



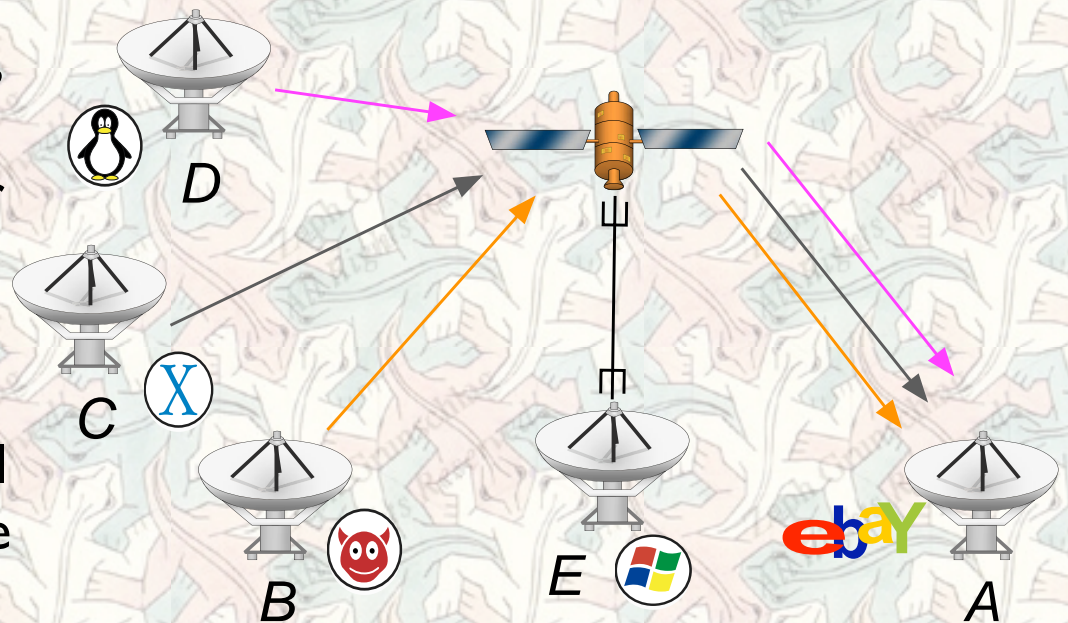
# Public Key Encryption

Zeph Grunschlag

# Public Key Encryption

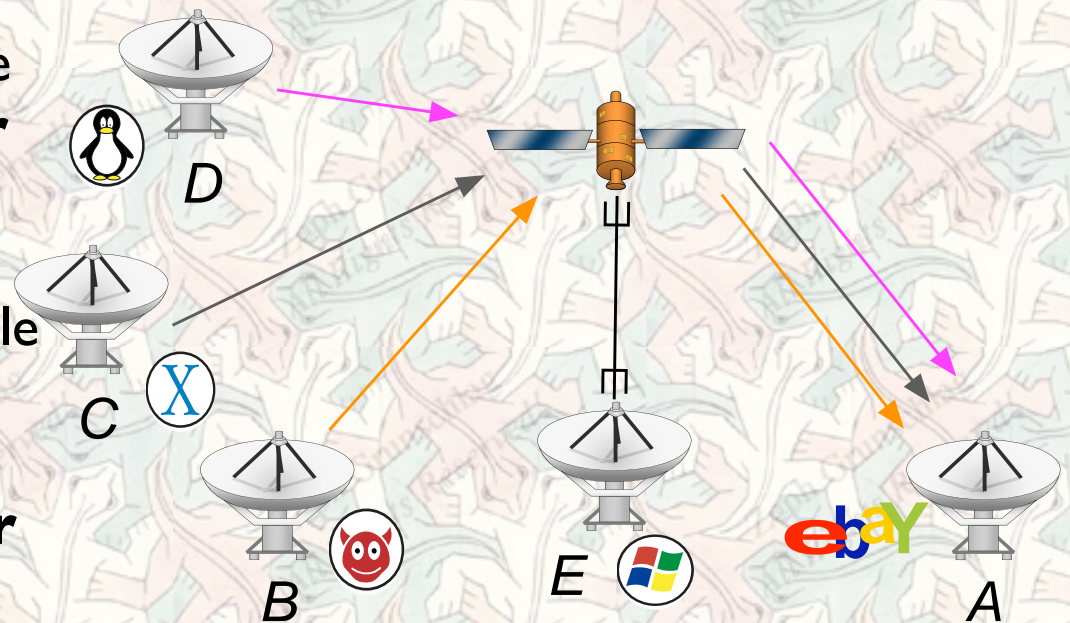
**PROBLEM:** Several individuals: Bob, Carla, David, ... wish to send messages to Alice over insecure channel eavesdropped by Eve

**GOAL:** Each individual encrypts their message without having to pre-establish secret key



# Public Key Encryption

FIRST ATTEMPT: Alice publishes a **trapdoor one-way function**, easy for *B*, *C*, *D* to encrypt with, impossible for others to decrypt, easy for *A* to decrypt with secret **trapdoor information**.



# Asymmetric Key

Each key  $K$  splits up into two parts:

- $P(K)$  - **public key** used for encryption with the function  $e_{PK}$
- $T(K)$  - **trapdoor** or **private key** used for decryption with the function  $d_{TK}$

$P(K)$  should be impossible to compute from  $T(K)$  at a minimum (**key security**).

# RSA

$K = (p, q, e)$  with  $p, q$  primes of equal bitlength,  $e$  is relatively prime to both  $p-1$  and  $q-1$

- $PK = P(K) = (n, e)$  with  $n = pq$
- $TK = T(K) = (n, d)$  same  $n$ ,  $d = e^{-1} \bmod \phi(n)$
- $P, C$  depend on  $K$ :  $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$

- Encrypt by exponentiating:

$$e_{PK}(x) = x^e \bmod n$$

- Decrypt by extracting root (raise to the  $d$ ):

$$d_{TK}(y) = y^d \bmod n$$

# Security Levels

- **Key Security** - PKE System should not allow computing  $TK$  from  $PK$ . This is considered a *total break*
- **Decryption Security** - should not allow computing a significant number plaintexts from ciphertexts. Considered *partial break*
- **Message Distinguishability** - shouldn't be able to tell which of two plaintexts got encrypted to a particular ciphertext
- **Semantic Security** - shouldn't be able to learn even a single bit of info about plaintext from ciphertext

# Trap-Door PKE Issues

Trap-door functions as defined are deterministic. Since  $e_{PK}$  is public information, Eve can compute as well and compare to eavesdropped messages:

E.g. suppose message space is limited to  $\{\text{ATTACK}, \text{RETREAT}\}$ . Eve pre-computes  $e_{PK}$  on each message and checks to see which one was sent by Bob.

**CONCLUSION:** Any secure PKE system must be randomized.

# Security of RSA

Intuitive Security: No known method of extracting  $e$ 'th roots mod  $n$  without knowing  $\phi(n)$

CLAIM: For  $n = pq$ , computing  $\phi(n)$  is equivalent to factoring  $n$ .

Key Security THM: If a BPP algorithm exists for finding a valid  $d$  from  $(n,e)$ , then a BPP algorithm for factoring  $n = pq$  exists.

Open Question: Can factoring be reduced to decrypting RSA?

# RSA - Issues

- As presented, is deterministic so suffers from the insecurities of deterministic PKE.
- Algebraic simplicity implies not decryption secure under chosen ciphertext attack
- Low decryption exponents can be cracked

# Rabin

$K = (p, q)$  with  $p, q$  primes of equal bitlength

- $PK = pq = n =$  product of the primes
- $TK = K =$  primes factors of  $n$
- $P, C$  depend on  $K$ :  $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$
- Encrypt by squaring:  $e_{PK}(x) = x^2 \bmod n$
- Decrypt by square-roots:  $d_{TK}(x) = \sqrt{x} \bmod n$ 
  1. Compute  $\pm y^{\frac{p+1}{4}} \bmod p, \pm y^{\frac{q+1}{4}} \bmod q$
  2. Patch back with CRT to get 4 roots

# Rabin - Analysis

- $d_{TK}(x)$  multivalued.

**FIX:** Include uniquely identifying information in plaintext

**Decryption Security THM:** If there is a BPP algorithm for decrypting Rabin, then there is a BPP algorithm for factoring such  $n$ .

**Negative THM:** Key insecure under chosen ciphertext attack

# Semantic Security

## Deficiencies of Stinson

May not be able to decrypt ciphertext but still discover some important bits.

DEF:  $\text{LSB}(x)$  is the least significant bit of  $x$  so is 0 for  $x$  odd and 1 for  $x$  even.

“THM”: An algorithm for finding the LSB of plaintexts from RSA ciphertexts would give an algorithm for decrypting RSA without  $TK$ .

Stinson only proves this for deterministic algorithms -almost useless.

FIX: Probabilistic Cryptography...