Week 2

CRYPTOGRAPHIC TOOLS

USER AUTHENTICATION

Chapter 2

Cryptographic Tools

Cryptography – What is it?

- Scrambling of Information so that it is unreadable to attackers
  - Usually requires an algorithm to encrypt and decrypt
- Transforming information into a secure form
  - Passwords in databases
  - PII (Personal Identifiable Information)
- Storing important data in cleartext form (unencrypted) can lead to major consequences

CIA Triad

Cryptography in context of to the CIA triad

- Confidentiality
  - Only authorized parties (with proper decrypting tools) can view it
- Integrity
  - Ensures information is correct and unaltered through encryption
- Availability
  - Encryption/decryption is quick and is not disruptive

Cryptographic Algorithms

- Symmetric Encryption Algorithms
- Hashing Algorithms
- Asymmetric Encryption Algorithms

Symmetric Encryption

- The “conventional” single key encryption performed by encrypting and decrypting information with a shared key
- Requirements include
  - A strong encryption algorithm
  - Securely shared secret key between sender and receiver

Symmetric Encryption Process

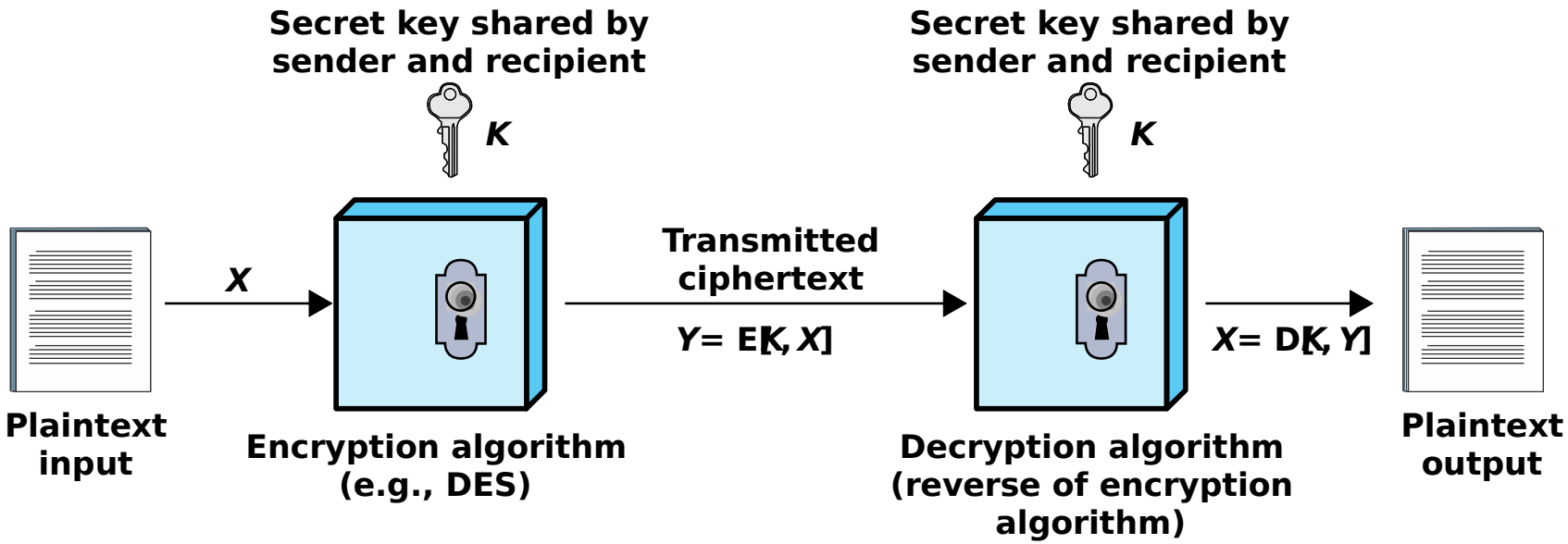


Figure 2.1 Simplified Model of Symmetric Encryption

Comparison of 3 Popular Symmetric Block Encryption Algorithms

	DES	Triple DES	AES
Plaintext block size (bits)	64	64	128
Ciphertext block size (bits)	64	64	128
Key size (bits)	56	112 or 168	128, 192, or 256

DES = Data Encryption Standard
AES = Advanced Encryption Standard

Average Time Required for Exhaustive Key Search (Block Cipher)

Key size (bits)	Cipher	Number of Alternative Keys	Time Required at 10^9 decryptions/s	Time Required at 10^{13} decryptions/s
56	DES	$2^{56} \approx 7.2 \times 10^{16}$	2^{55} ns = 1.125 years	1 hour
128	AES	$2^{128} \approx 3.4 \times 10^{38}$	2^{127} ns = 5.3×10^{21} years	5.3×10^{17} years
168	Triple DES	$2^{168} \approx 3.7 \times 10^{50}$	2^{167} ns = 5.8×10^{33} years	5.8×10^{29} years
192	AES	$2^{192} \approx 6.3 \times 10^{57}$	2^{191} ns = 9.8×10^{40} years	9.8×10^{36} years
256	AES	$2^{256} \approx 1.2 \times 10^{77}$	2^{255} ns = 1.8×10^{60} years	1.8×10^{56} years

Stream Cipher and Pseudorandom Byte Generator

- Each stream of data fed in as plain text is encrypted one byte at a time
- A Pseudorandom byte generator which operates and generates a stream of 8-bit numbers based on the provided key
- The data and the generated 8-bit number is combined using the bitwise exclusive OR (XOR) operation
- Stream cipher can be as secure as a block cipher with proper design of the pseudorandom number generator

Block vs Stream Cipher

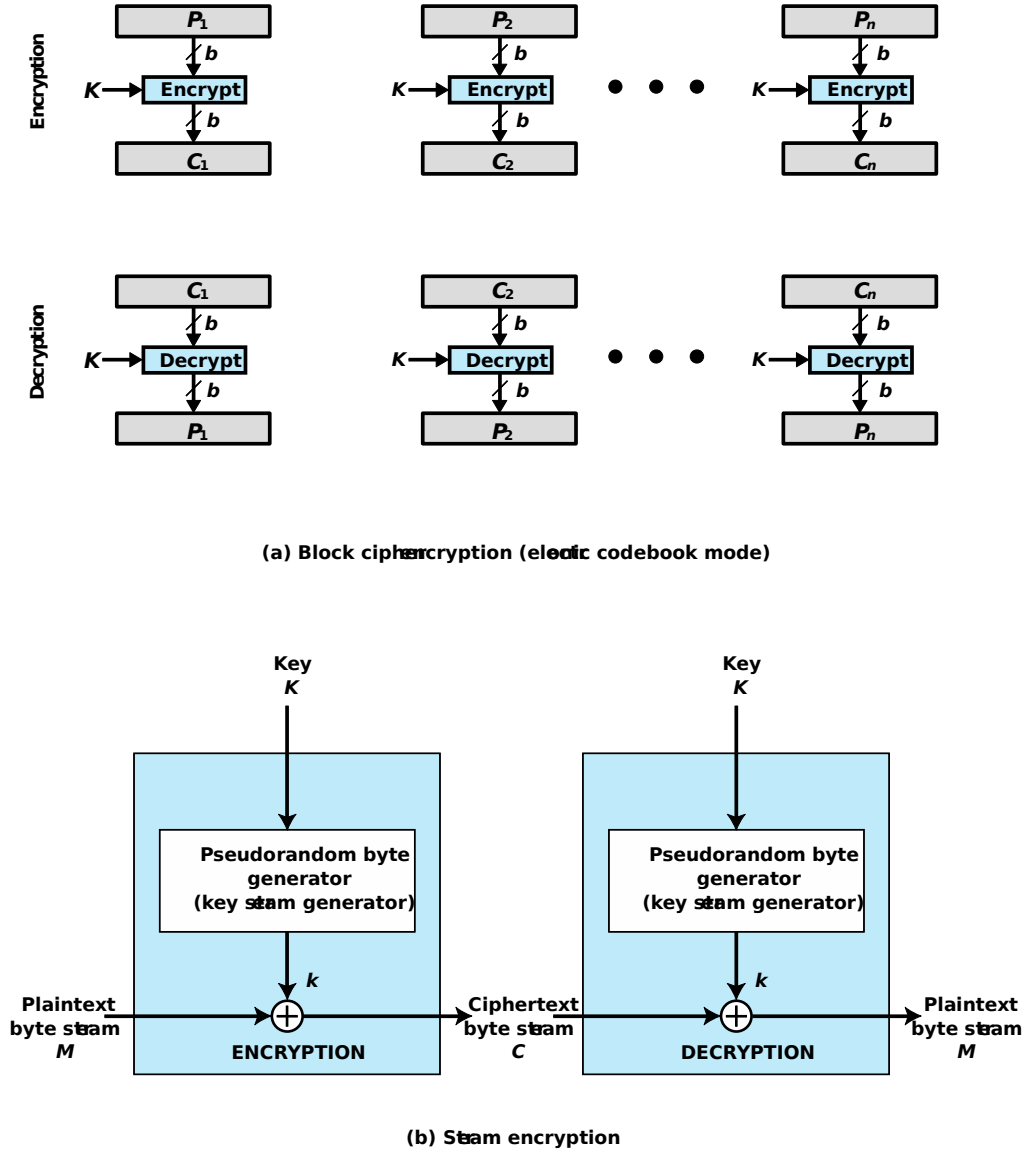
Block Cipher

- Processes the input one block of elements at a time
- Produces an output block for each input block
- Can reuse keys
- More common

Stream Cipher

- Processes the input elements continuously
- Produces output one element at a time
- Primary advantage is that they are almost always faster and use far less code
- Encrypts plaintext one byte at a time
- Pseudorandom stream is one that is unpredictable without knowledge of the input key

Block vs Stream Cipher



(a) Block cipher encryption (electronic codebook mode)

(b) Stream encryption

Figure 2.2 Types of Symmetric Encryption

Block vs Stream Cipher

- Stream cipher is fast if plaintext data is short
- Stream cipher consume more processing power if plain text is long
- Stream cipher more prone to attack because the engine generating stream does not vary much

Attacking Symmetric Encryption

CRYPTANALYTIC ATTACK

- Relies on
  - Having information about the nature of algorithm
  - Having information about the data and its characteristic
  - Some sample of cipher and text pairs
- Based on deducing and reverse engineering the plaintext or key being used
  - Once broken, all future and past messages encrypted with the key are compromised

BRUTE-FORCE ATTACKS

- Try all possible keys until an intelligible translation is obtained
  - On average requires half of all possible keys to be tried before achieving success

Practical Security Issues

Symmetric encryption is applied to a unit of data larger than a single 64-bit or 128-bit block.

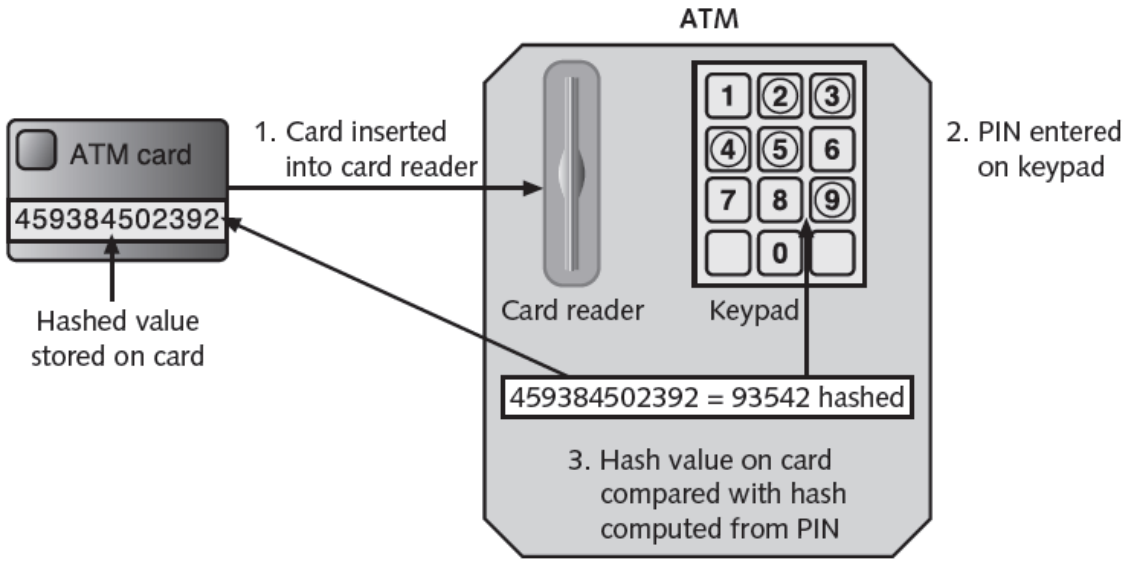
This means that data must be broken up (Email messages, database data)

The simplest approach to this type of multi-block encryption is ECB (Electronic Codebook), which is not secure for lengthy messages.

Cryptanalysts may be able to exploit regularities in the plain text and make it easier to decrypt.

Hashing Algorithm

- Most basic type of algorithm
- Creates a unique digital fingerprint using given data (fixed size)
- Contents cannot be used to reveal the original given data
- Primarily used for comparison purposes



- Other examples
  - Software executables (md5 hash)

Hash Function

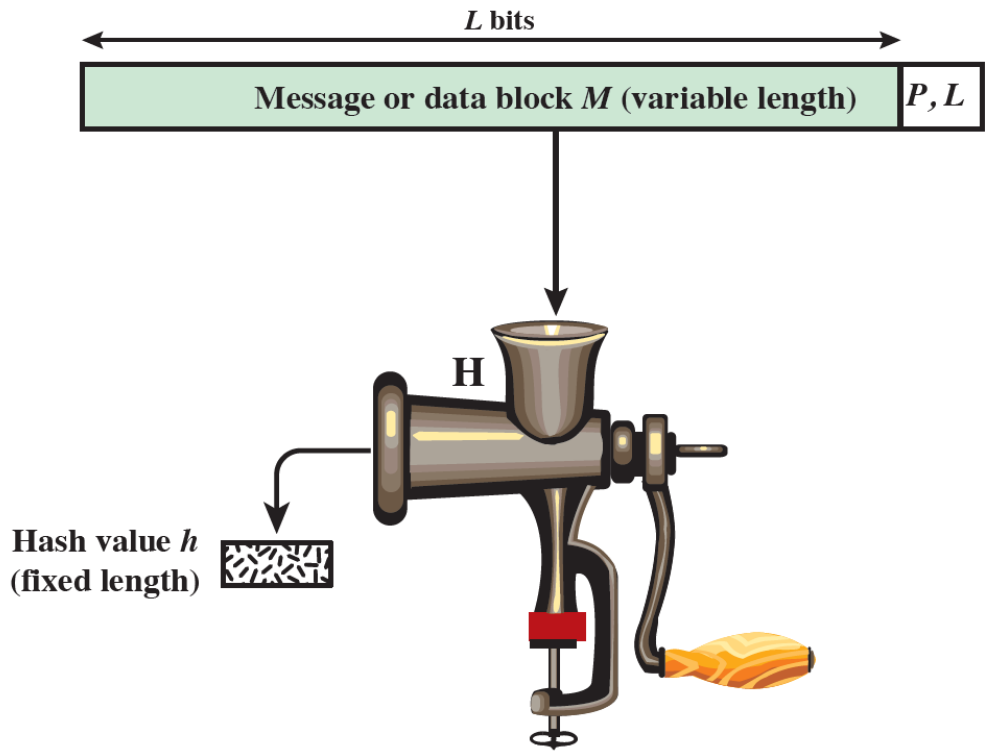
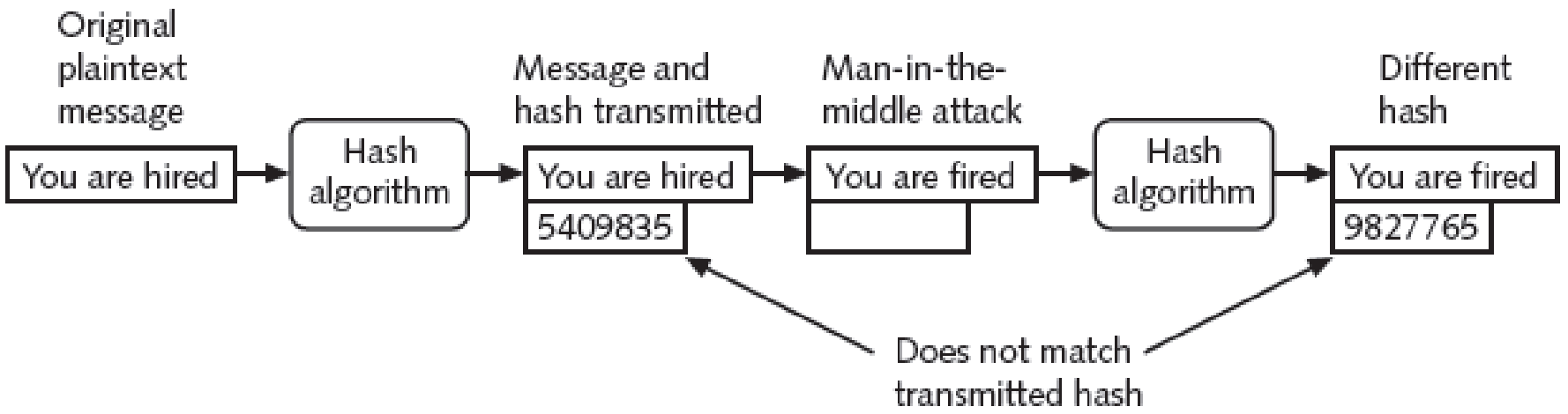


Figure 2.4 Cryptographic Hash Function; $h = H(M)$

Hashing Algorithms – Data Integrity

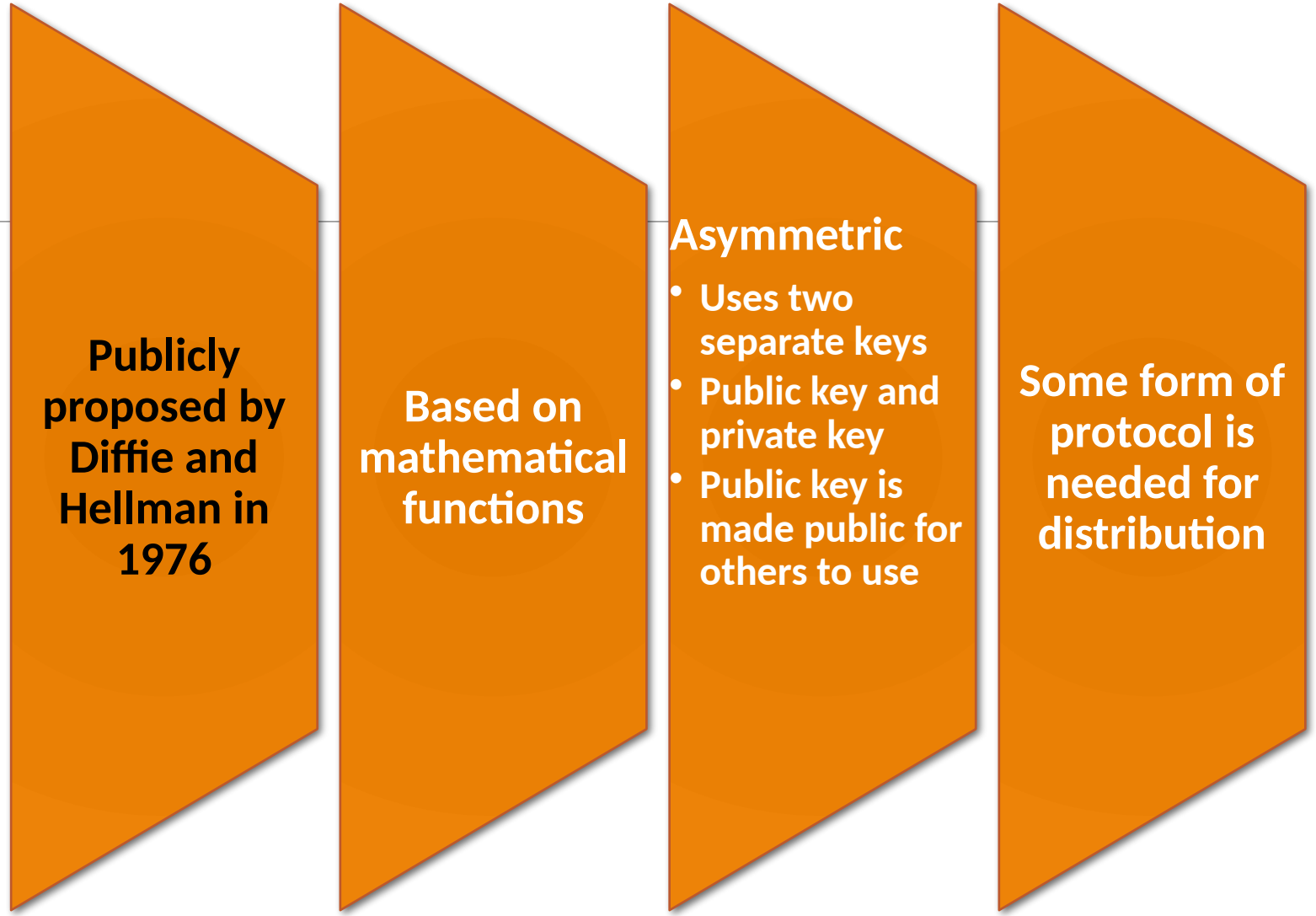


Hash Function Requirements

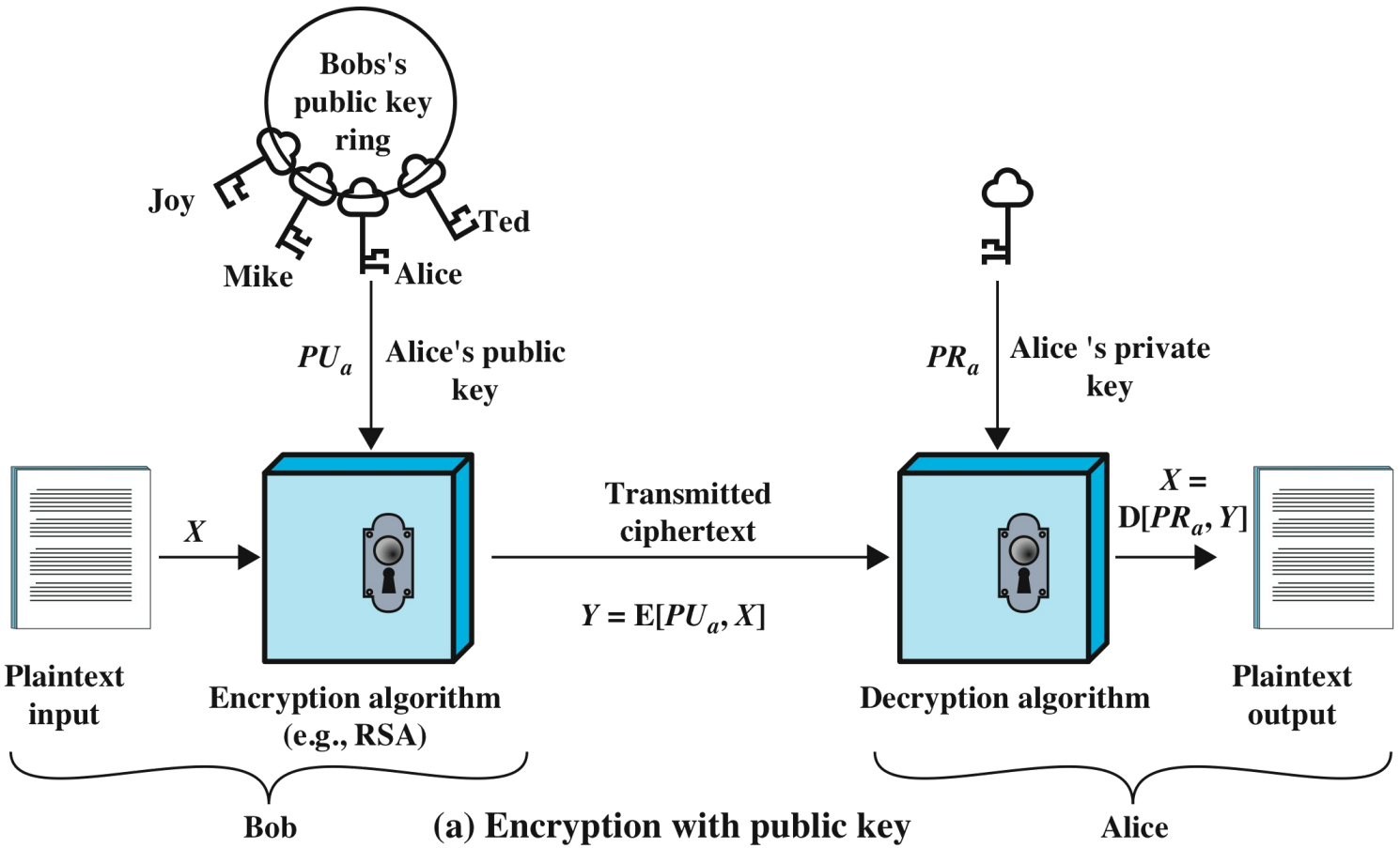
- A given hash function H
  - Can be applied to a block of data of any size
  - Produces a fixed-length output
  - Is relatively easy to compute for any given x
  - Is not reversible (pre-image resistant)
  - It is not possible to find an alternative message with the same hash value as a given message (Weak collision resistant)
  - Does not produce the same digest given two different inputs (Strong collision resistant)

Asymmetric Encryption

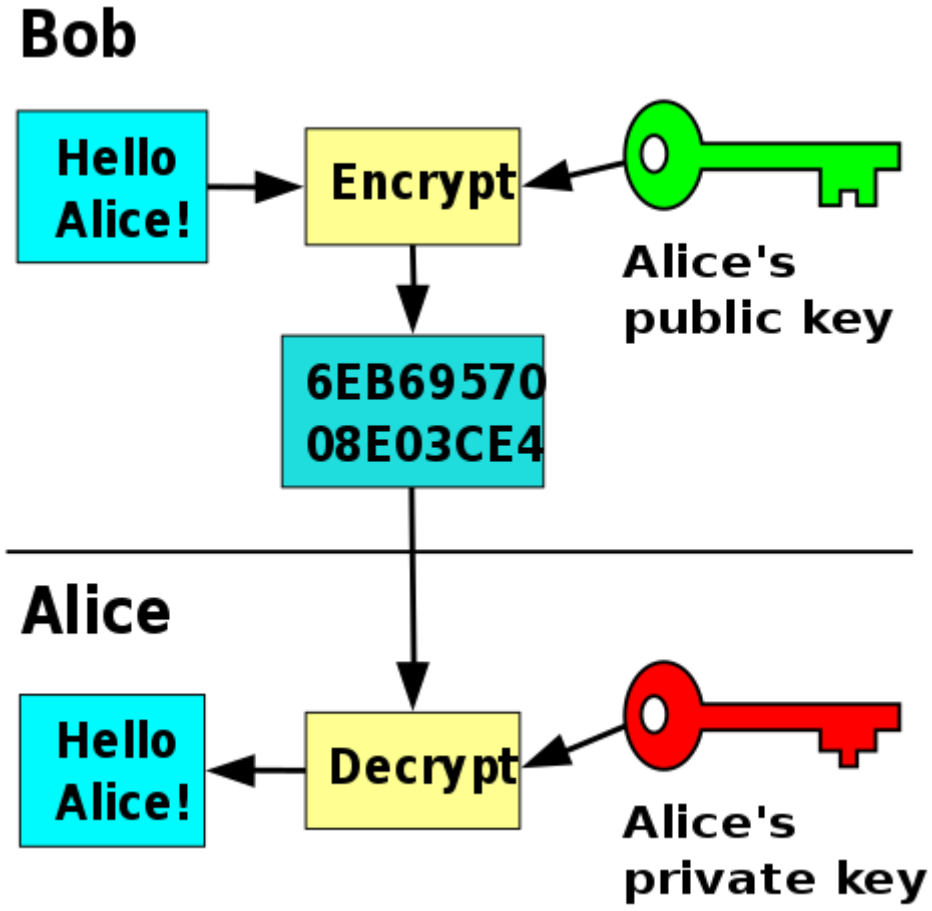
Public-key Encryption Structure



How Does Public Key Encryption Work?

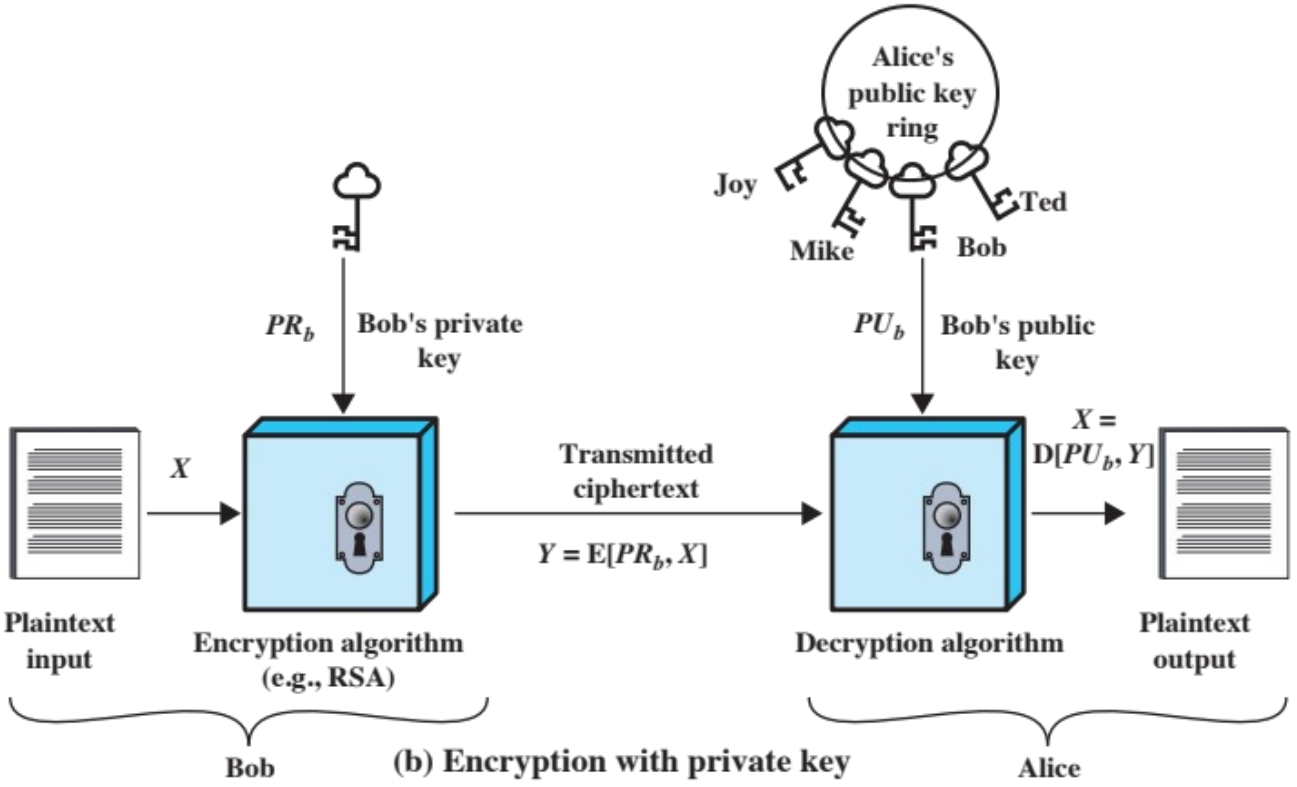


(a) Encryption with public key



Public Key Encrypti on Example

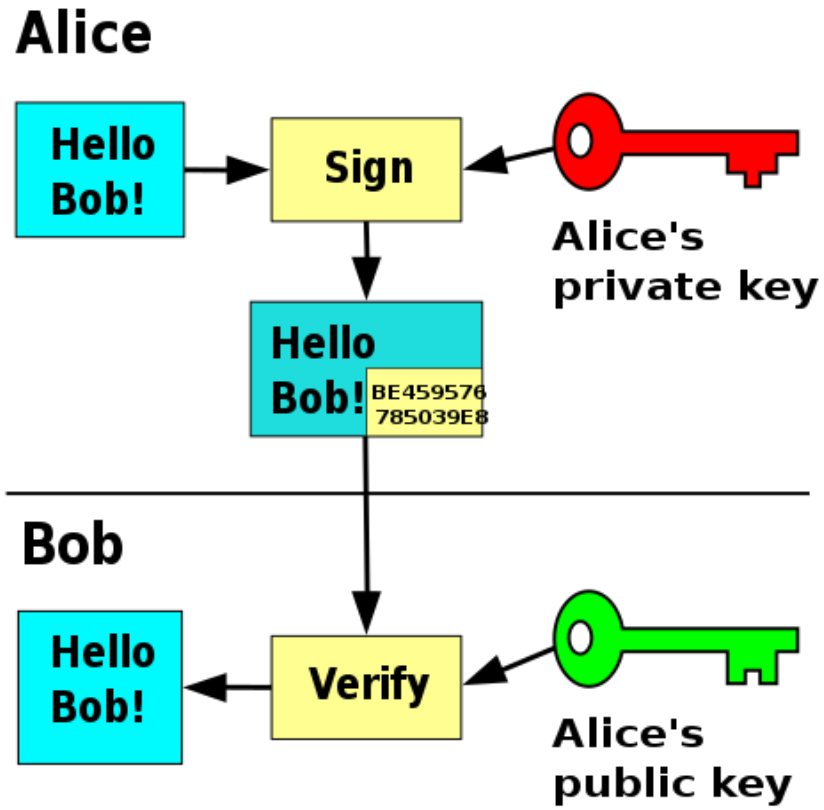
Encrypting email

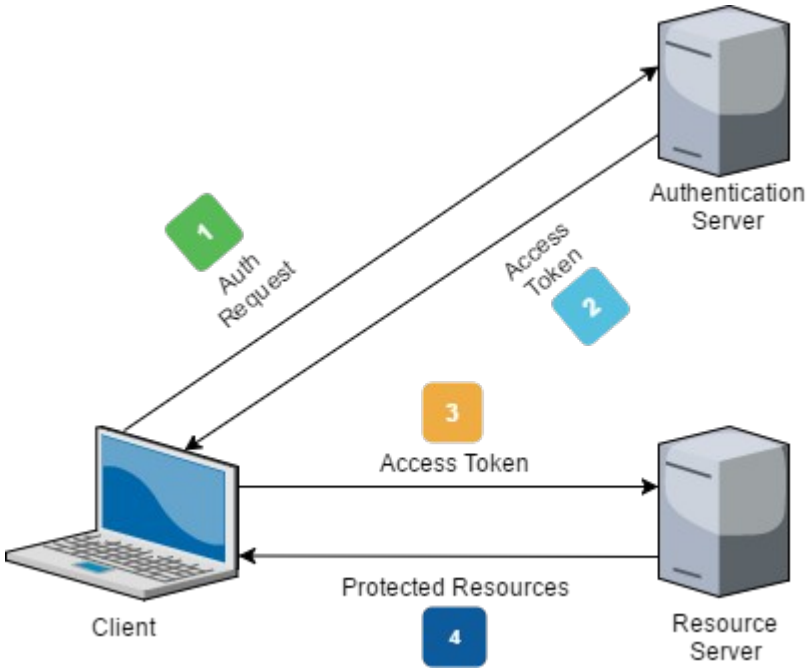


(b) Encryption with private key

Figure 2.6 Public-Key Cryptography

What about Private key Cryptography?





JWT token / verification

Private Key Cryptography Example

Public-key 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)

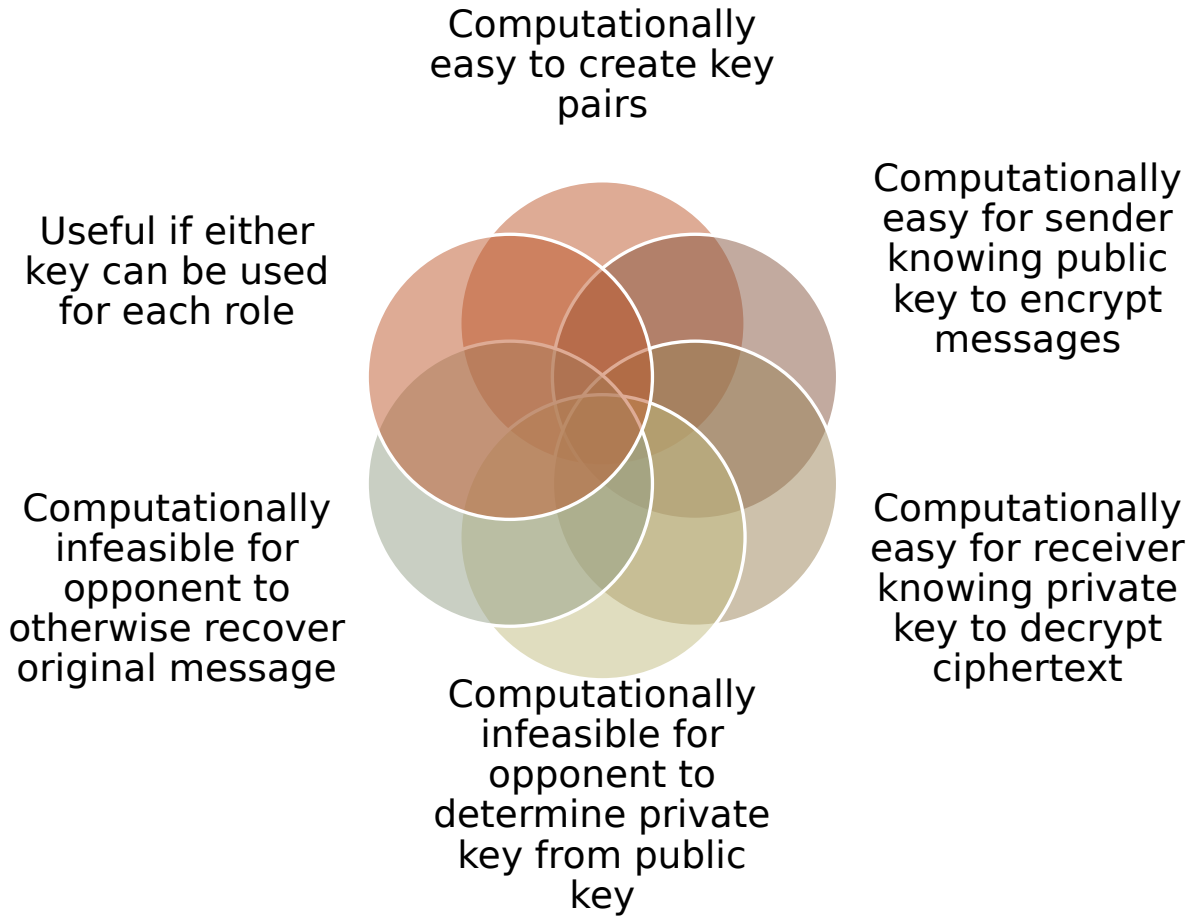
- Provides only a digital signature function with SHA-1
- Cannot be used for encryption or key exchange

Elliptic curve cryptography (ECC)

- Security like RSA, but with much smaller keys

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 Systems



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)

Digital Signature Depicted

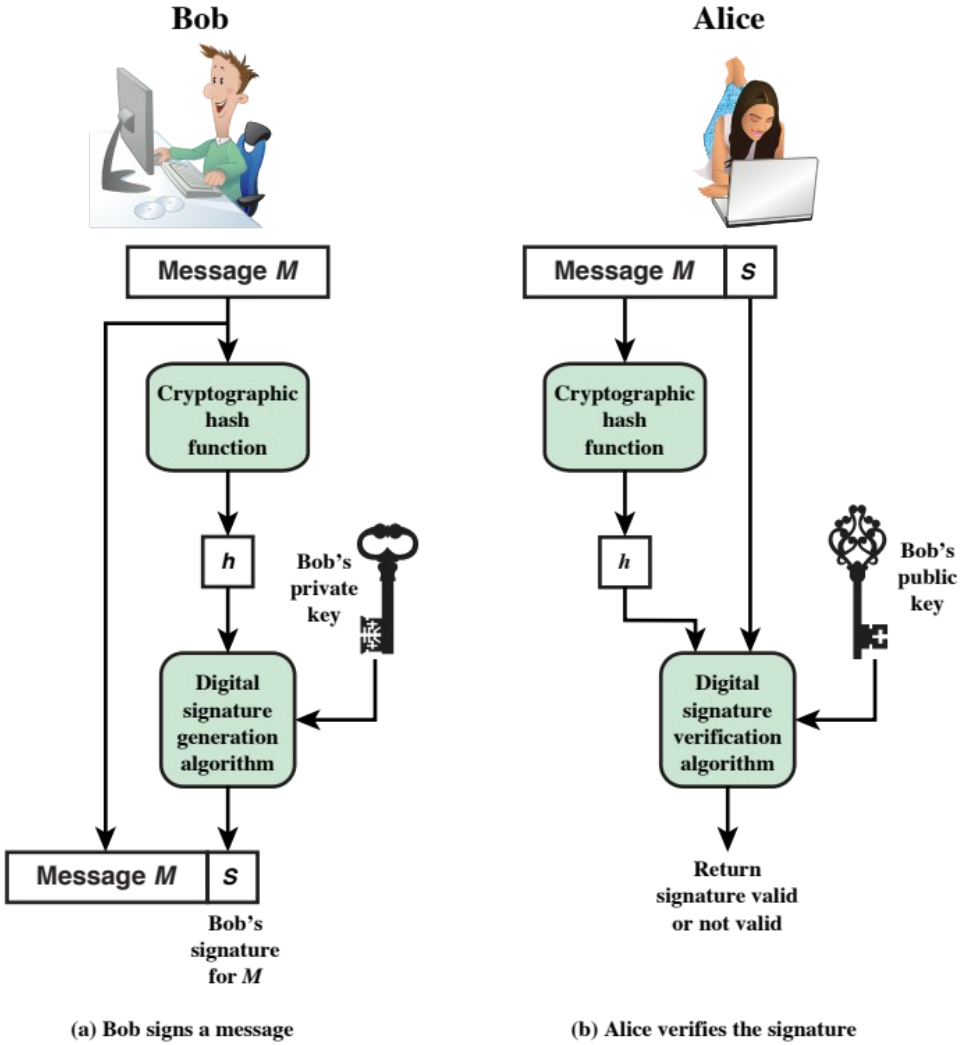


Figure 2.7 Simplified Depiction of Essential Elements of Digital Signature Process

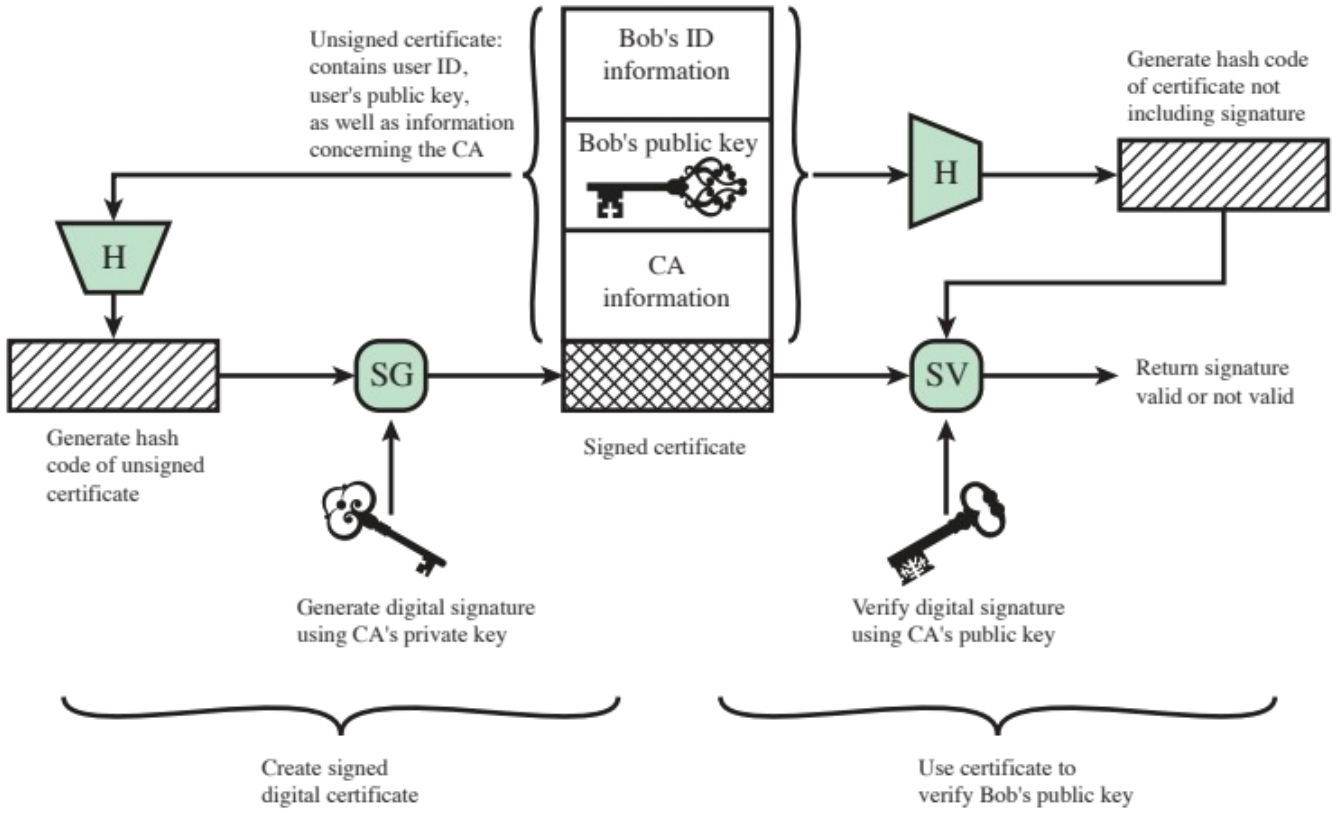


Figure 2.8 Public-Key Certificate Use

Certificate Authorities ' (CA) role in Digital Signatures

How do we know Bob's public key was used by Bob and that the sender was actually Bob?

Role of Random Numbers in Encryption

- Keys for public-key algorithms
- Stream key for symmetric stream cipher
- Symmetric key for use as a temporary session key
- Handshaking to prevent replay attacks
- Session key

Random Number Requirements

Randomness

- Criteria:
  - Uniform distribution
    - Frequency of occurrence of each of the numbers should be approximately the same
  - Independence
    - No one value in the sequence can be inferred from the others

Unpredictability

- Each number is statistically independent of other numbers in the sequence
- Opponent should not be able to predict future elements of the sequence on the basis of earlier elements

Random vs Pseudorandom

Cryptographic applications typically make use of algorithmic techniques for random number generation

- Algorithms are deterministic and therefore produce sequences of numbers that are not statistically random

Pseudorandom numbers are:

- Sequences produced that satisfy statistical randomness tests
- Likely to be predictable

True random number generator (TRNG):

- Uses a nondeterministic source to produce randomness
- Most operate by measuring unpredictable natural processes
  - e.g. radiation, gas discharge, leaky capacitors
- Increasingly provided on modern processors

Chapter 3

User Authentication

User Authentication - Definition

NIST SP 800-63-3 (Digital Authentication Guideline, October 2016) defines digital user authentication as:

- “The process of establishing confidence in user identities that are presented electronically to an information system.”

Security Requirements

Table 3.1 Identification and Authentication Security Requirements (SP 800-171)

	Basic Security Requirements:
1	Identify information system users, processes acting on behalf of users, or devices.
2	Authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.

	Derived Security Requirements:
3	Use multifactor authentication for local and network access to privileged accounts and for network access to non-privileged accounts.
4	Employ replay-resistant authentication mechanisms for network access to privileged and non-privileged accounts.
5	Prevent reuse of identifiers for a defined period.
6	Disable identifiers after a defined period of inactivity.
7	Enforce a minimum password complexity and change of characters when new passwords are created.
8	Prohibit password reuse for a specified number of generations.
9	Allow temporary password use for system logons with an immediate change to a permanent password.
10	Store and transmit only cryptographically-protected passwords.
11	Obscure feedback of authentication information.

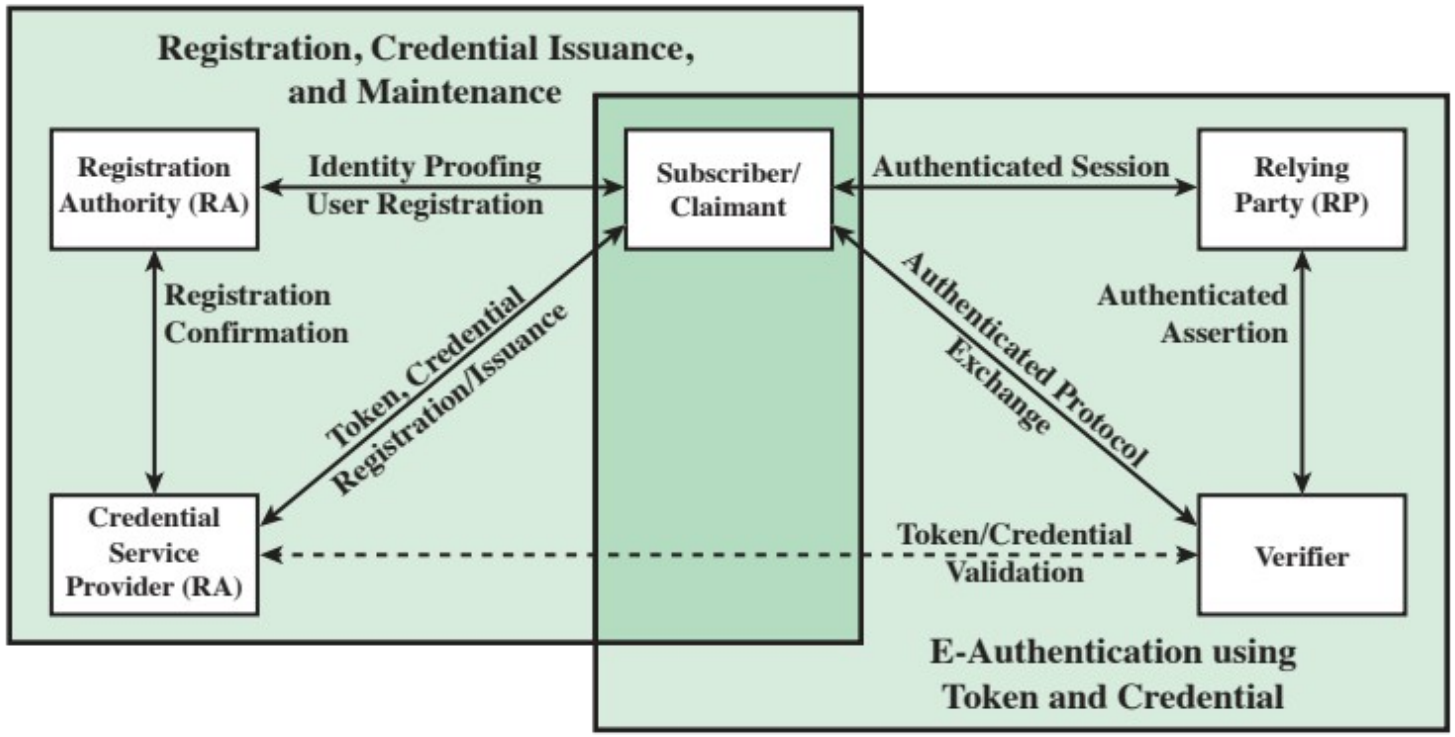


Figure 3.1 The NIST SP 800-63-2 E-Authentication Architectural Model

Authentication Architecture

Think about a website that accepts social login. The website in this case is the “Relying Party”

4 Means of Authentication

- Something the individual knows
  - PW, Pin, answers to challenge questions
- Something the individual possess (token)
  - Smartcard, electronic key card, physical key
- Something the individual is (biometrics)
  - Fingerprint, face id
- Something the individual does (dynamic biometrics)
  - Voice pattern, handwriting, typing rhythm

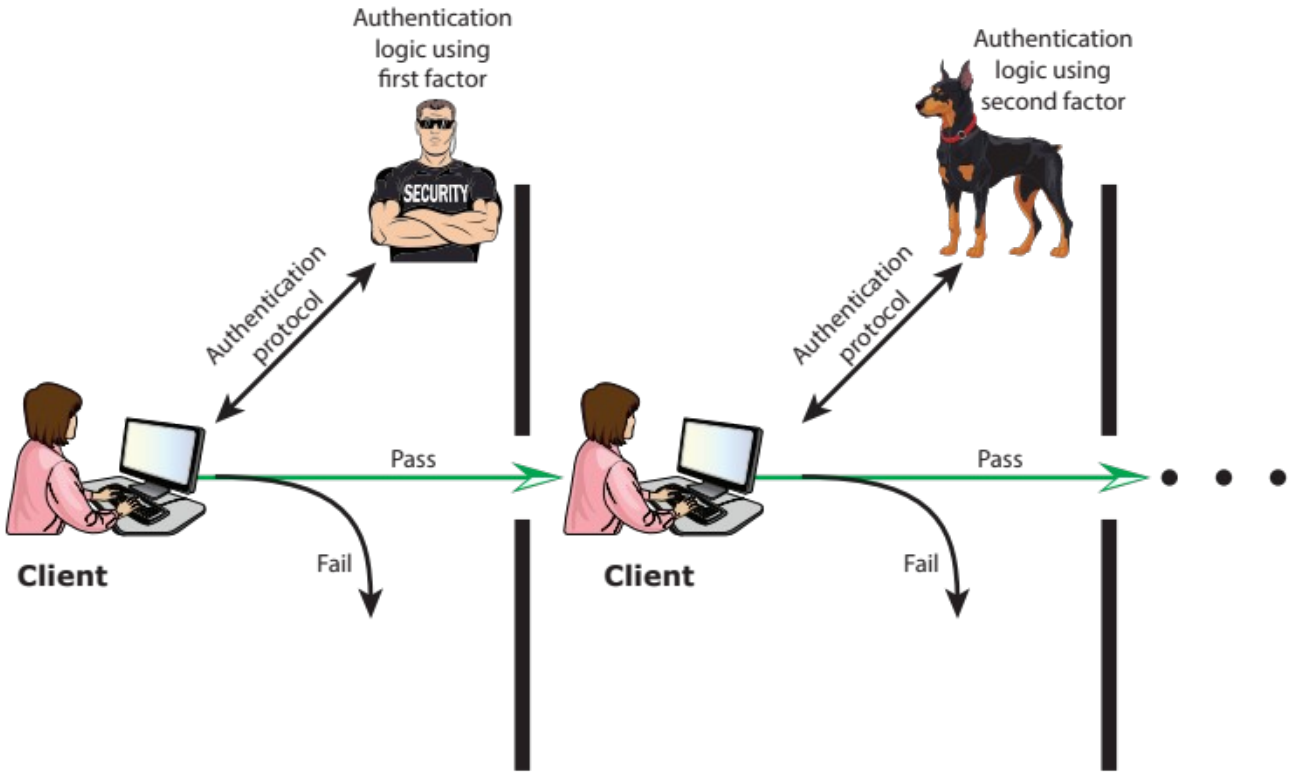
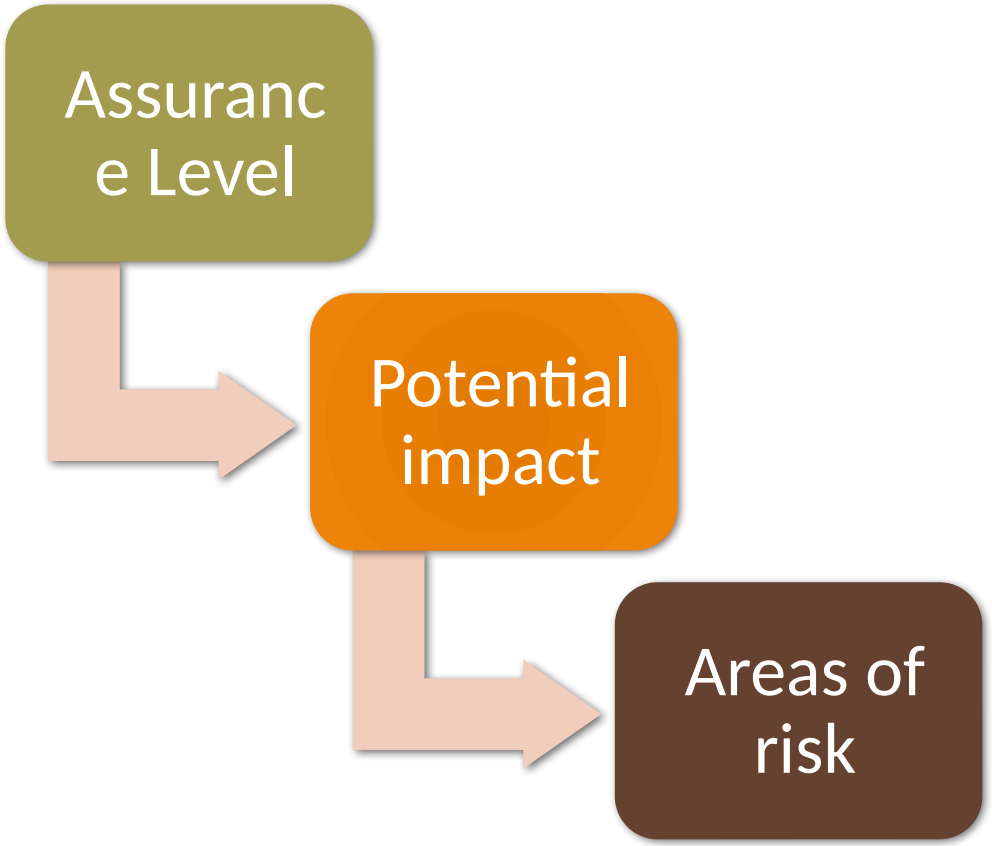


Figure 3.2 Multifactor Authentication

Multi-factor Authentication

Risk Assessment for User Authentication



Assurance Level

Describes an organization's degree of certainty that a user has presented a credential that refers to his or her identity

More specifically is defined as:

The degree of confidence in the vetting process used to establish the identity of the individual to whom the credential was issued

The degree of confidence that the individual who uses the credential is the individual to whom the credential was issued

Four levels of assurance

Level 1
- Little or no confidence in the asserted identity's validity

Level 2
- Some confidence in the asserted identity's validity

Level 3
- High confidence in the asserted identity's validity

Level 4
- Very high confidence in the asserted identity's validity

Potential Impact

FIPS 199 defines three levels of potential impact on organizations or individuals should there be a breach of security:

- Low
  - An authentication error could be expected to have a limited adverse effect on organizational operations, organizational assets, or individuals
- Moderate
  - An authentication error could be expected to have a serious adverse effect
- High
  - An authentication error could be expected to have a severe or catastrophic adverse effect

Maximum Potential Impacts for Each Assurance Level

Potential Impact Categories for Authentication Errors	Assurance Level Impact Profiles			
	1	2	3	4
Inconvenience, distress, or damage to standing or reputation	Low	Mod	Mod	High
Financial loss or organization liability	Low	Mod	Mod	High
Harm to organization programs or interests	None	Low	Mod	High
Unauthorized release of sensitive information	None	Low	Mod	High
Personal safety	None	None	Low	Mod/ High
Civil or criminal violations	None	Low	Mod	High

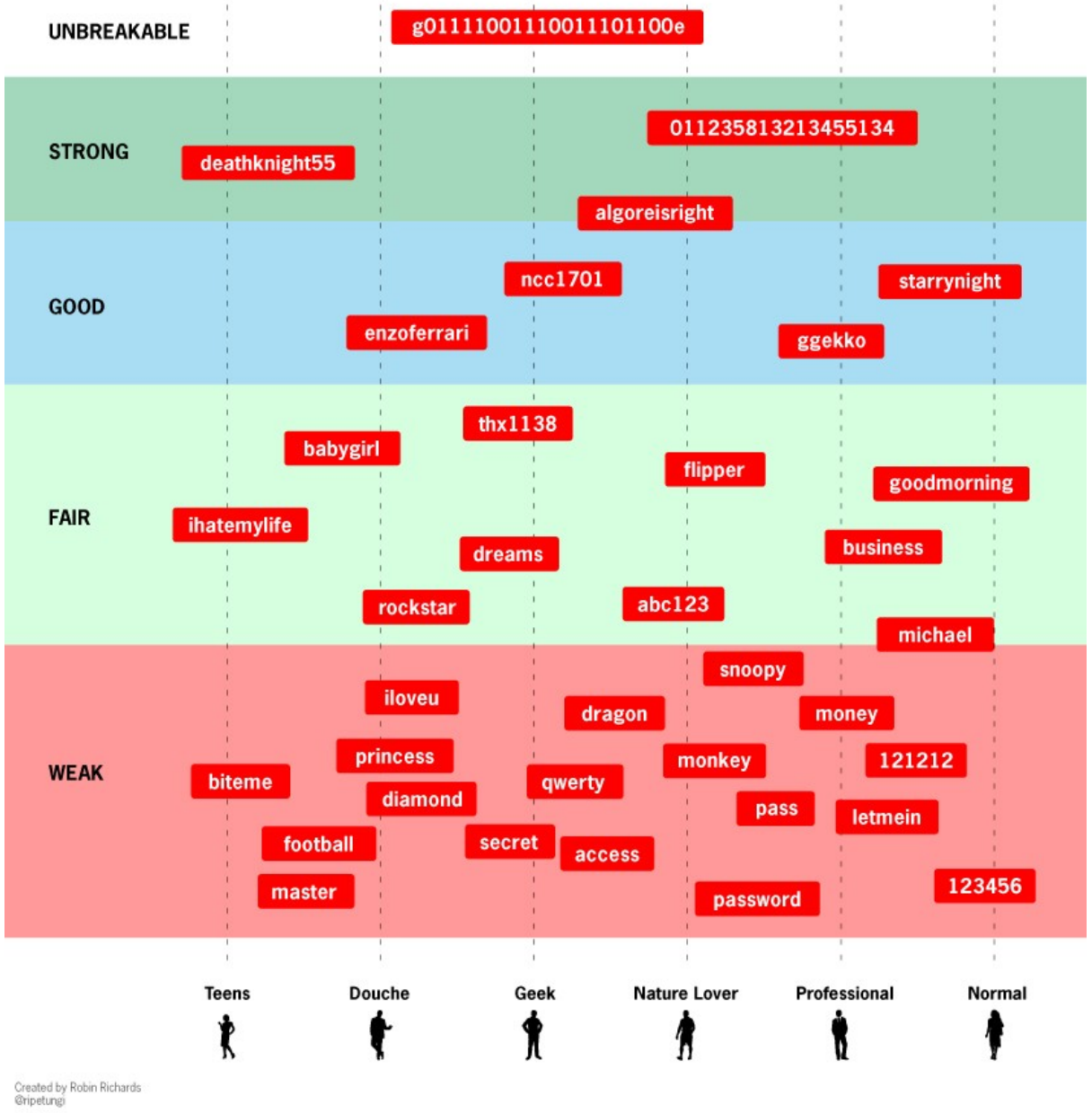
Password-based Authentication

Widely used line of defense against intruders

- User provides name/login and password
- System compares password with the one stored for that specified login

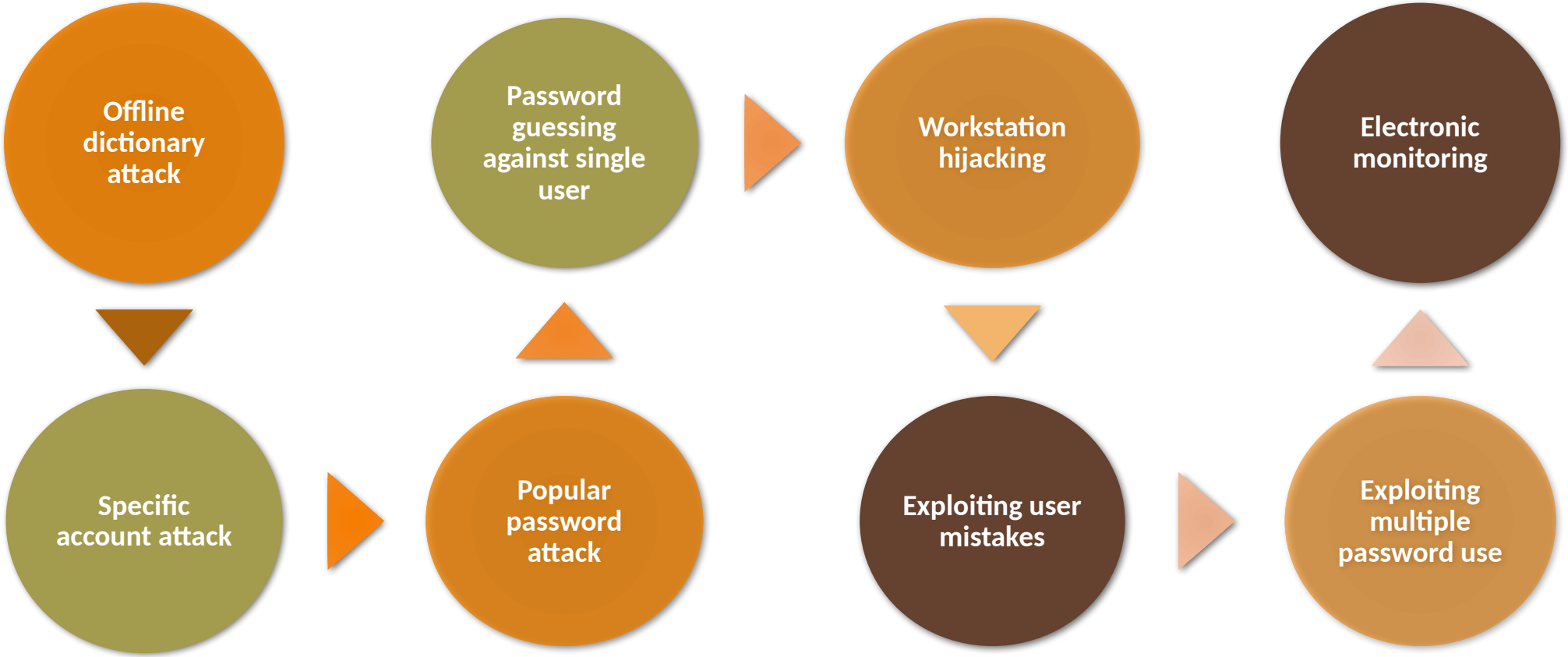
The user ID:

- Determines that the user is authorized to access the system
- Determines the user's privileges
- Is used in discretionary access control



Some Examples of Passwords

Password Vulnerability



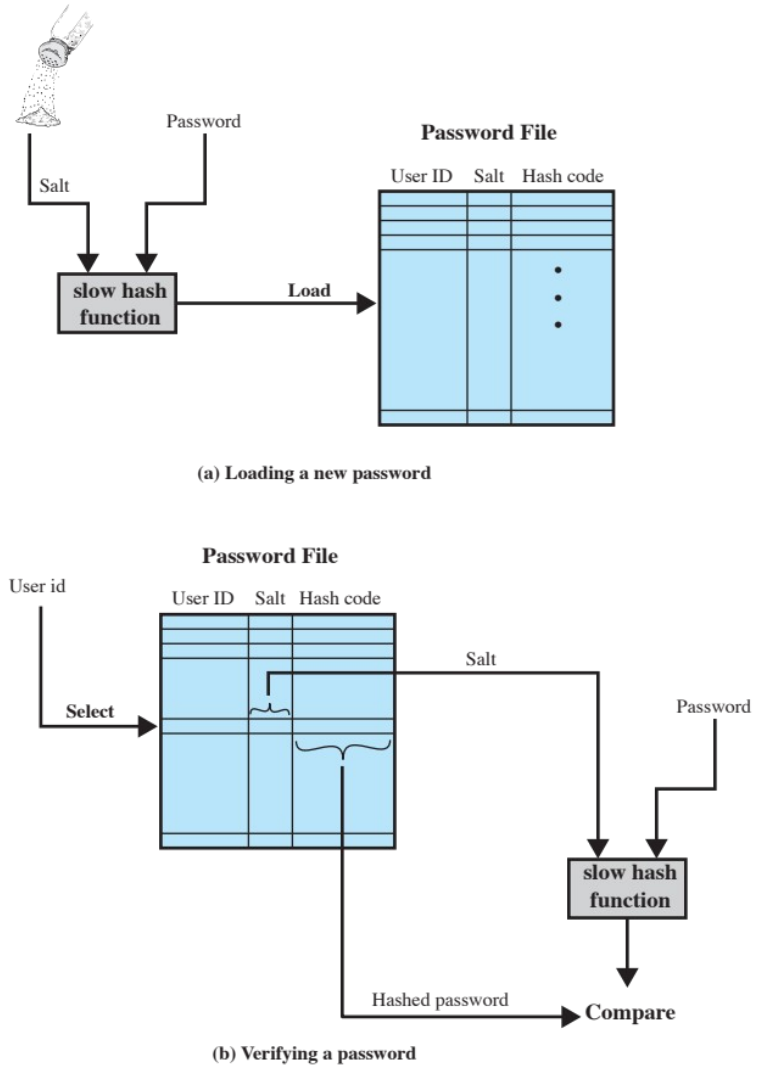


Figure 3.3 UNIX Password Scheme

Password files in Unix

Password Guessing

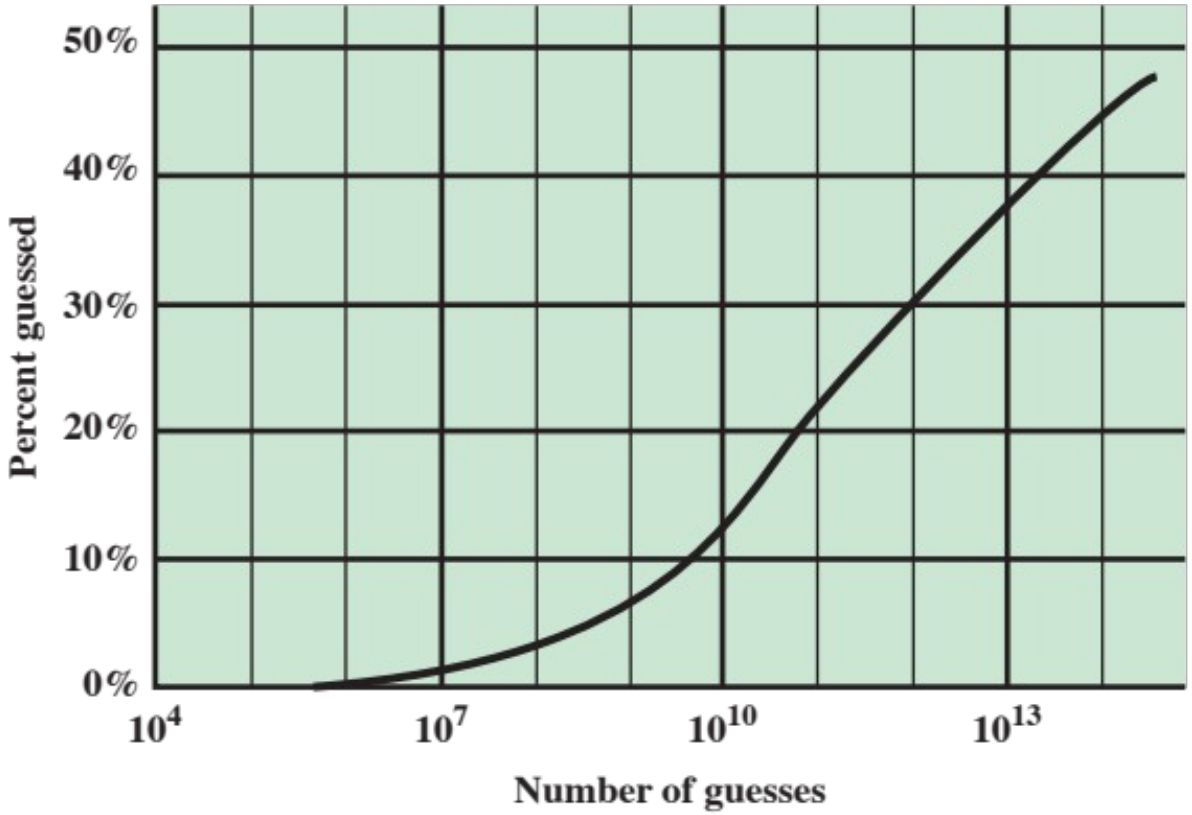
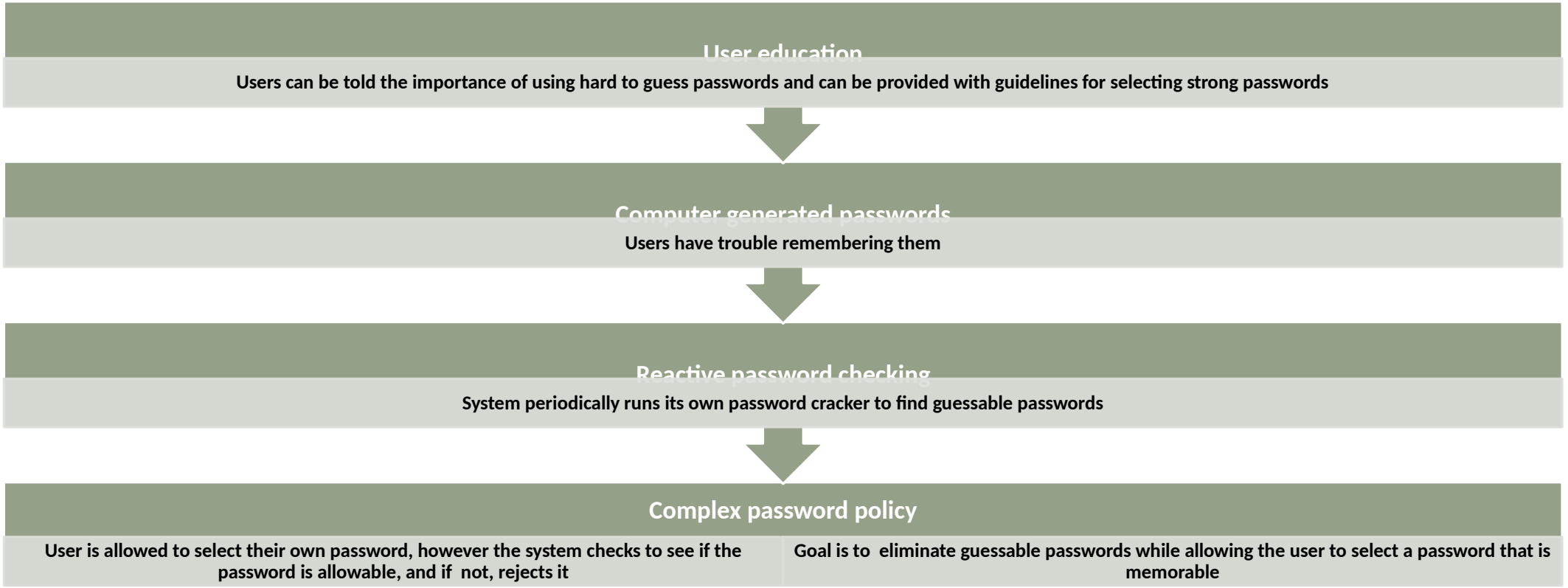


Figure 3.4 The Percentage of Passwords Guessed After a Given Number of Guesses

Password Selection Strategies



Biometric Authentication

Attempts to authenticate an individual based on unique physical characteristics

Based on pattern recognition

Is technically complex and expensive when compared to passwords and tokens

Physical characteristics used include:

- Facial characteristics
- Fingerprints
- Hand geometry
- Retinal pattern
- Iris
- Signature
- Voice

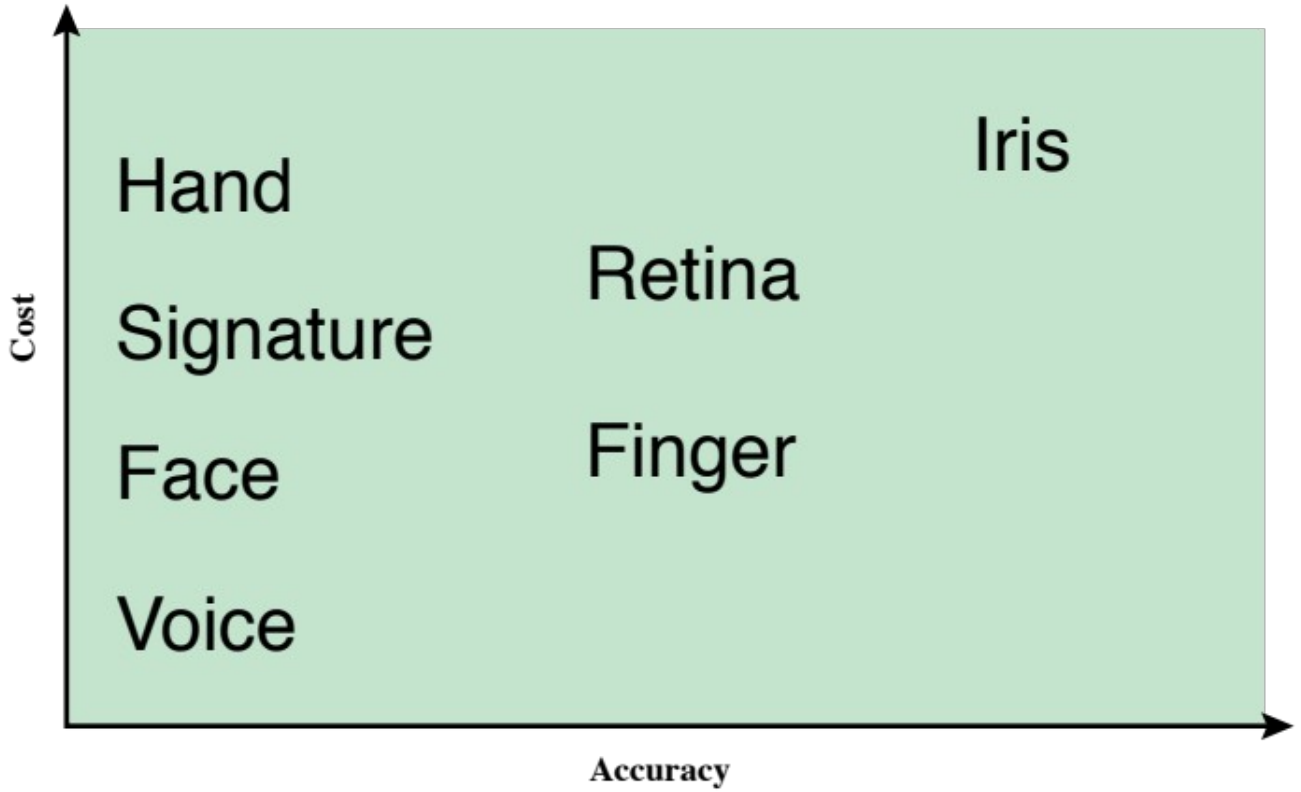


Figure 3.8 Cost Versus Accuracy of Various Biometric Characteristics in User Authentication Schemes.

Biometrics - Trade Offs

Security Issues

- **Eavesdropping**
- **Adversary attempts to learn the password by some sort of attack that involves the physical proximity of user and adversary**
- **Host Attacks**
- **Directed at the user file at the host where passwords, token passcodes, or biometric templates are stored**
- **Replay**
- **Adversary repeats a previously captured user response**
- **Client Attacks**
- **Adversary attempts to achieve user authentication without access to the remote host or the intervening communications path**
- **Trojan Horse**
  An application or physical device masquerades as an authentic application or device for the purpose of capturing a user password, passcode, or biometric
- **Denial-of-Service**
- **Attempts to disable a user authentication service by flooding the service with numerous authentication attempts**

User Authentication Attacks

Attacks	Authenticators	Examples	Typical defenses
Client attack	Password	Guessing, exhaustive search	Large entropy; limited attempts
	Token	Exhaustive search	Large entropy; limited attempts, theft of object requires presence
	Biometric	False match	Large entropy; limited attempts
Host attack	Password	Plaintext theft, dictionary/exhaustive search	Hashing; large entropy; protection of password database
	Token	Passcode theft	Same as password; 1-time passcode
	Biometric	Template theft	Capture device authentication; challenge response
Eavesdropping, theft, and copying	Password	"Shoulder surfing"	User diligence to keep secret; administrator diligence to quickly revoke compromised passwords; multifactor authentication
	Token	Theft, counterfeiting hardware	Multifactor authentication; tamper resistant/evident token
	Biometric	Copying (spoofing) biometric	Copy detection at capture device and capture device authentication
Replay	Password	Replay stolen password response	Challenge-response protocol
	Token	Replay stolen passcode response	Challenge-response protocol; 1-time passcode
	Biometric	Replay stolen biometric template response	Copy detection at capture device and capture device authentication via challenge-response protocol
Trojan horse	Password, token, biometric	Installation of rogue client or capture device	Authentication of client or capture device within trusted security perimeter