Toward dynamic epistemic verification of zero-knowledge protocols *

Gabriele Costa **Cosimo Perini Brogi**
IMT School for Advanced Studies Lucca

* Work supported by the project SERICS – SWOPS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union NextGenerationEU

ITASEC24
08 – 12 April 2024

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Zero-Knowledge

Formal

Definition

Let us assume Turing machines as models for computation.
An interactive proof system with Turing machines (P, V) for a given language L is zero-knowledge if for any probabilistic polynomial time Turing machine verifier $\hat{V}$ there exists a probabilistic polynomial time Turing machine simulator S such that

$$\forall x \in L, z \in \{0,1\}^*, \mathtt{View}_{\hat{V}}[P(x) \leftrightarrow \hat{V}(x,z)] = S(x,z),$$

where $\mathtt{View}_{\hat{V}}[P(x) \leftrightarrow \hat{V}(x,z)]$ is a record of the interactions between $P(x)$ and $\hat{V}(x,z)$.

Zero-Knowledge

Comprehensible

Pandora and *Vulcan*

Suppose Pandora is tetrachromat: she can distinguish between the colours of two pebbles that would be identical to a trichromat.[a]
She wants to prove to a trichromat Vulcan that the two pebbles are *not* identical.
They proceed as follows:

P turns her back and V tosses a coin.
With probability 50% he leaves the pebbles as they are, and with probability 50% switches the right pebble with the left piece.
P needs to guess whether V switched the pebbles or not.

---

[a] That is: a "normal viewer".

John William Waterhouse, "Pandora" (Public domain, via Wikimedia Commons)

Guillaume Coustou the Younger, "Vulcan" (Public domain, via Wikimedia Commons)

"Roses", nnice/Flickr/CC BY 2.0

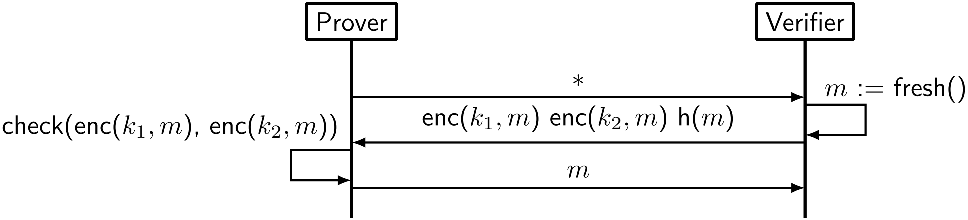
IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Our goal

This talk

- Introduce a new protocol, named Broken Key Protocol (BKP).
- Introduce a new protocol specification language (SPEC) to *describe* BKP.
- Introduce an abstract semantics – based on *relational models for dynamic epistemic logic* – for SPEC-statements.
- *Verify* that a single run of BKP satisfies three security desiderata – expressed in the formal language of DEL:
  $\Rightarrow$ Zero-knowledge
  $\Rightarrow$ Proof of knowledge
  $\Rightarrow$ No repudiation.

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

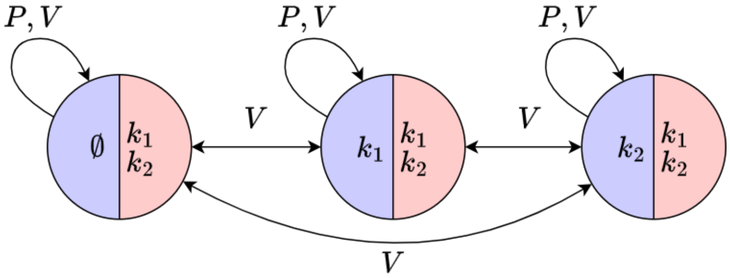
$$S_P \triangleq \quad \twoheadrightarrow_V : *; \twoheadleftarrow_V : x, y, z; [\mathtt{comp}(x, y)][z = \mathtt{h}(\mathtt{trydec}(k, x, y))] \twoheadrightarrow_V : \mathtt{trydec}(k, x, y)$$

Honest verifier

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

Dynamic epistemic logic

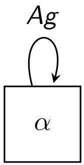
Models for states



Dynamic epistemic logic

Models for actions/events

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

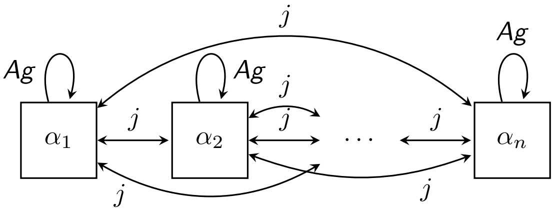


"Sending an expression is a public action that can be performed whenever the sender is able to construct the value of that expression; after the event, that value is stored in the local information of the receiver."

Dynamic epistemic logic

Models for actions/events

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

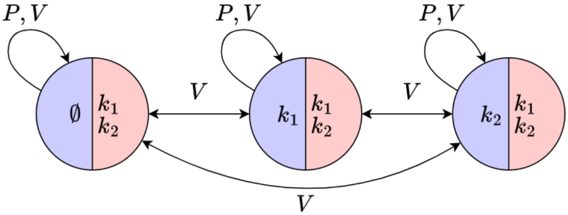


"Receiving information from the agent i as an equivalence class of sending statements from the same agent."

DEL-verification

Performing S_P

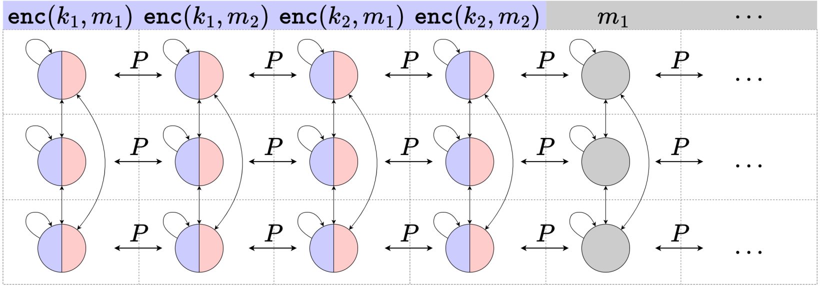
$$S_P \triangleq \rightarrowtriangle_V : *; \leftarrowtriangle_V : x, y, z; [\texttt{comp}(x, y)][z = \texttt{h}(\texttt{trydec}(k, x, y))] \rightarrowtriangle_V : \texttt{trydec}(k, x, y)$$



DEL-verification

Performing S_P

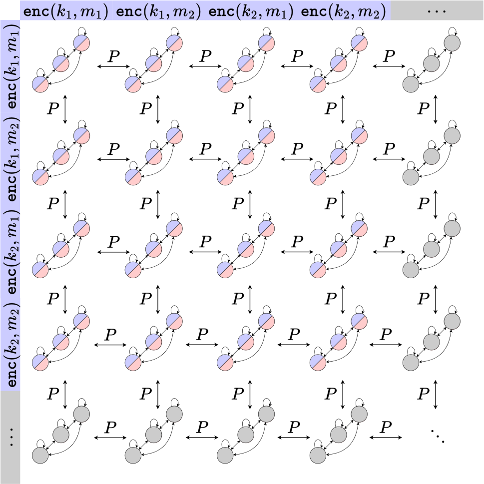
$$S_P \triangleq \rightarrow_V : *; \leftarrow_V : x, y, z; [\mathtt{comp}(x,y)][z = \mathtt{h}(\mathtt{trydec}(k,x,y))] \rightarrow_V : \mathtt{trydec}(k,x,y)$$



DEL-verification

Performing S_P

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



DEL-verification

Performing S_P

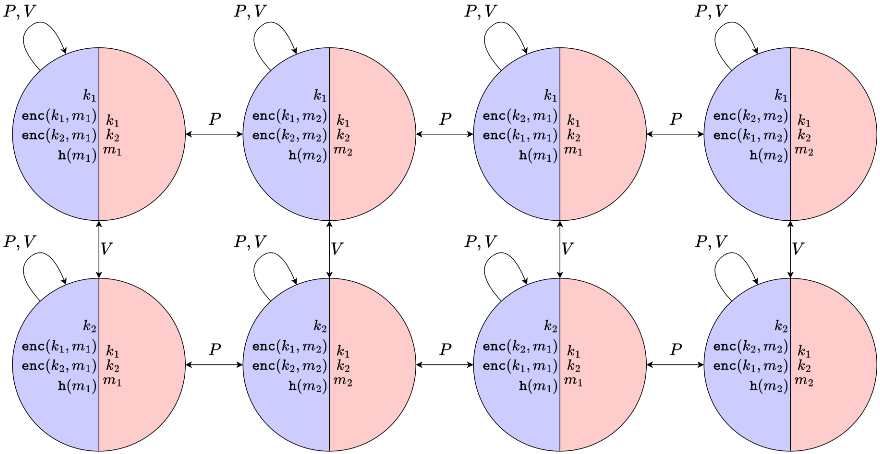
$$S_P \triangleq \rightarrow_V : *; \leftarrow_V : x, y, z; [\texttt{comp}(x,y)][z = \texttt{h}(\texttt{trydec}(k,x,y))] \rightarrow_V : \texttt{trydec}(k,x,y)$$

• • •

DEL-verification

Performing S_P

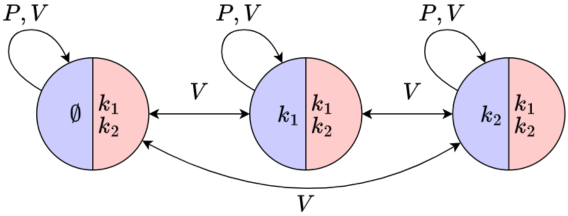
$$S_P \triangleq \rightarrow_V : *; \leftarrow_V : x, y, z; [\mathtt{comp}(x, y)][z = \mathtt{h}(\mathtt{trydec}(k, x, y))] \rightarrow_V : \mathtt{trydec}(k, x, y)$$



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

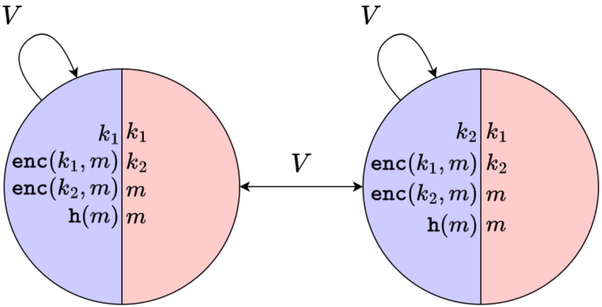
$$S_V \triangleq \leftarrow_P : *; m := \mathtt{fresh}(); \rightarrow_P : \mathtt{enc}(k_1, m), \mathtt{enc}(k_2, m), \mathtt{h}(m); \leftarrow_P : x; [x = m]\mathtt{skip}$$

• • •

DEL-verification

Performing S_V

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



Put in perspective

- We sketched a new methodology based on Dynamic Epistemic Logic to characterise Zero Knowledge protocols, specified in a simple formal language.
- We illustrated this DEL-verification approach to a specific new protocol (BKP), showing the evolution of epistemic states along the protocol execution from the view-points of *each participant* (prover and verifier).

That suggests that it is possible indeed to

◇ 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.

◇ 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!

IMT SCHOOL FOR ADVANCED STUDIES LUCCA

Put in perspective

- We sketched a new methodology based on Dynamic Epistemic Logic to characterise Zero Knowledge protocols, specified in a simple formal language.
- We illustrated this DEL-verification approach to a specific new protocol (BKP), showing the evolution of epistemic states along the protocol execution from the view-points of *each participant* (prover and verifier).

That suggests that it is possible indeed to

◇ 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.

◇ 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!