Agenda

PROBLEM DOMAIN: NUMBER THEORY:
APPLICATION DOMAIN: CRYPTOGRAPHY

- BASICS OF CRYPTOGRAPHY
  - SECRECY OR CONFIDENTIALITY
  - SHARED KEY AND PUBLIC KEY SYSTEMS

Cryptography - Secrecy

- Communication from **A(lice)** to **B(ob)**:
  - **A** sends a message **M** to **B** on *a public channel*
    - i.e. *any one can read the channel*

- (*Desired*) Property of said communication:
  - ***Secrecy*** or ***Confidentiality***:
    - *No one other than* **A** *and* **B** *can "get" the message*!

(StrawMan) Protocol for Secrecy

- **StrawMan Protocol**:
  1. **A** applies a function ***f*** on message **M**
     i.e. computes **M' = *f*(M)**
  2. **A** sends **M'** to **B** on a public channel
  3. **B** receives **M'** and *inverts*
     i.e. **B** applies f^{-1} on **M'** to get **M**
- **Secrecy Requirement**:
  - *f^{-1} cannot be computed by any one other than* **A** *and* **B**.

(StrawMan) Protocol for Secrecy [contd.]

- **StrawMan Protocol**:
  1. **A** applies a function ***f*** on message **M** i.e. computes **M′** = ***f*****(M)**
  2. **A** sends **M′** to **B** on a public channel
  3. **B** receives **M′** and *inverts* i.e. **B** applies f^{-1} on **M′** to get **M**
- **Secrecy Requirement**:
  - *f^{-1} cannot be computed by any one other than* **A** *and* **B**.
- **Solution**: *Keep **f** and f^{-1} secret!*
- **Pragmatics:**
  - *Obscurity is not security !*
  - *Complexity weakens security:*
    - Every pair of communicators will require their own functions
      - i.e. **O(N*N)** functions for a group of **N** communicators
        - i.e. this is not suitable for mass usage!

Secrecy and Encryption

- **TinMan Protocol:**
  - **A** applies a function **E** on message **M** and a key $\mathbf{K_A}$
    i.e. computes $\mathbf{M' = E(M, K_A)}$
  - **A** sends **M'** to **B**
  - **B** receives **M'** and inverts it
    - i.e. **B** applies a function $\mathbf{E^{-1}}$ on **M'** and a key $\mathbf{K_B}$ to get **M**
    - i.e. **B** computes $\mathbf{M = E^{-1}(M', K_B)}$

- **Note:**
    - **E** is referred to as an ***encryption*** function and $\mathbf{E^{-1}}$ is referred to as a ***decryption*** function. They are public.

End of Note.

Secrecy: TinMan Protocol: Requirements

- **TinMan Protocol:**
  - **A** applies a function **E** on message **M** and a key K_A i.e. computes $M' = E(M,K_A)$
  - **A** sends **M'** to **B**
  - **B** receives **M'** and inverts it by applying a function E^{-1} on **M'** and a key K_B to get **M** i.e. $M = E^{-1}(M',K_B)$
- **Secrecy Requirement**:
  - K_B must not be known to any one other than **A** and **B**.
  - Without K_B, $E^{-1}(M', K_B)$ cannot be computed.

Shared Key Encryption

- **TinMan Protocol:**
  - **A** sends **M'** = **E(M,K_A)** to **B**
  - **B** receives **M'** and computes **M = E^{-1} (M', K_B)**
- **Solution 1 :** ***Shared Key encryption***:
  - **A** and **B** share a secret **(K_A, K_B)**
  - **K_A** and **K_B** can be computed easily from each other
    - simplest case: **K_A == K_B**
- **Pragmatics:**
  - Every pair of communicators **A** and **B** will require a pair of keys **(K_A, K_B)**
    - i.e. **O(N*N)** keys (rather, ***key-pairs***) are required for a group of **N** communicators

Public Key Encryption

- **TinMan Protocol:**
  - **A** sends $\mathbf{M' = E(M, K_A)}$ to **B**
  - **B** receives **M'** and computes $\mathbf{M = E^{-1}(M', K_B)}$
- **Solution 2** : ***Public Key encryption***:
  - K_B is private to **B**: *denote it* K_{Bv} ,
  - K_A is public (*but associated with* **B**): *denote it* K_{Bu}
    - *K_{Bv} cannot be computed easily from K_{Bu}*
- **Pragmatics:**
  - Every receiver **B** will require a pair of keys (K_{Bv}, K_{Bu})
  - All public keys can be published (*say, in a directory*)!
    - i.e. **N** key-pairs are required for a group of **N** communicators

Public Key Encryption: IronMan Protcol

- **IronMan Protocol:**
  - **A** sends **M' = E(M,K_{Bu})** to **B**
  - **B** receives **M'** and computes **M = E^{-1} (M', K_{Bv})**
- **Secrecy Requirement:**
  - **E** and **E^{-1}** are computable in polynomial time with keys **K_{Bu}** and **K_{Bv}** respectively but
    - *they are not computable in polynomial time without!*
- Public Key encryption:
  - **Pragmatics:** ***K_{Bv}*** *should not be computable in polynomial time from* ***K_{Bu}***