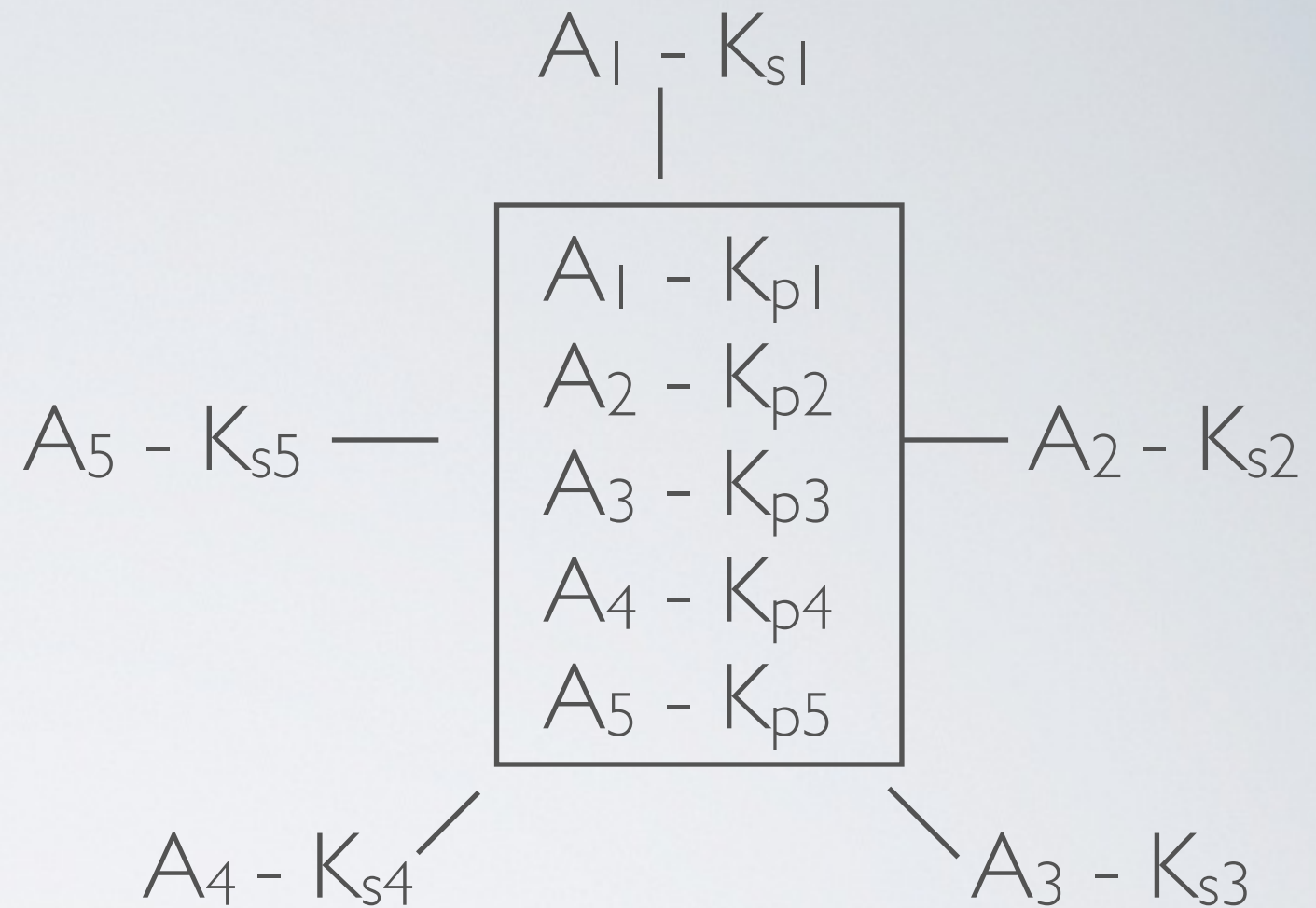


# Asymmetric Protocols

Thierry Sans

# Key exchange the public-key solution



Each  $A_i$  has a pair  $(K_p, K_s)$  and  $K_p$  is made public

➡ ... details coming later

➡ This is how the web works !

# The Needham-Shroeder public-key protocol

## Assumptions

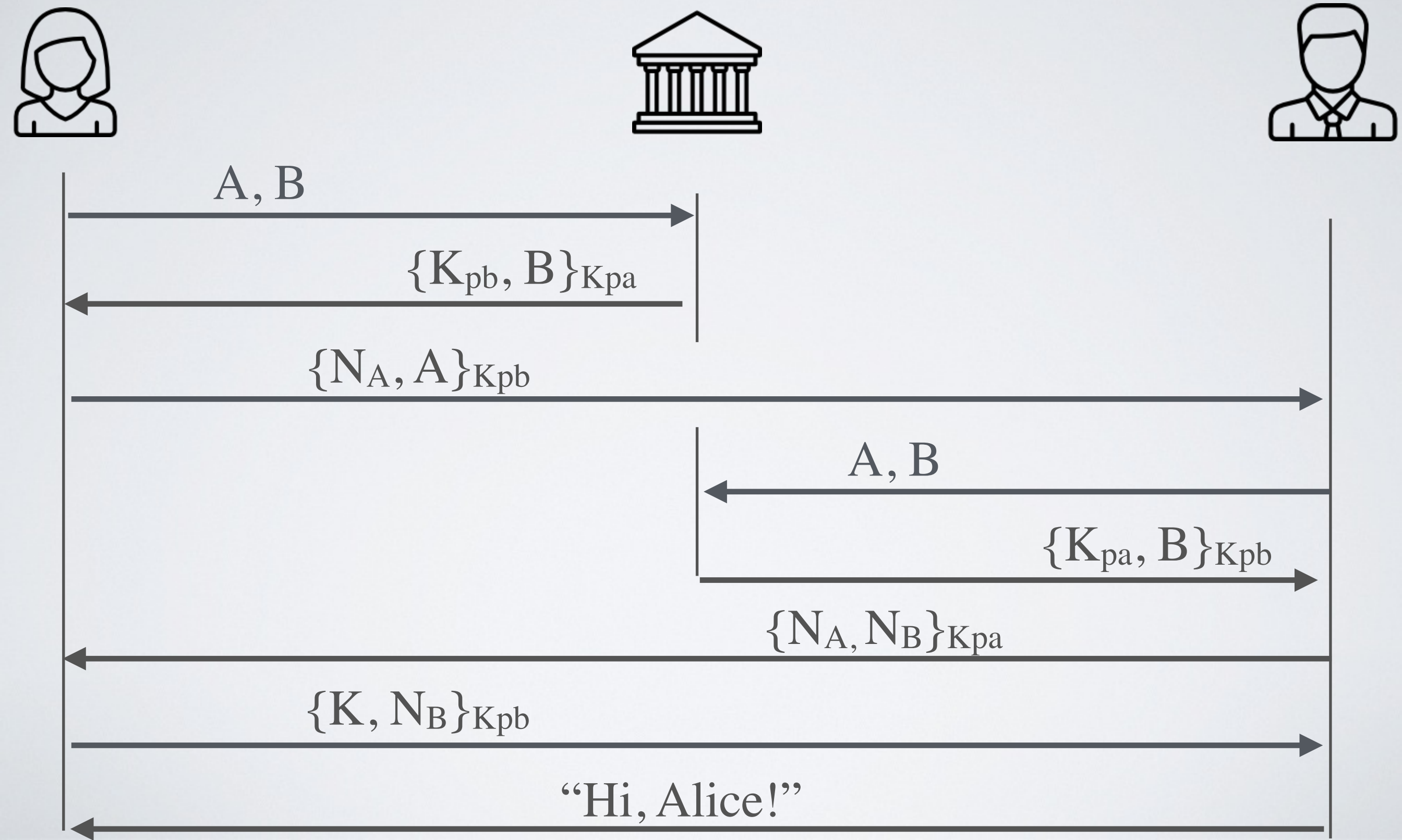
- 4 principals : Alice, Bob, Mallory and a Public-Key Server
- Secret keys:  $K_{sa}$ ,  $K_{sb}$ ,  $K_{sm}$
- Public keys:  $K_{pa}$ ,  $K_{pb}$ ,  $K_{pm}$
- A, B, M and S talk to each other using the same protocol

## Goals

When two parties want to engage in the communication, they want to

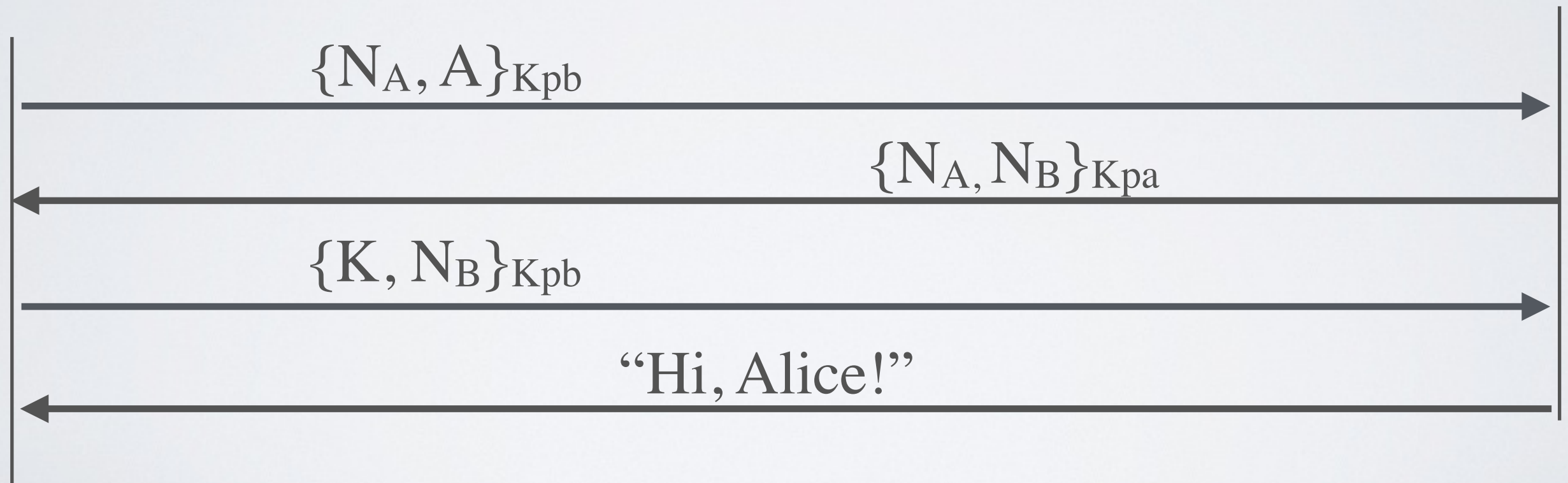
1. make sure that they talk to the right person (authentication)
2. establish a session key  $K$

# The vulnerable Needham-Shroeder public-key Protocol (1978)

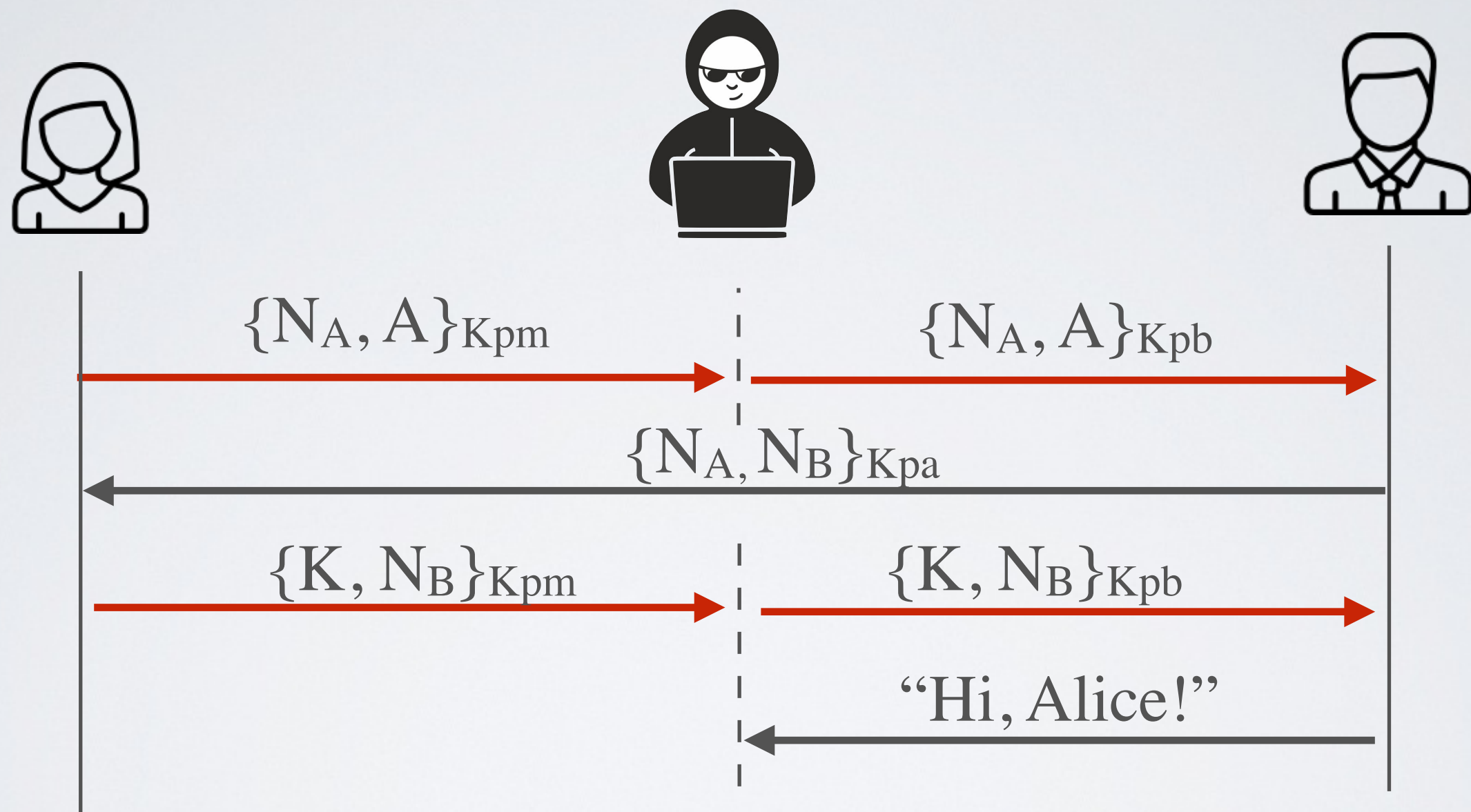




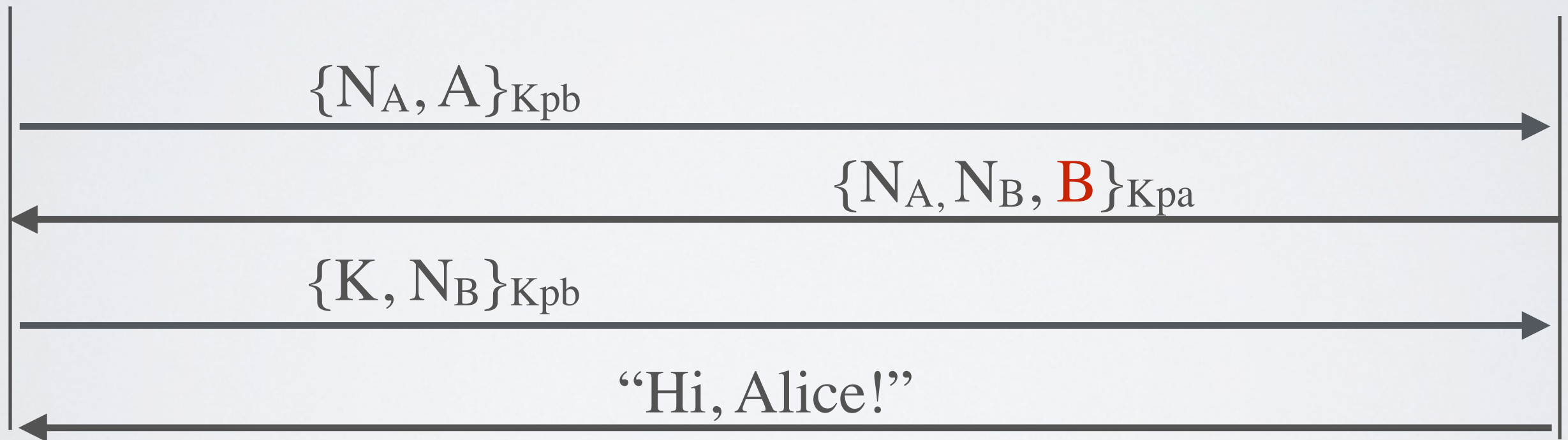
# The simplified but yet vulnerable Needham-Shroeder public-key Protocol (1978)



# Man-in-the-middle attack (Lowe's 1995)



# Lowe's fix (1995)



# Trust Models



# Two trust models

How to establish the authenticity of the binding between someone and its public key ?

Decentralized trust model

➔ **Web of Trust**

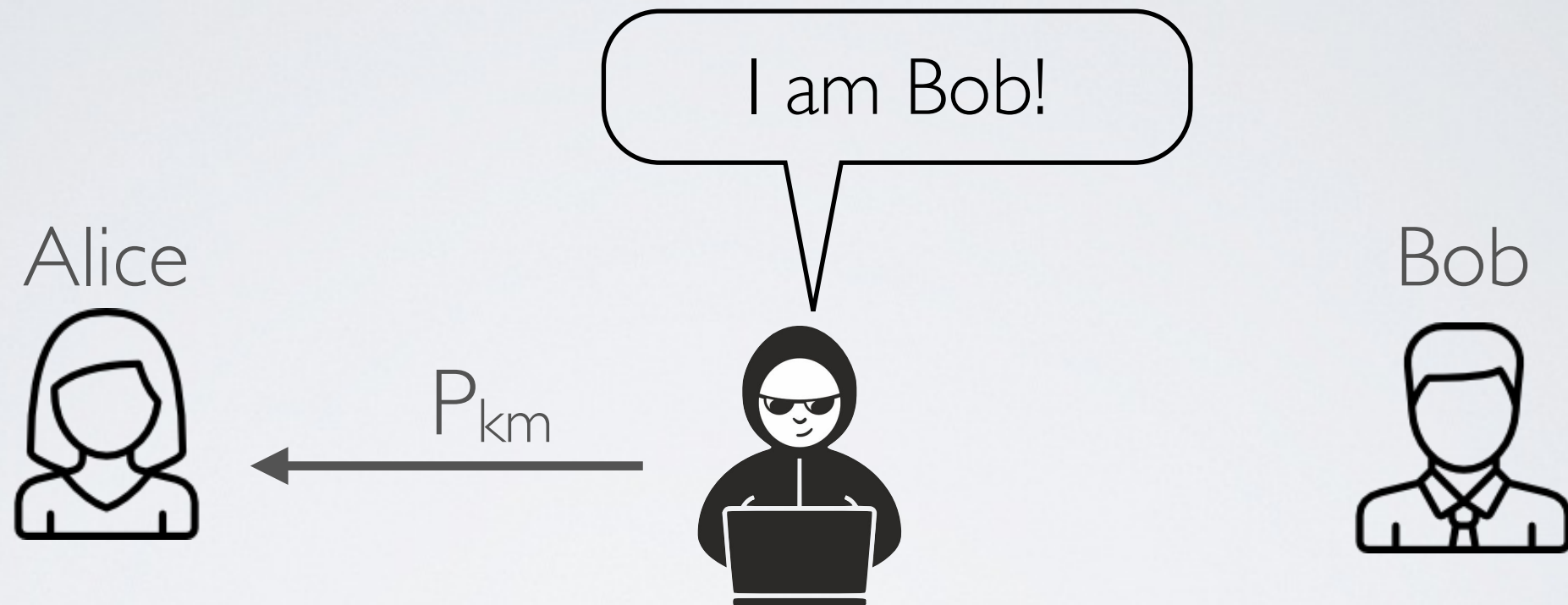


Centralized trust model

➔ **PKI - Public Key Infrastructure**



# Do you trust the GPG key ?

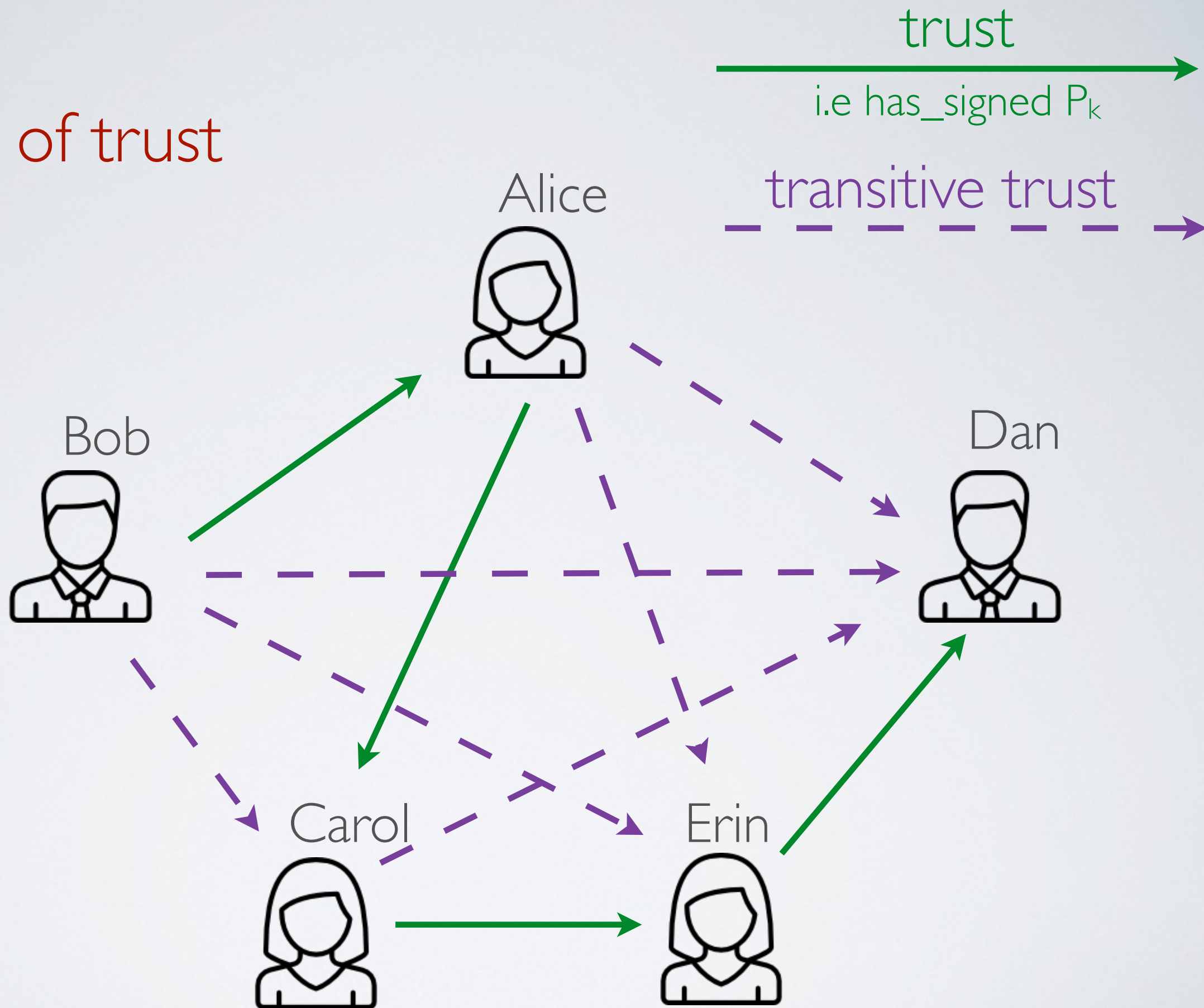


Alice should verify Bob's public key fingerprint

- either by communicating with Bob over another channel
- or by trusting someone that already trusts Bob

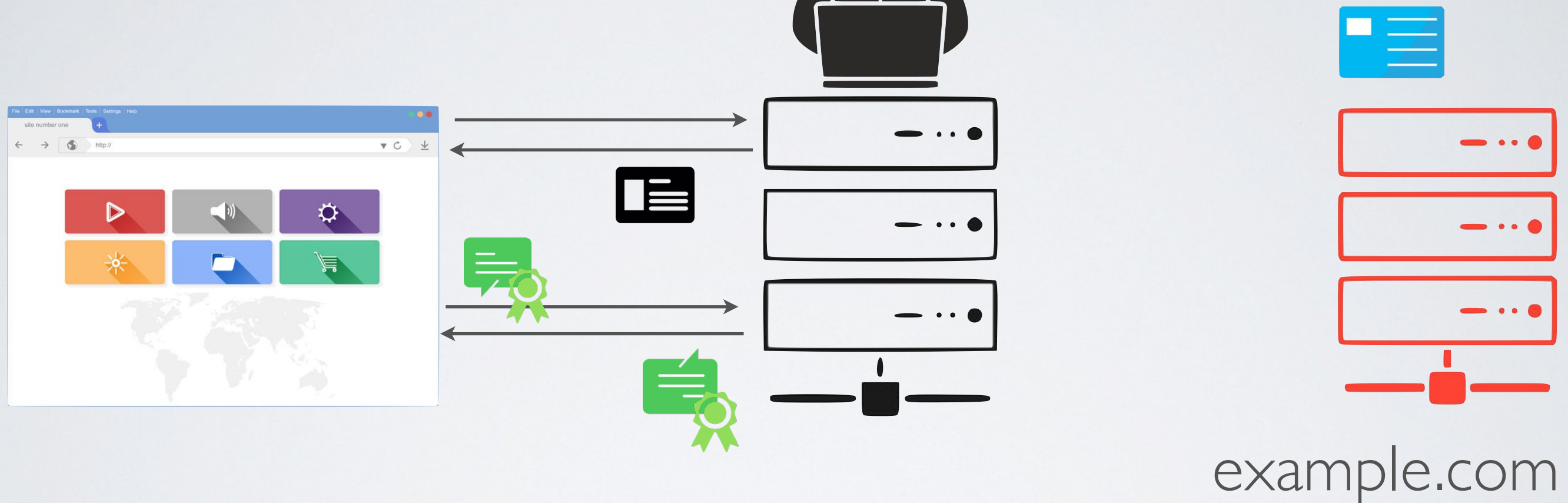
➡ **the web of trust**

# The web of trust



# Do you trust the network ?

I am `example.com`!



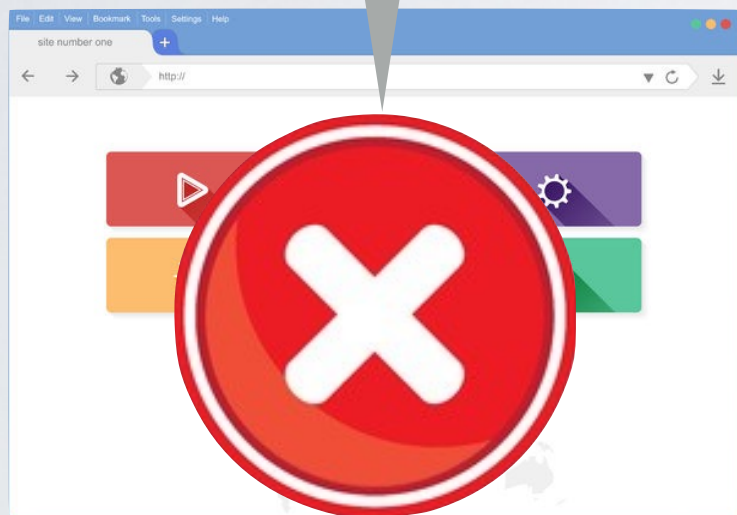
The browser should verify the certificate

➔ **PKI - Public Key Infrastructure**



# Generating and using (self-signed) certificates

**I don't know**



Who are you?



I am example.com





# Self-signed certificates are not trusted by your browser



## Your connection is not private

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

[Hide advanced](#)

[Reload](#)

bitbucket.org normally uses encryption to protect your information. When Chrome tried to connect to bitbucket.org this time, the website sent back unusual and incorrect credentials. Either an attacker is trying to pretend to be bitbucket.org, or a Wi-Fi sign-in screen has interrupted the connection. Your information is still secure because Chrome stopped the connection before any data was exchanged.

You cannot visit bitbucket.org right now because the website uses HSTS. Network errors and attacks are usually temporary, so this page will probably work later.

NET::ERR\_CERT\_DATE\_INVALID



## This Connection is Untrusted

You have asked Firefox to connect securely to **www.domainname.tld** but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

### What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

[Get me out of here!](#)

### ► Technical Details

#### ▼ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

[Add Exception...](#)

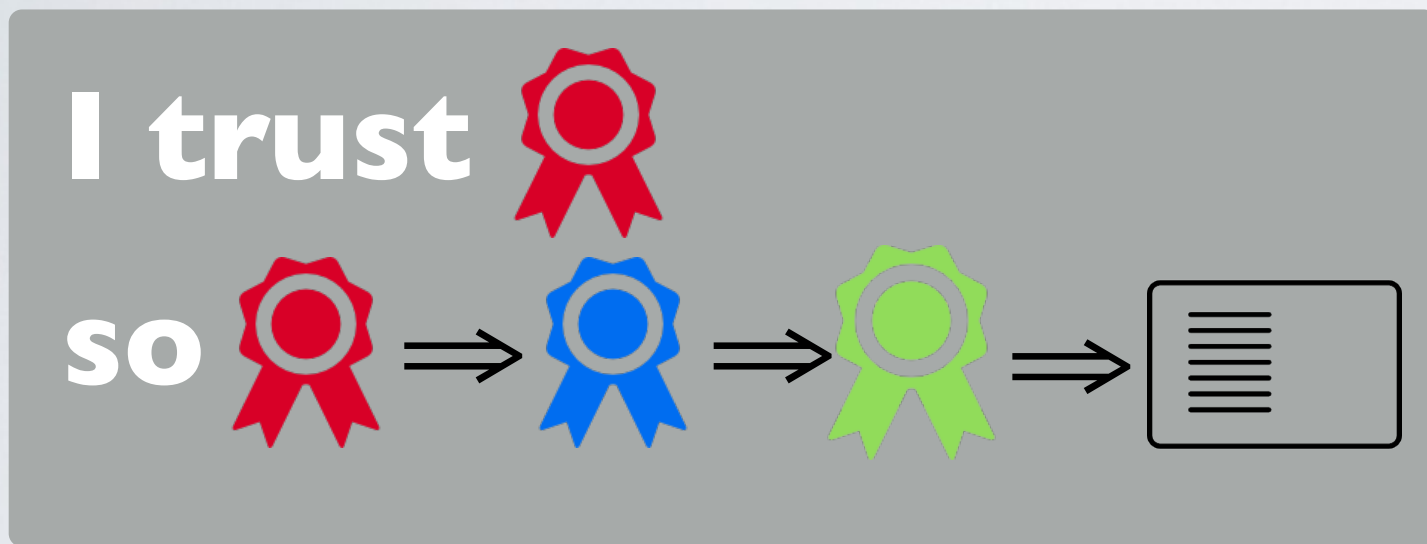


# Signed Certificate

# Certificate Authority (CA)



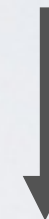
# The Chain of Trust



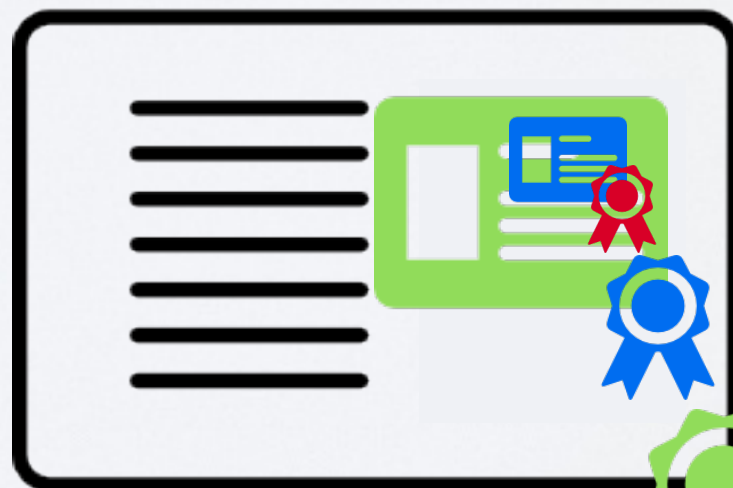
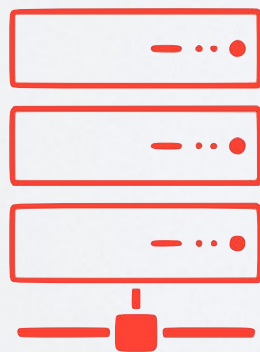
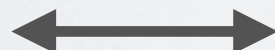
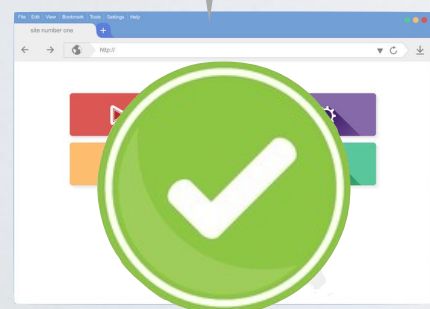
Root CA



Intermediate  
CA



Intermediate  
CA





# Your browser trusts many root CAs **by default**

Keychain Access

Click to unlock the System Roots keychain.

Keychains

- login
- Microsoft\_Int...diate\_Certificates
- Local Items
- System
- System Roots**

Category

- All Items
- Passwords
- Secure Notes
- My Certificates
- Keys
- Certificates**

**GeoTrust Global CA**  
Root certificate authority  
Expires: Saturday, May 21, 2022 at 7:00:00 AM Arabian Standard Time  
✓ This certificate is valid

| Name  | Kind               | Expires                         | Keychain            |
|---|--------------------|---------------------------------|---------------------|
| Echoworx Root CA2                                       | certificate        | Oct 7, 2030, 1:49:13 PM         | System Roots        |
| EE Certification Centre Root CA                         | certificate        | Dec 18, 2030, 2:59:59 AM        | System Roots        |
| Entrust Root Certification Authority                    | certificate        | Nov 27, 2026, 11:53:42 PM       | System Roots        |
| Entrust Root Certification Authority - EC1              | certificate        | Dec 18, 2037, 6:55:36 PM        | System Roots        |
| Entrust Root Certification Authority - G2               | certificate        | Dec 7, 2030, 8:55:54 PM         | System Roots        |
| Entrust.net Certification Authority (2048)              | certificate        | Dec 24, 2019, 9:20:51 PM        | System Roots        |
| Entrust.net Certification Authority (2048)              | certificate        | Jul 24, 2029, 5:15:12 PM        | System Roots        |
| ePKI Root Certification Authority                       | certificate        | Dec 20, 2034, 5:31:27 AM        | System Roots        |
| Federal Common Policy CA                                | certificate        | Dec 1, 2030, 7:45:27 PM         | System Roots        |
| <b>GeoTrust Global CA</b>                               | <b>certificate</b> | <b>May 21, 2022, 7:00:00 AM</b> | <b>System Roots</b> |
| GeoTrust Primary Certification Authority                | certificate        | Jul 17, 2036, 2:59:59 AM        | System Roots        |
| GeoTrust Primary Certification Authority - G2           | certificate        | Jan 19, 2038, 2:59:59 AM        | System Roots        |
| GeoTrust Primary Certification Authority - G3           | certificate        | Dec 2, 2037, 2:59:59 AM         | System Roots        |
| Global Chambersign Root                                 | certificate        | Sep 30, 2037, 7:14:18 PM        | System Roots        |
| Global Chambersign Root - 2008                          | certificate        | Jul 31, 2038, 3:31:40 PM        | System Roots        |
| GlobalSign  | certificate        | Mar 18, 2029, 1:00:00 PM        | System Roots        |
| GlobalSign  | certificate        | Jan 19, 2038, 6:14:07 AM        | System Roots        |
| GlobalSign  | certificate        | Jan 19, 2038, 6:14:07 AM        | System Roots        |
| GlobalSign  | certificate        | Dec 15, 2021, 11:00:00 AM       | System Roots        |
| GlobalSign Root CA                                      | certificate        | Jan 28, 2028, 3:00:00 PM        | System Roots        |
| Go Daddy Class 2 Certification Authority                | certificate        | Jun 29, 2034, 8:06:20 PM        | System Roots        |
| Go Daddy Root Certificate Authority - G2                | certificate        | Jan 1, 2038, 2:59:59 AM         | System Roots        |
| Government Root Certification Authority                 | certificate        | Dec 31, 2037, 6:59:59 PM        | System Roots        |
| Hellenic Academic and Research Institutions RootCA 2011 | certificate        | Dec 1, 2031, 4:49:52 PM         | System Roots        |
| Hongkong Post Root CA 1                                 | certificate        | May 15, 2023, 7:52:29 AM        | System Roots        |

177 items

# Real attacks

## Google Security Blog

The latest news and insights from Google on security and safety on the Internet

### An update on attempted man-in-the-middle attacks

August 29, 2011

Posted by Heather Adkins, Information Security Manager

Today we received reports of attempted SSL man-in-the-middle (MITM) attacks against Google users, whereby someone tried to get between them and encrypted Google services. The people affected were primarily located in Iran. The attacker used a fraudulent SSL certificate issued by DigiNotar, a root certificate authority that should not issue certificates for Google (and has since revoked it).

Google Chrome users were protected from this attack because Chrome was able to [detect](#) the fraudulent certificate.

## Google Security Blog

The latest news and insights from Google on security and safety on the Internet

### Enhancing digital certificate security

January 3, 2013

Posted by Adam Langley, Software Engineer

Late on December 24, Chrome detected and blocked an unauthorized digital certificate for the "\*.google.com" domain. We investigated immediately and found the certificate was issued by an [intermediate certificate authority](#) (CA) linking back to TURKTRUST, a Turkish certificate authority. Intermediate CA certificates carry the full authority of the CA, so anyone who has one can use it to create a certificate for any website they wish to impersonate.



# Limitation of secure channels

