The Context of Cryptography

Ramki Thurimella
Professional Paranoia or Don’t Trust Anything Blindly principle

Cryptography by itself is useless, just as a lock by itself is useless

Different weaknesses are useful to different attackers

An Attacker who breaks cryptography has low chance of getting detected
The Weakest Link Property

- A security system is only as strong as its weakest link

- Exercise: What could be some weak links in a web-based email systems?
The Adversary

- Most engineers have to contend with problems like
  - storms, heat, and wear and tear—predictable problems
- the adversary is
  - intelligent, clever, malicious, devious, and unpredictable *(don’t play by the rules)*
- Don’t know
  - Who it is
  - Motivation
  - The resources available to the Adversary
- Security systems last for > 10 years
  - But the adversary waits for 5 years, use new technology and launch an attack
The Adversary (cont.)

- Many security systems last for > 10 years,
  - the adversary does research for 5 years
  - Wait until technology is developed and
  - Launch an attack

- Fundamental imbalance between
  - System designer
    - Must defend against all (\(\forall\)) attacks
  - Attacker
    - Only needs to find one (\(\exists\)) attack

- Must think like an attacker
Professional Paranoia

- Affects rest of your life
- Can see how you can cheat others and others can cheat you
- Helps you observe things that others don’t notice
- Broader Benefits
  - This mindset helps you realize when a security problem might exist during design
  - Can get others to fix it
Discussing attacks

- Know the distinction: Attack on your work is NOT an attack on you.
- Know that: Other DO NOT know the distinction, i.e. if you attack their work, they think you are attacking them personally.
  - Be diplomatic: Ask “Have you thought about what might happen if someone did this?”
Threat Model

- *Every system can be attacked*
- Security system functions like a lock: Allow some people and deny others
- When we say a system *secure*, it usually means
  
  *Provides sufficient security of assets against certain classes of threats*

- SET protocol failure: developed for online shopping with credit cards (CC) and uses encryption to prevent
  - eavesdropping
  - The vendor from knowing CC number (this feature shifted the risk to customer who rejected this feature in favor of traditional cards)
Important to do a careful risk analysis
Analyze your assets and threats
One mistake in modeling risk can render the whole system useless
- SET is a good example
- Fuser: Online webmail aggregator
  - Failed because users did not trust a 3rd party to keep all their passwords with the company
Cryptography—not the solution

- SET is an example
- Scenario:
  - Protect file F on computer C from being read by others
  - Choice 1: Access control given by underlying OS
  - Choice 2: Encrypt F with a good (i.e. long) key K
    - Dilemma: Where do we store K
      - USB Drive: could lose it
      - Leave K on C and use access control to protect K?!?!
Cryptography—not the solution

- Systems become weaker when cryptography used incorrectly
- But, serves a crucial role in many security systems
- Many times used to satisfy regulatory controls & used incorrectly (only gives the appearance of security)
Cryptography is Difficulty

- Even experts design is broken a few years later
- Weakest link property + adversarial setting = difficult for security engineer
- No known testing methods to check if a system is secure
- Publish & get other experts to look at it
- Some methods do not receive scrutiny of other experts
Why bother then?

- Some narrow aspects of cryptography are well understood because they have been around for long.
- This course shows how to use this knowledge to build a practical system.
- Take what you learn here, work with expert cryptographers to design and analyze your new system.
Cryptography Is The Easy Part

- Like a lock, has well-defined boundaries
- On the other hand, entire system security is more elusive
- Holistic security involves organizational procedures to
  - grant access
  - check other procedures are being followed
  Harder
- Quality of software affects overall system security
Many Open Questions

- Key management and key storage
- Poor software quality
- Network security
- When users are added to the mix
- The problem of system security becomes VERY hard
- In relation to that, cryptography is easy
Generic Attacks

- Digital Rights Management (DRM) to protect audio & video
- If the user
  - Plays the audio
  - records audio
  - No security system can prevent it
- This is a generic attack
- Important to recognize what the generic attacks against a system are
  - Will not waste time trying to fix an unfixable problem
Security VS performance (cont.)

- Other professions
  - If it fails, back off and become conservative
- Computer Industry
  - Many security failures
  - But, just seems to plod along
- How much CPU?
  - Even 90% is OK
  - What good is a system that fast but insecure?
Security VS performance

- Other professions
  - First build what is safe and that works
  - Secondarily, reduce cost

- Computer Industry
  - Reversed the priorities
  - Strict efficiency demands
    - Speed of the system
    - Time to market
  - Unfortunately this results in security cost-cutting
Security Sermon

- Security first, second, and third, and performance somewhere way down
- Unfortunately the realities of marketplace trump the need for security
- Security is mix of
  - Prevention
  - Detection
  - Response
- Cryptography role is in the prevention which has to be very good so that the other two aspects are not overwhelmed
Security VS Features

- Complexity is the worst enemy of security
- If there are 20 different options which can be either on or off:
  To check functionality—test only frequently used combinations
  To check for security—test all million+ combinations
- Simple system does NOT mean small system
- Modularize: provide a clear simple interface between different parts