Defining Public Key Encryption

Slides by Prof. Jonathan Katz.
Lightly edited by me.

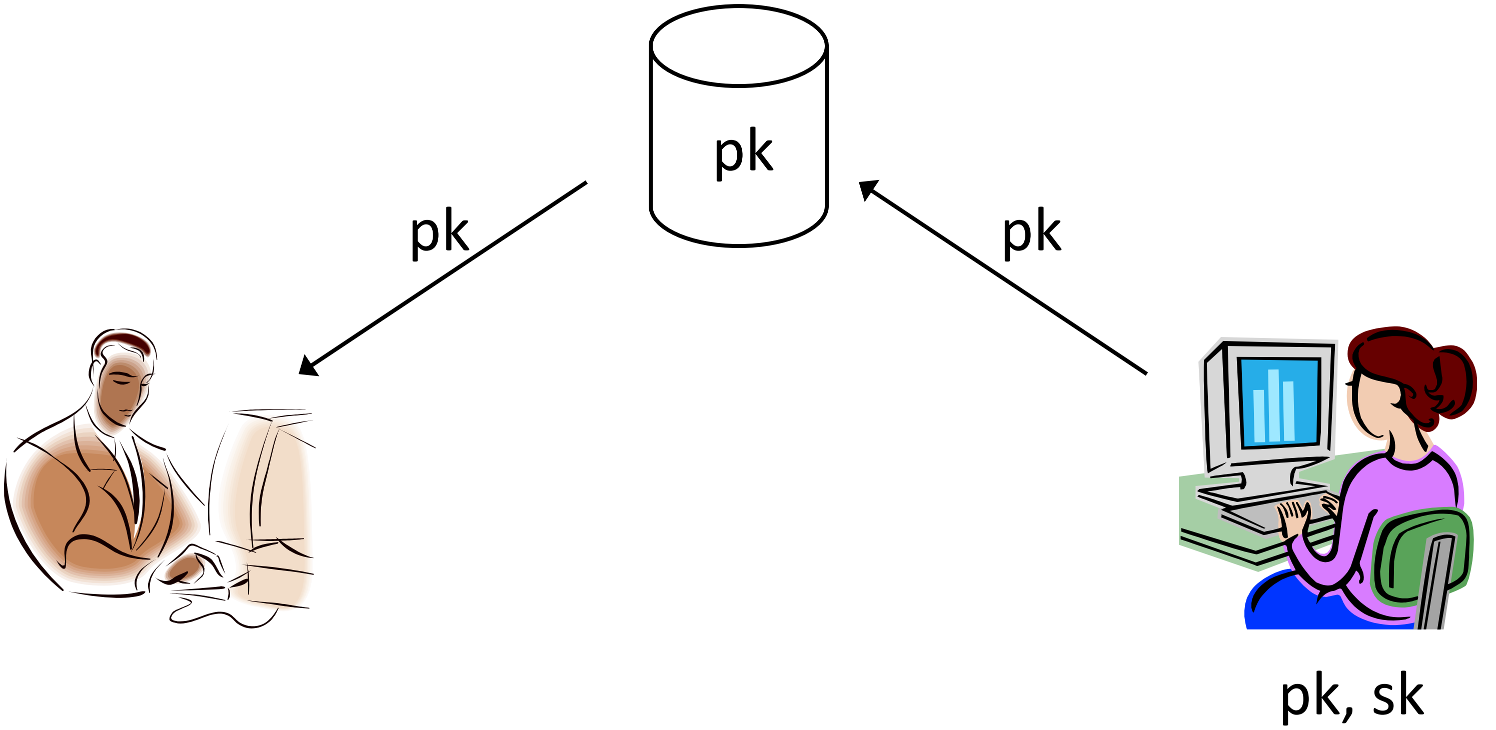
Review: private-key setting

- Two (or more) parties who wish to securely communicate *share* a uniform, secret key k in advance
- Same key k used for sending or receiving
  - Either party can send or receive
  - If multiple parties share a key, no way to distinguish them from one another based on the key
- Secrecy of k is critical
  - No security if attacker knows k

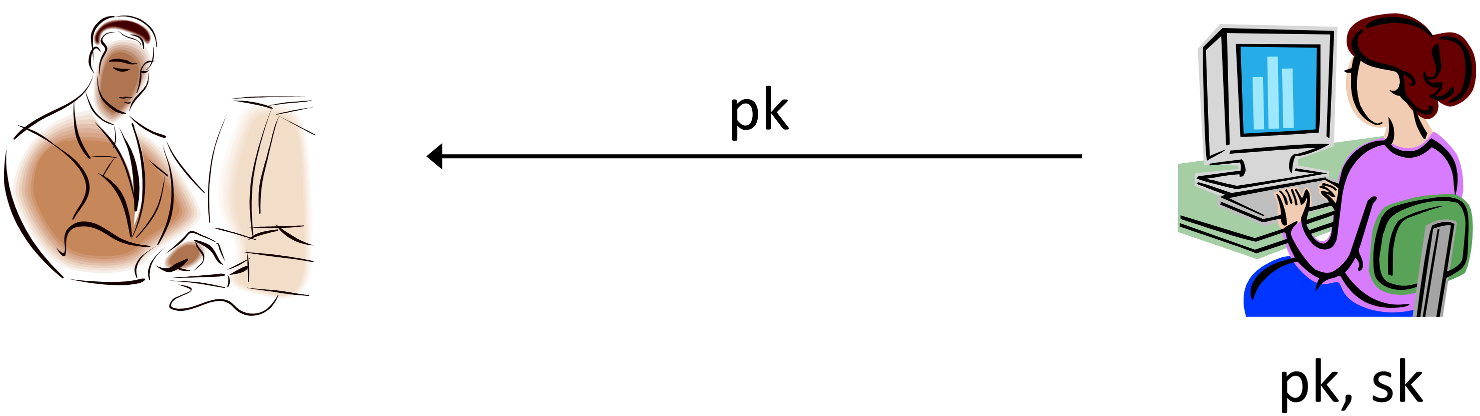
The public-key setting

- One party generates a *pair* of keys: public key pk and private key sk
  - Public key is widely disseminated
  - Private key is kept secret, and shared with no one
- Private key used by the party who generated it; public key used by anyone else
  - Also called *asymmetric* cryptography
- Security must hold even if the attacker knows pk

Public-key distribution I



Public-key distribution II



Public-key distribution

- Previous figures (implicitly) assume parties are able to obtain correct copies of each others' public keys
  - I.e., the attacker is *passive* during key distribution
- We will revisit this assumption later

Primitives

	Private-key setting	Public-key setting
Secrecy	Private-key encryption	Public-key encryption
Integrity	Message authentication codes	Digital signature schemes

How does this address the drawbacks of private-key crypto…?

- Key distribution
  - Public keys can be distributed over *public* (but authenticated) channels
- Key management in system of N users
  - Each user stores 1 private key and N-1 *public keys*; only N keys overall
  - Public keys can be stored in a central, public directory
- Applicability to "open systems"
  - Even parties who have no prior relationship can find each others' public keys and use them

Public-key vs. private-key crypto

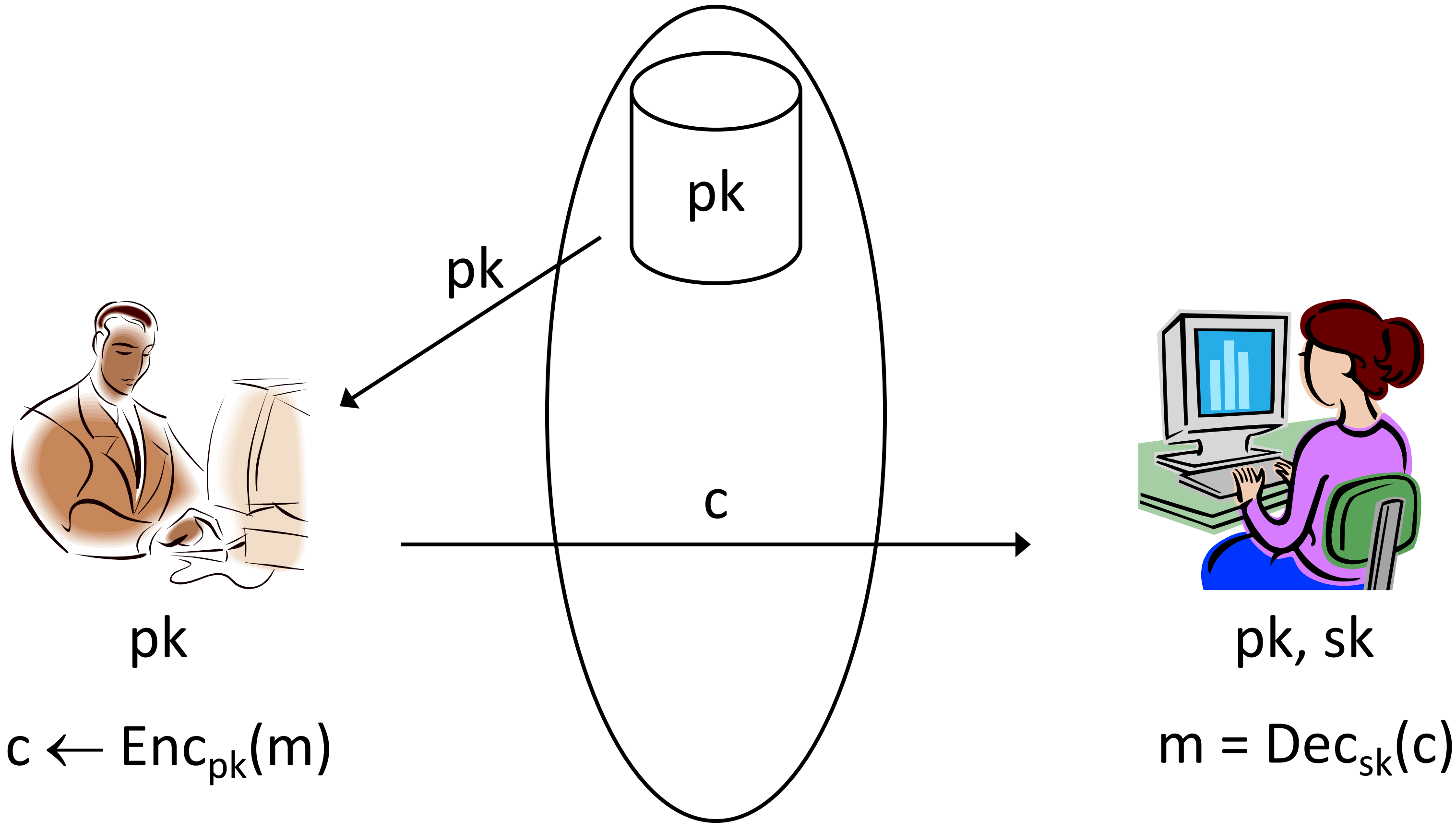
- Public-key cryptography is *strictly stronger* than private-key cryptography
  - Parties who wish to securely communicate could simply each generate public/private keys and then share them with each other
  - Use appropriate key depending on who is sending or receiving

Why study private-key crypto?

- Public-key crypto is roughly 2-3 orders of magnitude *slower* than private-key crypto
- Also 2-10× higher communication
  - If private-key crypto is an option, better to use it!

- As we will see, private-key cryptography is used for efficiency even in the public-key setting

Public-key encryption



Public-key encryption

- A public-key encryption scheme consists of three PPT algorithms:
  - Gen: *key-generation algorithm* that on input 1^n outputs (pk, sk)
  - Enc: *encryption algorithm* that on input pk and a message m outputs a ciphertext c
  - Dec: *decryption algorithm* that on input sk and a ciphertext c outputs a message m or an error $\perp$

For all m and (pk, sk) output by Gen,
$Dec_{sk}(Enc_{pk}(m)) = m$

Notes on the definition

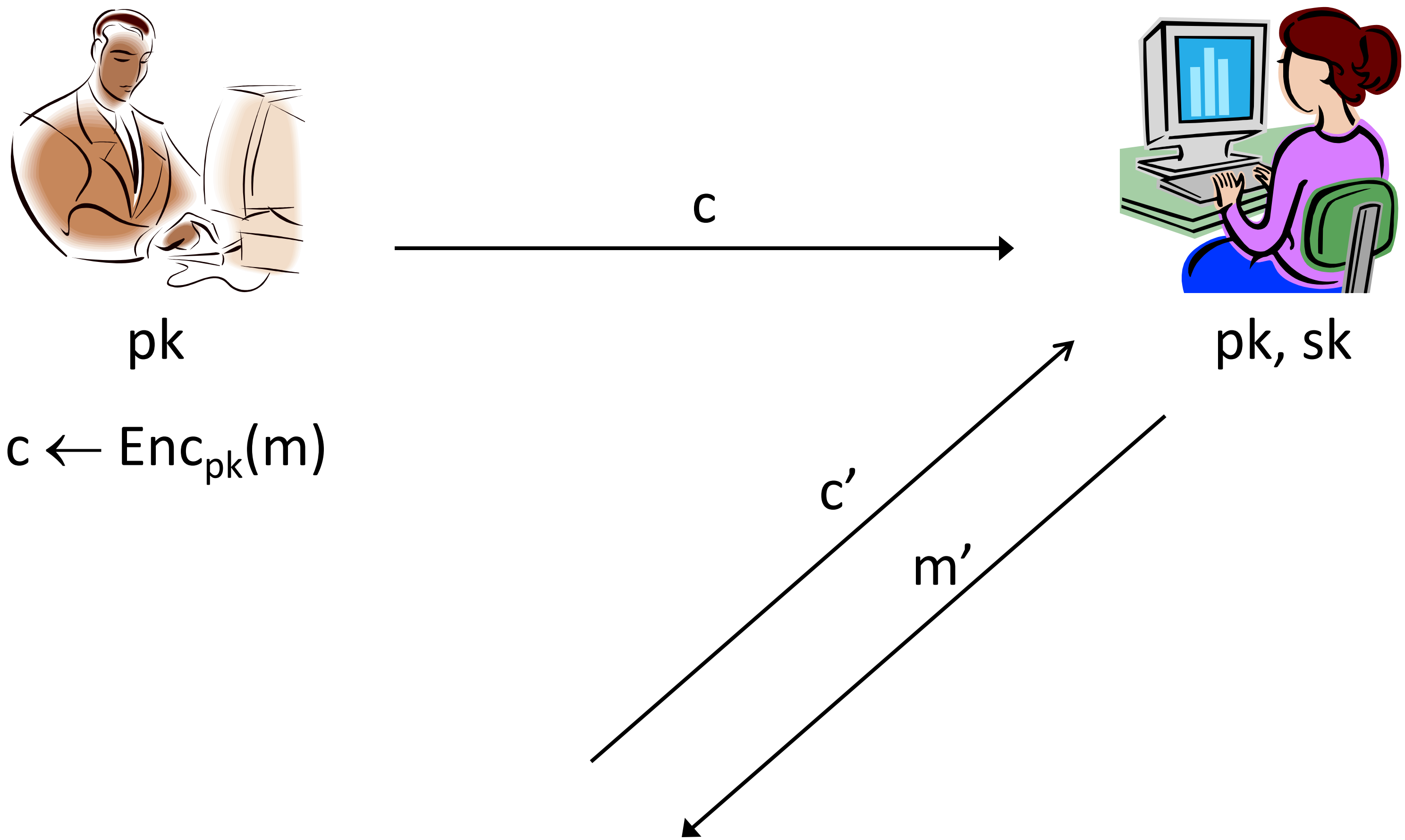
- No encryption oracle?!
  - Encryption oracle redundant in public-key setting

$\Rightarrow$ No *perfectly secret* public-key encryption

$\Rightarrow$ No *deterministic* public-key encryption scheme can be CPA-secure

$\Rightarrow$ CPA-security implies security for encrypting multiple messages (as in the private-key case)

Chosen-ciphertext attacks



Chosen-ciphertext attacks

- Chosen-ciphertext attacks are arguably even a greater concern in the public-key setting
  - Attacker might be a legitimate sender
  - Easier for attacker to obtain full decryptions of ciphertexts of its choice
- Related concern: *malleability*
  - I.e., given a ciphertext c that is the encryption of an unknown message m, might be possible to produce ciphertext c' that decrypts to a related message m'
  - This is also undesirable in the public-key setting

Chosen-ciphertext attacks

- Can define CCA-security for public-key encryption by analogy to the definition for private-key encryption
  - See book for details