Toward dynamic epistemic verification of zero-knowledge protocols

Cosimo Perini Brogi ⋆
IMT School for Advanced Studies Lucca

⋆ (j.w.w. Gabriele Costa)

Second Secure Software From First Principles Workshop
13 – 14 June 2024, Lucca

IMT SCUOLA ALTI STUDI LUCCA

Our goal

- Explore a new methodology based on the abstract semantics for Dynamic Epistemic Logic to analyse Zero Knowledge protocols, specified in a new protocol specification language (SPEC), quite simple.

Our goal

For today

- Here, we illustrate this DEL-verification approach to a specific new protocol, named Broken Key Protocol (BKP), verifying that the evolution of epistemic states along the protocol execution from the view-points of *each participant* (*honest* prover and verifier)

  satisfies:
  - ⇒ Zero-knowledge
  - ⇒ Proof of knowledge
  - ⇒ No repudiation

  expressed in the formal language of DEL.

More details in our conference paper:

- ✓ G. Costa, C. Perini Brogi. *Toward dynamic epistemic verification of zero-knowledge protocols*, in Proceedings of the Italian Conference on Cyber Security (ITASEC 2024), Salerno, Italy, April 8-11, 2024. (*To appear*).

Our goal

For today

- Here, we illustrate this DEL-verification approach to a specific new protocol, named Broken Key Protocol (BKP), verifying that the evolution of epistemic states along the protocol execution from the view-points of *each participant* (*honest* prover and verifier)

satisfies:
$\Rightarrow$ Zero-knowledge
$\Rightarrow$ Proof of knowledge
$\Rightarrow$ No repudiation

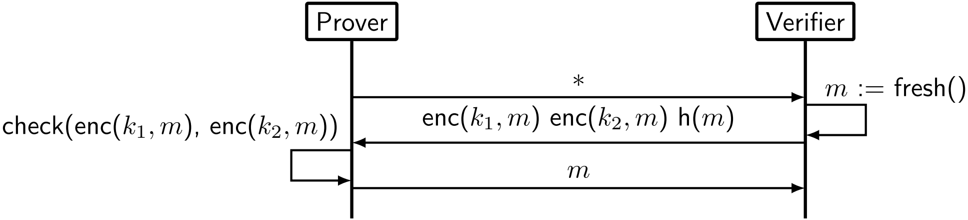
expressed in the formal language of DEL.

More details in our conference paper:

✓ G. Costa, C. Perini Brogi. *Toward dynamic epistemic verification of zero-knowledge protocols*, in Proceedings of the Italian Conference on Cyber Security (ITASEC 2024), Salerno, Italy, April 8-11, 2024. (*To appear*).

IMT SCUOLA ALTI STUDI LUCCA

Broken Key Protocol



Simple Protocol Epistemic Calculus

Statements

A *protocol statement* S is a term generated through the following grammar.

$$S ::= x := e \mid \rightarrowtriangle_A\colon e \mid \leftarrowtriangle_B\colon x \mid [g]S \mid S; S'$$

Structural Operational Semantics

$$\frac{\langle \sigma, S\rangle \longrightarrow \langle \sigma', S''\rangle}{\langle \sigma, S; S'\rangle \longrightarrow \langle \sigma', S''; S'\rangle}\ \text{(Seq 1)} \qquad \frac{\langle \sigma, S\rangle \longrightarrow \langle \sigma', \cdot\rangle}{\langle \sigma, S; S'\rangle \longrightarrow \langle \sigma', S'\rangle}\ \text{(Seq 2)}$$

$$\frac{[\![g]\!]_\sigma = \mathbf{1}}{\langle \sigma, [g]S\rangle \longrightarrow \langle \sigma, S\rangle}\ \text{(Cond 1)} \qquad \frac{[\![g]\!]_\sigma = \mathbf{0}}{\langle \sigma, [g]S\rangle \longrightarrow \skull}\ \text{(Cond 2)} \qquad \frac{[\![e]\!]_\sigma = v}{\langle \sigma, x := e\rangle \longrightarrow \langle \sigma[v/x], \cdot\rangle}\ \text{(Asgn)}$$

$$\frac{[\![e]\!]_\sigma = v}{\langle \sigma, \rightarrowtriangle_A\colon e\rangle \longrightarrow \langle \sigma, \cdot\rangle \uparrow_{A,v}}\ \text{(Send)} \qquad \frac{\langle \sigma, S\rangle \longrightarrow \langle \sigma', S''\rangle \uparrow_{A,v}}{\langle \sigma, S; S'\rangle \longrightarrow \langle \sigma', S''; S'\rangle \uparrow_{A,v}}\ \text{(Send-P)}$$

$$\frac{}{\langle \sigma, \leftarrowtriangle_B\colon x\rangle \longrightarrow \langle \sigma, \cdot\rangle \downarrow_{B,x}}\ \text{(Recv)} \qquad \frac{\langle \sigma, S\rangle \longrightarrow \langle \sigma', S''\rangle \downarrow_{B,x}}{\langle \sigma, S; S'\rangle \longrightarrow \langle \sigma', S''; S'\rangle \downarrow_{B,x}}\ \text{(Recv-P)}$$

SPEC-description of BKP

Honest prover

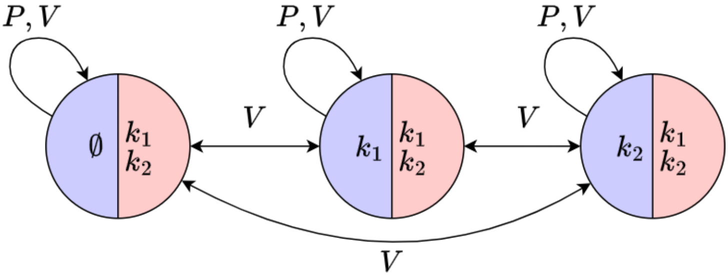
$$S_P \triangleq \quad \rightarrow_V : *; \leftarrow_V : x, y, z; [\mathtt{comp}(x, y)][z = \mathtt{h}(\mathtt{trydec}(k, x, y))] \rightarrow_V : \mathtt{trydec}(k, x, y)$$

Honest verifier

$$S_V \triangleq \quad \leftarrow_P : *; m := \mathtt{fresh}(); \rightarrow_P : \mathtt{enc}(k_1, m), \mathtt{enc}(k_2, m), \mathtt{h}(m); \leftarrow_P : x; [x = m]\mathtt{skip}$$

Dynamic epistemic logic

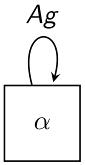
Models for states



Dynamic epistemic logic

Models for actions/events

The action model $\langle\!\langle \rightarrowtriangle_i \colon e \rangle\!\rangle_j$ for agent j sending e to agent i:

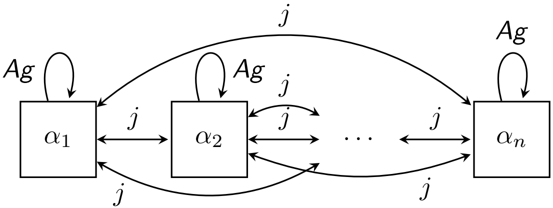


"Sending an expression is a public action that can be performed whenever the sender is able to construct the value of that expression; after the event, that value is stored in the local information of the receiver."

Dynamic epistemic logic

Models for actions/events

The action model $\langle\!\langle \leftarrowtriangle_i \colon x \rangle\!\rangle_j$ for agent j receiving values on variable x from agent i:

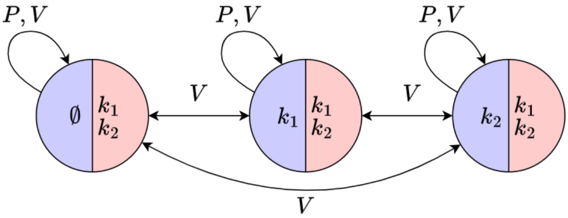


"Receiving information from the agent i as an equivalence class of sending statements from the same agent."

DEL-verification

Performing S_P

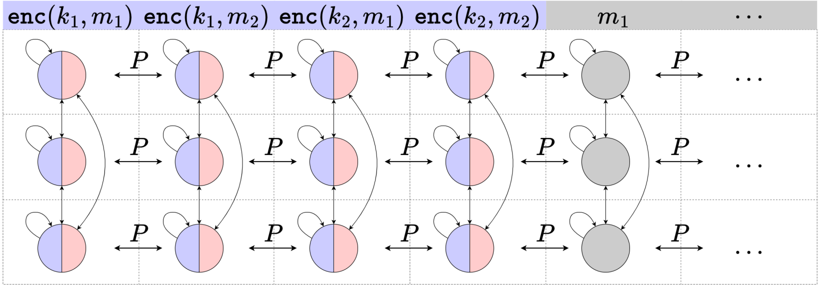
$$S_P \triangleq \twoheadrightarrow_V : *; \leftarrow_V : x, y, z; [\mathtt{comp}(x,y)][z = \mathtt{h}(\mathtt{trydec}(k,x,y))] \twoheadrightarrow_V : \mathtt{trydec}(k,x,y)$$



DEL-verification

Performing S_P

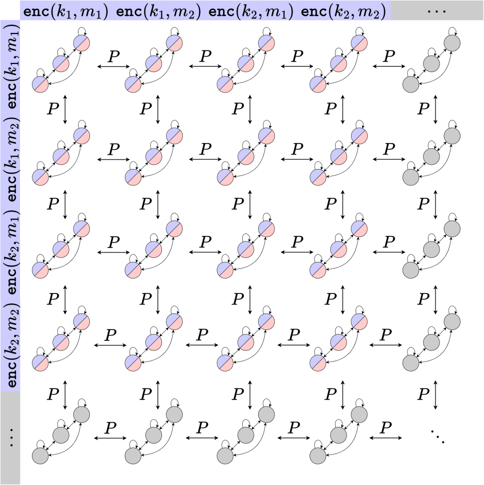
$$S_P \triangleq \rightarrow_V : *; \leftarrow_V : x, y, z; [\texttt{comp}(x, y)][z = \texttt{h}(\texttt{trydec}(k, x, y))] \rightarrow_V : \texttt{trydec}(k, x, y)$$



DEL-verification

Performing S_P

$$S_P \triangleq \rightarrowtriangle_V : *; \leftarrowtriangle_V : x, y, z; [\mathtt{comp}(x,y)][z = \mathtt{h}(\mathtt{trydec}(k,x,y))] \rightarrowtriangle_V : \mathtt{trydec}(k,x,y)$$



DEL-verification

Performing S_P

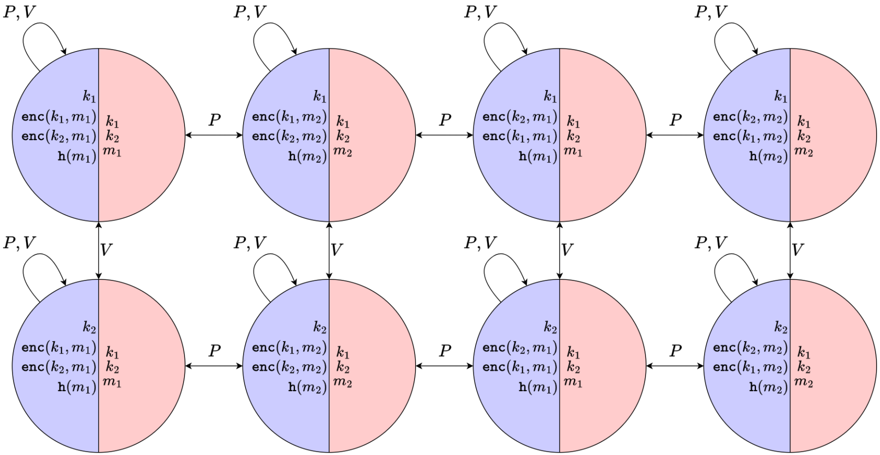
$$S_P \triangleq \multimap_V : *; \leftspoon_V : x, y, z; [\mathtt{comp}(x, y)][z = \mathtt{h}(\mathtt{trydec}(k, x, y))] \multimap_V : \mathtt{trydec}(k, x, y)$$

• • •

DEL-verification

Performing S_P

$$S_P \triangleq \rightarrowtriangle_V : *; \leftarrowtriangle_V : x, y, z; [\mathtt{comp}(x, y)][z = \mathtt{h}(\mathtt{trydec}(k, x, y))] \rightarrowtriangle_V : \mathtt{trydec}(k, x, y)$$



Zero knowledge: $\varphi_{\mathsf{ZK}} \triangleq \neg K_V(\mathsf{has}_P(k_1)) \wedge \neg K_V(\mathsf{has}_P(k_2))$

Proof of knowledge: $\varphi_{\mathsf{PoK}} \triangleq K_V(\mathsf{has}_P(k_1) \vee \mathsf{has}_P(k_2))$

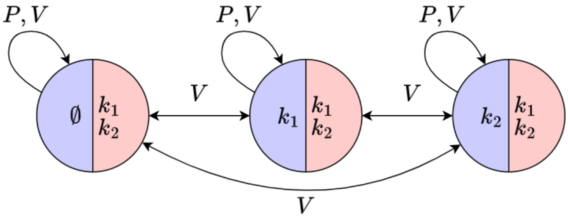
No repudiation: $\varphi_{\mathsf{NR}} \triangleq K_V(K_P(K_V(\mathsf{has}_P(k_1) \vee \mathsf{has}_P(k_2)))))$

IMT SCUOLA ALTI STUDI LUCCA

DEL-verification

Performing S_V

$$S_V \triangleq \leftarrow_P : *; m := \texttt{fresh}(); \rightarrow_P : \texttt{enc}(k_1, m), \texttt{enc}(k_2, m), \texttt{h}(m); \leftarrow_P : x; [x = m]\texttt{skip}$$



DEL-verification

Performing S_V

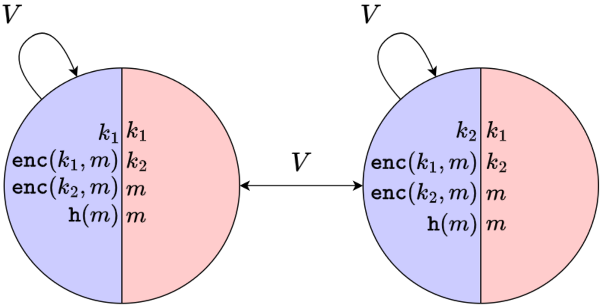
$$S_V \triangleq \leftarrow_P : *; m := \mathtt{fresh}(); \rightarrow_P : \mathtt{enc}(k_1, m), \mathtt{enc}(k_2, m), \mathtt{h}(m); \leftarrow_P : x; [x = m]\mathtt{skip}$$

$$\bullet \quad \bullet \quad \bullet$$

DEL-verification

Performing S_V

$$S_V \triangleq \leftarrow_P : *; m := \texttt{fresh}(); \rightarrow_P : \texttt{enc}(k_1, m), \texttt{enc}(k_2, m), \texttt{h}(m); \leftarrow_P : x; [x = m]\texttt{skip}$$



Proof of knowledge: $\varphi_{\mathsf{PoK}} \triangleq K_V(\mathsf{has}_P(k_1) \vee \mathsf{has}_P(k_2))$

Put in perspective

- Employ the capabilities and flexibility of non-classical logics, and, in particular, dynamic epistemic logic, in
  - formalising zero-knowledge scenarios and protocols;
  - abstracting the logical structure behind cryptographic and mathematical aspects of zero-knowledge interactions;
  - verifying security desiderata of zero-knowledge protocols.
- Store meta-theoretical results for the combination SPEC+DEL.
- Integrate existing models and automated tools for verification of zero-knowledge proofs with efficient and DEL-based modelling techniques (modulo some engineering adjustments).

Many thanks for listening!