

# Computational Soundness of Symbolic Zero Knowledge Proofs

Esfandiar Mohammadi

Master Seminar

Information Security and Cryptography Group

Max-Planck Institute for Software Systems

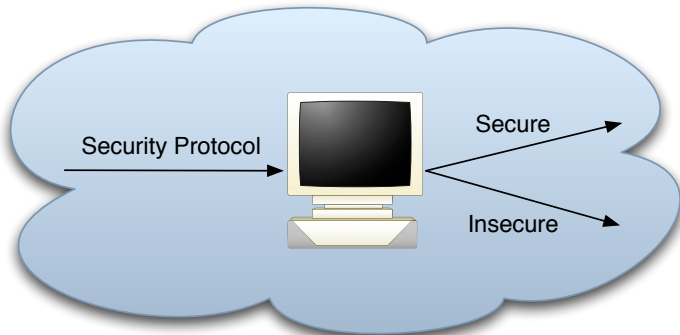
Saarland University

Advisors: Prof. Dr. Michael Backes and Dr. Dominique Unruh

June the 17th, 2008

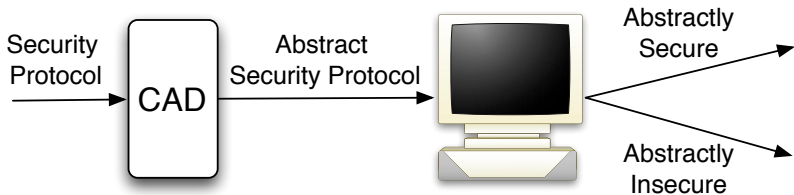
# Motivation

- Security Protocols are hard to verify by hand due to their complexity
- Automated verification is desirable



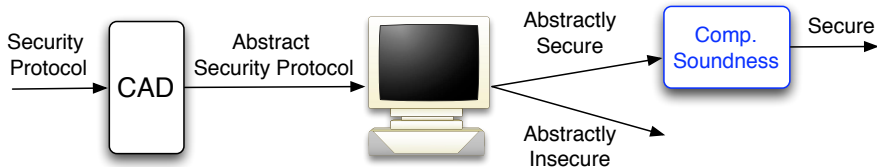
# Motivation

- Proofs are done in a "concrete" model (e.g. the turing machine model)
  - ⚡ Too complex for computers
- Simpler: Term model
  - Abstract the cryptographic primitives as terms
  - Restrict the adversary to "useful" actions
- Aim: Formulate and verify protocols in the abstract model



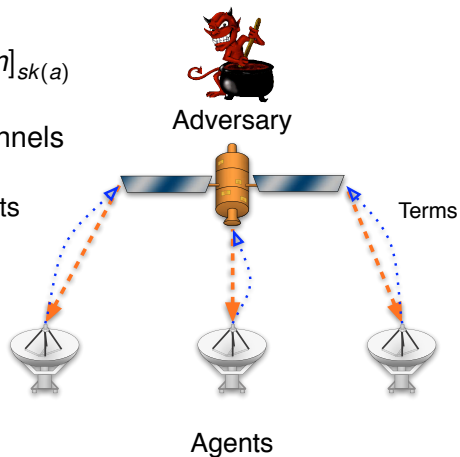
# Motivation

- But, “abstract security”  $\approx$  security in the concrete model?
  - “Computational Soundness”



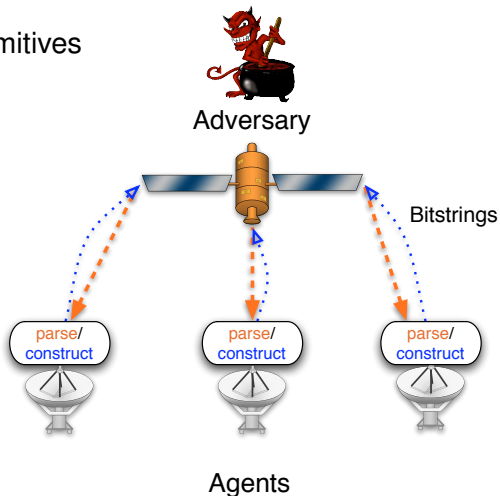
# Abstract Model

- terms e.g.:  $\{m\}_{ek(a)}$ ,  $[m]_{sk(a)}$
- active attacker:  
adversary controls channels
  - adversary sends messages and agents respond
- protocol: set of question-response patterns  $(q, r)$



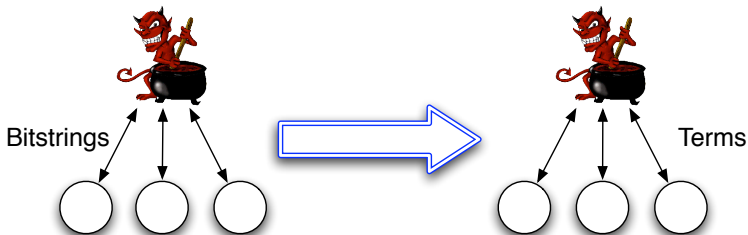
# Concrete Model

- real cryptographic primitives
- again, protocol: set of question-response patterns  $(q, r)$ 
  - ⇒ messages **parsed** to abstract terms
  - ⇒ answer of the agent **constructed** from abstract term



# Computational Soundness

- Secure in abstract model  $\Rightarrow$  Secure in concrete Model
- In contraposition: For every concrete adversary there is a corresponding abstract adversary
- Translate the communication from the concrete model to the abstract model

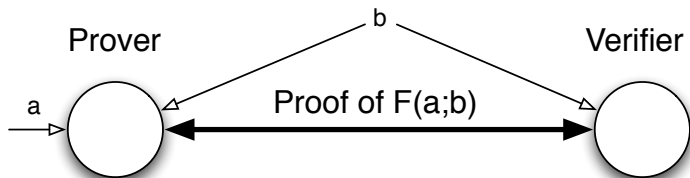


# How Much Can the Adversary Learn?

- Public knowledge
  - Knowledge from the corrupted agents
  - The responses of the agents and what is deducible from it
- ⇒ We need a set of deduction rules

- e.g. 
$$\frac{\vdash m}{\vdash \{m\}_{ek(b)}}$$

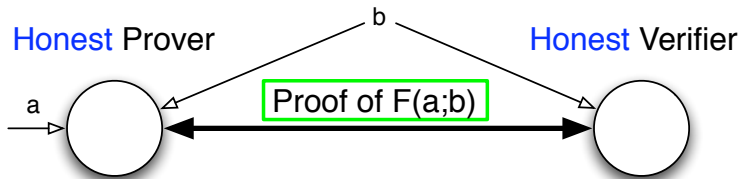
# Zero-Knowledge Proof Systems



- Public statement-formula  $F$ , private part  $a$ , public part  $b$
- Zero-Knowledge Proof System fulfills
  - **Completeness**
  - **Soundness**
  - **Zero-Knowledge**
  - Proof of Knowledge in addition **Extractability**

# Zero-Knowledge Proof Systems (cont'd)

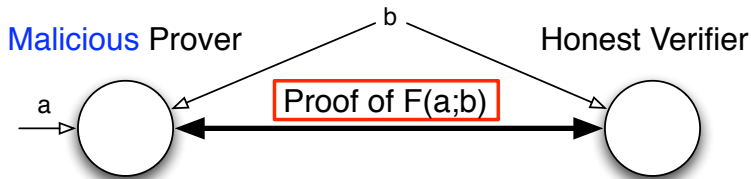
## Completeness



- **Completeness** Honest provers convince honest verifiers of true statements ( $F(a;b) = \text{true}$ )

# Zero-Knowledge Proof Systems (cont'd)

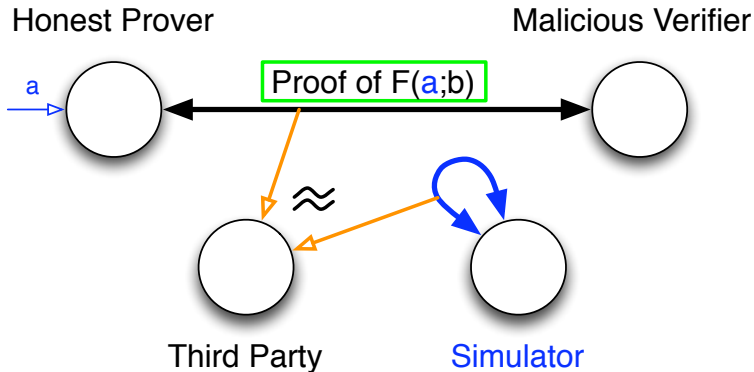
## Soundness



- **Soundness** Malicious provers can hardly convince honest verifiers of false statements ( $F(a;b) = \text{false}$ )

# Zero-Knowledge Proof Systems (cont'd)

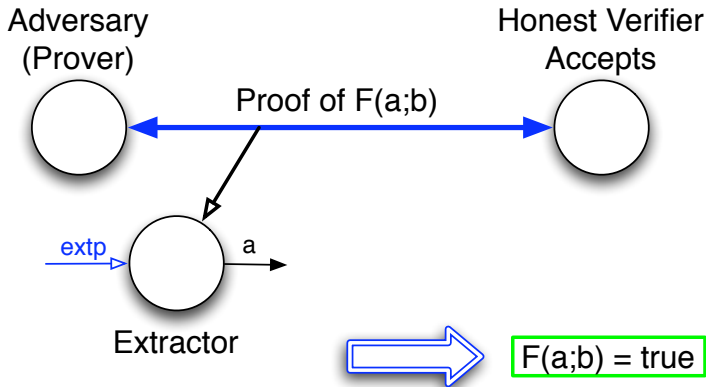
Zero-Knowledge



- **Zero-Knowledge** If  $F(a;b)$  is true, a third party cannot tell if the **proof** is real or **simulated** (without knowing  $a$ ).

# Zero-Knowledge Proof Systems (cont'd)

## Extractability - Proof of Knowledge



- **Extractability** If **Proof** of  $F(a;b)$  convinces the verifier and if **a** is extractible given **extp**, then  $F(a;b)$  is valid.

# ZK-Proofs in the Abstract Model

- ZK-proof as term:  $ZK_F(\underline{a}; \underline{b})$ 
  - Public statement-formula  $F$ , private part  $\underline{a}$ , public part  $\underline{b}$
- Statement-formulas  $F$ : boolean expressions over equations between terms
  - e.g.  $(\{a\}_{ek(b_1)} = b_2)$
- additional deduction rules:
  - e.g. 
$$\frac{\vdash \underline{a} \quad \vdash \underline{b} \quad F(\underline{a}; \underline{b}) \text{ valid}}{\vdash ZK_F(\underline{a}; \underline{b})}$$

# Computational Soundness of ZK

- Backes & Unruh: Computational Soundness of Symbolic Zero-Knowledge Proofs against active Attackers
- Strong requirements to the Zero-Knowledge Proof System
  - I. Non-malleability
  - II. Extractability

# Computational Soundness of ZK (cont'd)

## Problem I: Non-Malleability

- In the abstract model proofs are not malleable
    - e.g. given  $ZK_F(a; b)$  and  $ZK_{F'}(a'; b')$ 
      - ⚡  $ZK_{F \wedge F'}(a, a'; b, b')$  is not deducible without knowing  $a, a'$
- ⇒ We need the ZK Proof System to be non-malleable

# Computational Soundness of ZK (cont'd)

## Problem II: Extract the Witness

- Recall: We translate the communication in the concrete model to the abstract model
- When we parse the bitstring representing a zero-knowledge proof from the adversary:
  - ⚡ How to obtain the witness?

# Computational Soundness of ZK (cont'd)

## Problem II: Extract the Witness

- Recall: We translate the communication in the concrete model to the abstract model
- When we parse the bitstring representing a zero-knowledge proof from the adversary:
  - ⚡ How to obtain the witness?
- ZK Proof System needs to fulfill Extractability
  - Extraction algorithm is needed

# What I Am Going to Do?

## I. Including malleable proofs



- Find appropriate deduction rules to combine known ZK-proofs

## II. Without Extractability

- Already sketched: soundness proof for a simplified version with deterministic signatures instead of encryptions
- Next step: randomness, randomized encryptions

Thank you for your attention!

# References

-  [BackesUnruh08] Michael Backes, Dominique Unruh: Computational Soundness of Symbolic Zero-Knowledge Proofs Against Active Attackers. CSF 2008: 255-269
-  [CortierWarinschi] Véronique Cortier, Bogdan Warinschi: Computationally Sound, Automated Proofs for Security Protocols. ESOP 2005: 157-171