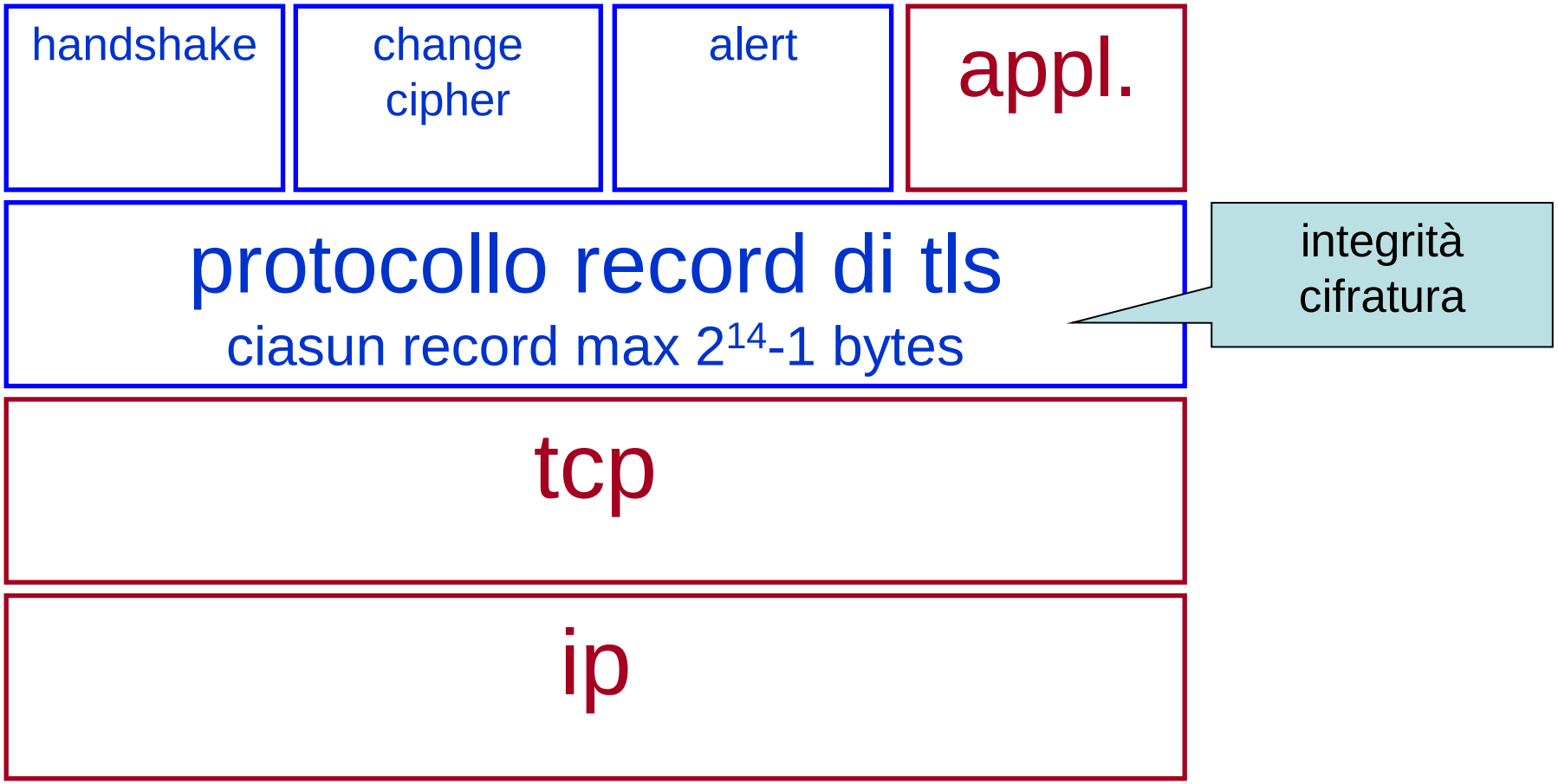
applicazioni della crittografia
protocolli di trasporto ssl, tls e ssh

descrizione generale

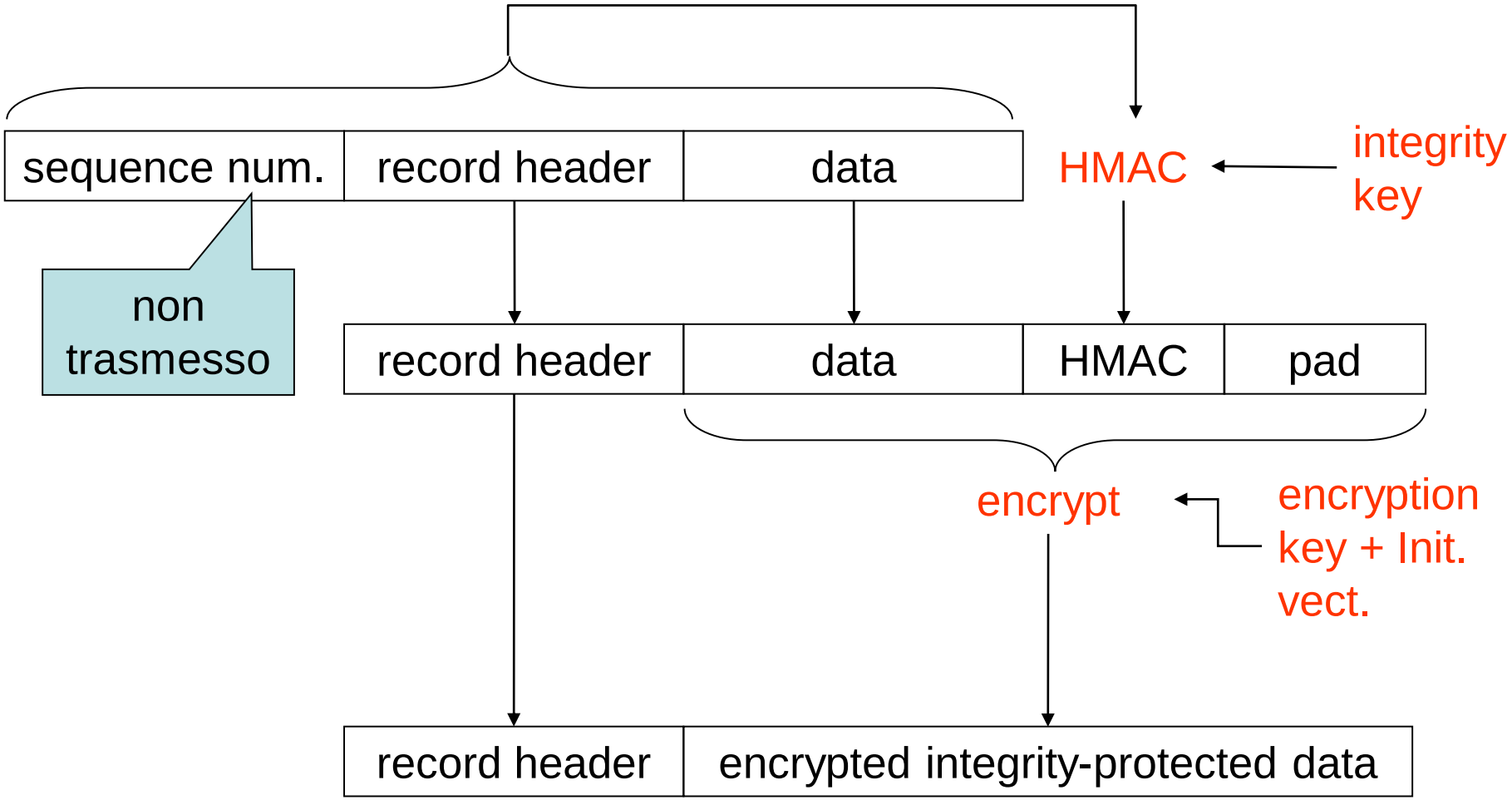
- Secure Socket Layer (Netscape)
  - versione 2, obsoleta, qualche vulnerabilità
  - versione 3
- Transport Layer Security (IETF, rfc2246)
  - versione 1, molto simile a SSLv3 ma incompatibile
  - v1.1, v1.2, v1.3

- protocolli del tutto generali
  - usati spesso per http (https su porta 443)
  - usati anche per imap, pop, telnet
- supportati dalle applicazioni più diffuse
- sono protocolli piuttosto complessi

il rapporto con la pila osi



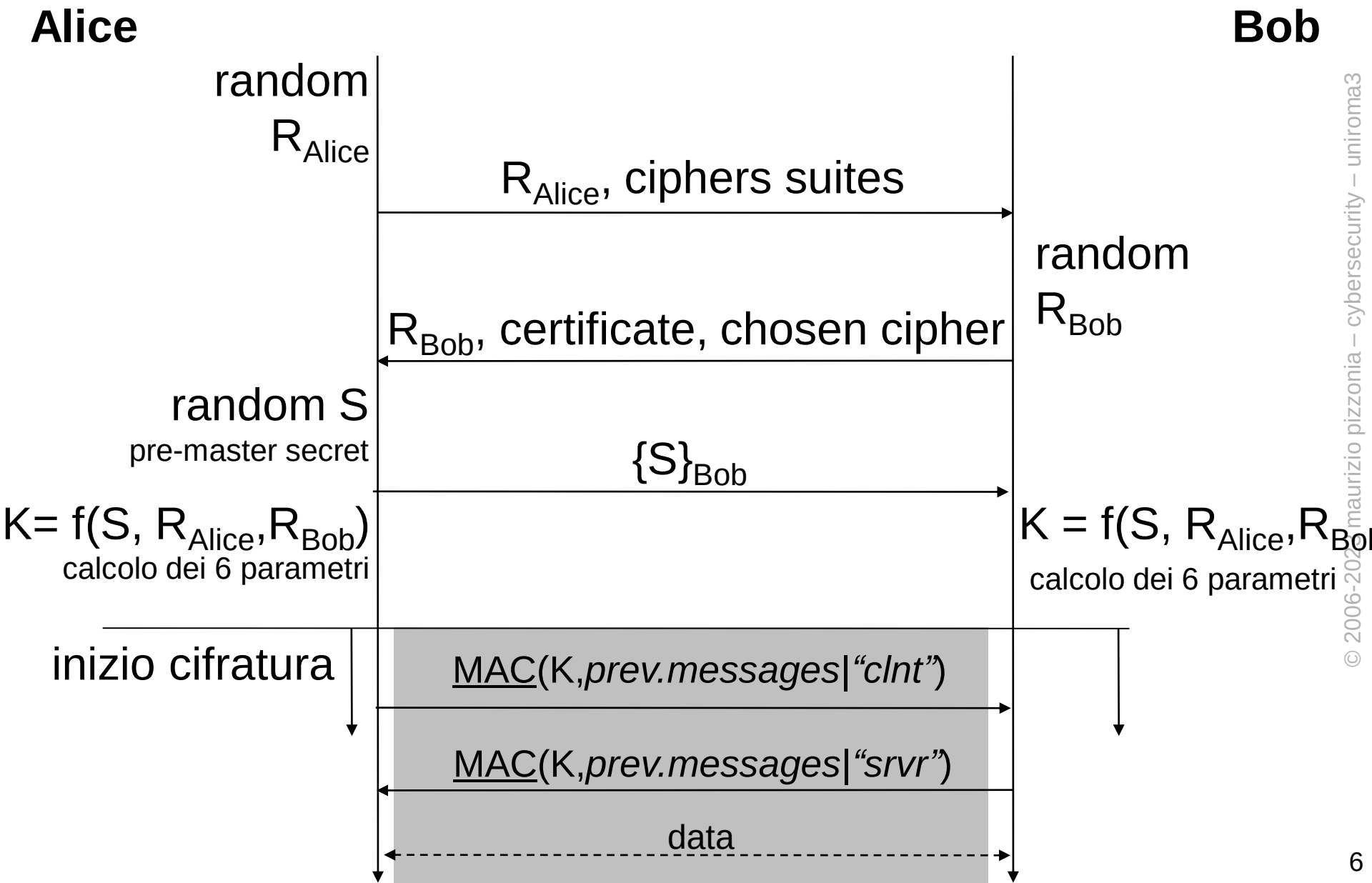
cifratura dei record



stato della connessione

- per iniziare una connessione cifrata i due devono accordarsi su...
  - algoritmo di cifratura
  - hash function per HMAC
  - come scambiare “la chiave” (pre-master secret)
- ... e su i seguenti 3 segreti per ciascuna direzione (totale 6)
  - integrity protection key
  - encryption key
  - Initialization Vector (necessario per molti algoritmi di cifratura a blocchi, es. DES)
  - sono tutti calcolati a partire dal pre-master secret

scambio rsa



esercizio

- perché SSLv3/TLS inseriscono un controllo di integrità per l'handshake?

varianti

- il client può fornire un proprio certificato per essere autenticato
- diffie-hellman
  - il server interviene nella creazione di S
  - autenticato con RSA o DSS
- diffie-hellman ephimeral
  - le chiavi vengono generate per la sessione e poi dimenticate
  - forward secrecy
- session resumption

cipher suites di TLS

- una cipher suite è un insieme di algoritmi da usare per la cifratura, l'integrita, e lo scambio di chiavi
- esempio di stringa identificativa di una cipher suite

TLS_RSA_WITH_3DES_EDE_CBC_SHA

chiavi scabiate con RSA

cifratura: 3DES nella variante EDE CBC

integrità: HMAC con SHA-1

alcune chipher suites di TLS

default, usato solo per handshake

TLS_NULL_WITH_NULL_NULL

TLS_RSA_WITH_NULL_MD5
TLS_RSA_WITH_NULL_SHA
TLS_RSA_EXPORT_WITH_RC4_40_MD5
TLS_RSA_WITH_RC4_128_MD5
TLS_RSA_WITH_RC4_128_SHA
TLS_RSA_EXPORT_WITH_RC2_CBC_40_MD5
TLS_RSA_WITH_IDEA_CBC_SHA
TLS_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_RSA_WITH_DES_CBC_SHA
TLS_RSA_WITH_3DES_EDE_CBC_SHA

TLS_DH_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_DSS_WITH_DES_CBC_SHA
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA
TLS_DH_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_RSA_WITH_DES_CBC_SHA
TLS_DH_RSA_WITH_3DES_EDE_CBC_SHA

forward secrecy

TLS_DHE_DSS_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_DSS_WITH_DES_CBC_SHA
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA
TLS_DHE_RSA_EXPORT_WITH_DES40_CBC_SHA
TLS_DHE_RSA_WITH_DES_CBC_SHA
TLS_DHE_RSA_WITH_3DES_EDE_CBC_SHA

TLS_DH_anon_EXPORT_WITH_RC4_40_MD5
TLS_DH_anon_WITH_RC4_128_MD5
TLS_DH_anon_EXPORT_WITH_DES40_CBC_SHA
TLS_DH_anon_WITH_DES_CBC_SHA
TLS_DH_anon_WITH_3DES_EDE_CBC_SHA

deprecati perché DH non autenticato è vulnerabile a MitM

elliptic curve cryptography (ECC)

- stessi algoritmi nuova definizione di gruppo
- molto più efficiente a parità di sicurezza
- da RFC4492...

Symmetric	ECC	DH/DSA/RSA
80	163	1024
112	233	2048
128	283	3072
192	409	7680
256	571	15360

esempi di cipher suites con ECC

- TLS_ECDH_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDH_anon_WITH_3DES_EDE_CBC_SHA

currently recommended suites in TLS 1.2

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

TLS_**DHE**_RSA_WITH_AES_128_GCM_SHA256
TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

TLS_ECDHE_ECDSA_WITH_**CHACHA20**_**POLY1305**_SHA256
TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256
TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305

adopted to avoid relying only on AES
(What if a vulnerability is found in AES? We need a recommended alteranative!)

TLS 1.3

- simpler cypher suites names
- certain algorithms are inferred by other means among limited alternatives
  - key exchange method: DHE or ECDHE
  - authentication: RSA or ECDSA
- shorter suites list

TLS_AES_256_GCM_SHA384

TLS_CHACHA20_POLY1305_SHA256

TLS_AES_128_GCM_SHA256

TLS_AES_128_CCM_8_SHA256

TLS_AES_128_CCM_SHA256

- also dropped support for some rarely used, or weak, features
  - Compression, CBC, Non-AEAD ciphers, Renegotiation of encryption parameters, RC4, DSA, MD5, SHA1, RSA Key Exchange, DH, ECDH, integrity-only ciphers

TLS 1.2 vs 1.3

- TLS 1.2 is not deprecated
  - TLS 1.1 and 1.0 are deprecated
- TLS 1.2 is not going to be deprecated soon
- TLS 1.3 is…
  - simpler
  - hard to configure insecurely
  - supported only by recent software
  - suitable for the vast majority of use-cases
  - unsuitable for certain “niche” use-cases
    - e.g. very small or low power devices, certain industrial security settings

ssh

- ssh è un concorrente di ssl/tls
  - v1 (vulnerabile), v2 attualmente in uso
- del tutto generale
  - usato sopratutto come telnet criptato
  - si può fare tunneling di qualsiasi cosa in ssh (opzioni –L e –R)
    - ma ora si può fare anche con ssl (vedi “stunnel”)
- companion protocols/commands
  - scp, sftp
- diffusione
  - famoso (implementazione open: openssh)
  - ampiamente supportato
  - standardizzato
    - rfc 4250-4256 e seguenti
  - supporta autenticazione RSA e anche certificati