MTAT.07.003 Cryptology II

Design of Complex Protocols

Sven Laur
University of Tartu

Taxonomy of Security Goals

The devil is in the details

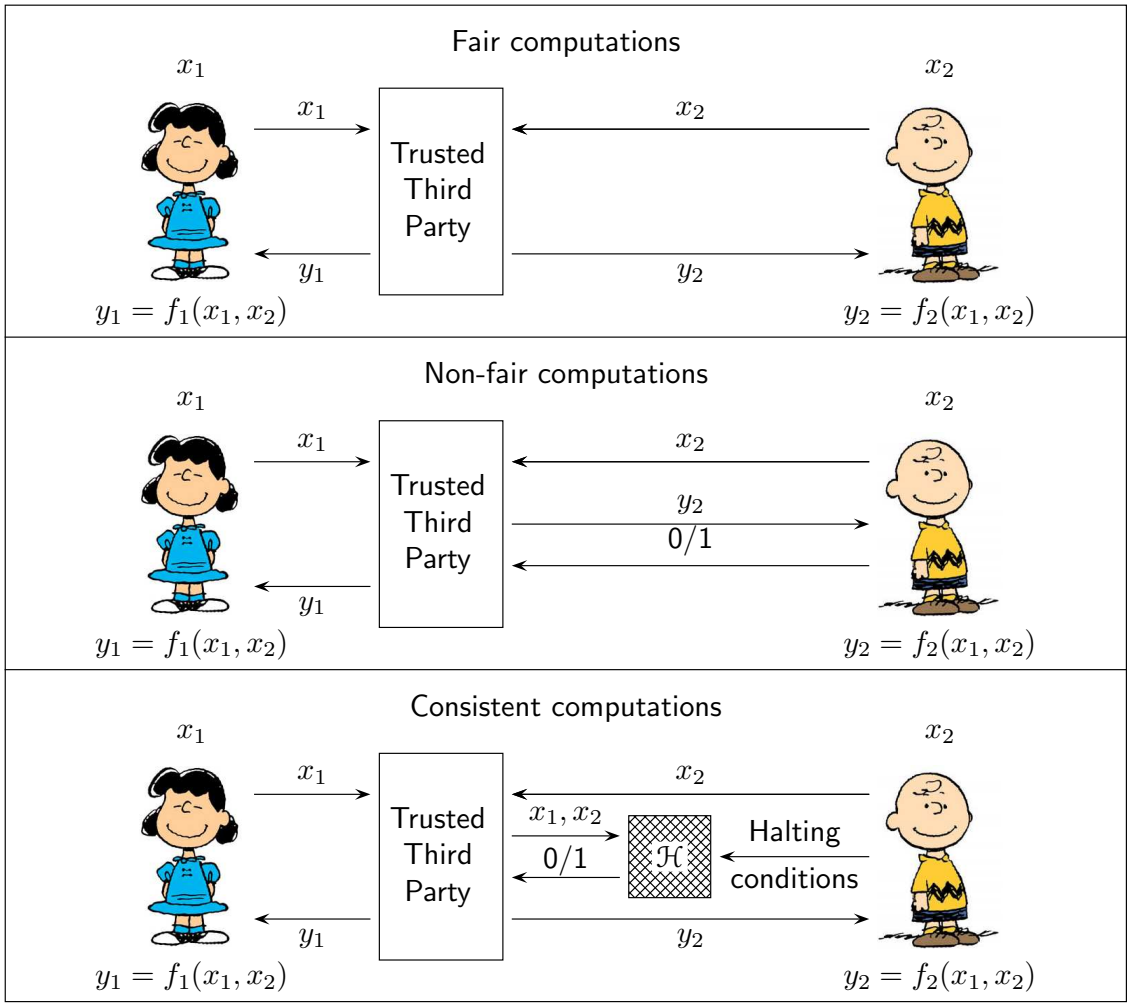
The ideal vs real world principle gives arise to *radically* different security depending on the exact properties we require from the mapping $\mathcal{A}, \mathcal{B} \overset{\mathcal{S}}{\mapsto} \mathcal{A}^{\circ}$.

- **Quantitative properties:**
  - How does $\varepsilon = \varepsilon(t_{\mathrm{re}}, t_{\mathrm{pred}})$ behave?
  - How comparable is $t_{\mathrm{id}} = t_{\mathrm{id}}(t_{\mathrm{re}}, t_{\mathrm{pred}})$ with t_{re}?
  - How large values of t_{pred} lead to reasonable values of t_{id} and ε?
- **Qualitative properties:**
  - What is the tolerated adversarial behaviour?
  - Which model is used for idealised computations?
  - What type of predicates $\mathcal{B}(\cdot)$ are considered relevant?
  - In which context the protocol is executed?

Taxonomy of security levels

	Input privacy	Output consistency	Complete security
Malicious behaviour	Privacy of inputs is guaranteed even against malicious behaviour.	Malicious behaviour that alters outputs is detectable. Public complaints reveal information about inputs.	Malicious behaviour is detectable. Public complaints reveal no information about inputs.
Semi-honest behaviour	Privacy of inputs is guaranteed. Privacy of outputs is not guaranteed.	Protocols implement the desired functionality.	Privacy of inputs and outputs is guaranteed. Protocols implement the desired functionality.

Taxonomy of ideal world models



Taxonomy of corruption models

Static corruption model. An adversary must choose which participants to corrupt before the execution of the computations.

- ▷ This model is adequate for small well-protected networks.

Dynamic corruption model. An adversary can choose which participants to corrupt during the execution of the computations.

- ▷ This model is adequate for larger networks provided that the protocol is executed in a relatively short time-frame.

Mobile corruption model. An adversary can corrupt participants and withdraw from them during the execution of the computations.

- ▷ This model is adequate for protocols that have a long life span.

Taxonomy of tolerated contexts (1/2)

Stand-alone security. Security of a protocol is considered in the setting where no other computations are carried out.

- ▷ The stand-alone setting is simple enough to analyse in practice.
- ▷ Security in more complex settings is often characterised through the stand-alone model by imposing additional constraints.

Sequential composability. A protocol is sequentially composable if it preserves security in the computational contexts where

- ▷ some computations are done before the protocol,
- ▷ some computations will be done after the protocol,
- ▷ no side computations are carried out during the protocol,
- ▷ the beginning and end of the protocol is clearly detectable for all parties.

Taxonomy of tolerated contexts (2/2)

Composability wrt specific contexts. In many cases, it is advantageous to use sub-protocols in order to implement complex functionality.

- ▷ The compound protocol is secure only if the sub-protocols preserve their security in the context induced by the compound protocol.
- ▷ As a result, we must either prove security directly for the compound protocol or show that the security is preserved in this context.

As an example, consider parallel self-composability of zero-knowledge proofs.

Universal composability. A protocol is universally composable if it preserves security in all contexts that use protocol as a black box:

- ▷ the context provides inputs and fresh randomness,
- ▷ the context uses only the outputs of the protocol.

Taxonomy of other taxonomies

A *communication model* specifies how messages are transferred between participants and how an adversary can influence message transmission.

An *execution and timing model* specifies how the participants carry out their computations and whether the adversary can use timing information.

Setup assumptions characterise how system wide parameters are generated. There are a wide spectrum of setup assumptions:

- ▷ plain model
- ▷ common reference string model
- ▷ public key infrastructure model

Finally, there is wide spectrum of models that describe possible *side-channel attacks*, e.g., power analysis, spectral analysis, hardware tampering, etc.

Beyound Universal Composability

Things we did not covered this year

How to minimise parts of a protocol that has to be run in isolation?

- **Bootstrapping principle**. We use common reference string model to create universally composable coin-flipping protocol and then use its output as an input to itself. As a result, we can amplify the length of common reference string by any polynomial factor.
- **Shared setup.** We use the same trusted setup-phase for many protocols. As this might compromise protocols and turn them *insecure*, we have to reprove the security of such compound protocols.
  - This can be done directly as in the Bellare-Rogaway model.
  - This can be done by refining the restrictions on the security proof so that protocol remains secure even in the case of shared setup. For instance, all security definitions of encryption algorithms are formalised such way that ciphertext remains secure even if an adversary has seen a ciphertext generated with the same key.