Universal composability for designing and analyzing cryptoprotocols

István Vajda
vajda@hit.bme.hu

Agenda

- Brief overview of universal composability
- Example analysis of a secure routing protocol
- Example for modular design
- Example for anonymous communication
- Modelling hash functions in UC-framework

Detailed research reports

I.Vajda. Cryptographically Sound Security Proof for On-Demand Source Routing Protocol EndairA.
Cryptology ePrint Archive Report 2011/103. http://eprint.iacr.org/2011/103.pdf

I.Vajda. Framework for Security Proofs for Reactive Routing Protocols in Multi-Hop Wireless Networks.
Cryptology ePrint Archive Report 2011/237. http://eprint.iacr.org/2011/220.pdf

I.Vajda. New look at impossibility result on Dolev-Yao models with hashes.
Cryptology ePrint Archive Report 2011/335. http://eprint.iacr.org/2011/335.pdf

I.Vajda. UC framework for anonymous communication
Cryptology ePrint Archive Report 2011/682. http://eprint.iacr.org/2011/682.pdf

Brief overview of universal composability

Traditional approach: security assessment of stand alone protocol problems

- Insufficient in general protocol environments:
execution in complex protocol environment (multi-instance, multi-execution), like the Internet
- Security guaranties when the protocol is used as a component of a larger system

Example: Consider the following key exchange protocol:

1.A → B : $\{K\}K_b$
2.B → A : $\{N\}K$
3.A → B : $\{sig_A(N)\}K$

K_b : public encryption key of party B
K : fresh session key generated by party A
N : nonce generated by B
$sig_A(N)$: party A signs N

Party B cannot be sure that the fresh session key is known only by party A and B.

Brief overview of universal composability:
example interleaving attack

Attack: adversary X is able to impersonate party A to party B:

A → X : $\{K\}K_x$

X → B : $\{K\}K_b$

B → X : $\{N\}K$

X → A : $\{N\}K$

A → X : $\{sig_A(N)\}K$

X → B : $\{sig_A(N)\}K$

A security patch:

1.A → B : $\{K\}K_b$

2.B → A : $\{N\}K$

3.A → B : $\{sig_A(B, K, N)\}K$

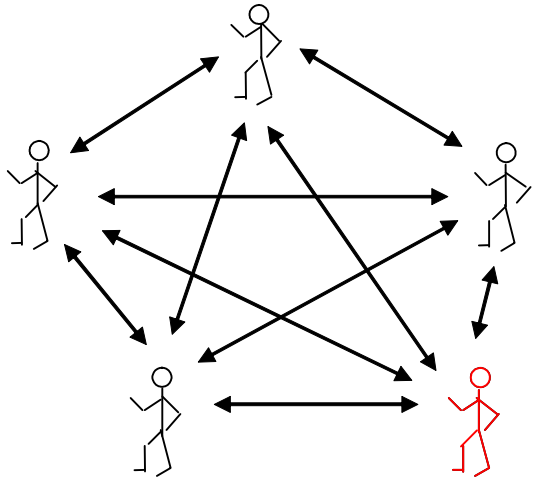
Brief overview of universal composability:
ideal protocol and secure realization

Ideal protocol carries out the cryptographic task in (ideally) secure way

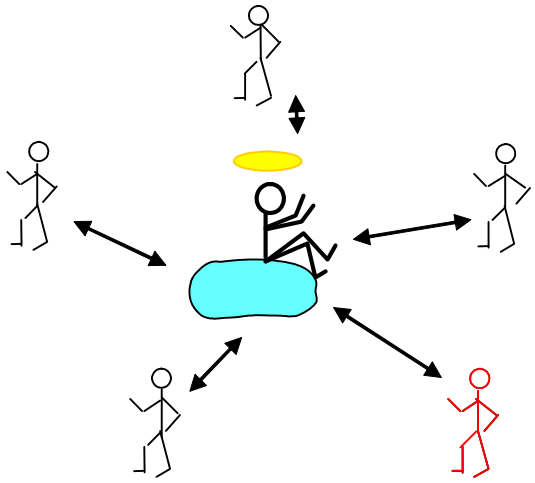
→ Our examples: anonymous communication, random hash

A (real) protocol **UC-realizes** the task**:**

Informal: any damage that can be caused by an adversary interacting with the (real) protocol can also be caused by an adversary interacting with the ideal protocol for the task (simulatability)



Real protocol



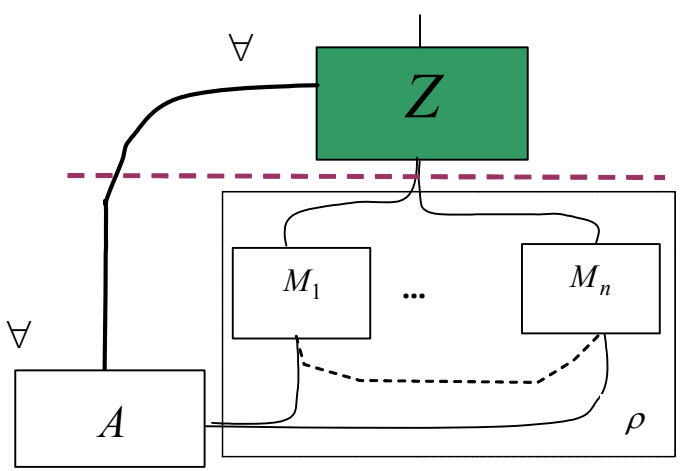
Ideal protocol

Brief overview of universal composability:
the environment

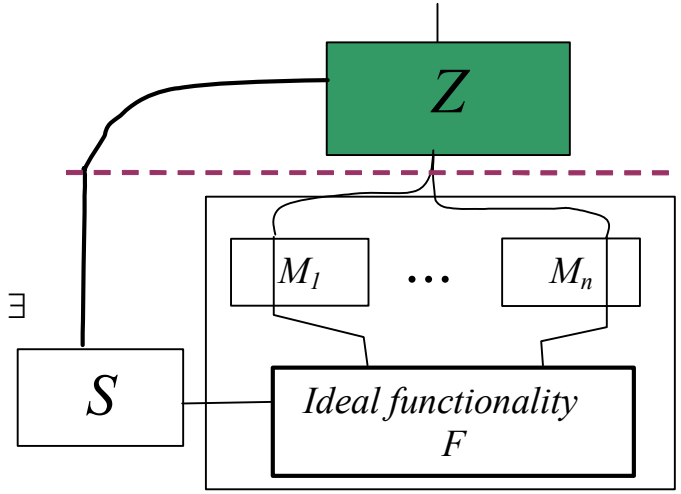
Security with respect to **interactive environment** (Z):

- *A* and *Z* are allowed to interact freely throughout the run of the protocol
  (i.e. not only at input and output events)

- the environment is an interactive distinguisher between the real and the ideal system
  (UC-realization ↔ indistinguishability of views)



Real configuration



Ideal configuration

Computational indistinguishability

Distinguishing algorithm $Z : \{0,1\}^{\infty} \to \{0,1\}$
distinguish discrete probability distributions *D* and *D'* from one sample and computational resource limit t.

The output of *Z* is 1, if it decides on *D* , otherwise it is 0.

Definition: Distributions *D* and *D'* are **(t,ε)-indistinguishable**, if

$$d_t(D, D') \leq \varepsilon$$

where

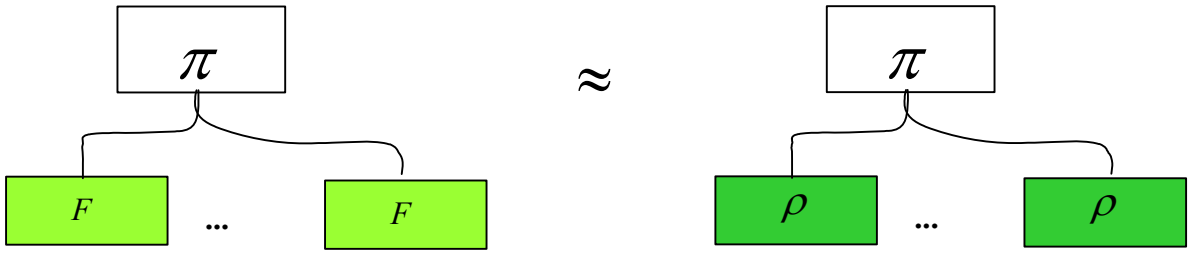
$$d_t(D, D') = \max_Z \left| \underset{x \leftarrow D}{P}(Z(x) = 1) - \underset{x' \leftarrow D'}{P}(Z(x') = 1) \right|$$

Brief overview of universal composability: universal composition

F – hybrid protocol: protocol π is an F – hybrid protocol, if the parties also make calls to ideal protocol/functionality F

Composed protocol π^{ρ} : replacing each call to a new instance of F with a new instance of protocol ρ

Th: *If protocol ρ UC-realizes ideal functionality F, then for any adversary A interacting with protocol π^{ρ} there exists an adversary A_F , interacting with protocol π, such that, no enviroment can tell which of this two systems it interacts with.*



F-hybrid protocol Composed protocol

Brief overview of universal composability: interpretations of UC

Interpretations/usage:

1. UC secure protocols maintain their security within any potocol environment (guarantee aspect)

2. modular design and analyis of complex protocols
   Our example: ad-hoc routing protocols

3. ease of analysis: the task is defined and analyzed for a stand alone protocol problem

   → the security in multi-party, multi-instance setting is guaranteed via the universal composition theorem

Brief overview of universal composability: automated analysis

Approaches:

1. Standard approaches for formal analysis of distributed systems and protocols

Disadvantages:

they cannot model

- computational bounds on processes and adversaries
- randomized protocols

(Security of real cryptographic primitives is guaranteed only in a computational and probabilistic sense!)

2. Dolev-Yao type symbolic analysis: crypto-primitives substituted by symbolic operations

Disadvantage: the analysis does not provide cryptographic soundness

2*. Pfitzmann-Waidner's approach: composable Dolev-Yao symbolic operations

→ Symbolic analysis can be done with cryptographic soundness guarantee!

Our examples

Brief overview of universal composability: (un)realizability

The realizability issue:

Which cryptographic tasks (ideal functionalities) are UC-realizable?
(under which set-up and computational assumptions)

Standard examples:

coin tossing, commitment, zero knowledge, oblivious transfer

Unrealizable: if only authenticated communication channels (F_{AUTH})
Realizable: if also trust set-up models (e.g. common random string functionality, F_{CRS})

Our example: hash function

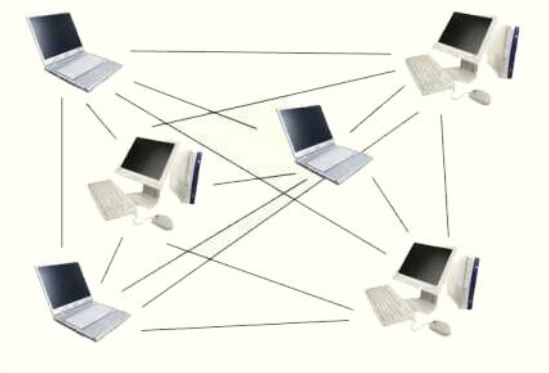
Agenda

- Brief overview of universal composability
- **Example analysis of a secure routing protocol**
- Example for modular design
- Example for anonymous communication
- Modelling hash functions in UC-framework

Secure routing protocol

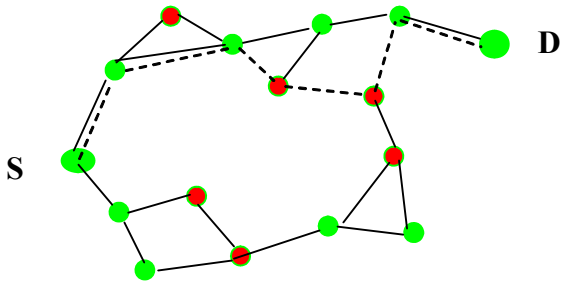
Ad-hoc (wireless) networks:

- a collection of autonomous (mobile) nodes that communicate with each other over wireless links without any central administration
- each host has to act as a router for hosts within the limited range of wireless transmission



Importance of secure routing ↔ informal analysis

Task of secure route discovery

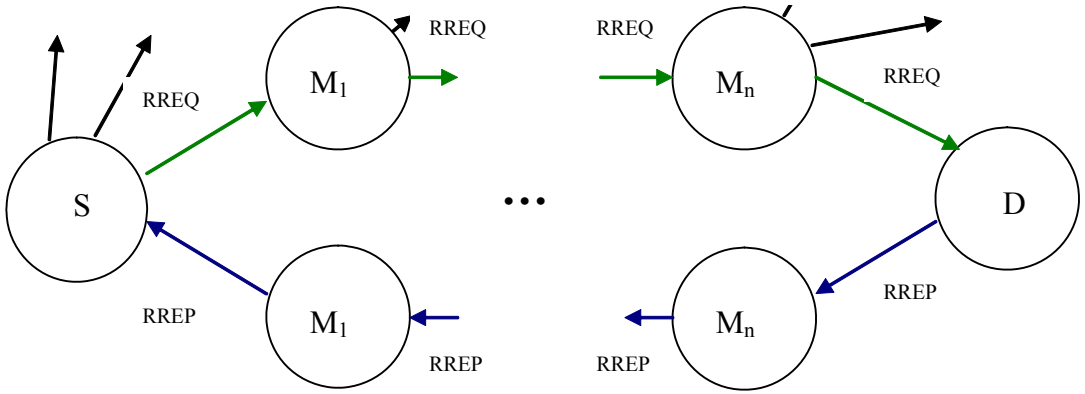


Protocol EndairA:
route acquisition

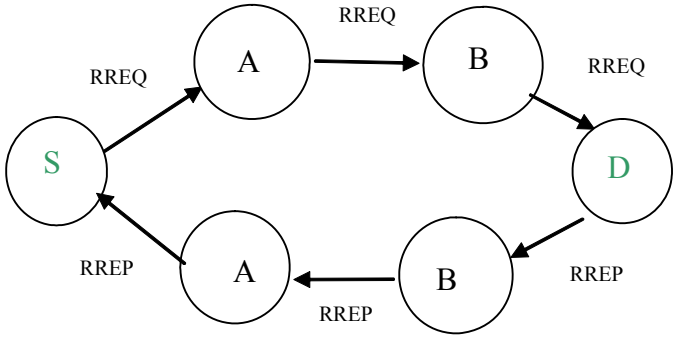
On-demand route acquisition protocol :

- Route Request: a source node initiates a route discovery towards a destination node by creating and broadcasting a Route Request (RREQ) message

- Route Reply: when the destination receives a RREQ, it initiates a Route Reply (RREP) phase and sends a corresponding message towards the source on the route received in the RREQ message



Protocol EndairA:
the protocol



$S \rightarrow * :$ {rreq, S, D, id, ()}
initiator generates a route request (IDs: initiator, target, request identifier)

$A \rightarrow * :$ {rreq, S, D, id, (A)}
intermediate node appends its identifier and re-broadcasts the request

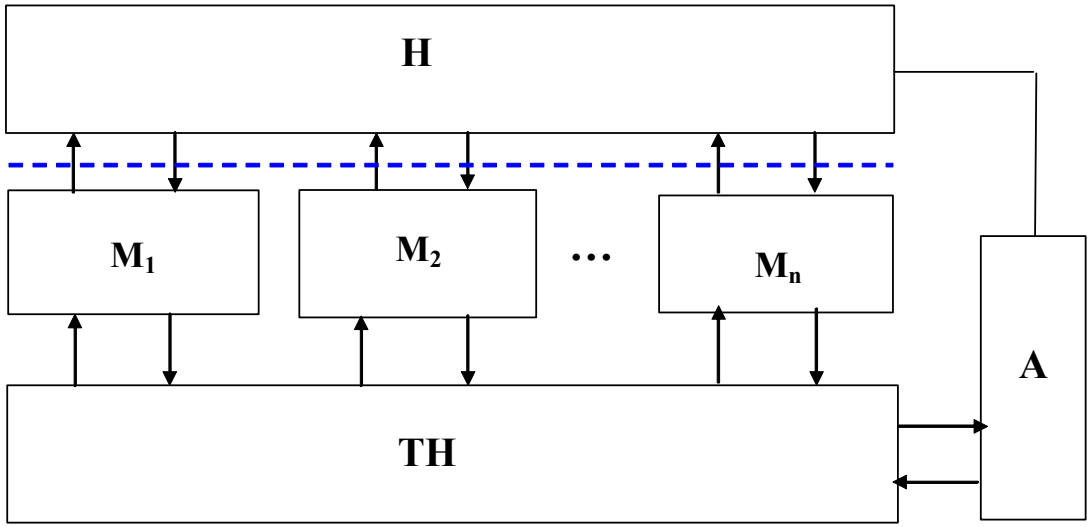
$B \rightarrow * :$ {rreq, S, D, id, (A,B)}

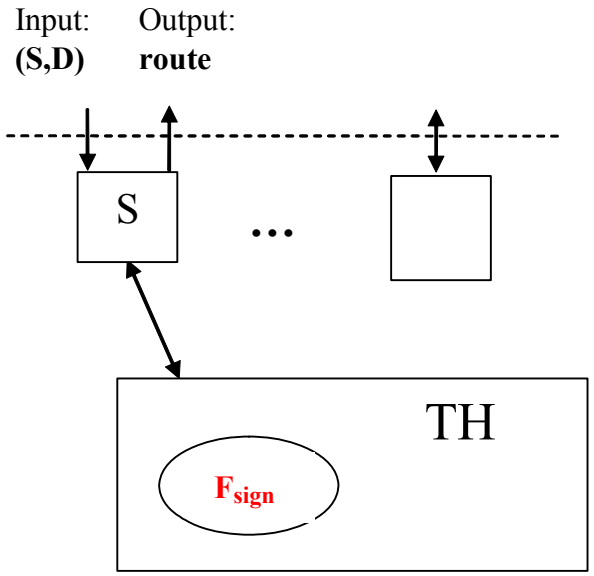
$D \rightarrow B :$ {(rrep, S, D, (A,B), sig_D)}
target generates a route reply with digital signature on the received route
reply is sent back to the initiator on the reverse of the route found in the request

$B \rightarrow A :$ {((rrep, S, D, (A,B), sig_D) sig_B)}
intermediate node verifies: IDs and digital signatures of neighbours

$A \rightarrow S :$ {(((rrep, S, D, (A,B), sig_D) sig_B) sig_A)}
initiator verifies: if the first identifier belongs to a neighbor; all the signatures in the reply

Protocol EndairA:
ideal system





Corollary of the Composition Theorem:
If an F-hybrid protocol π UC-realizes an ideal functionality G, then so does composed protocol $\pi^{\rho/F}$.

Protocol EndairA:
security requirement

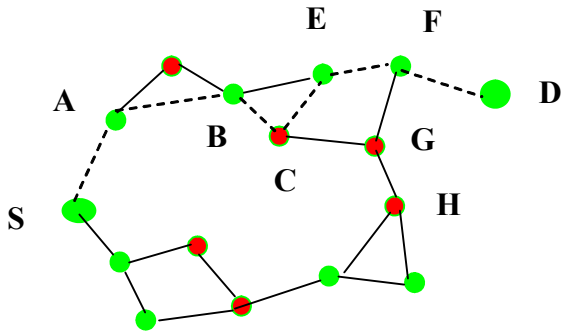
***Discovered Route Requirement*:**

If $C_1, C_2, \ldots C_m$ are the honest nodes on the discovered route output by the protocol, then in fact,
these nodes are all the honest nodes, in the given order, on an existing route from the initializing node *S* ($=C_1$) to the destination node D ($=C_m$).

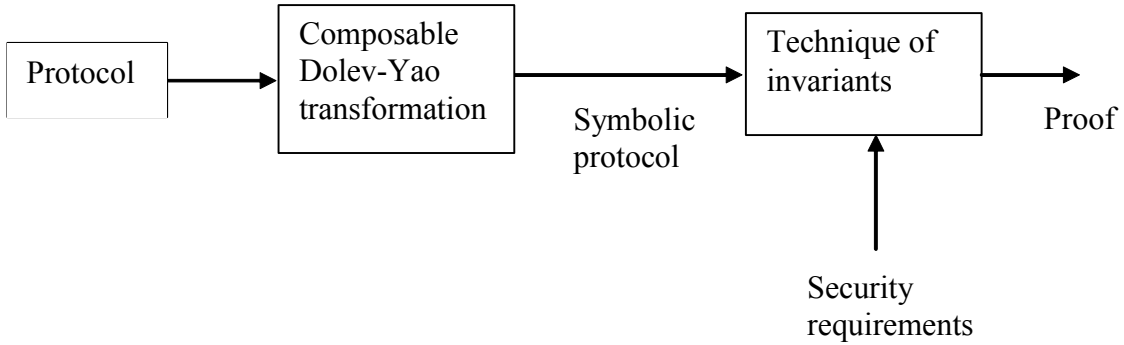
Route 1: S-ABCEF-D

Route 2: S-ABGCHEF-D

Route 3: ~~*S-AHCEF-D*~~



Protocol EndairA:
theorem/proof technique



Theorem: Protocol EndairA "UC-realizes Discovered Route Requirement".

Proof:

Invariants:

Inv. 1 *(Correct time order of signatures)* The order of signatures in the chain uniquely determines the order of time when they were generated.

Inv. 2 *(Correct time order of having control over TH)* The order of signatures in the chain uniquely determines the order of time the control held by a node having a handle to the corresponding secret key.

Inv. 3 *(Existing route)* At the time of route acquisition there existed a communication route between any two nodes which had handle to secret signing keys corresponding to two signatures in the chain.

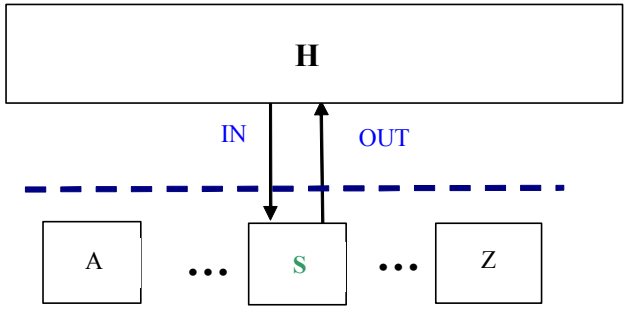
Agenda

- Brief overview of universal composability
- Example analysis of secure routing protocol
- **Example for modular design**
- Example for anonymous communication
- Modelling hash functions in UC-framework

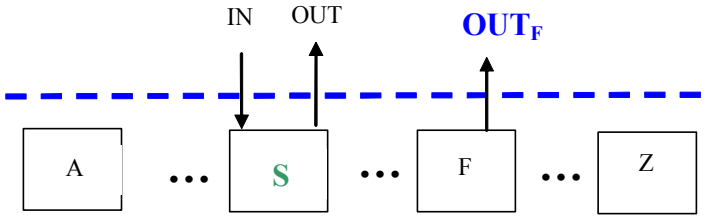
Modular design:
the idea

Decomposition of the security requirement:

breaking down the (global) security requirement against the system into (local) security requirements against the (honest) protocol machines of the system



User machine initializes session and receives result via protocol machine S



All participating nodes report their contribution to the route discovery process directly to H via a virtual interface and not via the mediation of a chain of nodes

Advantages: easier (UC-) design, analysis, fault detection

Modular design:
Applications: On-demand source routing

Global Req:
Discovered Route Requirement

Local Req:
1. Identifier of the protocol machine
2. Session identification information (minimally S, D, session id)
3. Identifier of those protocol machines in time order, which have processed the protocol message before it arrived to machine

where an identifier corresponds to the true identity of the machine if it is an honest machine, otherwise, it is one from the set of the identifiers of adversarial machines.

Theorem: (Decomposition of global security requirement)
Global security requirement is fulfilled if and only if honest nodes comply with local security requirement.

Example (fault detection): ***Ariadne***
During the RREQ phase an intermediate node *C* appends its public identifier, id_c to the received message *m* and signs the result:
(m, id_c) $sign_c$

It is an insecure implementation of the ideal_channel. E.g. adversary *A* is able to remove the signature (at the end of the message arriving to it), and substitute it to get:
(m, id_A) $sign_A$ → *non-existing route*

Modular design:
Applications: On-demand Vector Distance Protocols

Global Req:

Routing entries of honest nodes must be correct, where a routing entry (*C*,*E*,*F*,*c*) is correct, if there exists a route starting at node *C* and ending at node *F* via next hop *E* such that on this route:

1.1.) each honest node (*U*) has a routing entry with ending node *F* such that the next hop points to

1.1.1) an honest node (*V*), if *U* and *V* are (direct) neighbors,

1.2.2) an honest node (*V*), if *U* and *V* are pseudo neighbors,

1.1.3) any of the adversarial nodes, otherwise.

1.2.) the sum of costs over honest nodes on the route is less than or equal *c*

Local Req:

1.) The ideal honest machine sets the corresponding routing entry:
ending node id is set to the true identity of the node launching the session phase (RREQ phase, RREP phase),

2.) next hop id is set to sender id from whom the input has been received, where the sender id is the true identity of the sender if it is an honest machine, otherwise it is one from the set of the identifiers of the adversarial machines.

3.) cost of usage of a honest machine in a session cannot be influenced by the adversary

Theorem: Protocol ARAN is UC-secure

Conclusions/routing protocols

- first UC-secure routing protocols

  The flaws in routing protocols can be very subtle, therefore it is very difficult to discover them by informal reasoning!

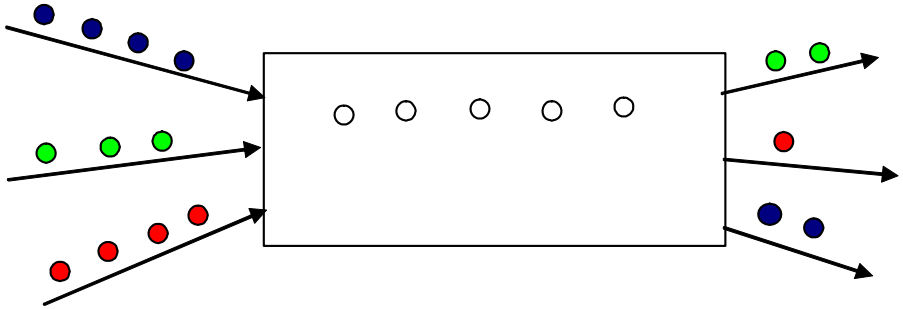
- modular design for practical protocols

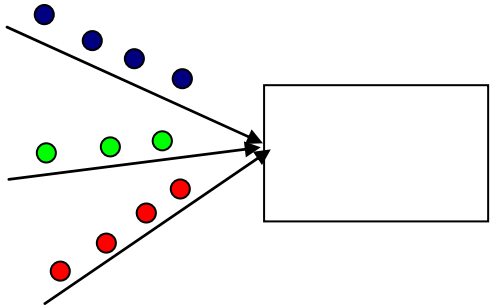
  Easier design/analysis/fault detection

Agenda

- Brief overview of universal composability
- Example analysis of secure routing protocol
- Example for modular design
- **Example for anonymous communication**
- Modelling hash functions in UC-framework

Anonymous communication:
anonymity notions



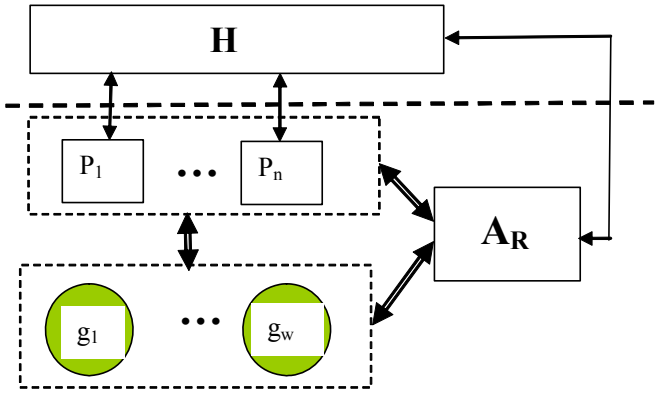


- ***Unlinkability***
- ***Sender/receiver anonymity***
- ***Relationship anonymity***

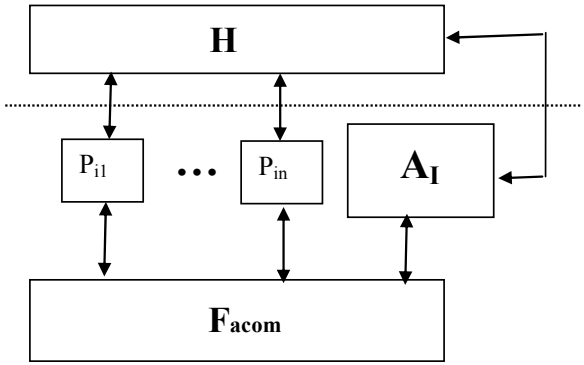
Sender/receiver anonymity → Relationship anonymity

TolI: Tolerable imperfection

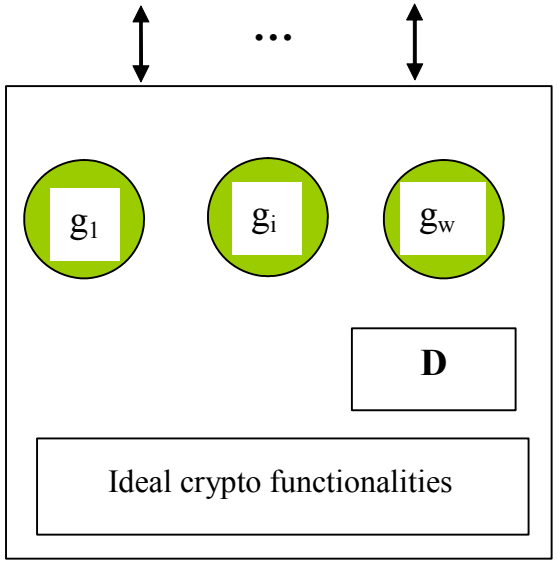
Anonymous communication
real/ideal models



real model



ideal model



ideal functionality F_{acom}

Anonymous communication
ideal model

1. Suppose H initializes a run: $H \rightarrow P_s$: (P_s, P_r, m)

   $P_s \rightarrow F_{anom}$: (P_s, P_r, m)

   $D :\Leftarrow (ind := size++, type := data, \arg := (P_S, P_r, m), hnd_{P_S})$

   F_{anon} "informs" the adversary (ToI)

2. Suppose the adversary decides to attack: compromise, initiate message sending, send guesses to H

3. Suppose F_{anon} makes a step of anonymization:

   $D :\Leftarrow (ind := size++, type := annon-data, \arg := (g_i, l_i, c_i), hnd_A)$

   F_{anon} "informs" the adversary (honest g_i): g_i: access by ToI
   c_i: access
   l_i: no access

4. Suppose F_{anon} decides to output: all output messages are sent to all receivers

5. Suppose F_{anon} , according to the level of compromization, gives control to the adversary and no longer guarantees anything (in a steps 3-4)

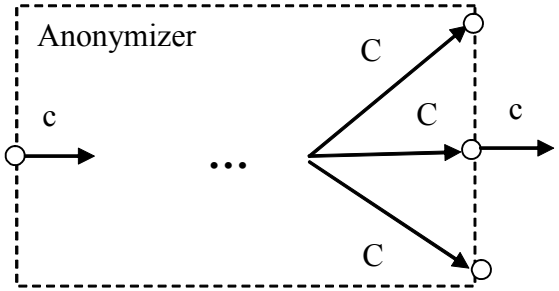
Anonymous communication:
protection of anonymity/ideal model

Principle: The adversary should not gain any anonymity-related information ***in excess to a priori (Tol)*** *from observing the run of the anonymizer network.*

1.) when inputs are sent to the anonymizer

2.) when the anonymizer “calculates”

3.) when the anonymizer sends outputs



The anonymizer sends all output packets onto all output channels

→ Receivers must be able to identify and select packets intended to them (coding).

π_Q

Anonymous communication:
Indistinguishability based anonymization vs. simulatability

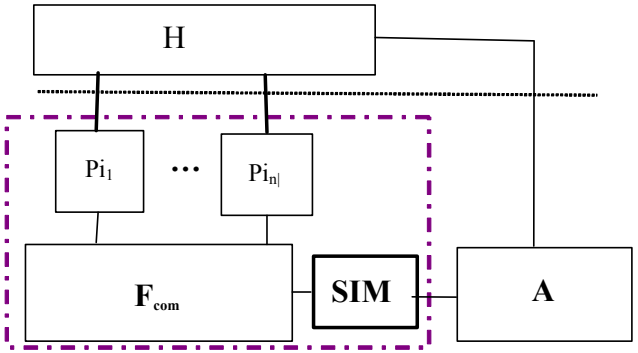
Indistinguishability "game": IND anonymizer

Hevia, D. Micciancio. An indistinguishability-based characterization of anonymous channels.
InNikita Borisov and Ian Goldberg, editors, *Privacy Enhancing Technologies: Eighth International Symposium, PETS 2008*, pages 24-43. Springer-Verlag, LNCS 5134, July 2008.

Theorem: Under *global passive adversary* an anonymous communication scheme *Q* is an IND_anonymizer if and only if π_Q it UC-realizes ideal functionality F_{anon}.

Corollary (extension to *adaptive adversary*):
Assume an adaptive adversary, the corruption operation of which can be simulated by overhearing the communication channels used for controlling the corruption attack.
An anonymous communication scheme *Q* is an IND_anonymizer with such an adaptive adversary if and only if π_Q UC-realizes ideal functionality F_{anon} with corresponding adaptive adversary.

Anonymous communication:
Proof details



UC → IND

If *Q* is not an IND_anonymizer then π_Q fails to securely realize ideal functionality F_{anon}.

A_Q : IND- adversary successfully attacks *Q*

Reduction proof:
An environment (A,H) is constructed, which can distinguish the real and the ideal* systems with the same advantage

UC-Adversary A runs adversary A_Q in a simulation of the IND_anonymizer game:
- Generates a pair of TоI-constrained test messages;
- Chooses random bit *b* and sends message to the system (real or ideal*) via *H* for anonymous transmission;
- Lets adversary A_Q see the same view of the system;
- When A_Q outputs decision *b'*, *A* outputs *b* xor *b'* via *H.*

P(A outputs 0 / real) =1/2+ε, P(A outputs 0 / ideal) =1/2

→ P(A out 0/real) - P(A out 0/ideal) = ε (non-negligible)

Anonymous communication

• a generic ideal model for anonymous communication together with a proof system within UC-framework

• this model extends earlier models

• our definition of anonymity is equivalent to the notion of computational indistinguishability (even in case of adaptive adversary)

Agenda

- Brief overview of universal composability
- Example analysis of secure routing protocol
- Example for modular design
- Example for anonymous communication
- Modelling hash functions in UC-framework

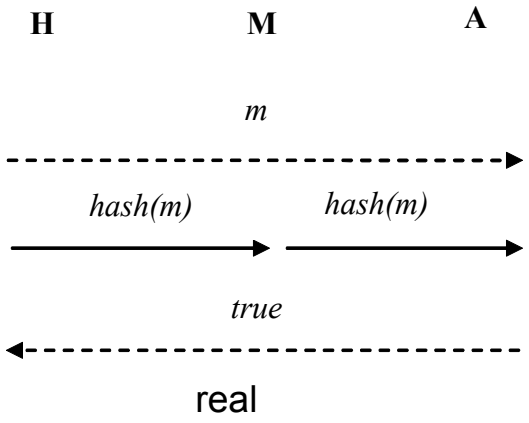
Modelling hash functions in UC-framework:
An unrealizability result

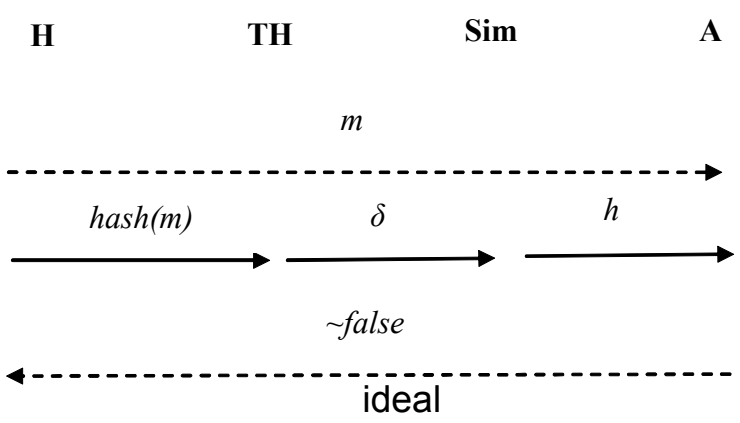
Protocols with hashes Dolev-Yao style models do not have cryptographically sound realization in the sense of BRSIM/UC in the standard model of cryptography.

M. Backes, B. Pfitzmann, and M. Waidner. Limits of the Reactive Simulatability/UC of Dolev-Yao Models for Hashes. *Cryptology ePrint Archive: Report 2006/068.* also in *Workshop on Formal and Computational Cryptography (FCC 2006) (2006)*

- commitment
- ideal hash function

$\longrightarrow$ simulation-failure





Modelling hash functions in UC-framework: Solutions

1. Random oracle model (F_{RO}-hybrid model), TTP

2. Random hash primitive in the standard model of cryptography (r_hash):

Properties of the ideal primitive r_hash*:*

i.) ***Ideal collision freeness***

ii.) ***Ideal secrecy***

iii.) ***Sender-controlled access to the verification algorithm***

Modelling hash functions in UC-framework:
Real model of r_hash

r_hash: *{0,1}* x V* → random variable over $\{0,1\}^{r\_hash\_len(k)}$

is a random mapping over the message space, where *V* is the power set of *U*.
It can be evaluated efficiently and has the following properties:

i.) collision free

ii.) random variables ***r_hash(m₁,Vset)*** and ***r_hash(m₂,Vset)*** are **indistinguishable** for any $m_1 \neq m_2$ and any Vset *in U.*
We specialize this hash primitive by introducing auxiliary function

$Rhash(r,m,Vset): \{0,1\}^{rand\_len(k)} \times \{0,1\}^* \times V \rightarrow \{0,1\}^{r\_has\_len(k)}$

such that if we substitute random value into parameter *r* we get *r_hash(m,Vset).*

iii.) the verification algorithm *Ver(m,h,id)* is defined as follows:

$Ver: \{0,1\}^* \times \{0,1\}^{r\_hash\_len(k)} \times ID \rightarrow \{0,1,\downarrow\}$

where the inputs are the following, in order: hash value (*h*), message (*m*), user identifier (*id*). The evaluation of the output is the following. First, algorithm

$r\_decrypt(h,id) \rightarrow \{r, \downarrow\}$

is called, which **outputs ↓ if, *id not in Vset*** else it outputs *r*.

Ver() outputs ↓, if r_decrypt() outputs ↓. Otherwise, the output is 1 if , *h=Rhash(r,m,Vset)*, else it is 0.

Modelling hash functions in UC-framework: Construction

Theorem: The ideal *r-hash* model is UC-securely implemented by the real *r-hash* model in the standard model of cryptography, assumed that honest users authorize only honest users to carry out verification.

Constructions / implementation of the real model:

Canetti's idea: random hashing with indistinguishability property
(R.Canetti. Towards Realizing Random Oracles: Hash Functions That Hide All partial Information. *Advance in Cryptology – CRYPTO '97*, LNCS 1294., pp.455-469, 1997.)

(i.) A collision free hash function *h(z)*.

(ii.) Random function *F(r,x)*, random parameter *r,* input *x (=h(z))* , :

- provides the indistinguishability property in input *x*
- *F(r,x) = (r, F'(r,x))* , *F'(r,x)* is one way in both inputs *r* and *x*
- random parameter *r*, is protected and is revealed only for intended users (*Vset*)

Modelling hash functions in UC-framework: Construction

- Construction for F(r,x) over group G:

$F(r,x)=(r, p^r x)$

where

$p \leftarrow_{rand} G$, known publicly

$r \leftarrow_{rand} S$, $S=\{1,2,\ldots,|G|\}$, kept secret

- The ElGamal public key encryption transformation over *G* provides IND-CPA security:

key pair: $pk=g^z$ $(=X)$, $sk=z$

encryption: $E_{pk}(m)=(g^y, X^y m)$

where

$z \leftarrow_{rand} G$, $y \leftarrow_{rand} G$, g is a generator of *G*.

- *Casting:* $p \sim X$, $r \sim y$, $x \sim m$

$p^r x \sim X^y m$

Conclusions/impossibility result on hashes

- proposal of a new type of random hash primitive (collision free, ideal secrecy, sender-controlled access to the verification algorithm).

- UC-models of the primitive (ideal, real properties of the primitive)

- the proposed ideal hash mapping has cryptographically sound realization in the standard model of cryptography

- construction

Research directions

In general

•definition of ideal functionalities in a way that relaxes requirements against realization

•capturing cryptographic tasks in UC-framework (e.g. electronic commerce applications)

•set-up assumptions (corresponding ideal functionalities) for general feasibility results

•modular design techniques (partitioning of large systems, tasks)

In particular

Invitation for research cooperation in topics of this talk