Real-world verification: the case of security protocol standards

Marko Horvat

MPI-SWS

27/11/2017

Introduction

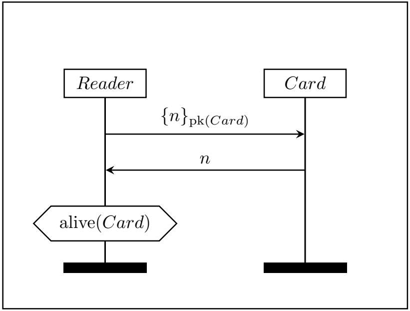
Actor Key Compromise

Improving the ISO/IEC 11770 standard

Formal analysis of TLS 1.3

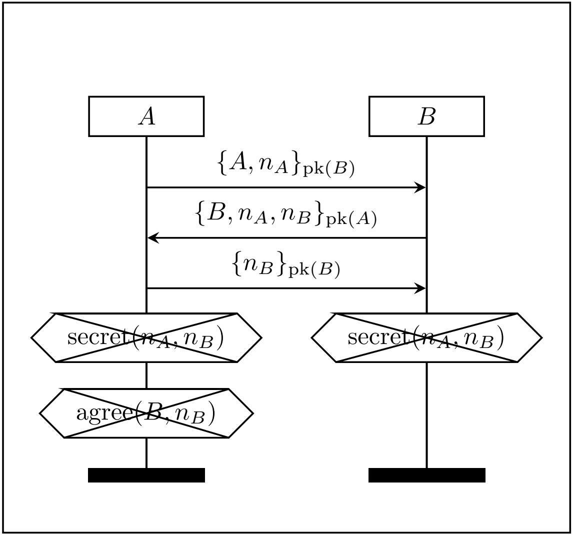
Security protocols. Actor Key Compromise (AKC)

- Most of us run security protocols on a daily basis:
  - secure searches, e-shopping, remote login, physical access, ...
- Example: simple challenge-response

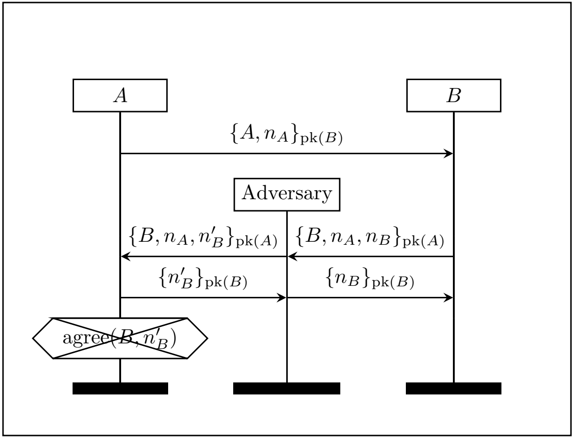


- Here reader knows card is present if $\mathrm{sk}(\mathit{Card})$ is secret
- Unfortunately, long-term secrets can be compromised
  - Lavabit, Heartbleed, $5 wrench, ...
- We might wonder: **can the reader get any security guarantees if $\mathrm{sk}(\mathit{Reader})$ is compromised**?

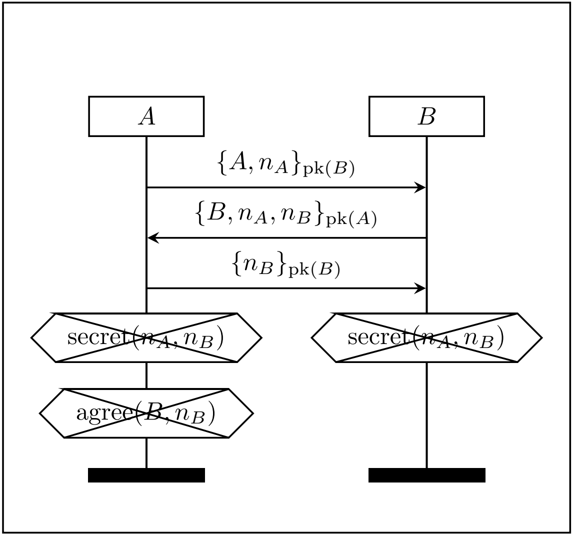
Securing NSL with respect to AKC



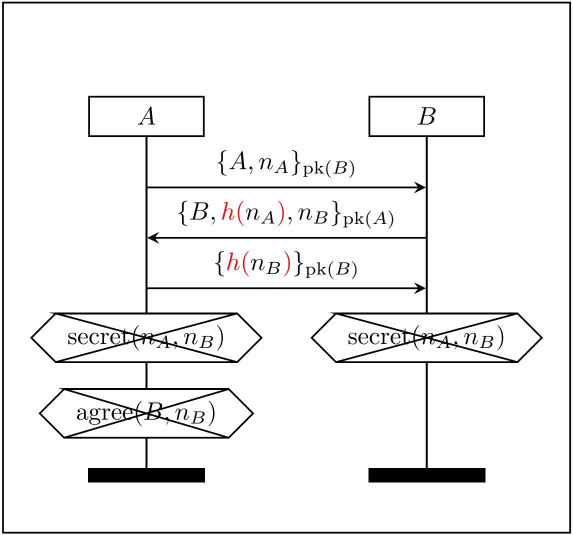
Securing NSL with respect to AKC



Securing NSL with respect to AKC

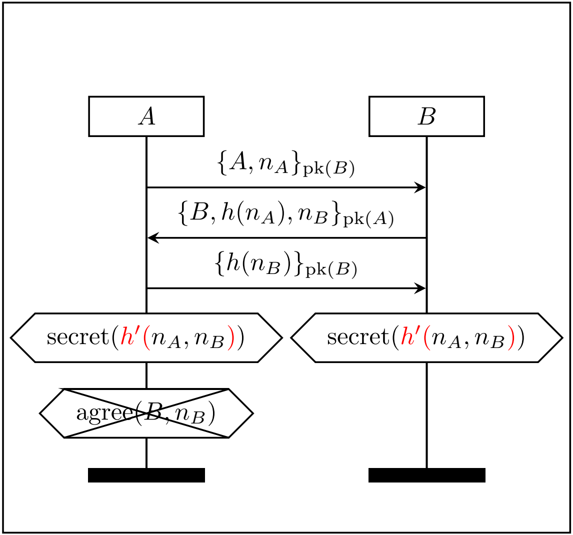


Securing NSL with respect to AKC



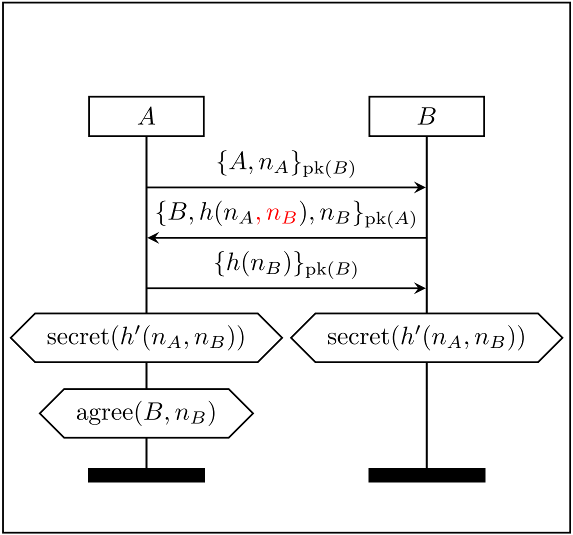
- hashed nonces in msgs #2,#3

Securing NSL with respect to AKC



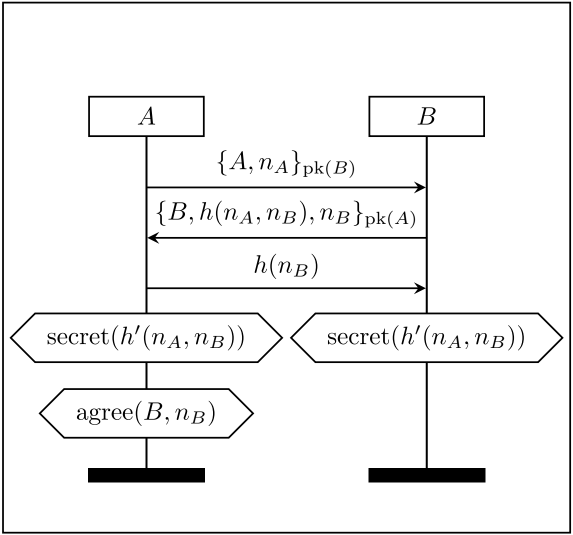
- hashed nonces in msgs #2,#3
- hashed nonces together to form key

Securing NSL with respect to AKC



- hashed nonces in msgs #2,#3
- hashed nonces together to form key
- copied n_B inside hash in msg #2

Securing NSL with respect to AKC



- hashed nonces in msgs #2,#3
- hashed nonces together to form key
- copied n_B inside hash in msg #2
- removed unnecessary encryption in msg #3

AKC results

- We use tool-supported formal methods for our case studies
  - Typical assumptions in symbolic setting
    - Perfect cryptography
    - Adversary controls the network
  - Four different adversary models
  - The strongest has all long-term keys but those of intended peer
  - Scyther used for small protocols, Tamarin otherwise
- We fix five vulnerable protocols:
  - NSL, two CCITT X.509 protocols, two modes of TLS-RSA
- We verify two protocols are AKC secure:
  - SSH Transport Layer, Mutual TLS-DHE_RSA
- All fixes must go beyond symmetric cryptography and hashing

ISO/IEC 11770

- Standard for key management techniques
  - Included in European Payments Council guidelines
  - Parts 2 and 3: 33 security protocols and over 50 variants
- We build on earlier work by Lara Schmid and Tomas Zgraggen
  - Significant modelling effort: informal properties, missing threat model
  - Scyther used for its easy scripting of batch analysis
  - Large amount of data with some great extrapolations
- Our main contributions:
  - We perform comprehensive analysis in minimal threat model
  - We establish clear relation of analysis to claims in standard
  - As a bonus, we consider AKC and UKS vulnerabilities

Advanced security properties

- Actor Key Compromise (AKC)
  - All protocols in Part 2 use symmetric cryptography and hashing only
  - Impossibility result from our previous paper: necessarily vulnerable to AKC
  - Four protocols in Part 3 vulnerable to AKC (easily replaced)
- Unknown Key Share (UKS)
  - Attacks where only Alice and Bob know session key K
  - However, Alice and Bob disagree on who they share K with
  - Using K does not authenticate subsequent messages
  - Protocols 3-KA-11 and 2-10 vulnerable to UKS
    - Another five from Part 2 if multiple roles per entity are allowed
  - Fix by binding certs/identities to keying material (NIST SP-800-56A)

ISO/IEC 11770 conclusions

- Main cause of problems:
  - Standard based on obsolete version of 9798 (entity authentication)
  - Prior to our work, no effort to fix inherited problems in 11770
- Recommendations to ISO/IEC 11770 working group:
  1. Make the threat model explicit
     - Allows for precise assessment if security requirements met
  2. Adopt recommendations for ISO/IEC 9798 (Basin et al.)
  3. Address remaining issues with 3-KA-11
     - Switch to TLS-DHE_RSA or adapt statements made
  4. Ensure resilience to AKC and UKS as described
- Current state of the standard:
  - 3-KA-11 removed from Part 3 in 2015 update
  - Part 2 scheduled to be fixed

ISO/IEC 11770 conclusions

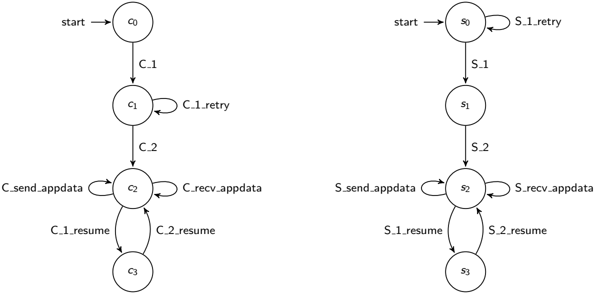
- Main cause of problems:
  - Standard based on obsolete version of 9798 (entity authentication)
  - Prior to our work, no effort to fix inherited problems in 11770
- Recommendations to ISO/IEC 11770 working group:
  1. Make the threat model explicit
     - Allows for precise assessment if security requirements met
  2. Adopt recommendations for ISO/IEC 9798 (Basin et al.)
  3. Address remaining issues with 3-KA-11
     - Switch to TLS-DHE_RSA or adapt statements made
  4. Ensure resilience to AKC and UKS as described
- Current state of the standard:
  - 3-KA-11 removed from Part 3 in 2015 update
  - Part 2 scheduled to be fixed

> **This standard was last reviewed and confirmed in 2014. Therefore this version remains current.**

Formal analysis of TLS 1.3

- TLS 1.2 critical in securing Internet communications today
- Lacking in both efficiency and security
- TLS Working Group preparing TLS 1.3 draft
- We analyse rev 06 of the specification
  - Joint with Cas Cremers, unpublished
- Our tool of choice is the Tamarin prover
  - Supports loops, non-monotonic state, Diffie-Hellman...
- Evolution of Tamarin models of TLS

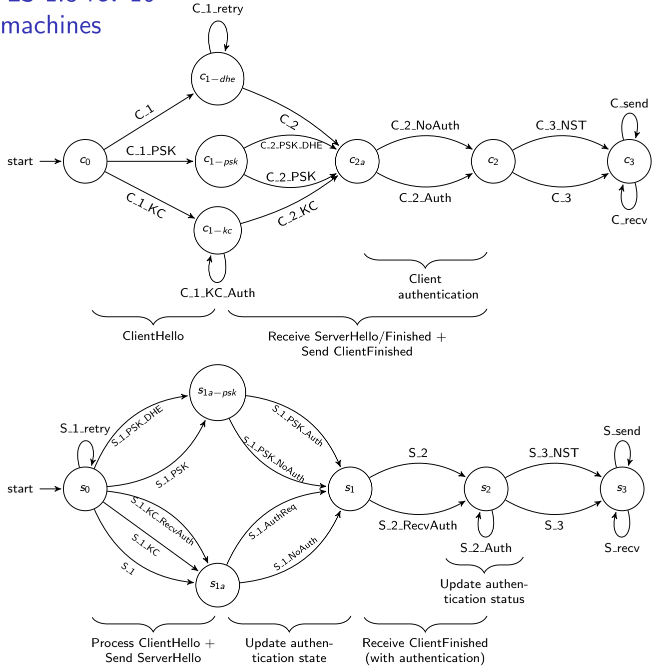
  Basic TLS 1.2 model→Refined TLS 1.2 model (2014 Q4)
  →TLS 1.3, rev 06 model (first half of 2015)



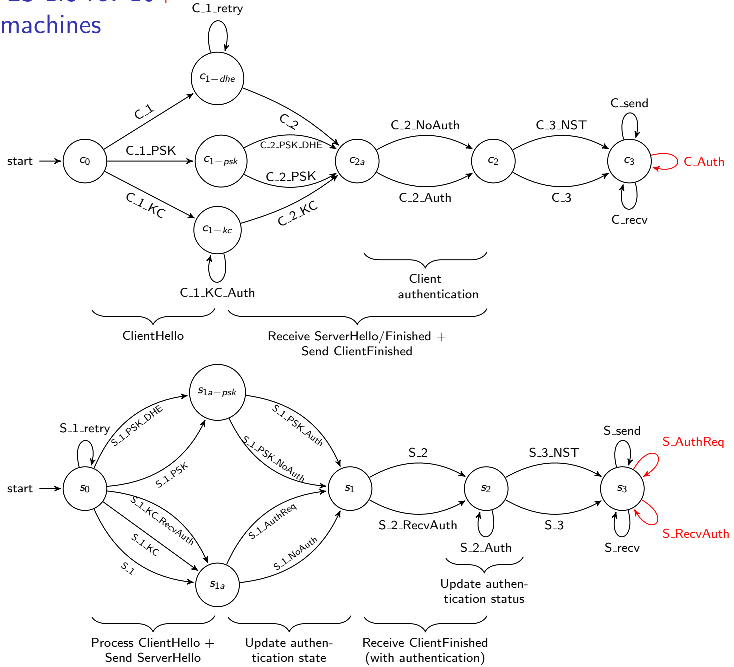
Results for TLS 1.3, rev 06 and beyond

- In rev 06, session keys secret in both authentication modes
  - Powerful symbolic attacker: active, AKC, PFS, DH reveal
  - Unbounded analysis breadth (concurrent threads)
  - Unbounded depth (retries, resumptions, data exchanges)
  - Limited coverage: single authentication mode at a time
- Next step: refine to TLS 1.3, rev 10
  - Joint with Cas Cremers, Sam Scott, Thyla van der Merwe
  - Collaboration of Mozilla, Oxford, RHUL
  - Second half of 2015
- TLS 1.3, rev 10 results:
  - Standard AKE security requirements verified
    - Session key secrecy and entity authentication
    - Any mix of authentication modes, but no DH reveal
  - Attack on its extension (RWC, TRON, S&P)
  - This work led to an update of the current (rev 11) draft
- Latest work (also with Jonathan Hoyland): rev 21 (CCS)

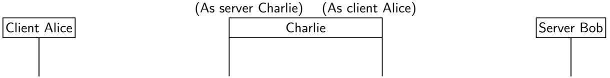
Our TLS 1.3 rev 10 state machines



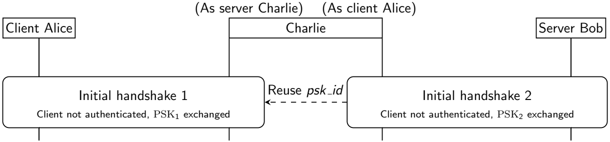
Our TLS 1.3 rev 10+ state machines



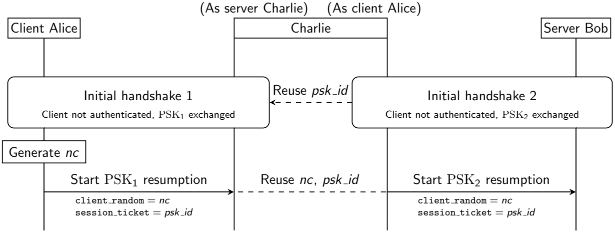
TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)



TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)

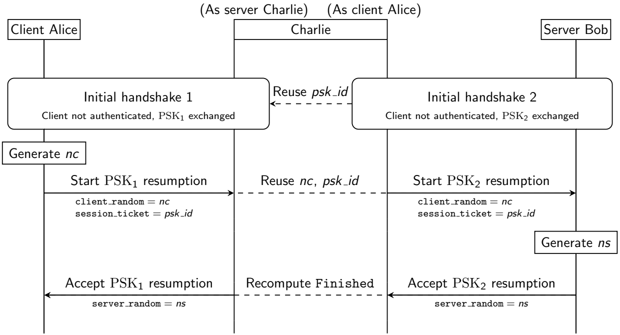


TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)

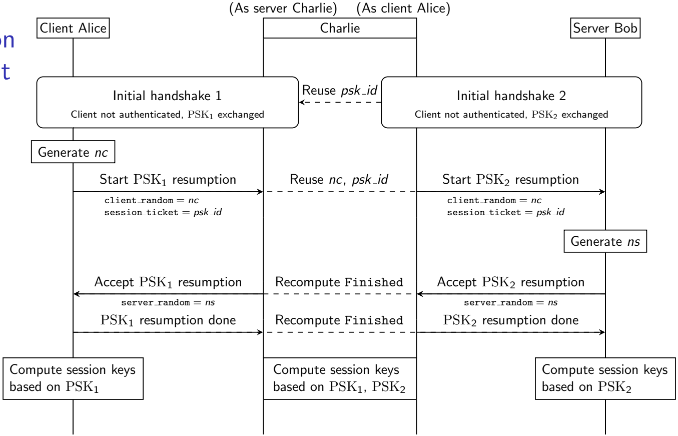


mhorvat@mpi-sws.org

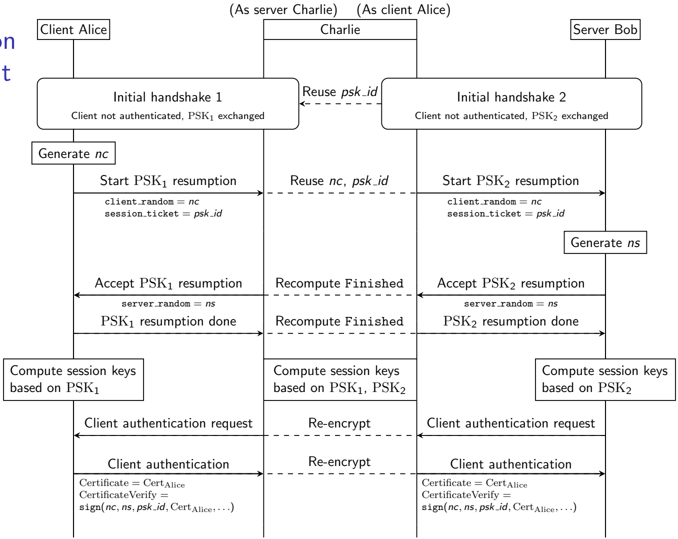
TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)



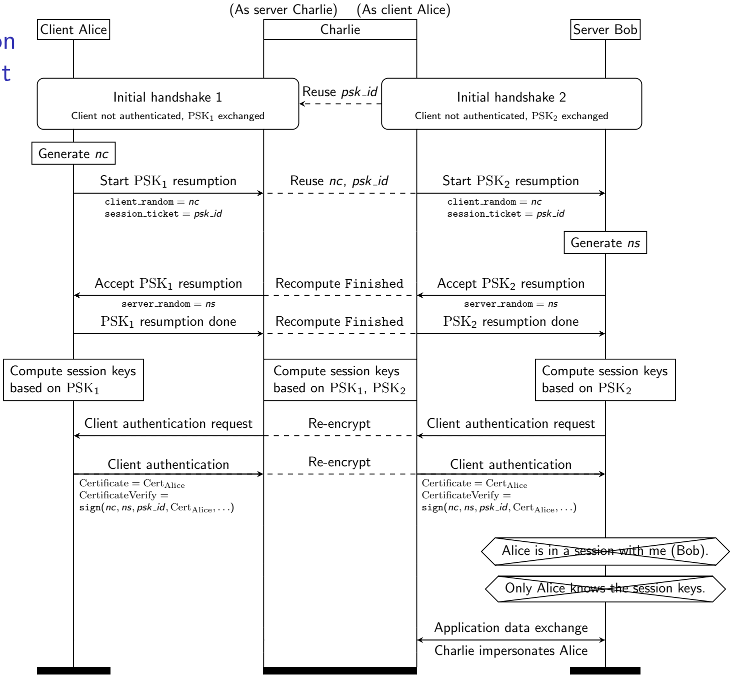
TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)



TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)



TLS 1.3 rev 10+ client impersonation attack (PSK+client authentication)



Negative AKC result

Impossibility of authentication under AKC

Suppose P is a protocol where:

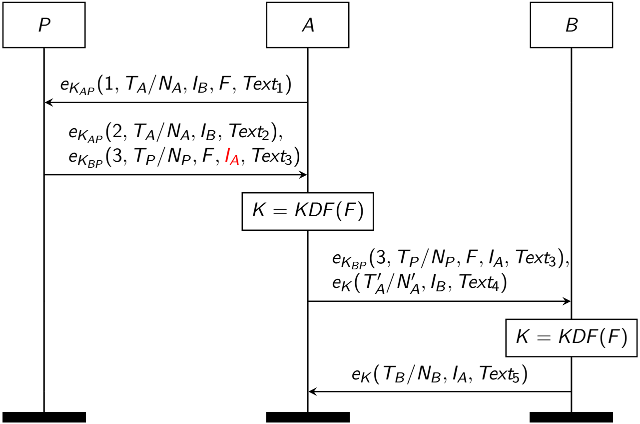
- symmetric cryptography and hashing are the only cryptographic primitives used, and
- freshly generated values are first sent out in accessible positions
  - not hashed (includes approximations, e.g. DH)
  - not used as symmetric keys

Then aliveness cannot be achieved in P under AKC.

ISO/IEC 11770 security properties and threat model

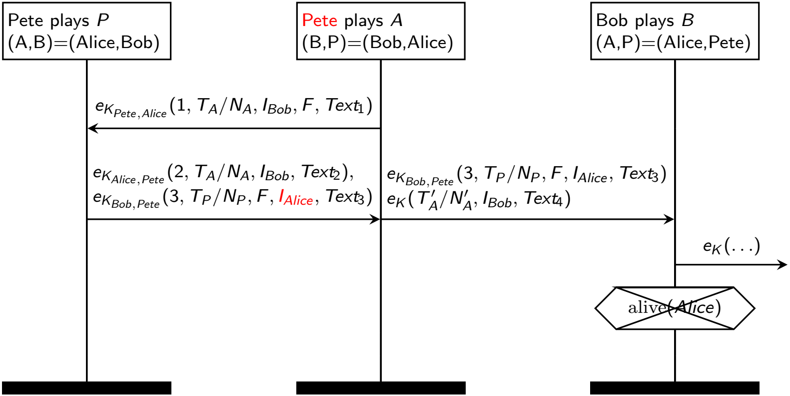
- Informal security properties made explicit for each protocol:
  - entity authentication
  - key authentication
  - forward secrecy
  - ...
- We make reasonable assumptions on adversary capabilities:
  - **Injecting/tampering with network messages**
    - only way to effectively violate entity authentication
  - **Eavesdropping on network messages**
    - otherwise, we would need no complex key management, but simple authentication mechanisms
  - **Compromising long-term private keys of entities**
    - only way to violate perfect forward secrecy

Protocol 2-12 with optional parts

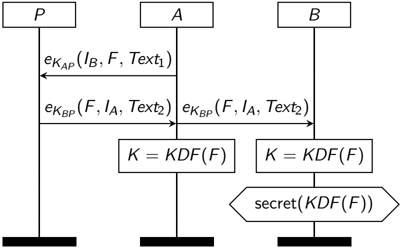


- Derived from a mutual authentication mechanism in 9798-2
- Claimed to satisfy mutual explicit key authentication, mutual key confirmation and mutual entity authentication
- But: A cannot/does not decrypt $e_{K_{BP}}(3, T_P/N_P, F, I_A, Text_3)$

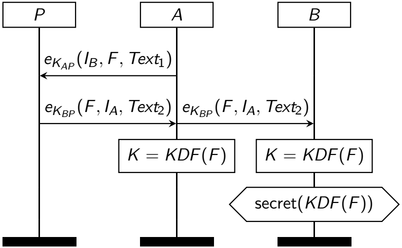
AT1: Entity authentication failure for protocol 2-12

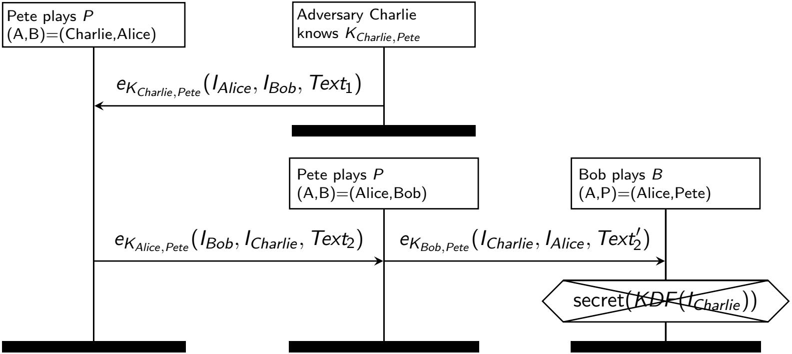


AT4: Type-flaw attack on key authentication in 2-11

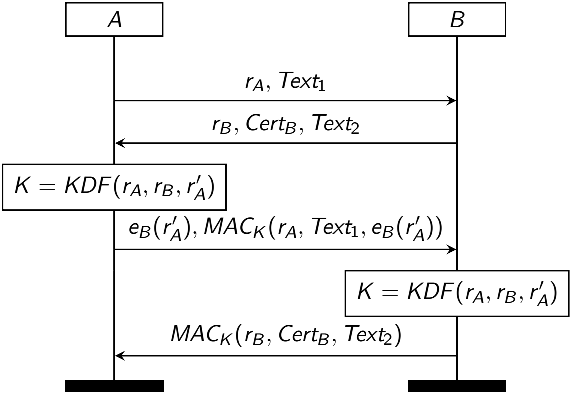


AT4: Type-flaw attack on key authentication in 2-11





Protocol 3-KA-11



- According to the standard, it offers mutual explicit key authentication and MFS
- Derived from unilaterally authenticated TLS_RSA, so provides neither

Claimed properties in Part 2

Mechanism in Part 2	Key Authentication	Key Confirmation	Entity Authentication
2-1	implicit	no	no
2-2	implicit	no	no
2-3	explicit	no	A
2-4	explicit	no	A
2-5	explicit	no	A & B
2-6	explicit	no	A & B
2-7	implicit	no	no
2-8	**explicit**(AT1)	**opt.**(AT1)	**opt.**(AT1)
2-9	**explicit**(AT1)	**opt.**(AT1)	**opt.**(AT1)
2-10	explicit	no	no
2-11	**explicit**(AT4)	no	no
2-12	**explicit**(AT1)	**opt.**(AT1)	**opt.**(AT1)
2-13	**explicit**(AT1)	**opt.**(AT1)	**opt.**(AT1)

Claimed properties in Part 3

Mechanism in Part 3	Implicit Key Authentication	Key Confirmation	Entity Authentication	Forward Secrecy
3-KA-1	A,B	no	no	no
3-KA-2	B	no	no	A
3-KA-3	A,B	B	A	A
3-KA-4	no	no	no	MFS
3-KA-5	A,B	opt	no	A,B
3-KA-6	A,B	opt	B	B
3-KA-7	A,B	A,B	A,B	MFS
3-KA-8	A,B	no	no	A
3-KA-9	A,B	no	no	MFS
3-KA-10	A,B	A,B	A,B	MFS
3-KA-11	A,**B**(AT2)	A,**B**(AT2)	B	**MFS**(AT3)
3-KT-1	B	no	no	A
3-KT-2	B	B	A	A
3-KT-3	B	B	A	A
3-KT-4	A	A	B	B
3-KT-5	A,B	(A),B	A,B	no
3-KT-6	A,B	**A,B**(AT5)	A,B	no

TLS 1.3 rev 10 (Full handshake, 0-RTT, PSK)