Article

Efficient Verification of Cryptographic Protocols with Dynamic Epistemic Logic

Xiaojuan Chen ¹ and Huiwen Deng ^{2,*} ¹ Business College, Southwest University, Chongqing 402460, China; chenxiaojuan@swu.edu.cn² School of Computer and Information Science, Southwest University, Chongqing 400715, China

* Correspondence: huiwend@swu.edu.cn; Tel.: +86-13983492827

Received: 20 April 2020; Accepted: 22 May 2020; Published: 21 September 2020

Abstract: The security of cryptographic protocols has always been an important issue. Although there are various verification schemes of protocols in the literature, efficiently and accurately verifying cryptographic protocols is still a challenging research task. In this work, we develop a formal method based on dynamic epistemic logic to analyze and describe cryptographic protocols. In particular, we adopt the action model to depict the execution process of the protocol. To verify the security, the intruder's actions are analyzed. We model exactly the protocol applying our formal language and give the verification models according to the security requirements of this cryptographic protocol. With analysis and proof on a selected example, we show the usefulness of our method. The result indicates that the selected protocol meets the security requirements.

Keywords: cryptographic protocol; security verification; action model; dynamic epistemic logic

1. Introduction

The security of cryptographic protocols is crucial in the design of the protocols as cryptographic protocols need to consider attacks from attackers. If there is any security flaw in a cryptographic protocol, the protocol can be attacked by attackers or intruders. Therefore, the security verification of cryptographic protocols is particularly important for checking whether the protocol meets security goals. To verify the security of cryptographic protocols, there are many methods. In these methods, logic-based methods play an important role. There are many works of literature (e.g., [1–5]) on the application of logical methods to analyze and verify cryptographic protocols. The earliest analysis method of logic is based on the Dolev-Yao model [6]. It is well known that the Dolev-Yao model opens the way for the study of formalization. Since then, a variety of cryptographic protocol analysis methods based on logic have come out. Most of the protocols analyzed by epistemic logic are authentication protocols in the existing literature. They use the method of model checking to analyze. For example, in the paper [7], the researchers propose two virtual agents to complete encryption and decryption, and construct a model checker to check the protocol. In other papers, KD45 system's belief logic is used. It needs to describe the belief logic of the protocol, and then establish reasoning rules to infer the security of the protocol.

In particular, the cryptographic protocol we mentioned here refers to the protocol using classical cryptography. With the research of quantum computer, quantum cryptography is a research trend. The latest development of quantum cryptography was given in the paper [8]. This article introduced a lot about quantum key distribution (QKD). Similarly, Davide Bacco and Beatrice Da Lio et al. were also doing research on quantum key distribution [9]. They illustrated how to use QKD technology to break through the limitation of low key rate and produce high key rate on a 37-core fiber. At present, the wide application of quantum technology has great challenges. Security verification of quantum cryptography protocol may be more challenging. It will be our next research content. Now, we construct an accurate

dynamic epistemic logic language to describe the protocol with classical cryptography and give the action model according to the action characteristics. The action model is used to accurately describe the protocol action, and the execution process of the protocol action is the operation process of the protocol. The execution process of action is also expressed clearly and intuitively, so the whole operation process of the cryptographic protocol is completely formalized. Our verification does not need complex rules and reasoning, only through the action description table can get the results.

We summarize our work as follows:

- Applying dynamic epistemic logic, a formal language is developed to describe the cryptographic protocol. The syntax and semantics of this language are given. Cryptographic primitives and special propositions are built.
- The action model is constructed to display action elements. Through update execution, the model is updated. The execution of actions results in changes between models. We model the protocol accurately. In a sense, the logic description is the logic analysis for cryptographic protocols.
- With a logical language and the action model, a specific example is verified. The protocol is that a secret message is transmitted between two users who only have their own key. The goal of the protocol is that only the two users know the secret message and the outside intruder can not know the secret message after the execution of the protocol. We give the goal models. Through analysis and proof, we show that the protocol meets the intended requirements and the protocol is secure.

The rest of the paper is arranged as follows. The second section displays related work about verification and analysis of the cryptographic protocol. The third section demonstrates a logic language we develop and the related basic knowledge. In Section 4, we present a specific cryptographic protocol and formalize it. Section 5 presents the verification process of the cryptographic protocol. In Section 6, we give our conclusion and expectation.

2. Related Work

Since the Dolev-Yao model created a way of logical analysis for the cryptographic protocol, all kinds of logical analysis methods for cryptographic protocol have been introduced. The BAN (Burrows, Abadi and Needham) logic [1] is a pioneer. BAN logic verifies the security of cryptographic protocol by the formal method. This initiative has greatly stimulated the interest of researchers in the formal analysis of cryptographic protocols, and has become a milestone in the analysis of cryptographic protocols. The formalization of BAN logic is built on multiple types of model logic. In the paper [1], they gave a set of reasoning rules firstly based on the BAN logic. They analyzed the security of protocols by beliefs deduction. In the follow up research, many drawbacks in the BAN logic were found. So, the BAN logic is extended to GNY logic [10], VO logic [11] and MB logic [12]. BAN logic is constantly improved in application. Today, BAN logic is still used as a verification tool for security protocols or authentication protocols in radio frequency identification systems [13], Internet of things systems [14,15], and remote authentication protocols [16].

AT logic [17] was proposed by Abadi and Tuttle. The problem of BAN logic is improved in the AT logic in terms of syntax, semantics and rules. The paper [18] and the book [19] employed SVO logic to analyze cryptographic protocols. The SVO logic has more strict definitions about the protocol logic. The protocol model based on the SVO logic has a wider range of applications.

There still are many logics for analyzing cryptographic protocols, such as non-monotonic logic [20], time-dependent cryptographic protocol logic [21], protocol composition logic [22] and so on. In all logic analysis, there are four analysis types to find attacks in cryptographic protocols.

Finding a new attack by inductive proof, the authors of [2] analyzed three protocols: a recursive protocol, Needham-Schroeder protocol, and Otway-Rees. Lowe's attack and a new attack were discovered. Another paper [23] also analyzed two variants of Yahalom.

The authors of [4] used rewriting calculus to display the Needham–Schroeder Public-Key protocol. With rewrite rules, they gave all actions in the protocol for all agents, including attackers. By these strategies and a set of rewrite rules, they could find attacks in the protocol.

There have been many works of literature (e.g., [24–28]) that employ process calculus. Typed process calculus displays security information (e.g., [29–33]) by the idea of standard static-analysis. On operating cryptographic data, static-analysis has some inadequacies. So, the work in [34] improved the solution of these operations. They described the protocols and gave proof for the security of the protocols based on the typed process calculus.

Model checking is still a widely used method to verify protocol security. The authors of [3] proposed a method that uses abstract data and deductive rules to prove the results of model checkers. Hans van Ditmarsch et al. [7] developed a dynamic epistemic model checker. In the follow-up work (e.g., [7,35,36]), many researchers verified and analyzed security protocols with dynamic epistemic logic. The authors of [37] proposed a variant of epistemic logic methods to verify authentication protocols.

At present, the development and application of logic go far beyond the field of philosophy [38–40]. Dynamic epistemic logic has been widely used in philosophy, sociology and computer science [41–47], and also in the verification and analysis of security protocols (see [7,36,37,48]).

Although many scholars have done a lot of work in this field, none of them is once and for all. Different protocols need different methods to analyze or verify. In this paper, dynamic epistemic logic is used to describe and analyze a specific cryptographic protocol, so as to effectively verify the security of the protocol.

3. A Dynamic Epistemic Language for Cryptographic Protocols

We develop a language based on the dynamic epistemic logic to describe cryptographic protocols. We extend the epistemic logic with the action model (that refers to the paper [49]). The structure of propositions is refined by three predicates ‘has’, ‘const’ and ‘tg’ (‘has’ denotes an agent *has* a certain message, ‘const’ denotes a certain message can be derived from the agent’s information set and ‘tg’ denotes to *tag* an action in the cryptographic protocol that has been performed).

We define this language as $\mathcal{L}_{cryp}^{A,B}$, where *cryp* means cryptography, *A* is a set of agents, and the set $B \subseteq A$ of the agents at epistemic states who are participants in the protocol we want to analyze. We use $\mathcal{A}$ to denote a set of actions.

3.1. Syntax of the Language $\mathcal{L}_{cryp}^{A,B}$

Before giving the syntax of the language, we define some basic concepts about this language.

Definition 1 (Message). *M* is taken as the set of messages.

Message *m* in the protocol as logic term is defined by:

$$m ::= n \mid k \mid \{m\}_k \mid (m, m')$$

where *n* is a nonce or encoding of a message in ASCII; generally, *n* denotes a plain message. Also, *k* is crypto-key. $\{m\}_k$ is the encryption of *m* with key *k* (ciphertext message) and (m, m') intuitively, represents concatenating of *m* and *m'*.

Definition 2 (Propositions). In this system, the set Φ_{cryp}^A as the set of basic propositions *p* that is defined by:

$$p ::= has_a m \mid const_a m \mid tg(\sigma)$$

where $a \in A$, $m \in M$ and $\sigma \in \mathcal{A}$. $has_a m$ means agent *a* possesses message *m* (the message is received by agent *a* or initial distribution). $const_a m$ means agent *a* can construct *m* from the messages he possesses. $tg(\sigma)$ denotes action σ is tagged. Let $\Phi_I, \Phi_T, \Phi_{tg}$ be subsets of Φ_{cryp}^A that contain the above three propositions, respectively. Then, we will define action models.

We will give a definition of about action models according to the idea of the paper [49,50]. The condition of an action execution is called *precondition*. Without corresponding preconditions, the action can not be executed. In this system, all preconditions belong to Φ_{cryp}^A , which is a part of the language $\mathcal{L}_{cryp}^{A,B}$.

Facts that will hold after the execution of an action are called *postcondition*. Postcondition displays the changing of basic facts after running the action. Postcondition is divided into two parts: the one is Pos_I that shows changes for the information sets including messages that are sent by the agents, another is Pos_{AL} , which illustrates the changes for action labels that show the actions have been performed by the agents.

Definition 3 (Action model). Based on the logical language $\mathcal{L}_{cryp}^{A,B}$, an action model $\mathfrak{A}$ is a structure: $\mathfrak{A} = (\mathcal{A}, \sim_{a \in B}, Pre, Pos_I, Pos_{AL})$, where $\mathcal{A}$ is non-empty and finite set of actions, $\sim_a$ is an equivalence relation (reflexive, symmetric and transitive) on $\mathcal{A}$ for all $a \in B$. $Pre : \mathcal{A} \rightarrow \Phi_{cryp}^A$ assigns a propositional precondition for every action. We employ the idea of paper [49] to deal with the postcondition. Φ is a set of propositions for the language $\mathcal{L}_{cryp}^{A,B}$. For the sake of simplicity of description, we will use a substitution in this model structure. Function $\Phi \rightarrow \mathcal{L}$ is a substitution for $\mathcal{L}$. It maps basic propositions of the language to their variant. The $SUB(\mathcal{L})$ is taken as a set of substitutions from Φ to $\mathcal{L}$. So, $Pos_I : \mathcal{A} \rightarrow SUB(\mathcal{L}_{cryp}^{A,B})$ gives a substitution for $\mathcal{L}$ that has property $Pos_I(\sigma)(p) \in \Phi_I \cup \{\top, \perp\}$ for all $p \in \Phi_I$ and $\sigma \in \mathcal{A}$. $Pos_{AL} : \mathcal{A} \rightarrow SUB(\mathcal{L}_{cryp}^{A,B})$ gives a substitution for $\mathcal{L}$, which has the property that $Pos_{AL}(\sigma)(p) \in \Phi_{AL} \cup \{\top, \perp\}$ for all $p \in \Phi_{AL}$ and $\sigma \in \mathcal{A}$.

We replace ‘has_am’ with ‘m ∈ I_a’ for the substitution mapping to $\top$. Meanwhile, the message m is added to the information set of agent a after the execution of the action.

We write ‘ σ^+ ’ to tag current action for the substitution mapping $tg(\sigma)$ to $\top$, and similarly, we use ‘ σ^- ’ to tag premise action for the substitution mapping $tg(\sigma)$ to $\perp$. This will label (unlabel) σ after the execution of the action. We will use these substitutions to formalize all postconditions in the protocol.

Definition 4 (Syntax of the language $\mathcal{L}_{cryp}^{A,B}$). Given are agents A and atoms Φ_{cryp}^A . Formulas of the language $\mathcal{L}_{cryp}^{A,B}$ are defined by:

$$\varphi ::= \top \mid p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid [\alpha]\varphi$$

$$\alpha ::= (\mathfrak{A}, \sigma) \text{ where } p \in \Phi_{cryp}^A, a \in A, \sigma (\sigma \in \mathcal{A}) \text{ is an action in the action model } \mathfrak{A}.$$

The meaning of the formulas is the same as that of ordinary dynamic epistemic logic. $K_a\varphi$ means agent a knows φ . $[\mathfrak{A}, \sigma]\varphi$ means after the execution of action σ in $\mathfrak{A}$, φ holds.

3.2. Semantics of the Language $\mathcal{L}_{cryp}^{A,B}$

Definition 5 (Models of the language $\mathcal{L}_{cryp}^{A,B}$). The model is a tuple:

$$\mathfrak{M} = (W, \{R_a\}_{a \in B}, I, AL)$$

where W is a non-empty set of a possible world, R_a is binary equivalence relations on W for all $a \in B$, I means information sets, $I_{w,a}$ means that agent a possesses the set of information at the world w , that is $I : W \times A \rightarrow \mathcal{P}(M)$. AL denotes the action label that is $AL : W \rightarrow \mathcal{P}(\mathcal{A})$. For $\sigma \in \mathcal{A}$, $tg(\sigma)$ means action σ in the set $\mathcal{A}$ has been tagged.

I and AL correspond to the valuation of basic propositions. If $m \in I_{w,a}$, has_am is satisfied at the world w . The m is the message that agent a possesses by receiving or by some initial distribution. Message m is also constructed by the agent from messages in his information set. That is $const_a m$. In the following, we cite the idea of the paper [51] and give the construction rules.

Definition 6 (Construction rules in the cryptography). There are many rules determining which messages an agent can construct from his information set:

$$\frac{m \quad k}{m_k} \quad \frac{m_k \quad k}{m} \quad \frac{m \quad m'}{(m, m')} \quad \frac{(m, m')}{m} \quad \frac{(m, m')}{m'}$$

where m_k denotes a message m encrypted with key k . (m, m) denotes concatenation of two messages. In these expressions, above the line is the condition, below the line is the result. The first one says, if an agent has message m and a key k in his information set, he should be able to construct m_k . The others are similar and will not be described.

The satisfaction relations of basic propositions are defined as follows.

Definition 7 (Semantics of basic propositions). Let $\mathfrak{M}$ be a model for $\mathcal{L}_{cryp}^{A,B}$, $w \in W$.

$$\begin{aligned}\mathfrak{M}, w &\models has_a m && \text{iff } m \in I_{w,a} \\ \mathfrak{M}, w &\models const_a m && \text{iff } m \in \overline{I_{w,a}} \\ \mathfrak{M}, w &\models tg(\sigma) && \text{iff } \sigma \in AL(w)\end{aligned}$$

The first one says if the model $\mathfrak{M}$ satisfies agent a possesses message m in the world w , if and only if (iff), m is in information set of agent a in the world w . The second says agent a constructs m at $(\mathfrak{M}, w)$, if and only if, m can be derived from the information set of agent a at the world w . $\overline{I_{w,a}}$ denotes that can be derived from $I_{w,a}$. The last one illustrates that $tg(\sigma)$ holds at $(\mathfrak{M}, w)$, if and only if, σ is tagged at the world w .

Having different basic propositions at the different world displays the transformation between models. This updating execution is defined as follows.

Definition 8 (Updated model). Let $\mathfrak{M}$ be a model for $\mathcal{L}_{cryp}^{A,B}$ and $\mathfrak{A}$ be an action model, the updated model $\mathfrak{M} \circ \mathfrak{A} = (W', \{R'_a\}_{a \in B}, I', AL')$ is defined by:

$$\begin{aligned}W' &= \{\langle w, \sigma \rangle \mid \mathfrak{M}, w \models Pre(\sigma)\} \\ R'_a &= \{(\langle w, \sigma \rangle, \langle v, \sigma' \rangle) \mid wR_a v \text{ and } \sigma \sim_a \sigma'\} \\ I'(\langle w, \sigma \rangle, a) &= \{m \mid \mathfrak{M}, w \models Pos_I(\sigma)(has_a m)\} \\ AL'(\langle w, \sigma \rangle) &= \{\beta \mid \mathfrak{M}, w \models Pos_{AL}(\sigma)(tg(\beta))\}\end{aligned}$$

Definition 9 (Semantics of the language $\mathcal{L}_{cryp}^{A,B}$). Given are epistemic state $(\mathfrak{M}, w)$ with $\mathfrak{M} = (W, \{R_a\}_{a \in B}, I, AL)$, and action model $\mathfrak{A} = (\mathcal{A}, \sim_{aa \in B}, Pre, Pos_I, Pos_{AL})$. Semantics for formulas of the language $\mathcal{L}_{cryp}^{A,B}$ are defined by:

$$\begin{aligned}\mathfrak{M}, w &\models \top && \text{iff true} \\ \mathfrak{M}, w &\models \neg \varphi && \text{iff } \mathfrak{M}, w \not\models \varphi \\ \mathfrak{M}, w &\models \varphi \wedge \psi && \text{iff } \mathfrak{M}, w \models \varphi \text{ and } \mathfrak{M}, w \models \psi \\ \mathfrak{M}, w &\models K_a \varphi && \text{iff for all } v, \text{ if } wR_a v \text{ then } \mathfrak{M}, v \models \varphi \\ \mathfrak{M}, w &\models [\mathfrak{A}, \sigma] \varphi && \text{iff if } \mathfrak{M}, w \models Pre(\sigma) \text{ then } \mathfrak{M} \circ \mathfrak{A}, \langle w, \sigma \rangle \models \varphi\end{aligned}$$

4. Formalizing Cryptographic Protocol

4.1. The Verification Problem of the Cryptographic Protocol

Protocol description: We use *Crypto* to denote the cryptographic protocol. Actually, the execution process of a protocol is a sequence of actions. In cryptography, a protocol is described as the form $A \rightarrow B : m$ (agent ' A sends message m to B '), where $m \in M$ is a message that has a certain pattern. These patterns have been defined as definition 1 in the specific system.

Instantiations: Action pattern is an important part in the execution of the cryptography. We usually instantiate action patterns by *instantiation* θ . *Instantiation* θ is a map from the parameters of the protocol to their respective domain. An execution of a protocol can have several instantiations interleaving. In this system, we employ Θ to denote a set of all instantiations. In this protocol, the number of parameters is finite, and Θ is too.

Network environment: We assume that the network on which the protocol can be run consists of an input-buffer and an output-buffer just as that defined in the work [52]. We take the two parts

as two special agents *In* and *Out*. In the protocol, the trusted agents can only send a message to *In*. The attacker or intruder could eavesdrop messages from *In*. The trusted agents receive messages from *Out*.

The verification model: The verification model is the model for this protocol to archive the goal pattern. After running of the protocol *Crypto*, a requirement ϕ ($\phi \in \mathcal{L}_{crypt}^{A,B}$) should hold. So, we formalize the verification model as below:

$$\mathfrak{M} \models \bigwedge_{\theta \in \Theta} [\text{Crypto}_{Intr}, \sigma(\theta)] \phi(\theta)$$

where

- $\mathfrak{M}$ is a model that we have defined previously, including all assumptions at the initial state, and all epistemic information for all the agents.
- Crypto_{Intr} denotes the action model. *Intr* denotes intruder's actions. This action model consists of all the actions in the protocol *Crypto* and all possible intruder's actions.
- σ is an action in the action model *Crypto* (generally the final one).
- ϕ is an $\mathcal{L}_{crypt}^{A,B}$ -formula. The formulas are either static epistemic formulas or propositions.

$\bigwedge_{\theta \in \Theta} [\text{Crypto}_{Intr}, \sigma(\theta)] \phi(\theta)$ means after an execution of all possible actions in the model *Crypto*, ϕ holds. This is the model of the protocol goals.

4.2. Modeling Action Models

The changes of the language model depend on the action execution in the protocol. That is to say, the action execution results in a change of model. So, we firstly give the action model. We have given the initial model $\mathfrak{M}$ and the action model Crypto_{Intr} . Assuming set *Ag* be the set of agents in this protocol, we will take *T* ($T \in \text{Ag}$) as the intruder's name that performs the intruder's actions. So, in this system, $A = \text{Ag} \cup \{In, Out\}$, and $B = \text{Ag}$.

We give the action model Crypto_{Intr} . Intruder's actions and actions of the protocol requirement are conducted according to the intruder model and the protocol specification from the set of the actions $\mathcal{A}$ in Crypto_{Intr} .

In the protocol, $a \rightarrow b : m$ means that agent *a* sends message *m* to agent *b*. We split the action into two parts: 'send' (σ) $a \rightarrow In : m$ and 'receive' (δ) $Out \rightarrow b : m$. The preconditions of actions for execution are that agent *a* can construct *m* from his information set: $Pre(\sigma) = const_a m$. Postcondition of the action σ is $Pos_I(\sigma) = m \in I_{In}$. We use σ^+ or σ^- to mark or unmark actions. Similarly, the precondition of action δ is: $Pre(\delta) = has_{Out} m$. We stipulate that the buffer can not construct any new message. Postcondition of δ is: $Pos_I = m \in I_b$ and a possible action marking. A specific action in $\mathcal{A}$ is transformed into a parameterized set of the action model. We will model the intruder's actions.

Intruder's actions: In the network model, we assume the intruder can eavesdrop all information from the buffer. So, the form of intruder's actions is either $In \rightarrow T : m$ or $T \rightarrow Out : m$. That is to say, on the one hand, the intruder can take any message from the in-buffer; on the other hand, the intruder can construct any message and put it into the out-buffer. These actions are described as parameterized actions in the action model as Table 1.

Table 1. Action model of the intruder.

Action	Direction	Message	Pre	Pos _I	Pos _{AL}
take <i>m</i>	$In \rightarrow T$	<i>m</i>	$has_{In} m$	$m \in I_T$	-
fake <i>m</i>	$T \rightarrow Out$	<i>m</i>	$const_T m$	$m \in I_{Out}$	-

Epistemic relations between actions: In the system, an agent can only distinguish the actions that he has performed. The other actions are indistinguishable for the agent. We use $\sim_a$ to denote

indistinguishable epistemic relation between actions for all agent a . We will construct $\sim_T$ to display indistinguishable actions for the intruder.

Initial Model: In the protocol, at the initial state, details in the model $\mathfrak{M} = (W, \{R_a\}_{a \in B}, I, AL)$ can be described as below.

- W : a non-empty set of worlds.
- R_a : Let wR_av if and only if $I_{(w,a)} = I_{(v,a)}$ (R_a is equivalence relations for a or a can not distinguish w from v).
- I : I is the set of information for the agents. We say an agent's information set is always based on a certain world. So, any agent a (except the buffer) and its information set I define a world w . Generally, we write it as $I_{w,a}$. In the initial state, we stipulate the buffer is empty. Therefore, $I_{(w,In)} = I_{(w,Out)} = \emptyset$.
- AL : In the initial model, we let $AL(w) = \emptyset$, for all $w \in W$.

In the following, we will analyze a specific cryptographic protocol with the formalization method on the above-mentioned.

4.3. Analysis of a Specific Cryptographic Protocol

In an insecure network, Alice has a secret message m and wants to send it to Bob. They only have their own key. We call Alice as Sender (S) and Bob as Receiver (R). The Sender has her own key k_s . The Receiver has his own key k_r . They only can decrypt data that is encrypted by himself. This protocol is as below.

1. $S \rightarrow R : m_s$
2. $R \rightarrow S : m_{sr}$
3. $S \rightarrow R : m_r$

At the first step, Sender encrypts m with k_s and gets $\{m\}_{k_s}$. For simplicity, we write $\{m\}_{k_s}$ as m_s . Then, the Sender sends m_s to Receiver. After it is received, the Receiver encrypts m_s with k_r and gets m_{sr} . m_{sr} is sent back to Sender. Sender decrypts m_{sr} and gets m_r . At the last step, Sender sends m_r to Receiver. Receiver can decrypt m_r and gets m . Finally, m is transmitted from S to R . We apply the encryption that is commutative: $m_{sr} = m_{rs}$. Here, the key can be a symmetric key or asymmetric key. We formalize this protocol. In this protocol, $M = \{m, m_s, m_{sr}, m_r\}$, $Ag = \{S, R, T\}$, where, S is the Sender, R is the Receiver, T is an intruder. At first, we give the initial model figure and then formalize the execution of the protocol according to the actions of the protocol itself.

In the protocol, we have $M = \{m, m_s, m_{sr}, m_r\}$. However, in the initial state, only S has m , others do not have m . We give a figure to display which agent has secret message m . According to $I_{w,S}, I_{w,R}, I_{w,T}$, we draw the picture. Each world w is expressed by the agent name that has m . The initial model is shown in Figure 1.

Figure 1. Initial model of the protocol.

In Figure 1, the solid lines present the worlds at both ends are distinguishable for S , and the dotted lines are for R . The dashed lines are for T . Accordingly, the dotted lines and the dashed lines are indistinguishably a relation for S . The solid lines and the dashed lines are indistinguishably a relation for R . The solid lines and the dotted lines are indistinguishability relation for T .

So, we have $\mathfrak{M}, S \models has_S m \wedge \neg has_R m \wedge \neg has_T m$ in the initial state.

Table 2 describes actions that are stipulated in the protocol. Now, we add actions from intruder(T) in Table 3. We assume the intruder is a weak intruder. He can not do anything but eavesdropping. He only can get all messages from In-buffer and forward it to Out-buffer. He can not construct any other message because he does not have any key.

Table 2. Action model in the protocol.

Action	Direction	Message	Pre	Pos _I	Pos _{AL}
σ_1	$S \rightarrow In$	m_s	$const_S m_s$	$m_s \in I_{In}$	σ_1^+
δ_1	$Out \rightarrow R$	m_s	$has_{Out} m_s$	$m_s \in I_R$	δ_1^+
σ_2	$R \rightarrow In$	m_{sr}	$const_R m_{sr}$	$m_{sr} \in I_{In}$	δ_1^-, σ_2^+
δ_2	$Out \rightarrow S$	m_{sr}	$has_{Out} m_{sr}$	$m_{sr} \in I_S$	δ_2^+
σ_3	$S \rightarrow In$	m_r	$const_S m_r$	$m_r \in I_{In}$	δ_2^-, σ_3^+
δ_3	$Out \rightarrow R$	m_r	$has_{Out} m_r$	$m_r \in I_R$	δ_3^+

Table 3. Action model including intruder's actions.

Action	Direction	Message	Pre	Pos _I	Pos _{AL}
σ_1	$S \rightarrow In$	m_s	$const_S m_s$	$m_s \in I_{In}$	σ_1^+
take m_s	$In \rightarrow T$	m_s	$has_{In} m_s$	$m_s \in I_T$	-
fake m_s	$T \rightarrow Out$	m_s	$has_T m_s$	$m_s \in I_{Out}$	-
δ_1	$Out \rightarrow R$	m_s	$has_{Out} m_s$	$m_s \in I_R$	δ_1^+
σ_2	$R \rightarrow In$	m_{sr}	$const_R m_{sr}$	$m_{sr} \in I_{In}$	δ_1^-, σ_2^+
take m_{sr}	$In \rightarrow T$	m_{sr}	$has_{In} m_{sr}$	$m_{sr} \in I_T$	-
fake m_{sr}	$T \rightarrow Out$	m_{sr}	$has_T m_{sr}$	$m_{sr} \in I_{Out}$	-
δ_2	$Out \rightarrow S$	m_{sr}	$has_{Out} m_{sr}$	$m_{sr} \in I_S$	δ_2^+
σ_3	$S \rightarrow In$	m_r	$const_S m_r$	$m_r \in I_{In}$	δ_2^-, σ_3^+
take m_r	$In \rightarrow T$	m_r	$has_{In} m_r$	$m_r \in I_T$	-
fake m_r	$T \rightarrow Out$	m_r	$has_T m_r$	$m_r \in I_{Out}$	-
δ_3	$Out \rightarrow R$	m_r	$has_{Out} m_r$	$m_r \in I_R$	δ_3^+

5. Verification of the Protocol

5.1. The Protocol Targets

In the initial state, S and R have their own key. Only S has a secret message m . Our goal is that R will have m but T can not have m after the execution of all actions. So, we give the target formulas.

Target 1: After the execution of the last action δ_3 , R will have secret message m .

$$\mathfrak{M} \models \bigwedge_{\theta \in \Theta} [Crypto_{Intr}, \delta_3(\theta)] K_{Ag} const_R m(\theta)$$

Target 2: T does not have m at the initial state. After execution of the last action δ_3 , T still does not have secret message m .

$$\mathfrak{M} \models \bigwedge_{\theta \in \Theta} \neg has_T m(\theta) \rightarrow [Crypto_{Intr}, \delta_3(\theta)] K_{Ag} \neg const_T m(\theta)$$

If this protocol satisfies Target 1 and 2, then the protocol is secure.

5.2. Verification of the Protocol

Proposition 1. For any set of message terms M , and any secret message m : without k , if $m \notin \overline{M}$ then $m \notin \overline{m_k \cup M}$.

Proof. From the construction rules, if an agent has m_k and corresponding k then he can compute m . Without k , if m can not be derived from M , then m can not be constructed even if m_k is added, because of lacking a premise condition. According to this proposition, T can not compute secret message m , because all messages he has from the In-buffer are encrypted data and he does not have the corresponding k . $\square$

In this system, messages in the information set for any agent are always increasing and never decreasing or deleting. So, we have:

Proposition 2. For any message m and any agent a , if agent a can construct message m , then he still can construct message m after the execution of the protocol actions.

$$\mathfrak{M} \models \text{const}_a m \rightarrow [\text{Crypto}_{\text{Intr}}] \text{const}_a m$$

Proposition 3. Target 2 is globally true in the model $\mathfrak{M}$.

In the initial state, only S has the secret message m and others do not have m . Of course, the intruder has m impossibly because any action is not done yet.

Now, we prove that Target 2 is still true after the execution of the sequence of the actions.

Proof. We assume there exists a world w that satisfies the below case:

$$\mathfrak{M}, w \not\models \neg \text{has}_T m \rightarrow [\text{Crypto}_{\text{Intr}}] \neg \text{const}_T m$$

If the above formula holds, then there exists an action sequence $\sigma_1, \sigma_2, \dots, \sigma_n (\sigma_i \in \mathcal{A})$ such that

$$\mathfrak{M}, w \models \neg \text{has}_T m \wedge [\text{Crypto}_{\text{Intr}}, \sigma_1] \dots [\text{Crypto}_{\text{Intr}}, \sigma_n] \text{const}_T m$$

From Table 3, we see that:

$$\mathfrak{M}, w \models \neg \text{has}_T m \wedge [\text{Crypto}_{\text{Intr}}, \sigma_1] \dots [\text{Crypto}_{\text{Intr}}, \delta_3] \neg \text{const}_T m$$

So, the assumption contradicts it.

The intruder only can get the messages from the In-buffer. So, if $m \in I_{In}$ then $m \in I_T$. We list the information in the In-buffer $\bigcup_{1 \leq i \leq n} I_{(w_i, In)} = \{m_s, m_{sr}, m_r\}$. Therefore, all the information T has is encrypted. Without corresponding k , he can not compute and get secret message m . Therefore, Target 2 is satisfied. $\square$

Proposition 4. Target 1 does not hold in the initial state. But after the running of the last action δ_3 , Target 1 holds.

Proof. In the initial state, R does not have m and R only has his own key k_r (that is assumed in the protocol). According to Table 3, the information set of R is $\bigcup_{1 \leq i \leq n} I_{(w_i, R)} = \{m_s, m_{sr}, m_r\}$. Also, R has k_r and m_r . According to the construction rules, R can construct m , namely, $\text{const}_R m$.

Therefore, after the execution of all actions, we have:

$$\mathfrak{M} \models [\sigma_1][take][fake][\delta_1][\sigma_2][take][fake][\delta_2][\sigma_3][take][fake][\delta_3]const_R m$$

which conforms to Target 1 (and everybody in the protocol knows that R has an encrypted message and corresponding key and R can construct m). $\square$

Therefore, this protocol is secure.

6. Conclusions and Expectation

We have introduced a formal language based on dynamic epistemic logic to verify the security of the cryptographic protocol. Especially, we use the action model to describe the execution process of a protocol. The updated model displays transformation between models. We have given the verification model and proved that this protocol meets these requirements. Compared with papers [7,48], our method does not need to establish complex models or rules to derive the security of the protocol. We can formalize the whole protocol running process only through an action model. This process is simple, intuitive and efficient.

In the future research, we will extend the frame of the language $\mathcal{L}_{crypt}^{A,B}$ to make it more expressive and able to express different cryptographic protocols. We should build a complete and sound axiomatization for the extended language. In this protocol, we assumed the attacker is a passive attacker who only can eavesdrop the messages from the network. In the future study, we will assume the attacker is an active attacker and he has more abilities to launch active attacks against the cryptographic protocol, and we plan to build an automatic detection system for security verification including protocols with quantum cryptography as a general modeling tool.

Author Contributions: Conceptualization, X.C. and H.D.; methodology, X.C.; validation, X.C.; formal analysis, X.C.; investigation, X.C.; resources, H.D.; writing—original draft preparation, X.C.; writing—review and editing, H.D.; supervision, H.D.; project administration, X.C. and H.D.; funding acquisition, X.C. and H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Project of Fundamental Research Funds for the Central Universities (No. XDJK2020B035), and Research Project of Education and Teaching Reform of Southwest University in 2020 (No. 2020JY055), and Special Project of Fundamental Research Funds for the Central Universities (No. SWU2009107), and Special Key Project of Technological Innovation and Application Development from Bureau of Science and Technology in Chongqing (No. cstc2019jscx gksbx0103), and the Key Base Project of Chongqing under Grant 16SKB040, and the Great Project of National Social Science Fund under Grant14ZDB016.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Burrows, M.; Abadi, M.; Needham, R.M. A Logic of Authentication. *ACM Trans. Comput. Syst.* **1990**, *8*, 18–36. [\[CrossRef\]](#)
2. Paulson, L.C. The Inductive Approach to Verifying Cryptographic Protocols. *J. Comput. Secur.* **1998**, *6*, 85–128. [\[CrossRef\]](#)
3. Roscoe, A.W.; Broadfoot, P.J. Proving Security Protocols with Model Checkers by Data Independence Techniques. *J. Comput. Secur.* **1999**, *7*, 147–190. [\[CrossRef\]](#)
4. Cirstea, H. Specifying Authentication Protocols Using Rewriting and Strategies. In Proceedings of the Practical Aspects of Declarative Languages, Third International Symposium, PADL 2001, Las Vegas, NV, USA, 11–12 March 2001; pp. 138–152. [\[CrossRef\]](#)
5. Lowe, G. Breaking and Fixing the Needham-Schroeder Public-Key Protocol Using FDR. *Softw. Concepts Tools* **1996**, *17*, 93–102.
6. Dolev, D.; Yao, A.C. On the security of public key protocols. *IEEE Trans. Inf. Theory* **1983**, *29*, 198–207. [\[CrossRef\]](#)

7. Van Ditmarsch, H.; Van Eijck, J.; Hernández-Antón, I.; Sietsma, F.; Simon, S.; Soler-Toscano, F. Modelling Cryptographic Keys in Dynamic Epistemic Logic with DEMO. In *Highlights on Practical Applications of Agents and Multi-Agent Systems*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 155–162.
8. Pirandola, S.; Andersen, U.L.; Banchi, L.; Berta, M.; Bunandar, D.; Colbeck, R.; Englund, D.; Gehring, T.; Lupo, C.; Ottaviani, C.; et al. Advances in Quantum Cryptography. *arXiv* **2019**, arXiv:1906.01645.
9. Bacco, D.; Da Lio, B.; Cozzolino, D.; Da Ros, F.; Guo, X.; Ding, Y.; Sasaki, Y.; Aikawa, K.; Miki, S.; Terai, H.; et al. Boosting the secret key rate in a shared quantum and classical fibre communication system. *Commun. Phys.* **2019**, *2*. [[CrossRef](#)]
10. Gong, L.; Needham, R.; Yahalom, R. Reasoning about belief in cryptographic protocols. In Proceedings of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 7–9 May 1990; pp. 234–248.
11. van Oorschot, P. Extending Cryptographic Logics of Belief to Key Agreement Protocols. In Proceedings of the 1st ACM Conference on Computer and Communications Security, CCS '93, Fairfax, VA, USA, 3–5 November 1993; ACM: New York, NY, USA, 1993; pp. 232–243.
12. Mao, W.; Boyd, C. Towards formal analysis of security protocols. In Proceedings of the Computer Security Foundations Workshop VI, Franconia, NH, USA, 15–17 June 1993; pp. 147–158.
13. Mansoor, K.; Ghani, A.; Chaudhry, S.A.; Shamshirband, S.; Ghayyur, S.; Mosavi, A. Securing Iot-Based RFID Systems: A Robust Authentication Protocol Using Symmetric Cryptography. *Sensors* **2019**, *19*, 4752. [[CrossRef](#)]
14. Boussada, R.; Hamdane, B.; Elhdhili, M.E.; Saïdane, L.A. Privacy-preserving aware data transmission for IoT-based e-health. *Comput. Netw.* **2019**, *162*, 106866. [[CrossRef](#)]
15. Ghani, A.; ul Hassan, K.M.; Mehmood, S.; Chaudhry, S.A.; Saqib, M.N. Security and key management in IoT wireless sensor networks: An authentication protocol using symmetric key. *Int. J. Commun. Syst.* **2019**, *32*, 1–18. [[CrossRef](#)]
16. Wang, F.; Xu, G.; Gu, L. A Secure and Efficient ECC-Based Anonymous Authentication Protocol. *Secur. Commun. Netw.* **2019**, *2019*, 1–13. [[CrossRef](#)]
17. Abadi, M.; Tuttle, M.R. A Semantics for a Logic of Authentication (Extended Abstract). In Proceedings of the Tenth Annual ACM Symposium on Principles of Distributed Computing, PODC '91, Montreal, QC, Canada, 19–21 August 1991; ACM: New York, NY, USA, 1991; pp. 201–216.
18. Syverson, P.F.; van Oorschot, P.C. On unifying some cryptographic protocol logics. In Proceedings of the 1994 IEEE Computer Society Symposium on Research in Security and Privacy, Oakland, CA, USA, 16–18 May 1994; pp. 14–28.
19. Syverson, P.; Van Oorschot, P. *A Unified Cryptographic Protocol Logic*; AD-a464 967; Defense Technical Information Center: Fort Belvoir, VA, USA, 1996.
20. Moser, L. A logic of knowledge and belief for reasoning about computer security. In Proceedings of the Computer Security Foundations Workshop II, Franconia, NH, USA, 18–20 June 1989; pp. 57–63.
21. Lei, X.; Xue, R.; Yu, T. A Timed Logic for Modeling and Reasoning about Security Protocols. *IACR Cryptol. EPrint Arch.* **2010**, *2010*, 645.
22. Datta, A.; Derek, A.; Mitchell, J.C.; Roy, A. Protocol Composition Logic (PCL). *Electr. Notes Theor. Comput. Sci.* **2007**, *172*, 311–358. [[CrossRef](#)]
23. Paulson, L.C. Relations Between Secrets: The Yahalom Protocol. In Proceedings of the Security Protocols, 7th International Workshop, Cambridge, UK, 19–21 April 1999; pp. 73–84.
24. Milner, R.; Parrow, J.; Walker, D. A Calculus of Mobile Processes, I. *Inf. Comput.* **1992**, *100*, 1–40. [[CrossRef](#)]
25. Hoare, C.A.R. Communicating Sequential Processes. *Commun. ACM* **1978**, *21*, 666–677. [[CrossRef](#)]
26. Milner, R. *A Calculus of Communicating Systems*; Springer: Berlin/Heidelberg, Germany, 1982.
27. Huszti, A.; Olah, N. Security analysis of a cloud authentication protocol using applied pi calculus. *IJIPT* **2019**, *12*, 16–25. [[CrossRef](#)]
28. Hirschi, L.; Baelde, D.; Delaune, S. A method for unbounded verification of privacy-type properties. *J. Comput. Secur.* **2019**, *27*, 277–342. [[CrossRef](#)]
29. Abadi, M. Secrecy by Typing in Security Protocols. In Proceedings of the Theoretical Aspects of Computer Software, Third International Symposium, TACS '97, Sendai, Japan, 23–26 September 1997; pp. 611–638.
30. Abadi, M.; Blanchet, B. Secrecy types for asymmetric communication. *Theor. Comput. Sci.* **2003**, *298*, 387–415. [[CrossRef](#)]

31. Gordon, A.D.; Jeffrey, A. Authenticity by Typing for Security Protocols. In Proceedings of the 14th IEEE Computer Security Foundations Workshop (CSFW-14 2001), Cape Breton, NS, Canada, 11–13 June 2001; pp. 145–159.
32. Gordon, A.D.; Jeffrey, A. Types and Effects for Asymmetric Cryptographic Protocols. In Proceedings of the 15th IEEE Computer Security Foundations Workshop (CSFW-15 2002), Cape Breton, NS, Canada, 24–26 June 2002; pp. 77–91.
33. Cardelli, L.; Ghelli, G.; Gordon, A.D. Secrecy and Group Creation. In Proceedings of the CONCUR 2000—Concurrency Theory, 11th International Conference, University Park, PA, USA, 22–25 August 2000; pp. 365–379.
34. Abadi, M.; Blanchet, B. Analyzing security protocols with secrecy types and logic programs. *J. ACM* **2005**, *52*, 102–146. [\[CrossRef\]](#)
35. Gattinger, M.; Van Eijck, J.V. Towards Model Checking Cryptographic Protocols with Dynamic Epistemic Logic. In Proceedings of the LAMAS (LAMAS 2015), Istanbul, Turkey, 4 May 2015. Available online: <http://www.irit.fr/~Emiliano.Lorini/LAMAS2015/accepted.htm> (accessed on 30 June 2020).
36. Chen, X.; Deng, H. Analysis of Cryptographic Protocol by Dynamic Epistemic Logic. *IEEE Access* **2019**, *7*, 29981–29988. [\[CrossRef\]](#)
37. Ahmadi, S.; Fallah, M.S. An Omniscience-Free Temporal Logic of Knowledge for Verifying Authentication Protocols. *Bull. Iran. Math. Soc.* **2018**, *44*, 1243–1265. [\[CrossRef\]](#)
38. Luis Uso-Domenech, J.; Antonio Nescolarde-Selva, J.; Segura-Abad, L. Proposal for the Formalization of Dialectical Logic. *Mathematics* **2016**, *4*, 69. [\[CrossRef\]](#)
39. Uso Domenech, J.L.; Antonio Nescolarde-Selva, J.; Segura-Abad, L. Dialectical Multivalued Logic and Probabilistic Theory. *Mathematics* **2017**, *5*, 15. [\[CrossRef\]](#)
40. Yang, E. Fixpointed Idempotent Uninorm (Based) Logics. *Mathematics* **2019**, *7*, 107. [\[CrossRef\]](#)
41. Smets, S.; Velzquez-Quesada, F.R. The Creation and Change of Social Networks: A Logical Study Based on Group Size. In Proceedings of the International Workshop on Dynamic Logic, Brasilia, Brazil, 23–24 September 2017.
42. Van Ditmarsch, H.; Knight, S. Announcement as effort on topological spaces. *Synthese* **2019**, *196*, 2927–2969. [\[CrossRef\]](#)
43. Christian, B.J.; Mattias, S. A Dynamic Solution to the Problem of Logical Omniscience. *J. Philos. Log.* **2019**, *48*, 501–521.
44. Philippe, B.; David, F.D.; Emiliano, L. The Dynamics of Epistemic Attitudes in Resource-Bounded Agents. *Stud. Log.* **2019**, *107*, 457–488.
45. Li, Y.; Kooi, B.; Wang, Y. A dynamic epistemic framework for reasoning about conformant probabilistic plans. *Artif. Intell.* **2019**, *268*, 54–84. [\[CrossRef\]](#)
46. Alexandru, B.; Nick, B.; Sonja, S. A Topological Approach to Full Belief. *J. Philos. Log.* **2019**, *48*, 205–244.
47. Badura, C.; Berto, F. Truth in Fiction, Impossible Worlds, and Belief Revision. *Australas. J. Philos.* **2018**, *2*, 1–16. [\[CrossRef\]](#)
48. Al Bataineh, O.; Van Der Meyden, R. Abstraction for Epistemic Model Checking of Dining Cryptographers-based Protocols. In Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge, Groningen, The Netherlands, 12–14 July 2011; pp. 247–256.
49. van Ditmarsch, H.P.; Kooi, B.P. Semantic results for ontic and epistemic change. *arXiv* **2006**, arXiv:cs/0610093.
50. Francien, D.; Yanjing, W. Dynamic epistemic verification of security protocols: framework and case study. In Proceedings of the Workshop on Logic, Rationality and Interaction, Beijing, China, 5–9 August 2007; pp. 129–144.
51. Paulson, L.C. Proving Properties of Security Protocols by Induction. In Proceedings of the 10th IEEE Workshop on Computer Security Foundations, CSFW '97, Rockport, MA, USA, 10–12 June 1997; IEEE Computer Society: Washington, DC, USA, 1997; pp. 70–83.
52. Cremers, C.; Mauw, S. Operational Semantics of Security Protocols. In *Models, Transformations and Tools*; Leue, I.S., Systä, T., Eds.; Springer: Berlin/Heidelberg, Germany, 2003; Volume 3466, pp. 66–89.