# Optimistic Non-Repudiation Protocol Analysis

Laurent Vigneron
*Joint work with Judson Santiago*

LORIA, Nancy-Université

WISTP'07, Heraklion, May 10, 2007

# Outline

# Outline

# Introduction



**Context:**

- Security of communications over an open network (wireless or not)
- Handled at software level by cryptographic protocols

**Standard properties** intensively studied:

- Secrecy
- Authentication

Efficient analysis methods and automatic tools already exist for several years

# Introduction



**Some security properties are rarely considered:**

- Non-repudiation
- Fair exchange

**What is non-repudiation?**

- Impossibility to deny participation to the communication

**What is the role of non-repudiation protocols?**

- To generates evidences of participation to the protocol
  *Easy!. . .* by digital signatures for example
- But, need of **fairness**: reciprocity and synchronization of non-repudiation
  *Much more difficult:* a trusted third party (TTP) is needed for fair exchanges

# Non-Repudiation Protocols



**What are the different kinds of non-repudiation protocols?**

- *With full involvement of a TTP:* used as delivery agent of evidences
  Problem: strong activity of the TTP; may be a bottleneck
  *Example:* Fair Zhou-Gollmann protocol
- *Optimistic protocols:* use of a TTP only if needed
  Based on the use of several protocols
  Permits each party to complete its protocol, even in case of problem
  *Example:* Cederquist-Corin-Dashti protocol

# Outline

# Example of Optimistic Non-Repudiation Protocol



**The Cederquist-Corin-Dashti protocol:** [2005]
Given two participants $A$ and $B$,
and a *trusted third party* ($TTP$),

Fair exchange of a message $M$ from $A$ to $B$,
with evidences of origin and of receipt,
using, if necessary, the $TTP$

Decomposition into three sub-protocols:

- a main sub-protocol beetween $A$ and $B$
- a protocol with the TTP for aborting the exchange
- a protocol with the TTP for resolving the exchange

# The Cederquist-Corin-Dashti Protocol



**Main sub-protocol:**
Normal run of the CCD protocol: exchange of evidences between $A$ and $B$

1. $A \rightarrow B : \{M\}_K.EOO_M$
   where $EOO_M = \{B.TTP.H(\{M\}_K).\{K.A\}_{Kttp}\}_{inv(Ka)}$
2. $B \rightarrow A : EOR_M$
   where $EOR_M = \{EOO_M\}_{inv(Kb)}$
3. $A \rightarrow B : K$
4. $B \rightarrow A : EOR_K$
   where $EOR_K = \{A.H(\{M\}_K).K\}_{inv(Kb)}$

**At the end:** $A$ and $B$ know $M$, and can prove the participation of each other to the communication

# The Cederquist-Corin-Dashti Protocol



**Abort sub-protocol:**

$A$ can decide to abort the protocol:

1. $A \rightarrow TTP : \{\texttt{abort}.H(\{M\}_K).B.\{K.A\}_{Kttp}\}_{inv(Ka)}$
2. $TTP \rightarrow A :$

   $E_{TTP}$ if protocol resolved

   where $E_{TTP} = \{A.B.K.H(\{M\}_K)\}_{inv(Kttp)}$

   $AB_{TTP}$ otherwise

   where $AB_{TTP} = \{A.B.H(\{M\}_K).\{K.A\}_{Kttp}\}_{inv(Kttp)}$

The $TTP$ helps $A$ to definitively abort the protocol, or to resolve it (if too late for abort)

# The Cederquist-Corin-Dashti Protocol



**Resolve sub-protocol**

$A$ and $B$ can ask the $TTP$ to resolve the protocol:

1. $A/B \rightarrow TTP : EOR_M$
2. $TTP \rightarrow A/B :$

| | |
|---|---|
| $AB_{TTP}$ | if protocol aborted |
| $E_{TTP}$ | otherwise |

The $TTP$ helps $A$ or $B$ to get its evidences (if not aborted)

**With those three sub-protocols,** each participant will always terminate its protocol run

# Outline

# Properties of Non-Repudiation Protocols



Many properties exist, but three main ones.

**Evidences of receipt for $A$:**

- Proof that the message $M$ has been received by $B$ and can be decrypted

**Evidences of origin for $B$:**

- Proof that the message $M$ has been sent by $A$

**Fairness:**

- Evidences of origin and of receipt, or none of them

# Properties of Non-Repudiation Protocols



**Evidences for the CCD Protocol:**

- Evidences of receipt for $A$:
  - $\{M\}_K$, $EOR_M$, and either $EOR_K$ or $E_{TTP}$

  ($E_{TTP}$ given by the $TTP$)
- Evidences of origin for $B$:
  - $\{M\}_K$, $EOO_M$, and $K$

  ($K$ may be given by the $TTP$, via $E_{TTP}$)
- Main property: fairness of the exchange of $M$

# Outline

# Analysis of Non-Repudiation Protocols



- No existing tool dedicated to such analysis
- Several tools for the analysis of cryptographic protocols
- Choice of AVISPA: `www.avispa-project.org`
- Built-in properties: secrecy and authentication
- Problem: representation of non-repudiation properties?
- Solution: use of LTL (linear time logic) formulas and agent knowledge
- Consequences:
  - Extension of the AVISPA specification language, for agent knowledge
  - Extension of AtSe (back-end of AVISPA), for considering simple LTL formulas

# Analysis of Non-Repudiation Protocols



**Representation of properties as LTL formulas:**

- If, at the end of the protocol run, $A$ has its evidences or $A$ is dishonest and the intruder has evidences of receipt, then $B$ should have the evidences of origin
- If, at the end of the protocol run, $B$ has its evidences or $B$ is dishonest and the intruder has evidences of origin, then $A$ should have the evidences of receipt
- Either both above properties are satisfied, or none of them (*fairness*)

**Automatic analysis of CCD with AVISPA/AtSe:**
two attacks found.

# First attack on the CCD Protocol



Given two honest agents $A$ and $B$,
and an intruder $i$ controling the network:

```
1. A -> i(B)    : {M}_K.EOOM
*** timeout for A ***
2. A -> i(TTP)  : ABORT
3. i(A) -> B    : {M}_K.EOOM
4. B -> i(A)    : EORM
5. i(A) -> TTP  : RESOLVE   (=EORM)
6. TTP -> i(A)  : ETTP
*** timeout for B ***
7. B -> i(TTP)  : RESOLVE
8. i(TTP) -> A  : ETTP
9. i(TTP) -> B  : ETTP
```

Use of timeout for aborting/resolving the protocol: $EOR_M$ unknown by $A$.

# First attack on the CCD Protocol



**Originality of the attack:**

- $A$ cannot prove that $B$ knows $M$, but it can guess it, as the protocol has been resolved
- $B$ has used the $TTP$ for decrpyting $M$;
  it has evidences that $A$ has sent $M$;
  it does not know that $A$ does not have its evidences
- The $TTP$ thinks both $A$ and $B$ have asked for resolution, and that both have their evidences

**Remark:** in this scenario, only one protocol run between honest agents!

# Second attack on the CCD Protocol



A variant of the first attack, where $B$ is dishonest ($i$):

```
1. A -> i   : {M}_K.EOOM
2. i -> TTP : RESOLVE
3. TTP -> i : ETTP
*** timeout for A ***
4. A -> TTP : ABORT
5. TTP -> A : ETTP
```

Again, at the end, $A$ does not know $EOR_M$

# Correction of the Protocol



**Origin of the flaw:**
Lack of information sent by the $TTP$ to $A$ when abort required, but protocol already resolved

**Solution:** in that case, the $TTP$ sends $E_{TTP}.EOR_M$

**Test of this solution:**

- One-session scenarios: with and without dishonest agents
- Two-sessions scenarios: with and without dishonest agents

$\longrightarrow$ No attack found by AVISPA/AtSe!
...but this is not a proof of correctness...

# Outline

# Conclusion



- Protocols are simple to design, but still difficult to verify
- Efficient search for attacks with AVISPA/AtSe:
  - A powerful specification language, for messages exchanges and properties description
  - Automatic attack search engine
  - Analysis of standard scenarios, covering various cases
  - But no formal proof of correctness
- Difficulties encountered:
  - Combination of sub-protocols
  - Non-determinism of protocols steps
  - Combination of parallel sessions
  - Definition of a dishonest agent ($\neq$ intruder), and analysis with it
  - Definition of non-repudiation and fair exchange properties