Basic Concepts(I)

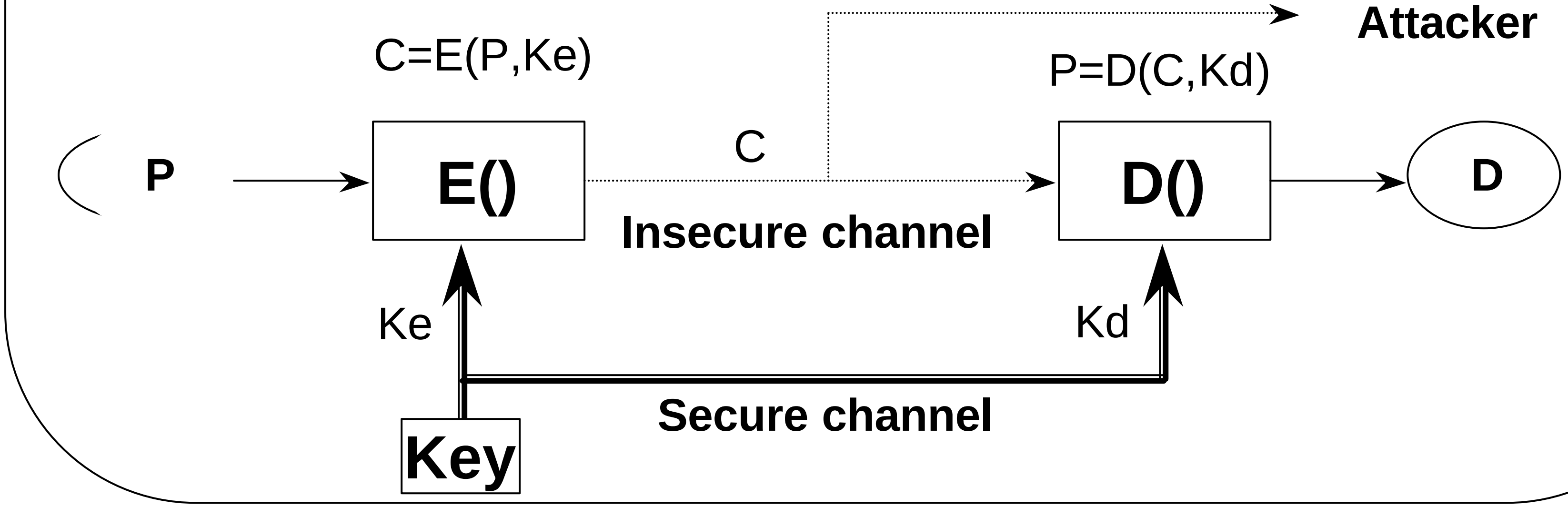
- ✍ **Cryptology**

  = Crypto(Hidden) + Logos (word)

  = Cryptography + Cryptanalysis

  = Code Writing + Code Breaking

- ✍ **Encryption(Decryption),Key,Plaintext,Ciphertext, Deciphertext**



Basic Concept(II)

- ✍ Channel
  - – Secure : trust, registered mail, tamper-proof device
  - – Insecure : open, public channel
- ✍ Entity
  - – Sender (Alice)
  - – Receiver (Bob)
  - – Adversary (Charlie)
    - ✍ Passive attack : wiretapping ->Privacy
    - ✍ Active attack : modification,impersonation

      -> Authentication

Basic Concepts(III)

- ✍ **Classification of cryptoalgorithms**
  - – by date
    - ✍ Traditional(~19C) : Ceaser
    - ✍ Mechanical(WW I, II) : Rotor Machine, Purple
    - ✍ Modern('50~) : DES, IDEA, AES
  - – by number of keys
    - ✍ Conventional : {1,single,common} key, symmetric
    - ✍ Public key cryptosystem : {2,dual} keys, asymmetric
  - – by size of plaintext
    - ✍ Block Cipher
    - ✍ Stream Cipher

Classification of Security

- ✍ **Unconditionally secure : unlimited power of adversary, perfect (Ex : one-time pad)**
- ✍ **Complexity-theoretic secure : complexity theoretic, adversary with polynomial-time power**
- ✍ **Provably secure**
- ✍ **Computationally secure**
- ✍ **Feasible secure**

Block Cipher

- **Characteristics**
  - **Based on Shannon's Theorem(1949)**
    - **Repetitive use of Confusion (Substitution) and Diffusion (Permutation)**
    - **Iteration : Weak -> Strong**
  - **Same P => Same C**
  - **{|P| = |C|} ? 64 bit, |P| ? |K| ? 56 bit**
  - **Memoryless configuration**
  - **Operate as stream cipher depending on mode**
  - **Shortcut cryptanalysis (DC, LC etc) in 90's**

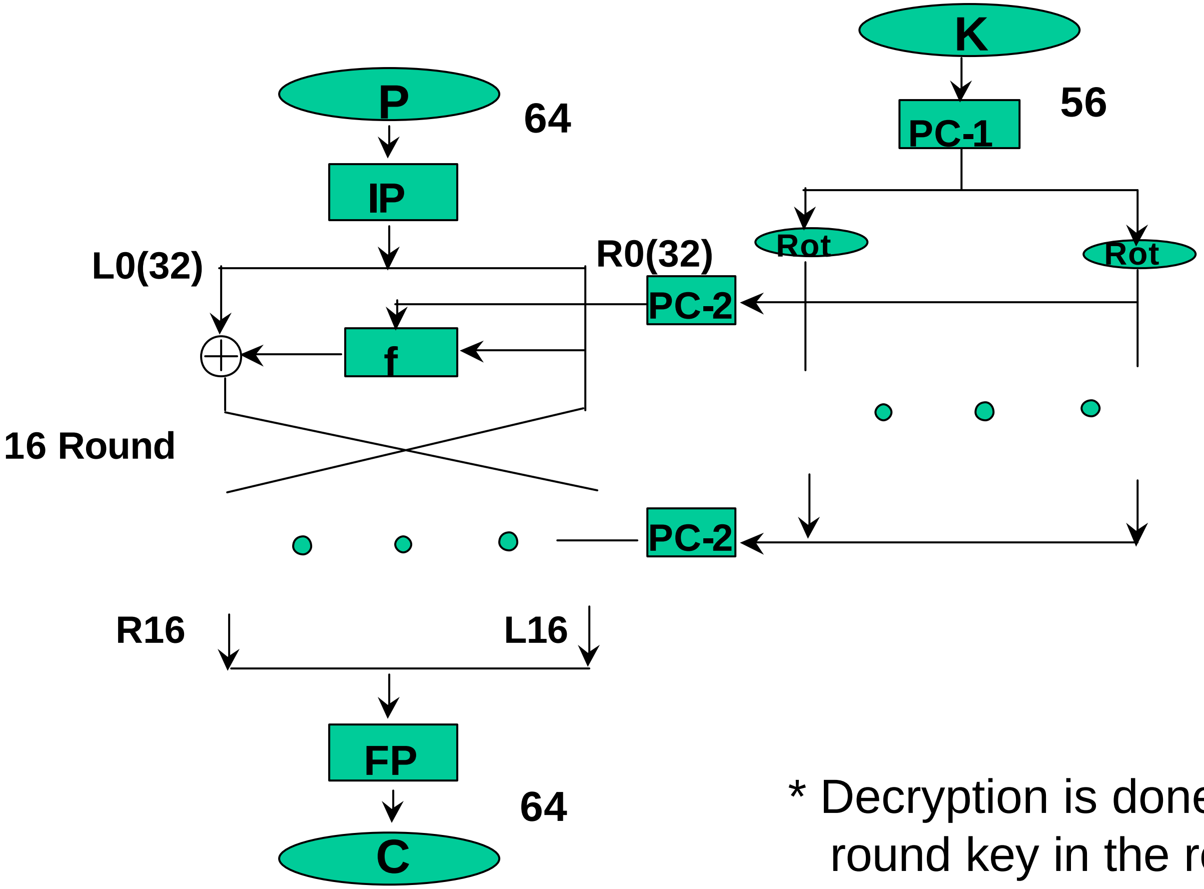
*** DC :Differential Cryptanalysis, LC : Linear Cryptanalysis**

Design Criteria of DES

- Provide a high level of security
- Completely specify and easy to understand
- Security must depend on key, not algorithm
- Available to all users
- Adaptable for use in diverse applications
- Economically implementable in electronic device
- Efficient to use
- Able to be validated
- Exportable

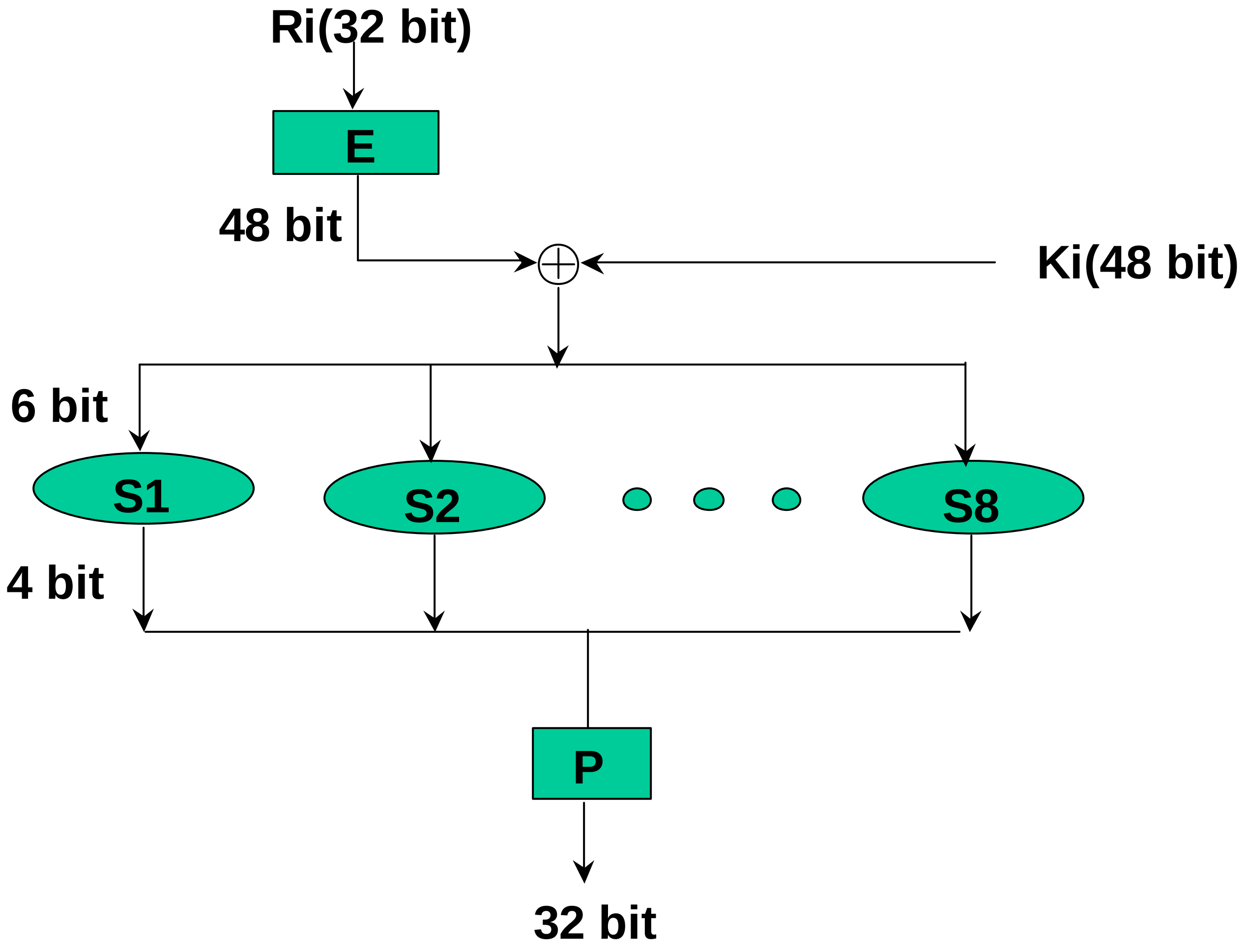
*** Federal Register, May 15, 1973**

Structure of DES



* Decryption is done by executing round key in the reverse order

f-function of DES



Criticism of DES

- Short key size : 112 -> 56 bits by NSA
- Classified design criteria
- Hidden trapdoor
- Revision of standard every 5 yrs after 1977 by NIST

DES Key Search Machine

- ✍ **Diffie & Hellman ('77)**
  - – 10^6 keys/sec VLSI
  - – Cost = $20,000,000
- ✍ **Wiener ('93)**
  - – 5 x 10^7 keys/sec
  - – 1 Frame :10$/VLSI x 5,760 =$100,000
  - – 10 Frames : $1,000,000
  - – 3.5hr in average

DES Challenge(I)

- **RSA Data Security Inc's protest against US's export control('97)**
  - $10,000('97) award
  - Key search machine by Internet Loveland's Rocker Verser
  - 60.1 Billion/1 day Key search, Succeed in 18 quadrillion operations and 96 day
    - 25% of Total 72 quadrillion (1q=10^{15} =0.1경)
    - 90MHz, 16MB Memory Pentium(700 Million/sec)
  - http://www.rsa.com/des/

DES Challenge(II, III)

- ✍ Distributed.Net + EFF
  - – 100,000 PC on Network
  - – 56hr
- ✍ EFF
  - – http://www.eff.org/DES cracker
  - – Specific tools
  - – 22hr 15min
  - – 250,000$

Strengthening DES

- Key size expansion
  - Double Encryption
    - e_k:$E_2(K_2,E_1(K_1,P))$, d_k:$D_1(K_1,D_2(K_2,C))$
    - Meet-in-the-middle attack
    - No increase of practical key size
  - Triple Encryption
    - e_k:$E(K_1,D(K_2,E(K_1,P)))$, d_k:$D(K_1,E(K_2,D(K_1,C)))$
    - e_k:$E(K_1,D(K_2,E(K_3,P)))$, d_k:$D(K_3,E(K_2,D(K_1,C)))$
    - 112 or 168 bits

Summary of block ciphers

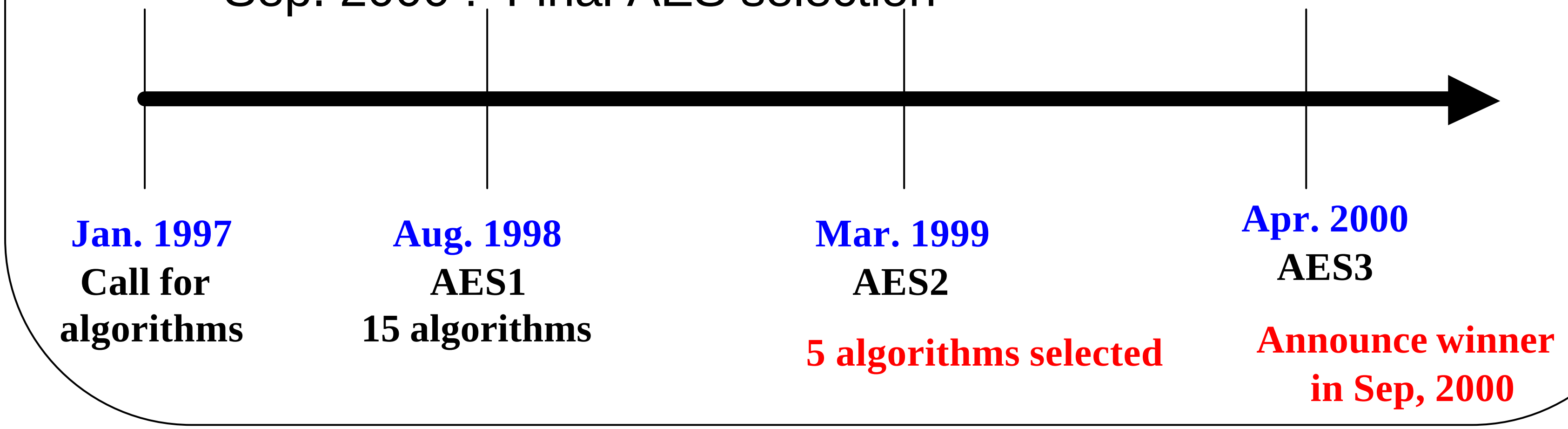
Algorithm	Year	Country	Pt/Ct	Key	Round
DES	1977	USA	64	56	16
FEAL	1987	Japan	64	64	4,8,16,32
GOST	1989	Russia	64	256	32
IDEA	1990	Swiss	64	128	8
LOKI	1991	Australia	64	64	16
SKIPJACK	1990	USA	64	80	32
MISTY	1996	Japan	64	128	>8
SEED	1998	Korea	128	128	16

AES requirements

- Block cipher
  - 128-bit blocks
  - 128/192/256-bit keys
- Worldwide-royalty free
- More secure than Triple DES
- More efficient than Triple DES

AES Calendar

- Jan. 2, 1997 : Announcement of intent to develop AES and request for comments
- Sep. 12, 1997 : Formal call for candidate algorithms
- Aug. 20-22, 1998 : First AES Candidate Conference and beginning of Round 1 evaluation (15 algorithms), Rome, Italy
- Mar. 22-23, 1999 : Second AES Candidate Conference, NY, USA
- Sep. 2000 : Final AES selection



Jan. 1997
Call for algorithms

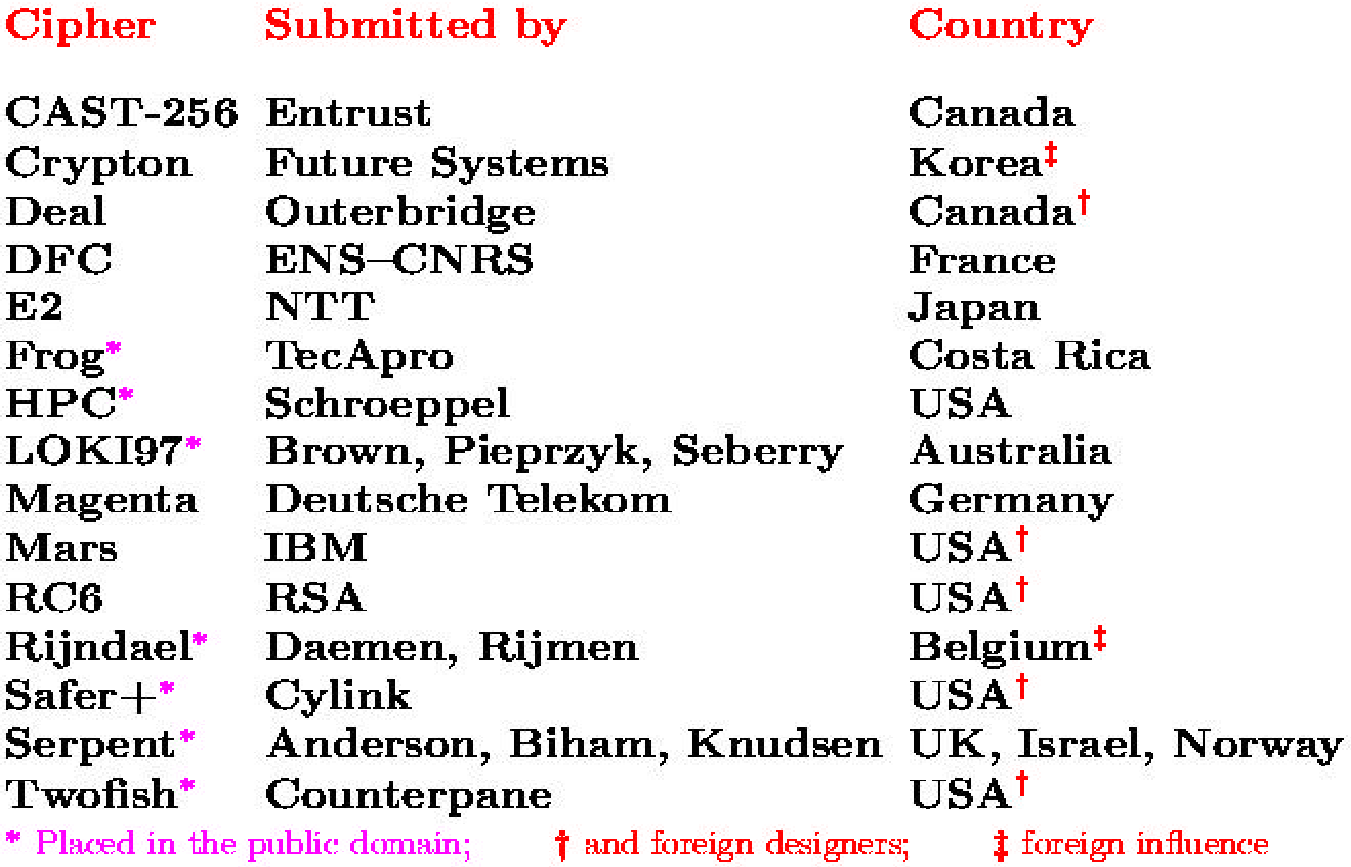
Aug. 1998
AES1
15 algorithms

Mar. 1999
AES2
5 algorithms selected

Apr. 2000
AES3
Announce winner in Sep, 2000

AES1 algorithms

- 15 algorithms are proposed at AES1 conference

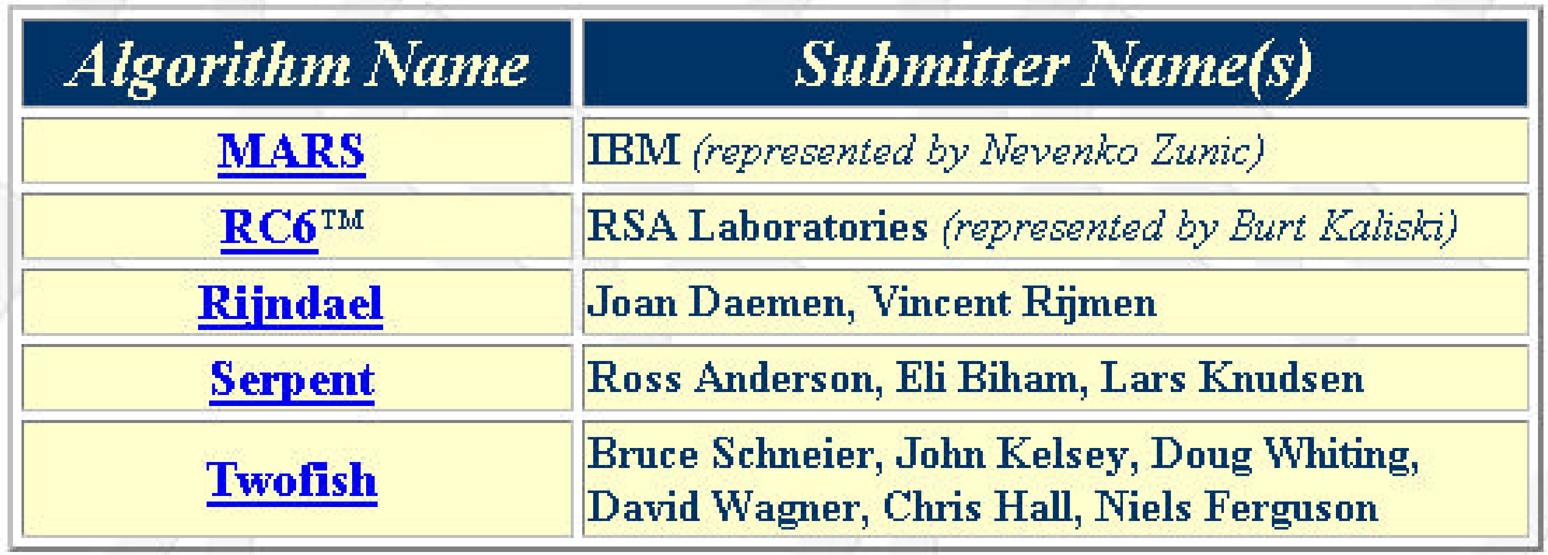


Cipher	Submitted by	Country
CAST-256	Entrust	Canada
Crypton	Future Systems	Korea‡
Deal	Outerbridge	Canada†
DFC	ENS–CNRS	France
E2	NTT	Japan
Frog*	TecApro	Costa Rica
HPC*	Schroeppel	USA
LOKI97*	Brown, Pieprzyk, Seberry	Australia
Magenta	Deutsche Telekom	Germany
Mars	IBM	USA†
RC6	RSA	USA†
Rijndael*	Daemen, Rijmen	Belgium‡
Safer+*	Cylink	USA†
Serpent*	Anderson, Biham, Knudsen	UK, Israel, Norway
Twofish*	Counterpane	USA†

* Placed in the public domain; † and foreign designers; ‡ foreign influence

AES Round 2 Algorithms

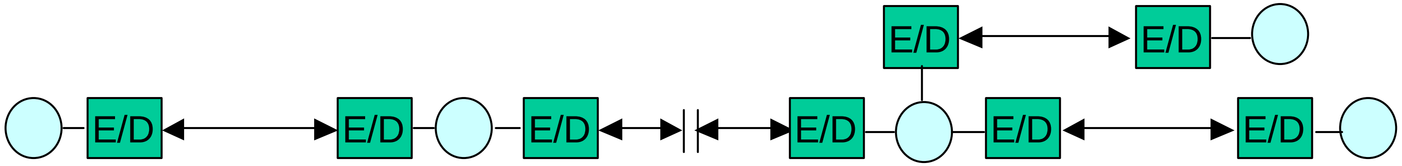
- After AES2 conference, NIST selected the following 5 algorithms as the round 2 candidate algorithm.



Algorithm Name	*Submitter Name(s)*
MARS	IBM *(represented by Nevenko Zunic)*
RC6™	RSA Laboratories *(represented by Burt Kaliski)*
Rijndael	Joan Daemen, Vincent Rijmen
Serpent	Ross Anderson, Eli Biham, Lars Knudsen
Twofish	Bruce Schneier, John Kelsey, Doug Whiting, David Wagner, Chris Hall, Niels Ferguson

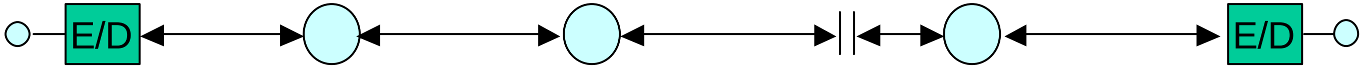
Operation of E/D device

(1) link-by-link



Ex : M/W Link, Satellite Link etc

(2) end-to-end



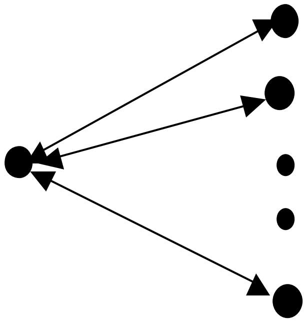
Ex : Telephone, Fax, Data Terminal etc

(3) Hybrid operation: (1) + (2)

Problem of Symmetric Cryptosystems

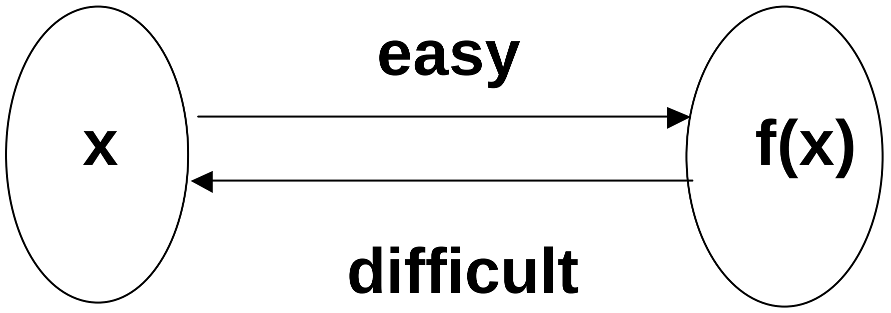
- **<u>Key management</u>**
  - **Keep secret key in secret**
  - **Over complete graph with *n* nodes,**

    $_{n}C_{2} = n(n-1)/2$ pairs secret keys are required.
  - **(Ex) n=100, 99 x 50 = 4,950 keys**



Concepts of PKC(I)

- ✍ **1-way ft.**
  - ✍ **Given x, easy to compute f(x).**
  - ✍ **Difficult to compute $f^{-1}(x)$ for given f(x).**

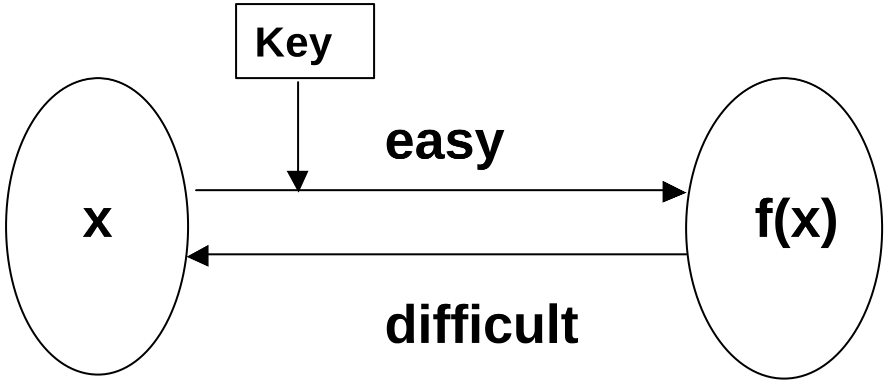


Ex) $f(x)= x^5 + x^3 + x^2+1$

Concepts of PKC(II)

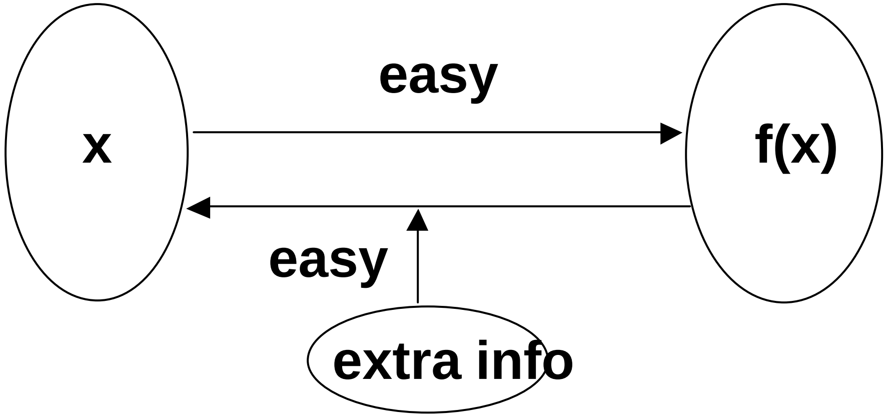
- ✍ **Keyed 1-way ft :**

  1-way ft with a key



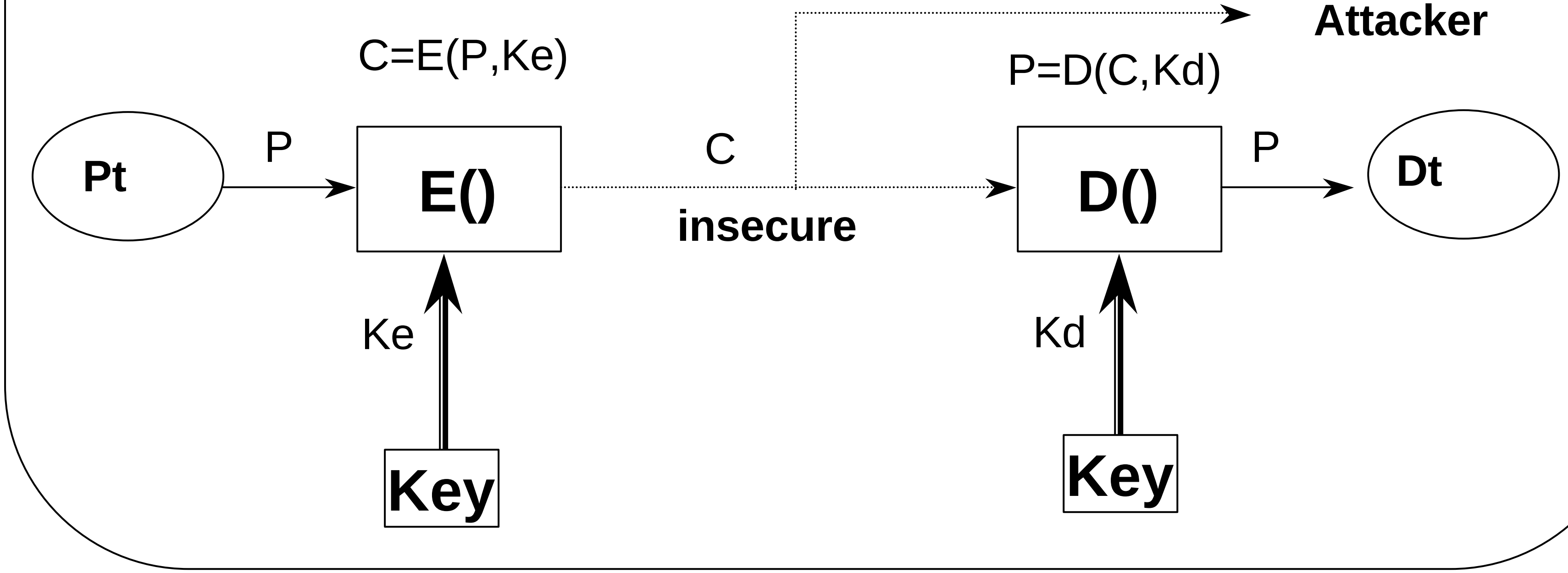
Concepts of PKC(III)

- 1-way trapdoor ft.
  - Given x, easy to compute f(x)
  - Easy to compute $f^{-1}(x)$ for given f(x) and some information -> trapdoor information



Concepts of PKC(IV)

- ✍ Use two keys
  - ✍ Given public key, easy to compute -> anyone can lock.
  - ✍ Only those has secret key, compute inverse -> only who has it can unlock, vice versa.



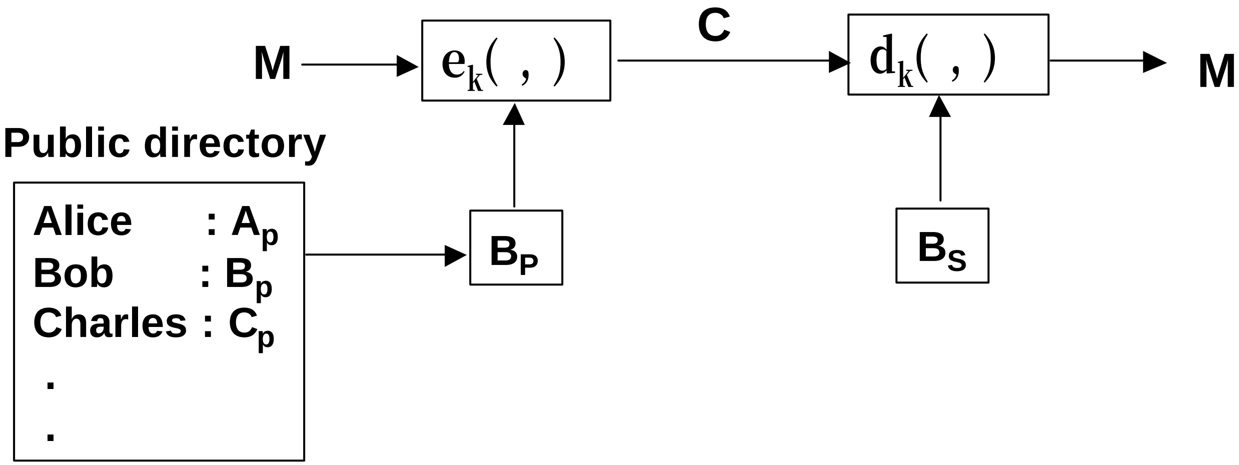
What service PKC provides ?(I)

✍ For Privacy

- Encrypt M with Bob’s public key : C = $e_K(B_p,M)$

- Decrypt C with Bob’s private key : D = $d_K(B_s,C)$

*Anybody can generate C, but only B can recover C.



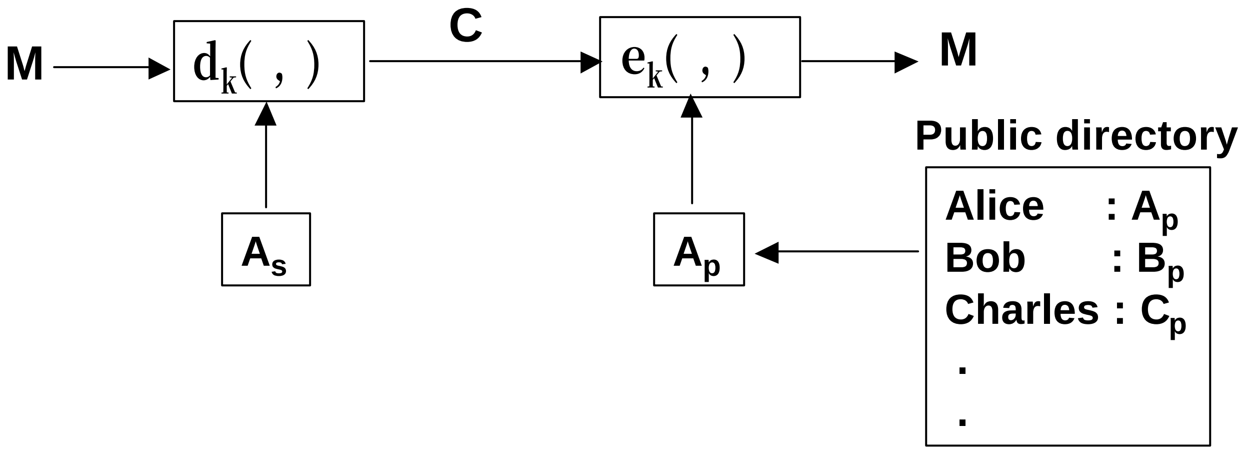
What service PKC provides ?(II)

✍ **For authentication(Digital Signature)**

- Encrypt M with Alice's private key : C = $d_K(A_s, M)$

- Decrypt C with Alice's public key : D = $e_K(A_p, C)$

* Only Alice can generate C, but anybody can recover C.



What service PKC provides ?(III)

- Identification
- Non-Repudiation
- Applicable to various cryptographic protocols
- Hybrid use with symmetric cryptosystem

Comparision

O : merit
X : demerit

Cryptosystem Item	Symmetric		Asymmetric
Key relation	Enc. key = Dec. key		Enc. Key ? Dec. key
Enc. Key	Secret		Public,{private}
Dec. key	Secret		Private,{public}
Algorithm	Secret	Public	Public
Typical ex.	Skipjack	DES	RSA
Key Distribution	Req'd (X)		Not req'd (O)
Number of keys	Many(X),keep many partners' secret key		Low(O), keep his pri. Key only
Secure authentication	Hard(X)		Easy(O)
E/D Speed	Fast(O)		Slow(X)

RSA Scheme(I)

- **For large 2 primes *p,q***
- ***n=pq , ?(n)=(p-1)(q-1) : Euler phi ft.***
- **Select random *e* s.t. gcd(?*(n), e*) = 1**
- **Compute *ed* = 1 mod ?*(n)* -> *ed* = *k?(n) +1***
- **Public key = {e, n}, secret key = {d, {n}}**
- **For given M in [0, n-1],**
- **Encryption, $C = M^e$ mod n**
- **Decryption, $D = C^d$ mod n**

(Proof) $C^d = (M^e)^d = M^{ed} = M^{k?(n) +1} = M \{M^{?(n)}\}^k = M$

RSA Scheme(II)

- ***p*=3, *q*=11**
- ***n* = *pq* = 33, ?*(n)* =(*p*-1)(*q*-1)=2 x10 = 20**
- ***e* = 3 *s.t*. gcd(*e*, ?*(n)*)=(3,20)=1**
- **Choose d *s.t*. *ed* =1 mod?*(n),* 3d=1mod 20, d=7**
- **Public key ={e,n}={3,33}, private key ={d}={7}**

- **M =5**
- **C = M^{e} mod n = 5^{3} mod 33 =26**
- **M =C^{d} mod n = 26^{7} mod 33= 5**

Requirements of Digital Signature

- ✍ **Efficiency**
- ✍ **Unforgeability : only signer can generate**
- ✍ **Authentication of a signer:**
- ✍ **Not reusable : not to use for other message**
- ✍ **Unalterable : No modification of signed message**
- ✍ **Non-repudiation : not denying the act of signing**

Elements of Digital Signature

- Consists of 6 elements ***(M,Mh,A,K,S,V)***
  - ***M*** **: message space**
  - ***Mh*** **(or *Ms)* : signing space**
  - ***A*** **: signature space**
  - ***K*** **: key space**
  - **For K? *K*, ? signing alg.** $\mathbf{sig_K}$ **?** ***S*** **and its corresponding verification alg.** $\mathbf{ver_K}$ **?** ***V*****.**
  - **Each** $\mathbf{sig_K}$ **:** ***M*****?** ***A*** **and** $\mathbf{ver_K}$ **:** ***M*** **x** ***A*** **? {t,f} are fts s.t.,** $\mathbf{ver_K}$**(x,y)= t if y =** $\mathbf{sig_K}$**(x) or** $\mathbf{ver_K}$**(x,y)=f if y ?**$\mathbf{sig_K}$**(x)**

Digital signature with appendix(I)

(1) Signature generation

(a) get secret key, K_s

(b) m'=h(m) : hash algorithm and $s^*=sig_{Ks}(m')$

(c) m, s^* : signature

(2) Signature verification

(a) obtain public key, Kp

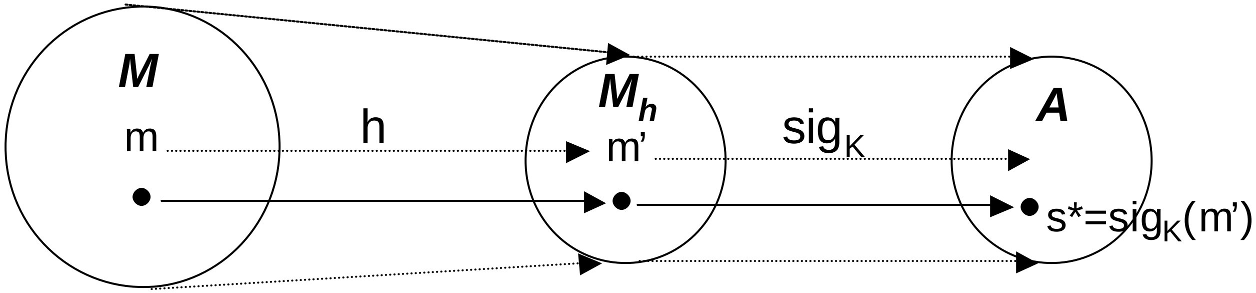
(b) compute m'=h(m) and $u=ver_{Kp}(m',s^*)$

(c) accept signature iff u=true.

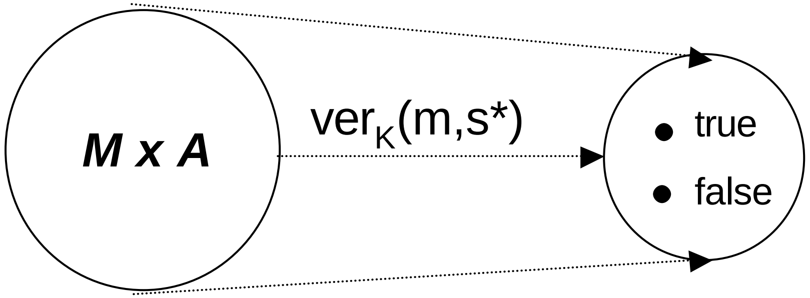
(Ex.) DSA, ElGamal, Schnorr

Digital signature with appendix(II)

(a) signing



(b) verification



Digital signature with message recovery(I)

(1) Signature generation

(a) get secret key, Ks

(b) m’=R(m) : redundancy ft and $s^{*}=sig_{Ks}(m’)$

(c) s^{*} : signature

(2) Signature verification

(a) obtain public key K_{P}

(b) compute m’= $ver_{Kp}(s^{*})$

(c) verify that m’? M_{R} (if m’ ? M_{R}, then reject)

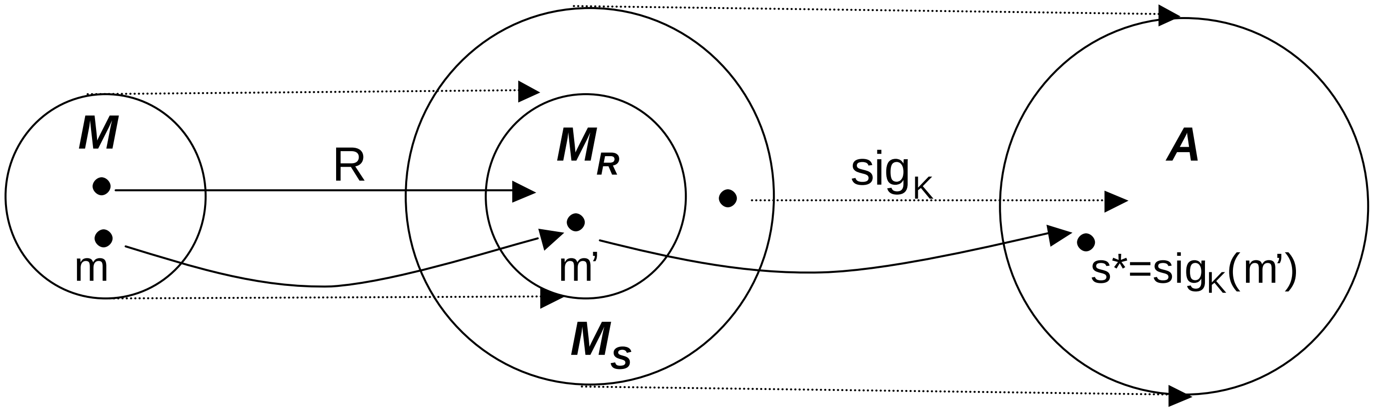
(d) recover m from m’ by computing $R^{-1}(m’)$

(Ex.) RSA, Rabin, Nyberg-Rueppel

* R() and $R^{-1}()$ are easy to compute.

Digital signature with message recovery(II)

(a) signing



(b) verification

Omitted.

R: redundancy ft
e.g., 1:1 ft
M_R : image of R

*****This scheme can be easily changed to digital signature with appendix s.t., hashing before signing**.**

Comparison of Signature

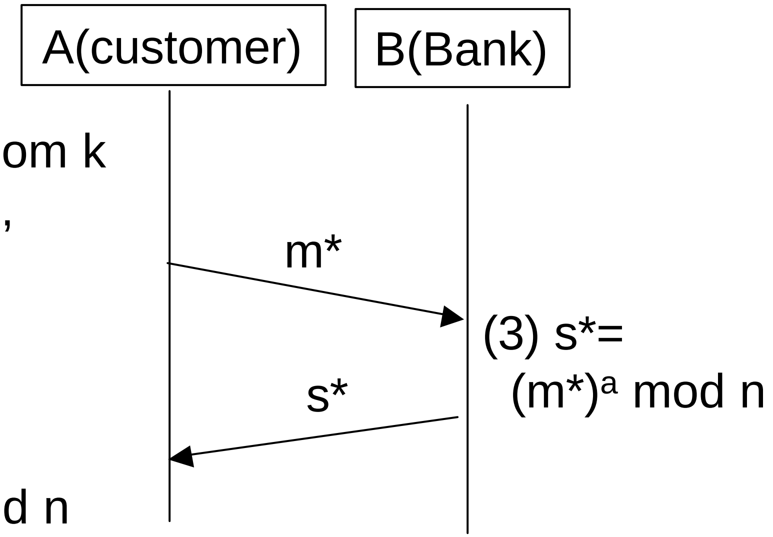
Item	Handwritten	Digital
Result of Signature	Fixed	Variable
Digital Copy	Difficult	Easy
Operation	Simple	Mathematical
Legality	Yes	Yes
Forgeability	Possible	Impossible
Tool	Pen	Computer
Auxiliary Tool	Not Necessary	Necessary(Hash ft)

Applied Digital Signature

- Blind signature
- One-time signature
  - Lamport scheme
  - Bos-Chaum scheme
- Undeniable signature
  - Chaum-van Antwerpen scheme
- Fail-stop signature
  - van Heyst-Peterson scheme
- **Group Signature :** group member can generate signature if dispute occurs, identify member.

Chaum's Blind Signature(I)

- Without B's knowing message M itself, A can get a signature of M from B.
- RSA scheme, B's public key :{n,b}, secret key:{a}



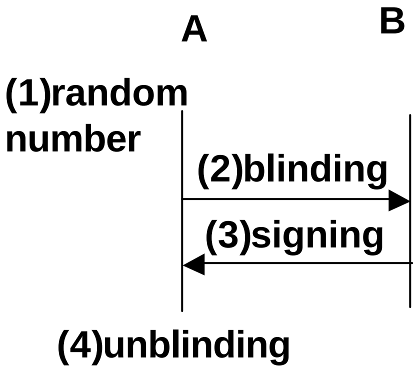
(1) select random k s.t. gcd(n,k)=1, 1<k<n-1

(2) $m^*=mk^b \bmod n$

(3) $s^*= (m^*)^a \bmod n$

(4) $s=k^{-1} s^* \bmod n$

(signature of M by B : $k^{-1}(mk^b)^a= k^{-1} m^a k^{ba} = m^a$)



$g(S_B f(m))=S_B(m)$

f:blinding ft

g:unblinding ft only A knows

f(m) : blinded message

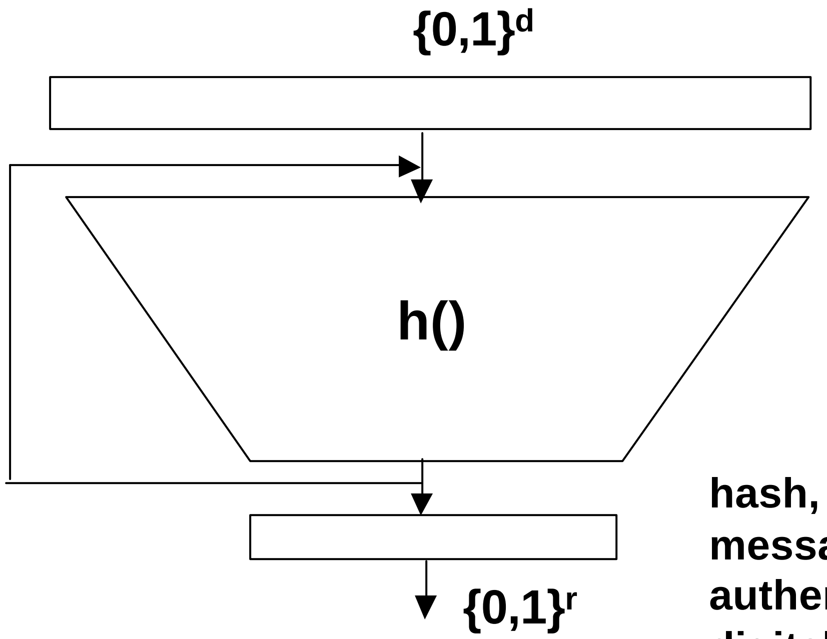
Chaum’s Blind Signature(II)

(Preparation) p=11, q=3, n=33,?(n)= 10 * 2=20
gcd(a, ?(n))=1 => a=3, ab =1 mod ?(n) => 3 b = 1 mod 20 => b=7
B’s public key :{n,b}={33,7}, secret key ={a}={3}

(1) A’s blinding of m=5
select k s.t. gcd(k,n)=1 => gcd(k,33)=1 => k=2
$m^* = m\ k^b$ mod n= 5 2^7 mod 33=640= 13 mod 33
(2) B’s signing
$s^* = (m^*)^a$ mod n = 13^3 mod 33 =2197=19 mod 33
(3) A’s unblinding
$s=k^{-1}\ s^*$ mod n (2 k^{-1}=1 mod 33 => k=17)
= 17 19 mod 33 =323=26 mod 33
* Original Signature : m^a mod n = 5^3 mod 33 =125 =26 mod 33

Hash function

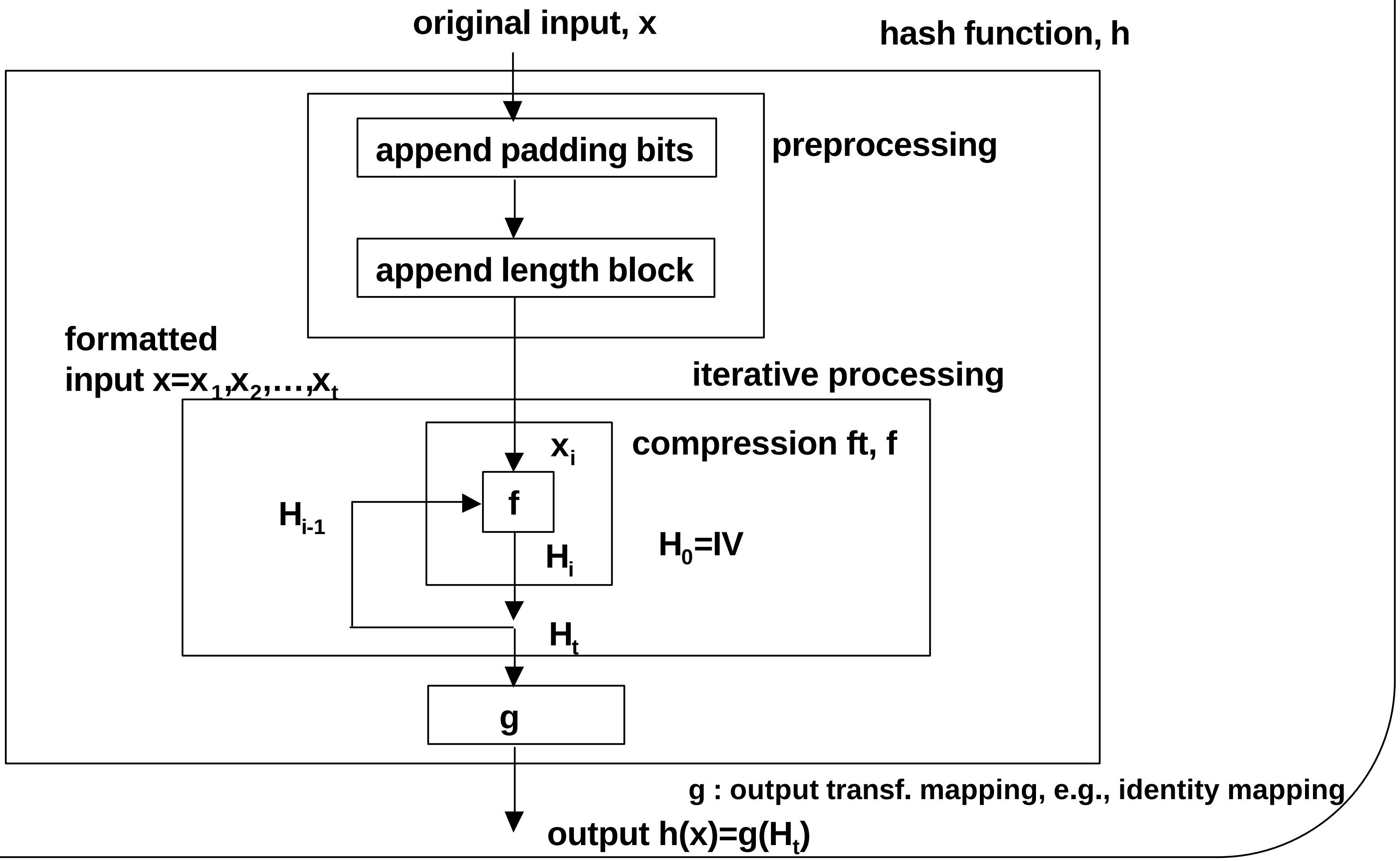
- Compress a binary string with an arbitrary length into a fixed short message
- Used for digital signature, integrity, authentication etc.



d > r

hash, hash code/value/result
message digest,checksum,MIC,
authentication tag, seal,compression
digital fingerprint, imprint

Detailed Configuration of Hash Function



Requirements of Hash function

- **Compression**
- **One-wayness**

  : If y=h(x) is given, it is computational infeasible to compute x
- **Collision-free**

  : It is computational infeasible to find a pair (x, x'), x? x' satisfying h(x)=h(x').
- **Efficiency**
  - **Easy to compute f(x) for a given x.**

Classification of Hash ft

- Keyed hash : MAC (Message Authentication Code)
- Unkeyed hash : MDC (Manipulation Detection Code),
  - 1WHF(One Way Hash Function)
  - CFHF(Collision-Free Hash Function)
- Dedicated Hash function
  - MD5, SHA-1

Summary

name	designer	year	characteristics	security
MD4	R.L.Rivest (USA)	‘91	Boolean ft 3R, 128bit	collision (‘95) 2^{20} operation
MD5	R.L.Rivest (USA)	‘92	Boolean ft 4R, 128bit	primitive ft’s collision(‘96)
HAVAL	Y.Zheng (Australia)	‘92	expand MD5 3,4,5R/128,160,192,224,256bit	
SHS	NIST	‘91	Boolean ft Modified MD4, 4R,160bit	
HAS -160	KISA (Korea)	‘98	Boolean ft	

Applications

- **Used together with a signature scheme**
- **Integrity service for MIC (Message Integrity Code) (Ex: anti-virus)**
- **passwd ft in UNIX OS**
- **Keyed Hash Ft (MAC)**
- **Identification in Challenge-response protocol**