Lecture Notes on Jan/11/2023

Chapter 1:
Computer Security Introduction and Review
(Part 02)

CYENG 351: Embedded Secure Networking

Instructor: Dr. Shayan (Sean) Taheri

Gannon University (GU)

Access Control and the Origins of Computer Security Theory (Cont.)

- Computer scientists have formalized the idea of access control, building models, and mathematically proving different policies over the years.
- **Access Control Matrix**
  - It provides a theoretical foundation for defining what security is.
  - It does not provide a practical method for implementing security for a system.
  - The most versatile and widely used model.
  - It is comprised of a grid, with resources on one axis and entities that can access those resources on the other.
  - The entries in the grid represent the rights those entities have over the corresponding resources.
  - We can represent all security situations for any system using the model.
  - Difficult to use it for any practical purposes in its complete form (i.e., Complete Formal Treatment) due to the number of possibilities.
  - **Solution:** Simplify the concept to represent larger ideas → Simplifying the matrix for looking at systems in a consistent and logical manner.
  - Having a logical and consistent representation → Allowing us to compare and contrast different security mechanisms and policies as they apply to a given system.
  - Rights and Concepts Applicable to Users, Resources, and Systems are only considered for simplification of the full model and explaining different security policies: **Read**, **Write**, and **Grant**.

Access Control and the Origins of Computer Security Theory (Cont.)

➢ Access Control Matrix

- **Read (R):** The ability to access a particular resource to gain its current state, without any ability to change that state.
- **Write (W):** The ability to change the state of a particular resource.
- **Grant (G) - Important:** The ability of a user to give access rights (including grant privileges) to another user → Allowing an expansion of rights to other users, and represents a possible security problem.

	Alice (manager)	Bob (IT admin)	Carl (Normal user)	Donna (Temporary user)
C:\Users\Alice	RWG	RWG		
C:\Users\Bob	R	RWG		
C:\Users\Carl	R	RWG	RWG	
C:\Users\Donna	R	RWG		RW

R = Can read from directory

W = Can Write to directory

G = Can grant other users read, write, or grant permission

Access Control Matrix

Access Control and the Origins of Computer Security Theory (Cont.)

➢ Access Control Matrix

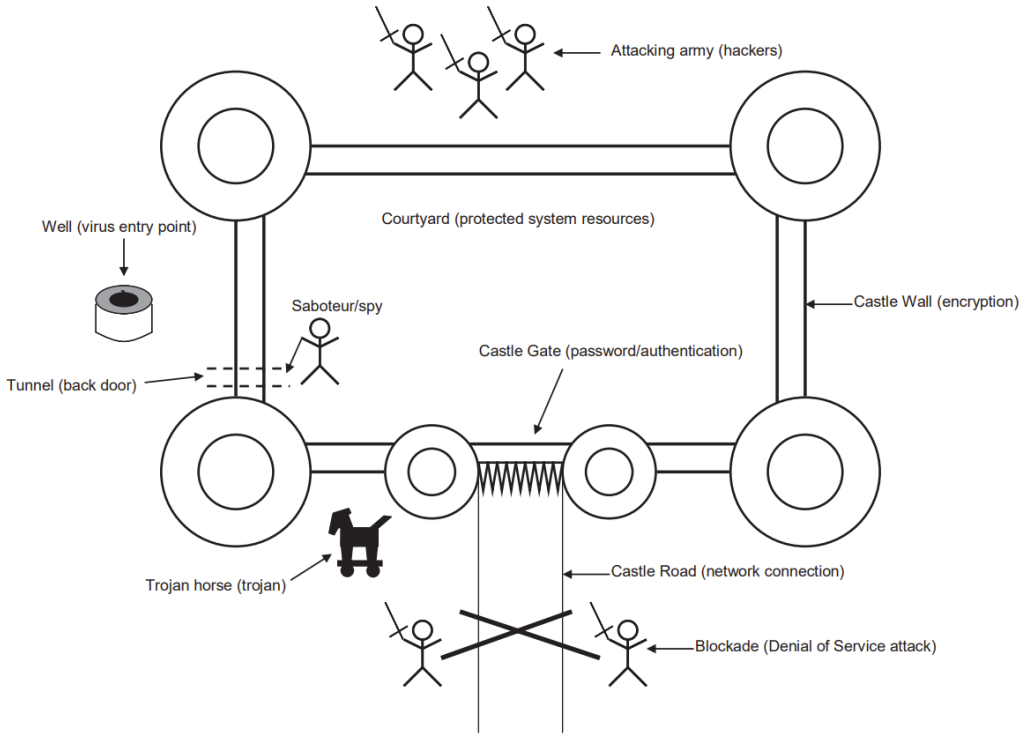
- Using the defined rights → Analyzing any given system and how secure it is, or is not.
- Using the matrix built from our system → Mathematically guarantee that certain states will or will not be entered.
- If we can prove that the only states the system enters are secure (that is, no unauthorized entities can get rights they are not entitled to, purposefully or inadvertently) → Then, we can be sure that the system is secure.
- The problem with the access control matrix model → It has been proven that this problem is undecidable in the general case → When any user is given the **grant** right, since it opens the possibility of an inadvertent granting of a right to an unauthorized user.
- However, it simply and efficiently represents security concepts in overall.

Security Policies

- **Security Policy** → It does provide a practical method for implementing security for a system (as opposed to **Matrix Access Control**).
- **Main Idea** → It is a set of rules that must be applied to and enforced by the system to guarantee some predefined level of security.
- Analogous to a legal system's penal code.
- It defines what entities (people and other systems) should and should not do.
- It requires a list of all the things that should be protected
- This list should form the basis for the security policy, which should be an integral part of the design.
- Each feature of the application must be accounted for in the policy.
- Things to take into account are: ***the considered application, the hardware used, the development tools and language used, the physical properties of the application (where is it), and who is going to be using it***.

Security Policies (Cont.)

- Castle Example
  - Consider your application as a castle.
  - You need to protect the inhabitants from incoming attacks and make sure they are content (at least if they are not happy living in a castle).
  - Your policy is then a checklist of all the things that might allow the inhabitants to come to harm.
  - There are the obvious things (get them out of the way first) like locking the castle gate and requiring proof of identity before allowing someone to enter (the password in an electronic application), or making sure the inhabitants can do their jobs (the application performs its tasks).
  - **Think like an enemy** → To develop a useful policy.
  - The enemy might not care about taking total control of the castle.
  - The enemy may want to prevent it from functioning properly by stopping incoming traders from reaching the castle (denial of service attack).



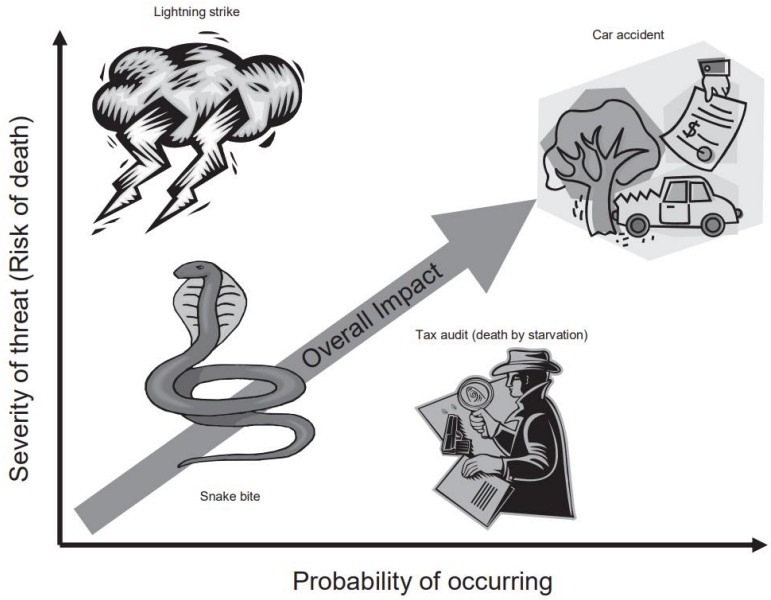
Application as a Castle

Security Policies (Cont.)

- Castle Example
  - A more subtle attack might be something that is not noticed at first, but later becomes a serious problem, like hiring an inhabitant to sabotage the defenses (disgruntled employee making it easier for hackers).
  - Poisoning the water supply (viruses).
  - The enemy may also rely on cleverness, such as the mythical Trojan horse (Trojan horses, basically viruses that give hackers a doorway directly into a system).
  - The enemy may not even be recognizable as an enemy, in the case of refugees from a neighboring war-ravaged country suddenly showing up and eating up all the available food and resources (worms like Blaster and Sasser come to mind).
  - Many problems in security are very similar in vastly different contexts.

Security Policies (Cont.)

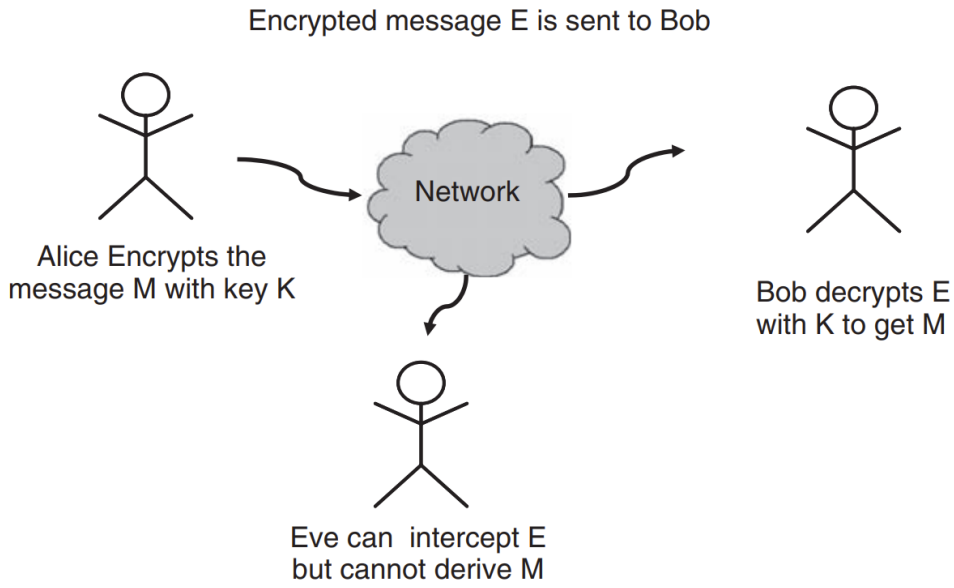
- Take **the probability of each attack** that results in each of those scenarios and multiply it with **the severity level** → Give you an idea of **the relative importance of protecting against a particular attack**.
- In the case of security, **the critic** would likely be **an expert security consultant** who you hire to look over your application design for catching possible problems (this is recommended for applications where security is truly important).
- Keeping things **secret** (**security through obscurity**) would seem to provide an additional level of security, and indeed it does, but it detracts from the overall security of the system → It is unlikely that an individual (or even an organization) can think of all the possibilities for security breaches alone, it requires different viewpoints and experience to see **what might lead to problems later**.
- Almost all of today's most widely used algorithms and protocols are wide open for review by all.
- A **good security policy** should be able to stand on its own without having to be kept secret.



Severity of Threats

Cryptography

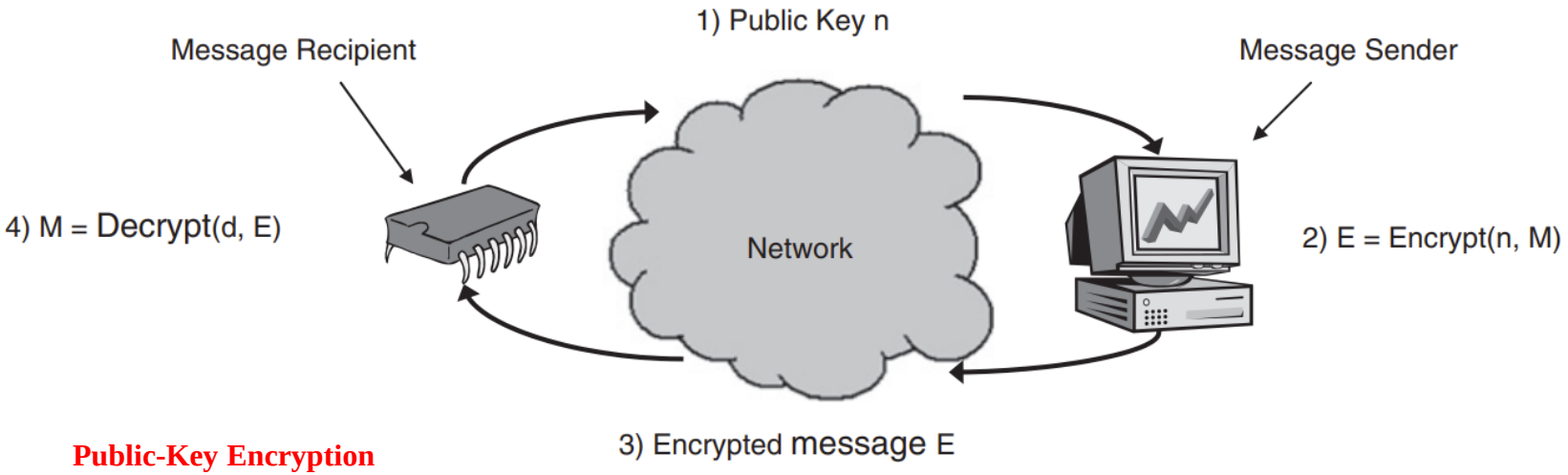
- What are the mechanisms that are used to enforce the security policies? The most important one is Cryptography.
- **Cryptography** → The science of encoding data such that a person or machine cannot easily (or feasibly) derive the encoded information without the knowledge of some secret key, usually a large, difficult to calculate number.
- The newest form of encryption is quantum cryptography, a form of cryptography that utilizes the properties of subatomic particles and quantum mechanics to encode data in a theoretically unbreakable way.
- **Symmetric Cryptography**
  - It is characterized by the use of a single secret key to encrypt and decrypt secret information.
  - This use of a single key is where the name symmetric came from, the same algorithm and key are used in both directions - hence the entire operation is symmetric.



Symmetric-key Encryption Example

Cryptography (Cont.)

- Symmetric-key cryptography has some serious drawbacks for computer security →
  - How do you give a secret key to someone you have never met (which is exactly what needs to happen for e-commerce)?
  - What do you do if your key is compromised or stolen?
  - How do you get a new key to the recipient of your messages?
- Symmetric-key algorithms are the simplest, **fastest** cryptographic algorithms we know of. → It is useful when **the speed** and **the bandwidth** are important.
- **Asymmetric Cryptography (a.k.a. Public-Key Cryptography)**
  - Public-key algorithms use different keys for both encryption and decryption (hence the asymmetry).
  - One of these keys is typically referred to as the public-key, since this key is usually published in some public place for anyone to access, and the other key is called private-key.



Public-Key Encryption

Assignment

- **Reading Assignment:**
  - Stapko, T., 2011. **Practical embedded security: building secure resource-constrained systems**. Elsevier.
    - ✓"Chapter 1: Computer Security Introduction and Review", Pages 3-13.

Questions?