How to keep a secret: Leakage Deterring Public Key Cryptography

Aggelos Kiayias

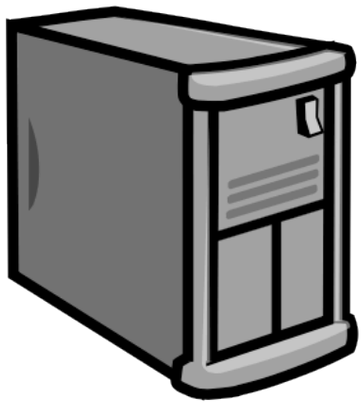
Qiang Tang

Question:

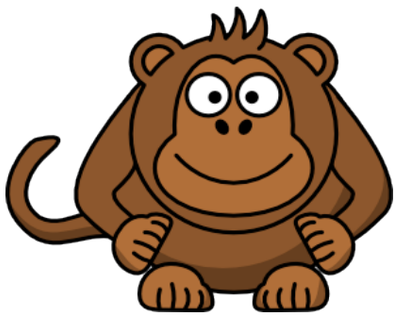
- In a PKI setting, how can we **prevent** a key owner from **leaking** software that **(partially) implements** her cryptographic function?
- Objective : motivate *accountability* amongst users & prevent the sharing of keys.

CA

e-mail
server



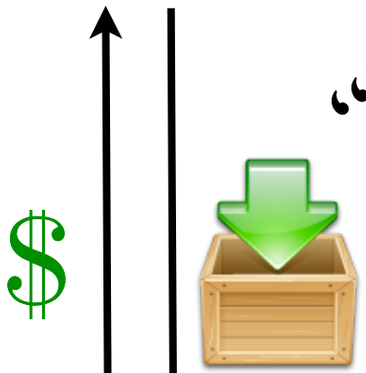
$

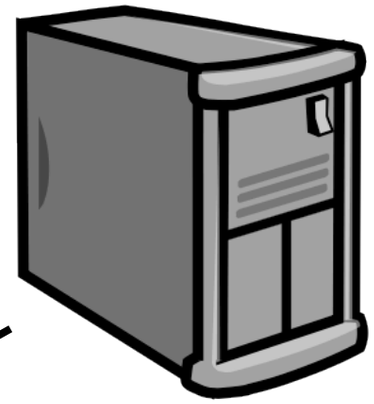


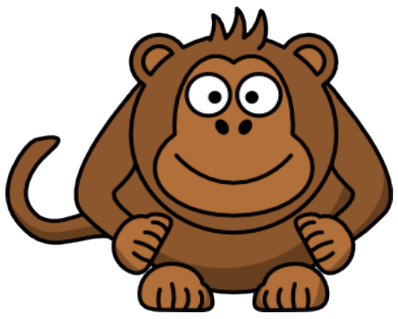
“Give me the ability to read e-mails from pig!”

CA

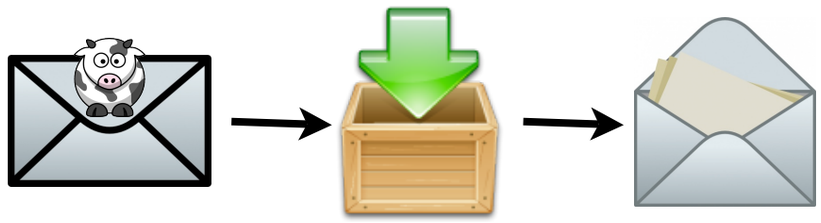
"Decryption box that decrypts e-mails from pig only!"







"Give me the ability to read e-mails from pig!"



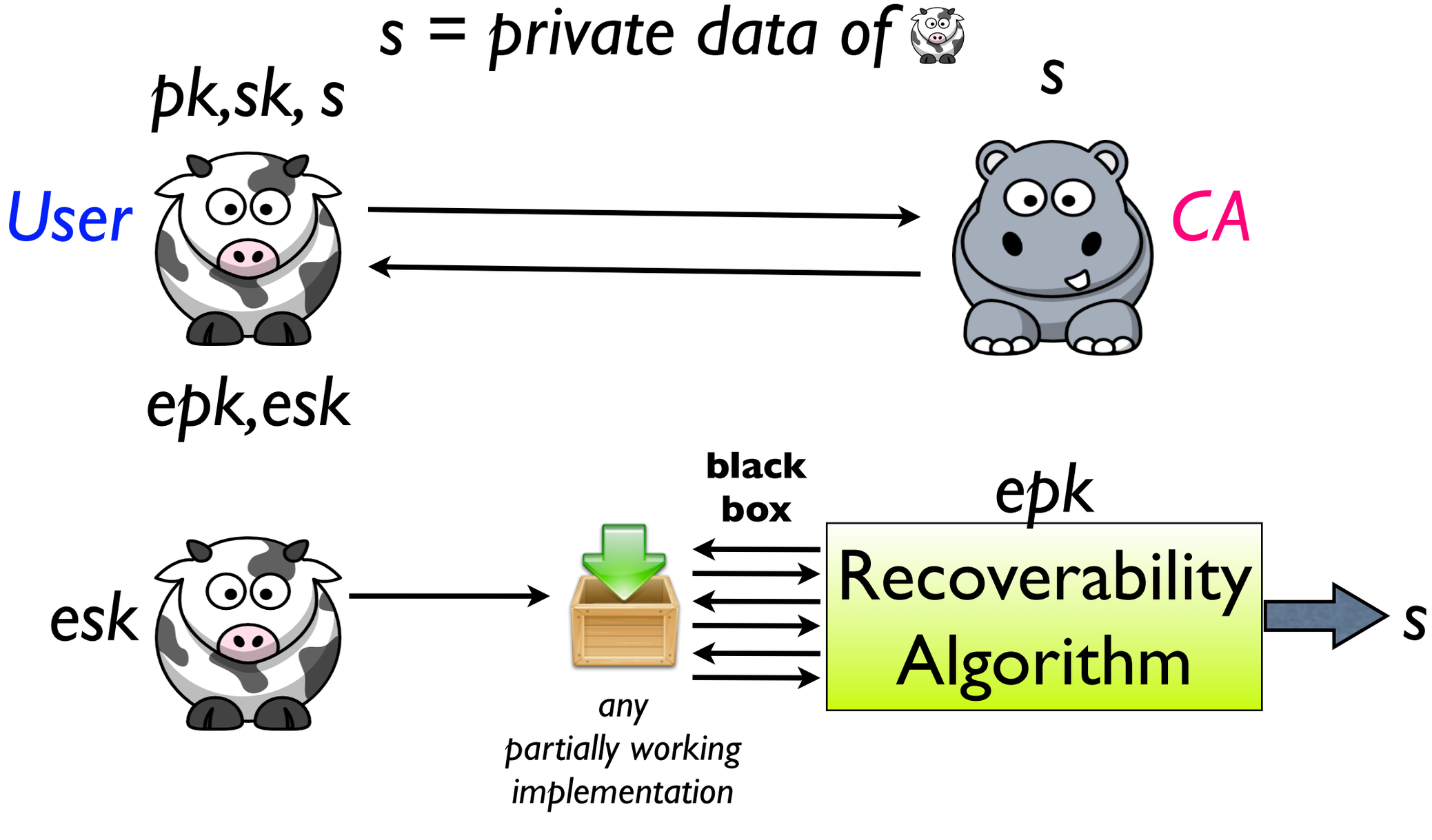
Leakage Deterring Cryptography

- **The Main Challenge**: the adversary is the secret-key owner.

Leakage Deterring Cryptography

- **The Main Challenge**: the adversary is the secret-key owner.
- Motivating idea for solution: **self-enforcement** (Dwork-Lotspiech-Naor'97) ... but **on steroids**.

Leakage Deterring Cryptography



Effectively...

- We turned a *semantic property* of software (i.e. a *guarantee* that a program works in "some way") into a decryption key that can unlock hidden information.

Results

- LD Public-key encryption.
  - based on homomorphic encryption:
    (constant ciphertexts)
    - and under general assumptions.
      (ciphertext proportional to min-entropy of plaintext distr.)
- LD Digital Signatures.
- LD Identification.

many open questions still remain!

How to keep a secret: Leakage Deterring Public Key Cryptography

[to appear in ACM CCS 2013]

Aggelos Kiayias

Qiang Tang