

INFO 6001
Revision 2
Information Security

Lesson 3

Lesson 3 Agenda

- Housekeeping
- Course Status Update
- User Authentication

Housekeeping

- Test #1 is in two weeks (worth 15%)
- This course number and the Section ID 1 or 1 PT or 1OL must be on your email subject line to get attention,!
- Good activity on the discussion forum, but many students are still MIA. The Introduction at start of the DF has been re-opened. The marks for Week 1 have been assigned but you must do your Introduction if you want any further weekly points.
- Remember, you must post, answer and read at least 2 others to get all 4 points.

Housekeeping

- Discussion Forum minimum activity – weekly marks (out of 4):
 - 1 read or 1 answer – 1 point each
 - 1 post – 2 points
 - 1 post and 1 answer – 3 points
 - 1 post, and 2 reads – 3 points
 - 1 post, 2 reads and 1 answer (or more) – 4 points
- Test #1 is in two weeks (worth 15%)
- Good activity on the discussion forum, but many students are still MIA. The Introduction at start of the DF has been re-opened. The marks for Week 1 have been assigned but you must do your Introduction if you want any further weekly points.
- Remember, you must post, answer and read at least 2 others to get all 4 points.

Course Tracking Status

- INFO-6001 on FOL:

- We have had 2 **Lesson** sessions & 2 **Lab** sessions.
- There have been 2 Lab **Assignments**. Each are worth **3.75%** for a total of **7.5%**.
- We have had 0 **Test(s)** worth **0%**
- This represents **7.5%** of the overall course marks for the term.
- You will be working on Lab Assignment 3 during this week's Lab Session. It will be due 1 hour after the end of the Session.

- Upcoming:

- Week 5 - Test 1 (15%)

Chapter 3

User Authentication (Public Key Cryptography and Message Authentication)



This weeks learning outcomes

- Define the term *message authentication code*
- Explain the **requirements** for a message authentication code
- Explain a **secure hash function**
- Understand the operation of **SHA-512**
- Define and explore **HMAC**
- Discuss the basic principles of **public-key cryptosystems**
- Discuss the **RSA algorithm**
- Define **Diffie-Hellman** key exchange
- Understand a **man-in-the-middle** attack



Guiding Questions for this week...

1. How are encryption and message authentication similar, and how are they different?
2. Why are digital signatures being trusted as authentic – even for large purchases like houses and cars?
3. When would you use a one-way hash function?
4. What are the three ways a message can be authenticated?
5. How does the length of a hash value make it more secure?
6. Describe a scenario where you would recommend using a hashed MAC?
7. What are the 6 ingredients of a PKE scheme?
8. What are some issues with the Diffie-Hellman key exchange?



Message Authentication

- **Encryption** protects against **passive attack** (eavesdropping)
- A different requirement is to protect against **active attack** (falsification of data and transactions)
 - Protection against such attacks is known as **message authentication**
- Message authentication is a procedure that allows communicating parties to verify that received messages are authentic
 - The two important aspects are to verify:
 - That the **contents** of the message **have not been altered**
 - That the **source is authentic**

Approaches to Message Authentication

Using **conventional encryption**

- Symmetric encryption alone is not a suitable tool for data authentication
 - We assume that only the sender and receiver share a key, so only the genuine sender would be able to encrypt a message successfully
 - The receiver assumes that no alterations have been made and that sequencing is proper ***if*** the message includes an ***error detection code*** and a ***sequence number***
 - If the message includes a ***timestamp***, the receiver assumes that the message has not been delayed beyond that normally expected for network transit

Without message encryption

- An **authentication tag** is generated and appended to each message for transmission
- The message itself is not encrypted and can be read at the destination independent of the authentication function at the destination
- Because the message is not encrypted, message ***confidentiality*** is not provided

Message Authentication Code

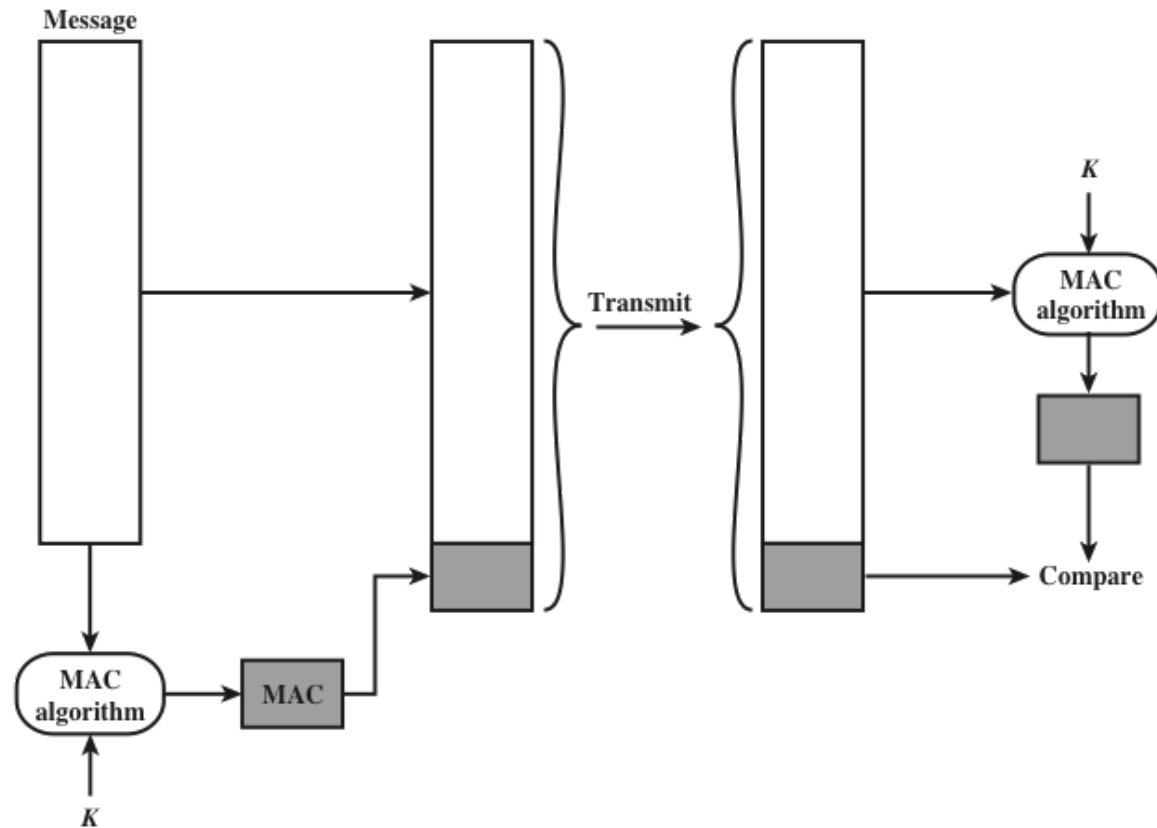


Figure 3.1 Message Authentication Using a Message Authentication Code (MAC)



One-way Hash Functions

- Accepts a variable-size message M as input and produces a fixed-size message digest $H(M)$ as output
- Does not take a secret key as input
- To authenticate a message, the message digest is sent with the message in such a way that the message digest is authentic

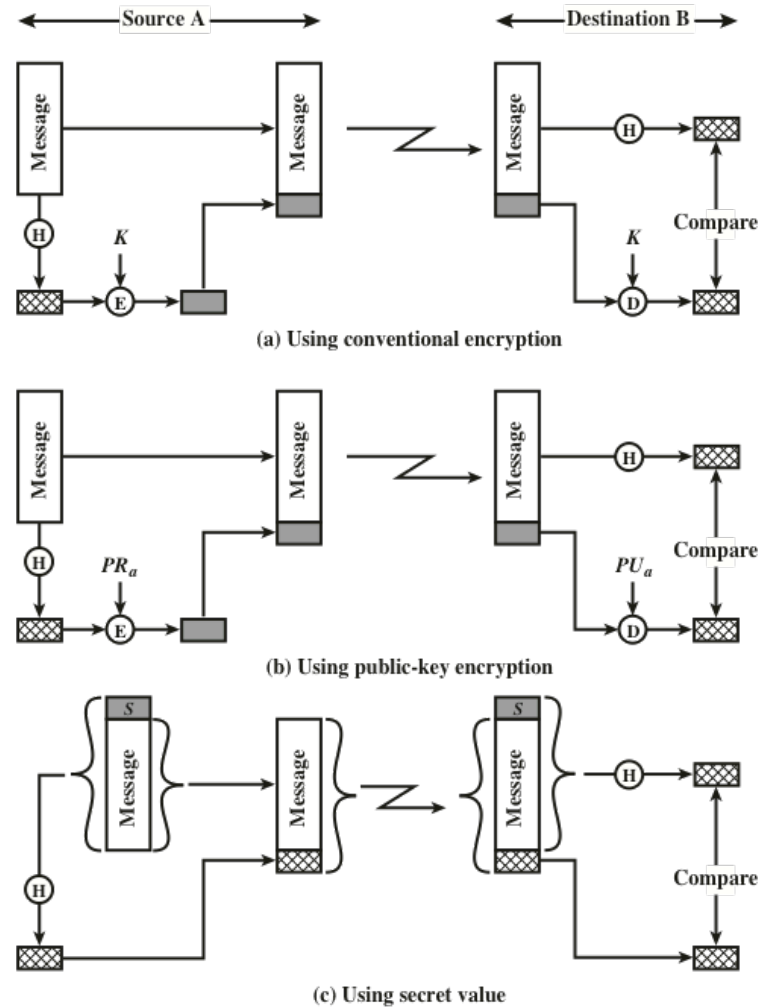


Figure 3.2 Message Authentication Using a One-Way Hash Function



Secure Hash Functions

Secure Hash Functions

- Is important not only in message authentication but in digital signatures
- Purpose is to produce a “fingerprint” of a file, message, or other block of data
- To be useful for message authentication, a hash function H must have the following properties:

1.

- H can be applied to a block of data of any size.

2.

- H produces a fixed-length output.

3.

- $H(x)$ is relatively easy to compute for any given x , making both hardware and software implementations practical.

4.

- For any given code h , it is computationally infeasible to find x such that $H(x) = h$. A hash function with this property is referred to as one-way or preimage resistant.

5.

- For any given block x , it is computationally infeasible to find $y \neq x$ with $H(y) = H(x)$. A hash function with this property is referred to as second preimage resistant. This is sometimes referred to as weak collision resistant.

6.

- It is computationally infeasible to find any pair (x, y) such that $H(x) = H(y)$.
- A hash function with this property is referred to as collision resistant. This is sometimes referred to as strong collision resistant.

Security of Hash Functions

- There are two approaches to attacking a secure hash function:
 - Cryptanalysis
 - Involves exploiting logical weaknesses in the algorithm
 - Brute-force attack
 - The **strength** of a hash function against this attack depends solely on the **length** of the hash code produced by the algorithm





	bit 1	bit 2	• • •	bit n
block 1	b_{11}	b_{21}		b_{n1}
block 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
block m	b_{1m}	b_{2m}		b_{nm}
hash code	C_1	C_2		C_n

Figure 3.3 Simple Hash Function Using Bitwise XOR



The SHA Secure Hash function

- SHA was developed by NIST and published as a federal information processing standard (FIPS 180) in 1993
- Was revised in 1995 as SHA-1 and published as FIPS 180-1
 - The actual standards document is entitled “Secure Hash Standard”
- Based on the hash function **MD4** and its design closely models MD4
- Produces 160-bit hash values
- In 2005 NIST announced the intention to phase out approval of SHA-1 and move to a reliance on SHA-2 by 2010



Table 3.1

Comparison of SHA Parameters

	SHA-1	SHA-224	SHA-256	SHA-384	SHA-512
Message Digest Size	160	224	256	384	512
Message Size	$< 2^{64}$	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block Size	512	512	512	1024	1024
Word Size	32	32	32	64	64
Number of Steps	80	64	64	80	80

Note: All sizes are measured in bits.

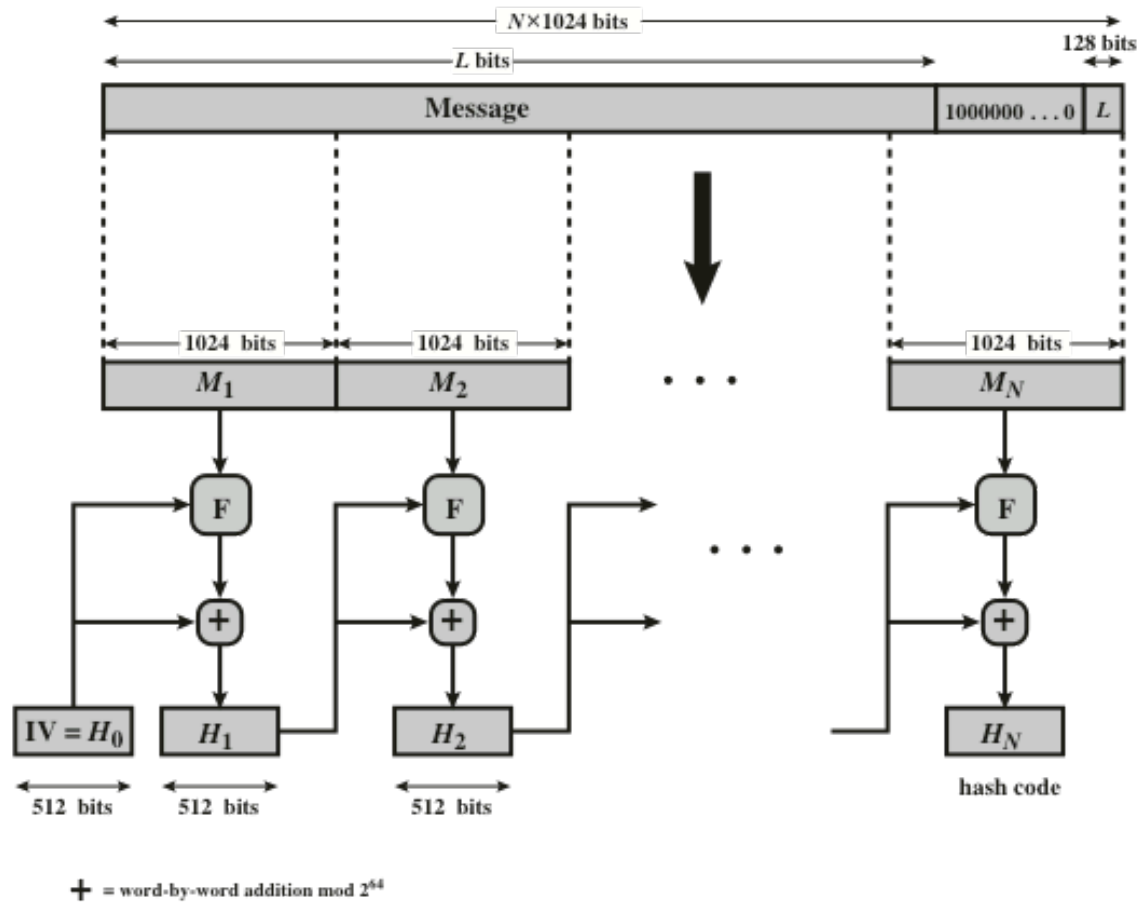


Figure 3.4 Message Digest Generation Using SHA-512

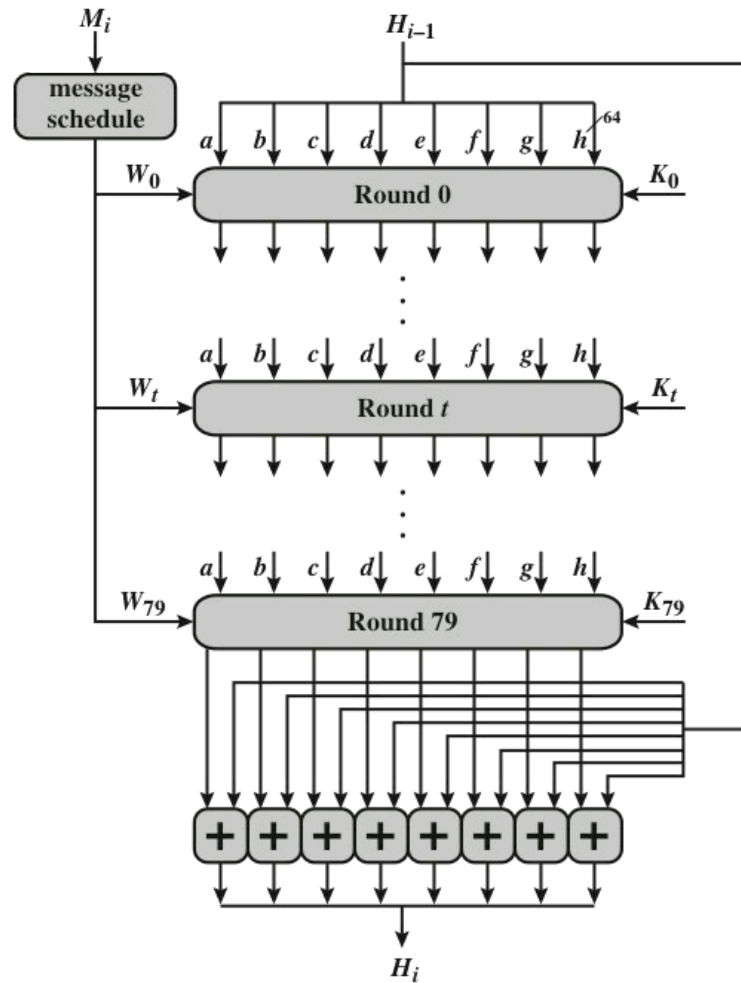


Figure 3.5 SHA-512 Processing of a Single 1024-Bit Block

SHA-3

1. It must be possible to replace SHA-2 with SHA-3 in any application by a simple drop-in substitution. Therefore, SHA-3 must support hash value lengths of 224, 256, 384, and 512 bits.

2. SHA-3 must preserve the online nature of SHA-2. That is, the algorithm must process comparatively small blocks (512 or 1024 bits) at a time instead of requiring that the entire message be buffered in memory before processing it.

Basic requirements that must be satisfied by any candidate for SHA-3

Message Authentication Codes

HMAC

- There has been an increased interest in developing a MAC derived from a cryptographic hash code, such as SHA-1
 - Cryptographic hash functions generally execute faster in software than conventional encryption algorithms such as DES
 - Library code for cryptographic hash functions is widely available
 - A hash function such as SHA-1 was not designed for use as a MAC and cannot be used directly for that purpose because it does not rely on a secret key
- There have been a number of proposals for the incorporation of a secret key into an existing hash algorithm
 - The approach that has received the most support is HMAC

HMAC

- Has been issued as RFC 2104
- Has been chosen as the mandatory-to-implement MAC for IP Security
- Is used in other Internet protocols, such as Transport Layer Security (TLS) and Secure Electronic Transaction (SET)



HMAC Design Objectives

- To use, without modifications, available hash functions --- in particular, hash functions that perform well in software, and for which code is freely and widely available
- To allow for easy replaceability of the embedded hash function in case faster or more secure hash functions are found or required
- To preserve the original performance of the hash function without incurring a significant degradation
- To use and handle keys in a simple way
- To have a well understood cryptographic analysis of the strength of the authentication mechanism based on reasonable assumptions on the embedded hash function

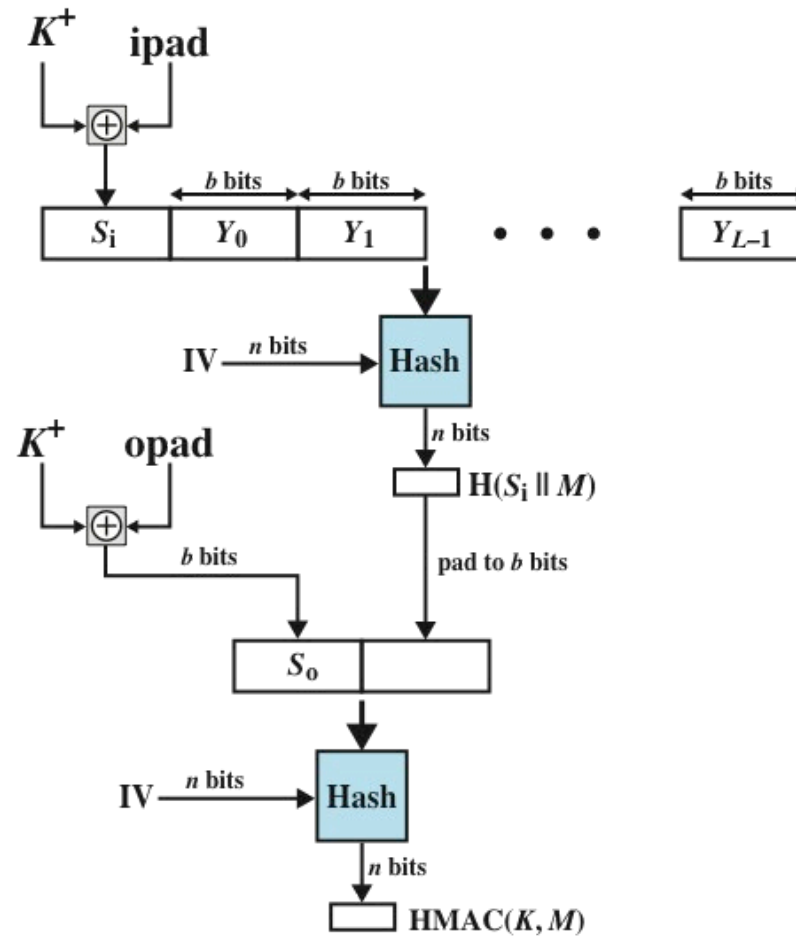
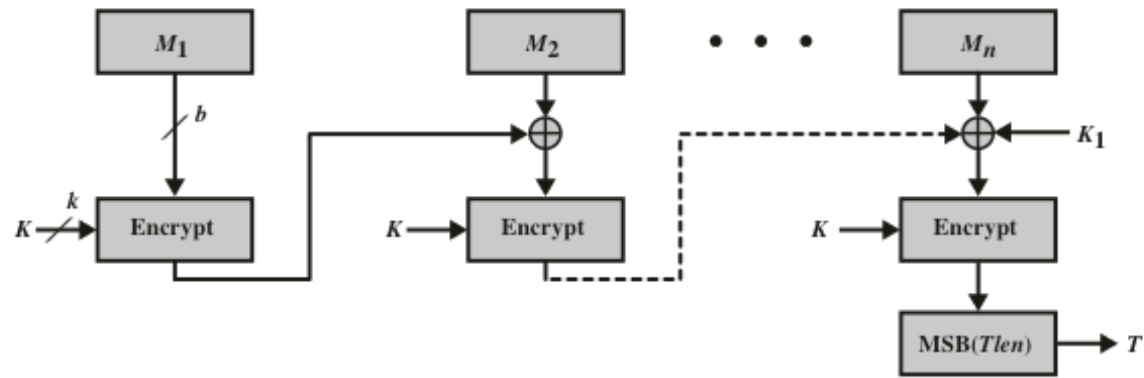
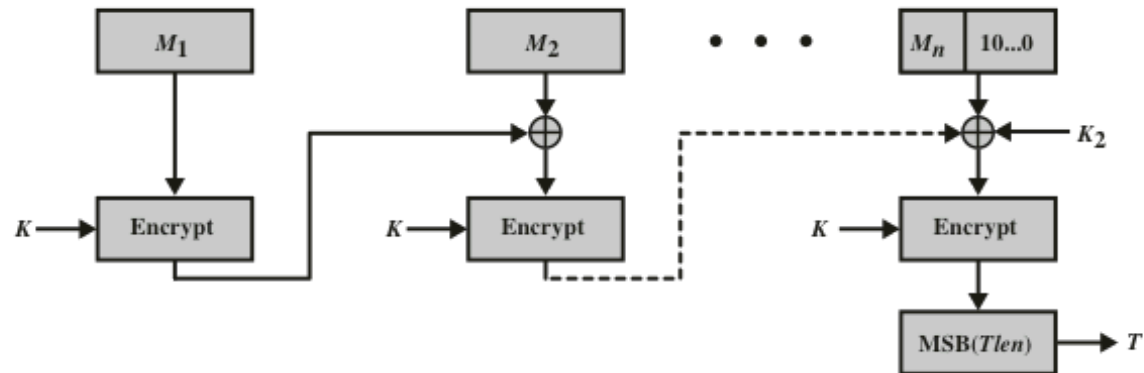


Figure 3.6 HMAC Structure



(a) Message length is integer multiple of block size

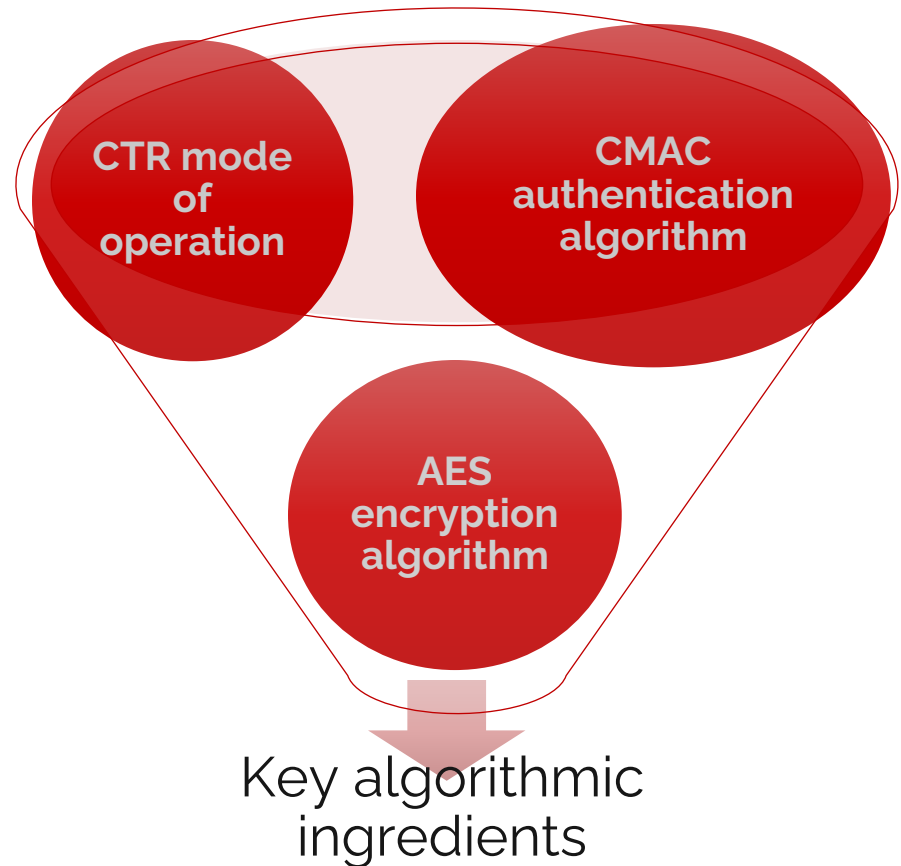


(b) Message length is not integer multiple of block size

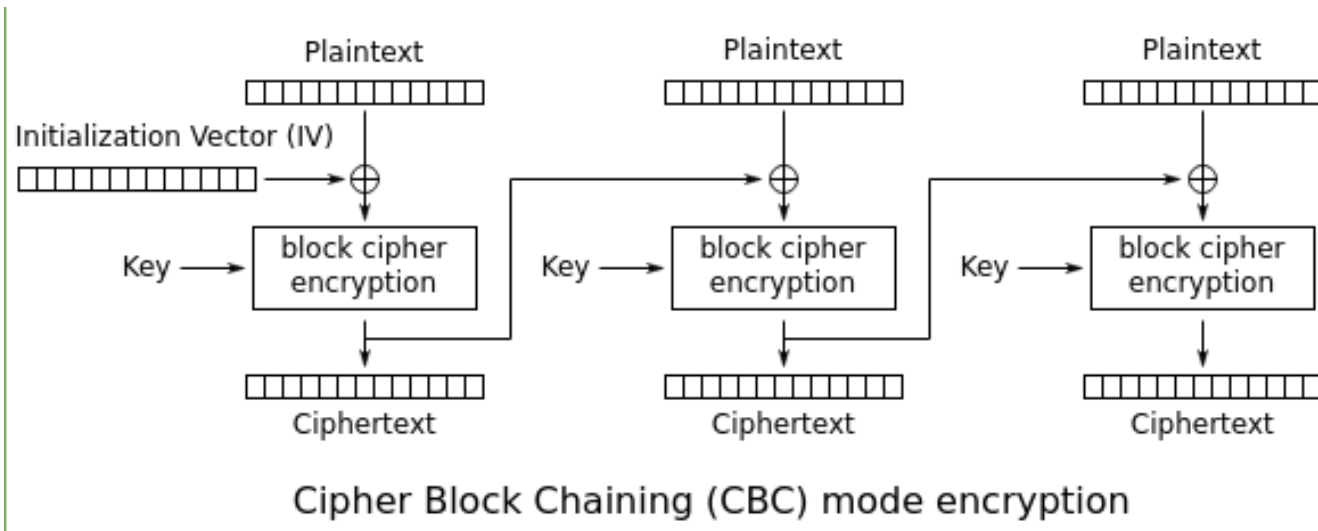
Figure 3.7 Cipher-Based Message Authentication Code (CMAC)

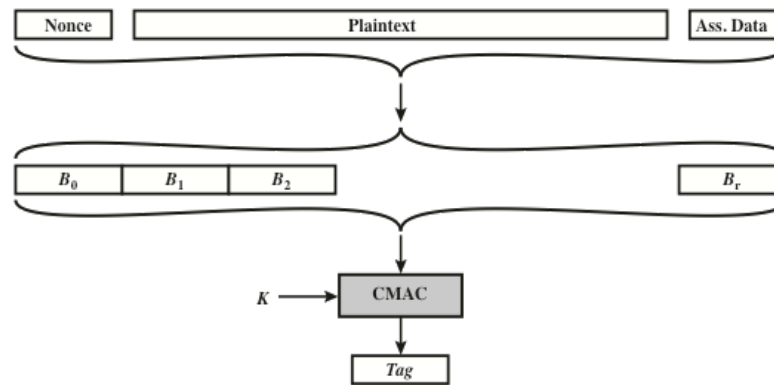
Counter with Cipher Block Chaining-Message Authentication Code (CCM)

- NIST standard SP 800-38C
- Referred to as an *authenticated encryption* mode
 - “Authenticated encryption” is a term used to describe encryption systems that simultaneously protect confidentiality and authenticity of communications
- A single key is used for both encryption and MAC algorithms

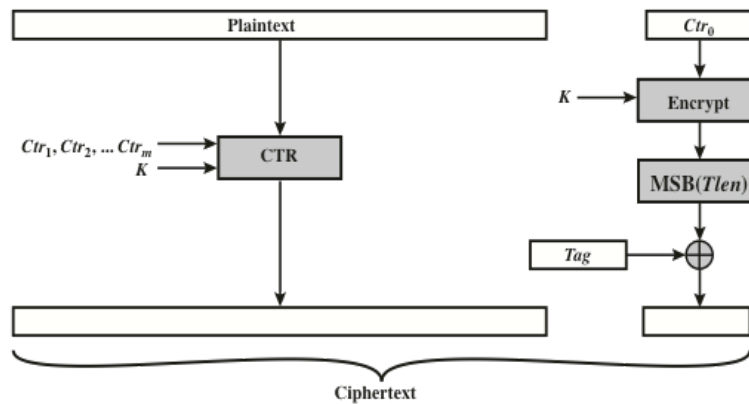


from last week.....





(a) Authentication



(b) Encryption

Figure 3.8 Counter with Cipher Block Chaining-Message Authentication Code (CCM)

Public Key Cryptography Principles



Public-Key Encryption Structure

- First publicly proposed by Diffie and Hellman in 1976
- Based on mathematical functions rather than on simple operations on bit patterns
- Is asymmetric, involving the use of two separate keys

Misconceptions:

- Public-key encryption is more secure from cryptanalysis than conventional encryption
- Public-key encryption is a general-purpose technique that has made conventional encryption obsolete
- There is a feeling that key distribution is trivial when using public-key encryption, compared to the rather cumbersome handshaking involved with key distribution centers for conventional encryption

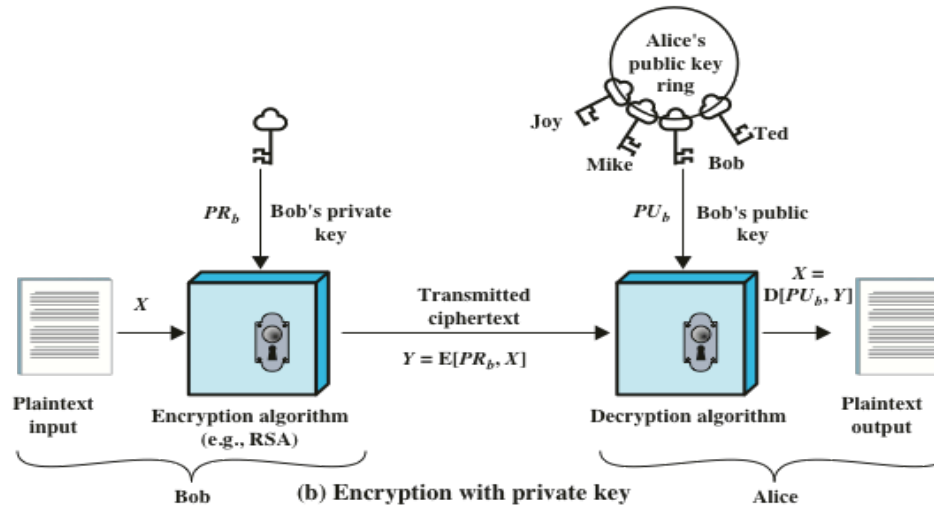
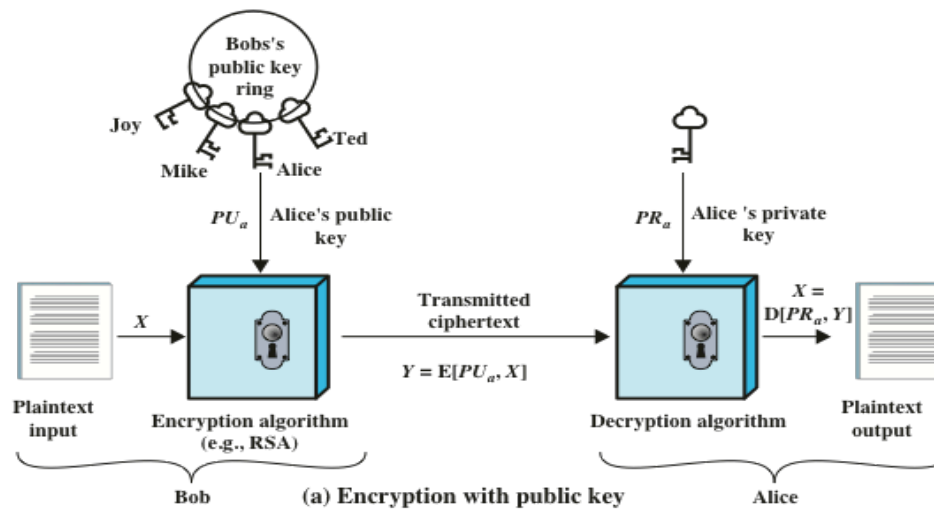


Figure 3.9 Public-Key Cryptography



Applications for Public-key Cryptosystems

- Public-key systems are characterized by the use of a cryptographic type of algorithm with two keys, one held private and one available publicly
- Depending on the application, the sender uses either the sender's private key, the receiver's public key, or both to perform some type of cryptographic function

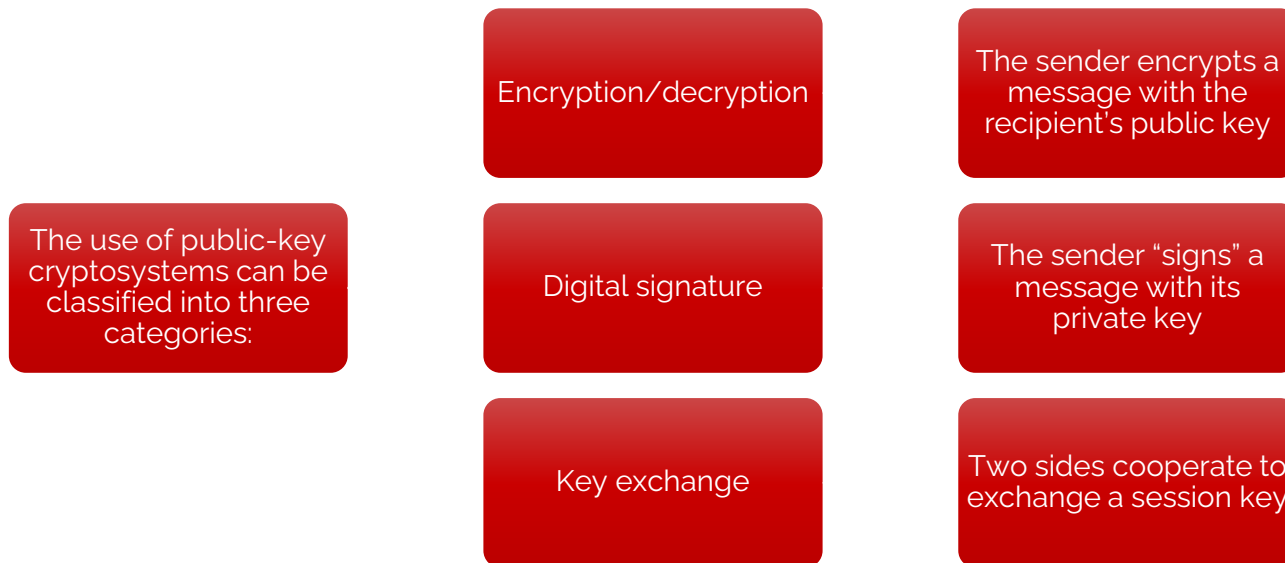




Table 3.2

applications for public-key cryptosystems

Algorithm	Encryption/Decryption	Digital Signature	Key Exchange
RSA	Yes	Yes	Yes
Diffie-Hellman	No	No	Yes
DSS	No	Yes	No
Elliptic Curve	Yes	Yes	Yes

Public Key Cryptography Algorithms



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 3.10 The RSA Algorithm

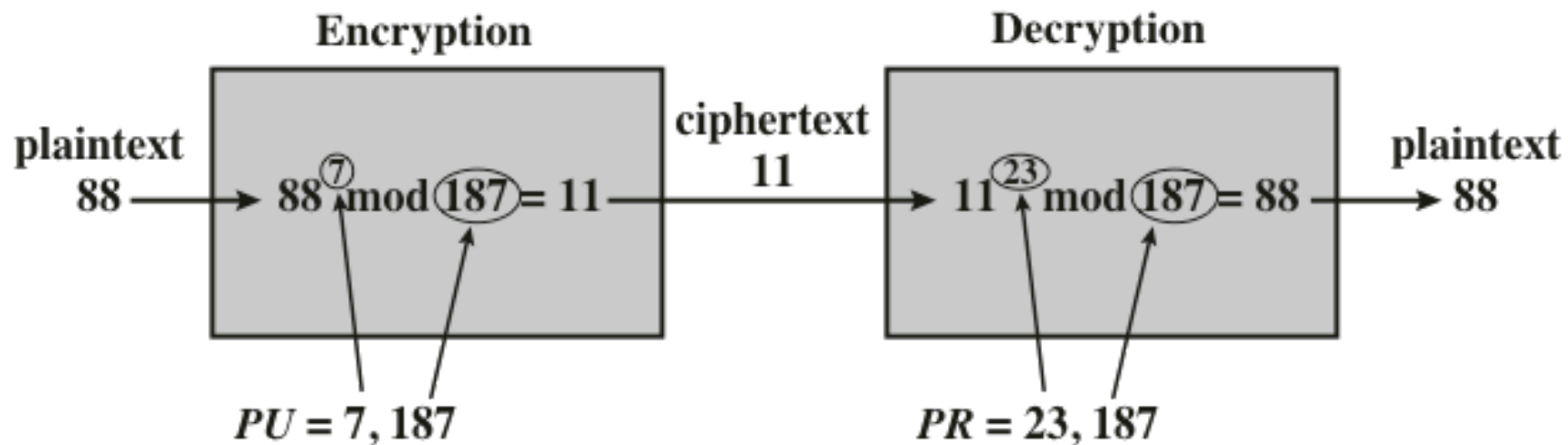
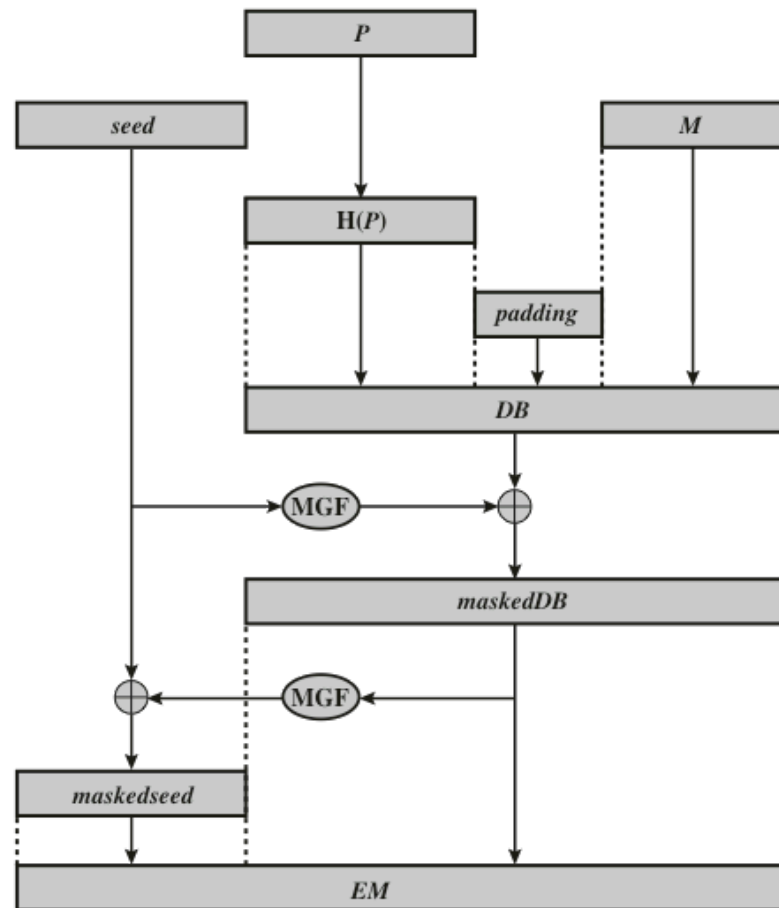


Figure 3.11 Example of RSA Algorithm



Security considerations

- The security of RSA depends on it being used in such a way as to counter potential attacks
- Possible attack approaches are:
 - ***Mathematical attacks***
 - ***Timing attacks***
 - ***Chosen ciphertext attacks***
- To counter sophisticated chosen ciphertext attacks, RSA Security Inc recommends modifying the plaintext using a procedure known as **optimal asymmetric encryption padding** (OAEP)



P = encoding parameters
 M = message to be encoded
 H = hash function

DB = data block
 MGF = mask generating function
 EM = encoded message

Figure 3.12 Encryption Using Optimal Asymmetric Encryption Padding (OAEP)

Diffie-Hellman Key Exchange

- First published public-key algorithm
- A number of commercial products employ this key exchange technique
- Purpose of the algorithm is to enable two users to exchange a secret key securely that then can be used for subsequent encryption of messages
 - The algorithm itself is limited to the exchange of the keys
- Depends for its effectiveness on the difficulty of computing discrete logarithms





Alice



Bob

Alice and Bob share a prime q and α , such that $\alpha < q$ and α is a primitive root of q

Alice generates a private key X_A such that $X_A < q$

Alice calculates a public key $Y_A = \alpha^{X_A} \bmod q$

Alice receives Bob's public key Y_B in plaintext

Alice calculates shared secret key $K = (Y_B)^{X_A} \bmod q$



Alice and Bob share a prime q and α , such that $\alpha < q$ and α is a primitive root of q

Bob generates a private key X_B such that $X_B < q$

Bob calculates a public key $Y_B = \alpha^{X_B} \bmod q$

Bob receives Alice's public key Y_A in plaintext

Bob calculates shared secret key $K = (Y_A)^{X_B} \bmod q$



Figure 3.13 Diffie-Hellman Key Exchange

Man-In-The-Middle Attack

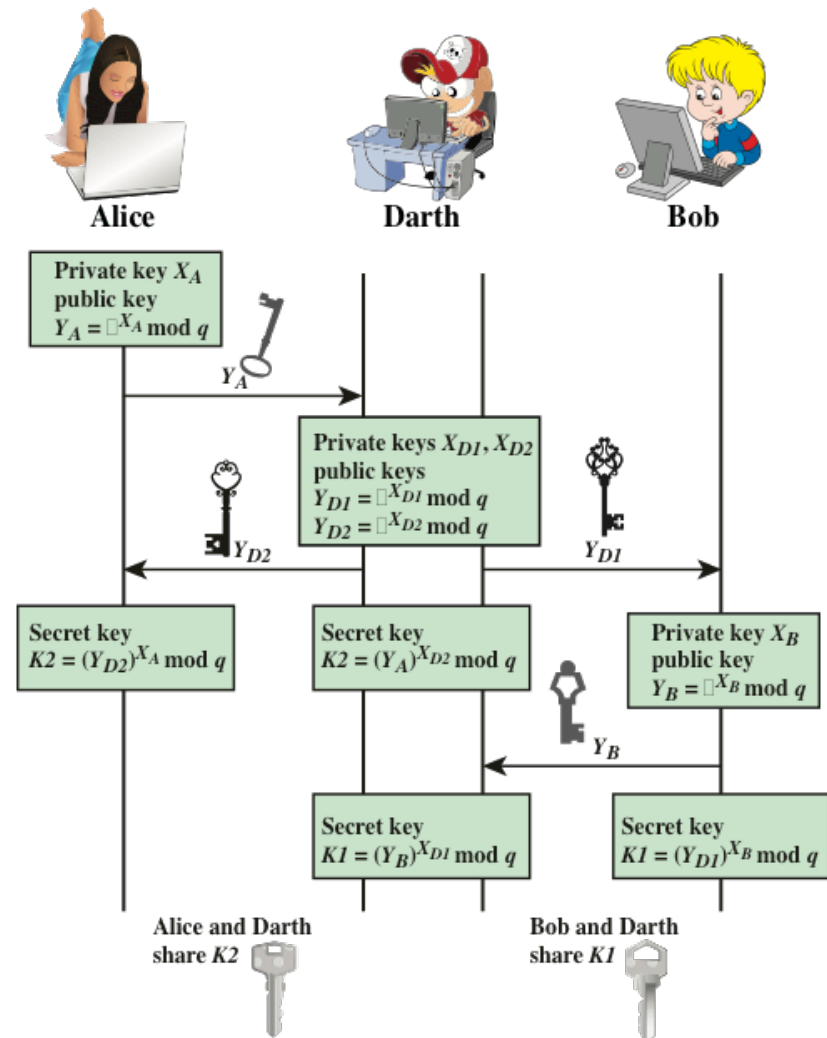


Figure 3.14 Man-in-the-Middle Attack



Other Public Key Crypto Algorithms

Elliptic-curve cryptography (ECC)

- Technique is based on the use of a mathematical construct known as the elliptic curve
- Principal attraction of ECC compared to RSA is that it appears to offer equal security for a far smaller bit size, thereby reducing processing overhead
- The confidence level in ECC is not yet as high as that in RSA

Digital Signatures



Digital Signature standard (DSS)

- FIPS PUB 186
- Makes use of the SHA-1 and presents a new digital signature technique, the Digital Signature Algorithm (DSA)
- Originally proposed in 1991 and revised in 1993 and again in 1996
- Uses an algorithm that is designed to provide only the digital signature function
- Unlike RSA, it cannot be used for encryption or key exchange



Digital signatures

- NIST FIPS PUB 186-4 (Digital Signature Standard (DSS)) defines a digital signature as:
 - **“the result of a cryptographic transformation of data that, when properly implemented, provides a mechanism for verifying origin authentication, data integrity, and signatory non-repudiation”**
- Thus, a digital signature is a data-dependent bit pattern, generated by an agent as a function of a file, message, or other form of data block
- FIPS 186-4 specifies the use of one of three digital signature algorithms:
 - Digital signature algorithm (DSA)
 - RSA digital signature algorithm
 - Elliptic curve digital signature algorithm (ECDSA)

Digital Signature Generation and Verification

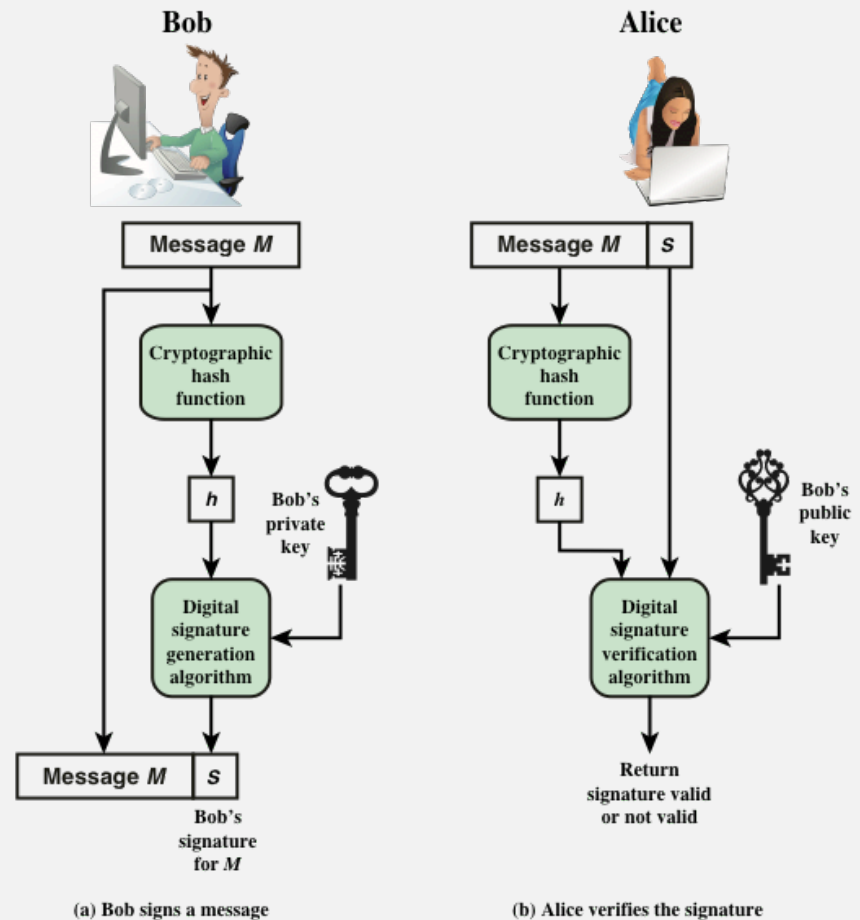


Figure 3.15 Simplified Depiction of Essential Elements of Digital Signature Process

RSA Digital Signature Algorithm

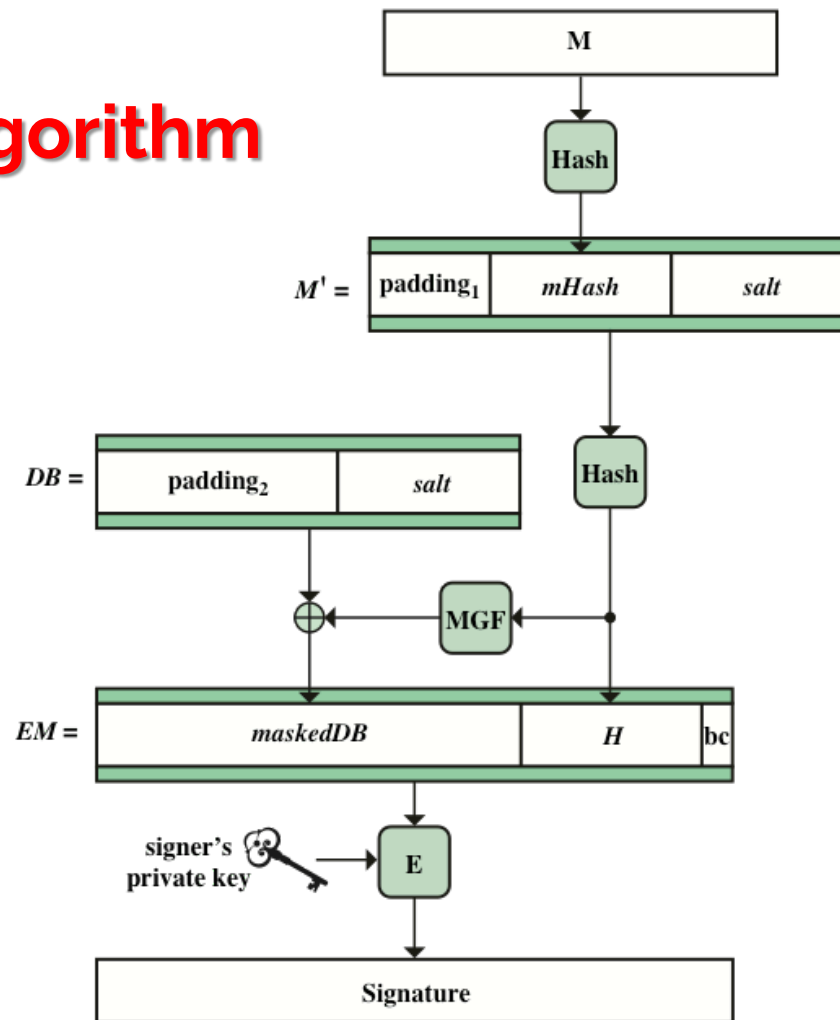


Figure 13.16 RSA-PSS Encoding and Signature Generation



Summary

- Approaches to message authentication
 - Authentication using conventional encryption
 - Message authentication without message encryption
- Secure hash functions
 - Hash function requirements
 - Security of hash functions
 - Simple hash functions
 - The SHA secure hash function
 - SHA-3
- Digital signatures
 - Digital signature generation and verification
 - RSA digital signature algorithm
- Message authentication codes
 - HMAC
 - MACs based on block ciphers
- Public-key cryptography principles
 - Public-key encryption structure
 - Applications for public-key cryptosystems
 - Requirements for public-key cryptography
- Public-key cryptography algorithms
 - The RSA public-key encryption algorithm
 - Diffie-Hellman key exchange
 - Other public-key cryptography algorithms

Week 3 Discussion Options

1. Digital signatures are growing in popularity. Write a post about what the risks and the benefits are to using DSS for major financial purchases?

•Or

2. Read this article from threatpost.com where they project what we will see in 2021 from an infosec standpoint. <https://threatpost.com/5-cybersecurity-trends-2022/177273/> Which of their forecasts (if any) do you agree or disagree with? Justify your answer.

•Or

- Anything else related to the course content or ISM program that is worthy of discussion