A Finite Equivalence of Verifiable Multi-secret Sharing

Hui Zhao
Computer Science and Technology School, Shandong University of Technology, No. 12 Zhangzhou Road, Zhangdian, Zibo, Shandong 255049, China

Mingchu Li
Software School, Dalian University of Technology, No. 2 Liaoning Road, Ganjingzi, Dalian, Liaoning 116024, China

Kouichi Sakurai
Department of Informatics, Kyushu University, 744 Motooka, Nishi-ku, Fukuoka 819-0395, Japan

Yizhi Ren
Software School, Hangzhou Dianzi University, No. 2 Liaoning Road, Ganjingzi, Hangzhou, Zhejiang 116024, China

Jonathan Z. Sun
School of Computing, The University of Southern Mississippi, 118 College Drive, Box 5106, Hattiesburg, MS 39406

Fengying Wang[*]
Computer Science and Technology School, Shandong University of Technology, No. 12 Zhangzhou Road, Zhangdian, Zibo, Shandong 255049, China

Received 10 March 2011
Accepted 9 December 2011

Abstract

We give an abstraction of verifiable multi-secret sharing schemes that is accessible to a fully mechanized analysis. This abstraction is formalized within the applied pi-calculus by using an equational theory which characterizes the cryptographic semantics of secret share. We also present an encoding from the equational theory into a convergent rewriting system, which is suitable for the automated protocol verifier ProVerif. Based on that, we verify the threshold certificate protocol in ProVerif.

Keywords: Pi-calculus; Secret-sharing; Formal analysis; Protocol verifier.

1. Introduction

pi-calculus are now widely considered a particularly salient approach for formally analyzing security protocols.[1] One of the central challenges in the analysis of complex and industrial-size protocols is the expressiveness of the formalism used in the formal analysis and its capability to model complex cryptographic operations. While such protocols traditionally relied only on the basic cryptographic operations such as encryption and digital signatures, modern cryptography has invented more sophisticated primitives with unique security features that go far beyond the traditional understanding of cryptography to solely offer secrecy and authenticity of a communication. Secret share constitutes such a prominent primitive.

In 1994, Dawson et al.[12] proposed multi-secret sharing (MSS) schemes. In such schemes, several secrets can be shared during one secret-sharing process. In 2004, Yang et al. (YCH)[16] proposed a new MSS, which is based on two-variable one-way function and allows the construction of several secrets in parallel. In 2005, Shao

[*] Professor Fengying Wang is the corresponding author, E-mail: Eric.hui.zhao@gmail.com.

Published by Atlantis Press

and Cao (SC)[14] proposed an efficient verifiable multi-secret sharing based on YCH and feldman's schemes. In 2006, Zhao et al. (ZZZ)[15] proposed a practical verifiable multi-secret sharing based on YCH schemes.[16] These unique security features that secret shares offer combined with the possibility to efficiently implement some of these proofs have paved these proofs the way to constitute very powerful building blocks for the construction of sophisticated cryptographic protocol.

In a verifiable multi-secret sharing scheme with threshold (l, t), we have a set of l players, indexed 1, ..., l, and a trusted dealer. There is also a share verification algorithm and a share combining algorithm. In the dealing phase, the dealer takes messages which can be called dealer parameters as input and generates secret keys SK_1, ..., SK_l and verification keys VK_1, ..., VK_l. After the dealing phase, the dealer or player i can take as input messages which are called player parameters, along with the secret key SK_i assigned by dealer, and generate secret share D_i. Besides, the share verification algorithm takes secret share D_i and verification key VK_i as input messages to determine if the secret share D_i is valid. The share combining algorithm takes as input messages t valid secret shares and output a secret. A secret is a message or a sequence of messages that can be built upon dealer parameters and player parameters through basic cryptographic operations.

Compared with the single-secret sharing schemes, the verifiable multi-secret sharing schemes have the following security features:

- It allows the constructing of several secrets in parallel and the computation is efficient;
- It is multi-use (once the secret has been constructed, it is not necessary for the dealer to redistribute a fresh secret share over a security channel to every player);
- It allows the participant to verify the validity of secret shares of the other participants.

Due to the complexity of verifiable multi-secret sharing schemes, it is very difficult to devise the abstraction of secret-sharing proofs. In 1999, Yew et al. [18] defined an abstraction of single-secret sharing schemes in Coq, which is based on a clear separation between the modeling of reliable participants and that of the adversary. But this definition is too simple to hold all the security features of verifiable multi-secret sharing schemes above. [9-11]

Our main contributions are as follows: We give an abstraction of verifiable multi-secret sharing schemes within the applied pi-calculus[4] by using an equational theory which characterizes the cryptographic semantics of secret shares. Based on that, we transform our abstraction into a convergent rewriting system, which is suitable for ProVerif[17] , a well-established tool for the mechanized analysis of different security properties.

We express cryptographic protocols in the applied pi-calculus. We devise an equational theory that characterizes the semantic properties of secret shares, and that allows for abstract operation about such proofs. The design of the theory guarantees the soundness and the completeness of verifiable multi-secret sharing schemes with threshold (l, t) as well as the actual secret shares' properties: First, it is impossible or at least computationally infeasible, to produce any secret with knowledge of any $t-1$ or fewer valid secret shares. On the other hand, knowledge of any t or more valid secret shares makes all the secrets easily computable.

ProVerif[17] constitutes a well-established automated protocol verifier based on Horn clauses resolution that allows for the verification of observational equivalence and of different trace-based security properties such as authenticity. We present a mechanized encoding of our equational theory into a convergent rewriting system that ProVerif can efficiently cope with. We prove that the encoding preserves observational equivalence.

In this paper, we review the applied pi-calculus in section 2. Section 3 contains the equational theory for abstract operation about verifiable multi-secret sharing in the applied pi-calculus. This equational theory is rewritten into an equivalent finite theory in terms of a convergent rewriting system in section 4. Section 5 elaborates on the analysis of the threshold certificate protocol, the description of its security properties.

2. Review of The Applied Pi-calculus

The syntax of the applied pi-calculus[3] is given as follows. Terms are defined by means of a *signature* Σ, which consists of a set of function symbols, each with an arity. The set of terms T_Σ is the free algebra built from names, variables, and function symbols in Σ applied to arguments. We let μ range over names and variables. Terms are equipped with an equational theory E, i.e., an equivalence relation on terms that is closed under substitution of terms and under application of term contexts (terms with a hole). We write $E \mapsto M = N$ for an equality, modulo E.

The grammar of processes (or plain processes) is defined as follows. The *null* process 0 does nothing; The *restriction* process $n.P$ generates a fresh name n and then behaves as P; The *conditional* process *if* $M = N$ *then* P *else* Q behaves as P if $E \mapsto M = N$, and as Q otherwise; The *input* process $u(x).P$ receives a message N from the channel u and then behaves as $P\{N/x\}$; The *output* process $\bar{u}(N).P$ outputs the message N on the channel u and then behaves as P; The *replication* process $P \mid Q$ executes P and Q in parallel; $!P$ generates an unbounded number of copies of P.

A *context* is a process or an extended process with a hole. An *evaluation context* is a context without private function symbols whose hole is not under a replication, a conditional, an input, or an output process. A context $C[\ ]$ closes P if $C[P]$ is closed.

As in the pi-calculus, the semantics is defined in terms of *structural equivalence* ($\equiv$) and *internal reduction* ($\to$). Structural equivalence states which processes should be considered equivalent up to syntactic re-arrangement. Internal reduction defines the semantics for extended processes.

Definition 1. *Structural equivalence* ($\equiv$) *is the smallest relation on extended processes that satisfies the rules in Table 1 and that is closed under renaming of bound names and variables, and under application of evaluation contexts.*

Table 1. Structure Equivalence.

PAR-A	$A_1 \mid (A_2 \mid A_3) \equiv (A_1 \mid A_2) \mid A_3$
PAR-C	$A_1 \mid A_2 \equiv A_2 \mid A_1$
REPL	$!P \equiv P \mid !P$
RES-/0	$\nu n.\, 0 \equiv 0$
RES-C	$\nu u.\, \nu u'.\, A \equiv \nu u'.\, \nu u.\, A$
RES-PAR	$A_1 \mid \nu u{:}A_2 \equiv \nu u(A_1 \mid A_2)$ *if* $u \notin free(A_1)$
ALIAS	$\nu x.\, \{M/x\} \equiv 0$
SUBST	$\{M/x\} \mid A \equiv \{M/x\} \mid A\{M/x\}$
REWRITE	$\{M/x\} \equiv \{N/x\}$ *if* $E \mapsto M = N$

Definition 2. *Internal reduction* ($\to$) *is the smallest relation on extended processes that satisfies the rules in Table 2 and that is closed under structural equivalence and under application of evaluation contexts.*

Table 2. Internal reduction.

$\bar{a}(x).\, P \mid a(x).\, Q \to P \mid Q$
$\dfrac{E \mapsto M = N \quad M, N ground}{if \;\; M = N \;\; then \;\; P \;\; else \;\; Q \to P}$
$\dfrac{not \;\; E \mapsto M = N \quad M, N ground}{if \;\; M = N \;\; then \;\; P \;\; else \;\; Q \to Q}$

We write $P \Downarrow u$ to denote that process P can send a message on channel u. Observational equivalence ($\approx$) constitutes an equivalence relation that captures the equivalence of processes with respect to their dynamic behavior.

Definition 3. *Observational equivalence* ($\approx$) *is the largest symmetric relation* $\mathbb{R}$ *between closed extended processes with the same domain such that* $P \,\mathbb{R}\, Q$ *implies*:

- *if* $P \Downarrow u$, *then* $Q \Downarrow u$;
- *if* $P \to^{*} P'$, *then* $Q \to^{*} Q'$ *and* $P' \,\mathbb{R}\, Q'$ *for some* Q';
- $C[P] \,\mathbb{R}\, C[Q]$ *for all closing evaluation contexts* $C[\ ]$.

3. An Equational Theory of Secret-sharing

3.1. *An underlying cryptographic base theory*

The base equational theory E_{base} we consider in this paper is given in Table 3. In E_{base}, function *ntuples* is for constructing the tuple of n messages; enc_{sym}, dec_{sym}, enc_{asym}, and dec_{asym} are for encrypting and decrypting messages by symmetric and asymmetric cryptography, respectively; sign and check are for signing messages and verifying signatures; *pk* is for modeling public key and *h* is for hashing. We encode numbers in binary form via the functions $string_0$, $string_1$, and *empty* (e.g., a bitstring 10 would be encoded as $string_1(string_0(empty))$). Arithmetic operations are modeled as destructors. For instance, the greater-equal relation is defined by the destructor $ge(M_1, M_2)$, which returns true if M_1 is greater equal then M_2. With this encoding, numbers and cryptographic messages are disjoint sets of values. In our example theory, infix notation is used for the functions *eq*, $\wedge$, and $\vee$, which model equality test, conjunction, and disjunction.

Table 3. A base equational theory containing basic cryptographic primitives and logical operators.

Σ_{base} = {*ntuples*, *ith*$_n$, *enc*$_{sym}$, *dec*$_{sym}$, *enc*$_{asym}$, *dec*$_{asym}$, *sign*, *check*, *pk*, *h*, $\wedge$, $\vee$, *eq*, *true*; *false*}

E_{base} is the smallest equational theory satisfying the following equations defined over all x, y, z:

$ith_n(ntuples(x_1, \ldots, x_n)) = x_i.$
$dec_{sym}(enc_{sym}(x, y), y) = x.$
$dec_{asym}(enc_{asym}(x, pk(y)), y) = x.$
$check(sign(x, y), pk(y)) = x.$
$eq(x, x) = true.$
$ge(string_1(empty)), string_0(empty))) = true.$
$\wedge\ (true, true) = true.$
$\wedge\ (false, x) = false.$
$\wedge\ (x, false) = false.$
$\vee\ (false, false) = false.$
$\vee\ (true, x) = true.$
$\vee\ (x, true) = true.$

3.2. *Illustrative example*

A (*l*, *t*)-threshold signatures scheme is a protocol that allows any subset of *t* players out of *l* to generate a signature, but that disallows the creation of a valid signature if fewer than *t* players participate in the protocol[11].

3.3. *The equational theory for secret-sharing*

Our equational theory E_{base} for abstract operations about secret-sharing and its components are explained in the following. Secret-sharing process with threshold (*l*, *t*) is represented as a term of the form $SSP_{l,t}(\tau)$, name τ is used to identify specified secret-sharing process, we abuse notation by writing $\tau_{l,t}$ which represents $SSP_{l,t}(\tau)$; The secret key for secret share is represented as a term of the form $SSK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F})$, while natural number *m*, called the proof's identity Id, can be used to identify specified secret key in secret-sharing process and we have $m \in [1, l]$; Further, the secret share is represented as a term of the form $SS_{i,j,k}(\widetilde{N}, SSK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F}), \widetilde{F})$. In the definitions above, $\widetilde{M}$ and $\widetilde{N}$, denotes sequence $M_1...M_i$ and $N_1...N_j$ of terms, respectively, and $\widetilde{F}$ denotes sequence $F_1...F_k$ of (*i*, *j*)-functions which constitutes a function over those terms, see below; Hence, $SS_{i,j,k}$ is a function of arity *j*+*k*+1 and $SSK_{i,j,k}$ is a function of arity *i*+*k*+2.

The (*i*, *j*)-function *F* constitutes a constant without names and variables, which is built upon distinguished nullary functions α_i and β_i with $i \in \mathbb{N}$, $\mathbb{N}$ is the set of natural numbers.

Definition 4. *We call a term an* (*i*, *j*)*-function if the term contains neither names nor variables, and if for every* α_m *and* β_n *occurring therein, we have* $m \in [1, i]$ *and* $n \in [1, j]$.

The values α_i and β_j in *F* constitute placeholders for the terms M_i, called dealer parameter, and N_j, called player parameter. In our abstraction model, the relationship between secret and dealer parameters, player parameters can be defined through (*i*, *j*)-functions. For instance,

$$SS_{2,1,1}\ (h(M), SSK_{2,1,1}(SK, pk(SK), 1,\ \tau_{l,t},\ \widetilde{F}),\ \widetilde{F})\ ;$$

$$\widetilde{F} = sign(\beta_1, \alpha_1)$$

denotes a secret share for the secret *sign*(*h*(*M*), *SK*) in a (*l*, *t*)-threshold signatures scheme. In the dealing phase, the dealer generates a private key and public key pair (*SK*, *pk*(*SK*)), then the dealer uses $SSK_{2,1,1}$ to generate secret keys SK_1, ..., SK_l. After the dealing phase, player *i* can use $SS_{i,j,k}$ to generate secret share D_i with SK_i. Similarly to secret key for secret share $SSK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F})$, the corresponding verification key for secret share is represented as a term of form $SVK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F})$ and we have $m \in [1, l]$.

Verification of secret shares with respect to verification key for secret share is modeled as a function $SVer_{i,j,k}$ of arity *k*+2 that is defined by the following equational rule:

$$SVer_{i,j,k}(SSK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F}), SS_{i,j,k}(\widetilde{N}, SSK_{i,j,k}(\widetilde{M}, m, \tau_{l,t}, \widetilde{F}), \widetilde{F})) = true \qquad (1)$$

Combination of r secret shares with respect to (*i*, *j*)-functions is modeled as function $SCombin_{i,j,k,r}$ of arity *r*+*k* that is defined by the following equational rule:

$$SCombin_{i,j,k,r}\ (SS_{i,j,k}(\widetilde{N}, SSK_{i,j,k}(\widetilde{M}, i_1, \tau_{l,t}, \widetilde{F}), \widetilde{F}), \ldots, SS_{i,j,k}(\widetilde{N}, SSK_{i,j,k}(\widetilde{M}, i_r, \tau_{l,t}, \widetilde{F}), \widetilde{F})) = \widetilde{F}\ \{\widetilde{M}/\widetilde{\alpha}\}\ \{\widetilde{N}/\widetilde{\beta}\}$$

iff

- $i_m \neq i_n$ *for* $1 \leq m, n \leq r$ *and* $m \neq n$;
- $r \geq t.$ (2)

These rules guarantee in the abstract model the soundness and the completeness of the verifiable multi-secret sharing schemes with threshold (l, t) as well as the actual secret share properties that knowledge of any $t-1$ or fewer valid secret shares leaves the secret completely undetermined and knowledge of any t or more valid secret shares makes the secret easily computable.

4. Towards a Mechanized Analysis of Secret-sharing

The equational theory E_{SS} defined in the previous section is not suitable for existing tools for mechanized security protocol analysis. The reason is that the number of possible (i, j)-functions, and thus the number of equational rules in E_{SS}, is infinite. Hence, it is very difficult to devise the abstraction of secret-sharing proofs possibly generated by the environment, which represents the adversary with various capabilities. In this section, we specify an equivalent equational theory in terms of a convergent rewriting system. This theory turns out to be suitable for Proverif, a well established tool for mechanized verification of different security properties of cryptographic protocol specified in a variant of the applied pi-calculus.[17]

4.1. *A finite specification of secret-sharing*

The central idea of our finite equivalent theory is to focus on the secret-sharing proofs used within the process specification and to abstract away from the additional ones that are possibly generated by the environment. This makes finite the specification of the equational theory.

First, we track secret shares generated or combined in the process specification by a set TR of tetrads of the form (i, j, k, $\widetilde{F}$), where $\widetilde{F}$ is sequence of k (i, j)-functions of secret-sharing schemes. Second, we record h, which is the largest threshold of secret-sharing schemes used in the process specification. For term M and process P, we let $terms(M)$ denote the set of subterms of M and $terms(P)$ denote the set of terms in P. We can now formally define the notion of (TR, h)-validity of terms and processes.

Definition 5. *A term Z is* (TR, h)*-valid if and only if the following conditions hold*:

(i) *For every* $SSK_{\underset{\sim}{i,j,k}}(\widetilde{M}, \underset{\sim}{M}, N, \widetilde{F})$, $SVK_{i,j,k}(\underset{\sim}{\widetilde{M}}, M, N, \widetilde{F})$, $SS_{i,j,k}(\underset{\sim}{\widetilde{M}}, \underset{\sim}{M}, \widetilde{F})$, $SVer_{i,j,k}(M, N, \widetilde{F})$, *and* $SCombin_{i,j,\underset{\sim}{k,r}}(\widetilde{M}, \widetilde{F}) \in terms(Z)$, *we have*

   (a) $(i, j, k, \widetilde{F}) \in TR$;

   (b) *for every* F_l *in* $\widetilde{F}$, *we have* $F_l \in T_{\Sigma_{base} \cup \{\alpha_m, \ \beta_n | m \in [1,i], n \in [1,j]\}}$;

   (c) *for every* $(i, j, k, \widetilde{F}) \in TR$ *such that* $E_{SS} \mapsto \widetilde{F}' = \widetilde{F}$, *we have* $\widetilde{F}' = \widetilde{F}$.

(ii) *For every* $l \in \mathbb{N}$, α_l *and* β_l *occur in Z only inside of* (i, j)*-function of Z.*

(iii) *For every* $SSP_{l,t}(M) \in terms(Z)$, *we have* $l \in [1, h]$.

A process P is (TR, h)*-valid if and only if M is* (TR, h)*-valid for every* $M \in terms(P)$.

We check that each secret share generation, verification and combination is tracked in TR (i). We also check that the secret can be divided into at most h different secret shares in the process specification (iii).

4.2. *Compilation into finite form*

It now remains to encode the secret-sharing proofs generated by the environment. These proofs are possibly different from the ones specified in the process. We include in $\Sigma_{SS}^{TR,h}$ the function symbols $SSK_{i,j,k}^{\widetilde{F}}$, $SVK_{i,j,k}^{\widetilde{F}}$, $SS_{i,j,k}^{\widetilde{F}}$, $SVer_{i,j,k}^{\widetilde{F}}$, $SCombin_{i,j,k,r}^{\widetilde{F}}$. We then replace every term $SSK_{i,j,k}(\widetilde{M}, M, N, \widetilde{F})$, $SVK_{i,j,k}(\widetilde{M}, M, N, \widetilde{F})$, $SS_{i,j,k}(\widetilde{M}, M, \widetilde{F})$, $SVer_{i,j,k}(M, N, \widetilde{F})$, *and* $SCombin_{i,j,k,r}(\widetilde{M}, \widetilde{F})$ with $SSK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, M, N)$, $SVK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, M, N)$, $SS_{i,j,k}^{\widetilde{F}}(\widetilde{M}, M)$, $SVer_{i,j,k}^{\widetilde{F}}(M, N)$, *and* $SCombin_{i,j,k,r}^{\widetilde{F}}(\widetilde{M})$ respectively. Since $\widetilde{F}$ is uniquely determined by $SSK_{i,j,k}^{\widetilde{F}}$, $SVK_{i,j,k}^{\widetilde{F}}$, $SS_{i,j,k}^{\widetilde{F}}$, $SVer_{i,j,k}^{\widetilde{F}}$, and $SCombin_{i,j,k,r}^{\widetilde{F}}$, it can be omitted from the protocol specification.

For combination of different secret shares in (TR, h)-valid form, which are with the same secret in the same (l, t)-threshold secret-sharing process, it suffices to check whether the arity of secret shares is more that t. Thus, we include in $\Sigma_{SS}^{TR,h}$ the functions $PCombin_{i,j,k,r}^{\widetilde{F}}$ and $SCVer_{i,j,k,r}^{\widetilde{F}}$. $SCVer_{i,j,k,r}^{\widetilde{F}}$ is used to determine if the secret can be computable from r different secret shares in a secret-sharing scheme with threshold (l, t) and can be modeled as follows.

$$\begin{aligned}
& SCVer_{i,j,k,r}^{\widetilde{F}}(SS_{i,j,k}^{\widetilde{F}}(\widetilde{N}, SSK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, i_1, \tau_{l,t})), \ldots, \\
& SS_{i,j,k}^{\widetilde{F}}(\widetilde{N}, SSK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, i_r, \tau_{l,t}))) = eq(t, r) \vee \\
& SCVer_{i,j,k,r-1}^{\widetilde{F}}(SS_{i,j,k}^{\widetilde{F}}(\widetilde{N}, SSK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, i_1, \tau_{l,t})), \ldots, \\
& SS_{i,j,k}^{\widetilde{F}}(\widetilde{N}, SSK_{i,j,k}^{\widetilde{F}}(\widetilde{M}, i_{r-1}, \tau_{l,t}))) \\
& for\ r \succ 1 \qquad (3)
\end{aligned}$$

$$SCVer^{\widetilde{F}}_{i,j,k,1}(SS^{\widetilde{F}}_{i,j,k}(\widetilde{N},\ SSK^{\widetilde{F}}_{i,j,k}(\widetilde{M},\ i,\ \tau_{l,t}))=eq(t,\ 1) \quad (4)$$

Thus, combination of r different secret shares is modeled by the following equational rules:

$$SCombin^{\widetilde{F}}_{i,j,k,r}(\widetilde{M}) = PCombin^{\widetilde{F}}_{i,j,k,r}(\widetilde{M},\ SCVer^{\widetilde{F}}_{i,j,k,r}(\widetilde{M})) \quad (5)$$

$$PCombin^{\widetilde{F}}_{i,j,k,r}\ (SS^{\widetilde{F}}_{i,j,k}(\widetilde{N},\ SSK^{\widetilde{F}}_{i,j,k}\ (\widetilde{M},\ i_1,\ \tau_{l,t})),\ \ldots,$$
$$SS^{\widetilde{F}}_{i,j,k}(\widetilde{N},\ SSK^{\widetilde{F}}_{i,j,k}\ (\widetilde{M},\ i_r,\ \tau_{l,t}))) = \widetilde{F}\ \{\widetilde{M}/\widetilde{\alpha}\}\ \{\widetilde{N}/\widetilde{\beta}\} \quad (6)$$

The $PCombin^{\widetilde{F}}_{i,j,k,r}$ and $SCVer^{\widetilde{F}}_{i,j,k,r}$ functions are private, hence they cannot be used by the adversary.
We now define the dynamic compilation of term and process.

Definition 6. *The* (TR,h)*-dynamic compilation is the partial function* $\sigma : T_{\Sigma_{SS}} \rightarrow T_{\Sigma^{TR,h}_{SS}}$ *recursively defined as follows*:

$$SSK_{i,j,k}(\widetilde{M}, M, N, \widetilde{F})\,\sigma = SSK^{\widetilde{F}}_{i,j,k}(\widetilde{M}, M, N);$$

$$SVK_{i,j,k}(\widetilde{M}, M, N, \widetilde{F})\,\sigma = SVK^{\widetilde{F}}_{i,j,k}(\widetilde{M}, M, N);$$

$$SS_{i,j,k}(\widetilde{M}, M, \widetilde{F})\,\sigma = SS^{\widetilde{F}}_{i,j,k}(\widetilde{M}, M);$$

$$SVer_{i,j,k}(M, N, \widetilde{F})\,\sigma = SVer^{\widetilde{F}}_{i,j,k}(M, N);$$

$$SCombin_{i,j,k,r}(\widetilde{M}, \widetilde{F})\,\sigma = SCombin^{\widetilde{F}}_{i,j,k,r}(\widetilde{M});$$

$$f(M_1, \ldots, M_i)\,\sigma = f(M_1\sigma, \ldots, M_i\sigma);$$

$x = x$; $n = n$, *where*

- $(i, j, k, \widetilde{F}) \in TR$;
- $r \in [1, h]$.

The following theorem finally states that observational equivalence is preserved under dynamic compilation and hence asserts the soundness of the encoding from the infinite specification into the finite specification.

Theorem 1. *Let P and Q be* (TR, h)*-valid processes, and* σ *be the* (TR, h)*-dynamic compilation, we have that* $P\sigma \approx_{E^{TR,h}_{SS}} Q\sigma \Leftrightarrow A \approx_{E_{SS}} B$.

The proof of this theorem is given in Appendix.

5. Mechanized Analysis of Threshold Certificate Protocol

We then analyze the security properties of (l, t)-threshold certificate protocol[17] with $E^{TR,h}_{SS}$.

The goal of threshold certificate protocol is to enable secret-sharing scheme to resist player's cheating. Each player gets secret share and a (l, t)-threshold agreement certificate in share distributing phase of the proposed secret-sharing scheme. The (l, t)-threshold agreement certificate can prevent the leakage of the secret share and can be verified. Thus, any player who submits a false certificate will be detected. And, no information about the secret shares can be computed from the cheating.
The threshold certificate protocol is composed of three subprotocols: the secret distributing protocol, the secret reconstruction protocol and the secret recovering protocol. The secret distributing protocol allows players and off-line TTP to get secret share and (l, t)-threshold agreement certificate from dealer. The secret reconstruction protocol enables more than t players to reconstruct the secret. The recovering protocol enables more than t players and off-line TTP to recover the secret.
Every Player has a unique ID. Off-line TTP also has a group of special IDs. We assume further dealer has a key-pair called endorsement key (EK) for each secret-sharing scheme as well as a publicly known identity bsn_D.

5.1. *Secret distributing protocol*

Suppose that the dealer wants to share the secret $sign(h(N_D), SK)$ among players whose IDs are ID_1, ..., ID_l according to the (l, t)-threshold secret-sharing policy, where $ID_i \neq ID_j$ if $i \neq j$. (Note that ID_1, ..., ID_l are the public IDs of the players.) For convenience, player ID_i denotes a player whose ID is ID_i. Let the special IDs of the off-line TTP be ID_{T_i} where $ID_{T_i} \notin \{ID_1, ..., ID_l\}$; and $i = 1; 2; \ldots; l{-}t+1$.

(i) The dealer uses a $(2l - t + 1,\ l + 1)$-threshold secret-sharing scheme to partition the secret $sign(h(N_D), SK)$ into $2l{-}t{+}1$ shares $(ID_1, ttpss_1)$, $(ID_2, ttpss_2)$, …, $(ID_l, ttpss_l)$ and $(ID_{T_1}, ttpss_{T_1})$, $(ID_{T_2}, ttpss_{T_2})$, ..., $(ID_{T_{l-t+1}}, ttpss_{T_{l-t+1}})$.

(ii) The dealer generates the signatures $sign(ttpss_1, SK)$, $sign(ttpss_2, SK)$, …, $sign(ttpss_{T_{l-t+1}}, SK)$.

(iii) The dealer uses a (l, t)-threshold secret-sharing scheme to partition secret $sign(h(N_D), SK)$ into l shares (ID_1, ss_1), (ID_2, ss_2),..., (ID_l, ss_l).

(iv) The dealer sends $(ID_{T_1}, ttpss_{T_1})$, $(ID_{T_2}, ttpss_{T_2})$, ..., $(ID_{T_{l-t+1}}, ttpss_{T_{l-t+1}})$ to the off-line TTP over a secure channel.

(v) The dealer sends $(ttpss_i, sign(ttpss_i, SK), ss_i)$ to player ID_i over a secure channel, for each $i = 1, 2, ..., l$.

It is noted that the part $(ttpss_i, sign(ttpss_i, SK))$ is called the (l, t)-threshold agreement certificate of the player ID_i.

In our calculus, we can model the dealer in the secret distributing protocol according to as follow.

$\mathit{define}\ \mathit{ssproof} = \mathit{sign}(\mathit{beta}1, \mathit{alpha}1).$

$\mathit{dealer}_1 = \prod_{ID_i \in ID}.$

$\mathit{let}\ \mathit{ttpssk} = SSK^{ssproof}_{2,1,1}(SK, pk(SK), i, SSP_{2l-t+1,\, l+1}(pi))\ \mathit{in}$

$\mathit{let}\ \mathit{ttpsvk} = SVK^{ssproof}_{2,1,1}(SK, pk(SK), i, SSP_{2l-t+1,\, l+1}(pi))\ \mathit{in}$

$\mathit{let}\ \mathit{ttpss} = SS^{ssproof}_{2,1,1}(h(N_D), \mathit{ttpssk})\ \mathit{in}$

$\mathit{let}\ \mathit{sscert} = \mathit{sign}(\mathit{ttpss}, SK)\ \mathit{in}$

$\mathit{let}\ \mathit{ssk} = SSK^{ssproof}_{2,1,1}(SK, pk(SK), i, SSP_{l,\, t}(pi))\ \mathit{in}$

$\underline{\mathit{let}\ s}\mathit{s} = SS^{ssproof}_{2,1,1}(h(N_D), \mathit{ssk})\ \mathit{in}$

$\overline{\mathit{oeb}}(ID_i, \mathit{ttpss}, \mathit{sscert}, \mathit{ss})) \mid !\mathit{oeb}(= ID_i).\ \overline{\mathit{oeb}}(ID_i, \mathit{ttpsvk}).$

$\mathit{dealer}_2 = \prod_{ID_{T_j} \in ID_T}.$

$\mathit{let}\ \mathit{ttpssk} = SSK^{ssproof}_{2,1,1}(SK, pk(SK), l+j, SSP_{2l-t+1,\, l+1}(pi))\ \mathit{in}$

$\underline{\mathit{let}\ t}\mathit{tpss} = SS^{ssproof}_{2,1,1}(h(N_D), \mathit{ttpssk})\ \mathit{in}$

$\overline{\mathit{oeb}}\ ((ID_{T_j}, \mathit{ttpss}).$

$\mathit{dealer} = \nu pi.\ \nu SK.\ \nu N_D.$

$(!\,\bar{c}(\mathit{bsn}_D, pk(SK)) \mid \mathit{dealer}_1 \mid \mathit{dealer}_2).$

Here the define statement defines an abbreviation *ssproof* for the (i, j)-function we use in all secret-sharing proofs.

5.2. *Secret reconstruction protocol*

After successfully executing the secret distributing protocol, each player gets secret share and a (l, t)-threshold agreement certificate from dealer. Players ID_{i_1}, ID_{i_2}, ..., $ID_{i_k} \in ID_G$ will reconstruct the secret where $ID_G \subseteq ID$, they may perform the following procedure:

(i) Each player ID_{i_j} submits her/his $(2l - t + 1, l + 1)$-threshold share, $(ttpss_{i_j}, sign(ttpss_{i_j}, SK))$, which we call the player ID_{i_j}'s (l, t)-threshold agreement certificate $sscert_{i_j}$.

(ii) Each player verifies the (l, t)-threshold agreement certificate $(ttpss_{i_j}, sscert_{i_j})$ by the equation $check(sscert_{i_j}, pk(SK)) = true$ or *not*, for each $j = 1, 2, ..., k$. If the number of the certificates which pass the verification is less than t, then they stop the procedure according to the (l, t)-threshold access structure. Otherwise, they perform the following steps.

(iii) Each player submits her/his secret share (ID_{i_j}, ss_{i_j}) to reconstruct the secret $sign(h(N_D), SK)$.

The process for the secret reconstruction protocol is reported as follow.

$\mathit{recon}^i_1 = \bar{c}(\mathit{ttpss}, \mathit{ttpcert}).\ cp^{i,2}(= ID_i).\ \bar{c}(ID_i, \mathit{ss}).$

$\mathit{recon}^i_2 = c(\mathit{ttpss}, \mathit{ttpcert}).$

$\quad \underline{\mathit{if}\ \mathit{check}(\mathit{ttpcert}}, PK)\ \underline{= \mathit{tru}}\mathit{e}\ \mathit{then}$

$\quad \underline{\mathit{auth}}\mathit{enticate}^i(\mathit{ttpss}).\ cp^{i,2}(\mathit{ttpss}, \mathit{ttpc}\underline{\mathit{ert}).}$

$\quad cp^{i,3}(ID_j, \mathit{ttpss}, \mathit{ttpcert}), c(= ID_j, \mathit{ss}),\ cp^{i,1}(\mathit{ss}).$

$\mathit{recon}^i_3 = !\mathit{init}^i(ID_j).\ (\mathit{recon}^i_1 \mid \mathit{recon}^i_2).$

$\mathit{recon}^i_4 = cp^{i,\,1}(ss_1).\ cp^{i,\,1}(ss_2).\ \ldots\ .\ cp^{i,\,1}(ss_t).$

$\quad \underline{\mathit{let}\ s = 1}\mathit{th}_1(\mathit{SCombin}^{ssproof}_{2,1,1,t}(ss_1, ss_2, \ldots, ss_t))\ \mathit{in}$

$\quad \mathit{obtain}^i(s).$

$\mathit{recon}^i_5 = !\underline{cp}^{i,\,3}(ID_x, \mathit{ttpss}, \mathit{ttpcert}),\ \overline{\mathit{oeb}}(ID_i, \underline{ID_x, \mathit{ttp}}\mathit{ss}) \mid$

$\quad \mathit{oeb}(ID_i, ID_i, \mathit{ttpss}) \mid \mathit{oeb}(= ID_i, s).\ \mathit{obtain}^i(s).$

$\mathit{player} = c(= \mathit{bsn}_D, PK).$

$\quad \prod_{ID_i \in ID_G}.\ \mathit{oeb}(= ID_i, \mathit{ttpss}, \mathit{ttpcert}, \mathit{ss}).\ cp^{i,\,1}(\mathit{ss}).$

$\quad cp^{i,\,2}(\mathit{ttpss}, \mathit{ttpcert}).\ (\mathit{recon}^i_3 \mid \mathit{recon}^i_4 \mid \mathit{recon}^i_5).$

To simplify our model, here the restricted channel $cp^{i,\,2}$ is used to decide if at least t different (l, t)-threshold agreement certificates pass the verification of ID_i.

5.3. *Secret recovering protocol*

Suppose that some of the players don' t submit their (l, t)-threshold secret shares. The remaining players may send $\{(ID_{i_j}, ttpss_{i_j})\}^k_{j=1}$ to the off-line TTP and request her/him to reconstruct the secret. In the following, we present the procedure the offline TTP performs in response to the request:

(i) The off-line TTP verifies the received $(2l-t+1, l+1)$-threshold secret shares. If $k \prec t$, he rejects the request and stops the procedure. Otherwise, she/he performs the following steps.

(ii) The off-line TTP uses the $(2l-t+1, l+1)$-threshold secret shares $\{(ID_{i_j}, ttpss_{i_j})\}^k_{j=1} \cup \{(ID_{T_j}, ttpss_{T_j})\}^{l-t+1}_{j=1}$ to reconstruct the secret $sign(h(N_D), SK)$.

(iii) The off-line TTP sends the secret $sign(h(N_D), SK)$ to the player ID_{i_j}, for each $j = 1, 2, ..., k$.

It is obvious that the off-line TTP will solve the secret $sign(h(N_D), SK)$ for all players ID_{i_j} who submitted the (l, t)-threshold agreement certificates which are used to solve $sign(h(N_D), SK)$.

The process for the secret recovering protocol is reported as follow.

$$recov_1 = \prod_{ID_{T_j} \in ID_T} . \; oeb(= ID_{T_j}, ttpss). \; ! \; \overline{cp^{ttp,j}} \; (ttpss).$$

$$recov_2 = \prod_{ID_i \in ID_G} . \; !oeb(= ID_i, ID_x, ttpss). \; \overline{oeb} \, (ID_x).$$

```
        $oeb(= ID_x, ttpsvk).$
        $\underline{if\ SVer}_{2,1,1}^{ssproof}\ (ttpsvk, ttpss) = true\ then$
        $\underline{cp^{ttp}\ (ID_i, tt}pss).\ ccp^{ttp}\overline{(=} ID_i, s).$
        $distribute^{ttp}\ (ID_x, s).\ \overline{oeb}\,(ID_x, s)\ |$
        $cp^{ttp}(= ID_i, ttpss_1).\ cp^{ttp}(= ID_i, ttpss_2).\ ....$
        $cp^{ttp}(= ID_i, ttpss_t).$
        $cp^{ttp,\,1}(ttpss_{t+1}).\ cp^{ttp,\,2}(ttpss_{t+2}).\ ....$
        $cp^{ttp,\,l-t+1}\ (ttpss_{l+1}).$
        $let\ s = 1th_1(\ SCombin_{2,1,1,l+1}^{ssproof}\ (ttpss_1, ttpss_2, ...,$
        $tt\underline{pss_{1+1}}))\ in$
        $!\ ccp^{ttp}\ (ID_i, s).$
$ttp = recov_1 \mid recov_2$
```

5.4. *Authenticity of the protocol*

We will now discuss the authenticity property of the threshold certificate protocol and how to model it in our calculus. Firstly, we can define this protocol as follow:

$System = dealer \mid player \mid ttp.$

The security goal of threshold certificate protocol is to enable secret-sharing scheme to resist player's cheating. Thus, any player who submits a false certificate will be detected. And, no information about the secret shares can be computed from the cheating.

$GS(ID_G, ID_j, t)$ is defined to include all the sets which contain t elements different from ID_j in ID_G. Thus, the authenticity property of threshold certificate protocol is defined as the fulfillment of the following trace properties:

$$\forall ID_i \in ID_G. \; \overline{obtain^i} \; (sign(h(N_D), SK) \rightarrow (\bigvee_{ID_{G_i} \in GS(ID_G, ID_i, t-1)} . \bigwedge_{j \in ID_{G_i}} . \bigvee_{ID_{G_j} \in GS(ID_G, ID_j, t-1)} . \bigwedge_{k \in ID_{G_j}} . \; \overline{authenticate^j} \; (SS_{2,1,1}^{ssproof} (\, h(N_D), SSK_{2,1,1}^{ssproof} (SK, pk(SK), k, SSP_{2l-t+1,\, l+1}(pi)))) \vee \overline{distribute^{ttp}} ((ID_i, sign(h(N_D), SK)). \tag{7}$$

Which means that if a player obtains the secret $sign(h(N_D), SK)$, then either there exists at least $t\ -1$ other players who verify at least $t\ -1$ player's certificates in the same run or the secret $sign(h(N_D), SK)$ is from TTP to resist player's cheating.

Trace properties such as above can be verified with the mechanized prover ProVerif.

In ProVerif script, the parallel compositions are replaced with replicated inputs, for example, $\prod_{ID_i \in ID}$ are replaced with $!exp\_ID(ID_i, i)$ on the restricted channel exp_ID. We also add the events $\overline{authenticate}$($authenticate_i$, $ttpss$), $\overline{obtain}$ ($obtain_i$, s), and $\overline{distribute}$(ID_x, s) just before $authenticate^i$ ($ttpss$), $obtain^i$ (s), and after $distribute^{ttp}$ (ID_x, s) respectively.

ProVerif shows that the threshold certificate protocol satisfies the authenticity property.

6. Conclusion

We devise an abstraction of verifiable multi-secret sharing schemes in the applied pi-calculus. An equational theory for terms characterizes the semantic properties of secret share. Moreover, we propose an encoding into a finite specification in terms of a convergent rewriting system that is accessible to a fully mechanized analysis.

We regard the threshold certificate protocol as the case study, because this protocol packages many ideas that appear in the field of secret-sharing. Furthermore, this case study contributes to the results for the specification and verification of verifiable multi-secret sharing protocol that should be useful beyond the analysis of the threshold certificate protocol.

Acknowledgements

This research is supported by the Natural Science Foundation of Shandong Province under Grant ZR2010FL003.

References

1. M. Abadi, Secrecy by typing in security protocols, *J. the ACM.* 46(5) (1999) 749-786.
2. M. Abadi, and C. Fournet, Mobile values, new names, and secure communication, in *Proc. 28th Symposium on Principles of Programming Languages* (ACM Press, 2001), pp. 104-115.

3. M. Abadi, B. Blanchet, and C. Fournet, Just fast keying in the pi calculus, *ACM Transactions on Information and System Security*. 10(3) (2007) 9.
4. M. Abadi and A. D. Gordon, A calculus for cryptographic protocols: The spi calculus, *Information and Computation*. 148(1) (1999) 1-70.
5. H. Zhao, M. Li, K. Sakurai and Y. Ren, A Finite Equivalence Theory of Verifiable Multi-secret Sharing, in *Proc. Computer Security Symposium* (Okinawa, Japan, 2009), pp. 1032-1041.
6. H. Zhao, M. Li, K. Sakurai and Y. Ren, Mechanized Analysis of Verifiable Multi-secret Sharing in the Applied Pi-calculus, in *Proc. Symposium on Cryptography and Information Security* (Otsu, Japan, 2010), pp. 953-958.
7. L. Bachmair and H. Ganzinger, Rewrite-based equational theorem proving with selection and simplification, *J. of Logic and Computation*. 4(3) (1994) 217-247.
8. M. Backes, A. Cortesi, R. Focardi, and M. Maffei, A calculus of challenges and responses, in *Proc. 5th ACM Workshop on Formal Methods in Security Engineering* (ACM Press, 2007), pp. 101-116.
9. A. Shamir, How to share a secret, *Communications of the ACM*. 22(11) (1979) 612-613.
10. P. Feldman, A practical scheme for non-interactive verifiable secret sharing, in *Proc. 28th Annual IEEE Symposium on Foundations of Computer Science* (1987), pp. 427-437.
11. V. Shoup, Practical threshold signatures, in *Proc. EUROCRYPT 2000* (Springer-Verlag, 2000), pp. 207-220.
12. A. J. He, E. Dawson, Multistage secret sharing based on one-way function, *Electronics Letters*. 30(19) (1994) 1591-1592.
13. H.-Y. Chien, J. K. Tseng, A practical (*t*, *n*) multi-secret sharing scheme, *IEICE Transactions on Fundamentals of Electronics, Communications and Computer 83-A*. 12(2000) 2762-2765.
14. J. Shao, Z. F. Cao, A new efficient (*t*, *n*) verifiable multi-secret sharing (VMSS) based on YCH scheme, *Applied Mathematics and Computation*. 168(1) (2005) 135-140.
15. J. Zhao, J. Zhang, R. Zhao, A practical verifiable multi-secret sharing scheme, *Computer Standards and Interfaces*. 29(1) (2007) 138-141.
16. C. C. Yang, T. Y. Chang, M. S. Hwang, A (*t*, *n*) multi-secret sharing scheme, *Applied Mathematics and Computation*. 151 (2004) 483-490.
17. B. Blanchet, An efficient cryptographic protocol verifier based on Prolog rules, in *Proc. 14th IEEE Computer Security Foundations Workshop* (IEEE Computer Society Press, 2001), pp. 82-96.
18. K. M. Yew, M. Z. Rahman, and S. P. Lee, Formal Verification of Secret Sharing protocol Using Coq, in *Proc. 5th Asian Computing Science Conference on Advances in Computing Science table of contents* (Springer-Verlag , 1999), pp. 381-382.
19. Chao-Wen Chan, Chin-Chen Chang and Zhi-Hui Wang, Cheating Resistance for Secret Sharing, in *Proc. IEEE International Conference on Networks Security,Wireless Communications and Trusted Computing* (2009), pp 840-846.
20. H. Zhao, M. Li and X. Fan, A Formal Separation Method of Protocols to Eliminate Parallel Attacks in Virtual Organization, Security and Communication Networks. 4(12) (2010) 1461-1468.

Appendix A. The Proof of Theorem 1

We first define the notion of an execution trace. This requires reviewing the labeled operational semantics that allow us to reason about processes that interact with their environment.[3]

Definition 7 *The set of execution traces of an extended process P, written traces(P), is defined as follows*:
$traces(P) = \{\mu_1\ \phi(P_1),\ \ldots,\ \mu_n\ \phi(P_n)\} | P \rightarrow^* \xrightarrow{\mu_1} P_1\ \ldots \rightarrow^* \xrightarrow{\mu_n} P_n\}$.

In the following, we let *s* range over execution traces. We then review the notions which come from [3].

Definition 8 *Two term M and N are equal in a frame* ϕ, *written* $(M = N)\phi$, *if and only if* $\phi \equiv \nu \tilde{n}.\sigma$, $M\sigma \equiv N\sigma$, *and* $\{\tilde{n}\} \cap (fn(M) \cup fn(N)) = \varnothing$ *for some name en and substitution* σ.

Definition 9 *Two closed frame* ϕ *and* ψ *are statically equivalent, written* $\phi \approx^s \psi$ *if and only if* $dom(\phi) = dom(\psi)$ *and for all terms M and N, it holds that* $(M = N)\ \phi$ *if and only if* $(M = N)\psi$.
We say that two closed extended processes are statically equivalent, written $P \approx^s Q$ *if and only if their frames are statically equivalent.*

We now define the notion of labeled bisimilarity[3], which constitutes an equivalent notion of observational equivalence. Labeled bisimilarity does not rely on the universal quantification over evaluation contexts used in the definition of observational equivalence.

Definition 10 *Labelled bisimilarity* ($\approx^l$) *is the largest symmetric relation* $\mathbb{R}$ *on closed extended processes such that* $P\ \mathbb{R}\ Q$ *implies*:

- $P \approx^s Q$;
- *if* $P \rightarrow P'$, *then* $Q \rightarrow Q'$ *and* $P'\ \mathbb{R}\ Q'$ *for some* Q';

- *if* $P \xrightarrow{\mu} P'$ *and* $fv(\mu) \subseteq dom(P)$ *and* $bn(\mu) \cap fn(Q) = \varnothing$, *then* $Q \rightarrow^{*} \xrightarrow{\mu} \rightarrow^{*} Q'$ *and* $P' \mathbb{R} Q'$ *for some* Q'.

We finally state the well-known equivalence between observational equivalence and labeled bisimilarity[3].

Theorem 2 *Observational equivalence coincides with labelled bisimilarity*: $\approx\, = \,\approx^{l}$.

Lemma 1 *let* ϕ *be an* (*TR*, *h*)-*valid frame and* σ *be an* (*TR*, *h*)-*compilation. Then for any ground terms* M_1, $M_2 \in T_{\Sigma_{SS}}$ *in* (*TR*, *h*)-*normal form with respect to* ϕ, *we have* $E_{SS} \mapsto M_1 = M_2 \;\Leftrightarrow\; E_{ss}^{TR,h} \mapsto M_1\sigma = M_2\sigma$.

Proof. We prove the $\Leftarrow$ implication by induction on the length of the derivation of M_1. We first discuss the interesting base case:
$M_1\sigma = SVer^{\tilde{F}}_{i,j,k}(SVK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, m, \tau_{l,t}), SS^{\tilde{F}}_{i,j,k}(\tilde{N}\sigma, SSK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, m, \tau_{l,t}), M_2\sigma = True$, it must be the case that $(i, j, k, \tilde{F}) \in TR$, otherwise M_1 is not in (*TR*, *h*)-normal form with respect to ϕ. Therefore, we get $M_1 = true$. We have $M_1 = M_2$ as desired.
$M_1\sigma = SCombin^{\tilde{F}}_{i,j,k,r}(SS^{\tilde{F}}_{i,j,k}(\tilde{N}\sigma, SSK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, i_1, \tau_{l,t})), \ldots, SS^{\tilde{F}}_{i,j,k}(\tilde{N}\sigma, SSK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, i_r, \tau_{l,t}))), M_2\sigma = \tilde{F}\{\widetilde{M}\sigma/\tilde{\alpha}\}\{\tilde{N}\sigma/\tilde{\beta}\}$, it must be the case that

- $(i, j, k, \tilde{F}) \in TR$;
- $i_m \neq i_n$ *for* $1 \leq m, n \leq r$ *and* $m \neq n$.

Otherwise M_1 is not in (*TR*, *h*)-normal form with respect to ϕ.
Further, we get that $SCVer^{\tilde{F}}_{i,j,k,r}(SS^{\tilde{F}}_{i,j,k}(\tilde{N}\sigma, SSK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, i_1, \tau_{l,t})), \ldots, SS^{\tilde{F}}_{i,j,k}(\tilde{N}\sigma, SSK^{\tilde{F}}_{i,j,k}(\widetilde{M}\sigma, i_r, \tau_{l,t}))) = true$.
Thus, we get $r \geq t$ and $M_1 = \tilde{F}\{\widetilde{M}/\tilde{\alpha}\}\{\tilde{N}/\tilde{\beta}\}$. We have $M_1 = M_2$ as desired.
The proof of the implication is similar.

Lemma 2 *let A be an extended process such that* $A \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P)$, *for some* (*TR*, *h*)-*valid extended process* $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P)$, *let* σ *be the* (*TR*, *h*)-*compilation. Then the following statements hold*:

(i) *For every B,* $A \rightarrow_{E_{SS}} B$ *if and only if there exists an* (*TR*, *h*)-*valid extended process* $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}'/\tilde{x}'\}|\, P')$ $\equiv_{E_{SS}} B$ *such that* $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P)\sigma \rightarrow_{E_{SS}^{TR,h}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}'/\tilde{x}'\}|\, P')\sigma$.

(ii) *For every* μ *containing only terms in* (*TR*, *h*)-*normal form with respect to* $v\tilde{n}.v\tilde{y}.\{\widetilde{M}/\tilde{x}\}$ *and every B,* $A \xrightarrow{\mu}_{E_{SS}} B$ *if and only if there exist an* (*TR*, *h*)-*valid extended process* $v\tilde{n}'.v\tilde{y}'.(\{\widetilde{M}'/\tilde{x}'\}\,|\, P') \equiv_{E_{SS}} B$ *such that* $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P)\sigma \xrightarrow{\mu}_{E_{SS}} v\tilde{n}'.v\tilde{y}'.(\{\widetilde{M}'/\tilde{x}'\}|P')\sigma$ where

(a) *If* $\mu = a(M)$, *then* $\tilde{n} = \tilde{n}'$, $\tilde{y} = \tilde{y}', x$, *for some* $x \notin \{\tilde{x}\}$, *and* $\{\widetilde{M}'/\tilde{x}'\} = \{\widetilde{M}/\tilde{x}\}|\{M/x\}$.

(b) *If* $\mu = \bar{a}(b)$, *then* $\tilde{n} = \tilde{n}'$, $\tilde{y} = \tilde{y}'$, *and* $\{\widetilde{M}'/\tilde{x}'\} = \{\widetilde{M}/\tilde{x}\}$.

(c) If $\mu = vb.\bar{a}(b)$, *then* $\tilde{n} = \tilde{n}'$, $\tilde{y} = \tilde{y}'$, *and* $\{\widetilde{M}'/\tilde{x}'\} = \{\widetilde{M}/\tilde{x}\}|\{M/x\}$.

(d) *If* $\mu = vx.\bar{a}(x)$, *then* $\tilde{n} = \tilde{n}'$, $\tilde{y} = \tilde{y}'$, *and* $\{\widetilde{M}'/\tilde{x}'\} = \{\widetilde{M}/\tilde{x}\}|\{M/x\}$, *for some* (*TR*, *h*)-*valid M.*

Proof. We prove statement 1 by case on the internal reduction rule. Let us first deal with the "only if" implication.
COMM: There exist M, x, Q, P_1, P_2 such that $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P) \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|Q|\bar{a}(M).P_1|a(x).P_2$ and $Q\,|\,\bar{a}(M).P_1\,|a(x).P_2$ is (*TR*, *h*)-valid. By α-renaming, we can assume that $x \notin fv(P_1)$. We also have that $B \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|Q|P_1|P_2\{M/x\}$.
By ALIAS, RES-PAR, and SUBST, we get $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P) \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.vx.(\{\widetilde{M}/\tilde{x}\}|\{M/x\}|Q|\bar{a}\,(\mathrm{x}).P_1|a(x).P_2$. Since σ behaves as the identity function on variables and names and it is defined on (*TR*, *h*)-valid terms and processes, we get $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P)\,\sigma \equiv_{E_{SS}^{TR,h}} v\tilde{n}.v\tilde{y}.v\tilde{x}.(\{\widetilde{M}/\tilde{x}\}|\{M/x\}|Q|\bar{a}\,(\mathrm{x}).P_1|a(x).P_2)\,\sigma$. We also have that $v\tilde{n}.v\tilde{y}.vx.(\{\widetilde{M}/\tilde{x}\}|\{M/x\}|Q|\bar{a}\,(x).P_1|a(x).P_2)\,\sigma \rightarrow_{E_{SS}^{TR,h}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|Q|P_1|P_2\{M/x\})\,\sigma$, as desired.
Notice that internal reduction is closed by structural equivalence. It is easy to see that $P_2\{M/x\}$ is (*TR*，*h*)-valid since *M* occurs in the (*TR*，*h*)-valid process $\bar{a}(M).P_1$ and it is thus (*TR*，*h*)-valid as well.
THEN: There exist M, N, Q, P_1, P_2 such that $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\, P) \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\mathrm{Q}|\, if(M{=}N)\, then\, P_1\, else\, P_2)$, for some M, N, Q, P_1, P_2 such that the process (Q| *if* (*M*=*N*) *then* P_1 *else* P_2) is (*TR*, *h*)-valid and $E_{ss} \mapsto M\{\widetilde{M}/\tilde{x}\} = N\{\widetilde{M}/\tilde{x}\}$. We also have that $B \equiv_{E_{SS}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|Q|P_1$. Similarly, we get $v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)\,\sigma \equiv_{E_{SS}^{TR,h}} v\tilde{n}.v\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|Q|\, if\,(M = N)\; then\; P_1\; else\; P_2)\,\sigma$. Since $\widetilde{M}$ is in (*TR*, *h*)-normal form with respect to and *M* is (*TR*, *h*)-valid, it is easy to see $(M\{\widetilde{M}/\tilde{x}\})$ is in (*TR*, *h*)-normal form with respect to $v\tilde{n}.v\tilde{y}.\{\widetilde{M}/\tilde{x}\}$. The reason is the same for $(N\{\widetilde{M}/\tilde{x}\})$. By Lemma1, we get $E_{ss}^{TR,h} \mapsto$

$(M\{\widetilde{M}/\tilde{x}\})\sigma$ =(N$\{M/x\})\sigma$. The result follows from THEN and structural equivalence.

ELSE: The reasoning is similar to the one in the previous item.

We now prove that the process reduction, as defined by the labeled transition systems, is preserved as well. We proceed by cases on the label:

If $\mu = a(M)$, there exist x, P', Q such that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P) \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}(\{\widetilde{M}/\tilde{x}\}|a(x).P'|Q)$ and $B \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P'\{M/x\}|Q)$. Similarly, we have that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)\sigma \equiv_{E_{ss}^{TR,h}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(x).P'|Q)\sigma$. By alpha-renaming, we can assume that $x \notin \tilde{x}$ and, by SCOPE, we derive $free(M) \cap \{\tilde{n}, \tilde{y}\} = \varnothing$. By IN, ALIAS, SUBST and RESPAR, we get $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(x).P'|Q)\sigma \xrightarrow{\mu\sigma} \nu\tilde{n}.\nu\tilde{y}.\nu x.(\{\widetilde{M}/\tilde{x}\}|\{M/x\}|P'|Q)\sigma$.

If $\mu = \bar{a}(b)$, the output term is a free channel. We have that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P) \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(x).P'|Q)$ and $B \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P'|Q)$. By SCOPE, $a \notin \tilde{n}$ and $b \notin \tilde{n}$. Similarly, we have that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)\sigma \equiv_{E_{ss}^{TR,h}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\bar{a}(b).P'|Q)\sigma$. The result follows from OUT- ATOM and SCOPE.

If $\mu = \nu b.\bar{a}(b)$, the output term is a private channel. We have that $\nu\tilde{n}.\nu b.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P) \equiv_{E_{SS}} \nu\tilde{n}.\nu b.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(x).P'|Q)$ and $B \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P'|Q)$. Similarly, we have that $\nu\tilde{n}.\nu b.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)\sigma \equiv_{E_{ss}^{TR,h}} \nu\tilde{n}.\nu b.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\bar{a}(b).P'|Q)\sigma$. The result follows from OUT-ATOM and OPEN-ATOM.

if $\mu = \nu x.\bar{a}(x)$, we have that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P) \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(M).P'|Q)$ and $B \equiv_{E_{SS}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|\{M/x\}|P'|Q)$, for some $x \notin \tilde{x}$ and with $fv(M) \subseteq \tilde{x}$. Similarly, we have that $\nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)\sigma \equiv_{E_{ss}^{TR,h}} \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|a(M).P'|Q)\sigma$. The result follows from ALIAS, RES-PAR, OUT-ATOM and OPEN-ATOM.

In all cases, it is easy to see that the resulting extended process is (*TR*, *h*)-valid. The proof for the "if" implication is similar and relies on the fact that σ is injective when applied to terms in (*TR*, *h*)-normal form.

Lemma 3 let ϕ be an (*TR*, *h*)-valid frame and σ be an (*TR*, *h*)-compilation. Then for every *M*, *N* in (*TR*, *h*)-normal form with respect to ϕ such that $(free(M) \cup free(N)) \cap bound(\phi) = \varnothing$, we have that $(M = N)\phi \Leftrightarrow (M\sigma = N\sigma)\phi\sigma$.

Proof. The proof follows from Lemma 1 and Definition 5.

Lemma 4 let ϕ be an (*TR*, *h*)-valid frame. For every *M*, *N* such that $(M = N)\phi$ holds or $(M\sigma = N\sigma)\phi\sigma$ holds, we have that *M*, *N* are in (*TR*, *h*)-normal form with respect to ϕ.

Proof. By an inspection of the equational rules in $E_{ss}^{TR,h}$ and definition 5.

Lemma 5 *let ϕ and ψ be (TR, h)-valid frames. Let σ be (TR, h)-compilation. If $\phi\sigma \approx^{s}_{E_{SS}^{TR,h}} \psi\sigma$, then $\phi \approx^{s}_{E_{SS}} \psi$.*

Proof. By definition 11, we have to prove that $E_{SS} \mapsto (M = N)\phi \Leftrightarrow E_{SS} \mapsto (M = N)\psi$, for every $M, N \in T_{E_{SS}}$, only if $E_{SS}^{TR,h} \mapsto (M' = N')\phi\sigma \Leftrightarrow E_{SS}^{TR,h} \mapsto (M' = N')\psi\sigma$, for every $M', N' \in T_{E_{SS}^{TR,h}}$. Suppose that there exist *M*, *N* such that $(M = N)\phi$ holds and $(M = N)\psi$ does not hold. By Lemma 4, we can assume that *M*, *N* are in (*TR*, *h*)-normal form with respect to ϕ. By Lemma 3, $(M\sigma = N\sigma)\phi\sigma$ holds and $(M\sigma = N\sigma)\psi\sigma$ does not hold. Therefore, we get $(M = N)\psi$ hold.

Similarly, we can prove that $E_{SS}^{TR,h} \mapsto (M' = N')\phi\sigma \Leftrightarrow E_{SS}^{TR,h} \mapsto (M' = N')\psi\sigma$, for every $M', N' \in T_{E_{SS}^{TR,h}}$, only if $E_{SS} \mapsto (M = N)\phi \Leftrightarrow E_{SS} \mapsto (M = N)\psi$, for every $M, N \in T_{E_{SS}}$.

Lemma 6 *let A and B be extended processes such that $A \xrightarrow{a(M)} A'$, $B \xrightarrow{a(M)} B'$, and $\phi(A) \approx^{s}_{E_{SS}} \phi(B)$. Then for every N such that $E_{SS} \mapsto M\phi(A) = N\phi(A)$ and $A \xrightarrow{a(N)} A'$, we have that $B \xrightarrow{a(N)} B'$.*

Proof. Since the frames of the two extended processes are statically equivalent, we have that $E_{SS} \mapsto M\phi(A) = N\phi(A)$ and $dom(\phi(A)) = dom(\phi(B))$. Possibly after applying α-renaming on bound names, we get the result by applying IN, SCOPE, and STRUCT.

We can finally show that verifying labeled bisimilarity on extended processes obtained by the compilation suffice to prove labeled bisimilarity on the original extended processes. With Lemma 7, we can prove Theorem 1 as desired.

Lemma 7 *Let A and B be extended process such that $A = \nu\tilde{n}.\nu\tilde{y}.(\{\widetilde{M}/\tilde{x}\}|P)$, $B = \nu\tilde{n}'.\nu\tilde{y}'.(\{\widetilde{M}'/\tilde{x}'\}|P')$, for some (TR, h)-valid processes P and P'. Let σ be the (TR, h)-compilation. If $A\sigma \approx^{l}_{E_{SS}^{TR,h}} B\sigma$, then $A \approx^{l}_{E_{SS}} B$.*

Proof. Since $A\sigma \approx^{l}_{E_{SS}^{TR,h}} B\sigma$,we can consider the smallest symmetric relation $\mathbb{R}$ satisfying the condition 1, 2, and 3 of Definition 12 and such that $A\sigma\ \mathbb{R} B\sigma$. Given σ, let us define the relation $\mathbb{R}$ as the smallest symmetric relation satisfying the following conditions:

- For every (*TR*, *h*)-valid *A*, *B* such that $A\sigma\ \mathbb{R}' B\sigma$. We have that $A\,\mathbb{R} B$.
- For every *A*, *B*, A', B' such that $A\,\mathbb{R} B$, $A \equiv_{E_{SS}} A'$ and $B \equiv_{E_{SS}} B'$, we have that $A'\ \mathbb{R}\ B'$.

We want to prove that $\mathbb{R}$ satisfies the conditions 1, 2, and 3 of Definition 12.

Condition 1: We want to prove that for every *A*, *B* such that $A\,\mathbb{R} B$, we have that $\phi(A) \approx^{s}_{E_{SS}} \phi(B)$. If $A\,\mathbb{R} B$, then there exist (*TR*, *h*)-valid A', B' such that $A \equiv_{E_{SS}} A'$ and $B \equiv_{E_{SS}} B'$, and $A'\phi\ \mathbb{R}\ B'\phi$. By definition of $\mathbb{R}'$, $\phi(A'\sigma) \approx^{s}_{E_{SS}^{TR,h}} \phi(B'\sigma)$. By Lemma 5, $\phi(A') \approx^{s}_{E_{SS}} \phi(B')$. Since structural equivalence preserves static equivalence, $\phi(A) \approx^{s}_{E_{SS}} \phi(B)$, as desired.

Condition 2:We want to prove that for every *A*, *B* such that $A\,\mathbb{R} B$, we have that if $A \rightarrow A_1$, then $B \rightarrow^{*} B_1$ and $A_1\ \mathbb{R} B_1$ for some B_1. If $A\,\mathbb{R} B$, then there exist (*TR*, *h*)-valid A', B' such that $A \equiv_{E_{SS}} A'$ and $B \equiv_{E_{SS}} B'$, and $A'\sigma\ \mathbb{R}'\ B'\sigma$. By Lemma 2, for every A_1 such that $A \rightarrow A_1$, there exists a (*TR*, *h*)-valid A_1' such that $A' \rightarrow A_1'$, $A_1' \equiv_{E_{SS}} A_1$ and $A'\sigma \rightarrow A_1'\sigma$, we can find similar B_1 and B_1' for *B* and B', respectively. By Definition of $\mathbb{R}'$, it is easy to see that $A_1'\sigma\ \mathbb{R}'\ B_1'\sigma$. By Definition of $\mathbb{R}$, $A_1'\ \mathbb{R}\ B_1'$ and, since σ is closed by structural equivalence, $A_1\,\mathbb{R} B_1$, as desired.

Condition 3: We want to prove that for every *A*, *B* such that $A\,\mathbb{R} B$, we have that if $A \xrightarrow{\mu} A_1$ and $fv(\mu) \subseteq dom(A)$ and $bn(\mu) \cap fn(B) = \varnothing$, then $B \rightarrow^{*} \xrightarrow{\mu} \rightarrow^{*} B_1$ and $A_1\ \mathbb{R}\ B_1$ for some B_1. If $A\,\mathbb{R} B$, then there exist (*TR*, *h*)-valid A', B' such that $A \equiv_{E_{SS}} A'$ and $A \equiv_{E_{SS}} B'$, and $A'\phi\ \mathbb{R}'\ B'\phi$. By Lemma2 and Lemma 6, for every A_1 such that $A \xrightarrow{\mu} A_1$, there exist a (*TR*, *h*)-valid A' such that $A' \xrightarrow{\mu} A_1'$, $A_1' \equiv_{E_{SS}} A_1$ and $A'\sigma \rightarrow A_1'\sigma$, we can find similar B_1 and B_1' for *B* and B', respectively. By Definition of $\mathbb{R}'$, it is easy to see that $A_1'\sigma\ \mathbb{R}'\ B_1'\sigma$. By Definition of $\mathbb{R}$, $A_1'\ \mathbb{R}\ B_1'$ and, since ϕ is closed by structural equivalence, $A_1\,\mathbb{R} B_1$, as desired.

Therefore, $A \approx^{l}_{E_{SS}} B$, as desired.

Theorem 1 then follows directly from Lemma 7 since $\approx^{l}$ and $\approx$ coincide in the applied pi-calculus.