



Data Security: Public Key Cryptography

Dr. Dan Massey

Read Computer Security: Principle and Practices Chapter 2

Additional Details From Chapter 21

Chap 21: Public-Key Cryptography and Message Authentication

Responsible only for the content in these slides

Returning to Our Motivating Problems:

Confidentiality: how could we encrypt a message? (email, web/http, sms, etc)

Apply AES – select key size and mode

But assumes sender and receiver share a secret key

Integrity: how could we authenticate a message?

Apply Message Authentication Code

But assumes sender and receiver share a secret key

(availability – not primary motivation now, provided the approach is feasible)

Public-Key Encryption Structure

Publicly proposed by Diffie and Hellman in 1976

Based on mathematical functions

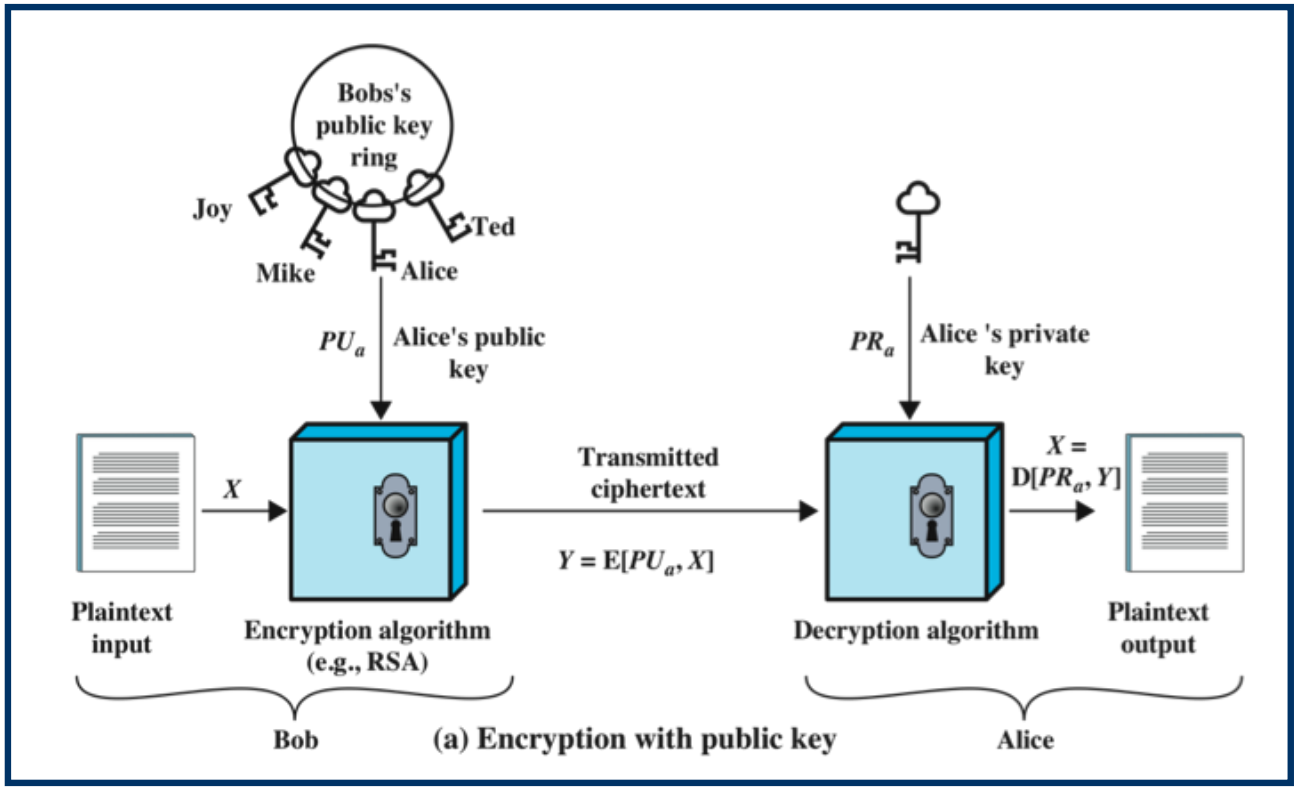
Asymmetric

- Uses two separate keys
- Public key and private key
- Public key is made public for others to use

Some form of protocol is needed for distribution

Public and Private Key Pairs

- No longer share a single secret (symmetric) key
- Entity generates a key pair
  - *Private key* known only to the entity
  - *Public key* made available to everyone
- Confidentiality:
  - Anyone can send confidential data to the entity
  - Encipher a message using the entity's public key
  - Entity deciphers message using the private key
- Integrity:
  - Entity enciphers message with its private key
  - Anyone can decipher the message with the entity's public key
  - Can be used to show message was authentic and unaltered.



- **Plaintext**
  - Readable message or data that is fed into the algorithm as input
- **Encryption algorithm**
  - Performs transformations on the plaintext
- **Public and private key**
  - Pair of keys, one for encryption, one for decryption
- **Ciphertext**
  - Scrambled message produced as output
- **Decryption key**
  - Produces the original plaintext

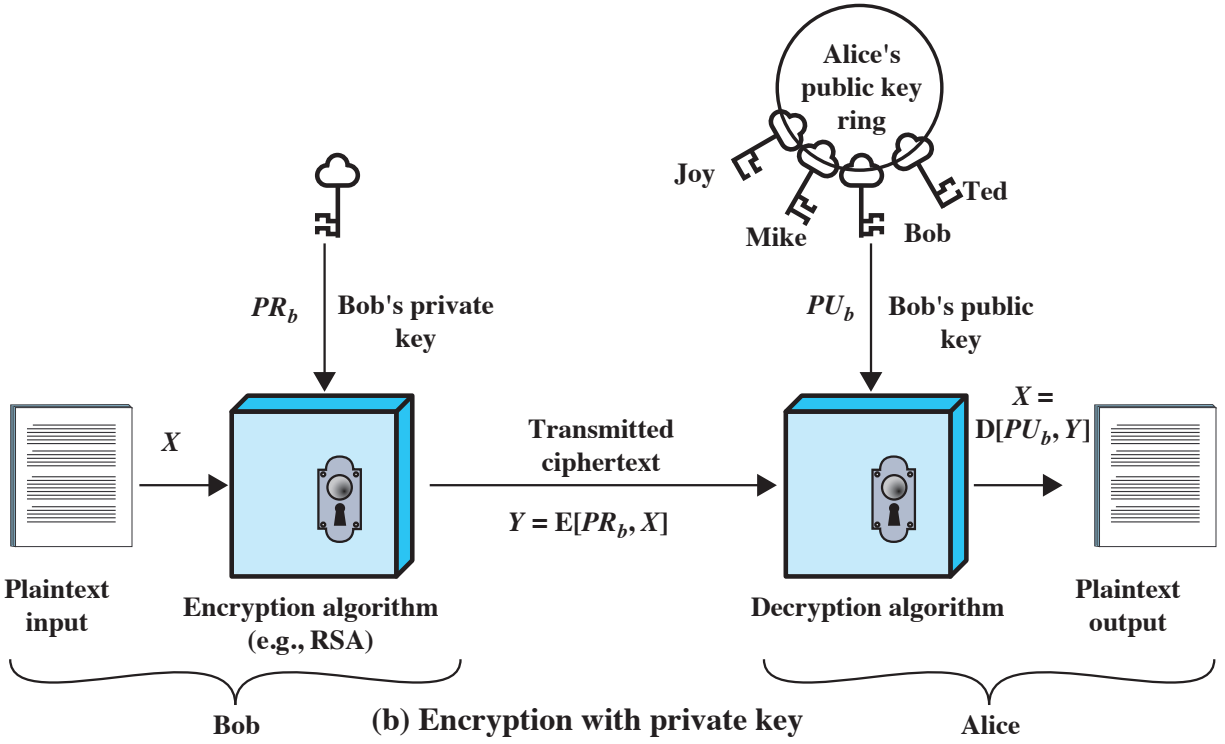


Figure 2.6 Public-Key Cryptography

- User encrypts data using his or her own private key
- Anyone who knows the corresponding public key will be able to decrypt the message

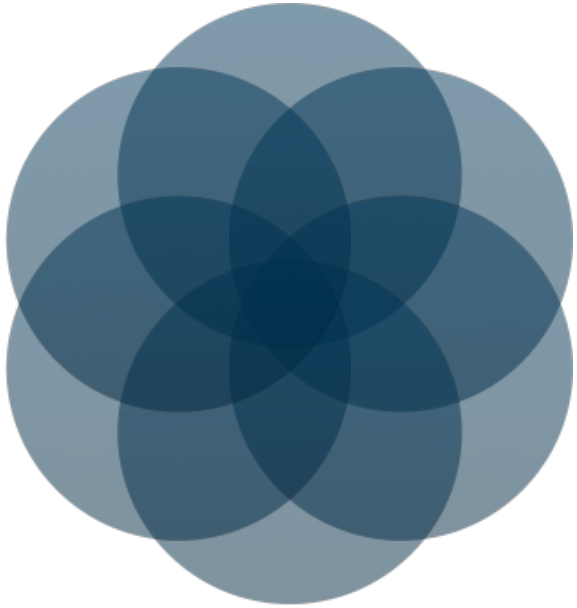
Applications for Public-Key Cryptosystems

Algorithm	Digital Signature	Symmetric Key Distribution	Encryption of Secret Keys
RSA	Yes	Yes	Yes
Diffie-Hellman	No	Yes	No
DSS	Yes	No	No
Elliptic Curve	Yes	Yes	Yes

Requirements for Public-Key Cryptosystems

Computationally easy to create key pairs

Useful if either key can be used for each role



Computationally easy for sender knowing public key to encrypt messages

Computationally infeasible for opponent to otherwise recover original message

Computationally easy for receiver knowing private key to decrypt ciphertext

Computationally infeasible for opponent to determine private key from public key

Asymmetric Encryption Algorithms

RSA (Rivest, Shamir, Adleman)	Developed in 1977	Most widely accepted and implemented approach to public-key encryption	Block cipher in which the plaintext and ciphertext are integers between 0 and n-1 for some n.
Diffie-Hellman key exchange algorithm	Enables two users to securely reach agreement about a shared secret that can be used as a secret key for subsequent symmetric encryption of messages	Limited to the exchange of the keys	
Digital Signature Standard (DSS)	Provides only a digital signature function with SHA-1	Cannot be used for encryption or key exchange	
Elliptic curve cryptography (ECC)	Security like RSA, but with much smaller keys		

Basics of the RSA Algorithm (1/2)

- Relies on the difficulty of determining the number of numbers relatively prime to a large integer *n*
- Totient function $\phi(n)$
  - Number of positive integers less than *n* and relatively prime to *n*
    - *Relatively prime* means with no factors in common with *n*
- Example: $\phi(10) = 4$
  - 1, 3, 7, 9 are relatively prime to 10
- Example: $\phi(21) = 12$
  - 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20 are relatively prime to 21

Basics of the RSA Algorithm (2/2)

- Choose two large prime numbers p, q
  - Let $n = pq$; then $\phi(n) = (p-1)(q-1)$
  - Choose $e < n$ such that e is relatively prime to $\phi(n)$.
  - Compute d such that $ed \bmod \phi(n) = 1$
- Public key: (e, n); private key: d
- Encipher: $c = m^e \bmod n$
- Decipher: $m = c^d \bmod n$
- *Key Intuition:*
  - *Q1: Given p and q, compute n=pq, pick e, and compute d*
  - *Q2: Given (e,n), find p, q, d.*
  - *Note that both problems can be solved. Do you want Q1 or Q2?*

Example: Confidentiality (1/2)

- Take $p = 7$, $q = 11$, so $n = pq = 77$ and $\phi(n) = (p-1)(q-1) = 60$
- Alice chooses $e = 17$, relatively prime to $\phi(n)=60$
- making $d = 53$, $ed = 17*53 = 901 \bmod (\phi(n)=60) = 1$
- Bob wants to send Alice secret message HELLO (07 04 11 11 14)
  - $c = m^e \bmod n = 07^{17} \bmod 77 = 28$
  - $04^{17} \bmod 77 = 16$
  - $11^{17} \bmod 77 = 44$
  - $11^{17} \bmod 77 = 44$
  - $14^{17} \bmod 77 = 42$
- Bob sends 28 16 44 44 42

Example: Confidentiality (2/2)

- Alice receives 28 16 44 44 42
- Alice uses private key, $d = 53$, to decrypt message:
  - $m = c^d \bmod n = 28^{53} \bmod 77 = 07$
  - $16^{53} \bmod 77 = 04$
  - $44^{53} \bmod 77 = 11$
  - $44^{53} \bmod 77 = 11$
  - $42^{53} \bmod 77 = 14$
- Alice translates message to letters to read HELLO
  - No one else could read it, as only Alice knows her private key and that is needed for decryption

Example: Integrity/Authentication (1/2)

- Take $p = 7$, $q = 11$, so $n = 77$ and $\phi(n) = 60$
- Alice chooses $e = 17$, making $d = 53$
- Alice wants to send Bob message HELLO (07 04 11 11 14) so Bob knows it is what Alice sent (no changes in transit, and authenticated)
  - $s = m^d \bmod n = 07^{53} \bmod 77 = 35$
  - $04^{53} \bmod 77 = 09$
  - $11^{53} \bmod 77 = 44$
  - $11^{53} \bmod 77 = 44$
  - $14^{53} \bmod 77 = 49$
- Alice sends 35 09 44 44 49

Example: Integrity/Authentication (2/2)

- Bob receives 35 09 44 44 49
- Bob uses Alice's public key, $e = 17$, $n = 77$, to decrypt message:
  - $m = s^e \bmod n = 35^{17} \bmod 77 = 07$
  - $09^{17} \bmod 77 = 04$
  - $44^{17} \bmod 77 = 11$
  - $44^{17} \bmod 77 = 11$
  - $49^{17} \bmod 77 = 14$
- Bob translates message to letters to read HELLO
  - Alice sent it as only she knows her private key, so no one else could have enciphered it
  - If (enciphered) message's blocks (letters) altered in transit, would not decrypt properly

Example Limitations

- Encipher message in blocks considerably larger than the examples here
  - If 1 character per block, RSA can be broken using statistical attacks (just like classical cryptosystems)
  - Attacker cannot alter letters, but can rearrange them and alter message meaning
    - Example: reverse enciphered message of text ON to get NO

Security of RSA

Brute force

- Involves trying all possible private keys

Mathematical attacks

- There are several approaches, all equivalent in effort to factoring the product of two primes

Timing attacks

- These depend on the running time of the decryption algorithm

Chosen ciphertext attacks

- This type of attack exploits properties of the RSA algorithm

Progress in Factorization

Number of Decimal Digits	Number of Bits	Date Achieved
100	332	April 1991
110	365	April 1992
120	398	June 1993
129	428	April 1994
130	431	April 1996
140	465	February 1999
155	512	August 1999
160	530	April 2003
174	576	December 2003
200	663	May 2005
193	640	November 2005
232	768	December 2009

Other Public-Key Algorithms

Digital Signature Standard (DSS)

- FIPS PUB 186
- Makes use of SHA-1 and the Digital Signature Algorithm (DSA)
- Originally proposed in 1991, revised in 1993 due to security concerns, and another minor revision in 1996
- Cannot be used for encryption or key exchange
- Uses an algorithm that is designed to provide only the digital signature function

Elliptic-Curve Cryptography (ECC)

- Equal security for smaller bit size than RSA
- Seen in standards such as IEEE P1363
- Confidence level in ECC is not yet as high as that in RSA
- Based on a mathematical construct known as the elliptic curve

Returning to Our Motivating Problems:

Confidentiality: how could we encrypt a message? (email, web/http, sms, etc)

Encrypt Message with receiver's public key

Integrity: how could we authenticate a message?

Sign message with sender's private key

(availability – not primary motivation now, provided the approach is feasible)

Digital Signatures

- NIST FIPS PUB 186-4 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)

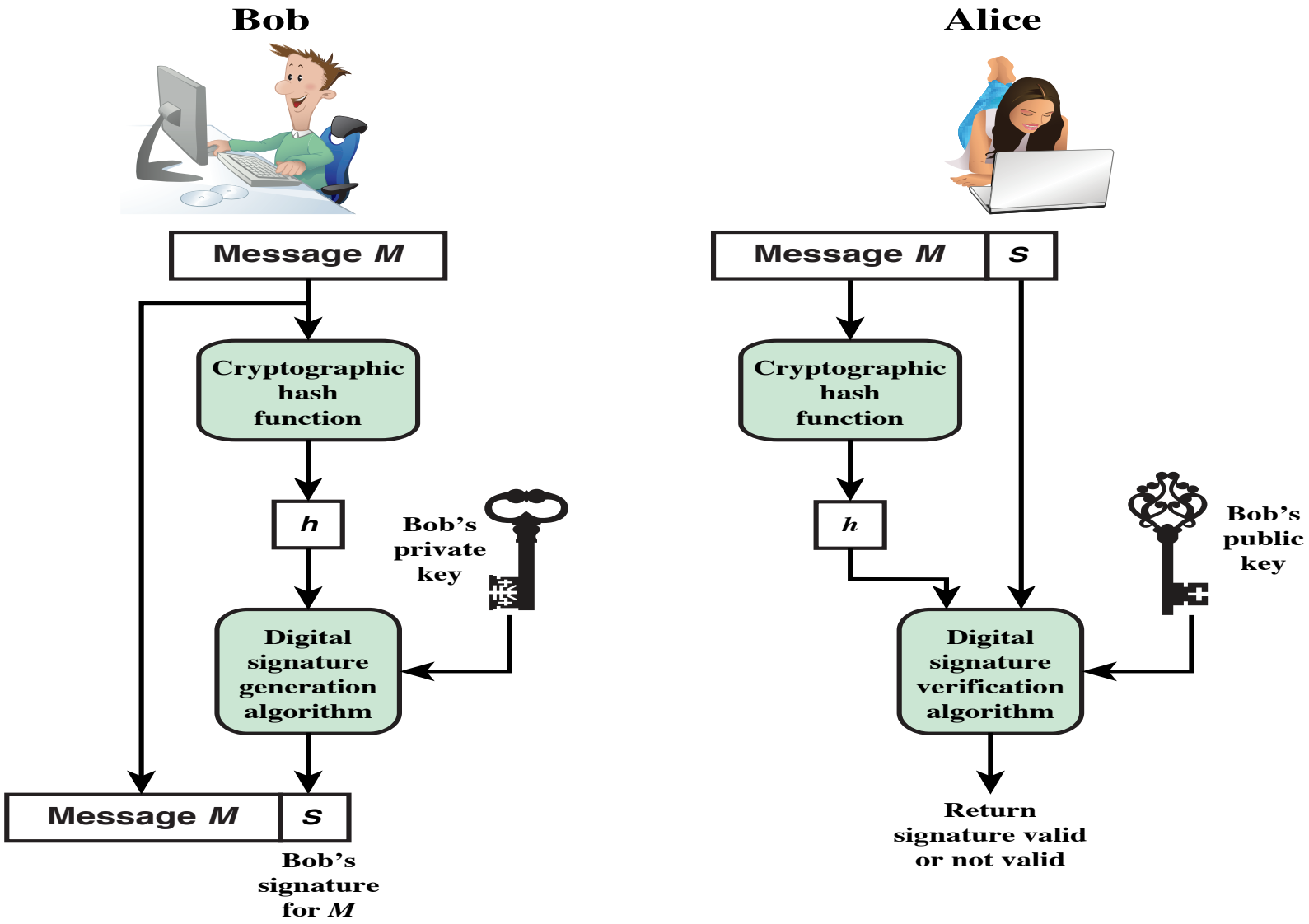


Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process

Heilmeier Questions

- What are you trying to do? Articulate your objectives using absolutely no jargon.
- How is it done today, and what are the limits of current practice?
- What is new in your approach and why do you think it will be successful?
- Who cares? If you succeed, what difference will it make?
- What are the risks?
- How much will it cost?
- How long will it take?
- What are the mid-term and final “exams” to check for success?