CYBR372 - Week 1: Context of Cryptography

Terms in this set (11)

Cryptography	Science and the art of secrets (cryptology). Branch of mathematics. Concerns with secrecy in communications Concerns with confidentiality and integrity principles.
Applications of cryptography	Secure communication - protocols. File encryption. Content protection. User authentication.
Don't reinvent the wheel principle	Easy to get it wrong. Exposure to side-channel attacks.
Cryptography as a component of a system	By itself is useless, only a single component. Still critical, need to get it right - hard to trace if cracked. May not be the weakest component necessarily.
Weakest link property	Security is only as strong as its weakest link Addition of security may weaken the system (implementation matters!). Identify this with attack trees.

CYBR372 - Week 1: Context of Cryptography

Attack Tree	Multi-layer tree diagram covering the exposure of vulnerabilities of a system (attack surfaces). Need to only concern about key components. Can be complex and time consuming with multiple layers in covering the whole system. Layers represented by defense in depth property.
Adversarial setting	A harsh and dynamic environment. Context about the environment is unknown, anything can happen. No knowledge is given about adversaries.
Professional paranoia	The pessimistic security mindset of continuously thinking about vulnerabilities of system to protect. Attack the system and not the person.
Threat modelling	Threat analysis. Identifying assets and threats.
Cryptography is:	- not the solution, only a component that may make the system weaker. - difficult - hard to design good cryptography systems, the lack of testing will lead to a poor implementation. - the easiest component - many available and tested libraries in place that are well established.

Security criteria	Security vs performance - just throw more processing power. Costs - just find the middle ground to solve the issue, bad mindset of finding the cheapest solution vs the most secure. Security vs features - Keep It Simple Stupid, complexity is the enemy. Testing only exposes functionality, security is the absence of functionality. The solution is a simple system. Simple does not equal small - coherent modular system design with minimal dependencies Security vs evolving systems - future proofing is hard, have to maintain the system throughout its lifecycle.