Cryptography was the art of secret coding

Computer & Network Security
Lecture 8 : Message Authentication and Hash Functions

Nur Azman Abu
FTMK, UTeM
Mobile : 0177722007
Email : nura@utem.edu.my

Tuesday 16 September 2008

Content

- Message Authentication
- Data Authentication Algorithm

- Elements of Crypto-System
- Hashing Functions
- Secure Hashing Algorithm
- Random Numbers
- SHA-2 Design
- Birthday Paradox

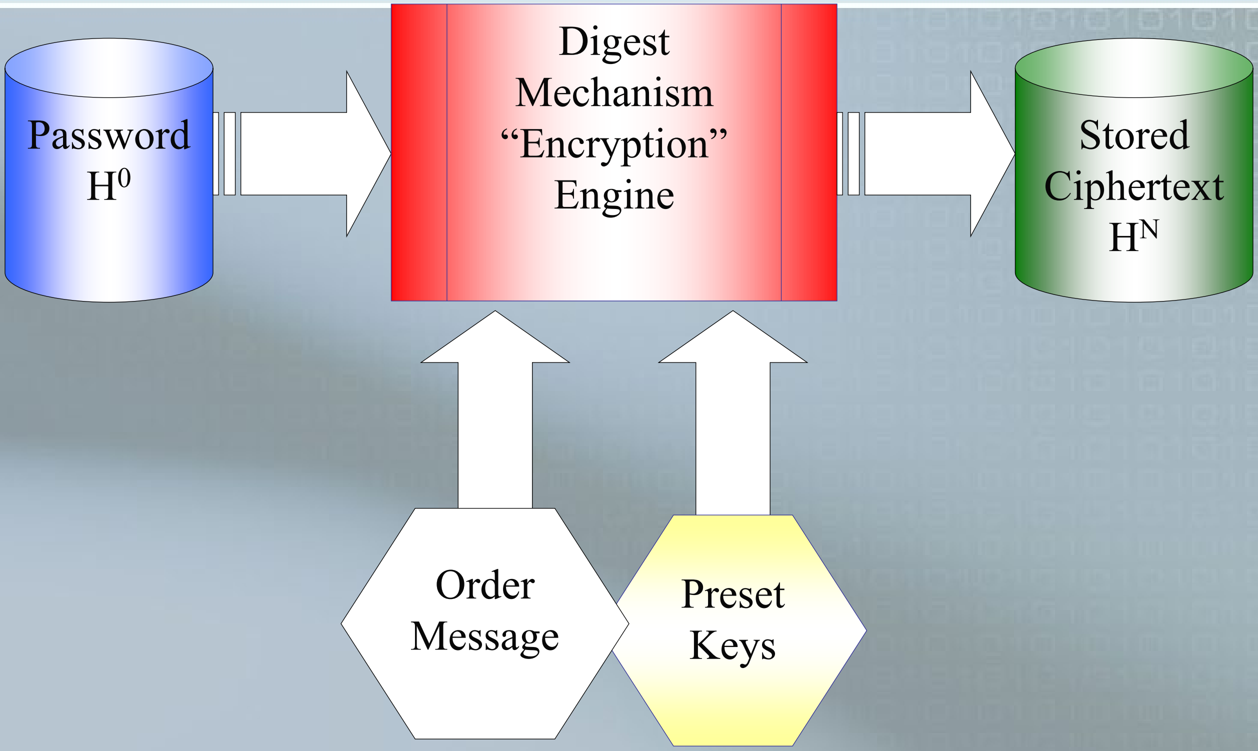
Please …..read….. Dictionary Attack!

Asymmetric

Symmetric

OTK

SHA-2 means SHA-256



SHA-256, described in Section 2 of this paper, is a 256-bit hash and is meant to provide 128 bits of security against collision attacks.

Message Authentication

- message authentication is concerned with:
  - protecting the integrity of a message
  - (i.e. protection from modification)
  - validating identity of originator
  - non-repudiation of origin (dispute resolution)
- will consider the security requirements
- then three alternative functions used:
  - message encryption
  - message authentication code (MAC)
  - hash function

Generically this is the problem of message authentication, and in eCommerce applications is arguably more important than secrecy.

References:

Elements of Crypto-System

Symmetric Encryption	–DES, AES, …
Public Key Infrastructure	–RSA, Elliptic Curve, NTRU…
Key Exchange	–Diffie–Helmann
Hashing Function	–MD5, MD6, SHA1, SHA2, SHA3

Security Requirements

- disclosure
- traffic analysis
- masquerade
- content modification
- sequence modification
- timing modification
- source repudiation
- destination repudiation

Message Encryption

- message encryption by itself also provides a measure of authentication
- if symmetric encryption is used then:
  - receiver know sender must have created it
  - since only sender and receiver now key used
  - know content cannot of been altered
  - if message has suitable structure, redundancy or a checksum to detect any changes

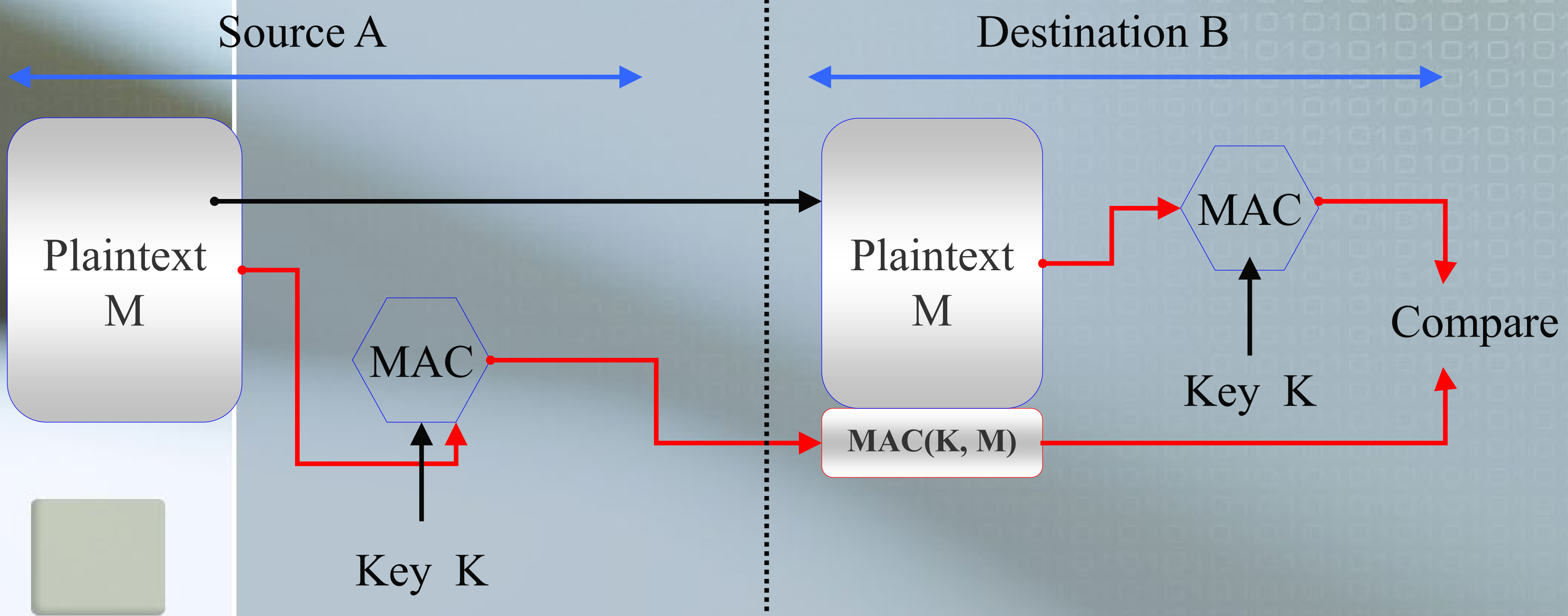
Message Encryption

- if public-key encryption is used:
  - encryption provides no confidence of sender
  - since anyone potentially knows public-key
  - however if
    - sender **signs** message using his private-key
    - then encrypts with recipients public key
    - have both secrecy and authentication
  - again need to recognize corrupted messages
  - but at cost of two public-key uses on message

Message Authentication Code (MAC)

- generated by an algorithm that creates a small fixed-sized block
  - depending on both message and some key
  - like encryption though need not be reversible
- appended to message as a **signature**
- receiver performs same computation on message and checks it matches the MAC
- provides assurance that message is unaltered and comes from sender

Message Authentication Code



Message Authentication Codes

- as shown the MAC provides authentication
- can also use encryption for secrecy
  - generally use separate keys for each
  - can compute MAC either before or after encryption
  - is generally regarded as better done before
- why use a MAC?
  - sometimes only authentication is needed
  - sometimes need authentication to persist longer than the encryption (e.g. archival use)
- note that a MAC is not a digital signature

MAC Properties

- a MAC is a cryptographic checksum
  - $MAC = C_K(M)$
  - condenses a variable-length message M
  - using a secret key K
  - to a fixed-sized authenticator
- is a many-to-one function
  - potentially many messages have same MAC
  - but finding these needs to be very difficult

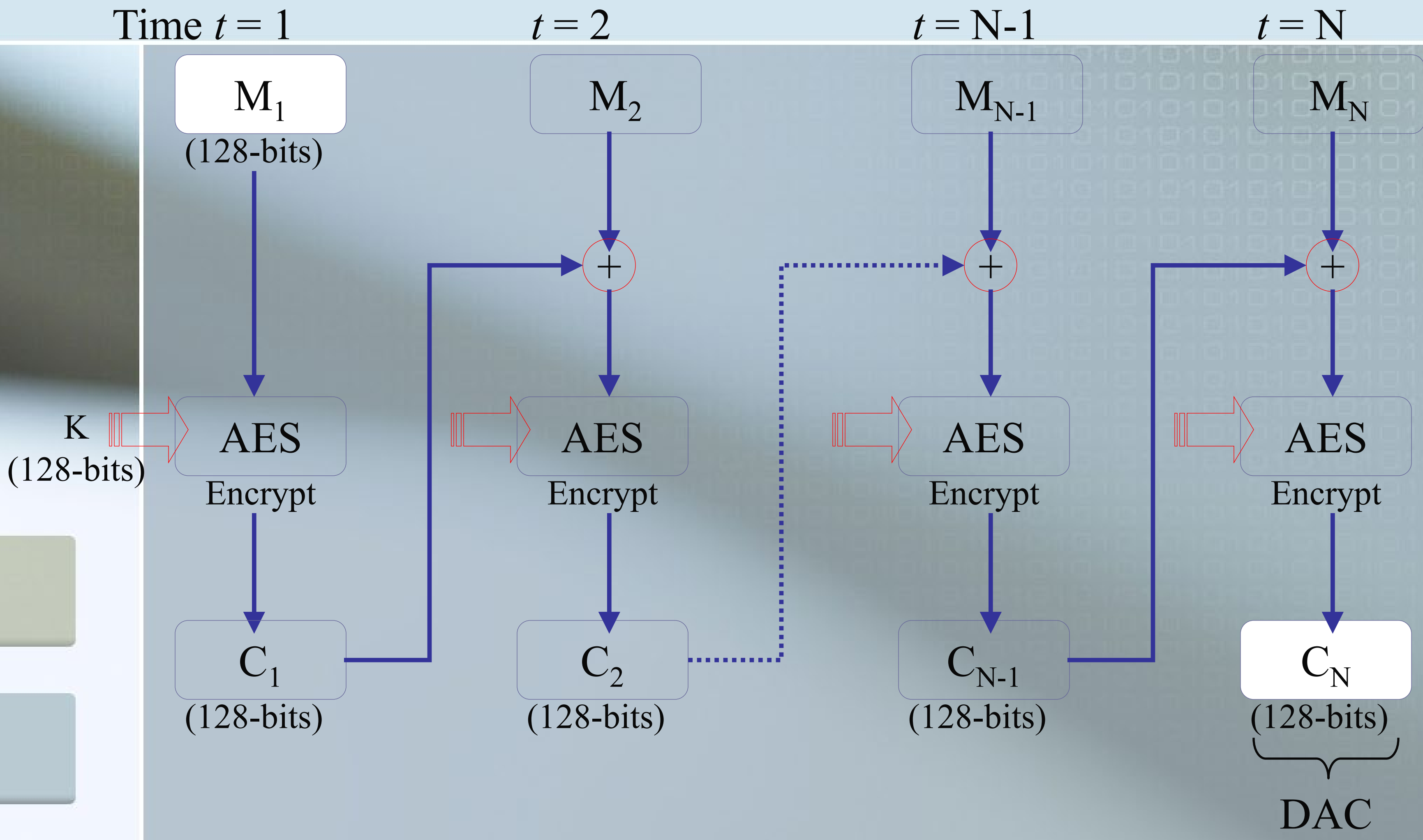
Requirements for MACs

- taking into account the types of attacks
- need the MAC to satisfy the following:
  - knowing a message and MAC, is infeasible to find another message with same MAC
  - MACs should be uniformly distributed
  - MAC should depend equally on all bits of the message

Using Symmetric Ciphers for MACs

- can use any block cipher chaining mode and use final block as a MAC
- **Data Authentication Algorithm (DAA)** is a widely used MAC based on AES-CBC
  - using Initial Vector IV=0 and zero-pad of final block
  - encrypt a message using AES in Cipher Block Chain (CBC) mode
  - and send just the final block as the MAC
    - or the leftmost M bits ($16 \le M \le 64$) of final block
- but final MAC is now too small for security

Data Authentication Algorithm



FIPS PUB 113 / ANSI X9.17 MAC based on AES-CBC

Hash Functions

- condenses arbitrary message to fixed size
  - $h = \mathrm{H}(\mathrm{M})$
- usually assume that the hash function is public and not keyed
  - unlike MAC which is keyed
- hash used to detect changes to message
- It is a one-way function
- can use in various ways with message
- most often to create a digital signature

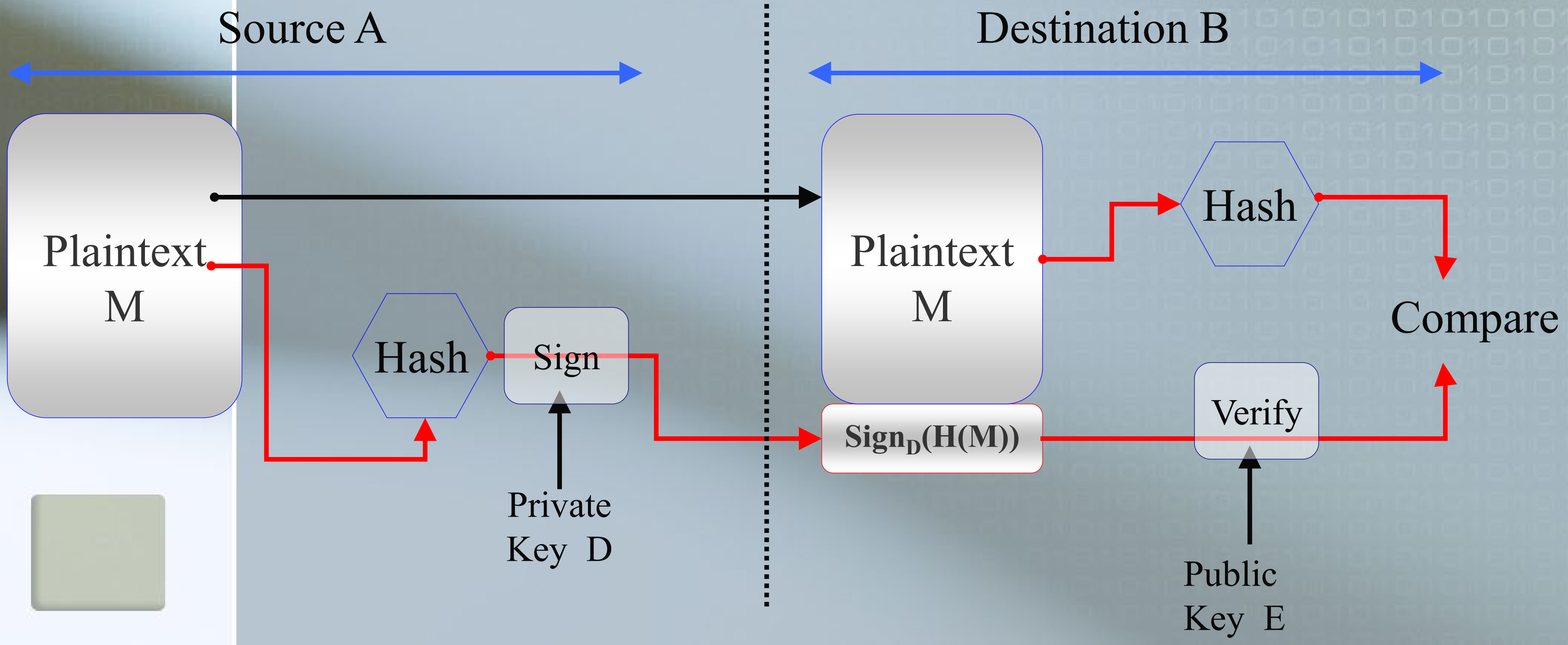
Hash Function

- A hash function plays an important role in cryptography.
- First, it is a form of encryption that takes some plaintext input and transforms it into a fixed-length encrypted output called the *message digest*.
- The output digest is a fixed-size set of bits that serves as a unique "DNA" for the original message.
- If the original message is altered and then hashed again, it will produce a different signature.
- Thus, hash functions can be used to detect altered and forged documents.
- Hashing provides message integrity, assuring recipients that the contents of a message have not been altered or corrupted.
- Second, Hash functions are *one-way*, meaning that it is easy to compute the message digest but very difficult to revert the message digest back to the original plaintext.

Hash function features

- A hash function should be improbable for two different messages to ever produce the same message digest. Changing a single bit in one message will produce an entirely different message digest.
- It should be improbable to produce a message that has some desired or predefined output (target message digest).
- It should be improbable to reverse the results of a hash function. This is possible because a message digest could have been produced by an almost infinite number of messages.
- The hash algorithm itself does not need to be kept secret. It is made available to the public. Its security comes from its ability to produce one-way hashes.
- The resulting message digest is a fixed size. A hash of a short message will produce the same size digest as a hash of a full set of messages.

Hash Functions & Digital Signatures



Basic Uses of Hash Functions : It shows the hash being “signed” with the senders private key, thus forming a digital signature.

Requirements for Hash Functions

- can be applied to any sized message M
- produces fixed-length output h
- is easy and efficient to compute h = H(M) for any message M
- given h is infeasible… difficult to find x s.t. H(x) = h
  - one-way property
- given x is infeasible to find y s.t. H(y)=H(x)
  - weak collision resistance
- is infeasible to find any x, y s.t. H(y)=H(x)
  - strong collision resistance

Simple Hash Functions

- are several proposals for simple functions
- based on XOR of message blocks
- not secure since can manipulate any message and either not change hash or change hash also
- need a stronger cryptographic function
- Another simple example is a summation
- Another area of study is error correcting code

Birthday Attacks

- might think a 64-bit hash is secure
- but by **Birthday Paradox** is not
- **birthday attack** works thus:
  - opponent generates $2^{m/2}$ variations of a valid message all with essentially the same meaning
  - opponent also generates $2^{m/2}$ variations of a desired fraudulent message
  - two sets of messages are compared to find a pair with same hash (probability > 0.5 by birthday paradox)
  - have user sign the valid message, then substitute the forgery which will have a valid signature
- conclusion is that need to use larger MAC/hash

Block Ciphers as Hash Functions

- can use block ciphers as hash functions
  - using $H_0 = 0$ and zero-pad of final block
  - compute: $H_{\boldsymbol{i}} = E_{M_{\boldsymbol{i}}} [H_{\boldsymbol{i}-1}]$
  - and use final block as the hash value
  - similar to CBC but without a key
- resulting hash is too small (64-bit)
  - both due to direct birthday attack
  - and to "meet-in-the-middle" attack
- other variants also susceptible to attack

Hash Functions & MAC Security

- like block ciphers have:
- **brute-force** attacks exploiting
  - strong collision resistance hash have cost $2^{m/2}$
    - 128-bit hash looks vulnerable, 160-bits better
  - MACs with known message-MAC pairs
    - can either attack key-space (exhaustive key search) or MAC
    - at least 128-bit MAC is needed for security

Hash Functions & MAC Security

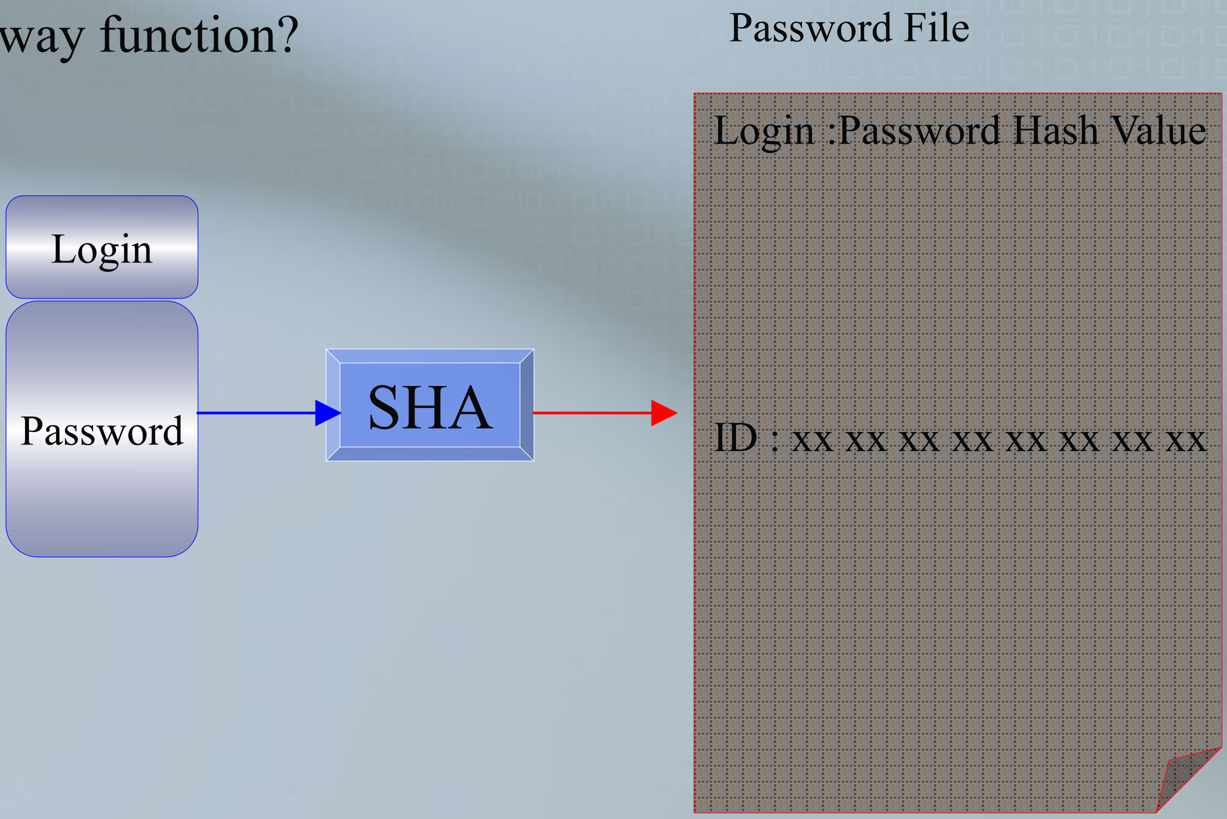
- **cryptanalytic attacks** exploit structure
  - like block ciphers want brute-force attacks to be the best alternative
- have a number of analytic attacks on iterated hash functions
  - Column Vector $CV_{\boldsymbol{i}} = f[CV_{\boldsymbol{i}-1}, M_{\boldsymbol{i}}]$; $H(M) = CV_N$
  - typically focus on collisions in function f
  - like block ciphers is often composed of rounds
  - attacks exploit properties of round functions

Application of Hash function

- Password protection mechanism
- Secure Session Key
  - Security also lies in keeping the key secret from the administrator
- Digital Signature
  - Digital Signature is an authentication mechanism that enables the owner/sender/creator of a message to attach a code that acts as a signature.
  - The signature guarantees the source and integrity of the message.
  - Hash function is utilized and integrated the signing of the order or document.
- Public Key Authentication
  - A certificate authority (CA) is an authority in a network that issues and manages security credentials and public keys for message encryption. As part of a public key infrastructure (PKI),
  - a CA checks with a registration authority (RA) to verify information provided by the requestor of a digital certificate .
  - If the RA verifies the requestors information, the CA can then issue a certificate.
  - The PKI Certificate has to come along with the hash digest value of the ownership plus the public key for verification by other parties before using the public key.
  - This is to make sure the public key is valid and belong to the right ownership.
- Message Digest of a forensic file…i.e. Evidence
  - Integrity protection
  - To show that a file is in tact ~ unchanged!

Proper Password Protection

- Non-Reversible by Administrator
- Ethical
- Secure Hashing Algorithm
- Modern
- One-way function?



Where is the password?

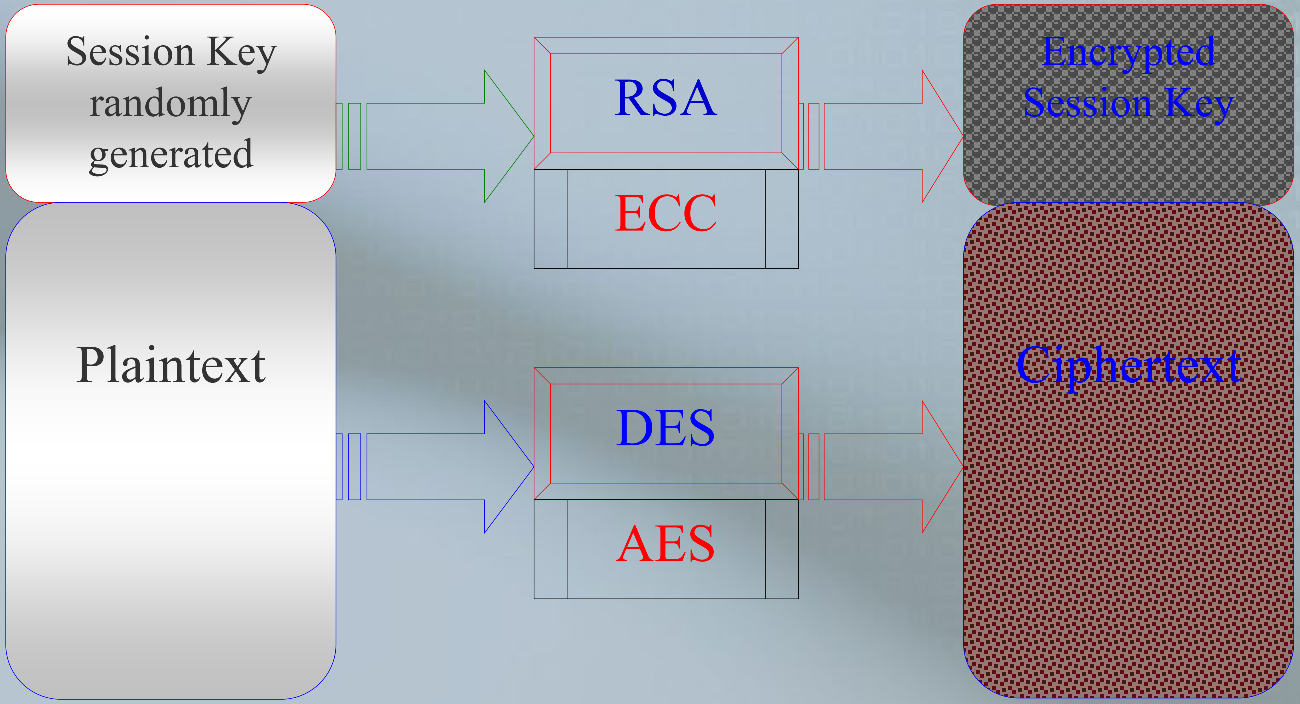
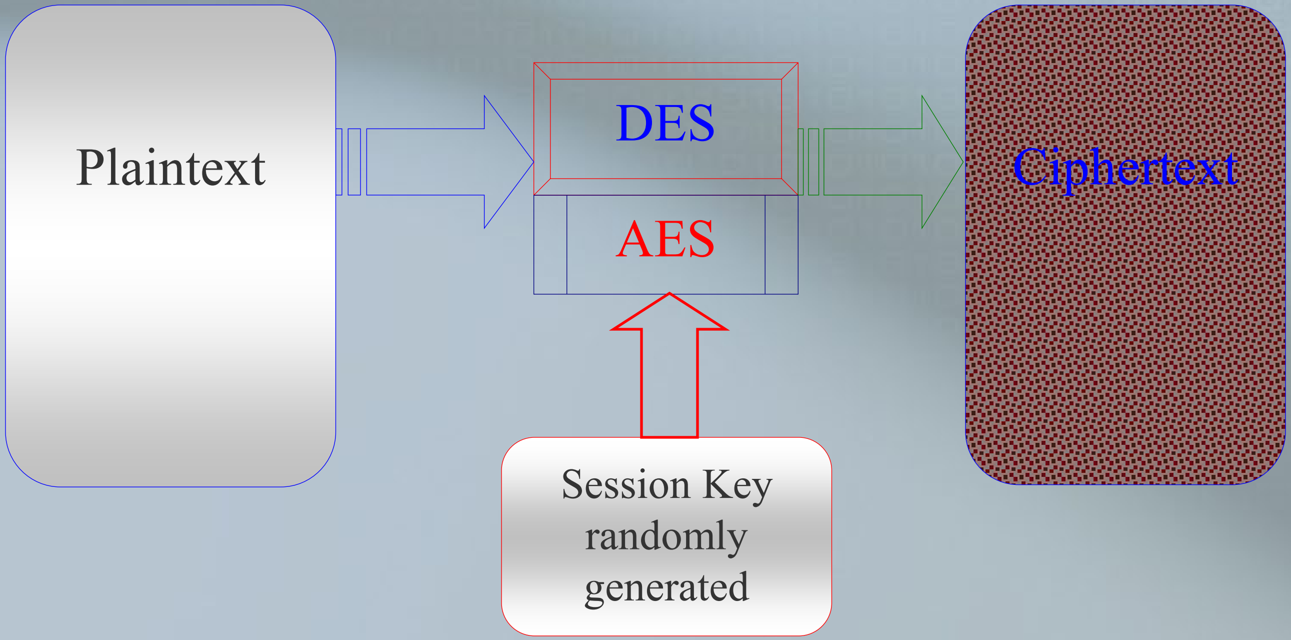


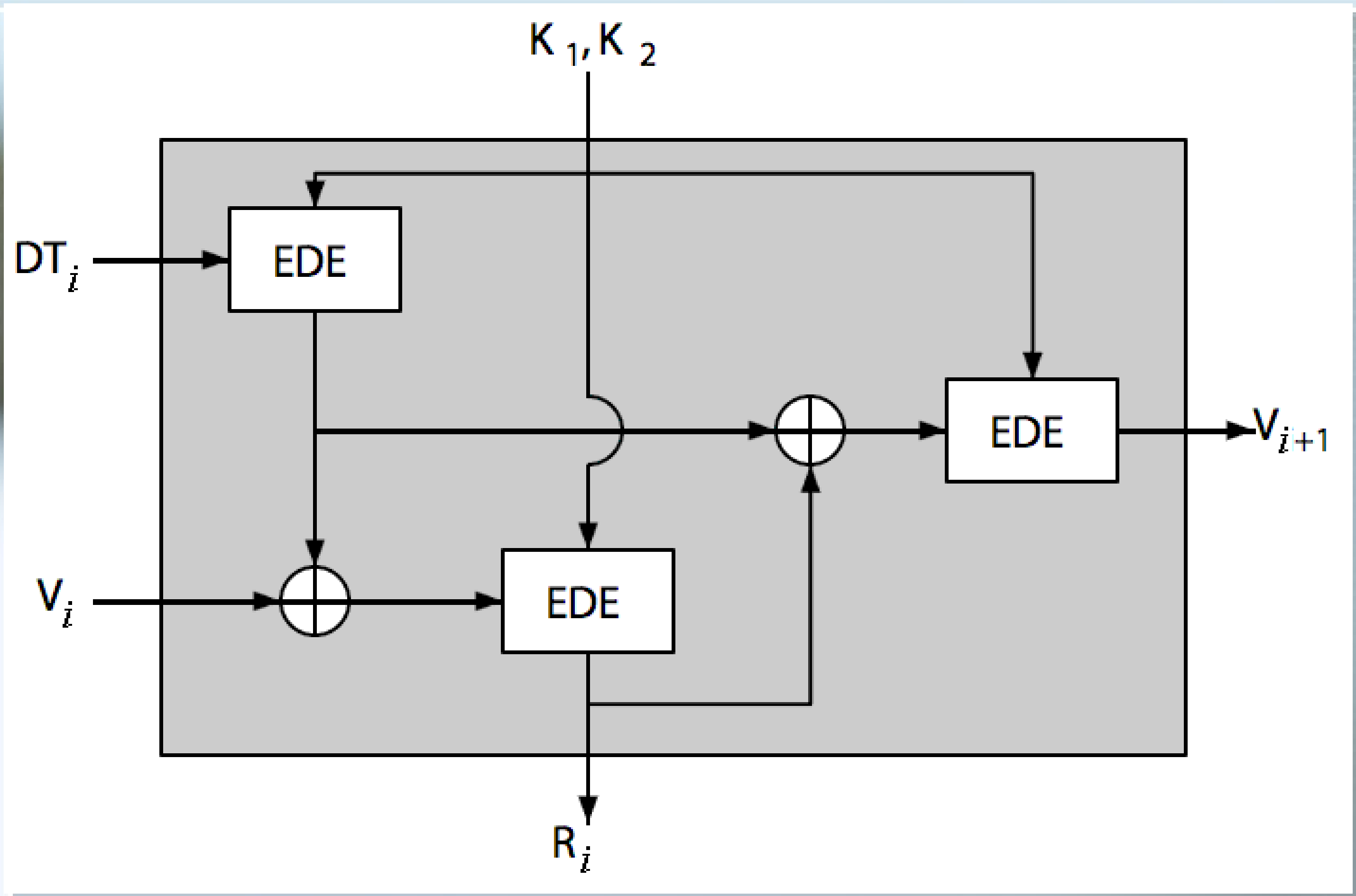
Figure 1 : Common Strategy of Encryption Mode combining Public Key Infrastructure and Standard Symmetric Encryption
Session key coming out of PRNG must be hashed!

Session Key

- Temporary Key
- Used For Encryption Of Data Between Users
- For One Logical Session Then Discarded
- Use Once Only
- Commonly generated by PRNG
- Session key coming out of PRNG must be hashed!



ANSI X9.17 PRNG



ANSI X9.17 PRG

- One of the strongest (cryptographically speaking) PRNGs is specified in ANSI X9.17.
- It uses date/time & seed inputs and 3 triple-DES encryptions to generate a new seed & random value.
  - DT_i - Date/time value at the beginning of i^{th} generation stage
  - V_i - Seed value at the beginning of i^{th} generation stage
  - R_i - Pseudorandom number produced by the i^{th} generation stage
  - K_1, K_2 - DES keys used for each stage

- Then compute successive values as:
  - $R_i = EDE([K_1, K_2], [V_i \text{ XOR } EDE([K_1, K_2], DT_i)])$
  - $V_{i+1} = EDE([K_1, K_2], [R_i \text{ XOR } EDE([K_1, K_2], DT_i)])$

- Several factors contribute to the cryptographic strength of this method. The technique involves a 112-bit key and three EDE encryptions for a total of nine DES encryptions.
- The scheme is driven by two pseudorandom inputs, the date and time value, and a seed produced by the generator that is distinct from the pseudo-random number produced by the generator.
- Thus the amount of material that must be compromised by an opponent is overwhelming.

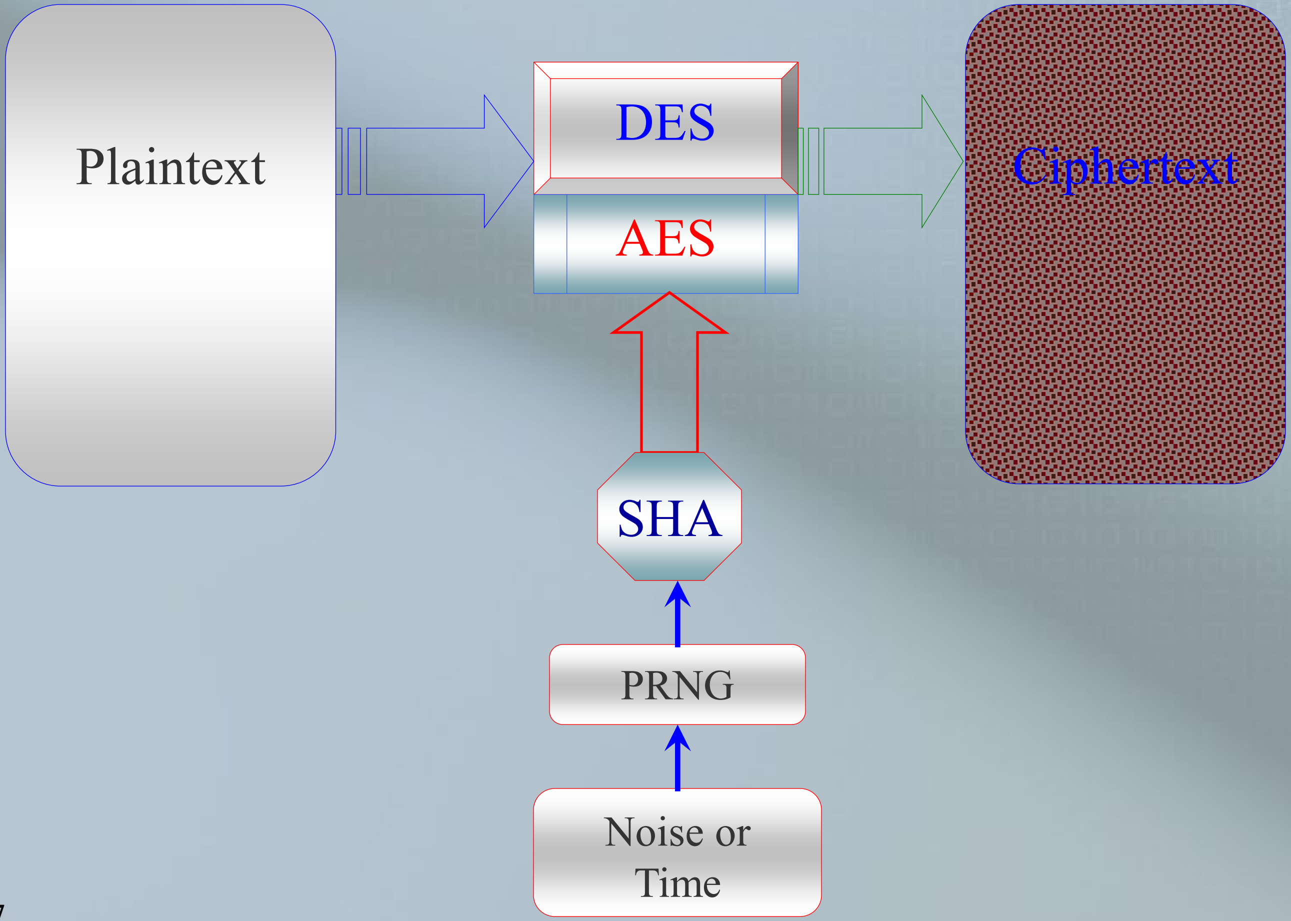
Natural Random Noise

- best source is natural randomness in real world
- find a regular but random event and monitor
- do generally need special h/w to do this
  - eg. radiation counters,
  - radio noise,
  - audio noise,
  - thermal noise in diodes,
  - leaky capacitors,
  - mercury discharge tubes etc
- starting to see such h/w in new CPU's
- problems of **bias** or uneven distribution in signal
  - have to compensate for this when sample and use
  - best to only use a few noisiest bits from each sample

A true random number generator (TRNG) uses a nondeterministic source to produce randomness. Most operate by measuring unpredictable natural processes, such as pulse detectors of ionizing radiation events, gas discharge tubes, and leaky capacitors. Special hardware is usually needed to capture the readout. A true random number generator may produce an output that is biased in some way. Various methods of modifying a bit stream to reduce or eliminate the bias have been developed.

Session Key

- Commonly generated by PRNG
- Session key coming out of PRNG must be hashed!



Digital Signature

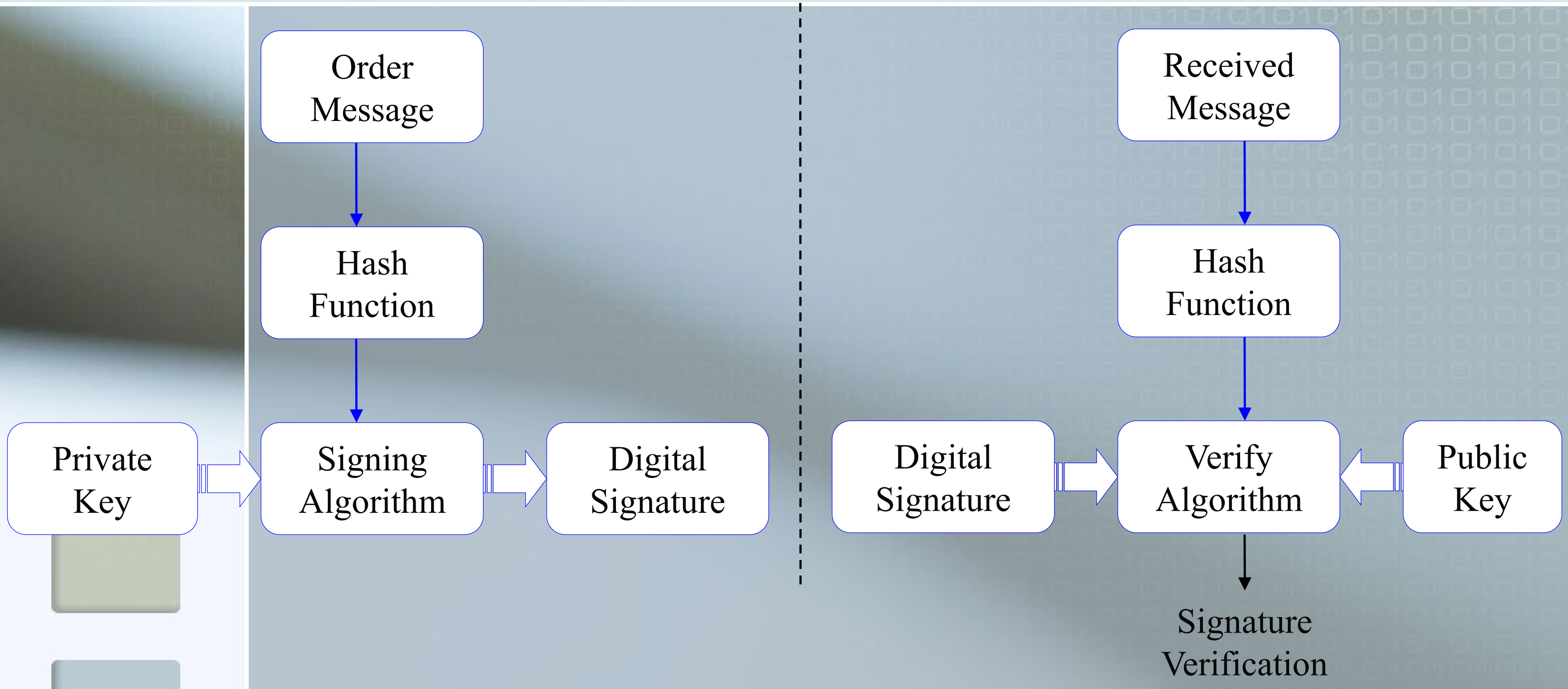
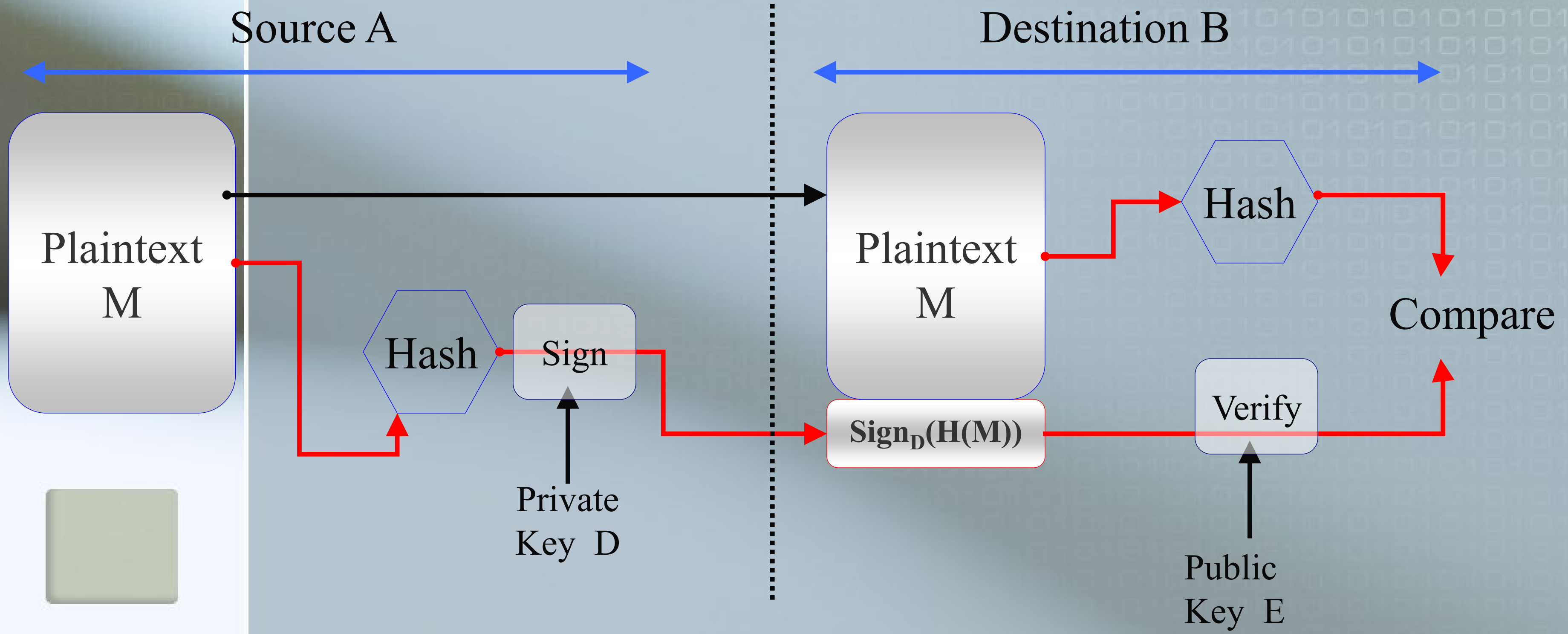


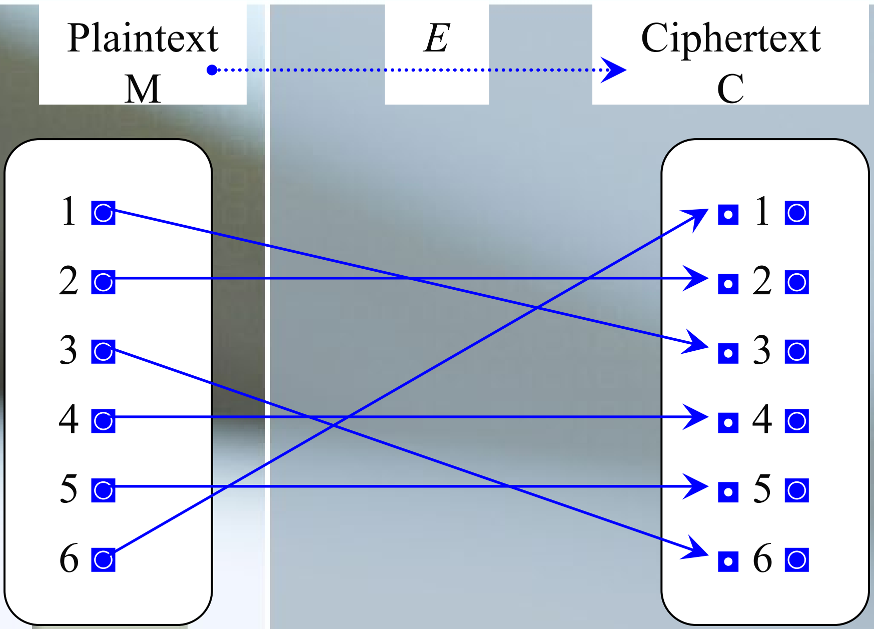
Figure 2. Basic digital signature verification process.

Hash Functions & Digital Signatures

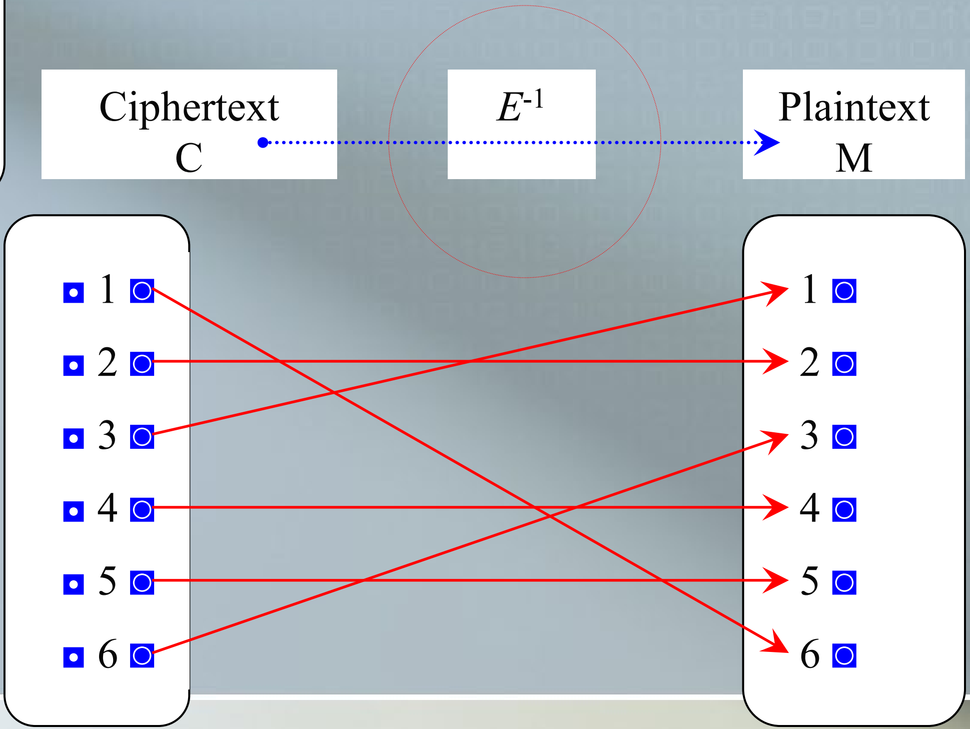


Basic Uses of Hash Functions : It shows the hash being “signed” with the senders private key, thus forming a digital signature.

example : Let the plaintext set be M = {1, 2, …, 6} and ciphertext set be C= {1, 2, …, 6}. Take a sample function $E(m) = 3^m \bmod 7$.



One-way function means the inverse is difficult to compute!

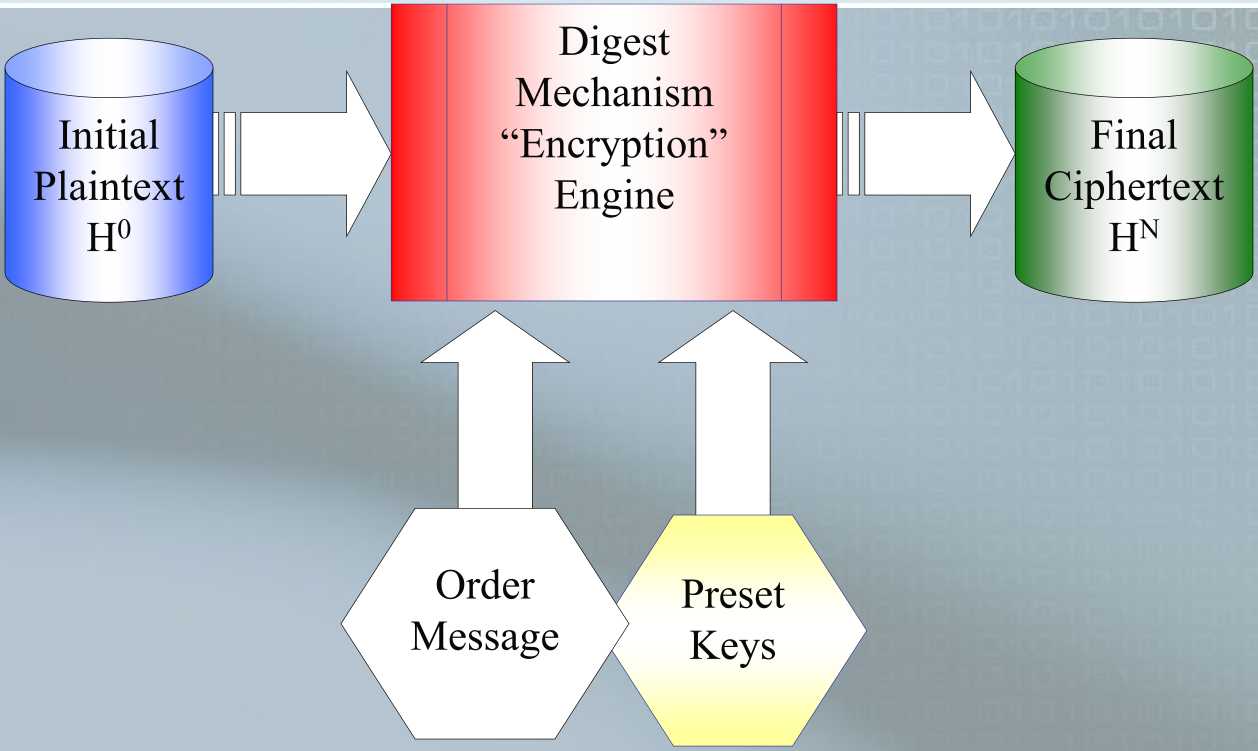


Inverse Function
The inverse of 1-1 function take the reverse process from ciphertext range set to the domain plaintext set.

Secure Hashing Algorithm SHA-2

- An *n*-bit hash is a map from arbitrary length messages to n-bit hash values.
- An *n*-bit cryptographic hash is an n-bit hash which is one-way and collision-resistant function.
- One way function means that given a hash value, it should require work equivalent to about 2^n hash computations in order to find any message that hashes to that value.
- Collision-resistant means that finding any two messages which hash to the same value should require work equivalent to about $2^{n/2}$ hash computations.
- Since the goal of the new Advanced Encryption Standard (AES) is to offer, at its three crypto-variable sizes, 128, 192, and 256 bits of security, there is a need for companion hash algorithms which provide similar levels of enhanced security.

SHA-2 means SHA-256



SHA-256, described in Section 2 of this paper, is a 256-bit hash and is meant to provide 128 bits of security against collision attacks.

SHA-256 Overview

- SHA-256 operates in the manner of MD4, MD5, and SHA-1: The message to be hashed is first
  - (1) padded with its length in such a way that the result is a multiple of 512 bits long, and then
  - (2) parsed into 512-bit message blocks $M^{(1)}$, $M^{(2)}$, ……, $M^{(N)}$.

- The message blocks are processed one at a time: Beginning with a fixed initial hash value $H^{(0)}$, sequentially compute
  - $H^{(i)} = H^{(i-1)} + C_{M^{(i)}}(H^{(i-1)})$,
  - where C is the SHA-256 compression function and + means word-wise mod 2^{32} addition.

  - $H^{(N)}$ is the hash value of input M.

Description of SHA-256

- The SHA-256 compression function operates on a 512-bit message block and a 256-bit intermediate hash value.
- It is essentially a 256-bit block cipher algorithm which encrypts the intermediate hash value using the message block as key.
- Hence there are two main components to describe:
  - the SHA-256 compression function, and
  - the SHA-256 message schedule.
- We will use the following notation:

Symbol	Operation
$\oplus$	Bitwise XOR
$\wedge$	Bitwise AND
$\vee$	Bitwise OR
$\neg$	Bitwise complement
$+$	Mod 2^{32} addition
R^n	Right shift by n bits
S^n	Right rotation by n bits

Table 1: Notation
All of these operators act on 32-bit words.

Prescribed Input Initial Hash Value $H^{(0)}$

The initial hash value $H^{(0)}$ is the following sequence of 32-bit words. They are obtained by taking the fractional parts of the square roots of the first eight primes):

$H_1^{(0)}$ = 6A09E667
$H_2^{(0)}$ = BB67AE85
$H_3^{(0)}$ = 3C6EF372
$H_4^{(0)}$ = A54FF53A
$H_5^{(0)}$ = 510E527F
$H_6^{(0)}$ = 9B05688C
$H_7^{(0)}$ = 1F83D9AB
$H_8^{(0)}$ = 5BE0CD19

$\sqrt{2}$	1.	414	213	562	373	095	048	801	688	10
	1.	0110	1010	0000	1001	1110	0110	0110	0111	2
	1.	**6**	**A**	**0**	**9**	**E**	**6**	**6**	**7**	16
$\sqrt{3}$	1.	732	050	807	568	877	293	527	446	10
	1.	1011	1011	0110	0111	1010	1110	1000	0101	2
	1.	**B**	**B**	**6**	**7**	**A**	**E**	**8**	**5**	16
$\sqrt{5}$	2.	236	067	977	499	789	696	409	173	10
	10.	0011	1100	0110	1110	1111	0011	0111	0010	2
	2.	**3**	**C**	**6**	**E**	**F**	**3**	**7**	**2**	16
$\sqrt{7}$	2.	645	751	311	064	590	590	501	615	10
	10.	1010	0101	0100	1111	1111	0101	0011	1010	2
	2.	**A**	**5**	**4**	**F**	**F**	**5**	**3**	**A**	16
$\sqrt{11}$	3.	316	624	790	355	399	849	114	932	10
	11.	0101	0001	0000	1110	0101	0010	0111	1111	2
	3.	**5**	**1**	**0**	**E**	**5**	**2**	**7**	**F**	16
$\sqrt{13}$	3.	605	551	275	463	989	293	119	221	10
	11.	1001	1011	0000	0101	0110	1000	1000	1100	2
	3.	**9**	**B**	**0**	**5**	**6**	**8**	**8**	**C**	16
$\sqrt{17}$	4.	123	105	625	617	660	549	821	409	10
	100.	0001	1111	1000	0011	1101	1001	1010	1011	2
	4.	**1**	**F**	**8**	**3**	**D**	**9**	**A**	**B**	16
$\sqrt{19}$	4.	358	898	943	540	673	552	236	981	10
	100.	0101	1011	1110	0000	1100	1101	0001	1001	2
	4.	**5**	**B**	**E**	**0**	**C**	**D**	**1**	**9**	16

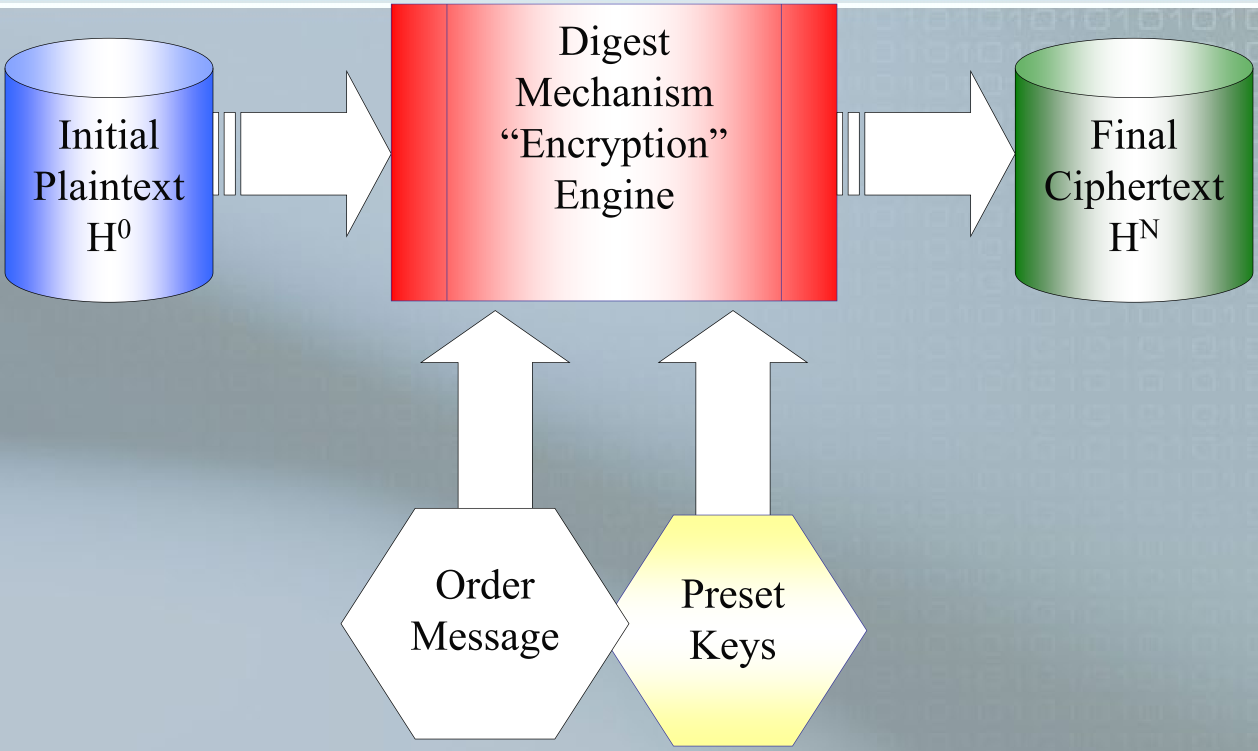
Initial Hash Value $H^{(0)}$

The initial hash value $H^{(0)}$ is the following sequence of 32-bit words. They are obtained by taking the fractional parts of the square roots of the first 8 primes):

$H_1^{(0)}$ = 6A09E667
$H_2^{(0)}$ = BB67AE85
$H_3^{(0)}$ = 3C6EF372
$H_4^{(0)}$ = A54FF53A
$H_5^{(0)}$ = 510E527F
$H_6^{(0)}$ = 9B05688C
$H_7^{(0)}$ = 1F83D9AB
$H_8^{(0)}$ = 5BE0CD19

Please refer the SHA2version2 document!

SHA-2 means SHA-256



SHA-256, described in Section 2 of this paper, is a 256-bit hash and is meant to provide 128 bits of security against collision attacks.

A sequence of constant words,
$K_0, K_1, K_2, K_3, \ldots\ldots, K_7$
$K_8, \ldots\ldots\ldots\ldots\ldots\ldots, K_{15}$
$K_{16}, \ldots\ldots\ldots\ldots\ldots\ldots, K_{23}$
$K_{24}, \ldots\ldots\ldots\ldots\ldots\ldots, K_{31}$
$K_{32}, \ldots\ldots\ldots\ldots\ldots\ldots, K_{39}$
$K_{40}, \ldots\ldots\ldots\ldots\ldots\ldots, K_{47}$
$K_{48}, \ldots\ldots\ldots\ldots\ldots\ldots, K_{55}$
$K_{56}, K_{57}, K_{58}, \ldots\ldots, K_{63}$
is being used in SHA-256. In hex, these are given by

428A2F98	71374491	B5C0FBCF	E9B5DBA5	3956C25B	59F111F1	923F82A4	AB1C5ED5
D807AA98	12835B01	243185BE	550C7DC3	72BE5D74	80DEB1FE	9BDC06A7	C19BF174
E49B69C1	EFBE4786	0FC19DC6	240CA1CC	2DE92C6F	4A7484AA	5CB0A9DC	76F988DA
983E5152	A831C66D	B00327C8	BF597FC7	C6E00BF3	D5A79147	06CA6351	14292967
27B70A85	2E1B2138	4D2C6DFC	53380D13	650A7354	766A0ABB	81C2C92E	92722C85
A2BFE8A1	A81A664B	C24B8B70	C76C51A3	D192E819	D6990624	F40E3585	106AA070
19A4C116	1E376C08	2748774C	34B0BCB5	391C0CB3	4ED8AA4A	5B9CCA4F	682E6FF3
748F82EE	78A5636F	84C87814	8CC70208	90BEFFFA	A4506CEB	BEF9A3F7	C67178F2

These are the first 32 bits of the fractional parts of the cube roots of the first 64 primes.

Diagrams

The SHA-256 compression function is pictured below:

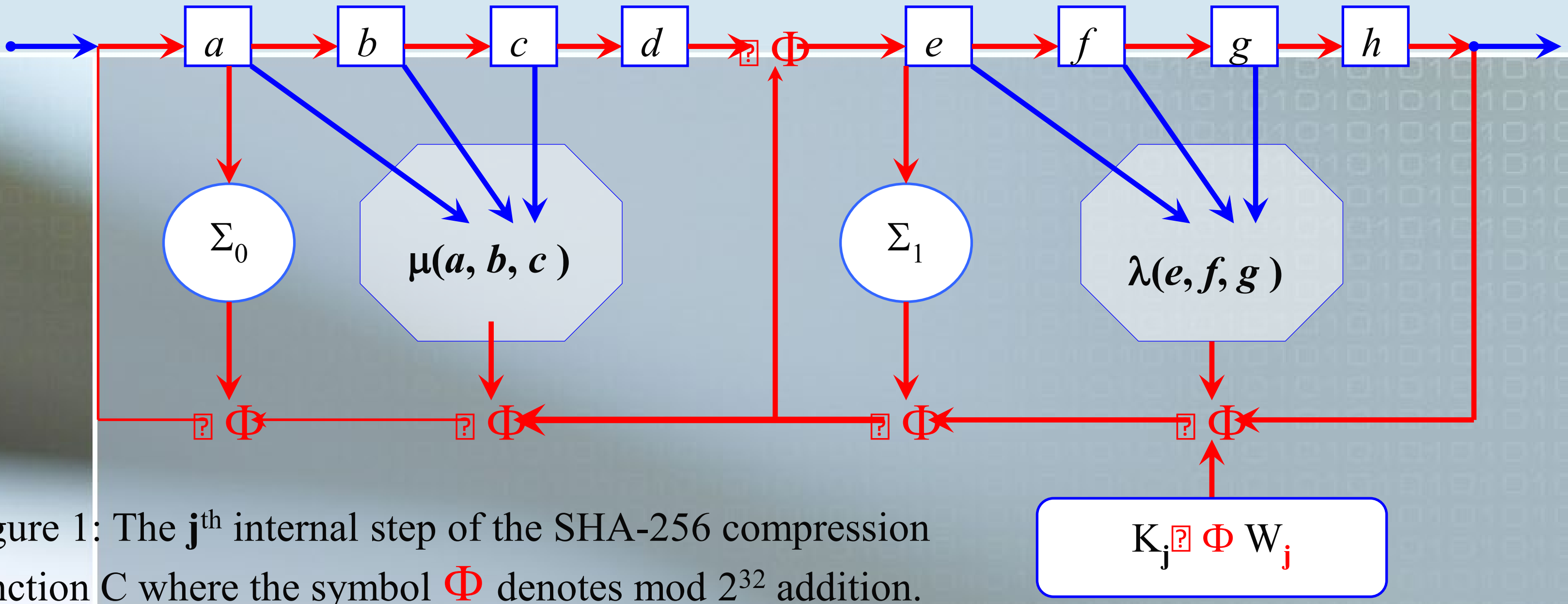


Figure 1: The j^th internal step of the SHA-256 compression function C where the symbol Φ denotes mod 2^{32} addition.

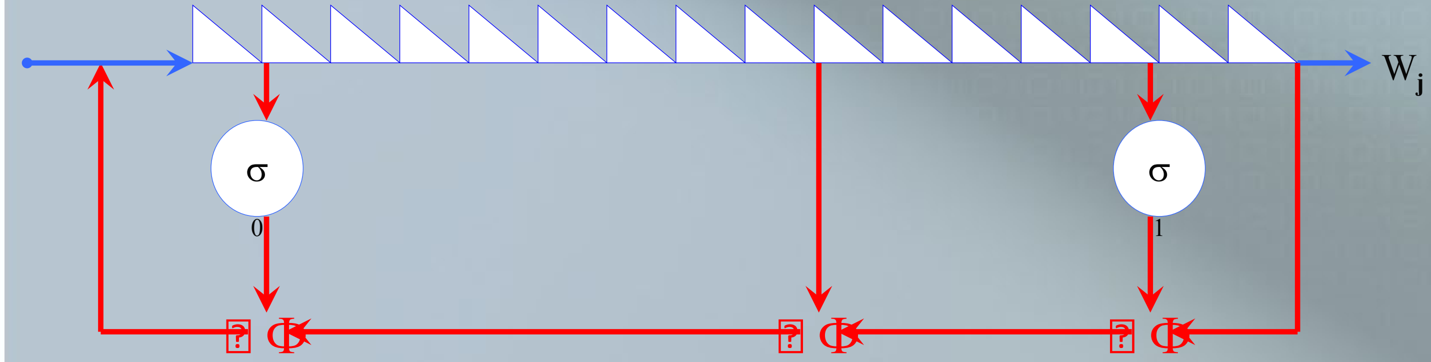


Figure 2: SHA-256 message schedule

The registers here are loaded with W_0, W_1, W_2, W_3, …. , W_{15}

Secure Hashing Algorithm

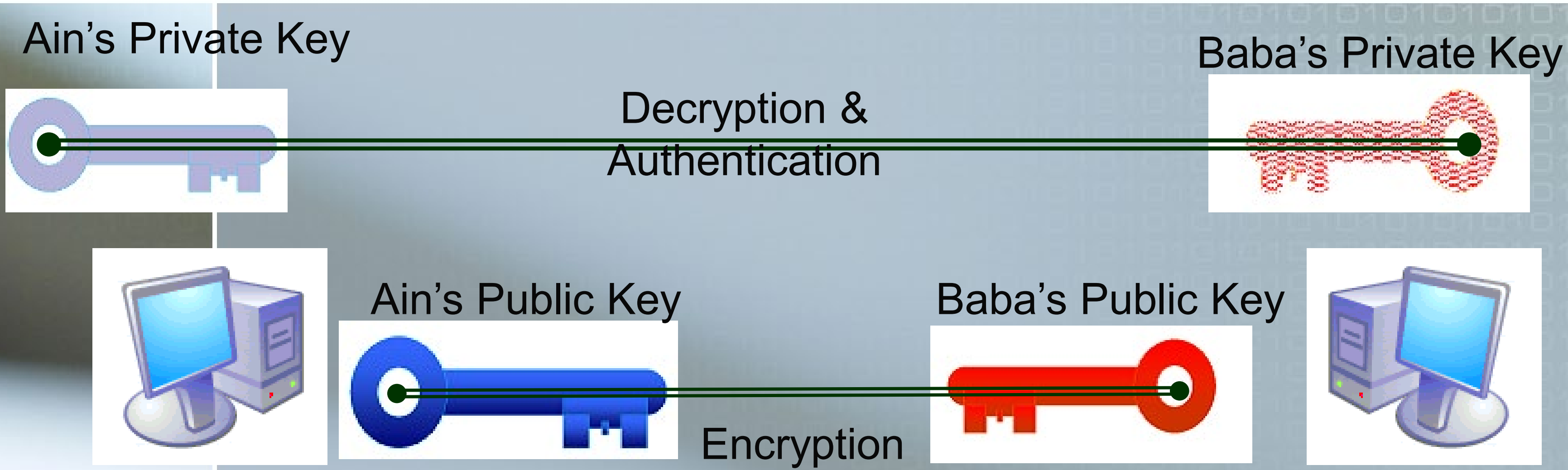
- There are 2 prominent algorithms in Hashing functions.
  - First, the most popularly used technique is MD5.
  - Second, the well accepted standard is secure hashing algorithm SHA-1.

- Nevertheless, SHA-256 is chosen in this class as it is considered to be the primary next-generation algorithm.

- **MD-5**
  - A hash function designed by Ron Rivest, one of the inventors of the RSA public-key encryption scheme.
  - The MD-5 algorithm produces a 128-bit output. Note that MD-5 is now known to have some weaknesses and should be avoided if possible.
  - SHA-1 is generally recommended.

- **SHA-1 (Secure Hash Algorithm-1)**
  - SHA-1 is an MD-5-like algorithm that was designed to be used with the Digital Signature Standard (DSS).
  - NIST (National Institute of Standards and Technology) and NSA (National Security Agency) are responsible for SHA-1.
  - The SHA-1 algorithm produces a 160-bit MAC.
  - This longer output is considered to be more secure than MD-5.

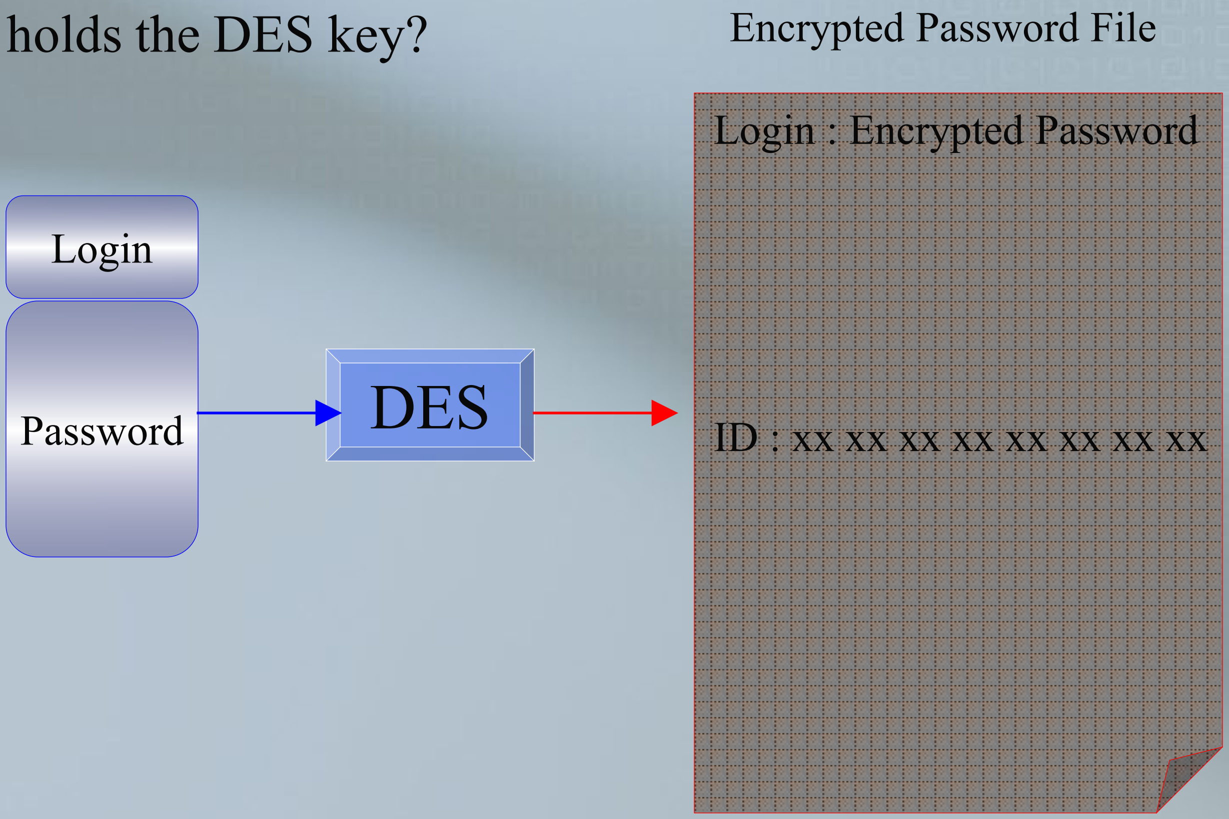
Key Elements in Modern CryptoSystem



- PRNG and TRNG
- Symmetric Encryption :AES
- SHA :SHA2
- PKI :RSA, ECC, NTRU
- Key Exchange
- Threshold Scheme? :Newton Interpolation

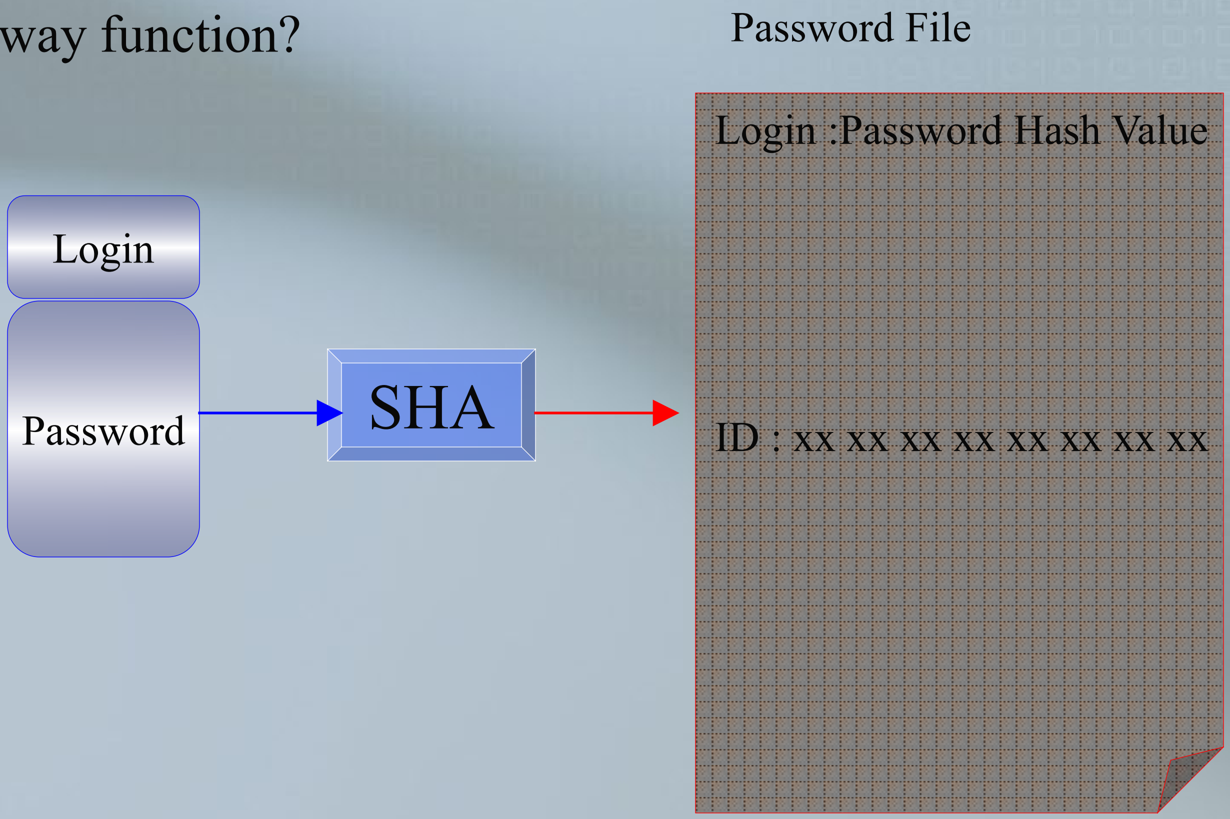
Traditional Password Protection

- Reversible by Administrator
- Unethical
- DES
- Traditional
- Who holds the DES key?



Proper Password Protection

- Non-Reversible by Administrator
- Ethical
- Secure Hashing Algorithm
- Modern
- One-way function?



Conclusion

Thus, hash functions, though not encryption algorithms in their own right, provide significant security services, mainly identity authentication. They form an important basis for all modern cryptosystems.

References:

1. **William Stallings, Cryptography and Network Security : Principles and Practice, 3rd Edition, Prentice Hall, 2003.**

2. Charles P. Pfleeger, Security in Computing, Prentice Hall, 3rd Edition December 2002.

3. **A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1996.**

Asymmetric

Symmetric

OTK