

## Distributed Systems

### 24. Cryptographic Systems: A Brief Introduction

Paul Krzyzanowski  
Rutgers University  
Fall 2018

November 26, 2018

© 2018 Paul Krzyzanowski

1

## Cryptography $\neq$ Security

Cryptography may be a component of a secure system

Adding cryptography may not make a system secure

November 26, 2018

© 2018 Paul Krzyzanowski

2

## Cryptography: what is it good for?

- **Confidentiality**
  - others cannot read contents of the message
- **Authentication**
  - determine origin of message
- **Integrity**
  - verify that message has not been modified
- **Nonrepudiation**
  - sender should not be able to falsely deny that a message was sent

November 26, 2018

© 2018 Paul Krzyzanowski

3

## Confidentiality

November 26, 2018

© 2018 Paul Krzyzanowski

4

## Encryption

Plaintext (cleartext) message  $P$

Encryption  $E(P)$

Produces **Ciphertext**,  $C = E(P)$

Decryption,  $P = D(C)$

Cipher = cryptographic algorithm

November 26, 2018

© 2018 Paul Krzyzanowski

5

## Terms: types of ciphers

- **Symmetric algorithm**
  - Shared key
  - $C = E_k(P)$   $P = D_k(C)$
  - Key length  $\rightarrow$  difficulty of attack
- **Public key algorithm**
  - Key pair: **private key** ( $k$ ) & a **shared public key** ( $K$ )
  - $C = E_k(P)$   $P = D_K(C)$
  - $C = E_K(P)$   $P = D_k(C)$

November 26, 2018

© 2018 Paul Krzyzanowski

6

## Key distribution

Secure key distribution is the biggest problem with symmetric cryptography

November 26, 2018

© 2018 Paul Krzyzanowski

7

## Distributing Keys

- **Manual: pre-shared keys**
  - Initial configuration, out of band (send via USB key, recite, ...)
- **Trusted third party**
  - Knows all keys
  - Alice creates a **session key**
  - Encrypts it with her key – sends to Trent
  - Trent decrypts it and sends it to Bob
  - Alternatively: Trent creates a session key – encrypts it for Alice & for Bob
- **Public key cryptography**
  - Alice encrypts a message with Bob's public key
  - Only Bob can decrypt
- **Diffie-Hellman**
- **Hybrid cryptosystems**

November 26, 2018

© 2018 Paul Krzyzanowski

8

## Diffie-Hellman Key Exchange

### Key distribution algorithm

- First algorithm to use public/private “keys”
- Not public key encryption
- Uses a **one-way function**  
Based on difficulty of computing discrete logarithms in a finite field compared with ease of calculating exponentiation
- Compute  
**common key** =  $f(\text{your\_private\_key}, \text{their\_public\_key})$
- Eavesdroppers cannot compute this

November 26, 2018

© 2018 Paul Krzyzanowski

9

## Hybrid Cryptosystems

- **Session key**: randomly-generated key for one communication session
- Use a **public key algorithm** to send the session key
- Use a **symmetric algorithm** to encrypt data with the session key

Public key algorithms are almost never used to encrypt messages

- MUCH slower; vulnerable to *chosen-plaintext attacks*
- RSA-2048 approximately 55x slower to encrypt and 2,000x slower to decrypt than AES-256

November 26, 2018

© 2018 Paul Krzyzanowski

10

## Message Integrity

November 26, 2018

© 2018 Paul Krzyzanowski

11

## Hash functions

- **Cryptographic hash function** (also known as a **digest**)
  - Input: arbitrary data
  - Output: fixed-length bit string
- Properties of a cryptographic hash,  $H = \text{hash}(M)$ :
  - **One-way function**
    - Given  $H$ , it should be difficult to compute  $M$
  - **Collision resistant**
    - Given  $H$ , it should be difficult to find  $M'$ , such that  $H = \text{hash}(M')$
    - For a hash of length  $L$ , a perfect hash would take  $2^{(L/2)}$  attempts
  - **Efficient**
    - Computing  $H$  should be computationally efficient

November 26, 2018

© 2018 Paul Krzyzanowski

12

### Message Authentication Codes vs. Signatures

- Message Authentication Code (MAC)**
  - Hash of message encrypted with a symmetric key:
    - An intruder will not be able to replace the hash value
- Digital Signature**
  - Hash of message encrypted with the owner's private key
    - Alice encrypts the hash with her **private key**
    - Bob validates it by decrypting it with her **public key** & comparing with  $hash(M)$
  - Provides **non-repudiation**: recipient cannot change the encrypted hash

November 26, 2018 © 2018 Paul Krzyzanowski 13

### Digital signatures: public key cryptography

Alice generates a hash of the message

November 26, 2018 © 2018 Paul Krzyzanowski 14

### Digital signatures: public key cryptography

Alice encrypts the hash with her private key  
This is her **signature**.

November 26, 2018 © 2018 Paul Krzyzanowski 15

### Digital signatures: public key cryptography

Alice sends Bob the message & the encrypted hash

November 26, 2018 © 2018 Paul Krzyzanowski 16

### Digital signatures: public key cryptography

- Bob decrypts the hash using Alice's public key
- Bob computes the hash of the message sent by Alice

November 26, 2018 © 2018 Paul Krzyzanowski 17

### Digital signatures: public key cryptography

If the hashes match, the signature is valid  
– the encrypted hash *must* have been generated by Alice

November 26, 2018 © 2018 Paul Krzyzanowski 18

### Digital signatures: multiple signers

Charles:

- Generates a hash of the message,  $H(P)$
- Decrypts Alice's signature with Alice's public key
  - Validates the signature:  $D_a(S) \stackrel{?}{=} H(P)$
- Decrypts Bob's signature with Bob's public key
  - Validates the signature:  $D_b(S') \stackrel{?}{=} H(P)$

November 26, 2018 © 2018 Paul Krzyzanowski 19

### Covert AND authenticated messaging

If we want to keep the message secret  
 – combine encryption with a digital signature

Use a session key:

- Pick a random key,  $K$ , to encrypt the message with a symmetric algorithm
- encrypt  $K$  with the public key of each recipient
- for signing, encrypt the hash of the message with sender's private key

November 26, 2018 © 2018 Paul Krzyzanowski 20

### Covert and authenticated messaging

Alice generates a digital signature by encrypting the message with her private key

November 26, 2018 © 2018 Paul Krzyzanowski 21

### Covert and authenticated messaging

Alice picks a random key,  $K$ , and encrypts the message  $P$  with it using a symmetric cipher

November 26, 2018 © 2018 Paul Krzyzanowski 22

### Covert and authenticated messaging

Alice encrypts the session key for each recipient of this message using their public keys

November 26, 2018 © 2018 Paul Krzyzanowski 23

### Covert and authenticated messaging

The aggregate message is sent to Bob & Charles

November 26, 2018 © 2018 Paul Krzyzanowski 24

