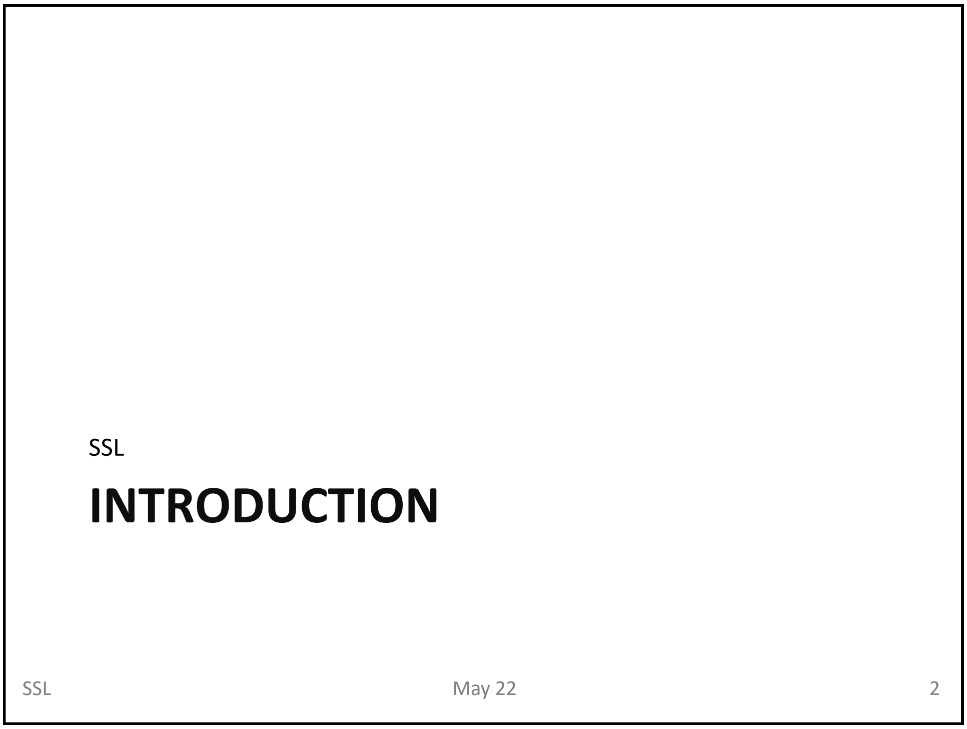
Secure Socket Layer (SSL)

Gianluca Dini
Dept. of Ingegneria dell'Informazione
University of Pisa
Email: gianluca.dini@unipi.it

Version: 2021-05-10

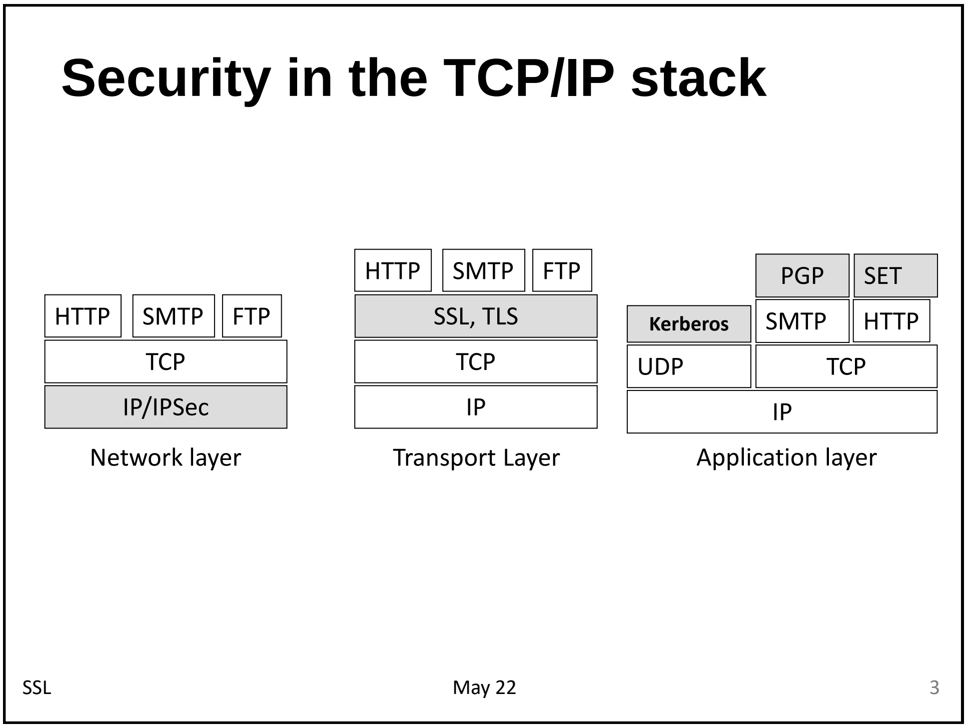
1



SSL

INTRODUCTION

2



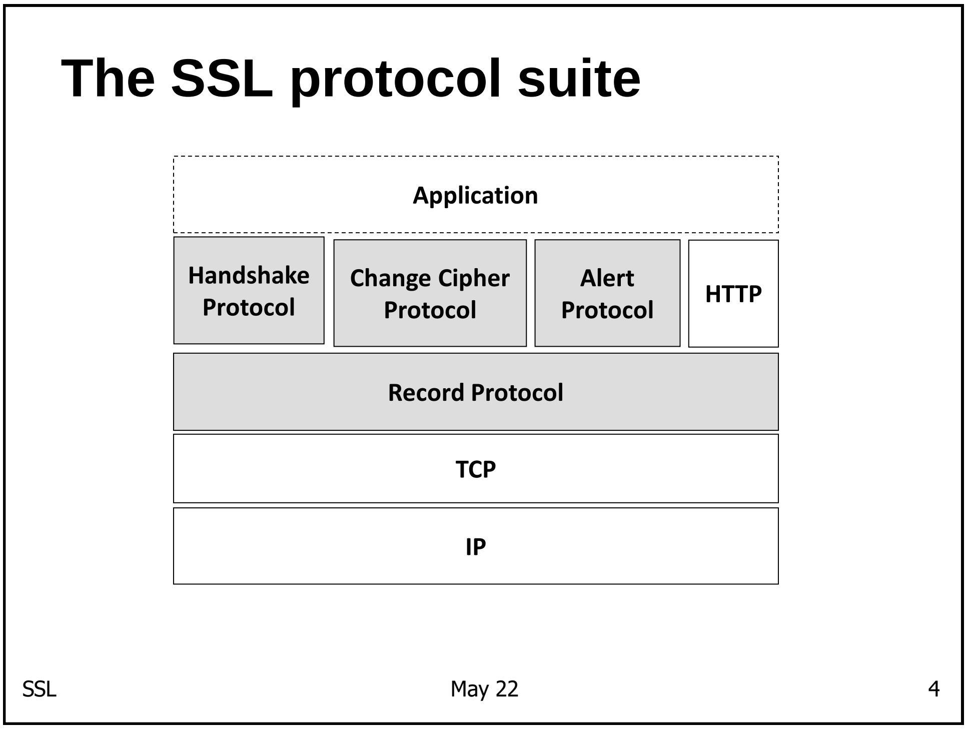
Security in the TCP/IP stack

HTTP | SMTP | FTP
TCP
IP/IPSec
Network layer

HTTP | SMTP | FTP
SSL, TLS
TCP
IP
Transport Layer

PGP | SET
Kerberos | SMTP | HTTP
UDP | TCP
IP
Application layer

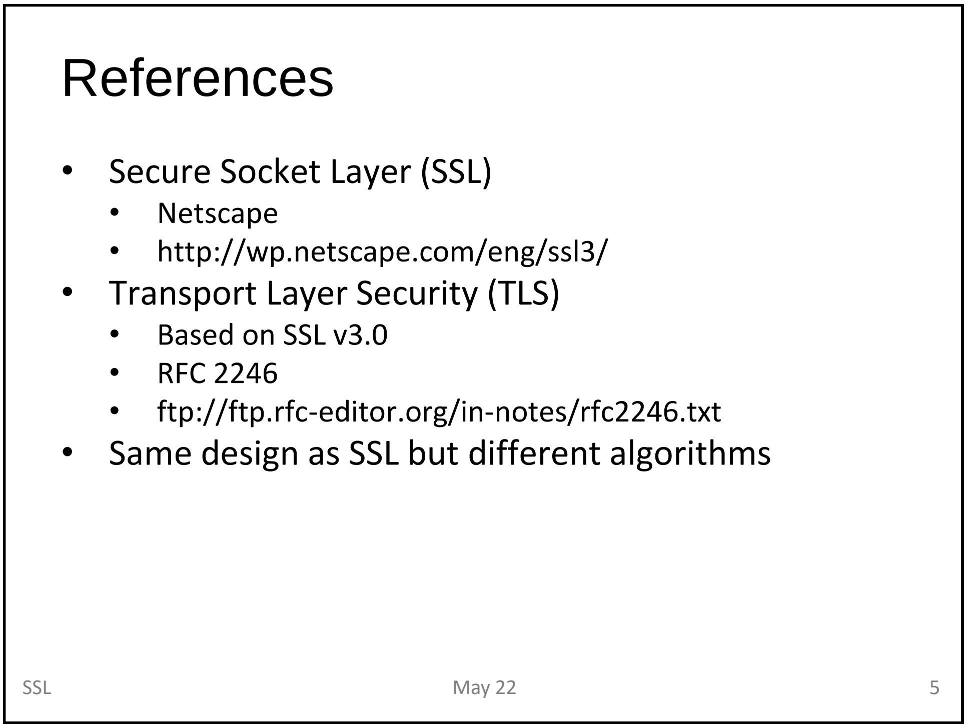
3



The SSL protocol suite

Application
Handshake Protocol | Change Cipher Protocol | Alert Protocol | HTTP
Record Protocol
TCP
IP

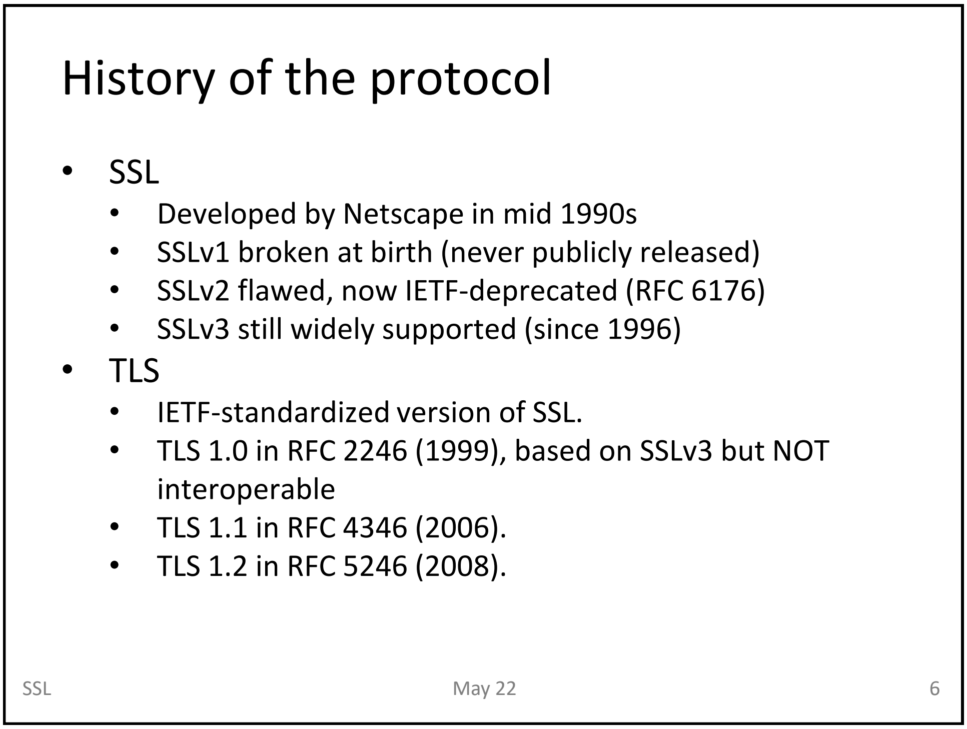
4



References

- Secure Socket Layer (SSL)
  - Netscape
  - http://wp.netscape.com/eng/ssl3/
- Transport Layer Security (TLS)
  - Based on SSL v3.0
  - RFC 2246
  - ftp://ftp.rfc-editor.org/in-notes/rfc2246.txt
- Same design as SSL but different algorithms

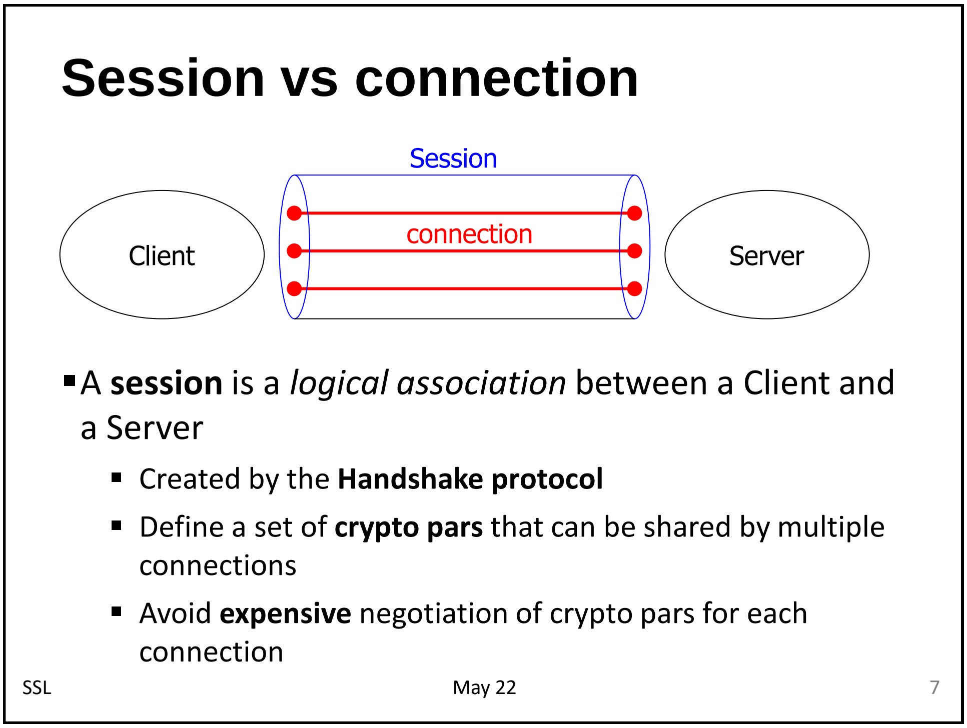
5



History of the protocol

- SSL
  - Developed by Netscape in mid 1990s
  - SSLv1 broken at birth (never publicly released)
  - SSLv2 flawed, now IETF-deprecated (RFC 6176)
  - SSLv3 still widely supported (since 1996)
- TLS
  - IETF-standardized version of SSL.
  - TLS 1.0 in RFC 2246 (1999), based on SSLv3 but NOT interoperable
  - TLS 1.1 in RFC 4346 (2006).
  - TLS 1.2 in RFC 5246 (2008).

6

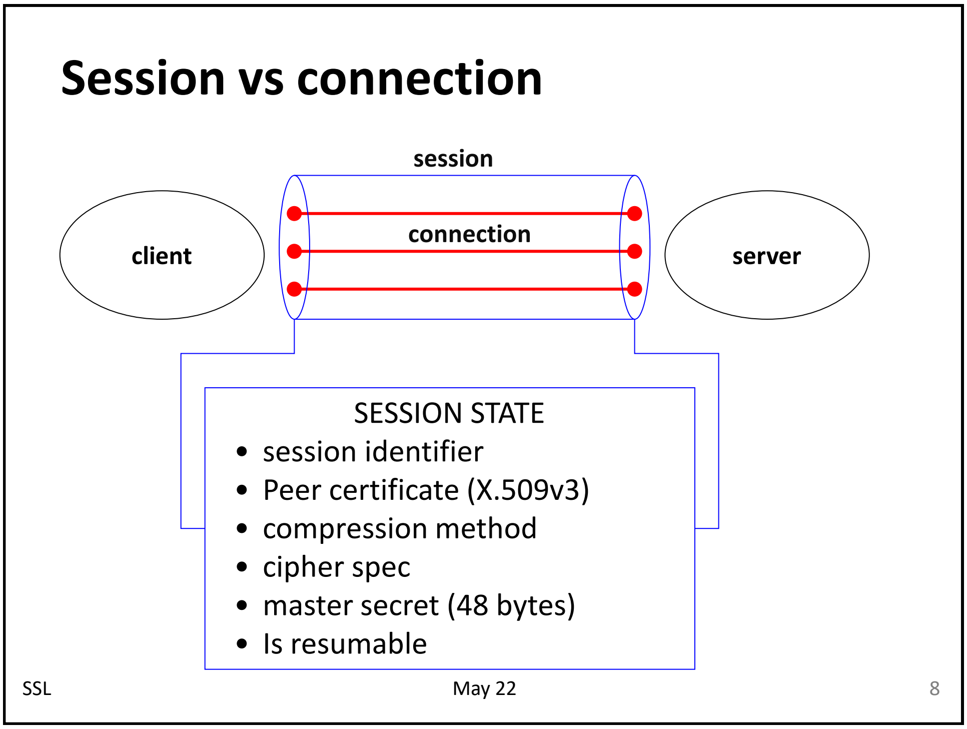


Session vs connection

Session
Client
connection
Server

- A **session** is a *logical association* between a Client and a Server
  - Created by the **Handshake protocol**
  - Define a set of **crypto pars** that can be shared by multiple connections
  - Avoid **expensive** negotiation of crypto pars for each connection

7



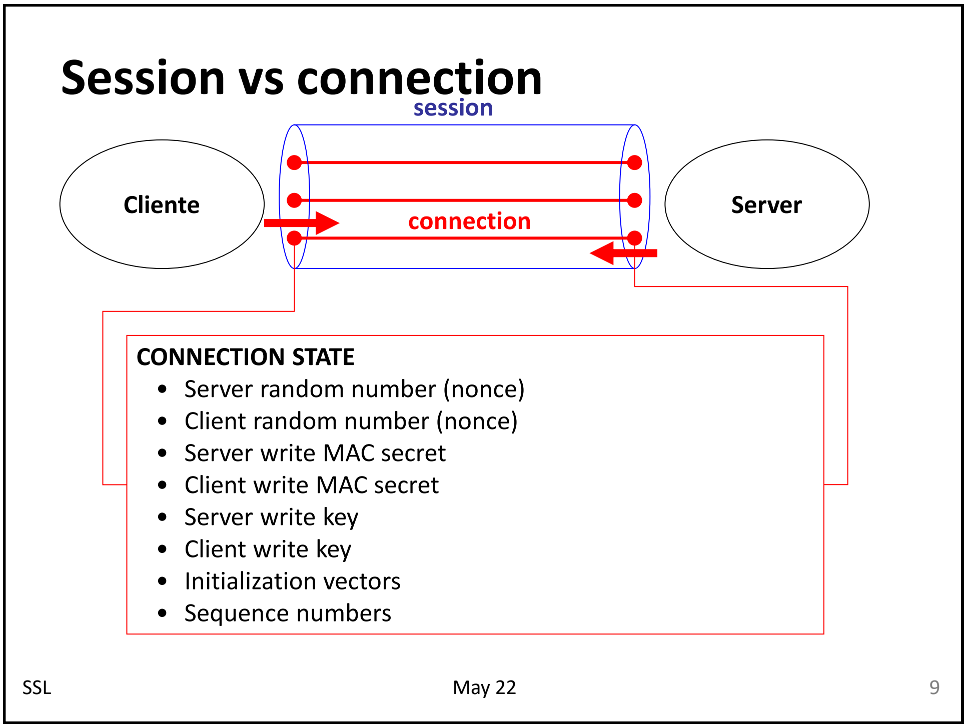
Session vs connection

session
client
connection
server

SESSION STATE

- session identifier
- Peer certificate (X.509v3)
- compression method
- cipher spec
- master secret (48 bytes)
- Is resumable

8

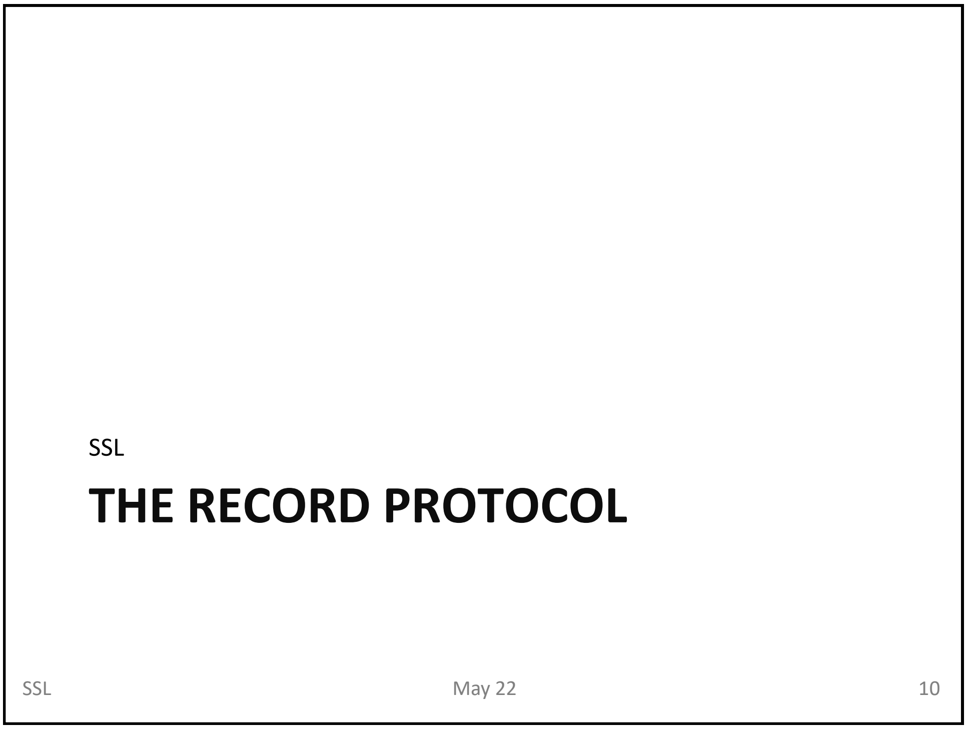


Session vs connection

CONNECTION STATE

- Server random number (nonce)
- Client random number (nonce)
- Server write MAC secret
- Client write MAC secret
- Server write key
- Client write key
- Initialization vectors
- Sequence numbers

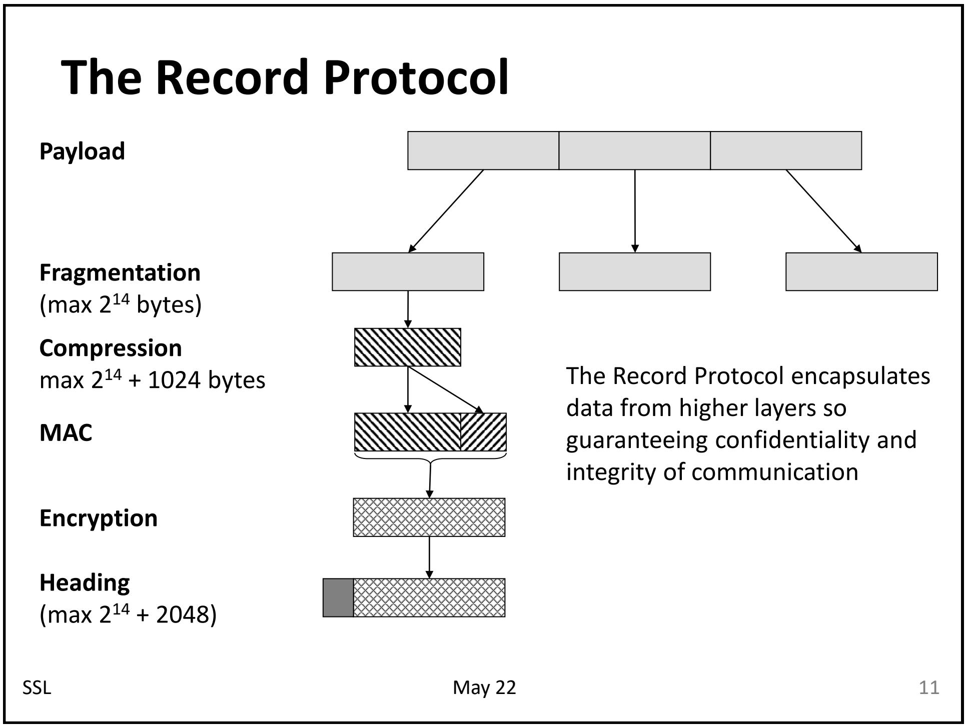
9



SSL

THE RECORD PROTOCOL

10



The Record Protocol

Payload

Fragmentation
(max 2^{14} bytes)

Compression
max 2^{14} + 1024 bytes

MAC

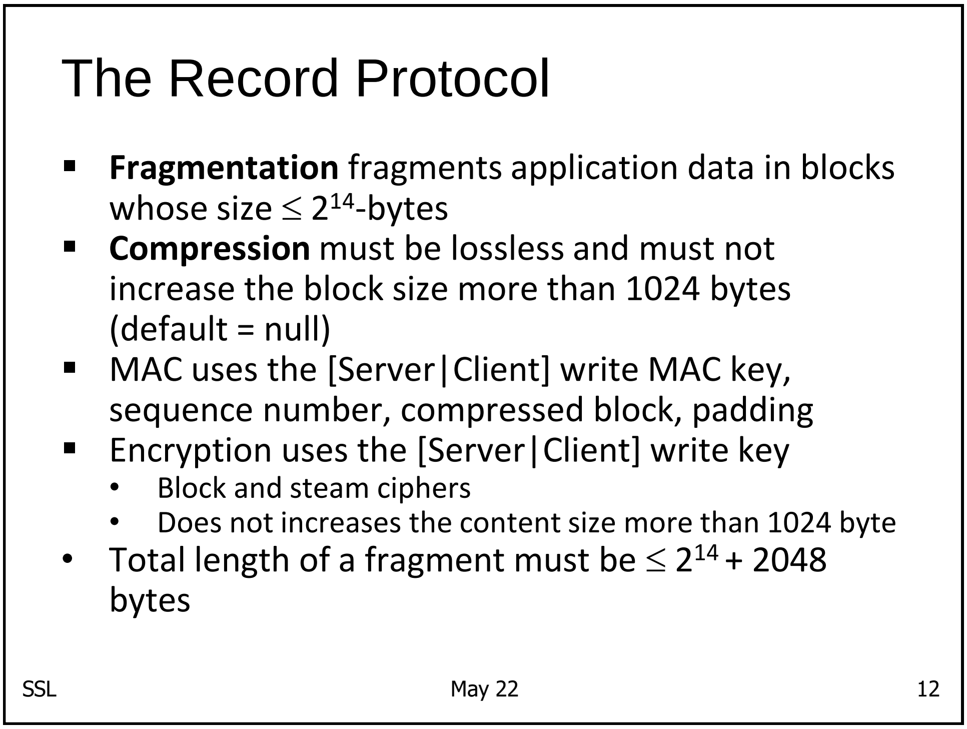
Encryption

Heading
(max 2^{14} + 2048)

The Record Protocol encapsulates data from higher layers so guaranteeing confidentiality and integrity of communication

SSL May 22 11

11

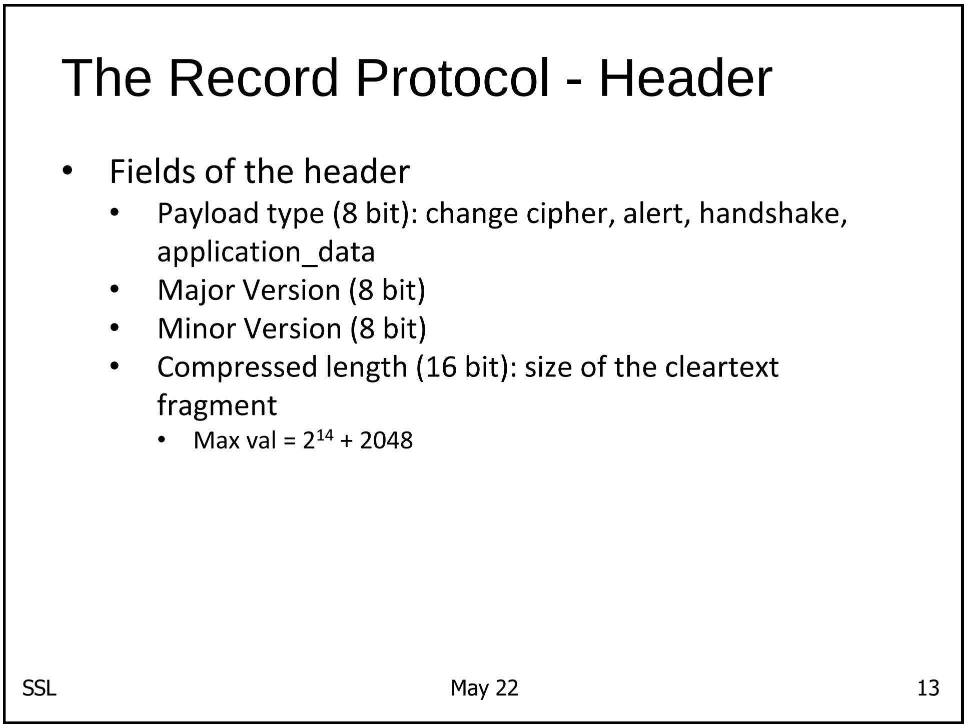


The Record Protocol

- **Fragmentation** fragments application data in blocks whose size ≤ 2^{14}-bytes
- **Compression** must be lossless and must not increase the block size more than 1024 bytes (default = null)
- MAC uses the [Server|Client] write MAC key, sequence number, compressed block, padding
- Encryption uses the [Server|Client] write key
  - Block and steam ciphers
  - Does not increases the content size more than 1024 byte
- Total length of a fragment must be ≤ 2^{14} + 2048 bytes

SSL May 22 12

12

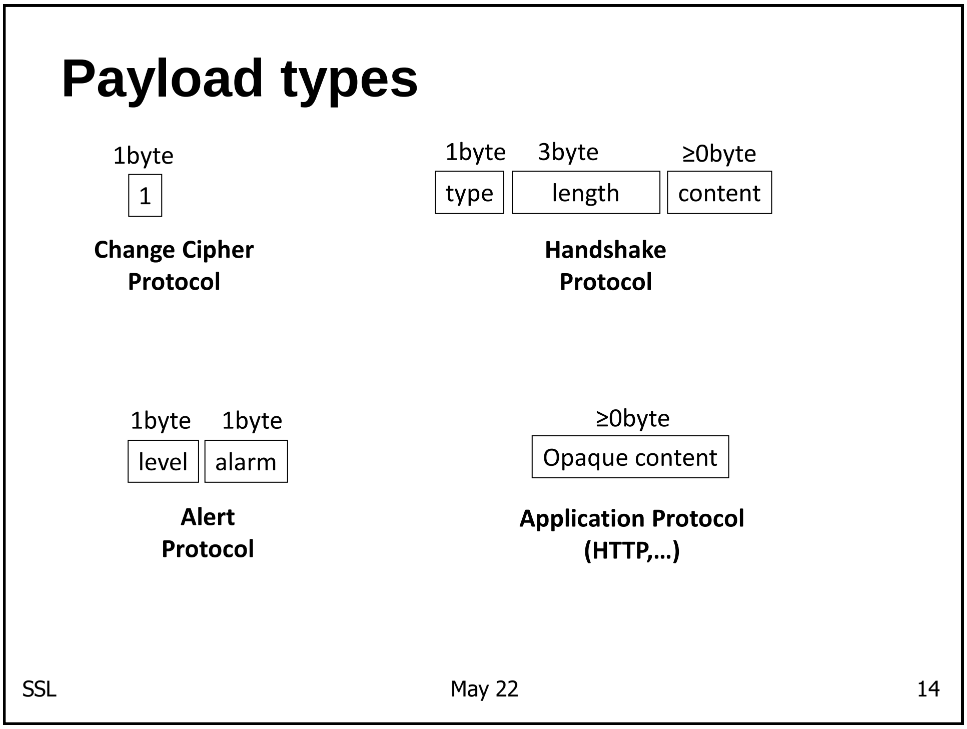


The Record Protocol - Header

- Fields of the header
  - Payload type (8 bit): change cipher, alert, handshake, application_data
  - Major Version (8 bit)
  - Minor Version (8 bit)
  - Compressed length (16 bit): size of the cleartext fragment
    - Max val = 2^{14} + 2048

SSL May 22 13

13



Payload types

1byte
1

Change Cipher Protocol

1byte	3byte	≥0byte
type	length	content

Handshake Protocol

1byte	1byte
level	alarm

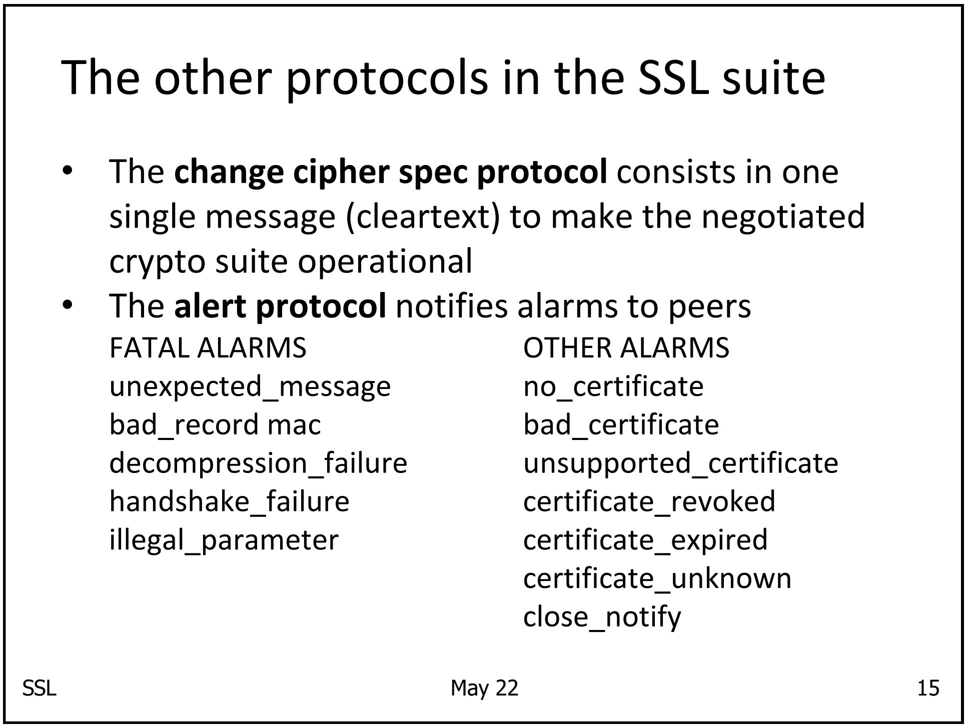
Alert Protocol

≥0byte
Opaque content

Application Protocol (HTTP,…)

SSL May 22 14

14



The other protocols in the SSL suite

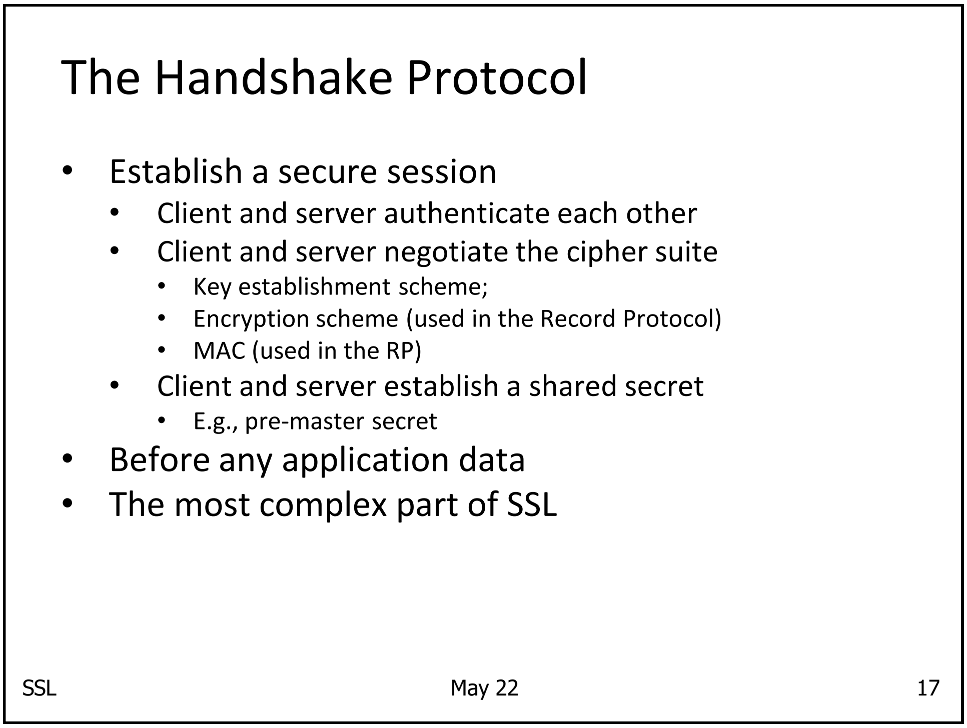
- The **change cipher spec protocol** consists in one single message (cleartext) to make the negotiated crypto suite operational
- The **alert protocol** notifies alarms to peers

  FATAL ALARMS
  unexpected_message
  bad_record mac
  decompression_failure
  handshake_failure
  illegal_parameter

  OTHER ALARMS
  no_certificate
  bad_certificate
  unsupported_certificate
  certificate_revoked
  certificate_expired
  certificate_unknown
  close_notify

SSL

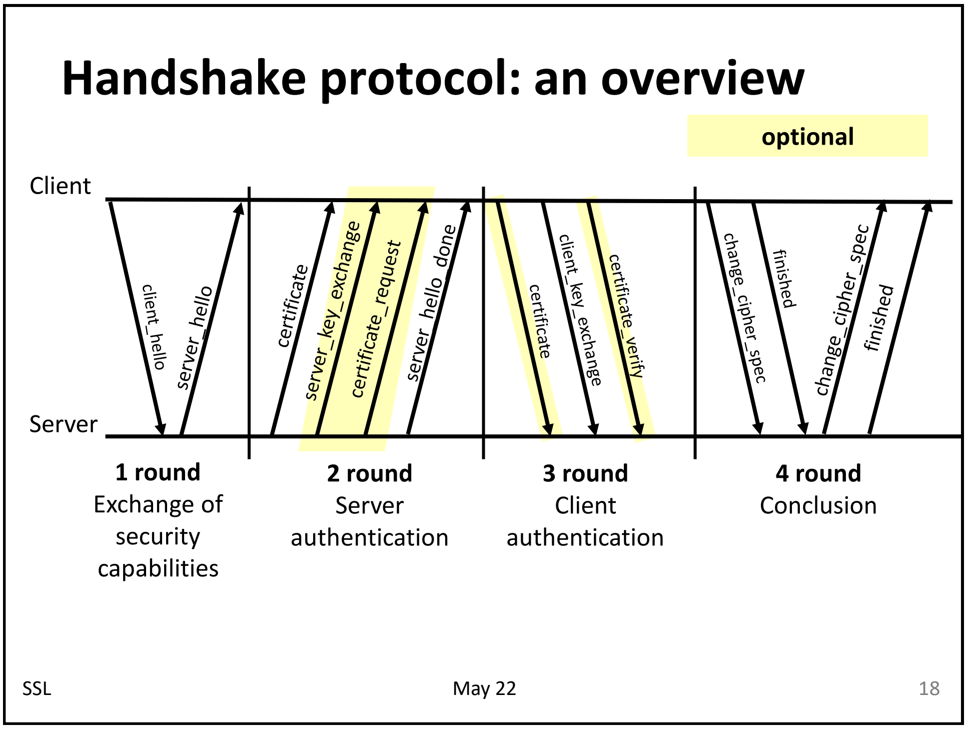
THE HANDSHAKE PROTOCOL



The Handshake Protocol

- Establish a secure session
  - Client and server authenticate each other
  - Client and server negotiate the cipher suite
    - Key establishment scheme;
    - Encryption scheme (used in the Record Protocol)
    - MAC (used in the RP)
  - Client and server establish a shared secret
    - E.g., pre-master secret
- Before any application data
- The most complex part of SSL

17



Handshake protocol: an overview

optional

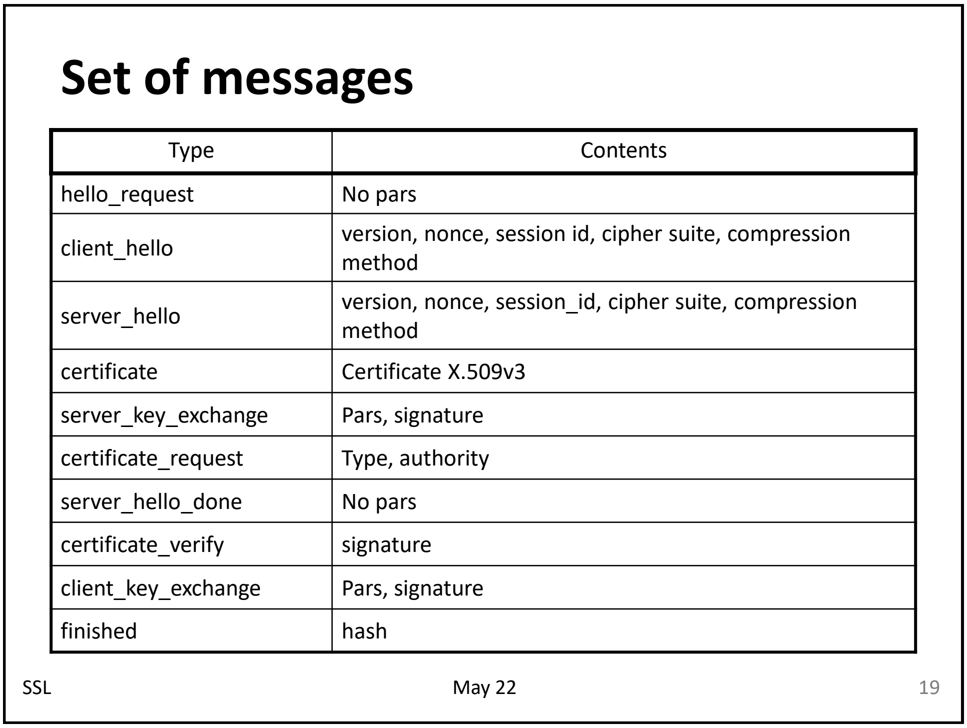
1 round Exchange of security capabilities

2 round Server authentication

3 round Client authentication

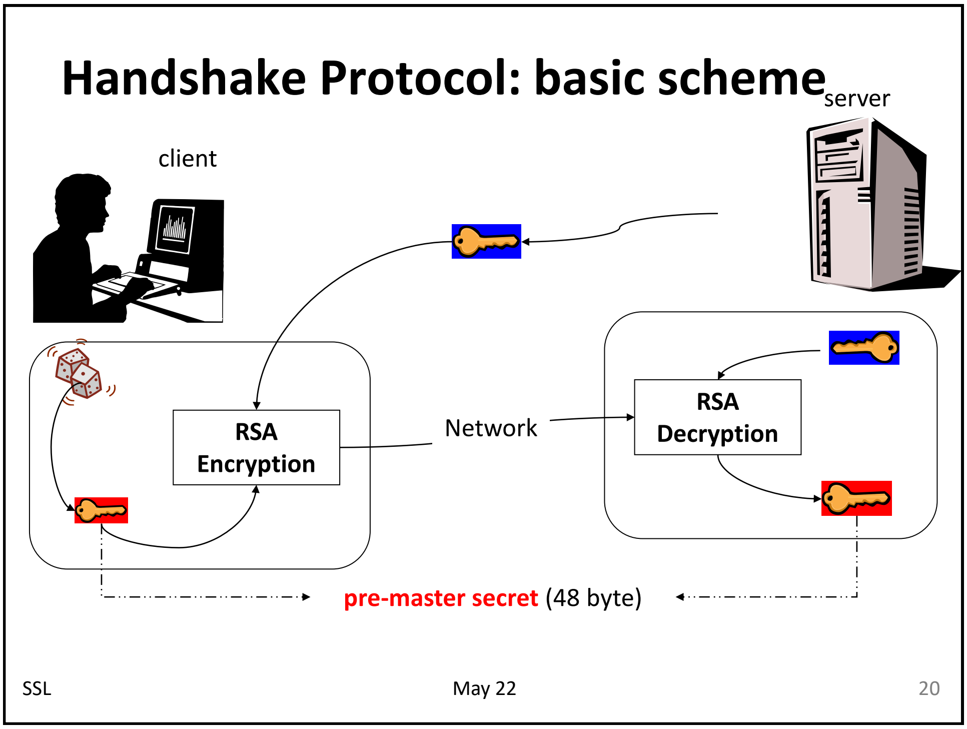
4 round Conclusion

18



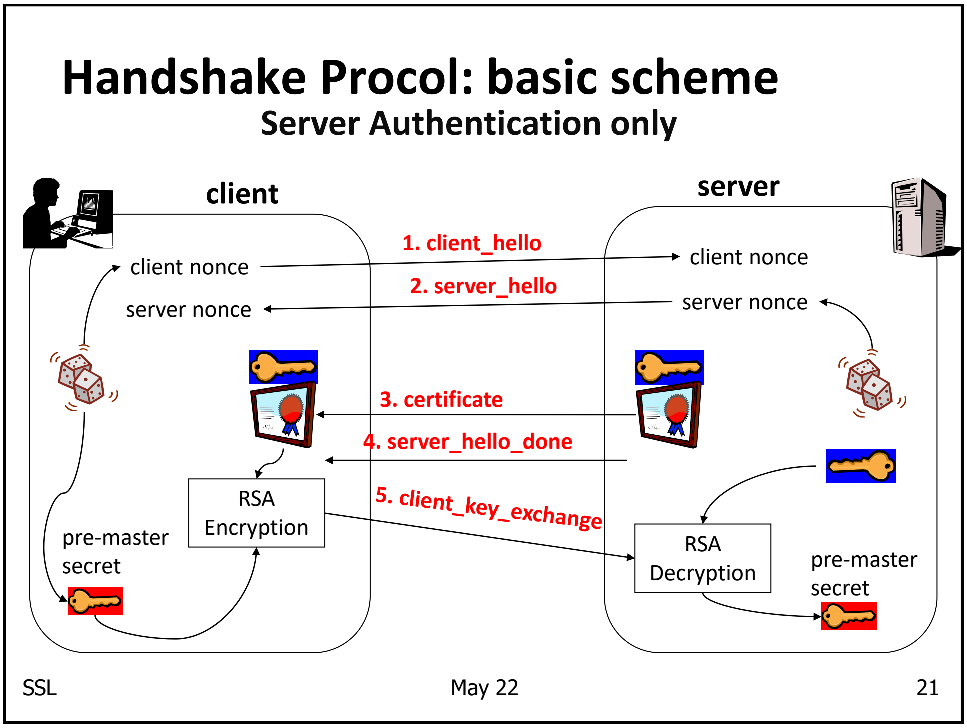
Set of messages

Type	Contents
hello_request	No pars
client_hello	version, nonce, session id, cipher suite, compression method
server_hello	version, nonce, session_id, cipher suite, compression method
certificate	Certificate X.509v3
server_key_exchange	Pars, signature
certificate_request	Type, authority
server_hello_done	No pars
certificate_verify	signature
client_key_exchange	Pars, signature
finished	hash



Handshake Protocol: basic scheme

pre-master secret (48 byte)



Handshake Procol: basic scheme

Server Authentication only

client

server

client nonce

server nonce

1. client_hello

2. server_hello

3. certificate

4. server_hello_done

5. client_key_exchange

RSA Encryption

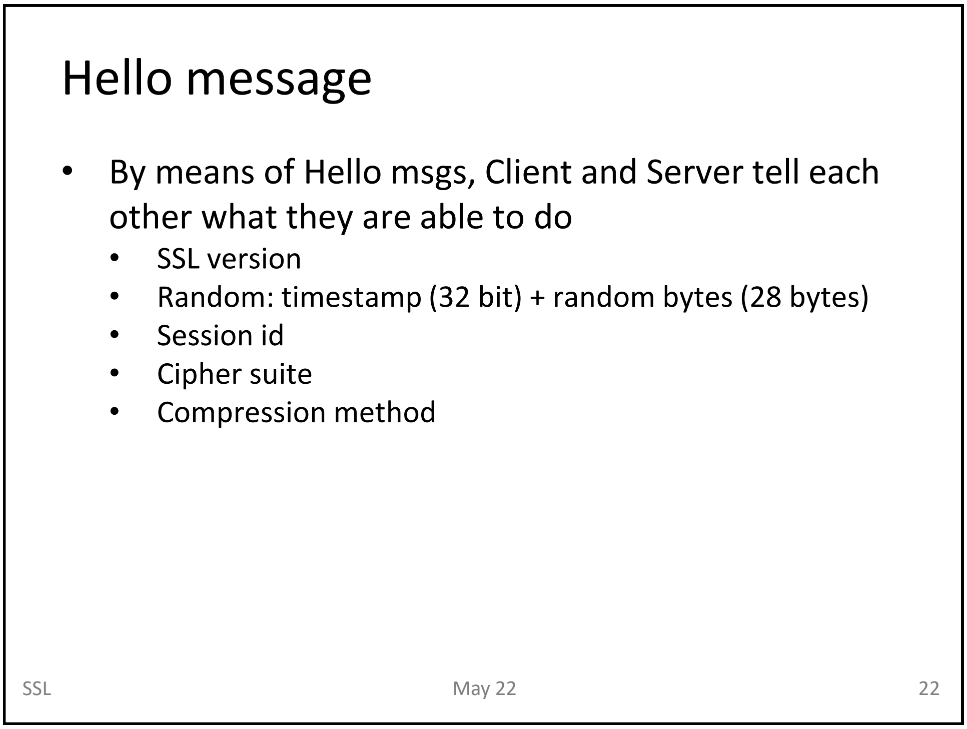
RSA Decryption

pre-master secret

pre-master secret

SSL May 22 21

21

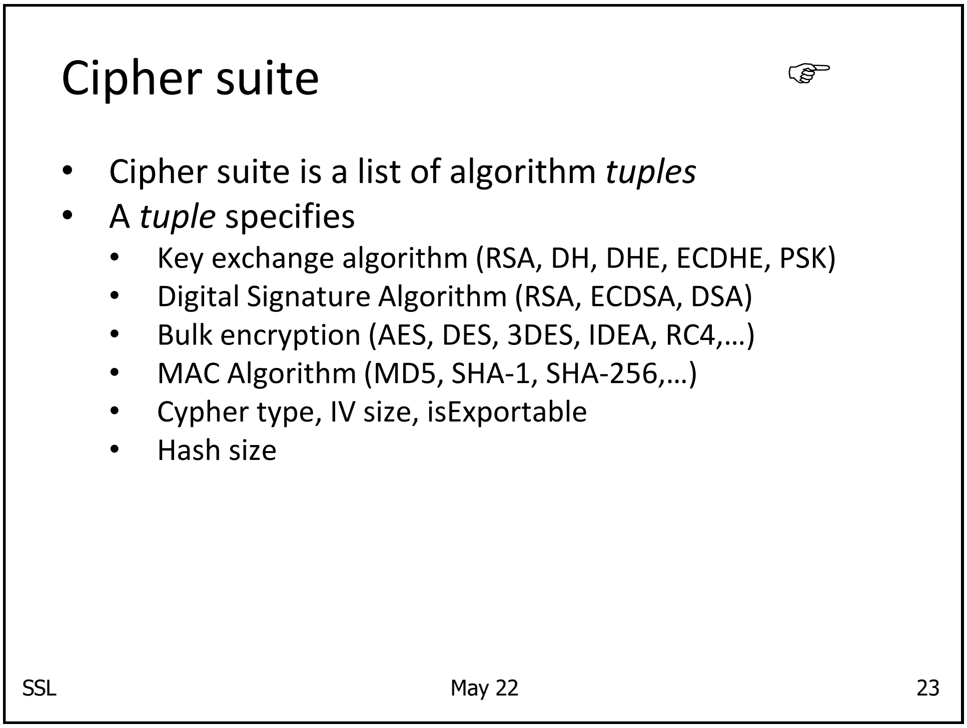


Hello message

- By means of Hello msgs, Client and Server tell each other what they are able to do
  - SSL version
  - Random: timestamp (32 bit) + random bytes (28 bytes)
  - Session id
  - Cipher suite
  - Compression method

SSL May 22 22

22

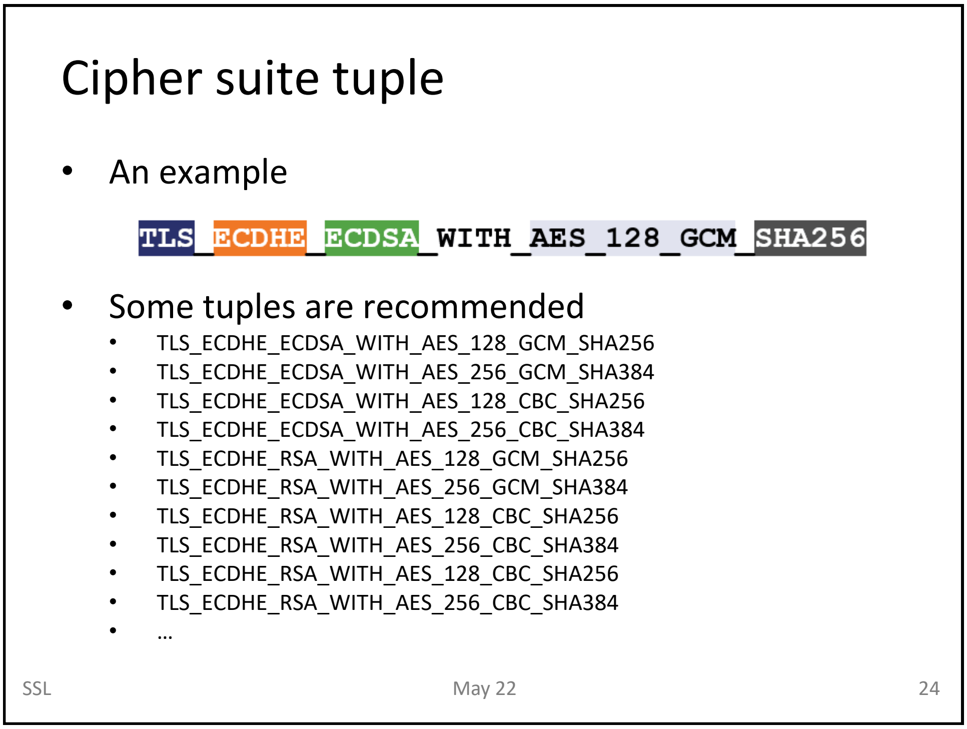


Cipher suite ☞

- Cipher suite is a list of algorithm *tuples*
- A *tuple* specifies
  - Key exchange algorithm (RSA, DH, DHE, ECDHE, PSK)
  - Digital Signature Algorithm (RSA, ECDSA, DSA)
  - Bulk encryption (AES, DES, 3DES, IDEA, RC4,…)
  - MAC Algorithm (MD5, SHA-1, SHA-256,…)
  - Cypher type, IV size, isExportable
  - Hash size

SSL May 22 23

23



Cipher suite tuple

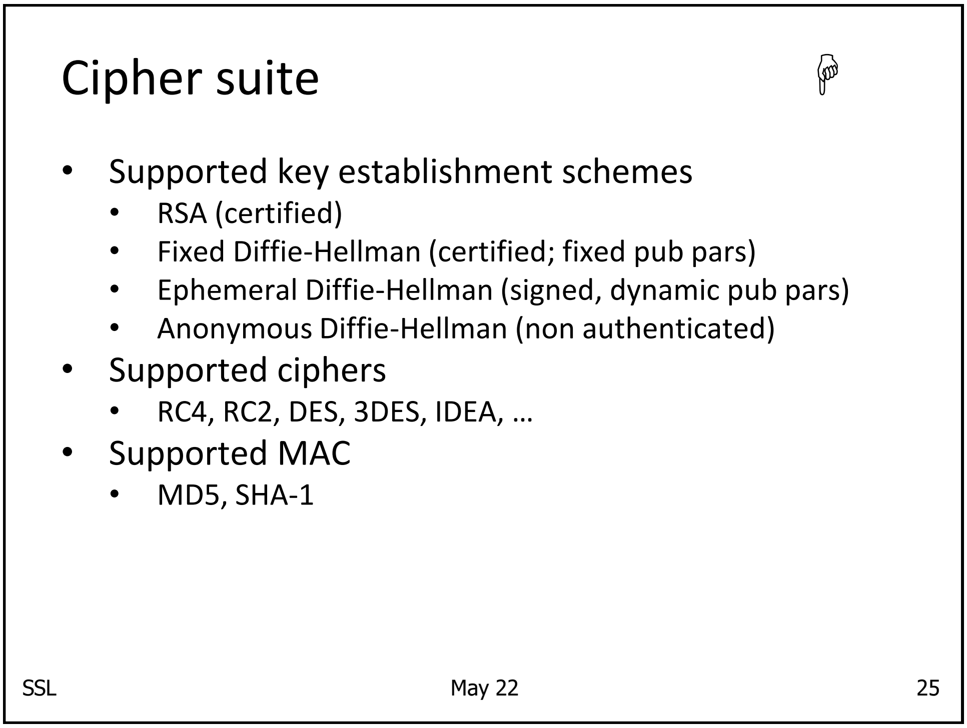
- An example

  TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

- Some tuples are recommended
  - TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
  - TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
  - TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
  - …

SSL May 22 24

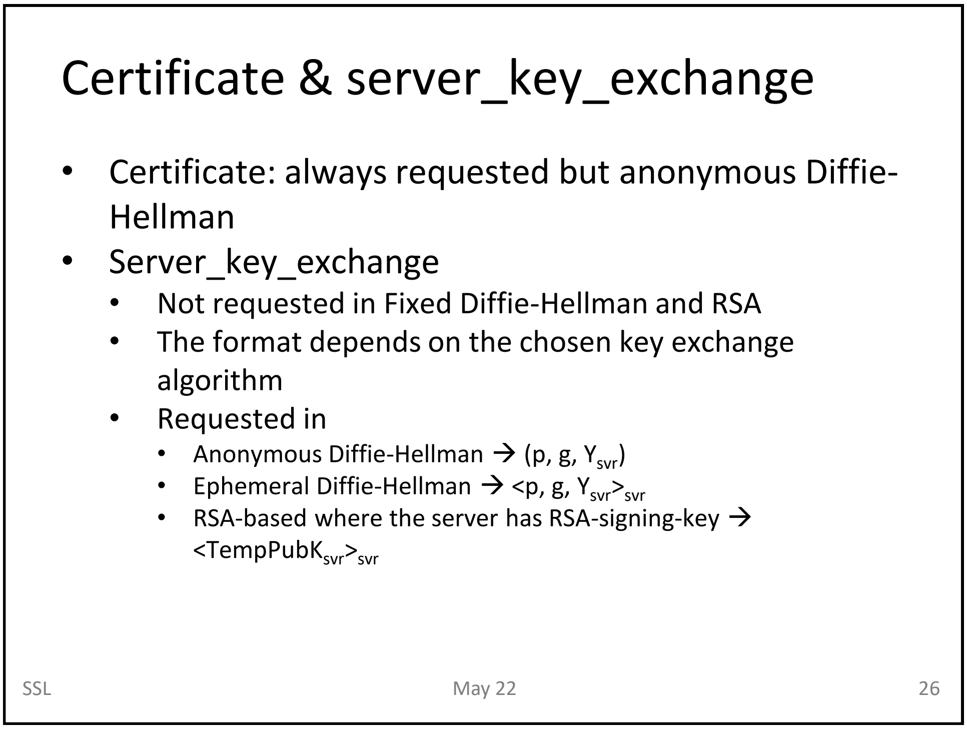
24



Cipher suite

- Supported key establishment schemes
  - RSA (certified)
  - Fixed Diffie-Hellman (certified; fixed pub pars)
  - Ephemeral Diffie-Hellman (signed, dynamic pub pars)
  - Anonymous Diffie-Hellman (non authenticated)
- Supported ciphers
  - RC4, RC2, DES, 3DES, IDEA, …
- Supported MAC
  - MD5, SHA-1

25



Certificate & server_key_exchange

- Certificate: always requested but anonymous Diffie-Hellman
- Server_key_exchange
  - Not requested in Fixed Diffie-Hellman and RSA
  - The format depends on the chosen key exchange algorithm
  - Requested in
    - Anonymous Diffie-Hellman → (p, g, Y_{svr})
    - Ephemeral Diffie-Hellman → $<p, g, Y_{svr}>_{svr}$
    - RSA-based where the server has RSA-signing-key → $<TempPubK_{svr}>_{svr}$

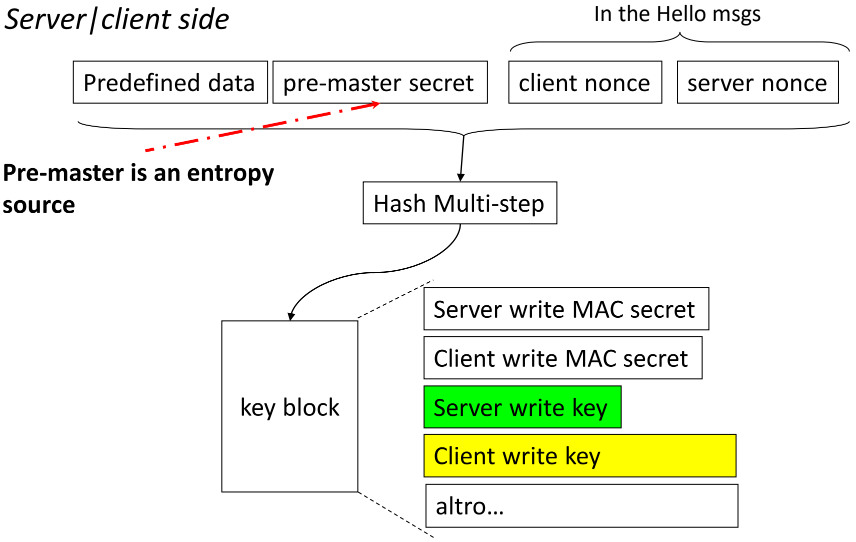
26

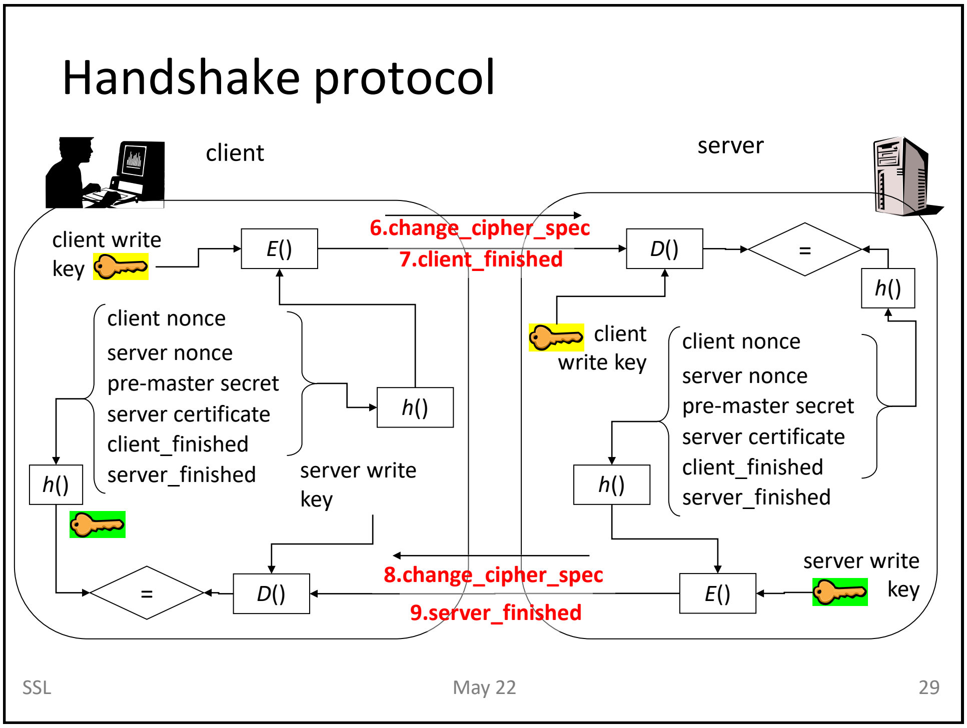
Client_key_exchange message

- The message format depends on the chosen key establishment
  - RSA: pre-master secret
  - ANONYMOUS OR EPHEMERAL DH: (p, g, Y)$_{clnt}$
  - FIXED DH: void payload, public pars will be sent in a certificate message (client → server)

Key generation

Server|client side





Handshake protocol

client

server

client write key

E()

6.change_cipher_spec

7.client_finished

D()

=

h()

client nonce
server nonce
pre-master secret
server certificate
client_finished
server_finished

h()

client write key

client nonce
server nonce
pre-master secret
server certificate
client_finished
server_finished

h()

server write key

h()

D()

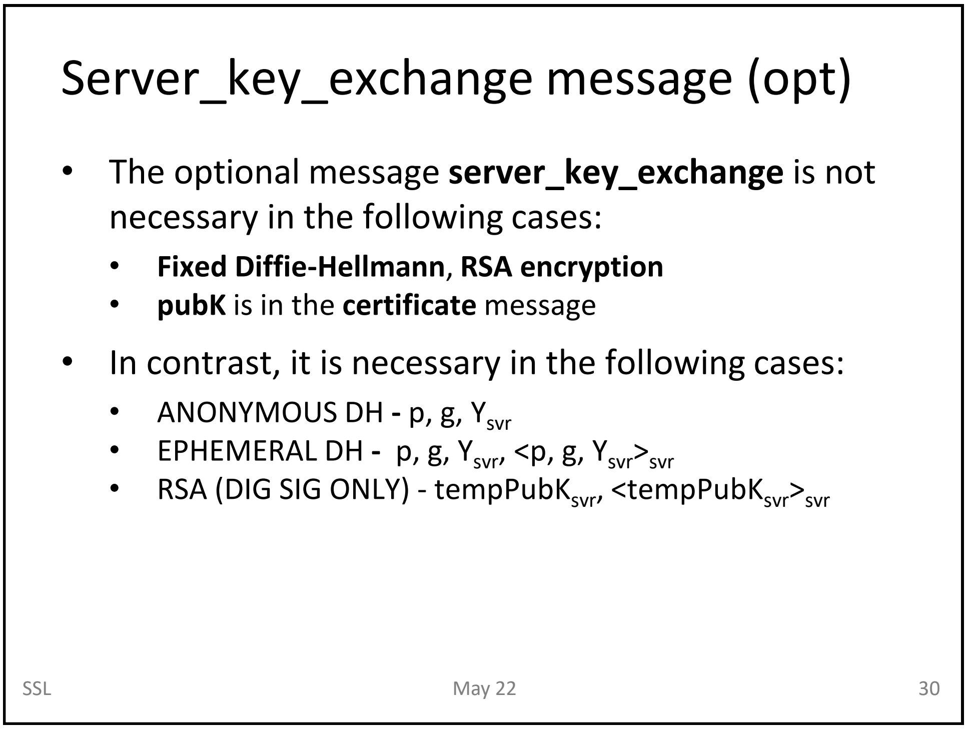
=

8.change_cipher_spec

9.server_finished

E()

server write key



Server_key_exchange message (opt)

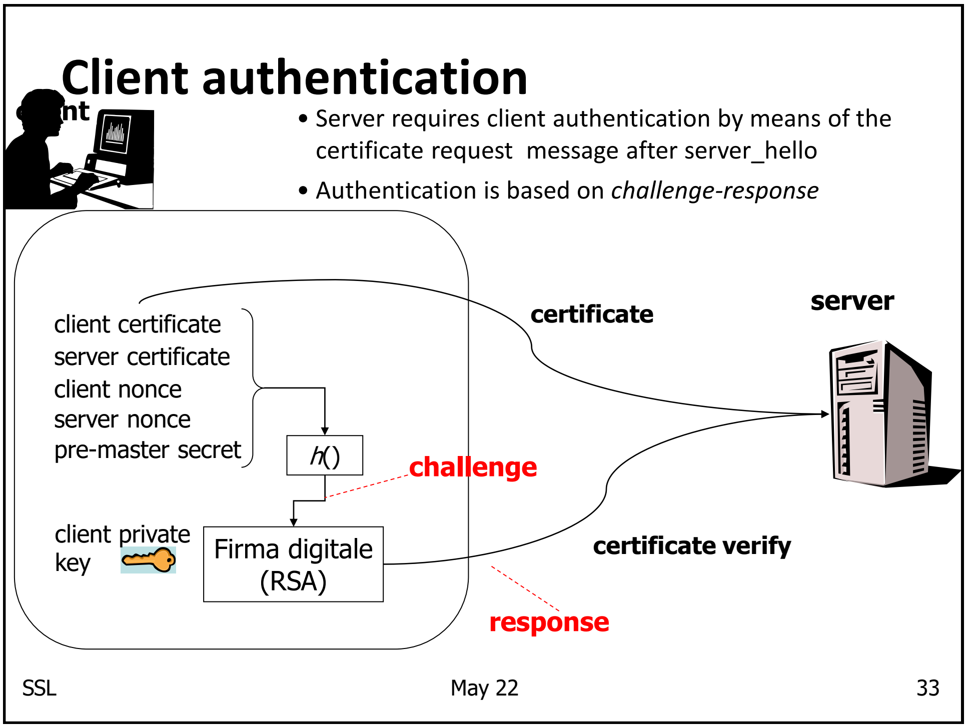
- The optional message **server_key_exchange** is not necessary in the following cases:
  - **Fixed Diffie-Hellmann, RSA encryption**
  - **pubK** is in the **certificate** message
- In contrast, it is necessary in the following cases:
  - ANONYMOUS DH **-** p, g, Y_{svr}
  - EPHEMERAL DH **-** p, g, Y_{svr}, $<p, g, Y_{svr}>_{svr}$
  - RSA (DIG SIG ONLY) - $tempPubK_{svr}$, $<tempPubK_{svr}>_{svr}$

certificate_request message

- Server may issue a **certificate_request** unless anonymous Diffie-Hellmann is used
- The message has two parameters
  - **Certificate_type**: type of digital signature and its use
    - (RSA | DSS) + (only signature | fixed Diffie-Hellmann | Ephemeral DH)
  - **Certificate_authorities:** acceptable certification authorities

Client authentication

- Handshake Protocol authenticates the server by default
- How can the client be authenticated?
  - Typically, the client is authenticated at the application level (password, credit card number, …)
- However, SSL also supports client authentication w.r.t. the server



Client authentication

- Server requires client authentication by means of the certificate request message after server_hello
- Authentication is based on *challenge-response*

33

Certificate & certificate_verify message

- Client sends the a certificate message if the server requested it
  - No_certificate alert if required certificate is not available
- The client sends certificate_verify message to provide explicit proof of signing privK possession

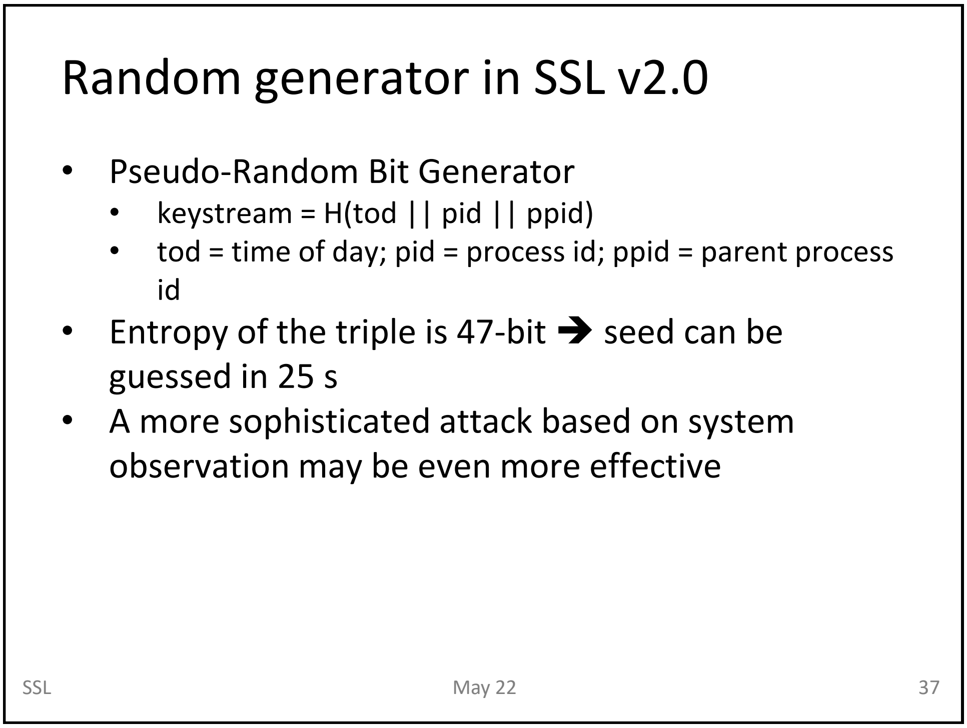
34

Security

- Handshake Protocol
  - Nonces in client hello and server hello
    - Nonces make it possible generate a fresh master secret and avoid replay attacks
  - Certificates
    - Avoid MIM
  - Random quantities
    - Pre-master secret and nonces must be unpredictable
- Record Protocol
  - A block is numbered, authenticated and encrypted
  - Avoid block replay, reordering and substitution
  - Cipher “protects” the MAC

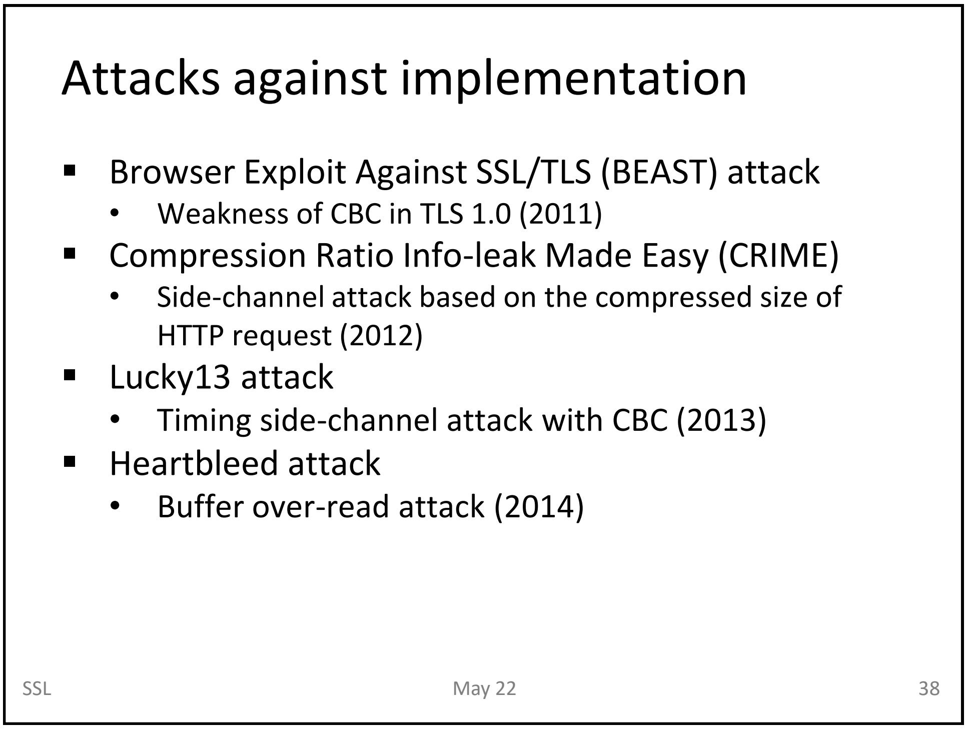
SSL

HISTORY: PITFALLS AND ATTACKS



Random generator in SSL v2.0

- Pseudo-Random Bit Generator
  - keystream = H(tod || pid || ppid)
  - tod = time of day; pid = process id; ppid = parent process id
- Entropy of the triple is 47-bit ➔ seed can be guessed in 25 s
- A more sophisticated attack based on system observation may be even more effective



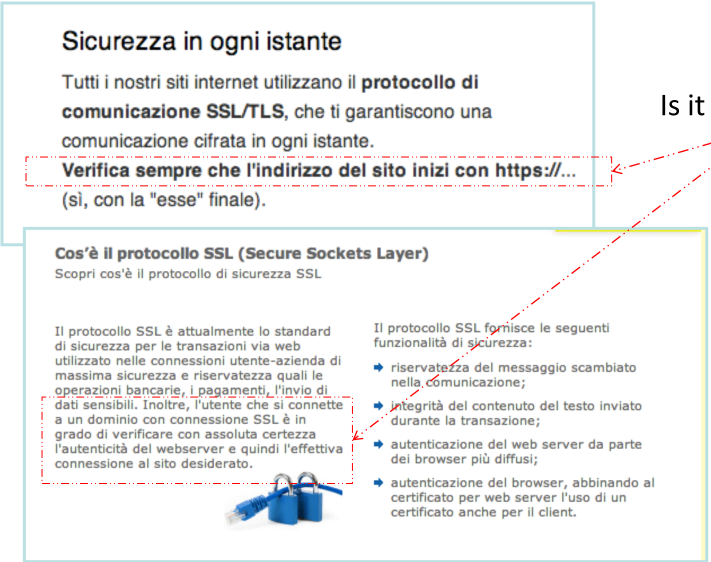
Attacks against implementation

- Browser Exploit Against SSL/TLS (BEAST) attack
  - Weakness of CBC in TLS 1.0 (2011)
- Compression Ratio Info-leak Made Easy (CRIME)
  - Side-channel attack based on the compressed size of HTTP request (2012)
- Lucky13 attack
  - Timing side-channel attack with CBC (2013)
- Heartbleed attack
  - Buffer over-read attack (2014)

SSL

ON USING SSL IN E-COMMERCE

SSL in action



Sicurezza in ogni istante

Tutti i nostri siti internet utilizzano il **protocollo di comunicazione SSL/TLS**, che ti garantiscono una comunicazione cifrata in ogni istante.

Verifica sempre che l'indirizzo del sito inizi con https://... (sì, con la "esse" finale).

Cos'è il protocollo SSL (Secure Sockets Layer)

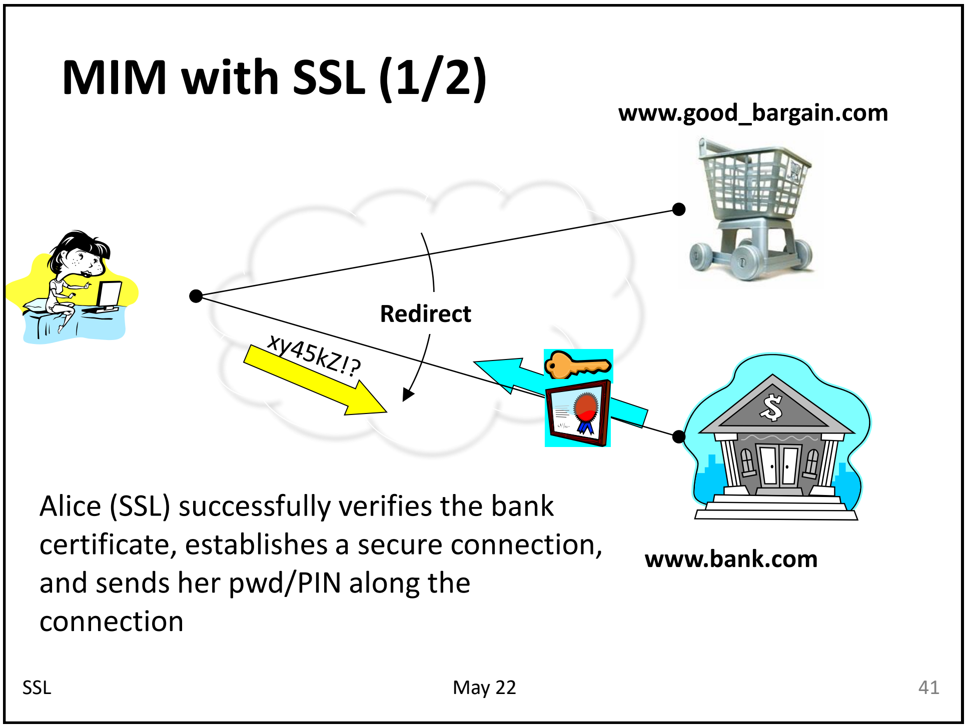
Scopri cos'è il protocollo di sicurezza SSL

Il protocollo SSL è attualmente lo standard di sicurezza per le transazioni via web utilizzato nelle connessioni utente-azienda di massima sicurezza e riservatezza quali le operazioni bancarie, i pagamenti, l'invio di dati sensibili. Inoltre, l'utente che si connette a un dominio con connessione SSL è in grado di verificare con assoluta certezza l'autenticità del webserver e quindi l'effettiva connessione al sito desiderato.

Il protocollo SSL fornisce le seguenti funzionalità di sicurezza:

- riservatezza del messaggio scambiato nella comunicazione;
- integrità del contenuto del testo inviato durante la transazione;
- autenticazione del web server da parte dei browser più diffusi;
- autenticazione del browser, abbinando al certificato per web server l'uso di un certificato anche per il client.

Is it really true?

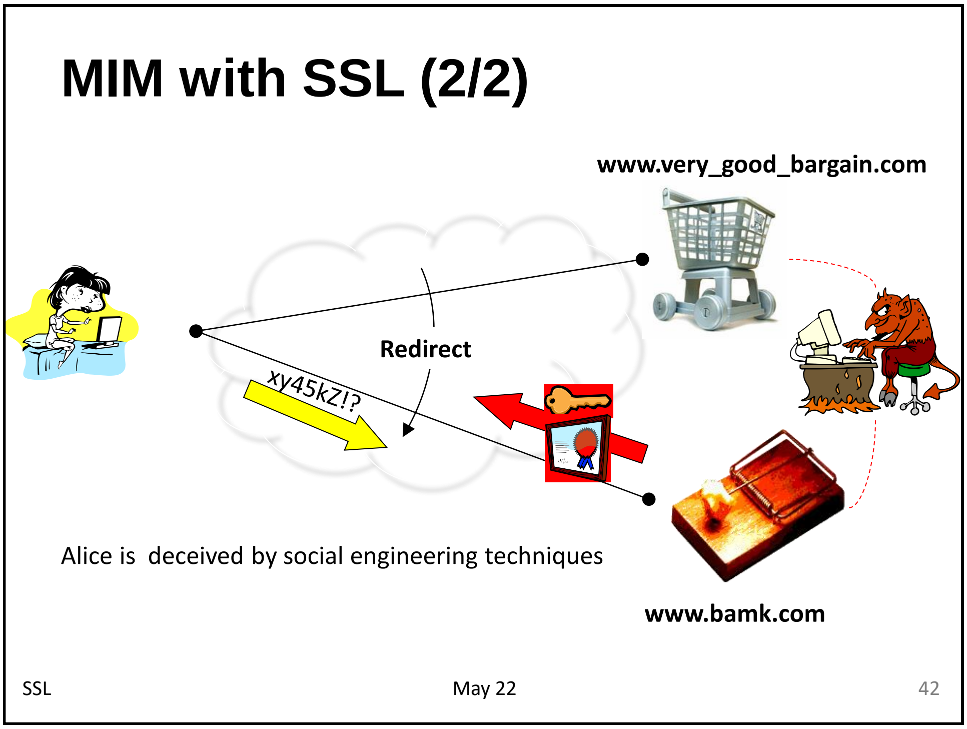


MIM with SSL (1/2)

Alice (SSL) successfully verifies the bank certificate, establishes a secure connection, and sends her pwd/PIN along the connection

SSL May 22 41

41

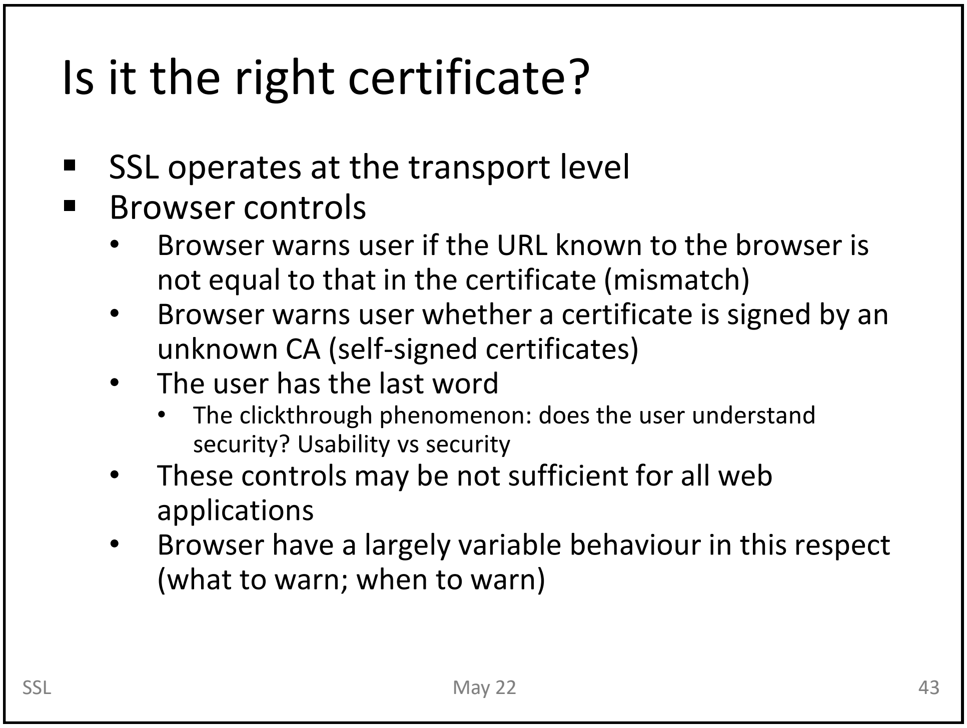


MIM with SSL (2/2)

Alice is deceived by social engineering techniques

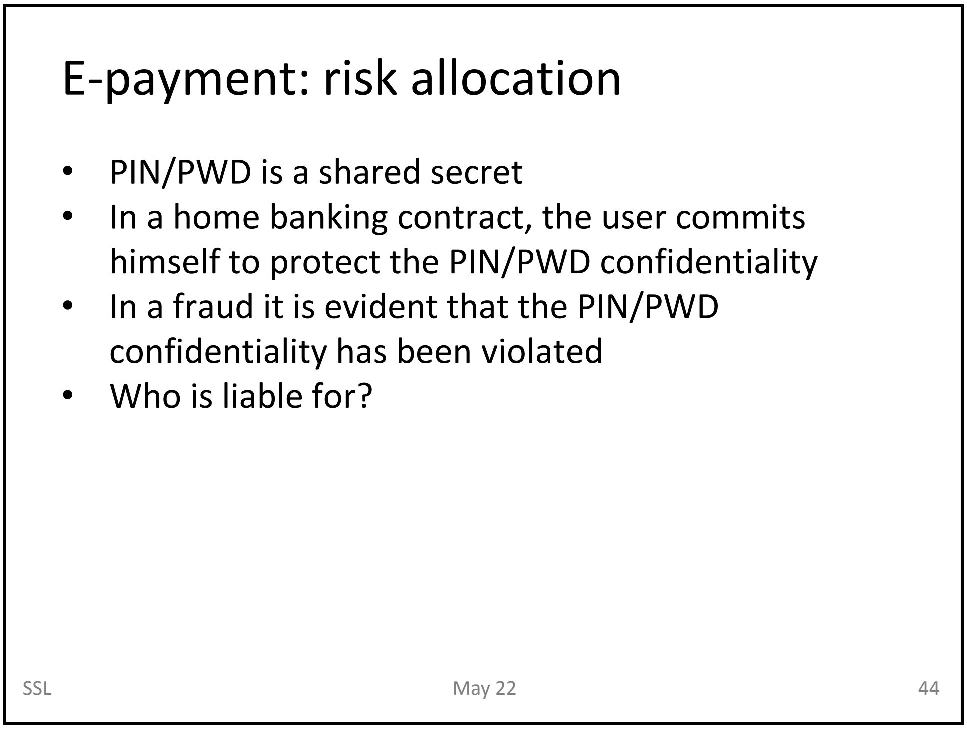
SSL May 22 42

42



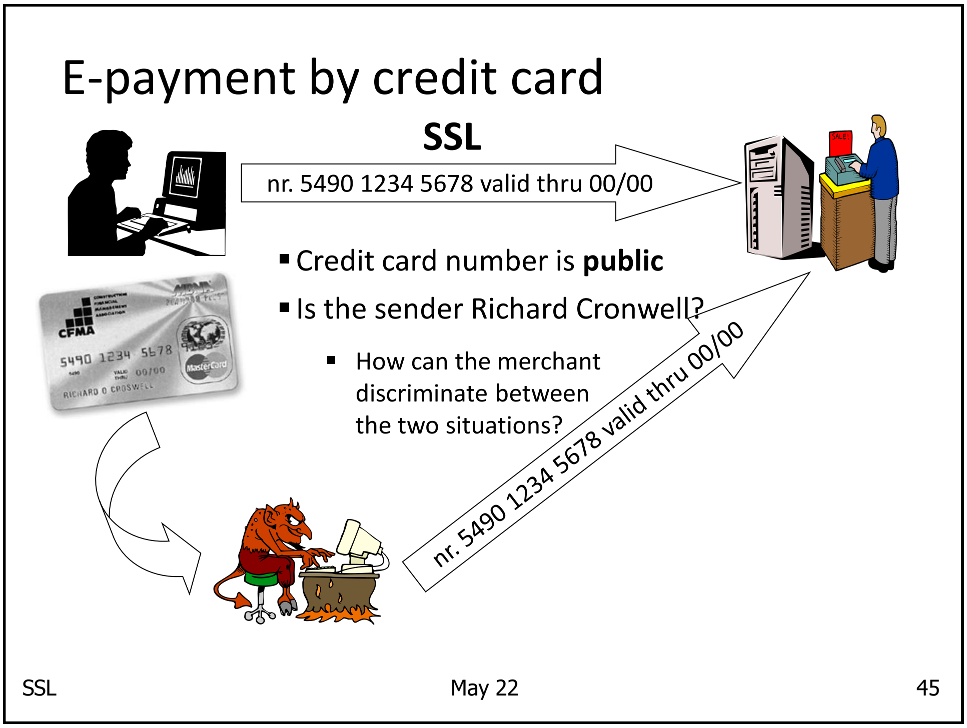
Is it the right certificate?

- SSL operates at the transport level
- Browser controls
  - Browser warns user if the URL known to the browser is not equal to that in the certificate (mismatch)
  - Browser warns user whether a certificate is signed by an unknown CA (self-signed certificates)
  - The user has the last word
    - The clickthrough phenomenon: does the user understand security? Usability vs security
  - These controls may be not sufficient for all web applications
  - Browser have a largely variable behaviour in this respect (what to warn; when to warn)



E-payment: risk allocation

- PIN/PWD is a shared secret
- In a home banking contract, the user commits himself to protect the PIN/PWD confidentiality
- In a fraud it is evident that the PIN/PWD confidentiality has been violated
- Who is liable for?



E-payment by credit card

SSL

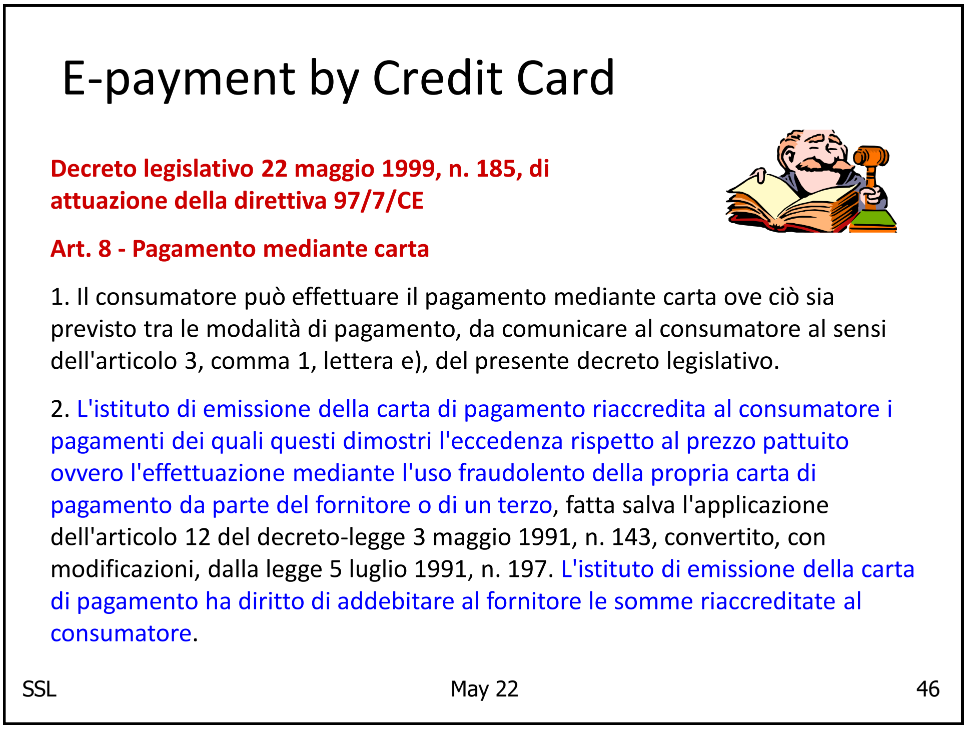
nr. 5490 1234 5678 valid thru 00/00

- Credit card number is **public**
- Is the sender Richard Cronwell?
  - How can the merchant discriminate between the two situations?

nr. 5490 1234 5678 valid thru 00/00

SSL May 22 45

45



E-payment by Credit Card

Decreto legislativo 22 maggio 1999, n. 185, di attuazione della direttiva 97/7/CE

Art. 8 - Pagamento mediante carta

1. Il consumatore può effettuare il pagamento mediante carta ove ciò sia previsto tra le modalità di pagamento, da comunicare al consumatore al sensi dell'articolo 3, comma 1, lettera e), del presente decreto legislativo.

2. L'istituto di emissione della carta di pagamento riaccredita al consumatore i pagamenti dei quali questi dimostri l'eccedenza rispetto al prezzo pattuito ovvero l'effettuazione mediante l'uso fraudolento della propria carta di pagamento da parte del fornitore o di un terzo, fatta salva l'applicazione dell'articolo 12 del decreto-legge 3 maggio 1991, n. 143, convertito, con modificazioni, dalla legge 5 luglio 1991, n. 197. L'istituto di emissione della carta di pagamento ha diritto di addebitare al fornitore le somme riaccreditate al consumatore.

SSL May 22 46

46

E-payment by Credit Card

- 🇮🇹 **Gli istituti di emissione**, cui compete l'autorizzazione dell'operazione di pagamento, nonché i soggetti che rendono tecnicamente possibile la transazione on-line, **sono tenuti a controllare la correttezza del numero della carta e la data della sua scadenza ma non anche la corrispondenza tra il numero fornito e l'effettivo titolare**
- 🇺🇸 Gli istituti di emissione verificano la corrispondenza tra numero della carta di credito comunicato per effettuare una transazione on-line ed il nominativo fornito da colui che la effettua.

  Ad esempio, l'**Address Verification Service (AVS)** verifica che l'indirizzo di consegna sia quello con cui il possessore della carta è registrato
- 🇪🇺 In Europa il grado di sicurezza nelle transazioni on-line è minore e quindi il commercio elettronico è destinato ad incontrare resistenze anche da parte dei fornitori di che sopportano rischi elevati

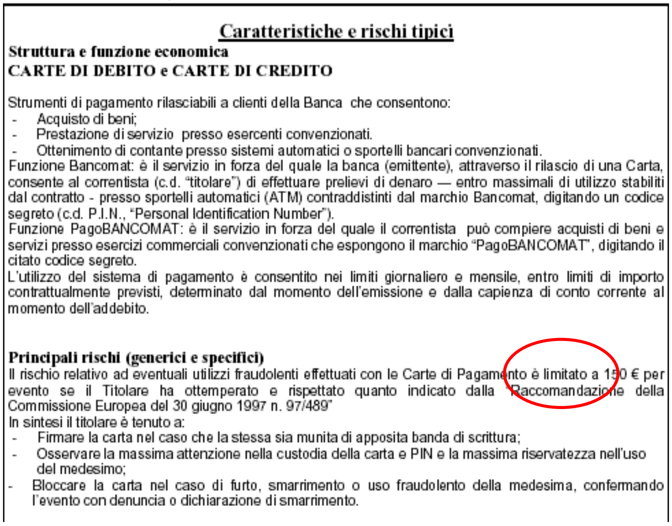
E-payment by Credit Card: risk allocation

- 🇮🇹 Il fornitore di beni o servizi on-line **è tenuto ad accollarsi il rischio** della rivalsa degli istituti di emissione qualora, in caso di uso fraudolento delle carta, questi riaccreditano le corrispondenti somme al legittimo titolare.
- 🇮🇹 La legge **non consente** al fornitore di liberarsi dall'obbligo della restituzione delle somme agli istituti di emissione qualora dimostri
  1. di avere usato tutte le cautele necessarie e possibili ad evitare l'uso fraudolento della carta di credito
  2. che il fatto è stato causato dal caso fortuito.
- 🇮🇹 I fornitori dovranno usare tutte le cautele del caso per potere, nel caso di uso fraudolento di carte di credito, perlomeno rintracciare l'illegittimo utilizzatore e rivalersi su questo.

  Le conseguenze derivanti dall'addebito delle somme riaccreditate al titolare della carta potrebbero poi essere annullate contraendo una **assicurazione** a copertura dei danni (economici) derivanti da tale circostanza.

E-payment by Credit Card

Foglio informativo sulle operazioni e servizi offerti alla clientela (CariPrato)



Caratteristiche e rischi tipici

Struttura e funzione economica
CARTE DI DEBITO e CARTE DI CREDITO

Strumenti di pagamento rilasciabili a clienti della Banca che consentono:

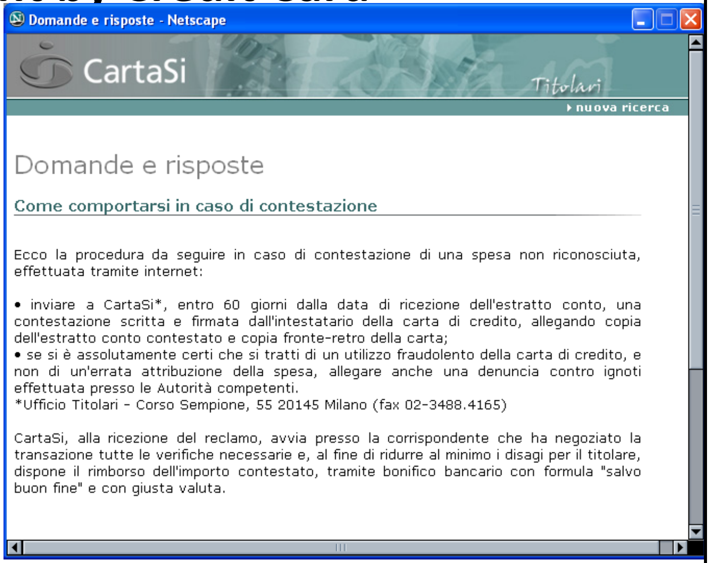
- Acquisto di beni;
- Prestazione di servizio presso esercenti convenzionati.
- Ottenimento di contante presso sistemi automatici o sportelli bancari convenzionati.

Funzione Bancomat: è il servizio in forza del quale la banca (emittente), attraverso il rilascio di una Carta, consente al correntista (c.d. "titolare") di effettuare prelievi di denaro — entro massimali di utilizzo stabiliti dal contratto - presso sportelli automatici (ATM) contraddistinti dal marchio Bancomat, digitando un codice segreto (c.d. P.I.N., "Personal Identification Number").
Funzione PagoBANCOMAT: è il servizio in forza del quale il correntista può compiere acquisti di beni e servizi presso esercizi commerciali convenzionati che espongono il marchio "PagoBANCOMAT", digitando il citato codice segreto.
L'utilizzo del sistema di pagamento è consentito nei limiti giornaliero e mensile, entro limiti di importo contrattualmente previsti, determinato dal momento dell'emissione e dalla capienza di conto corrente al momento dell'addebito.

Principali rischi (generici e specifici)
Il rischio relativo ad eventuali utilizzi fraudolenti effettuati con le Carte di Pagamento è limitato a 150 € per evento se il Titolare ha ottemperato e rispettato quanto indicato dalla "Raccomandazione della Commissione Europea del 30 giugno 1997 n. 97/489"
In sintesi il titolare è tenuto a:

- Firmare la carta nel caso che la stessa sia munita di apposita banda di scrittura;
- Osservare la massima attenzione nella custodia della carta e PIN e la massima riservatezza nell'uso del medesimo;
- Bloccare la carta nel caso di furto, smarrimento o uso fraudolento della medesima, confermando l'evento con denuncia o dichiarazione di smarrimento.

E-payment by Credit Card



Domande e risposte

Come comportarsi in caso di contestazione

Ecco la procedura da seguire in caso di contestazione di una spesa non riconosciuta, effettuata tramite internet:

- inviare a CartaSi*, entro 60 giorni dalla data di ricezione dell'estratto conto, una contestazione scritta e firmata dall'intestatario della carta di credito, allegando copia dell'estratto conto contestato e copia fronte-retro della carta;
- se si è assolutamente certi che si tratti di un utilizzo fraudolento della carta di credito, e non di un'errata attribuzione della spesa, allegare anche una denuncia contro ignoti effettuata presso le Autorità competenti.

*Ufficio Titolari - Corso Sempione, 55 20145 Milano (fax 02-3488.4165)

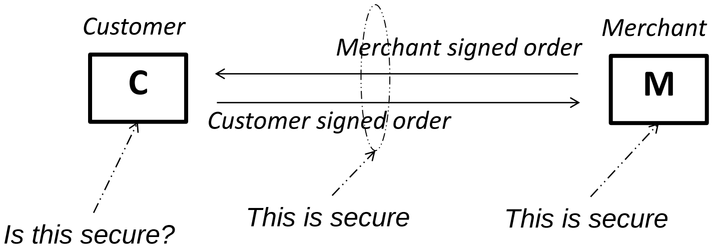
CartaSi, alla ricezione del reclamo, avvia presso la corrispondente che ha negoziato la transazione tutte le verifiche necessarie e, al fine di ridurre al minimo i disagi per il titolare, dispone il rimborso dell'importo contestato, tramite bonifico bancario con formula "salvo buon fine" e con giusta valuta.

Secure Electronic Transactions

- SET was built to answer to these problems
- SET has been designed and implemented in the late 90’s
  - Commissioned by Visa and Mastercard
  - Involves all (IBM, Microsoft,…)
- SET was a failure
  - Too “heavy” and too expensive
  - Specifications takes more than 1000 pages (!)
- We are interested in the risk allocation

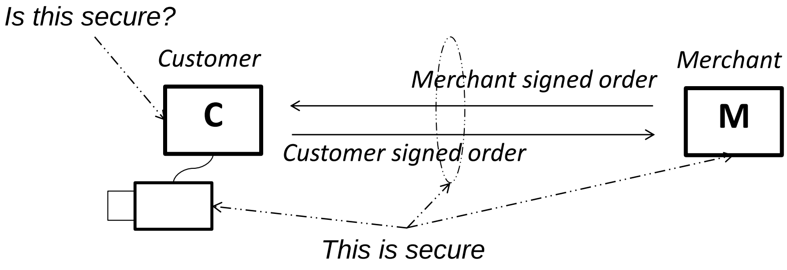
Secure Electronic Transactions

- SET requires a PKI in place
- A (privK, pubK) pair is stored at M and C
- *If an order is signed by your key you cannot repudiate it*
  - **The risk is allocated on the customer**
- M and C are assumed trusted devices!
  - Stealing a privK is equivalent to stealing a file



Secure Electronic Transactions

- Do smart cards help?
  - Loosing a piece of plastic vs. loosing a file
  - Is what you see what you sign?



SSL: Pros and Cons

- Pros
  - SSL is a well-designed, robust and secure protocol
- Cons
  - SSL protects communication only
  - User has to check security parameters
  - SSL is vulnerable to name spoofing