Secure transactions

The problems of e-commerce and transaction security

- Problems of remoteness
  - Trust factor between parties
  - Use of sensitive data
  - Atomicity of transaction
- Internet protocol problems
  - Authentication
  - Confidentiality
- Transparence and critical mass problem

Main protocols

- Three main proposals, over the years, for transaction security
  - HTTP over SSL (Secure Socket Layer), or HTTPS
    - ☺ Communication confidentiality/integrity
    - ☺ Mutual authentication
    - ☹ No guarantees on data usage
    - ☹ No strict authentication of client
  - S/HTTP: Secure HTTP, security-oriented extension of HTTP
    - ☹ Missing critical mass support
  - SET (Secure Electronic Transaction)
    - ☺ Guarantees on data usage and transaction security enforcement
    - ☹ Missing critical mass support

Glossary

- **cardholder**: customer using a credit card
- **issuer**: credit card circuit which issued the card
- **merchant**: online vendor
- **acquirer**: the vendor's bank/processing agency which accepts payment requests
- **payment gateway**: the system which the acquirer offers to the merchan to process payment requests

Secure Socket Layer (SSL)

SSL

- Originally designed by Netscape for securing web communication; it became a de facto standard also for other protocols
- IETF standardized TLS, which can be considered version “3.1” of SSL
- SSL enforces:
  - Confidentiality and integrity of the communications
  - Server authentication
  - Client authentication (optionally)
- Uses both symmetric and asymmetric cryptography for performance reasons

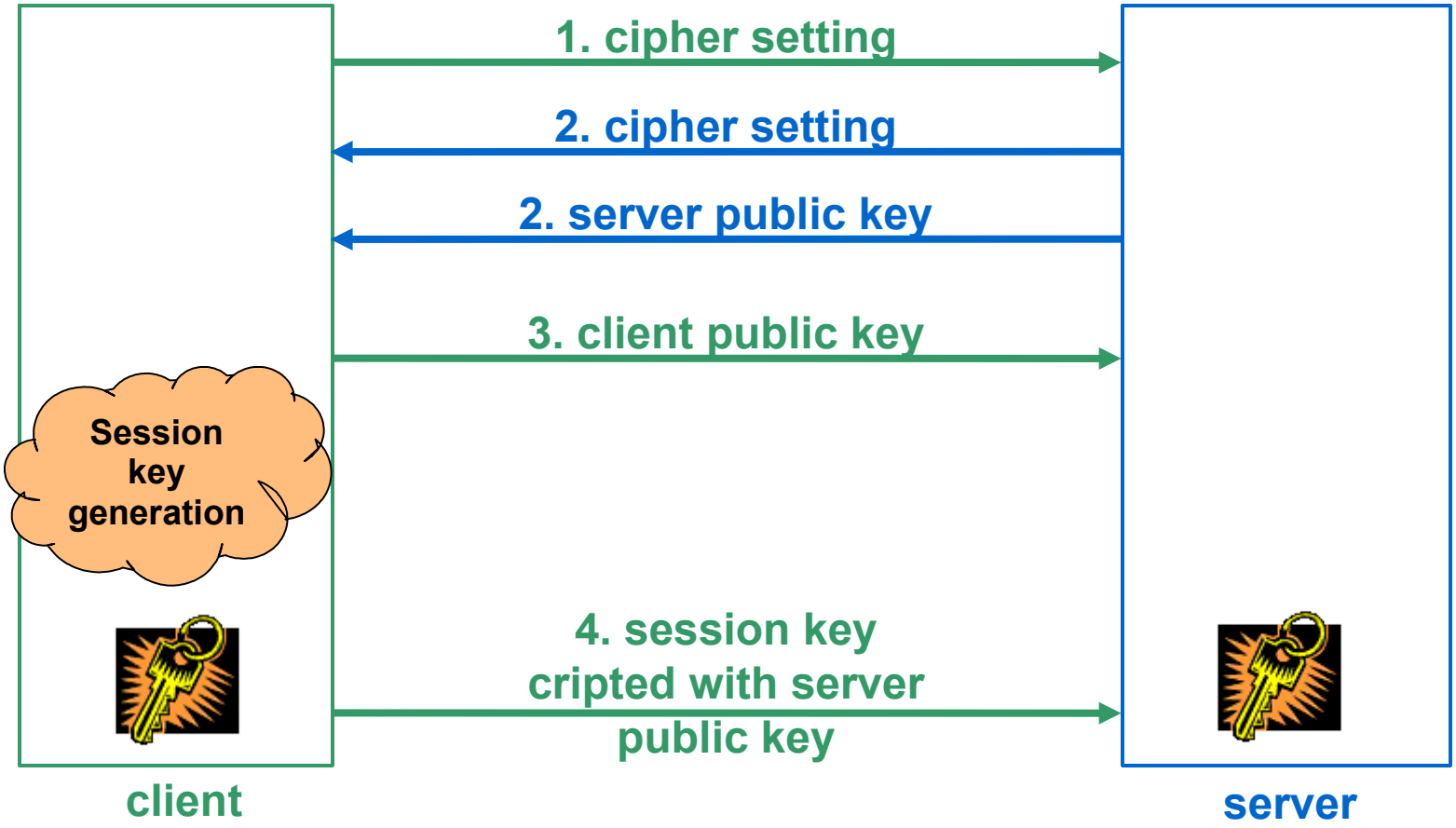
Fasi di SSL

- SSL phases:
  - **handshake**: using asymmetric crypto, client and server agree on a secret, to use in the following phase
  - **Secure connection**: using the secret previously exchanged, client and server communicate securely using symmetric cryptography (more efficient)

Cipher setting

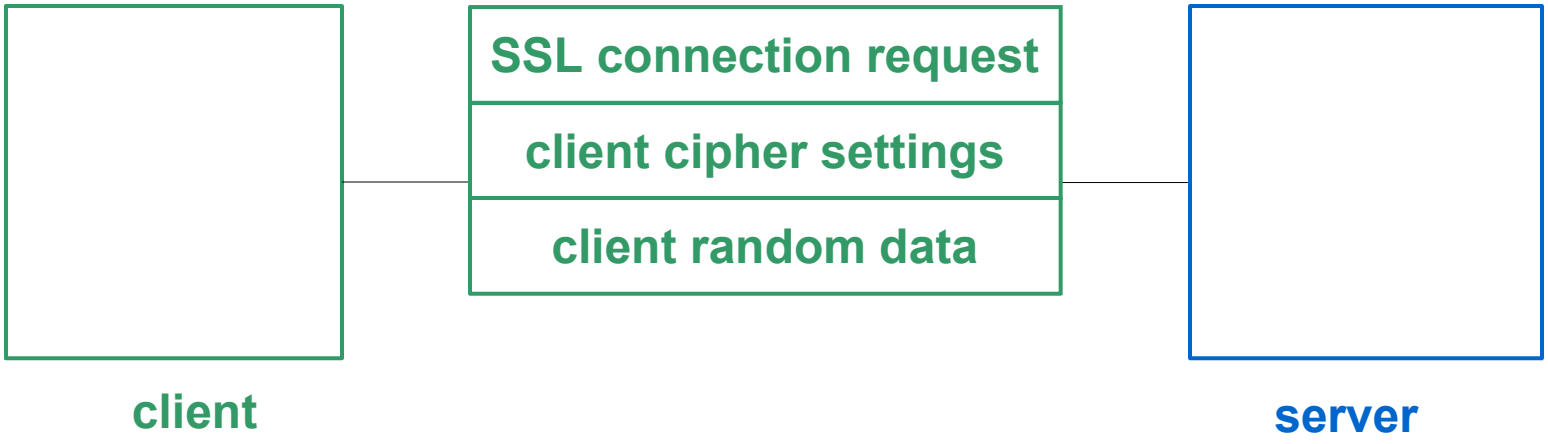
- SSL designed to be flexible wrt technical evolution
- Client and server may use different suites of algorithms for symmetric and asymmetric encryption, as well as for hashing
- The set of algorithm that a server or client is able to use is called its **cipher setting**
- During handshake, cipher settings are compared to agree on shared algorithms
- The standard mandates implementation of a minimal cipher setting: DES, RC2, RC4 and IDEA (symmetric), RSA and DSS (authentication/ signature), SHA-1 and MD5 (hashing), X.509 (certificates), Diffie-Hellman and RSA (key exchange)

SSL handshake: overview



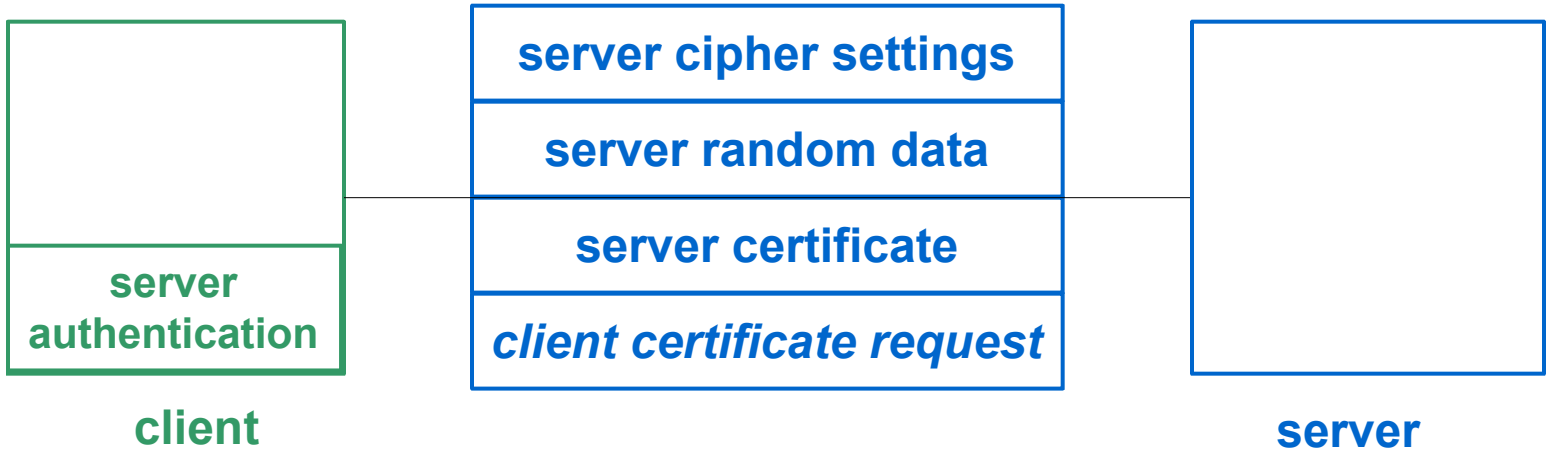
SSL handshake: details

1. Connection Request



SSL handshake: details

2. Connection Response



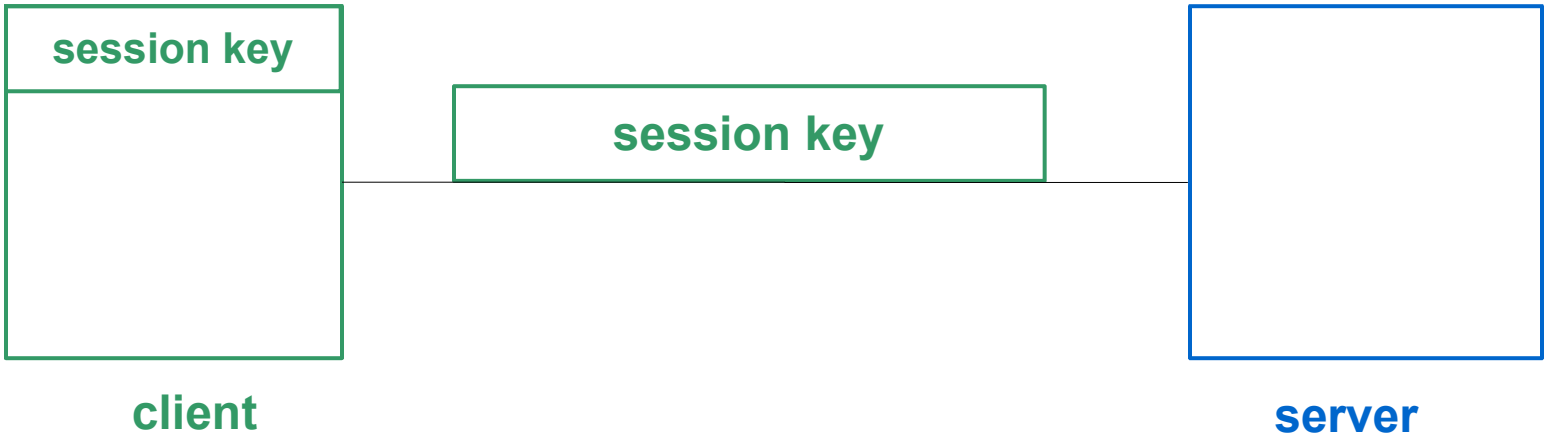
SSL handshake: details

3. Client Authentication

- *Optional step:* if not required, the client sends just an asymmetric key of a disposable pair, without a certificate

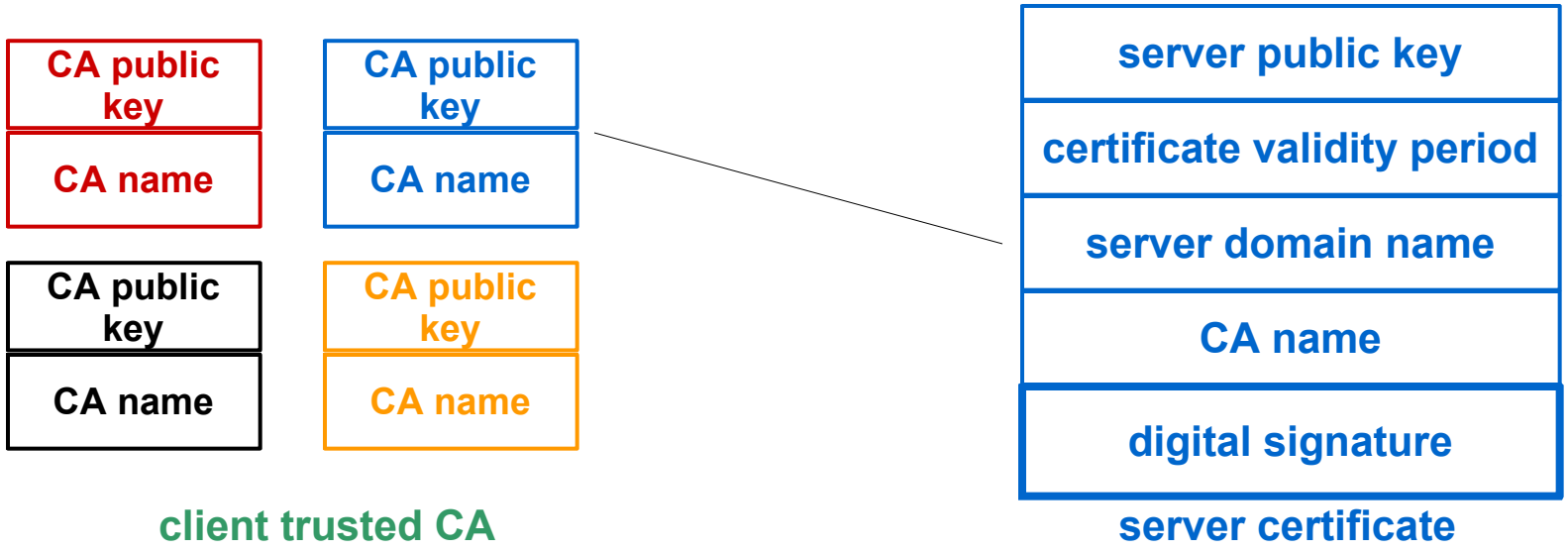
SSL handshake: details

4. Session Key Exchange



- If using D-H: key is already exchanged as a result of exchanging public keys
- If using RSA: **client** generatest a symmetric key, encrypts it with server's public key (because only the server is authenticated)

SSL handshake: server autentication

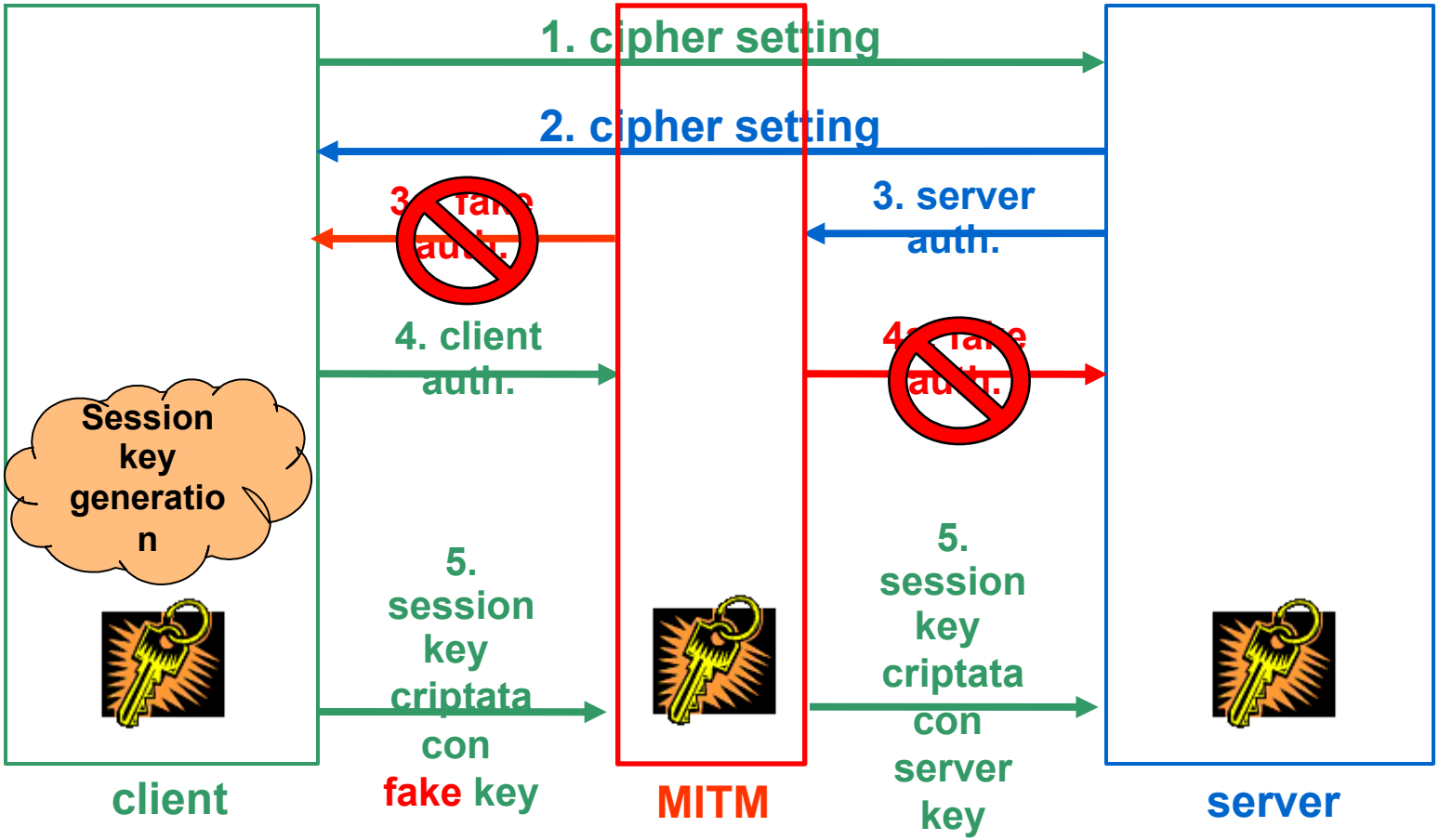


1. Is the certificate in the validity period?
2. Is the server root CA trusted?
3. Is the certificate valid?
4. Is it revoked?
5. *Is the name of the server in the certificate the same which I requested?*

Resistence to *man-in-the-middle*

- Could a MITM attack violate SSL?
- The MITM could, while impersonating the client towards the server and viceversa:
  - Send a fake certificate (i.e. signed by a non-trusted CA)
  - Send a good certificate with a fake name
  - Send a good certificate but substitute the public key (making it invalid)
  - Let everything through

SSL handshake vs. MITM



Example: MITM attack

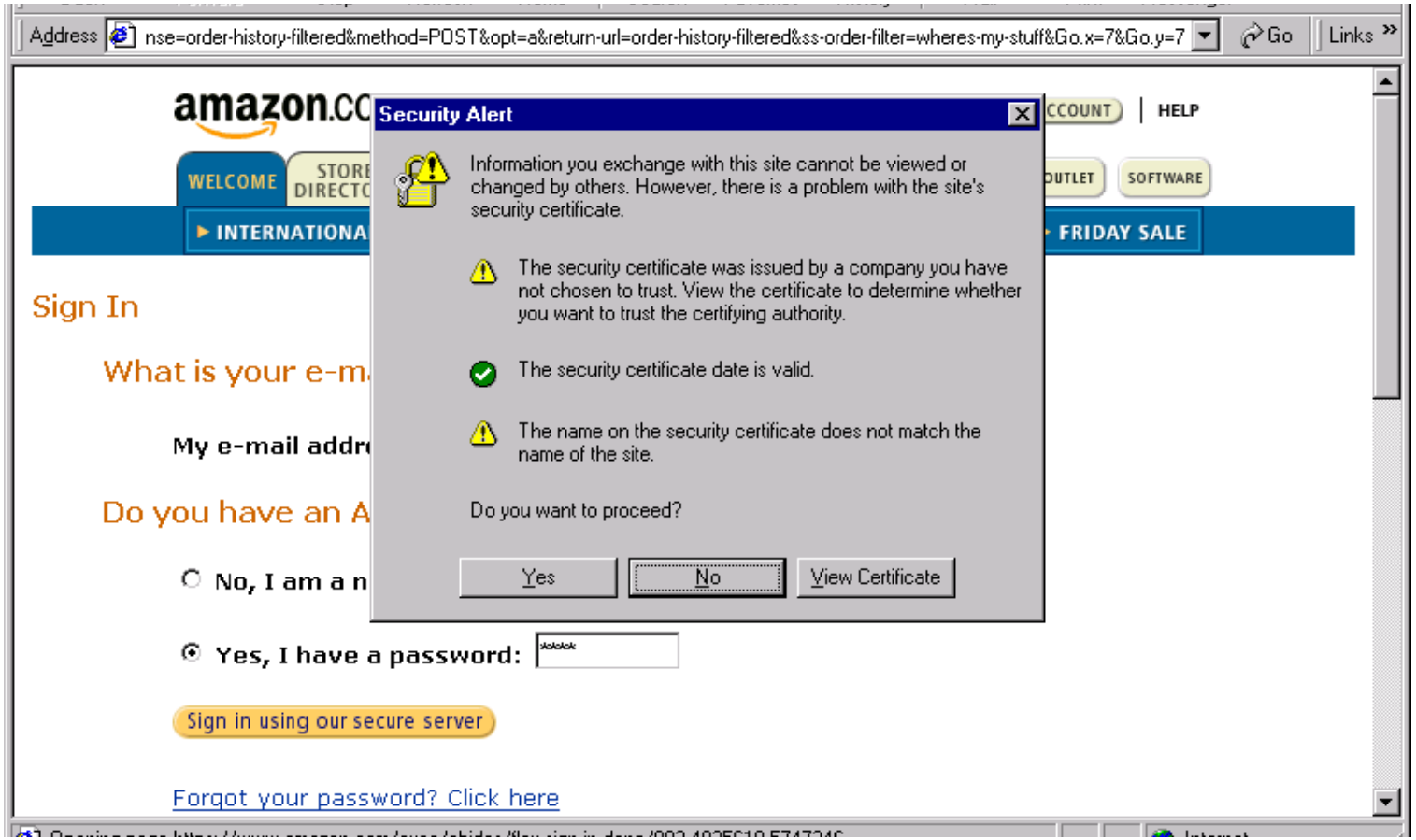
- Using dnsspoof to resolve a website (e.g. www.amazon.com) to the aggressor PC (e.g. 15.1.1.25)

```
C:\>ping www.amazon.com

Pinging www.amazon.com [15.1.1.25] with 32 bytes of data:

Reply from 15.1.1.25: bytes=32 time<10ms TTL=249
Reply from 15.1.1.25: bytes=32 time<10ms TTL=249
Reply from 15.1.1.25: bytes=32 time<10ms TTL=249
Reply from 15.1.1.25: bytes=32 time<10ms TTL=249
```

- Now 15.1.1.25 can act as a proxy, and try to intercept payment data...

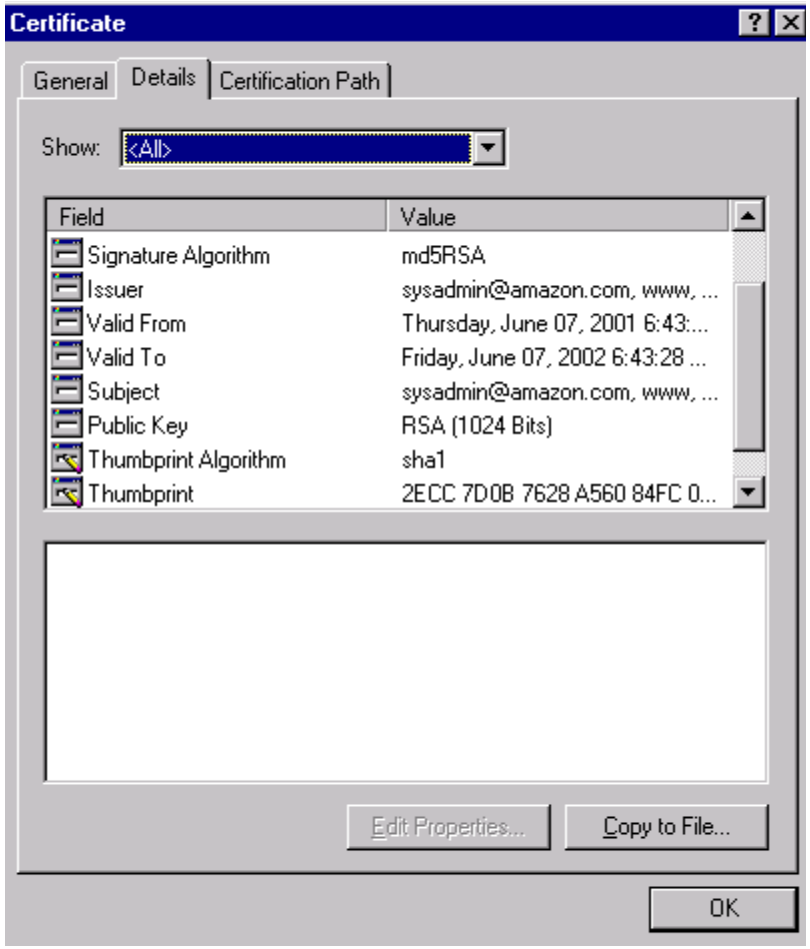
MITM attack (2)

- “webmitm”, a tool in dsniff, allows us to send out a different certificate. Here's the user view:



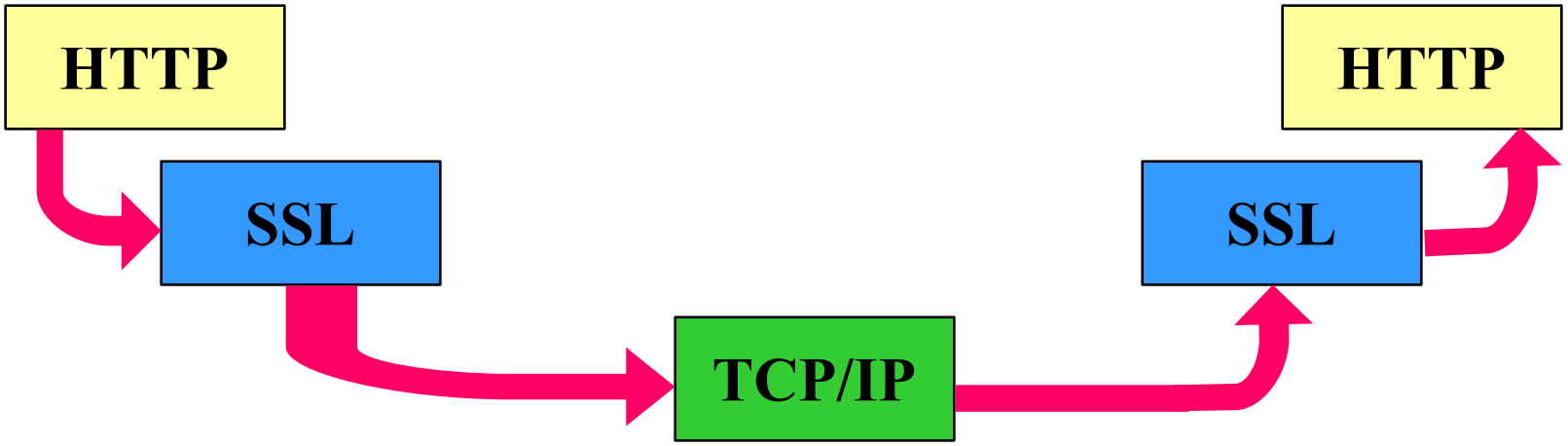
MITM attack (3)

- And if the user “checks” before clicking OK, what is he going to check?



HTTPS

- HTTP over an SSL secure channel
- Assigned to port 443
- S/HTTP is a more powerful, yet unused alternative



SSL summary

It does:

- Protect transmissions
  - Confidentiality
  - Integrity
- Ensures authentication
  - Of server
  - Of client (optoionally)

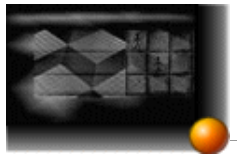
It does not:

- Protect data before transmission or after transmission
  - On server
  - On client (e.g. from trojan)
  - By abuser (e.g. non-honest merchant)
- Protect from PKI violations
  - Usage of insecure certs
  - Key mismanagement
- Ensure against user stupidity

S/HTTP

- S/HTTP is a transaction security protocol developed by IETF, extending and following HTTP specs
- IETF Draft Standard, unused
- CMS/MOSS (Crypto Message Std / Multipart Obj. Security Std) defining headers for encrypted objects
- Uses multiple layers of algorithms, with independence mechanisms
- Most browsers support it, but it has never gained extensive usage

Secure Electronic Transaction (SET)



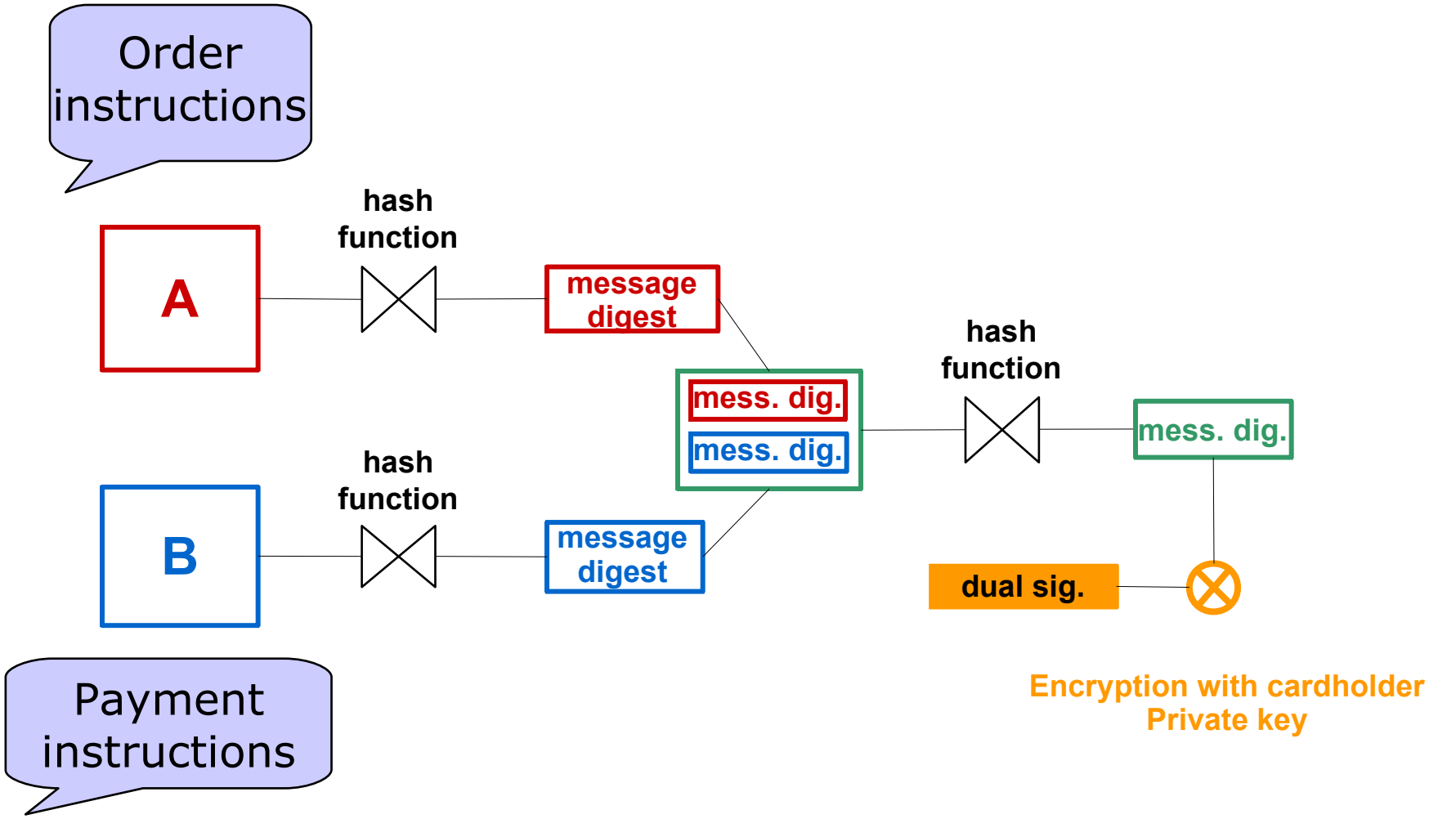
Transactions, not connections!

- Problem:
  - SSL protects connections, not transaction data which is given to the merchant
- Solution
  - Give the *merchant* only order data
  - Make payment data accessible only to the trusted *payment gateway*
  - Have *merchant* deal with the *payment gateway*
  - Have *payment gateway* verify order coherence
- Yes, but how ?

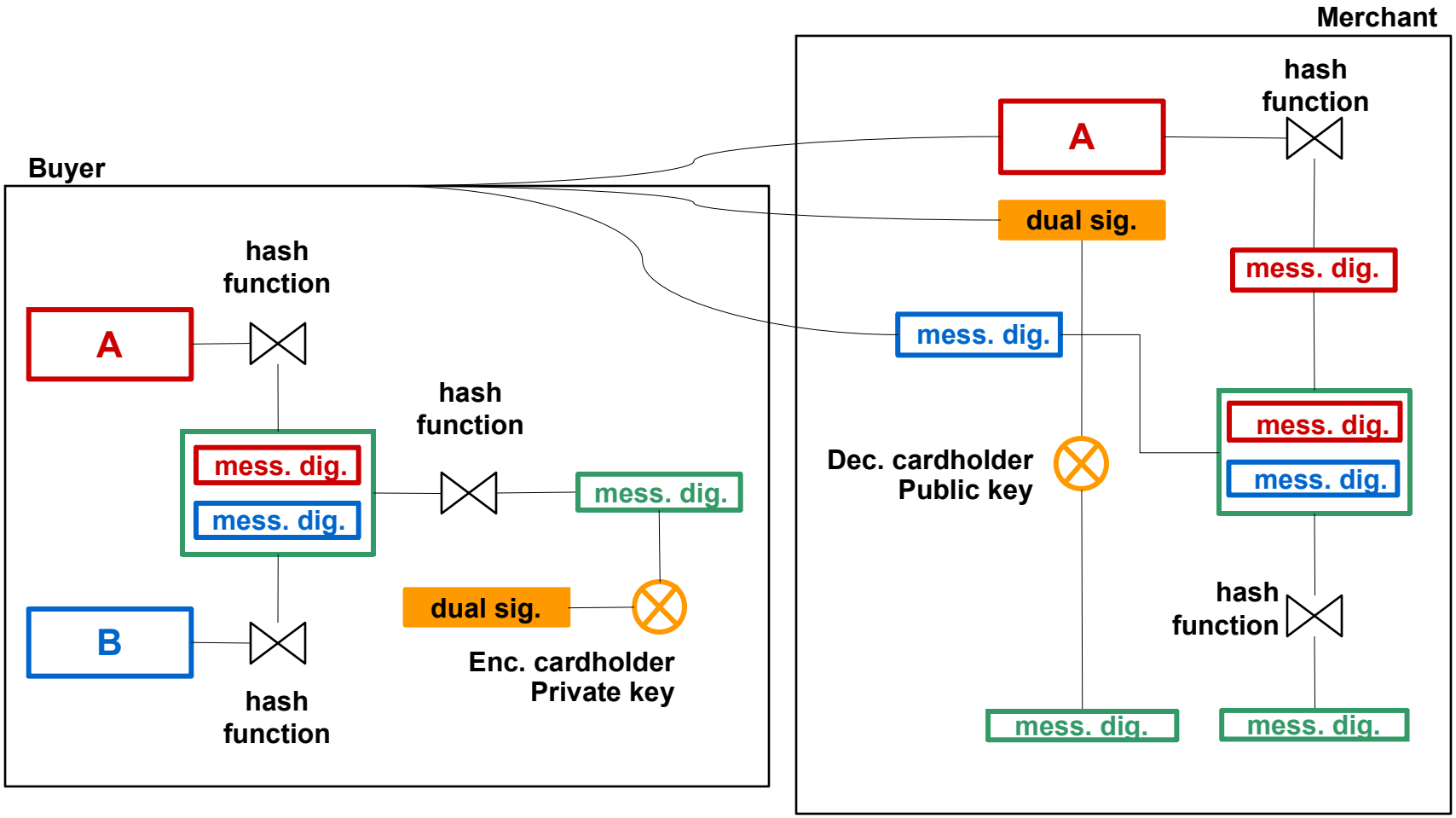
Dual signature

- SET introduces the concept of a dual digital signature
  - A mechanism for interacting with two different subjects, revealing to each only its own portion of data
- Dual signature creates two *message digests*, one per message part, and a combined dual signature

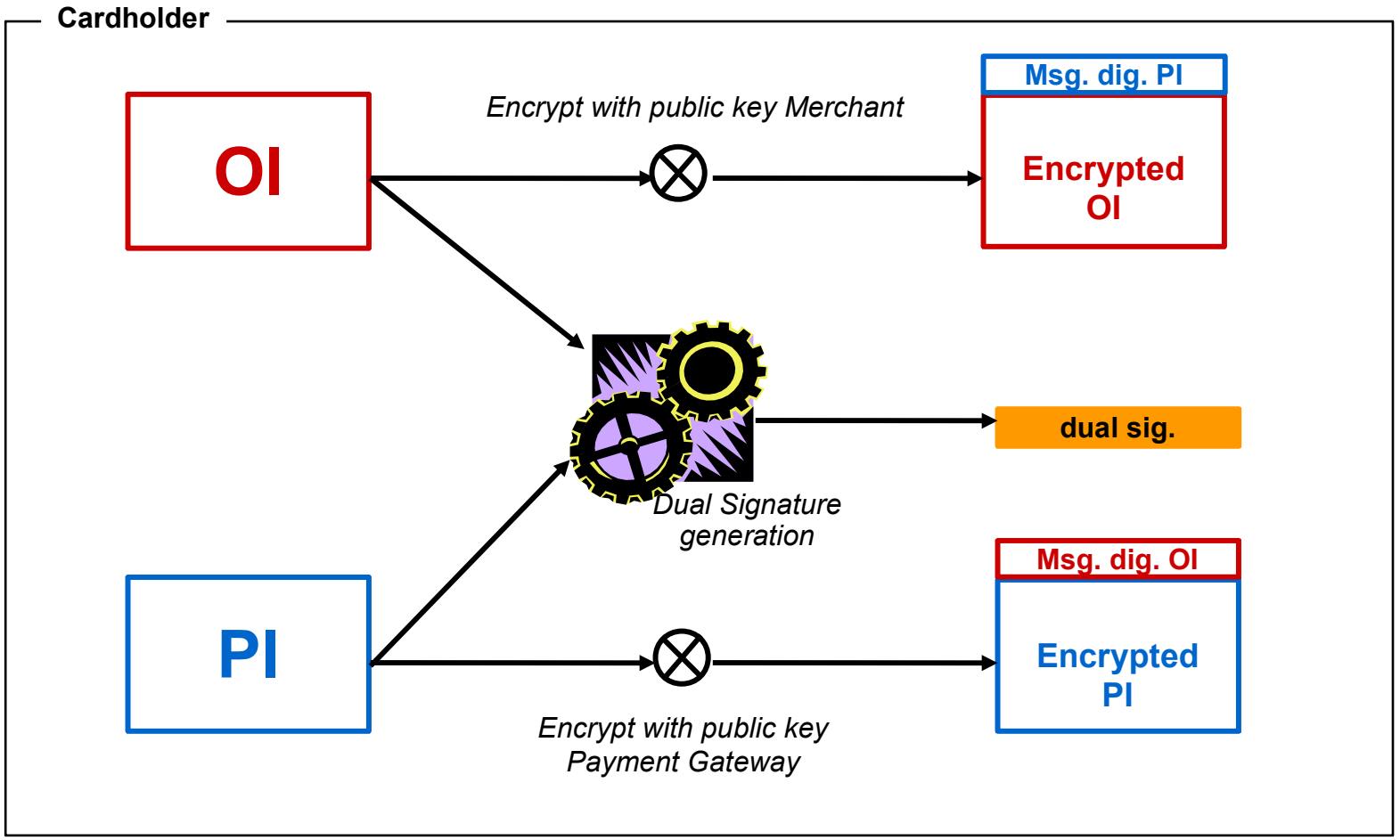
dual signature generation



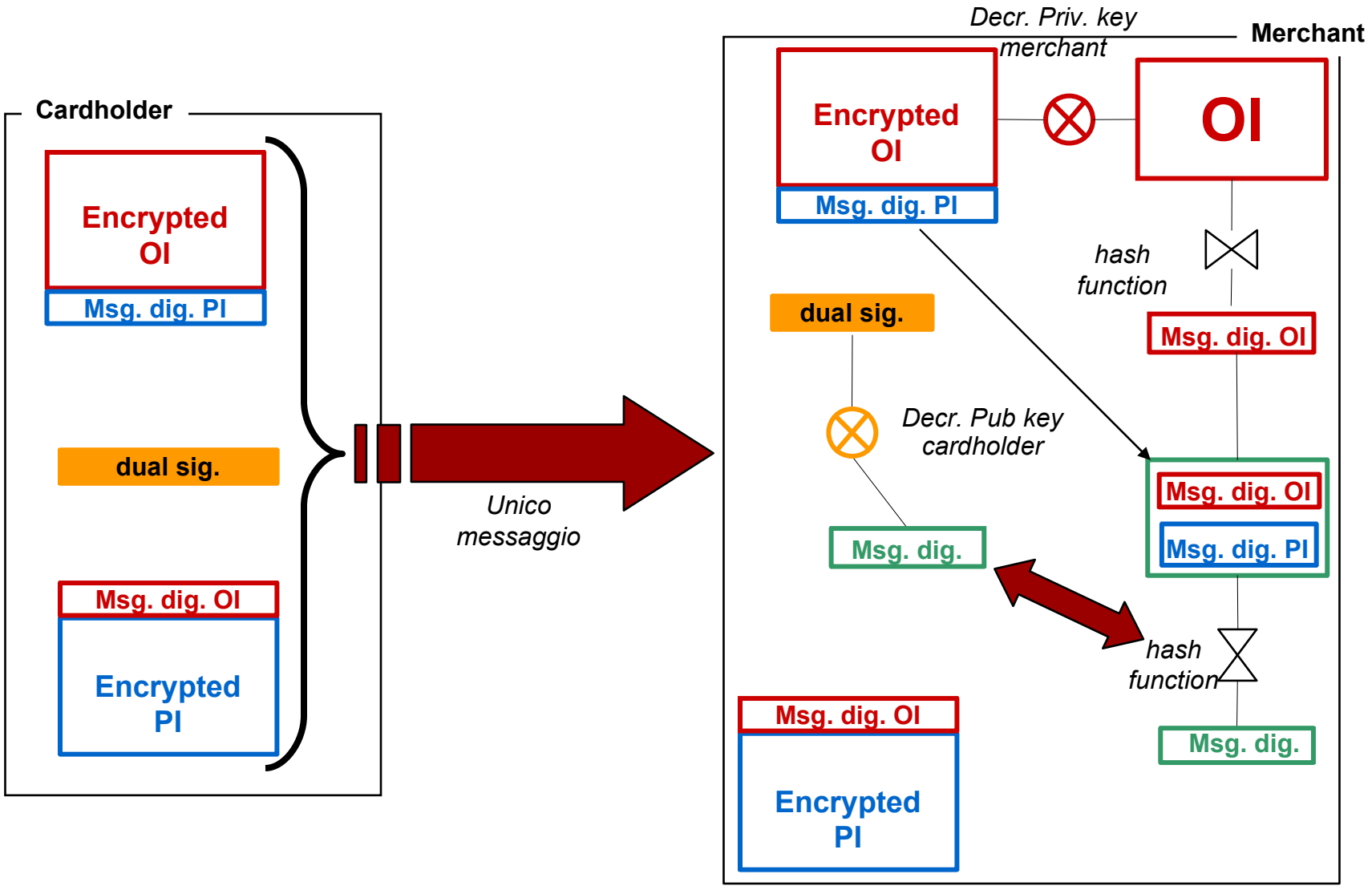
Verification



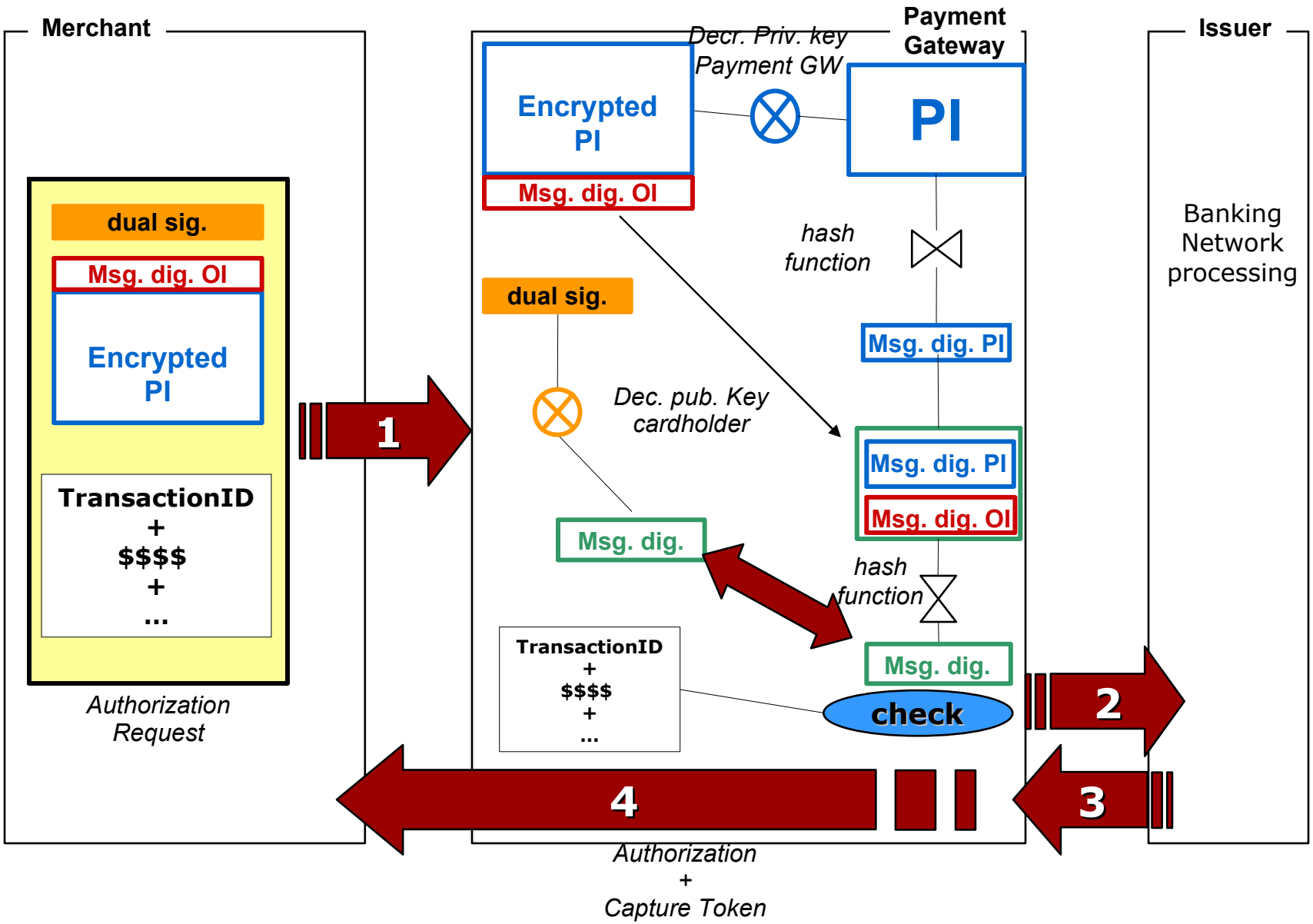
Request preparation



Merchant verification



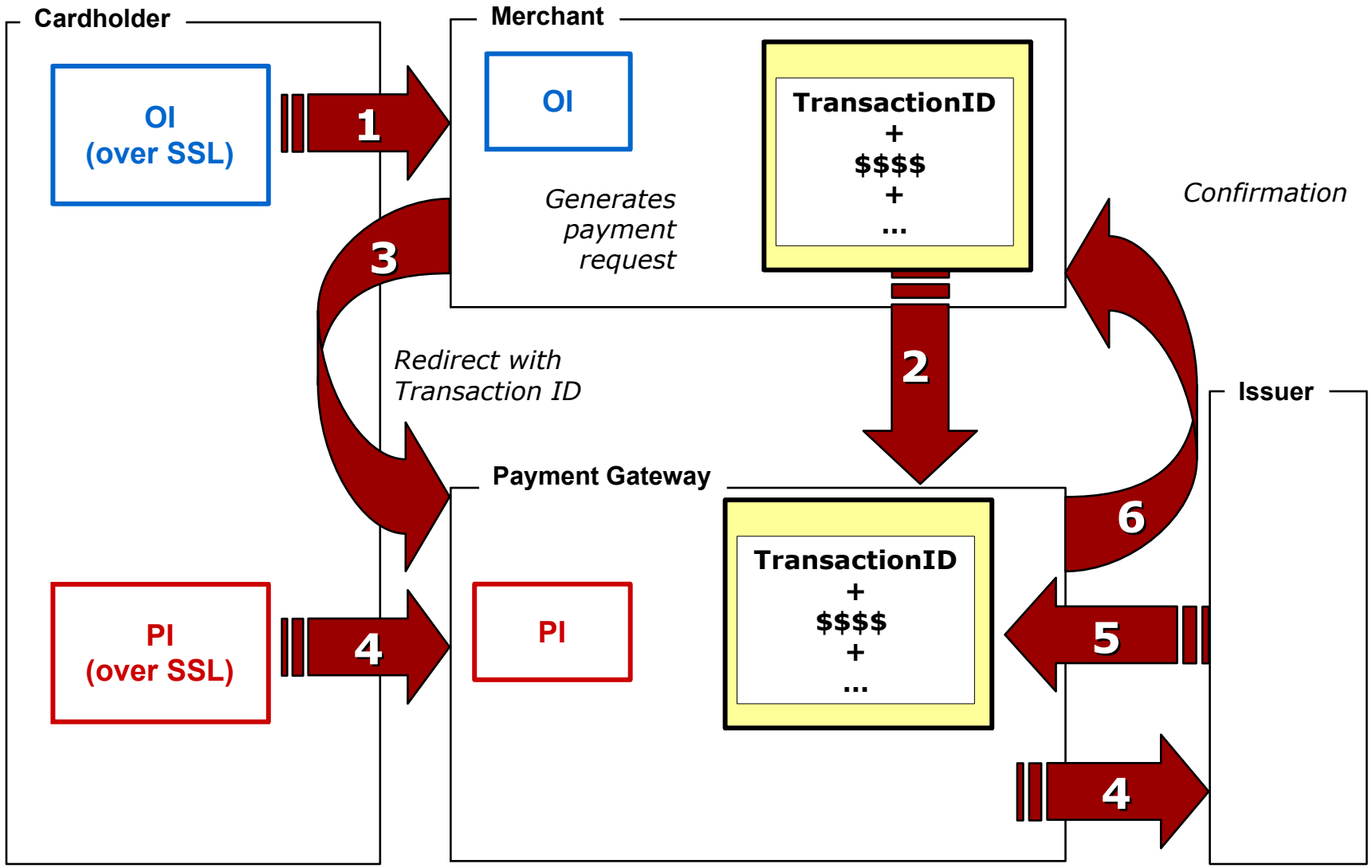
Payment authorization



SET limitations

- SET requests that besides Merchant and Payment Gateway, also the Cardholder has a digital certificate
- Therefore, a pre-registration of the cardholder is needed!
- Non-transparent = less critical mass = failure
- Notwithstanding its elegance and the fact it was pushed by VISA+MasterCard!
- Commonly, a “simplified” version of SET is used in e-commerce as an architectural solution

A “fake” SET


Cardholder
Merchant
OI
(over SSL)
1
OI
TransactionID
+
$$$$
+
...
Generates
payment
request
Confirmation
3
Redirect with
Transaction ID
2
Issuer
Payment Gateway
6
TransactionID
+
$$$$
+
...
PI
(over SSL)
4
PI
5
4