

ENEE 457: Computer Systems Security

8/31/16

Lecture 2

Computer Security Goals and Principles

Charalampos (Babis) Papamanthou

Department of Electrical and Computer Engineering
University of Maryland, College Park



Computer Security Goals

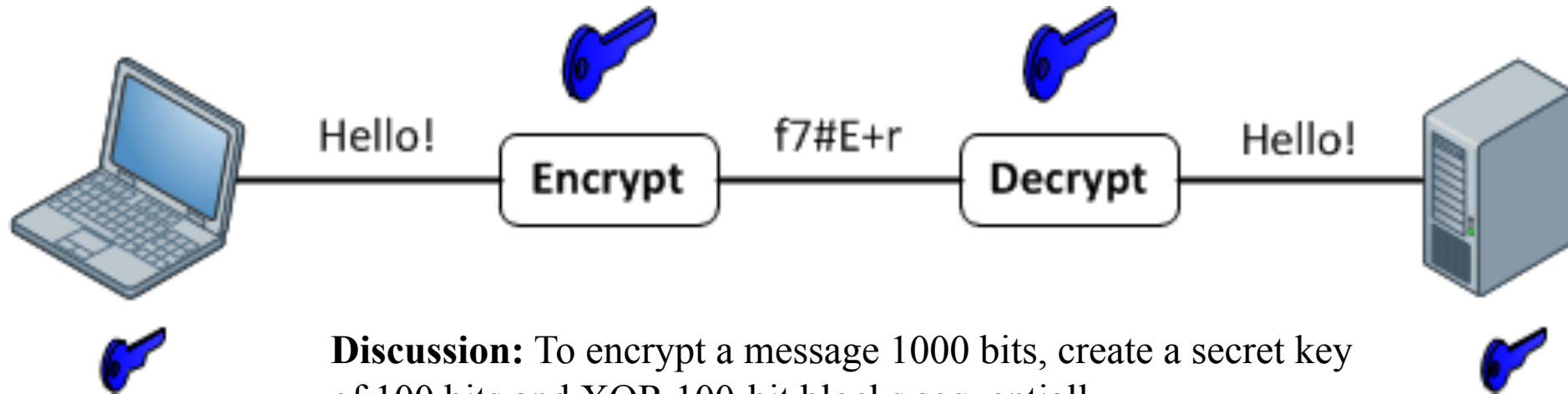
- Confidentiality
- Integrity
- Availability
- Authenticity
- Anonymity

Confidentiality

- It is the avoidance of the unauthorized disclosure of information
- It involves the protection of data, providing access for those who are allowed to see it while disallowing others from learning anything about its content
- E.g., nobody should be able to read the emails I am sending to my friends, except for my friends

Tools for confidentiality

- **Encryption:** the transformation of information using a secret, called an encryption key, so that the transformed information can only be read using another secret, called the decryption key (which may, in some cases, be the same as the encryption key)



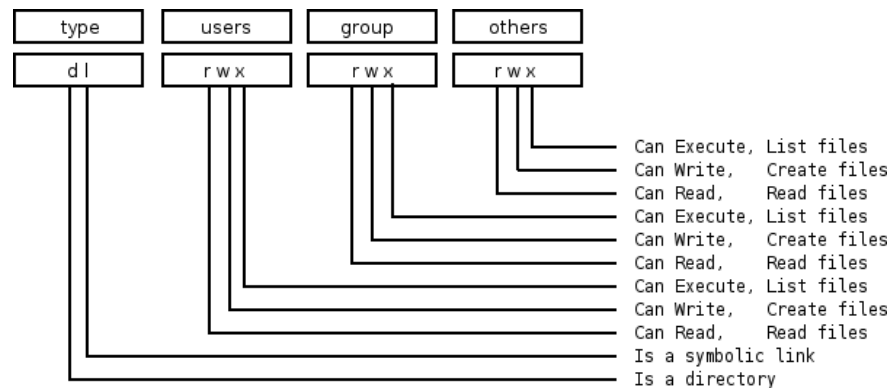
Discussion: To encrypt a message 1000 bits, create a secret key of 100 bits and XOR 100-bit blocks sequentially.

Does this reveal the content of the message?

Is this good enough?

Tools for confidentiality

- **Access control:** rules and policies that limit access to confidential information to those people and/or systems with a “need to know”
 - This need to know may be determined by identity, such as a person’s name or a computer’s serial number, or by a role that a person has, such as being a manager or a computer security specialist



Tools for confidentiality

- **Authentication:** the determination of the identity or role that someone has. This determination can be done in a number of different ways, but it is usually based on a combination of
 - something the person has (like a smart card or a radio key fob storing secret keys)
 - something the person knows (like a password)
 - something the person is (like a human with a fingerprint)



Integrity

- The property that information has not be altered in an unauthorized way
- Tools:
 - **Checksums:** the computation of a function that maps the contents of a file to a numerical value. A checksum function depends on the entire contents of a file and is designed in a way that even a small change to the input file (such as flipping a single bit) is highly likely to result in a different output value.
 - Discussion: Can we use the checksum $f(x) = x \bmod M$?

Availability

- **Availability:** the property that information is accessible and modifiable in a timely fashion by those authorized to do so
- **Tools:**
 - **Physical protections:** infrastructure meant to keep information available even in the event of physical challenges.
 - **Computational redundancies:** computers and storage devices that serve as back-ups in the case of failures
 - E.g., error-correcting codes

Other important Security goals

- Authenticity



- Anonymity



Authenticity

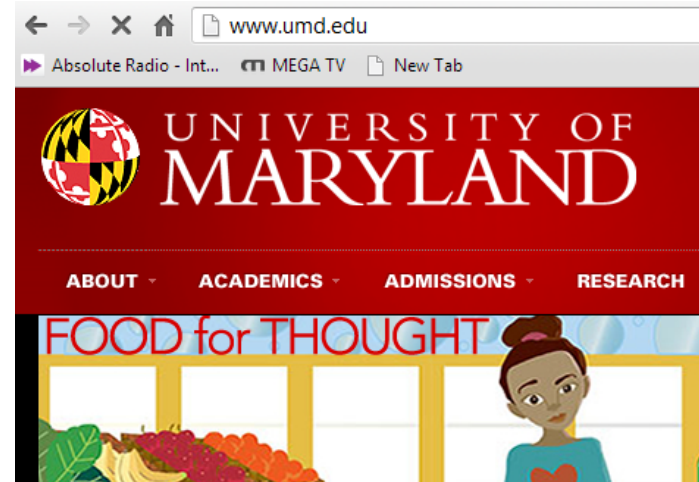
- Authenticity is the ability to determine that statements, policies, and permissions issued by persons or systems are genuine
- Primary tool:
 - **Digital signatures:** These are cryptographic computations that allow a person or system to commit to the authenticity of their documents in a unique way that achieves nonrepudiation, which is the property that authentic statements issued by some person or system cannot be denied

Anonymity

- **Anonymity:** the property that certain records or transactions not to be attributable to any individual
- **Tools:**
 - **Aggregation:** the combining of data from many individuals so that disclosed sums or averages cannot be tied to any individual (differential privacy)
 - **Proxies:** trusted agents that are willing to engage in actions for an individual in a way that cannot be traced back to that person (mixers in Bitcoin)
 - **Pseudonyms:** fictional identities that can fill in for real identities in communications and transactions, but are otherwise known only to a trusted entity (your fake Gmail account)

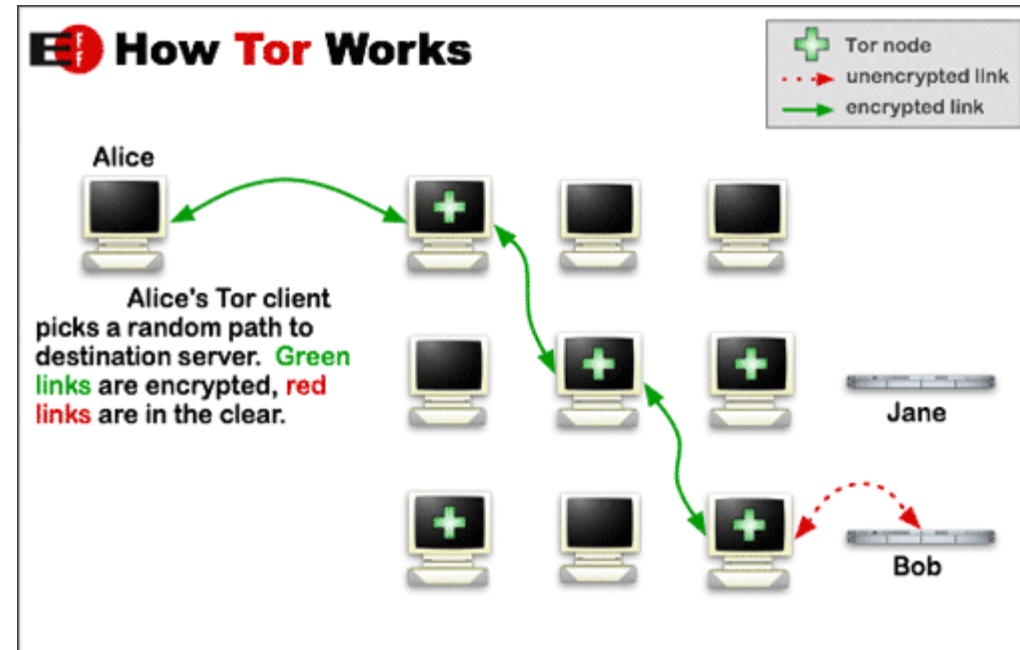
Examples: HTTPS protocol

- Confidentiality
- Integrity
- Authenticity



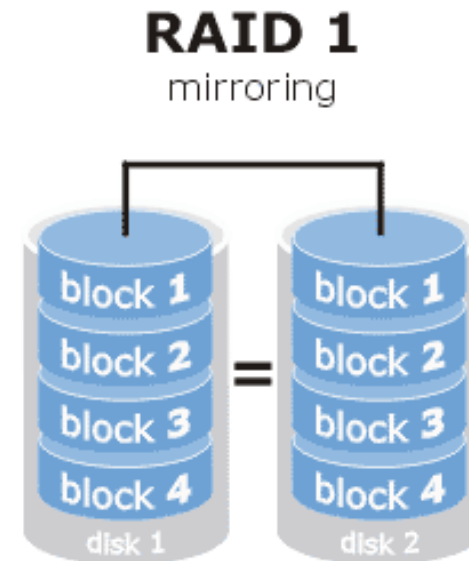
Examples: TOR protocol

- Anonymity



Examples: RAID technology

- Availability



Error correcting codes

- Given blocks b_1, b_2, \dots, b_n , we want to store them with redundancy
- If we just duplicate them and store copies c_1, c_2, \dots, c_n , it is no good. The attacker can just delete 2 blocks (e.g., b_1 and c_1) and cause damage
- Alternatively, consider the polynomial $p(x) = (x - b_1)(x - b_2) \dots (x - b_n)$
- Store b_1, b_2, \dots, b_n and $p(r_1), p(r_2), \dots, p(r_n)$ for random r_1, r_2, \dots, r_n
- Now even if the attacker deletes **any** n out of $2n$ blocks that we have stored, we can always recover our initial data
- How? Polynomial interpolation!
- As long as n out of $2n$ points are stored intact, we can always recover the initial data

Some Security Principles

Economy of mechanism

- This principle stresses simplicity in the design and implementation of security measures
 - Example: Avoid multiple interconnecting software modules (running in different machines) to implement a security property (e.g., input of password in one machine, checking of password in another)

Complete mediation

- The idea behind this principle is that every access to a resource must be checked for compliance with a protection scheme
 - As a consequence, one should be wary of performance improvement techniques that save the results of previous authorization checks, since permissions can change over time
 - For example, an online banking web site should require users to sign on again after a certain amount of time, say, 15 minutes, has elapsed

Open design

- According to this principle, the security architecture and design of a system should be made publicly available
 - Security should rely only on keeping cryptographic keys secret
 - Open design allows for a system to be scrutinized by multiple parties, which leads to the early discovery and correction of security vulnerabilities caused by design errors
 - The open design principle is the opposite of the approach known as security by obscurity, which tries to achieve security by keeping cryptographic algorithms secret and which has been historically used without success by several organizations

Fail-safe defaults

- If a system fails, it should return to the most secure mode
- Example: How do firewalls work?
 - They receive packets
 - They make decision whether to forward or drop the packet
 - If a firewall is attacked, or fails, it should ALWAYS reject a packet
 - This is good for security (at that point, functionality is gone anyways!)

Separation of responsibility

- Split a privilege so that at least two parties need to agree for an action to be performed
 - In a theatre you have two people making sure you are allowed to get in
 - A cashier that sells you the ticket and issues a receipt
 - Another employee outside of the performance room that checks the ticket
- Why this is good?
 - Cashiers are low-paid employees that might be tempted to do a favor to a friend
 - The other employee keeps them honest
- Computer system analogy
 - Check for access control in two places, so that if one gets compromised (e.g., through a buffer overflow), the other will still work

Design security in from the start

- Internet is a bad example of that
 - Now we have to patch it up (DNSSEC, IPSEC)
- Do not focus only on functionality, especially when you design network applications
 - Assume the attacker will get you
 - If something can go wrong, it will...

Psychological acceptability

- This principle states that user interfaces should be well designed and intuitive, and all security-related settings should adhere to what an ordinary user might expect
 - E.g., if a security administrator asks you to pick a 17 letters completely random password and change it every month, most likely you will write it down on a piece of paper
 - If you take 3 minutes to figure out how to set up secure connection with your friend, probably you will communicate insecurely

Work factor

- According to this principle, the cost of circumventing a security mechanism should be compared with the resources of an attacker when designing a security scheme
 - A system developed to protect student grades in a university database, which may be attacked by snoopers or students trying to change their grades, probably needs less sophisticated security measures than a system built to protect military secrets, which may be attacked by government intelligence organizations

Compromise recording (better to detect than to prevent)

- This principle states that sometimes it is more desirable to record the details of an intrusion than to adopt more sophisticated measures to prevent it
 - Internet-connected surveillance cameras are a typical example of an effective compromise record system that can be deployed to protect a building in lieu of reinforcing doors and windows
 - The servers in an office network may maintain logs for all accesses to files, all emails sent and received, and all web browsing sessions