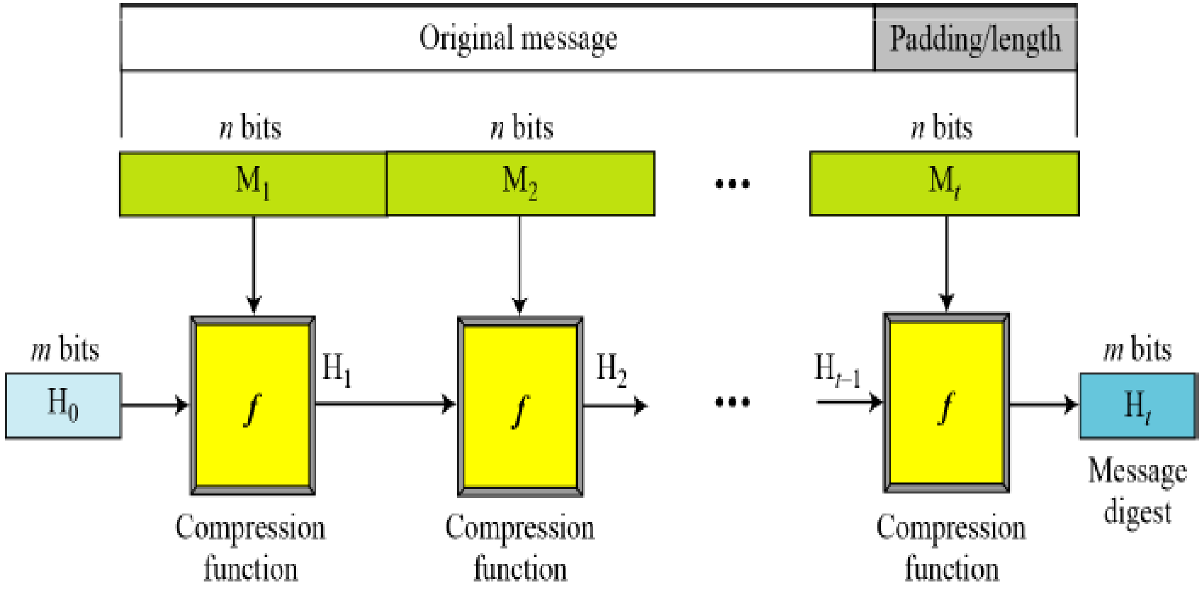
By the end of this session

- Discuss the general ideas behind cryptographic hash functions.

Iterated Hash Function

- All cryptographic hash function need to create a fixed size message digest from a variable size message.
- Rather building Hash function with variable-size input go for fixed size input and use it multiple times.
- The fixed size input functions is referred as Compression function, which takes n-bit string to create m-bit string. Where n>m.
- This process is named as Iterated cryptographic hash function.

Merkle-Damgard Scheme



Hash function is collision resistant if the compression function is collision resistant.

- It is an primitive iterated hash function that is collision resistant, if the compression function is collision resistant.
- The scheme uses the following steps:
  - The original message length and padding are appended to create an augmented message that can be divided into blocks of n bits.
  - The message is then considered as t blocks, each of n bits. We call each block M_1, M_2, M_3…,M_t. We call the intermediate message digests as H_1, H_2…H_t. H_t is the final Message digest.
  - Before starting the iteration, the digest H_0 is set to a fixed value and called Initial vector
  - The compression function at each iteration operates on H_{i-1} and M_i to create a new H_i.

Two Groups of Compression Functions

- Hash functions made from Scratch
  - MD5
  - SHA
  - RIPMED
- Hash functions based on Block Ciphers
  - Rabin Scheme
  - Davies-Meyer Scheme
  - Matyas-Meyer-Oseas Scheme
  - Miyaguchi-Preneel Scheme

Hash Functions made from scratch

- Cryptographic hash functions uses compression functions that are made from scratch.

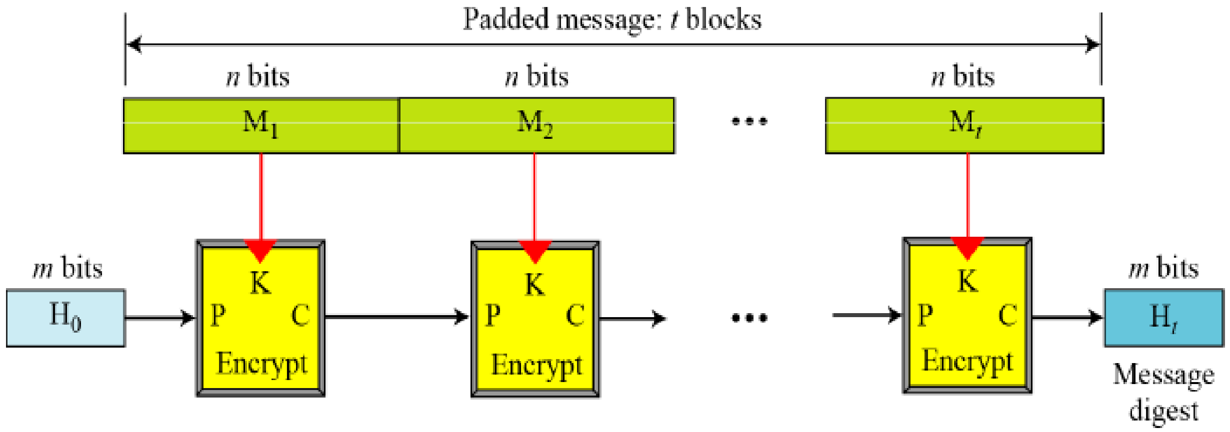
Message Digest (MD):

- Designed by Ron Rivest.
- Referred as MD2, MD4, MD5.
- MD5 divides the message into 512 bits and creates a 128 bit digest.

Secure Hash Algorithms (SHA):

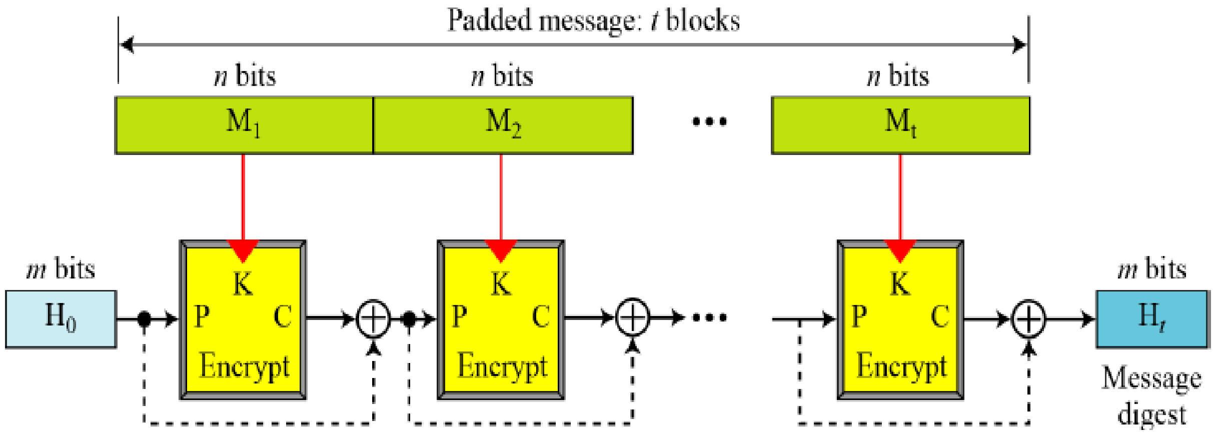
- Developed by NIST and published as a FIPS 180.
- Mostly based on MD5.
- Later revised as FIPS 180-1 named as SHA-1, then revised as FIPS 180-2 has four versions: SHA-224, SHA-256, SHA-512.

Hash Functions based on Block Ciphers – Rabin Scheme



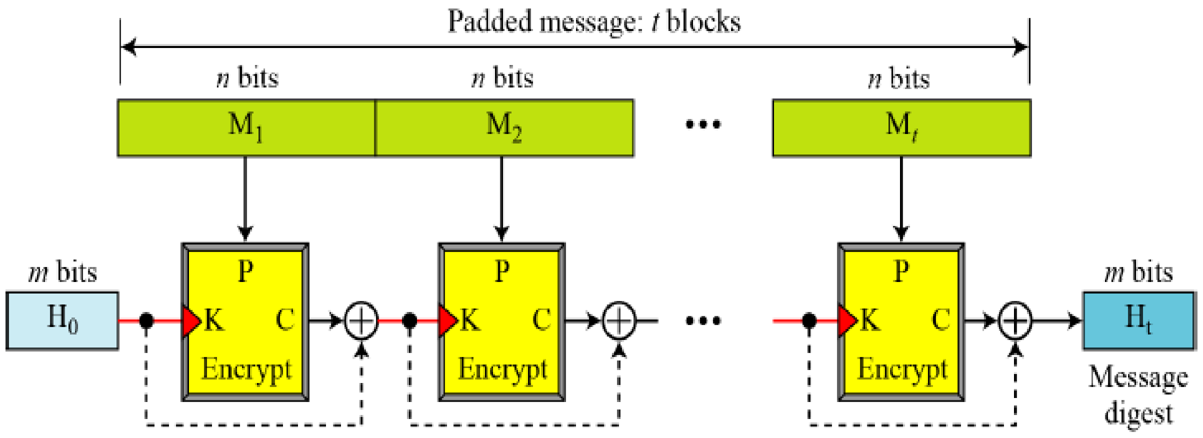
- Based on Merkle-Damgard scheme.
- Compression function is replaced by any encrypting cipher.
- The message block is used as the key and previously created digest is used as plaintext.
- The message digest size is the ciphertext size created by the block cipher.
- Vulnerable to Meet-in-the-Middle attack.

Hash Functions based on Block Ciphers – Davies-Meyer Scheme



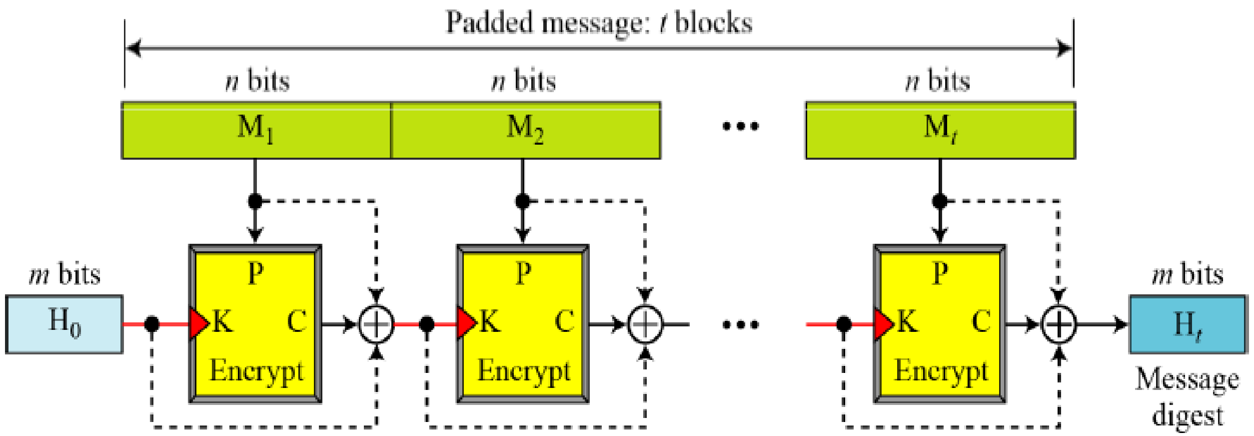
- Same as Rabin scheme except that this uses forward feed to protect against meet-in-the-middle attack.

Hash Functions based on Block Ciphers – Matyas-Meyer-Oseas Scheme



- Dual version of Davies-Meyer scheme.
- The Message block and IV can be used interchangeable as Plaintext and key.
- This scheme is possible only with the cipher where plaintext and ciphertext sizes are same.

Hash Functions based on Block Ciphers – Miyaguchi-Preneel Scheme

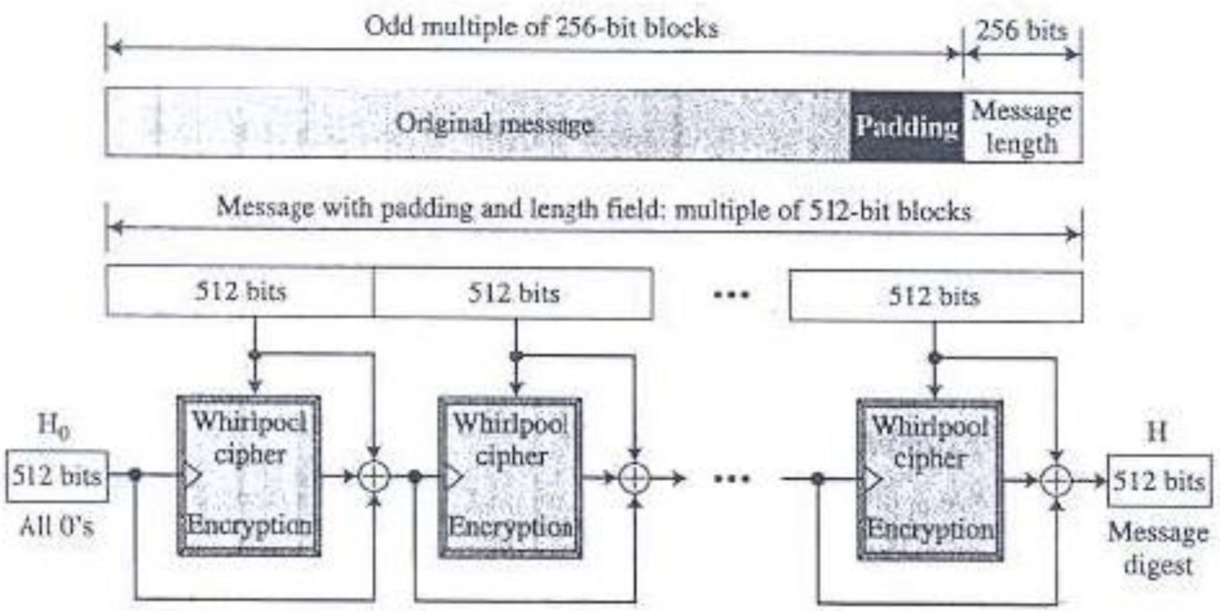


- Extended version of Matyas-Meyer-Oseas scheme.
- To make more stronger against attacks, the plaintext, the cipher key and cipher text are all xored to create new digest.

By the end of this session…

- Describe the working of Whirlpool cryptographic hash function.

Whirlpool

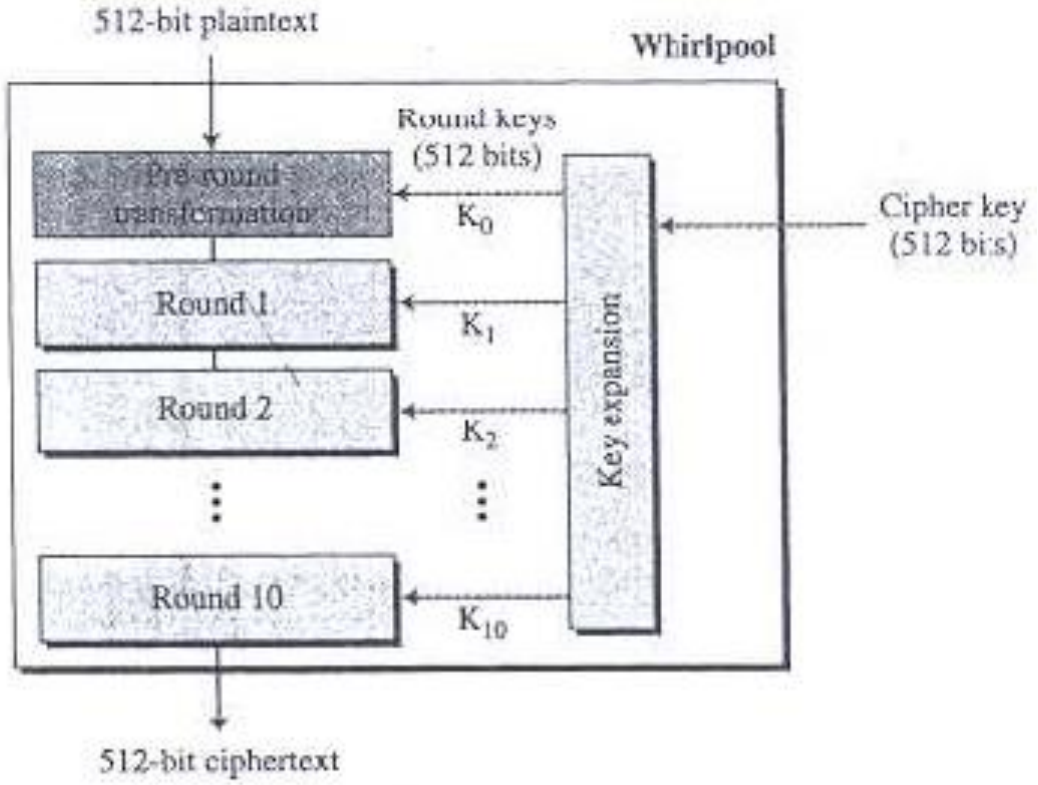


- Designed by Vincent Rijmen and Paulo S. L. M. Barreto.
- Endorsed by New European Schemes for Signatures, Integrity, and Encryption (NESSIE).
- It is an iterated cryptographic hash function based on Miyaguchi-Preneel scheme and uses a symmetric key block cipher as compression function.
- The block cipher used is modified AES cipher

Preparation

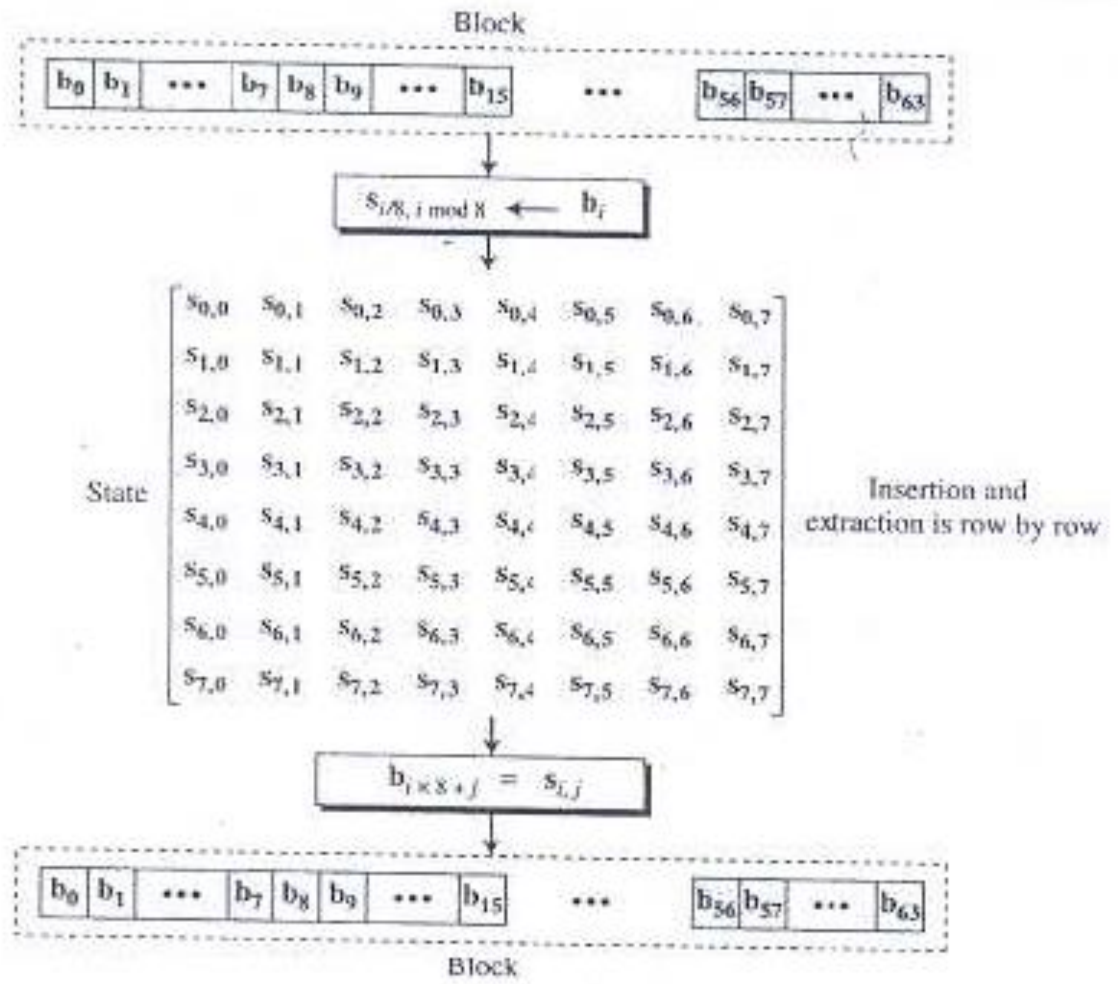
- Requires the length of the original message less than 2^{256} bits.
- The message need to be padded before processed with 1-bit followed by necessary number of 0-bits.
- A block of 256 bits is added to define the length of original message.
- After padding and adding length, the augmented message size is a multiple of 512 bits.
- Produces a digest of 512 bits with initial vector of size 512 bits.

Whirlpool Cipher - Rounds



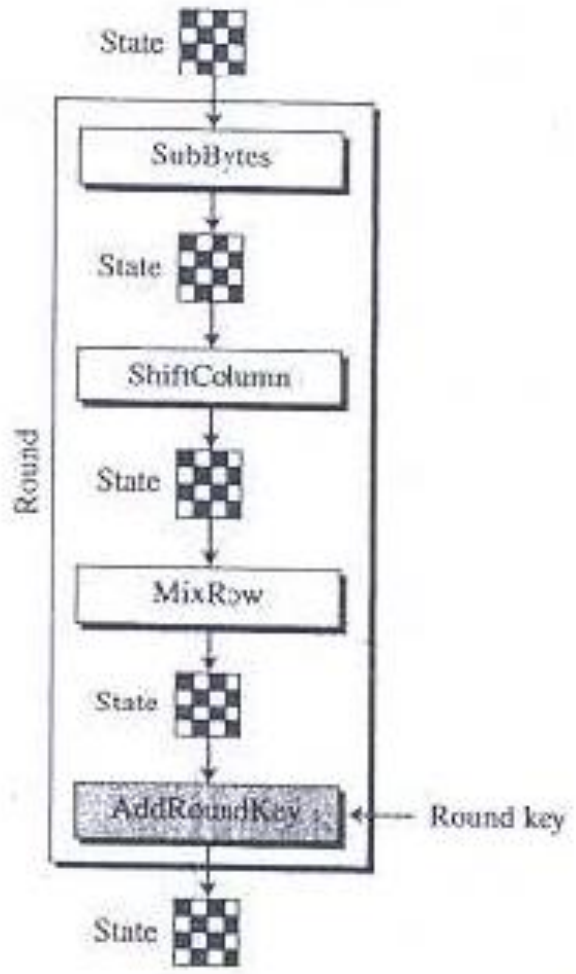
- Is an non-Feistel cipher like AES and a block cipher user in hash algorithm.
- Has 10 rounds.
- The block size and the key size are 512 bits.
- Uses 11 round keys, K0 to K10, each of 512 bits.

Whirlpool cipher – States and Blocks



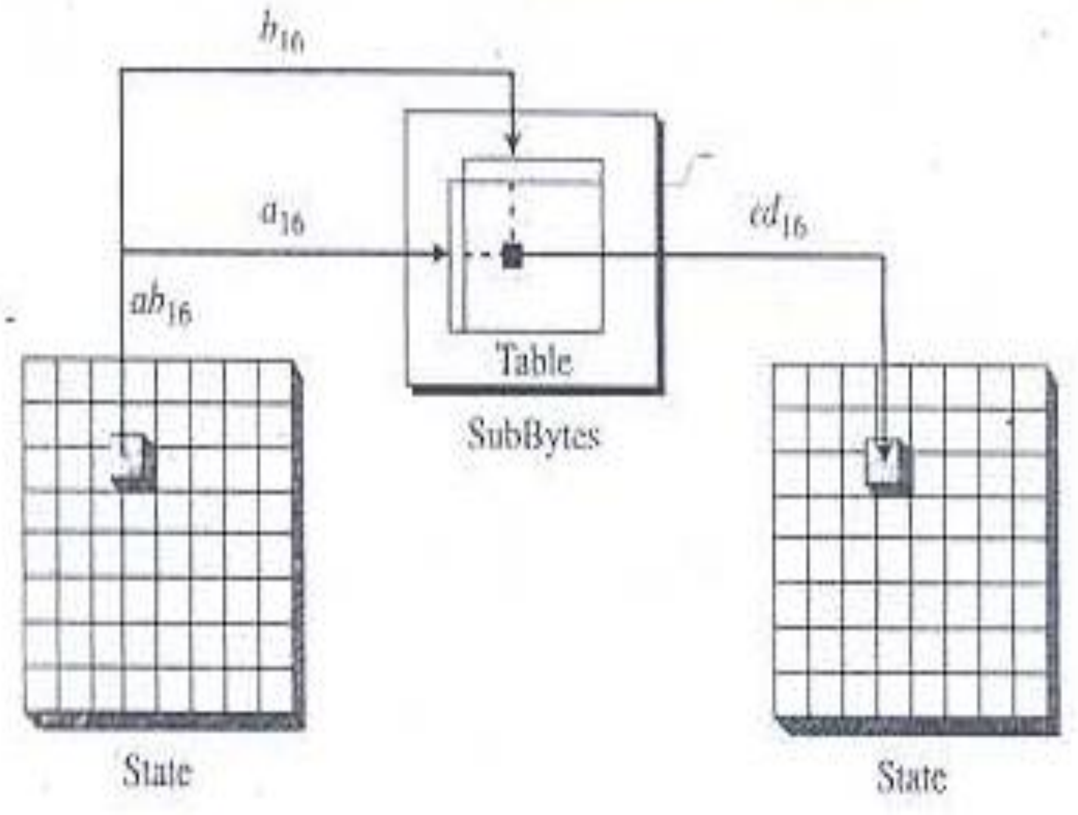
- Like AES, Whirlpool cipher uses states and blocks.
- The size of block or state is 512 bits.
- A block is considered as a row matrix of 64 bytes.
- A state is considered as a square matrix of 8 X 8 bytes.
- Unlike AES, the block to state and state to block transformation is done row by row.

Whirlpool cipher – Structure of Each Round



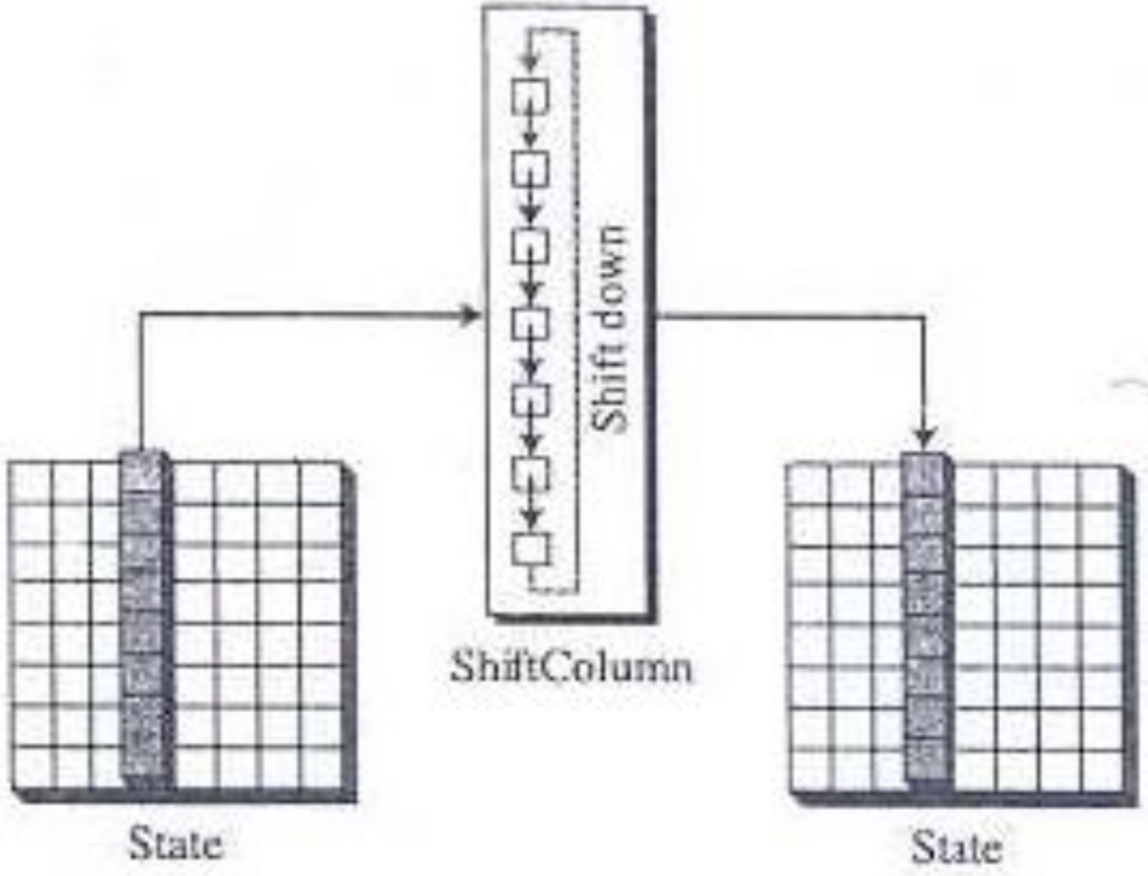
- SubBytes
- Shift Columns
- MixRow
- AddRoundKey

Whirlpool cipher – SubBytes

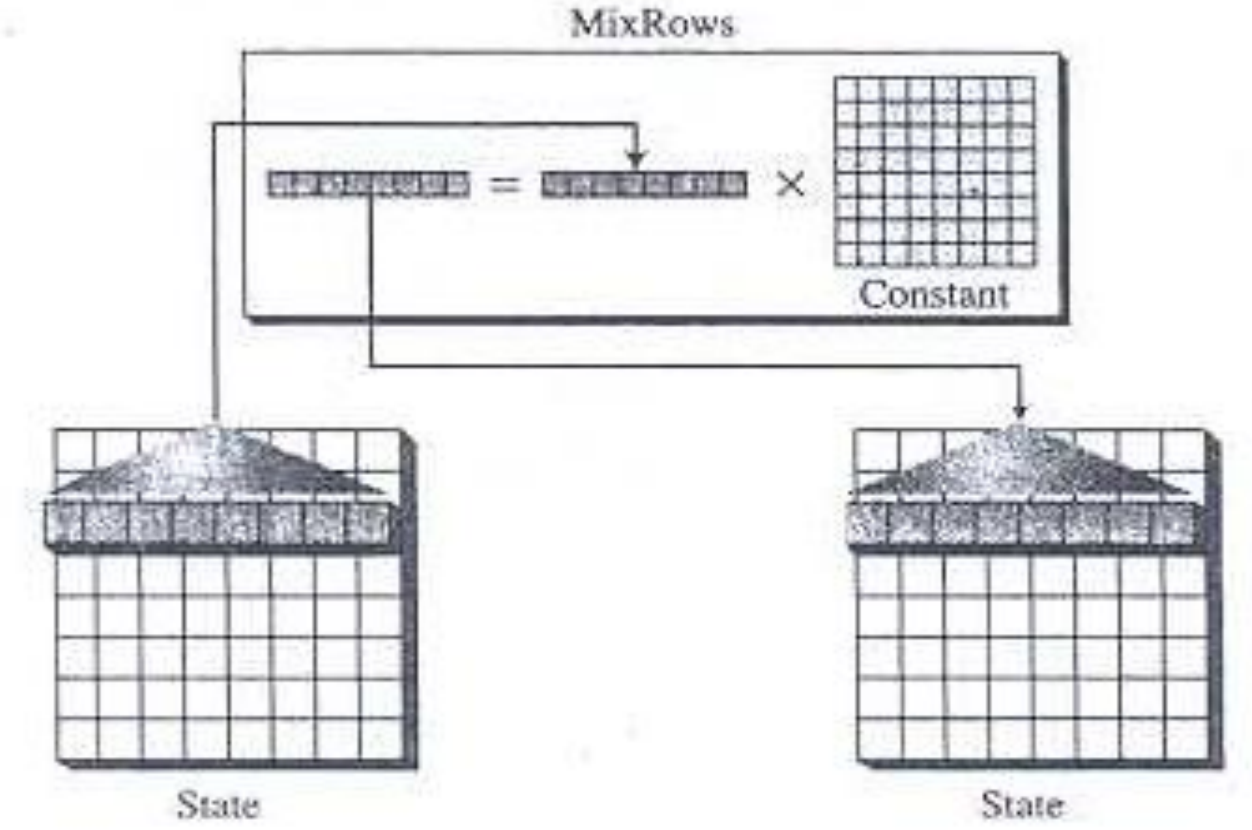


	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	18	23	C6	E8	87	B8	01	4F	36	A6	D2	F5	79	6F	91	52
1	16	BC	9B	8E	A3	0C	7B	35	1D	E0	D7	C2	2E	4B	FE	57
2	15	77	37	E5	9F	F0	4A	CA	58	C9	29	0A	B1	A0	6B	85
3	BD	5D	10	F4	CB	3E	05	67	E4	27	41	8B	A7	7D	95	C8
4	FB	EF	7C	66	DD	17	47	9E	CA	2D	BF	07	AD	5A	83	33
5	63	02	AA	71	C8	19	49	C9	F2	E3	5B	88	9A	26	32	B0
6	E9	0F	D5	80	BE	CD	34	48	FF	7A	90	5F	20	68	1A	AE
7	B4	54	93	22	64	F1	73	12	40	08	C3	EC	DB	A1	8D	3D
8	97	00	CF	2B	76	82	D6	1B	B5	AF	6A	50	45	F3	30	EF
9	3F	55	A2	EA	65	BA	2F	C0	DE	1C	FD	4D	92	75	06	8A
A	B2	E6	0E	1F	62	D4	A8	96	F9	C5	25	59	84	72	39	4C
B	5E	78	38	8C	C1	A5	E2	61	B3	21	9C	1E	43	C7	FC	04
C	51	99	6D	0D	FA	DF	7E	24	3B	AB	CE	11	8F	4E	B7	EB
D	3C	81	94	F7	9B	13	2C	D3	E7	6E	C4	03	56	44	7E	A9
E	2A	BB	C1	53	DC	0B	9D	6C	31	74	F6	46	AC	89	14	E1
F	16	3A	69	09	70	B6	C0	ED	CC	42	98	A4	28	5C	F8	86

Whirlpool cipher – ShiftColumns



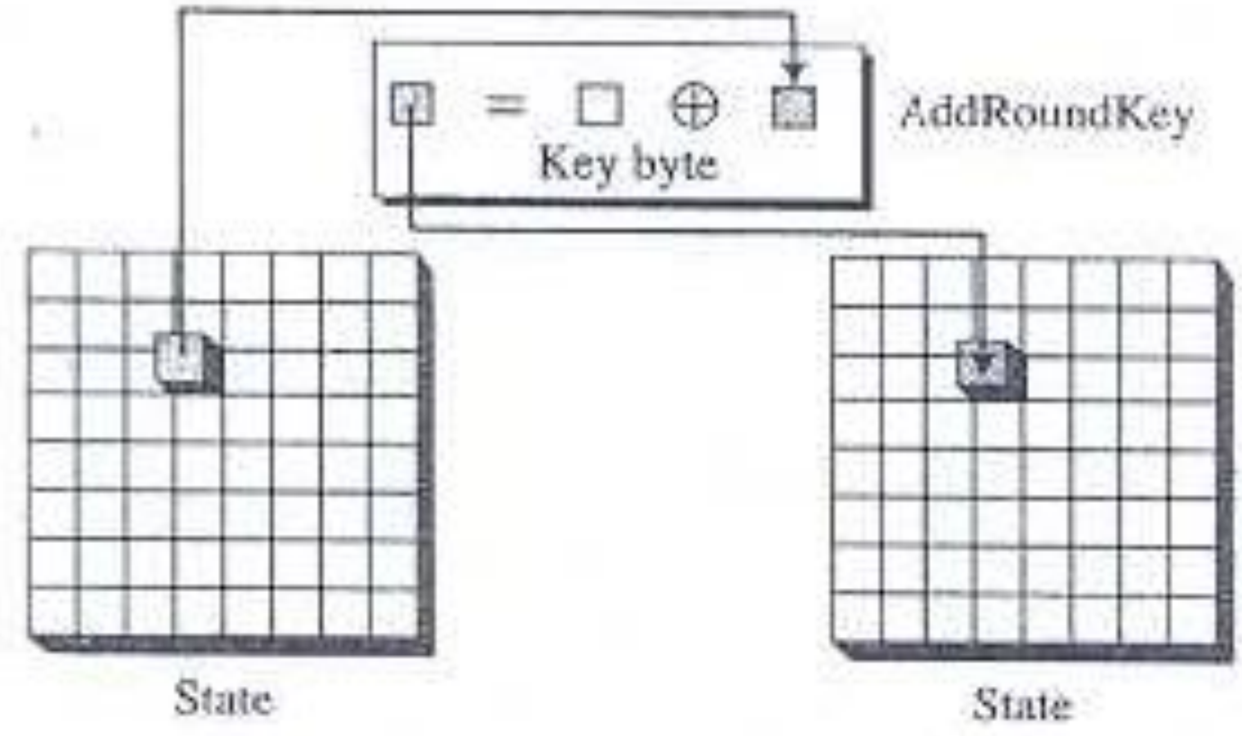
Whirlpool cipher – MixRows



$$\begin{bmatrix} 01 & 01 & 04 & 01 & 08 & 05 & 02 & 09 \\ 09 & 01 & 01 & 04 & 01 & 08 & 05 & 02 \\ 02 & 09 & 01 & 01 & 04 & 01 & 08 & 05 \\ 05 & 02 & 09 & 01 & 01 & 04 & 01 & 08 \\ 08 & 05 & 02 & 09 & 01 & 01 & 04 & 01 \\ 01 & 08 & 05 & 02 & 09 & 01 & 01 & 04 \\ 04 & 01 & 08 & 05 & 02 & 09 & 01 & 01 \\ 01 & 04 & 01 & 08 & 05 & 02 & 09 & 01 \end{bmatrix}$$

Constant matrix

Whirlpool cipher – AddRoundKey



Whirlpool cipher – Summary

Block size: 512 bits
Cipher key size: 512 bits
Number of rounds: 10
Key expansion: using the cipher itself with round constants as round keys
Substitution: SubBytes transformation
Permutation: ShiftColumns transformation
Mixing: MixRows transformation
Round Constant: cubic roots of the first eighty prime numbers

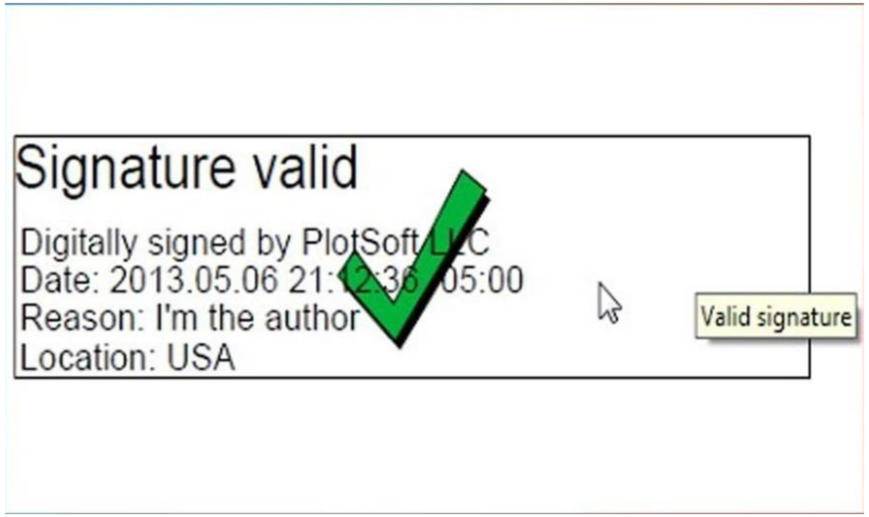
By the end of this session…

- Compare between conventional signatures and digital signatures
- Describe the process of digital signature

Digital Signatures

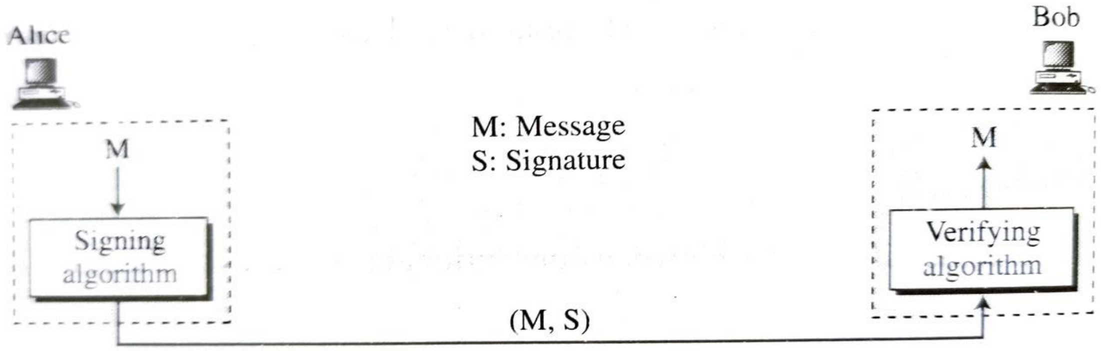
- A person signs a document to show that it is originated from her or was approved by her.
- Eg:
  - A notice from an authorized entity
  - A check from a bank customer
  - A painting from an artist
- Signature is the symbol of authentication
- What if we want authentication for electronic documents.

Conventional signature vs Digital Signature



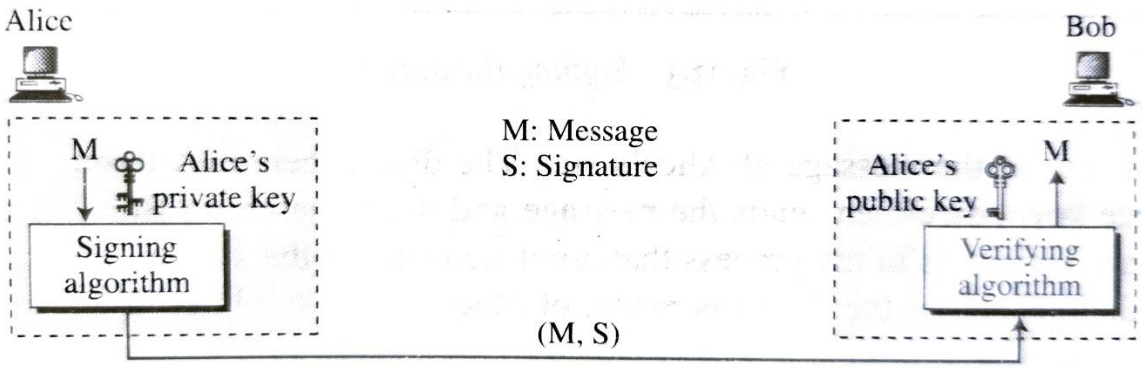
- Inclusion
- Verification method
- Relationship
- Duplicity

Process



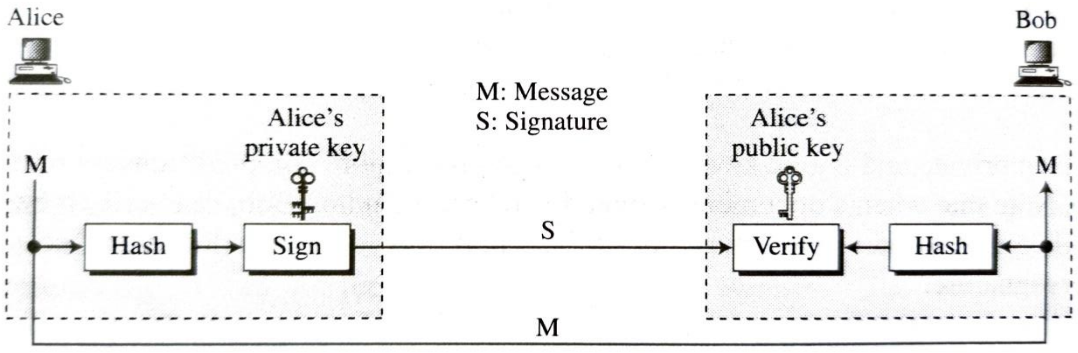
- Sender uses signing algorithm to generate signature of a message.
- Receiver uses the verifying algorithm to verify the signature.

Need of Keys



- Sender uses his private key to generate the signature.
- Receiver uses sender's public key to verify the signature.
- A digital signature needs a public key cryptosystem.

Signing the Digest



- Usually, Asymmetric cryptosystem is very inefficient while dealing with long messages.
- Digital signature schemes require asymmetric cryptosystem.
- Solution is to sign the digest of the message.

By the end of this session…

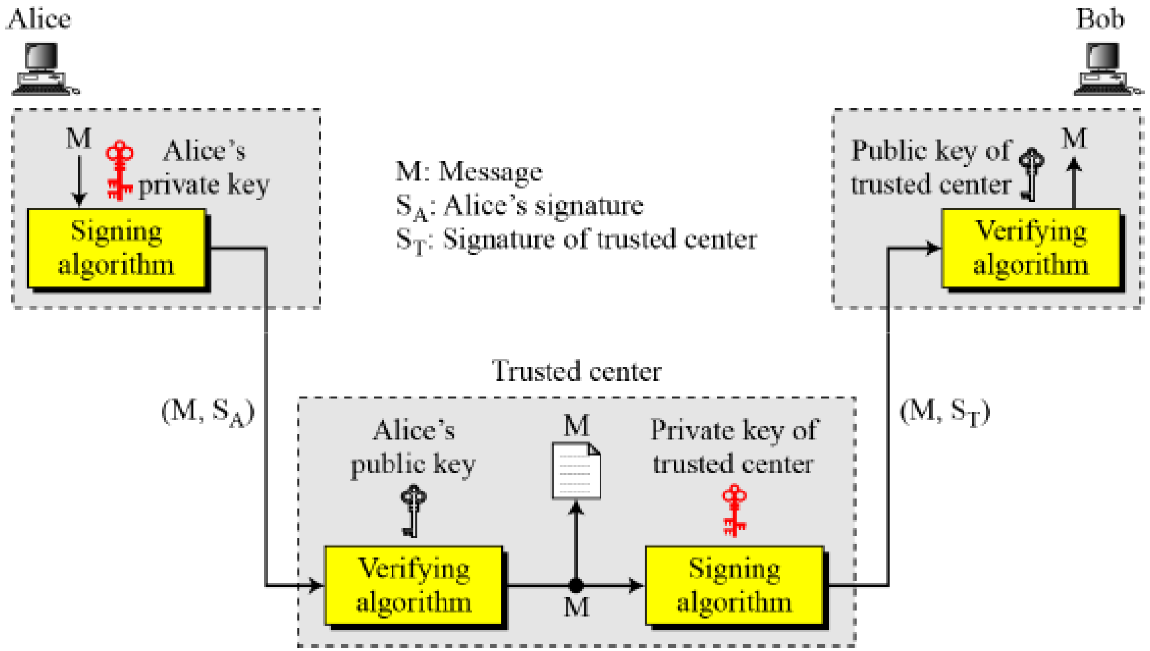
- Explain the security services provided by a digital signature
- Describe attacks on digital signatures
- Describe the working of RSA digital signature scheme.

Security services

	Mechanism							
Service	**Encipherment**	**Digital Signature**	**Access Control**	**Data Integrity**	**Authentication Exchange**	**Traffic Padding**	**Routing Control**	**Notarization**
Peer entity authentication	Y	Y			Y			
Data origin authentication	Y	Y						
Access control			Y					
Confidentiality	Y						Y	
Traffic flow confidentiality	Y					Y	Y	
Data integrity	Y	Y		Y				
Nonrepudiation		Y		Y				Y
Availability				Y	Y			

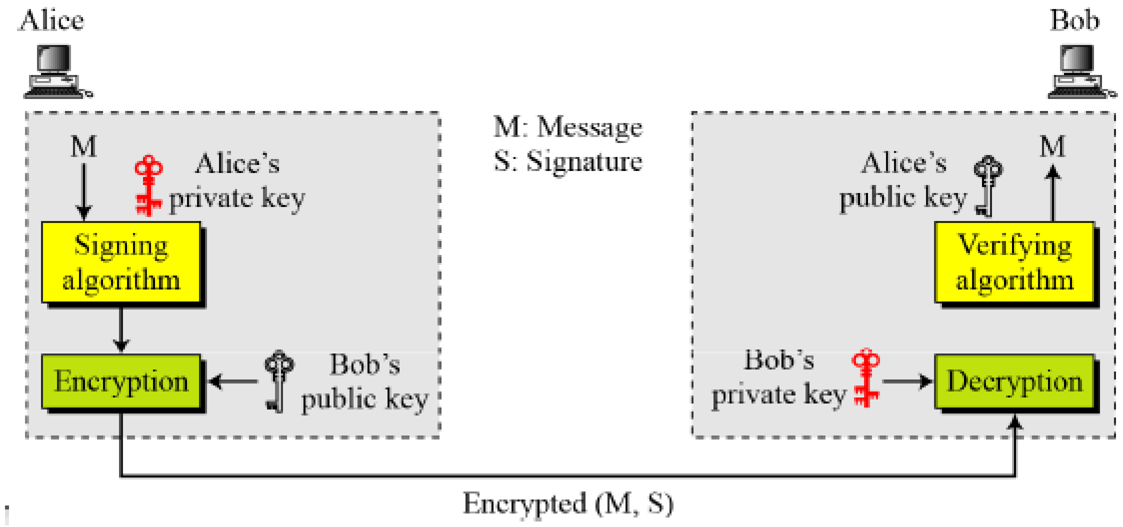
- Message Authentication
- Message Integrity
- Nonrepudiation
- Confidentiality with Encipherment

Nonrepudiation



- Alice signs a message and then denies it, can Bob later prove that Alice actually signed it?
- Alice may change the private key and claim that signature is not authentic.
- One solution is to use trusted third party.

Confidentiality



- Digital signature alone can't provide confidentiality.
- The message and signature must be encrypted using either secret-key or public-key cryptosystem.

Attacks on Digital signatures

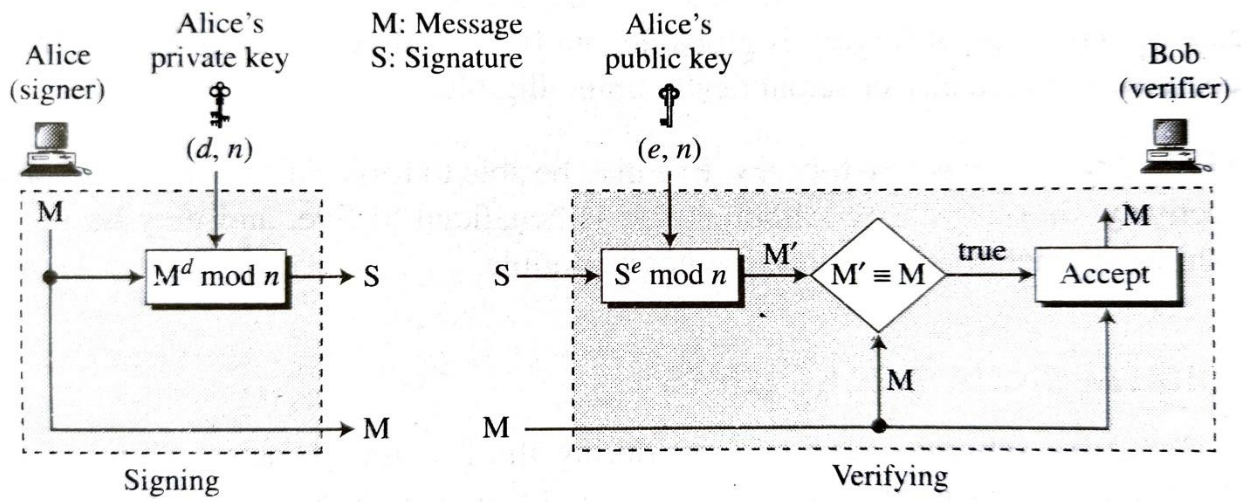
Attack types

- Key-Only Attack
- Known-Message Attack
- Chosen-Message Attack

Forgery types

- Existential Forgery
- Selective Forgery

RSA Digital signature scheme



- Alice creates a signature out of the message using her private exponent,

$$s = M^d \; mod \; n$$

and sends the message and the signature to Bob.

- Bob receives M and S. Bob applies Alice's public exponent to the signature to create a copy of the message

$$M^1 = \; S^e \; mod \; n$$

- If $M^1 \equiv M \; mod \; n$, Bob accepts the message.

RSA Digital signature scheme - Example

Alice: p=823, q=953, n=784319, e=313 and d=160009.

$$M: 19070, S = 19070^{160009} \; mod \; 784319$$
$$= 210625 \; mod \; 784319$$

Bob: e=313, M = 19070, S = 210625

$$M^1 = 210625^{313} \; mod \; 784319$$
$$= 19070 \; mod \; 784319$$

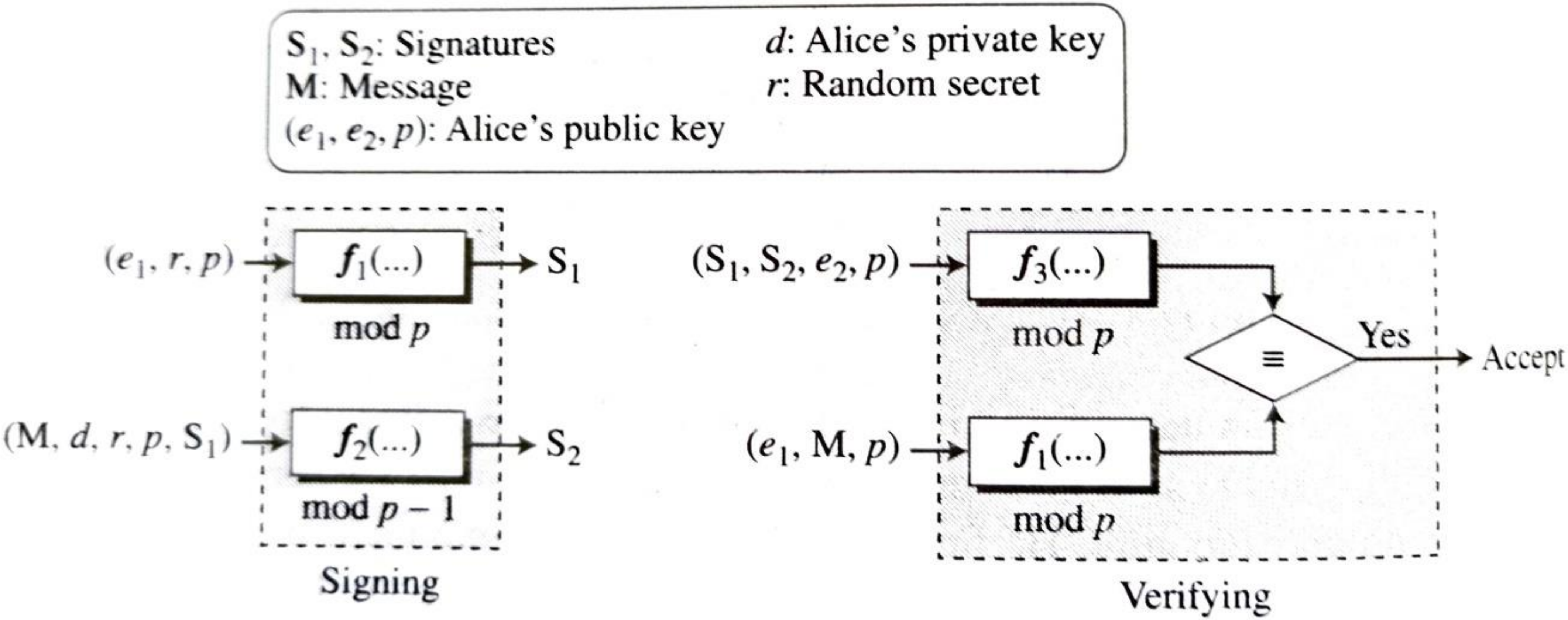
Attacks on RSA Signature

- Key-Only Attack – Existential Forgery
- Known-Message Attack – Existential Forgery
- Chosen-Message Attack – Selective Forgery

By the end of this session…

- Describe the working of ElGamal Digital Signature Scheme.

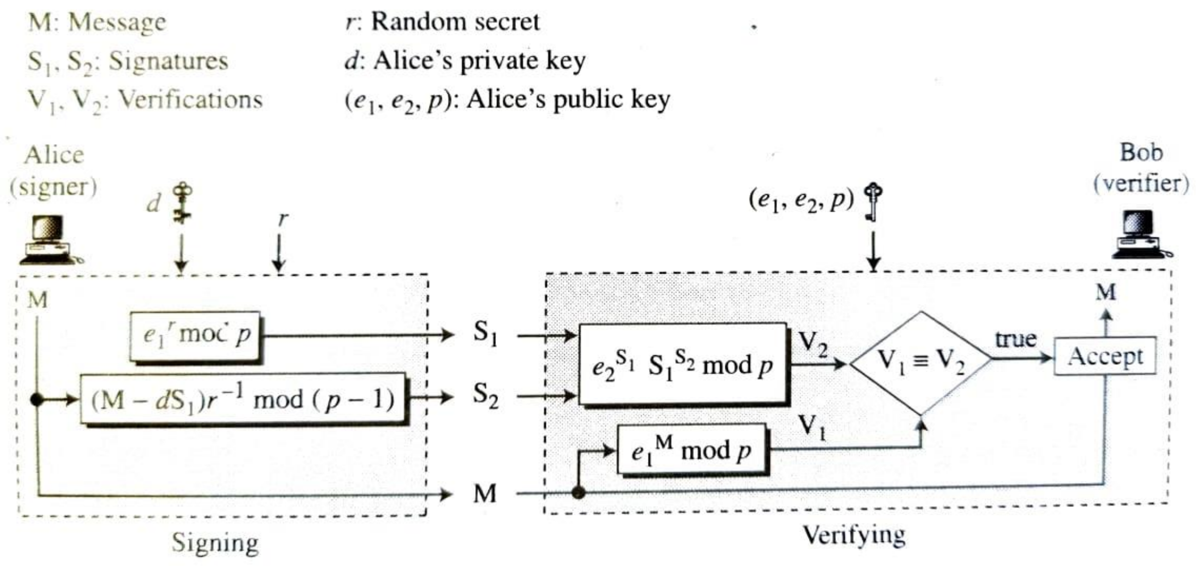
ElGamal Digital Signature Scheme



Key Generation

- Let p be the prime number and e_1 be a primitive element in Z_p^*.
- Alice selects her private key d < p-1 and calculates $e_2 = e_1^d \bmod p$
- Alice's public key is the tuple (e_1, e_2, p) and private key is d.

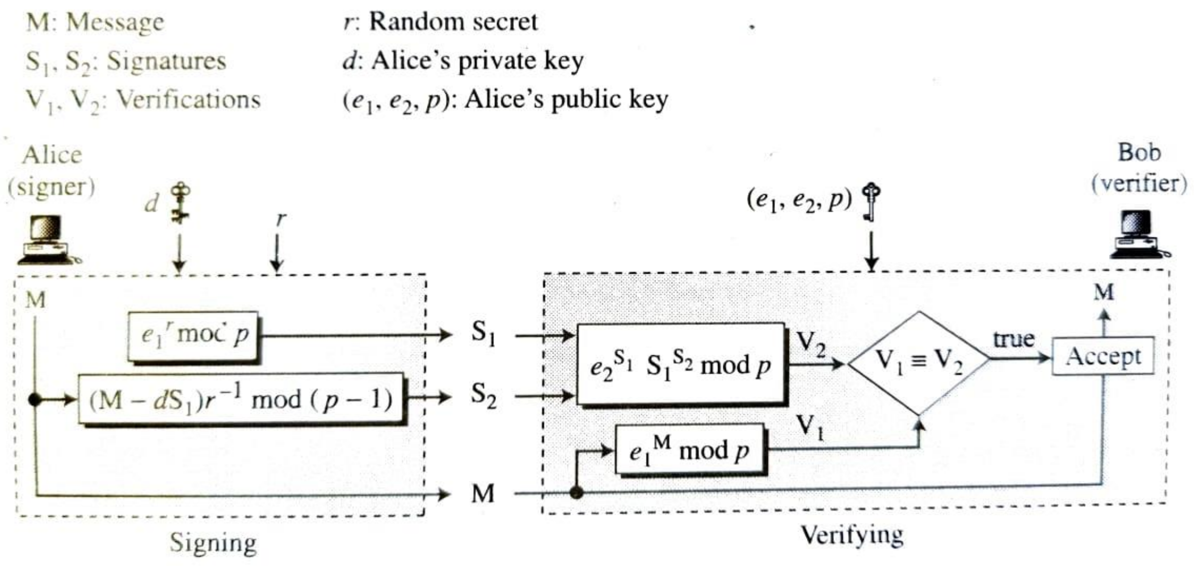
Verifying and Signing



Signing:

1. Alice chooses a secret random number r.
2. Alice calculates the first signature $S_1 = e_1^r \; mod \; p$
3. Alice calculates the second signature $S_2 = (M - d \times S_1) \times r^{-1} \; mod \; p - 1$
4. Alice sends M, S_1 and S_2 to Bob

Verifying and Signing – Contd...



Verifying: Receiver Bob receives M, S_1 and S_2

1. Bob checks to see if 0<S_1<p and 0<S_2<p-1.
2. Bob calculates $V_1 = e_1^M \ mod \ p$
3. Bob calculates $V_2 = e_1^{S_1} \times S_1^{S_2} \ mod \ p$
4. If V_1 is congruent to V_2 the message is accepted,

Example

Alice:

p=3119, e_1= 2, d=127

Calculates $e_2 = 2^{127}\ mod\ 3119 = 1702.$

Chooses r = 307.

Announces 2, 1702, 3119 as public key.

M = 320

$S_1 = e_1^r = 2^{307} = 2083\ mod\ 3119$

$S_2 = (M - d \times S_1) \times r^{-1}\ mod\ p - 1$

= (320 – 127 x 2083) x 307 $^{-1}$ = 2105 mod 3119.

Bob:

$V_1 = e_1^M\ mod\ p = 2^{320}$

$= 3006\ mod\ 3119$

$V_2 = e_1^{S_1} \times S_1^{S_2}\ mod\ p$

$= 1702^{2083} \times 2083^{2105}$

$= 3006\ mod\ 3119$

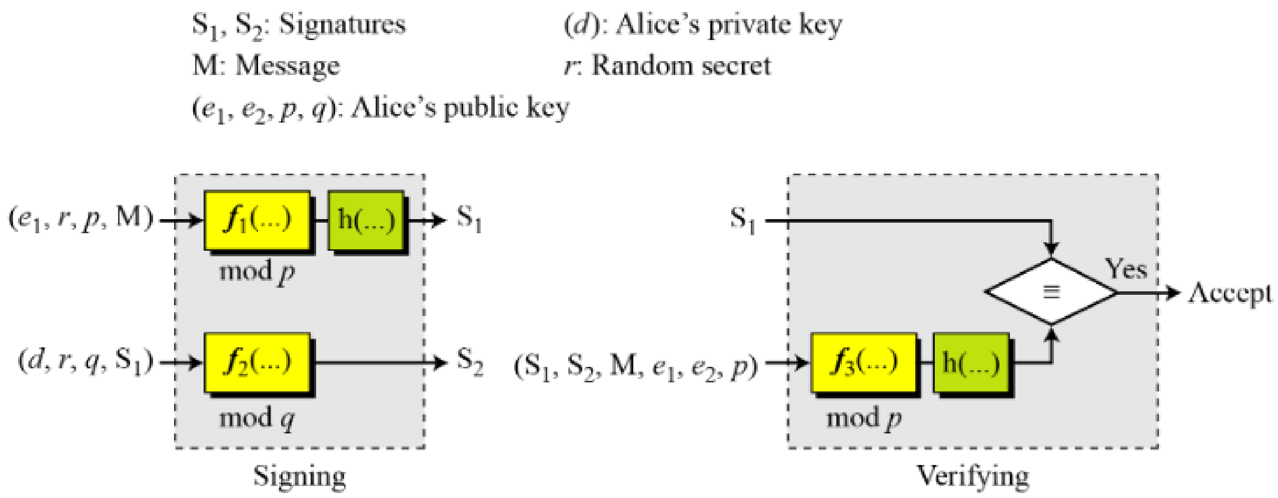
Attacks on ElGamal Signature scheme

- Key only forgery
- Known-message forgery

By the end of this session…

- Describe the working of Schnorr Digital Signature Scheme.

Schnorr Digital Signature Scheme



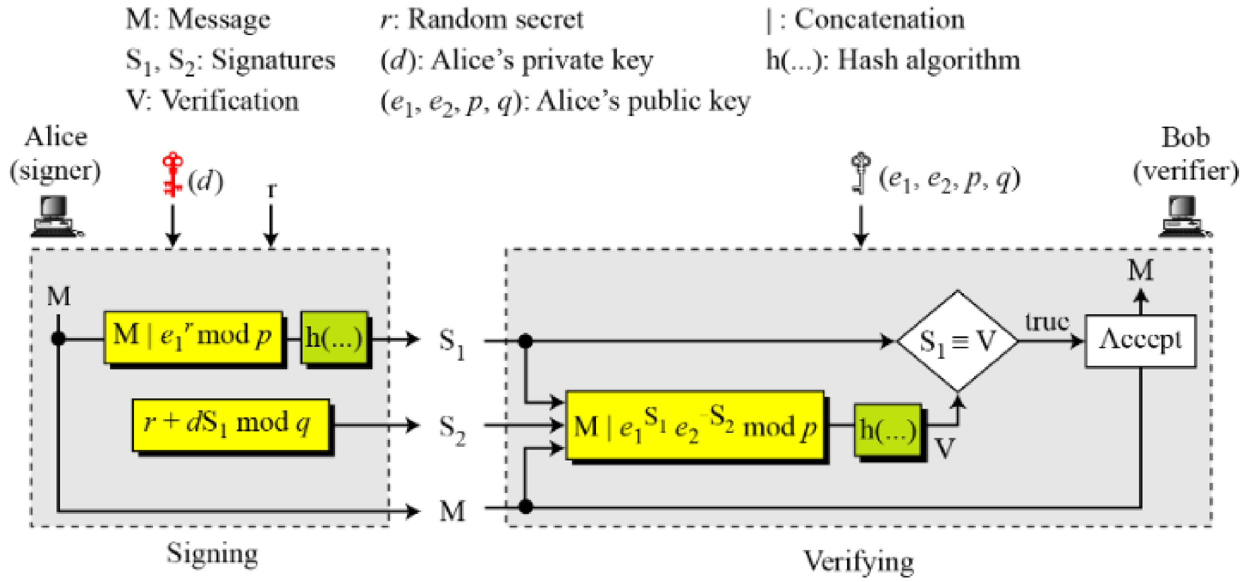
- p in ElGamal digital signature scheme needs to be large and atleast of 1024 bits to generate a signature as large as 2048 bits.
- To reduce the size of the signature, Schnorr proposed a new scheme based on ElGamal, but with reduced signature size.
- Signing process uses two functions to create two signatures and verification process uses one function.

Key Generation

- Alice selects a prime p (1024 bits).
- Alice selects another prime q. The prime q needs to divide (p-1).
- Alice chooses e_1 to be qth root of 1 modulo p.
  - Alice chooses a primitive element e_0 in Z_p.
  - Calculates $e_1 = e_0^{(p-1)/q} \ mod \ p$.
- Alice chooses an integer d, as the private key
- Alice calculates $e_2 = e_1^d \ mod \ p$
- Alice public key is (e_1, e_2, p, q) and private key is d.

- p = 2267
- q = 103
- $e_1 = 354$
  - $e_0 = 2$
  - $2^{22} \ mod \ 2267$
- d = 30
- $e_2 = 354^{30} \ mod \ 2267$ = 1206
- Public key (354, 1206,2267,103)
- Private key d = 30

Signing and Verifying



Signing:

1. Alice chooses a random number r.
2. Alice calculates the first signature
$$S_1 = h(M|e_1^r \ mod \ p)$$
3. Alice calculates the second signature
$$S_2 = r + d \times S_1 \ mod \ q$$
4. Alice sends $M, S_1, and \ S_2$

Verifying:

1. Bob calculates $V = h(M|e_1^{S_2}.e_2^{S_1} \ mod \ p)$
2. If S_1 is congruent to V mod p, the message is accepted.

Example

- Public Key = (354, 1206,2267,103) and Private key d = 30.

Signing:

- Random number r = 11, Message M = 1000
- Calculates $S_1 = h(M|e_1^r \; mod \; p) = h(1000|354^{11} \; mod \; 2267) = h(1000630) = 200$
- Calculates $S_2 = r + d \times S_1 \; mod \; q = 11 + 30 \times 200 \; mod \; 103 = 35$

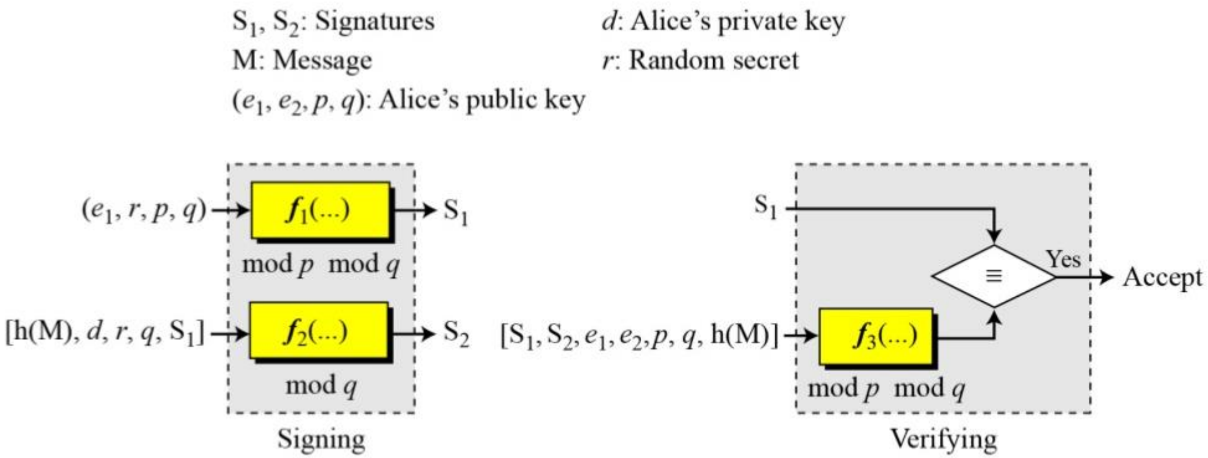
Verifying:

- Calculates $V = h\left(M|e_1^{S_2}.\, e_2^{S_1} \; mod \; p\right) = h(1000|354^{35}.\, 1206^{200} \; mod \; 2267) = h(1000|630) = 200$
- Here S_1 and V are equal and the signature is valid.

By the end of this session…

- Describe the working of Digital Signature Standard (DSS)

Digital Signature Standard



- Adopted by NIST in 1994 and published DSS as FIPS 186.
- DSS uses DSA based on ElGamal scheme with some ideas from the Schnorr scheme.
- The size of the prime is 512 bits, later the size made as variable.
- Signing process uses two functions to create two signatures
- Verifying process uses one function and the output of the function is compared with first signature.
- Uses two public moduli: p and q.

Key Generation

- Alice chooses a prime p, between 512 and 1024 bits, but must be multiple of 64.
- Alice chooses a 160-bit prime q, such that q divides (p-1).
- Alice uses two multiplication groups $< Z_p^*, \times >$ and $< Z_q^*, \times >$; the second is a subgroup of the first.
- Alice creates e_1 to be the qth root 1 modulo p.
- Alice chooses d as the private key and calculates $e_2 = e_1^d$.
- Alice's public key is (e_1, e_2, p, q) and private key is (d)

Verifying and Signing

Verifying:

- Bob checks to see if $0 < S_1 < q$.
- Bob checks to see if $0 < S_2 < q$.
- Bob calculates a digest of M.
- Bob calculates $V = \left[\left(e_1^{h(M)S_2^{-1}} . e_2^{S_1.S_2^{-1}}\right) \ mod \ p\right] mod \ q$.
- If S_1 is congruent to V, then the signature is valid

Signing:

- Alice chooses a random number r.
- Alice calculates first signature $S_1 = (e_1^r \ mod \ p) \ mod \ q$.
- Alice creates a digest of message h(M).
- Alice calculates the second signature $S_2 = (h(M) + dS_1)r^{-1} \ mod \ q$.
- Alice sends M, S_1, S_2 to Bob.

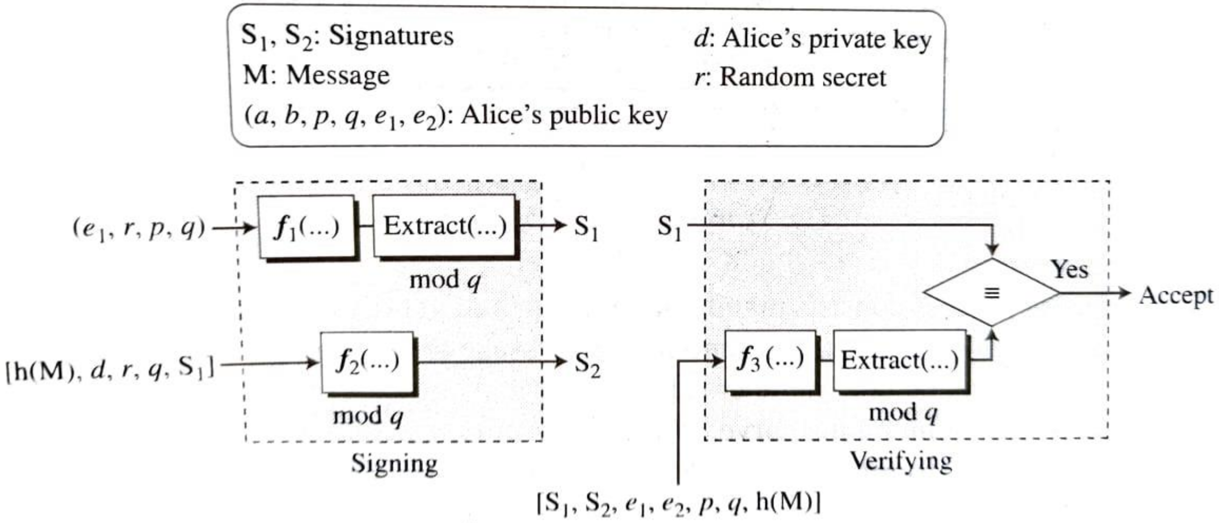
Example

- p = 8081, q= 101, $e_0 = 3$, d = 61, h(M) = 5000 and r = 61.
- Public key (6968,2038,8081,101) and Private key (61)
- $S_1 = 54 \; and \; S_2 = 40.$

By the end of this session…

- Describe the working of Elliptic curve Digital Signature Scheme.
- Outline the variations and applications for digital signatures.

Elliptic Curve Digital Signature Scheme



- DSA based on Elliptic curves and sometimes referred as ECDSA.
- Signing process, uses two functions and an extractor to create two signatures.
- Verifying process, the output of one function is compared with first signature.
- Functions f_1 and f_3 actually create points on the curve. f_1 creates a point on elliptic curve with sender's private key and f_3 creates the same point with the public key of sender

Key Generation

- Alice chooses an elliptic curve $E_p(a, b)$ with p a prime number.
- Alice chooses another prime number q to be used in the calculation.
- Alice chooses the private key d, an integer.
- Alice chooses $e_1(\dots, \dots)$, a point on the curve.
- Alice calculates $e_2(\dots, \dots) = \mathrm{d} \times e_1(\dots, \dots)$, another point on the curve.
- Alice public key is (a, b, p, q, e_1, e_2) and her private key is d.

Signing Process.

- Alice chooses a secret random number r, between 1 and q-1.
- Alice selects a third point on the curve, $P(u, v) = r \times e_1(\ldots, \ldots)$
- Alice uses the first coordinates of $P(u, v)$ to calculate the first signature $S_1 = u \bmod q$.
- Alice uses the digest of the message, her private key, and the secret random number r, and the S_1 to calculate the second signature $S_2 = (h(M) + d \times S_1)r^{-1} \bmod q$.
- Alice sends M, S_1, S_2.

Verifying

- Bob uses M, S_1, S_2 to create two intermediate results, A and B:

$A = h(M)S_2^{-1}\ mod\ q$ and $B = S_2^{-1}S_1\ mod\ q$.

Bob then reconstructs the third point $T(x, y) = A \times e_1(\ldots, \ldots), B \times e_2(\ldots, \ldots)$

- Bob uses the first coordinate of T(x,y) to verify the message. If $x = S_1\ mod\ q$, the signature is valid.

Variations and Applications of Digital Signatures

- Two Variations
  - Timestamped Signatures
  - Blind Signatures
  - Undeniable Digital Signatures
    - Signing protocol
    - Verifying protocol
    - Disavowal protocol

By the end of this session…

- Explain Kerberos as a KDC and Authentication protocol.

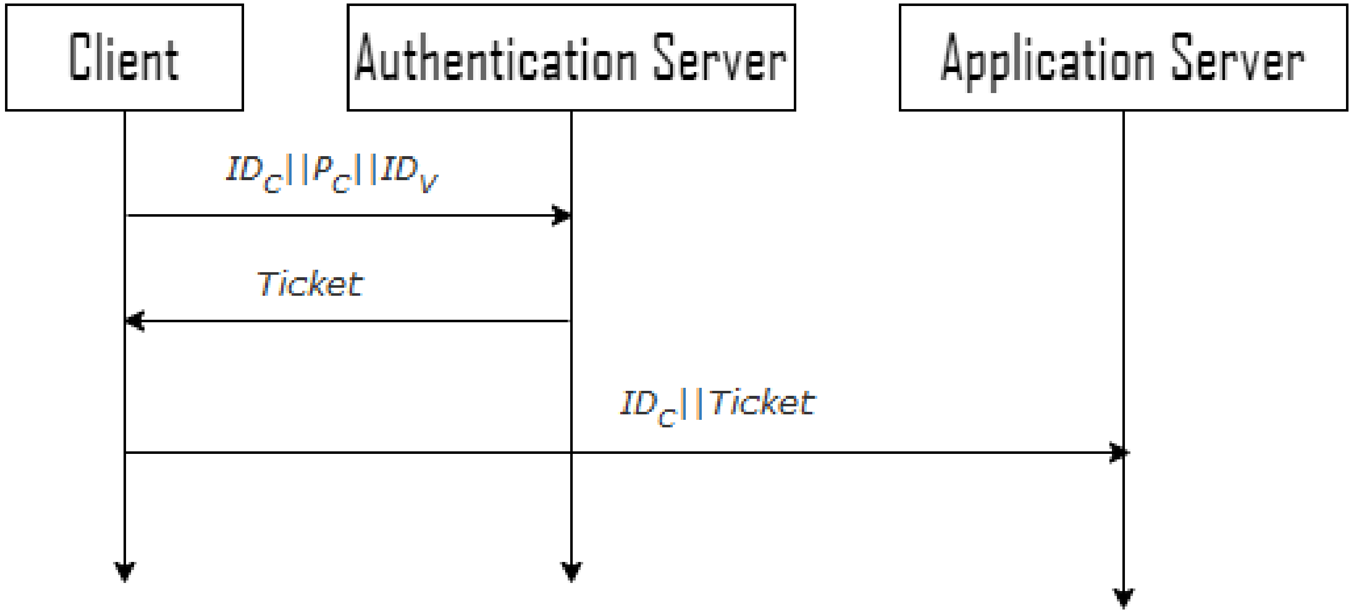
KERBEROS

- Is an authentication protocol and a KDC.
- Used by most popular systems including Windows server 2000.
- Named after a three-headed dog in Greek mythology that guards the gates.
- Originally designed by MIT.

KERBEROS - Servers

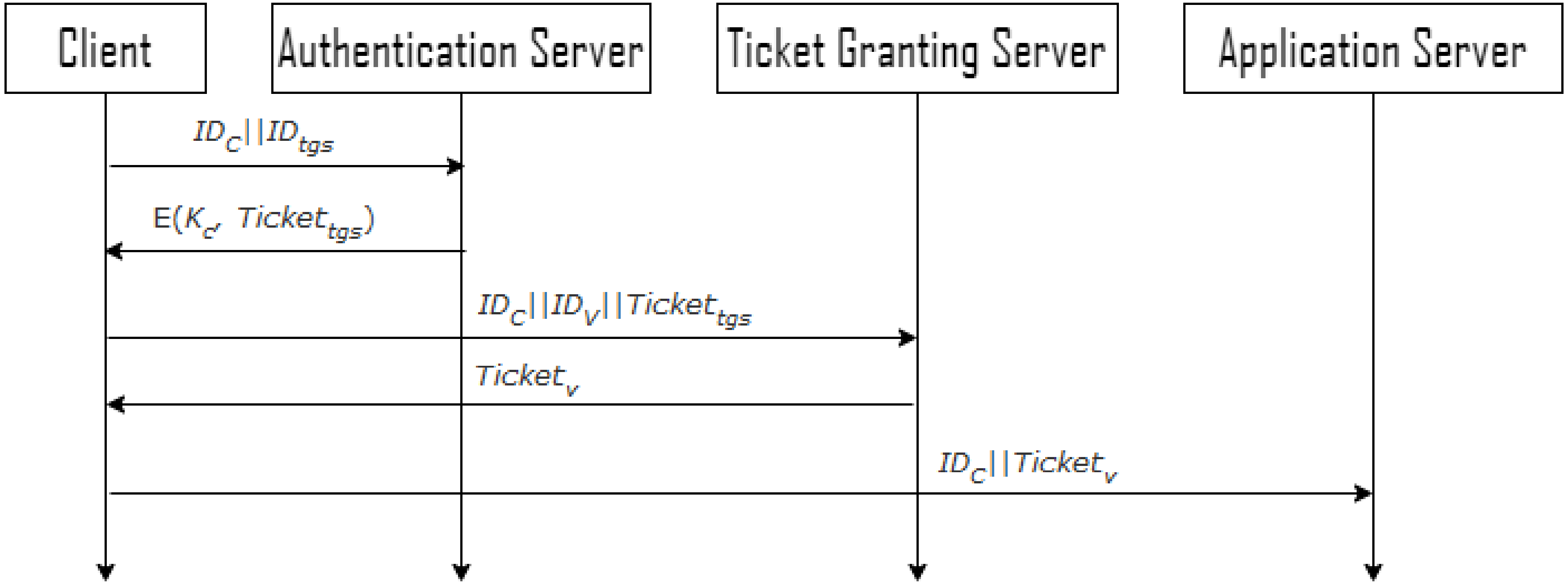
- Three Servers
  - Authentication Server
  - Ticket Granting Server
  - Application Server

A simple authentication protocol



$$Ticket = E(K_v, [ID_C || AD_C || ID_V])$$

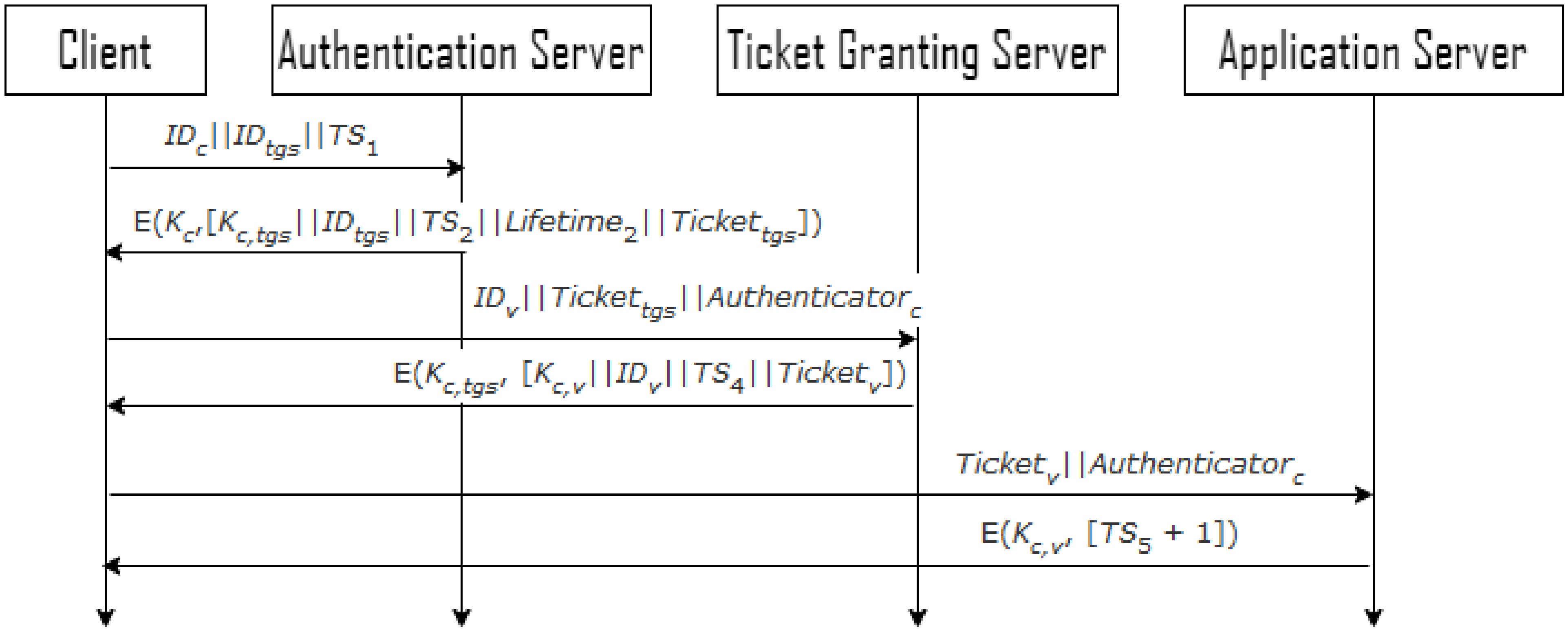
A more secure authentication protocol



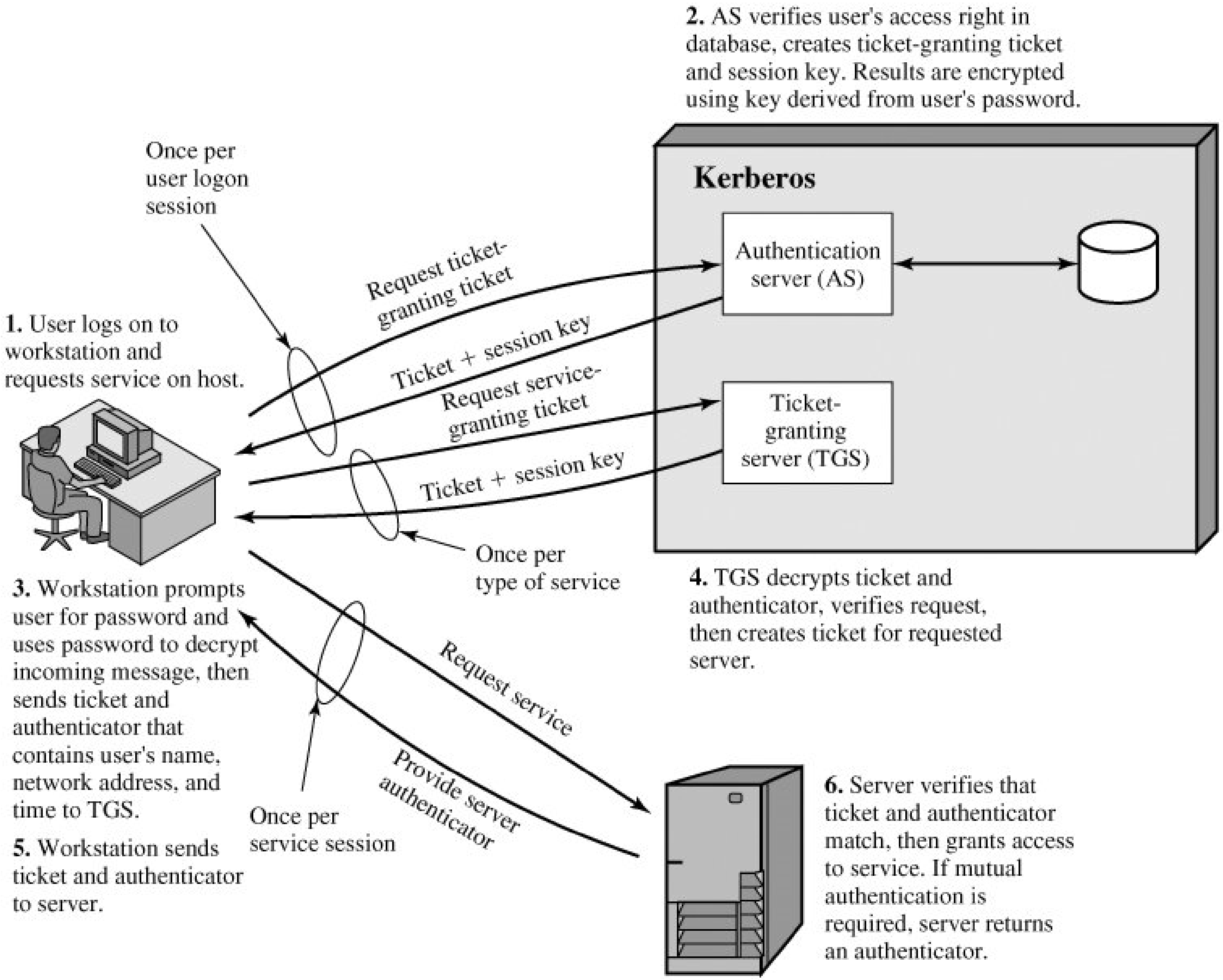
$Ticket_{tgs} = E(K_{tgs}, [ID_C || AD_C || ID_{tgs} || TS_1 || Lifetime_1])$

$Ticket_v = E(K_v, [ID_C || AD_C || ID_v || TS_2 || Lifetime_2])$

Kerberos 4 Authentication protocol



Kerberos


2. AS verifies user's access right in database, creates ticket-granting ticket and session key. Results are encrypted using key derived from user's password.
Kerberos
Authentication server (AS)
Ticket-granting server (TGS)
Once per user logon session
Request ticket-granting ticket
Ticket + session key
Request service-granting ticket
Ticket + session key
Once per type of service
1. User logs on to workstation and requests service on host.
3. Workstation prompts user for password and uses password to decrypt incoming message, then sends ticket and authenticator that contains user's name, network address, and time to TGS.
4. TGS decrypts ticket and authenticator, verifies request, then creates ticket for requested server.
Request service
Provide server authenticator
Once per service session
5. Workstation sends ticket and authenticator to server.
6. Server verifies that ticket and authenticator match, then grants access to service. If mutual authentication is required, server returns an authenticator.

By the end of this session…

- Describe the concept of symmetric key agreement and discuss the two common methods of session key creation.

Symmetric key agreement

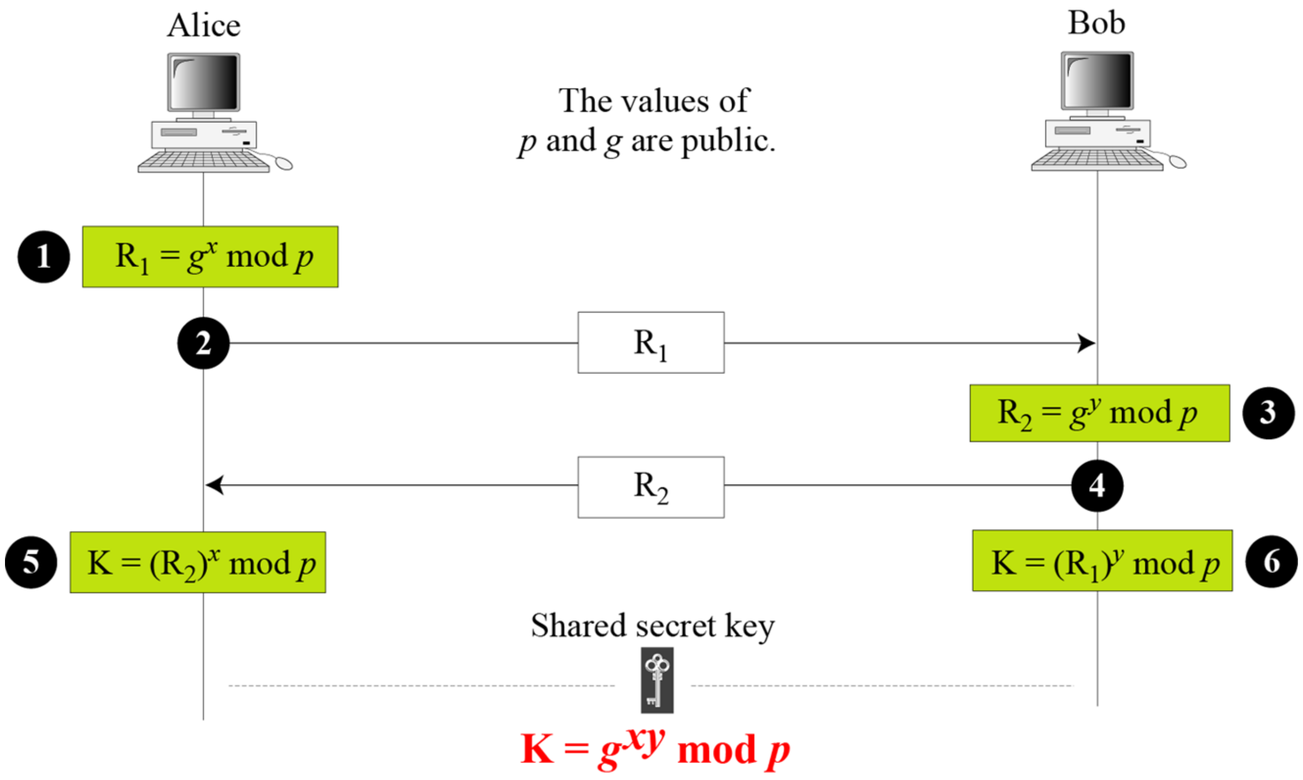
- Alice and Bob can create a session key between themselves without using a KDC.
- Two common methods:
  - Diffie-Hellman key agreement protocol
  - Station-to-Station protocol

Diffie-Hellman Key agreement protocol

- Need to choose two public global parameters p and g.
  - p is a large prime
  - g is a generator of order p-1.

- Selects a private key to generate public key based on discrete logarithm problem.

Contd...



1. Alice chooses a large random number x and calculates
$$R_1 = g^x \bmod p$$
2. Bob chooses another large random number y and calculates
$$R_2 = g^y \bmod p$$
3. Alice sends R_1 to Bob and Bob sends R_2 to Alice.
4. Alice calculates $k = R_2^x \bmod p$.
5. Bob calculates $k = R_1^x \bmod p$.

Example

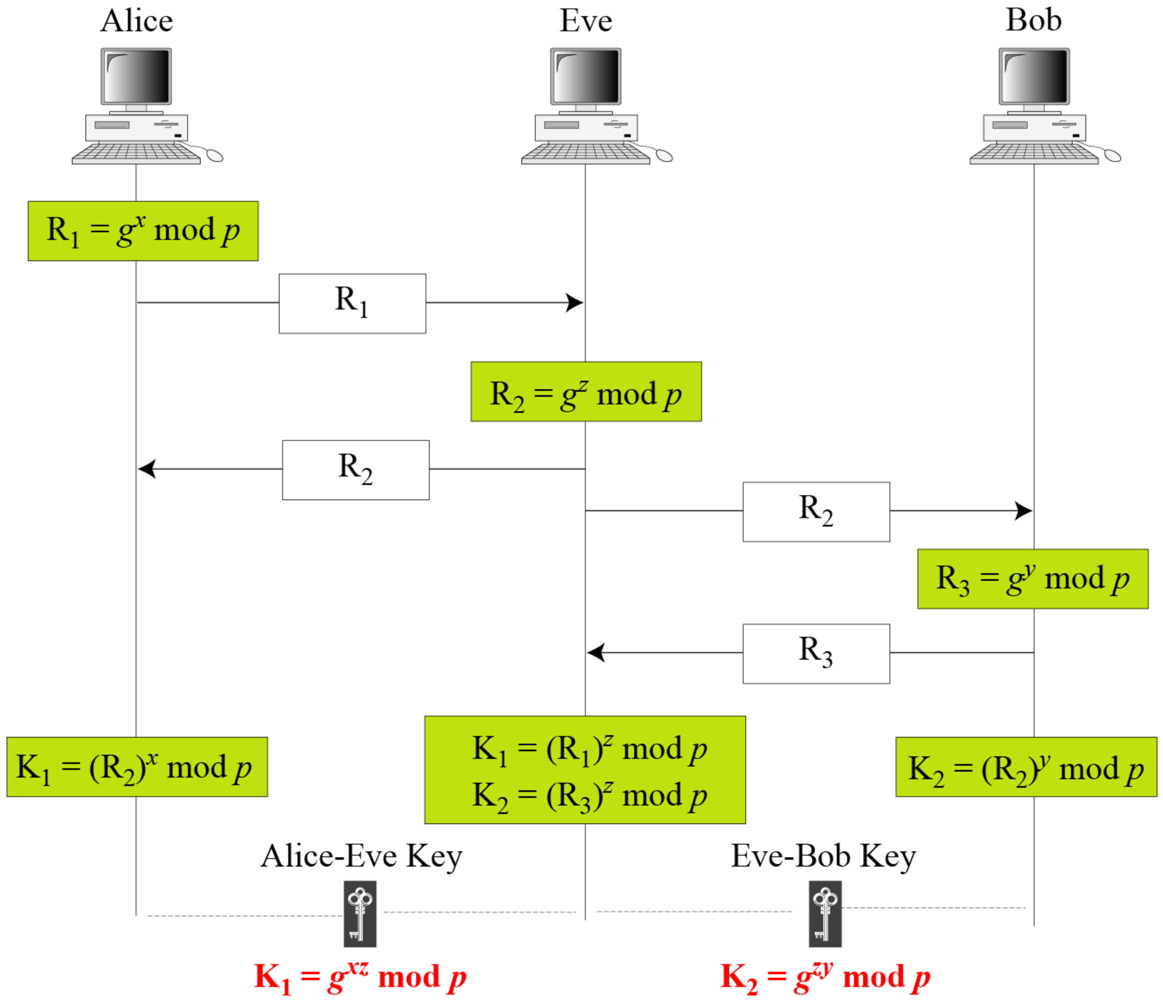
Assume that *g* = 7, *p* = 23, x=3 and y=6. The steps are as follows:

1. Alice chooses *x* = 3 and calculates R_1 = 73 mod 23 = 21.
2. Bob chooses *y* = 6 and calculates R_2 = 76 mod 23 = 4.
3. Alice sends the number 21 to Bob.
4. Bob sends the number 4 to Alice.
5. Alice calculates the symmetric key K = 43 mod 23 = 18.
6. Bob calculates the symmetric key K = 216 mod 23 = 18.
7. The value of K is the same for both Alice and Bob; g^{xy} mod *p* = 718 mod 35 = 18.

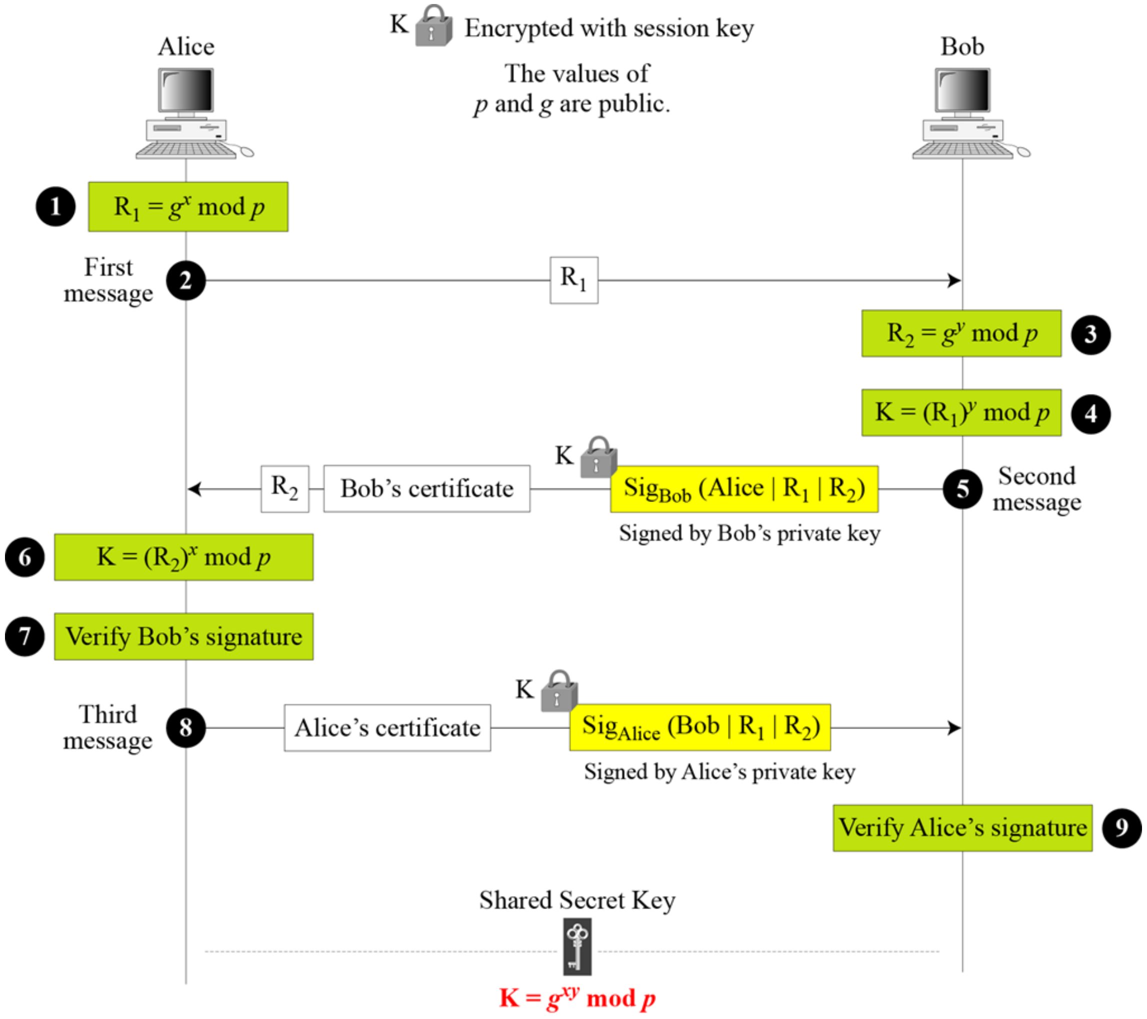
Attacks on Diffie – Hellman: Discrete Logarithm Attack

- If eve can get x from R1 and y from R2, He can easily get secret key k.
- To make Diffie – Hellman strong:
  - The prime p must be very large (morethan 300 decimal digits).
  - The prime p must be choosen such that p-1 has atleast one large factor.
  - The generator must be choosen from the group $< Z_p^*, \times >$
  - Bob and Alice must destroy x and y after they calculated k.

Attacks on Diffie – Hellman: Man in the middle attack



Station – to – Station Key Agreement



By the end of this session…

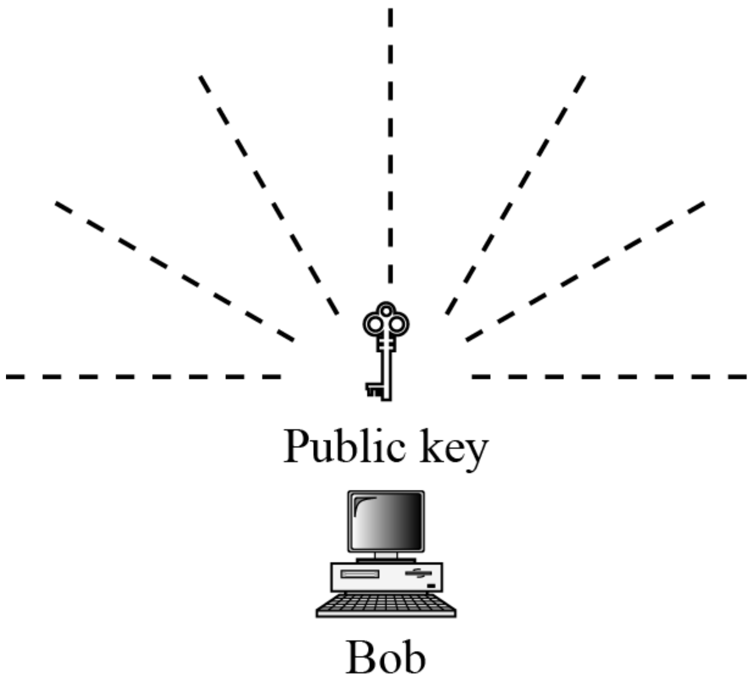
- Summarize the idea of public-key distribution and explain its duties.

Public key Distribution

- In asymmetric-key cryptography, everyone has access to everyone's public key
- Public keys are available to public

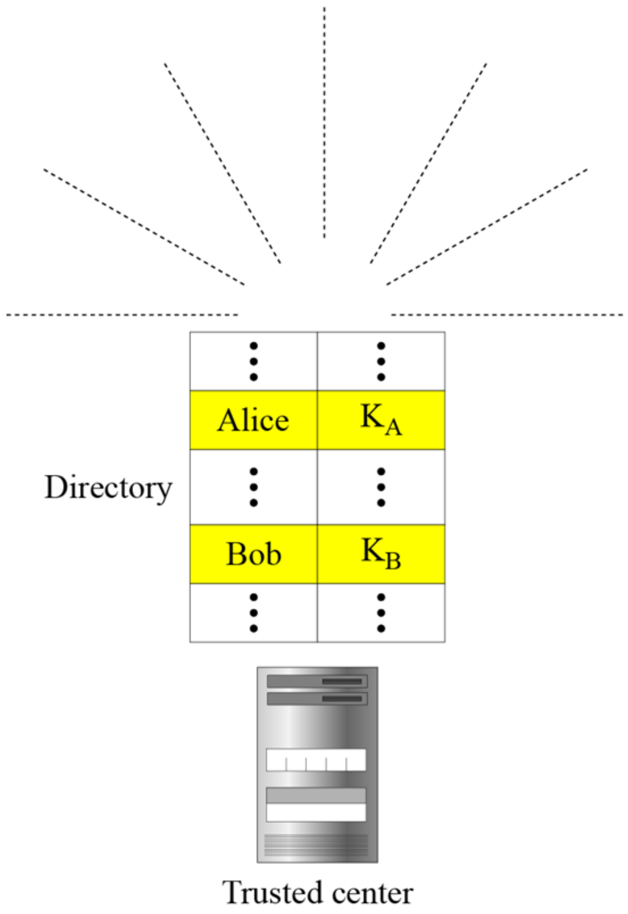
- Public Announcement
- Trusted Center
- Controlled Trusted Center
- Certification Authority
- X.509
- Public Key Infrastructure

Public Announcement



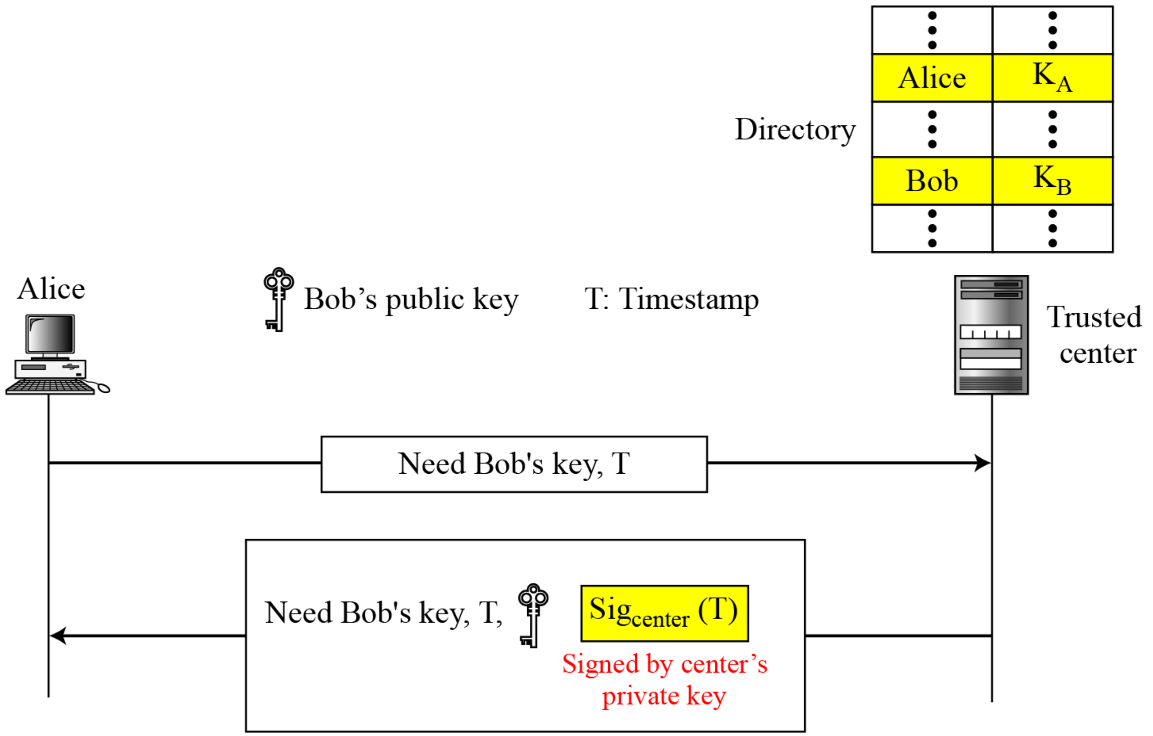
- Not secure
  - Subject to forgery
  - Eve can sign a document claiming as Bob
  - Also vulnerable if Alice directly requests Bob's public key.

Trusted center



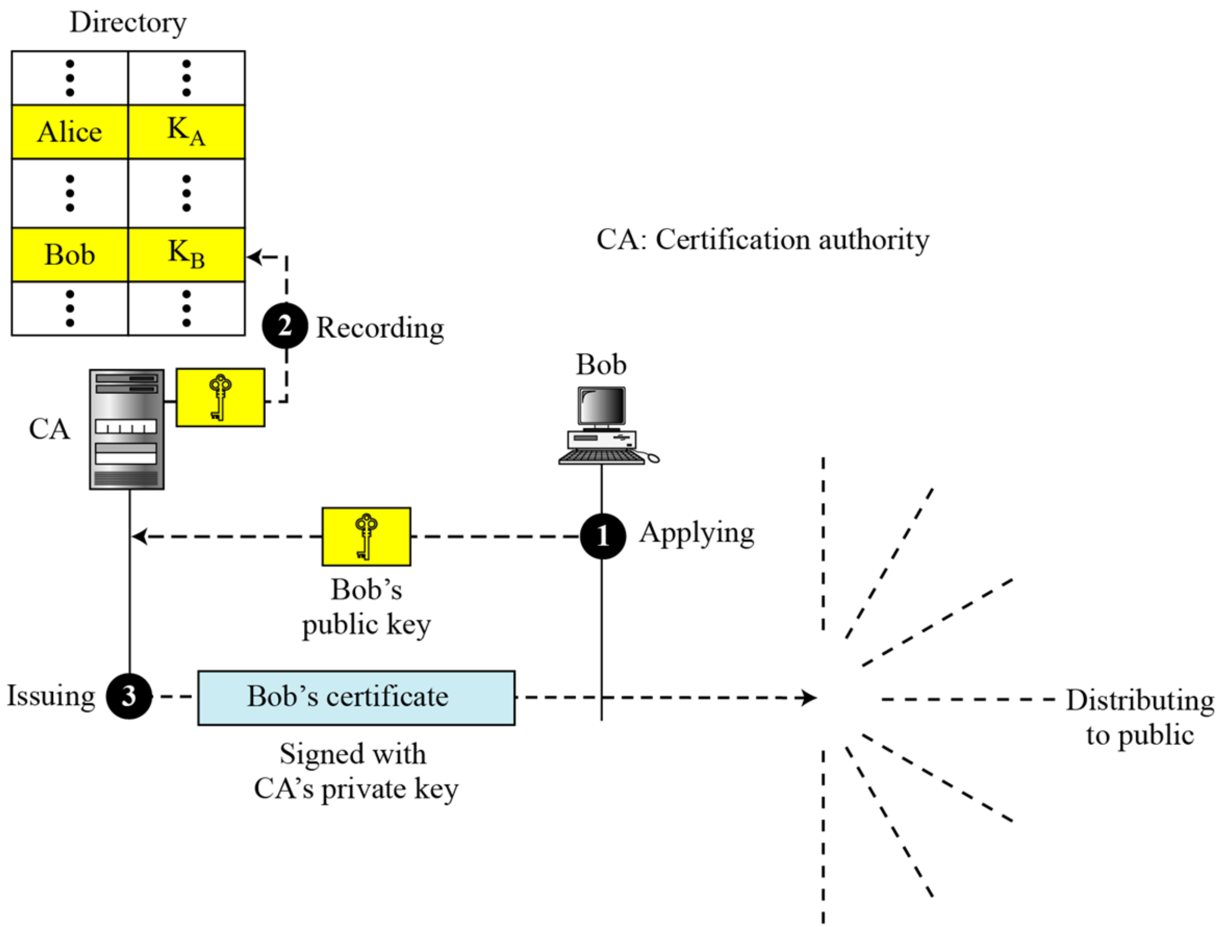
- Trusted center retains a directory of public keys.
- The directory is dynamically updated.
- Users generates their public and private keys, and registers in the trusted center with their public keys.
- The directory can be publicly advertised by the trusted center.

Controlled Trusted Center



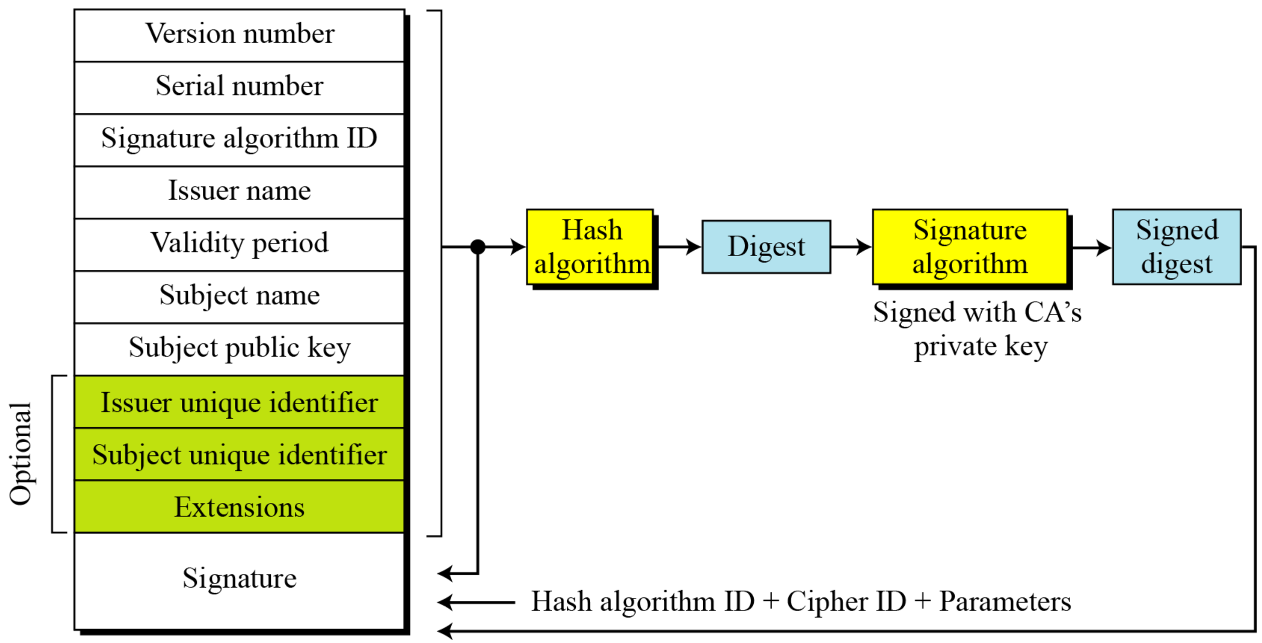
- A higher level security is needed while announcing the public key.
  - Interception
  - Modification of response
- Includes a timestamp and a signature of the authority.
- Creates a heavy load on the Trusted center.

Certification Authority



- Bob wants two things:
  - Wants people to know his public key.
  - Wants no one to accept a forged public key as his.
- Certification Authority as a state organization or federal issues certificates that binds a public key with an entity.

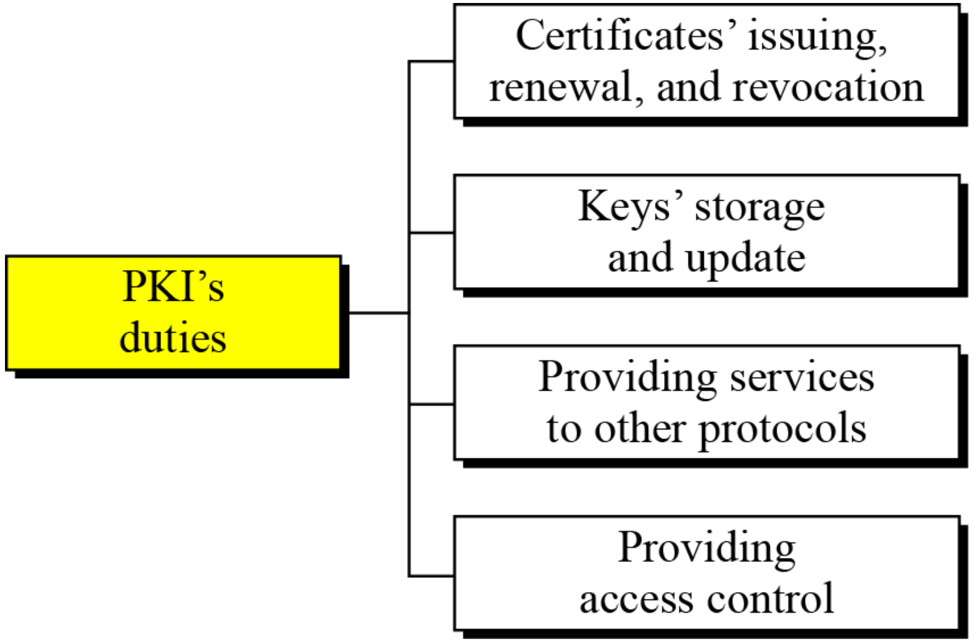
X.509



- Requires a universal format for certificate.
- The ITU has designed X.509, as a standard gives universal format for certificates.

- Maintenance of certificate:
  - Certificate Renewal
  - Certificate Revocation
  - Delta Revocation

Public Key Infrastructure



- Is a model for creating, distributing and revoking certificates based on X.509.
- Created by IETE.


Thank you!