Refining Key Agreement against Strong Adversaries

Joseph Lallemand, supervised by David Basin, with Christoph Sprenger

joseph.lallemand@ens-cachan.fr

ENS Cachan – ETH Zürich

Context

Key agreement protocols are central in many security architectures, hence methods for their rigorous development and analysis are of vital importance. We propose an approach to developing key agreement protocols that are correct by construction in the presence of strong adversaries. Our work brings together two separate lines of research:

- Verification methods, to check that protocols are secure against compromising adversaries, *i.e.* adversaries that not only are in control of the communication network (Dolev-Yao model), but can also compromise (learn) various data from the participants, such as session keys or long-term keys.
- Development methods for protocols that are correct by construction, such as refinement methods (see on the right). Refinement is a popular formalism for developing systems together with their correctness guarantees.

Refinement

Protocols are modelled by transition systems formed of states, some of which are initial, and events. Let T_a and T_c be such systems (*abstract* and *concrete*), and π a mapping from concrete to abstract states. T_c *refines* T_a with *refinement mapping* π, written $T_c \sqsubseteq_\pi T_a$, if:

- π maps initial states to initial states;
- for every concrete event evt_c, there is an abstract event evt_a such that $\pi^{-1} \circ evt_c \circ \pi \subseteq evt_a$.

These conditions essentially ensure that T_c simulates T_a. The following result then implies that invariants are preserved by a chain of refinements.

Invariant preservation

Suppose $T_2 \sqsubseteq_\pi T_1$. Then

1. $T_3 \sqsubseteq_{\pi'} T_2$ implies $T_3 \sqsubseteq_{\pi\circ\pi'} T_1$, and
2. $reach(T_1) \subseteq J$ implies $\pi(reach(T_2)) \subseteq J$.

Development of a family of key agreement protocols

We derive a family of protocols, such as:

- $\mathsf{dh3}_{\mathsf{asym}}$: signed Diffie-Hellman
- $\mathsf{sk3}_{\mathsf{asym}}$: some modes of IKEv1 or SKEME

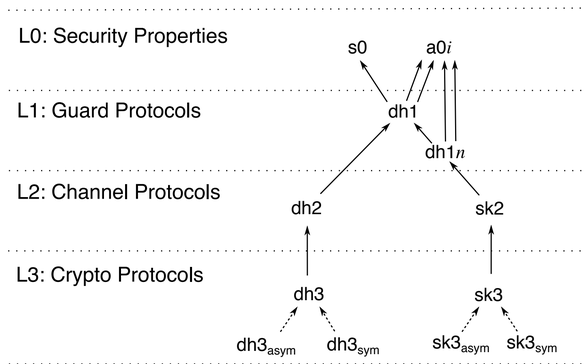

Figure 4: Refinement graph of our case study (→=refinement, ⇢=instantiation)

Compromising adversary

Following [1], we consider an arbitrary session of the protocol, and let the adversary dynamically compromise various resources of the participants:

- any agent's long-term key, except the participants of that session: *dynamically compromising* Dolev-Yao attacker.
- the long-term keys of the considered session's participants *after* that session is finished: we strengthen standard secrecy to *perfect forward secrecy*.
- session keys from other sessions: secrecy in this setting additionally guarantees *key independence*.

Equational theory

Our framework includes modelling of non-atomic messages and secrets on all levels, as required to express the Diffie-Hellman protocol, as well as support for equational reasoning by quotienting the set of messages by an equational theory.
We use here the equation associated with Diffie-Hellman exponentiation:

$$\mathsf{exp}(\mathsf{exp}(x, y), z) = \mathsf{exp}(\mathsf{exp}(x, z), y)$$

Modularity

At L3 we implement channels parametrically, under some assumptions that any concrete implementation must satisfy. This approach is modular: we can derive a parametric L3 protocol and then instantiate it in many different ways.
We provide in our work two instantiations based on symmetric and asymmetric cryptography, and other ones could be defined.

Contributions

1. A refinement-based approach to develop secure by construction protocols.
2. We substantially extend the scope of previous work [2] [3], by extending the classes of both protocols and properties supported, which required a major redesign of that work (equational theories, complex messages, compromising adversaries…);
3. Our approach is modular, as we prove protocols parametrically with respect to the cryptographic functions used to provide security guarantees;
4. Finally, we validate our approach by developing a family of Diffie-Hellman-based key agreement protocols. This shows it can be applied to real-life protocols.

Development statistics and discussion

We have entirely formalized this work in the proof assistant Isabelle/HOL (Table 1). Our development method is stepwise and systematical: global security properties are mapped to security guards on the exchanged information, then to actual network messages. We provide a sizeable infrastructure, which can be applied to other case studies. States and attacker events at all levels are also reusable. The protocol events follow a canonical structure.The refinement mappings are simple, the invariants are mostly canonical, with similar proofs for different protocols.

	theories	definitions	lemmas	lines
Infrastructure (refinement, modelling, L0)	(15 theories)	67	661	5500
Level 1	dh1, dh1n	44	117	1828
Diffie-Hellman	dh2, dh3	56	137	2071
SKEME/IKEv1	sk2, sk3	56	146	2468

Table 1: Specification and proof statistics (time in seconds)

Approach

We consider a 4-level hierarchy of increasingly concrete models:

Level 0 *Security properties.* Simple, protocol-independent specifications of secrecy and authentication, where security invariants are trivial to show.

Level 1 *Guard protocols. Runs* contain the state of a protocol role instance during its execution. They communicate by reading each other's memory. *Security guards* (= logical conditions) ensure secrecy or authenticity if needed. The attacker interacts loosely with the protocol: unguarded reads are non-deterministic, and he can learn any message that does not let him derive a protocol secret. (*e.g.* Fig. 1)

Level 2 *Channel protocols.* Runs communicate using communication channels with security properties, such as authentic or confidential channels. The intruder is closer to the protocol: he can access channels depending on their security properties and gains compromising abilities. (*e.g.* Fig. 2)

Level 3 *Cryptographic protocols.* We implement channel messages by cryptographic operations using long-term keys. This refinement is parametrized by implementation, subject to a set of assumptions. We can then obtain different protocol variants by instantiating the implementation in different ways. The attacker is a Dolev-Yao adversary with compromising abilities. (*e.g.* Fig. 3)

We then establish a succession of refinements, gradually concretizing the wanted L0 properties (here T_0) into a L3 model (here T_3):

$$T_3 \sqsubseteq_{\pi_{23}} T_2 \sqsubseteq_{\pi_{12}} T_1 \sqsubseteq_{\pi_{01}} T_0$$

The L3 protocol inherits the security properties through the refinements: it is secure by construction.

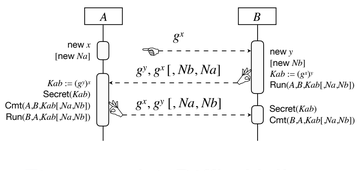

Figure 1: L1 Diffie-Hellman

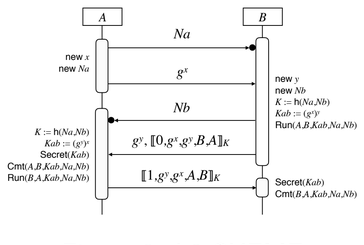

Figure 2: L2 SKEME

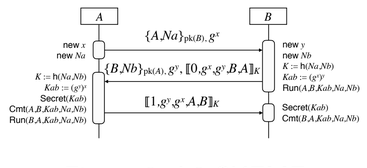

Figure 3: L3 SKEME

Future work

1. Additional forms of compromise, *e.g.* Key Compromise Impersonation
2. More general equational theories
3. More general channel implementations (*e.g.* probabilistic/hybrid encryption)
4. (Even) more automated refinement proofs

References

[1] D. A. Basin and C. Cremers.
Know your enemy: Compromising adversaries in protocol analysis.
ACM Trans. Inf. Syst. Secur., 17(2):7:1–7:31, 2014.

[2] C. Sprenger and D. A. Basin.
Developing security protocols by refinement.
In E. Al-Shaer, A. D. Keromytis, and V. Shmatikov, editors, *ACM Conference on Computer and Communications Security, CCS 2010*, pages 361–374. ACM, 2010.

[3] C. Sprenger and D. A. Basin.
Refining key establishment.
In S. Chong, editor, *IEEE Computer Security Foundations Symposium, CSF 2012*, pages 230–246. IEEE Computer Society, 2012.