Beveiliging van netwerken en computers

CHAPTER 2
BASIC CONCEPTS

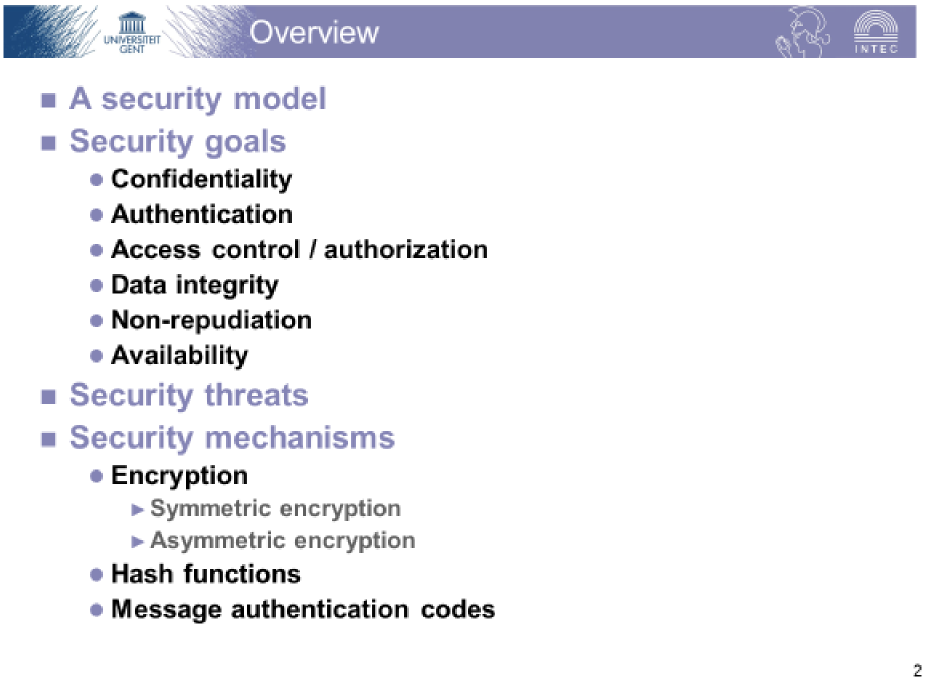
PROF. DR. IR. ELI DE POORTER

eli.depoorter@ugent.be

GHENT UNIVERSITY – IMEC

IDLAB

http://idlab.technology | *http://idlab.ugent.be*

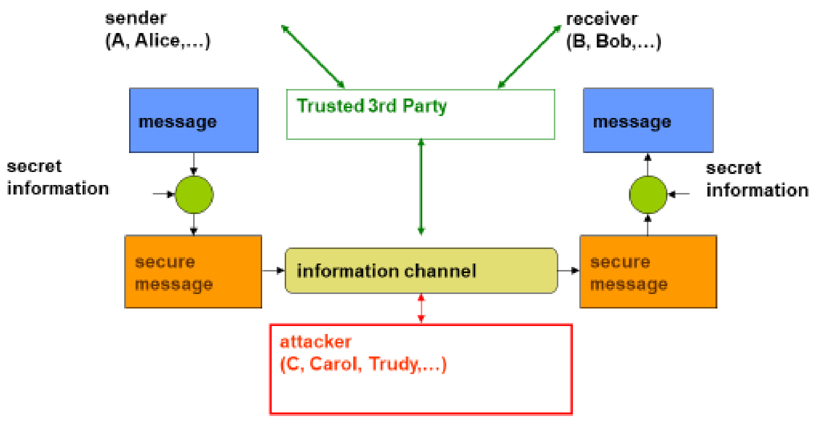


Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

2

Network security model



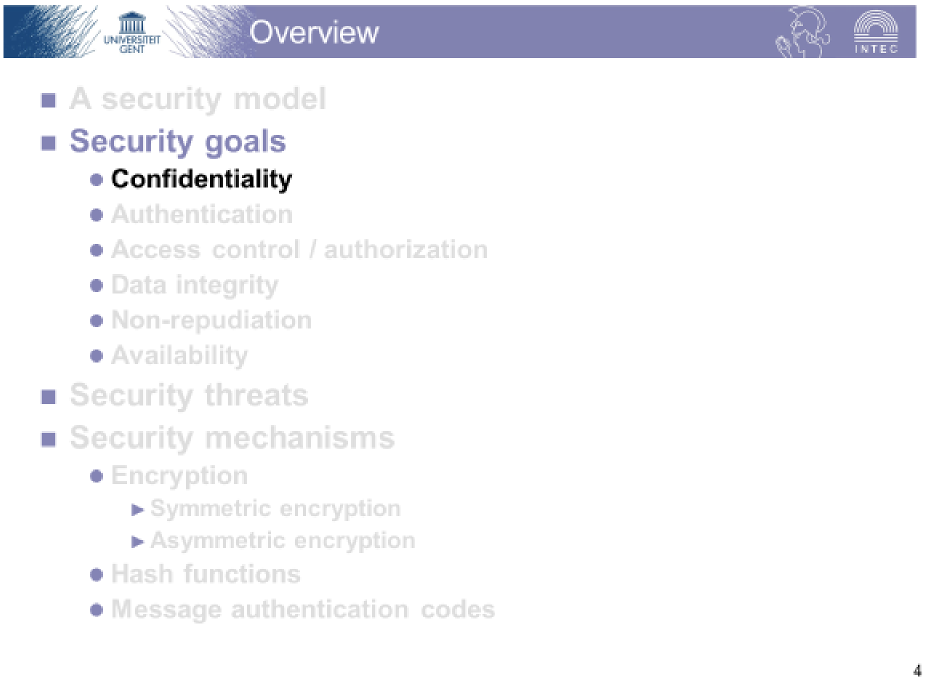
3

Modern security principles originate from Auguste Kerckhoffs in the 19th century: a cryptosystem should be secure even if everything about the system, except the key, is public knowledge (Kerckhoff's principle). This means that typically the security transformation is publicly known (and is often standardized), and only the secret information (key) used in the security transformation to convert the message into a secure message is kept secret. The full method for secure information exchanges between A(lice) and B(ob) may require a prior exchange of several messages, including both data and control messages.

Alice and Bob are two commonly used placeholder names to make security exchanges easier to follow. The names are used for convenience; for example, "Alice sends a message to Bob encrypted with his public key" is easier to follow than "Party A sends a message to Party B encrypted by Party B's public key." These names were originally used by Ron Rivest in the 1978 Communications of the ACM article presenting the RSA cryptosystem, and in ”A Method for Obtaining Digital Signatures and Public-Key Cryptosystems” published April 4, 1977, revised September 1, 1977, as technical Memo LCS/TM82.

Alice and Bob are archetypes in cryptography, Eve is also common. Names further down the alphabet are less common. Other (less frequently used) names include:

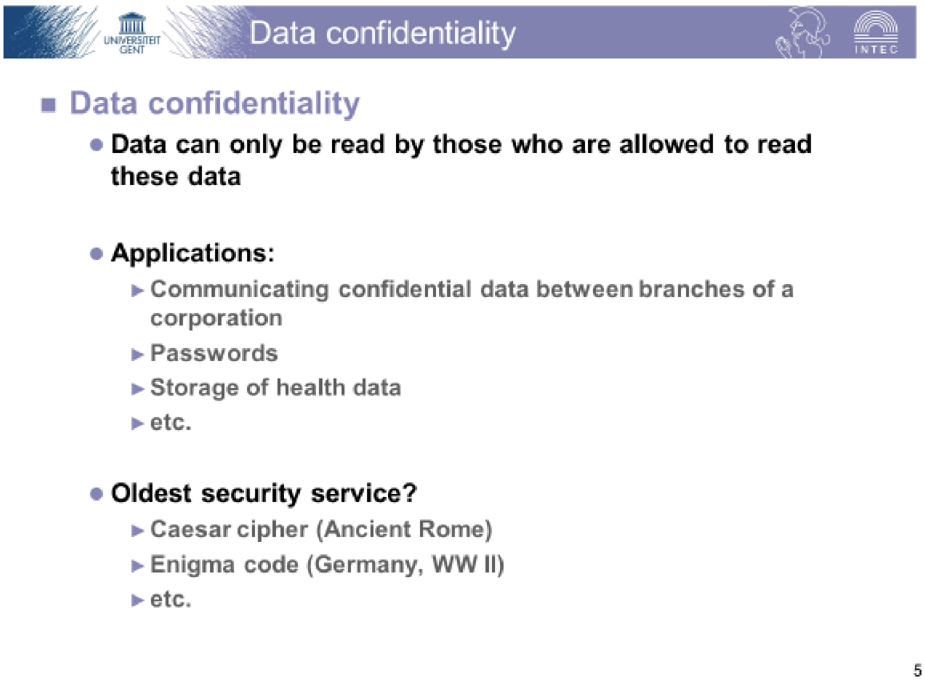
- Carol, Chuck, Carlos or Charlie, as a third participant in communications, sometimes with malicious intent.
- Craig, the password cracker (usually encountered in situations with stored hashed/salted passwords).
- Dan or Dave, a fourth participant.
- Eve, an eavesdropper, is usually a passive attacker. While she can listen in on messages between Alice and Bob, she cannot modify them.
- Trudy, a malicious attacker (an intruder). Unlike the passive Eve, this one is the active man-in-the-middle attacker who can modify messages, substitute his/her own messages, replay old messages, and so on. The difficulty of securing a system against Trudy is much greater than against Eve.
- Walter, a warden, may be needed to guard Alice and Bob in some respect, depending on the protocol being discussed.
- Wendy, a whistleblower, is an insider with privileged access who may be in a position to divulge the information.



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

4

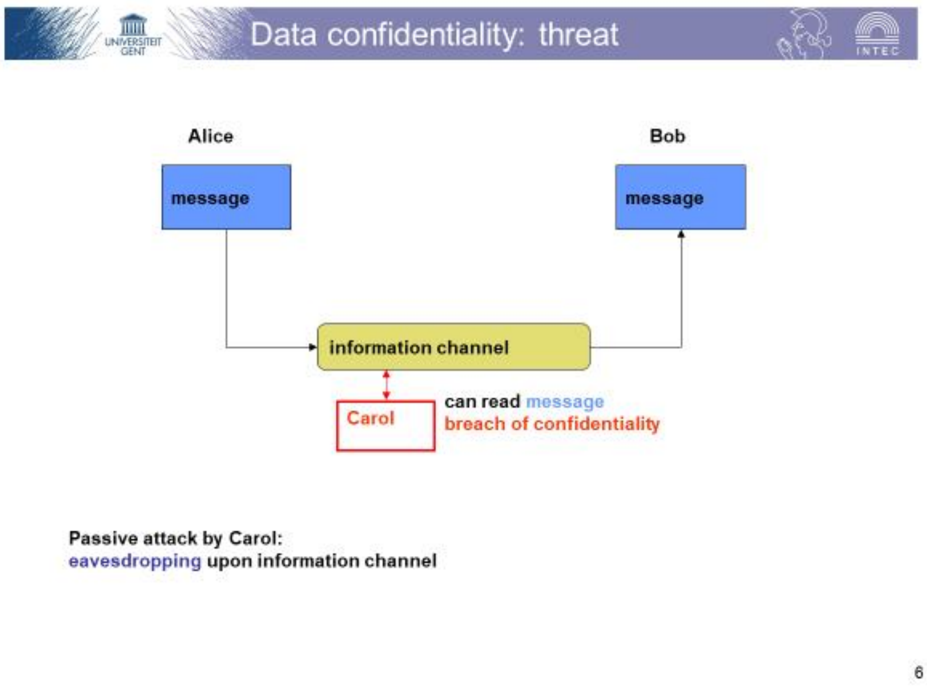


Data confidentiality

- Data confidentiality
  - **Data can only be read by those who are allowed to read these data**
  - **Applications:**
    - Communicating confidential data between branches of a corporation
    - Passwords
    - Storage of health data
    - etc.
  - **Oldest security service?**
    - Caesar cipher (Ancient Rome)
    - Enigma code (Germany, WW II)
    - etc.

5

Confidentiality is translated as “vertrouwelijkheid” in Dutch.

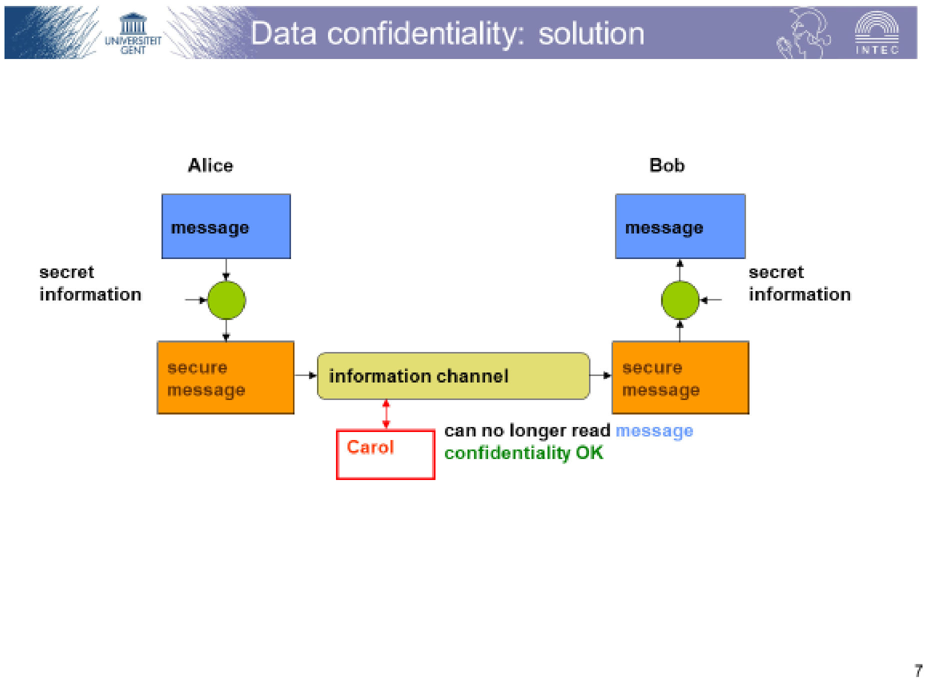


Data confidentiality: threat

Passive attack by Carol:
eavesdropping upon information channel

6

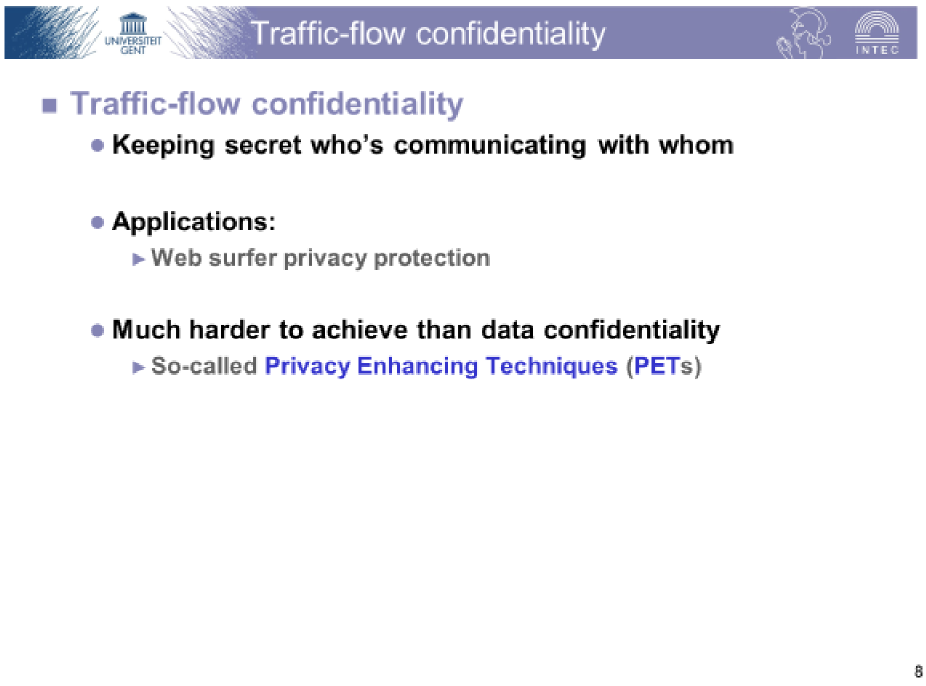
E.g. Alice submits a tender for a procurement by Bob. Neither party uses any special security mechanism (e.g. simple communication using e-mail, ftp, etc.). Carol can read the transmitted message. Carol can then submit her own proposal, adapted using her knowledge of Alice’s offer.



Data confidentiality: solution

7

To make sure the message no longer is readable for Carol, a security transformation (encryption) is introduced, allowing only those (i.c. Alice en Bob) who have access to secret information (secret key) to recover the original message. Carol can still intercept the secure message, but she won't be able to derive the original message from this.

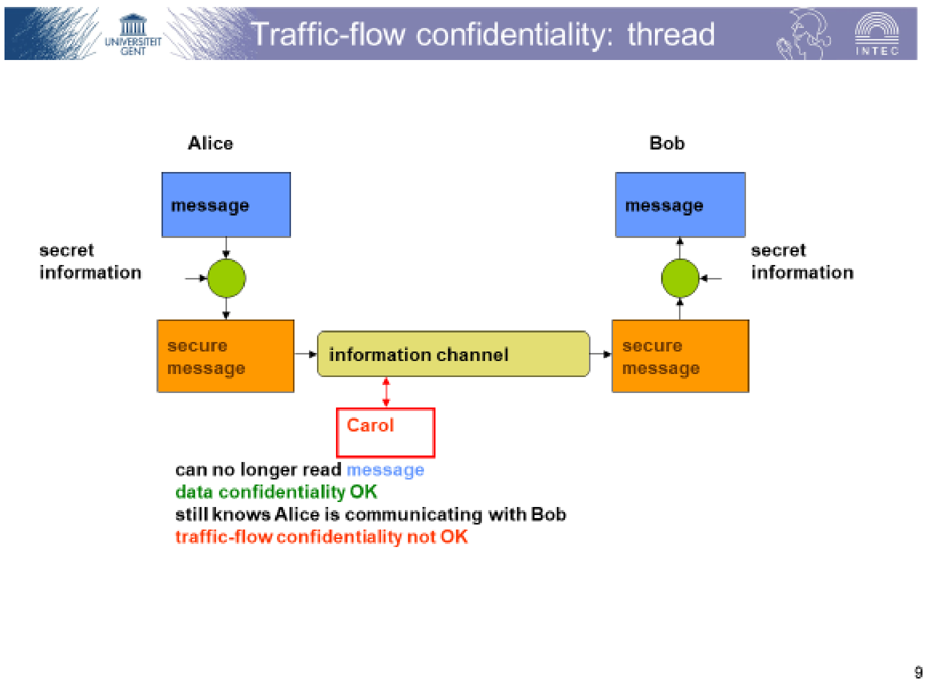

UNIVERSITEIT GENT
INTEC

Traffic-flow confidentiality

- Traffic-flow confidentiality
  - Keeping secret who's communicating with whom
  - Applications:
    - Web surfer privacy protection
  - Much harder to achieve than data confidentiality
    - So-called Privacy Enhancing Techniques (PETs)

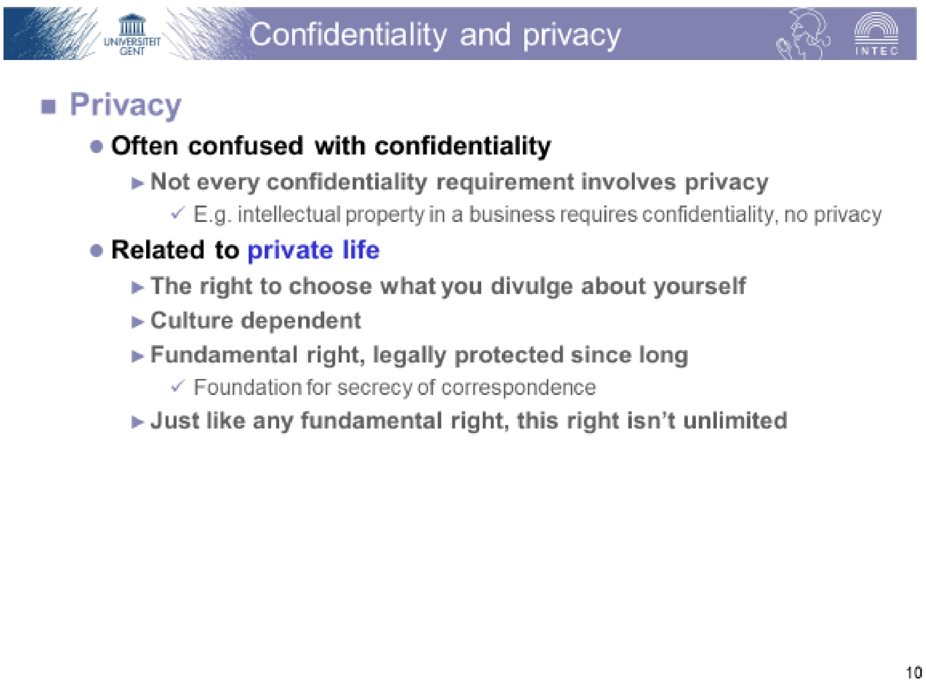
8

As an examples of a existing privacy enhancing technologies (PET) communication anonymizers exist that hide the real online identity (email address, IP address, etc.) of a user and replacing it with a non-traceable identity (disposable / one-time email address, random IP address of hosts participating in an anonymizing network, pseudonym, etc.). A well-known PET is the onion routing protocol used by the Tor anonymous network.



Passive attack by Carol: tapping the information channel between both parties A(lice) and B(ob).

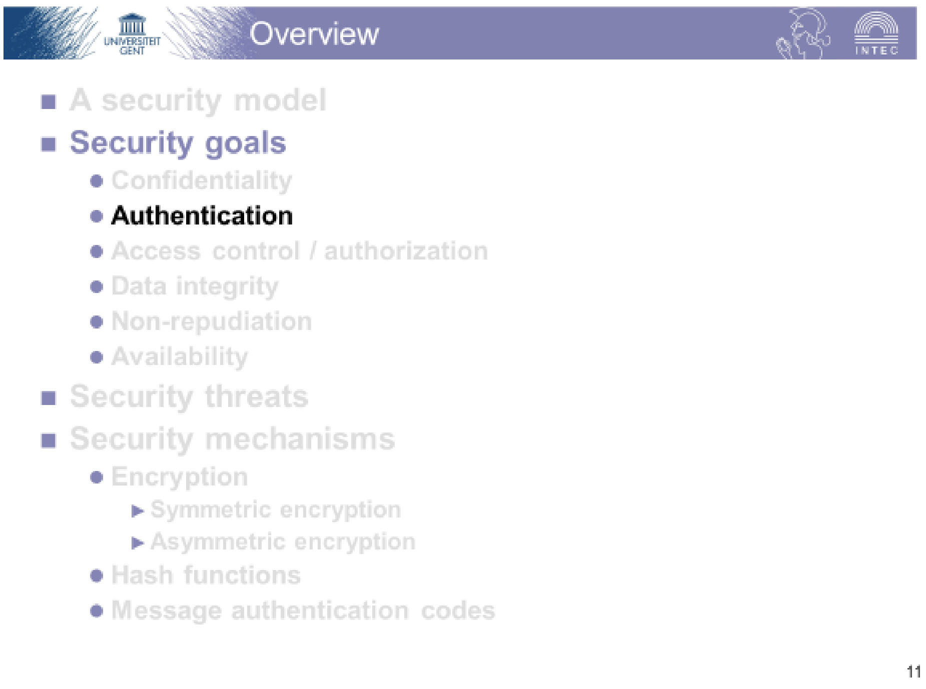
With the solution for data confidentiality Carol can’t read the transmitted message any longer, but she can still deduce that Alice is communicating with Bob (e.g. from IP headers, or by analyzing the communication flow outgoing from Alice and incoming at Bob: “traffic-analysis”).



Confidentiality and privacy

- Privacy
  - **Often confused with confidentiality**
    - **Not every confidentiality requirement involves privacy**
      - ✓ E.g. intellectual property in a business requires confidentiality, no privacy
  - **Related to private life**
    - **The right to choose what you divulge about yourself**
    - **Culture dependent**
    - **Fundamental right, legally protected since long**
      - ✓ Foundation for secrecy of correspondence
    - **Just like any fundamental right, this right isn't unlimited**

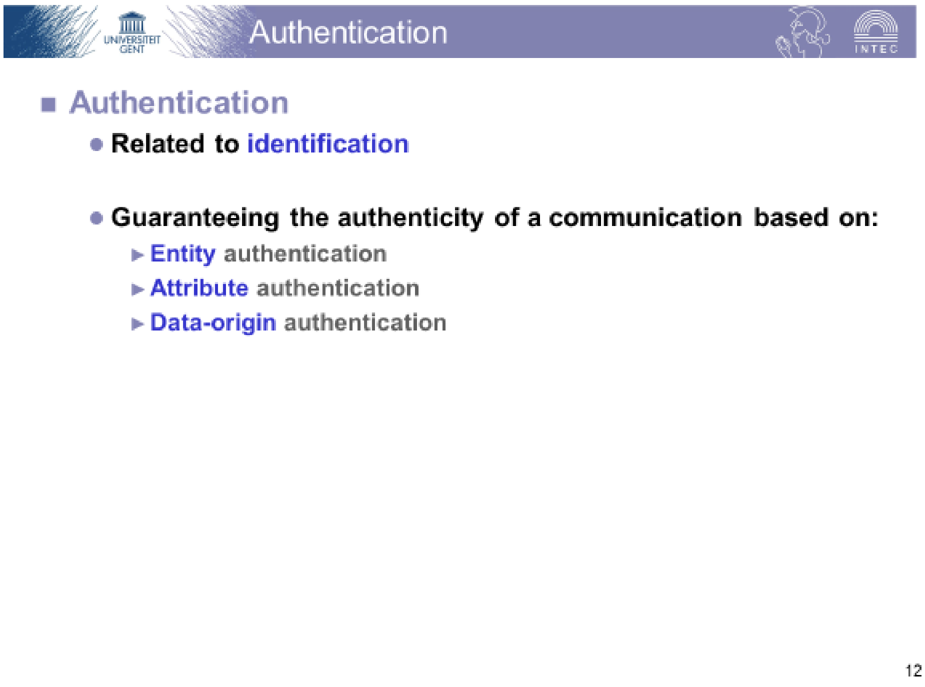
10



Overview

- A security model
- Security goals
  - Confidentiality
  - **Authentication**
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

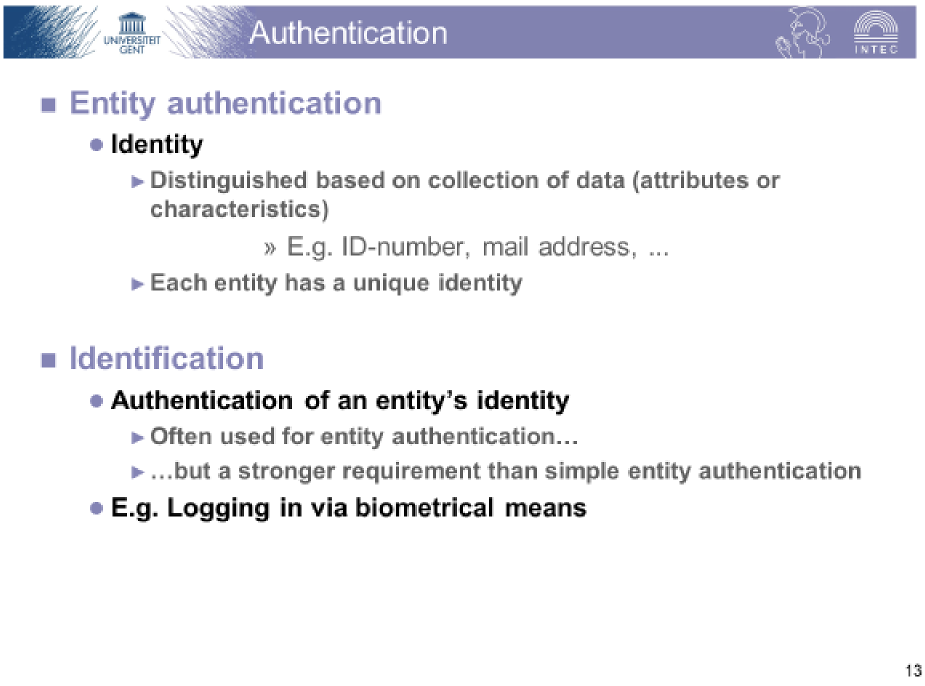
11



Authentication

- **Authentication**
  - Related to identification
  - Guaranteeing the authenticity of a communication based on:
    - Entity authentication
    - Attribute authentication
    - Data-origin authentication

12



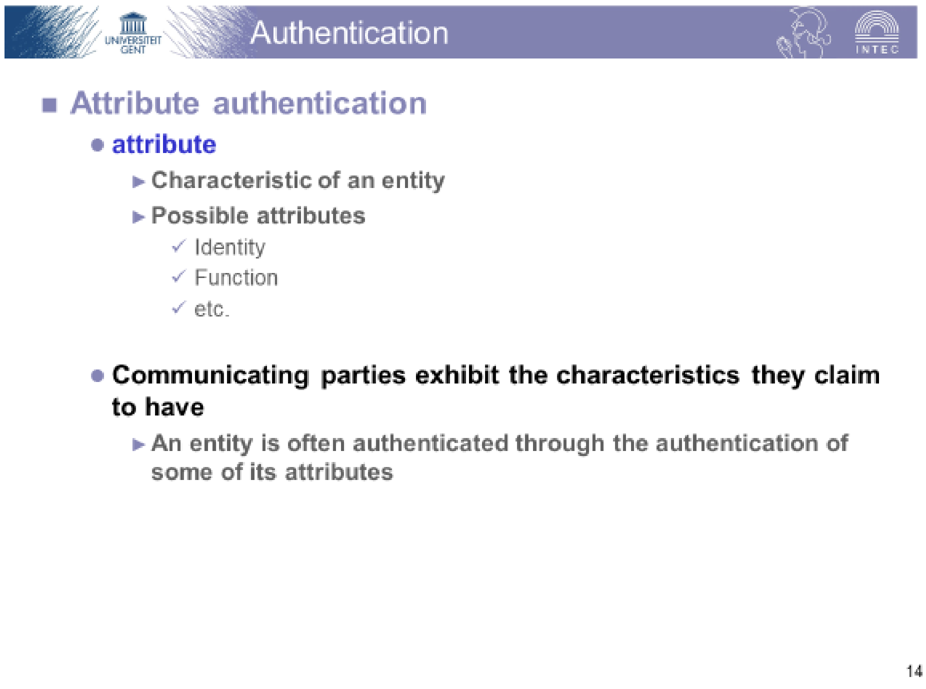
Authentication

- **Entity authentication**
  - Identity
    - Distinguished based on collection of data (attributes or characteristics)
      - » E.g. ID-number, mail address, ...
    - Each entity has a unique identity
- **Identification**
  - Authentication of an entity's identity
    - Often used for entity authentication…
    - …but a stronger requirement than simple entity authentication
  - E.g. Logging in via biometrical means

13

According to the Merriam-Webster dictionary, an entity is "something that has separate and distinct existence and objective or conceptual reality". An entity typically has a number of attributes (or characteristics) that, taken together, form a unique combination. Entity authentication requires a validation to check if communicating parties are who they claim to be. In practical implementations, entity authentication is often based on authentication of some attributes of these entities. An example is a person's first name, last name, date and hour of birth and place of birth that together uniquely identify a person. Alternatively, the identity number of a passport is also an attribute that uniquely identifies a person.

Identification is used when an identity needs to be uniquely represented and the identity needs to be validated. An example where entity authentication doesn't really require the entity's identification is the registration with many websites. During registration the user generates a login and password, and provides some data (name, address, email address, etc.) to the website. Typically the only important data is the e-mail address (to which possibly the password will be sent), while all other data absolutely don't need to be authentic. The next time the user logs in to this website, he'll be authenticated via the login and password (he is then deemed the authentic owner of this couple). But one can't assert his identity is authentic, since it wasn't really verified at any time.

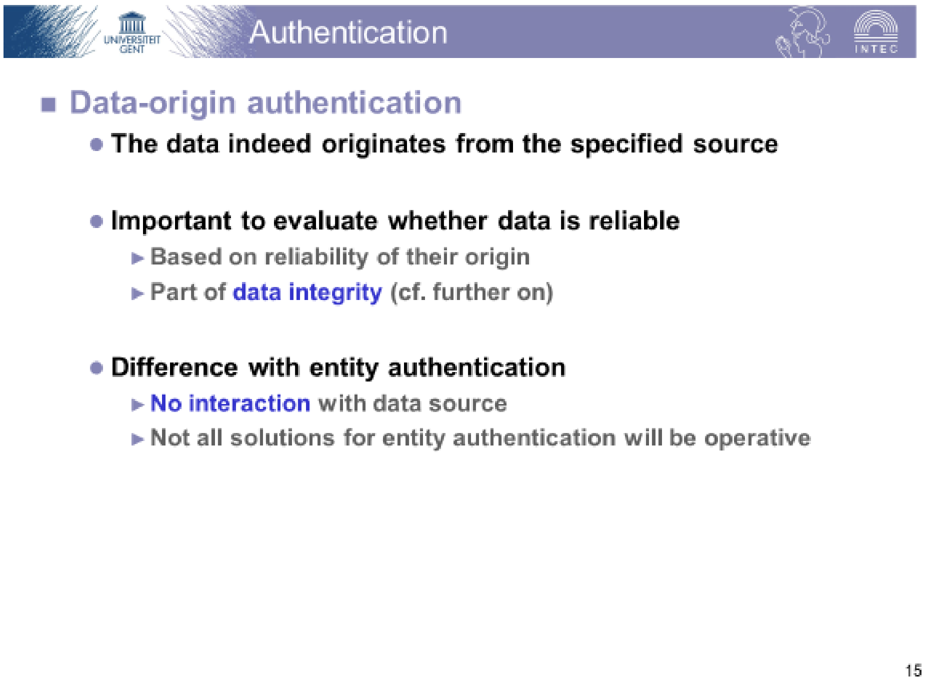


Authentication

- Attribute authentication
  - attribute
    - Characteristic of an entity
    - Possible attributes
      - ✓ Identity
      - ✓ Function
      - ✓ etc.
  - **Communicating parties exhibit the characteristics they claim to have**
    - An entity is often authenticated through the authentication of some of its attributes

14

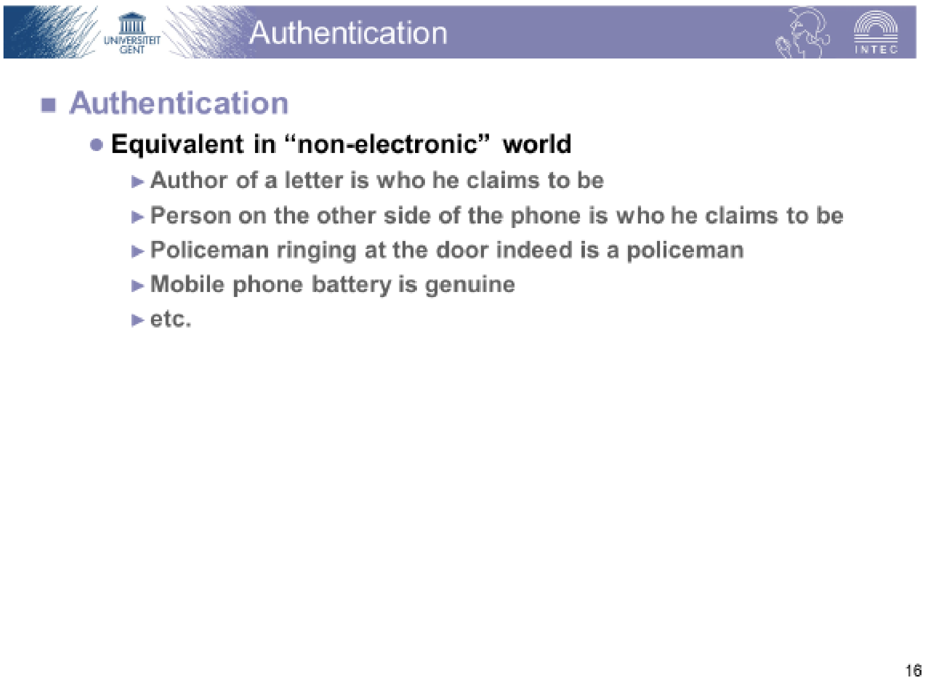
An already sufficiently authenticated entity (e.g. using an electronic identity card, which adequately captures the identity of a communicating entity) may need to authenticate additional attributes (e.g. the fact that he is a doctor to access certain medical information).



Authentication

- **Data-origin authentication**
  - **The data indeed originates from the specified source**
  - **Important to evaluate whether data is reliable**
    - Based on reliability of their origin
    - Part of **data integrity** (cf. further on)
  - **Difference with entity authentication**
    - **No interaction** with data source
    - Not all solutions for entity authentication will be operative

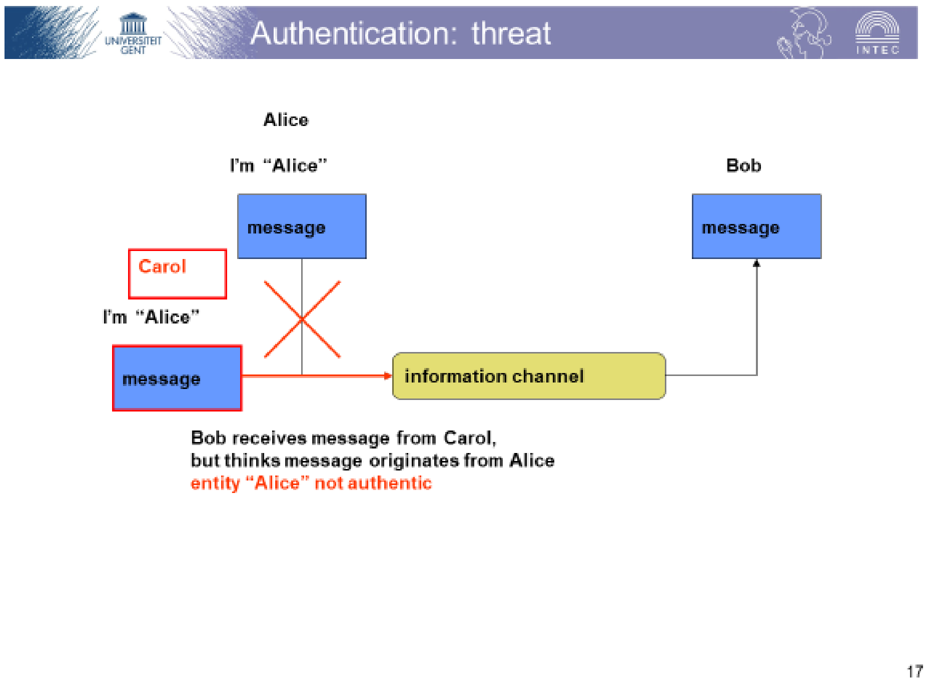
15



Authentication

- **Authentication**
  - **Equivalent in “non-electronic” world**
    - Author of a letter is who he claims to be
    - Person on the other side of the phone is who he claims to be
    - Policeman ringing at the door indeed is a policeman
    - Mobile phone battery is genuine
    - etc.

16



Authentication: threat

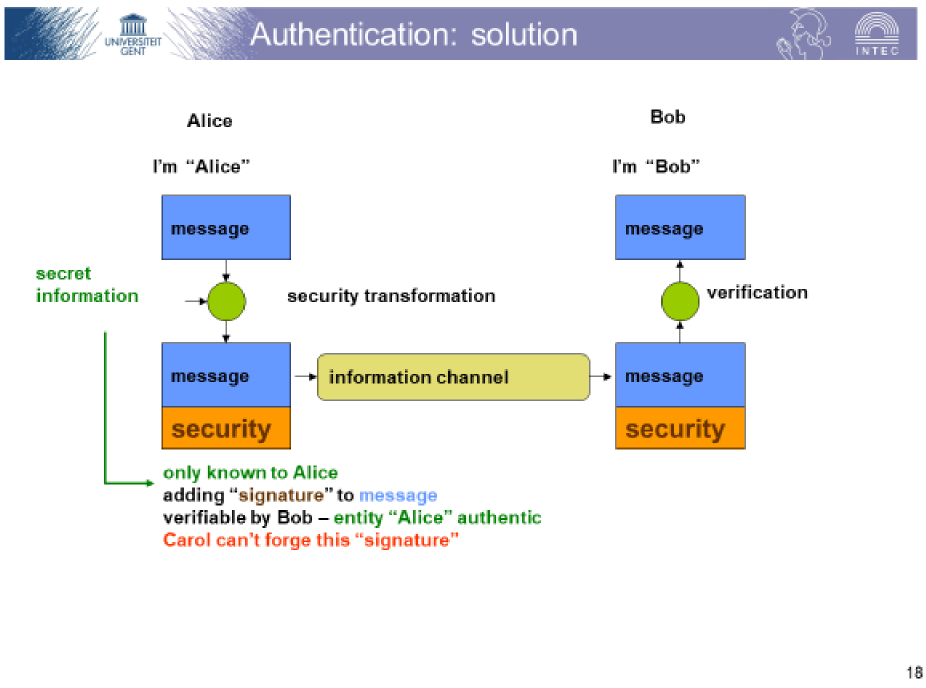
Bob receives message from Carol,
but thinks message originates from Alice
entity "Alice" not authentic

17

Active attack by Carol on the communication between Alice and Bob, e.g. by hijacking the communication, or more simply by sending a forged message.

Example 1) Alice communicates with website of Bank Bob. Bob needs to know whether it is indeed Alice who logs in, and not Carol masquerading as Alice in order to gain access to Alice's accounts at Bank Bob. (**authenticity of the entity** Alice; conversely, Alice may also want to be sure she's communicating with the website of Bank Bob and not with a fake website set up by Carol).

Example 2) Bob has received a message from software vendor Alice (e.g. by email) with a major security patch for the software in attachment. Before Bob installs the patch, he'd like to be certain it really originates from Alice… and isn't some piece of malware coming from Carol. (**authenticity of the origin** of the data received by Bob).



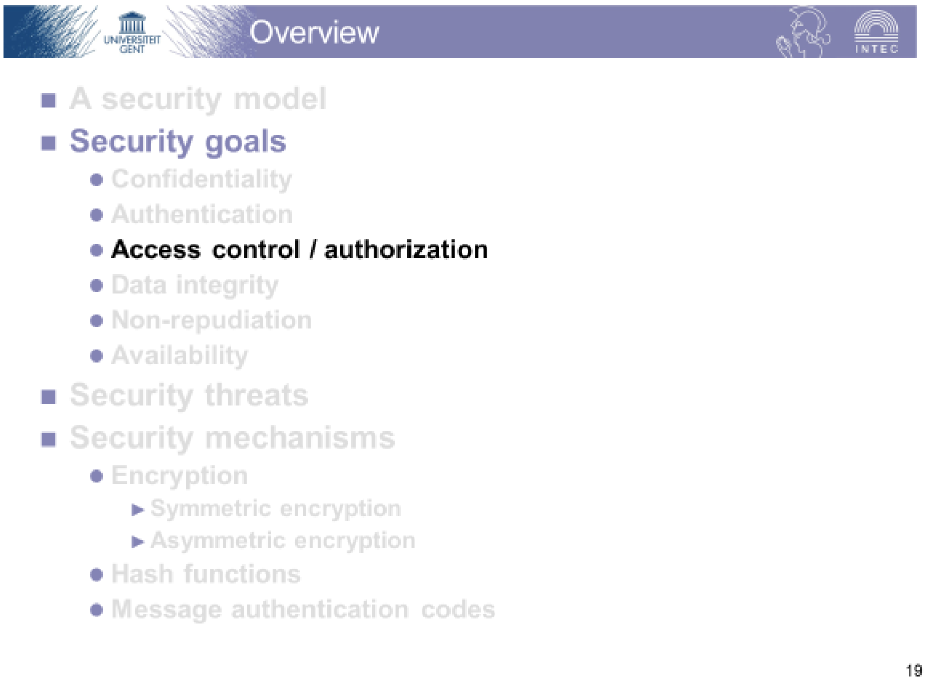
Make sure Carol can't generate a message in which she claims to be someone else anymore. Add some "signature" to Alice's message. This "signature" binds the entity Alice to the contents of the message, so it must:

- be linked to the message (another message has a different "signature")
- be based on secret information only Alice possesses
- be verifiable, even by those who don't know this secret information

The digital signature is based on this principle, and also guarantees the authenticity of the origin of the data.

Note that a potential threat still lurks in this diagram. If Carol eavesdrop upon the information channel she still can intercept a secure message and then forward it again to Bob and in this way still impersonate Alice (replay).

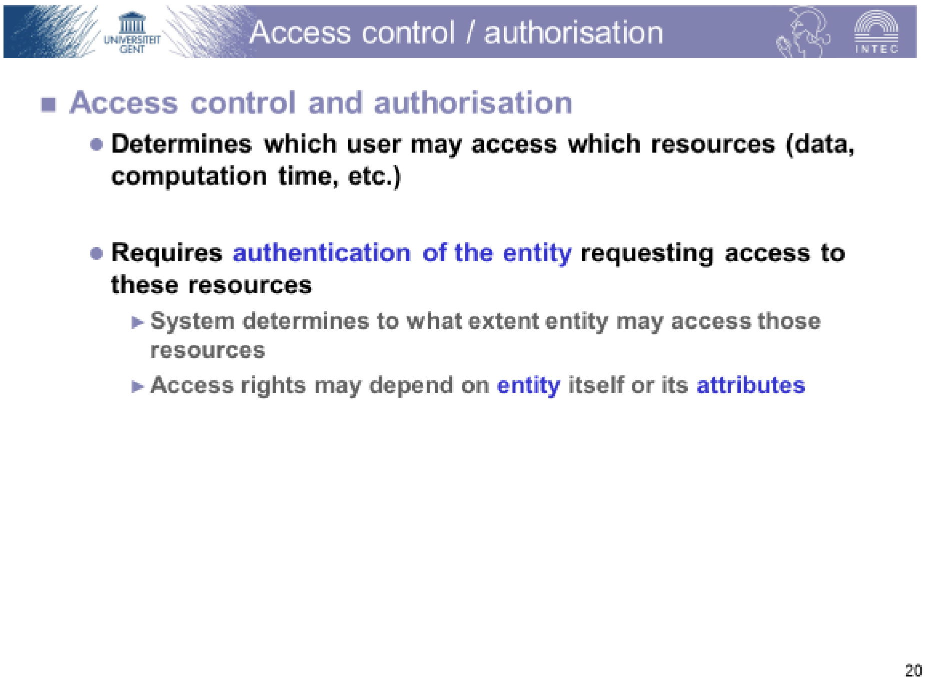
An alternative solution is for Alice to use an (encrypted) password as secret information, that Bob will recognise as Alice's. The danger is that Carol still can eavesdrop on the forwarded password and then use it again (sometimes even if it is encrypted). A remedy against this threat is to ensure the communication between Alice and Bob can also be confidential. A possible implementation could be: sending the password over a TLS/SSL connection (see later for TLS/SSL).



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - **Access control / authorization**
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

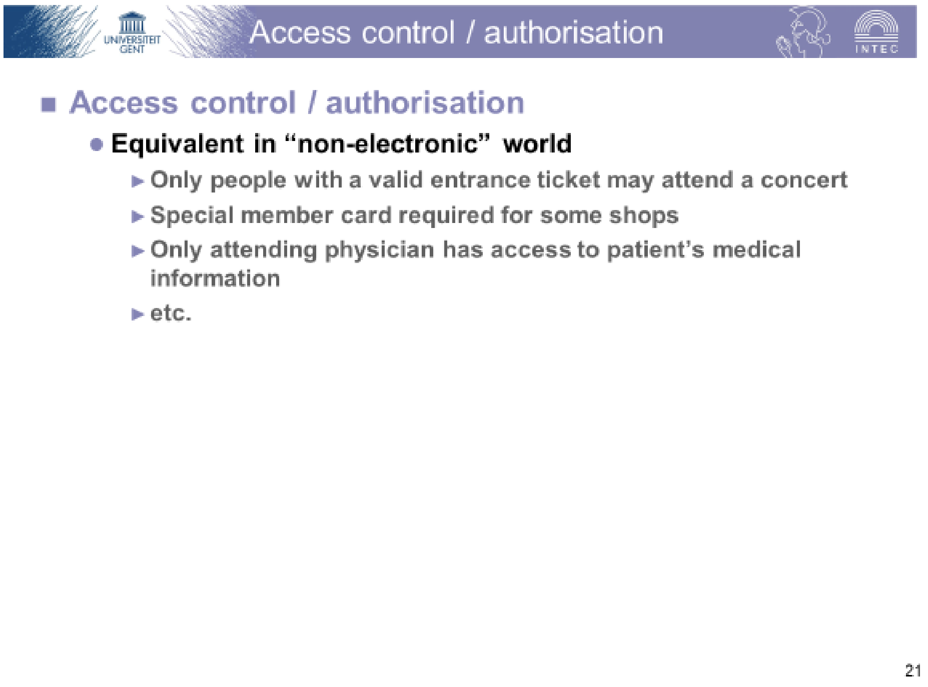
19



Access control / authorisation

- Access control and authorisation
  - **Determines which user may access which resources (data, computation time, etc.)**
  - **Requires authentication of the entity requesting access to these resources**
    - System determines to what extent entity may access those resources
    - Access rights may depend on entity itself or its attributes

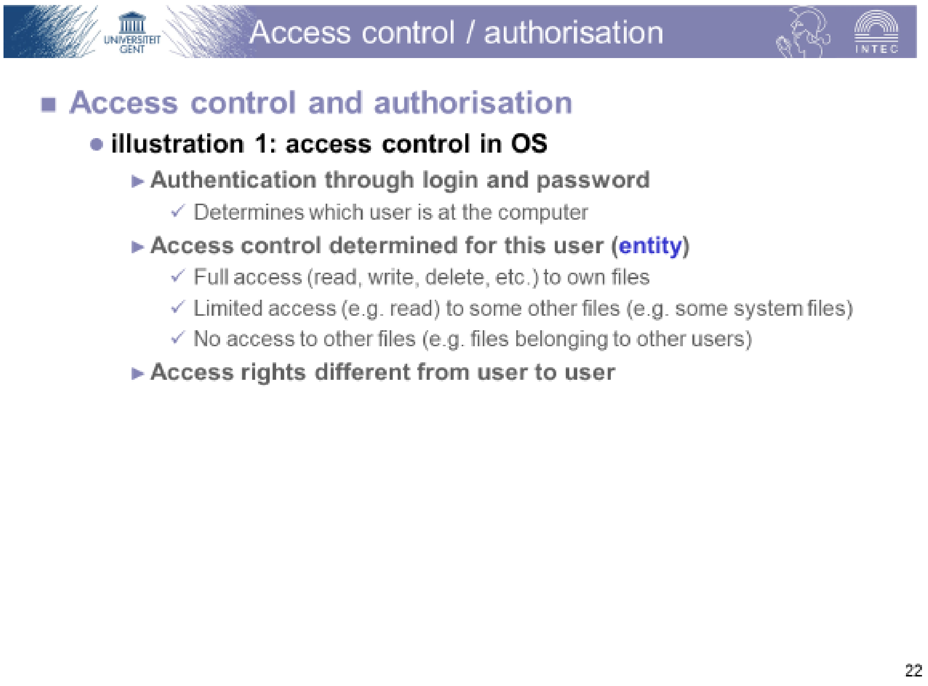
20



Access control / authorisation

- Access control / authorisation
  - **Equivalent in "non-electronic" world**
    - Only people with a valid entrance ticket may attend a concert
    - Special member card required for some shops
    - Only attending physician has access to patient's medical information
    - etc.

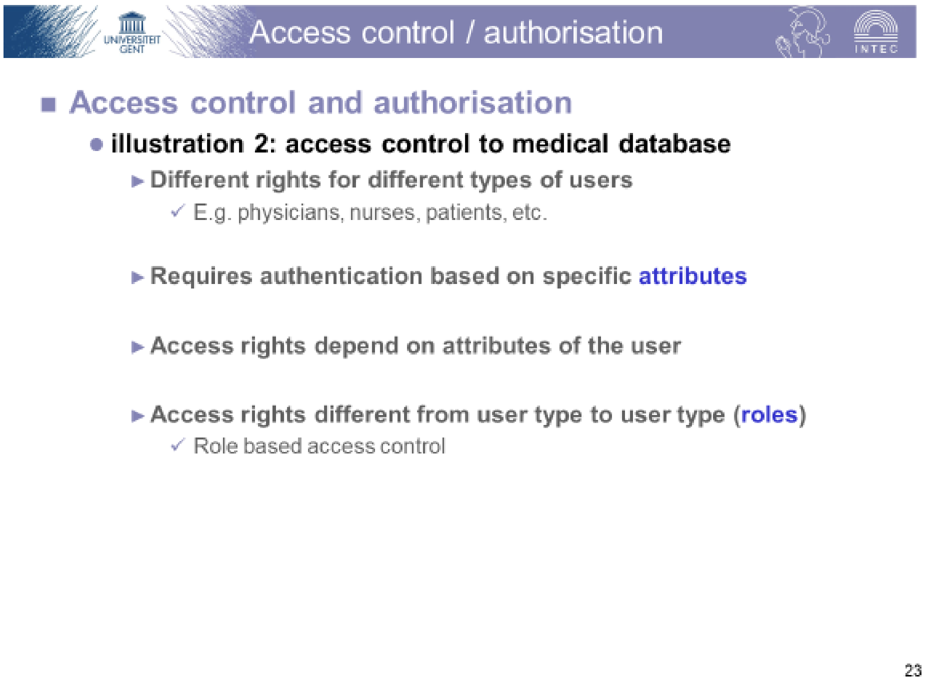
21



Access control / authorisation

- Access control and authorisation
  - **illustration 1: access control in OS**
    - Authentication through login and password
      - ✓ Determines which user is at the computer
    - Access control determined for this user (entity)
      - ✓ Full access (read, write, delete, etc.) to own files
      - ✓ Limited access (e.g. read) to some other files (e.g. some system files)
      - ✓ No access to other files (e.g. files belonging to other users)
    - Access rights different from user to user

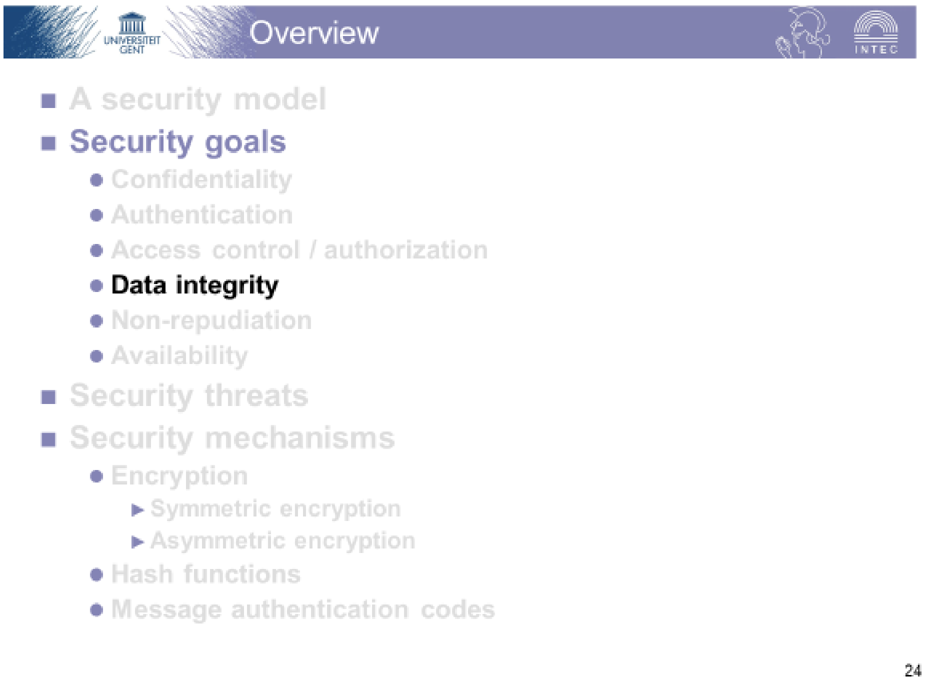
22



Access control / authorisation

- Access control and authorisation
  - **illustration 2: access control to medical database**
    - Different rights for different types of users
      - ✓ E.g. physicians, nurses, patients, etc.
    - Requires authentication based on specific **attributes**
    - Access rights depend on attributes of the user
    - Access rights different from user type to user type (**roles**)
      - ✓ Role based access control

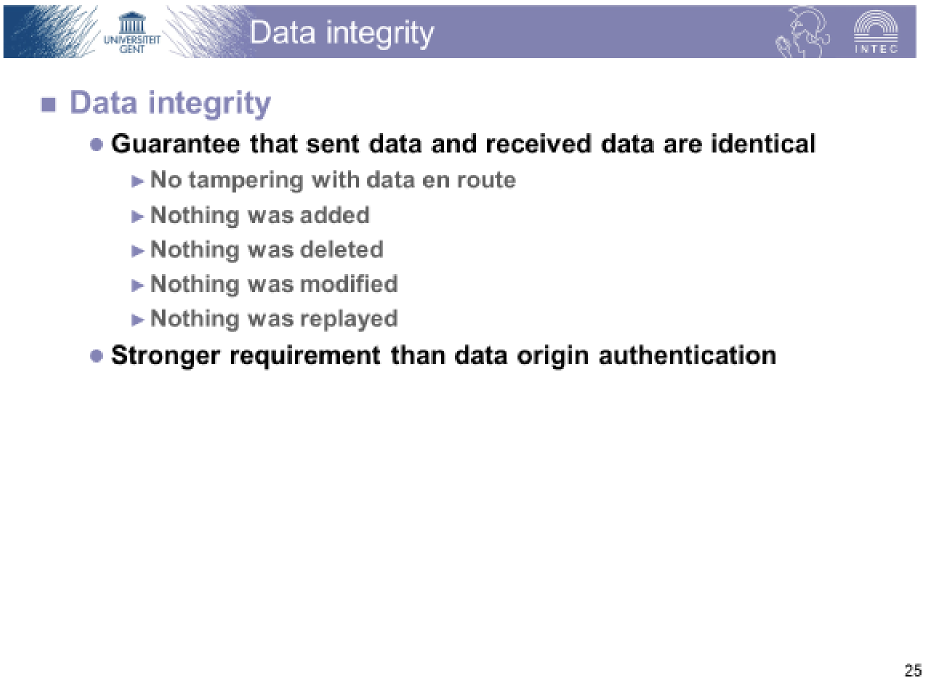
23



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - **Data integrity**
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

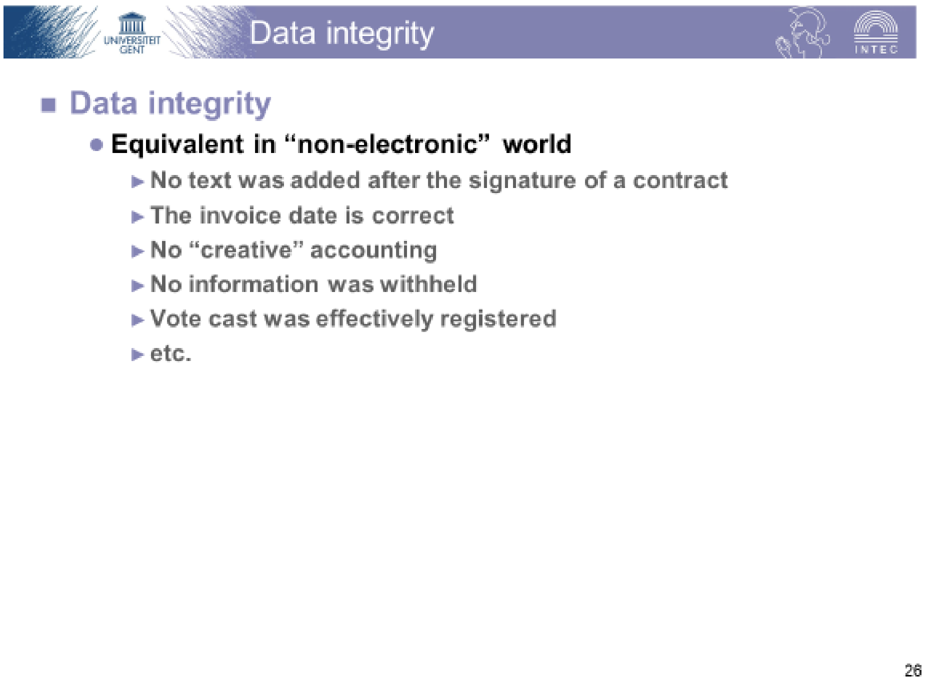
24



Data integrity

- Data integrity
  - Guarantee that sent data and received data are identical
    - No tampering with data en route
    - Nothing was added
    - Nothing was deleted
    - Nothing was modified
    - Nothing was replayed
  - Stronger requirement than data origin authentication

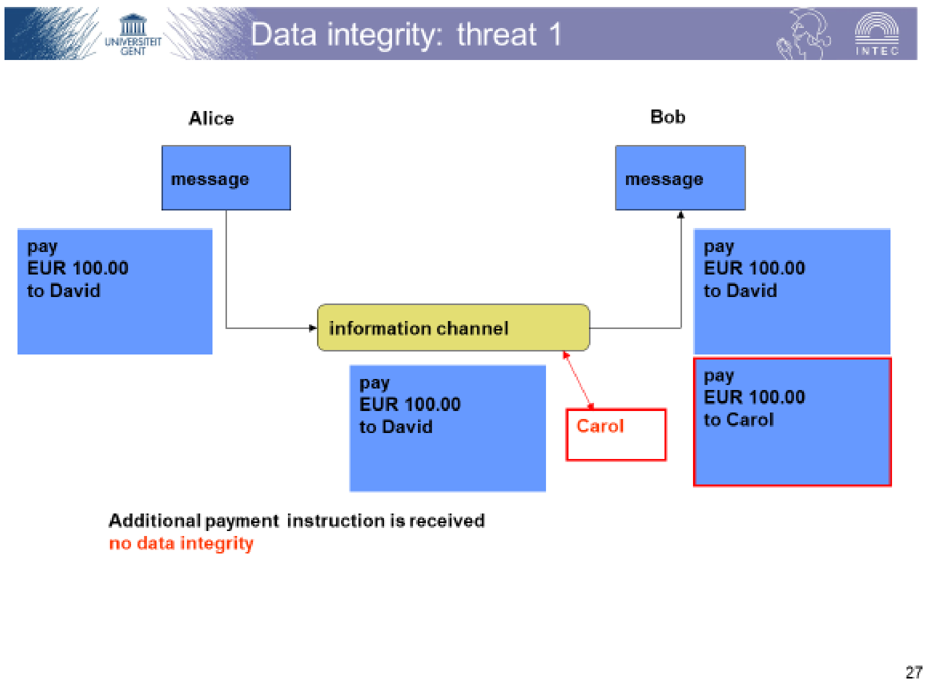
25



Data integrity

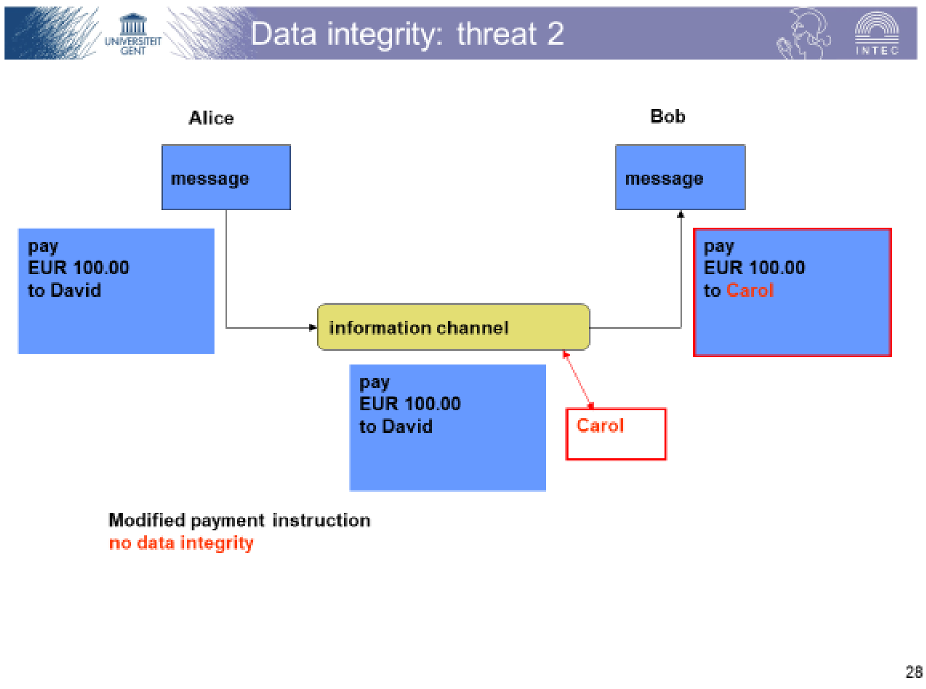
- Data integrity
  - Equivalent in “non-electronic” world
    - No text was added after the signature of a contract
    - The invoice date is correct
    - No “creative” accounting
    - No information was withheld
    - Vote cast was effectively registered
    - etc.

26



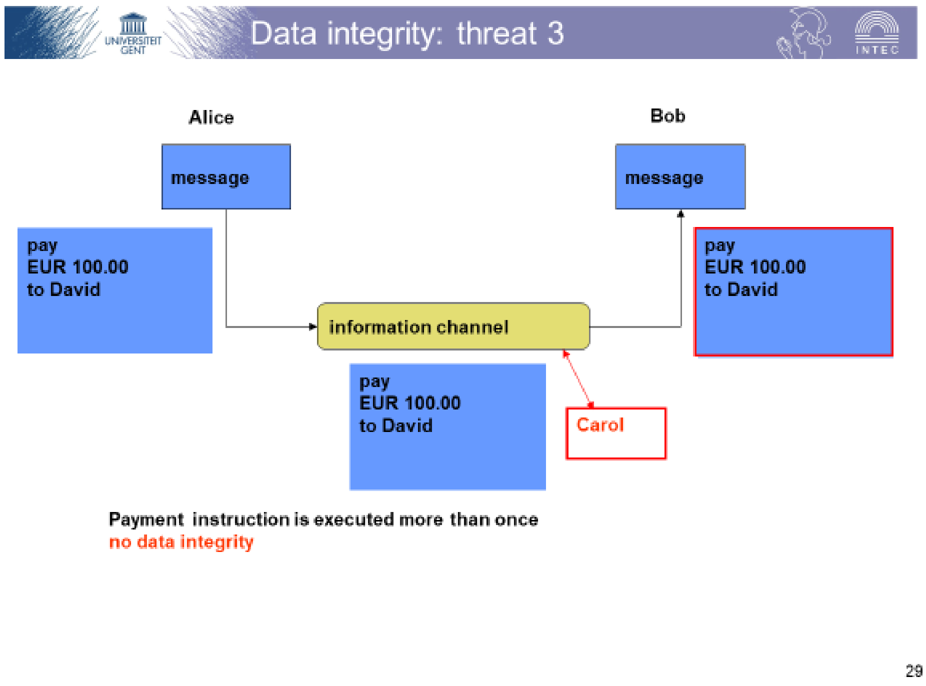
E.g. Alice sends message to bank Bob instructing to pay EUR 100.00 to David. Alice would like that this exact amount is paid to D(avid)…and that no other payments are stealthily executed.

Active attack by Carol: injection of an additional message.



E.g. Alice sends message to bank Bob instructing to pay EUR 100.00 to David. Alice would like that this exact amount is paid to D(avid)…and not to some other person.

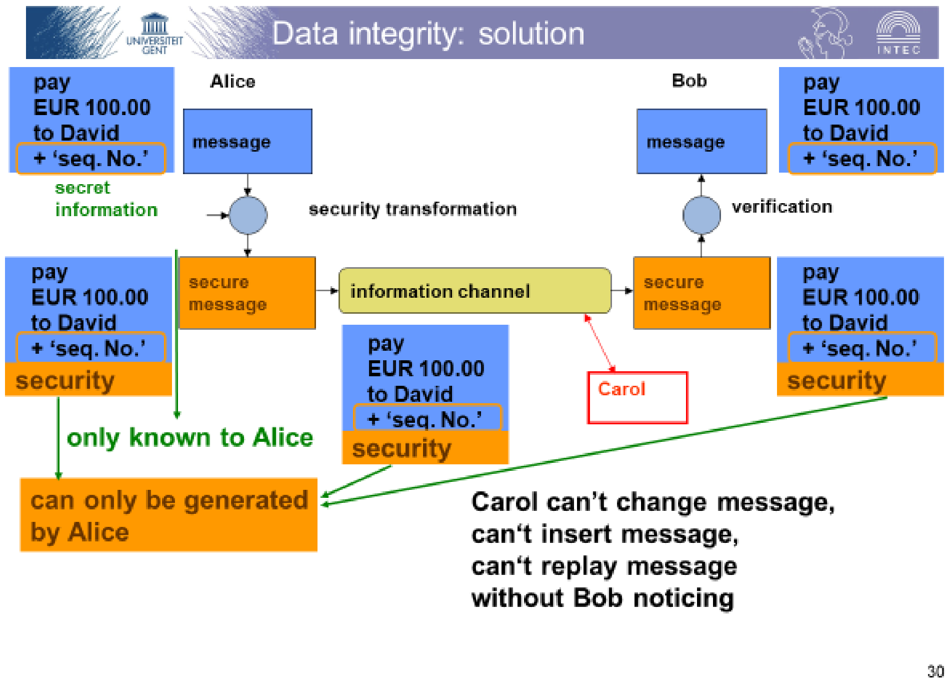
Active attack by Carol: message modification.



Data integrity: threat 3

Payment instruction is executed more than once
no data integrity

E.g. Alice sends message to bank Bob instructing to pay EUR 100.00 to David. Alice would like that this exact amount is paid to D(avid)…and doesn't want a double execution of the payment.
Active attack by Carol: fraudulent reuse of data ("replay").


UNIVERSITEIT GENT
INTEC
pay EUR 100.00 to David + 'seq. No.'
Alice
message
Bob
message
pay EUR 100.00 to David + 'seq. No.'
secret information
security transformation
verification
pay EUR 100.00 to David + 'seq. No.' security
secure message
information channel
secure message
pay EUR 100.00 to David + 'seq. No.' security
pay EUR 100.00 to David + 'seq. No.' security
Carol
only known to Alice
can only be generated by Alice
30

Data integrity: solution

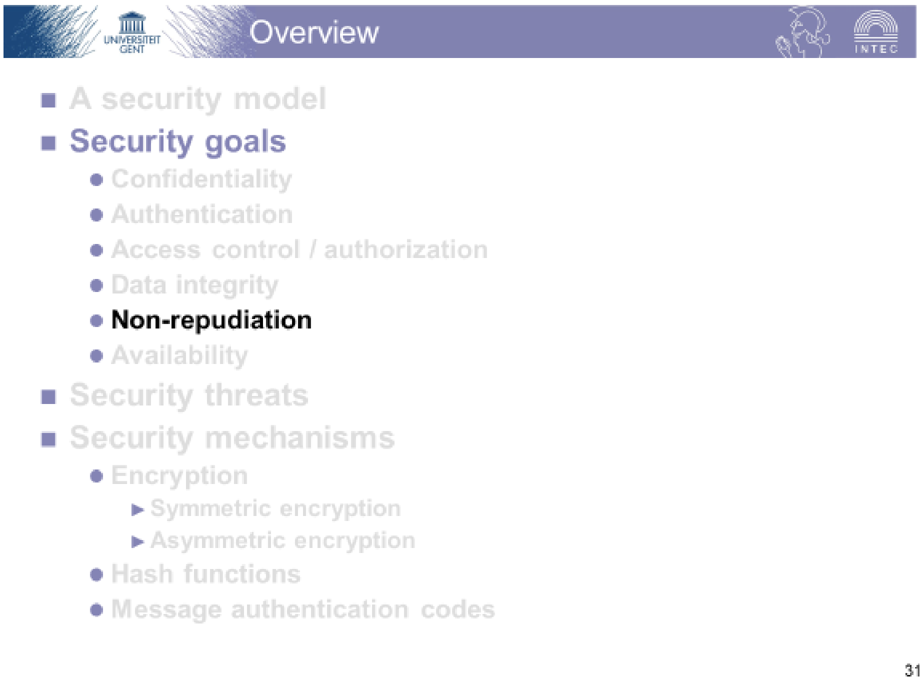
Carol can't change message,
can't insert message,
can't replay message
without Bob noticing

In this case, a digital signature can be a solution. This signature makes it impossible for Carol surreptitiously to modify a message or to insert a message that hasn't been created by Alice.

The sequence number added to the communication aims at thwarting a "replay". In practice, this won't be a simple serial number (e.g. 0,1,2,etc.), but it will satisfy some security requirements (to avoid the replay of an old sequence number in a later session). A possible choice may be the use of a (sufficiently large) "nonce", which is incremented for each subsequent message. Another possibility is to use a time value instead of a regular sequence number (but beware of desynchronised clocks).

Some use modes of symmetric encryption may also thwart replay attacks (see later).

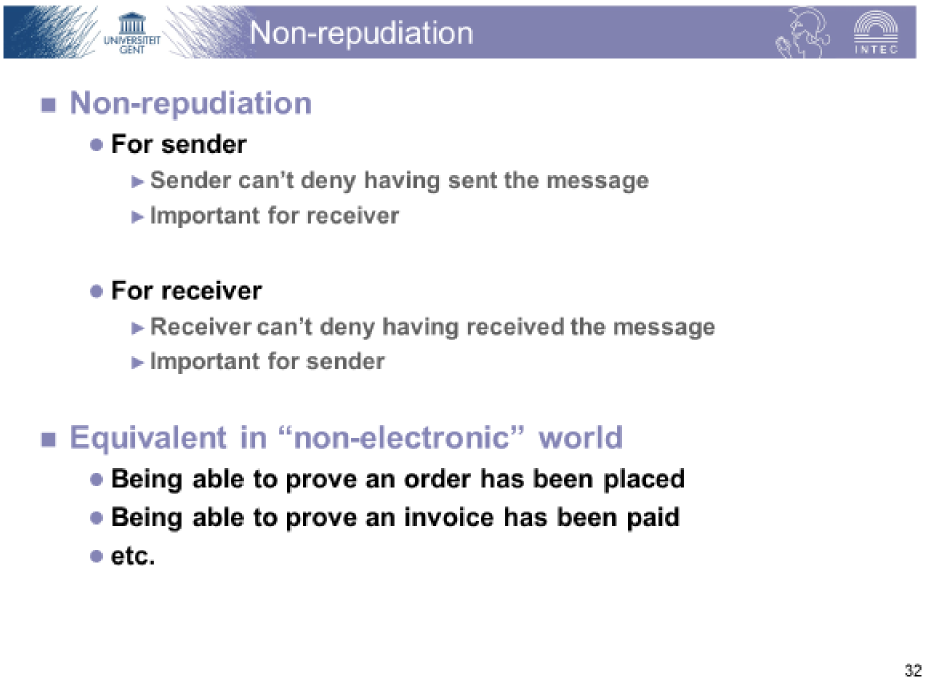
Observe that this schema doesn't guarantee full data integrity, as Carol could still cause the message **not** to reach Bob. This aspect of data integrity can only be achieved with a bi-directional communication between Alice and Bob (receipt acknowledgement).



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - **Non-repudiation**
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

31



Non-repudiation

- Non-repudiation
  - **For sender**
    - Sender can't deny having sent the message
    - Important for receiver
  - **For receiver**
    - Receiver can't deny having received the message
    - Important for sender
- Equivalent in "non-electronic" world
  - **Being able to prove an order has been placed**
  - **Being able to prove an invoice has been paid**
  - **etc.**

32

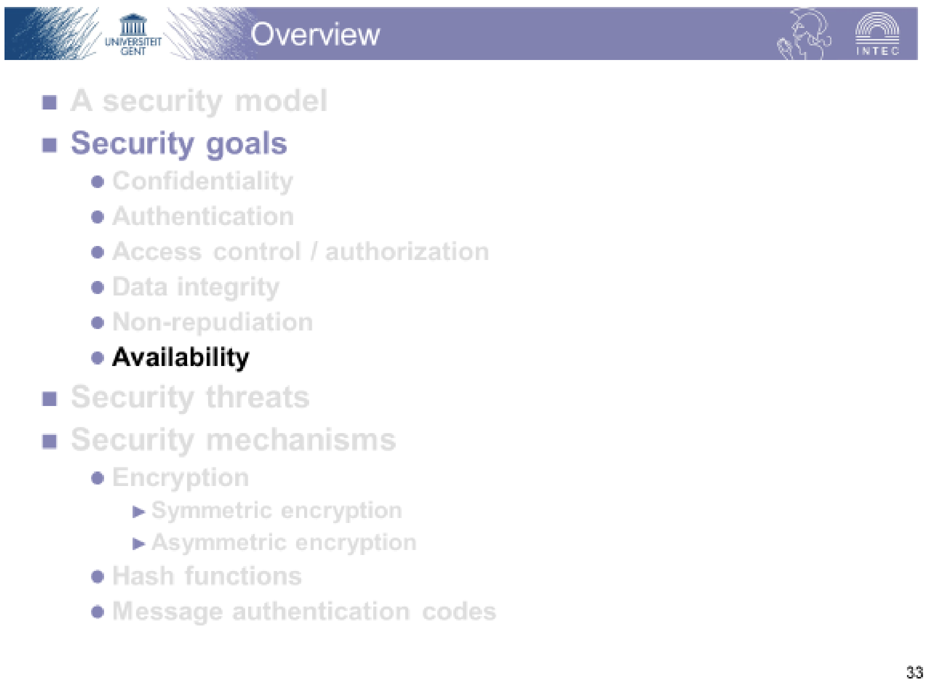
Repudiation is translated in Dutch as "verwerping", "ontkenning" or "verloochening".

In order to achieve non-repudiation (for the sender) for a transaction, the receiver should be able to prove that the sender did send this message. This implies measures have been taken to ensure an intruder couldn't have sent the message, i.e.:

- Sender authentication
- Data integrity of the communication between sender and receiver
- The receiver will have to keep the "signature" of the message

The schema shown for data integrity would achieve these goals.

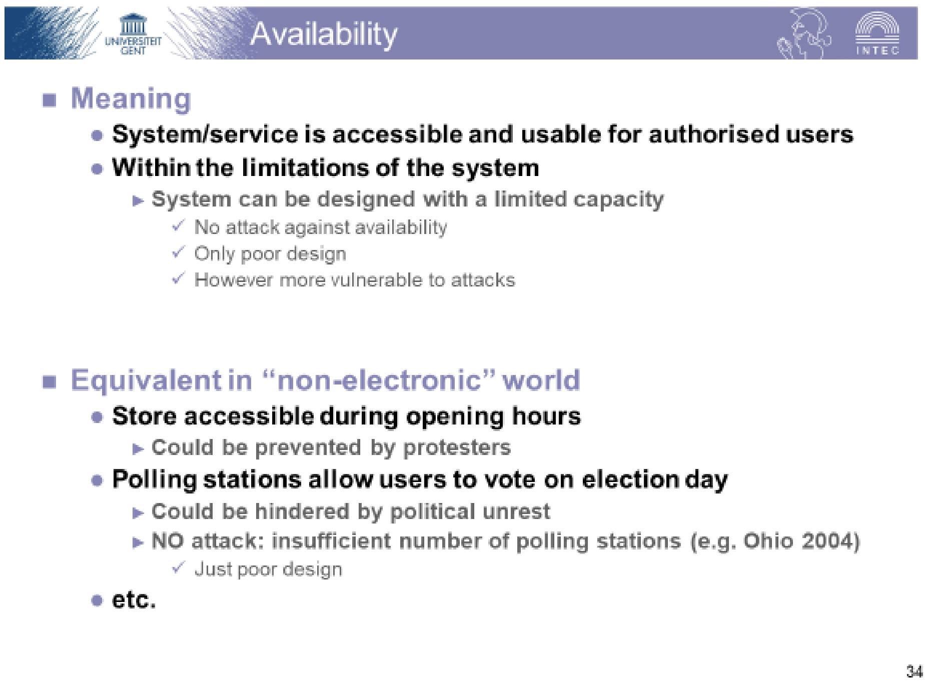
Non-repudiation for the receiver is harder to achieve. It requires a reaction from the receiver to the sender. The non-repudiation of this reaction guarantees the desired functionality.



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - **Availability**
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

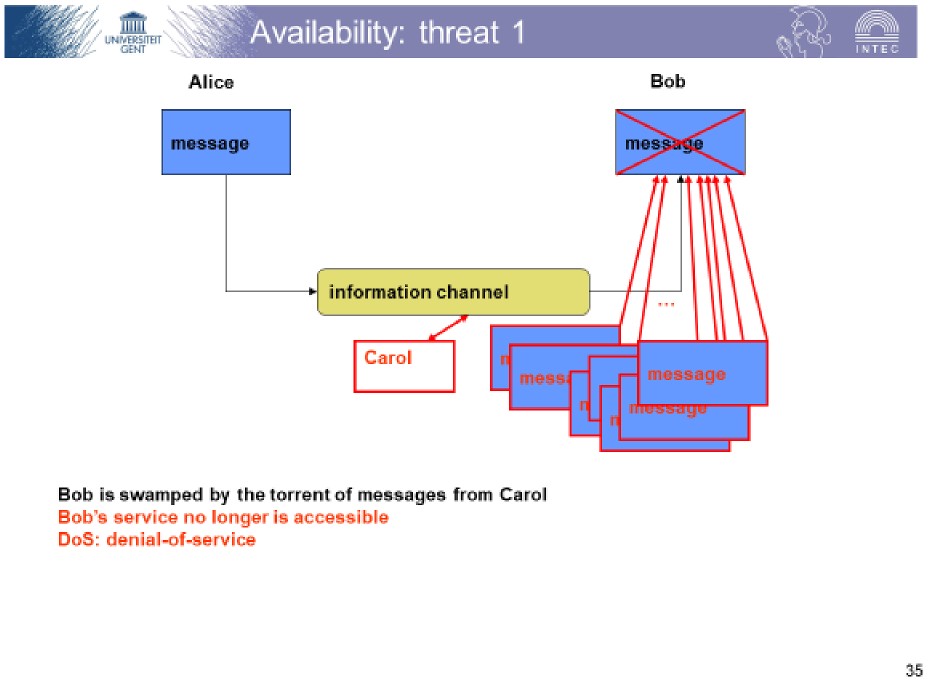
33



Availability

- Meaning
  - **System/service is accessible and usable for authorised users**
  - **Within the limitations of the system**
    - System can be designed with a limited capacity
      - ✓ No attack against availability
      - ✓ Only poor design
      - ✓ However more vulnerable to attacks
- Equivalent in "non-electronic" world
  - **Store accessible during opening hours**
    - Could be prevented by protesters
  - **Polling stations allow users to vote on election day**
    - Could be hindered by political unrest
    - NO attack: insufficient number of polling stations (e.g. Ohio 2004)
      - ✓ Just poor design
  - **etc.**

34



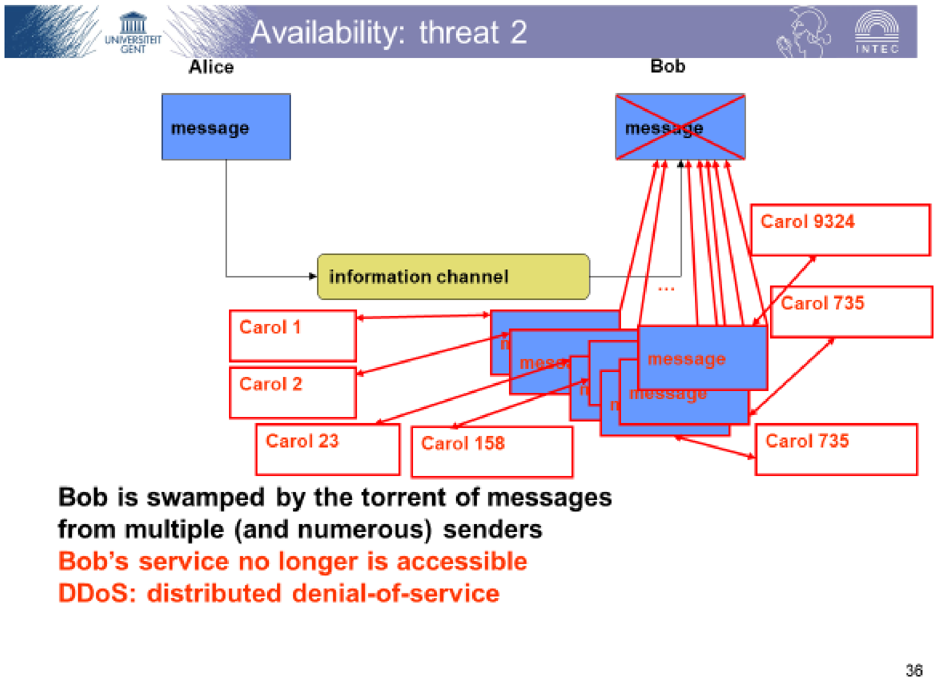
Availability: threat 1

Bob is swamped by the torrent of messages from Carol
Bob's service no longer is accessible
DoS: denial-of-service

35

E.g. Alice requests information from website Bob. Under normal circumstances Bob should be able to receive, process, and answer Alice's message.

Carol executes an active attack: denial-of-service. This requires true broadband access (better than DSL or cable; university networks are sometimes used as launch pads for this kind of attacks). Authentication of communicating parties makes such an attack a lot harder (although the initial phase of the authentication protocol could still be attacked).



Availability: threat 2

Bob is swamped by the torrent of messages from multiple (and numerous) senders
Bob's service no longer is accessible
DDoS: distributed denial-of-service

36

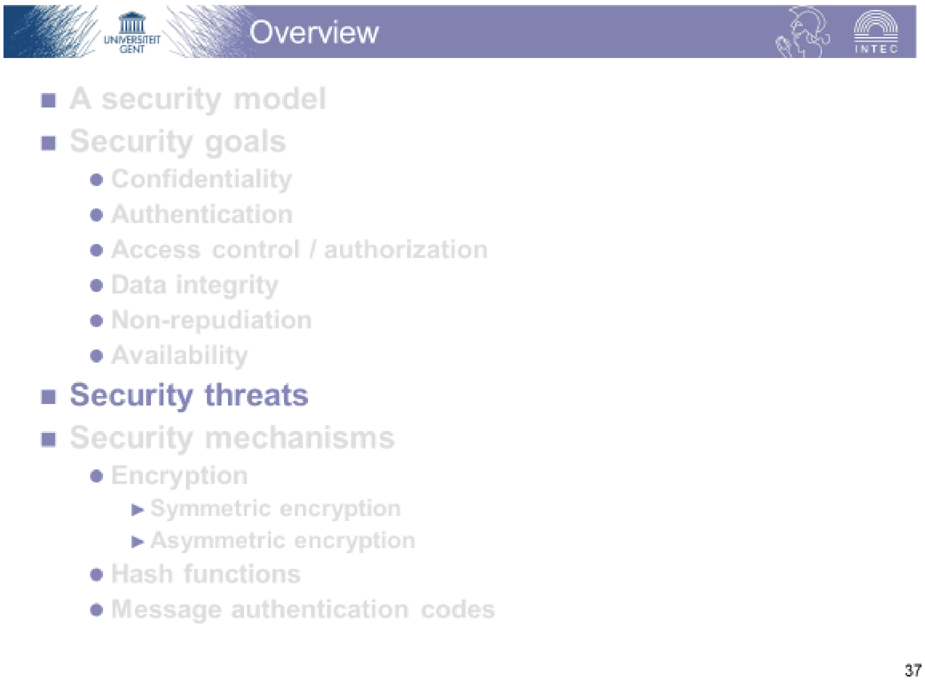
E.g. Alice requests information from website Bob. Under normal circumstances Bob should be able to receive, process, and answer Alice's message.

"Carols" execute an active attack: distributed denial-of-service. A large number of attackers Carol 1, Carol 2, etc. overwhelm Bob's website under a torrent of messages. These messages may come from:

- A large group of participants
- A large group of computers contaminated by malware, where unsuspecting users are deployed for the attack (e.g. MyDoom.A worm, which performed a successful DoS attack against the SCO website)

Broadband access isn't required for this kind of attack (regular DSL or cable will be largely sufficient). This attack is much harder to fend off than regular DoS.

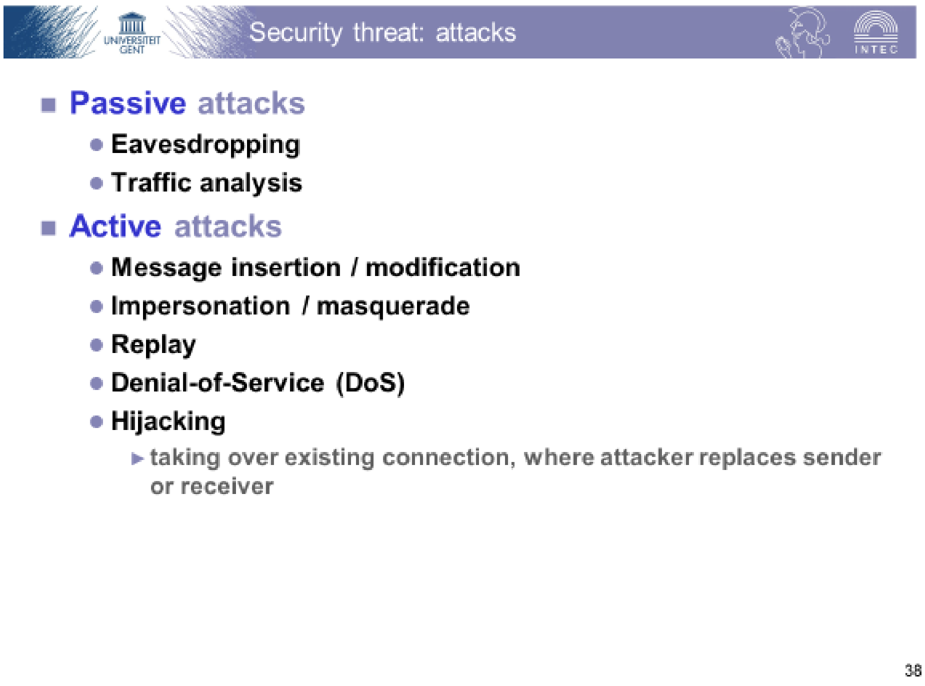
Practical solutions against DDoS typically involve non-cryptographic techniques (firewall, IDS, etc.) that analyse incoming communications so that anomalies can be detected and dealt with (filtering out the anomalous traffic). If needed, it may be desirable to operate a temporary graceful shutdown of the service (which is still to be preferred over a system crash, possibly involving some data loss).



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

37

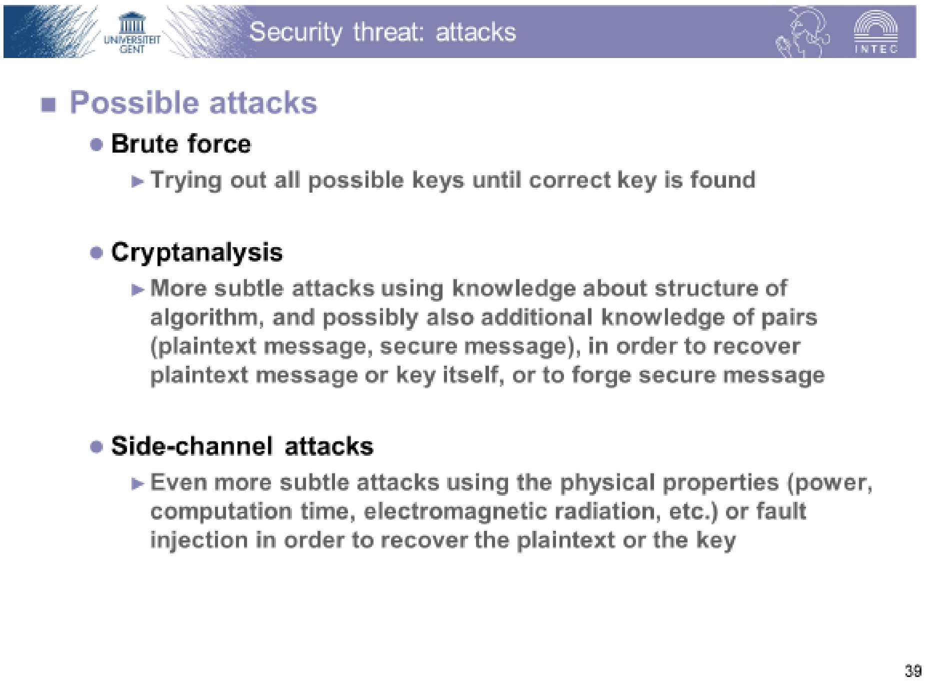


Security threat: attacks

- Passive attacks
  - Eavesdropping
  - Traffic analysis
- Active attacks
  - Message insertion / modification
  - Impersonation / masquerade
  - Replay
  - Denial-of-Service (DoS)
  - Hijacking
    - taking over existing connection, where attacker replaces sender or receiver

38

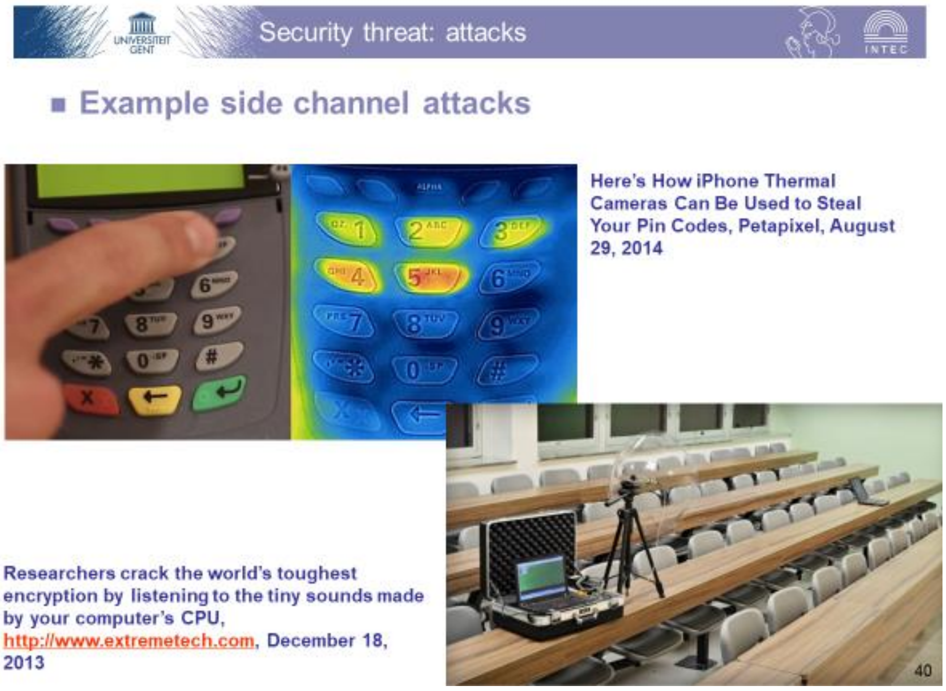
Passive attacks offer a more limited potential to the attacker, but they are significantly harder to detect, as they don't modify the transmitted data. This is certainly true if part of the communication channel is totally open (e.g. a wireless connection). Active attacks do modify the transmitted data or the data stream, and can thus be detected more easily. However, they give the attacker a lot more options.



Security threat: attacks

- Possible attacks
  - Brute force
    - Trying out all possible keys until correct key is found
  - Cryptanalysis
    - More subtle attacks using knowledge about structure of algorithm, and possibly also additional knowledge of pairs (plaintext message, secure message), in order to recover plaintext message or key itself, or to forge secure message
  - Side-channel attacks
    - Even more subtle attacks using the physical properties (power, computation time, electromagnetic radiation, etc.) or fault injection in order to recover the plaintext or the key

39



Security threat: attacks

- Example side channel attacks

Here's How iPhone Thermal Cameras Can Be Used to Steal Your Pin Codes, Petapixel, August 29, 2014

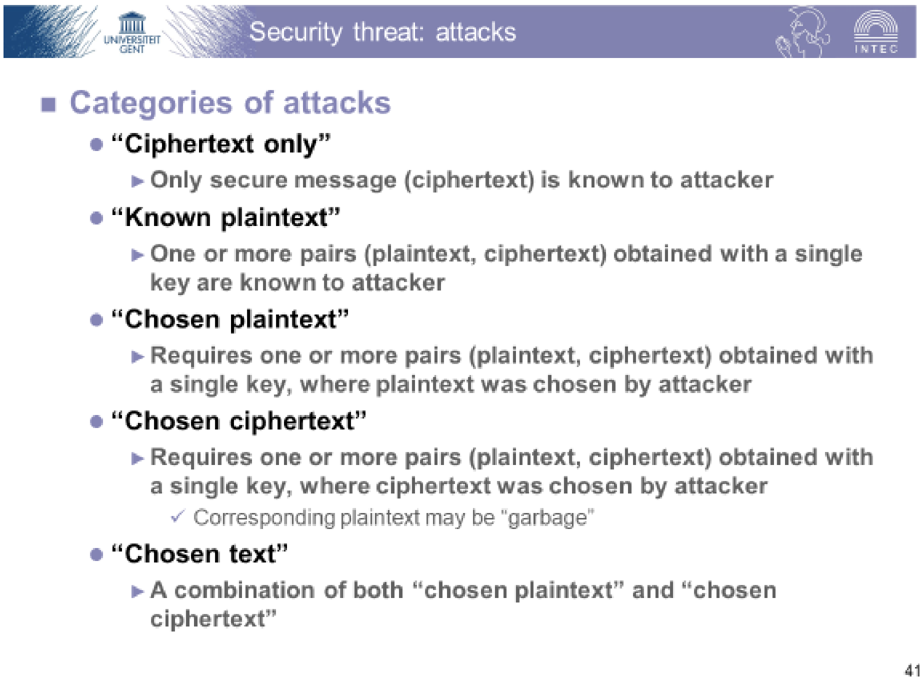
Researchers crack the world's toughest encryption by listening to the tiny sounds made by your computer's CPU, http://www.extremetech.com, December 18, 2013

40

Timing information, power consumption, electromagnetic leaks or even sound can provide an extra source of information, which can be exploited to break the system.

http://petapixel.com/2014/08/29/heres-iphone-thermal-cameras-can-used-steal-pin-codes/

http://www.extremetech.com/extreme/173108-researchers-crack-the-worlds-toughest-encryption-by-listening-to-the-tiny-sounds-made-by-your-computers-cpu



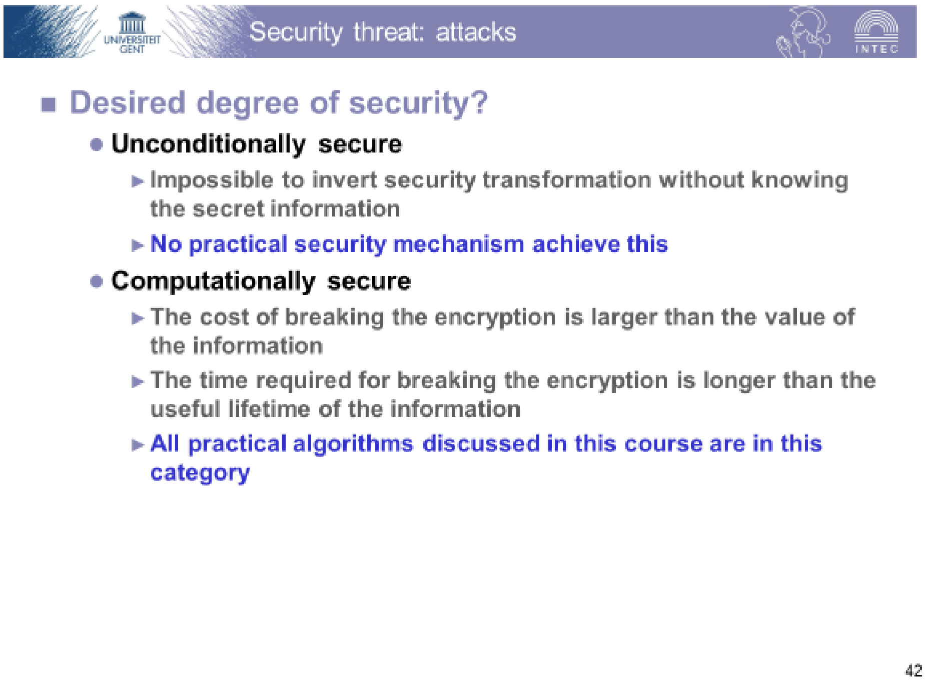
Security threat: attacks

- Categories of attacks
  - "Ciphertext only"
    - Only secure message (ciphertext) is known to attacker
  - "Known plaintext"
    - One or more pairs (plaintext, ciphertext) obtained with a single key are known to attacker
  - "Chosen plaintext"
    - Requires one or more pairs (plaintext, ciphertext) obtained with a single key, where plaintext was chosen by attacker
  - "Chosen ciphertext"
    - Requires one or more pairs (plaintext, ciphertext) obtained with a single key, where ciphertext was chosen by attacker
      - ✓ Corresponding plaintext may be "garbage"
  - "Chosen text"
    - A combination of both "chosen plaintext" and "chosen ciphertext"

41

The term ciphertext (or secure message) typically refers to the encrypted message, whereas the term plaintext is used for the original text. The same terminology is used for text or other data structures, since any digital data source can be represented as plaintext. “Ciphertext only” attacks can be mounted using information that is readily available to the attacker (it only requires eavesdropping upon the information channel). It is however much harder to perform successful cryptanalysis with a “ciphertext only” attack. Conversely, it may be very hard to mount a “chosen text” attack as finding the desired (plaintext, ciphertext) pairs is typically hard. However, it offers more opportunities for cryptanalysis.

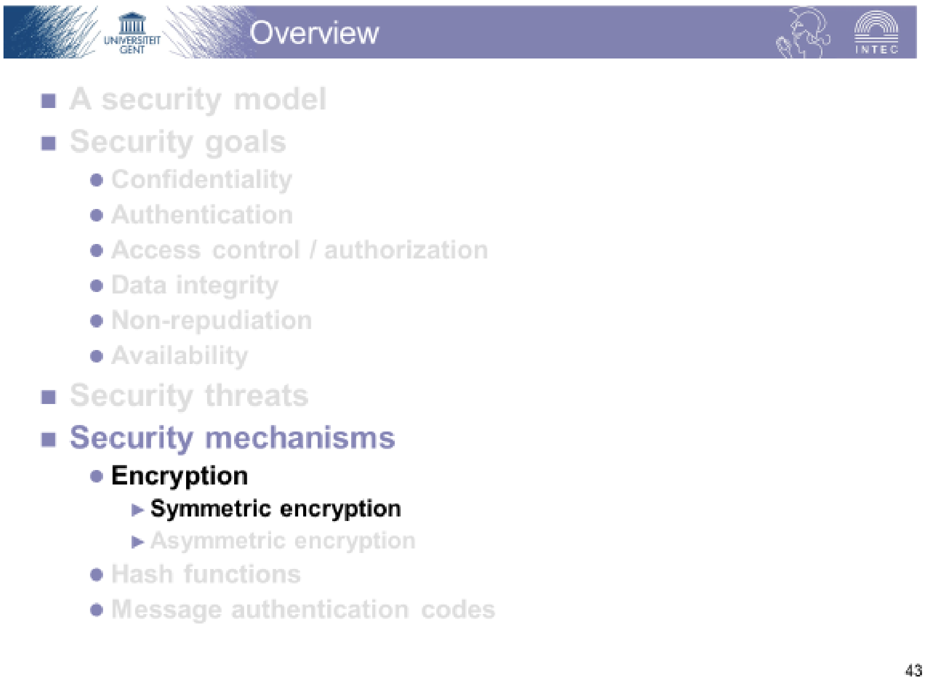
Some algorithms are very safe against “ciphertext only” attacks, but won’t withstand other types of attacks. This may be acceptable if the algorithm is used correctly. Other algorithms will even withstand a “chose text” attack. This is something to bear in mind when considering security mechanisms.



Security threat: attacks

- Desired degree of security?
  - Unconditionally secure
    - Impossible to invert security transformation without knowing the secret information
    - No practical security mechanism achieve this
  - Computationally secure
    - The cost of breaking the encryption is larger than the value of the information
    - The time required for breaking the encryption is longer than the useful lifetime of the information
    - All practical algorithms discussed in this course are in this category

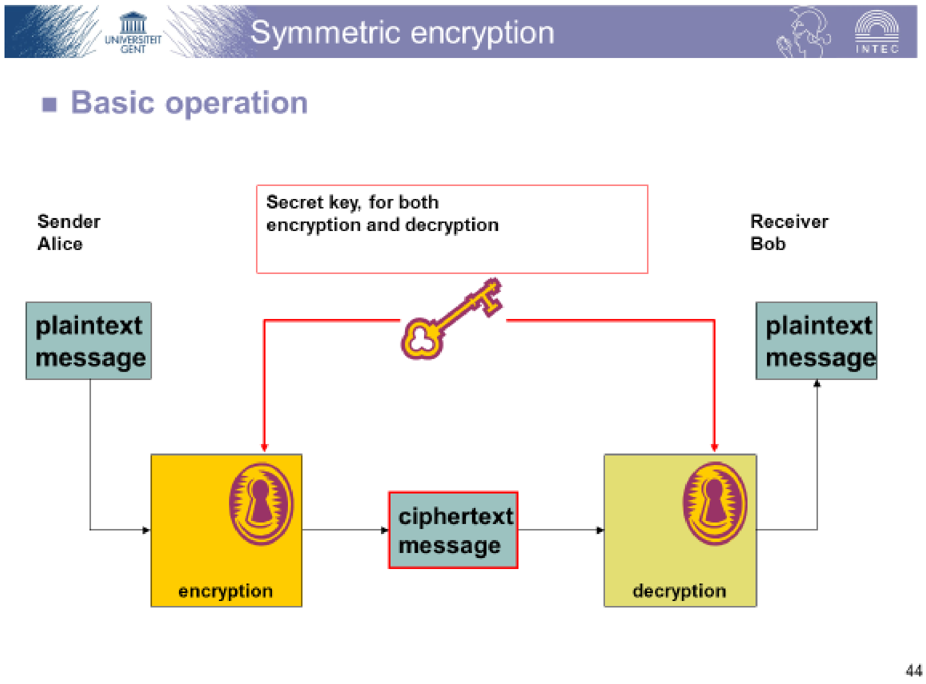
42



Overview

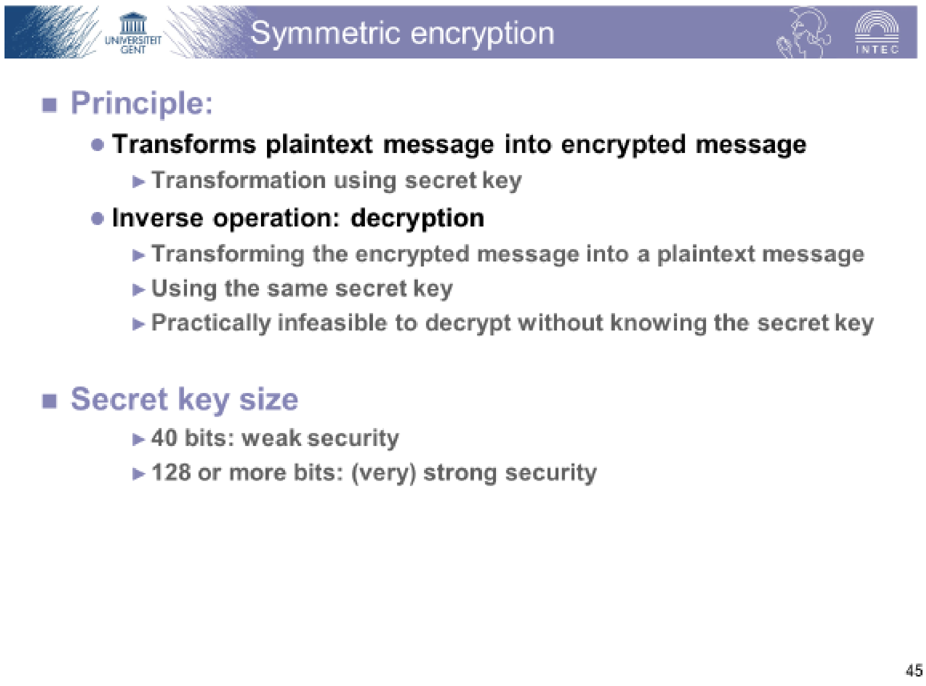
- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - **Encryption**
    - **Symmetric encryption**
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

43



Symmetric encryption

- Basic operation

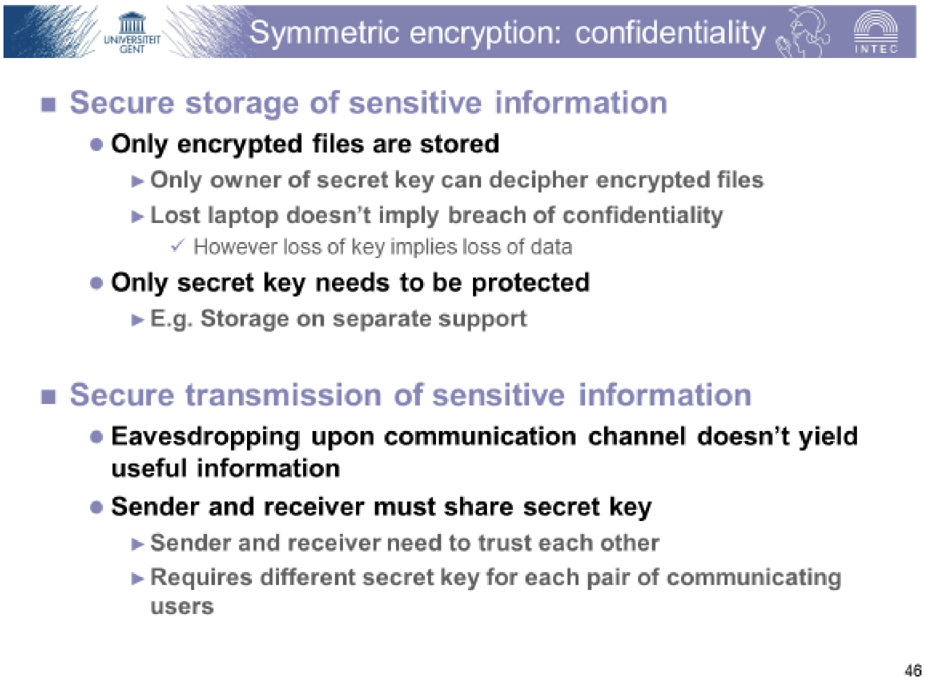


Symmetric encryption

- **Principle:**
  - **Transforms plaintext message into encrypted message**
    - Transformation using secret key
  - **Inverse operation: decryption**
    - Transforming the encrypted message into a plaintext message
    - Using the same secret key
    - Practically infeasible to decrypt without knowing the secret key
- **Secret key size**
    - 40 bits: weak security
    - 128 or more bits: (very) strong security

45

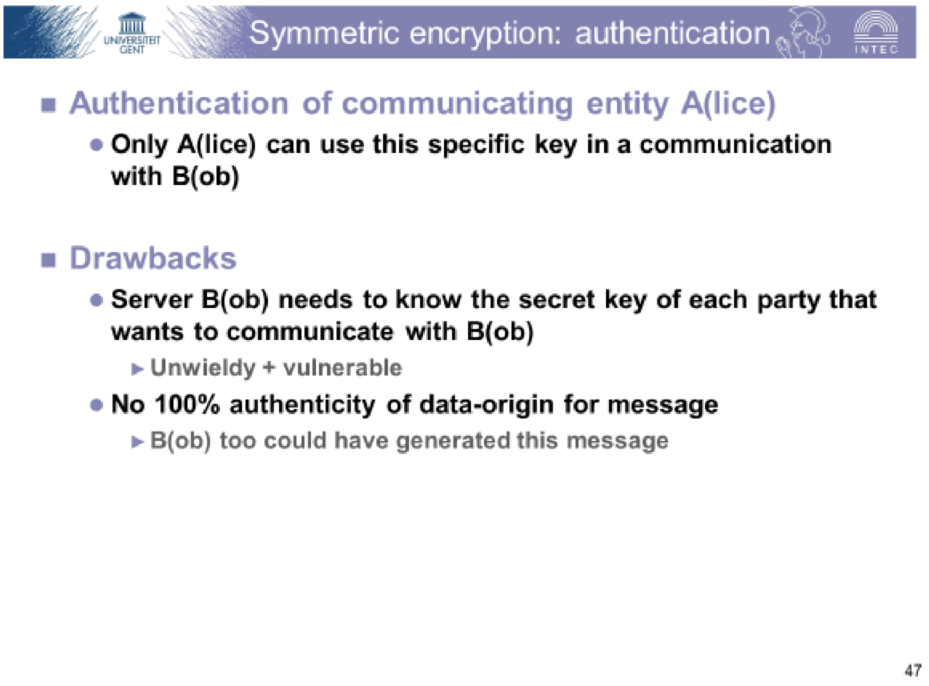
Symmetric encryption is the most traditional form of encryption and is also referred to as “conventional encryption”.



Symmetric encryption: confidentiality

- Secure storage of sensitive information
  - Only encrypted files are stored
    - Only owner of secret key can decipher encrypted files
    - Lost laptop doesn't imply breach of confidentiality
      - ✓ However loss of key implies loss of data
  - Only secret key needs to be protected
    - E.g. Storage on separate support
- Secure transmission of sensitive information
  - Eavesdropping upon communication channel doesn't yield useful information
  - Sender and receiver must share secret key
    - Sender and receiver need to trust each other
    - Requires different secret key for each pair of communicating users

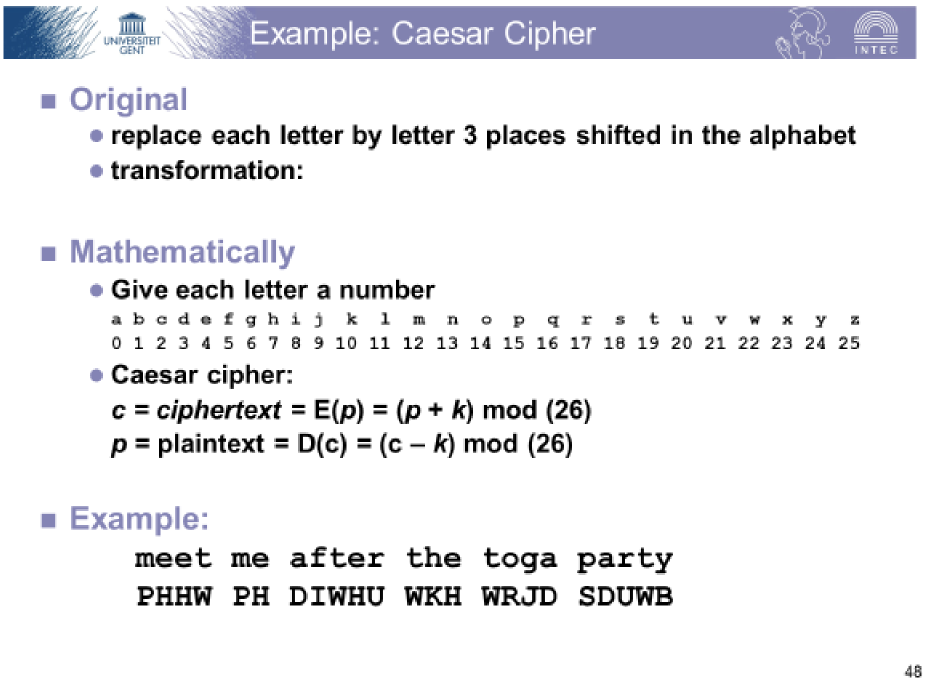
46



Symmetric encryption: authentication

- Authentication of communicating entity A(lice)
  - Only A(lice) can use this specific key in a communication with B(ob)
- Drawbacks
  - Server B(ob) needs to know the secret key of each party that wants to communicate with B(ob)
    - Unwieldy + vulnerable
  - No 100% authenticity of data-origin for message
    - B(ob) too could have generated this message

47



Example: Caesar Cipher

- Original
  - replace each letter by letter 3 places shifted in the alphabet
  - transformation:
- Mathematically
  - Give each letter a number

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

  - Caesar cipher:

    c = *ciphertext* = E(p) = (p + k) mod (26)

    p = plaintext = D(c) = (c – k) mod (26)
- Example:

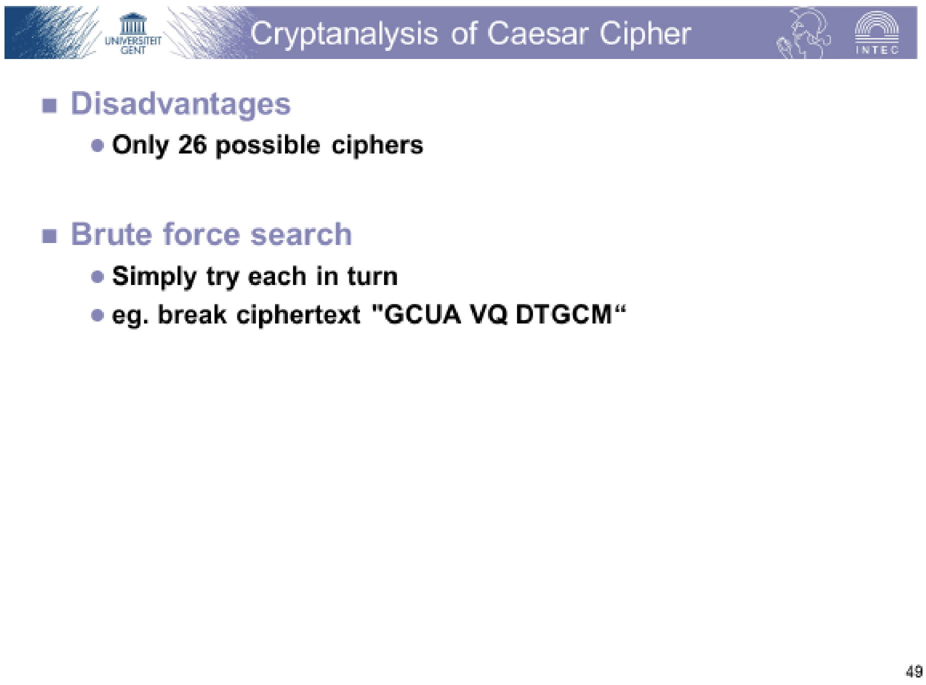
```
meet me after the toga party
PHHW PH DIWHU WKH WRJD SDUWB
```

48

Substitution ciphers are encryption methods where letters of plaintext are replaced by other letters or by numbers or symbols. Whilst the early Greeks described several substitution ciphers, the first attested use in military affairs of one was by Julius Caesar, described by him in *Gallic Wars*. We call any cipher using a simple letter shift a **caesar cipher**, not just those with shift 3 as originally used.

This mathematical description uses **modulo (clock) arithmetic**. Here, when you reach Z you go back to A and start again. Mod 26 implies that when you reach 26, you use 0 instead (i.e. the letter after Z, or 25 + 1 goes to A or 0).

Example: howdy (7,14,22,3,24) encrypted using key *f* (i.e. a shift of 5) is MTBID



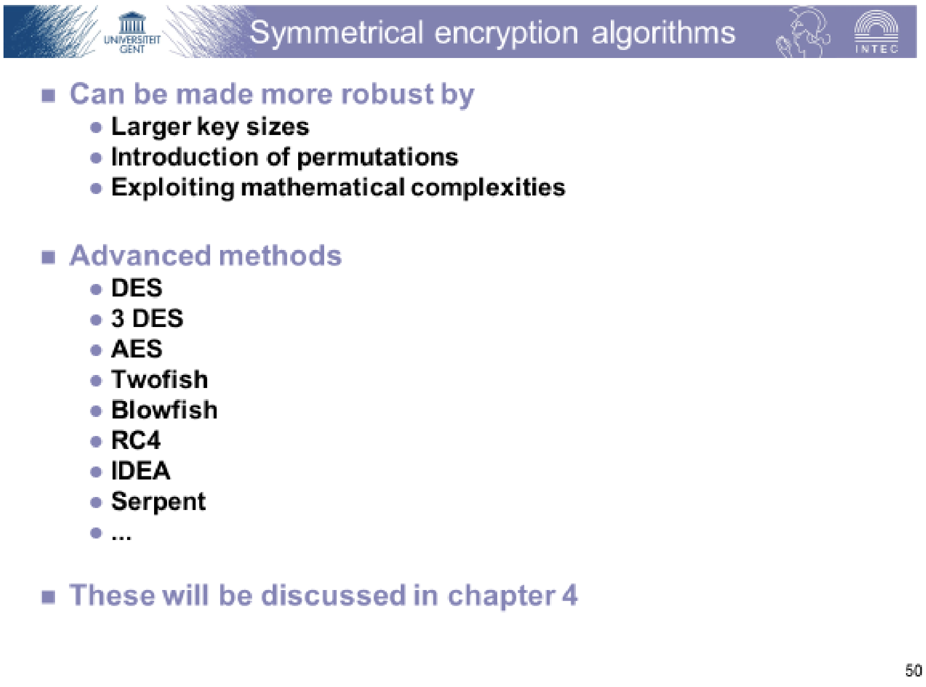
Cryptanalysis of Caesar Cipher

- Disadvantages
  - Only 26 possible ciphers
- Brute force search
  - Simply try each in turn
  - eg. break ciphertext "GCUA VQ DTGCM“

49

With a Caesar cipher, there are only 26 possible keys, of which only 25 are of any use, since mapping A to A etc doesn't really obscure the message! An attacker can try each of the keys (shifts) in turn, until can recognise the original message.

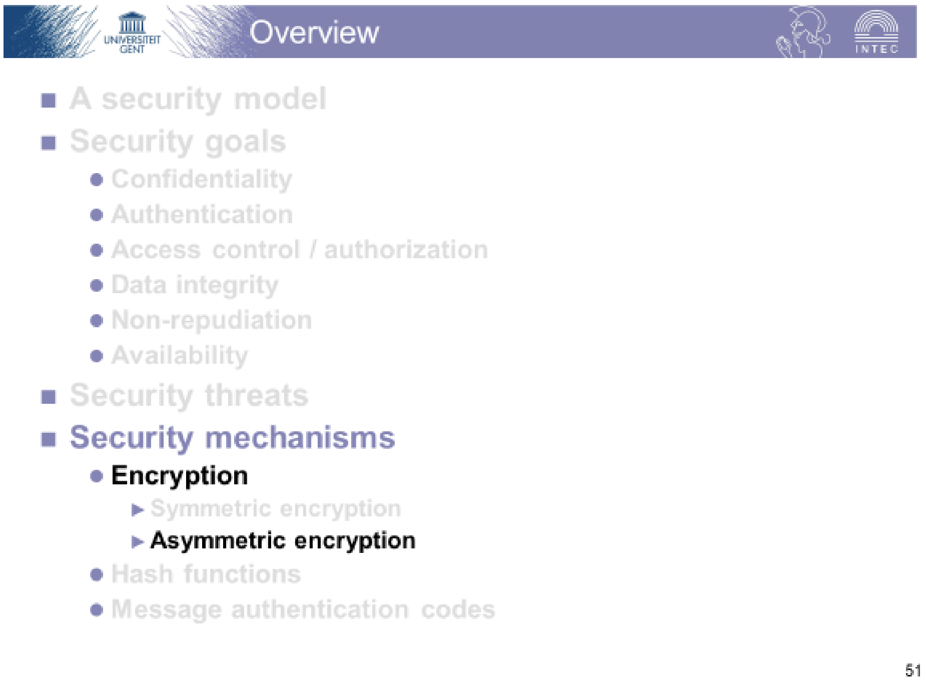
Note: as mentioned before, you need to be able to **recognise** when have an original message (i.e. is it English). Usually easy for humans, hard for computers. Visually recognizing the presence of an original message is much harder when compressed data is used.



Symmetrical encryption algorithms

- Can be made more robust by
  - Larger key sizes
  - Introduction of permutations
  - Exploiting mathematical complexities
- Advanced methods
  - DES
  - 3 DES
  - AES
  - Twofish
  - Blowfish
  - RC4
  - IDEA
  - Serpent
  - ...
- These will be discussed in chapter 4

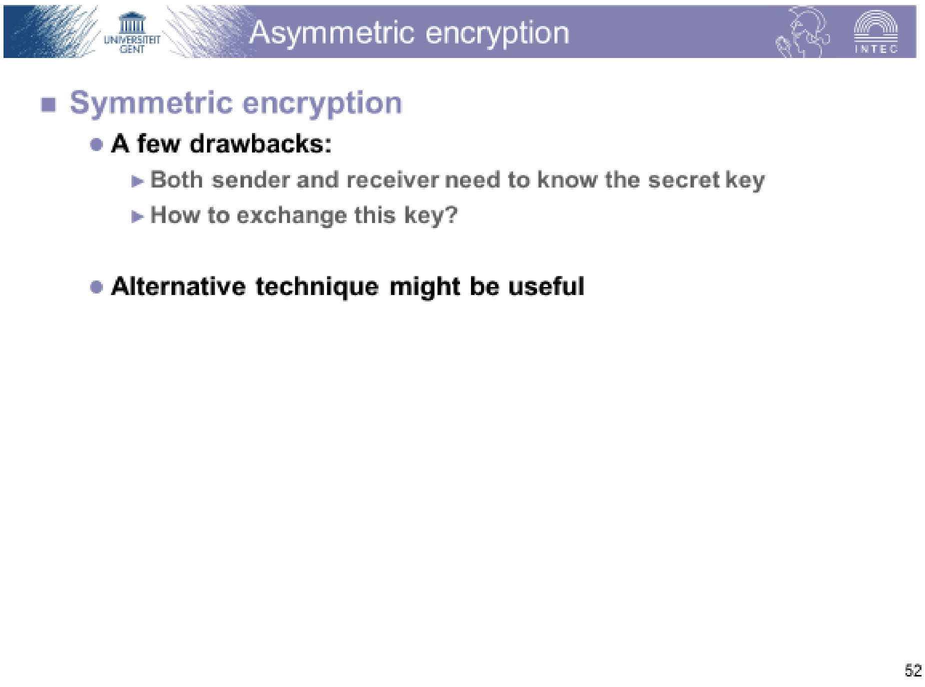
50



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - Message authentication codes

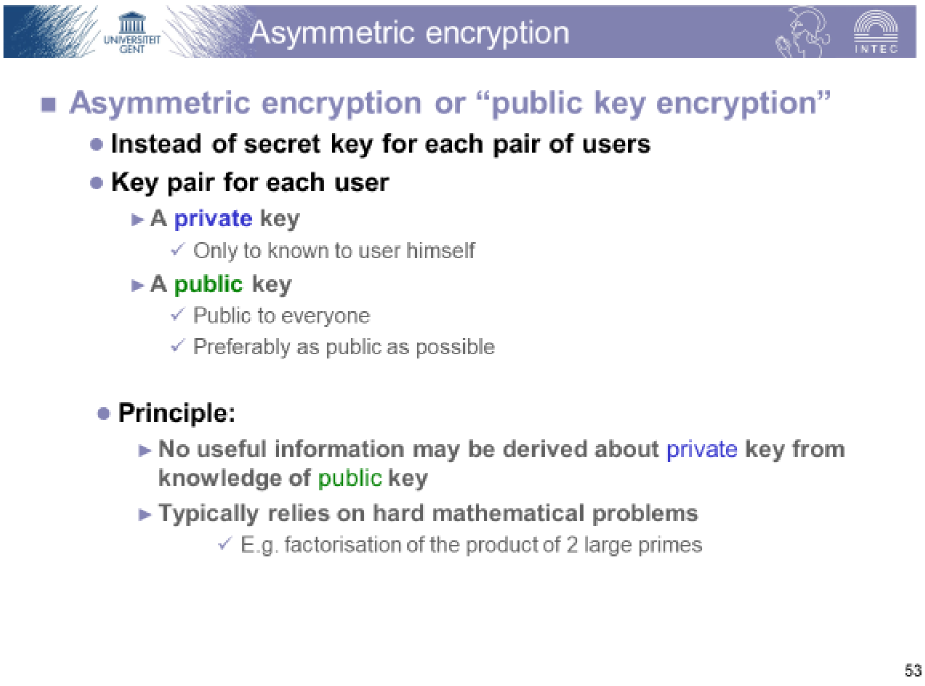
51



Asymmetric encryption

- Symmetric encryption
  - A few drawbacks:
    - Both sender and receiver need to know the secret key
    - How to exchange this key?
  - Alternative technique might be useful

52



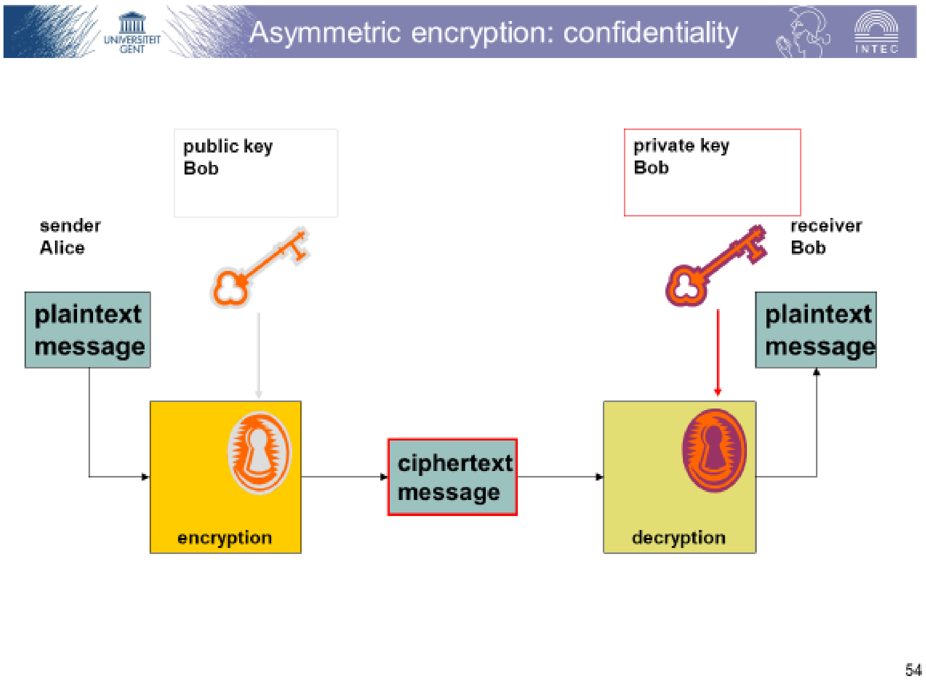
Asymmetric encryption

- Asymmetric encryption or “public key encryption”
  - Instead of secret key for each pair of users
  - Key pair for each user
    - A private key
      - ✓ Only to known to user himself
    - A public key
      - ✓ Public to everyone
      - ✓ Preferably as public as possible
  - Principle:
    - No useful information may be derived about private key from knowledge of public key
    - Typically relies on hard mathematical problems
      - ✓ E.g. factorisation of the product of 2 large primes

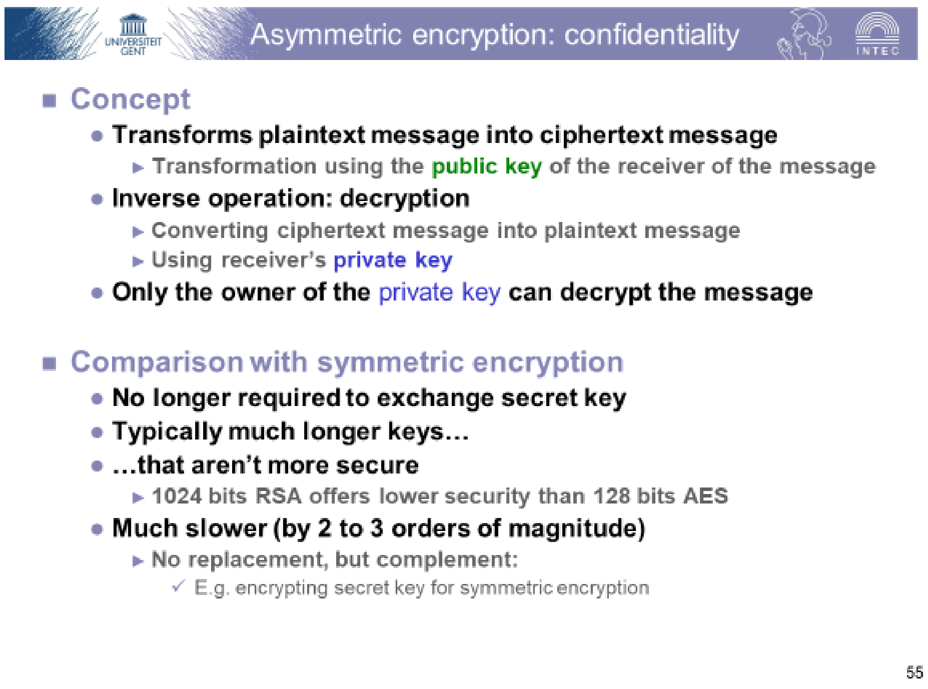
53

In this course we use the expression “**private key**” for the secret information in an **asymmetric** encryption algorithm, and the expression “**secret key**” for the secret information in a **symmetric** encryption algorithm. The sole purpose of this wording is to distinguish between both types of algorithms. However, in literature about security and cryptography one may occasionally encounter the expression “secret key” for asymmetric encryption. The context should make clear whether one is dealing with a symmetric or with an asymmetric algorithm.

A consequence of the reliance on a hard mathematical problem is that the security of the algorithm is threatened not only by increasing computer power, but also by possible improvements in the solution techniques of the mathematical problem. This happened in the 90's with prime factorisation. The complexity of solving this problem was significantly reduced by the introduction of the GNFS and may even be further reduced in the future, hereby threatening RSA for not too long a key length (1024 bits; 768 bits should be considered broken by now).



In a key pair for asymmetric encryption we write KU_A for A's public key van A, and KR_A for A's private key.

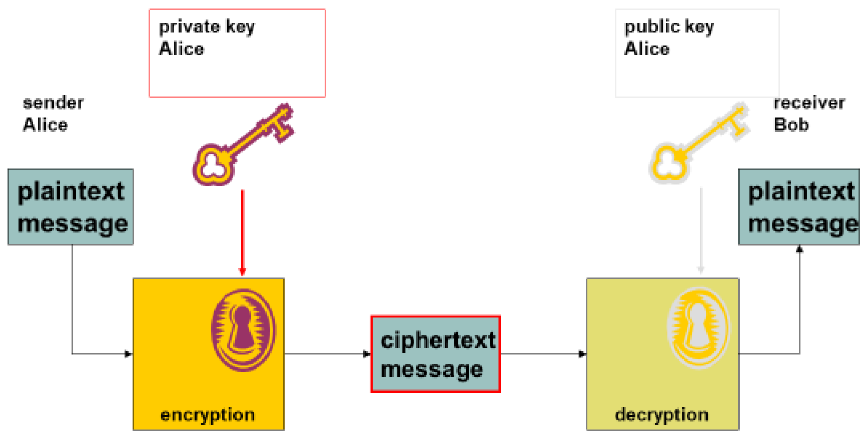


Asymmetric encryption: confidentiality

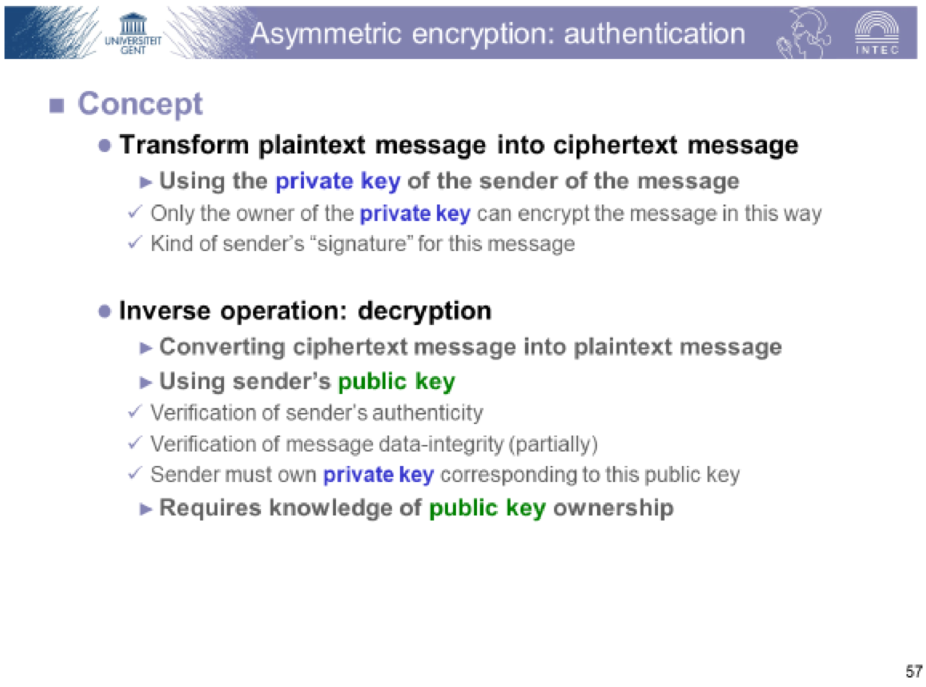
- Concept
  - Transforms plaintext message into ciphertext message
    - Transformation using the public key of the receiver of the message
  - Inverse operation: decryption
    - Converting ciphertext message into plaintext message
    - Using receiver's private key
  - Only the owner of the private key can decrypt the message
- Comparison with symmetric encryption
  - No longer required to exchange secret key
  - Typically much longer keys…
  - …that aren't more secure
    - 1024 bits RSA offers lower security than 128 bits AES
  - Much slower (by 2 to 3 orders of magnitude)
    - No replacement, but complement:
      - ✓ E.g. encrypting secret key for symmetric encryption

55

Asymmetric encryption: authentication



56



Asymmetric encryption: authentication

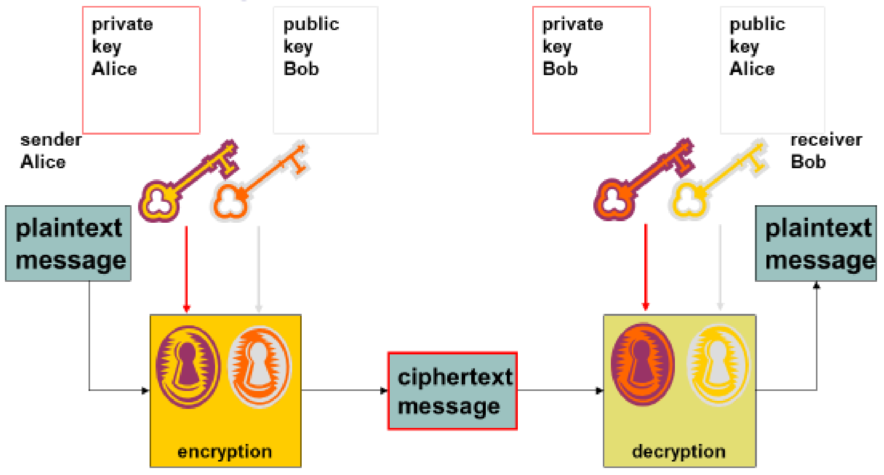
- Concept
  - Transform plaintext message into ciphertext message
    - Using the **private key** of the sender of the message
      - ✓ Only the owner of the **private key** can encrypt the message in this way
      - ✓ Kind of sender's "signature" for this message
  - Inverse operation: decryption
    - Converting ciphertext message into plaintext message
    - Using sender's **public key**
      - ✓ Verification of sender's authenticity
      - ✓ Verification of message data-integrity (partially)
      - ✓ Sender must own **private key** corresponding to this public key
    - Requires knowledge of **public key** ownership

57

Data-integrity is only partially achieved. It is indeed impossible for an attacker to modify the message, but without additional measures, the attacker could still replay or suppress the message. This is sometimes called **message authentication**.

Asymmetric encryption

Confidentiality + authentication

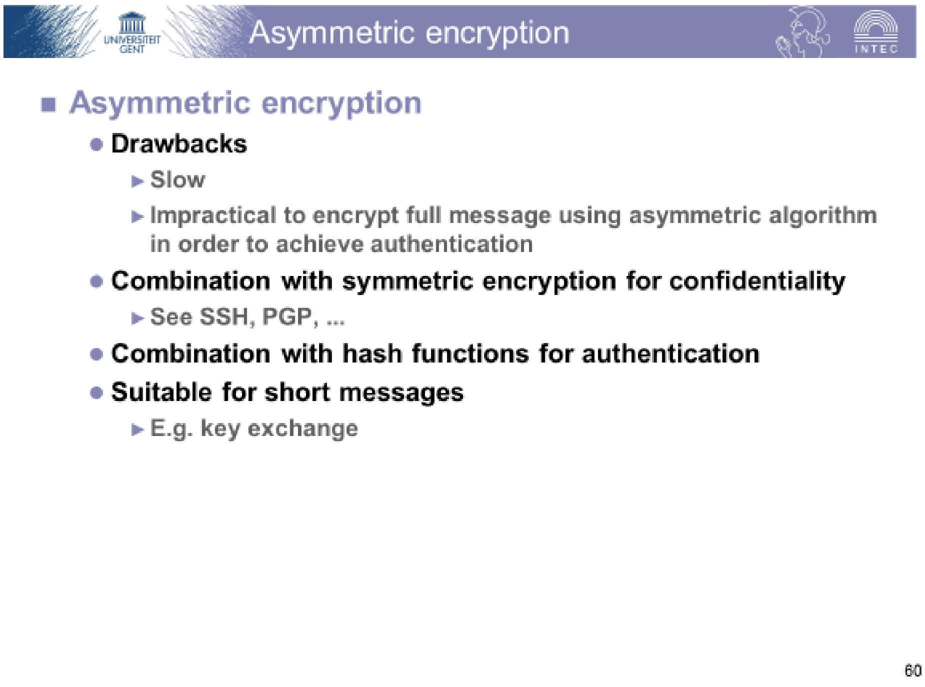


58

Asymmetric encryption

- Combining confidentiality + authentication
  - Only sender A(lice) can send message encrypted using A(lice)'s private key
    - Authenticity of entity A(lice)
    - Data-integrity of message (partially)
    - To be verified using A(lice)'s **public key**
  - Only receiver B(ob) can decrypt message that was encrypted using B(ob)'s public key
    - Confidentiality of communication between A(lice) and B(ob)
  - Requires that each party knows the public key of the other party

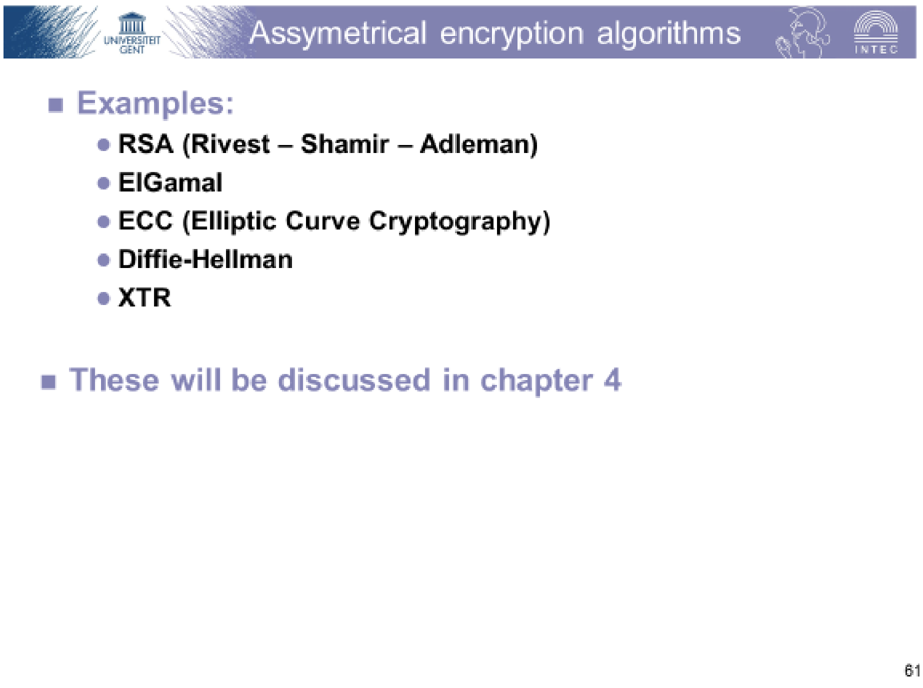
59



Asymmetric encryption

- Asymmetric encryption
  - Drawbacks
    - Slow
    - Impractical to encrypt full message using asymmetric algorithm in order to achieve authentication
  - Combination with symmetric encryption for confidentiality
    - See SSH, PGP, ...
  - Combination with hash functions for authentication
  - Suitable for short messages
    - E.g. key exchange

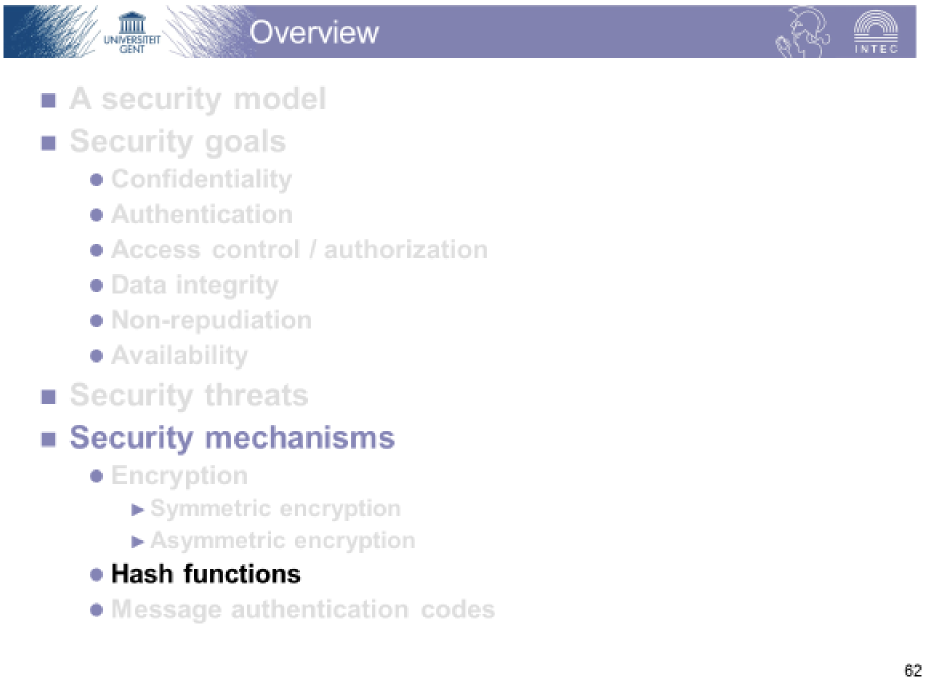
60



Assymetrical encryption algorithms

- Examples:
  - RSA (Rivest – Shamir – Adleman)
  - ElGamal
  - ECC (Elliptic Curve Cryptography)
  - Diffie-Hellman
  - XTR
- These will be discussed in chapter 4

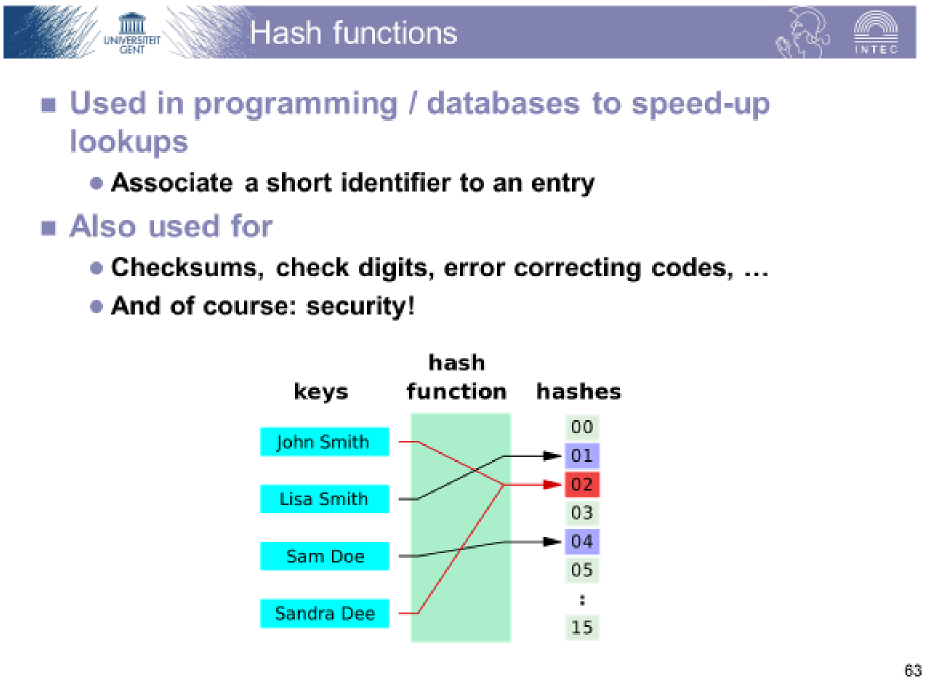
61



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - **Hash functions**
  - Message authentication codes

62



Hash functions

- Used in programming / databases to speed-up lookups
  - Associate a short identifier to an entry
- Also used for
  - Checksums, check digits, error correcting codes, …
  - And of course: security!

keys
hash function
hashes
John Smith
Lisa Smith
Sam Doe
Sandra Dee
00
01
02
03
04
05
:
15

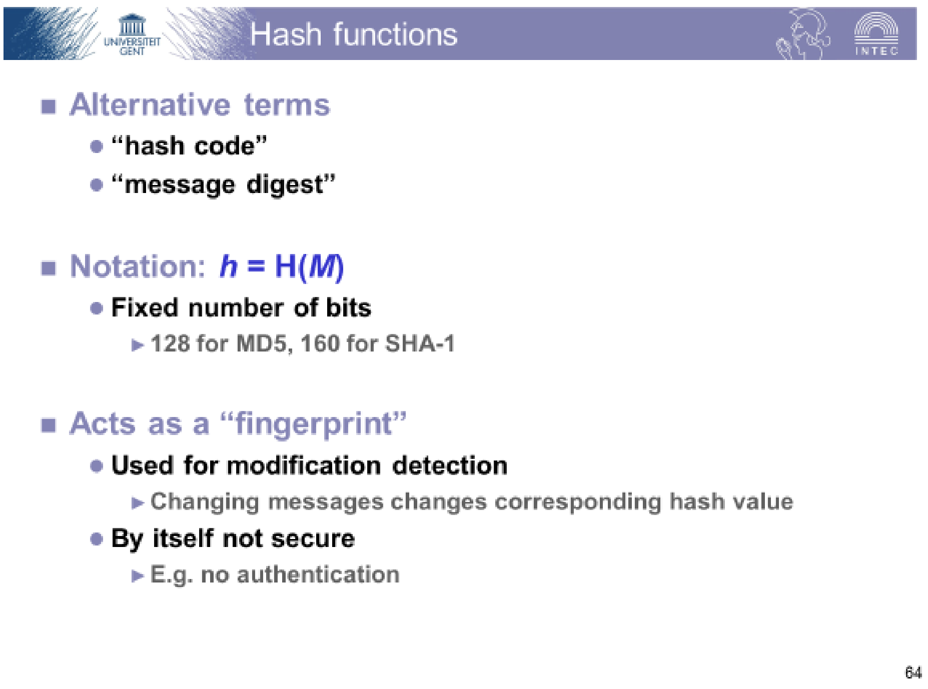
63

Hash functions are primarily used in hash tables, to quickly locate a data record (e.g., a dictionary definition) given its search key (the headword). Specifically, the hash function is used to map the search key to an index; the index gives the place in the hash table where the corresponding record should be stored. Typically, the domain of a hash function (the set of possible keys) is larger than its range (the number of different table indexes), and so it will map several different keys to the same index. Therefore, each slot of a hash table is associated with (implicitly or explicitly) a set of records, rather than a single record. For this reason, each slot of a hash table is often called a bucket, and hash values are also called bucket indices. Thus, the hash function only hints at the record's location — it tells where one should start looking for it. Still, in a half-full table, a good hash function will typically narrow the search down to only one or two entries. The figure demonstrates a hash function that maps names to integers from 0 to 15. There is a collision between keys "John Smith" and "Sandra Dee".

A **checksum or hash sum** is a small-size datum from a block of digital data for the purpose of detecting errors which may have been introduced during its transmission or storage. It is usually applied to an installation file after it is received from the download server. By themselves checksums are often used to verify data integrity, but should not be relied upon to also verify data authenticity.

A **check digit** is a form of redundancy check used for error detection on identification numbers (e.g. bank account numbers) which have been input manually. It is analogous to a binary parity bit used to check for errors in computer-generated data. It consists of a single digit (sometimes more than one) computed by an algorithm from the other digits (or letters) in the sequence input.

Forward error correction (FEC) or channel coding are techniques used for controlling errors in data transmission over unreliable or noisy communication channels. The central idea is the sender encodes the message in a redundant way by using an error-correcting code (ECC). The redundancy allows the receiver to detect a limited number of errors that may occur anywhere in the message, and often to correct these errors without retransmission.

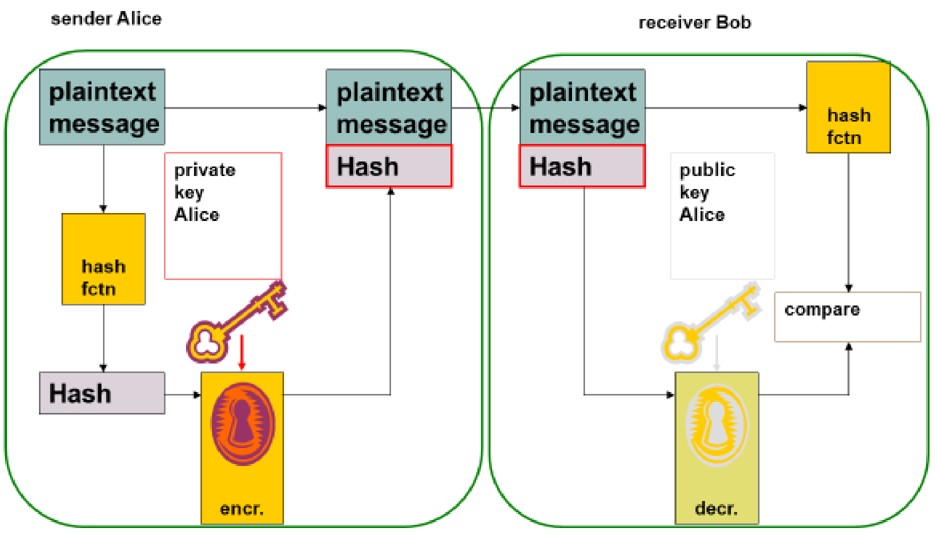


Hash functions

- Alternative terms
  - "hash code"
  - "message digest"
- Notation: h = H(M)
  - Fixed number of bits
    - 128 for MD5, 160 for SHA-1
- Acts as a "fingerprint"
  - Used for modification detection
    - Changing messages changes corresponding hash value
  - By itself not secure
    - E.g. no authentication

64

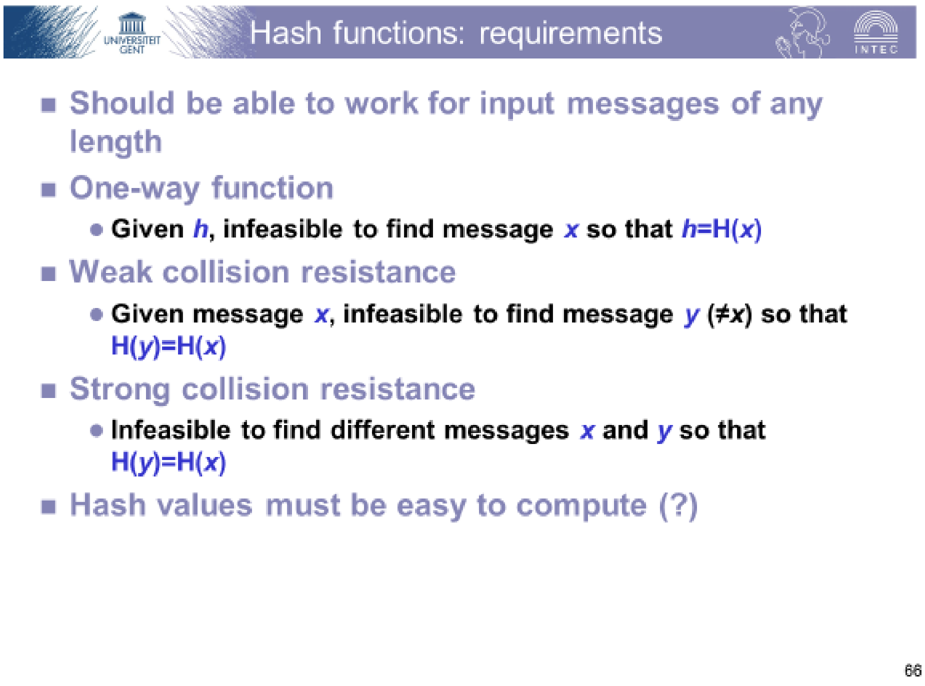
Hash functions



65

In this use of a hash function for message authentication, an asymmetric encryption algorithm is used (to encrypt the non-secret hash value). Authentication of communicating entity A(lice) is provided, since only A(lice) can use this particular key in communication with B(ob). Authentication of the sent message is provided, since a modified message would correspond to different (encrypted) hash value. To further ensure that message tampering is not possible, a counter sequence is typically added to the message and encrypted together with the hash value.

The encrypted hash value is the “digital signature” of the corresponding plaintext message. This is one of the most common schemes for digital signatures (see later), where RSA or ElGamal can be used as an encryption algorithm.



Hash functions: requirements

- Should be able to work for input messages of any length
- One-way function
  - Given *h*, infeasible to find message *x* so that *h*=H(*x*)
- Weak collision resistance
  - Given message *x*, infeasible to find message *y* (≠*x*) so that H(*y*)=H(*x*)
- Strong collision resistance
  - Infeasible to find different messages *x* and *y* so that H(*y*)=H(*x*)
- Hash values must be easy to compute (?)

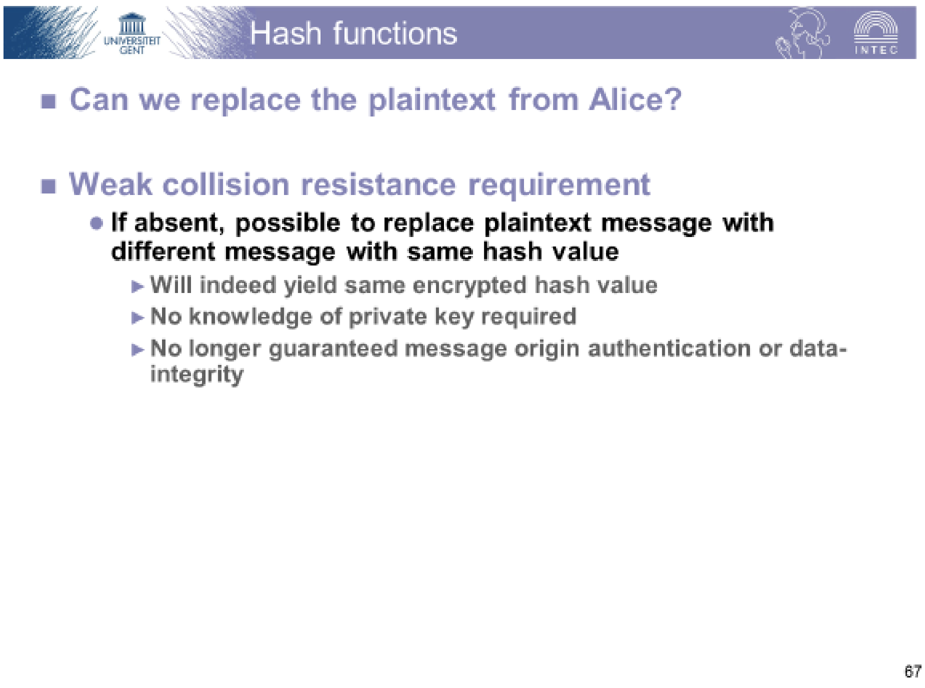
66

The main goal of the “one-way function” property is to make it impossible to recover the original message from the hash value only. This is generally not a major issue as a hash function typically maps a huge message space on a much smaller hash value space. This is however essential when the message space is rather small (e.g. when hashing passwords).

We’ll see why weak and strong collision resistance are important when dealing with the generation of digital signatures. Both forms of collision resistance requirement exclude too simple and too naive hash function implementations (e.g. using XOR operations).

Strong collision resistance isn’t required for all applications.

Note that the requirement “easy to compute” is not always valid. When encrypting passwords, special hash algorithms or “key derivation functions” (such as bcrypt, scrypt or PBKDF) are used that require more time to compute. This has the advantage that attackers that somehow gain access to the hashed passwords can’t too easily calculate which hash code is derived from each password. Algorithms such as bcrypt include adaptive functionality: over time, the variables such as the iteration count can be increased to make it slower in order to keep the algorithm resistant to brute-force search attacks even with increasing computation power.

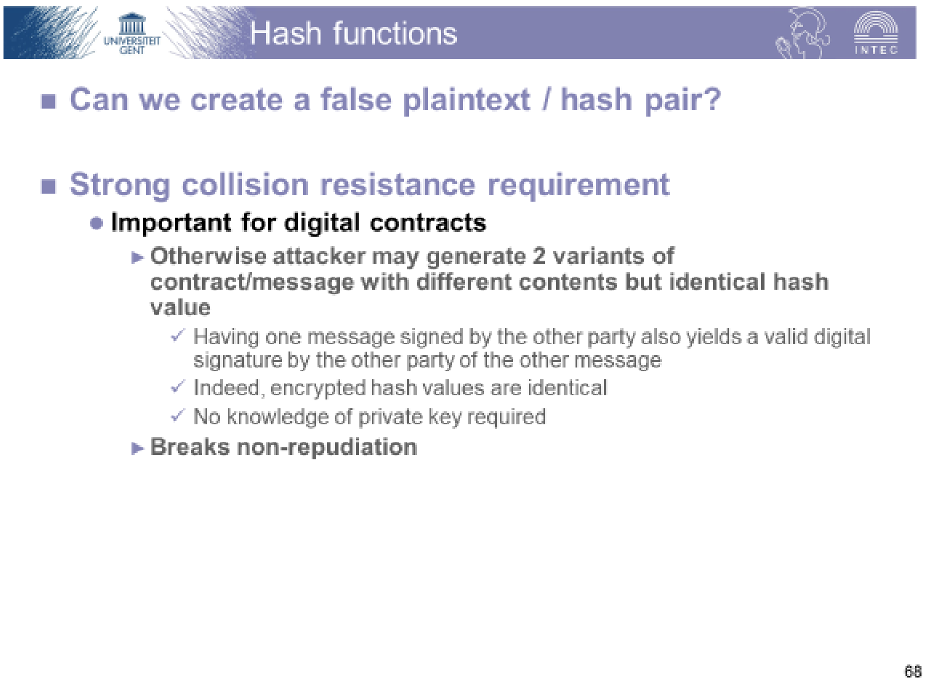


Hash functions

- Can we replace the plaintext from Alice?
- Weak collision resistance requirement
  - If absent, possible to replace plaintext message with different message with same hash value
    - Will indeed yield same encrypted hash value
    - No knowledge of private key required
    - No longer guaranteed message origin authentication or data-integrity

67

For a good hash function with n bits hash value, generating a forged message should be hard, typically in the order of 2^n. I.e. it should require a brute force attack in which almost all possible plaintext messages should be generated before a plaintext message is found with the same hash value.



Hash functions

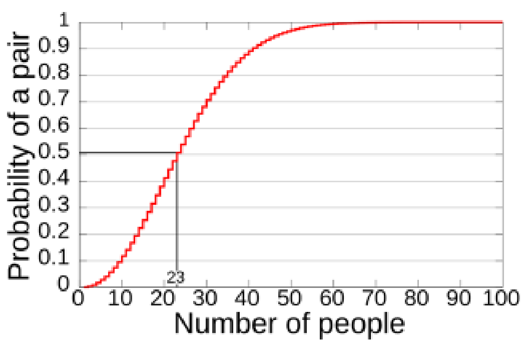
- Can we create a false plaintext / hash pair?
- Strong collision resistance requirement
  - Important for digital contracts
    - Otherwise attacker may generate 2 variants of contract/message with different contents but identical hash value
      - ✓ Having one message signed by the other party also yields a valid digital signature by the other party of the other message
      - ✓ Indeed, encrypted hash values are identical
      - ✓ No knowledge of private key required
    - Breaks non-repudiation

68

For a good hash function with n bits hash value, generating a forged message should be hard, typically in the order of $2^{n/2}$. I.e. it should require a brute force attack in which $2^{n/2}$ combinations are generated before two plaintext / hash pairs are found with the same hash values.

Hash functions

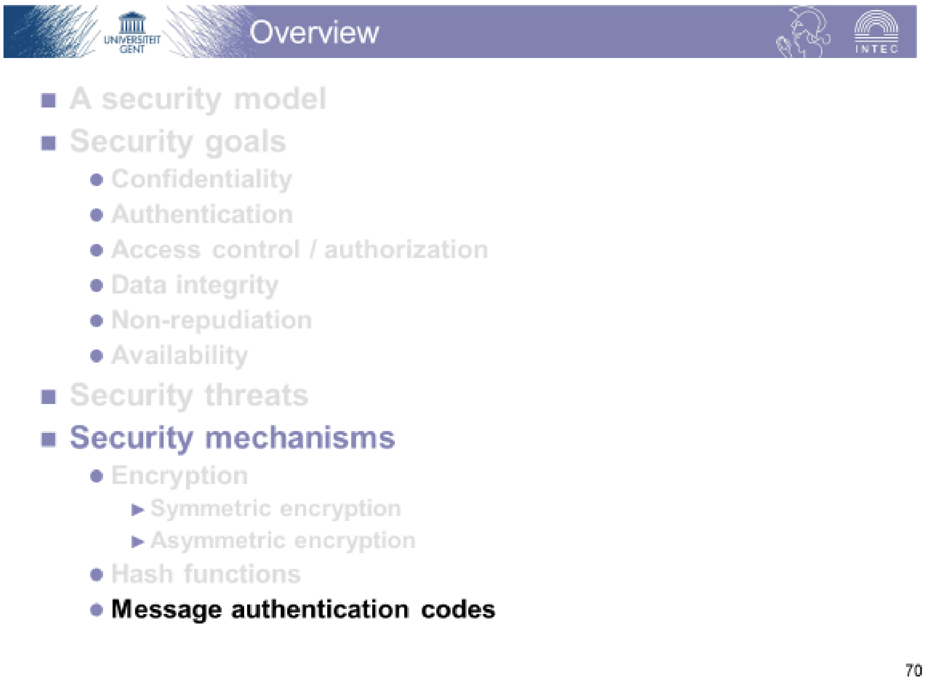
- Problem with strong collision resistance
  - **Much harder to achieve than weak collision resistance**
  - **Vulnerable to "birthday" attack**



Computed probability of at least two people sharing a birthday amongst X people.

69

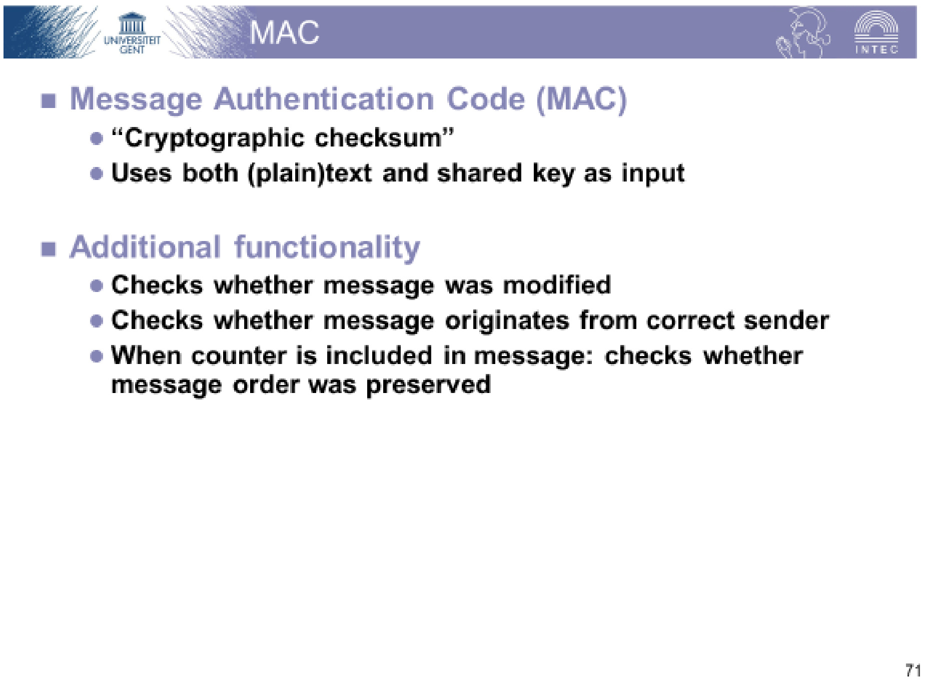
Intuitively this can be understood as follows. Although it is difficult to find a person with the same birthday as you (probability 1 in 365), the likelihood that at least two persons in a group shares a birthday is much higher. Similarly, the probability of finding a plaintext / hash value pair with same hash value amongst a group of plaintext messages is much higher than identifying a plaintext with the same hash as a specific previous plaintext message.



Overview

- A security model
- Security goals
  - Confidentiality
  - Authentication
  - Access control / authorization
  - Data integrity
  - Non-repudiation
  - Availability
- Security threats
- Security mechanisms
  - Encryption
    - Symmetric encryption
    - Asymmetric encryption
  - Hash functions
  - **Message authentication codes**

70

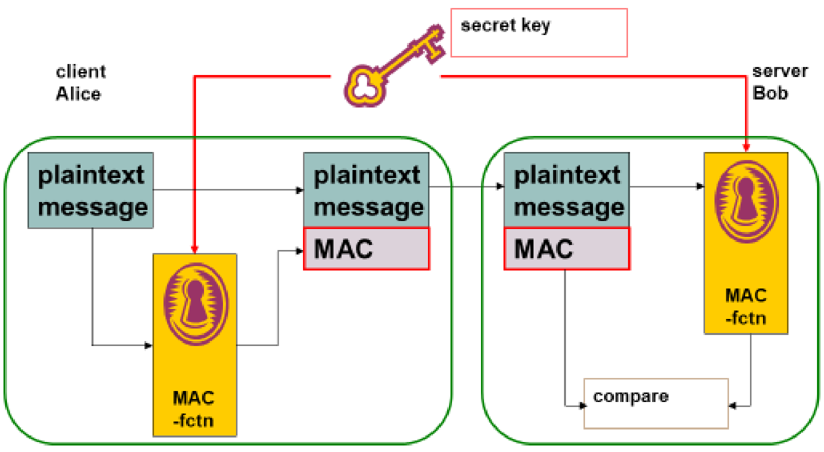


MAC

- Message Authentication Code (MAC)
  - **"Cryptographic checksum"**
  - **Uses both (plain)text and shared key as input**
- Additional functionality
  - **Checks whether message was modified**
  - **Checks whether message originates from correct sender**
  - **When counter is included in message: checks whether message order was preserved**

71

MAC

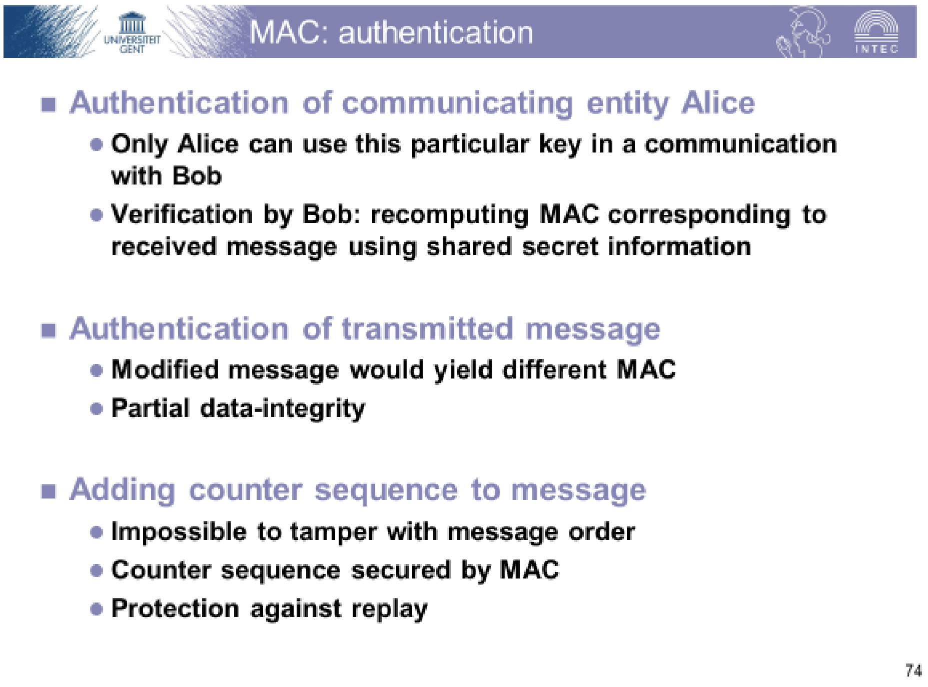


72

MAC

- MAC vs symmetric encryption
  - **Similarity: both sender and receiver share a secret key**
    - Similar authentication mechanism
    - Drawback that receiver also needs to know secret key
  - **Differences with symmetric encryption**
    - No confidentiality function
    - "Irreversible encryption": decryption impossible
- MAC vs hash function
  - **MAC also depends on secret key, while hash value only depends on message**
- Notation: $MAC = C_K(M)$
  - **where K: secret key ; M: message**

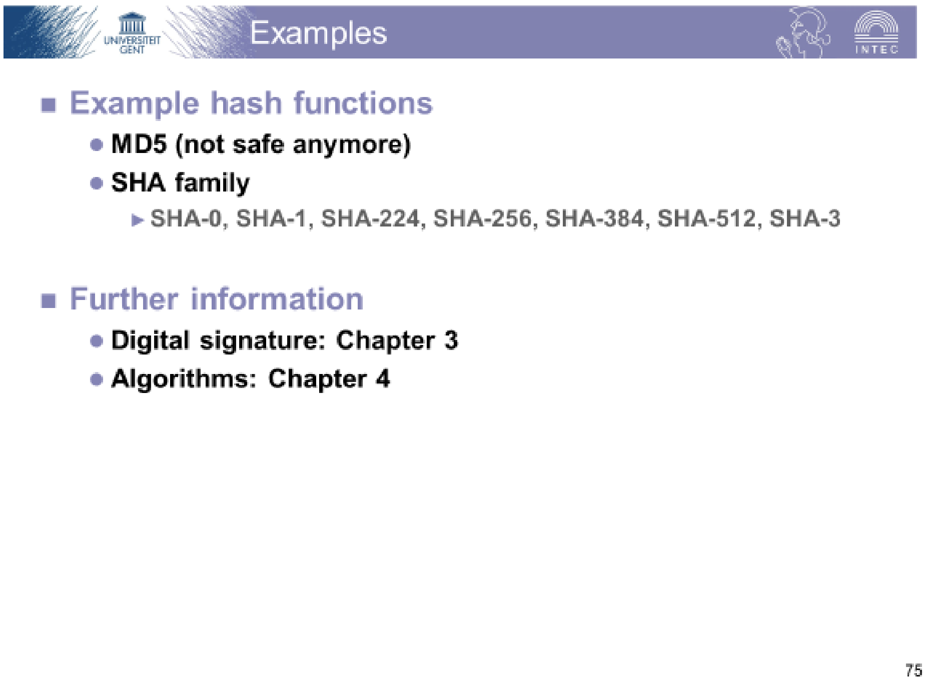
73



MAC: authentication

- Authentication of communicating entity Alice
  - Only Alice can use this particular key in a communication with Bob
  - Verification by Bob: recomputing MAC corresponding to received message using shared secret information
- Authentication of transmitted message
  - Modified message would yield different MAC
  - Partial data-integrity
- Adding counter sequence to message
  - Impossible to tamper with message order
  - Counter sequence secured by MAC
  - Protection against replay

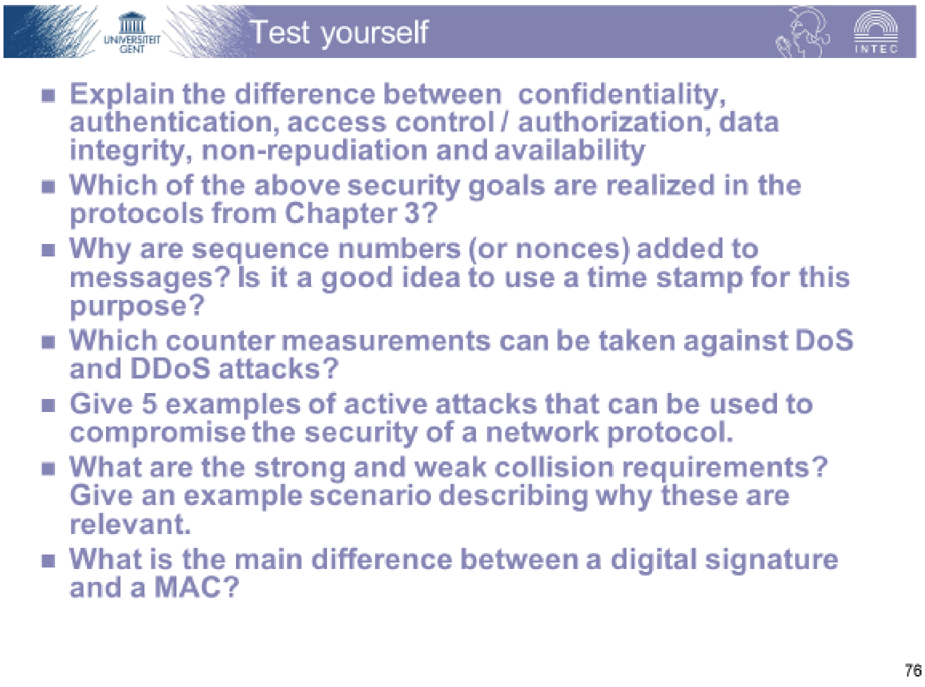
74



Examples

- Example hash functions
  - MD5 (not safe anymore)
  - SHA family
    - SHA-0, SHA-1, SHA-224, SHA-256, SHA-384, SHA-512, SHA-3
- Further information
  - Digital signature: Chapter 3
  - Algorithms: Chapter 4

75



Test yourself

- Explain the difference between confidentiality, authentication, access control / authorization, data integrity, non-repudiation and availability
- Which of the above security goals are realized in the protocols from Chapter 3?
- Why are sequence numbers (or nonces) added to messages? Is it a good idea to use a time stamp for this purpose?
- Which counter measurements can be taken against DoS and DDoS attacks?
- Give 5 examples of active attacks that can be used to compromise the security of a network protocol.
- What are the strong and weak collision requirements? Give an example scenario describing why these are relevant.
- What is the main difference between a digital signature and a MAC?

76