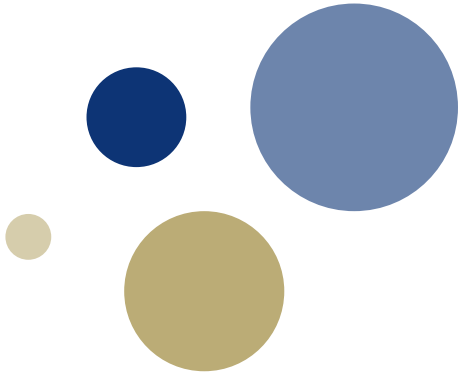


Symmetric Key Exchange with Full Forward Security and Robust Synchronization

Colin Boyd, Gareth T. Davies, Bor de Kock, Kai Gellert, Tibor Jager and Lise Millerjord

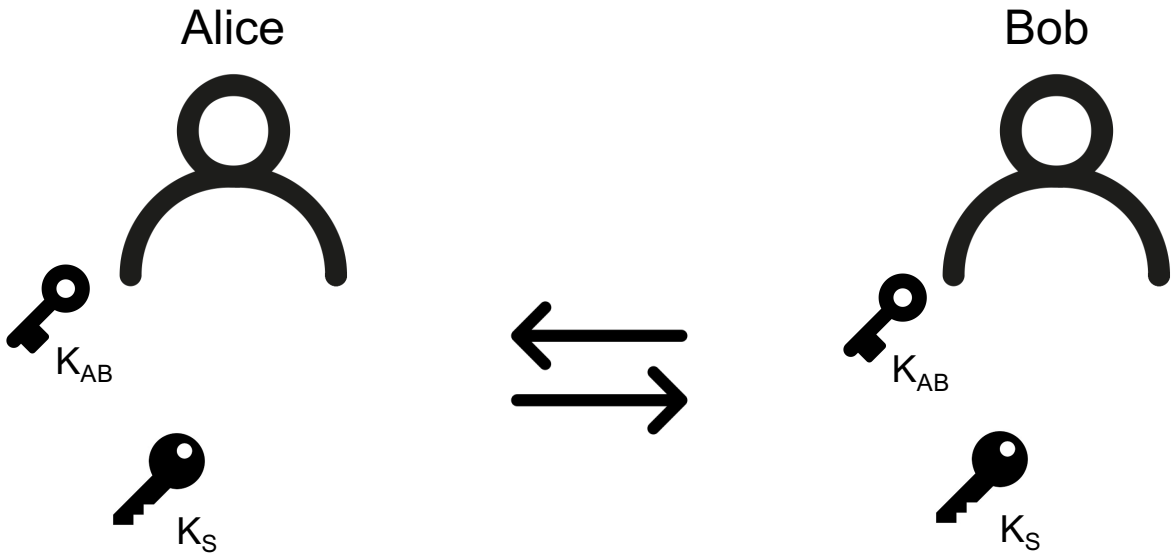
The Problem

- Authenticated key exchange for very constrained devices
- Pre-shared symmetric keys
- Forward security
- Synchronization
- Concurrent Correctness

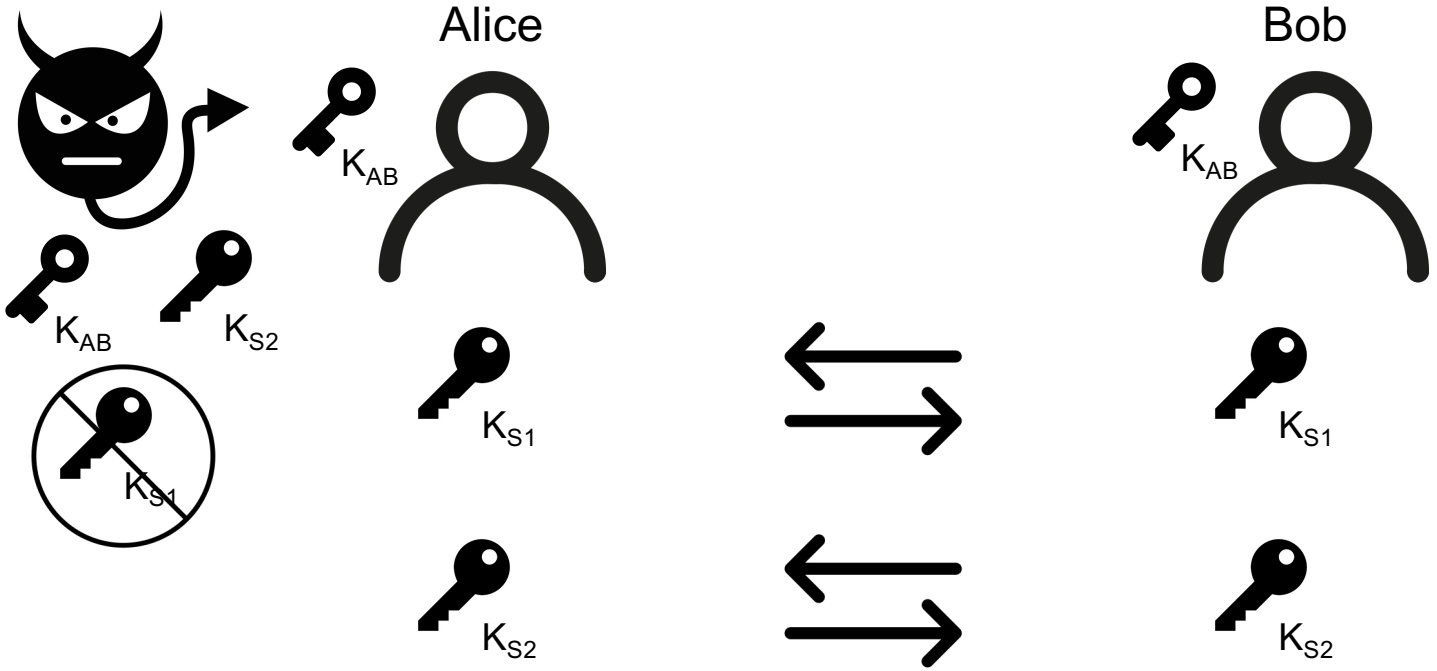
Our Contributions

- 3 very efficient AKE protocols with linear key evolving
- 2 AKE protocols with non-linear key evolving
- Framework for protocol analysis
- Formalization of synchronization robustness as a security property

Authenticated Key Exchange - AKE



Forward Security



Achieving Forward Security

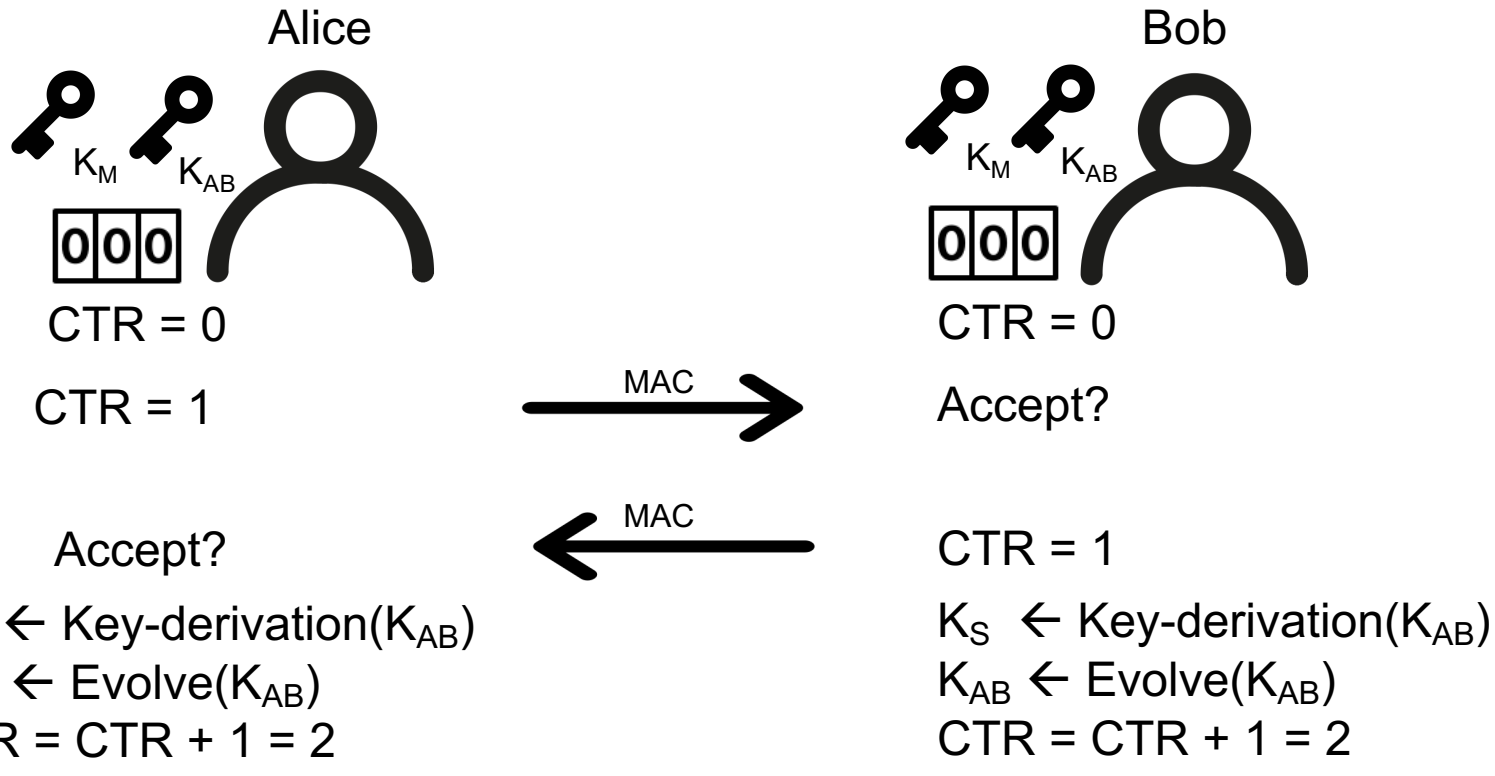
- Evolve keys to obtain forward security
- Time-based evolution [Dousti and Jalili, 2015]
- Triggered evolution: evolve after session key derivation

Challenges

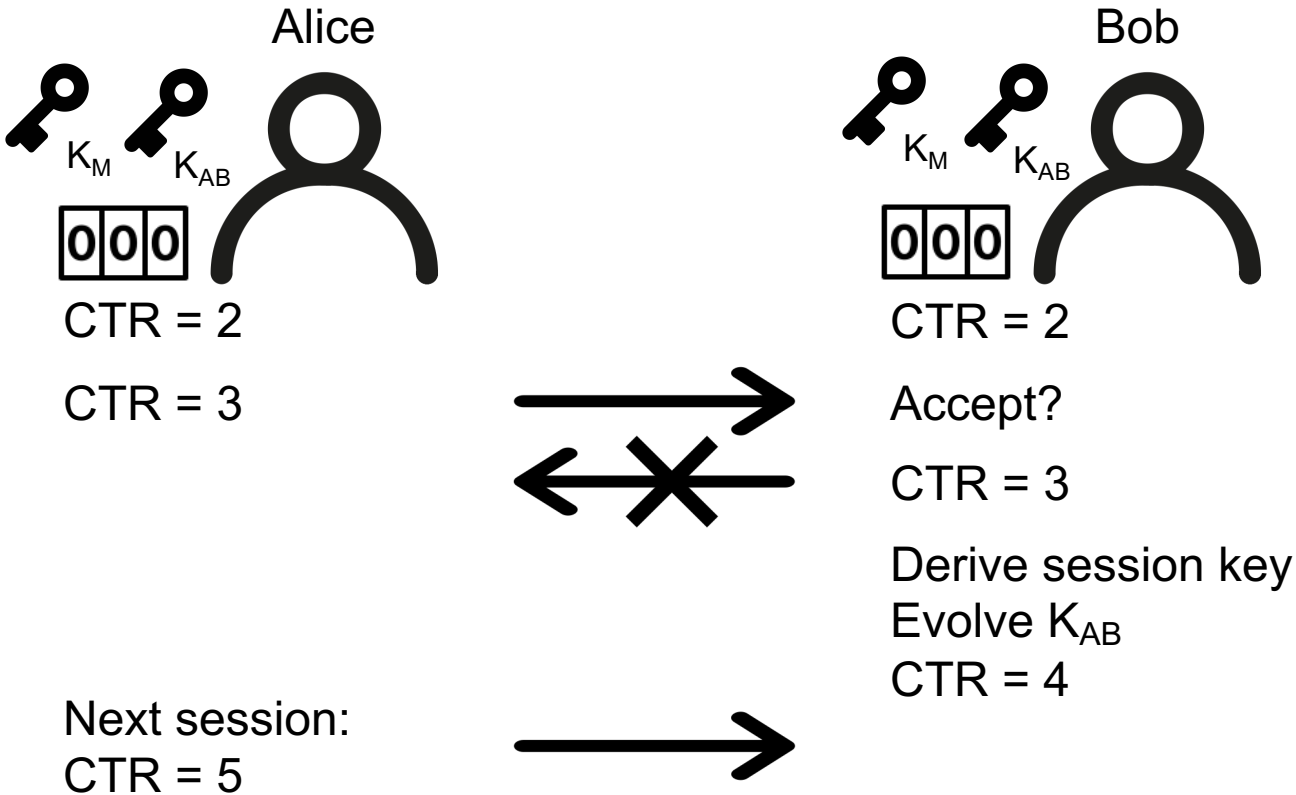
- Synchronization - both parties needs to have evolved the same number of steps
- Concurrent correctness – parallel sessions cause problems when one session evolves shared key material before the other session is ready

Protocol	Auth.	# of Messages	Sync. Rob. Weak	Sync. Rob. Full	Conc. Corr.	Forward Security
SAKE [ACF20]	Mutual	5	✗	✗	✗	✓
SAKE-AM [ACF20]	Mutual	4	✗	✗	✗	✓

Example protocol: LP2



Example protocol: LP2



Linear-evolving protocols

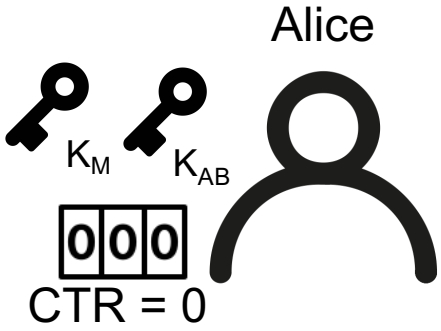
- 3 protocols – 1, 2 and 3 messages
- 1-message protocol: one-way authentication
- 3-message protocol: key confirmation, bounded gap

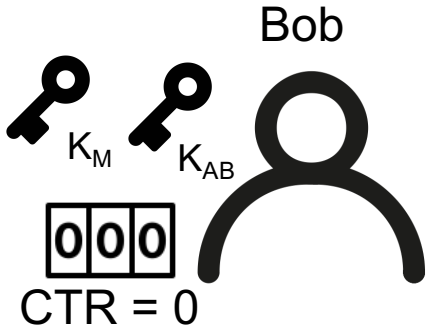
Protocol	Auth.	# of Messages	Sync. Rob. Weak	Sync. Rob. Full	Conc. Corr.	Forward Security
LP1	R only	1	✓	✗	✗	✓
LP2	Mutual	2	✓	✗	✗	✓
LP3	Mutual	3	✓	✗	✗	✓

Security model

- Framework for protocol analysis
- AKE model from [Bellare Rogaway 94, Li et al 2014]
- Model lacks notion of concurrent correctness and synchronization
- Formalization of synchronization robustness – the ability to compute keys in future sessions if something goes wrong

Concurrent Correctness



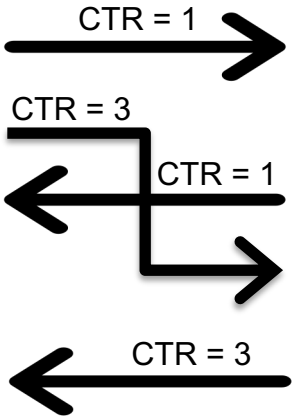


Initiate session 1: CTR = 1

Initiate session 2: CTR = 3

Accept?
Alice is at CTR = 3,
Aborts session 1

Accept?
Alice accepts with CTR = 4



Accept?

CTR = 1, session completes,
Bob accepts with CTR = 2

Accept?
CTR = 3, session completes,
Bob accepts with CTR = 4

Synchronization Robustness

- Captures the ability of two parties succeeding in exchanging a session key in the future, no matter what has happened previously.
- If the parties get out of sync, we need to be able to resynchronize.
- This definition formalizes this requirement and comes in a weak and a strong flavour.

Weak Synchronization Robustness

- Definition: Any honestly executed, uninterrupted session will succeed no matter what has happened before.
  - Concurrent sessions were initiated
  - Messages in previous sessions were dropped, reordered or altered (so that they were not accepted)
  - Parties are arbitrarily many steps out of sync
  - Either way: the next session Alice and Bob are allowed to execute without any interruption will succeed
- LP2: Allowing role reversal will make the protocol fail to meet this requirement

Full Synchronization Robustness

- Definition: Any honestly executed session will succeed, no matter what else is going on with concurrent sessions or previous sessions.
  - Arbitrary many concurrent sessions are allowed
  - The adversary may interleave messages with concurrent sessions arbitrarily
  - The adversary may make any previous or concurrent sessions fail, but the one that is allowed to complete honestly will succeed
- Linearly evolving protocols fail this requirement

Non-linear key evolution

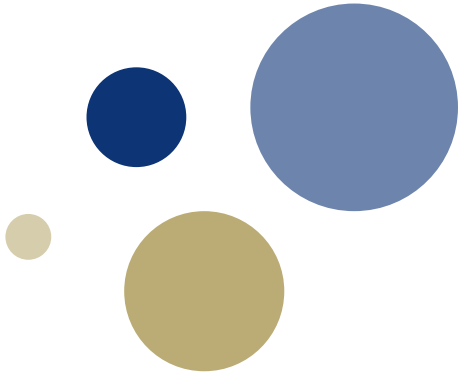
- Need something different to achieve full synchronization robustness
- Use puncturable pseudorandom functions [Sahai Waters 2014]
- Definition: A PPRF is a PRF with an extra algorithm Punct(k, x) such that
  - Evaluating on a punctured value fails
  - Puncturing on an already punctured value returns the same key
  - Puncturing is commutative – the order in which you puncture values does not matter
- Session key is determined by evaluating on the session nonce
- All concurrent sessions can succeed: puncturing only affects key material of that particular session

The non-linearly evolving protocols

Protocol	Auth.	# of Messages	Sync. Rob. Weak	Sync. Rob. Full	Conc. Corr.	Forward Security
PP1	R only	1	✓	✓	✓	✓
PP2	Mutual	2	✓	✓	✓	✓

The protocols and their properties

Protocol	Auth.	# of Messages	Sync. Rob. Weak	Sync. Rob. Full	Conc. Corr.	Forward Security
SAKE [ACF20]	Mutual	5	✗	✗	✗	✓
SAKE-AM [ACF20]	Mutual	4	✗	✗	✗	✓
LP1	R only	1	✓	✗	✗	✓
LP2	Mutual	2	✓	✗	✗	✓
LP3	Mutual	3	✓	✗	✗	✓
PP1	R only	1	✓	✓	✓	✓
PP2	Mutual	2	✓	✓	✓	✓



Thank you for your attention

Colin Boyd, Gareth T. Davies, Bor de Kock, Kai Gellert, Tibor Jager and Lise Millerjord

https://eprint.iacr.org/2021/702