Aalto University

Network Security:
RSA handshake (TLS 1.2 and earlier)

Tuomas Aura, Aalto University

CS-E4300 Network security

Public-key encryption of session key

- Public-key encryption of the session key:

1. A → B: A, B, PK_A
2. B → A: A, B, $E_A(SK)$

PK_A = A's public key

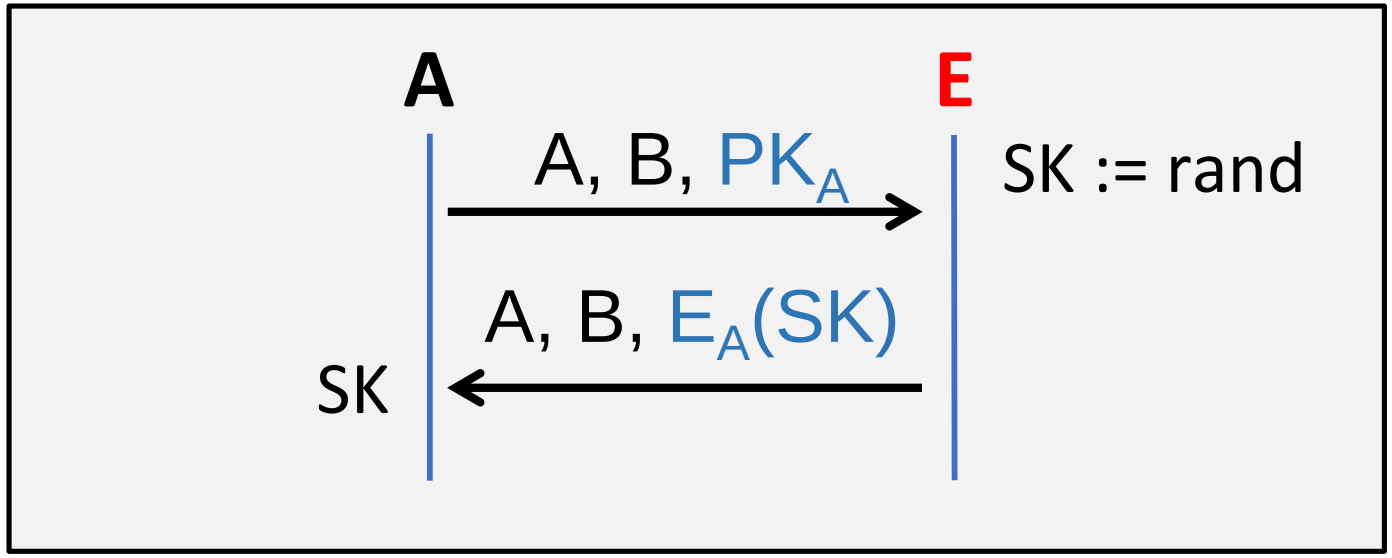
SK = session key

$E_A(\ldots)$ = encryption with A's public key

Note:
The protocol is not secure like this. Please read further.

Impersonation and MitM attacks

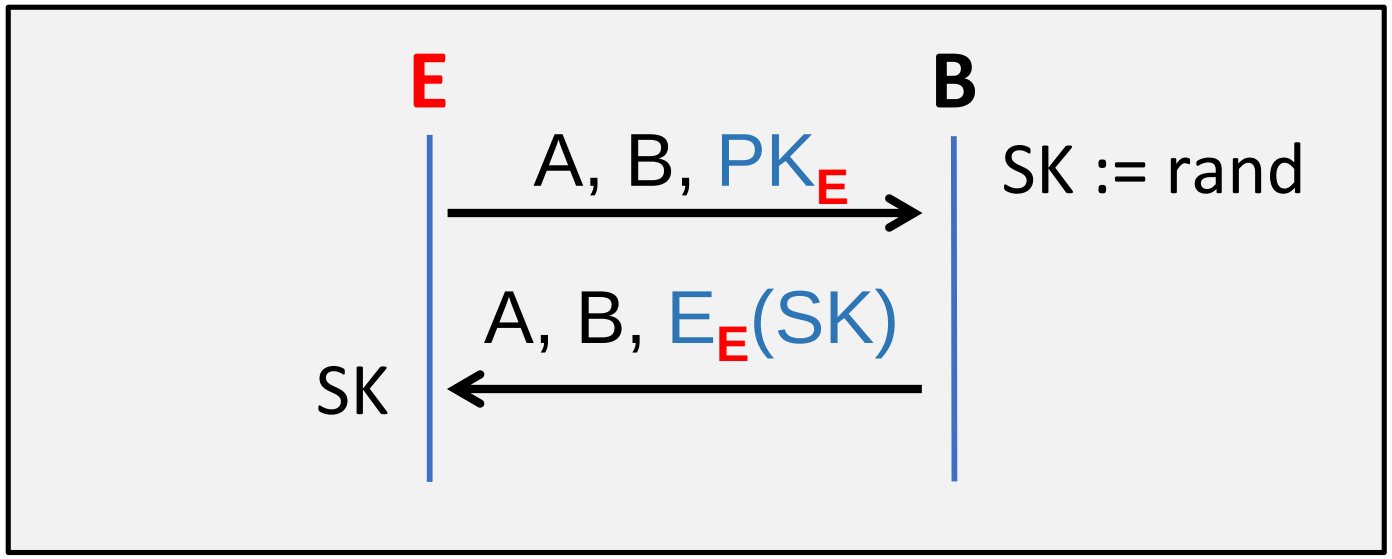
- Unauthenticated key exchange with public-key encryption suffers from the same impersonation and man-in-the-middle attacks as DH



- A has a shared secret, but with whom?

Impersonation and MitM attacks

- Impersonating A is similarly possible because B does not know whether the public key really belongs to A:



- B has a shared secret, but with whom?

Authenticated key exchange

- Authenticated key exchange with public-key encryption:

1. A → B: A,B, N_A, $Cert_A$
2. B → A: A,B, N_B, $E_A(KM)$, S_B("Msg2",A,B,N_A,N_B,$E_A(KM)$), $Cert_B$, MAC_{SK}(A, B, "Responder done.")
3. A → B: A,B, MAC_{SK}(A, B, "Initiator done.")

SK = h(N_A, N_B, KM)

Somewhat realistic protocol (compare with TLS_RSA)

Why nonces and not SK = KM?

KM = random key material (random bits) generated by B

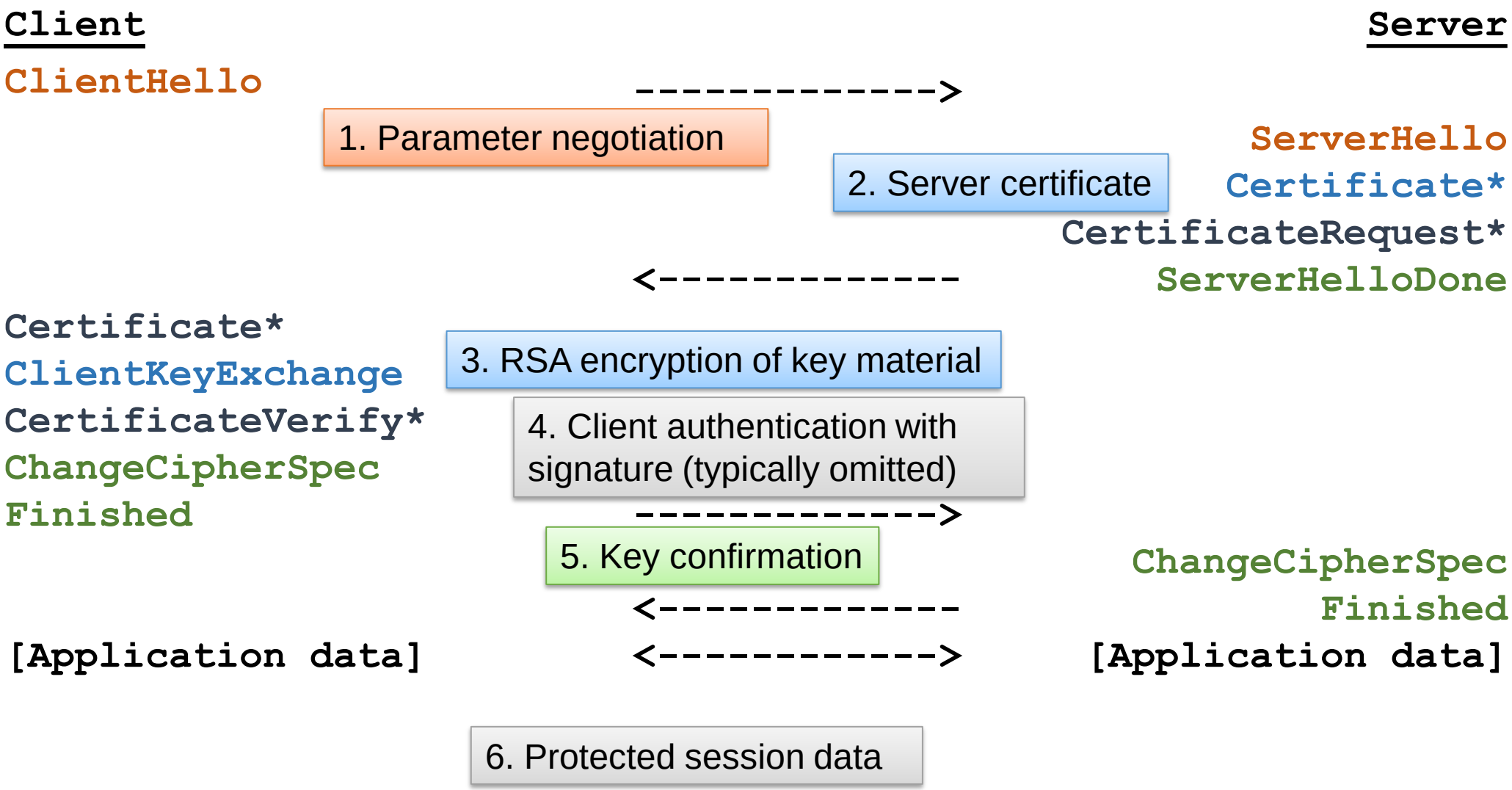
$Cert_A$, $E_A(...)$ = A's certificate and public-key encryption to A

$Cert_B$, $S_B(...)$ = B's certificate and signature

$MAC_{SK}(...)$ = MAC with the session key

To match with the previous slide:
A = Server, B = Client

TLS_RSA handshake



TLS_RSA handshake

1. C → S: Versions, N_C , SessionId, CipherSuites
2. S → C: Version, N_S , SessionId, CipherSuite
   $Cert_S$, [Root CAs]
3. C → S: [$Cert_C$]
   E_S(pre_master_secret),
   [$Sign_C$(all previous messages including)]
   ChangeCipherSpec
   MAC_{SK} ("client finished", all previous messages)
4. S → C: ChangeCipherSpec
   MAC_{SK}("server finished", all previous messages)

E_S = RSA encryption (PKCS #1 v1.5) with S's public key from $Cert_S$
pre_master_secret = random byte string chosen by C
master_secret = h(pre_master_secret, "master secret", N_C, N_S)

TLS_RSA handshake

1. C → S: Versions, N_C, SessionId, CipherSuites
2. S → C: Version, N_S, SessionId, CipherSuite
   $Cert_S$, [Root CAs]
3. C → S: [$Cert_C$]
   E_S(pre_master_secret),
   [$Sign_C$(all previous messages including)]
   ChangeCipherSpec
   MAC_{SK} (“client finished”, all previous messag
4. S → C: ChangeCipherSpec
   MAC_{SK}("server finished“, all previous messages)

Which security properties?

- Secret, fresh session key
- Mutual or one-way authentication
- Entity authentication, key confirmation
- Perfect forward secrecy (PFS)
- Contributory key exchange
- Downgrading protection
- Identity protection
- Non-repudiation
- Plausible deniability
- DoS resistance

E_S = RSA encryption (PKCS #1 v1.5) with S’s public key from $Cert_S$

pre_master_secret = random byte string chosen by C

master_secret = h(pre_master_secret, “master secret”, N_C, N_S)