MTAT.07.003 Cryptology II

Sequential, Parallel and Universal Composability

Sven Laur
University of Tartu

Disclaimer

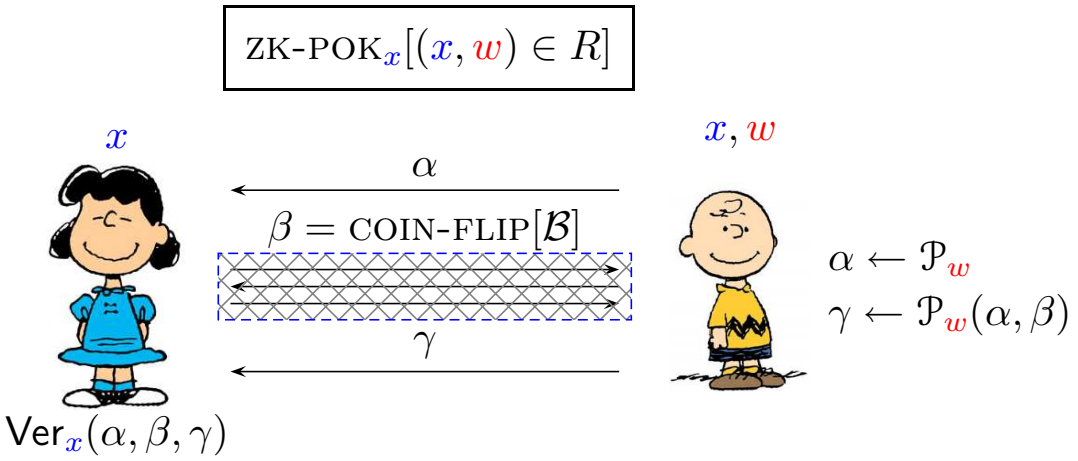
In the following slides, we sketch only the main principles of composability results and therefore omit all irrelevant technical details.

- ▷ We omit precise quantification of output distributions.
- ▷ We omit precise quantification of running times.
- ▷ We omit precise quantification of tolerated behaviour.
- ▷ We assume existence of a global synchronisation service.
- ▷ We assume that the adversary follows synchronisation signals.

Also note that we do not formalise the precise communication protocol between the challenger and the adversary. The corresponding formulation would take an entire lecture and would not give additional insight.

What is Composability?
Why Is It Useful?

An illustrative example



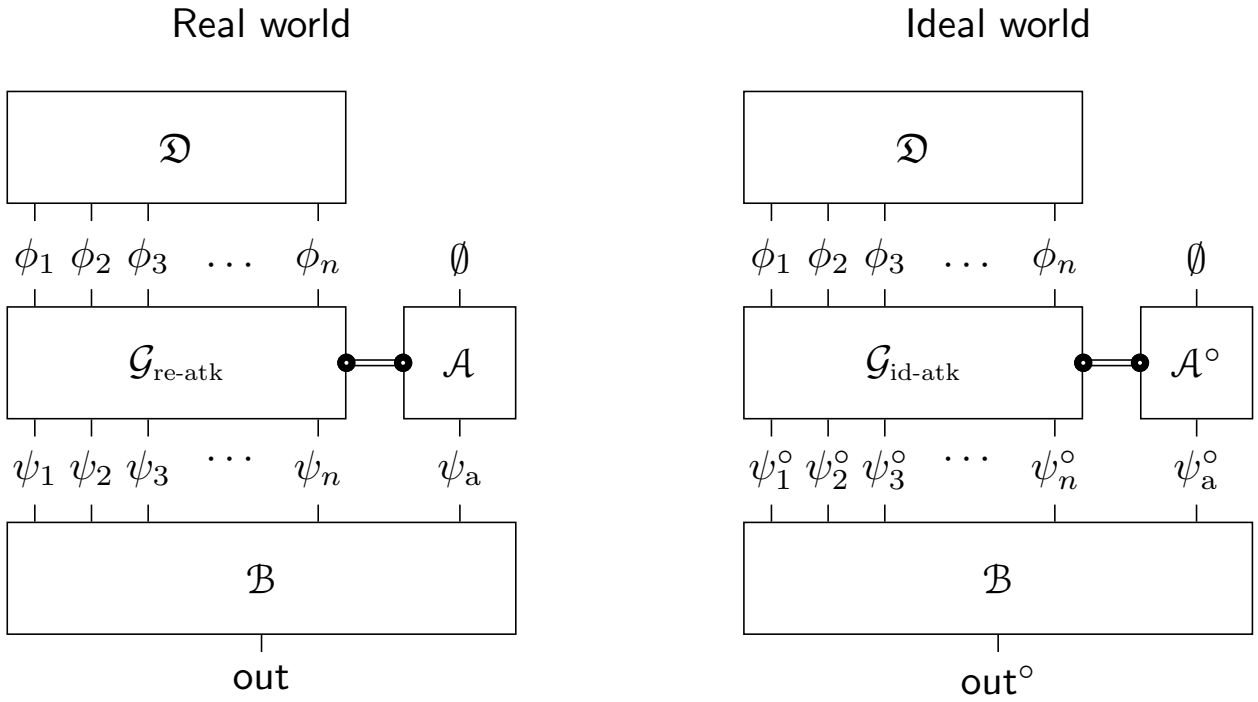
A zero-knowledge proof of knowledge can be constructed from a sigma protocol and a coin-flipping protocol. However, there are some issues.

- ▷ None of the protocols is not run in the stand-alone setting.
- ▷ We need several instances of Blum protocol to generate enough coins.

Hence, we formally need to reprove the security of the zero-knowledge proof from scratch. The latter is tedious and requires *unnecessary* mental effort.

Stand-alone Security And Beyond

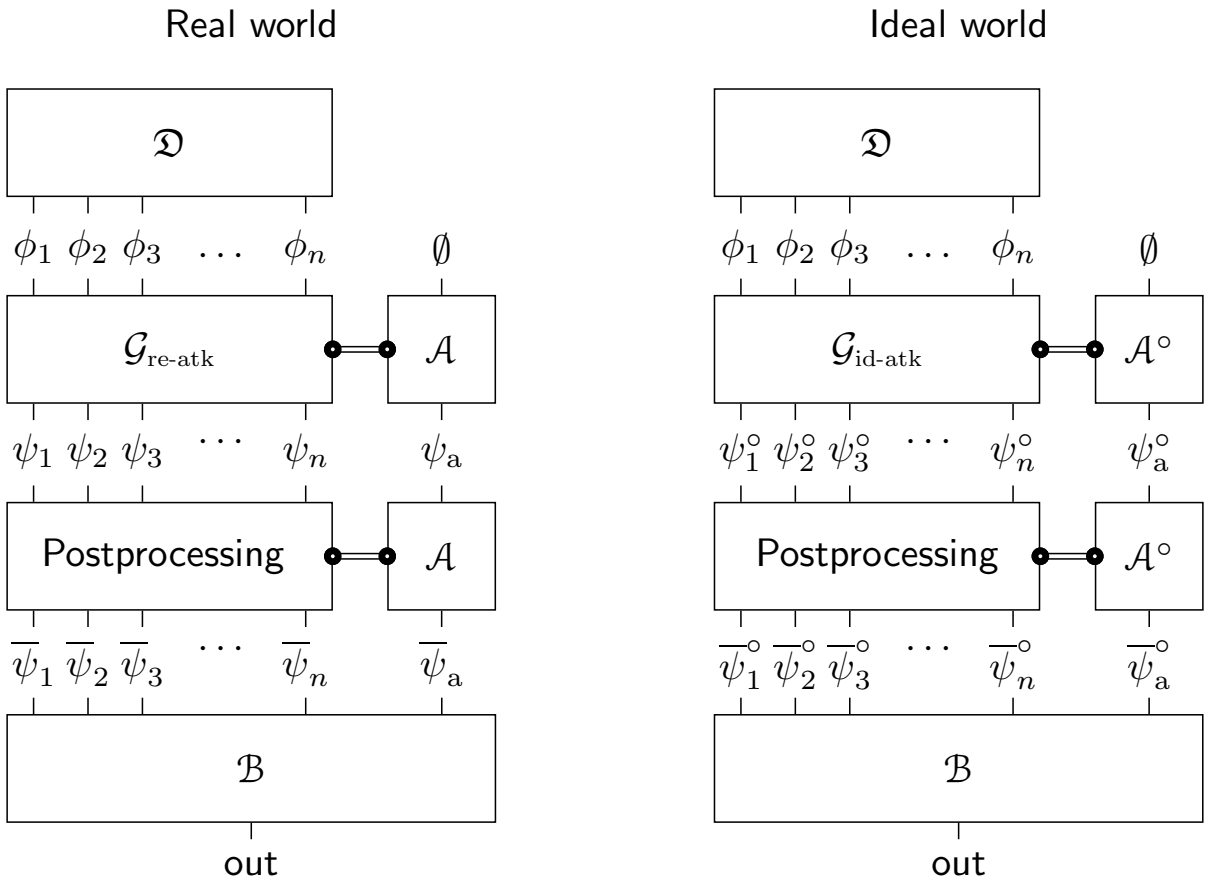
Pictorial definition of stand-alone security



A protocol is secure in the stand-alone model if there exists a mapping $\mathcal{A}, \mathcal{B} \mapsto \mathcal{A}^\circ$ such that for any input distribution $\mathfrak{D}$:

$$|\Pr[\text{out} = 1] - \Pr[\text{out}^\circ = 1]| \leq \varepsilon \ .$$

Security in post-processing contexts



We explicitly assume that everybody knows when post-processing starts.

Obvious conclusion

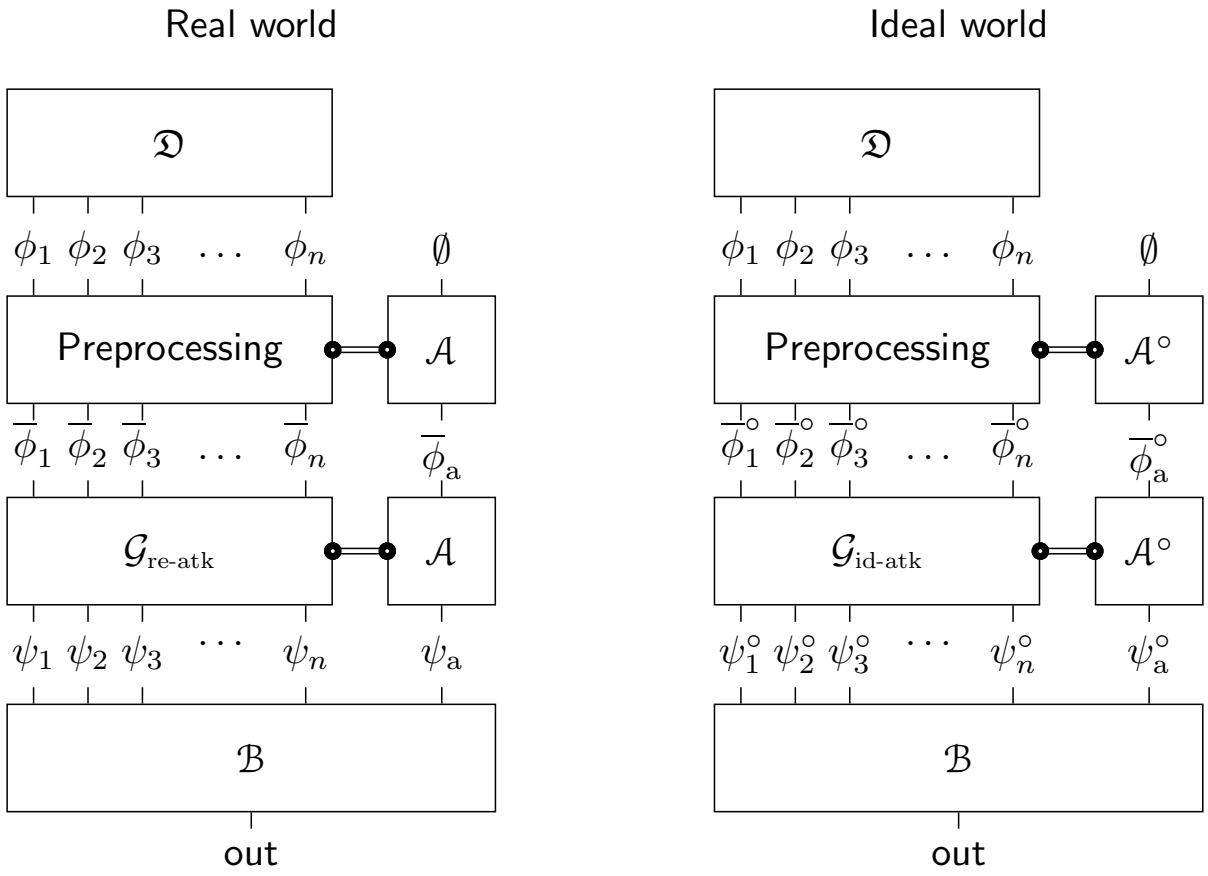
Theorem. Stand-alone security is preserved in a post-processing context if all parties are guaranteed to start post-processing phase after the protocol.

Proof

- ▷ Note that we can view the post-processing phase together with the predicate $\mathcal{B}(\cdot)$ as a new predicate $\mathcal{B}'(\cdot)$.
- ▷ For the formal proof, one must show that the outcome of the post-processing phase does depends only on the outputs ψ.
- ▷ The latter can be technically difficult if there is no global synchronisation service—we have to prove that timings do not change the outcome.

□

Security in pre-processing contexts



We explicitly assume that everybody knows when pre-processing starts.

Obvious conclusions

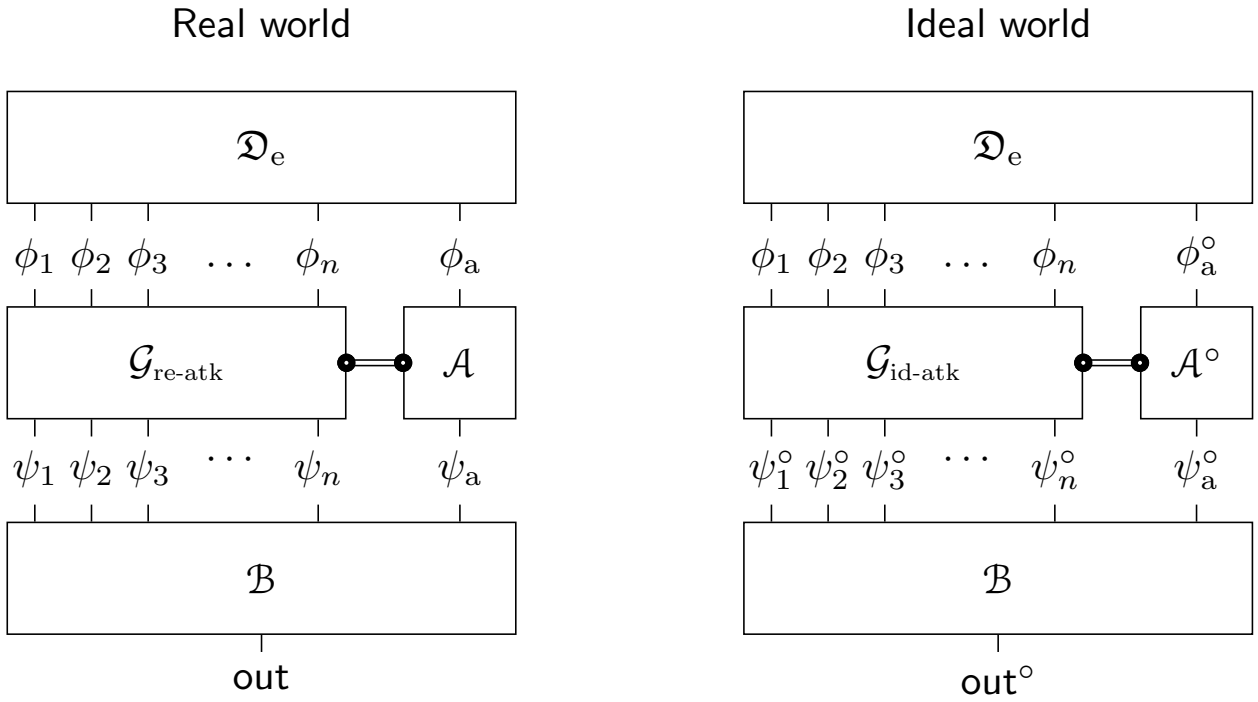
Trivial observation 1. If the ideal world adversary $\mathcal{A}^\circ$ behaves identically to $\mathcal{A}$ in the pre-processing phase, then the distributions $\overline{\phi}_{\mathrm{e}}$ and $\overline{\phi}^\circ_{\mathrm{e}}$ coincide.

Trivial observation 2. A real world adversary can use the input $\overline{\phi}_{\mathrm{a}}$ for tuning the attack against protocol. Consequently, the stand-alone security is *formally* not sufficient for guaranteeing security in pre-processing contexts.

Clarifying remark 1. If the adversary is static, then the input $\overline{\phi}_{\mathrm{a}}$ can be combined with the input $\overline{\phi}_i$ of a corrupted participant and thus we can *prove* that stand-alone security is preserved in pre-processing contexts.

Clarifying remark 2. For dynamic corruption model, there are explicit counterexamples of pre-processing contexts that do not preserve security.

Strong stand-alone security



A protocol is secure in the stand-alone model if there exists a mapping $\mathcal{A}, \mathcal{B} \mapsto \mathcal{A}^{\circ}$ such that for any extended input distribution $\mathfrak{D}_{\mathrm{e}}$:

$$\left|\Pr\left[\mathsf{out} = 1\right] - \Pr\left[\mathsf{out}^{\circ} = 1\right]\right| \leq \varepsilon \ .$$

Obvious conclusion

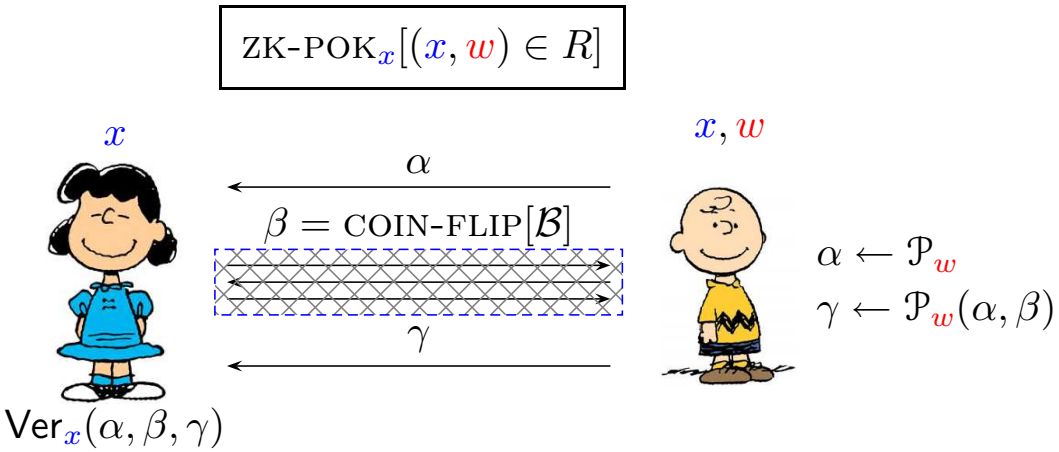
Theorem. Strong stand-alone security is preserved in a pre-processing context if pre-processing phase is always ended before the protocol.

Proof

- ▷ If the ideal world adversary $\mathcal{A}^\circ$ behaves identically to $\mathcal{A}$ in the pre-processing phase, then the distributions $\overline{\phi}$ and ϕ° coincide.
- ▷ Note that we can view the output $\overline{\phi}$ of a pre-processing phase as an extended input distribution $\mathfrak{D}_\mathrm{e}$ and then use the mapping $\mathcal{A}, \mathcal{B} \mapsto \mathcal{A}^\circ$ to define the second stage of the ideal world adversary.
- ▷ For the formal proof, we must assume that the decomposition is well defined and that the outcome depends only on the state $\overline{\phi}$.
- ▷ The latter can be technically difficult if there is no global synchronisation service—we have to prove that timings do not change the outcome.

□

An illustrating example



The coin-flipping protocol is executed in a computational context where

- ◇ the transfer of a message α forms a pre-processing context
- ◇ the transfer of a message γ forms a post-processing context
- ◇ pre- and post-processing context are clearly separated form the protocol.

Hence, (strong) stand-alone security of the coin-flipping protocol is sufficient for stand-alone security of the zero-knowledge protocol.

Sequential Composability

Central composability result

Theorem. Strong stand-alone security is preserved in all contexts, where pre- and post-processing phases are clearly separated from the protocol and no side-computations are carried out during the protocol execution.

Corollary. Strong stand-alone security guarantees sequential composability.

Usage restrictions. Strong stand-alone security is sufficient if honest participants are willing to accept the following constraints:

- ▷ They use these protocols in a black-box manner.
- ▷ They use separation signals for pre- and post-processing phases.
- ▷ They stop all side-computations during the execution of protocols.
- ▷ They consider all parties that violate these constrains as corrupted.

An explicit example

If we execute the Blum protocol π sequentially ℓ times, then we can also stack simulators sequentially to get the ideal world adversary.

$\mathcal{G}_{\text{real}}^{\mathcal{P}_1^*}$

$(\phi_1, \phi_2) \leftarrow \mathfrak{D}$

Run π to get (ψ_1^1, ψ_2^1)

$(\phi_1, \phi_2) \leftarrow (\psi_1^1, \psi_2^1)$

Run π to get (ψ_1^2, ψ_2^2)

$\ldots$

return $(\psi_1^\ell, \psi_2^\ell)$

$\mathcal{G}_{\text{ideal}}^{(\mathcal{S}^*)^{\mathcal{P}_1^*}}$

$(\phi_1, \phi_2) \leftarrow \mathfrak{D}$

Use $\mathcal{S}$ to get (ψ_1^1, ψ_2^1)

$(\phi_1, \phi_2) \leftarrow (\psi_1^1, \psi_2^1)$

Use $\mathcal{S}$ to get (ψ_1^2, ψ_2^2)

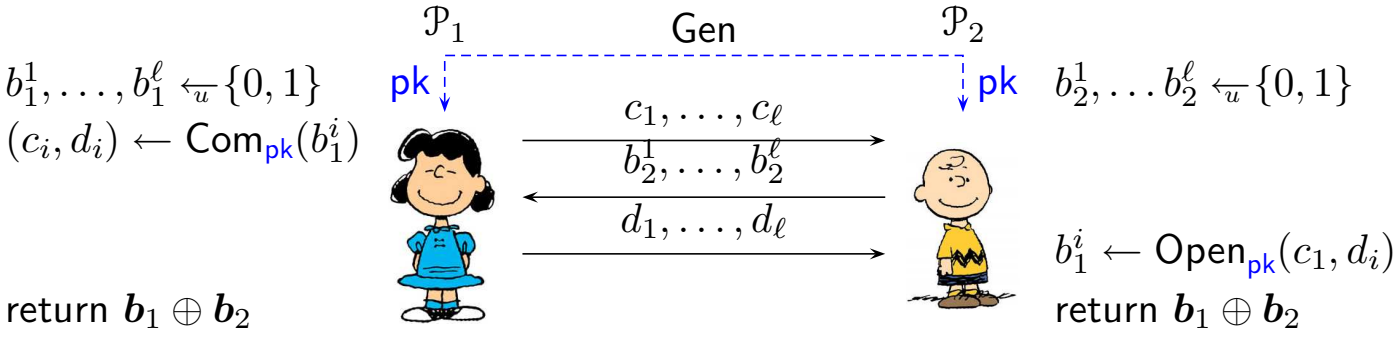
$\ldots$

return $(\psi_1^\ell, \psi_2^\ell)$

The final difference is a sum of individual differences.

Parallel composability

An interesting example



The simulation of this protocol is significantly more complex, since a single change in the vector $c_1, \ldots, c_n$ can alter any of the values $b_2^1, \ldots, b_2^\ell$.

- ▷ The number of potential replies $b_2^1, \ldots b_2^\ell$ grows exponentially wrt ℓ.
- ▷ We cannot one-by-one alter values $c_1, \ldots, c_\ell$ to get the correct output.
- ▷ Classical simulation strategies have exponential time-complexity wrt ℓ.

Central impossibility result

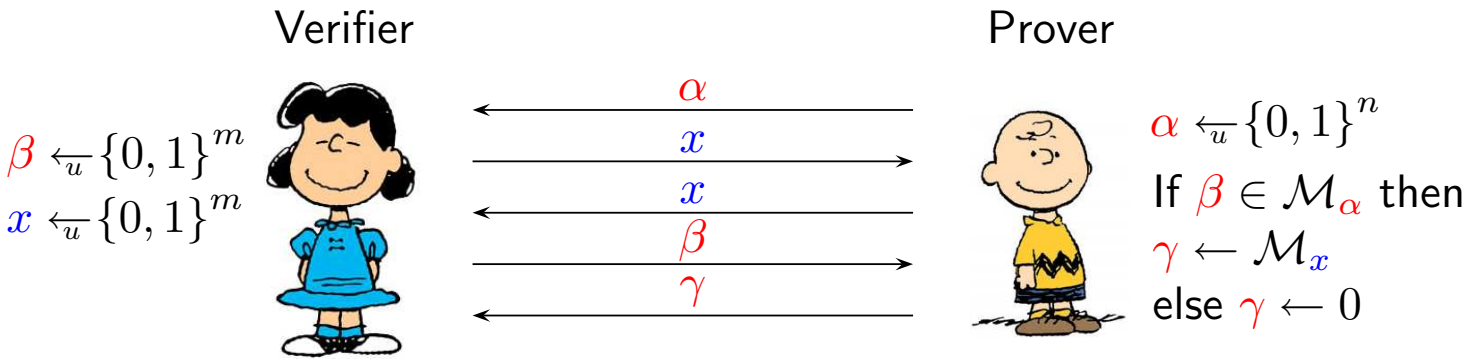
Theorem. There exists zero-knowledge proofs that are secure in the strong stand-alone security model, but their parallel composition is insecure.

Corollary. Strong stand-alone security is *insufficient* for parallel composability.

Further remarks:

- ▷ This result was first proved by Goldreich and Krawczyk in 1990.
- ▷ This result does not imply that we need protocols with stronger security guarantees if we use parallel composition in our construction.
- ▷ Instead, it just shows that there is *no general strategy* for constructing a new simulator from individual simulators of sub-protocols.

The first step towards a counter-example

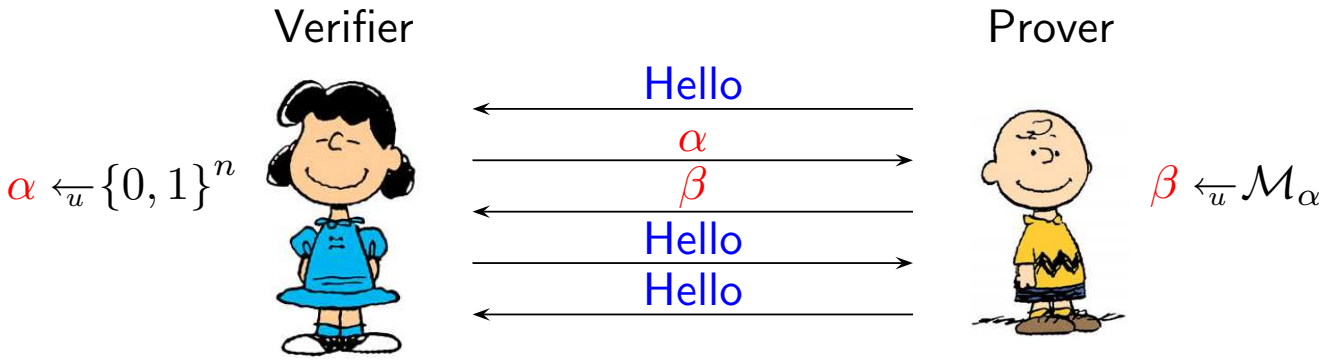


Let $\{\mathcal{M}_\alpha\}_{\alpha\in\{0,1\}^n}$ be a family of (t,ε)-unsamplable sets, i.e., no t-time adversary can find $y \in \mathcal{M}_\alpha$ with probability more than ε if $\alpha \xleftarrow{u} \{0,1\}^n$.

Such a family of sets can be constructed from (t,ε)-one-way function $f : \{0,1\}^n \to \{0,1\}^\ell$ by defining $\mathcal{M}_\alpha = \{x \in \{0,1\}^n : f(x) = f(\alpha)\}$.

For obvious reasons, this protocol is simulatable for all t-time verifiers $\mathcal{V}_*$, since $\Pr[\gamma \neq 0] \leq \varepsilon$ and we can simulate all other messages.

The second step towards a counter-example

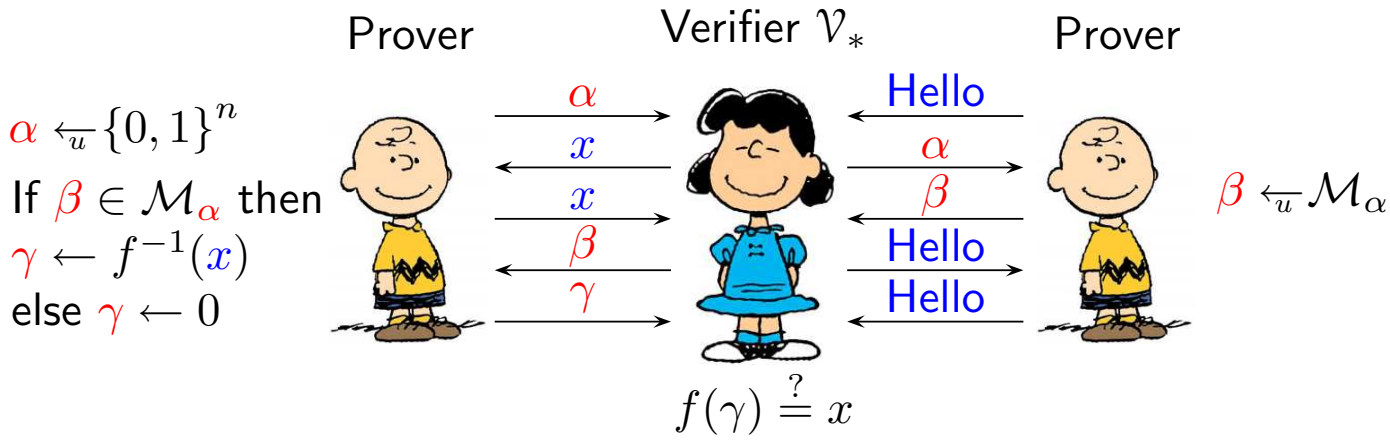


Let $\{\mathcal{M}_\alpha\}_{\alpha\in\{0,1\}^n}$ be a family of (t,ε)-pseudorandom sets, i.e., no t-time adversary can distinguish distributions $x \leftarrow_u \mathcal{M}_\alpha$ and $x \leftarrow_u \{0,1\}^n$ with an advantage more than ε for any index $\alpha \in \{0,1\}^n$

With some technical effort it is possible to construct a family of sets $\{\mathcal{M}_\alpha\}$ that is both (t,ε)-unsamplable and (t,ε)-pseudorandom.

Hence, if a simulator $\mathcal{S}$ replaces β by a uniformly chosen $\hat{\beta} \in \{0,1\}^n$, then the output of $\mathcal{S}^{\mathcal{V}_*}$ remains still computationally close to the output of $\mathcal{V}_*^{\mathcal{P}}$.

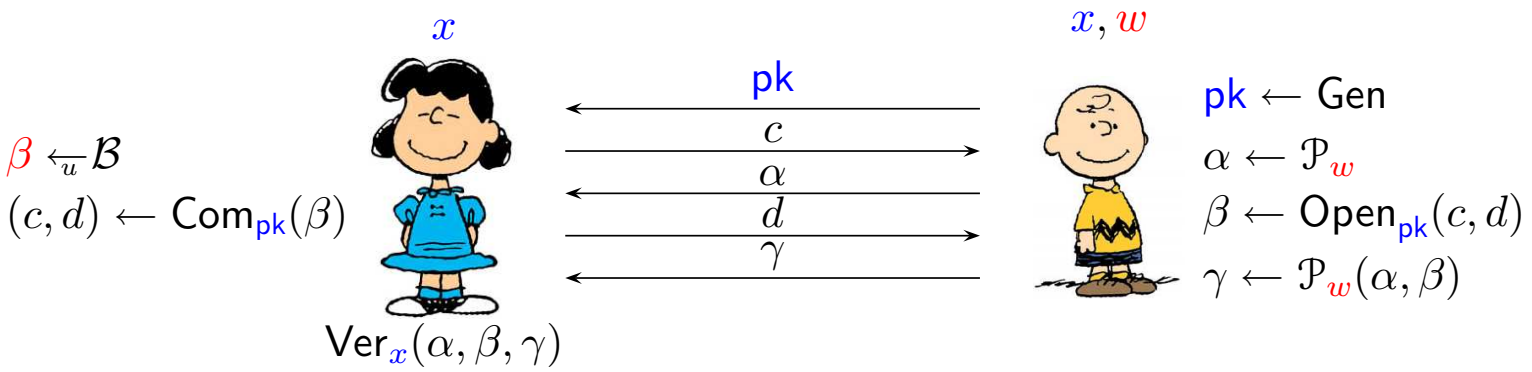
Insecurity of parallel composition



A malicious verifier $\mathcal{V}_*$ can cleverly reuse messages to allure the prover $\mathcal{P}$ to reveal the value of $f^{-1}(x)$ that is infeasible to compute otherwise.

More precisely, the value β is pseudorandom in the stand-alone setting, whereas in the parallel execution context the prover $\mathcal{P}$ provides a trapdoor mechanism to distinguish β from uniform distribution.

Zero-knowledge can be parallelly composable



We can run the strengthened sigma protocols depicted above in parallel without rapid decrease in the security level.

- ▷ A parallel execution of first two messages can be viewed as a list commitment $c_1, \ldots, c_n$ to the challenges $\beta_1, \ldots, \beta_n$.
- ▷ The following messages correspond to sigma protocols $\pi_1, \ldots, \pi_n$.
- ▷ Since parallel composition of sigma protocol is also a sigma protocol, the compound protocol is also a strengthened sigma protocol.

Detailed analysis

If we gradually convert the malicious verifier $\mathcal{V}_*$ to ideal world verifier $\mathcal{V}_\circ$ by replacing first the protocol by a dedicated black-box simulator $\mathcal{S}_1$ and then the second protocol by $\mathcal{S}_2$ and so on we get a truly inefficient simulator.

- ▷ For fixed simulator error ε, the knowledge extractor $\mathcal{K}_1$ in the simulator $\mathcal{S}_1$ increases the running time by the factor $\frac{1}{\varepsilon}$.
- ▷ The second simulator also increases the running time by the factor $\frac{1}{\varepsilon}$.
- ▷ Hence, the summary slowdown for the iterative simulator is exponential in the number of protocols that are run in parallel.

On the other hand, if we construct a simulator directly for the final compound protocol, then the slowdown factor remains $\Theta\left(\frac{1}{\varepsilon}\right)$.

Further comments

Note that the counterexample given by Goldreich and Krawczyk is artificial, since the parallel execution of zero-knowledge protocols creates a context, where a malicious verifier can use a prover as a dedicated distinguisher.

- ▷ More formally, these protocols have correlated message spaces and thus messages are meaningful even out of their protocol scope.
- ▷ Most protocols can be designed so that the adversary cannot correlate message spaces of different protocols.
- ▷ As a result, the counterexample is non-convincing and one should provide a more natural counterexample.

Universal composability

Functional definition

Let $\varrho\langle\cdot\rangle$ be a computational context that uses protocol π in a black-box manner and let π° be the corresponding ideal functionality. Then we can observe the security properties of two compound protocols

$$\varrho\langle\pi\rangle \qquad \text{and} \qquad \varrho\langle\pi^\circ\rangle \ .$$

More precisely, we must fix an input distribution $\mathfrak{D}_{\mathrm{e}}$ and a security goal $\mathcal{B}(\cdot)$ to define ideal and real world games $\mathcal{G}_{\mathrm{real}}$ and $\mathcal{G}_{\mathrm{ideal}}$.

A protocol is universally composable if for all relevant context $\varrho\langle\cdot\rangle$ and security goals $\mathcal{B}(\cdot)$, there exists a mapping $\mathcal{A}, \varrho\langle\cdot\rangle, \mathcal{B} \mapsto \mathcal{A}^\circ$ such that for any input distribution $\mathfrak{D}_{\mathrm{e}}$:

$$|\Pr[\mathsf{out} = 1] - \Pr[\mathsf{out}^\circ = 1]| \leq \varepsilon \ .$$

Security criterion

Theorem. Let us consider only contexts $\varrho\langle\cdot\rangle$, where the beginning and the end of a protocol is clearly separated form other computations. Then a protocol is universally composable iff there exists a non-rewinding black-box interface $\mathcal{I}\langle\cdot\rangle$ that converts any real world adversary $\mathcal{A}$ to an ideal world adversary $\mathcal{I}\langle\mathcal{A}\rangle$ such that for any input distribution $\mathfrak{D}_{\mathrm{e}}$:

$$|\Pr[\mathsf{out} = 1] - \Pr[\mathsf{out}^{\circ} = 1]| \leq \varepsilon \ .$$

Necessity:

- ▷ Consider a silent partner attack, where adversary $\mathcal{A}_*$ follows orders from a non-corruptible party $\mathcal{P}_i$ that treats its input ϕ_i as a program code.
- ▷ Since the adversary $\mathcal{A}_*$ knows nothing about the future attack steps and the correspondence cannot depend on ϕ_i, the corresponding ideal world adversary must be able to do on-line simulation.

Triviality theorem

Theorem. A protocol without a trusted setup and honest majority cannot preserve privacy of inputs and be universally composable at the same time.

Proof

- ▷ As the protocol is universally composable, there exists an non-rewinding interface between the real world adversary and the trusted third party.
- ▷ The interface is essentially an attack description, since it shows how to extract inputs of the corrupted parties by simulating the behaviour of other honest nodes.
- ▷ As the adversary can corrupt the same set of nodes in the real world and mimic the interface, he or she can indeed violate input privacy and control the outputs of honest parties.

□

Obvious conclusions

Observation 1. A trusted setup phase is essential to achieve universal composability in settings where honest parties are in minority. Input-private two-party protocols without a setup phase cannot be universally composable.

Observation 2. It is straightforward to achieve universal composability by using trusted setup such as the common reference string model. However, it is reasonable to minimise the number of system-wide parameters.

Observation 3. Universally composable protocols do not have to preserve security if they share the same trusted setup.

- ▷ We need more fine-grained security definitions that make the analysis of such protocols tractable.
- ▷ Hence, we must extent the classical definition of universal composability.