

KDC CA and Lightweight Cryptography

Hemant Ghayvat

Department of Computer Science and Media Technology

hemant.ghayvat@lnu.se



SYMMETRIC-KEY DISTRIBUTION

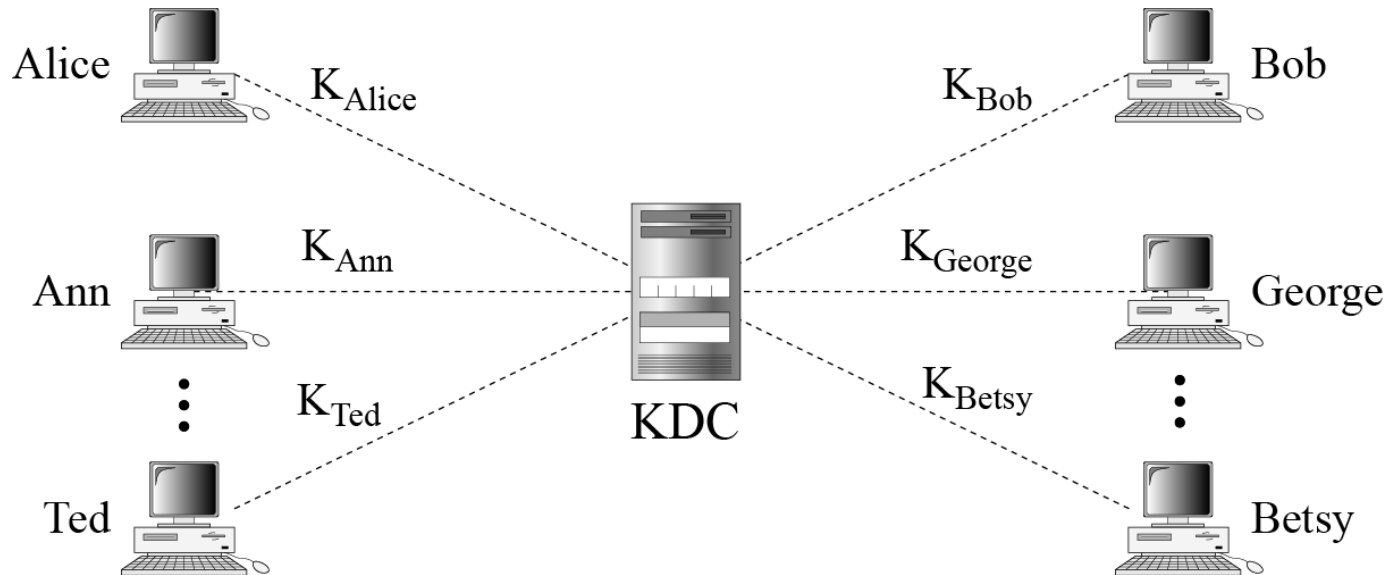
Symmetric-key cryptography is more efficient than asymmetric-key cryptography for enciphering large messages. Symmetric-key cryptography, however, needs a shared secret key between two parties. The distribution of keys is another problem.



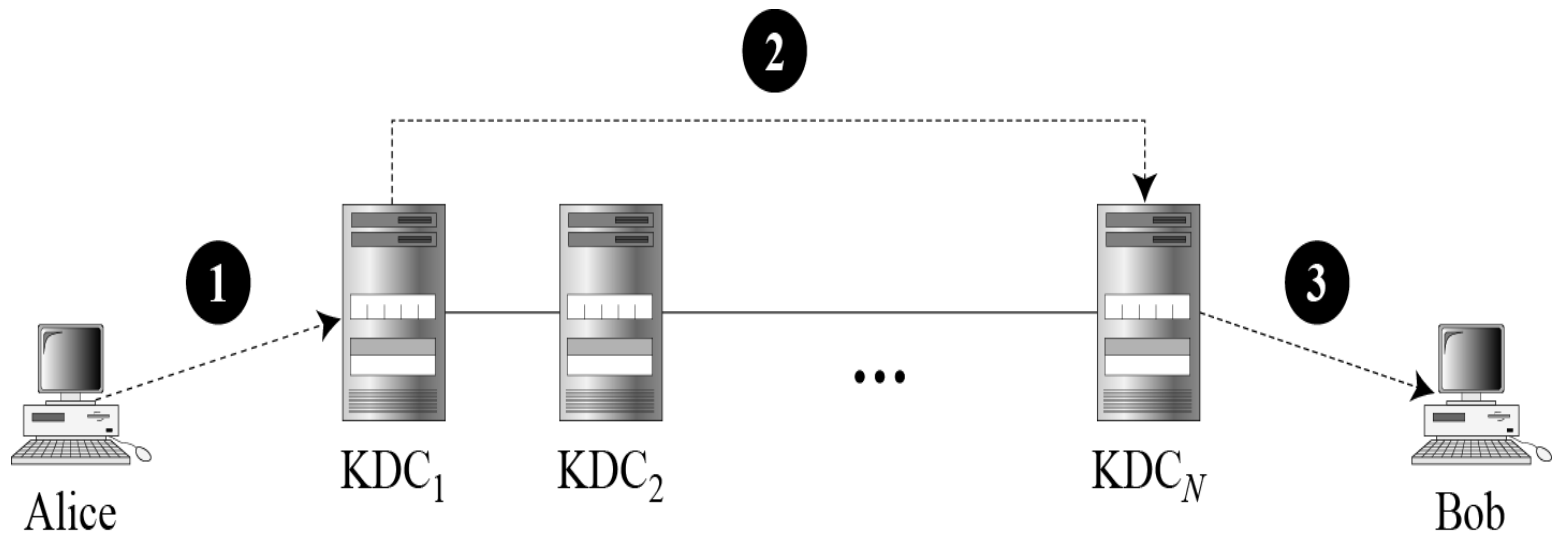


Key-Distribution Center: KDC

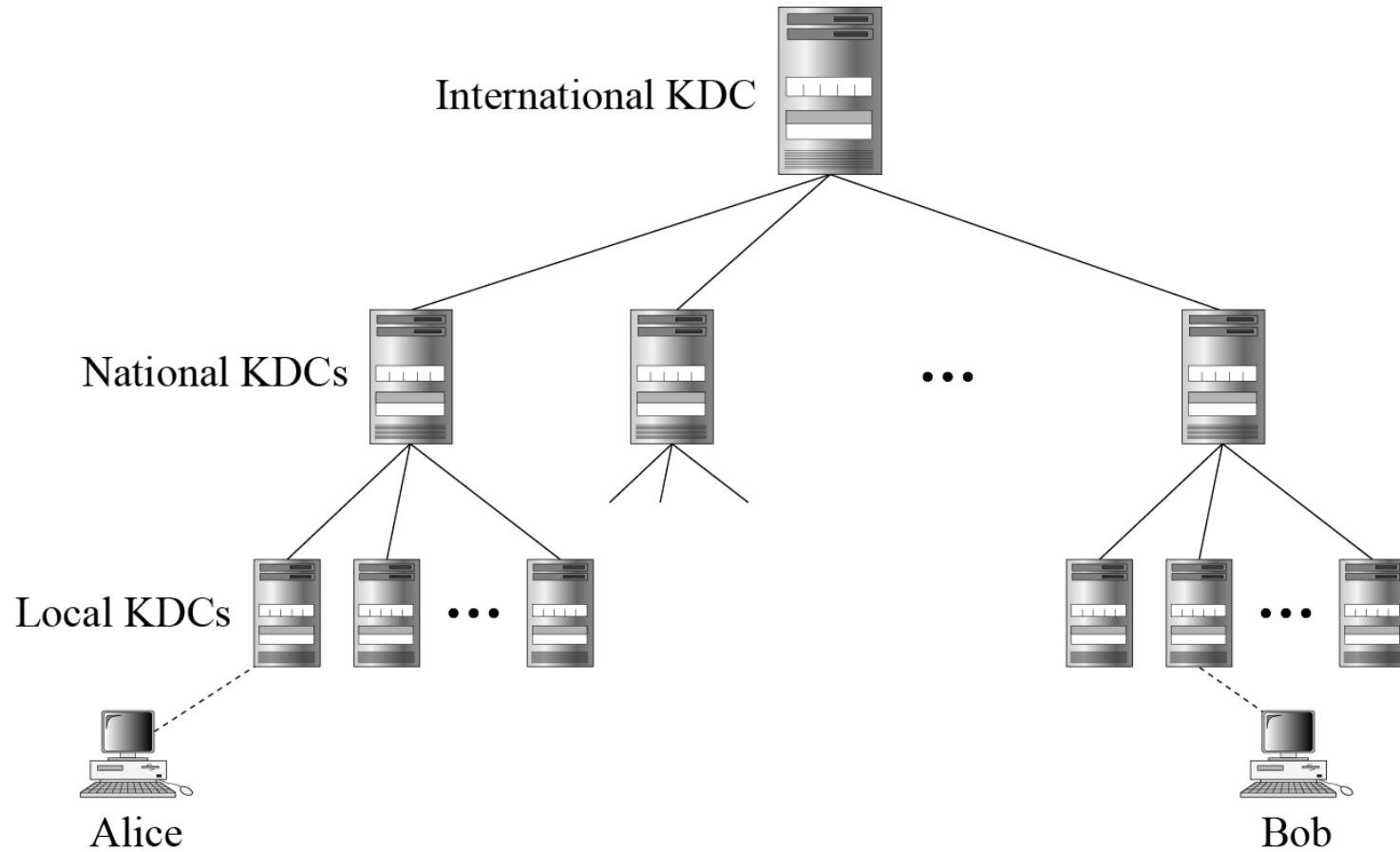
Key-distribution center (KDC)



Flat Multiple KDCs.



Hierarchical Multiple KDCs





Session Keys


A KDC creates a secret key for each member. This secret key can be used only between the member and the KDC, not between two members.

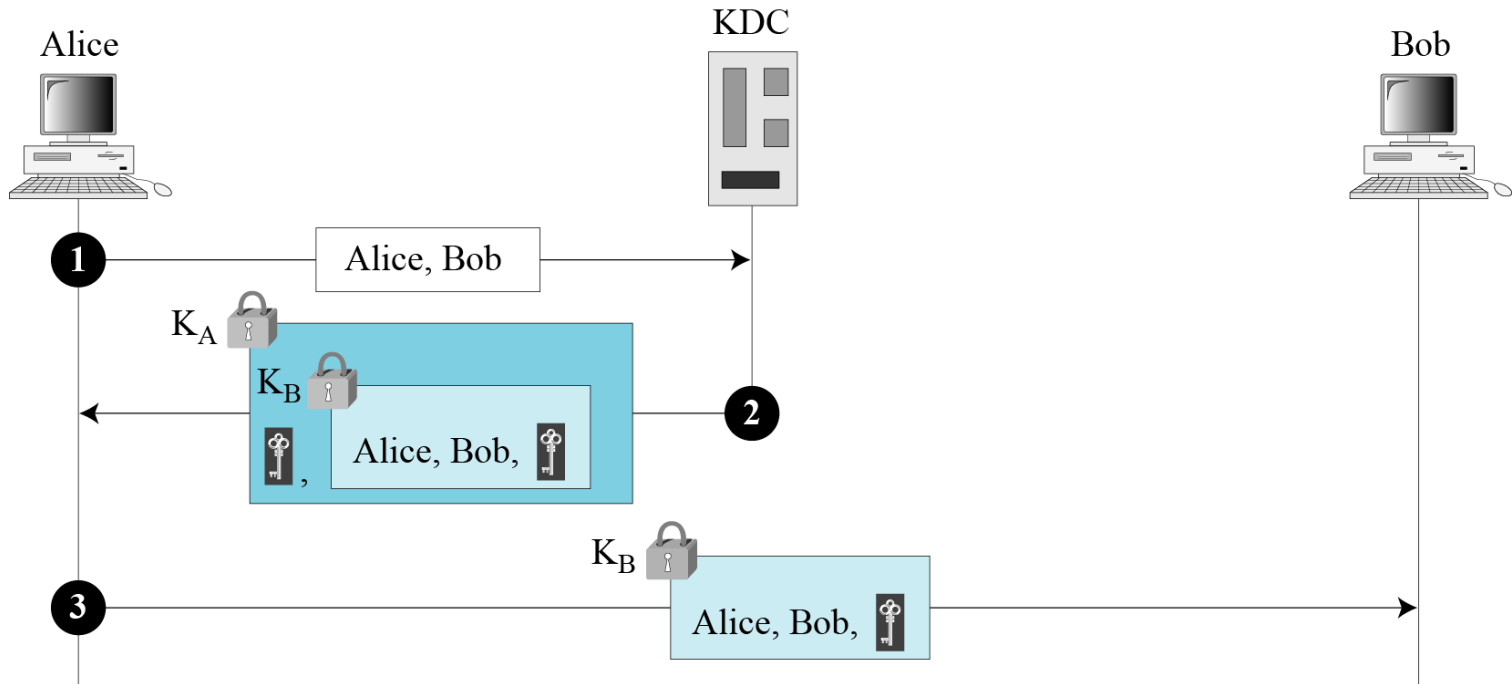
A session symmetric key between two parties is used only once



A Simple Protocol Using a KDC

K_A  Encrypted with Alice-KDC secret key  Session key between Alice and Bob

K_B  Encrypted with Bob-KDC secret key KDC: Key-distribution center





FINALLY

3% MILK

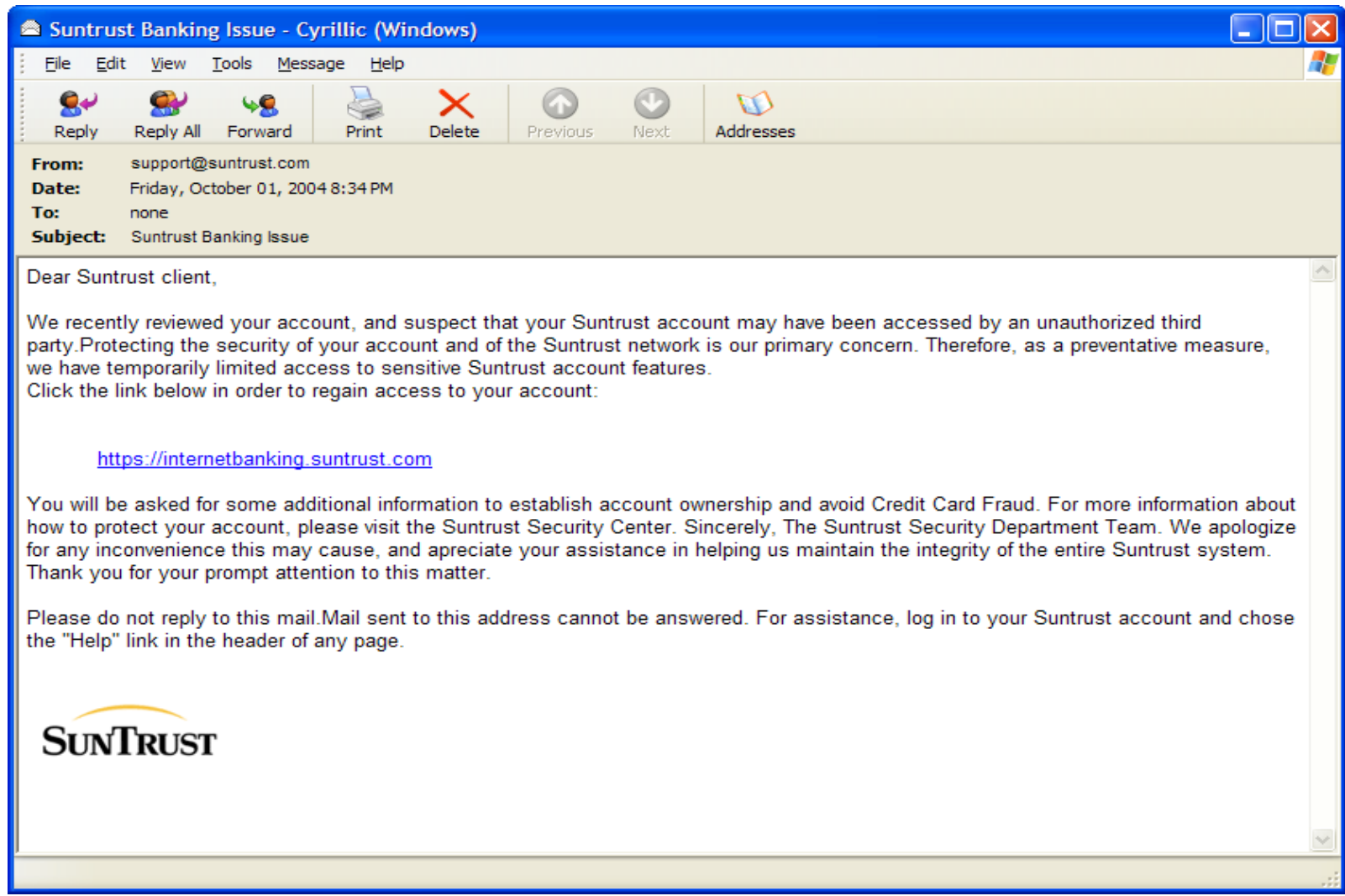
KERBEROS

Kerberos is an authentication protocol, and at the same time a KDC, that has become very popular. Several systems, including Windows 2000, use Kerberos. Originally designed at MIT, it has gone through several versions.

The Kerberos protocol works by using a trusted **third-party authentication server**, which issues tickets that can be used by clients to authenticate to services.



Motivation



Application of Karberos

1. Single sign-on (SSO) systems: Kerberos can be used as a foundation for SSO systems that allow users to authenticate once and then access multiple applications without needing to enter their credentials again.
2. Remote access: Kerberos can be used to provide secure remote access to systems and resources, such as VPNs, remote desktops, and file shares.
3. Online banking: Some banks use Kerberos to provide strong authentication for their online banking systems, helping to protect against fraud and unauthorized access.
4. Cloud computing: Kerberos can be used to provide secure authentication and authorization in cloud computing environments, such as for accessing cloud-based applications or services.
5. Messaging and collaboration: Kerberos can be used to secure messaging and collaboration systems, such as email, chat, and groupware, by providing strong authentication and authorization for users and services



Kerberos

A practical authentication service

Kerberos: three headed dog in Greek mythology,
the guardian of the entrance of Hades

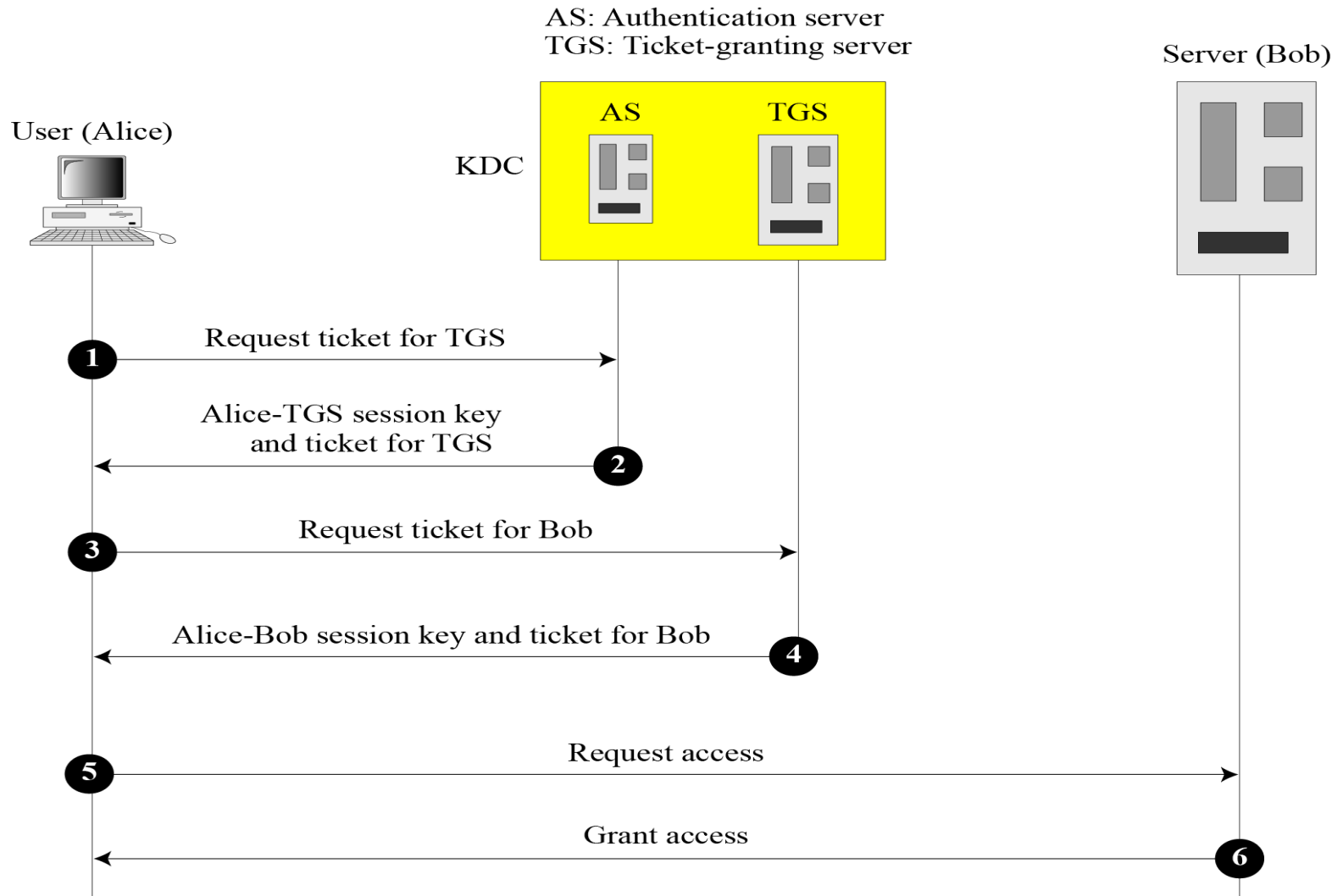
Those three heads in security: AAA
(Authentication, Accounting, Audit)

However, in Kerberos the last two heads never
implemented

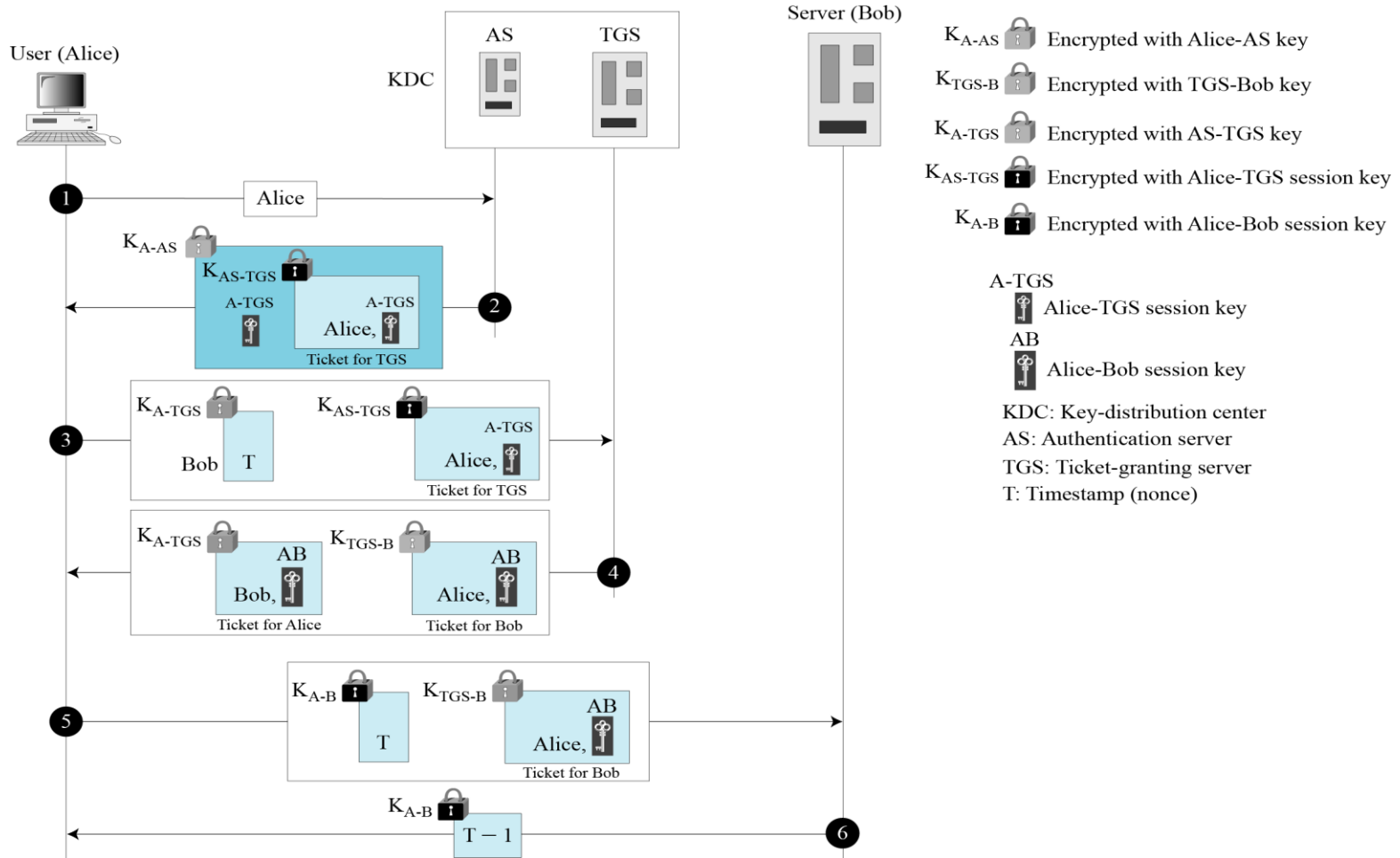




Kerberos servers



Kerberos example



A Simple Authentication Dialogue without TGT

where

C = client Alice , AS = authentication server , V = Bob server

ID_C = identifier of user C , ID_V = identifier of V Bob

P_C = password of user on C to access Bob, AD_C = network address of C Alice

K_V = secret encryption key shared by AS and V Bob

the user logs on to a workstation and requests access to server V .

The client module C in the Alice user's workstation enters the user's password and then sends a message to the AS that includes the Alice user's ID, the Bob server's ID, and the user's password.

The AS checks its database to see if the user has supplied the proper password for this user ID and whether this user is permitted access to server V Bob.



A Simple Authentication Dialogue

- an authentication server (AS)
 - that knows the passwords of all users and stores these in a centralized database.
 - shares a unique secret key with each server
 - These keys have been distributed physically or in some other secure manner.

(1) C \longrightarrow AS: $ID_C || P_C || ID_V$

(2) AS \longrightarrow C: Ticket

(3) C \longrightarrow V: $ID_C || \text{Ticket}$

– Ticket = $E(K_v, [ID_C || AD_C || ID_V])$



A Simple Authentication Dialogue

- The AS creates a ticket that contains the user's ID Alice and network address (where Alice and Bob are located) and the server's ID Bob.
- This ticket is encrypted using the secret key shared by the AS and this server Bob.
- This ticket is then sent back to C Alice.
- C Alice sends a message to V Bob containing C's ID and the ticket.
- Bob V decrypts the ticket and verifies that the user ID Alice in the ticket is the same as the unencrypted user ID in the message.
- There are two ID's one which verified by the AS encrypted with secret key and other which is coming with plaintext of Alice



A More Secure Authentication Dialogue

Two problems in previous dialogue :

- a. each ticket can be used only once.
- b. plaintext transmission of the password [message (1)].

a scheme for avoiding plaintext passwords and a new server, known as the ticket-granting server (TGS).



A More Secure Authentication Dialogue

Once per user logon session:

(1) $C \longrightarrow AS: ID_C || ID_{tgs}$

(2) $AS \longrightarrow C: E(K_c, Ticket_{tgs})$

Once per type of service:

(3) $C \longrightarrow TGS: ID_C || ID_v || Ticket_{tgs}$

(4) $TGS \longrightarrow C: Ticket_v$

Once per service session:

(5) $C \longrightarrow V: ID_C || Ticket_v$

$Ticket_{tgs} = E(K_{tgs}, [ID_C || AD_C || ID_{tgs} || TS1 || Lifetime1])$

$Ticket_v = E(K_v, [ID_C || AD_C || ID_v || TS2 || Lifetime2])$

TS1: Allows AS to verify that client's clock is synchronized with that of AS

TS2: Informs TGS of time this ticket was issued



A More Secure Authentication Dialogue

1. The client requests a ticket-granting ticket on behalf of the user by sending its user's ID and password to the AS, together with the TGS ID, indicating a request to use the TGS service.
2. The AS responds with a ticket that is encrypted with a key that is derived from the user's password. When this response arrives at the client, the client prompts the user for his or her password, generates the key, and attempts to decrypt the incoming message. If the correct password is supplied, the ticket is successfully recovered.

Now that the client has a ticket-granting ticket, access to any server can be obtained with steps 3 and 4:

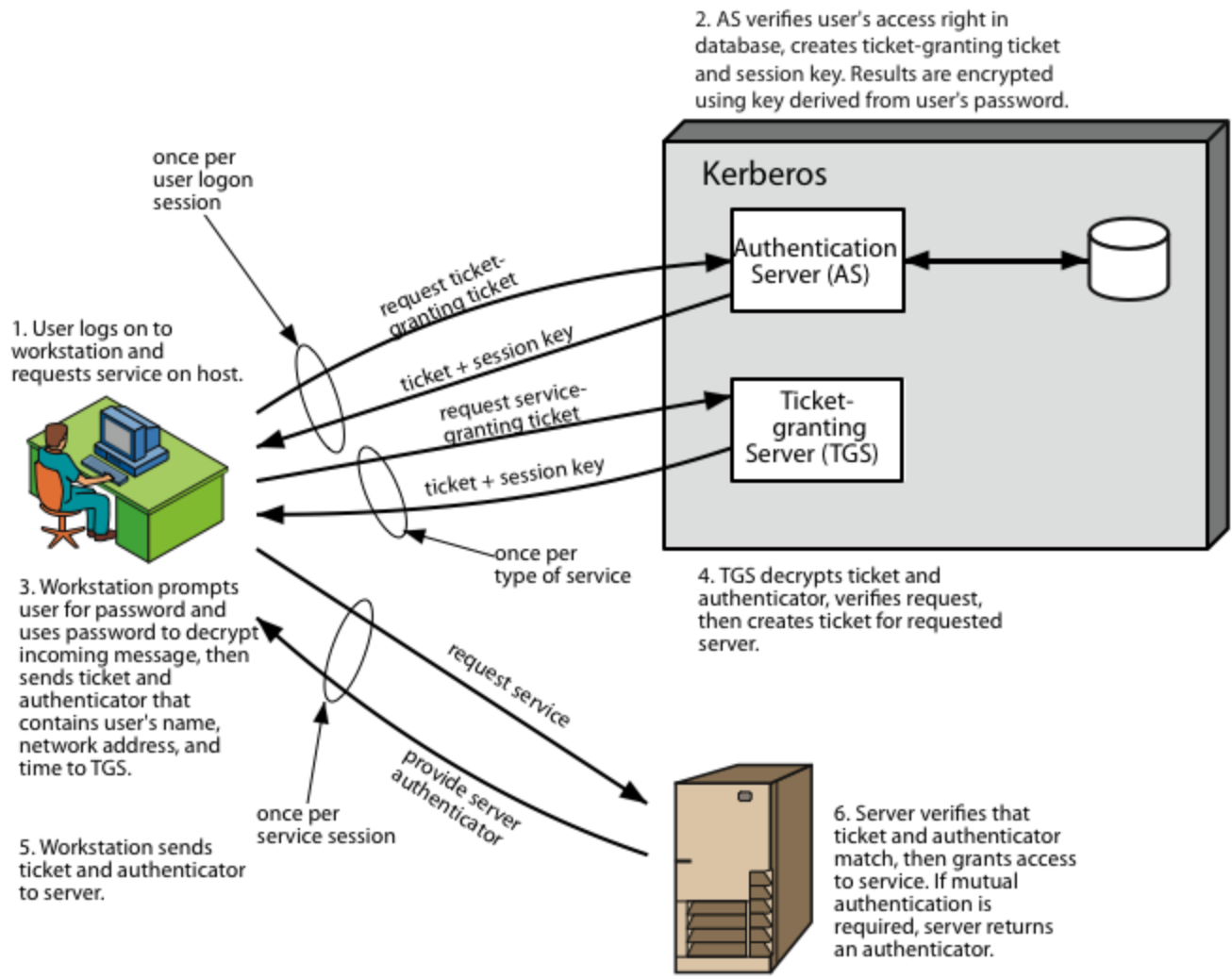


A More Secure Authentication Dialogue

3. The client requests a service-granting ticket on behalf of the user. For this purpose, the client transmits a message to the TGS containing the user's ID, the ID of the desired service, and the ticket-granting ticket.
4. The TGS decrypts the incoming ticket and verifies the success of the decryption by the presence of its ID. It checks to make sure that the lifetime has not expired. Then it compares the user ID and network address with the incoming information to authenticate the user. If the user is permitted access to the server V , the TGS issues a ticket to grant access to the requested service.
5. The client requests access to a service on behalf of the user. For this purpose, the client transmits a message to the server containing the user's ID and the service-granting ticket. The server authenticates by using the contents of the ticket.



Kerberos 4 Overview



The Version 4 Authentication Dialogue

Problems:

remain in previous dialogue the lifetime associated with the ticket-granting ticket deny the true service to the user.

Solution:

1. A network service (the TGS or an application service) must be able to prove that the person using a ticket is the same person to whom that ticket was issued.
2. servers authenticate themselves to users (Authenticator_s)



Kerberos 4 Overview

a basic third-party authentication scheme

have an Authentication Server (AS)

- users initially negotiate with AS to identify self
- AS provides a non-corruptible authentication credential (ticket granting ticket TGT)

have a Ticket Granting server (TGS)

- users subsequently request access to other services from TGS on basis of users TGT



The Version 4 Authentication Dialogue

- (1) C \longrightarrow AS $ID_c || ID_{tgs} || TS_1$
(2) AS \longrightarrow C $E(K_c, [K_{c,tgs} || ID_{tgs} || TS_2 || Lifetime_2 || Ticket_{tgs}])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} || ID_c || AD_c || ID_{tgs} || TS_2 || Lifetime_2])$$

(a) Authentication Service Exchange to obtain ticket-granting ticket

- (3) C \longrightarrow TGS $ID_v || Ticket_{tgs} || Authenticator_c$
(4) TGS \longrightarrow C $E(K_{c,tgs}, [K_{c,v} || ID_v || TS_4 || Ticket_v])$

$$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} || ID_c || AD_c || ID_{tgs} || TS_2 || Lifetime_2])$$

$$Ticket_v = E(K_v, [K_{c,v} || ID_c || AD_c || ID_v || TS_4 || Lifetime_4])$$

$$Authenticator_c = E(K_{c,tgs}, [ID_c || AD_c || TS_3])$$

(b) Ticket-Granting Service Exchange to obtain service-granting ticket

- (5) C \longrightarrow V $Ticket_v || Authenticator_c$
(6) V \longrightarrow C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)

$$Ticket_v = E(K_v, [K_{c,v} || ID_c || AD_c || ID_v || TS_4 || Lifetime_4])$$

$$Authenticator_c = E(K_{c,v}, [ID_c || AD_c || TS_5])$$

(c) Client/Server Authentication Exchange to obtain service





Kerberos Version 5

developed in mid 1990's

specified as Internet standard RFC 1510

provides improvements over v4

- addresses environmental shortcomings
 - encryption alg, network protocol, byte order, ticket lifetime, authentication forwarding, interrealm auth
- and technical deficiencies
 - double encryption, non-std mode of use, session keys, password attacks



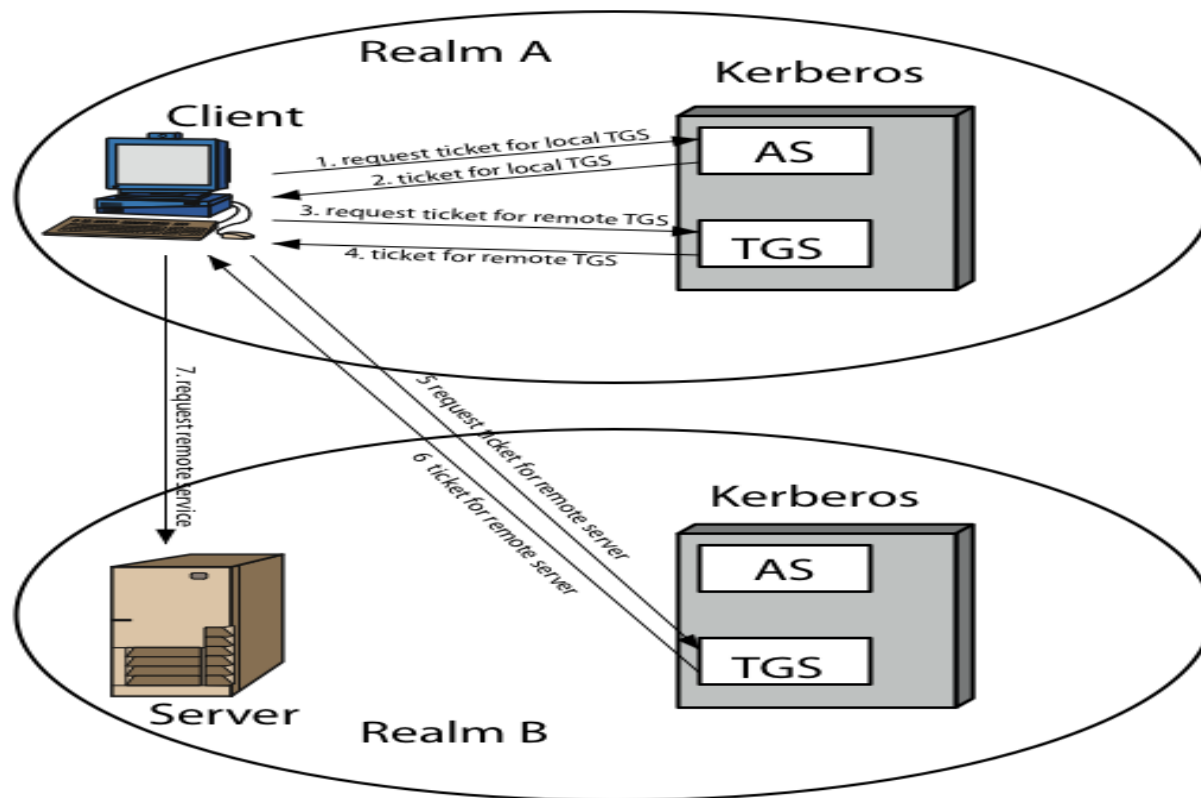
Kerberos Version 5

The minor differences between version 4 and version 5 are briefly listed below:

- 1) Version 5 has a longer ticket lifetime.
- 2) Version 5 allows tickets to be renewed.
- 3) Version 5 can accept any symmetric-key algorithm.
- 4) Version 5 uses a different protocol for describing data types.
- 5) Version 5 has more overhead than version 4.



Kerberos Realms: Realm names are used for network routing and authentication. They provide the identification required to forward authentication requests to the server that holds the user's credentials.



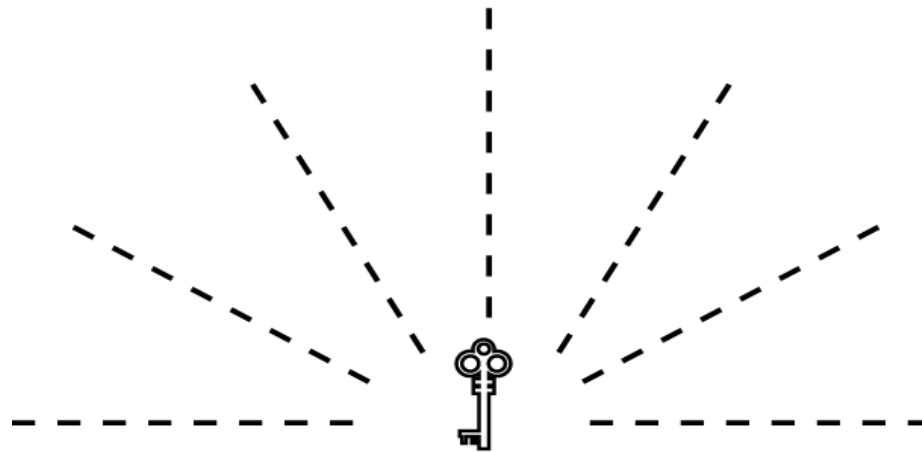
PUBLIC-KEY DISTRIBUTION

In asymmetric-key cryptography, people do not need to know a symmetric shared key; everyone shields a private key and advertises a public key.



Public Announcement

Announcing a public key



Public key

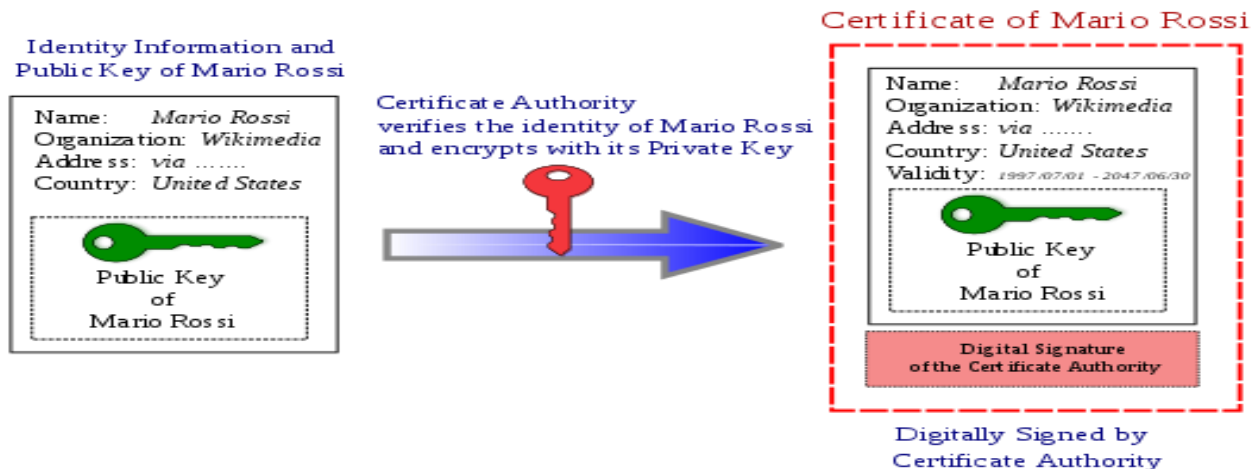


Bob



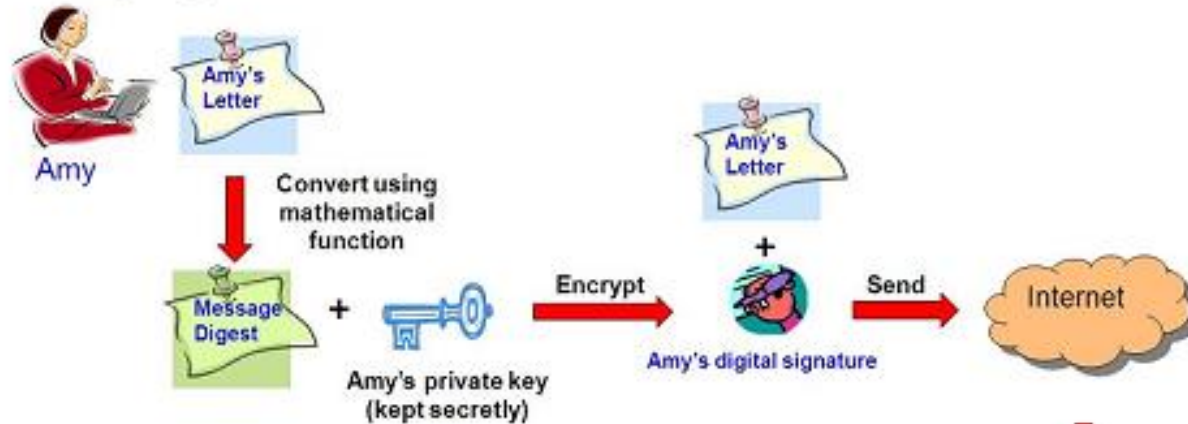
Certification Authority

- Certification authority generates the signature (using the sender's private key in the encryption process)
- This signature would be generated for the user public key and user identity

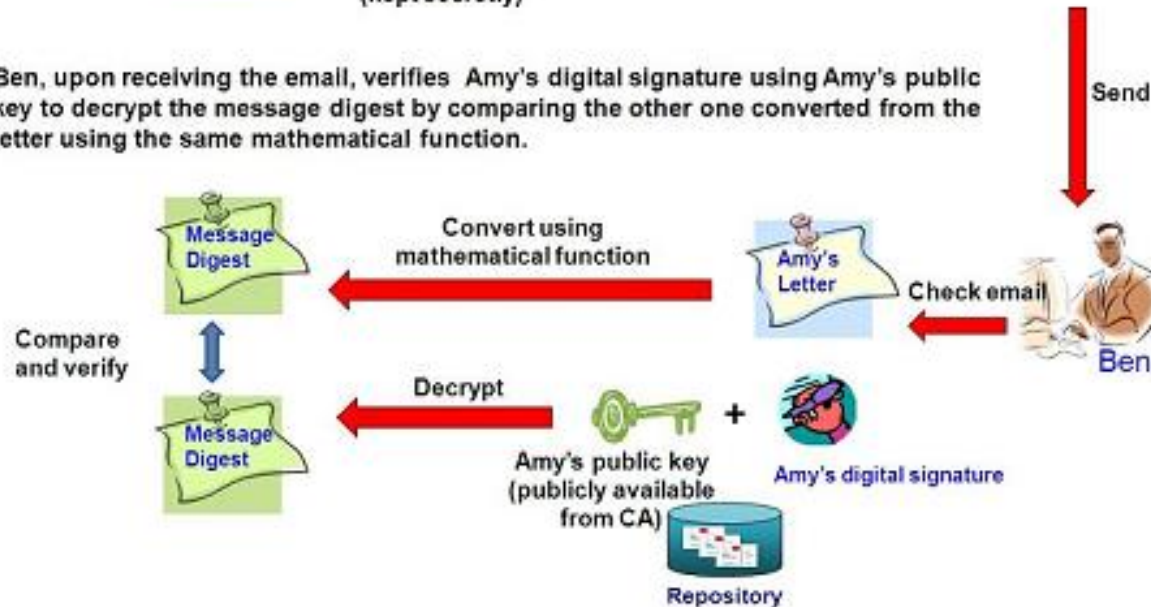


Digital Signature

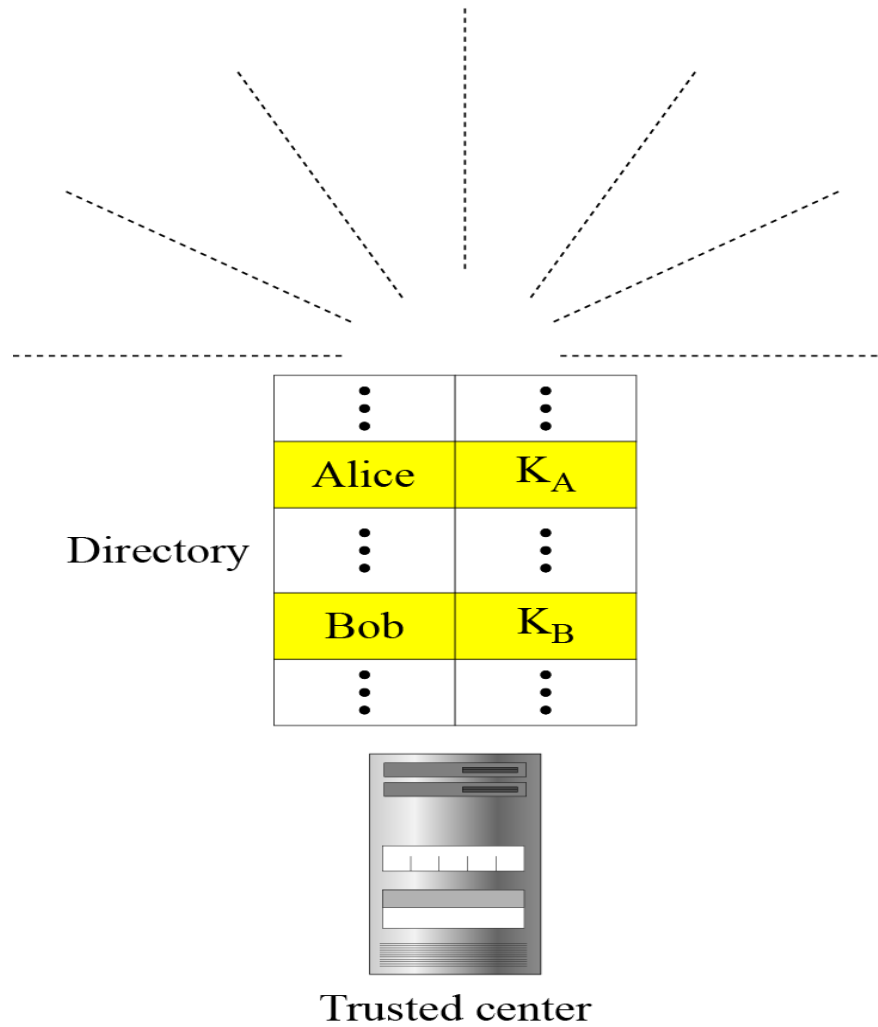
1. Amy converts her letter into a message digest by using a mathematical function. She then creates her digital signature by encrypting the message digest using her private key. Her letter, together with her digital signature are sent to Ben via email.



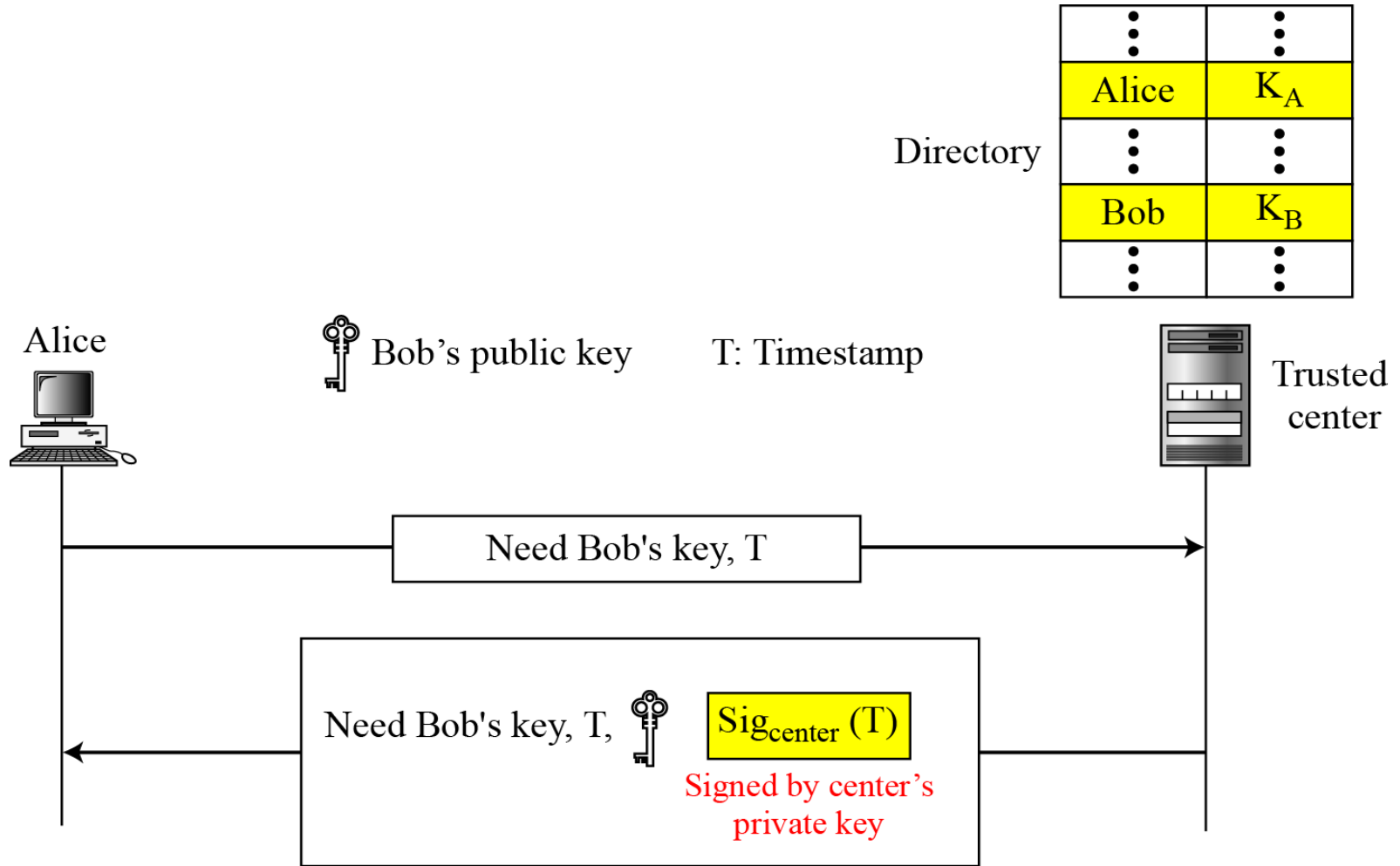
2. Ben, upon receiving the email, verifies Amy's digital signature using Amy's public key to decrypt the message digest by comparing the other one converted from the letter using the same mathematical function.



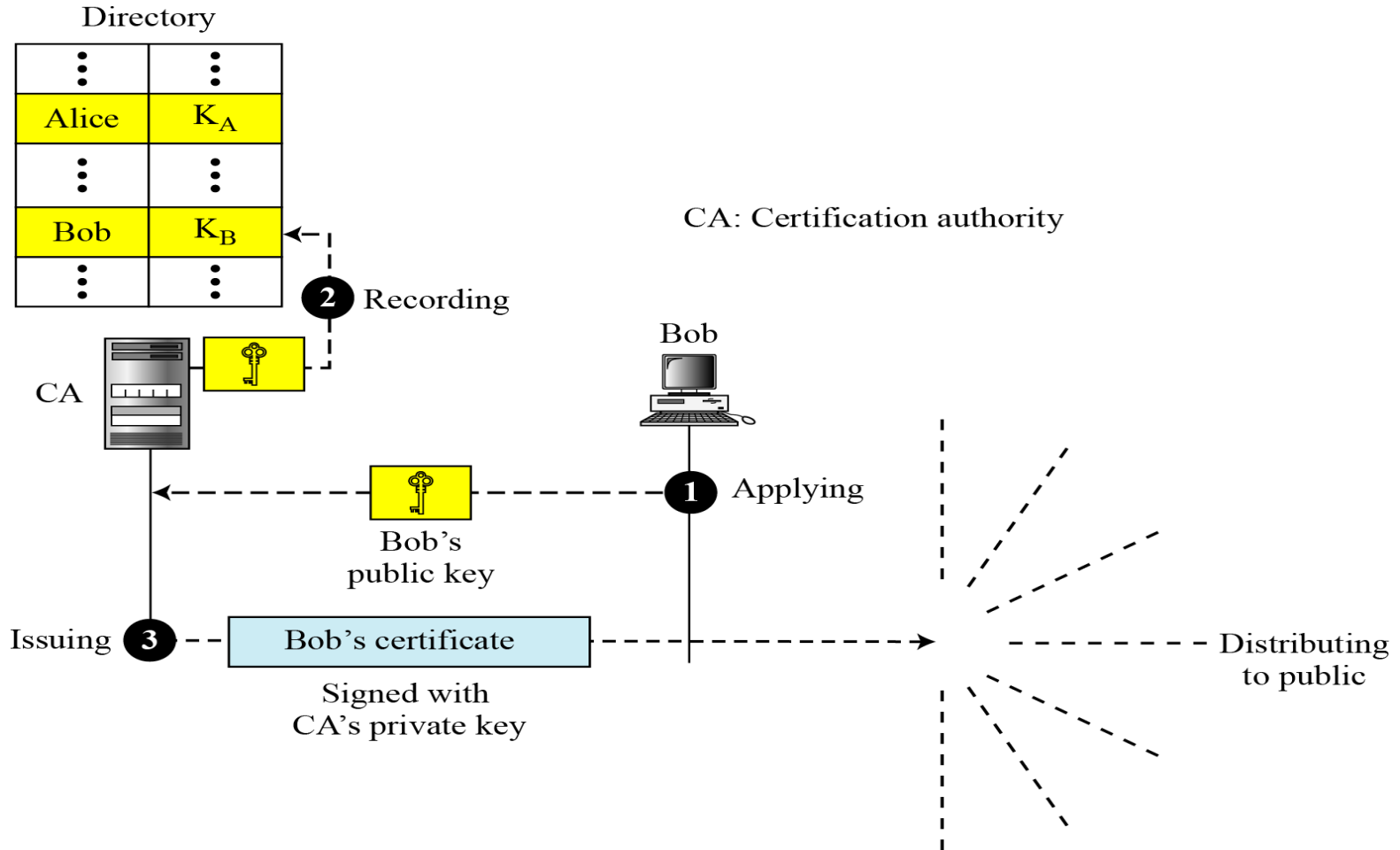
Trusted Center



Controlled Trusted Center



Certification Authority



X.509-format of a Certificate

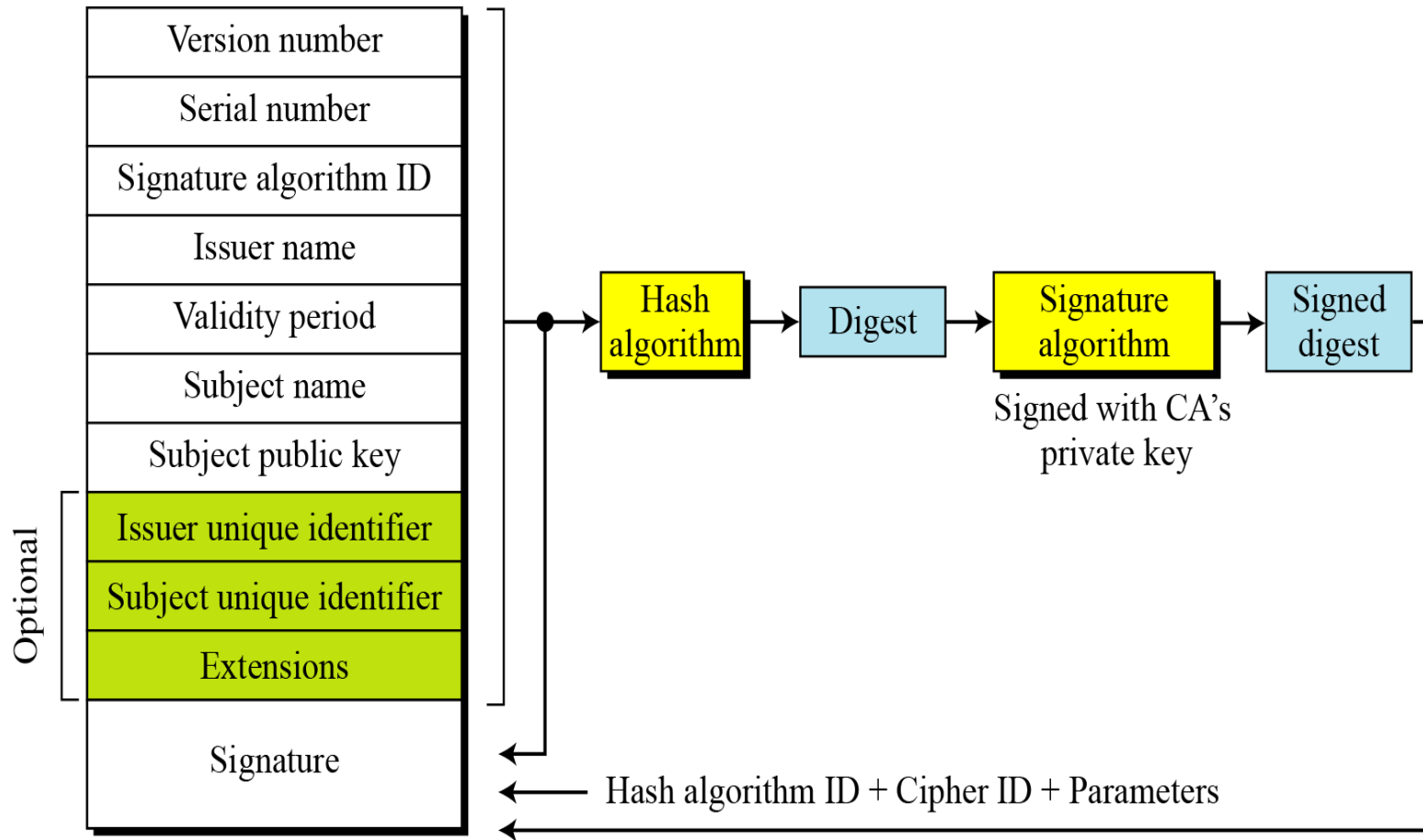


Figure 15.10 X.509 Public-Key Certificate Use

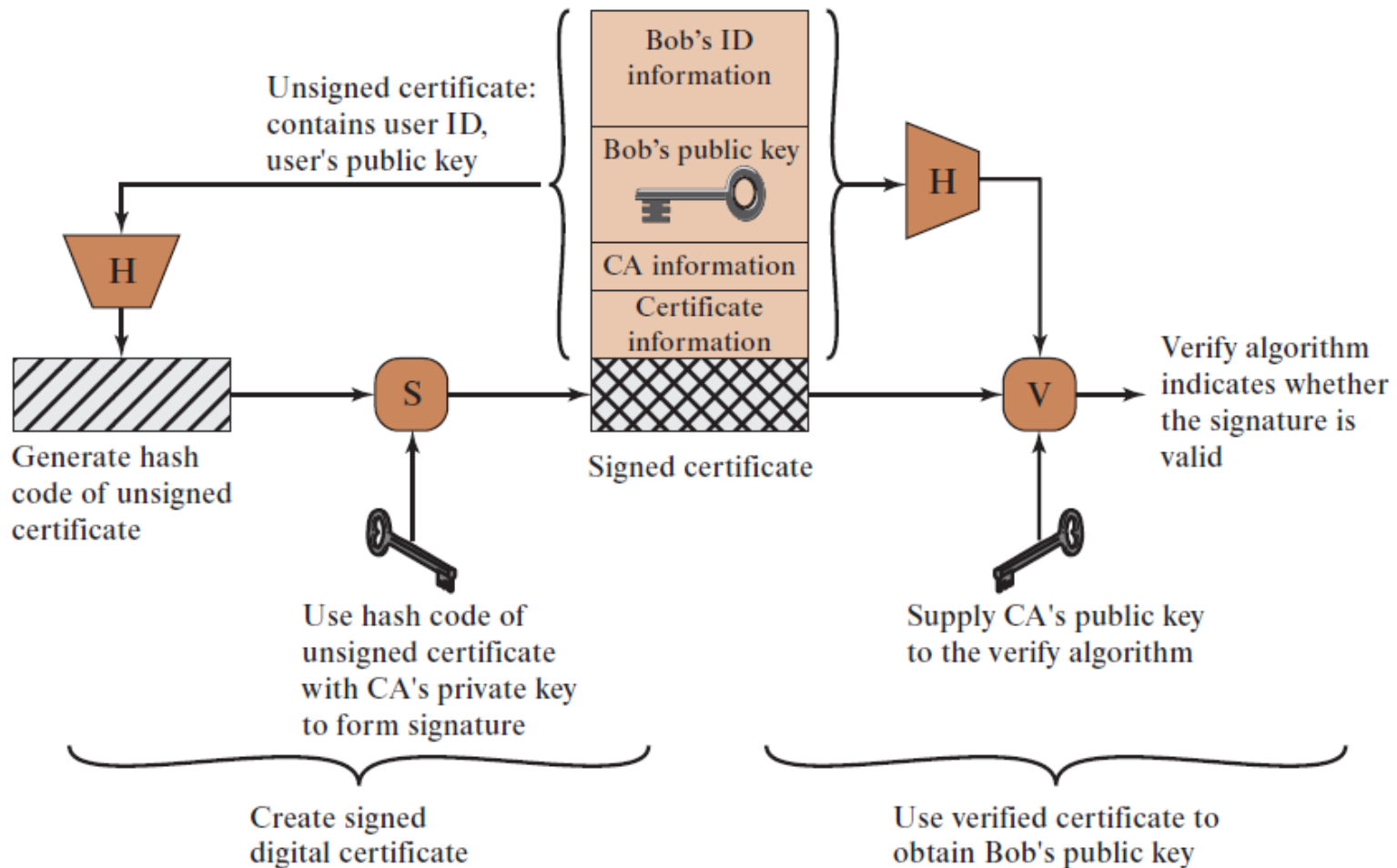
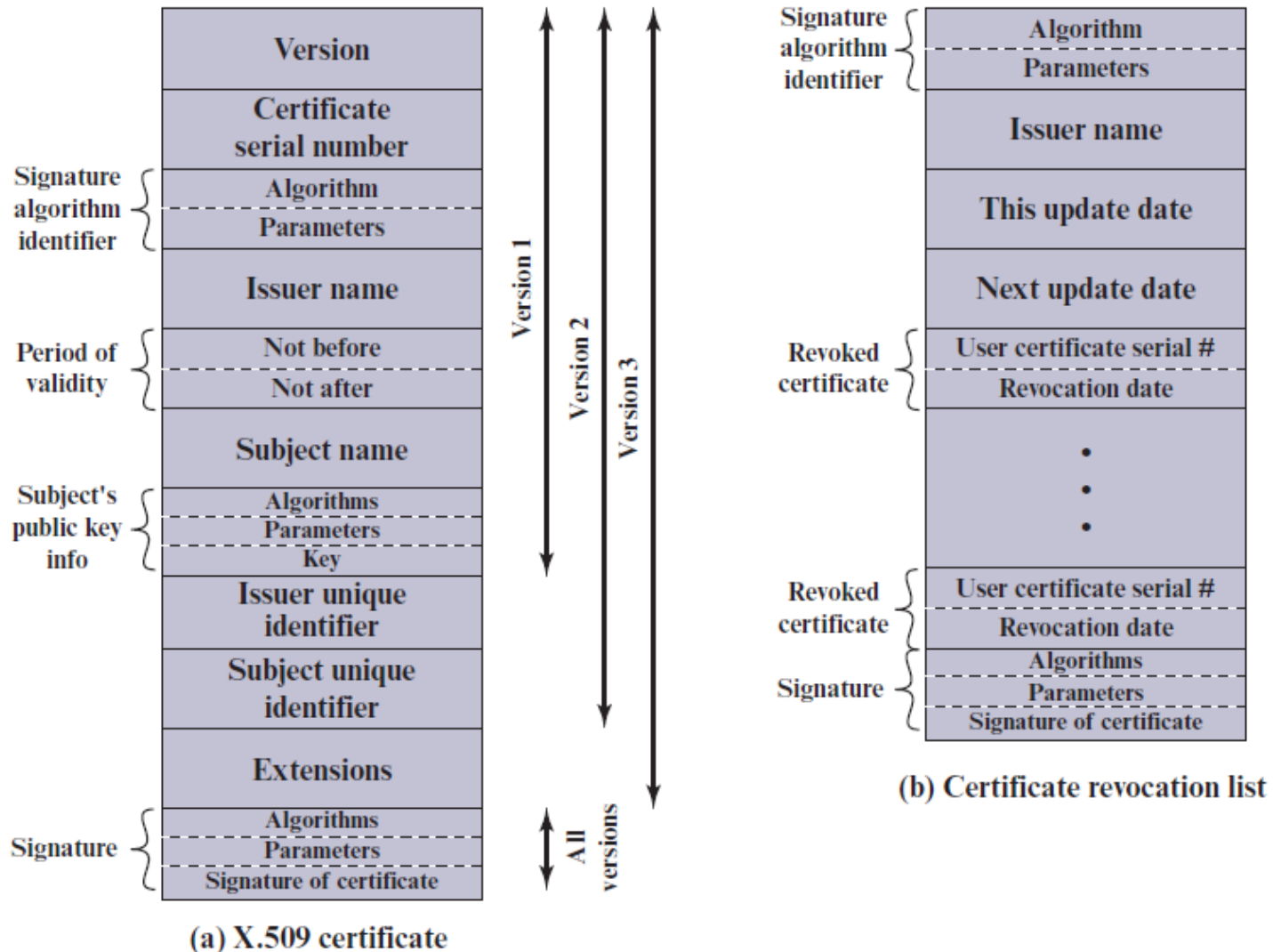


Figure 15.11 X.509 Formats



(a) X.509 certificate

(b) Certificate revocation list



Obtaining a Certificate

User certificates generated by a CA have the following characteristics:

- Any user with access to the public key of the CA can verify the user public key that was certified
- No party other than the certification authority can modify the certificate without this being detected

Because certificates are unforgeable, they can be placed in a directory without the need for the directory to make special efforts to protect them

In addition, a user can transmit his or her certificate directly to other users

Once B is in possession of A's certificate, B has confidence that messages it encrypts with A's public key will be secure from eavesdropping and that messages signed with A's private key are unforgeable



Certificate Revocation

Each certificate includes a period of validity

- Typically a new certificate is issued just before the expiration of the old one

It may be desirable on occasion to revoke a certificate before it expires, for one of the following reasons:

- The user's private key is assumed to be compromised
- The user is no longer certified by this CA
- The CA's certificate is assumed to be compromised

Each CA must maintain a list consisting of all revoked but not expired certificates issued by that CA

- These lists should be posted on the directory



CA Hierarchy

if both users share a common CA then they are assumed to know its
public key

otherwise CA's must form a hierarchy

use certificates linking members of hierarchy to validate other CA's

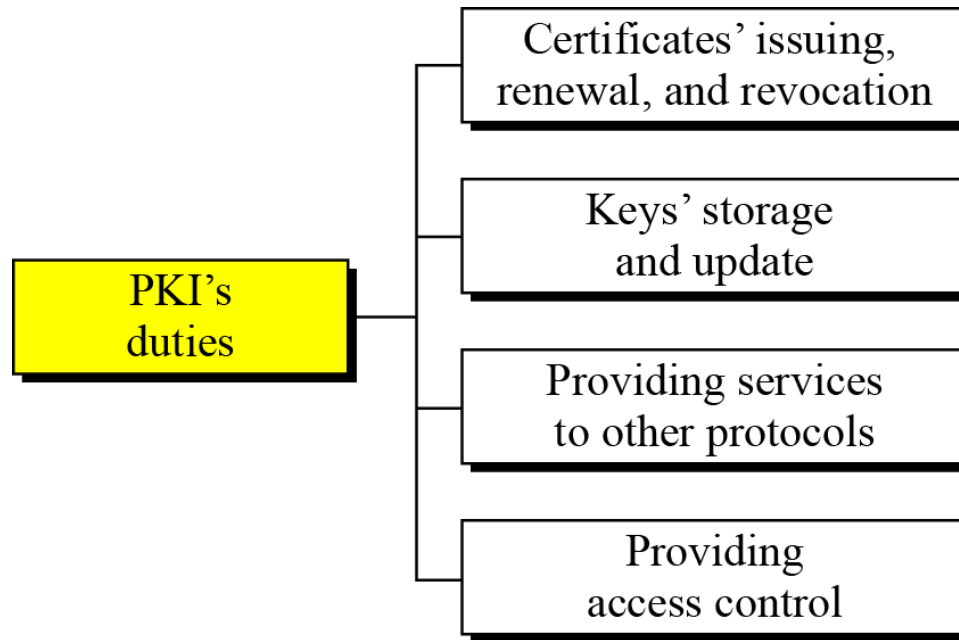
- each CA has certificates for clients (forward) and parent (backward)

each client trusts parents certificates

enable verification of any certificate from one CA by users of all other
CAs in hierarchy



Public-Key Infrastructures (PKI)



Chapter 14

Lightweight Cryptography and Post-Quantum Cryptography

Lightweight Cryptography Concepts

Lightweight cryptography is a subfield of cryptography concerned with the development of cryptographic algorithms for **resource-constrained** devices

The term *lightweight* refers to the characteristic that a cryptographic algorithm makes **minimal resource demands on the host system**

For many existing cryptographic standards, the algorithms incorporate **trade-offs** between **security, performance, and cost requirements** that make them unsuitable for implementation in resource-constrained devices

Lightweight cryptography includes attempts to develop efficient **implementations of conventional cryptographic algorithms** as well as the design of **new lightweight algorithms**



Embedded Systems (1 of 2)

The term *embedded system* refers to the use of electronics and software within a product that has a specific function or set of functions

We can also define an embedded system as any device that includes a computer chip, but that is not a general-purpose workstation, desktop, or laptop computer

Today, many devices that use electric power have an embedded computing system



Embedded Systems (2 of 2)

Types of devices with embedded systems include:

- cell phones
- digital cameras
- video cameras
- calculators
- microwave ovens
- home security systems
- washing machines
- lighting systems
- thermostats
- printers
- various automotive systems
- tennis rackets
- electric toothbrushes
- and numerous types of sensors and actuators in automated systems



Microcontrollers (1 of 2)

A *microcontroller* is a single chip that contains the processor, nonvolatile memory for the program (ROM or flash), volatile memory for input and output (RAM), a clock, and an I/O control unit

- It is also called a “computer on a chip”
- Microcontrollers come in a range of physical sizes and processing power



Microcontrollers (2 of 2)

The processor portion of the microcontroller has a much lower silicon area than other microprocessors and much higher energy efficiency

Billions of microcontroller units are embedded each year in products from toys to appliances to automobiles

- Typically they are used as dedicated processors for specific tasks
- They are integral parts of modern industrial technology and are among the most inexpensive ways to produce machinery that can handle extremely complex functionalities

Another typical feature of a microcontroller is that it does not provide for human interaction

- The microcontroller is programmed for a specific task, embedded in its device, and executes as and when required

Deeply Embedded Systems (1 of 2)

Are a subset of embedded systems

Have a processor whose behavior is difficult to observe both by the programmer and the user

Use a microcontroller

Are not programmable once the program logic for the device has been burned into ROM

Have no interaction with a user



Deeply Embedded Systems (2 of 2)

Are dedicated, single-purpose devices that detect something in the environment, perform a basic level of processing, and then do something with the results

Deeply embedded systems often have wireless capability and appear in networked configurations

The IoT depends heavily on deeply embedded systems

Typically, deeply embedded systems have extreme resource constraints in terms of memory, processor size, time, and power consumption



Constrained Devices

A constrained device is a device with limited volatile and nonvolatile memory, limited processing power, and a low data rate transceiver

Many devices in the IoT are resource constrained

Typical constrained devices are equipped with 8- or 16-bit microcontrollers that possess very little RAM and storage capacities

Resource-constrained devices are often equipped with an IEEE 802.15.4 radio, which enables low-power low-data-rate wireless personal area networks (WPANs) with data rates of 20–250 kbps and frame sizes of up to 127 octets



Categories of Constraints for Lightweight Cryptography (1 of 2)

Chip area

- Is of concern when a cryptographic algorithm is implemented in hardware
- Typically expressed in gate equivalents (GEs)

Energy consumption

- Is a function of several factors including the processing time, the chip area, the operating frequency, and the number of bits transmitted between entities

Program code size and RAM size

- Cryptographic algorithms need to be compact in terms of code and make use of minimal RAM during execution



Categories of Constraints for Lightweight Cryptography (2 of 2)

Communications transmissions rate

- Very constrained devices may be capable of very limited data rates
- The amount of security related data that needs to be transmitted needs to be extremely small

Execution time

- For some devices execution time is constrained by the amount of time the device is present in the communication zone



Radio Frequency Identification (RFID)

A data collection technology that uses electronic tags attached to items to allow the items to be identified and tracked by a remote system

Is increasingly becoming an enabling technology for IoT

The main elements of an RFID system are tags and readers

- RFID tags are small programmable devices with an attached antenna, used for object, animal, and human tracking
- RFID readers acquire and sometimes rewrite information stored on RFID tags that come within operating range
 - Readers are usually connected to a computer system that records and formats the acquired information for further uses



RFID Devices (1 of 2)

Counterfeit goods:

RFID tags can be cloned or modified in order for counterfeit products or parts to pass as genuine.

Authentication can counter this threat.

Environmental logging:

Tampering with information such as temperature logs can pose a threat to the supply chain management of products such as fresh goods and medical supplies.

Data and device authentication can counter this threat.

Privacy of Electronic Product Code (EPC):

The EPC is designed to be stored on an RFID tag and it provides a universal identifier for every physical object anywhere in the world. This raises serious privacy issues if such tags are attached to personal items. Therefore, the tag must also identify the reader as trusted before divulging traceable information.



RFID Devices (2 of 2)

Antitheft:

Data may be written to the tag to indicate to an exit portal **whether or not that item has been sold**. Persistent memory write and lock operations must be protected to prevent theft.

Returns:

When a tag is returned to a store or manufacturer, an authenticated reset/write mechanism allows it to be reused. The tags maintain some amount of persistent memory; read, write, and lock operations to this memory must be authenticated to prevent tamper and unauthorized modification. Authenticated reads allow data to be visible only for the tag's owner.



Electronic Home Appliances and Smart TV

A number of home appliances are now equipped with embedded processors that provide a range of services and may be connected to the Internet

To lower cost, these embedded systems are generally very constrained and are almost constantly under full load, leaving limited resources for security features

These devices are vulnerable to unauthorized access that may tamper with the control signals or issue illegal commands that would lead to abnormal operations

These devices will also usually have updateable software making authentication methods important



Smart Agricultural Sensors

Environmental sensors in agricultural settings can improve productivity and yield

- For example, the sensors can operate with actuators to control the timing and amount of watering and to automatically open and close greenhouse windows and to schedule pest control

Requirements for sensor networks include

- autonomously driven
- small size
- low power consumption
- low cost so that large numbers of sensors can be employed

These devices need to be tamper resistant to prevent sabotage



Medical Sensors

Wireless medical sensors permit health monitoring of patients outside of a hospital setting, capturing and transmitting a number of medical and health-related measures

These devices are generally extremely small and use very little power



Industrial Systems (1 of 2)

In factories, the transportation, processing, and assembly operations have been automated to improve operational efficiency

Several machine tools and robots can be connected by a network to share manufacturing information and to manage the processes based on the data collected by sensors

Through a network, it is also possible to store information at a single place and to manage the equipment from a central location



Industrial Systems (2 of 2)

When connected to the Internet, these systems can be vulnerable both to the exposure of data and to sabotage

The risk is especially high in the case of critical public infrastructure, such as power distribution systems, nuclear power plants, water treatment, and air traffic control

The execution of unauthorized commands or the failure to execute authorized commands can lead to significant and even catastrophic damage

Thus, authentication, authorization, and availability mechanisms are essential



Automobiles

Modern automobiles provide both in-vehicle communication as well as wireless communication with external entities via small embedded systems

These onboard embedded devices are part of what are termed vehicle communications systems, which are networks in which vehicles and roadside units are the communicating nodes, providing each other with information, such as safety warnings and traffic information

They can be effective in avoiding accidents and traffic congestion

Among security concerns are authentication to ensure that all the communications are accurate and can't be spoofed, and privacy to ensure that the communications can't be used to track cars



Figure 14.1 Lightweight Cryptography Trade-Offs

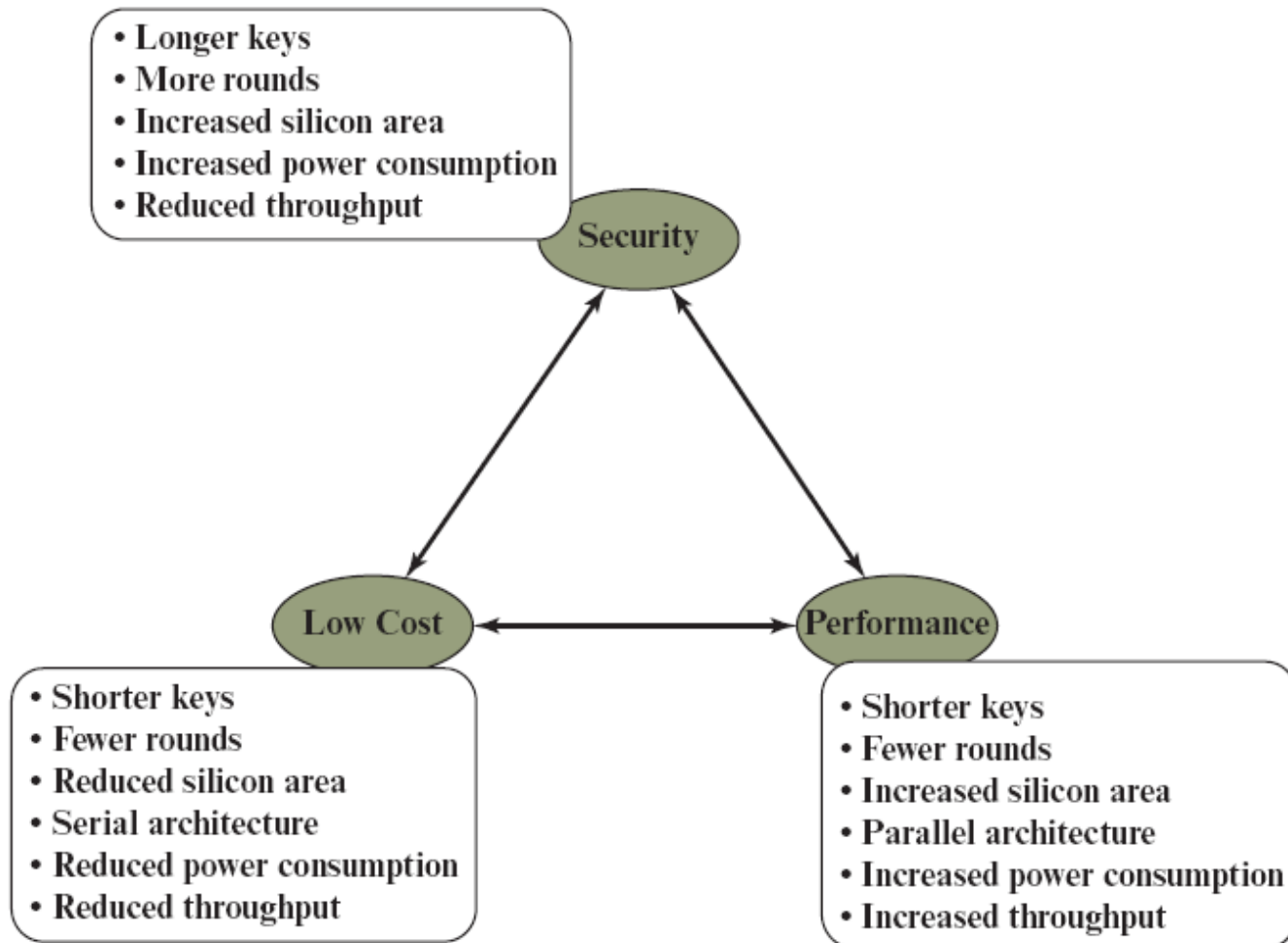
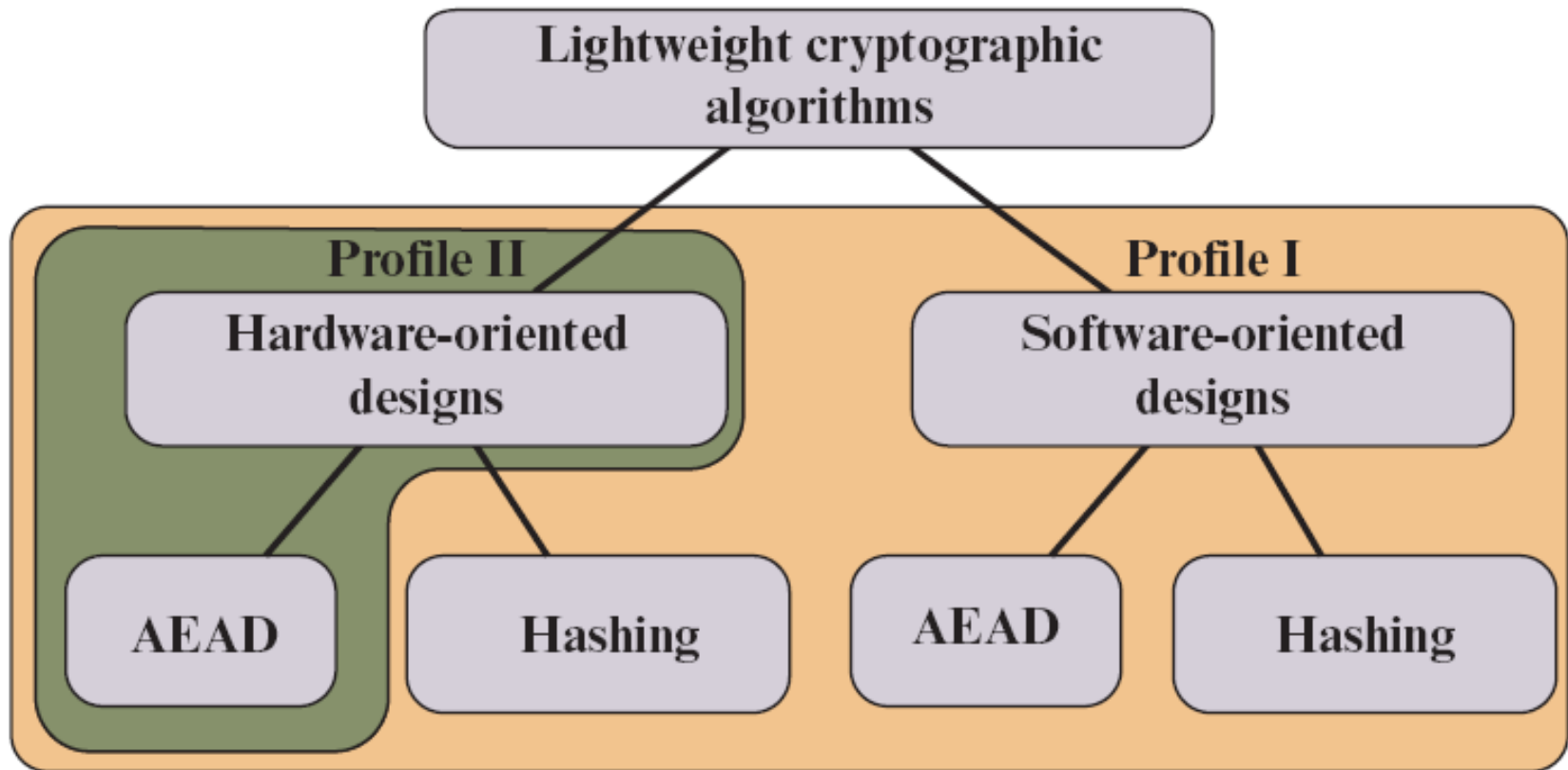


Figure 14.2 Profiles for Lightweight Cryptography



AEAD = Authenticated Encryption with Associated Data



Side-Channel Attack

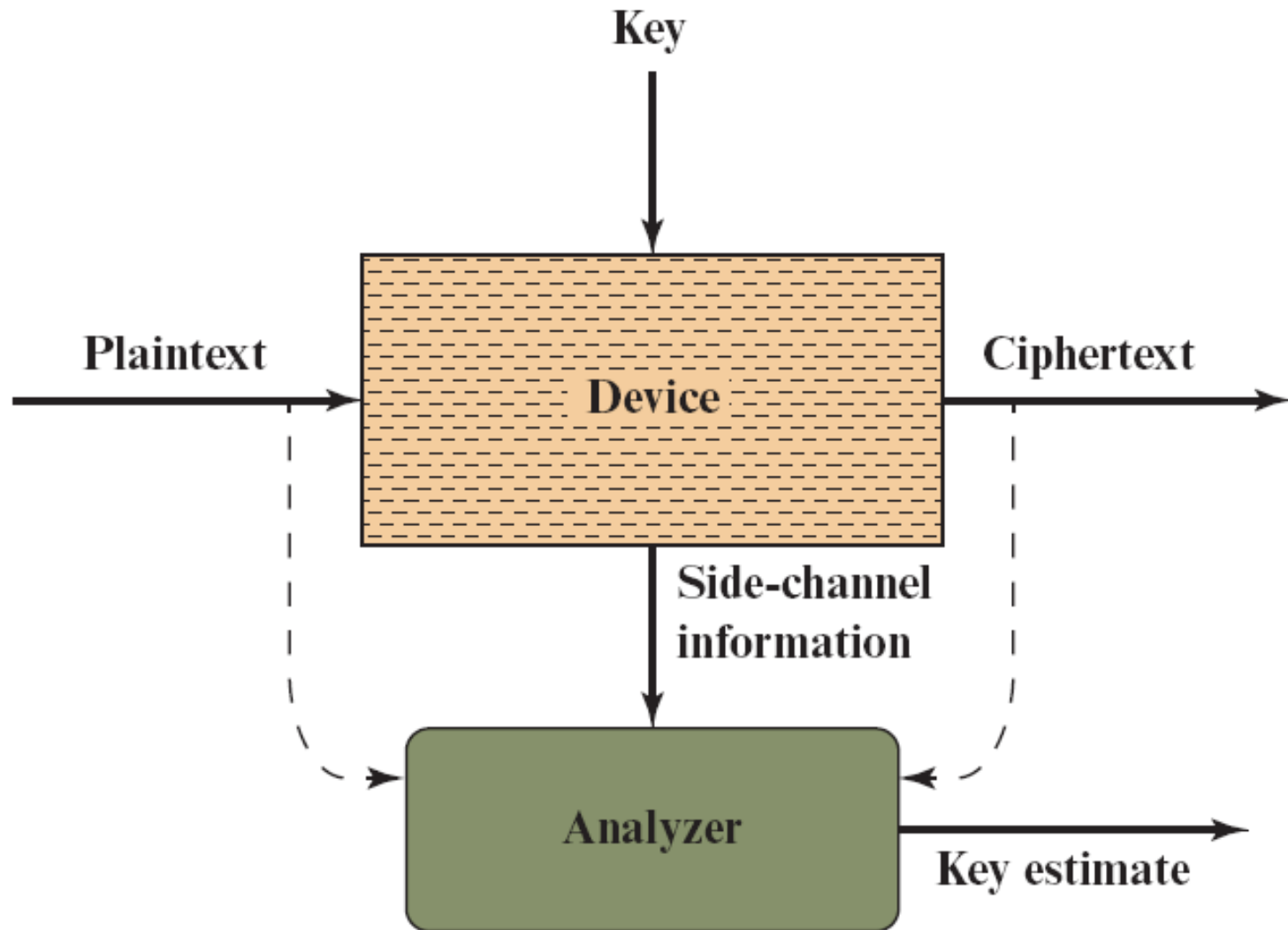
An attack enabled by leakage of information from a physical cryptosystem

An attacker exploits the physical environment to recover some leakage that can be used to break the cryptographic algorithm

Characteristics that could be exploited in a side-channel attack include **running time, power consumption, and electromagnetic and acoustic emissions**



Figure 14.3 Side-Channel Attack



References

1. http://www.sfu.ca/~ljilja/ENSC427/News/Kurose_Ross/Chapter_8_V7.0_Accessible.pdf
2. Stallings, W., Cryptography and Network Security: Principles and Practice 0133354695, 9780133354690.
3. **A.K. Dewdney, The New Turning Omnibus, pp. 250-257, Henry Holt and Company, 2001.**
4. www.whatis.com (search for kerberos)
5. Bryant, W. Designing an Authentication System: A Dialogue in Four Scenes. <http://web.mit.edu/kerberos/www/dialogue.html>
6. Kohl, J.; Neuman, B. “The Evolution of the Kerberos Authentication Service” <http://web.mit.edu/kerberos/www/papers.html>
7. <http://www.isi.edu/gost/info/kerberos/>





Lnu.se