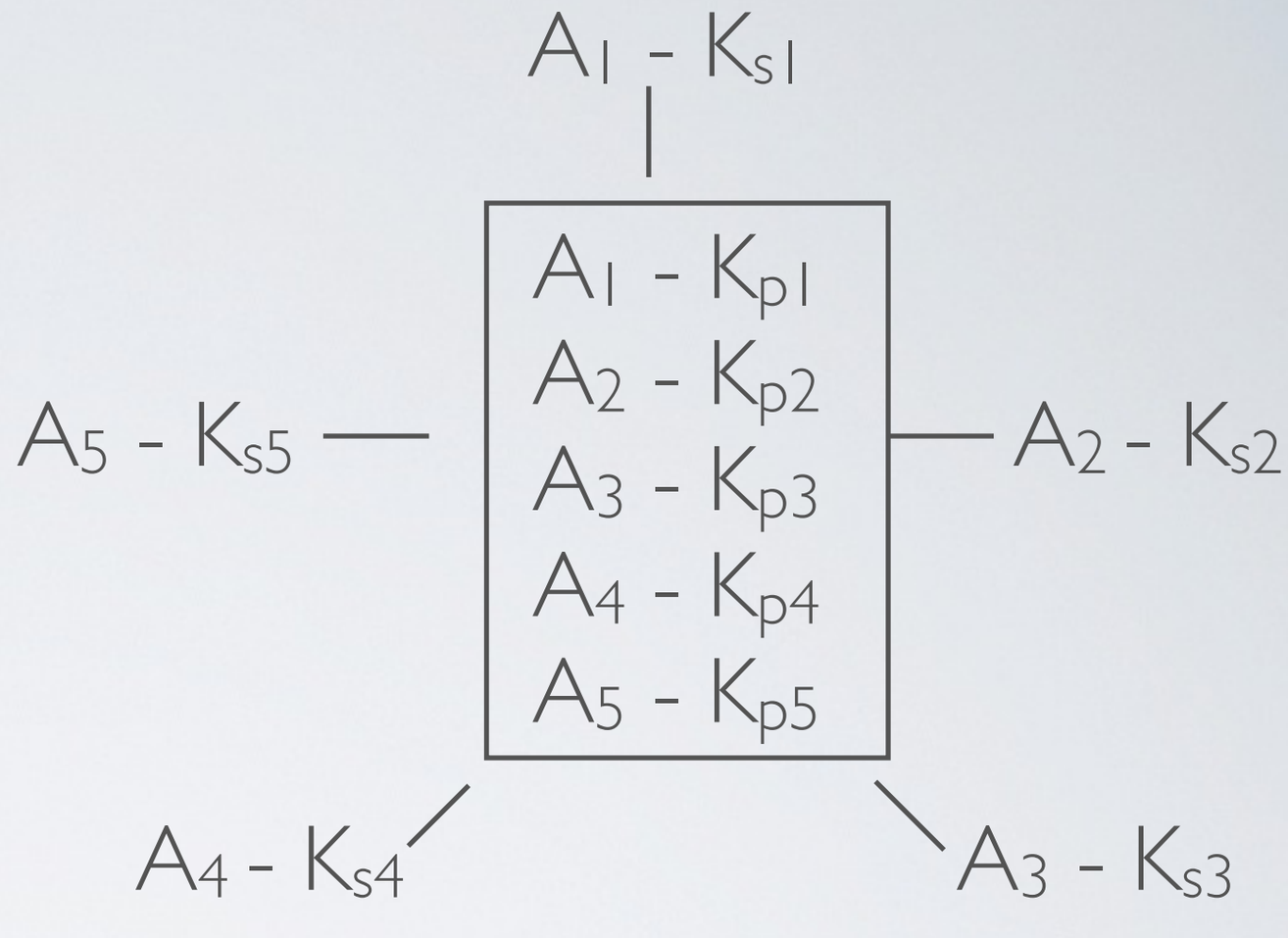
Asymmetric Protocols

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-Schroeder 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)



A, B

$\{K_{pb}, B\}_{Kss}$

$\{N_A, A\}_{Kpb}$

B, A

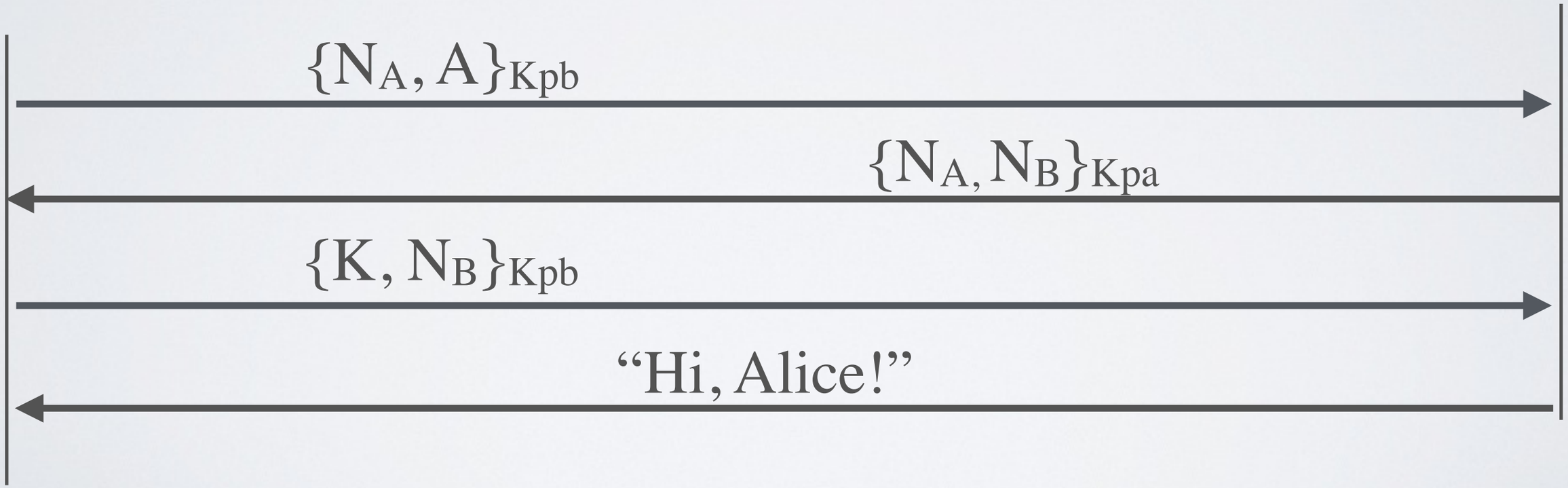
$\{K_{pa}, A\}_{Kss}$

$\{N_A, N_B\}_{Kpa}$

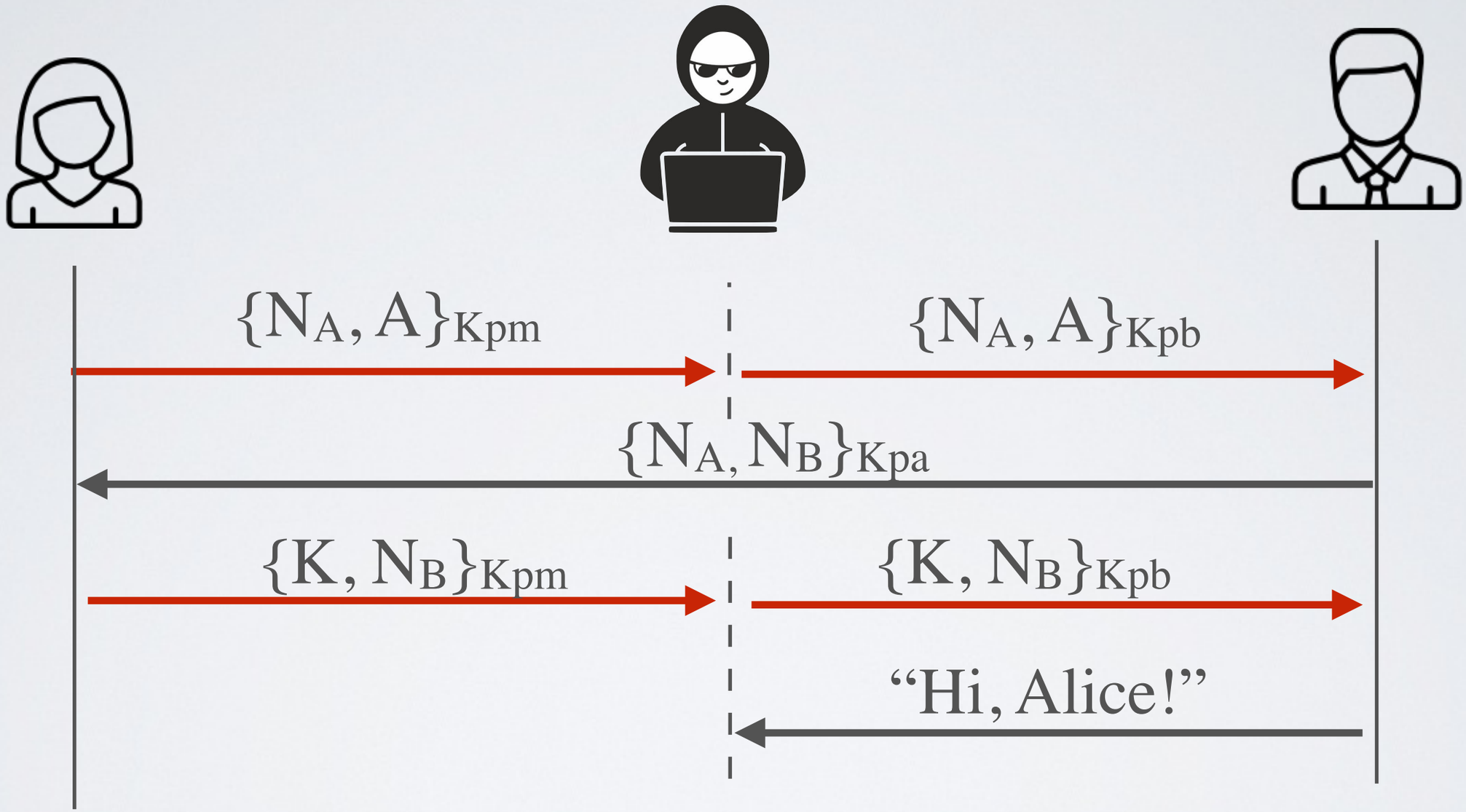
$\{K, N_B\}_{Kpb}$

“Hi, Alice!”

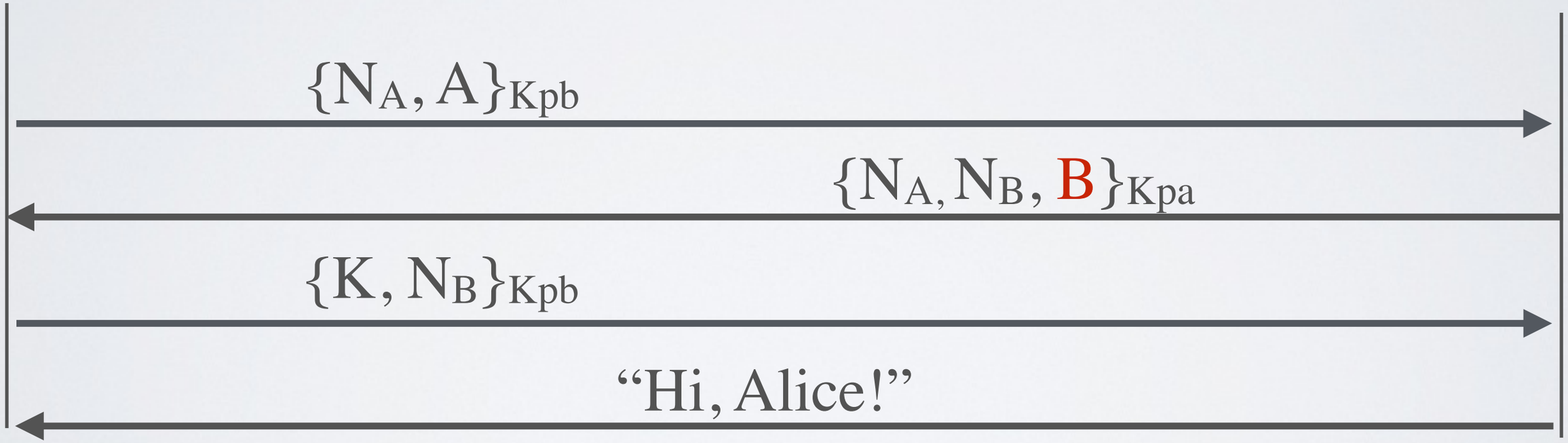
The simplified but yet vulnerable Needham-Shroeder public-key Protocol (1978)



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 ?

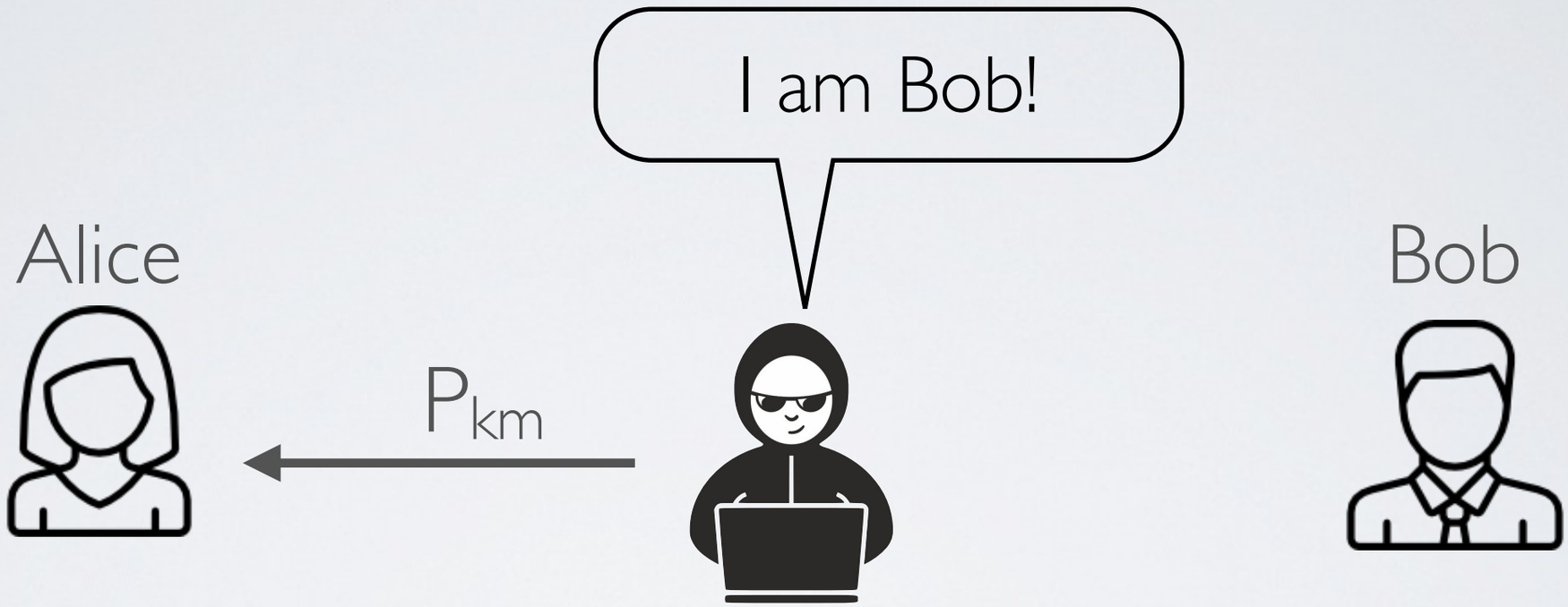
Decentralized trust model

➡ **Web of Trust**

Centralized trust model

➡ **PKI - Public Key Infrastructure**

Do you trust the GPG key ?

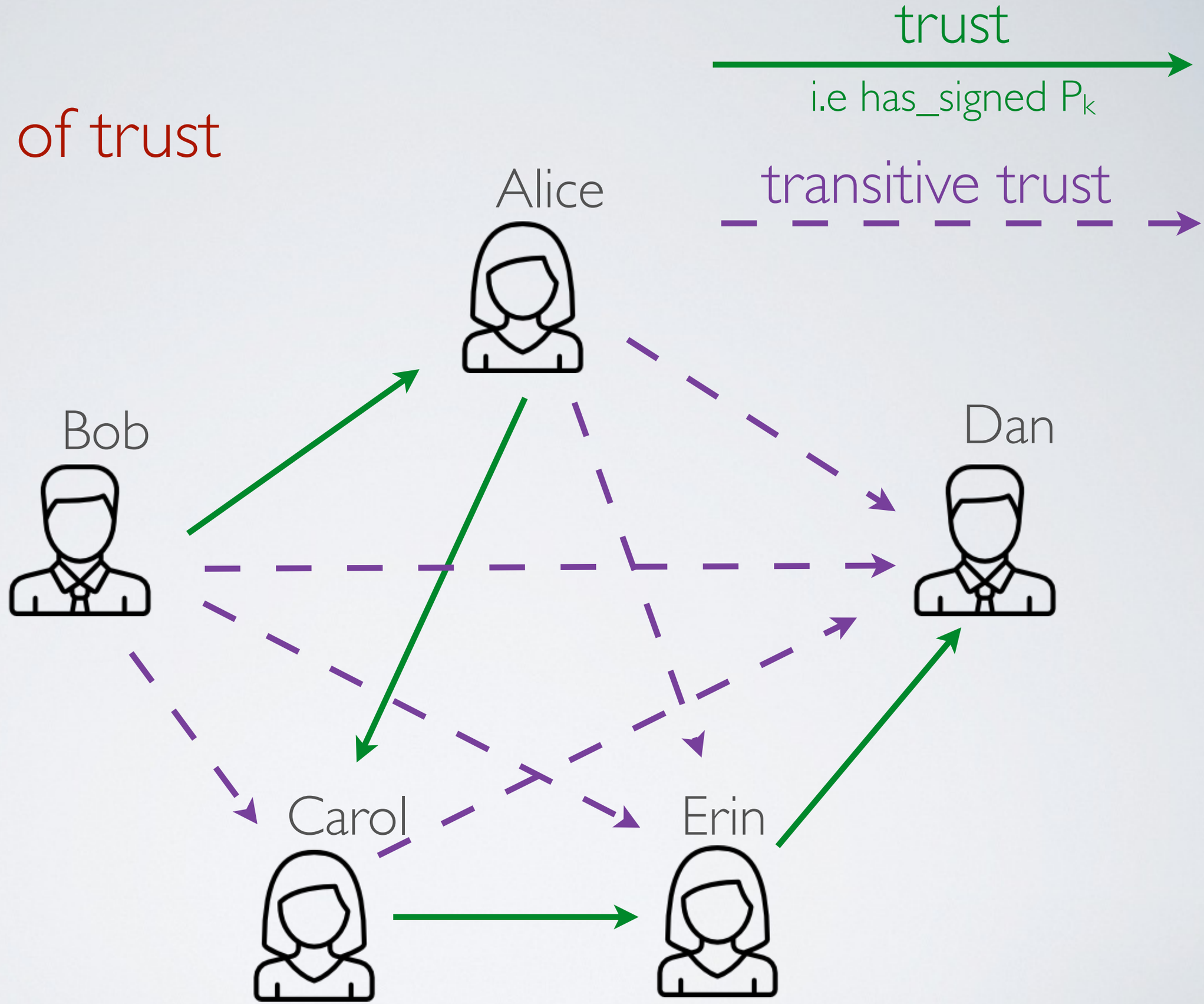


Alice should verify Bob's public key fingerprint

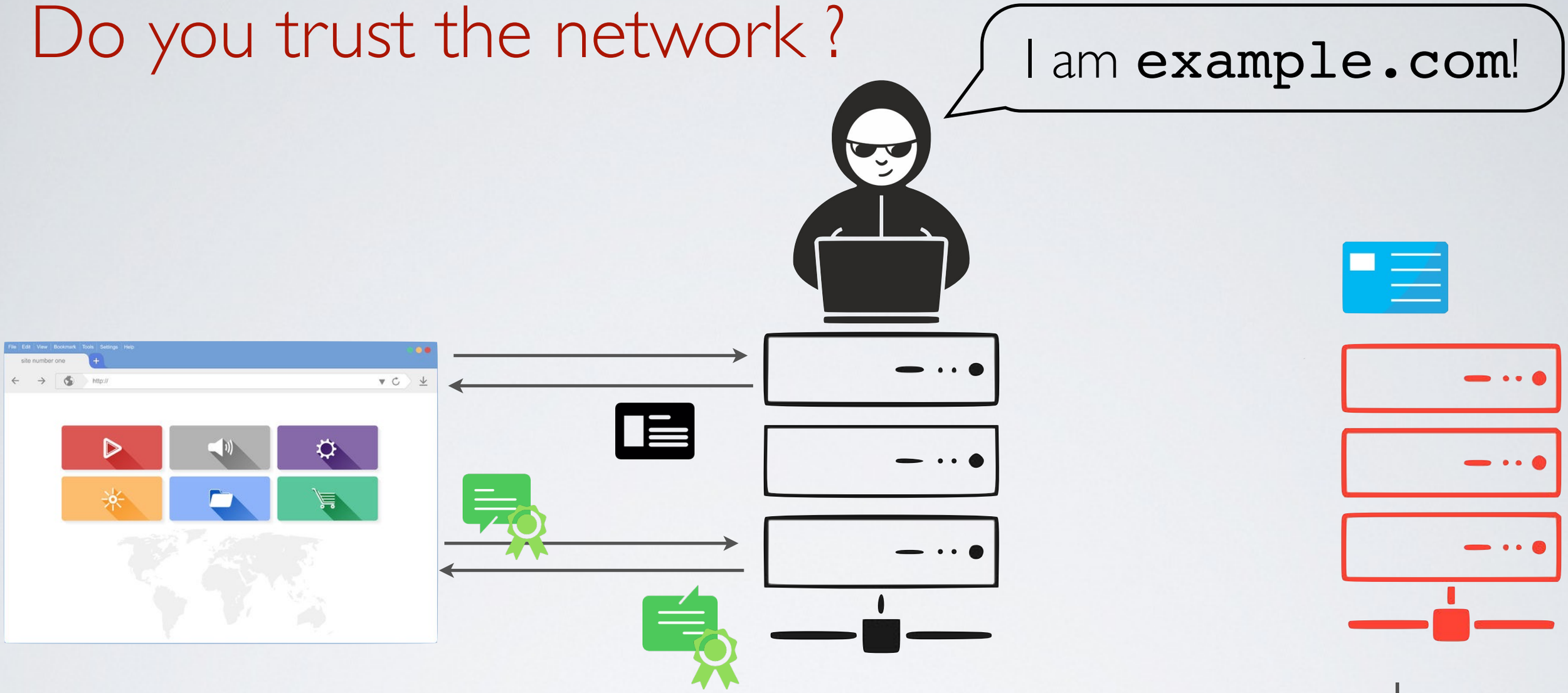
- either by communicating with Bob over another channel
- or by trusting someone that already trusts Bob

➡ **the web of trust**

The web of trust



Do you trust the network ?



The browser should verify the certificate

➡ **PKI - Public Key Infrastructure**

Generating and using (self-signed) certificates

Self-signed certificates are not trusted by your browser

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...

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

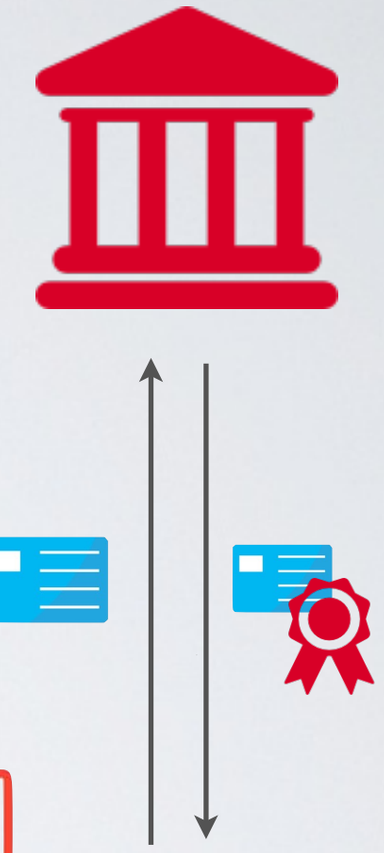
bitbucket.org normally uses encryption to protect your information. When Chrome tried to connect to bitbucket.org this time, the website sent back unusual and incorrect credentials. Either an attacker is trying to pretend to be bitbucket.org, or a Wi-Fi sign-in screen has interrupted the connection.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

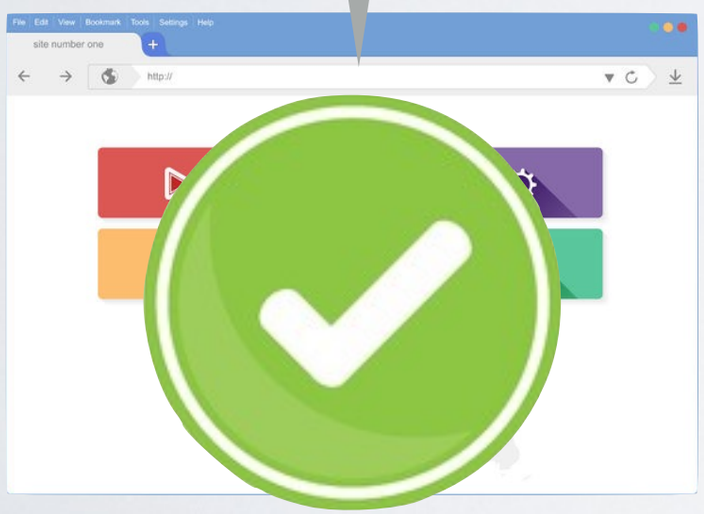
You cannot visit bitbucket.org right now because the website uses HSTS. Network errors and attacks are usually temporary, so this page will probably work later.

NET::ERR_CERT_DATE_INVALID

Signed Certificate

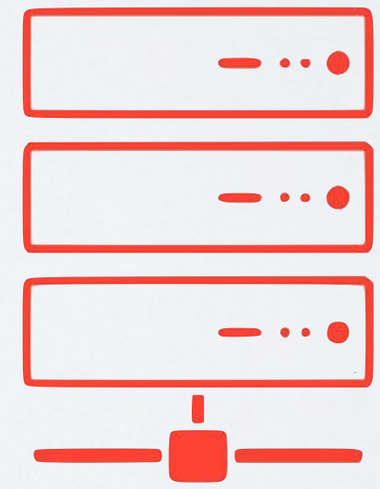
Certificate Authority (CA)



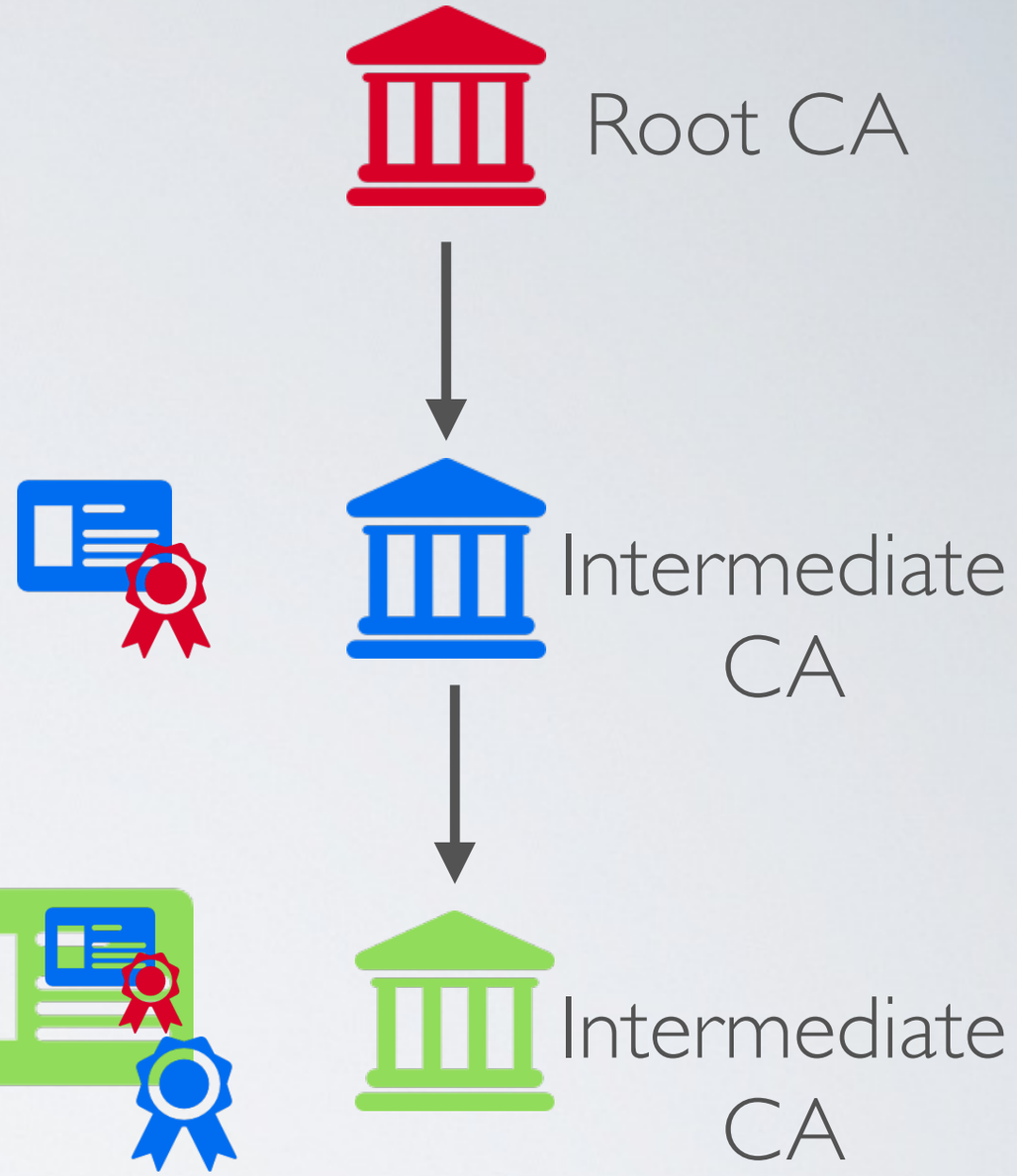


Who are you?

I am example.com

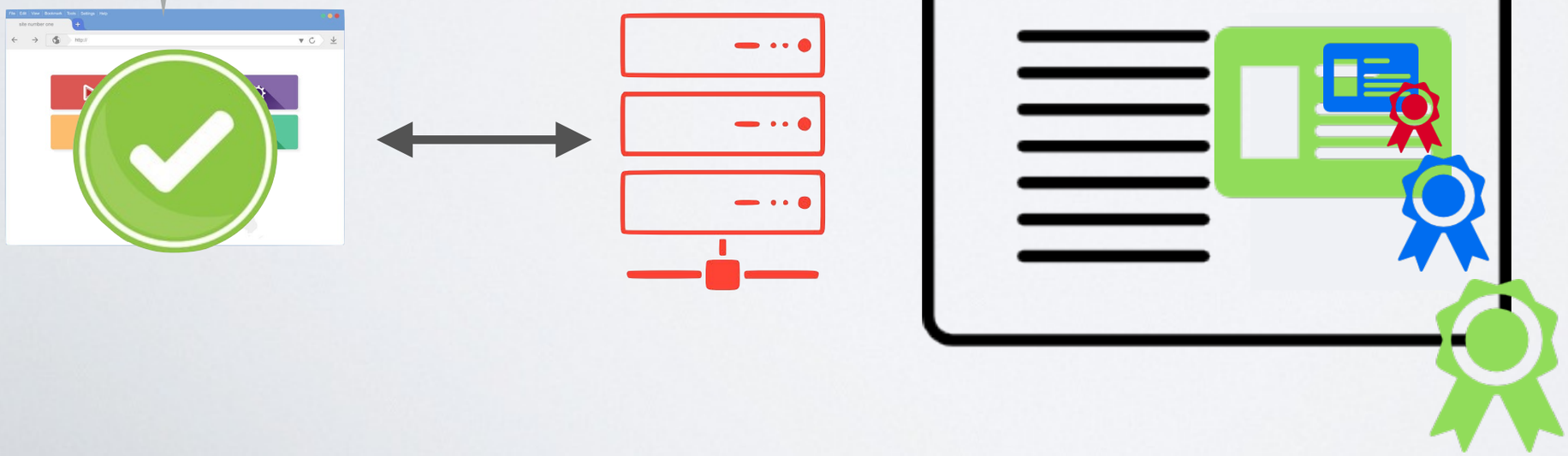


The Chain of Trust

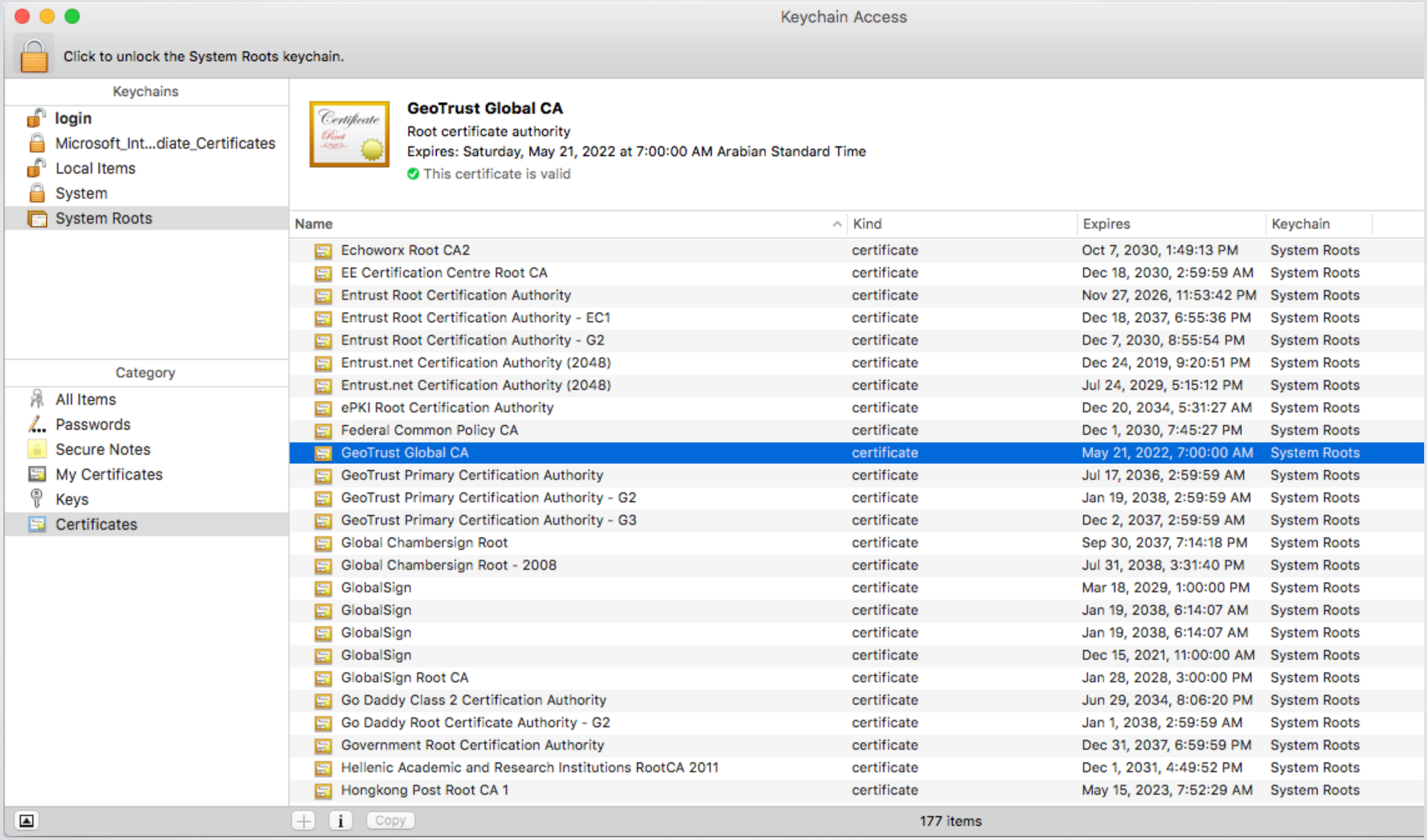


I trust

so



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



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 or

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