SECURITY PROPERTIES VERIFICATION OF SECURITY PROTOCOLS

Martin TOMÁŠEK*

*Department of Computers and Informatics, Faculty of Electrical Engineering and Informatics, Technical University of Košice, Letná 9, 042 00 Košice, Slovak Republic, tel.: +421 55 602 3178, e-mail: martin.tomasek@tuke.sk

ABSTRACT

We introduce security protocols by analyzing and verifying their properties. We use spi-calculus, an extension of the π-calculus, that enables us to consider cryptographic issues in more details. In this work we represent the security protocol as a process and we use the behavioral equivalences for describing secrecy and authenticity properties. Our goal is to design the practical procedure for verification of security protocols.

Keywords: *protocol, spi-calculus, secrecy, authenticity, π-calculus*

1. INTRODUCTION

Cryptographic protocols are used today to provide security in various applications. Cryptographic protocols are rules for exchange of messages between participants, and rely on cryptographic algorithms like encryption and decryption. Experience has shown that even very simple protocols which seem secure may have subtle flaws, even if the underlying cryptographic algorithms are secure. An extension of the π-calculus, the spi-calculus [1], was proposed as a formal notation for describing and reasoning about cryptographic protocols.

The objective of our work is to find a practical method of modeling and verifying cryptographic protocols using spi-calculus and validate it on specific communicating protocols. We analyze cryptographic protocols and their security properties. By means of basic knowledge about process algebras we use spicalculus for specification of cryptographic protocols. We develop and evaluate common formal method for the verification of cryptographic protocols.

2. CALCULUS OF SECURITY PROTOCOL

A protocol $P = C^* \cup C$, where clauses in C use symbols from Σ, predicates from $P^* \cup P$, and contain predicates from P^* only in the body. We can write $\bar{c}\langle M\rangle.P$ to denote a process that sends the message M on channel c after which it executes the process P. Then $c(x)M$ denotes a process that is listening on the channel c and if it receives some message M on this channel then it will execute the process $Q[M/x]$. We may compose these two processes in parallel to get a bigger process, denoted as $\bar{c}\langle M\rangle.P \mid c(x).Q$. Now the two smaller processes may communicate on the channel c after which they will execute the process $P \mid Q[M/x]$ [2]. The cryptographic protocol is communicating protocol, which uses the cryptography to achieve security goals. Basic cryptographic algorithms are DES, RSA, and DSA, and may be vulnerable if key is too short.

2.1. Abstract Syntax of the Calculus

The abstract syntax of the spi-calculus [1] is divided into two parts, terms and processes.

$L, M, N ::=$	**Terms**
$\mid n$	Name
$\mid (M, N)$	Pair
$\mid 0$	Zero
$\mid suc(M)$	Successor
$\mid x$	Variable
$\mid \{M\}_N$	Shared key encription
$P, Q, R ::=$	**Processes**
$\mid M\langle N\rangle.P$	Output - process is ready to output on channel m
$\mid M(x).P$	Input - process is ready to input from channel m
$\mid P \mid Q$	Composition - behaves as process P and Q running in parallel
$\mid (\nu n)P$	Restriction is a process that makes a new, private name n, which may occur in P
$\mid\ !P$	Replication behaves as a finite number of copies of P running in parallel
$\mid [M\ is\ N]P$	Match behaves as P if the terms M and N are the same; otherwise it is stuck (it does nothing)
$\mid 0$	Nil process does nothing
$\mid let\ (x, y) = M\ in\ P$	Pair splitting processes $let\ (x, y) = M\ in\ P$ behaves as $P[N/x][L/y]$ if the term M is the pair (N,L)
$\mid case\ M$ of $0 : P\ suc(x) : Q$	Integer case behaves as P if the term M is 0, as $Q[N/x]$ if M is $suc(N)$
$\mid case\ L$ of $\{x\}_N\ in\ P$	Shared key decryption

2.2. Semantic of the Calculus

Let $fn(M)$ and $fn(P)$ be a set of free names in term M and process P. Let $fv(M)$ and $fv(P)$ be the set of free variables in term M and process P. Closed processes are processes without any free variables. [3]

Reaction relation; $P \to Q$ means that there exists a reaction between subprocesses of P such that the whole can step to process Q:

$$\bar{m}\langle N\rangle.P \mid m(x).Q \to P \mid Q[N/x] \qquad \text{Interaction}$$

Then we define the **reduction relation** $>$ on closed processes:

$!P > P \mid !P$	Replication
$[M\ is\ M]P > P$	Match
$let\ (x,y) = (M,N)\ isP > P[M/x][N/y]$	Let
$case\ 0\ of\ 0 : P\ suc(x) : Q > P$	Zero
$case\ suc(M)\ of\ 0 : P\ suc(x) : Q > Q[M/x]$	Successor
$case\ \{M\}_N\ of\ \{x\}_N\ in\ P > P[M/x]$	Decrypt

Structural equivalence is a relation on closed processes that satisfies the following rules and equation:

$P \mid 0 \equiv P$	Nil
$P \mid Q \equiv Q \mid P$	Commitment
$P \mid (Q \mid R) \equiv (P \mid Q) \mid R$	Association
$(vm)(vn)P \equiv (vn)(vm)P$	Switch
$(vn)0 \equiv 0$	Drop
$(vn)(P \mid Q) \equiv P \mid (vn)Q$ if $n \notin fn(P)$	Extrusion

Reduction $\dfrac{P > Q}{P \equiv Q}$ Reflection $\dfrac{}{P \equiv P}$ Symmetry $\dfrac{P \equiv Q}{Q \equiv P}$

Transitivity $\dfrac{P \equiv Q\ Q \equiv R}{P \equiv R}$ Parameterization $\dfrac{P \equiv P'}{P \mid Q \equiv P' \mid Q}$ Restriction $\dfrac{P \equiv P'}{(vm)P \equiv (vm)P'}$

With these rules we can complete reaction rules as follow:

$$\frac{P \equiv P'\ P' \to Q'\ Q' \equiv Q}{P \to Q} \quad \frac{P \to P'}{P \mid Q \to P' \mid Q} \quad \frac{P \to P'}{(vn)P \to (vn)P'}$$

Abadi and Gordon [1] use **testing equivalence** as the notion of equivalence. Two processes are testing equivalent, written $P \simeq Q$, if they are indistinguishable to any other process. For specification of testing equivalence [4] we first define barbs. **Barbs** define a predicate describing the channels, where output process can communicate. A barb β is an input or an output channel, where output channels are marked by a barb $\bar{m}$. P exhibits barb β, written $P \downarrow \beta$, is defined:

$m(x).P \downarrow m$	Input
$\bar{m}\langle M\rangle.P \downarrow \bar{m}$	Output

Barb Parametrization $\dfrac{P \downarrow \beta}{P \mid Q \downarrow \beta}$ Barb Restriction $\dfrac{P \downarrow \beta\ \beta \notin \{m, \bar{m}\}}{(vm)P \downarrow \beta}$ Barb Structural $\dfrac{P \equiv Q\ Q \downarrow \beta}{P \downarrow \beta}$

Test is a closed process R and a barb β. The process R is trying to see if the tested process can be made to exhibit barb β:

$P \sqsubseteq Q = for\ any\ test\ (R, \beta),$ $if(P \mid R) \Downarrow \beta\ then(Q \mid R \Downarrow \beta)$	Testing Preorder
$P \simeq Q = P \sqsubseteq Q\ and\ Q \sqsubseteq P$	Testing Equivalence

The idea about testing equivalence builds De Nicola and Hennesy [5].

3. SECURITY PROPERTIES AND VERIFICATION PROCEDURE

For the verification of cryptographic protocols it is useful first define security properties [6] of these protocols.

Secrecy: M is secret if a session that contains M is indistinguishable from any session containing some data M_0 in place of M (observational equivalence property). Global secrecy is when a message is secret all the time. Local secrecy is when a message is secret till the corresponding session has not ended.

Authenticity: If A accepts a message M as coming from B then B actually sent M. If A received a message of form M_1 then B sent a message of form M_2. If A got a message of form M then B was active. If A has got a message M n times then B sent it n times.

In this project we want to proceed verification of cryptographic protocols by means of validation of the secrecy and the authenticity. We define the safety property.

Definition 4.1: Safety

- **Authenticity**: B always replies F to the message M that A sends; an attacker cant cause B to apply F to some other message.
- **Secrecy**: The message M can't be read in transit from A to B; if F doesn't reveal M, then the whole protocol doesn't reveal M.

Protocol is safe only if both conditions, authenticity and secrecy, are satisfied. In summary, we have:

$Inst(M) \simeq Inst_{spec}(M),\ for\ all\ M$	Authenticity
$Inst(M) \simeq Inst(M')\ if\ F(M) \simeq F(M'),$ $for\ all\ M, M'$	Secrecy

3.1. Verification Procedure

We designed following procedure to verify the safety properties of the communication protocols:

1. Write the protocol into convenient form. The best is writing it with messages.
2. Make the spi-calculus description of this protocol.
3. Make specification from description of this protocol.
4. Verify authenticity:
   - Make specification for authenticity.
   - Verify authenticity by exhibiting auxiliary equivalences (strong bisimilarity, barbed equivalence, and barbed congruence).

5. Verify secrecy:
   - Prove restricted version of secrecy property $Inst(M) \simeq Inst(M')$ if $F(x)$ is $\bar{c}\langle * \rangle$.
   - Prove full secrecy property $Inst(M) \simeq Inst(M')$ if $F(M) \simeq F(M')$ using auxiliary equivalences.
6. If both authenticity and secrecy are valid, then the protocol is secure.

4. EXAMPLES

Two principals A and B share the key K_{AB}, we assume there is a public channel c_{AB} that A and B use for communication. The protocol is simply that A sends a message M under K_{AB} to B, on c_{AB}.

To verify the safety properties of the protocol we use the above procedure.

1. Message 1 $A \to B : \{M\}_{K_{AB}}$ on c_{AB}
2. Specification in spi-calculus

$$\begin{aligned} A(M) &= \overline{c_{AB}}\langle \{M\}_{K_{AB}} \rangle \\ B_{spec} &= c_{AB}(x).case\ x\ of\ \{y\}_{K_{AB}}\ in\ F(y) \\ Inst_{spec}(M) &= (\nu K_{AB})(A(M) \mid B_{spec}(M)) \end{aligned}$$

3. The main definitions are:

$$\begin{aligned} Inst(M) &= (\nu c_{AB})(\overline{c_{AB}}\langle M \rangle.0 \mid c_{AB}(x).F(x)) \\ Inst_{spec}(M) &= (\nu c_{AB})(\overline{c_{AB}}\langle M \rangle.0 \mid c_{AB}(x).F(M)) \end{aligned}$$

4. **Proposition 5.1**: For any closed term M, $Inst(M) \simeq Inst_{spec}(M)$.
   Only commitments of $Inst(M)$ and $Inst_{spec}(M)$ are:

$$\begin{aligned} Inst(M) &\xrightarrow{\tau} (\nu c_{AB})(0 \mid F(M)) \\ Inst_{spec}(M) &\xrightarrow{\tau} (\nu c_{AB})(0 \mid F(M)) \end{aligned}$$

   From definition of barbed congruence we know, that strong bisimilarity implies barbed congruence and barbed congruence implies testing equivalence.

$$\begin{aligned} Inst(M) &\sim_s Inst_{spec}(M) \\ Inst(M) &\sim Inst_{spec}(M) \\ Inst(M) &\simeq Inst_{spec}(M) \end{aligned}$$

5. First we prove restricted version of secrecy property. **Lemma 5.2**: $Inst(M) \simeq Inst(M')$ if $F(M')$ is $\bar{c}\langle * \rangle$, for any closed terms M and M'. Only commitment of $Inst(N)$ is: $Inst(N) \xrightarrow{\tau} (\nu c_{AB})(0 \mid \bar{c}\langle * \rangle)$ and so clearly $Inst(M) \sim_s Inst(M')$. Like in previous, $Inst(M) \simeq Inst(M')$. Now we can make calculation of full secrecy property, $Inst(M) \simeq Inst(M')$ if $F(M) \simeq F(M')$. In special case, where $F(x)$ is $\bar{c}\langle * \rangle$ we can write $Inst(M, (x)\bar{c}\langle * \rangle)$. We assume that c is a fresh name and y fresh variable and we write $\tau.F(N)$ for $(\nu c)(\bar{c}\langle * \rangle \mid c(y).F(N))$. Only commitments are:

$$\begin{aligned} (\nu c)(c_{AB}(x).\bar{c}\langle * \rangle \mid c(y).F(N)) &\xrightarrow{c_{AB}} (x)\tau.F(N) \\ c_{AB}(x).\tau.F(N) &\xrightarrow{c_{AB}} (x)\tau.F(N) \end{aligned}$$

   From these, we have: $(\nu c)(c_{AB}(x).\bar{c}\langle * \rangle \mid c(y).F(N)) \sim_s c_{AB}(x)\tau.F(N)$.
   As follows using proposition $F(N) \simeq \tau.F(N)$, facts that testing equivalence is congruence and that strong bisimilarity implies testing equivalence, we have:

$$\begin{aligned} Inst_{spec}(N) &= (\nu c_{AB})(\overline{c_{AB}}\langle N \rangle.0 \mid c_{AB}(x).F(N)) \\ &\simeq (\nu c_{AB})(\overline{c_{AB}}\langle N \rangle.0 \mid \\ &\quad c_{AB}(x).(\tau.F(N))) \\ &\simeq (\nu c_{AB})(\overline{c_{AB}}\langle N \rangle.0 \mid (\nu c)(c_{AB}(x).\bar{c}\langle * \rangle \\ &\quad \mid c(y)F(N))) \\ &\equiv (\nu c)((\nu c_{AB})(\overline{c_{AB}}\langle N \rangle.0 \mid \\ &\quad (\nu c)(c_{AB}(x).\bar{c}\langle * \rangle)) \mid c(y)F(N)) \\ &= (\nu c)(Inst(N, (x)\bar{c}\langle * \rangle) \mid c(y).F(N)) \end{aligned}$$

   And we obtain equation: $Inst_{spec} \simeq (\nu c)(Inst(N, (x)\bar{c}\langle * \rangle) \mid c(y).F(N))$. With this equation, Lemma 5.2, Proposition 5.1 and assumption $F(M) \simeq F(M')$ we can make following calculation.

$$\begin{aligned} Inst(M) &\simeq Inst_{spec}(M) \\ &\simeq (\nu c)(Inst(M, (x)\bar{c}\langle * \rangle) \mid c(y).F(M)) \\ &\simeq (\nu c)(Inst(M', (x)\bar{c}\langle * \rangle) \mid c(y).F(M')) \\ &\simeq Inst_{spec}(M') \\ &\simeq Inst(M') \end{aligned}$$

6. Authenticity and secrecy property are valid, protocol is secure.

5. CONCLUSIONS

This work describes verification of cryptography protocols with emphasis on authenticity and secrecy properties using spi-calculus. The main task was to design a common procedure of the verification, which can be applied on any cryptographic protocol. Presented results are based on Abadi's and Gordon's testing equivalence and auxiliary equivalences [3]. This approach is more suitable for automation than solution designed by Woo and Lam [2]. Future extension of this work may be a software implementation of designed procedure.

ACKNOWLEDGEMENT

This work was supported by the Slovak Research and Development Agency under the contract No. APVV-0008-10.

REFERENCES

[1] ABADI, M. – GORDON, A. D.: "A calculus for cryptographic protocols: The spi calculus," *Information and Computation*, vol. 148, no. 1, pp. 1 – 70, 1999.

[2] WOO, T. Y. C. – LAM, S. S.: “A semantic model for authentication protocols,” 1993.

[3] ABADI. M. – GORDON, A. D.: “Reasoning about cryptographic protocols in the spi calculus,” in *In CONCUR’97: Concurrency Theory*. Springer-Verlag, 1997, pp. 59–73.

[4] CLEAVELAND, R. – HENNESSY, M.: “Testing equivalence as a bisimulation equivalence,” 1993.

[5] de NICOLA, R. – HENNESSY, M.: “Testing equivalences for processes,” *Theoretical Computer Science*, vol. 34, pp. 83–133, 1984.

[6] GORDON, A. D. – JEFFREY, A.: “Authenticity by typing for security protocols,” *Journal of Computer Security*, p. 2003.

Received March 10, 2014, accepted June 3, 2014

BIOGRAPHY

Martin Tomášek received the master degree in computer science in 1998 and PhD degree in software and information systems in 2005 both at the Faculty of Electrical Engineering and Informatics of the Technical University of Koice, Slovakia. Currently he is an associate professor at the Department of Computers and Informatics of the Faculty of Electrical Engineering and Informatics of the Technical University of Koice, Slovakia. His research interests include distributed systems, component-based systems, and concurrency theory.