A high-level survey of cryptography
Caveats

- Everything I present will be (relatively) informal
  - I may simplify, but I will not say anything that is an outright lie…

- Cryptography offers formal definitions and rigorous proofs of security (neither of which we will cover here)
  - For more details, take CMSC 456

Goals of cryptography

- Crypto deals primarily with three goals:
  - Confidentiality
  - Integrity (of data)
  - Authentication (of resources, people, systems)

- Other goals also considered
  - E.g., non-repudiation
  - E-cash (e.g., double spending)
  - Secure distributed computation
  - Anonymity
  - ...

Private- vs. public-key settings

- There are two settings for cryptography:
  - Private-key / shared-key / symmetric-key / secret-key
  - Public-key
- The private-key setting is the “classical” one (thousands of years old)
- The public-key setting dates to the 1970s

Private-key cryptography

- The communicating parties share some information that is random and secret
  - This shared information is called a key
  - Key is completely unknown to an attacker
  - Key is uniformly random
- The key k must be shared in advance
  - We don't discuss (for now) how they do this
  - You can imagine they meet on a dark street corner and Alice hands a USB device (with a key on it) to Bob
Canonical  App 1: user-to-user

Alice  \( K \)  \( \leftrightarrow \)  \( shared\ info \)  \( K \)  Bob

---

Canonical  App 2: user-to-same-user

Alice  \( K \)  \( \rightarrow \)  Bob  \( K \)
Private-key cryptography

- Two complementary goals:
  - Secrecy and integrity

- For secrecy:
  - Private-key encryption

- For integrity:
  - Message authentication codes

Attacker types

- Passive eavesdropping vs. active interference

- Secrecy is a concern for passive or active adversaries

- Integrity is a concern for active adversaries
Private-key encryption

Functional definition

- Encryption algorithm:
  - Takes a key and a message (plaintext), and outputs a ciphertext
  - $c \leftarrow E_k(m)$, possibly randomized!
- Decryption algorithm:
  - Takes a key and a ciphertext, and outputs a message (or perhaps an error)
  - $m = D_k(c)$
- Correctness: for all $k$, we have $D_k(E_k(m)) = m$
- We have not yet said anything about security…
Secrecy

- Want secrecy against a passive eavesdropper who observed the ciphertext
- This adversary does not know the key
Security through obscurity?

- Always assume that the full details of crypto protocols and algorithms are public
  - Known as Kerckhoffs’s principle
  - The only secret information is the key
- “Security through obscurity” is a bad idea…
  - True in general; even more true in the case of cryptography
  - Home-brewed solutions are BAD!
  - Standardized, widely-accepted solutions are GOOD!

Security through obscurity?

- Why not?
- Easier to maintain secrecy of a key than an algorithm
  - Reverse engineering
  - Social engineering
  - Insider attacks
- Easier to change the key than the algorithm
- In general setting, much easier to share an algorithm than for everyone to use their own
A classic example: shift cipher

- Assume the English uppercase alphabet (no lowercase, punctuation, etc.)
  - View letters as numbers in \( \{0, \ldots, 25\} \)
- The key is a random letter of the alphabet
- Encryption done by addition modulo 26

- Is this secure?
  - Exhaustive key search
  - Automated determination of the key

Another example: substitution cipher

- The key is a random permutation of the alphabet
  - Note: key space is huge!
- Encryption done in the natural way

- Is this secure?
  - Frequency analysis
  - A large key space is necessary, but not sufficient, for security
Another example: Vigenere cipher

Use a word as a set of shift cipher inputs…

- More complicated version of shift cipher
- Believed to be secure for over 100 years
- Is it secure?
Attacking the Vigenere cipher

- Let $p_i$ (for $i=0, \ldots, 25$) denote the frequency of letter $i$ in English-language text
  - Known that $\Sigma p_i^2 \approx 0.065$
- For each candidate period $t$, compute frequencies $\{q_i\}$ of letters in the sequence $c_0, c_t, c_{2t}, \ldots$
  - For the correct value of $t$, we expect $\Sigma q_i^2 \approx 0.065$
    - For incorrect values of $t$, we expect $\Sigma q_i^2 \approx 1/26$
- Once we have the period, can use frequency analysis as in the case of the shift cipher

Moral of the story?

- Don't use "simple" schemes
- Don't use schemes that you design yourself
  - Use schemes that other people have already designed and analyzed…
Moving on…