CS 598 - COMPUTER SECURITY IN THE PHYSICAL WORLD

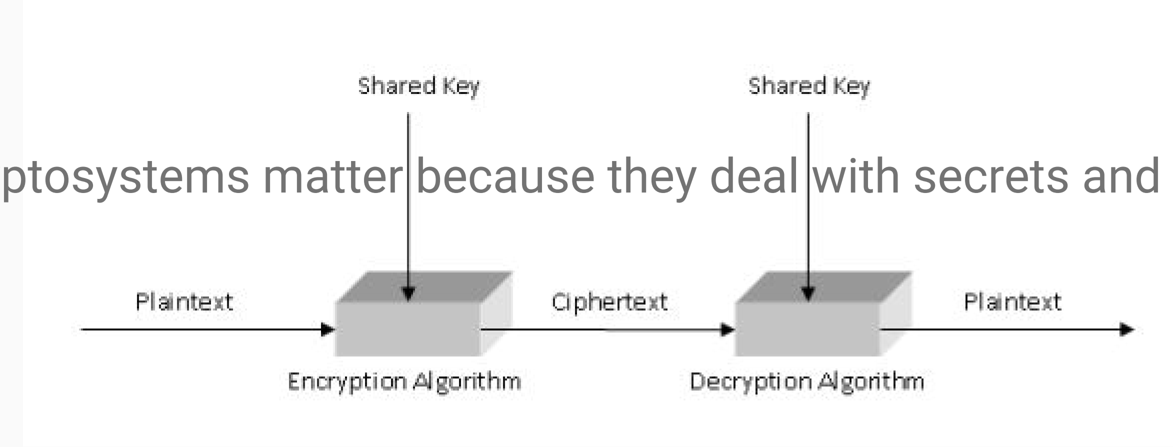
Why Cryptosystems Fail?

Ross J. Anderson - Communications of the ACM 1994

By Hassan Shahid Khan

What are cryptosystems?

- A suite of cryptographic algorithms (generation, encryption and decryption) needed to implement a particular security service, most commonly for achieving confidentiality.



- Cryptosystems matter because they deal with secrets and money.

A brief history of cryptosystems

- Cryptography was originally used by governments; military and diplomatic organisations used it to keep messages secret.

- Later, cryptographic mechanisms have been incorporated in a wide range of commercial systems.
- Introduced to the commercial world from the military by designers of automatic teller machine (ATM) systems in the 1970s.

No public feedback about how cryptosystems fail

- Unlike other engineering systems e.g airline industry, designers of cryptosystems get no feedback on their systems.

- Government or military institutions are very secretive about their systems and the mistakes that they made. Little known after 1945

- As a case study, Ross Anderson analyzed the failure models of the retail banking system.

Case Study: ATM Fraud

- ATMs use encryption to protect customers PINs.
- PIN is derived from the account number by encryption and sent in encrypted form on the line to the bank.
- Initial threat model presumed that attacks would be technically sophisticated, either cryptanalysis or via eavesdropping.

What fraudsters did

- Observing customers entering PIN’s and pick up discarded receipts.
  - Copy account number (on receipts!) to blank cards to loot customers
- Entering telephone cards
  - One british bank’s ATM believed that the previous card was entered again.
- Jackpotting (replay attack)
  - ATM networks do not encrypt/authenticate the authorization response to the ATM
  - Attacker can then record a ‘pay’ response and replay it until the ATM is empty
- Postal interception
  - Recorded delivery was not a requirement

What bank employees did

- Issued extra cards for themselves
  - Ignore complaints made by customers about phantom transactions
- Fit ATMs with devices recording PINs entered
- Exploiting test systems to calculate PINs
  - Some ATMs had a 14 digit code that gave out 10 bank notes.

Examples of bank policies

- Giving all customers the same PIN
  - Later found out to be a simple programming error
- Writing out encrypted PIN on the magnetic strip of a card
  - Criminals manipulated account numbers to match targets on magnetic strip to get access
- Check PINs only via offline systems
  - Issue policies like: First digit + Third digit = Second digit + Fourth digit
  - E.g 4455

A common observation

- Most successful attacks were not sophisticated at all!

- Anderson found that for ATM-related fraud, only two cases out of hundreds involved technical attacks.

Why was the threat model wrong?

- Expected criminals with a high level of technical expertise.
  - Blindly followed conventional military wisdom which stressed secrecy

- Human factors
  - Assumed employees implementing the systems would have appropriate expertise.
  - Mistakes in the application design and in the way the system is operated

The problem with security products

- Misguided focus towards building cryptosecurity products without addressing how to incorporate them in real systems.
  - A lot of failures occur at the implementation level.
- Poor certification process
  - Products should only be certified if they are simple enough for ordinary staff to use.
- Threat model assumes only one thing goes wrong at a time.
  - In reality, the majority of security failures is a combination of careless insiders and opportunist attacks.

Design For Robustness

- Robustness means security systems should be resilient against minor errors in design and operation, and provide redundancy against component failure.

- Overdesigning nor redundancy may be sufficient for security systems.
    - More rounds of bad algo do not necessarily make the system any more secure!

- Explicitness should be the organizing principle for security robustness.

Explicitness

- Explicitness means no implications; all objects and assumptions should be fully and clearly expressed.
    - The room for implicitness becomes an attack surface or source of failure when implementing a system
- A typical problem is to identify which objects in a system have security significance.
- Evaluating the significance of all objects in a system is not trivial.

Explicitness (cont.)

- Ways to check for explicit set of relationships:

- 1) Formal methods
  - Help us check desirable properties, verify correctness of protocols.

- 2) Integrating security with software engineering
  - Data dependency analysis
  - Agreeing upon a ‘concept of operations’ before detailed specifications.

- One should never use encryption without understanding what it is for
  - Abadı & Needham, 1994

What we can learn from safety systems

1. The specification should list all possible failure modes of the system.

2. It should explain what strategy has been adopted to prevent each of these failure modes or at least make them acceptable unlikely.

3. It should then spell out how these strategies are implemented, including the consequences when each single component fails.

4. The certification program must include a review by independent expert and test whether the system can be operated by people with the stated level of skills and experience.

Discussion Points

- Key contributions of the paper?
- Limitations of the paper?
- What is still relevant to security systems today?
- Are the same attack vectors prevalent today?
- Greater emphasis on understandability of system design + algorithms?
- Thoughts on the security certification process?