Tutorial 01

Security

Introduction

- whoami?
- Who are you?

DirectPoll website

- **http://etc.ch/AMuC**

What is one of the worst thing about DirectPoll website?

Doubts from Exercise Sheet 1?

Basic Terminal tricks

- **There are a lot of commands, what to do?**

```
whatis ls
which ls
ls --help
ls --help | grep human | less
man ls
```

```
sudo apt-get install tldr
tldr ls
```

Confidentiality and Integrity

- **Which of these schemes preserve integrity? Which preserve confidentiality?**

A. One Time Pad
B. Block Ciphers
C. MAC (Message Authentication Code)
D. SHA-1
E. AES (Advanced Encryption Standard)
F. MD5

Confidentiality and Integrity

●

A.	One Time Pad	Confidentiality	
B.	Block Ciphers	Confidentiality	
C.	MAC (Message Authentication Code)		Integrity
D.	SHA-1		Integrity
E.	AES (Advanced Encryption Standard)	Confidentiality	
F.	MD5		Integrity

One Time Pad

- **Inspired from one time pad, your friend introduces a new scheme. Will it work correctly? (Uses OR instead of XOR)**

  k ∈ {0, 1}^{n}

  m ∈ {0, 1}^{n}

  c ≔ m + k , m ← c + k

A. True

B. False

One Time Pad

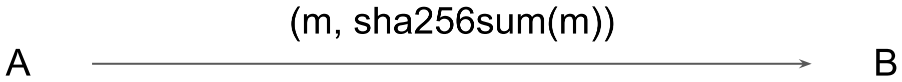
- False

Key (k)	1 0 1 0
Message (m)	1 1 0 0
Cipher text (k + m)	1 1 1 0
Decrypted (c + k)	1 1 1 0

Arbitrary schemes don’t work. See what makes XOR special.

Hash as Message Authentication Code

- **All information about protocol implementation is public. What kind of attack is possible here?**



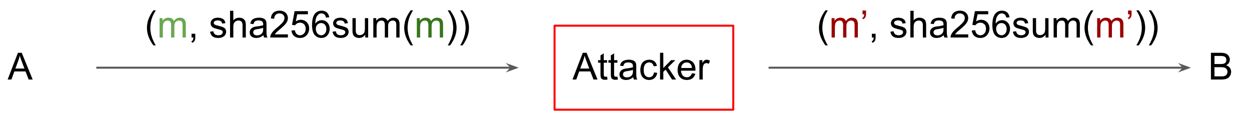
(m, sha256sum(m))

A

B

Hash as Message Authentication Code

- MITM. B has no way to verify that this was sent by A



Show hashes on terminal

- echo “text” | sha256sum
- echo “test” | sha256sum

One key to rule them all

- Slide 2 [Chapter 3] says:
    - “Symmetric crypto requires secret key between communication pairs.
      Number of keys: n(n-1)/2”

- **What is wrong with all “n” parties sharing the same symmetric key?**

One key to rule them all

- Assume n is large enough (let 1000).
    - If key gets leaked, all 1000 machines will need to replace their key.
    - If Machine C goes rogue: it can decrypt messages between A and B.
- Plus, we need a secure communication channel to exchange keys in symmetric cryptography. Instead, we can simply broadcast our public key in asymmetric cryptography.

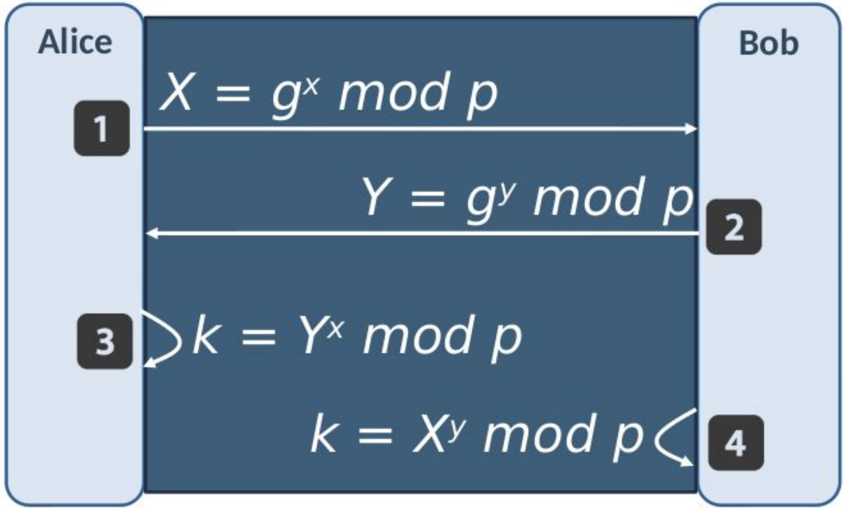
Diffie-Hellman Key Exchange

- **Which of these pairs are kept private?**

A. X, Y (Exponentiation results)

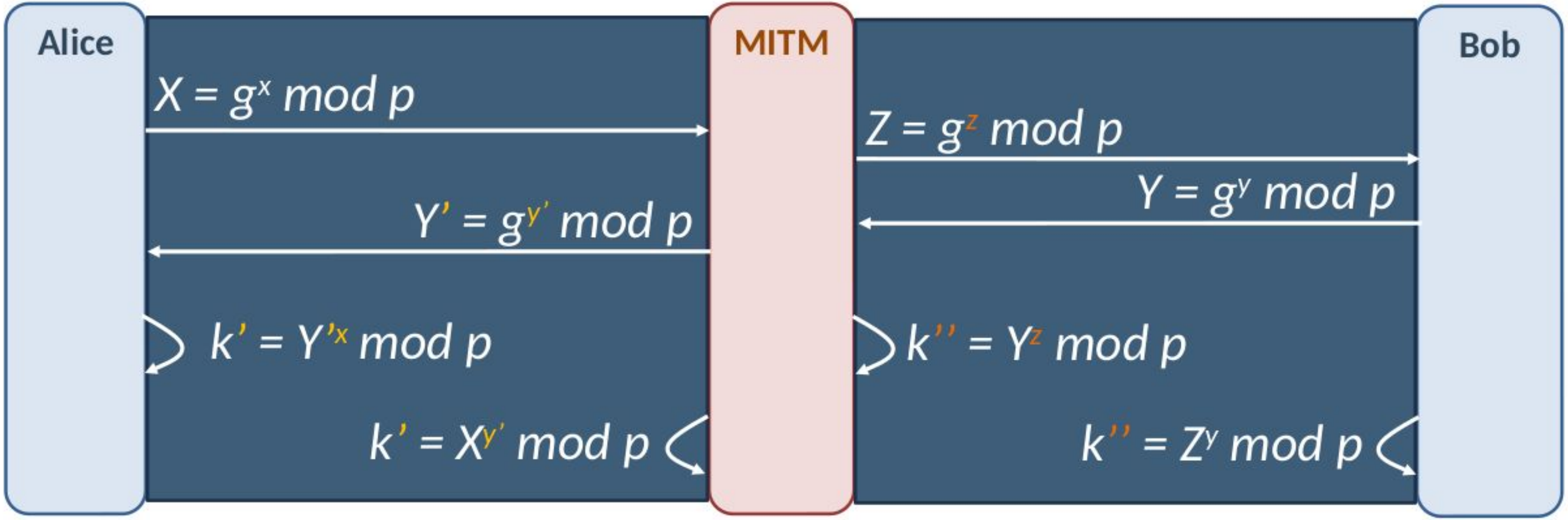
B. g, p (exponentiation base, divisor)

C. x, y



Diffie-Hellman Key Exchange

- x, y are kept private.



- Note that attacker uses publicly available g, p to do MITM

What happened to RSA?

- Let’s do an iteration of RSA: [private] [public]
  - 2 prime numbers: **p = 11** and **q = 7**
  - Compute RSA modulus **(n) = pq = 77**
  - **z = (p-1)(q-1) = 10 * 6 = 60**
  - Select **e < z**, such that it is relative prime to **z**; **e = 17**
  - Find d such that **ed mod z = 1**; **d = 53**
- Public key pair (e, n) = (17, 77); Private key pair (d, n) = (53, 77)
- Encryption: Cipher (c) = m^e mod n
- Decryption: m = c^d mod n

Example 1: m = 25; c = 25^{17} mod 77 = 9; Decrypt: 9^{53} mod 77 = 25

Example 2: m = 100; c = 100^{17} mod 77 = 67; Decrypt: 67^{53} mod 77 = 23

What happened to RSA?

- RSA with modulus n works with set of integers {0, …., n-1}
- Note that in example 2: result of decryption is 100 modulus 77

- Refer: Why RSA can't handle numbers above 76?

Who signed my papers?

- **A and B agreed on a deal. They need to sign the agreement. What should they use?**

A. **Message Authentication Code (MAC)**
B. **Digital Signatures using RSA**

Who signed my papers?

- A and B agreed on a deal. They need to sign the agreement. What should they use?

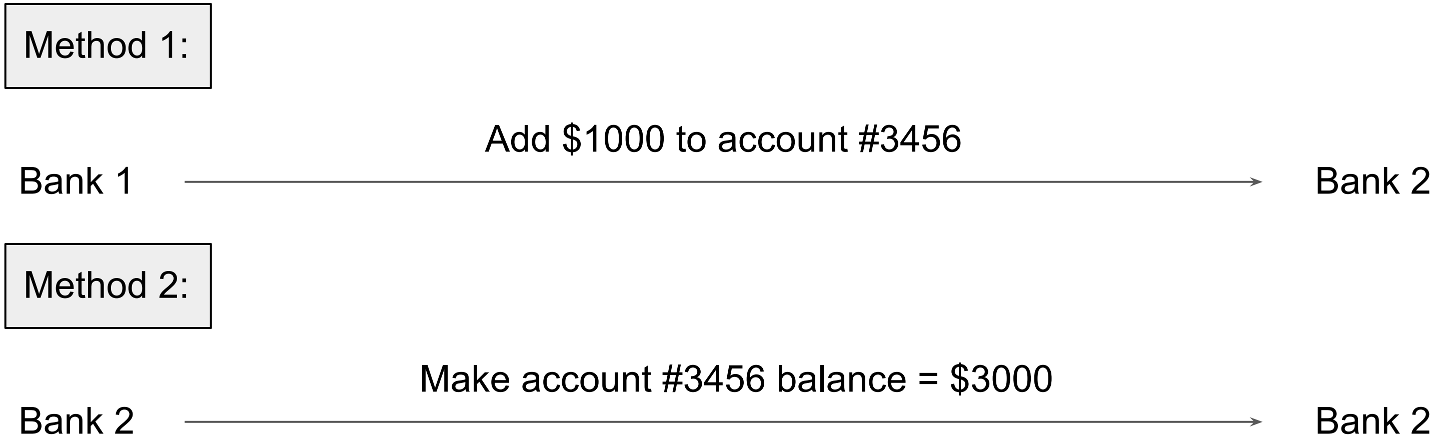
A. Message Authentication Code (MAC)

B. Digital Signatures using RSA

- **MAC requires both parties to share the key. Can not guarantee who signed the document.**
- **Both parties have different signing keys in RSA.**

Idempotent operation

- **Bob took the Security lecture last year and learned about replay attacks. He says method 2 is no longer susceptible to replay attacks, is he correct?** (Account #3456 currently has $2000, no other transactions happen until current one is committed.)



Method 1:

Add $1000 to account #3456

Bank 1 → Bank 2

Method 2:

Make account #3456 balance = $3000

Bank 2 → Bank 2

Idempotent operation

- Introduce idempotent operations.
- No. Replay attacks are still possible.

What can an attacker do?

- Read
- Manipulate
- Drop
- Repeat
- Inject
- Reflect
- ….
- ….

Perfect Forward Secrecy

- **Using a new set of RSA keys for every session achieves Perfect Forward Secrecy?**

A. **True**

B. **False**

Perfect Forward Secrecy

- Using a new set of RSA keys for every session achieves Perfect Forward Secrecy?

A. **True**
B. False

- Even if one of your RSA key gets leaked, data associated to that session will be leaked. All other sessions are still unharmed.
- But DH keys cheaper to generate than RSA keys and thus widely used.

Root CA == God Mode?

- **Who signs the root certificates?**

A. **Trusted Third Parties**

B. **Self-signed**

C. **Not signed**

- **How can it be misused?**

Root CA == God Mode?

- Who signs the root certificates?

A. Trusted Third Parties

B. Self-signed

C. Not signed

- How can it be misused?
  - Root CA can then issue certificate for Intermediary CA, which in turn can issue certificate for any arbitrary website.
  - Symantec issued “test” certificates for google.com in 2015 and 2017
  - DNS Poisoning + Mis-issued Certificate = Boom! (But google still has more measures)

Visit Certificates in Firefox

- Compare
  - Root CA
  - Intermediate CA
  - End Certificate

- Firefox
  - Web Developer tools: Networks tab > Security
  - about.config
  - security.ssl3

Colliding Certificates

- **Certifying Authorities sign hash of a certificate. If we use SHA256 for this hashing, we are safe from colliding certificates as mentioned in the lecture.**

A. **True**

B. **False**

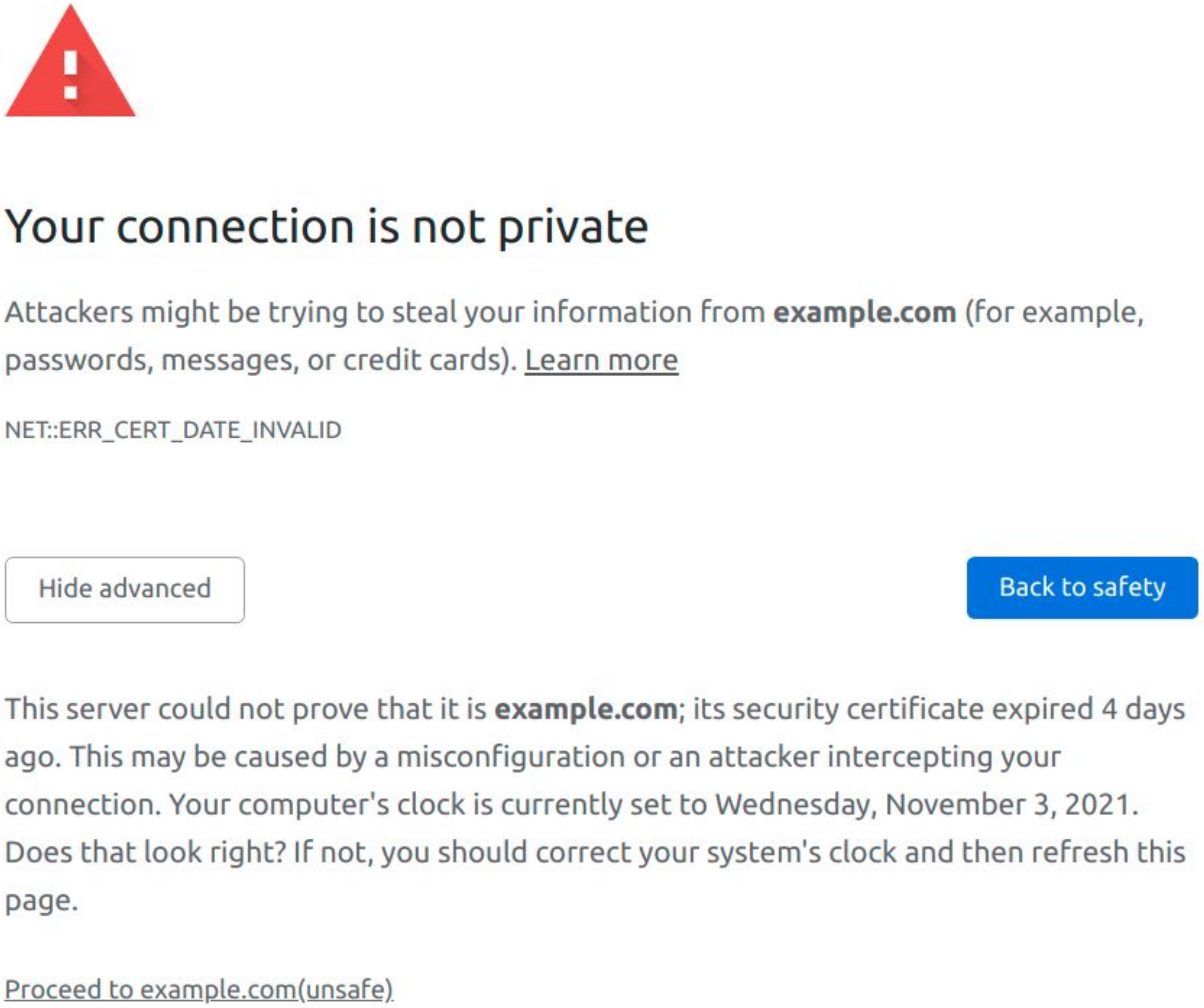
Colliding Certificates

- Certifying Authorities sign hash of a certificate. If we use SHA256 for this hashing, we are safe from colliding certificates as mentioned in the lecture.

A. **True**

B. False

- CAs used MD5 for hashing. Collision attacks exist against MD5. Use safer hash functions.
- Refer: The MD5 certificate collision attack, and what it means for Tor | Tor Blog



Your connection is not private

Attackers might be trying to steal your information from **example.com** (for example, passwords, messages, or credit cards). Learn more

NET::ERR_CERT_DATE_INVALID

This server could not prove that it is **example.com**; its security certificate expired 4 days ago. This may be caused by a misconfiguration or an attacker intercepting your connection. Your computer's clock is currently set to Wednesday, November 3, 2021. Does that look right? If not, you should correct your system's clock and then refresh this page.

Proceed to example.com(unsafe)

Is 90 days special?

- **Have a look at cispa.de. They use Let’s Encrypt. The certificate expires in 90 days. Why this interval?**
- **Does every connection become public on 91st day?**

Is 90 days special?

- A security decision. Crypto does not break on 91st day. Mainly to:
  - Prevent damage from misplaced/stolen keys.

- Refer: Why ninety-day lifetimes for certificates?

Feedback Form

- Would like to see anything different?
- Liked it, hated it, something can be improved?

- Link: https://docs.google.com/forms/d/e/1FAIpQLSfVW3Uh73PKfrAIjcsTzTtZVV_2bWobt-E9VRvXc1J3erHpVg/viewform