Autenticazione

Autenticazione

- L’autenticazione è uno degli elementi più critici nella sicurezza
- Essa permette ad un’entità (una persona o un sistema) di dichiarare la propria identità ad un’altra entità
- Una buona infrastruttura di autenticazione già protegge dalla maggior parte degli attacchi
- Autenticarsi significa disporre di **credenziali**
- Di solito l’entità che vuole autenticarsi deve dimostrare la conoscenza di un segreto all’altra

Tipi di autenticazione

- Autenticazione dei **dati**
  - prova dell'origine dei dati
  - normalmente associata all'integrità
- Autenticazione dei **peer**
  - prova dell'identità dell'altro estremo della comunicazione
  - autenticazione singola o mutua
  - autenticazione del client (ad esempio username + password)
  - autenticazione del server (meno frequente ma molto importante!)
- AuthN: Authentication
- AuthZ: Authorization

Message authentication code (MAC)

- Metodo usato per verificare l'integrità e l'autenticità di un messaggio
- Può anche verificare l'identità del mittente (diversamente dal solo hash)
- Richiede l'utilizzo di una chiave segreta (diversamente dal solo hash)
- Si può basare sulla combinazione di una funzione hash con una chiave segreta
- Oppure può sfruttare un cifrario simmetrico

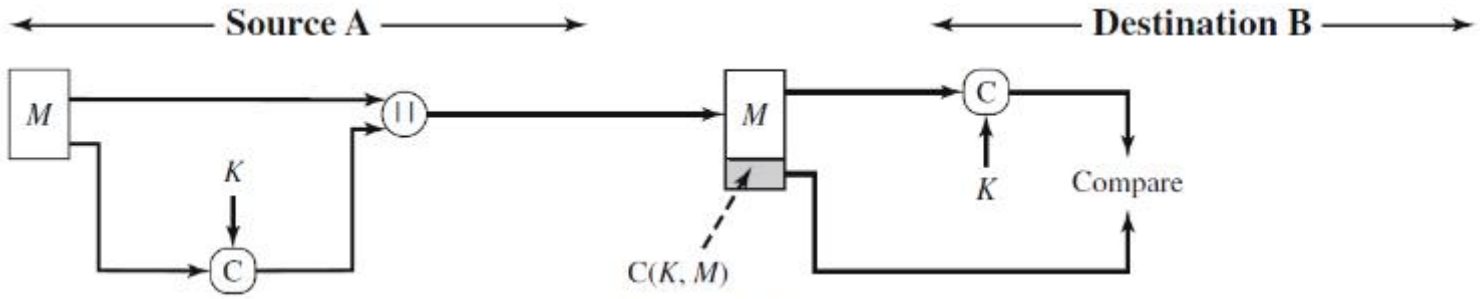
Message authentication code (MAC) - 2

- Le due parti (A e B) condividono una chiave segreta (K)
- Quando A deve inviare un messaggio (M) a B, ne calcola il MAC:

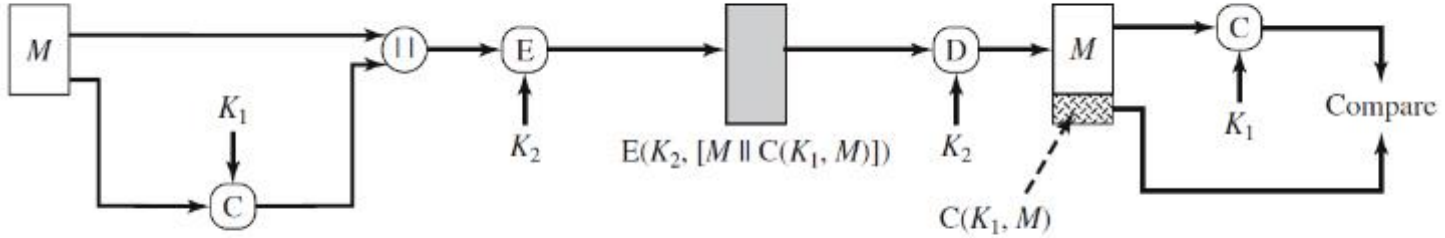
$$MAC = C(K, M)$$

  - C = funzione MAC
  - MAC = valore del MAC
- Il risultato viene trasmesso a B insieme al messaggio M
- B ripete il calcolo e verifica che il suo valore del MAC coincida con quello ricevuto

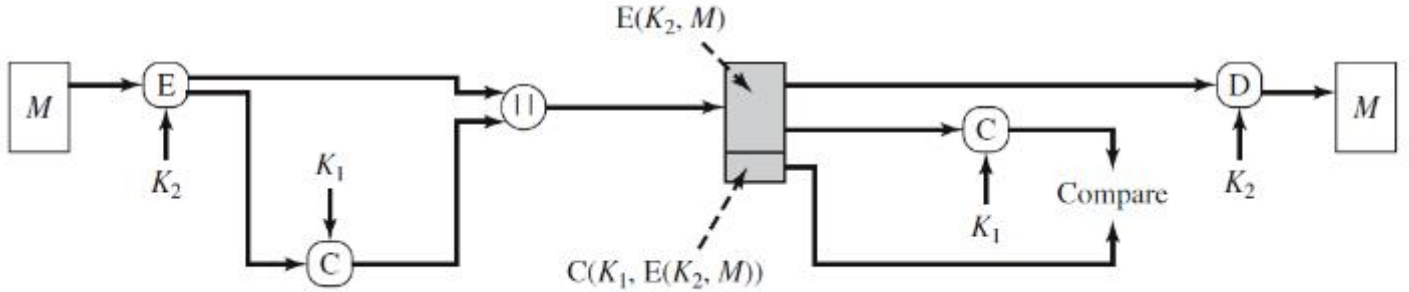
Message authentication code (MAC) - 3



(a) Message authentication



(b) Message authentication and confidentiality; authentication tied to plaintext



(c) Message authentication and confidentiality; authentication tied to ciphertext

Hash con chiave: HMAC

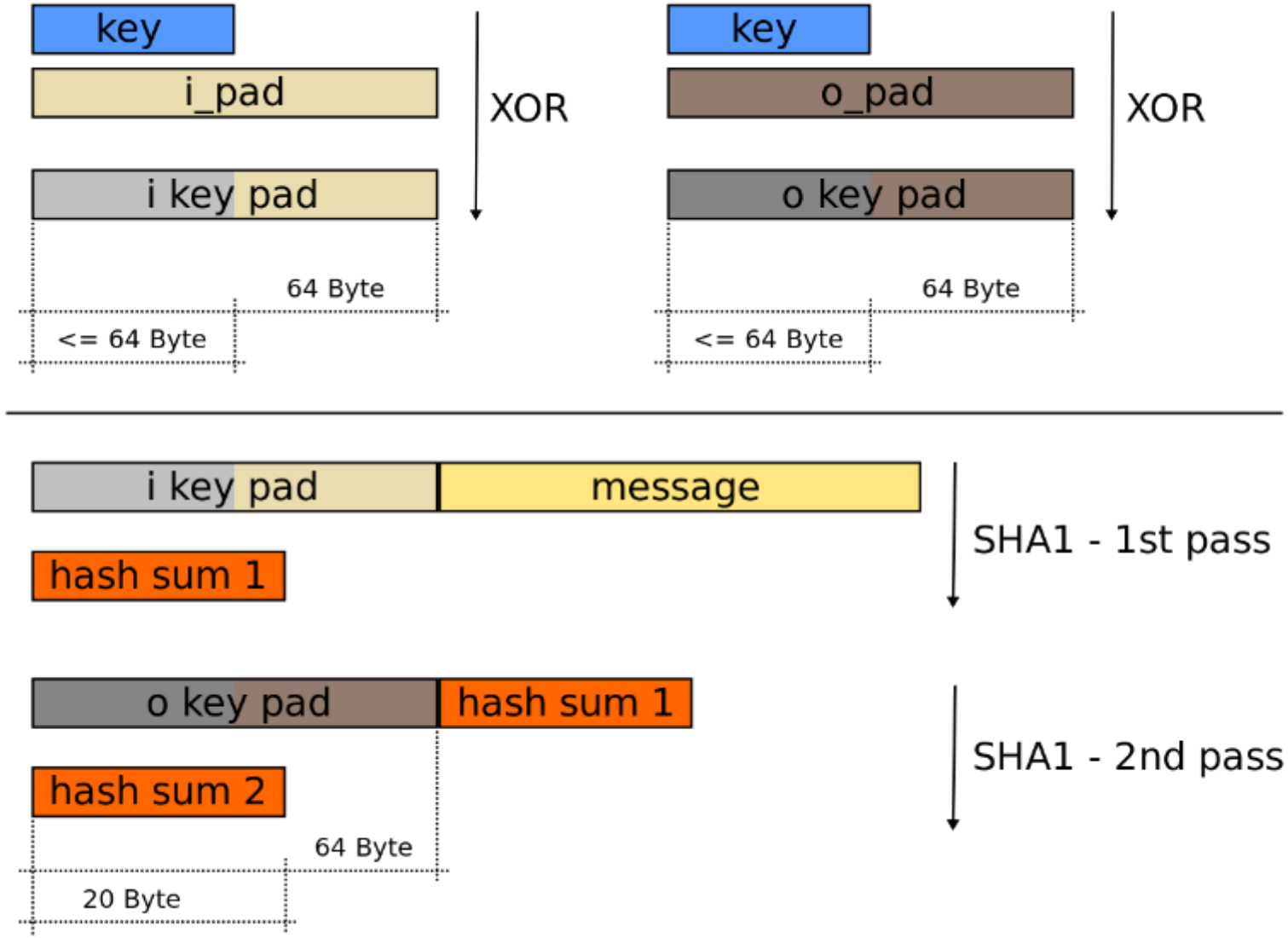
- Algoritmo MAC basato su una funzione hash
- Molte funzioni hash (come MD5, SHA-1 e SHA-2) usano uno stato interno che evolve in funzione dei blocchi del messaggio ed assume un valore finale che coincide col digest
- Questo le rende vulnerabili ad **attacchi basati su estensione** del messaggio:
  - Partendo dall'ultimo valore della variabile di stato, si può prolungare il messaggio e continuare il calcolo del digest
  - Ciò potrebbe consentire ad Eve di calcolare un HMAC valido per un messaggio esteso pur non conoscendo la chiave

Hash con chiave: HMAC (2)

- Per evitare tali attacchi, HMAC usa una doppia applicazione della funzione hash

- La chiave segreta viene inizialmente usata per derivare due altre chiavi:
  - la **chiave interna** viene usata dal primo passaggio dell'algoritmo per produrre un hash interno partendo dal messaggio
  - la **chiave esterna** viene usata dal secondo passaggio dell'algoritmo per calcolare il valore di HMAC partendo dal risultato del primo passaggio

HMAC basato su SHA-1



Autenticazione dei peer

- Un utente può avere **credenziali** di diverso tipo:
  - Quello che sa (password, pin…)
  - Quello che ha (badge, smartcard…)
  - Quello che è (impronte digitali, caratteristiche vocali, analisi della retina…)
  - Combinazioni delle precedenti

- Autenticazione a **singolo fattore**
  - Usa un solo tipo di credenziali per autenticare l'utente. Garantisce un livello di sicurezza minimo ed è sconsigliata per applicazioni sensibili (ad es. finanziarie).

- Autenticazione a **più fattori**
  - Usa due o più tipi di credenziali e fornisce livelli di sicurezza maggiori.

Authentication, Authorization, Accounting

- Dopo avere verificato l'identità del soggetto, il sistema deve determinare i suoi diritti e privilegi: questo è il processo di **autorizzazione**.
- Il sistema dovrebbe inoltre tenere traccia degli eventi relativi alle autenticazioni e autorizzazioni avvenute, tramite la raccolta di file di **log**.
- Tali log servono per ragioni di «contabilità» (**accounting**), ma sono anche utili per alcune funzioni di sicurezza, come il rilevamento di intrusioni.

Autenticazione singola o mutua

- **Autenticazione singola**: il client autentica il server oppure il server autentica il client
- **Autenticazione mutua**: il client autentica il server e allo stesso tempo il server autentica il client
- Esempio di autenticazione singola: consultazione di un sito informativo (Wikipedia)
  - Il client deve essere certo che il server è autentico
  - Il server non ha bisogno di autenticare il client
- Esempio di autenticazione mutua: consultazione della posta elettronica
  - Il client deve essere certo che il server è autentico
  - Il server deve accertarsi dell'identità del client

Requisiti dell'autenticazione

- **Non trasferibilità**: Alice (authenticator) non dovrebbe essere in grado di riutilizzare la prova fornita da Bob (client), cioè Alice non dovrebbe essere in grado di impersonare Bob.

- **Non sostituibilità**: Eve che osserva gli scambi tra Alice e Bob non dovrebbe essere in grado di impersonare né Alice né Bob.

- Tali requisiti devono essere soddisfatti anche quando Alice e Bob eseguono autenticazioni multiple e Eve può fare un numero infinito di tentativi di autenticazione.

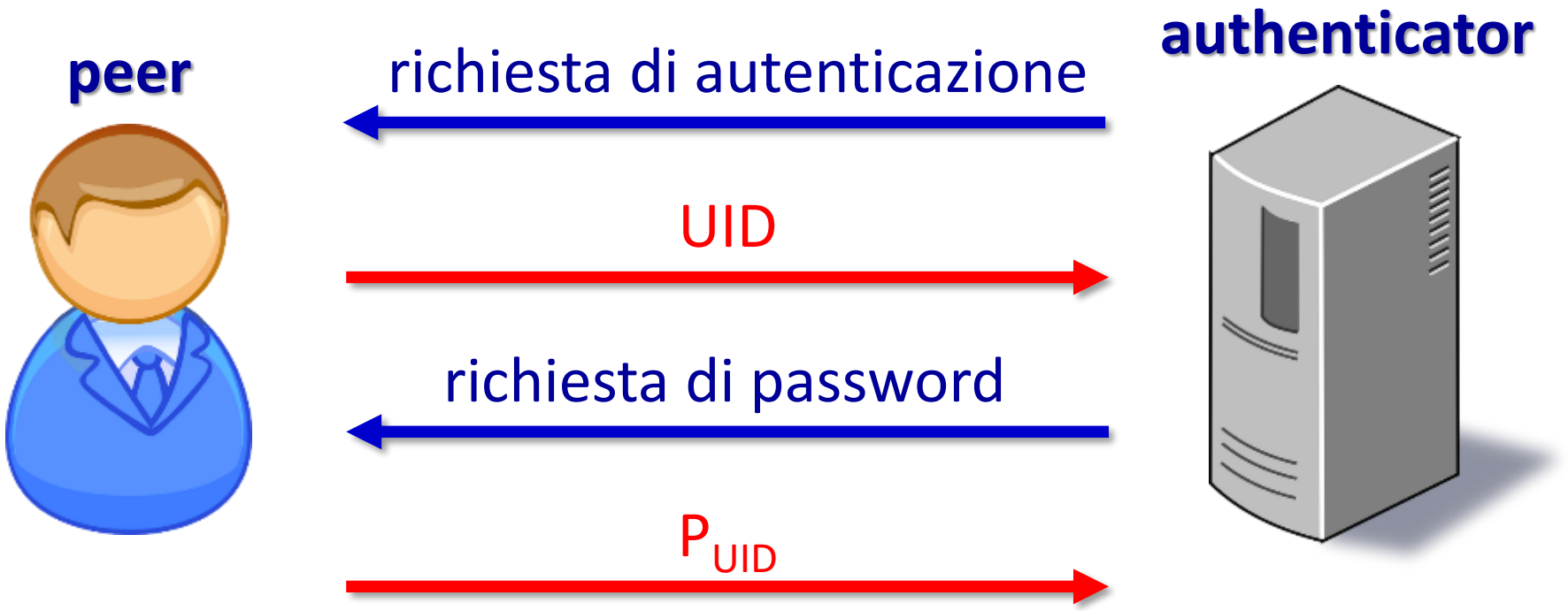
Protocolli di Autenticazione

- Lo scambio delle credenziali deve essere sicuro
- Per questo è fondamentale avere dei solidi meccanismi per gestire l'autenticazione
- Questi meccanismi sono chiamati **protocolli di autenticazione**
- Esistono numerosi protocolli di autenticazione che presentano diversi livelli di sicurezza:
  - PAP (Password Authentication Protocol)
  - CHAP (Challenge-Response Handshake Authentication Protocol)
  - MS-CHAP v1/v2 (variante Microsoft del CHAP)
  - EAP (Extensible Authentication Protocol)

PAP (Password Authentication Protocol)

- E’ la forma di autenticazione più semplice
- Richiede soltanto nome utente e password che vengono trasferiti sulla rete e confrontati con una tabella delle coppie consentite residente sul server
- E’ definito in RFC 1334
- La password attraversa la rete in chiaro
- PAP offre scarsa sicurezza e non può essere utilizzato più di una volta all’interno della stessa sessione

PAP


peer
authenticator
richiesta di autenticazione
UID
richiesta di password
P_{UID}

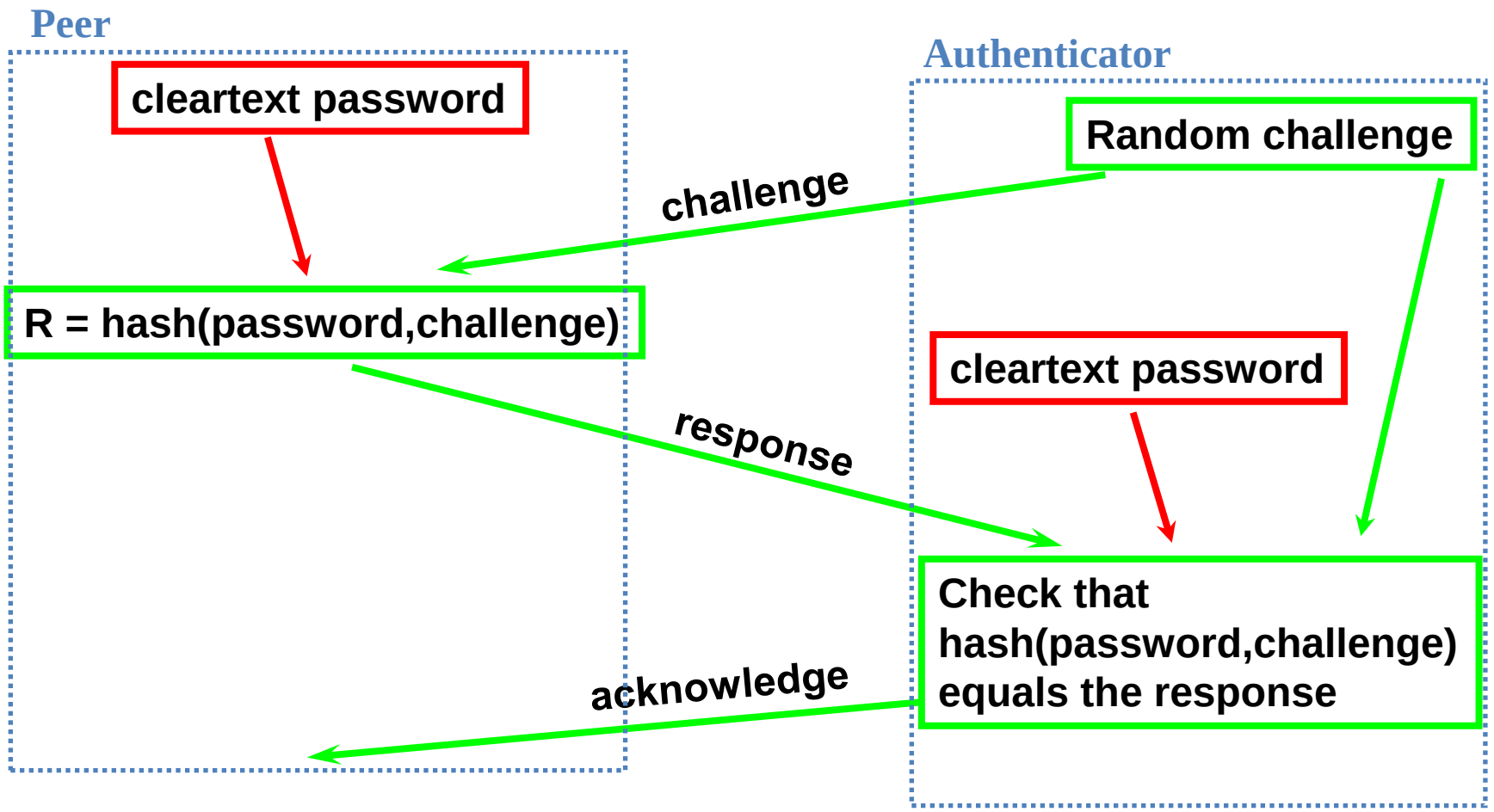
Memorizzare le password

- Non possono mai essere memorizzate in chiaro
- Per memorizzarle cifrate serve che il server memorizzi anche la chiave di cifratura
- E’ più sicuro memorizzare un digest della password
- Esistono però gli attacchi del dizionario
- Per contrastarli serve usare il **salt**

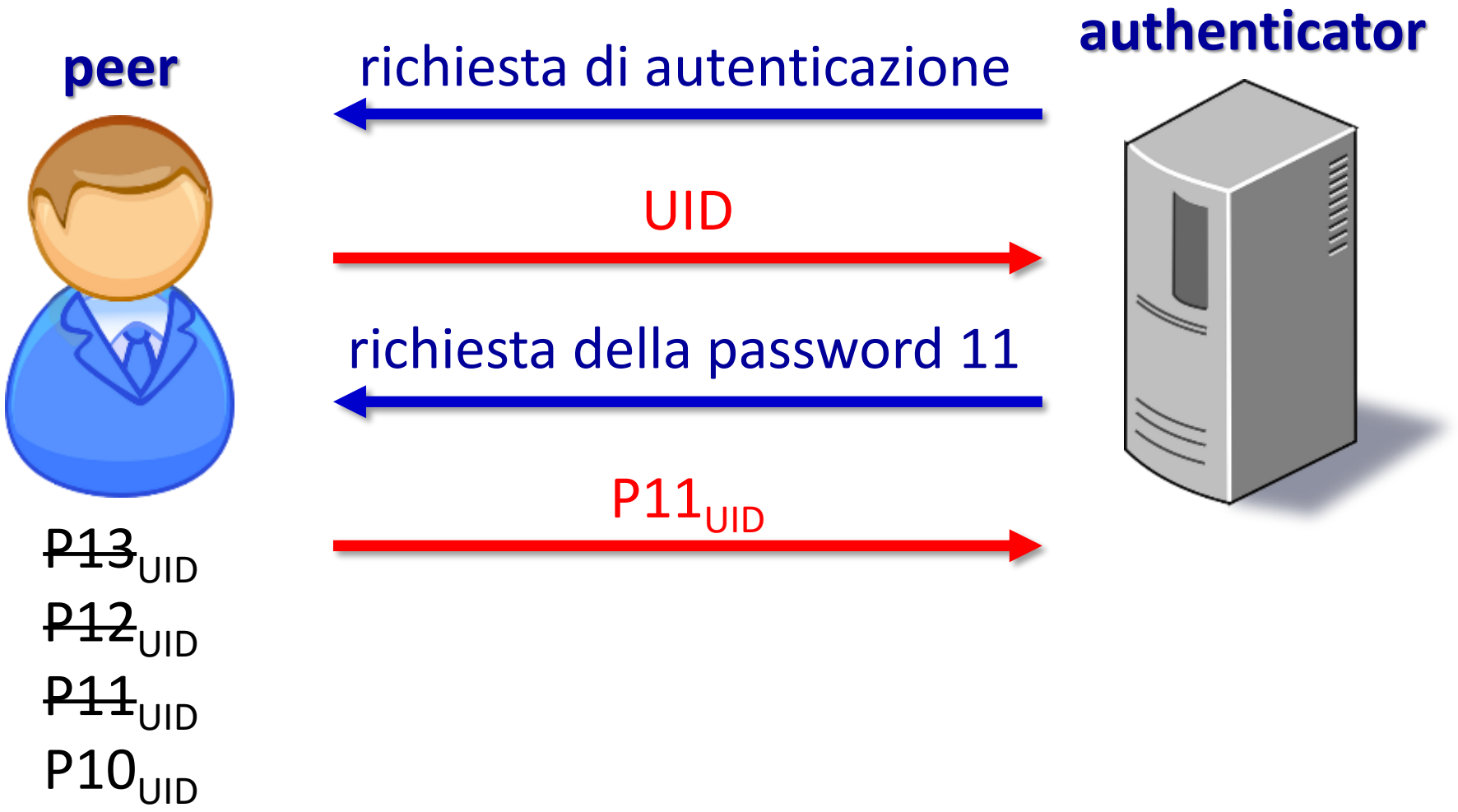
CHAP (Challenge-Handshake Authentication Protocol)

- Dopo aver stabilito la connessione, l'autenticatore invia una sfida al client che chiede di essere autenticato
- Il client prova di conoscere un segreto condiviso rispondendo alla sfida
- Può essere utilizzato più volte all'interno di una sessione per verificare se questa è stata dirottata
- E' definito in RFC 1994
- Non supporta la mutua autenticazione
- Richiede la disponibilità in chiaro del segreto condiviso

CHAP (Challenge-Handshake Authentication Protocol)

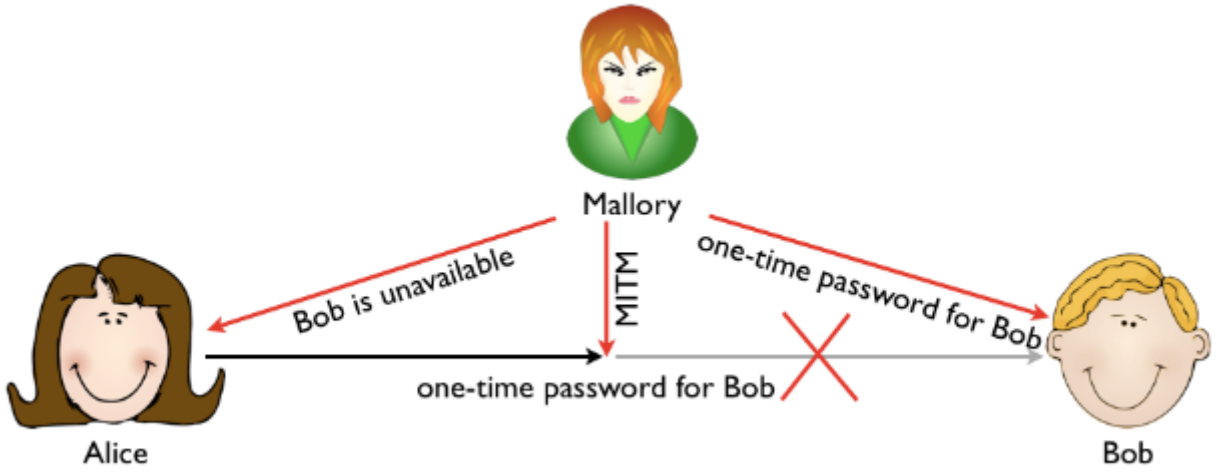


One-Time Password (OTP)



One-Time Password (OTP) - 2

- Ciascuna password è valida solo per un'esecuzione del protocollo di autenticazione
- Immune ad intercettazione (la password non si riusa)
- Soggetta ad attacchi MITM (senza protezione delle credenziali)

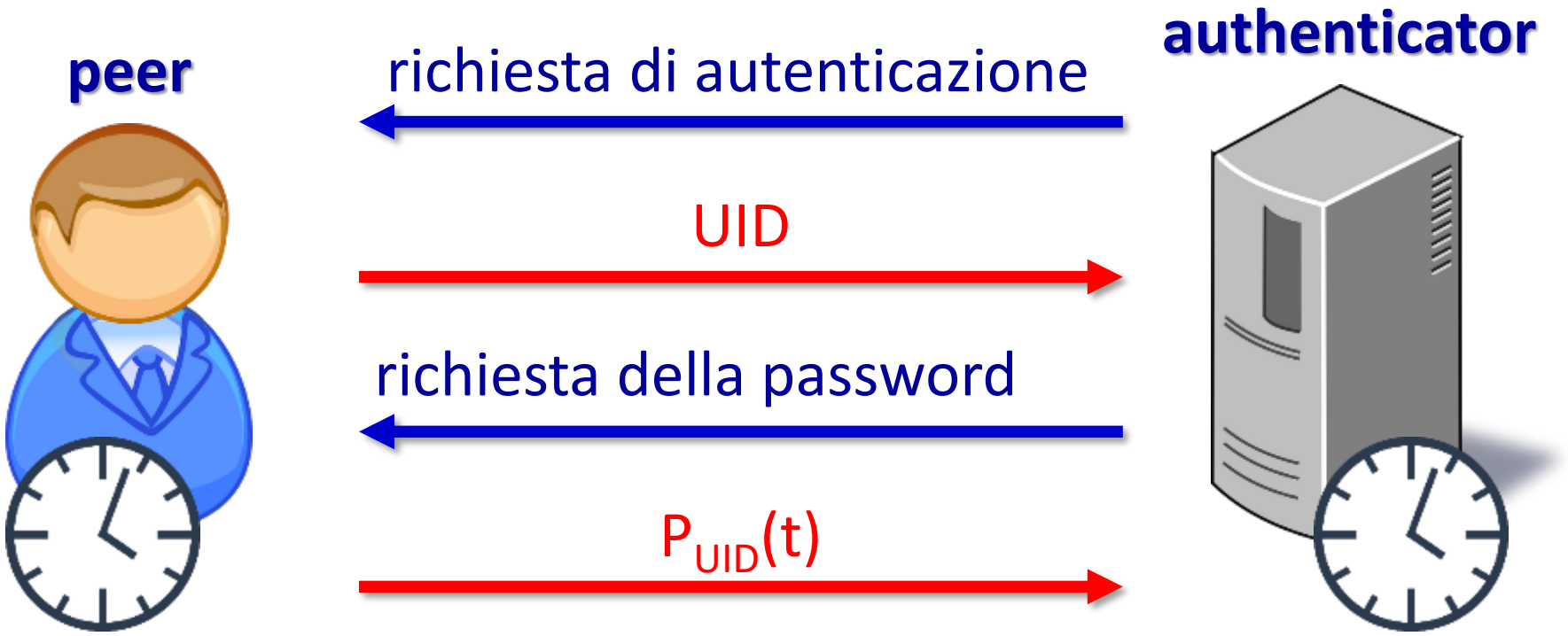


One-Time Password (OTP) - limiti

- Distribuzione delle password onerosa:
  - quantità di password
  - possibilità di esaurimento delle password

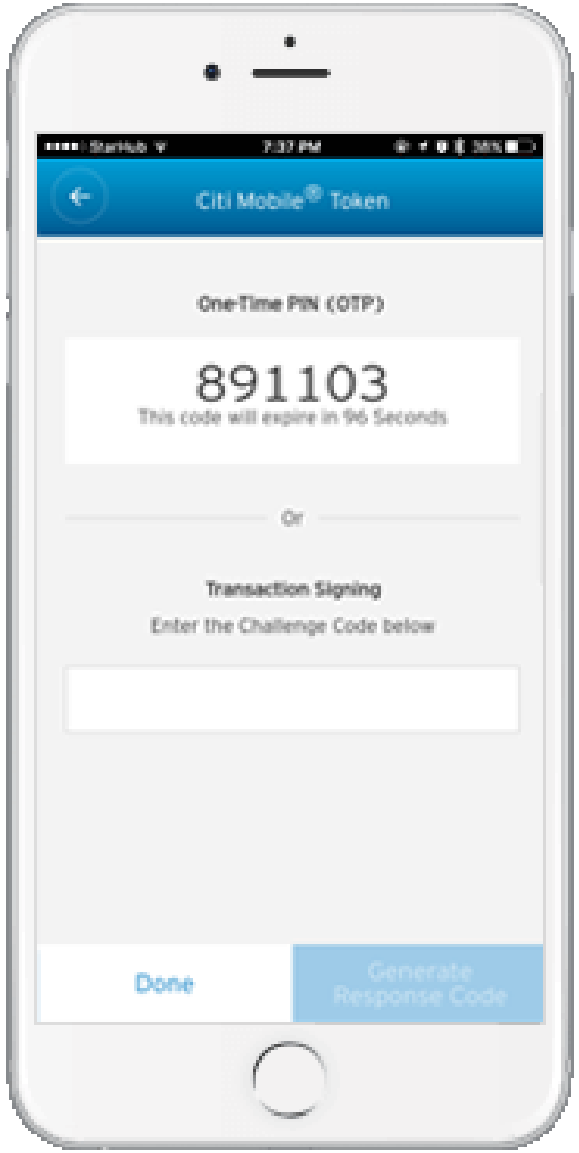
- **Soluzione**: generare le OTP tramite una funzione

OTP basata sul tempo



Il tempo può essere sostituito da un contatore

TOTP

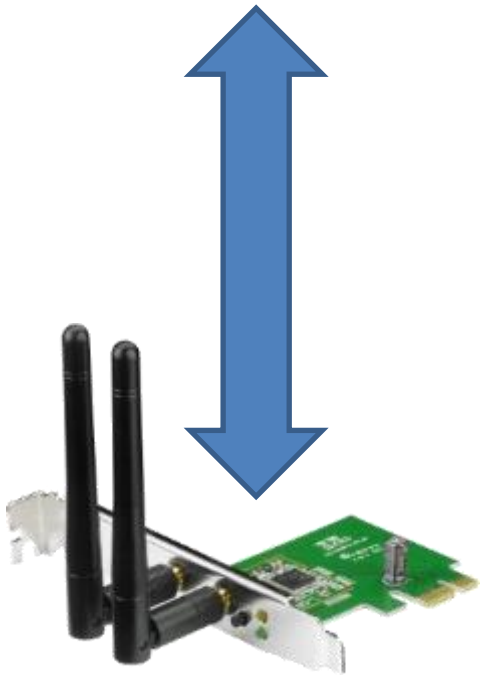

7:37 PM
Citi Mobile® Token
One-Time PIN (OTP)
891103
This code will expire in 96 Seconds
Or
Transaction Signing
Enter the Challenge Code below
Done
Generate Response Code


OTP c100
641966
ONE TIME PASSWORD

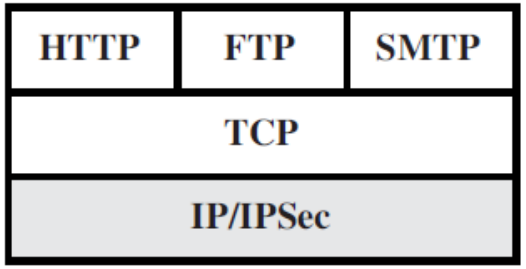
Protocolli per la Sicurezza delle Reti

Pila protocollare ISO/OSI e sicurezza

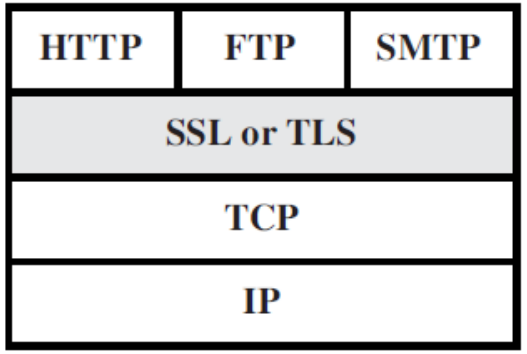
Data	**Application** Network Process to Application
Data	**Presentation** Data Representation and Encryption
Data	**Session** Interhost Communication
Segments	**Transport** End-to-End Connections and Reliability
Packets	**Network** Path Determination and IP (Logical Addressing)
Frames	**Data Link** MAC and LLC (Physical Addressing)
Bits	**Physical** Media, Signal, and Binary Transmission



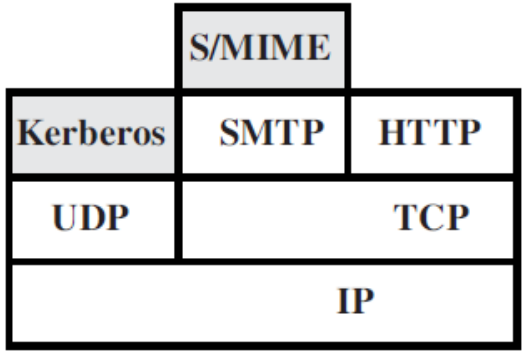
Approcci alla sicurezza delle reti



(a) Network level



(b) Transport level



(c) Application level

Sicurezza a livello Applicazione

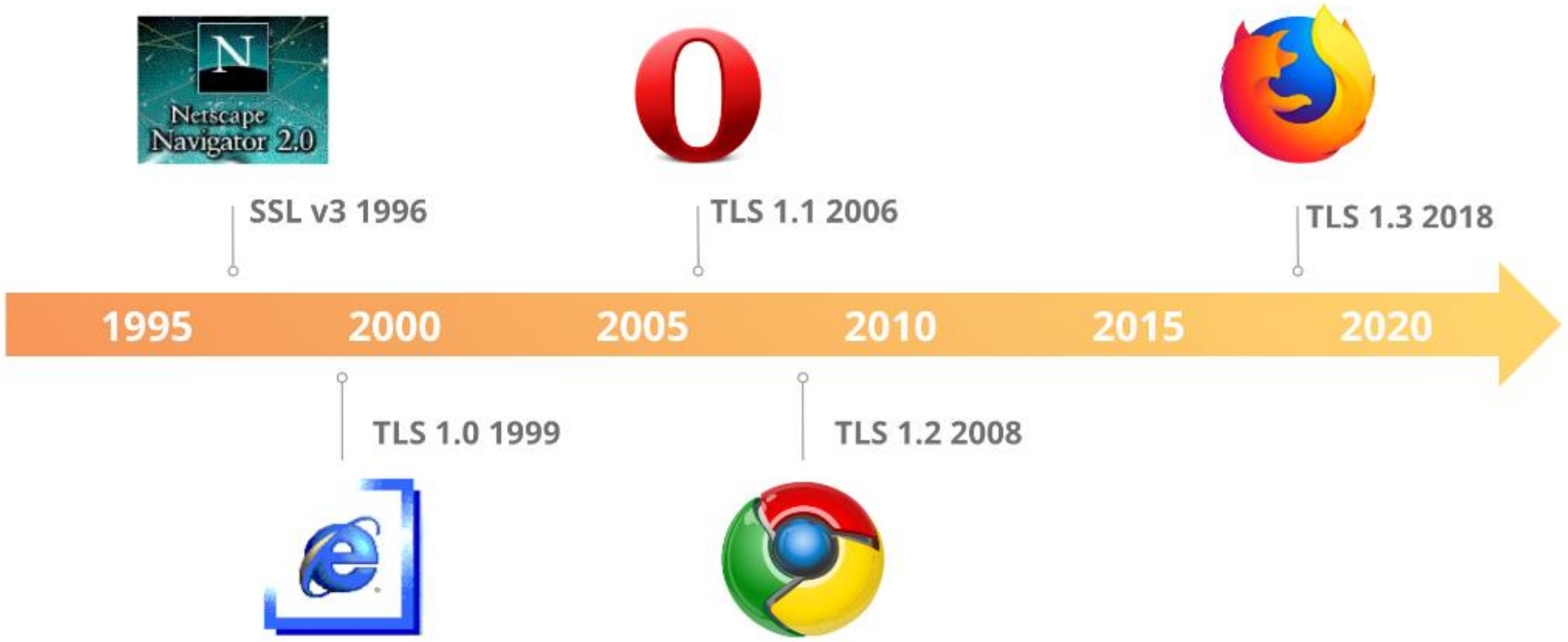
- La comunicazione tra applicazioni può includere funzioni di sicurezza custom o standard.

- Es: standard **S/MIME** (Secure / Multipurpose Internet Mail Extensions) per la sicurezza della posta elettronica.
  - Definito in RFC 3369, 3370, 3850 e 3851.
  - Supporta **crittografia a chiave pubblica** e **firma digitale**.
  - Originariamente sviluppato da RSA Data Security.
  - Usa la sintassi per messaggi sicuri PKCS#7 (Public Key Cryptography Standards), pure sviluppata da RSA Data Security e Cryptographic Message Syntax (CMS) standardizzata da IETF.
  - È incluso nella maggior parte dei moderni software di posta elettronica.

End-to-end encryption (E2EE)

- Denota comunicazioni che non vengono mai decifrate durante il trasferimento da mittente a destinatario.
- Proposto nel 2003 per proteggere la comunicazione GSM o TETRA tra il dispositivo mobile e l'infrastruttura di rete.
- Solo gli utenti che comunicano possono leggere i messaggi.
- Mira ad impedire le **intercettazioni**, incluse quelle effettuate dallo stesso fornitore del servizio o dell'infrastruttura di rete.
- Spesso implementato a livello di applicazione

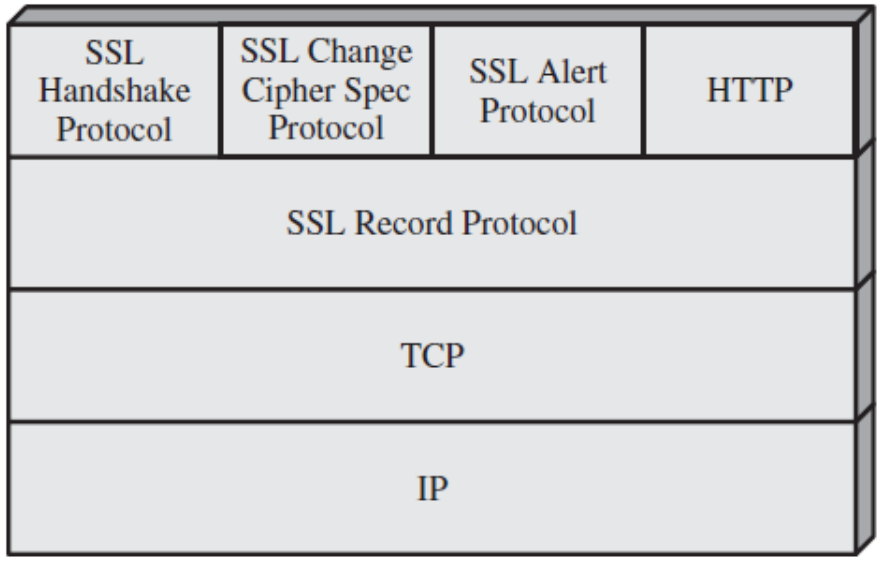
Sicurezza a livello Presentazione



Secure Socket Layer: storia

- **1994**: SSL introdotto da Netscape per servizi web sicuri (1.0 mai rilasciato per problemi di sicurezza).
- **1995**: SSL 2.0 rilasciato.
- **1996**: SSL 3.0 rilasciato (versione «stabile»).
- **1999**: TLS (**Transport Layer Security**) 1.0 rilasciato come standard IETF (RFC 2246): è un upgrade di SSL 3.0 ma non è interoperabile con SSL 3.0, sebbene le differenze tra i due siano limitate.
- **2006**: TLS 1.1 (RFC 4346) migliora TLS 1.0 (protezione contro attacchi a Cipher Block Chaining…).
- **2008**: TLS 1.2 (RFC 5246) rilasciato con diverse modifiche (aggiunte funzioni pseudo-random specifiche, aggiunte modalità AES, rimossi IDEA e DES ed altro).
- **2011**: IETF annuncia che SSL 2.0 è deprecato (RFC 6176) per problemi di sicurezza.
- **2015**: IETF annuncia che SSL 3.0 è deprecato (RFC 7568) e qualsiasi versione di TLS garantisce maggiore sicurezza di SSL.
- **2018**: TLS 1.3 (RFC 8446) rilasciato con modifiche rilevanti per semplificare il protocollo (rimosse SHA-1, MD5, RC4, DES e 3DES, handshake abbreviato, cifratura delle informazioni sul server, aggiunta della firma RSA-PSS…).

Protocolli SSL/TLS



- Due livelli protocollari:
  - **Record Protocol**: fornisce la base per servizi sicuri (compressione, cifratura).
  - **Handshake Protocol**, **Change Cipher Spec Protocol** e **Alert Protocol** svolgono funzioni di gestione di SSL/TLS appoggiandosi sul record protocol.

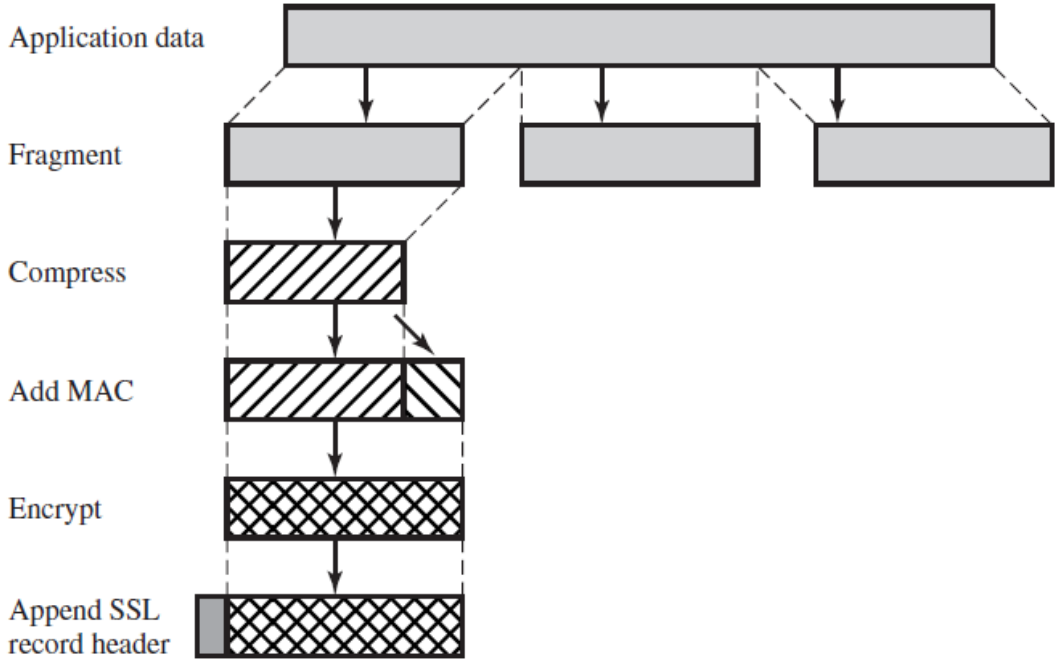
- Protocolli di livello 7 (ad es. HTTP, FTP…) poggiano su SSL/TLS per ottenere sicurezza (ad es. HTTP + SSL/TLS = HTTPS per navigazione web sicura).

Sessioni e connessioni SSL

- Una **connessione** SSL è intesa come un legame peer-to-peer (a livello trasporto), essa è temporanea ed associata ad una sessione
- Una **sessione** SSL è un'associazione tra un client ed un server.
- Le sessioni sono create dal Handshake Protocol e definiscono un set di parametri di sicurezza che possono essere condivisi tra connessioni multiple, evitando la necessità di negoziare i parametri di sicurezza per ciascuna connessione
- Tra una coppia di terminali (client e server) possono esistere connessioni multiple e contemporanee, solitamente nell'ambito di un'unica sessione

SSL Record Protocol

- Fornisce due servizi per le connessioni SSL:
  - **Confidenzialità**: cifratura dei dati sfruttando una chiave segreta definita dal Handshake Protocol
  - **Integrità**: controllo di integrità ed autenticità dei dati tramite un message authentication code (MAC) che sfrutta una chiave segreta definita dal Handshake Protocol

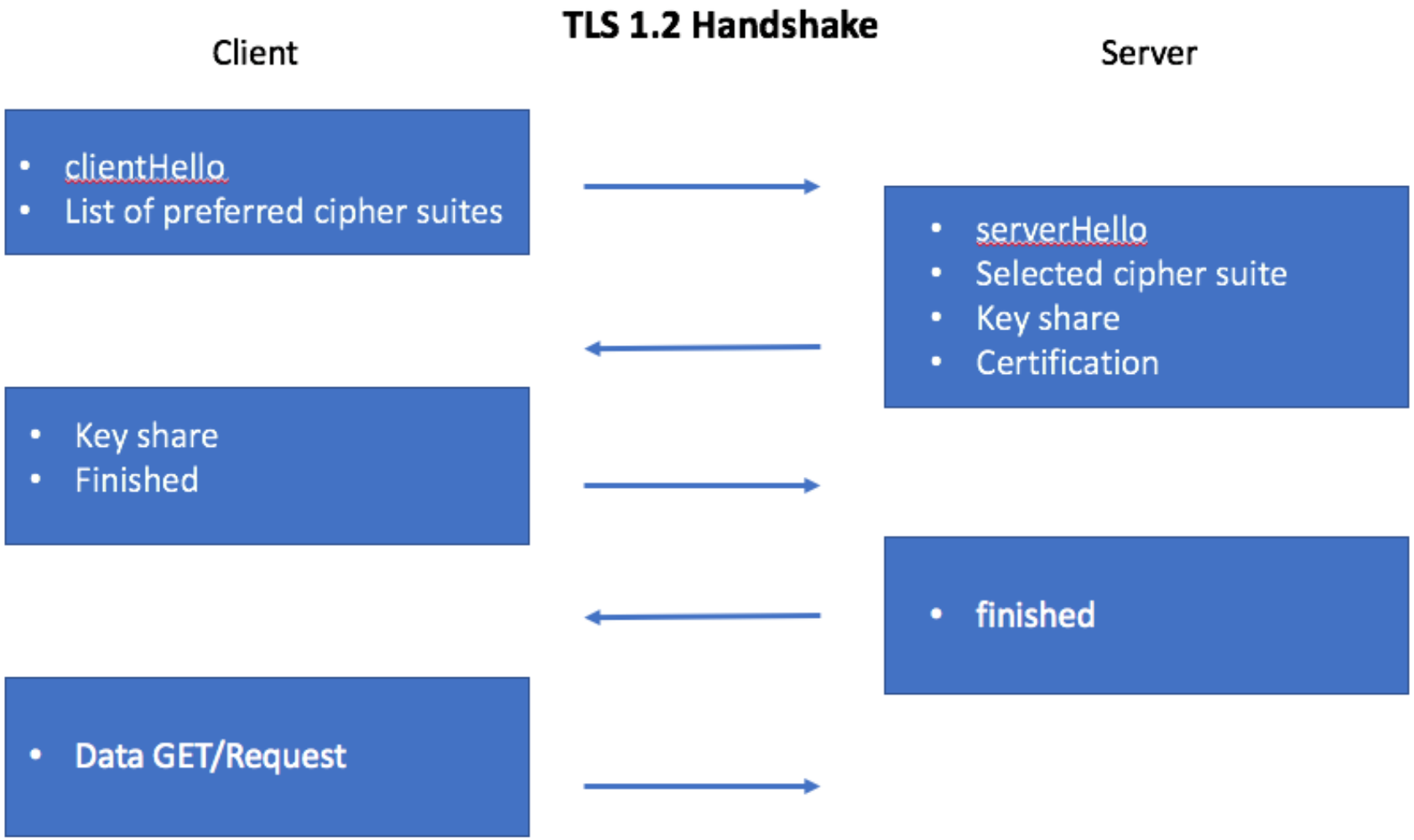


Transport Layer Security

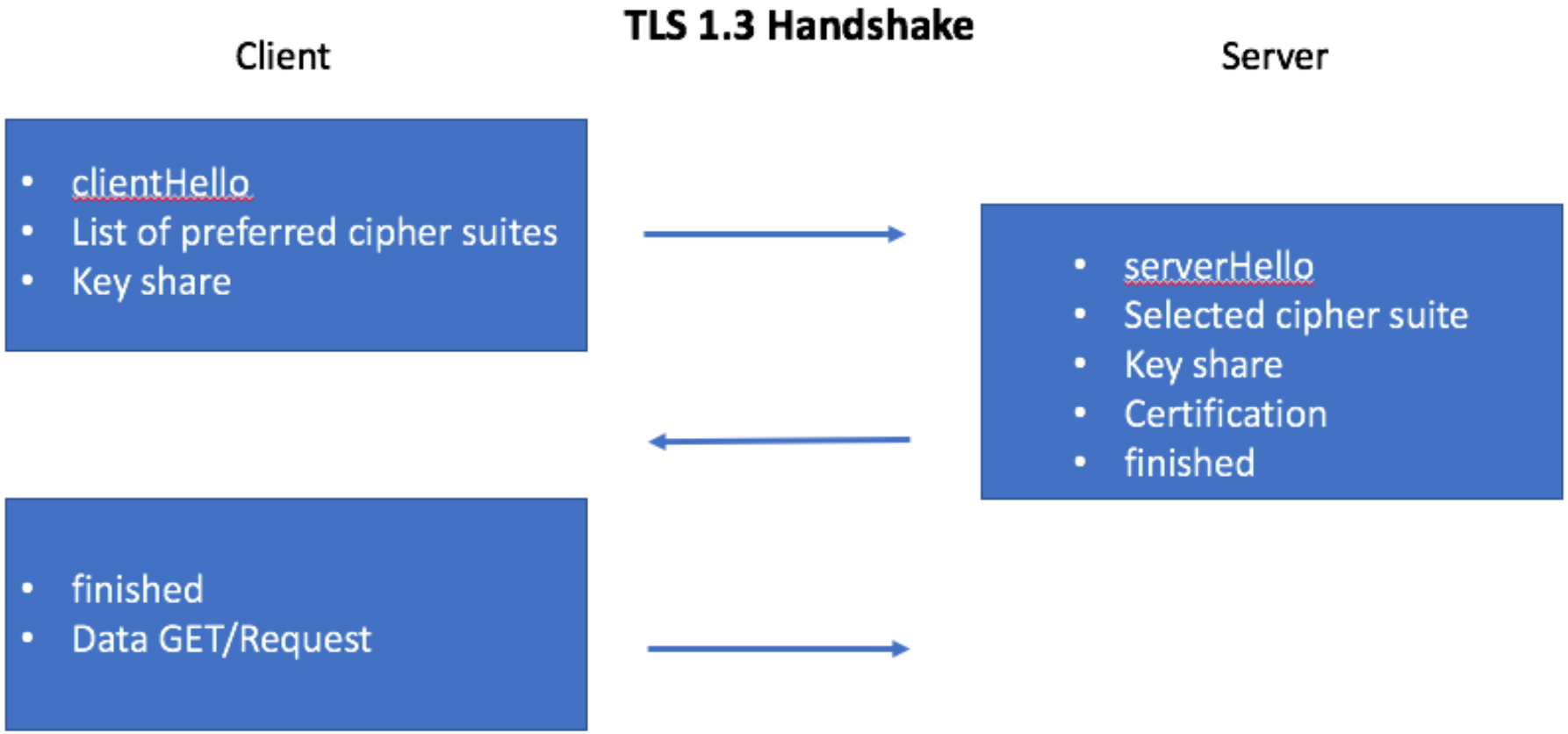
- TLS è un'iniziativa di standardizzazione IETF avente lo scopo di produrre una versione standard di SSL

- TLS include diverse migliorie rispetto ad SSL (ad esempio fa uso dell'algoritmo HMAC definito in RFC 2104 e di funzioni pseudorandom per espandere le chiavi segrete)

Handshake TLS 1.2



Handshake TLS 1.3



HTTPS (HTTP over SSL)

- Combinazione di HTTP e SSL per implementare comunicazioni sicure tra un web browser ed un web server
- Protocollo incluso in tutti i browser moderni (ma l'effettivo uso dipende dal supporto da parte del server)
- Quando è usato, l'url cambia da http://... a https://...
- HTTP normalmente usa la porta 80, mentre HTTPS la 443

HTTPS (HTTP over SSL)

- Quando HTTPS è usato, i seguenti elementi della comunicazione sono cifrati:
  - url delle pagine richieste dal client
  - contenuti delle pagine
  - contenuti dei moduli compilati dall'utente
  - cookies scambiati tra client e server
  - contenuti delle intestazioni HTTP
- HTTPS è documentato nella RFC 2818 (*HTTP Over TLS*)
- Non ci sono differenze fondamentali tra HTTP over SSL e HTTP over TLS (entrambi sono denominati HTTPS)

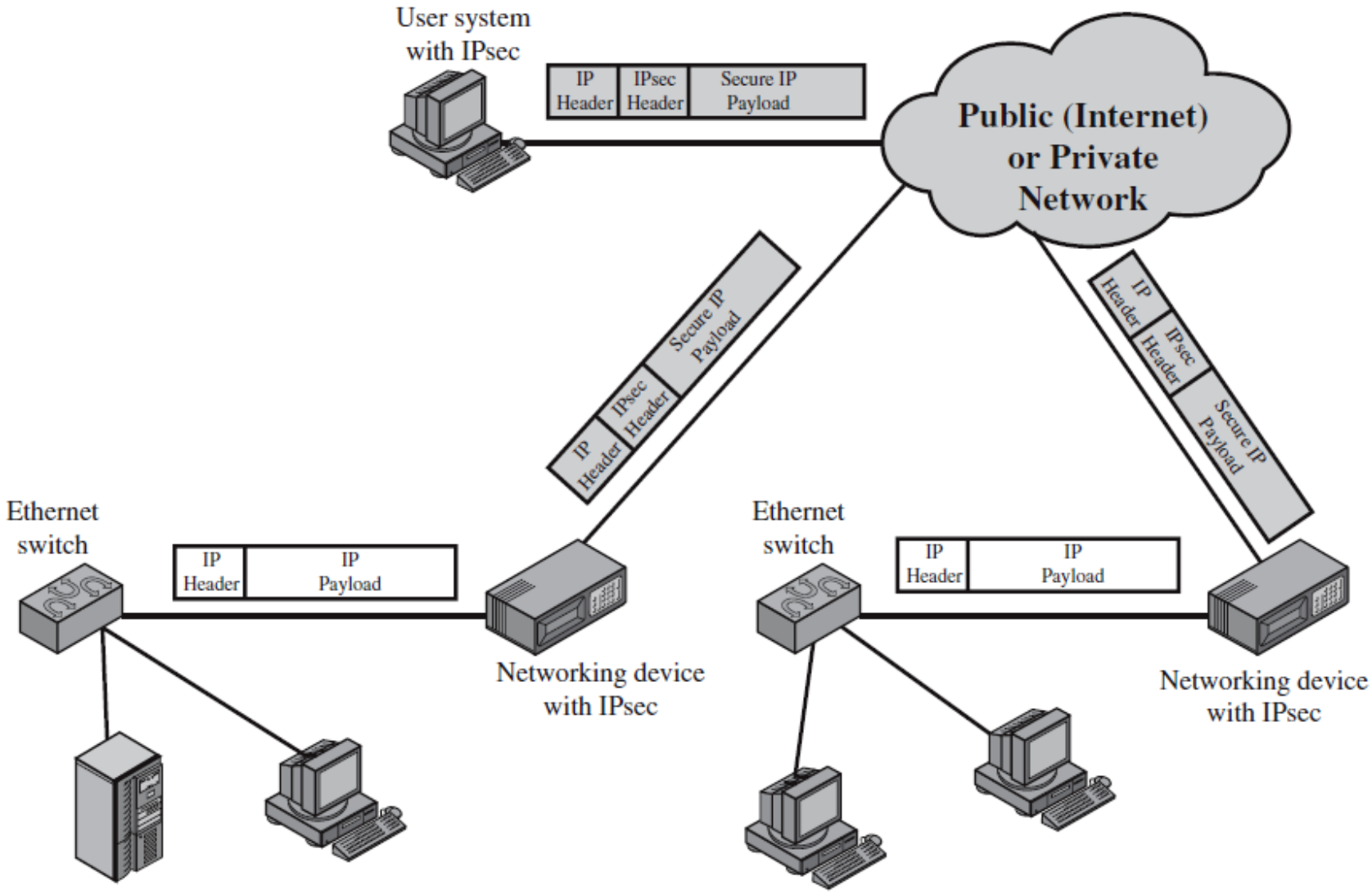
Sicurezza a livello rete: IP security

- Nel 1994, la Internet Architecture Board (IAB) emise un rapporto intitolato “Security in the Internet Architecture” (RFC 1636)
- Esso evidenziava la necessità di rendere sicure le infrastrutture di rete rispetto all'uso ed al controllo non autorizzato del traffico di rete
- Si esprimeva la necessità di rendere sicuro il traffico tra due end-user utilizzando tecniche di autenticazione e cifratura
- Tali tecniche sono state incluse nella versione più recente del protocollo IP (IPv6)
- Esse sono state progettate per essere usate anche con la versione precedente del protocollo IP (IPv4)

Esempi di uso di IPsec

- Realizzazione di una rete privata virtuale (**VPN**) tramite Internet o altra rete pubblica, evitando la necessità di reti dedicate
- Accesso remoto sicuro tramite Internet ad una rete privata
- Comunicazioni sicure tra reti private (extranet e intranet) di diversi proprietari
- Aumento della sicurezza nelle applicazioni di commercio elettronico e altri servizi online

Uso tipico di IPsec (VPN)



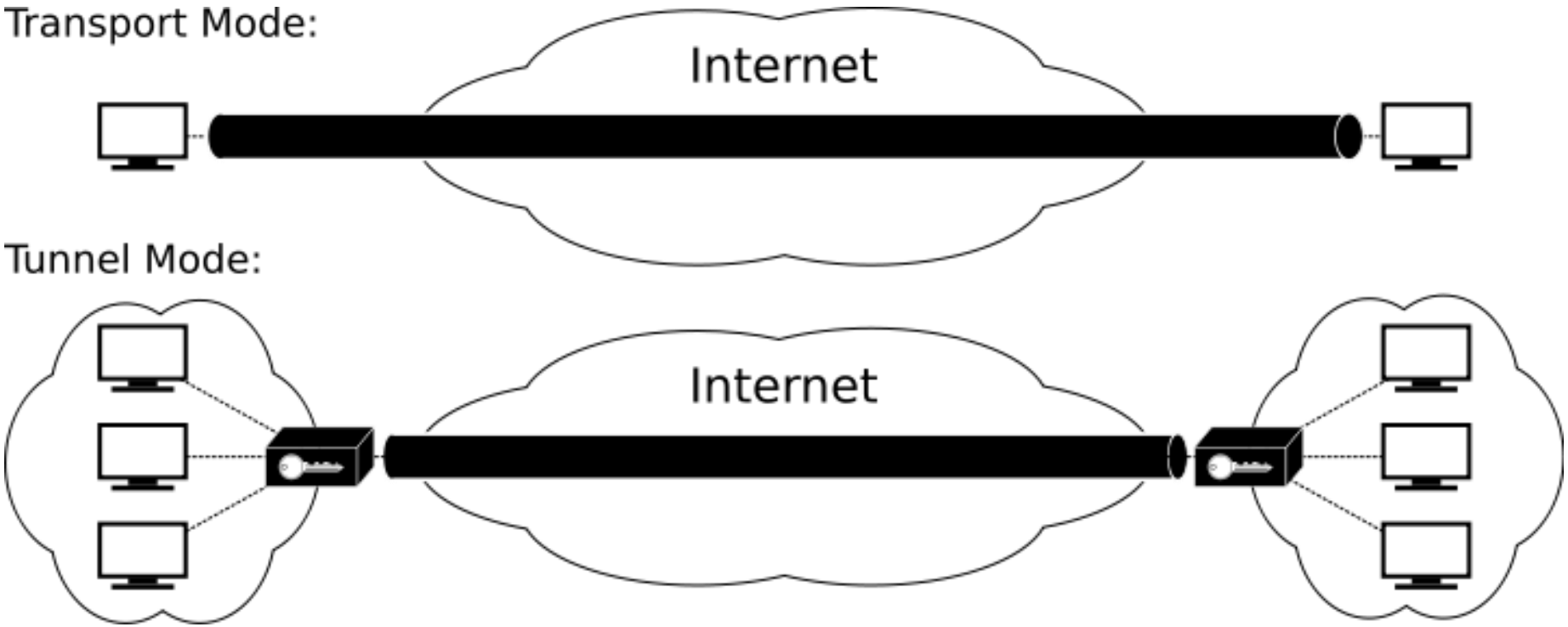
Vantaggi di IPsec

- IPsec fornisce sicurezza a livello IP, quindi coinvolge tutto il traffico di rete, senza bisogno di overhead a livelli più alti
- I dispositivi IPsec (router e firewall) perimetrali sono difficili da bypassare
- Lavorando a livello IP, IPsec è trasparente per le applicazioni ed i software di livello superiore
- Non serve la conoscenza e gestione di tecniche di sicurezza da parte dell'utente
- Si può utilizzare IPsec anche per singoli utenti o gruppi di utenti all'interno di una rete

Tecniche di IPsec

- IPsec utilizza tecniche crittografiche per tre scopi principali:
  - Autenticazione
  - Confidenzialità
  - Gestione delle chiavi
- Le specifiche di IPsec sono contenute in un gran numero di documenti IETF
- Essi definiscono le tecniche tramite cui IPsec implementa servizi sicuri a livello IP consentendo ai sistemi di:
  - Scegliere i protocolli di sicurezza
  - Scegliere gli algoritmi da usare
  - Ottenere le chiavi necessarie

Modalità operative IPsec



Virtual Private Network (VPN)

- Una VPN è una rete privata virtuale interconnessa tramite una rete pubblica (Internet) usando cifratura ed appositi protocolli per ottenere sicurezza
- Usare una rete pubblica (Internet) per connettere sedi dislocate consente notevoli risparmi rispetto all'uso di connessioni dedicate
- Inoltre questo permette l'accesso alla rete privata da qualsiasi luogo raggiunto dalla rete pubblica (Internet)
- Usando cifratura ed autenticazione ai livelli di rete più bassi (ad esempio tramite IPsec) si può ripristinare il carattere privato della rete
- E' sufficiente implementare tali tecniche nei dispositivi perimetrali (router o firewall)

Sicurezza delle Reti Wireless

IEEE 802.11 e SICUREZZA

- Le reti radio sono intrinsecamente meno sicure di quelle cablate
- Lo standard IEEE 802.11 prevedeva come primo protocollo di sicurezza il **WEP** (Wired Equivalent Privacy)
- Il WEP è stato definitivamente abbandonato nel 2001 proprio perché si è dimostrato che non era in grado di garantire la sicurezza delle reti
- Il protocollo che sostituisce e supera WEP è **802.11i**

OPERAZIONI del WEP

- Il frame da trasmettere viene dapprima sottoposto al controllo di integrità tramite un algoritmo CRC a 32 bit che produce come risultato 4 byte denominati ICV (Integrity Check Value)
- Il ICV viene appeso in coda al corpo del frame ed entrambi vengono cifrati
- La chiave per la cifratura è composta da una chiave condivisa di 40 bit a cui viene aggiunto un Initialization Vector di 24 bit
- I 64 bit che ne risultano servono come chiave per l'algoritmo RC4 che genera un keystream lungo esattamente quanto il frame aumentato del ICV
- Il keystream generato va in XOR con il corpo del frame aumentato del ICV per ottenerne la versione cifrata

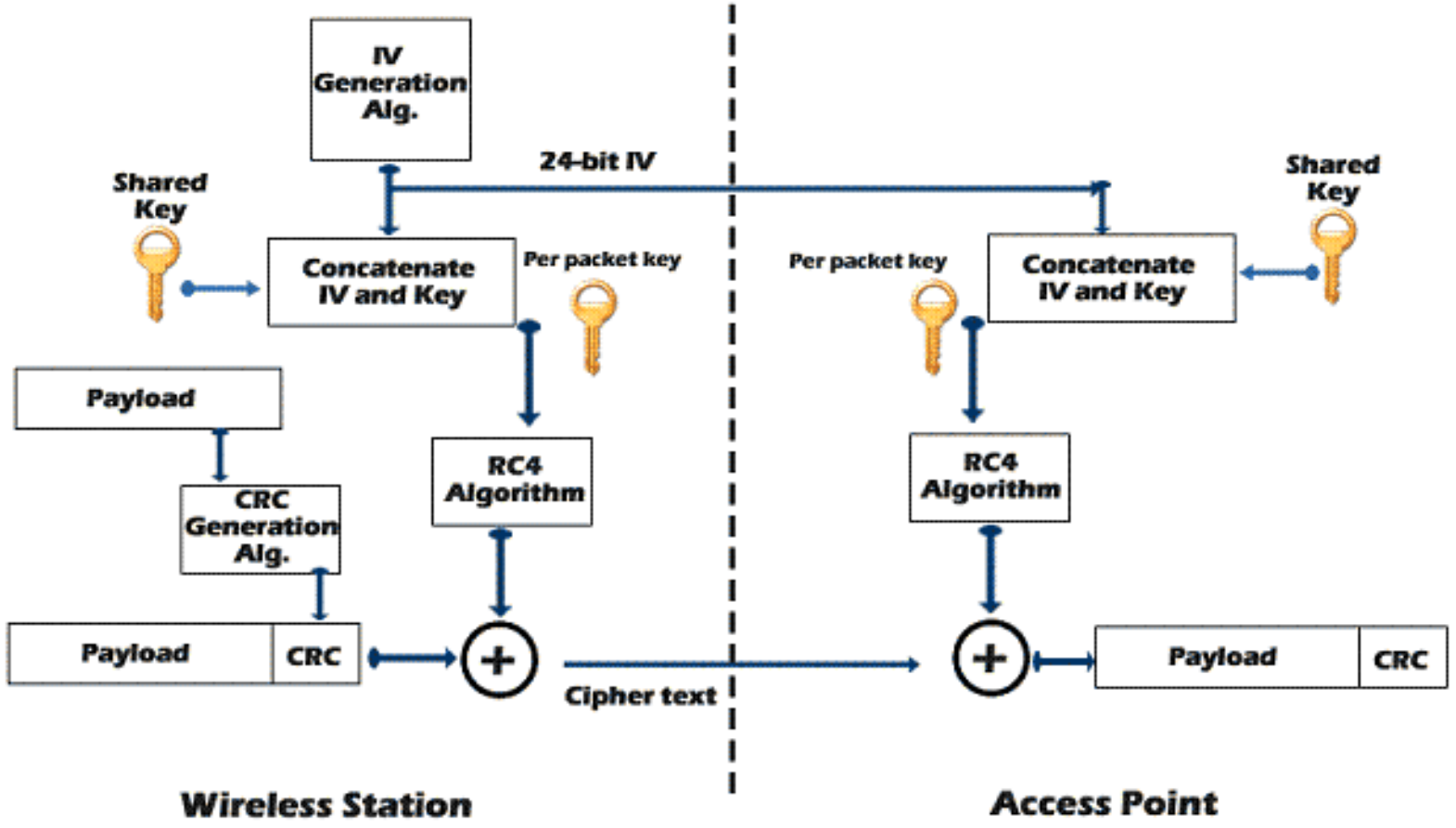
RC4

- Cifrario a flusso con complessità minima
- Sviluppato da Ron Rivest nel 1987
- È particolarmente debole quando l'inizio del keystream non viene scartato, oppure quando si usano chiavi correlate
- Il suo uso all'interno di TLS è stato proibito da IETF nel 2015 con la RFC 7465
- Anche Microsoft e Mozilla ne sconsigliano l'uso

RC4 – principio

- RC4 si basa su un circuito che genera un keystream pseudocasuale che viene usato per cifrare e decifrare in stile one-time-pad
- RC4 ha uno stato interno costituito da due elementi:
  - Una permutazione dei 256 possibili byte (*S*)
  - Due indici a 8 bit (*i* e *j*)
- La permutazione è inizializzata a partire da una chiave segreta di lunghezza variabile (tra 40 e 2048 bit) tramite il «key scheduling algorithm» (KSA)
- Dopodiché il «pseudo-random generation algorithm» (PRGA) genera il flusso di bit che costituisce il keystream

Incapsulamento WEP



WEP

- L’Initialization Vector (IV) del WEP è lungo 24 bit e viene trasmesso in chiaro
- Il segreto (condiviso) si limita soltanto ai primi 40 bit
- La cifratura del WEP prende il nome di 40 bit WEP, o 40+24 bit WEP o 64 bit WEP
- La versione a 128 bit non è stata in un primo momento standardizzata a causa del divieto imposto dalle leggi americane
- Essa è arrivata in un secondo momento sottoforma di diverse varianti (104 bit WEP, 128 bit WEP o 152 bit WEP) con conseguenti problemi di interoperabilità

Limiti del WEP

- Il riutilizzo del keystream è un forte punto di debolezza (due frame cifrati con lo stesso keystream hanno XOR identico alle loro versioni non cifrate) e l'unico modo per evitarlo è variare il IV, che però è trasmesso in chiaro.

- Il CRC è una tecnica di "integrity check" per niente sicura dal punto di vista crittografico: è perfettamente noto come la variazione di 1 bit influisce sul risultato del CRC. Le funzioni hash, invece, darebbero risultati più difficili da prevedere.

- La parte segreta della chiave (40 bit) deve essere condivisa tra gli interlocutori ma non è prevista una tecnica di condivisione sicura.

- Essa infatti può essere una delle 4 chiavi di default, oppure può essere impostata manualmente dall'amministratore di sistema. Non esiste una tecnica di distribuzione sicura delle chiavi.

Attacco al WEP (2001)

Weaknesses in the Key Scheduling Algorithm of RC4

Scott Fluhrer[1], Itsik Mantin[2], and Adi Shamir[2]

[1] Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134
sfluhrer@cisco.com
[2] Computer Science department, The Weizmann Institute, Rehovot 76100, Israel.
{itsik,shamir}@wisdom.weizmann.ac.il

Abstract. In this paper we present several weaknesses in the key scheduling algorithm of RC4, and describe their cryptanalytic significance. We identify a large number of weak keys, in which knowledge of a small number of key bits suffices to determine many state and output bits with non-negligible probability. We use these weak keys to construct new distinguishers for RC4, and to mount related key attacks with practical complexities. Finally, we show that RC4 is completely insecure in a common mode of operation which is used in the widely deployed Wired Equivalent Privacy protocol (WEP, which is part of the 802.11 standard), in which a fixed secret key is concatenated with known IV modifiers in order to encrypt different messages. Our new passive ciphertext-only attack on this mode can recover an arbitrarily long key in a negligible amount of time which grows only linearly with its size, both for 24 and 128 bit IV modifiers.

Crollo del WEP

- Adam Stubblefield, John Ioannidis ed Avi Rubin, nello stesso mese, portarono a termine con successo un attacco ad una rete wireless protetta con WEP. L'esperimento portò alla determinazione in breve tempo di tutti i 40 bit della chiave segreta.

- Entro la fine del mese Jeremy Bruestle e Blake Hegerle avevano già rilasciato **AirSnort**, il primo applicativo open source per il recupero delle chiavi in una rete wireless.

- A seguito di questi eventi, IEEE istituì un gruppo di lavoro che aveva il compito di elaborare un nuovo standard per la sicurezza delle reti wireless.

Il dopo-WEP

- La **Wi-Fi Alliance** è una associazione di produttori che certifica col proprio logo dispositivi WLAN
- Fondata nel **1999** con lo scopo di certificare l'interoperabilità dei dispositivi IEEE 802.11
- Un dispositivo conforme allo standard IEEE 802.11 può (ma non deve necessariamente) essere certificato Wi-Fi
- Dopo il crollo del WEP, la Wi-Fi alliance ha sviluppato tre nuove soluzioni per la sicurezza nelle WLAN:
  - Wi-Fi Protected Access (**WPA**)
  - Wi-Fi Protected Access 2 (**WPA2**)
  - Wi-Fi Protected Access 3 (**WPA3**)

WPA

- Wi-Fi Protected Access (**WPA**):
  - Disponibile dal **2003**, corrisponde a **IEEE 802.11i "draft"**
  - Richiede modifiche contenute rispetto a WEP
  - Introduce Temporal Key Integrity Protocol (TKIP) per avere una diversa chiave di 128 bit per ciascun pacchetto
  - Sostituisce CRC con un message authentication code (MAC)

- Wi-Fi Protected Access 2 (**WPA2**):
  - Disponibile dal **2004**, corrisponde a **IEEE 802.11i-2004**
  - Sostituisce RC4 con AES usato in modalità **CCM** (counter mode with cipher block chaining message authentication code (**CBC-MAC**))
  - Combina la modalità **CTR** per la confidenzialità con il **CBC-MAC** per l'autenticazione dei messaggi

Key Reinstallation Attack

- 2017: attacco di successo contro WPA2

WPA 3

- Annunciato a Gennaio 2018
- Usa cifratura a 128 bit in modalità WPA3-Personal ed a 192 bit in modalità WPA3-Enterprise
- Sostituisce il 4-way handshake con il protocollo di handshake chiamato «Dragonfly», che implementa “Simultaneous Authentication of Equals” (IEEE 802.11-2016)
- Garantisce forward secrecy