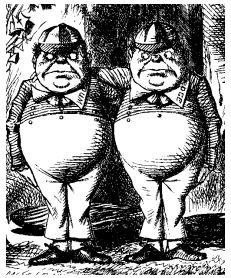
Information Security Introduction

The Cast of Characters

- Alice and Bob are the **good guys**

- Trudy is the bad "guy" →



- Trudy is our generic "intruder"

Alice's Online Bank

- Alice opens Alice's Online Bank (AOB)
- What are Alice's security concerns?
- If Bob is a customer of AOB, what are his security concerns?
- How are Alice's and Bob's concerns similar? How are they different?
- How does Trudy view the situation?

CIA

- CIA == Confidentiality, Integrity, and Availability
- AOB must prevent Trudy from learning Bob's account balance
- **Confidentiality:** prevent unauthorized *reading* of information
  - Cryptography used for confidentiality

CIA

- Trudy must not be able to change Bob's account balance
- Bob must not be able to improperly change his own account balance
- **Integrity:** detect unauthorized *writing* of information
  - Cryptography used for integrity

CIA

- AOB's information must be available whenever it's needed
- Alice must be able to make transaction
  - If not, she'll take her business elsewhere
- **Availability:** Data is available in a *timely manner* when needed
- Availability a relatively new security issue
  - Denial of service (DoS) attacks

Beyond CIA: Crypto

- How does Bob's computer know that "Bob" is really Bob and not Trudy?
- Bob's password must be verified
  - This requires some clever **cryptography**
- What are security concerns of pwds?
- Are there alternatives to passwords?

Beyond CIA: Protocols

- When Bob logs into AOB, how does AOB know that "Bob" is really Bob?
- As before, Bob's password is verified
- Unlike the previous case, **network** security issues arise
- How do we secure network transactions?
  - **Protocols** are critically important
  - Crypto plays a major role in security protocols

Beyond CIA: Access Control

- Once Bob is *authenticated* by AOB, then AOB must restrict actions of Bob
  - Bob can't view Charlie's account info
  - Bob can't install new software, and so on…
- Enforcing such restrictions: *authorization*
- **Access control** includes both authentication and authorization

Beyond CIA: Software

- Cryptography, protocols, and access control are all implemented in **software**
  - Software is foundation on which security rests
- What are security issues of software?
  - Real-world software is complex and buggy
  - Software flaws lead to security flaws
  - How does Trudy attack software?
  - How to reduce flaws in software development?
  - And what about malware?

Textbook

- The text consists of four major parts
  - Cryptography
  - Access control
  - Protocols
  - Software
- But, people cause lots of problems…

The People Problem

- People often break security
  - Both intentionally and unintentionally
  - Here, we consider an unintentional case
- For example, suppose you want to buy something online
  - Say, *Information Security: Principles and Practice*, 3rd edition from amazon.com

The People Problem

- To buy from amazon.com…
  - Your browser uses the SSL protocol
  - SSL relies on cryptography
  - Many access control issues arise
  - All security mechanisms are in software
- Suppose all of this security stuff works perfectly
  - Then you would be safe, right?

The People Problem

- What could go wrong?
- Trudy tries man-in-the-middle attack
  - SSL is secure, so attack does ***not*** "work"
  - But, Web browser warns of problem
  - What do you, the user, do?
- If user ignores warning, attack works!
  - None of the security mechanisms failed
  - But user *unintentionally* broke security

Cryptography

- "Secret codes"
- The book covers
  - Classic cryptography
  - Symmetric ciphers
  - Public key cryptography
  - Hash functions++
  - Advanced cryptanalysis

Access Control

- Authentication
  - Passwords
  - Biometrics
  - Other methods of authentication
- Authorization
  - Access Control Lists and Capabilities
  - Multilevel security (MLS), security modeling,
  - Firewalls, intrusion detection (IDS)

Protocols

- "Simple" authentication protocols
  - Focus on basics of security protocols
  - Lots of applied cryptography in protocols
- Real-world security protocols
  - SSH, SSL, IPSec, Kerberos
  - Wireless: WEP, GSM

Software

- Security-critical flaws in software
  - Buffer overflow
  - Race conditions, etc.
- Malware
  - Examples of viruses and worms
  - Prevention and detection

Software

- Software and testing
  - Open source, closed source, other topics
- Operating systems
  - Basic OS security issues
  - "Trusted OS" requirements
  - NGSCB: Microsoft's trusted OS for the PC
- Software is a BIG security topic
  - Lots of material to cover
  - Lots of security problems to consider

Think Like Trudy

- In the past, no respectable sources talked about "hacking" in detail
  - After all, such info might help Trudy
- Recently, this has changed
  - Lots of info on network hacking, malware, how to hack software, and more
  - Classes taught on virus writing, SRE, …

Think Like Trudy

- Good guys must think like bad guys!
- A police detective…
  - …must study and understand criminals
- In information security
  - We want to understand Trudy's methods
  - We might think about Trudy's motives
  - We'll often pretend to be Trudy

Think Like Trudy

- Is it a good idea to discuss security problems and attacks?
- Bruce Schneier, referring to *Security Engineering*, by Ross Anderson:
  - "It's about time somebody wrote a book to teach the good guys what the bad guys already know."

Think Like Trudy

- We must try to think like Trudy
- We must study Trudy's methods
- We can admire Trudy's cleverness
- Often, we can't help but laugh at Alice's and/or Bob's stupidity
- But, we **cannot** act like Trudy

Crypto

- **Cryptology** — The art and science of making and breaking "secret codes"
- **Cryptography** — making "secret codes"
- **Cryptanalysis** — breaking "secret codes"
- **Crypto** — all of the above (and more)

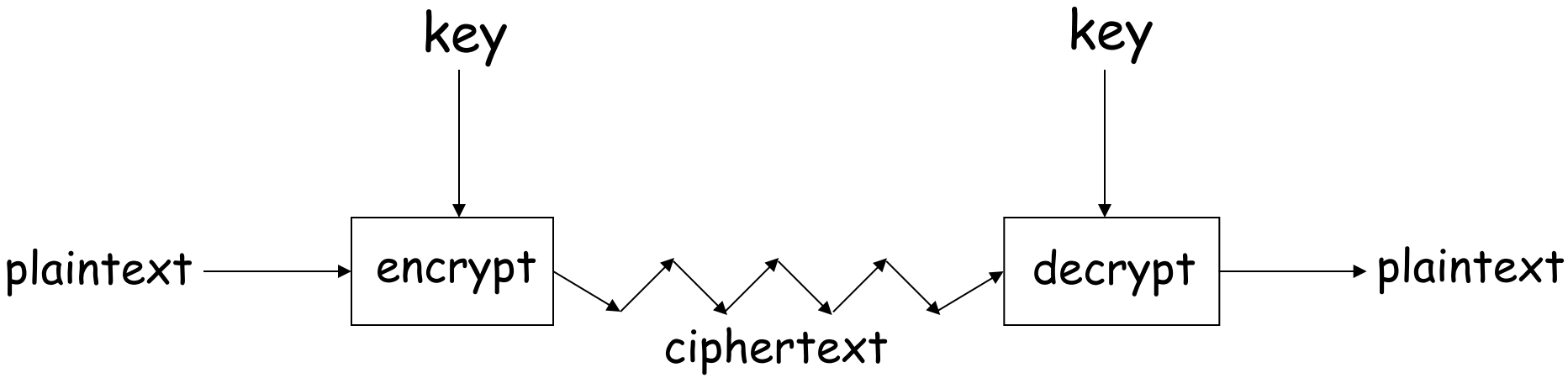
How to Speak Crypto

- A *cipher* or *cryptosystem* is used to *encrypt* the *plaintext*
- The result of encryption is *ciphertext*
- We *decrypt* ciphertext to recover plaintext
- A *key* is used to configure a cryptosystem
- A *symmetric key* cryptosystem uses the same key to encrypt as to decrypt
- A *public key* cryptosystem uses a *public key* to encrypt and a *private key* to decrypt

Crypto

- Basic assumptions
  - The system is completely known to the attacker
  - Only the key is secret
  - That is, crypto algorithms are not secret
- This is known as **Kerckhoffs' Principle**
- Why do we make such an assumption?
  - Experience has shown that secret algorithms tend to be weak when exposed
  - Secret algorithms never remain secret
  - Better to find weaknesses beforehand

Crypto as Black Box



A generic view of symmetric key crypto

Taxonomy of Cryptography

- **Symmetric Key**
  - Same key for encryption and decryption
- **Public Key** (or "asymmetric" crypto)
  - Two keys, one for encryption (public), and one for decryption (private)
- **Hash algorithms**
  - Can be viewed as "one way" crypto

Taxonomy of Cryptanalysis

- From perspective of info available to Trudy…
  - Ciphertext only — Trudy's worst case scenario
  - Known plaintext
  - Chosen plaintext
    - "Lunchtime attack"
    - Some protocols will encrypt chosen data
  - Adaptively chosen plaintext
  - Related key
  - Forward search (public key crypto)
  - And others…