Improving the ISO/IEC 11770 standard for key management techniques

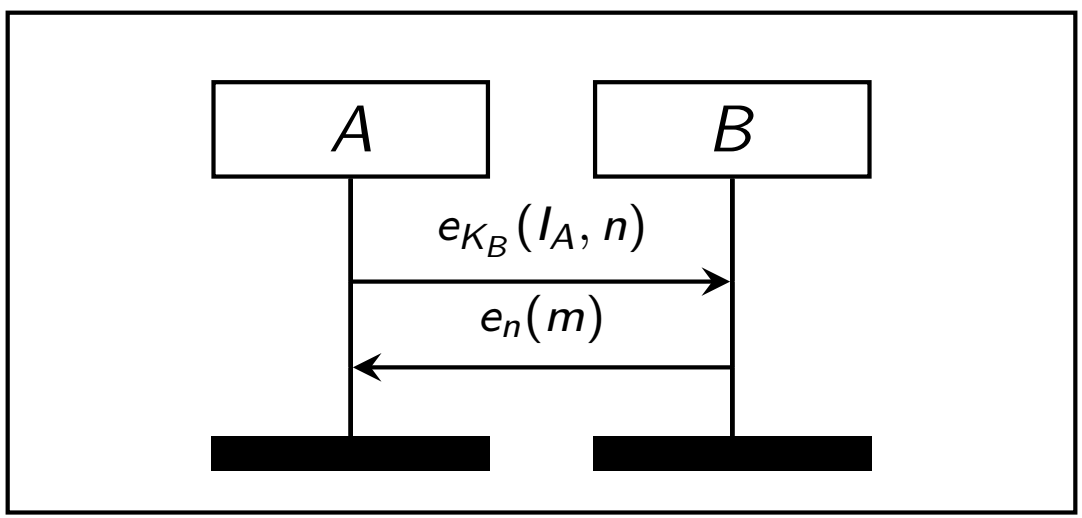
Cas Cremers, *Marko Horvat*

University of Oxford, UK

16 December 2014

ISO/IEC 11770

- ISO: International Organisation for Standardisation
- ISO standards often mandated by, e.g., oversight bodies
- Most previous formal analyses very limited in scope
- Exception: Basin et al. analysed ISO/IEC 9798 (entity authentication)
- IEC: International Electrotechnical Commission
- Our analysis: ISO/IEC 11770 (key management techniques)
- Key management: share secret, later encrypt to securely communicate



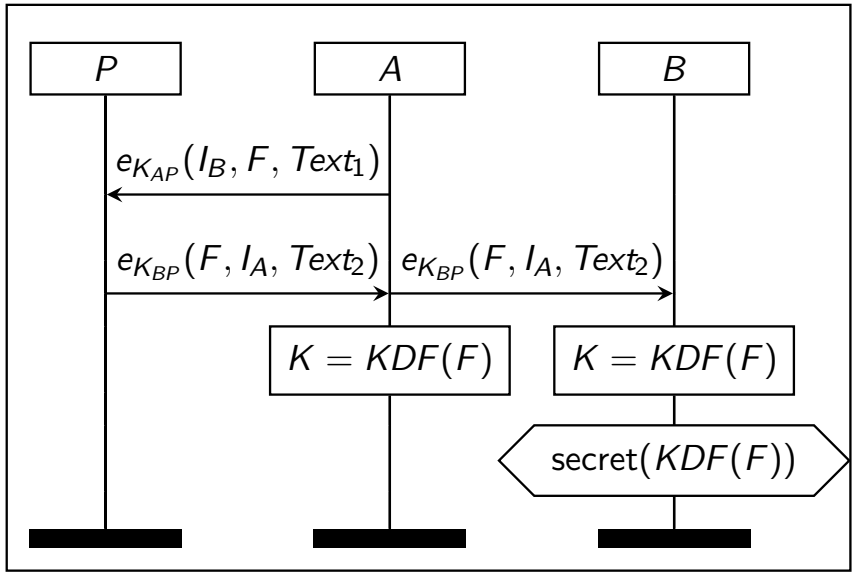
ISO/IEC 11770

- 30 security protocols and over 50 variants
- Part 2: symmetric techniques, Part 3: asymmetric techniques
- Many man hours spent on improving models and property specifications
- Two students supervised by Cas Cremers got their BSc for work on standard (Lara Schmid, Tomas Zgraggen)
- Students' work led to theoretically interesting properties for coarse protocol models
- Our aim: combine earlier work with precise models and property analyses that have clear relation to claims in standard
- Tool used: compromising adversaries branch of Scyther tool
- We uncover several incorrect claims and provide fixes
- Version of this report provided to the ISO/IEC working group

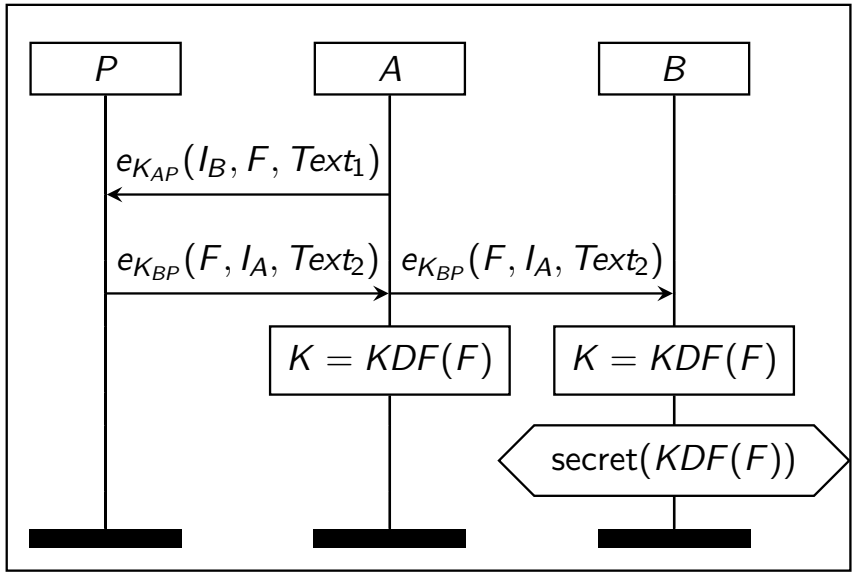
Security properties and threat model (a.k.a. *the* adversary)

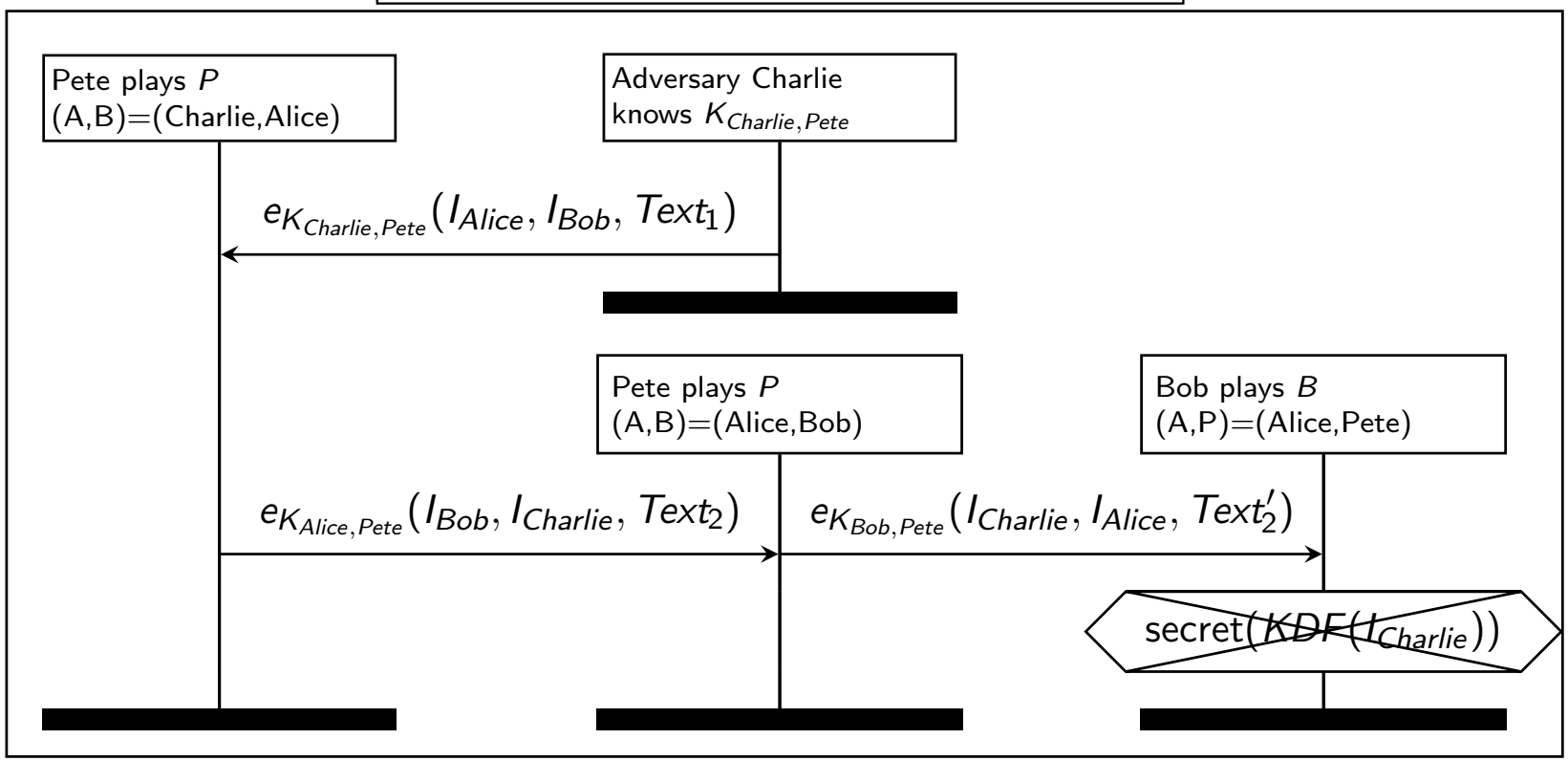
- Claimed security properties made explicit for each protocol:
  - entity authentication
  - key authentication
  - forward secrecy
  - ...
- However, definitions informal, so we have to choose formalisations
- Also, standard does not specify an explicit threat model
- We make reasonable assumptions on adversary's 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

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



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

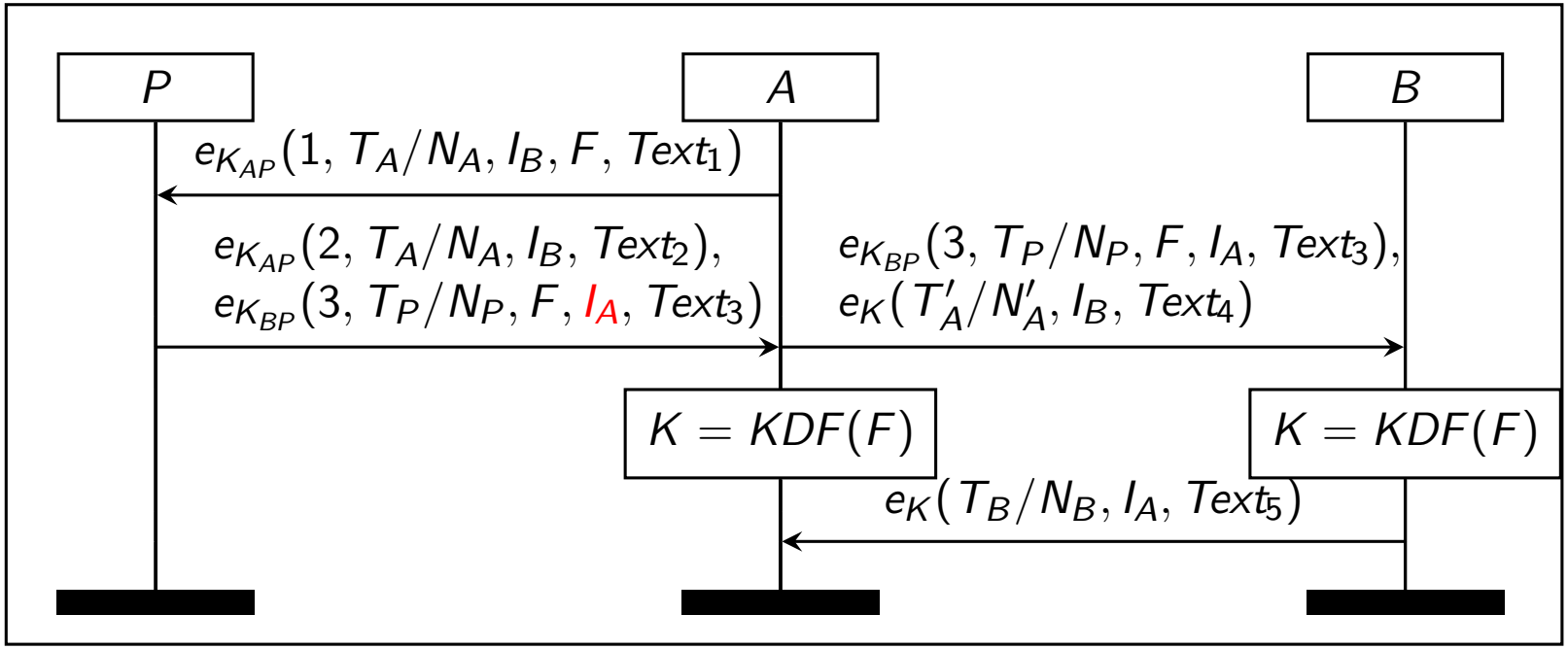




Claimed properties in Part 2

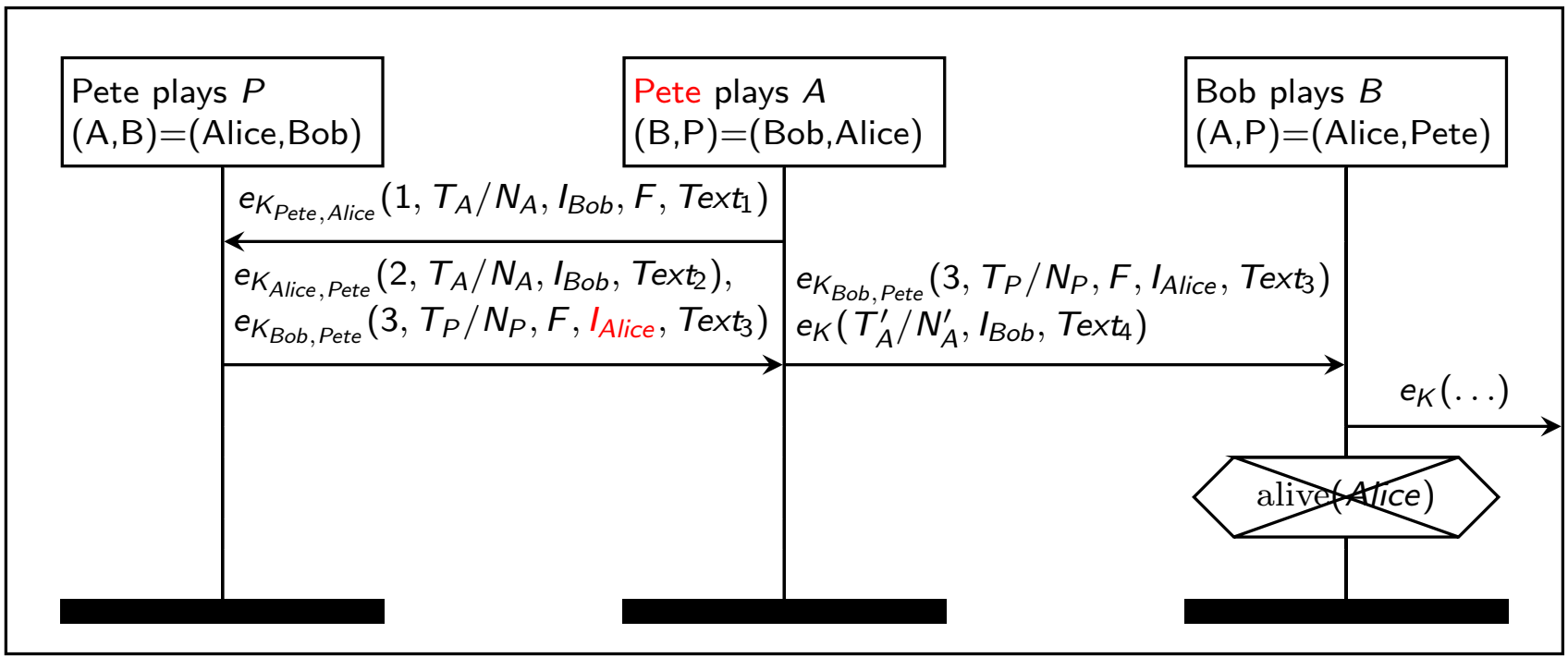
Mechanism in part 2	Key Authentication	Key Confirmation	Entity Authentication
2-1	implicit	no	no
2-2	implicit	no	no
2-3	explicit	no	A
2-4	explicit	no	A
2-5	explicit	no	A & B
2-6	explicit	no	A & B
2-7	implicit	no	no
2-8	**explicit**(AT1)	**opt.**(AT1)	**opt.**(AT1)
2-9	**explicit**(AT1)	**opt.**(AT1)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

Protocol 2-12 with optional fields



- Derived from a four-pass mutual authentication mechanism in ISO/IEC 9798-2
- Claimed to satisfy mutual explicit key authentication, mutual key confirmation and mutual entity authentication
- Role A cannot/does not decrypt $e_{K_{BP}}(3, T_P/N_P, F, I_A, Text_3)$
- If it did, A could check its identity I_A

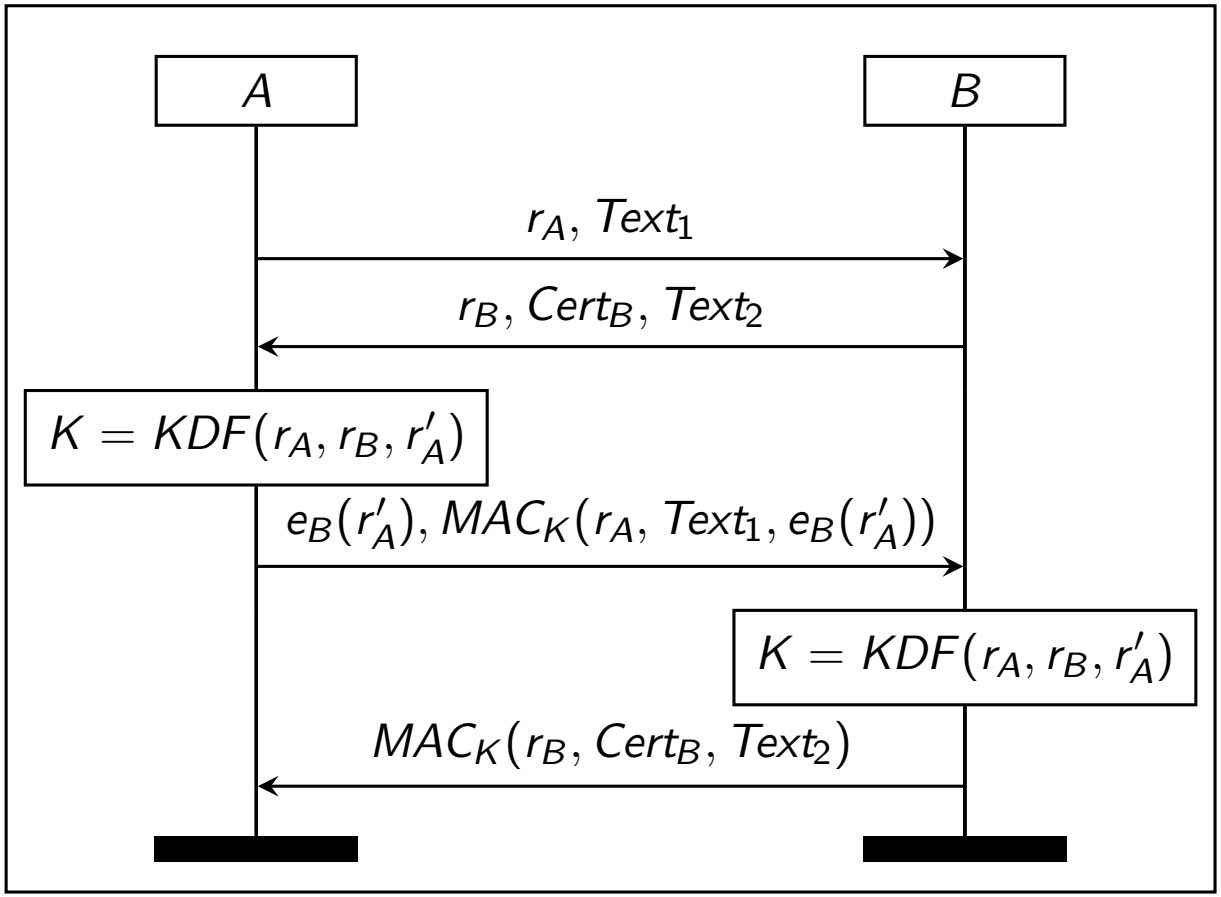
AT1: Entity authentication failure for protocol 2-12



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

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

Unknown key share (UKS) attacks

- Attacks in which only Alice and Bob know session key K
- However, Alice and Bob disagree on who they share K with
- Even though adversary does not learn K, using K does not authenticate subsequent messages
- 3-KA-11 is derived from TLS, but vulnerable to UKS attack
- All messages confirmed in TLS, including identities

Key compromise impersonation (KCI) resilience

- Desirable property of key exchange protocols
- KCI attack: adversary exploits his knowledge of the long-term private key of Alice to impersonate any entity in subsequent communication with Alice
- All protocols in part 2 use symmetric cryptography and hashing only
- Impossibility result from our previous paper: necessarily vulnerable to KCI attacks
- Four protocols from part 3 vulnerable to KCI (same as ¬PFS here)

Rec. 1: Improving protocols to achieve the stated properties

- Most straightforward way to fix most protocols is to adopt the recommendations made for ISO/IEC 9798 (Basin et al.)
- In particular, we require that:
  - no cryptographic data must be interchangeable, which can be enforced by including unique tags,
  - when optional fields are not used, then they must be set to empty, and
  - entities that perform the role of the TTP in the 2-8, 2-9, 2-12 and 2-13 protocols must not perform the A or B role.
- When this is done, only 3-KA-11 remains vulnerable to attacks

Rec. 2: Using appropriate key derivation functions (KDFs)

- If input to KDF includes identities of communicating parties, UKS is directly prevented
- We recommend making this an explicit requirement
- KDF from NIST SP-800-56A described in the standard meets this requirement

Rec. 3: Addressing remaining issues with 3-KA-11

- 3-KA-11 inherently does not offer perfect forward secrecy or mutual authentication
- Stripped down version of unilateral TLS-RSA handshake where security-relevant information has been removed
- Switching to, e.g., mutually authenticated TLS-DHE_RSA, changes environmental assumptions
- Simpler solution: adapt statements made
- False statements:
  - implicit key authentication for both entities
  - key confirmation for both entities
  - mutual explicit key authentication
  - mutual forward secrecy

Related work

- Long history of breaking and fixing different versions of 3-KT-6 and 2-12 (1998–2008)
- Researchers involved: Horng, Hsu, Mitchell, Yeun, Cheng, Comley, etc.
- Mostly due to interpreting identity fields as fresh keys, or using too few identifiers
- Mathuria and Sriram used Scyther to discover more complex type-flaw attacks on 2-13 and Cheng's and Comley's proposed fixed protocol
- Attacks rely on the possibility that complex fields can be interpreted as atomic
- In 2010, Chen and Mitchell looked at general concepts underlying type-flaw attacks
- Their countermeasures in some later versions of ISO standards

Conclusions

- In retrospect, some attacks should have been found by manual inspection, such as on flawed TLS variant
- Two ways in which standardisation bodies could be more proactive:
  - publish early drafts free of charge, to promote external analysis
  - be aware of analyses of standards on which they build
- No attempt was made to determine if protocols derived from ISO/IEC 9798 inherited same problems, which is in fact the case
- Applying the recommendations for ISO/IEC 9798 would have prevented all issues we showed, except for those with 3-KA-11