COMPUTATIONAL ANALYSIS OF THE EDHOC PROTOCOL

Baptiste COTTIER

SYMBOLIC VS COMPUTATIONAL SECURITY

	Symbolic	Computational
PRIMITIVES	Treated as blackboxes	Functions on bitstrings
MESSAGES	(typed) Terms	Bitstrings (1001101101)
ATTACKER	Restricted to compute only using these primitives	Any probabilistic polynomial-time algorithm
SECRECY	Attacker can not distinguish when the value of the secret changes	Attacker can not distinguish the secret from a random value

CONTEXT

- EDHOC constraints:
  - Small number of messages (ideally 3, or 4 with key-confirmation)
  - Small message size (~100 bytes in total)
  - Minimize code and memory footprint

- Analysis done in the **static-static** setting using:
  - 128 bits-security **Elliptic Curve DH**
  - 64-bits security **MAC** (trade-off to reduce communication)

Id	AEAD	Hash	MAC len	ECDH curve	Signature	Application AEAD	Note
0	AES-CCM-16-64-128	SHA-256	8	X25519	EdDSA	AES-CCM-16-64-128	constrained
1	AES-CCM-16-128-128	SHA-256	16	X25519	EdDSA	AES-CCM-16-64-128	constrained
2	AES-CCM-16-64-128	SHA-256	8	P-256	ES256	AES-CCM-16-64-128	constrained
3	AES-CCM-16-128-128	SHA-256	16	P-256	ES256	AES-CCM-16-64-128	constrained
4	ChaCha20/Poly1305	SHA-256	16	X25519	EdDSA	ChaCha20/Poly1305	
5	ChaCha20/Poly1305	SHA-256	16	P-256	ES256	ChaCha20/Poly1305	
6	A128GCM	SHA-256	16	X25519	ES256	A128GCM	
24	A256GCM	SHA-384	16	P-384	ES384	A256GCM	high-security
25	ChaCha20/Poly1305	SHAKE256	16	X448	EdDSA	ChaCha20/Poly1305	high-security

SECURITY GOALS

- Security Level of 128 bits: Minimum expected time needed to attack the protocol.

  With T the execution time of the protocol and ε the success probability of the attack, we have:

$$\mathrm{T}/\varepsilon \cong 2^{128}$$

- Applicative data confidentiality:
  - **Key-Privacy**: At most both participants know the final session key. By compromising the long-term credential of either peer, an attacker shall not be able to compute past session keys
  - **Mutual Authentication**: Exactly both participants have the material to compute the final session key
  - **Identity Protection**

COMPUTATIONAL ANALYSIS (TO BE PROVEN)

Key Privacy

- Equivalent to **Implicit Authentication**
- Relies on the **Computational Diffie-Hellman** assumption
  - Depends on the group size where Diffie-Hellman is considered

- Indistinguishability in the **Find-Then-Guess model**. The adversary is given access to oracles :
  - **Send**: models an active attack, in which the adversary may intercept a message and then either modify it, create a new one, or simply forward it to the intended participant.
  - **Reveal**: models the misuse of session keys by a user
  - **Test**: tries to capture the adversary's ability (or inability) to tell apart a real session key from a random one
    - ➢ Given several accesses to the **Send** and **Reveal** oracles, and only one access to the **Test** oracle, the attacker succeed if he can distinguish the session key from a random value

COMPUTATIONAL ANALYSIS (TO BE PROVEN)

Mutual Authentication

- Equivalent to Explicit Authentication

- Ends when both parties activate the following flags (initialized at 0):
  - **Accept**: asserts that we have the required material
  - **Terminate**: asserts that other party has the required material

- Relies on MAC security :
  - 64 bits MAC provides 128-bits security
  - *To check*: Is 128 bits security reached after few AEAD messages?

COMPUTATIONAL ANALYSIS (TO BE PROVEN)

Identity Protection

- The protocol should protect the identity of the parties:
  - against **active** attackers for the *Initiator*
  - against **passive** attackers por the *Responder*

- Security games:
  - Given two identities, an **active** attacker should not distinguish the *Initiator*
  - Given two identities, a **passive** attacker should not distinguish the *Responder*

PROTOCOL DECOMPOSITION