Proceedings of the 8th International Conference on Applied Informatics
Eger, Hungary, January 27–30, 2010. Vol. 1. pp. 135–148.

Demonstration of the Modified CSN-logic

Péter Takács[a], Tamás Mihálydeák[b]

[a]University of Debrecen, Faculty of Health
e-mail: vtp@de-efk.hu

[b]University of Debrecen, Faculty of Informatics
e-mail: mihalydeak@inf.unideb.hu

Abstract

We know a number of tools for examining cryptographic protocols. We present the modified CSN-logic in this article. We analyze the Needham-Schroeder protocol with this logical tool. We emphasize the important moments of the practical analysis: idealization, detectability of active attacks, bounded nature of the logical model.

Keywords: cryptographic protocols, formal verification, modified CSN-logic

MSC: 68Q60, 03B70, 03B42

1. Introduction

Cryptographic protocols are often used in today's communication tools. We meet them when we pay by credit card, when we use mobile phones, etc.. Since we handle our personal-, medical- and financial data in these systems, it is necessary to protect these systems. Several methods can provide an opportunity to examine the protocols (as a theoretical approach to computing, logical analysis, etc.). We study the method of the logical approach in this article. The ultimate goal of the process is to construct trusty, secure, adequate protocols.

The general scheme for analyzing cryptographic protocols with modal logic tools are the following. The first step is the protocol formalization. We describe the protocol steps of the fixed assets of formal logic. Sometimes this is called protocol idealization. The second step: we specify the initial assumptions. For example, set of communication partners and the quality of channels are given here. Thirdly: we specify the goals of the protocol. We use the logical axioms and postulates in the fourth step. We compare the results achieved in the fourth step with the protocol goals stated in the fifth step. The goal is to infer the objectives of the protocol from the formal protocol and from the initial assumptions. We use the above steps to examine Needham-Schroeder protocol in this article, our logic tool is the modified CSN-logic.

2. The modified version of the CSN-logic

The first significant result of the analysis protocol with logic tools was the BAN-logic. [1] BAN-logic was the direct ancestor of the CSN-logic. The first description of the CSN-logic was published in 1997 by T. Coffey and P. Saidha. [3] This system enables analysis of protocols that use public key encryption. T. Newe and T. Coffey extended the logic in 2003. The new system is capable of analysing public- and secret-key protocols. The original sources do not reflect the expected exactitude of the mathematical logic. We present an improved system, which is based on the CSN-logic. We specify the applied logic language, the notation system and the rules of inferences in our work. We modify the axiom system in lesser degree. We keep the original CSN-logic name referring to the authors.

2.1. The language of the modified CSN-logic

Detailed description of the modified CSN-logic system is attached in Appendix. Some important features of the logic are the following.

The CSN-logic is a many-sorted (multi-type) and multi-modal, first-order deduction system. The CSN-logic introduces new operators to describe the cryptographic protocols ("K" is the knowledge operator, "B" is the belief operator). The deduction system is based on a classical first-order deduction system. We extend the original system such as the deduction rules for the new operators. The CSN-system is an "epistemic-doxatic" system - by another classification. Thus, the CSN-logic combines the knowledge and belief operators. We have been studying the CSN system since 2006. We examined a number of protocols (MANA protocol family). [8] [9] [10] [11] Our aims are to refine and develop the system.

The CSN-logic is an ordered six-tuple:

$$L^{(CSN)} = \langle Sort, LC, Var, Con, Term, Form \rangle$$

where $Sort$ is the set of types, LC is the set of logical constants, Var is the infinite set of variables of language, Con is the infinite set of non-logical constants of the language, $Term$ is the set of terms, $Form$ is a set of formulas of language. 20 axioms are in the system. A1-A4 are logical axioms. A5-A20 are non-logical axioms. In addition, M1 - M5 are comments, which help the interpretation of the axioms and to prove theorems.

3. The Needham-Schroeder protocol

We show the application of theory in this section. We consider the Needham-Schroeder symmetric key protocol (NS-protocol, 1978). [6] This protocol aims to establish a session key between two parties on network and it is based on symmetric encryption algorithm. A, B, S entities (S server); n_A, n_B nonces (fresh messages); ks_{AB}, ks_{BS}, ks_{AS} symmetric keys; $\{\}_{ks_{AB}}$ encryption with key ks_{AB}. The protocol steps are the following (in Alice-Bob notation system).

1. $A \to S : A, B, n_A$
2. $S \to A : \{n_A, B, ks_{AB}, \{ks_{AB}, A\}_{ks_{BS}}\}_{ks_{AS}}$
3. $A \to B : \{ks_{AB}, A\}_{ks_{BS}}$
4. $B \to A : \{n_B\}_{ks_{AB}}$
5. $A \to B : \{n_B - 1\}_{ks_{AB}}$

The protocol is vulnerable to replay attack (Denning-Sacco 1981 [4]). If an attacker uses older and compromissed value for ks_{AB}, he can then replay the message step 3 to Bob, who will accept it, being unable to tell that the key is non fresh. The Kerberos protocol fixed this flaw (timestamp, nonces). [2]

3.1. Examination of the N-S protocol

The first step is the protocol formalization.
The idealization of the N-S protocol is the following.
1. $S(ch_1, A, t_1, \{A, B, n_A\})$
2. $R(ch_1, S, t_2, \{A, B, n_A\})$
3. $S(ch_1, S, t_3, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS}))$
4. $R(ch_1, A, t_4, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS}))$
5. $S(ch_2, A, t_5, E(\{ks_{AB}, A\}, ks_{BS}))$
6. $R(ch_2, B, t_6, E(\{ks_{AB}, A\}, ks_{BS}))$
7. $S(ch_2, B, t_7, E(n_B, ks_{AB}))$
8. $R(ch_2, A, t_8, E(n_B, ks_{AB}))$
9. $S(ch_2, A, t_9, E(\{n_B, 1\}, ks_{AB}))$
10. $R(ch_2, B, t_{10}, E(\{n_B, 1\}, ks_{AB}))$

The initial assumptions are the following.
I1. $\forall\Sigma \in ENT\backslash\{A, S\}\ \neg L_{\Sigma,t_0} ks_{AS},\ L_{A,t_0} ks_{AS},\ L_{S,t_0} ks_{AS}$.
Only A and S know key ks_{AS}.
I2. $\forall\Sigma \in ENT\backslash\{B, S\}\ \neg L_{\Sigma,t_0} ks_{BS},\ L_{B,t_0} ks_{BS},\ L_{S,t_0} ks_{BS}$.
Only B and S know key ks_{BS}.
I3. $CH(ch_1, pub)$, $ENT_{ch_1} = \{A, B, S\}$. The channel ch_1 is public, ENT_{ch_1} is the set of entities capable of using the channel ch_1.
I4. $CH(ch_2, pub)$, $ENT_{ch_2} = \{A, B, S\}$. The channel ch_2 is public, ENT_{ch_2} is the set of entities capable of using the channel ch_2.
I5. $ENT_{ch_1} = ENT_{ch_2} = \{A, B, S, E\}$. E is involved in case of passive attacks.

I6. $\forall t_i \, \forall \Psi \in ENT_{ch_j} \;\; K_{E,t_i}(R(ch_j, \Psi, t_i, m)) \quad (i \in \{1, \ldots 10\}, j \in \{1,2\})$.
E receives all messages.

The protocol goals and the proofs are the following.

Theorem 3.1 (G1.). *Entity A knows the secret key ks_{AB} at time t_4.*

$$L_{A,t_4} ks_{AB}$$

Proof: If A receives the message in protocol step 4, the following statements hold true:

$$K_{A,t_4} R(ch_1, A, t_4, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})) \qquad (1)$$

By axiom $A2(a)$, we have:

$$R(ch_1, A, t_4, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})) \qquad (2)$$

By axiom $A6(a)$, we obtain:

$$L_{A,t_4} E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS}) \qquad (3)$$

Finally, by $I1.$, axioms $A3(a)$, $A11(b)$ and $A11(d)$, we obtain:

$$L_{A,t_4} ks_{AB}. \;\square$$

Theorem 3.2 (G2.). *Entity A knows at time t_4: entity S sends message containing the key ks_{AB} at time $t_i < t_4$*

$$K_{A,t_4} S(ch_1, S, t_i, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS}))$$

and entity A knows at time t_4: entity S can use the key ks_{AB} at time t_i

$$K_{A,t_4} L_{S,t_i} ks_{AB}.$$

Proof: As starting point for our proof we use the (1) point of the proof $G1.$ theorem. By axiom $A6(a)$ and inference rule $K1(a)$, we have:
$\exists \Psi \in ENT_{ch_1} \backslash \{A\} \; \exists t_i < t_4$

$$K_{A,t_4} S(ch_1, \Psi, t_i, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})) \qquad (1)$$

Since in our model $ENT_{ch_1} \backslash \{A\} = \{B, S\}$, there are two possibilities.

$$K_{A,t_4} S(ch_1, B, t_i, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})) \qquad (2)$$
$$K_{A,t_4} S(ch_1, S, t_i, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})) \qquad (3)$$

By $I1.$ and $A12(a)$, (2) can be excluded. By axiom $A5(a)$, we obtain:

$$K_{A,t_4} L_{S,t_i} E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS}) \quad t_i < t_4 \qquad (4)$$

By using $I1.$, axioms $A11(b)$ and $A11(d)$, and step (4):

$$K_{A,t_4} L_{S,t_i} ks_{AB} \quad t_i < t_4. \quad q.e.d.$$

Theorem 3.3 (G3.). *Entity B knows the secret key ks_{AB} at time t_6.*
$L_{B,t_6} ks_{AB}$

Proof: The proof is similar to theorem $G1$. By step 6 of the protocol:

$$K_{B,t_6} R(ch_2, B, t_6, E(\{ks_{AB}, A\}, ks_{BS})) \quad (1)$$

We use $A2(a)$, $A6(a)$, $I2.$, $A3(a)$, $A11(b)$ and $A11(d)$ respectively and we obtain the statement. □

Theorem 3.4 (G4.). *Eavesdropper E does not know the secret key ks_{AB} at time t_{10}. The protocol is resistant to passive attack.* $\neg L_{E,t_{10}} ks_{AB}$.

Proof: By initial assumptions $I5.$ and $I6.$ (passive attack), we have:
(1) $K_{E,t_2}(R(ch_1, S, t_2, \{A, B, n_A\}))$
(2) $K_{E,t_4}(R(ch_1, A, t_4, E(\{n_A, B, ks_{AB}, E(\{ks_{AB}, A\}, ks_{BS})\}, ks_{AS})))$
(3) $K_{E,t_6}(R(ch_2, B, t_6, E(\{ks_{AB}, A\}, ks_{BS})))$
(4) $K_{E,t_8}(R(ch_2, A, t_8, E(n_B, ks_{AB})))$
(5) $K_{E,t_{10}}(R(ch_2, B, t_{10}, E(\{n_B, 1\}, ks_{AB})))$
Expressions (2) and (3) contain encrypted message elements. These messages can be decrypted with knowledge of keys ks_{AS} and ks_{BS}. These keys cannot be directly transferred, so entity E does not know these keys. Expressions (4) and (5) describe encryption with key ks_{AB}. E should know the key ks_{AB} from the previous terms. The expression (1) does not contain the key ks_{AB}. Thus, we can admit: E does not know the key even by capturing the messages. □

3.2. Active attack

Entity B receives the message containing the key ks_{AB} from entity A, according to protocol steps. S creates this message element, A essentially only transmits it. On this basis we can formulate statement similar to G2. theorem. Entity B knows at time t_6: entity A sends message containing the key ks_{AB} at time $t_i < t_6$

$$L_{B,t_6} S(ch_2, A, t_i, E(\{ks_{AB}, A\}, ks_{BS}))$$

and entity B knows at time t_6: entity A can use the key ks_{AB} at time t_i

$$K_{B,t_6} L_{A,t_i} ks_{AB}.$$

If we apply the skeleton of the G2 proof we get stuck. As starting point for our proof we use the (1) point of the proof $G3$. theorem. By axiom $A6(a)$ and inference rule $K1(a)$ we obtain:
$\exists \Psi \in ENT_{ch_2} \backslash \{B\} \; \exists t_i < t_6$

$$K_{B,t_6} R(ch_2, B, t_6, E(\{ks_{AB}, A\}, ks_{BS})) \quad (3.1)$$

Since in our model $ENT_{ch_2} \setminus \{B\} = \{A, S\}$, there are two possibilities.

$$K_{B,t_6} S(ch_2, A, t_i, E(\{ks_{AB}, A\}, ks_{BS})) \tag{2}$$

$$K_{B,t_6} S(ch_2, S, t_i, E(\{ks_{AB}, A\}, ks_{BS})) \tag{3}$$

(2) would be excluded by $I2$.. We cannot go further because B has no prior direct information about communications of A and S. We cannot prove the above statement in this way.

The failure of proof of the statements above may indicate weakness in the protocol. This weakness makes the Denning-Sacco-type attack possible. [4] The current formal methods are not suitable for the detection of active attacks. Active intervention means the modification, deletion, replacement or installation of new steps in the protocol. This represents new protocols, which means that a new analysis is needed similar to the foregoing.

4. Summary - plans and tasks

The presentation included a description of the CSN-logic. Some important and stressed experiences are as follows.

- The idealization is a very important step of the examination.
- The active attack always means a new protocol. It always means new examinations.
- The limitations of the logical model always should be considered.

Future goals, some of which relate to the CSN-logic are as follows.

- we must consider new protocols,
- we must develop the language (one-way channels (model bulletin board), channel mix),
- we should examine the axiom-system (reduction, consistency, independence, etc.),
- we should record intended interpretation in detail.

New questions emerged during the analysis and should be studied later. Let us analyze a situation in which two attackers are in the model. Does the passive attacker recognize the active attacker? How can this be modelled?

References

[1] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactins on Computer Systems*, 8(1):18–36, February 1990.

[2] L. Buttyán and I. Vajda. *Kriptográfia és alkalmazásai.* TypoTex, 2004.

[3] T. Coffey and P. Saidha. Logic for verifying public-key cryptographic protocols. *IEEE Proceedings Computers and Digital Techniques*, 144(1):28–32, 1997.

[4] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24.(8.):533–536., August 1981.

[5] M. Kudo and A. Mathuria. An extended logic for analyzing timed-release public-key protocols. In *Proceedings Information and Communication Security, Second International Conference, ICICS'99, Sysdney*, pages 9–11, November 1999.

[6] R. M. Needham and M. D. Schroeder. Using encryption for authentication in large networks of computers. *Communicatins of the ACM*, 21(12):993–999, 1978.

[7] T. Newe and T. Coffey. Formal verification logic for hybrid security protocols. *International Journal of Computer Systems Science & Engineeing*, pages 17–25, 2003.

[8] P. Takács. The extension of CSN-logic for multi-channel protocols. In *Proceedings of the 7th ICAI Conference, Eger*, pages 147–154, 2007. Reviewed by Zentralblatt für Mathematik.

[9] P. Takács and S. Vályi. Többcsatornás kriptográfiai protokollok vizsgálata a bővített CSN-logika eszközeivel. In *I. Nyíregyházi Doktorandusz Konferencia, DE-EK*, December 2007.

[10] P. Takács and S. Vályi. An extension of protocol verification modal logic to multi-channel protocols. *Tatra Mountains Mathematical Publications*, 41:153–166, 2008.

[11] P. Takács and S. Vályi. Javaslat a MANA II kriptográfiai protokoll korrekciójára. In *Informatika a felsőoktatásban 2008*, Augusztus 2008.

Appendix

The language of the CSN-logical system is the following ordered sextet [1]

$$L^{(CSN)} = \langle Sort, LC, Var, Con, Term, Form \rangle$$

where
Sort $= \{U, E, K, T, C\}$ is the set of types: U message type; E entity type; K key type; T time type; C channel type.
LC $= \{\neg, \rightarrow, \leftrightarrow, \wedge, \vee, \equiv, =, \forall, \exists, (,)\}$ is the set of logical constans of the language. We use them as it is often used in the first-order logic.
Var $= Var_U \cup Var_E \cup Var_K \cup Var_T \cup Var_C$ is the infinite set of variables of language. All variables have specified type. Var_δ denotes the set of δ type variables.
Con $= Con_U \cup Con_E \cup Con_K \cup Con_T \cup Con_C$ is the infinite set of non-logical constants of the language. All non-logical constants have defined types. Con_δ denotes the set of δ type non-logical constants. The set can be empty in certain types of case. $F(0)_\delta$ is the set of constant symbols (name-constants), $F(n)_\delta$ is the set of n-ary function symbols. Numbers in arguments indicate the number of parameters. It is usually given in a series of finite index $\langle \delta_1, \delta_2, \ldots, \delta_n, \delta \rangle$ for the function symbols. This specifies the type of the arguments ($\delta_i \in Sort$) and the type of the function symbol ($\delta \in Sort$). $P(0)$ is the set of propositional variables (relations of valence 0), $P(n)$ is the set of predicate symbols with valence n (number of arguments). It is usually given in a series of finite index $\langle \delta_1, \delta_2, \ldots, \delta_n \rangle$ ($\delta_i \in Sort$) for the predicate symbols.
Term $= Term_U \cup Term_E \cup Term_K \cup Term_T \cup Term_C$ is the set of terms of the language. Terms are given by inductive definition. $Term_\delta$ denotes the set of δ type terms. The set can be empty in certain types of cases. The general form of inductive definition for all types is the following:

- (a) $Var_\delta \cup F(0)_\delta \subseteq Term_\delta$.
- (b) If $f \in F(n)_\delta$, $(n = 1, 2, \ldots)$ and $s_1, s_2, \ldots, s_n \in Term$, then $f(s_1, s_2, \ldots, s_n) \in Term_\delta$.

Form is a set of formulas of language. Forms are given by inductive definition:

- (a) $P(0) \subseteq Form$.
- (b) If $s_1, s_2 \in Term_\delta$, then $(s_1 = s_2) \in Form$.
- (c) If $P \in P(n)$, $(n = 1, 2, \ldots)$ and $s_1, s_2, \ldots, s_n \in Term$, then $P(s_1, s_2, \ldots, s_n) \in Form$.
- (d) If $A \in Form$, then $\neg A \in Form$.
- (e) If $A, B \in Form$, then $(A \rightarrow B), (A \wedge B), (A \vee B), (A \equiv B) \in Form$.
- (f) If $x \in Var$, $A \in Form$, then $\forall x A, \exists x A \in Form$.

Additional details and characteristics of each type are as follows:

[1] The original CSN-logic was appearing two articles by T. Coffey, P. Saidha and T. Newe. [3] [7] This system is complemented by M. Kudo and A. Mathuria. [5] The first form of the multichannel scheme was establish by P. Takács and S. Vályi. [10] The Appendix contains a substantial revision of this system. We refer to the above-mentioned sources of intended interpretation of the system.

• i, j are general index variables. They run along the natural numbers.
• x, y, z are general variables. It is always given what types of variables are represented.
• We employ parentheses for clarity in the description in many cases. They should be read and interpreted as usual in mathematics.
• Free variables are implicitly quantified with universal quantifiers in the CSN axioms and inference rules.

U - message type

Characterization: description of messages in communication. MSG is a set of all messages. This set is infinite.
• Var_U: Set of message type variables. This set is infinite.
m, n, r, m_1, m_2, …, m_i, m_j, … are general message variables.
n_A, n_B, …, n_Σ, … are special message variables. They usually denote unique message elements (fresh messages - against replay attacks).
r_A, r_B, …, r_Σ, … are special message variables. They usually denote random numbers. Capital letter in the index denotes the entity who generates the message.
• Con_U:
$F(0)_U$:

(a) Transmitted signals (characters; bit-sequences; 1, 2 bytes with ASCII or Unicode coding) during the protocol communication are message-constants.

(b) Fixed meaning strings (commands, directions, for example: "enc", "dec", "0", "1", …) are message-constans. They are always in double quation-marks. We provide interpretation in all cases.

$F(n)_U$:

$\{m_1, m_2\}$ We can generate new messages by concatenation. Braces denote this construction. $\{\} \in F(2)$; $\langle U, U, U\rangle$.

$E(m, k)$ Encryption function - case of symmetric key cryptography. $E(m, ks_{(\Sigma,\Psi)})$ means: encryption of plaintext message m using the shared secret key of entities Σ and Ψ. $E \in F(2)$; $\langle U, K, U\rangle$.

$D(m, k)$ Decryption function - case of symmetric key cryptography. $D(x, ks_{(\Sigma,\Psi)})$ means: decryption of chiphertext message m using the shared secret key of entities Σ and Ψ. $D \in F(2)$; $\langle U, K, U\rangle$.

$e(m, k)$ Encryption function - case of public key cryptography. $e(m, k_\Sigma)$ means: encryption of plaintext message m using the public key k_Σ of entity Σ. $e(m, k_\Sigma^{-1})$ means: generate digital signature of message m using the secret key k_Σ^{-1} of entity Σ. $e \in F(2)$; $\langle U, K, U\rangle$.

$d(m, k)$ Decryption function - case of public key cryptography. $d(m, k_\Sigma^{-1})$ means: decryption of chiphertext message m using the secret key k_Σ^{-1} of entity Σ. $d(m, k_\Sigma)$ means: check of the digital signature of message m using the public key k_Σ of entity Σ. $d \in F(2)$; $\langle U, K, U\rangle$.

$h(m, k)$ Keyed hash function. $h(m, k)$ denotes the value of the keyed hash function. $h(m, k) \in F(2)$; $\langle U, K, U\rangle$.

$H(m)$ Hash function - MD series, SHA series, HAVAL, RIPEM, etc.. $H(m) \in F(1)$; $\langle U, U \rangle$.

Remarks:

1. $ss_{(\Sigma,\Psi)}$ is a shared secret for entities Σ and Ψ. $SS_{(\Sigma,\Psi)}$ is the set of good sharet secrets for entities Σ and Ψ.
2. We interpret function-pairs ($E2U(\Sigma)$, $U2E(m)$; $K2U(k)$, $U2K(m)$; $T2U(t)$, $U2T(m)$; $C2U(ch), U2C(m)$) in the case of entity-, key-, time- and channel-type variables. They make it possible to embed and take out entities, keys, time-points and channels to/from the messages (as strings). They represent type-conversion functions.

E - entity type

Characterization: description of communication partners. ENT is the set of all possible entities. ENT is a finite set.

• Var_E: Σ, Ψ, Γ, Λ, ... Set of entity type variables. This set is infinite.

• Con_E:

$F(0)_E$:

A, B, C, D, U, ... The name of entities follow traditional roles: communicating partners A, B; passive attacker E; absolutely reliable party T, etc..

$F(n)_E$:

$E2U(\Sigma)$ Type-conversion function: entity to message. $E2U(\Sigma)='\Sigma'$. $E2U \in F(1)$; $\langle E, U \rangle$.

$U2E(m)$ Type-conversion function: message to entity. $U2E('\Sigma')=\Sigma$. $U2E \in F(1)$; $\langle U, E \rangle$.

Remarks:

1. We interpret the sets of entities who can use the channels. In the case of public channel $ENT_{ch_i} = ENT$. In the case of secret channel we list the elements of the set: $ENT_{ch_i} = \{A, B, \ldots\}$. $ENT_{ch_i} \subseteq ENT$.

K - key type

Characterization: description of keys. KEY denotes the set of all possible keys.

• Var_K: Set of key type variables. This set is infinite. k general key-variable.

• Con_K:

$F(n)_K$:

$ks_{(\Sigma,\Psi)}$ Shared secret key - case of symmetric key cryptography. $ks_{(\Sigma,\Psi)}$ is a shared secret key for entity Σ and Ψ. $ks_{(\Sigma,\Psi)} \in F(2)$; $\langle E, E, K \rangle$.

k_Σ Public key - case of public key cryptography. k_Σ is a public key of entity Σ. $k_\Sigma \in F(1)$; $\langle E, K \rangle$.

k_Σ^{-1} Secret key - case of public key cryptography. k_Σ^{-1} is a secret key of entity Σ. $k_\Sigma^{-1} \in F(1)$; $\langle E, K \rangle$.

$k_{t_i}, k_{t_i}^{-1}$ Time-key. $k_{t_i} \in F(1)$; $\langle T, K \rangle$. $k_{t_i}^{-1} \in F(1)$; $\langle T, K \rangle$.

$K2U(k)$ Type-conversion function: key to message. $K2U(k_\Sigma) =' k_\Sigma{}'$. $K2U \in F(1)$; $\langle K, U \rangle$.

$U2K(m)$ Type-conversion function: message to key.
$U2K('k_\Sigma') = k_\Sigma$. $U2K \in F(1)$; $\langle U, K\rangle$.

Remarks:
1. $KS_{(\Sigma,\Psi)}$ denotes the set of good shared keys for entites Σ and Ψ.
2. We use the $ks_{\Sigma\Psi}$ notation for key $ks_{(\Sigma,\Psi)}$.

T - time type

Characterization: description of the time properties of protocols. $TIME$ denotes the set of all possible time in the protocol. This set is finite.

• Var_T: t, t_1, t_2, … t_i, t_j, …, t', t'', … Set of time type variables. This set is infinite.

• Con_T:

$F(0)_T$:
- (a) t_0 is the initial time of the examined protocol.
- (b) t_g is time of key generation.
- (c) τ is a timing point of examined protocol.

$F(n)_T$:

$T2U(t)$ Type-conversion function: time to message.
$T2U(t_i) =' t_i'$. $T2U \in F(1)$; $\langle T, U\rangle$.

$U2T(m)$ Type-conversion function: message to time.
$U2T('t_i') = t_i$. $U2T \in F(1)$; $\langle U, T\rangle$.

• $Form$:
- (a) If $t_1, t_2 \in Term_T$, then $(t_1 < t_2) \in Form$.

Remarks:
1. The $TIME$ set forms a linear ordered set. It is described by the axiom $A20(a)$.
2. $t_i \le t_j$, $t_i > t_j$, $t_i \ge t_j$ formulas are interpreted.

C - channel type

Characterization: description of communication channels. CH denotes the set of all possible channels. This set is finite.

• Var_C: ch, ch_1, ch_2, … ch_i, ch_j are channel variables. This set is infinite.

• Con_C:

$F(n)_T$:

$C2U(t)$ Type-conversion function: channel to message.
$C2U(ch_i) =' ch_i'$. $C2U \in F(1)$; $\langle C, U\rangle$.

$U2C(m)$ Type-conversion function: message to channel.
$U2C('ch_i') = ch_i$. $U2C \in F(1)$; $\langle U, C\rangle$.

Remarks:
1. It is necessary to describe the channel properties of the system. We distinguish two types of channels for the sake of simplicity. Let us denote $CH(ch_i, sec)$ the secure (protected) channel ch_i. Let us denote $CH(ch_i, pub)$ the public channel ch_i. If the type of the channel is given the set of users who are able to use the channel may be given. We use the notation $ENT_{ch_i} = \{\ldots\}$.

Operators and predicates

$K_{\Sigma,t}\Phi$	Knowledge operator of Hintikka. $K_{\Sigma,t}\Phi$ means: entity Σ knows statement Φ at time t.
$B_{\Sigma,t}\Phi$	Belief operator. $B_{\Sigma,t}\Phi$ means: entity Σ believes at time t that statement Φ is true.
$L_{\Sigma,t}x$	Knowledge predicate. $L_{\Sigma,t}x$ means: entity Σ knows and can reproduce object (message or key) x at time t.
$S(ch_i, \Sigma, t, m)$	Emission predicate. $S(ch_i, \Sigma, t, m)$ means: entity Σ sends message m at time t in channel ch_i.
$R(ch_i, \Sigma, t, m)$	Reception predicate. $R(ch_i, \Sigma, t, m)$ means: entity Σ receives message m at time t in channel ch_i.
$C(x, y)$	'Contains' predicate. $C(x, y)$ means: object x contains the object y.
$A(\Sigma, t, \Psi)$	Authentication predicate. $A(\Sigma, t, \Psi)$ means: entity Σ authenticates entity Ψ at time t.
$O_{\Sigma,t}(x, y)$	'Obtain' predicate. $O_{\Sigma,t}(x, y)$ means: entity Σ can obtain object y from object x at time t.

Inference rules

Let us denote α, β formulas; p, q statements of the language.
The inference rules of the CSN-logic are the following:

R1	$\alpha \wedge (\alpha \rightarrow \beta) \Rightarrow \beta$ *(modus ponens).*
R2(a)	$\alpha \Rightarrow K_{\Sigma,t}\alpha$ *(generelisation rule I).*
R2(b)	$\alpha \Rightarrow B_{\Sigma,t}\alpha$ *(generalisation rule II).*
R3	$(\alpha \wedge \beta) \Rightarrow \alpha$ *(simplification).*
R4	$(\alpha)\ ,\ (\beta) \Rightarrow (\alpha \wedge \beta)$ *(conjunction).*
R5	$\alpha \Rightarrow (\alpha \vee \beta)$ *(addition).*
R6	$\neg\neg\alpha \Rightarrow \alpha$ *(double negation).*
K1(a)	$K_{\Sigma,t}(p \wedge q) \Rightarrow K_{\Sigma,t}p \wedge K_{\Sigma,t}q$.
K2(a)	$K_{\Sigma,t}p \wedge K_{\Sigma,t}q \Rightarrow K_{\Sigma,t}(p \wedge q)$.

Axioms

A1(a)	$K_{\Sigma,t}p \wedge K_{\Sigma,t}(p \rightarrow q) \rightarrow K_{\Sigma,t}q$
A1(b)	$B_{\Sigma,t}p \wedge B_{\Sigma,t}(p \rightarrow q) \rightarrow B_{\Sigma,t}q$
A2(a)	$K_{\Sigma,t}p \rightarrow p$
A3(a)	$L_{\Sigma,t}x \rightarrow \forall t_i \geq t\ \ L_{\Sigma,t_i}x$
A3(b)	$K_{\Sigma,t}p \rightarrow \forall t_i \geq t\ \ K_{\Sigma,t_i}p$
A3(c)	$B_{\Sigma,t}p \rightarrow \forall t_i \geq t\ \ B_{\Sigma,t_i}p$
A4(a)	$L_{\Sigma,t}y \wedge C(y, x) \rightarrow \exists \Psi \in ENT\ \ L_{\Psi,t}x$
A4(b)	$C(x, x)$
A4(c)	$C(x, y) \wedge C(y, z) \rightarrow C(x, z)$
A4(d)	$C(e(m, k_\Sigma), m) \wedge C(d(m, k_\Sigma^{-1}), m)$

A5(a) $S(ch_i, \Sigma, t, m)$
$\rightarrow L_{\Sigma,t}m \wedge \exists\Psi \in ENT_{ch_i}\backslash\{\Sigma\}\ \exists t_i > t\ R(ch_i, \Psi, t_i, m)$

A6(a) $R(ch_i, \Sigma, t, m)$
$\rightarrow L_{\Sigma,t}m \wedge \exists\Psi \in ENT_{ch_i}\backslash\{\Sigma\}\ \exists t_i < t\ S(ch_i, \Psi, t_i, m)$

A6(b) $R(ch_i, \Sigma, t, m_1) \wedge C(m_1, m_2) \wedge O_{\Sigma,t}(m_1, m_2) \rightarrow \exists\Psi \in ENT\ \exists t_i < t\ \exists m_3\ (S(ch_i, \Psi, t_i, m_3) \wedge C(m_3, m_2) \wedge L_{\Psi,t_i}m_2 \wedge O_{\Sigma,t}(m_1, m_3) \wedge O_{\Sigma,t}(m_3, m_2))$

A7(a) $L_{\Sigma,t}m \wedge L_{\Sigma,t}k_\Psi \rightarrow L_{\Sigma,t}e(m, k_\Psi)$

A7(b) $L_{\Sigma,t}m \wedge L_{\Sigma,t}k_\Sigma^{-1} \rightarrow L_{\Sigma,t}d(m, k_\Sigma^{-1})$

A8(a) $\neg L_{\Psi,t}k_\Sigma \wedge \forall t_i < t\ \neg L_{\Psi,t_i}(e(m, k_\Sigma)) \wedge \neg(\exists n(R(ch_i\Psi, t_i, n) \wedge C(n, e(m, k_\Sigma)))) \rightarrow \neg L_{\Psi,t}(e(m, k_\Sigma))$

A8(b) $\neg L_{\Psi,t}k_\Sigma^{-1} \wedge \forall t_i < t\ \neg L_{\Psi,t_i}(d(m, k_\Sigma^{-1})) \wedge \neg(\exists n(R(ch_i, \Psi, t_i, n) \wedge C(n, d(m, k_\Sigma^{-1})))) \rightarrow \neg L_{\Psi,t}(d(m, k_\Sigma^{-1}))$

A9(a) $L_{\Sigma,t}k_\Sigma^{-1} \wedge \forall\Psi \in ENT\backslash\{\Sigma\}\ \neg L_{\Psi,t}k_\Sigma^{-1}$

A10(a) $L_{\Sigma,t}(d(m, k_\Sigma^{-1})) \rightarrow L_{\Sigma,t}m$

A11(a) $L_{\Gamma,t}m \wedge L_{\Gamma,t}ks_{(\Sigma,\Psi)} \rightarrow L_{\Gamma,t}(E(m, ks_{(\Sigma,\Psi)}))$

A11(b) $L_{\Gamma,t}m \wedge L_{\Gamma,t}ks_{\{\Sigma,\Psi\}} \rightarrow L_{\Gamma,t}(D(m, ks_{\{\Sigma,\Psi\}}))$

A11(c) $L_{\Sigma,t}m \wedge O_{\Sigma,t}(m, n) \rightarrow L_{\Sigma,t}n$

A11(d) $L_{\Sigma,t}m \wedge L_{\Sigma,t}n \rightarrow L_{\Sigma,t}\{m, n\}$

A11(e) $L_{\Sigma,t}\{m, n\} \rightarrow L_{\Sigma,t}m \wedge L_{\Sigma,t}n$

A12(a) $(\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \le t\ \neg L_{\Gamma,t_i}(E(m, ks_{(\Sigma,\Psi)}))\wedge$
$\neg(\exists n(R(ch_i, \Gamma, t_i, n) \wedge C(n, E(m, ks_{(\Sigma,\Psi)}))))$
$\rightarrow \neg L_{\Gamma,t}(E(m, ks_{(\Sigma,\Psi)})))$

A12(b) $(\neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \forall t_i \le t\ \neg L_{\Gamma,t_i}(D(m, ks_{(\Sigma,\Psi)}))\wedge$
$\neg(\exists n(R(ch_i, \Gamma, t_i, n) \wedge C(n, D(m, ks_{(\Sigma,\Psi)}))))$
$\rightarrow \neg L_{\Gamma,t}(D(m, ks_{(\Sigma,\Psi)})))$

A13(a) $\forall\Gamma \in ENT\backslash\{\Sigma, \Psi\}\ \neg L_{\Gamma,t}ks_{(\Sigma,\Psi)} \wedge \exists\Lambda \in \{\Sigma, \Psi\}\ L_{\Lambda,t}ks_{(\Sigma,\Psi)} \rightarrow ks_{(\Sigma,\Psi)} \in \{KS_{(\Sigma,\Psi)}\}$

A14(a) $\forall\Gamma \in ENT\backslash\{\Sigma, \Psi\}\ \neg L_{\Gamma,t}ss_{(\Sigma,\Psi)} \wedge \exists\Lambda \in \{\Sigma, \Psi\}L_{\Lambda,t}ss(\Sigma, \Psi) \rightarrow ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\}$

A15(a) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t}ss_{(\Sigma,\Psi)} \wedge ss_{(\Sigma,\Psi)} \in \{SS_{\{\Sigma,\Psi\}}\} \wedge R(\Sigma, t, m)) \wedge C(m, ss_{(\Sigma,\Psi)}) \wedge \forall t_i \le t\ \neg S(\Sigma, t_i, m)] \rightarrow K_{\Sigma,t}(S(\Psi, t_i, m))$

A15(b) $[A(\Sigma, t, \Psi) \rightarrow (L_{\Sigma,t}k_\Psi \wedge L_{\Sigma,t}m \wedge R(\Sigma, t, n) \wedge C(n, e(m, k_\Psi^{-1}))\ ] \rightarrow \forall t_i \le t\ K_{\Sigma,t}(S(\Psi, t_i, n))$

A16(a) $L_{\Sigma,t}m \wedge L_{\Sigma,t}k \rightarrow L_{\Sigma,t}h(m, k)$.

A17(a) $h(n_A, k_A) = h(n_B, k_B) \leftrightarrow n_A = n_B \wedge k_A = k_B$.

A18(a) $L_{\Sigma,t}m \rightarrow L_{\Sigma,t}h(m)$.

A19(a) $h(n_A) = h(n_B) \leftrightarrow n_A = n_B$.

A20(a) $\forall t \in TIME\ (t \le t) \wedge \forall t, s \in TIME\ (t \le s \wedge s \le t \rightarrow t = s) \wedge \forall t, s, r \in TIME\ (s \le t \wedge t \le r \rightarrow s \le r)$

Remarks

(M1) The type-conversion functions enable us to embed and take out entities, keys, time-points and channels to/from the messages (as strings).

(M2) The axioms do not contain any direct reference to the digital signature. We assume that individuals are able to prepare and verify the digital signature. Based on axiom $A7(a)$:
$L_{\Sigma,t}m \wedge L_{\Sigma,t}k_{\Sigma}^{-1} \rightarrow L_{\Sigma,t}\, e(m, k_{\Sigma}^{-1})$,
$L_{\Sigma,t}\, e(m, k_{\Psi}^{-1}) \wedge L_{\Sigma,t}k_{\Psi} \rightarrow L_{\Sigma,t}\, d(e(m, k_{\Psi}^{-1}), k_{\Psi}) = L_{\Sigma,t}\, m$.

(M3) The axioms $A11(c)$ and $A11(d)$ contain the possibility of connecting and decomposing the message elements.

(M4) In practice, we are simplifying the notation. We leave the marking of the type-conversion in all cases where it is not ambiguous. For example, we use $\{A, k_{\Sigma}, t\}$ instead of $\{E2U(A), K2U(k_{\Sigma}), T2U(t)\}$.

(M5) In some analyses we assume that the messages sent and received are not the same in the case of public channels. It means the application of the Dolev-Yao attacker model: the communication network is totally attackable. The attacker can intercept, change the messages and generate new messages.
If $CH(ch_i, pub)$ and $S(ch_i, A, t_1, n_A)$, then $R(ch_i, B, t_2, n_B)$.
If $CH(ch_i, sec)$ (channel protected) and $S(ch_i, A, t_1, n_A)$, then $R(ch_i, B, t_2, n_A)$.

Péter Takács
University of Debrecen, Faculty of Health, Nyíregyháza, Sóstói Srt. 2-4.

Tamás Mihálydeák
University of Debrecen, Faculty of Informatics, Debrecen, Egyetem Sqr. 1.