Universally Composable Symbolic Analysis

- for Two-Party Protocols based on Homomorphic Encryption

Morten Dahl and Ivan Damgård
Aarhus University

Symbolic Analysis

- Abstracts away details to facilitate analysis
  - formal proof
  - machine assistance
  - large systems / real-world applications

- Tool support
  - type systems
  - model checkers
  - theorem provers

Popular Choices

- Process algebra as basic model
  - keys -> unguessable symbols
  - encryption -> abstract term
  - polynomial time -> fixed set of attacker rules

- For instance
  - terms: enc(m, ek, r) and ekfor(dk)
  - rule: dec(enc(m, ekfor(dk), r), dk) = m

Popular Choices

- Classical primitives
  - encryption
  - signature
  - hash functions

- Security defined by Prop(p)
  - weak secrecy: “key k not deducible”
  - strong secrecy: “P(k_1) ≈ P(k_2)”

- (not least for real-world soundness)

Motivation

- Modern primitives somewhat neglected
    - homomorphic encryption

- ... yet could imagine many applications (special-purpose MPC)
    - Voting
    - Auctions
    - Secure Payments

- Goal is tool-aided method for formal analysis

This Work

- Two-party secure function evaluation protocols
  - homomorphic encryption, commitments, NIZK-PoK
  - Coin Flip, Oblivious Transfer, Triple generation

- Applied Pi-calculus for the symbolic model
  - well-known and suitable for ProVerif tool
  - show real-world soundness w.r.t. standard UC model

- So, **for the class of protocols we consider:**
  - **symbolic security implies UC security**

Contribution

- Symbolic model of homomorphic encryption
  - suitable for tool analysis

- Carry simulation/UC approach over to symbolic model
  - security properties as ideal functionalities
  - simulator extraction operations

- Real-world soundness of homomorphic encryption
  - for indistinguishability-based properties
  - no fixed security property

- Analysis of concrete OT protocol [DNO08]

Symbolic UC

- Natural to capture security for FSE by ideal functionalities
  - input from environment
  - corrupted players
  - strong secrecy: "Sender(x_0, x_1) ≈ Sender(0, x_1)" ??

- Usual benefits of UC
  - compositional / modular analysis (including single session)

- ... and little bonus "hybrid analysis"
  - hide sub-protocols using unsupported primitives

- See also: DKP09, BU13

Approach

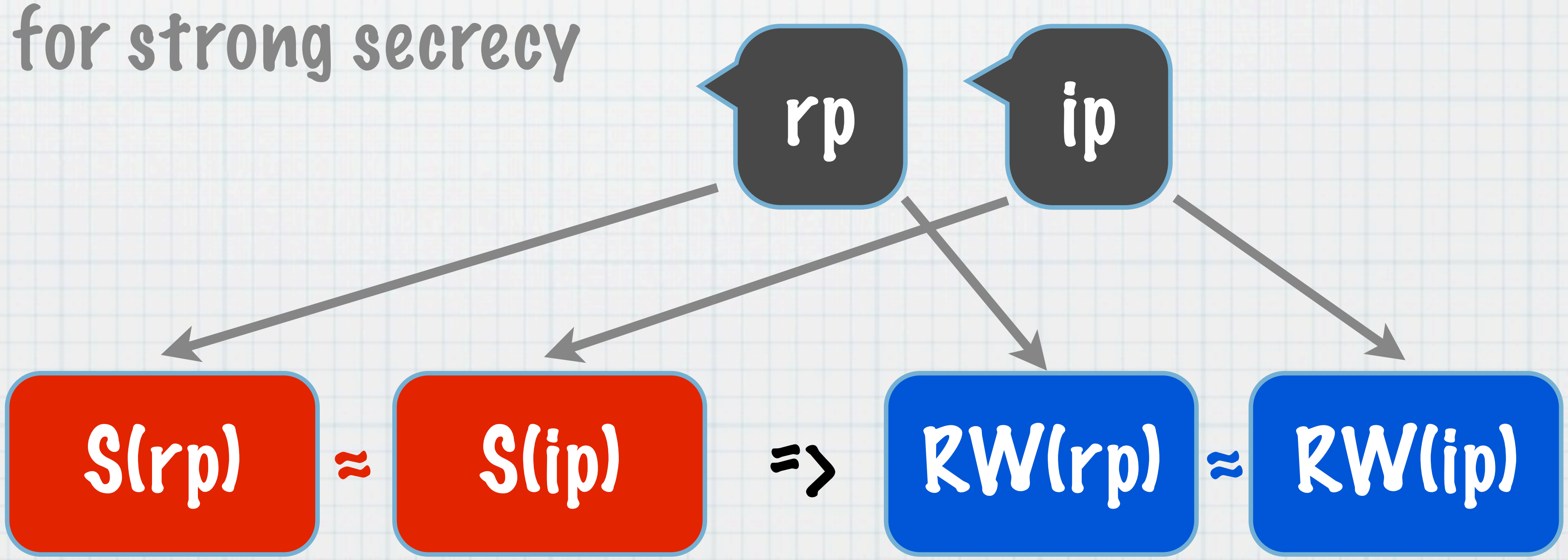
- Consider class of protocols
  - certain structure and black-box use of crypto
  - captured by high level language

- Define two interpretations of systems:
  - symbolic S(.) produces set of processes
  - computational RW(.) produces set of ITMs

- Theorem: indistinguishability carries over

How To Apply

- Methodology
  - express protocol and sub ideal functionalities
  - express target ideal functionality and simulator
  - show symbolic indistinguishability: S(rp) ≈ S(ip)
  - apply soundness theorem: RW(rp) ≈ RW(ip)
  - also works for strong secrecy



rp

ip

S(rp) ≈ S(ip) => RW(rp) ≈ RW(ip)

Protocol Language

- Used for expressing players, ideal functionalities, simulators

- Commitments
  - commit_T(..) --> [C, Proof_T]

- Homomorphic encryption
  - encrypt_T(..) --> [C, Proof_T]
  - eval_e(..) --> [C, C_1, ..., D_1, ..., Proof_e]
  - decrypt(..)

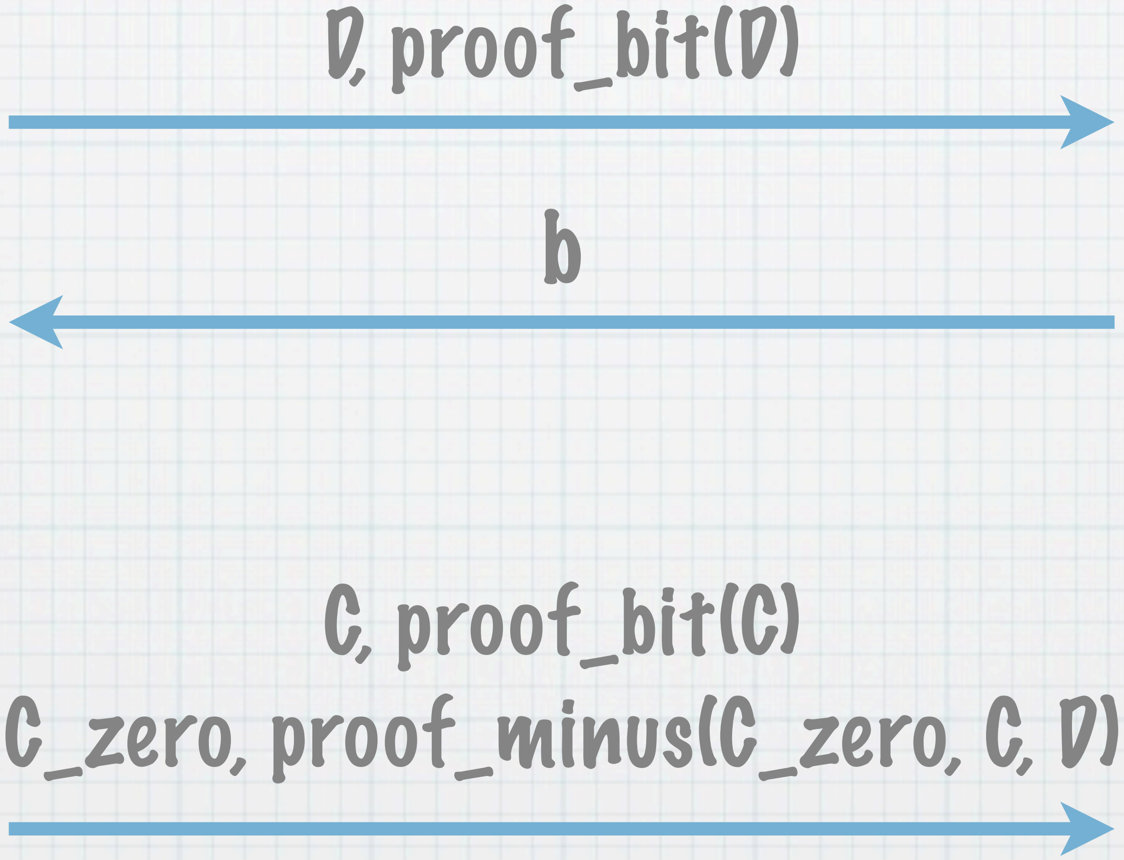
- NIZK-PoK
  - proof verification: verCommit_T(..), verEncrypt_T(..), verEval_e(..)
  - simulator witness extraction: extrCommit(..), extrEncrypt(..), ...

Coin Flip

Player A
knows crs_A, ek_B, ...
input bit a

Player B
knows ek_B, dk_B, ...
input bit b

commit_bit(a, r)



D, proof_bit(D)

b

encrypt_bit(ek_B, a)
eval_minus(C, a, r)

C, proof_bit(C)
C_zero, proof_minus(C_zero, C, D)

check decrypt(dk_B, C_zero) = 0

Soundness

- Third "intermediate" interpretation: I(p)
  - F_aux ideal crypto module
  - uniformly random handles instead of ciphertexts etc.
  - global memory with restricted access
  - fixed set of adversarial methods

- I(p_1) ≈ I(p_2) => RW(p_1) ≈ RW(p_2)
  - approximately that F_aux is realised in RW(.)

- S(p_1) ≈ S(p_2) => I(p_1) ≈ I(p_2)
  - already quite similar

I => RW

- Construct translator T
  - T[I(p)] ≈ RW(p)
  - use only adversarial methods

- hence I(p_1) ≈ I(p_2) => T[I(p_1)] ≈ T[I(p_2)]

Primitives

- Commitment scheme
  - well-spread, comp. binding, and comp. hiding

- Encryption scheme
  - homomorphic for set of expressions
  - well-spread, correct, history hiding, IND-CPA

- NIZK-PoK scheme
  - complete, comp. ZK, extractable

Translator T

- Network messages to adversary
  - honest: use dummy values
  - corrupt: obtain correct values through F_aux

- Network messages from adversary
  - easy when both honest
  - can extract most from proofs for a corrupt player
  - reject certain untranslatable messages

S => I

- Already close to each other
- Intermediate attacker forced to use F_aux (for encrypting etc.)
  - matchable by symbolic attacker with overwhelming prob.
  - fails only if he guesses a random handle
- By symbolic indistinguishability he sees the same in every activation in both cases
  - symbolic indistinguishability has weaker scheduling guarantees
  - ... small condition on protocols

Thank You !

- Two-party secure function evaluation protocols
    - homomorphic encryption, commitments, NIZK-PoK
    - Coin Flip, Oblivious Transfer, Triple generation

- Applied Pi-calculus for the symbolic model
    - well-known and suitable for ProVerif tool
    - show real-world soundness w.r.t. standard UC model

- So, **for the class of protocols we consider:**
    - **symbolic security implies UC security**