Journal of Information Security, **2010, 1, 56-67**
doi:10.4236/jis.2010.12007 Published Online October 2010 (http://www.SciRP.org/journal/jis)

Extending the Strand Space Method with Timestamps: Part II Application to Kerberos V*

Yongjian Li[1,2], Jun Pang[3]
[1]*Chinese Academy of Sciences, Institute of Software Laboratory of Computer Science, Beijing, China*
[2]*The State Key Laboratory of Information Security, Beijing, China*
[3]*University of Oldenburg Department of Computer Science Safety-critical Embedded Systems, Oldenburg, Germany*
E-mail: *lyj238@ios.ac.cn, jun.pang@informatik.uni-oldenburg.de*
Received June 23, 2010; *revised September* 14, 2010; *accepted July* 12, 2010

Abstract

In this paper, we show how to use the novel extended strand space method to verify Kerberos V. First, we formally model novel semantical features in Kerberos V such as timestamps and protocol mixture in this new framework. Second, we apply unsolicited authentication test to prove its secrecy and authentication goals of Kerberos V. Our formalization and proof in this case study have been mechanized using Isabelle/HOL.

Keywords: Strand Space, Kerberos V, Theorem Proving, Verification, Isabelle/HOL

1. Introduction

The first version of Kerberos protocol was developed in the mid eighties as part of project Athena at MIT [1]. Over twenty years, different versions of Kerberos protocols have evolved. Kerberos V (**Figure 1** and **Figure 2**) is the latest version released by the Internet Engineering Task Force (IETF) [2]. It is a password-based system for authentication and authorization over local area networks. It is designed with the following aims: once a client authenticates himself to a network machine, the process of obtaining authorization to access another network service should be completely transparent to him. Namely, the client only needs enter his password once during the authentication phase.

As we introduced in the previous paper [3], there are two novel semantic features in Kerberos V protocol. First, it uses timestamps to prevent replay attacks, so this deficiency of the strand space theory makes it difficult to analyze these protocols. Second, it is divided into three causally related multiple phases: authentication, authorization, and service protocol phases. One phase may be used to retrieve a ticket from a key distribution center, while a second phase is used to present the ticket to a security-aware server. To make matters more complex, Kerbeors uses timestamps to guarantee the recency of these tickets, that is, such tickets are only valid for an interval, and multiple sub-protocol sessions can start in parallel by the same agent using the same ticket if the ticket does not expire. Little work has been done to formalize both the timestamps and protocol mixture in a semantic framework.

The aim of this paper is practical. We hope to apply the extended theory in [3] to the analysis of Kerberos V protocol. Kerberos V is appropriate as our case study because it covers both timestamps and protocol mixture semantical features.

Structure of the Paper: Section 2 briefly introduces the overview of Kerberos V. Section 3 presents the formalization of Kerberos V. Sections 4 and 5 prove its secrecy and authentication goals. We discuss related work and conclude the paper in Section 6.

2. An Overview of Kerberos V

The protocol's layout and its message exchanging are presented in **Figure 1** and **Figure 2** separately. In the infrastructure of the Kerberos V protocol, there is a unique authentication server, and some (not necessarily only one) ticket granting servers. The latter assumption is different from that in [4], where only a unique ticket granting server exists.

*This is a revised and extended version of the homonymous paper appearing in the Proceedings the Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2007, IEEE Computer Society). The main modifications have been made on the presentation of the technical material, with the purpose of having full details. The first author is supported by grants (No.60496321, 60421001) from National Natural Science Foundation of China.

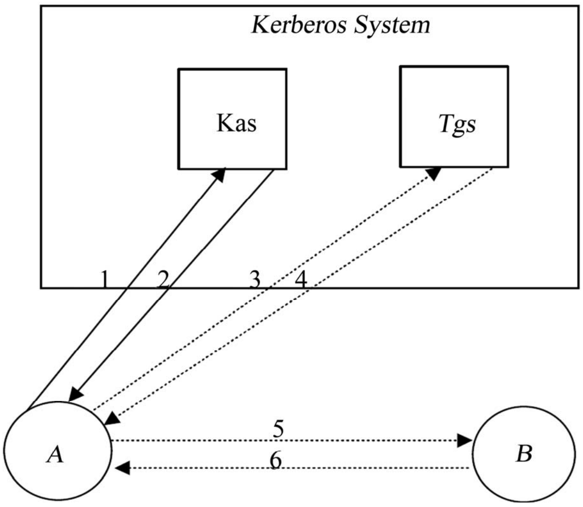

Figure 1. The layout of Kerberos V.

Authentication phase

$1.\ A \rightarrow Kas: \{\!|A, Tgs|\!\}$

$$2.\ Kas \rightarrow A: \{\!\Big|\ \underbrace{\{\!|A, Tgs, authK, Ta|\!\}_{K_{Tgs}}}_{authTicket}, \{\!|A, Tgs, authK, Ta|\!\}\ KA \Big|\!\}$$

Authorisation Phase

$$3.\ A \rightarrow Tgs: \{\!\Big|\ \{\!|A, Tgs, authK, Ta|\!\}_{K_{Tgs}}, \{\!|A, t_2|\!\}_{authK}, B \Big|\!\}$$

$$4. Tgs \rightarrow A: \{\!\Big|\ \underbrace{\{\!|A, B, seruK, T_S|\!\}_{K_B}}_{seruTicket}, \{\!|A, B, seruK, T_S|\!\}_{authK} \Big|\!\}$$

Service Phase

$$5.\ A \rightarrow B: \{\!|A, B, seruK, T_S|\!\}_{K_B}, \{\!|A, t_3|\!\}_{seruK} \Big|\!\}$$

$$6.\ B \rightarrow A: \{\!|t_3|\!\}_{seruK}$$

Figure 2. Kerberos V: message exchanging.

In order to access some network service, the client needs to communicate with two trusted servers **Kas** and *Tgs*. **Kas** is an authentication server (or the key distribution center) and it provides keys for communication between clients and ticket granting servers. *Tgs* is a ticket granting server and it provides keys for communication between clients and application servers. The full protocol has three phases each consisting of two messages between the client and one of the servers in turn. Messages 2 and 4 are different from those in Kerberos IV [1,4] in that nested encryption has been cancelled. Later we will show that this change does not affect goals of the protocol.

Detailed explanation about Kerberos V is delayed to Section 2, where the protocol is formalized in strand space model with our extensions. Here we only give an overview of the general principles to guarantee recency, secrecy and authentication in the design of Kerberos V. For recency,

- A regular sender should attach a timestamp to indicate the time when the message is issued; usually such a message is of the form $\{\!|\ldots,t,\ldots|\!\}_K$, where t is the time, K may be either a session key or long-term key.
- When a regular receiver the message $\{\!|\ldots,t,\ldots|\!\}_K$ first he need be ensured of K's secrecy to guarantee that the message is not froged by the penetrator. Second he check the recency of the message by comparing the timestamp t with the reception time. More formally, if the receiving node is n, then $time(n)$ should be no later than $cracktime(K)+t$, meaning that this message cannot be cracked at $time(n)$, which in turn indicates that the message $\{\!|\ldots,t,\ldots|\!\}_K$ is recent.

For an encrypted message $\{\!|h|\!\}_K$, the secrecy of a part of the plain message h also comes from both the secrecy of K and the recency of the message $\{\!|h|\!\}_K$ itself. That is to say, when a regular receives $\{\!|h|\!\}_K$ at time t, it must be ensured that the aforementioned two conditions must be guaranteed until t. From this, we can see that recency and secrecy are closely related with each other in a timed protocol framework.

Unsolicited tests are the main mechanism to guarantee authentication. Because a guarantee of the existence of a regular node can be drawn from an unsolicited test, a regular agent uses unsolicited test to authenticate its regular protocol participant in Kerberos V.

Now let us briefly review the main theoretical results in [3], which will be used in this work. For interesting readers, refer to [3] for preliminary definitions.

If an agent is not a penetrator then his shared key cannot be penetrated, which is formalized as follows:

Axiom 1 *If* $A \notin \mathbf{Bad}$, *then* $K_A \notin \mathbf{K}_{\mathcal{P}}$.

Lemma 1 is the main technique used to reason about authentication guarantee of a node n which is an unsolicited test for an encrypted term of the form $\{\!|h|\!\}_K$ (e.g., the tickets $\{\!|A, Tgs, authK, T_a|\!\}_{K_A}$, $\{\!|A, t|\!\}_{authK}$, and so on). That is to say, regular agents can use an unsolicited test with other properties of the protocol to guarantee that the agent who originates the term $\{\!|h|\!\}_K$ should be an intended regular agent.

Lemma 1 (Unsolicited authentication test) $\mathcal{B}$ *is a given bundle. Let* n *be an unsolicited test for* $\{\!|h|\!\}_K$. *Then there exists a positive regular node* m *in* $\mathcal{B}$ *such that* $m \preceq_{\mathcal{B}} n$ *and* $\{\!|h|\!\}_K \sqsubset term(m)$ *and* $\{\!|h|\!\}_K \not\sqsubset term(m')$ *for any node* m' *such that* $m' \preceq_{\mathcal{B}} m$.

Let a be an atomic message that uniquely originates at some node n, m be a positive penetrator node in a bundle $\mathcal{B}$ such that and $a \sqsubset term(m)$. Suppose M is a test suite for a w.r.t. m in the bundle $\mathcal{B}$. A strand's receiving nodes get messages which are all in $synth(M)$, but a new message, which is not in $synth(M)$, is sent in the strand, then the strand must be regular because a penetrator strand can not create such a term.

Lemma 2 *Let m be a positive node in a bundle $\mathcal{B}$, a be an atomic message that uniquely originates at a regular node n, M be a message set such that $suite(M,a,m,n,\mathcal{B})$, and $term(m') \in synth(M)$ for any node such that $m' \Rightarrow^{+} m$, and $term(m) \notin synth(M)$, then m is regular.*

We will illustrate these general principles in detail in the next sections when we formalize the semantics and prove secrecy properties of Kerberos V.

3. Formalizing Kerberos V

To model the time for a penetrator to break a message encrypted by a long-term shared key or a session key, we define two constants shrKcrackt ime and sessionKcrktime. The crack time of any regular agent's long-term shared key is the constant shrKcracktime,

Axiom 2 $cracktime(K_A)$ = shrKcracktime, *for any regular agent A in Kerberos V.*

The crack time of any session key originated by an authentication server is the constant sessionKcrktime.

Axiom 3 $cracktime(authK)$ = sessionKcrktime , *for any session key authK originated by* **Kas** .

The trace tr specifications of the regular strands of Kerberos V (see **Figure 2**) are defined as predicates:[1]

1) Part I (Authentication Phase)

- **Ag-I** $[i_1, A, Tgs, authK, T_a, authTicket, t_0, t_1]$ iff

$$tr(i_1) = \begin{bmatrix} (t_0,+,\{|A,Tgs|\}), \\ (t_1,-,\{|authTicket, \\ \quad \{|A,Tgs,authK,T_a|\}_{K_A}|\}) \end{bmatrix}$$

where $Tgs \in$ **TGSs** and $t_1 - T_a \le$ shrKcracktime .

- **AS** $[as, A, Tgs, authK, t_0, t_1]$ iff

$$tr(as) = \begin{bmatrix} (t_0,-,\{|A,Tgs|\}), \\ (t_1,+,\{|\{|A,Tgs,authK,t_1|\}_{K_{Tgs}}, \\ \quad \{|A,Tgs,authK,t_1|\}_{K_A}|\}) \end{bmatrix}$$

where $Tgs \in$ **TGSs** .

In the first phase, when **Kas** issues the second message

$$\{|\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}, \{|A,Tgs,authK,T_a|\}_{K_A}|\},$$

$authK$ is the session key that will be used for the client A to communicate with a ticket grant server Tgs, **Kas** attaches T_a with the message to indicate when this message is sent; if A receives this message at time t_1, A will check the condition $t_1 - T_a \le$ shrKcracktime to ensure the recency of this message. At the end of this phase, A obtains a ticket $authTicket$ and the session key $authK$ to communicate with Tgs.

2) Part II (Authorization Phase)

- **Ag-II** $[i_2, A, authK, authTicket, B, servK, T_S, servTicket, t_2, t_3]$ iff $\exists i_1, Tgs, T_a, t_0, t_1 . i_1 \mapsto i_2 \wedge$ **Ag-I** $[i_1, A, Tgs, authK, T_a, authTicket, t_0, t_1] \wedge$

$$tr(i_2) = \begin{bmatrix} (t_2,+,\{|authTicket, \\ \quad \{|A,t_2|\}_{authK}, B), \\ (t_3,-,\{|servTicket, \\ \quad \{|A,B,servK,T_S|\}_{authK}) \end{bmatrix}$$

where $Tgs \in$ **TGSs** and $t_3 - T_a \le$ shrKcracktime and $t_3 - T_S \le$ sessionKcrktime.

- **TGS** $[tgs, A, Tgs, authK, servK, B, T_a, T_0, t_0, t_1]$ iff

$$tr(tgs) = \begin{bmatrix} (t_0,-,\{|\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}, \\ \quad \{|A,T_{0\,authK},B|\}), \\ (t_1,+,\{|\{|A,B,servK,t_1|\}_{K_B}, \\ \quad \{|A,B,servK,t_1|\}_{authK}|\}) \end{bmatrix}$$

where $Tgs \in$ **TGSs,** $B \notin$ **TGSs,** $t_1 +$ sessionKcrktime $\le T_a +$ shrKcracktime .

In the second phase, the situation is more complex. Both Tgs and A need to check whether their received messages are recent by the same mechanism. Furthermore, Tgs also need ensure a side condition that

$$t_1 + \text{sessionKcrktime} \le T_a + \text{shrKcracktime}$$

to guarantee that the application server B only receives a recent service ticket. Informally speaking, this condition means that Tgs can guarantee any $authK$ that he receives can only be compromised later than $servK$ which is associated with the $authK$. We will comment this side condition in analysis in the third phase. At the end of this phase, A obtains a ticket $servTicket$ and the session key $servK$ to communicate with B.

[1] For simplicity, we assume any trace of a regular agent always respects the time order in Kerberos V protocol, and we do not include this side condition in the trace specifications.

3) Part III (Service Phase)

• **Ag-III** $[i_3, A, servK, servTicket, t_4, t_5]$ iff $\exists i_1, Tgs, authK, T_a, t_0, t_1, i_2, authTicket, B, T_S, t_2, t_3$.

$$tr(i_3) = \begin{bmatrix} \left(t_4, +, \{|servTicket, \{|A, t_4|\}_{servK}\right), \\ \left(t_5, -, \{|t_4|\}_{servK}\right) \end{bmatrix} \wedge$$

Ag-I $[i_1, A, Tgs, authK, T_a, authTicket, t_0, t_1] \wedge$

Ag-II $[i_2, A, authK, authTicket, B, servK, T_S, servTicket, t_2, t_3] \wedge i_1 \mapsto i_2 \wedge i_2 \mapsto i_3$

where $Tgs \in \mathbf{TGSs}$, $t_5 - T_a \le \text{shrKcracktime}$ and $t_5 - T_S \le \text{sessionKcrktime}$.

• **Apps** $[apps, A, B, servK, T_S, T_4, t_0, t_1]$ iff

$tr(apps) =$

$$\begin{bmatrix} (t_0, -, \{|\{|A, B, servK, T_S|\}_{K_B}, \{|A, T_4|\}_{servK}|\}), \\ (t_1, +, \{|t_1|\}_{servK}) \end{bmatrix}$$

where $t_0 - T_S \le \text{sessionKcrktime}$.

In the last phase, it is subtle for the application server B to check the recency of the message $\{|\{|A, B, servK, T_S|\}_{K_B}, \{|A, T_4|\}_{servK}|\}$. From the ticket $\{|A, B, servK, T_S|\}_{K_B}$, B knows that Tgs must have issued $\{|\{|A, B, servK, T_S|\}_{K_B}, \{|A, B, T_4, servK, T_S|\}_{authK}|\}$ at time T_S . The potential compromise of $servK$ is from the message $\{|A, B, servK, T_S|\}_{authK}$. A penetrator can either directly break $\{|A, B, servK, T_S|\}_{authK}$ to obtain $servK$, or have $authK$ first then decrypt the message $\{|A, B, servK, T_S|\}_{authK}$ to obtain $servK$. Since $authK$ is also a session key which is originated by **Kas** in an earlier time than T_s, the guarantee for the confidentiality of $authK$ is of extreme importance. The corresponding ticket $\{|A, Tgs, authK, T_a|\}_{K_{Tgs}}$ is not available for B, B cannot know the creation time of $authK$. So B cannot directly check whether $authK$ has been compromised. Fortunately, if *Tgs* can guarantee that any $authK$ which it receives will be compromised later than $servK$, associated with the $authK$, then it is enough for B to check $t_0 - T_S \le \text{sessionKcrktime}$ to ensure that the $authK$ has not been compromised. At the end of this stage, A and B authenticate each other, and A can access the service provided by B.

The authentication server **Kas** must obey the following principles to generate a session key $authK$:

- $authK$ must never be known initially to a penetrator, *i.e.*, $authK \notin \mathbf{K}_{\mathcal{P}}$;
- $authK$ must be uniquely originated;
- $authK$ is a symmetric key;
- $authK$ must not be the same as an agent's long-term shared key.

We summarize these principles as the following axiom:

Axiom 4 *For any authentication server strand* as *such that* $\mathbf{AS}[as, A, Tgs, authK, t_0, t_1]$, *we have* $authK \notin \mathbf{K}_{\mathcal{P}}$, $authK$ *uniquely originates in* $(as, 1)$, $authK = authK^{-1}$ *, and* $authK \ne K_B$ *for any agent* B .

A ticket grant server creates the session key $servK$ by three principles, which are similar to those which the authentication server obeys to create the session key $authK$.

Axiom 5 *For any ticket grant server strand* tgs *such that* $\mathbf{TGS}[tgs, A, Tgs, authK, servK, B, T_a, T_0, t_0, t_1]$, $servK \notin \mathbf{K}_{\mathcal{P}}$, $servK$ *uniquely originates in* $(tgs, 1)$, $servK = servK^{-1}$ *, and* $servK \ne K_B$ *for any agent* B .

In the following two subsections, we verify the secrecy and authentication properties of Kerberos V. We use similar ways for representing these security properties as in [5]. However, we may need formulate secrecy properties with temporal restrictions when we discuss them in a timed framework. A value v is secret for a protocol if for every bundle $\mathcal{C}$ of the protocol the penetrator cannot receive v in cleartext until some time t ; that is, there is no node n in $\mathcal{C}$ such that $term(n) = v$ and $time(n) \le t$. For Kerberos V, we mainly discuss the secrecy of a long-term key of a regular agent, and $authK$, $servK$ issued by servers. Authentication properties are specified as usual: for a participant B (e.g. acting as a responder), for a certain vector of parameters $\vec{x}$, if each time principal B completes a run of the protocol as a responder using $\vec{x}$ supposedly with A, then there is a run of the protocol with A acting as an initiator using $\vec{x}$ supposedly with B. And this is formalized as follows: there is a responder strand $Resp$ ($\vec{x}$) and the i-th node of the strand is in a bundle $\mathcal{C}$, then there is an initiator strand $Init$ ($\vec{x}$) and some j-th node of the initiator strand is in $\mathcal{C}$.

In order to prove the secrecy of a long-term key K_A, we only need use the well-founded induction principle on bundles. But the knowledge closure property on penetrators is needed when we prove the secrecy of some session key $authK$ or $servK$. For instance, in order to prove the secrecy of $authK$, we construct a set

$$M =_{df} \left\{ \{|A, Tgs, authK, T_a|\}_{K_A}, \{|A, Tgs, authK, T_a|\}_{K_{Tgs}} \right\} \cup \{t \mid authK \not\subset t\}.$$

We will show that for any node m in a Kerberos bundle $\mathcal{B}$, if $authK \sqsubset term(m)$ and $time(m) \leq$

$$T_a + \text{shrKcracktime},$$

then $term(m)$ must be in $synth(M)$. Intuitively, this fact holds because both the penetrator and regular strands can only emit a message which is in $synth(M)$. The penetrator can not decrypt or crack the messages

$$\{|A, Tgs, authK, T_a|\}_{K_A} \text{ and } \{|A, Tgs, authK, T_a|\}_{K_{Tgs}}$$

until time $T_a + \text{shrKcracktime}$, so it can only synthesize any messages which is in $synth(M)$; except a unique authentication server strand, any other regular strand can not emit any message which has $authK$ as a subterm until that time. But for the authentication server strand, he can only emit

$$\left\{\left| \{|A, Tgs, authK, t_1|\}_{K_{Tgs}}, \{|A, Tgs, authK, t_1|\}_{K_A} \right|\right\}$$

which is still in $synth(M)$. Our formal proof is by contradiction. If not so, by the well-founded induction principle on $\mathcal{B}$, we have a minimal element m such that $authK \sqsubset term(m)$ and $term(m) \notin synth(M)$. By the knowledge closure property, we can exclude the cases when m is in a penetrator strand. By case analysis on the form of the trace of regular strands, we can also exclude the case when m is in a regular strand. Thus, a contradiction is concluded.

In the following two sections, we give the detailed proof on the secrecy and authentication properties to show how to apply the proof techniques aforementioned. Note that we also have formalized all the proofs in Isabelle/HOL, and the proof scripts can be obtained at [6]. The paper proof here can be viewed as a text account of the mechanical proof scripts at [6].

4. Proving Secrecy Goals

In Kerberos V, a long-term key of a regular agent is never sent in the network, so it cannot be compromised. Let $\mathcal{B}$ be a bundle of Kerberos V. For any node in the bundle, the long-term key of a regular agent cannot be a part of the term of the node. In order to prove this lemma, we only need the well-founded induction principle on bundles.

Lemma 3 *Let* $n \in \mathcal{B}$. *If* $A \notin \mathbf{Bad}$, *then* $term(n) \neq K_A$.

Proof. Let

$$P =_{df} \{x \mid x \in \mathcal{B} \wedge K_A \sqsubset term(x)\}$$

We show that P is empty by contradiction. If there is a node $n' \in P$, then by the well-foundedness of a bundle, there exists a node m such that m is minimal in P. Namely, $m \in \mathcal{B}$, $K_A \sqsubset term(m)$, and for all $m' \in \mathcal{B}$, if $m' \preceq_{\mathcal{B}} m$ then $K_A \not\sqsubset term(m')$.

We prove that the sign of m is positive. If $sign(m) = -$, then by upward-closed property of a bundle there must be another node m'' in the bundle $\mathcal{B}$ such that $sign(m'') = +$ and $m'' \rightarrow m$. This contradicts with the minimality of m. Then m is either in a regular strand or in a penetrator strand.

• CASE 1: m is in a regular strand.

There are six cases. Here we only analyze the cases when m is in an authentication server strand $as \in \mathbf{AS}$ $[as, A, Tgs, authK, t_0, t_1]$ or m is in a client strand $i \in$ **Ag-II** $[i, A, authK, authTicket, B, servK, T_S, servTicket, t_2, t_3]$. The other cases are either straightforward or can be analyzed in a similarly.

If m is in an authentication server strand such that $as \in \mathbf{AS}\,[as, A, Tgs, authK, t_0, t_1]$. By inspection on the trace form of the strand, we have $m = (as, 1)$, $K_A \sqsubset term(as, 1)$, and

$$term(as,1) = \left\{\left| \{|A, Tgs, authK, t_1|\}_{K_{Tgs}}, \{|A, Tgs, authK, t_1|\}_{K_A} \right|\right\},$$

then

$$K_A \sqsubset \{|A, Tgs, authK, t_1|\}_{K_{Tgs}} \text{ or } K_A \sqsubset \{|A, Tgs, authK, t_1|\}_{K_A}.$$

In both cases, we can conclude that $K_A = authK$. But this contradicts with Axiom 4. If m is in a client strand such that $i \in \mathbf{Ag-II}[i, A, authK, authTicket, B, servK, T_S, servTicket, t_2, t_3]$. By inspection on the trace form of the client strand, we have $m = (i, 0)$, $K_A \sqsubset term(i, 0)$, and

$$term(i,0) = \{|authTicket, \{|A, t_2|\}_{authK}, B|\},$$

then $K_A \sqsubset authTicket$. But by the definition of the client strand, there exists some client strand i_1 suchthat$i_1 \mapsto i$ and **Ag-I** $[i_1, A, Tgs, authK, T_a, authTicket, t_0, t_1]$. From the definition of the strand , we have $authTicket \sqsubset term(i_1, 1)$. From this and $K_A \sqsubset authTicket$, we have (1) $K_A \sqsubset term(i_1, 1)$. From $i_1 \mapsto i$, we have (2) $(i_1, 1) \Rightarrow (i, 0)$. From (1) and (2), we can conclude that m is not minimal in P. This contradicts with the minimality of m.

• CASE 2: m is in a penetrator strand p.

Here we only analyze the cases when p is either K_K (key strand) or $C_{g,h}$ (concatenation). Other cases are either straightforward or can be analyzed in a similar way.

- p is K_K. We have $m = (p, 0)$ and $K_A \sqsubset K$. Then $K_A = K \in \mathbf{K}_{\mathcal{P}}$. This contradicts with Axiom 1.

- p is $C_{g,h}$. We have $m' = (p, 2)$ and $K_A \sqsubset \{|g, h|\}$. By the definition of $\sqsubset$, we have $K_A \sqsubset g$, or $K_A \sqsubset h$. If $K_A \sqsubset g$, then $K_A \sqsubset term(p, 0)$. This contradicts with the minimality of m. The case when $K_A \sqsubset h$ can be analyzed similarly.

If an authentication ticket $\{|A,Tgs,authK,T_a|\}_{K_A}$ or $\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}$ occurs as a subterm of a node in $\mathcal{B}$, A is not compromised, and Tgs is a ticket granting server, then it can be guaranteed that there must be an authentication server strand as in which $\{|A,Tgs,authK,T_a|\}_{K_A}$ and $\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}$ originate at time T_a. Therefore, T_a is the earliest time when $\{|A,Tgs,authK,T_a|\}_{K_A}$ and $\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}$ occur in $\mathcal{B}$. With the specification of the origination of the key $authK$ by **Kas** (formulated by Axiom 4), we also are ensured that T_a is the earliest time when $authK$ occurs in $\mathcal{B}$. The minimal property of T_a will be used in the proof of Lemma 5.

Lemma 4 *Let* $n \in \mathcal{B}$, $A \notin$ **Bad**, *and* $Tgs \in$ **TGSs**. *If* $\{|A,Tgs,authK,T_a|\}_{K_A}$ *or* $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset term(n)$, *then there exists an authentication server strand as such that* **AS**$[as, A, Tgs, authK, t_0, T_a]$ *for some* t_0, $(as,1) \in \mathcal{B}$, *and* $T_a \le time(n)$.

Proof.

Here we only prove the case $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset term(n)$. The other case can be proved in a similar way. First we prove that (1) n is an unsolicited test for the term $\{|A,Tgs,authK,T_a|\}_{K_A}$ We only need prove that K_A must be regular *w.r.t.* n. By Lemma 3, there is no node m in $\mathcal{B}$ such that $term(m) = K_A$, so K_A must be regular *w.r.t.* n.

From (1), by Lemma 1, there exists a positive regular node m in $\mathcal{B}$ such that $m \preceq_{\mathcal{B}} n$ and $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset term(m)$ and $\{|A,Tgs,authK,T_a|\}_{K_A} \not\sqsubset term(m')$ for any node m' such that $m' \preceq_{\mathcal{B}} m$.

From $m \preceq_{\mathcal{B}} n$ and $\mathcal{B}$ is a bundle, we can easily conclude $time(m) \le time(n)$ and $m \in \mathcal{B}$.

Now we prove that m must be in an authentication server strand. From the fact that m is regular, then we have six cases, here we select two cases when m is in an authentication server strand as such that **AS** [as, A', Tgs', $authK'$, t_0, t_1] or in an ticket granting server strand tgs such that **TGS**[tgs, A', Tg, $authK'$, $servK$, B', T_a', T_0, t_0, t_1].

- m is in an authentication server strand as such that **AS** [as, A', Tgs', $authK'$, t_0, t_1]. By inspection on the form of the strand, $m = (as,1)$ because m is positive. Obviously

$$term(m) =$$
$$\{|\{|A',Tgs',authK',t_1|\}_{K_{Tgs'}}, \{|A',Tgs',authK',t_1|\}_{K_{A'}}|\}$$

By $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset term(m)$, we have either (2) $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset \{|A',Tgs',authK',t_1|\}_{K_{Tgs'}}$ or (3) $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset \{|A',Tgs',authK',t_1|\}_{K_{A'}}$. From (2), we have $A = A'$ and $Tgs = Tgs'$ and $authK = authK'$ and $T_a = t_1$, so **AS**$[as, A, Tgs, authK, t_0, T_a]$. Case (3) can be prove similarly.

- m is in an ticket granting server strand such that **TGS** $[tgs, A', Tg, authK', servK, B', T_a', T_0, t_0, t_1]$. By inspection on the form of the strand, $m = (tgs,1)$ because m is a positive node. Obviously

$$term(m) = \{|\{|A',B',servK,t_1|\}_{K_{B'}}, \{|A',B',servK,t_1|\}_{authK'}|\}.$$

From $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset term(m)$, we have either (2) $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset \{|A',B',servK,t_1|\}_{K_{B'}}$ or (3) $\{|A,Tgs,authK,T_a|\}_{K_A} \sqsubset \{|A',B',servK,t_1|\}_{authK'}$. From (2), we can prove that $Tgs = B'$, then by the assumption $Tgs \in$ **TGSs**, we have $B' \in$ **TGSs**. But by the definition of the ticket granting server, we have $B' \notin$ **TGSs**. Therefore a contradiction is obtained. Case (3) can be proved similarly.

Once the authentication tickets $\{|A,Tgs,authK,T_a|\}_{K_A}$ or $\{|A,Tgs,authK,T_a|\}_{K_{Tgs}}$ are created by the authentication server **Kas** at T_a, then the session key $authK$ will be not compromised until the time $T_a + \mathrm{shrKcracktime}$.

Lemma 5 *Let* $n \in \mathcal{B}$, $A \notin$ **Bad**, *and* $Tgs \in$ **TGSs**. *If*

$$\{|A,Tgs,authK,T_a|\}_{K_A} \textit{ or}$$

$$\{|A,Tgs,authK,T_a|\}_{K_{Tgs}} \sqsubset term(n),$$

then for any node $m \in \mathcal{B}$ *such that* $time(m) \le T_a + \mathrm{shrKcracktime}$, $term(m) \ne authK$.

Proof. First we define two sets.

$$M =_{df} \left\{ \begin{array}{l} \{|A,Tgs,authK,T_a|\}_{K_A}, \\ \{|A,Tgs,authK,T_a|\}_{K_{Tgs}} \end{array} \right\} \cup \{t \mid authK \not\sqsubset t\}$$

$$P =_{df} \{m.m \in \mathcal{B} \wedge time(m) \le T_a + \mathrm{shrKcracktime} \wedge$$
$$term(m) \notin synth(M)\}.$$

We show that for any node $m \in \mathcal{B}$ such that $time(m) \le T_a + \mathrm{shrKcracktime}$, $term(m) \in synth(M)$. In order to prove this, we only need show P is empty. We prove the assertion by contradiction. If P is not empty, then by the well-foundedness of a bundle, (1) there exists a positive node m such that $m \in \mathcal{B}$, $term(m) \notin synth(M)$, $time(m) \le T_a + \mathrm{shrKcracktime}$, and for all $m' \in \mathcal{B}$, if $m' \preceq_{\mathcal{B}} m$ then $term(m) \in synth(M)$.

First from the fact that $\{|A,Tgs,authK,T_a|\}_{K_A}$ or $\{|A,Tgs,authK,T_a|\}_{K_{Tgs}} \sqsubset term(n)$ by the Lemma 4, then there exists an authentication server strand as such that $\mathbf{AS}[as, A, Tgs, authK, t_0, T_a]$ for some t_0, $(as,1) \in \mathcal{B}$, and $T_a \le time(n)$. From the definition of **AS** and Axiom 4, we have (2) $authK$ uniquely originates at $(as,1)$ and $time(as,1) = T_a$.

Next we prove that (3) **suite(**M, $authK$, m, $(as,1)$, $\mathcal{B}$). Here we need show both $\{|A, Tgs, authK, T_a|\}_{K_A}$ and $\{|A, Tgs, authK, T_a|\}_{K_{Tgs}}$ are components in $\mathcal{B}$. From $A \notin$ **Bad** and $Tgs \in$ **TGSs**, by Lemma 3, we have that neither K_A nor K_{Tgs} is compromised, and they are symmetry, therefore $regular(K_A^{-1}, m, \mathcal{B})$ and $regular(K_{Tgs}^{-1}, m, \mathcal{B})$; furthermore from $time(m) \le T_a +$ shrKcracktime, and by Axiom 2, $cracktime(K_A) =$ shrKcracktime, with (2), we have $time(m) \le time(as,1) + cracktime(K_A)$, similarly we have $time(m) \le time(as,1) + cracktime(K_{Tgs})$, so (3) is proved.

From (1), we have for any m' such that $m' \Rightarrow^+ m$, $term\ (m') \in synth(M)$. With (2)(3), by Lemma 2, we have m must be in a regular strand i, then there exist six cases. Here we analyze the cases when $\mathbf{AS}\ [i, A', Tgs', authK', t_0', t_1]$, other cases are more simpler. If m is in an authentication server strand $\mathbf{AS}\ [i, A', Tgs', authK', t_0', t_1]$. By inspection on the form of the strand, $m = (i,1)$ because m is positive. Obviously

$$term(m) =$$

$$\left\{\left|\{|A', Tgs', authK', t_1|\}_{K_{Tgs'}}, \{|A', Tgs', authK', t_1|\}_{K_{A'}}\right|\right\}.$$

Obviously $authK \sqsubset term\ (m)$, otherwise $term\ (m) \in M$. Therefore $authK \sqsubset \{|A', Tgs', authK', t_1|\}_{K_{A'}}$ or

$authK \sqsubset \{|A', Tgs', authK', t_1|\}_{K_{Tgs'}}$ then $authK = authK'$.

From the definition of Axiom 4, we have $authK$ uniquely originates from the strand i. Combining with (2), we have $as = i$, then $A = A'$, $Tgs = Tgs'$, $t_1 = T_a$, so

$$term(m)$$

$$= \left\{\left|\{|A, Tgs, authK, T_a|\}_{K_{Tgs}}, \{|A, Tgs, authK, T_a|\}_{K_A}\right|\right\}$$

$$\in synth(M)$$

This contradicts with the fact $term(m) \notin synth(M)$.

Therefore for any node $m \in \mathcal{B}$ such that $time(m) \le T_a +$ shrKcracktime, $term(m) \in synth(M)$. Next we only need prove that $authK \notin synth(M)$. We prove by contradiction, if $authK \in synth(M)$, by the rule inversion of definition of $synth$, we have $authK \in M$, this contradicts with the definition of M.

In order to prove the conclusion of Lemma 5, we need the conclusion of Lemma 4, which ensures us that a penetrator cannot crack the term $\{|A, Tgs, authK, T_a|\}_{K_A}$ (or $\{|A, Tgs, authK, T_a|\}_{K_{Tgs}}$) to obtain $authK$. Because the earliest time when $authK$ occurs in $\mathcal{B}$ is T_a and $authK$ can only occur in $\{|A, Tgs, authK, T_a|\}_{K_A}$ (or $\{|A, Tgs, authK, T_a|\}_{K_{Tgs}}$) the penetrator cannot crack such a term until $T_a +$ shrKcracktime, and what he can only do is to synthesize some term from M. Therefore, $authK$ must be safe until that time. Furthermore, the intermediate result of this proof tells us that $term(m)$ must be in $synth(M)$ for any node $m \in \mathcal{B}$ such that $time(m) \le T_a +$ shrKcracktime.

If both the tickets

$$\{|A, B, servK, T_S|\}_{authK} \text{ and } \{|A, Tgs, authK, T_a|\}_{K_A}$$

occur as a part of the term of a node in $\mathcal{B}$, A and B are not compromised, and B is not a ticket grant server, and $authK$ is still not compromised at the time when the above two tickets occur, then it can be guaranteed that A must have passed the first and second phases of the protocol, and a ticket grant server strand tgs must exist in $\mathcal{B}$, where two tickets

$$\{|A, B, servK, T_S|\}_{K_B} \text{ and } \{|A, B, servK, T_S|\}_{authK}$$

are issued for some session key $authK$. Similar to Lemma 4, this lemma ensures us that T_S is the earliest time when $servK$ occurs in $\mathcal{B}$, and this minimal property is needed in the proof of Lemma 7.

Lemma 6 *Let* $m, n \in \mathcal{B}$, $A \notin$ **Bad**, $Tgs \in$ **TGSs**. *If both*

$$\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(m),$$

and

$$\{|A, B, servK, T_S|\}_{authK} \sqsubset term(n)$$

and $time(n) \le T_a +$ shrKcracktime, *then there exists a ticket granting server strand* tgs *such that* $TGS[tgs, A, Tgs, authK, servK, B, T_a, T_0, t_0, T_S]$ *for some* T_0, t_0, $(tgs,1) \in \mathcal{B}$ *and* $T_S \le time(n)$.

Here we only give the proof sketch of this lemma. First we need show that (1) n is an unsolicited test for $\{|A, B, servK, T_S|\}_{authK}$ in $\mathcal{B}$. We need prove $regular(authK, n, \mathcal{B})$. This can be ensured by Lemma 5. Because $time(n) \le T_a +$ shrKcracktime, we have $term(n') \ne authK$ for any node n' such that $time(n') \le time(n)$. From (1), we can show that there is a regular node n' such that $n' \preceq_{\mathcal{B}} n$ and

$$\{|A, B, servK, T_S|\}_{authK} \sqsubset term\ (n') \text{ and}$$

$\{|A, B, servK, T_S|\}_{authK} \not\sqsubset$ *term* (m') for any node m' such that $m' \preceq_{\mathcal{B}} n$. By the case analysis on the form of regular strands, we can prove that n' must be in a ticket granting server strand tgs such that **TGS**[tgs, A, Tgs, $authK$, $servK$, B, T_a, T_0, t_0, T_S] for some T_0, t_0, $(tgs,1) = n'$ and $T_S = time(n')$.

Moreover, by the fact a ticket $\{|A, B, servK, T_S|\}_{authK}$ is originated at time T_S, then the session key $servK$ will not be compromised until the time

$$T_S + \text{sessionKcrktime}\,.$$

Because during the interval from T_S to

$$T_S + \text{sessionKcrktime}\,,$$

neither $\{|A, B, servK, T_S|\}_{authK}$ will be cracked, nor the session key $authK$ can be obtained by a penetrator to decrypt the ticket $\{|A, B, servK, T_S|\}_{authK}$.

Lemma 7 *Let* $m_0, n \in \mathcal{B}$, $A, B \notin \mathbf{Bad}$, $Tgs \in \mathbf{TGSs}$. *If both* $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(m_0)$ *and*

$$\{|A, B, servK, T_S|\}_{authK} \sqsubset term(n)\,,$$

and $time(n) \le T_a + \text{shrKcracktime}$, *then for any node* $m \in \mathcal{B}$ *such that* $time(m) \le T_S + \text{sessionKcr ktime}$, $term(m) \ne servK$.

Proof. First we define:

$$M =_{df} \left\{ \begin{matrix} \{|A, B, servK, T_S|\}_{K_B}, \\ \{|A, B, servK, T_S|\}_{authK} \end{matrix} \right\} \cup \{t \mid servK \not\sqsubset t\}.$$

We will show that for any node $m \in \mathcal{B}$ such that $time(m) \le T_S + \text{sessionKcrktime}$, $term\ (m) \in synth(M)$. We prove the assertion by contradiction.

Let $P =_{df} \{m.\ m \in \mathcal{B} \wedge time(m) \le T_S + \text{sessionKcrktime} \wedge term\ (m) \notin synth(M)\}$. If P is not empty, then by the well-foundedness of a bundle, (1) there exists a positive node m such that $m \in \mathcal{B}$, $time(m) \le T_S + \text{sessionKcrktime}$, $term\ (m) \notin synth(M)$, and for all $m' \in \mathcal{B}$, if $m' \preceq_{\mathcal{B}} m$ then $term\ (m') \in synth(M)$.

From the fact $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(m_0)$, by Lemma 4, there exists an authentication server strand as such that **AS**[$as, A, Tgs, authK, t_0, T_a$] for some t_0, $(as,1) \in \mathcal{B}$. From the definition of **AS**, we know (2) $authK$ uniquely originates at $(as,1)$ and $time(as,1) = T_a$.

From the fact that $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(m_0)$ and $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(n)$, by Lemma 6, then there exists a ticket granting server strand tgs such that **TGS**[tgs, A, Tgs, $authK$, $servK$, B, T_a, T_0, t_0, T_S] for some T_0, t_0. From the definition of **TGS** and Axiom 5, we have (3) $servK$ uniquely originates at $(tgs,1)$, $time(tgs,1) = T_S$,

$$\{|A, Tgs, authK, T_a|\}_{K_{Tgs}} \sqsubset term\ (tgs,0),$$

and $T_S + \text{sessionKcrktime} \le T_a + \text{shrKcracktime}$. From $\{|A, Tgs, authK, T_a|\}_{K_{Tgs}} \sqsubset term\ (tgs,0)$, by Lemma 4, we can easily conclude that $T_a \le time(tgs,0)$, then (4) $T_a \le time(tgs,1)$.

Next we prove that (5) $suite(M, servK, m, (tgs,1), \mathcal{B})$. Here we need show both $\{|A, Tgs, servK, T_a|\}_{K_B}$ and $\{|A, B, servK, T_S|\}_{authK}$ are components in $\mathcal{B}$. From $B \notin \mathbf{Bad}$, by Lemma 3, we have K_B are never compromised, similar to counterpart in Lemma 5, we can prove that $regular(K_B^{-1}, m, \mathcal{B})$. From

$$T_S + \text{sessionKcrktime} \le T_a + \text{shrKcracktime}$$

and $time(m) \le T_S + \text{sessionKcrktime}$, we have (6) $time(m) \le T_a + \text{shrKcracktime}$, with (4), we have $time(m) \le time(tgs,1) + cracktime(K_B)$. From (6) and $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(m)$ for any node n' such that $time(n') \le time(m)$, we have

$$time(n') \le T_S + \text{sessionKcrktime}\,,$$

then $time(n') \le T_a + \text{shrKcrktime}$, by Lemma 5, $term\ (n') \ne authK$; by Axiom 4, $authK$ is symmetry, therefore $authK^{-1} = authK$, so $regular(authK^{-1}, m, \mathcal{B})$. From $time(m) \le T_S + \text{sessionKcrktime}$, and by Axiom 4 again, we have $cracktime(authK) = \text{sessionKcrktime}$, then $time(m) \le T_S + cracktime(authK)$, with (3), we have $time\ (m) \le time\ (tgs,1) + cracktime\ (authK)$. Therefore (5) is proved.

From (1), we have for any m' such that $m' \Rightarrow^{+} m$, $term\ (m') \in synth\ (M)$. With (2)(5), by Lemma 2, we have m must be in a regular strand i, then there exists six cases. Here we analyze the cases when **AS**[i, A', Tgs', $authK'$, t_0', t_1] or **TGS**[i, A', Tgs', $authK'$, $servK'$, B', T_a', T_0', t_0', T_S'], other cases are more simpler.

If **AS**[i, A', Tgs', $authK'$, t_0', t_1], then by inspection on the form of the strand, $m = (i,1)$ because m is positive. Obviously

$$term(m) = \left\{ \left\{|A', Tgs', authK', t_1|\right\}_{K_{Tgs'}}, \left\{|A', Tgs', authK', t_1|\right\}_{K_{A'}} \right\}.$$

Obviously $servK \sqsubset term\ (m)$, otherwise $term(m) \in M$. Therefore $servK \sqsubset \{|A', Tgs', authK', T_a|\}_{K_{A'}}$ or

$$servK \sqsubset \{|A', Tgs', authK', T_a|\}_{K_{Tgs'}}$$

then $servK = authK'$. From Axiom 5, we have $authK$ uniquely originates from the strand i. Combining with (3), we can conclude i is both an authentication server strand and a ticket granting server strand, obviously this is a contradiction.

If $\mathbf{TGS}[i, A', Tgs', authK', servK', B', T_a', T_0', t_0', T_S']$, then by similar argument, we can prove that $servK'$ = $servK$, from Axiom 5, we have $servK$ uniquely originates from i, with (3), we have $i = tgs$, we can prove that

$$term(m) = \{|\{|A, B, servK, T_S|\}_{K_B}, \{|A, B, servK, T_S|\}_{authK}|\}$$

then $term\ (m) \in synth\ (M)$ this contradicts with (1).

At last we only need prove that $term\ (m) \in synth\ (M)$ implies that $term\ (m) \neq servK$, this is similar to counterpart in Lemma 5.

Both Lemma 6 and Lemma 7 have the assumption that $\{|A, Tgs, authK, T_a|\}_{K_A}$ is a subterm of the node n, which can guarantee that $authK$ must be a session key originated by an authentication server strand. The assumption $time(n) \leq T_a + \text{shrKcracktime}$ is used to guarantee that $authK$ is still safe at $time(n)$. Besides, the two terms $\{|A, B, servK, T_S|\}_{authK}$ and $\{|A, Tgs, authK, T_a|\}_{K_A}$ are intelligible for the client A, so these two lemmas are secrecy properties in the view of A.

In Lemmas 6 and 7, both $\{|A, B, servK, T_S|\}_{authK}$ and $\{|A, Tgs, authK, T_a|\}_{K_A}$ are unintelligible for an application server B because $authK$ and K_A cannot be known by B. So the two properties are not in B's view. B can only receive a message such as $\{|A, B, servK, T_S|\}_{K_B}$, can it be ensured that $servK$ is confidential when he receives the message $\{|A, B, servK, T_S|\}_{K_B}$? The following two lemmas are about the confidential information inferred from the message $\{|A, B, servK, T_S|\}_{K_B}$. They are secrecy properties in B's view.

Once a server ticket such as $\{|A, B, servK, T_S|\}_{K_B}$ occurs in a bundle, where A and B are not compromised, and B is not a ticket granting server, then conclusions similar to those in Lemma 6 and Lemma 7 can be drawn.

Lemma 8 *Let* $n \in \mathcal{B}$, $A, B \notin \mathbf{Bad}$, *and* $B \notin \mathbf{TGSs}$. *If* $\{|A, B, servK, T_S|\}_{K_B} \sqsubset term(n)$, *then there exists a ticket grant server strand tgs such that* $TGS[tgs, A, Tgs, authK, servK, B, T_a, T_0, t_0, T_S]$ *for some* Tgs, $authK$, T_a, T_0, t_0, $(tgs, 1) \in \mathcal{B}$ *and* $T_S \leq time(n)$.

Lemma 9 *Let* $n \in \mathcal{B}$, $A, B \notin \mathbf{Bad}$, *and* $B \notin \mathbf{TGSs}$. *If* $\{|A, B, servK, T_S|\}_{K_B} \sqsubset term(n)$, *then for any node* $m \in \mathcal{B}$ *such that* $time\,(m) \leq T_S + \text{sessionKcr ktime}$, $term(m) \neq servK$.

Here we summarize the main ideas used in the above proof of secrecy properties.

- For a long-term key of a regular agent, its secrecy is easily inferred because it is never sent as a part of a message. We only need the well-founded induction principle on bundles to prove this.
- But for a short session key $authK$ or $servK$, the cases are more complex because they are sent as a part in a message such as

$$\{|A, Tgs, authK, T_a|\}_{K_A} \text{ or } \{|A, B, servK, T_S|\}_{authK}.$$

In kerberos V, a session key such as $authK$ ($servK$) occurs as a part of a term of node n which is of the form $\{|h|\}_K$, where K can be either a long-term key or another short session key, and h also contains a timestamp t such as $T_a(T_S)$, which indicates the time when $\{|h|\}_K$ is t. As mentioned before, both secrecy of K and recency of $\{|h|\}_K$ should be guaranteed. Secrecy of K can be directly drawn from other lemmas on K. But for recency checking, firstly we need prove that the timestamp t indeed indicates the time when $\{|h|\}_K$ is originated. Lemmas 4, 6, 9 play a role in guaranteeing that t is the first time when $authK$ ($servK$) is originated. From this and the assumption that $time(n) \leq t + \text{cracktime}$ (K), the recency of $\{|h|\}_K$ can be proved.

5. Proving Authentication Goals

For convenience, we call that a strand i uses a term $\{|h|\}_K$ as an unsolicited test if there is a node n is in the strand i and is an unsolicited test for $\{|h|\}_K$ in a bundle $\mathcal{B}$. Because a guarantee of the existence of a regular node can be drawn from an unsolicited test, a regular agent uses unsolicited test to authenticate its regular protocol participant in Kerberos V.

The client strand in the authentication phase receives $\{|A, Tgs, authK, T_a|\}_{K_A}$ as an unsolicited test that authenticates the positive node of the authentication server strand. The intuition behind this authentication is quite straightforward. By case analysis on the form of i, we have $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(i,1)$, combining with the assumption that A is not compromised, by Lemma 4, we have $\{|A, Tgs, authK, T_a|\}_{K_A}$ can only be originated by an authentication server. For the sake of brevity, in the following discussion we use $P[x, *, \ldots, y]$ to denote $\exists x'.P[x, x', \ldots, y]$. $\mathcal{B}$ is a bundle of Kerberos V.

Lemma 10 *Let* $A \notin \mathbf{Bad}$. *If* i *is a client strand in the authentication phase such that* $\boldsymbol{Ag\text{-}I}[i, A, Tgs, authK, T_a, authTicket, t_0, t_1]$ and $(i,1) \in \mathcal{B}$, *then there exists an authentication server strand as such that* $AS[as, A, Tgs, authK, *, T_a]$, $(as, 1) \in \mathcal{B}$.

The ticket grant server strand uses $\{|A, T_0|\}_{authK}$ as an unsolicited test to authenticate the client strand in the authorization phase. This guarantee is ensured from the secrecy of $authK$, which is in turn guaranteed by the ticket $\{|A, B, authK, T_a|\}_{K_{Tgs}}$. By the trace specification of a ticket grant server strand, we have that

$$\left\{\left|\left\{\left|A, B, authK, T_a\right|\right\}_{K_{Tgs}}, \left\{\left|A, T_0\right|\right\}_{authK}\right|\right\}$$

is received by Tgs earlier than the time

$$T_a + \text{sessionKcrktime},$$

by Lemma 5, $authK$ is safe at that time.

Lemma 11 *Let* $A \notin \mathbf{Bad}$, $Tgs \in \mathbf{TGSs}$. *If* tgs *is a ticket grant server strand such that* **TGS**$[tgs, A, Tgs, authK, servK, B, T_a, T_0, t_0, t_1]$, *and* $(tgs,0) \in \mathcal{B}$, *then there exists a client strand* i *in the authorization phase such that* $(i,0) \in \mathcal{B}$ *and* ***Ag-II***$[i, A, authK, *, *, *, *, *, *, T_0, *]$.

Proof. By analysis on the form of tgs strand, we have (1) $\{|A,T_0|\}_{authK} \sqsubset (tgs,0)$, (2) $\{|A,B,authK,T_a|\}_{K_{Tgs}} \sqsubset term(tgs,0)$ and $time(tgs,0) \le T_a$ + shrKcrackt ime. From (2), by Lemma 5, we have that $term(m) \neq authK$ for any $m \in \mathcal{B}$ such that $time(m) \le time(tgs,0)$, therefore $regular(authK,(tgs,0),\mathcal{B})$. With (1), by Lemma 1, we have (3) there is a positive regular node m such that $m \preceq_{\mathcal{B}} (tgs,0)$ and $\{|A,T_0|\}_{authK} \sqsubset term(n)$ and $\{|A,T_0|\}_{authK} \not\sqsubset term(m')$ for any node m' such that $m' \preceq_{\mathcal{B}} m$. Obviously $m \in \mathcal{B}$.

Now we need prove that m must be in a client strand i in the authorization phase. From the fact that m is regular, then we have six cases, here we select two cases when (4) m is in a strand i such that **Ag-II**$[i, A', authK', authTicket', B', servK', T_S', servTicket', t_2, t_3]$ or (5) m is in a strand i such that **Ag_III**$[i, A', servK', servTicket', t_4, t_5]$. Other cases are more simpler.

If (4) holds, then $m = (i,0)$ because m is positive. From $\{|A,T_0|\}_{authK} \sqsubset term(n)$ and

$$term(m) = \left|\left\{authTicket', \{|A,t_2|\}_{authK}, B\right\}\right|,$$

we have either (6) $\{|A,T_0|\}_{authK} \sqsubset authTicket'$ or (7)

$$\{|A,T_0|\}_{authK} \sqsubset \{|A',t_2|\}_{authK'}.$$

If (6) holds, then by the definition of the client strand, there exists some client strand i_1 such that $i_1 \mapsto i$ and **Ag-I**$[i_1, A', Tgs', authK', T_a', authTicket', t_0', t_1']$. From the definition of the strand, we have

$$term(i_1,1) = \left\{\left|authTicket', \left\{\left|A', Tgs', authK', T_a'\right|\right\}_{K_{A'}}\right|\right\}.$$

From this and (6), we have (8) $\{|A,T_0|\}_{authK} \sqsubset term(i_1,1)$. From $i_1 \mapsto i$, $(i_1,1) \Rightarrow (i,0)$, then (9) $(i_1,1) \preceq_{\mathcal{B}} (i,0)$. But (8) and (9) contradicts with (3). If (7) holds, then $A = A'$, $T_0 = t_2$, $authK = authK'$. So the conclusion is obtained.

If (5) holds, then similar to the counterpart of the argument for case (4), we have either (10) $\{|A,T_0|\}_{authK} \sqsubset servTicket'$ or (11) $\{|A,T_0|\}_{authK} \sqsubset \{|A',t_4|\}_{servK'}$. For case (10), its proof is similar to that of case (6). If (11) holds, then (12) $A = A'$, $T_0 = t_4$, $authK = servK'$. By the definition of **Ag_III**, (13) there is a client strand i_2, i_1 such that $i_1 \mapsto i_2$ and $i_2 \mapsto i$ and **Ag-II**$[i_2, A', authK', authTicket', B', servK', T_S', servTicket', t_2, t_3]$ and **Ag-I** $[i_1, A', Tgs', authK', T_a', authTicket', t_0', t_1']$ for some $Tgs' \in$ **TGSs**. Obviously,

$$\{|A', Tgs', authK', T_a|\}_{K_{A'}} \sqsubset term(i_1,1),$$

$$\{|A', B', servK', T_S'|\}_{authK'} \sqsubset term(i_2,1),$$

and $time(i_2,1) \le T_a' + \text{shrKcracktime}$. From $i_1 \mapsto i_2$ and $i_2 \mapsto i$ and $(i,0) \in \mathcal{B}$, we have $(i_1,1) \in \mathcal{B}$ and $(i_2,1) \in \mathcal{B}$. From (12), and the assumption $A \notin$ **Bad,** by Lemma 6, (14) there is a ticket granting server strand tgs' such that **TGS**$[tgs', A', Tgs', authK', servK', B', T_a', T_0', t_0'', T_S']$ for some T_0', t_0''. But from (2), by Lemma 4, we have (15) there is an authentication server as such that $AS[as, A, Tgs, authK, t_0', T_a]$ for some t_0'. But from (12) and Axioms 4,5, we have $tgs' = as$ because $authK$ ($servK'$) uniquely originates from a strand, obviously this is a contradiction.

A client strand i_2 in the authorization phase receives $\{|A,B,servK,T_S|\}_{authK}$ as an unsolicited test. Note that $\{|A,B,servK,T_S|\}_{authK}$ is received in the second node in the client strand; furthermore, from the definition of **Ag-II**, we have that there exists a client strand i_1 in the authentication phase such that $i_1 \mapsto i_2$, and the ticket $\{|A,Tgs,authK,T_a|\}_{K_A}$ must be received at the second node of i_1; from the definition of **Ag-II**, $\{|A,B,servK,T_S|\}_{authK}$ must have been received at an earlier time than T_a + shrKcracktime, then by Lemma 5, it can be guaranteed that $authK$ must be safe at the time when the client strand receives $\{|A,B,servK,T_S|\}_{authK}$.

Lemma 12 *Let* $A, B \notin$ **Bad**. *If* i *is a client strand in the authorization phase such that* ***Ag-II*** $[i, A, authK, T_a, auth\text{-}Ticket, B, servK, T_S, servTicket, t_0, t_1]$ and $(i,1) \in \mathcal{B}$, *then there exists a client strand* i_0 *in the authentication phase, and a ticket grant server strand* tgs, *and some* Tgs *such that* $i_0 \mapsto i$ *and* ***Ag-I***$[i_0, A, Tgs, authK, T_a, authTicket, *, *]$ *and* **TGS**$[tgs, A, Tgs, authK, servK, B, T_a, *, *, T_s]$, *and* $(tgs,1) \in \mathcal{B}$, *and* $B \notin$ **TGSs**.

The application server B receives $\{|A,T_4|\}_{servK}$, which is an unsolicited test to guarantee that the first received message must be from a client strand in the service phase. This guarantee is ensured from the secrecy of $servK$, which is in turn guaranteed by the ticket $\{|A,B,servK,T_S|\}_{K_B}$. By the trace specification of an application server strand, we have that

$$\left\{\left|\left\{\left|A, B, servK, T_S\right|\right\}_{K_B}, \left\{\left|A, T\right|\right\}_{4servK}\right|\right\}$$

is received by B earlier than the time

$$T_S + \text{sessionKcrktime}.$$

By Lemma 9, $servK$ is safe at that time.

Lemma 13 *Let* $A, B \notin \mathbf{Bad}$, $B \notin \mathbf{TGSs}$. *If* b *is an application server strand such that* $\mathbf{Apps}[b, A, B, servK, T_S, T_4, t_0, t_1]$, and $(b,0) \in \mathcal{B}$, *then there are two client strands* i_2, i_3 *and some servTicket such that* $i_2 \mapsto i_3$ *and* $(i_3,0) \in \mathcal{B}$ *and* $\mathbf{Ag\_II}[i_2, A, *, *, B, servK, T_S, servTicket, *, *]$ *and* $\mathbf{Ag\_III}[i_3, A, servK, servTicket, T_4, *]$.

The client strand in the service phase uses $\{|T_4|\}_{servK}$ as an unsolicited test to authenticate the application server strand. This guarantee is also ensured from the secrecy of $servK$, which is in turn guaranteed by the ticket $\{|A, B, servK, T_S|\}_{authK}$, and $\{|A, Tgs, authK, T_a|\}_{K_A}$. By the trace specification of an application server strand, we have that $\{|T_4|\}_{servK}$ is received by A earlier than the time $T_S + \text{sessionKcrktime}$ and $T_a + \text{shrKcracktime}$. By Lemma 7, $servK$ is safe at that time.

Lemma 14 *Let* $A, B \notin \mathbf{Bad}$, $B \notin \mathbf{TGSs}$. *If* i_3 *is a client strand in a service phase such that* $\mathbf{Agent\_III}[i_3, A, servK, servTicket, t_4, *]$, *and* $(i_3,1) \in \mathcal{B}$, *and* i_2 *is a client strand in the authorization phase such that* $\mathbf{Agent\_II}[i_2, A, authK, authTicket, B, servK, T_S, servTicket, t_2, t_3]$ *and* $i_2 \mapsto i_3$, *then there exists an application server strand* b *such that* $\mathbf{Apps}[b, A, B, servK, T_S, t_4, ,*,*]$ *and* $(b,1) \in \mathcal{B}$.

Proof. By analysis on the form of strand i_3 and i_2, we have (1) $term(i_3,1) = \{|t_4|\}_{servK}$ and (2) $\{|A, B, servK, Ts|\}_{authK} \sqsubset term(i_2,1)$. By unfolding the definition of **Agent_II,** there exists a client strand i_1 such that $i_1 \mapsto i_2$ and **Ag-I** $[i_1, A, Tgs, authK, T_a, authTicket, t_0, t_1]$ and $Tgs \in \mathbf{TGSs}$ for some Tgs, T_a, t_0, t_1. Obviously, (3) $\{|A, Tgs, authK, T_a|\}_{K_A} \sqsubset term(i_1,1)$, $time(i_1,1) \leq T_a$ +shrKcracktime. From $i_1 \mapsto i_2$ and $i_2 \mapsto i_3$, we can easily conclude that $(i_1,1) \Rightarrow^+ (i_3,1)$ and $(i_2,1) \Rightarrow^+ (i_3,1)$. With $(i_3,1) \in \mathcal{B}$, we have $(i_2,1) \in \mathcal{B}$ and $(i_1,1) \in \mathcal{B}$. With (2)(3), by Lemma 7, (4) $term(m) \neq servK$ for any node m such that $time(m) \leq T_S$ +sessionKcrktime. By the definition of **Agent_III** we have (5) $time(i_3,1) \leq T_S$ +sessionKcrktime. From (4)(5), we have $term(m) \neq servK$ for any node m such that $time(m) \leq time(i_3,1)$, therefore *regular* $(servK, (i_3,1), \mathcal{B})$. So $(i_3,1)$ is an unsolicited test for $\{|T_4|\}_{servK}$ in $\mathcal{B}$. By Lemma 1, (5) there is a regular positive node m such that $m \preceq_{\mathcal{B}} (i_3,1)$ and $\{|t_4|\}_{servK} \sqsubset term(m)$ and $\{|t_4|\}_{authK} \not\sqsubset term(m')$ for any node m' such that $m' \preceq_{\mathcal{B}} m$. Obviously $m \in \mathcal{B}$. By simple case analysis, we have that m must be in an application server b such that $\mathbf{Apps}[b, A', B', servK', T_S', t_4', *, *]$, $m = (b,1)$. By the definition of **Apps**, $term(b,1) = t'_{4servK'}$. With $t_{4servK} \sqsubset term(b,1)$, we have (6) $servK' = servK$ and $t_4' = t_4$.

$$\text{Let } M =_{df} \left\{ \begin{array}{l} \{|A, B, servK, T_S|\}_{K_B}, \\ \{|A, B, servK, T_S|\}_{authK} \end{array} \right\} \cup \{t \mid servK \not\sqsubset t\}.$$

Obviously $time(b,0) \preceq_{\mathcal{B}} time(b,1) \preceq_{\mathcal{B}} time(i_3,1) \leq T_S$ + sessionKcrktime, by the proof of Lemma 7, we have $term(b,0) \in synth(M)$, *i.e.*,

$$\left\{\left| \{|A', B', servK', T_S'|\}_{K'_{B'}}, \{|t_4'|\}_{servK'} \right|\right\} \in synth(M).$$

By the definition of $synth$, we have

$$\{|A', B', servK', T_S'|\}_{K'_{B'}} \in synth(M),$$

then we have (7) $\{|A', B', servK', T_S'|\}_{K'_{B'}} \in M$ or (8) $\{|A', B', servK', T_S'|\} \in synth(M)$. If (7) holds, from (6) (7) and the definition of M, we have

$$\{|A, B, servK, T_S|\} = \{|A', B', servK', T_S'|\},$$

then $A = A'$ and $B = B'$ and $T_S = T_S'$. Therefore, the conclusion holds. If (8) holds, by the definition of *synth,* we have $servK' \in synth(M)$, with (6), we have $servK \in synth(M)$, then by the definition of *synth,* we have $servK \in M$, but this contradicts with the definition of M.

Roughly speaking, we need two steps to prove an authentication goal that if there is a regular responder strand $Resp\,(r, \vec{x})$ and the k-th node of the strand is in a bundle $\mathcal{B}$, then there is an initiator strand $Init\,(i, \vec{x})$ and some j-th node of the initiator strand is in $\mathcal{B}$. First we prove that (r,k) is an unsolicited test for some encrypted term $\{|h|\}_K$ in $\mathcal{B}$, which requires the secrecy of K. This is can be easily proved by the secrecy results on keys in section 3. Therefore, we have that there exists some regular node m in $\mathcal{B}$ by Lemma 2. Second, we need prove that m indeed is the intended node (i, j). In order to prove this, we need do case analysis on the form of the strand which m possibly lies in. This proof needs unicity property of some session keys and the results of unsolicted tests, namely, the facts that $\{|h|\}_K \sqsubset term(m)$ and m is minimal.

6. Conclusions and Related Work

Our main aim is to extend and mechanize the strand space theory to analyze Kerberos V, since mechanization in a theorem prover not only helps us model protocols rigorously and specify protocol goals without any ambiguity, it also guarantees a formal proof. Besides the essential inherence from the classic strand space method, our work is deeply inspired by Paulson and Bella’s work. We have directly used their formalization of message algebra, and have learned a lot about the semantics of timestamps and replay attacks from [4]. However, we model and analyze protocols in strand space theory rather than in Paulson's trace theory. In detail, we model behaviors of all the agents by strands, and mainly use the well-founded induction principle to prove properties. So in our Isabelle formalization, main efforts have been devoted to definitions and lemmas about strand space theory. e.g., we formalize strands, bundles, unique originality, the well-founded principle on bundles, and use this principle to prove important results such as unsolicited authentication test and regularity of keys.

In [4], the ability of a penetrator to crack a stale encrypted message is modelled by the Oops rule in the inductive definition of a trace, and the trace definition depends on the protocol under study. However, in the strand space theory, a penetrator’s abilities are modelled to be independent of the protocol, that is the main reason why we relate a key with a crack time, and model a penetrator’s ability of cracking a stale encrypted message by a new key cracking strand. The advantage of our method is that modelling a penetrator’s behavior remains independent and results such as the unsolicited authentication tests can be generalized.

Regarding verification of the Kerberos protocols, Mitchell *et al.* [7] analyzed a simplified version of the protocol by model checking, and Butler *et al.* [8] analyzed the Kerberos V protocol using MSR [9]. But they did not include timestamps and replay attacks in their model, in fact the former work ignored both nonces and timestamps, and the latter only considered the implementation of the Kerberos protocol basing on nonce.

7. References

[1] S. P. Miller, J. I. Neuman, J. I. Schiller and J. H. Saltzer, “Kerberos Authentication and Authorisation System,” Technical Report, Technical Plan Section E.2.1, MIT, Athena, 1989.

[2] K. R. C. Neuman and S. Hartman, “The Kerberos Network Authentication Service (v5),” Technical report, Internet RFC 4120, July 2005.

[3] Y. L. and J. Pang, “Extending the Strand Space Method with Timestamps: Part I the Theory,” *Journal of Information Security*, Vol. 1, No. 2, 2010, pp. 45-55.

[4] G. Bella, “Inductive Verification of Cryptographic Protocols,” PhD Thesis, Cambridge University Computer Laboratory, 2000.

[5] F. Javier Thayer, J. C. Herzog and J. D. Guttman, “Strand Spaces: Proving Security Protocols Correct,” *Journal of Computer Security*, Vol. 7, No. 1, 1999, pp. 191-230.

[6] Y. Li, “Strand Space and Security Protocols”. http://lcs.ios.ac.cn/~lyj238/strand.html.

[7] J. C. Mitchell, M. Mitchell and U. Stern, “Automated Analysis of Cryptographic Protocols Using Murphi,” *Proceedings of* 18*th Symposium on Security and Privacy*, 1997, pp. 141-153.

[8] L. Bozga, C. Ene and Y. Lakhnech, “A Symbolic Decision Procedure for Cryptographic Protocols with Time Stamps,” *Journal of Logic and Algebraic Programming*, Vol. 65, No. 1, 2005, pp. 1-35.

[9] F. Butler, I. Cervesato, A. Jaggard and A. Scedrov, “A formal Analysis of Some Properties of Kerberos 5 Using MSR,” *Proceedings of* 15*th IEEE Computer Security Foundations Workshop*, 2002, pp. 175-190.