La Sicurezza delle Applicazioni Internet richiami delle tecnologie di base

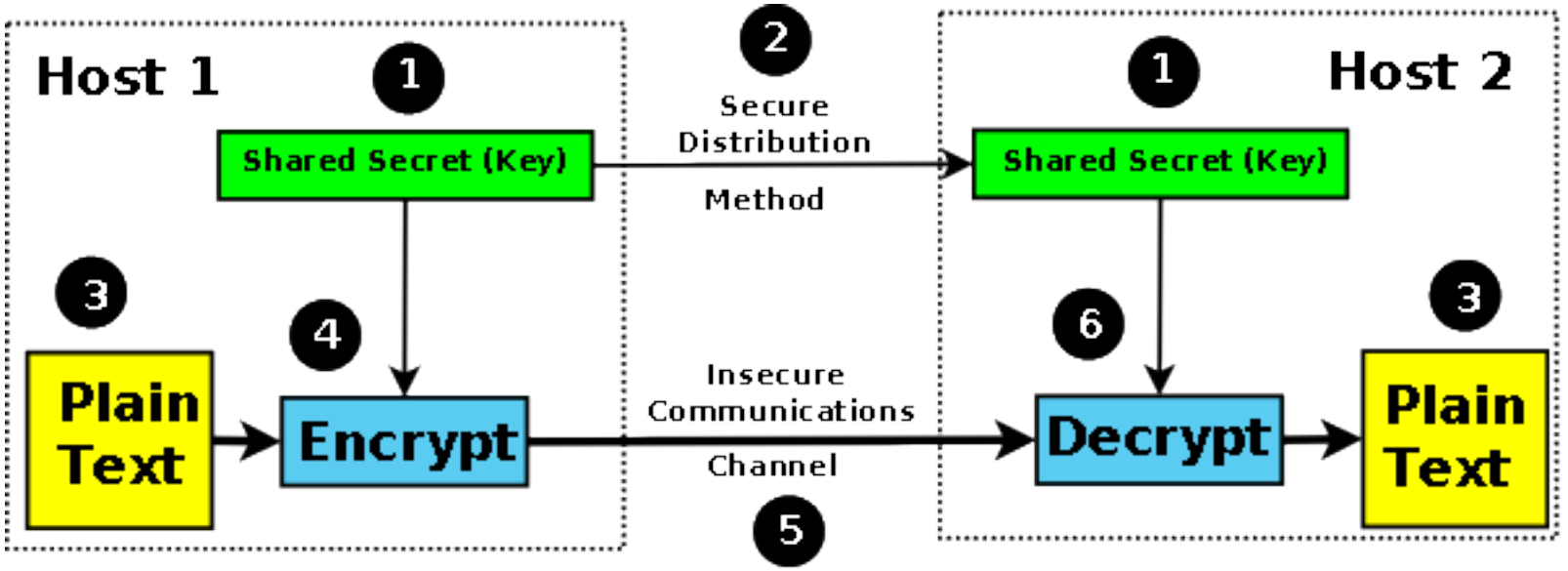
Tito Flagella
Laboratorio Applicazioni Internet - Università di Pisa

La cifratura simmetrica

La chiave utilizzata per cifrare è la stessa usata per decrifrare

- $M = D_k(E_k(M))$
- k è la chiave (unica)

La cifratura simmetrica



La cifratura Asimmetrica

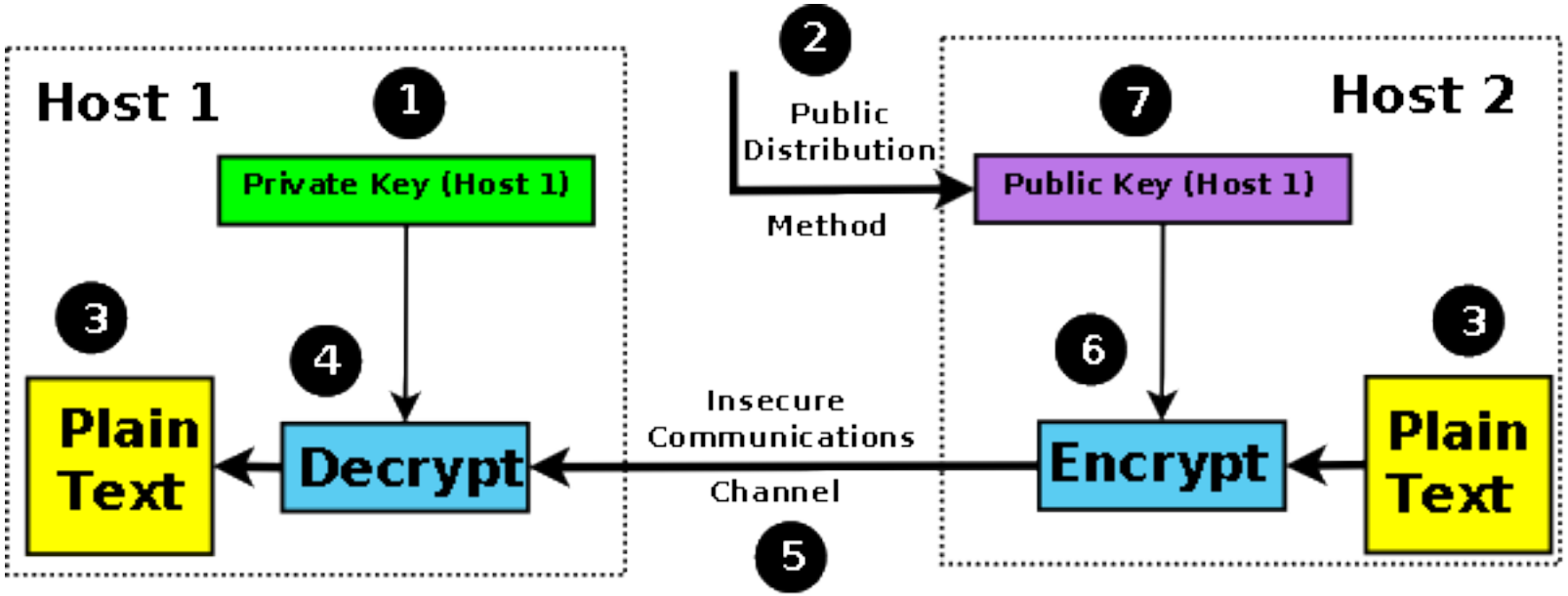
Si utilizzano due chiavi diverse, una per cifrare ed una per decifrare (la funzione in generale è la stessa)

$$M = D_{k1}(E_{k2}(M))$$

k1 e k2 sono generate in coppia

Conoscendo una chiave non è possibile dedurre l'altra

La cifratura Asimmetrica



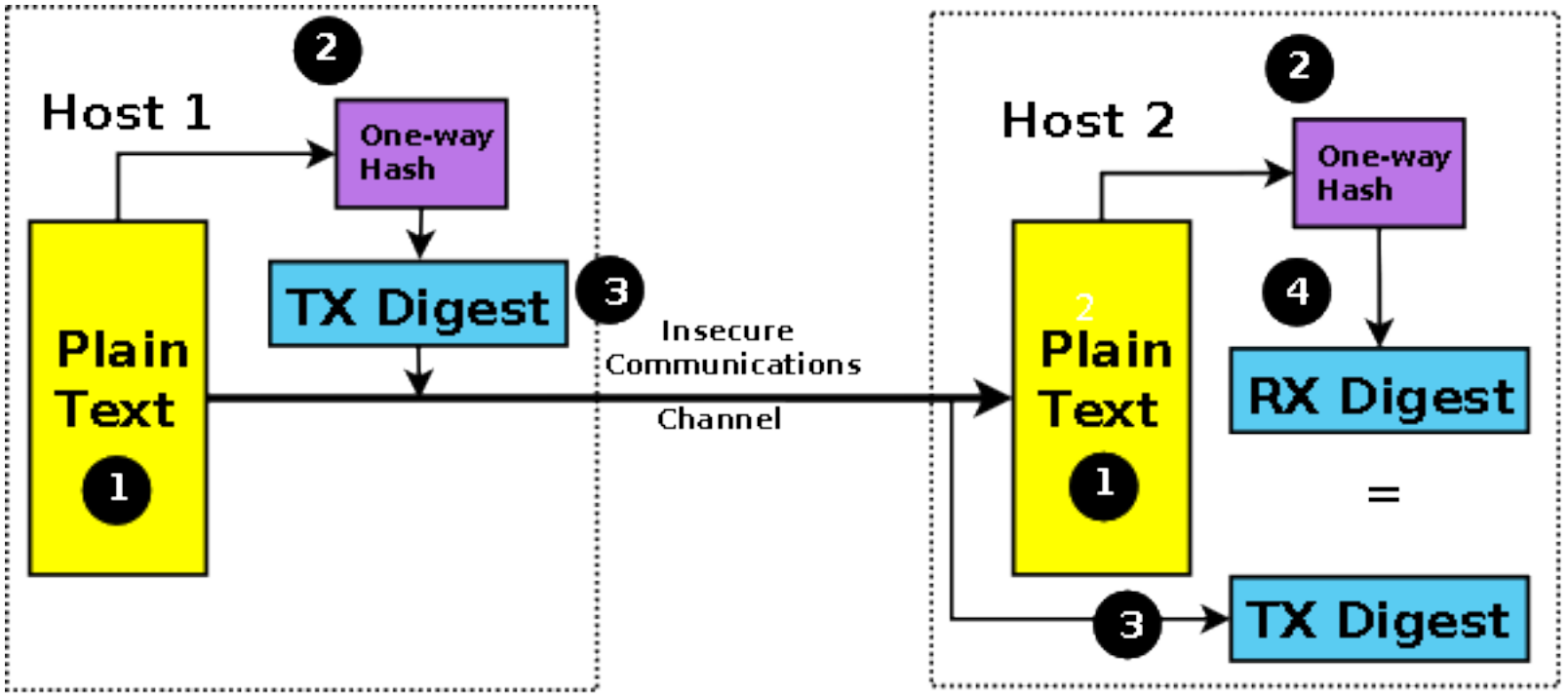
Le funzioni Hash

Dato un testo di lunghezza arbitraria restituiscono un *digest* breve e di lunghezza fissa

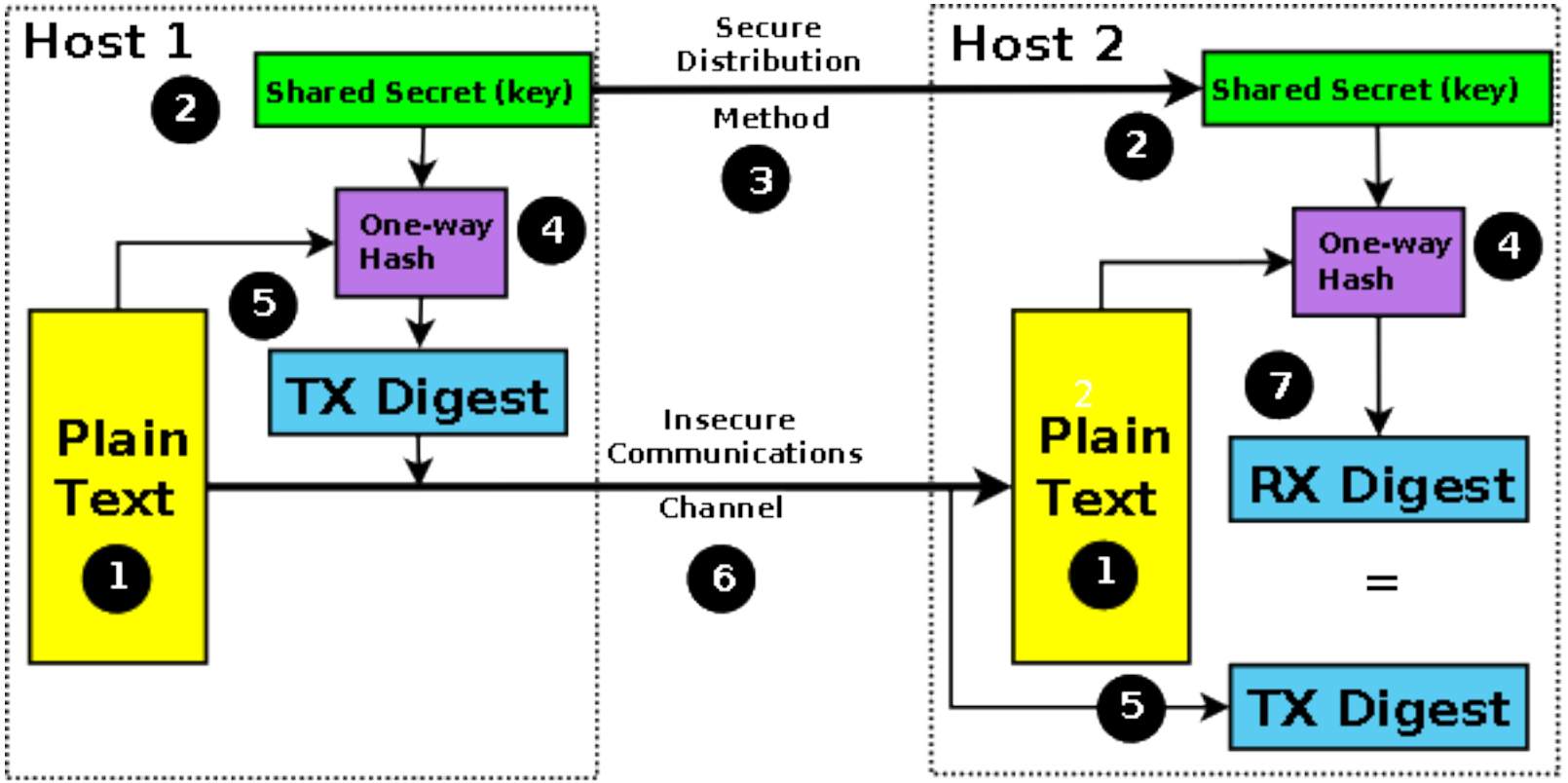
Dato un *digest,* non è possibile risalire al testo originale

Dato un *digest* D, non è possibile costruire un testo M che generi come *digest* D

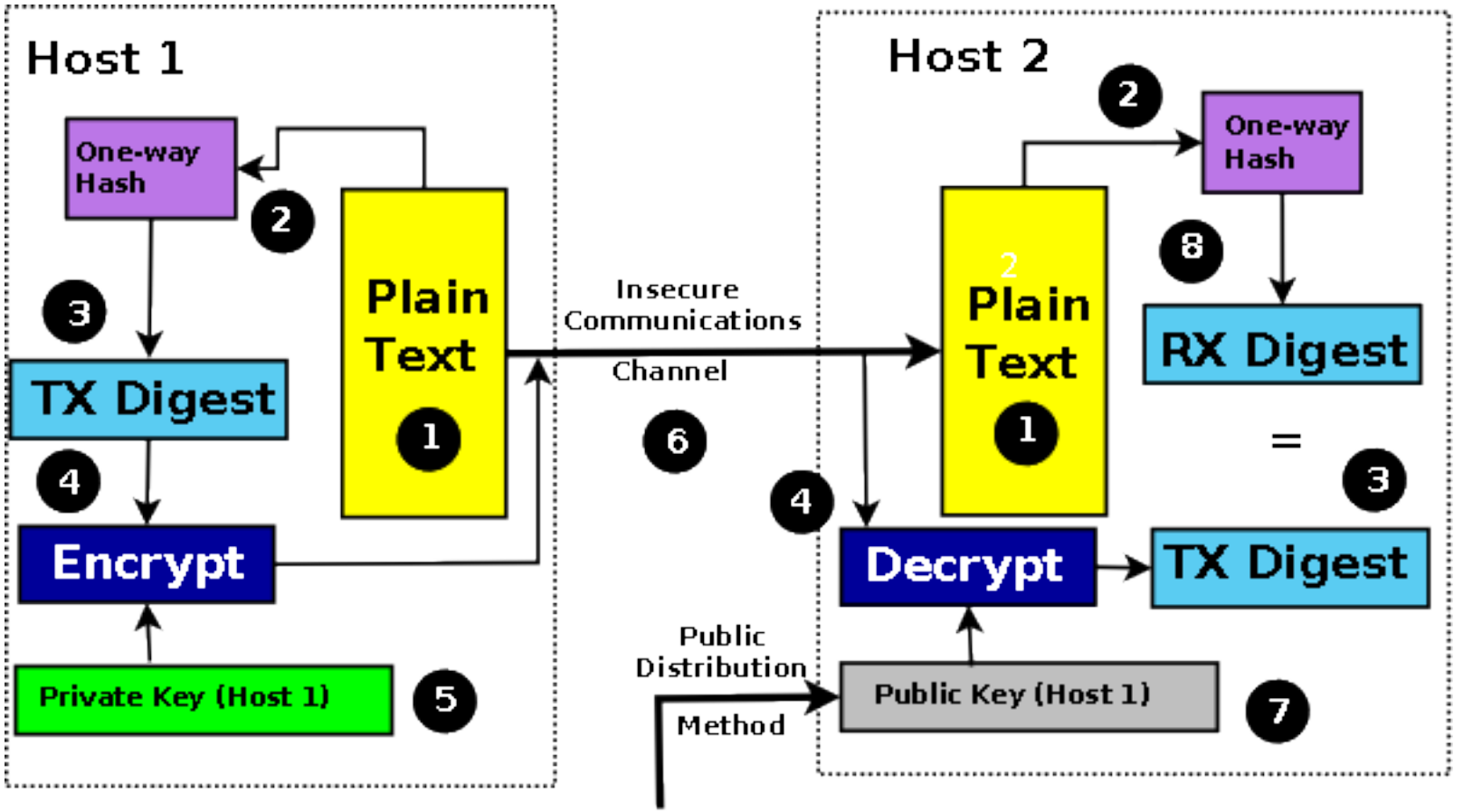
Le funzioni Hash



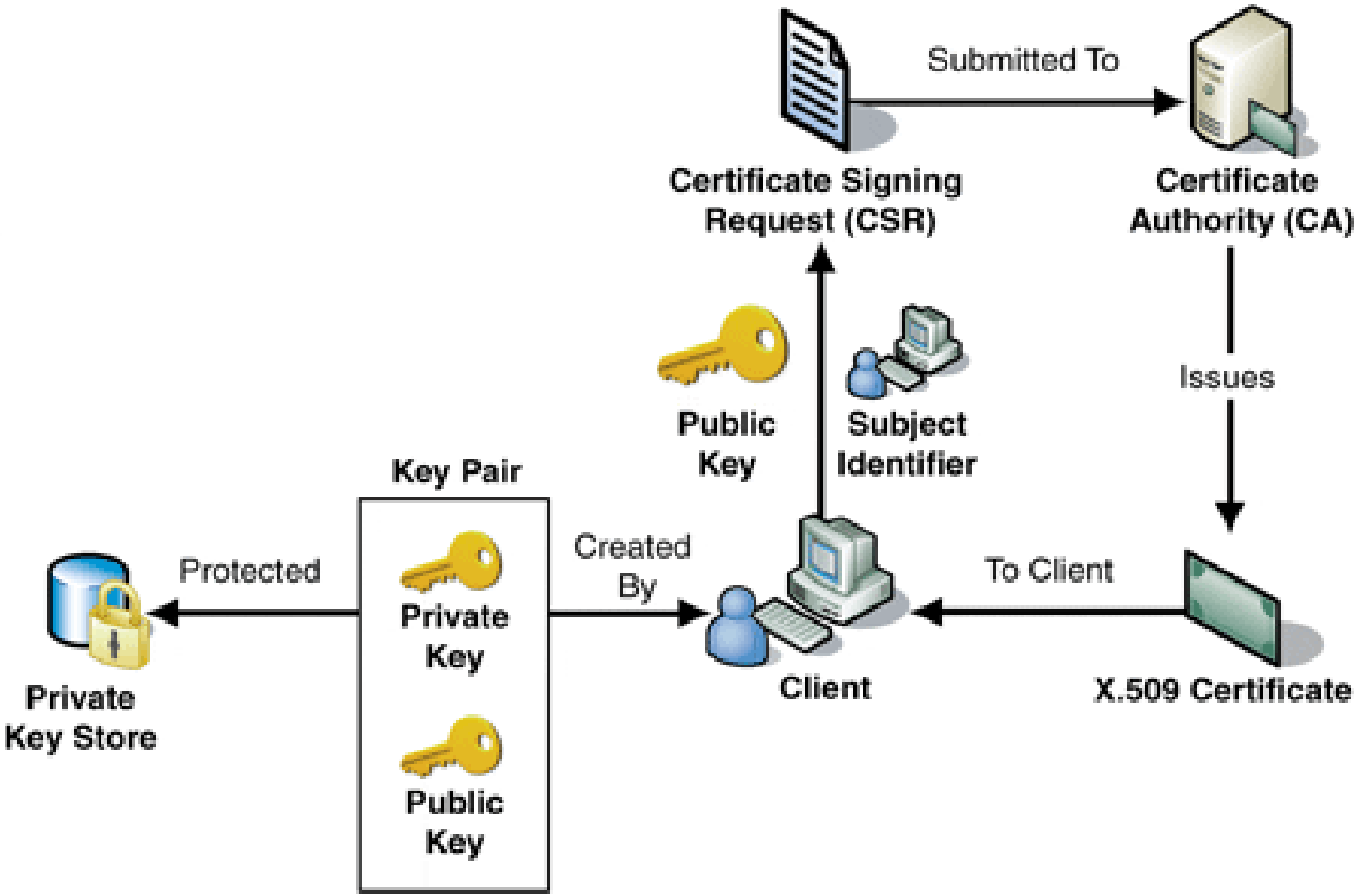
Le funzioni Hash sicure



Firma tramite Hash e Cifratura Asimmetrica



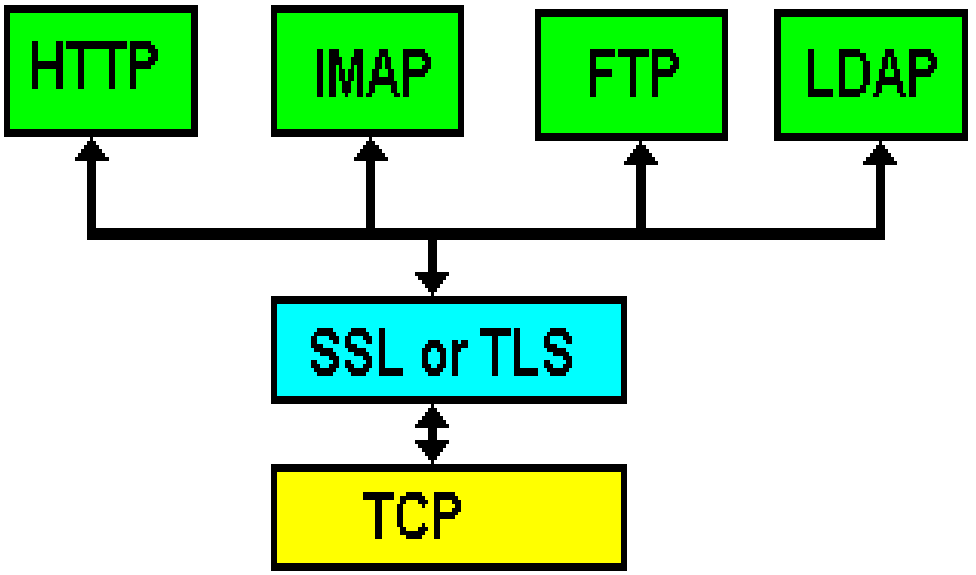
I Certificati X.509



Il protocollo SSL/TLS

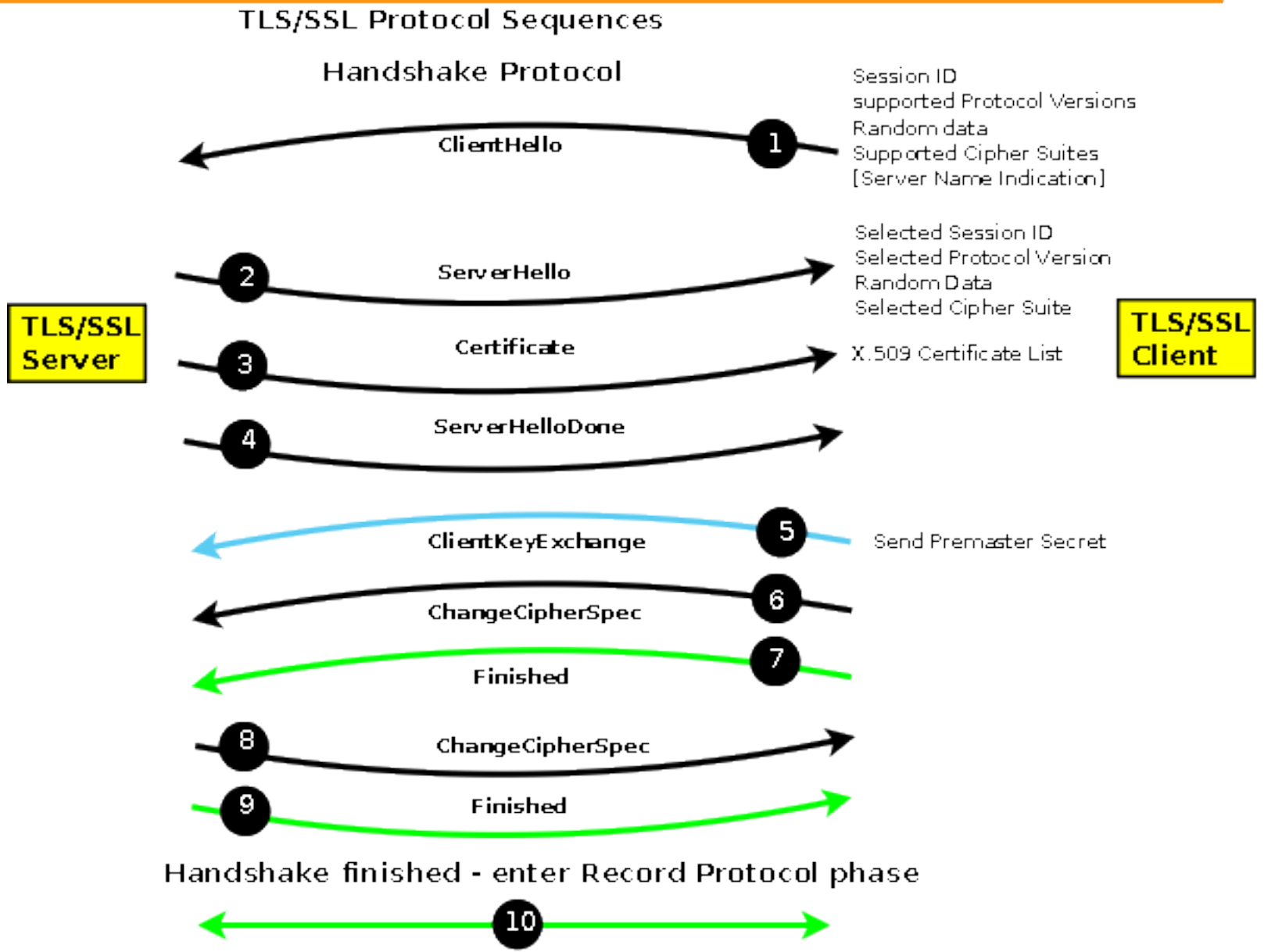
- Secure Sockets Layer (SSL) is a Netscape protocol originally created in 1992
- The IETF standardized Transport Layer Security (TLS) Version 1, a minor variation of SSL, in RFC 2246, Version 1.1 in RFC 4346 and Version 1.2 in RFC 5246.

Il protocollo SSL/TLS



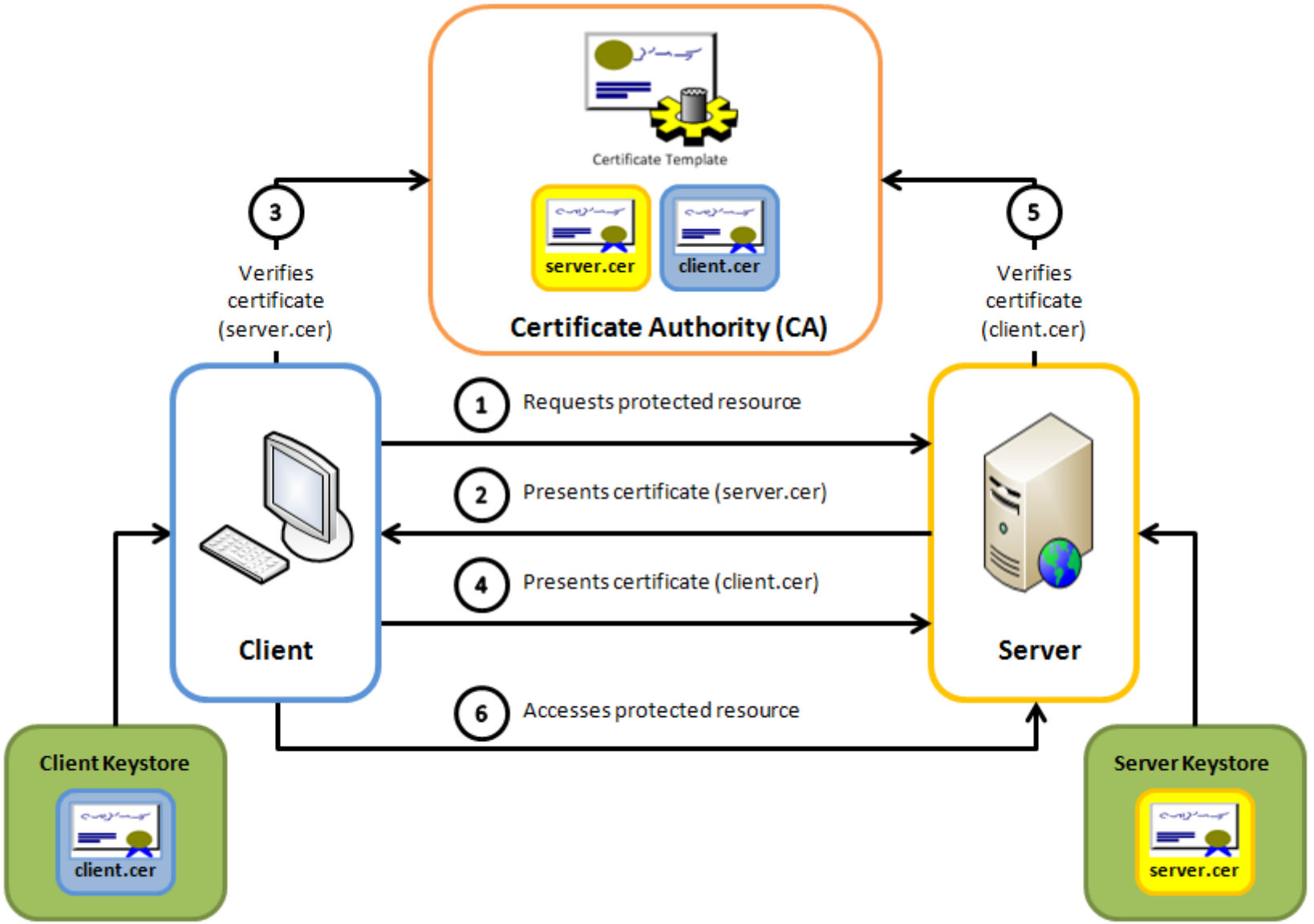
HTTP + SSL/TLS + TCP = HTTPS

Il Protocollo SSL/TLS



Nero: in chiaro, blue cifrati con chiave pubblica, verde cifrati con chiave simmetrica scambiata durante l'hadshake.

HTTPS Mutual Authentication



Mutual SSL authentication / Certificate based mutual authentication

Gestione Certificati in Java

Gestione dei KeyStore con keytool

$ keytool -genkey -alias KeyForPaul -keystore JohnsPrivateKey.store

$ keytool -export -alias KeyForPaul -file certfile.cer -keystore JohnsPrivateKey.store

$ keytool -import -alias PublicKeyFromJohn -file certfile.cer -keystore MyPublicKey.store

Uso dei keystore dalla JVM

$ java -Djavax.net.ssl.keyStore=/mypath/ks-name

-Djavax.net.ssl.trustStore=/mypath/ts-name

Configurazione dei keystore negli Application Server

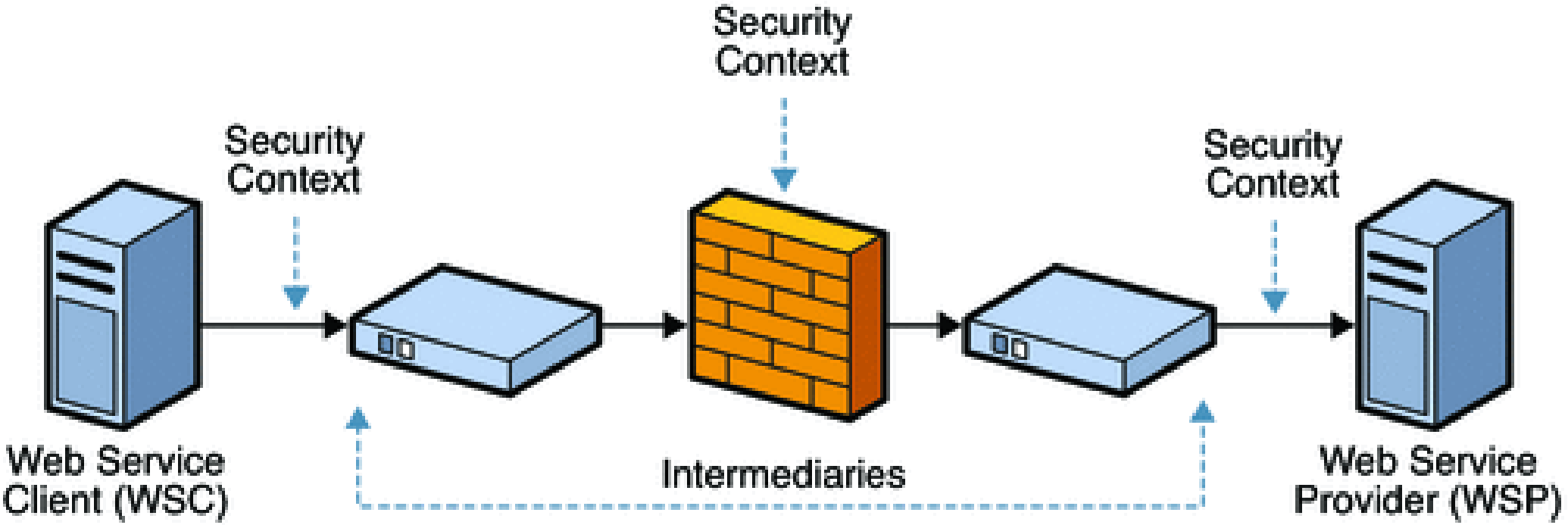
<Connector port="8443" protocol="HTTP/1.1" SSLEnabled="true"

maxThreads="150" scheme="https" secure="true"

clientAuth="false" sslProtocol="TLS"

keystoreFile="/home/u/user/keystore/shopping.jks" keystorePass="******"/>

Sicurezza Point to Point o End to End



- HTTPS garantisce la sicurezza a livello Trasporto (Point to Point)
- XML-Encryption o WS-Security garantiscono la sicurezza a livello Messaggio (End to End)