An Authentication Logic with Formal Semantics Supporting Synchronization, Revocation, and Recency*

Stuart G. Stubblebine
CertCo
55 Broad St. Suite 22
New York, NY 10004
stuart@stubblebine.com

Rebecca N. Wright
AT&T Labs – Research
180 Park Avenue
Florham Park, NJ 07932
rwright@research.att.com

Abstract

Distributed systems inherently involve dynamic changes to the value of security-relevant attributes such as the goodness of encryption keys, trustworthiness of participants, and synchronization between principals. Since concurrent knowledge is usually infeasible or impractical, it is often necessary for the participants of distributed protocols to determine and act on beliefs that may not be supported by the current state of the system. Policies for determining beliefs in such situations can range from extremely conservative, such as only believing statements if they are very recent, to extremely optimistic, such as believing all statements that are not yet known to be revoked. Such security policies often are heavily dependent on timing of received messages and on synchronization between principals.

We present a logic for analyzing cryptographic protocols that has the capability to specify time and synchronization details. This capability considerably advances the scope of known techniques both for expressing practical authentication policies of protocol participants as constraints, and for reasoning about protocol goals subject to these constraints. In the course of reasoning about protocol goals, one is able to exhibit sufficient conditions regarding trust between protocol participants, synchronization between protocol participants, and timeliness of message contents.

Our logic is flexible, and can support a wide range of security policies including the recently proposed "recent-secure authentication" policy for enforcing revocation in distributed systems. The ability to reason about the conjunction of individual participant policies and protocols will be especially important as public and private key infrastructures are deployed and new and unanticipated policies are put into use.

1 Introduction

In distributed systems, the lack of physical protection on a communication path means that communication is susceptible to attacks involving eavesdropping, modification, and impersonation. Protocols using cryptographic mechanisms such as encryption and digital signatures form the basis of security in distributed systems by providing protection against such attacks. Unfortunately, cryptographic protocol design has historically been error prone as evidenced by numerous vulnerabilities cited in the literature. However, analysis and design techniques have proven useful in detecting protocol vulnerabilities.

We present a logic for analyzing cryptographic protocols with a capability to specify time and synchronization details. This capability provides new techniques for expressing practical authentication policies of protocol participants and for reasoning about protocol goals subject to these policies. By using our logic to reason about properties of synchronization, recency, and revocation, one can prove protocol goals that are beyond the scope of existing analysis techniques. Using our logic, one can prove protocol goals such as the presence of participants at particular times, the goodness of keys at particular times, and beliefs of participants at particular times. More specifically, goals might involve the long-term evolution of beliefs in

*This work was done primarily while the first author was at AT&T. A preliminary version of this work appeared in [SW96].

In *IEEE Trans. on Software Engineering*, Vol. 28, No. 3, 2002, pp. 256–285.

practical systems characterized by dynamic changes to the goodness of keys, jurisdiction, authorization, and synchronization, the correctness of replay detection mechanisms using timestamps, or the time of presence and continued presence of remote principals.

1.1 Related Work

In their seminal paper [BAN89], Burrows, Abadi, and Needham introduced a logic for reasoning about authentication protocols. Considerable effort has been spent on understanding, extending, and improving the use of logics for cryptographic protocols [GNY90, AT91, GS91, Syv93, SvO96]. A number of formal semantics have only been proposed for some of these logics [AT91, WK96, SvO96, BM97]. However, the formal semantics is important due to the numerous and subtle yet important distinctions in their meanings. Related research has been performed in calculi for access control and authentication in distributed systems [LABW92, ABLP93, Stu95].

In the BAN logic and its successors, time is divided into two epochs, the past and the present. The past is all time before the current run of the protocol being considered; the present begins at the start of the current run. Since all beliefs held in the present are assumed to be stable during the current run of the protocol, principals are not allowed to change their beliefs during the run. In particular, in their logic, if a principal P believes that K is a good key at some point in the run of a protocol, then P must believe that K is a good key for the duration of the protocol.

Our underlying logic is based on the BAN logic, as reformulated by Abadi and Tuttle [AT91] . Unlike these logics, we qualify all statements with time to allow arbitrary revocation of keys, beliefs, and jurisdiction. We use a notation and formalism that is similar to the BAN-related logics to highlight the parallels between our logic and the BAN-related logics. Our formal semantics relating to time was inspired by the work of Moses and Bloom [MB95], and Fagin et. al. [FHMV95].

The work of Gaarder and Snekkenes [GS91] extends BAN logics by introducing the notion that a participant claims a formula is, was, or will be good in a time interval and the notation that the local clock of a participant has a time within an interval. The work makes the strong assumption that the duration of a protocol run must be short, and that participants commit themselves to believing that messages are correct for the duration of the protocol. The logic has the limiting characteristic of other BAN-like logics that beliefs may not be revoked at an arbitrary time. The assumptions used are too restrictive for reasoning about long-lived security associations. Furthermore, reasoning about synchronization and synchronization bounds is beyond the scope of their work and the logic is without a formal semantics. Our work removes these restrictions by allowing arbitrarily long protocol durations in which participants beliefs may change over time.

The work of Syverson on adding time to a logic of authentication [Syv93], although related, has a different focus than our work. His work incorporates temporal formalism into the semantics model of BAN logic [AT91] using temporal notions of "all points in the run prior to the current one," and "at some point in the run prior to the current one." Our logic, on the other hand, allows for arbitrary granularity of time.

Our work was directly motivated by earlier work of Stubblebine [Stu95]. His work introduced the notion of *recent-secure authentication*. Recent-secure authentication characterizes an authentication policy based on the recency of security information (such as keys) used in authenticating channel or message. In particular, it requires that at the time of authentication, verifiers must have received a recent account attesting to the accuracy of security-relevant information used for authentication. Authentication according to recent-secure policies ensures that revocation is fail-safe. That is, if security information is revoked at time t then no verifier will authenticate a message using revoked credentials after time $t + \delta$ where δ is the recency interval. Existing logics were insufficiently expressive to analyze recent-secure authentication policies and protocols using them. By adding recency and revocation to these logics, our work provides a formal framework for stating and analyzing recent-secure authentication policies and protocols.

1.2 Our Work

As stated above, our underlying logic is based on the BAN logic of authentication. The BAN logic of authentication describes beliefs of trustworthy parties involved in authentication, and the evolution of these beliefs while the principals communicate. This approach has had success explaining protocols and uncovering hidden assumptions and dangerous bugs. The BAN logic has unearthed many protocol flaws and is attributed with being a very cost-effective means of detecting (some) flaws [CJ97].

We chose to base our logic on the BAN logic for a number of reasons. Since timing issues can themselves be intuitively complex, we felt it was crucial to use the simplest possible starting point. At the time of our initial study [SW96], BAN was one of the most widely studied analysis methods for cryptographic protocols. Even as later methods such as the spi calculus [AG99] have been developed, BAN has been recognized to be simpler and more intuitive than other frameworks. Another important aspect of the BAN approach is that it forces one to be precise about the protocol assumptions and goals. Assumptions and goals concerning synchronization, revocation, and recency have been traditionally hidden. Also, assumptions and goals vary widely even with different instances of closely related protocols. Thus we believe that it is reasonable to take BAN as a starting point, and this is the approach we take in this paper. We believe our work demonstrates, in an intuitive way, the importance of considering security issues of synchronization, revocation, and recency in security protocol analysis. Building on other analysis methods such as the spi calculus may be a worthwhile endeavor for future research.

The main difference between our logic and the BAN logic and its successors is that we allow for the possibility of revocation of keys, jurisdictions, and more generally, of arbitrary beliefs. That is, we allow for a principal to hold a belief at one point in the run of a protocol, and not to hold that belief later in the run. We are also much more strict about the extension of beliefs from a given point in time to a later point. In particular, a belief at a given time can only be extended to a later time according to the policies that are specified as part of the assumptions of the protocol.

Another significant difference between our logic and the BAN logic is our use of timestamps. BAN treats all timestamps as nonces. As such, the important property of a timestamp is its freshness. However, there are cases where it is the *recentness*, rather than the freshness, of a timestamp that is important. This is the case, for example, with a Kerberos ticket [NT94] that may have been sent before, so is not fresh, but is still valid because it was created recently enough. Our logic explicitly captures the time of the timestamp and uses this time to reason about recentness.

In the BAN-like logics, the response to a nonce-based challenge is idealized as the same nonce as in the challenge. We feel that this is not the proper way to handle challenge-response protocols in our logic, since in particular, it is the freshness of both the challenge and the response that allows the recipient to determine the presence of the responder. In using our logic, the response should be idealized as it is said, allowing the differentiation between the freshness of the challenge and the freshness of the response. As an example of the usefulness of this approach, consider a protocol in which P issues a challenge X to Q at time t_1, Q responds with $X + 1$ encrypted with a shared key K at time t_2, and then Q sends the same response again at t_3. Our logic allows P to determine Q's presence between t_1 and t_2 because of the freshness of Q's response at time t_2, but does not allow P to determine Q's presence between t_2 and t_3, since the message at t_3 may have been a replay. This example illustrates how our logic allows for different beliefs to be concluded at different times during the run of a protocol.

We do not assume clock synchronization. Our logic allows for reasoning about trusted synchronization paths and synchronization bounds between protocol participants. In particular, we have added a synchronization construct to allow explicit assumptions about clock synchronization to be made where needed. Consequently, analysis makes explicit the impact that derivation of trustworthy synchronization paths and synchronization bounds have on achieving protocol goals.

Although the addition of time into our logic yields a somewhat more complicated logic, we feel the additional complexity is justified due to the considerable gains in being able to specify and analyze important properties of systems and policies found in practice. Also, we feel that the analysis technique can easily be reduced to implementation, thereby removing much of the complexity of analysis.

We define our logic in Section 2. In Section 3, we present examples of analysis using the logic. In two appendices, we present a model of computation and show that our logic is sound with respect to that model: in Appendix A, we define our model of computation and give a formal semantics for our logic; in Appendix B, we prove the soundness of our logic with respect to the model of computation.

2 The Logic

Before proceeding to a formal definition of our logic, we first give a brief overview of the logic. The logic consists of formulas that make statements about the knowledge and beliefs of the principals involved. We make statements such as P *believes*$_t$ X (P believes X holds at time t) and P *received*$_t$ X (P received X at time t), about messages such as $\{X\}_K$ (X encrypted by K) and $\lfloor X \rfloor_{K^{-1}}$ (X signed by K^{-1}). We give a simple logic and model, in which we assume each principal has a local concept of time; however, there are no inherent assumptions of clock synchronization.

In our basic formulas, a time refers to the local time of the principal in the formula. Where there is no principal, or more than one principal, involved, we explicitly state the principal whose time is being referenced. We use $[t_1, t_2]$ to indicate that a formula holds at all times between t_1 and t_2, and we use $\langle t_1, t_2 \rangle$ to indicate that a formula holds at some time between t_1 and t_2.

Our logic makes some simplifying assumptions similar to those made by related logics (cf. [BAN89]). We do not attempt to answer every important question about cryptographic protocols. Our goal is not to reason about implementation details such as the correct use of cryptographic functions. Principals are trusted to communicate honestly; hence, senders believe the truth of message contents. Also, principals can be trusted to keep secrets. In particular, although their keys may become compromised and thereby allow others to decrypt encrypted messages, principals are assumed not to pass on the decrypted contents themselves unless specified by the protocol. Keys are assumed to be adequate for their purposes and cryptosystems are assumed to be secure and properly used. For example, we assume that it is only possible for a principal to create the value $\{X\}_K$ that is the encryption of X with key K if the principal has K and knows X, and conversely, if a principal attempts to decrypt $\{X\}_K$ with the wrong key, the result will not be a message. In practice, this is easily achievable by formatting messages according to a specified format before encrypting them. Principals can recognize their own encrypted messages, and will not mistakenly attribute them to another principal.

We separate the protocol itself from security policies that specify, for example, the methods principals use for determining beliefs from previously received messages, and handling of revocation. Policies can range from conservative, such as only believing statements if they are very recent, to optimistic, such as believing all statements that are not yet known to be revoked. The policies are external to the "core" logic; specific policies determined by a protocol analyzer should be used in specific situations.

Finally, we also make certain assumptions about the use of our logic. In particular, we assume that users of our logic properly formalize protocols and policies and make "reasonable" assumptions when carrying out protocol analysis. For example, assumptions should not contradict each other, and should reflect the real world in which the correctness of the protocol is being claimed.

In Section 2.1, we formally define the set of messages and formulas in the logic. In Section 2.2, we define the axioms that allow us to derive one formula from another. Section 2.3 discusses belief-extension policies. Section 2.4 explains how the logic is used to analyze protocols.

2.1 Messages and formulas

Underlying the set of possible messages is the set $\mathcal{T}$ of *primitive terms*. $\mathcal{T}$ contains the natural numbers and several distinguished subsets of constant symbols called: *principals, public keys, private keys, shared keys, data constants* (such as nonces), and *primitive propositions*. Together, the set of public keys, private keys, and shared keys, are called *keys*. Each key has an *inverse*, which is also a key. Given a key K, we write K^{-1} to denote its inverse. If K is a public key, then K^{-1} is a private key. If K is a private key, then K^{-1} is a public key. If K is a shared key, then K^{-1} is a shared key and $K^{-1} = K$. As a notational convention, we typically use K^{-1} to denote a private key, even if the corresponding public key K has not explicitly been mentioned. We assume that the set of times is ordered. For simplicity, we represent times in our logic by natural numbers, though it would be possible to define times using any metric space.

Some messages have truth values while others, such as times, principal's names and nonces, do not. Messages that have truth values are called *formulas*. Messages and formulas in the logic are defined by mutual induction. Given the set $\mathcal{T}$ of primitive terms, we recursively define the set of messages, $\mathcal{M}_\mathcal{T}$ as follows.

M1. φ is a message if φ is a formula.

M2. X is a message if $X \in \mathcal{T}$.

M3. $F(X_1, \ldots, X_n)$ is a message if $X_1, \ldots, X_n \in \mathcal{T}$ and F is an n-ary function.

We assume in M3 that there are distinguished functions $\{X\}_K$, $\lfloor X \rfloor_{K^{-1}}$, and $(X_1, \ldots, X_n)$ that represent encryption, digital signatures, and concatenation, respectively. We will also use functions to represent constraints on keys $(\langle P, K, [t_b, t_e] \rangle)$; this usage will be illustrated in an example in Section 3.

The set of formulas, $\mathcal{F}_{\mathcal{T}}$, also recursively defined, is defined below. For each formula, we also provide an intuitive description of the intended meaning of the formula. Note, however, that the intuitive meanings are given only to help the reader. The real meaning of the formulas in the logic are given only implicitly by the behavior assigned to them by the axioms of Section 2.2. In Appendix A, we provide a model of computation and define the semantics of all formulas in that model of computation. The soundness of our logic shown in Appendix B demonstrates that the meanings given implicitly by the logic and the meanings given explicitly by the model are consistent with each other.

We specify formulas with three types of time: those that apply at a single time t; those that apply over a universal time interval $[t_1, t_2]$, meaning that the formula holds at all times between t_1 and t_2; and those that apply over an existential time interval $\langle t_1, t_2 \rangle$, meaning that the formula holds at some time between t_1 and t_2. If φ and ψ are formulas, t, t_1 and t_2 are natural numbers, P, Q, and R are principals, p is a primitive proposition, K is a key, and ϵ is a natural number, then the following are formulas.

F1. $p\ \mathit{at}_Q\ t$, $p\ \mathit{at}_Q\ [t_1, t_2]$, and $p\ \mathit{at}_Q\ \langle t_1, t_2 \rangle$:

The primitive proposition is true at time t on Q's clock, at all times between t_1 and t_2 on Q's clock, or during at least one time between t_1 and t_2 on Q's clock, respectively.

F2. $\neg\varphi$ and $\varphi \wedge \psi$

Respectively, φ is not true, and both φ and ψ are true. For simplicity, we have defined only the logical connectives $\neg$ and $\wedge$. Of course, other logical connectives such as $\vee$ and $\supset$ can be defined in the standard way from these.

F3. $t_1 \leq t_2$ and $t_1 \geq t_2$

The usual relations between natural numbers.

F4. $P\ \mathit{believes}_{t,Q}\ \varphi$, $P\ \mathit{believes}_{[t_1,t_2],Q}\ \varphi$, and $P\ \mathit{believes}_{\langle t_1,t_2 \rangle,Q}\ \varphi$

P believes that φ is true at time t on Q's clock (respectively at some or all times between t_1 and t_2 on Q's clock).

The remaining formulas follow the same interpretation of universal and existential time intervals: i.e. that the corresponding formula with a single time t holds at all times between t_1 and t_2 on S's clock, denoted $[t_1, t_2], S$, or at some time between t_1 and t_2 on a principal S's clock, denoted $\langle t_1, t_2 \rangle, S$. We continue to provide the intuitive interpretation of each formula for the single-time version, but omit the repetition of the interpretation of corresponding universal and existential versions.

F5. $P\ \mathit{controls}_{t,Q}\ \varphi$, $P\ \mathit{controls}_{[t_1,t_2],Q}\ \varphi$, and $P\ \mathit{controls}_{\langle t_1,t_2 \rangle,Q}\ \varphi$

P can be trusted as an authority on φ at time t on Q's clock. That is, if P says φ is true at this time, then it is. In this case, we say P has control over φ, or P has jurisdiction over φ.

F6. $P\ \mathit{received}_{t,Q}\ X$, $P\ \mathit{received}_{[t_1,t_2],Q}\ X$, and $P\ \mathit{received}_{\langle t_1,t_2 \rangle,Q}\ X$

P received the message X at or before time t on Q's clock. Sent and received messages include both the actual messages that were sent or received as part of a protocol and their contents.

F7. $P\ \mathit{says}_{t,Q}\ X$, $P\ \mathit{says}_{[t_1,t_2],Q}\ X$, and $P\ \mathit{says}_{\langle t_1,t_2 \rangle,Q}\ X$

P sent the message X (precisely) at time t on Q's clock.

F8. P $\mathit{said}_{t,Q}$ X, P $\mathit{said}_{[t_1,t_2],Q}$ X, and P $\mathit{said}_{\langle t_1,t_2\rangle,Q}$ X

P sent the message X at or before time t on Q's clock.

F9. $P \overset{K}{\leftrightarrow}_{t,R} Q$, $P \overset{K}{\leftrightarrow}_{[t_1,t_2],R} Q$, and $P \overset{K}{\leftrightarrow}_{\langle t_1,t_2\rangle,R} Q$

K is a good shared key for P and Q at time t on R's clock. That is, P and Q are the only principals who use K to encrypt messages.

F10. $\overset{K}{\Rightarrow}_{t,Q} P$, $\overset{K}{\Rightarrow}_{[t_1,t_2],Q} P$, and $\overset{K}{\Rightarrow}_{\langle t_1,t_2\rangle,Q} P$

K is a key that can be used to verify signatures from P at time t on Q's clock. That is, there is a corresponding key K^{-1} that only P use to sign messages.

Since we believe one of the useful features of a logic is that it requires its users to be explicit about assumptions being made, we distinguish between signature keys and encryption keys (though in some protocols or some implementations the same key may be both an encryption key and a signature key).

F11. $\overset{K}{\mapsto}_{t,Q} P$, $\overset{K}{\mapsto}_{[t_1,t_2],Q} P$, and $\overset{K}{\mapsto}_{\langle t_1,t_2\rangle,Q} P$

K is a key that only P can use to encrypt messages at time t on Q's clock. Only the corresponding key K^{-1} can be used to decrypt messages encrypted by K.

F12. P $\mathit{has}_{t,Q}$ K, P $\mathit{has}_{[t_1,t_2],Q}$ K, and P $\mathit{has}_{\langle t_1,t_2\rangle,Q}$ K

P is in possession of the key K at time t on Q's clock.

F13. $\mathit{fresh}_{t,P}$ X, $\mathit{fresh}_{[t_1,t_2],P}$ X, and $\mathit{fresh}_{\langle t_1,t_2\rangle,P}$ X

X is fresh at time t if it has not been said at or before time t (during the run of a protocol).

F14. $\epsilon\text{-}\mathit{sync}_{t,R}$ (P,Q), $\epsilon\text{-}\mathit{sync}_{[t_1,t_2],R}$ (P,Q), and $\epsilon\text{-}\mathit{sync}_{\langle t_1,t_2\rangle,R}$ (P,Q)

P and Q's clocks are within ϵ of each other at time t on R's clock.

F15. $(\exists t : t_1 \leq t \leq t_2)\varphi$, $(\exists t \geq t')\varphi$, and $(\exists t \leq t')\varphi$

F16. $(\forall t : t_1 \leq t \leq t_2)\varphi$, $(\forall t \geq t')\varphi$, and $(\forall t \leq t')\varphi$

The formula φ is true for some (in the first case) or for every (in the second) value of t between t_1 and t_2.

For notational simplicity, we also use the following shorthand notations, where either x is a time, or x is $[t_1,t_2]$ or $\langle t_1,t_2\rangle$ for times t_1 and t_2.

P $\mathit{believes}_x$ φ for P $\mathit{believes}_{x,P}$ φ	P $\mathit{controls}_x$ φ for P $\mathit{controls}_{x,P}$ φ
P $\mathit{received}_x$ X for P $\mathit{received}_{x,P}$ X	P says_x X for P $\mathit{says}_{x,P}$ X
P said_x X for P $\mathit{said}_{x,P}$ X	$P \overset{K}{\leftrightarrow}_x Q$ for $P \overset{K}{\leftrightarrow}_{x,P} Q$
$\overset{K}{\Rightarrow}_x P$ for $\overset{K}{\Rightarrow}_{x,P} P$	$\overset{K}{\mapsto}_x P$ for $\overset{K}{\mapsto}_{x,P} P$
P has_x K for P $\mathit{has}_{x,P}$ K	$\epsilon\text{-}\mathit{sync}_x$ (P,Q) for $\epsilon\text{-}\mathit{sync}_{x,P}$ (P,Q)

Note that because principals' clocks may not be synchronized, these shorthand formulas are not always symmetric, even when the fully specified formula is. For example, as we shall see, $P \overset{K}{\leftrightarrow}_{t,P} Q$ means the same thing as $Q \overset{K}{\leftrightarrow}_{t,P} P$, but $P \overset{K}{\leftrightarrow}_t Q$ does not mean the same thing as $Q \overset{K}{\leftrightarrow}_t P$, since in the former, t refers to the time when P's clock reads t, while in the latter t refers to the time when Q's clock reads t. These and other properties are fully specified by the axioms in Section 2.2.

2.2 Axioms

Our axiom system includes two inference rules. For any formulas φ and ψ, and any time t,

R1. Modus Ponens: From φ and $\varphi \supset \psi$ infer ψ.

R2. Necessitation: If $\vdash \varphi$, then infer $P\ \mathit{believes}_t\ \varphi$.

The notation $\vdash \varphi$ means that φ is a *theorem*, i.e. it is derivable from the axioms and inference rules alone. Since the $\vdash$ symbol is a metalinguistic symbol, it does not actually appear in any proofs in the logic. We write $\Gamma \vdash \varphi$ if from the set Γ of formulas, it is possible to derive φ using the axioms and inference rules.

We take all tautologies of propositional logic as axioms. In addition, we take as axioms all instances of the following axiom schemas. Following each axiom or group of axioms, we give a brief intuitive explanation.

Belief

A1. Belief and logical connective axioms.

a. $P\ \mathit{believes}_t\ \varphi \wedge P\ \mathit{believes}_t\ (\varphi \supset \psi) \supset P\ \mathit{believes}_t\ \psi$

b. $P\ \mathit{believes}_t\ \varphi \wedge P\ \mathit{believes}_t\ \psi \supset P\ \mathit{believes}_t\ (\varphi \wedge \psi)$

Belief and logical connectives interact in the expected manner.

A2. $P\ \mathit{believes}_t\ \varphi \equiv P\ \mathit{believes}_t\ P\ \mathit{believes}_t\ \varphi$

Multiple levels of belief are equivalent to a single level.

Time and Synchronization The addition of axioms relating to time and synchronization are where are logic differs most substantially from previous logics.

A3. Monotonicity axioms

a. $P\ \mathit{received}_t\ X \wedge t' \geq t \supset P\ \mathit{received}_{t'}\ X$

b. $P\ \mathit{said}_t\ X \wedge t' \geq t \supset P\ \mathit{said}_{t'}\ X$

c. $P\ \mathit{has}_t\ K \wedge t' \geq t \supset P\ \mathit{has}_{t'}\ K$

d. $\mathit{fresh}_{t,P}\ X \wedge t' \leq t \supset \mathit{fresh}_{t',P}\ X$

e. $\epsilon\text{-}\mathit{sync}_t\ (P,Q) \wedge \epsilon' \leq \epsilon \supset \epsilon'\text{-}\mathit{sync}_t\ (P,Q)$

Once a message has been received, it remains received. Similarly, a message that has been said remains said and a key that is had remains had. Freshness has a reverse monotonicity: if a message is fresh at a particular time, it is fresh at any earlier time. Finally, if two clocks are synchronized within some interval, then they are also synchronized in any smaller interval. Note that there are no monotonicity axioms associated with other kinds of statements such as *believes*, *controls*, and goodness of keys. This reflects the fact that these statements may be revoked, and are therefore nonmonotonic.

A4. Synchronization axioms

a. $\epsilon\text{-}\mathit{sync}_t\ (P,Q) \supset \epsilon\text{-}\mathit{sync}_{\langle t-\epsilon,t+\epsilon\rangle}\ (Q,P)$

b. $\epsilon_1\text{-}\mathit{sync}_{t,S}\ (P,Q) \wedge \epsilon_2\text{-}\mathit{sync}_{t,S}\ (Q,R) \supset (\epsilon_1+\epsilon_2)\text{-}\mathit{sync}_{t,S}\ (P,R)$

The reference to a principal's clock can be changed to another synchronized principal's clock. Synchronization bounds are accumulative.

A5. Time Conversion Axioms

a. $P\ \mathit{received}_{t,S}\ X \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R) \supset P\ \mathit{received}_{\langle t-\epsilon,t+\epsilon\rangle,R}\ X$

Corresponding axioms A5b–A5e hold for *says*, *said*, *at*, and *has*, respectively.

f. $P \ \mathit{believes}_{[t-\epsilon,t],S} \ \varphi \wedge \epsilon\text{-}\mathit{sync}_t \ (S,R) \supset P \ \mathit{believes}_{\langle t-\epsilon,t+\epsilon\rangle,R} \ \varphi$

The corresponding axiom A5g holds for *controls*.

h. $P \overset{K}{\leftrightarrow}_{[t-\epsilon,t],S} Q \wedge \epsilon\text{-}\mathit{sync}_t \ (S,R) \supset P \overset{K}{\leftrightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} Q$

Corresponding axioms A5i and A5j hold for verification keys and encryption keys respectively.

Time conversion axioms allow the a time on one principal's clock to be changed to an existential interval on a synchronized principal's clock. Note that time conversion on constructs that can be revoked, such as *believes*, *controls*, and key-related constructs, are required to hold on a time interval in order to allow time conversion. This is because otherwise, it would be possible for a formula to hold for such a short time on one principal's clock that it would never actually be true on another principal's clock.

Originator Identification

A6. $\overset{K}{\mapsto}_{t,S} Q \wedge R \ \mathit{received}_{t,S} \ \lfloor X \rfloor_{K^{-1}} \supset Q \ \mathit{said}_{t,S} \ X \wedge Q \ \mathit{said}_{t,S} \ \lfloor X \rfloor_{K^{-1}} \wedge Q \ \mathit{has}_{t,S} \ K^{-1}$

A7. $P \overset{K}{\leftrightarrow}_{t,S} Q \wedge R \ \mathit{received}_{t,S} \ \{X\}_{K^{-1}} \wedge \neg P \ \mathit{said}_{t,S} \ \{X\}_{K^{-1}} \supset Q \ \mathit{said}_{t,S} \ X \wedge Q \ \mathit{said}_{t,S} \ \{X\}_{K^{-1}} \wedge Q \ \mathit{has}_{t,S} \ K^{-1}$

Signatures can be used to deduce the originator of messages. For shared keys, encryption provides partial originator identification: the originator can only be uniquely determined if one of the key sharers is known not to have sent the message.

Receiving and Having

A8. $P \ \mathit{received}_t \ (X_1, \ldots, X_n) \supset P \ \mathit{received}_t \ X_i$

A9. $P \ \mathit{received}_t \ K \supset P \ \mathit{has}_t \ K$

A10. $P \ \mathit{received}_t \ \{X\}_K \wedge P \ \mathit{has}_t \ K^{-1} \supset P \ \mathit{received}_t \ X$

A11. $P \ \mathit{received}_t \ \lfloor X \rfloor_{K^{-1}} \supset P \ \mathit{received}_t \ X$

Principals can decrypt encrypted messages if they have decryption keys, and principals can read signed messages with or without the signature verification key. (However, in order to actually verify the originator of a signed message, it is necessary to have the key.)

Saying

A12. $P \ \mathit{said}_t \ (X_1, \ldots, X_n) \supset P \ \mathit{said}_t \ X_i$

A13. $P \ \mathit{says}_t \ (X_1, \ldots, X_n) \supset P \ \mathit{says}_t \ X_i$

A14. $P \ \mathit{said}_t \ \lfloor X \rfloor_{K^{-1}} \supset P \ \mathit{said}_t \ X$

A15. $P \ \mathit{says}_t \ \lfloor X \rfloor_{K^{-1}} \supset P \ \mathit{says}_t \ X$

A16. $P \ \mathit{said}_t \ X \supset (\exists t' \leq t) P \ \mathit{says}_{t'} \ X$

A17. $P \ \mathit{says}_t \ X \supset P \ \mathit{said}_t \ X$

Parts of messages that are said are also said. *Says* and *said* have the expected relationship. Note that A14 and A15 state that principals are responsible, for the contents of signed messages that they send (in the sense of being considered to have said them), even if it was not their keys used to sign them. If instead, P wants to forward a signed message $\lfloor X \rfloor_{K^{-1}}$ received from Q without being responsible for the contents (i.e. without enabling the deduction that $P \ \mathit{said} \ X$), this could be represented in our logic by having P send the message $Q \ \mathit{said}_t \ \lfloor X \rfloor_{K^{-1}}$. Alternately, if desired, the logic could be extended to include an explicit forwarding notation.

Recognition We represent recognition of message by belief of receipt, and provide an axiom stating that a principal recognizes encrypted messages that it receives if it has the corresponding decryption key.

A18. P *received*$_t$ $\{X\}_K \wedge P$ *has*$_t$ $K^{-1} \supset P$ *believes*$_t$ P *received*$_t$ $\{X\}_K$

Semantically, P's recognition of an encrypted message $M = \{X\}_K$ means that P recognizes that M is, in fact, a message encrypted with the key K, but does not necessarily mean that P can recognize all the components of X. (More details are provided in Appendix A.) Note that a similar recognition axiom could be added if desired for hash functions to reflect that a principal who recognizes X and receives $H(X)$ can also recognize $H(X)$.

Freshness

A19. *fresh*$_t$ $X \supset$ *fresh*$_t$ $\lfloor X \rfloor_K$

If a message is fresh, then its signature is also fresh. Axiom A19 is useful for reasoning about challenge-response protocols that use a signed nonce in response to a nonce. Similar axioms could be added for encrypted nonces and other common challenge-response mechanisms.

Nonce-Verification

A20. *fresh*$_{t_1,P}$ $X \wedge P$ *said*$_{t_2}$ $X \supset P$ *says*$_{\langle t_1,t_2 \rangle}$ X

If a message is fresh at time t_1 and has been said at time t_2, then it was actually uttered at some time between t_1 and t_2.

Jurisdiction

A21. P *controls*$_t$ $\varphi \wedge P$ *says*$_t$ $\varphi \supset \varphi$

Principals only make accurate statements about formulas they control.

Symmetry

A22. $P \stackrel{K}{\leftrightarrow}_{t,R} Q \equiv Q \stackrel{K}{\leftrightarrow}_{t,R} P$

A23. ϵ-*sync*$_{t,R}$ $(P,Q) \equiv \epsilon$-*sync*$_{t,R}$ (Q,P)

P and Q sharing a key means the same thing as Q and P sharing a key, and similarly for synchronization.

2.3 Extension of Beliefs

Unlike all earlier authentication logics our logic is somewhat restrictive in that there are no axioms that extend beliefs from one time to a later time. This is done for two reasons. First, it allows for the possibility of beliefs to change over time. Second, it supports the use of different policies for extension of beliefs in different situations.

In this section, we give several (oversimplified) examples of such policies as an illustration of the kinds of policies that can be expressed and how this could be done. Note that that policies are not axioms, but must be explicitly assumed as initial assumptions if they are to be used in a proof. Furthermore, the soundness of the logic as shown in Appendix B does not apply to proofs using these policies unless they too can be shown to be sound. In general, most policies are not sound in our general model of computation. Rather, having a particular policy can be thought of as an assumption on the part of the affected participants that the setting in which the protocol is executing is a restricted model of computation in which the policies are sound.

The simplest policy allows arbitrary extension of belief, essentially treating all beliefs as stable.

P1. **Stable beliefs**: P *believes*$_{t,S}$ $\varphi \wedge (t \leq t') \supset P$ *believes*$_{t',S}$ φ

This policy is simple, but useful. P1 results in a setting in which beliefs are stable, since any belief at a given time can be extended to a later time. We can extend beliefs if there are *recent* statements, if statements have not been received about something being *revoked*, and if statements satisfy a *time constraint* window. Extending a belief about a formula using the recency of a statement is not as straightforward as it may seem. Extending a belief incrementally without a stopping point is like making a belief stable. Consequently, we use a "constraint" tuple to constrain the window for which a belief may be extended to no more than δ time units from their original belief. We give generic policies for extending beliefs about public keys. The policies below assume some authority S has jurisdiction over (certain) keys. More generally, the relevant policy as well as the relevant jurisdiction assumption would both be believed by some principal. Similarly, some kind of synchronization assumption regarding S and P would be needed to carry out a useful analysis with these policies. Examples of such policies in use can be found in Section 3.

P2. **Extend keys if recent and satisfy constraint window**:
$P\ \mathit{believes}_{t_1}\ \overset{K_Q}{\Rightarrow}_{\hat{t},P} Q \wedge P\ \mathit{believes}_{t_1}\ S\ \mathit{says}_{\hat{t},P}\ \langle Q, K_Q, ([t_b, t_e], S)\rangle \wedge t_b \leq t_2 \leq t_e \wedge\ t_1 \leq t_2 \leq \hat{t} + \delta \supset$
$P\ \mathit{believes}_{t_2}\ \overset{K_Q}{\Rightarrow}_{t_2,P} Q$

P3. **Extend keys if not revoked**:
$P\ \mathit{believes}_{t_1}\ \overset{K_Q}{\Rightarrow}_{t_1,P}\ Q \wedge (t_1 \leq t_2) \wedge (\forall \hat{t} : t_1 \leq \hat{t} \leq t_2) \neg (P\ \mathit{believes}_{\hat{t}}\ S\ \mathit{says}_{\hat{t},P}\ \neg \overset{K_Q}{\Rightarrow}_{\hat{t},P}\ Q) \supset$
$P\ \mathit{believes}_{t_2}\ \overset{K_Q}{\Rightarrow}_{t_2,P} Q$

P2 says that formulas are believed for a window of time of length δ. These policies should not be assumed if the protocol allows for revocation of the particular formula specified in the policy. If the protocol allows for revocation, then other policies such as P3 would be more appropriate. P3 says that principals believe formulas until they receive notification to revoke the belief. Similarly, one could define policies based on recency of keys and on the belief that they are not yet revoked. Note that P3 is not generally sound, since a key is usually revoked only after it is compromised (rather than at the instant it is compromised), but it is a commonly used policy, and therefore important to consider.

It would also be possible to define more complicated policies. For example, one might wish to define a policy in which the extension of beliefs depends on the formula in question. There are several reasons for representing policies as inferences to be assumed in a proof. It allows more flexibility, as the policy can be tailored to an individual situation. In addition, it allows for consideration of a particular protocol with respect to multiple policies. The ability to reason about the conjunction of policies and protocols will be especially important as public and private key infrastructures are deployed and new and unanticipated policies are put into use.

2.4 Using the Logic

In this section, we sketch the use of our logic to perform syntactic analysis of protocols. Following [Mao95, SvO96], we replace the idealization step of [AT91, BAN89] by additional premises used in the analysis. A protocol is specified by syntactically describing the messages that are sent in the language of messages defined in Section 2.1.

Once the protocol is specified, we can attempt to prove protocol goals from a set of premises. For each message

$$P \rightarrow Q :\ X$$

in the specification, $Q\ \mathit{received}_t\ X$ is included in the premise set, where t is the time on Q's local clock when the message is received. We sometimes write $P \rightarrow_{\hat{t}} Q :\ X$ to denote that P sends the message X at the local time $\hat{t}$ on P's clock. In addition, the premise set includes three kinds of premises. First, *message interpretation* premises relate the received messages to the interpretation made by the principal receiving the message. Second, *protocol-specific policies* such as the belief-extension policies in Section 2.3 must be explicitly included as initial premises if they are to be used in the analysis. Finally, the premise set includes any *initial assumptions* such as shared keys between participants and control principals may have over various statements.

A *proof* consists of a sequence of formulas. Each formula must either be in the premise set, be an axiom, be derivable by necessitation, or be derivable from formulas earlier in the sequence by modus ponens. For any formula φ in the proof, the proof can be interpreted as saying that if the formulas in the premise set are true, then φ is true. Hence, in order for the results of a proof to be meaningful outside of the logic, the assumptions must be somehow justified. In particular, if any assumptions that are inferences are included in the premise set, the soundness of the logic (as shown in Appendix B) only applies if those assumptions can be shown to be true (i.e. sound) in that model or an appropriately restricted model. However, even if an analysis is carried out with assumptions not shown to be true, the results can still be useful in that if conclusions of the proof are later shown to be false, it implies that one of the assumptions was false, which may be useful feedback for a protocol designer.

In order to make all belief extension explicit, we recommend that the time of a principal's beliefs in any initial assumptions be no later than when the principal starts to operate. If this restriction is followed, one is less likely to overlook policy assumptions used to make formulas and beliefs current at the time of message receipt.

3 Examples

In Section 3.1, we present example techniques of the use of our logic. This is followed by an analysis of Web-based certificate verification in Section 3.2, a more complete analysis of the Kerberos protocol in Section 3.3, and a demonstration of how our logic handles the known flaws and fixes of the Wide Mouthed Frog protocol in Section 3.4.

3.1 Techniques

In this section, we analyze a protocol fragment that uses public key certificates to highlight several interesting aspects of the use of our logic. In order to carry out the analysis, we make several assumptions (labelled by B's) such as principals holding particular keys, times of message receipts, and so forth. As with the belief extension assumptions in Section 2.3, some of the assumptions are stated as implications in order to reflect the reasons a particular conclusion is drawn.

Expressing the Semantics of Message Timestamps Our logic enables one to express subtle but important distinctions in semantics for protocols having message timestamps. To express that a message timestamp gives the time a message is sent, we can use a formula of the form $P \ \textit{says}_t \ X$ where t is the value of the timestamp. Note that X need not take on a truth value. For example, P can be signing and forwarding encrypted text and a timestamp without knowledge of the corresponding plaintext. Messages of this form can be used to prove the time of presence of principal P, as illustrated later.

Public Key Identity Certificate For illustrative purposes, consider what we shall call a public key identity certificate where Q is a principal, K_Q is Q's public signature verification key, s_Q is the certificate serial number, t_b and t_e are the begin and end times of the key validity period, t_S is a timestamp indicating the time the certificate information was deemed accurate, and K_S^{-1} is the private signature key of a principal S certifying the message contents.

Message 1 $\quad S \rightarrow P: \ \lfloor Q, K_Q, s_Q, t_b, t_e, t_S \rfloor_{K_S^{-1}}$

For some t, our goal is to work towards deducing

Goal: $P \ \textit{believes}_t \ \overset{K_Q}{\Rightarrow}_{t,P} Q$

For the purposes of this example, we assume P initially has the server key and the above certificate is issued between t_b and t_e.

B1. $P \ \textit{has}_{t_0} \ K_S$

B2. $t_b \le t_S \le t_e$

We assume the receipt of Message 1 at some arbitrary time t_1 after the beginning of the session (i.e., t_0).

B3. P *received*$_{t_1}$ $\lfloor Q, K_Q, s_Q, t_b, t_e, t_S \rfloor_{K_S^{-1}}$

B4. $t_0 \le t_1$

P obtains the key from the contents of the signed message.

1. P *has*$_{t_1}$ K_Q *(by A9, B3, A8, and A11)*

As illustrated in the full example in Section 3.3, assumptions about interpreting received messages are made explicit. In assumption B5, we assume the receipt of this message is interpreted as the certifier saying at t_S a key tuple and the goodness of a key for the entity. (To minimize complexity of our notation we drop the serial number from this example.)

B5. P *received*$_{t_1}$ $\lfloor Q, K_Q, s_Q, t_b, t_e, t_S \rfloor_{K_S^{-1}} \wedge\ t_b \le t_S \le t_e \supset$
P *believes*$_{t_1}$ P *received*$_{t_1}$ $\lfloor S$ *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S, S} Q)\rfloor_{K_S^{-1}}$

We can now conclude the formula for the received message.

2. P *believes*$_{t_1}$ P *received*$_{t_1}$ $\lfloor S$ *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S, S} Q)\rfloor_{K_S^{-1}}$ *(by B5, B2, and B3)*

Assuming Stable Beliefs To verify the signature on the received message, P needs to believe the goodness of a signature verification key. We assume P believes in the goodness of a verification key with the certificate authority.

B6. P *believes*$_{t_0}$ $\overset{K_S}{\Rightarrow}_{t_0, P} S$

However, assumption B6 alone is inadequate to analyze statement 2 at the later time, t_1. For simplicity, assume the belief extension policy (i.e., P1) for the key of S. (Typically, one might assume such a stable belief only for some root key. Then one would reason about the quality of subordinate keys derived from certificates.)

B7. P *believes*$_{t_0}$ $\overset{K_S}{\Rightarrow}_{t_0, P} S \wedge t_1 \ge t_0 \supset P$ *believes*$_{t_1}$ $\overset{K_S}{\Rightarrow}_{[t_0, t_1], P} S$

This assumption says that beliefs about the goodness of S's signature verification key are stable as time progresses. We caution the reader that stable belief assumptions should be used sparingly in protocols that must tolerate security failures such as breaches of private keys. However in some designs, stable assumptions, although strong, might be justified, for example if keys are embedded in tamper-resistant hardware. However, in other designs stable assumptions about the goodness of a key reflects a design vulnerability. For example, there are numerous versions of Web browsers in use that have static keys embedded in the software. However, many versions of these browsers are now known to be vulnerable.

We obtain the desired belief about the goodness of the S's signing key.

3. P *believes*$_{t_1}$ $\overset{K_S}{\Rightarrow}_{[t_0, t_1], P} S$ *(by B7, B6, and B4)*

In practice, protocols can be designed to recover from the compromise of a key or the loss of jurisdiction. Unlike other BAN-like logics we can reason about these protocols using policies for belief extension such as "believe until revoked" or "believe if recent". Analysis assuming these policies is performed in a manner similar to the analysis of this protocol fragment and is illustrated later in this section.

Identifying the Originator Originator identification axioms deduce the originator of a message. We use the fact that we believed the signature verification key was good up to the time the message was received.

4. P *believes*$_{t_1}$ S *said*$_{t_1,P}$ $\lfloor S$ *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S,S} Q)\rfloor_{K_S^{-1}}$ *(by A1, A6, 2, and 3)*

This statement illustrates the technique of qualifying subscript times with the identifier of a principal when the time of the reference is different from the principal making the statement. The subscript t_1, P on *said* indicates that P believes this certificate was sent at or before t_1 on P's local clock.

Fresh Uses of Key Our logic helps avoid confusion about fresh keys and fresh messages created using keys. Defining freshness at the granularity of time, instead of at the granularity of BAN-like epochs, exposes the notion of fresh uses of keys. A key or formula is fresh until it is sent in a message. Furthermore, a message composed entirely from elements that are not necessarily fresh can itself be fresh. In this example, we take as an assumption that P believes that S's timestamped statement is fresh because P believes that S does not send more than one signed timestamp with the same time and that S has been following this policy for at least long enough to cover all recent messages. That is, in this case, recency is being used to derive freshness; recency is verified by assuming some synchronization:

B8. P *believes*$_{t_1}$ ϵ-*sync*$_{t_1}$ $(P, S) \wedge (t_S \leq t_1 + \epsilon) \wedge P$ *believes*$_{t_1}$ S *said*$_{t_1,P}$ $\lfloor S$ *says*$_{t_S}$ $X\rfloor_{K_S^{-1}} \wedge (t_0 \leq t_1) \supset$
P *believes*$_{t_1}$ *fresh*$_{\langle t_0, t_1\rangle, P}$ $\lfloor S$ *says*$_{t_S}$ $(X)\rfloor_{K_S^{-1}}$

Note the use of an inference as an assumption to explicitly indicate—and in some sense, justify—the relationship between the premise and the conclusion of the assumption, rather than using the simpler but less explicit assumption of the conclusion alone. Thus, using two additional assumptions:

B9. $\forall t \geq t_0,$ P *believes*$_t$ ϵ-*sync*$_t$ (P, S)

B10. $t_S \leq t_1 + \epsilon$

we obtain:

5. P *believes*$_{t_1}$ *fresh*$_{\langle t_0, t_1\rangle, P}$ $\lfloor S$ *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S,S} Q)\rfloor_{K_S^{-1}}$ *(by B8, B9, B10, 4, B4)*

Deducing the Time a Message was Sent We can deduce a message was at an indicated timestamp through a sequence of steps relying on knowing the sender of the message and knowing the sender had jurisdiction over her timestamp (statements 6–9).

6. P *believes*$_{t_1}$ S *says*$_{\langle t_0, t_1\rangle, P}$ $\lfloor S$ *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S,S} Q)\rfloor_{K_S^{-1}}$ *(by A16 and 4)*

We remove the signature.

7. P *believes*$_{t_1}$ S *says*$_{\langle t_0, t_1\rangle, P}$ S *says*$_{t_S}$ $(\langle Q, K_Q, ([t_b, t_e], S)\rangle, \overset{K_Q}{\Rightarrow}_{t_S,S} Q)$ *(by A13 and 6)*

We assume the stable belief that S has control over her message timestamps and then deduce the time the components of the message were sent.

B11. $\forall t \geq t_0, P$ *believes*$_t$ S *controls*$_{[t_0,t],P}$ $\forall t'_S, S$ *says*$_{t'_S}$ X

8. P *believes*$_{t_1}$ S *says*$_{t_S}$ $\overset{K_Q}{\Rightarrow}_{t_S,S} Q$ *(by A1, A21, 7, and B11)*

9. P *believes*$_{t_1}$ S *says*$_{t_S}$ $\langle Q, K_Q, ([t_b, t_e], S)\rangle$ *(by A1, A21, 7, and B11)*

Suppose as an initial assumption that P believes that S has jurisdiction, for all times after t_0, concerning the goodness of Q's key.

B12. P *believes*$_{t_0}$ $\forall t''_S, S$ *controls*$_{t''_S}$ $(\forall K'_Q) \overset{K'_Q}{\Rightarrow}_{t''_S,S} Q$

Assuming stable beliefs over jurisdiction, we can extend this belief to t_1 where $t_1 \geq t_0$ and we can instantiate the particular key and time in question. (For brevity, the details of this analysis are omitted.):

10. P *believes*$_{t_1}$ S *controls*$_{t_S}$ $\overset{K_Q}{\Rightarrow}_{t_S,S} Q$

11. P *believes*$_{t_1}$ $\overset{K_Q}{\Rightarrow}_{t_S,S} Q$ *(by A1, A21, 10, and 8)*

We have deduced that P believes that K_Q is a good key at t_S on S's clock. Where protocol correctness relies on P believing that K_Q is a good key at a time on P's clock, we need to derive a trusted clock synchronization path between S and P to determine what t_S on S's clock means in reference to P's clock.

Reasoning about Synchronization Our logic enables one to deduce trusted synchronization paths and to derive synchronization bounds between principals from initial assumptions and messages. This is a necessary step when protocol correctness depends on synchronization. For example, it is common practice to authenticate keys using a chain of certificates where each certificate is issued by a different entity according to its own clock. In this simplified example, we assume that principals synchronize their clocks to a single intermediate time server I.

B13. $\forall t \geq t_0,\ P$ *believes*$_t$ $\forall t'_S\ \epsilon_1$-*sync*$_{t'_S,S}$ (S, I)

B14. $\forall t \geq t_0,\ P$ *believes*$_t$ $\forall t'_S\ \epsilon_2$-*sync*$_{t'_S,S}$ (P, I)

Suppose we wish to deduce a trusted synchronization path and derive the degree of synchronization between P and S to interpret the time t_S on S's clock in terms of P's clock for the belief in statement 11. Since the synchronization formula is symmetric, we obtain

12. $\forall t \geq t_0,\ P$ *believes*$_t$ $\forall t'_S\ \epsilon_2$-*sync*$_{t'_S,S}$ (I, P) *(by A23 and B14)*

We apply the clock synchronization axiom.

13. P *believes*$_{t_1}$ ϵ-*sync*$_{t_S,S}$ (S, P) *(by A4b, B13, and 12)*

where $\epsilon = \epsilon_1 + \epsilon_2$. Hence, we have deduced a synchronization bound between S and P.

Converting the Time of Reference Converting the time of reference from one principal's clock to another's is relatively straightforward transitive operation given a belief about synchronization bounds for the time in question. We apply the time conversion axiom to obtain a belief about the goodness of Q's signature verification key in terms of P's clock.

14. P *believes*$_{t_1}$ $\overset{K_Q}{\Rightarrow}_{\langle t_S-\epsilon, t_S+\epsilon\rangle,P} Q$ *(by A5f, 11, and 13)*

Reasoning about Recency and Extending Beliefs In practice, it is sometimes important that principals believe that statements are recent. For example, a discussed by Stubblebine [Stu95], Kerberos [NT94] enables networked application servers, or *vendors*, to constrain the recency of client authentication with the *authentication server*. To allow enforcement of recency constraints, the time that clients authenticate to the authentication server is carried in a ticket that is presented later to vendors when requesting service. Vendors can choose to reject tickets if the time of initial authentication does not meet the vendor's policy for recency.

We now give an example for reasoning about recency. Suppose we have

B15. P *received*$_{t_2}$ $\lfloor X \rfloor_{K_Q^{-1}}$.

To reduce the chances that K_Q^{-1} is no longer a good key for Q, we can require tight constraints on the recency of the goodness of Q's key used in evaluating assumption B15. Suppose we have a recency policy for "extend keys if recent and satisfy a constraint window" (i.e., Policy P2) pertaining to Q's keys, where δ is the size of the constraint window:

B16. $\forall \hat{t} \in [t_s - \epsilon, t_s + \epsilon], (P \ \mathit{believes}_{t_1} \ \overset{K_Q}{\Rightarrow}_{\hat{t},P} Q \wedge P \ \mathit{believes}_{t_1} \ S \ \mathit{says}_{\hat{t}} \ \langle Q, K_Q, ([t_b, t_e], S)\rangle \wedge (t_b \leq t_2 \leq t_e) \wedge \ (t_1 \leq t_2 \leq \hat{t} + \delta) \supset P \ \mathit{believes}_{t_2} \ \overset{K_Q}{\Rightarrow}_{t_2,P} Q)$

and suppose further that t_2 satisfies the following time constraints:

B17. $(t_b \leq t_2 \leq t_e) \wedge (t_1 \leq t_2 \leq t_S + \delta)$

We now obtain our desired goal:

15. $P \ \mathit{believes}_{t_2} \ \overset{K_Q}{\Rightarrow}_{t_2,P} Q$ *(by B16, 9, and B17)*

Reasoning about Revocation Our logic allows us to reason about revocation of formulas. We give an example for reasoning about revocation of keys; however, similar techniques can be applied to any revocable formula (e.g., formulas involving control, synchronization, etc.). Suppose we assume a certificate revocation list similar to that of the X.509 standard [CCI93]. This revocation list contains a timestamp t, and zero or more serial numbers unambiguously identifying revoked keys. For simplicity, assume the revocation message contains one serial number. Suppose we have

B18. $P \ \mathit{believes}_{t_2} \ \overset{K_Q}{\Rightarrow}_{\hat{t},P} Q$

and receive the following revocation message:

Message 2 $\quad S \rightarrow P: \ \lfloor t_r, \text{ Revoked Certificates: } s_Q \rfloor_{K_S^{-1}}$

From the receipt of this message and the appropriate received message interpretation axiom we obtain the following. (For brevity, the details of this analysis are omitted.):

16. $P \ \mathit{believes}_{t_2} \ P \ \mathit{received}_{t_2} \ \lfloor S \ \mathit{says}_{t_r} \ \neg \overset{K_Q}{\Rightarrow}_{t_r,S} Q \rfloor_{K_S^{-1}}$

From the revocation list and other formulas, suppose we are able to further deduce the following:

17. $P \ \mathit{believes}_{t_3,S} \ S \ \mathit{says}_{t_r} \ \neg \overset{K_Q}{\Rightarrow}_{t_r,S} Q$

As an example, consider the goal to interpret the received message in statement B15 subject to a "Extend keys if not revoked" policy (i.e., analogous to Policy P3) concerning the goodness of Q's keys. That is, our policy assumption is the following:

B19. $P \ \mathit{believes}_{t_2,S} \ \overset{K_Q}{\Rightarrow}_{t_2,S} Q \wedge (t_2 \leq t_4) \wedge (\forall t_r : t_2 \leq t_r \leq t_4) \neg (P \ \mathit{believes}_{t_r,S} \ S \ \mathit{says}_{t_r} \ \neg \overset{K_Q}{\Rightarrow}_{t_r,S} Q) \supset P \ \mathit{believes}_{t_4,S} \ \overset{K_Q}{\Rightarrow}_{t_4,S} Q$

In this case, we are unable to obtain the extended belief $P \ \mathit{believes}_{t_4,S} \ \overset{K_Q}{\Rightarrow}_{t_4,S} Q$ for $t_4 \geq t_3$ due to 17. Therefore, we are prevented from using a revoked key to derive the originator of the received message in statement B15 at t_3. This example illustrates the basic procedure for reasoning about revocation. Using our logic, one can specify and reason about more complex revocation lists and more complex revocation policies.

3.2 Vulnerabilities of Web-Based Certificate Verification

A number of important applications rely on verification of public key certificates. These applications include secure Web based browsing using Secure Sockets Layer protocol (SSL) [FKK96] and secure email using Secure/Multipurpose Internet Mail Extensions (S/MIME)[DHRW97, DHR$^+$97].

Verification of public key certificates is mandated for parties relying on their accuracy. A primary method of allowing users to verify certificates is by going to the Web page and entering the details of the certificate to be verified on the Web form (e.g., Verisign Inc. Web page Oct. 98). The Web page returned identifies details about the certificate including the status of the certificate. The status may be 'verified', 'revoked', or 'expired'. We show this procedure can be vulnerable to an "accidental replay attack" due to caching on the local host.

We now briefly analyze Web-based certificate verification, present in detail the vulnerability mentioned above, and discuss possible fixes.

3.2.1 Protocol Description

Message 1 $\quad U \rightarrow_{t_1} B : \ \mathrm{info}_Q$

Message 2 $\quad B \rightarrow_{t_2} S : \ \lfloor \mathrm{info}_Q \rfloor_{K_{SB}}$

Message 3 $\quad S \rightarrow_{t_3} B : \ \lfloor \mathrm{info}_Q, \mathrm{status} \rfloor_{K_{SB}}$

Message 4 $\quad B \rightarrow_{t_4} U : \ (\mathrm{info}_Q, \mathrm{status})$

In Message 1, a user U supplies the browser B with a unique identifier info_Q for the certificate to be verified. In Message 2, the browser sends the certificate verification request under a shared key established through an SSL session between the server S and itself. In Message 3, the server responds with the status of the certificate. Finally, in Message 4, the browser informs the user of the status of the certificate.

3.2.2 Analysis

The user's verification policy requires the validation of a certificate prior to using it. We express this policy as the goal that, at the end of the protocol, the user believes the goodness of the public key in question. (Subsequently, the user can employ the key for the end application.)

Goal: $U \ \mathit{believes}_{t_4} \ \stackrel{K_Q}{\Rightarrow}_{t_4,U} Q$

Using certificate-based analysis similar to that described in Section 3.1, we assume the session key established between B and S using certificates stored in the browser is good throughout the protocol run.

B1. $\forall t \leq t_4, \ U \ \mathit{believes}_t \ S \stackrel{K_{SB}}{\leftrightarrow}_{t,U} B$

The user runs the protocol to check the validity of K_Q for Q at time t_4, obtaining

1. $U \ \mathit{received}_{t_4} \ (\mathrm{info}_Q, \mathrm{verified})$

To obtain the aforementioned goal, we assumed that statement 1 can be derived, and we also assume the following received message interpretation rule.

B2. $U \ \mathit{received}_t \ (\mathrm{info}_Q, \mathrm{verified}) \supset U \ \mathit{believes}_t \ \stackrel{K_Q}{\Rightarrow}_{t,U} Q$

However, careful examination of this rule reveals that it assumes that any response returned to the user is believed. Web-based information retrieval can be subject to abnormalities due to caching at the local host and caching in the network. Experiments conducted with Microsoft Internet Explorer™ running on Microsoft Windows 95™ confirms that locally cached SSL Web pages can be returned unexpectedly.

Does adding a timestamp by the server fix the problem? Consider, the enhanced rule:

B3. U *received*$_t$ $(\text{info}_Q, \text{verified}, t') \supset U$ *believes*$_t$ $\overset{K_Q}{\Rightarrow}_{t',U} Q$

However, it may be overly optimistic since it implicitly assumes synchronization with the verification server and this may not necessarily be a reasonable assumption. Instead, to be more precise we make the synchronization assumption explicit.

B4. U *received*$_t$ $(\text{info}_Q, \text{verified}, t') \wedge U$ *believes*$_t$ ϵ-*sync*$_t$ $(U, S) \supset$
U *believes*$_t$ $\overset{K_Q}{\Rightarrow}_{\langle t'-\epsilon, t'+\epsilon \rangle, U} Q$

In the absence of an automated approach integrated into the end application, the time in the server's response should be included in the information presented to the user so that she can manually check that the time on the response is recent. In this case, the certificate verification protocol would have a better chance of alerting the user of a cached and possibly invalid response.

This example clearly indicates that the security of a protocol cannot and must not be considered independently of the policies being used by its participants.

3.3 The Kerberos Version 5 Protocol

The Kerberos protocol [KN93, NT94] has been analyzed in the past. However, due to the use of time in our logic, our analysis differs in several critical ways from previous analyses. First, since we do not allow beliefs at one time to be extended to another time without explicit policy assumptions, it is necessary to make such assumptions in order to prove the desired protocol goals. Second, our analysis of the challenge-response portion of the protocol reflects the difference in our handling of nonce verification.

We demonstrate the suitability of our language for expressing and reasoning about clock synchronization and recency of messages. As we will see later, reasoning about message recency is necessary to evaluate the replay detection mechanism.

The principals of the protocol include a client C, an authentication server T, and a vendor V. In the first message, the client requests credentials from the authentication server for the vendor by sending her identity, the identity of the vendor she wishes to communicate with, an expiration time, and a nonce. In the second message, the authentication server responds with credentials consisting of a temporary encryption key and a *ticket*. The temporary encryption key is sent with the identity of the vendor, expiration time, and nonce from the first message. It is all sent encrypted in a long term key shared between the client and the authentication server. The ticket consists of the session key, the client's identity, and the expiration time, all encrypted under a key shared between the authentication server and the vendor. In the third message, the client sends an *authenticator* and also forwards the ticket. The authenticator proves that the request was originated by the principal to whom the ticket was issued. The authenticator includes the identity of the client, a timestamp generated by the client, and it may optionally contain other information used in requesting service from the vendor. (See §3.2.2 of RFC 1510 [KN93]). The timestamp proves that the message was recently generated and is not a replay (as further discussed below). Encrypting the authenticator in the session key proves that it was generated by a party possessing the session key. Since no one except the requesting principal and the server knows the session key (i.e. it is never sent over the network in the clear and we are assuming "perfect" cryptography) this guarantees the identity of the client.

Kerberos requires synchronization between a client and vendor be sufficiently fine such that the vendor can store and check for replayed messages during this window. Authenticators falling outside the window time out since the server can not check the storage to determine if the message is a replay. Given that we can deduce a synchronization path between principals originating and authenticating the message, analysis consists of deducing message authentication assuming a "believe if recent" policy where the recency interval δ is chosen to reflect the "length" of message storage on the server. However, one can not overlook the dependency of recency intervals on sufficiently small synchronization bounds. Protocol analysis using our logic makes this dependency explicit.

Kerberos protects against replay protection using timestamps. If the client time in the authenticator differs from the local server time by more than the allowable clock skew, then an error is returned.

3.3.1 Protocol Description: Kerberos Version 5

Message 1 $C \rightarrow_{\hat{t}_1} T: \; C, V, t_e, n_C$

Message 2 $T \rightarrow_{\hat{t}_2} C: \; \{K_{C,V}, V, t_e, n_C\}_{K_{C,T}}, \{K_{C,V}, C, t_e\}_{K_{T,V}}$

Message 3 $C \rightarrow_{\hat{t}_3} V: \; \{C, t_C\}_{K_{C,V}}, \{K_{C,V}, C, t_e\}_{K_{T,V}}$

Recall that the subscripts on the arrows indicate the local time of the sending processor at the time a message is sent. Correspondingly, we will denote the local time of the receiving processor at the time of receipt of the messages by t_1, t_2 and t_3, respectively. Note that because times are expressed in the local time of different principals and there may be synchronization inaccuracies, the times $\hat{t}_1, t_1, \hat{t}_2, t_2, \hat{t}_3$ and t_3 need not necessarily be in sequence.

3.3.2 Assumptions

Received messages

B1. C $\textit{received}_{t_2}$ $\{K_{C,V}, V, t_e, n_C\}_{K_{C,T}}$

B2. V $\textit{received}_{t_3}$ $\{C, t_C\}_{K_{C,V}}$

B3. V $\textit{received}_{t_3}$ $\{K_{C,V}, C, t_e\}_{K_{T,V}}$

Message order and relative timing

B4. $t_0 \leq t_1, t_2, t_3$

B5. $\hat{t}_1 \leq s_c \leq t_2$

In B5, s_c is the time on C's clock when V sent the second message. Note that although neither V nor C knows s_c, C knows that there must exist *some* time s_c that satisfies the constraints we set out here. That is, C knows that s_c is later than $\hat{t}_1$ and before t_2, since the second message contains the nonce that was generated by C at time $\hat{t}_1$ and so was not sent before then, and it was received at time t_2, so was not sent after then. Strictly speaking, it would be more correct to use an existential quantifier inside of all of C's beliefs and apply the axioms inside the quantification. However, doing so would require (a straightforward but cumbersome) extension of the logic, so we have chosen not to do so here.

B6. $t_2 \leq t_e - \epsilon_1$

As will be seen in B18, ϵ_1 is the assumed synchronization bound between C and T. If B6 does not hold, then C (who can check it after receiving Message 2) should abort and restart the protocol since in this case the message was delayed too long in transit, so the ticket is no longer valid and will be rejected by V.

B7. $s_v \leq t_3 \leq t_e - \epsilon_2$

B8. $t_C - \epsilon_2 \leq t_3 \leq t_C + \epsilon_2$

B9. $t_3 - \delta \leq t_C$ [1]

Not received assumptions

B10. $(\forall t' : t_3 - \delta \leq t' < t_3) \neg V$ $\textit{received}_{t'}$ $\{C, t_C\}_{K_{C,V}}$

The buffer storage period δ is used to detect replays of recent messages. For the purposes of this example, we assume that the authenticator has not previously been received by V in the specified recency interval.

[1] The specification requires that $|t_3 - t_C| \leq \delta$. Hence, a fast clock on a client could result in a message being rejected. However, it is not clear that this check does a lot to protect an adversary from blocking a message and forwarding it later when V's clock is close to t_C.

Interpretation of received messages

B11. C $received_{t_2}$ $\{K_{C,V}, V, t_e, n_C\}_{K_{C,T}} \wedge C$ has_{t_2} $K_{C,T,} \supset$
C $believes_{t_2}$ C $received_{t_2}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}} \wedge$
C $believes_{t_2}$ $\neg C$ $said_{t_2}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}}$

B12. V $received_{t_3}$ $\{K_{C,V}, C, t_e\}_{K_{T,V}} \wedge V$ has_{t_3} $K_{T,V} \supset$
V $believes_{t_3}$ V $received_{t_3}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}} \wedge$
V $believes_{t_3}$ $\neg V$ $said_{t_3}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}}$

B13. V $received_{t_3}$ $\{C, t_C\}_{K_{C,V}} \wedge V$ has_{t_3} $K_{C,V} \supset$
V $believes_{t_3}$ V $received_{t_3}$ $\{C$ $believes_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}} \wedge$
V $believes_{t_3}$ $\neg V$ $said_{t_3}$ $\{C$ $believes_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}}$

Keys

B14. C has_{t_0} $K_{C,T}$

B15. C $believes_{t_0}$ $C \overset{K_{C,T}}{\leftrightarrow}_{t_0} T$

B16. V has_{t_0} $K_{T,V}$

B17. V $believes_{t_0}$ $V \overset{K_{V,T}}{\leftrightarrow}_{t_0} T$

Synchronization

B18. C $believes_{t_2}$ $\epsilon_1\text{-}sync_{t_2}$ (C, T)

B19. V $believes_{t_3}$ $\epsilon_2\text{-}sync_{t_3}$ (V, T)

Belief extension policies

B20. $t_0 \leq t_2 \wedge C$ $believes_{t_0}$ $C \overset{K_{C,T}}{\leftrightarrow}_{t_0} T \supset C$ $believes_{t_2}$ $C \overset{K_{C,T}}{\leftrightarrow}_{t_2} T$

B21. $s_c \leq t_2 \leq t_e - \epsilon_1 \wedge C$ $believes_{t_2}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V \supset C$ $believes_{t_2}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_2} V$

B22. $t_0 \leq t_3 \wedge V$ $believes_{t_0}$ $V \overset{K_{V,T}}{\leftrightarrow}_{t_0} T \supset V$ $believes_{t_3}$ $V \overset{K_{V,T}}{\leftrightarrow}_{t_3} T$

B23. $s_c \leq t_3 \leq t_e - \epsilon_2 \wedge V$ $believes_{t_3}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V \supset V$ $believes_{t_3}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V$

Replay detection policy

B24. $(\forall t' : t_3 - \delta \leq t' < t_3) \neg V$ $received_{t'}$ $\{C, t_C\}_{K_{C,V}} \wedge (t_3 - \delta \leq t_C) \supset$
V $believes_{t_3}$ $(\forall t' : t' < t_3) \neg V$ $received_{t'}$ $\{C, t_C\}_{K_{C,V}}$

Freshness beliefs

B25. $\forall K_{C,V}, C$ $believes_{t_2}$ $fresh_{\hat{t}_1,C}$ $(\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}})$

B26. V $believes_{t_3}$ $fresh_{\langle -\infty, t_3 \rangle, V}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}}$

Jurisdiction beliefs

B27. C *believes*$_{t_2}$ $\forall K_{C,V}, T$ *controls*$_{[\hat{t}_1,t_2],C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{[\hat{t}_1,t_2]} V$

B28. V *believes*$_{t_3}$ $\forall K_{C,V}, T$ *controls*$_{[-\infty,t_3],V}$ $C \overset{K_{C,V}}{\leftrightarrow}_{[-\infty,t_3],V} V$

B29. V *believes*$_{t_3}$ C *controls*$_{[-\infty,t_3],V}$ C *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V$

3.3.3 Analysis

First, we derive beliefs from C's receipt of the second message to obtain a goal that C has a good shared key to communicate with V. We begin by examining the receipt of the first component of message 2.

1. C *believes*$_{t_2}$ $C \overset{K_{C,T}}{\leftrightarrow}_{t_2,C} T$ *(by B20, B4, and B15)*
2. C *believes*$_{t_2}$ C *has*$_{t_2}$ $K_{C,T}$ *(by B14, A3, B4, and R2)*
3. C *believes*$_{t_2}$ C *received*$_{t_2}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}}$ *(by B11, 2, and B1)*
4. C *believes*$_{t_2}$ $\neg C$ *said*$_{t_2}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}}$ *(by B11, 2, and B1)*
5. C *believes*$_{t_2}$ T *said*$_{t_2,C}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}}$ *(by A1, A7, 3, 1, and 4)*
6. C *believes*$_{t_2}$ T *says*$_{\langle\hat{t}_1,t_2\rangle,C}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V, n_C, t_e\}_{K_{C,T}}$ *(by A1, A20, B25, and 5)*
7. C *believes*$_{t_2}$ T *says*$_{\langle\hat{t}_1,t_2\rangle,C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V$ *(by A15 and 6)*
8. C *believes*$_{t_2}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_c} V$ *(by A1, A21, B27, and 7)*
9. C *believes*$_{t_2}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_2} V$ *(by B21, 8, B5, B6, and B18)*

The derivation of statement 2 requires the use of a belief extension policy concerning the goodness of the initial shared key. In practice, an auxiliary protocol might be used to give confidence in the goodness of this initial key. This could be integrated into our analysis by amending our belief extension policy to require a more restrictive extension such as "believe if not revoked" or "believe if recent and not revoked".

Next, we analyze the receipt of the second half of the third message. We derive a goal about V's belief of having a good shared key with C.

10. V *has*$_{t_3}$ $K_{T,V}$ *(by A3 and B16)*
11. V *believes*$_{t_3}$ V *received*$_{t_3}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}}$ *(by B12, B3, and 10)*
12. V *believes*$_{t_3}$ $\neg V$ *said*$_{t_3}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}}$ *(by B12, B3, and 10)*
13. V *believes*$_{t_3}$ $V \overset{K_{V,T}}{\leftrightarrow}_{t_3} T$ *(by B22, B4, and B17)*
14. V *believes*$_{t_3}$ T *said*$_{t_3,V}$ $\{C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e\}_{K_{T,V}}$ *(by A1, A7, 13, 12, and 11)*
15. V *believes*$_{t_3}$ T *says*$_{\langle-\infty,t_3\rangle,V}$ $(C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V, t_e)$ *(by A1, A20, B26, and 14)*
16. V *believes*$_{t_3}$ T *says*$_{\langle-\infty,t_3\rangle,V}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V$ *(by A13 and 15)*
17. V *believes*$_{t_3}$ $C \overset{K_{C,V}}{\leftrightarrow}_{s_v,V} V$ *(by A1, A21, B28, 16)*

We can extend the belief about the goodness of the key to the time the message is received.

18. V *believes*$_{t_3}$ $V \overset{K_{C,V}}{\leftrightarrow}_{t_3} C$ *(by B19, B23, B7, 17, and A22)*

Finally, we analyze the receipt of the authenticator in the third message. We obtain the second order belief that V believes that C believes the goodness of a shared key.

19. V *believes*$_{t_3}$ V *has*$_{t_3}$ $K_{C,V}$ *(by A1, 11, A11, A8, and A9)*

20. V *believes*$_{t_3}$ V *received*$_{t_3}$ $\{C$ *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}}$ *(by A1, B13, B2, and 19)*

21. V *believes*$_{t_3}$ $\neg V$ *said*$_{t_3}$ $\{C$ *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}}$ *(by A1, B13, B2, and 19)*

22. V *believes*$_{t_3}$ C *said*$_{t_3,V}$ $\{C$ *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}}$ *(by A1, A7, 17, 20, and 21)*

23. V *believes*$_{t_3}$ C *says*$_{\langle -\infty,t_3\rangle,V}$ $\{C$ *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V\}_{K_{C,V}}$ *(by A16 and 22)*

24. V *believes*$_{t_3}$ C *believes*$_{t_C}$ $C \overset{K_{C,V}}{\leftrightarrow}_{t_C} V$ *(by A1, A21, B29, and 23)*

Unlike other security analysis methods, we are able to reason about the particular replay detection mechanism because we are able to reason about time (e.g., buffer storage times and actual timestamps in messages). As captured in assumption B24, the replay detection mechanism requires that the timestamp in the authenticator is within some time bound of the message receipt. Also, it requires that the message had not been received within that cache window. We are able to conclude that V believes it has not received the message before.

25. V *believes*$_{t_3}$ $(\forall t' : t_3 - \delta \leq t' < t_3) \neg V$ *received*$_{t'}$ $\{C, t_C\}_{K_{C,V}}$ *(by B24, B10, and B9)*

3.4 Wide Mouthed Frog Protocol

The Wide Mouthed Frog Protocol [BAN89] is an authentication protocol with two known flaws and potential fixes for those flaws. In this section, we show that our logic can be used to reason about the time-based fixes for the flaws. This example illustrates complexities of time that can be specified and analyzed in our logic, an area in which the previous BAN logics had shortcomings.

Two attacks of the Wide Mouthed Frog protocol are described in the literature [CJ97]. We give some highlights in this section. The protocol is as follows.

Message 1 $A \rightarrow S: \; A, \{t_a, B, K_{ab}\}_{K_{as}}$

Message 2 $S \rightarrow B: \; \{t_s, A, K_{ab}\}_{K_{bs}}$

In the first message, A sends to the trusted intermediary S a timestamp, the destination identifier, and the proposed session key. If the timestamp is recent, then the second message is sent. If the timestamp in the second message is recent then one obtains belief about the recent presence of the originator A and the goodness of the new shared key. The formal specification requires S *believes*$_{t_2}$ ϵ-*sync*$_{t_2,S}$ (S, A), and B *believes*$_{t_4}$ ϵ-*sync*$_{t_4,B}$ (B, S). That is, S believes her clock is synchronized with A upon receipt of the message at t_2, and B believes her clock is synchronized with S upon receipt of the second message at t_4. The message verification policy requires that the first message was said no earlier than Δ time units before the current time on S's clock (adjusted for clock synchronization, ϵ).

$$t_A + \epsilon - \Delta \leq t_2 \quad \wedge \quad S \;\textit{believes}_{t_2}\; A \;\textit{says}_{[t_a-\epsilon,t_a+\epsilon],S}\; A, \{B, t_a, A \overset{K_{AB}}{\leftrightarrow}_{t_a,A} B\}_{K_{as}}$$

Similarly, the second message is to be received within some bound adjusted for B's clock.

The first attack consists of replaying the first message within an appropriate time window to S. S checks that the timestamp is recent. However, this message is said to cause reauthentication since S produces a new second message with an updated timestamp. This may or may not be an attack depending upon the interpretation of the specification. A clue that this attack might not really be an attack comes from the explicit assumption, "S believes the formula t_a is fresh." This means "t_a has not been sent in a message at any time before the current run of the protocol" [BAN89]. However, this assumption is not supported by anything else in the protocol description. One method of handling replays for timestamped messages is that described in the previous Kerberos example in Section 3.3. Briefly, S caches all requests that have timestamps within some interval of the server's clock. Request that are before the interval are old and invalid. Requests within the interval are checked against the cache to see that they are not replays. Requests that are too far in the future should also cause concern since they break either the clock synchronization assumption or the assumption that the time stamps used are actually clock readings. Unlike the BAN logic, we can specify assumptions that are much closer to the actual mechanism using negation, timed statements, and clock synchronization assumptions—not found in the BAN logic or any of its extensions. As in the Kerberos example (i.e. B10), we explicitly assume that the same message was not received within some time period, δ:

$$(\forall t' : t_2 - \delta \leq t' < t_2) \neg S \ \ \textit{received}_{t'} \ \ A, \{B, t_a, A \stackrel{K_{AB}}{\leftrightarrow}{}_{t_a, A} \ B\}_{K_{as}}$$

As with Kerberos (i.e., B24), the replay detection rule lets us conclude that the receiver believes the message was not received any time in the past.

$(\forall t' : t_2 - \delta \leq t' < t_2) \neg S \ \ \textit{received}_{t'} \ \ A, \{B, t_a, A \stackrel{K_{AB}}{\leftrightarrow}{}_{t_a, A} \ B\}_{K_{as}} \wedge (t_2 - \delta \leq t_A) \supset$
$S \ \ \textit{believes}_{t_2} \ \ (\forall t' : t' < t_2) \neg S \ \ \textit{received}_{t'} \ \ A, \{B, t_a, A \stackrel{K_{AB}}{\leftrightarrow}{}_{t_a, A} \ B\}_{K_{as}}$

We now move on to the second attack. The second method of attack is described as allowing one session to be recorded and then the attacker continuously uses S as an oracle until he wants to bring about reauthentication between A and B [CJ97].

Message 1 $\quad A \rightarrow S : \ A, \{t_a, B, K_{ab}\}_{K_{as}}$

Message 2 $\quad S \rightarrow B : \ \{t_s, A, K_{ab}\}_{K_{bs}}$

Message 1′ $\quad Z(B) \rightarrow S : \ B, \{t_s, A, K_{ab}\}_{K_{bs}}$

Message 2′ $\quad S \rightarrow Z(A) : \ \{t'_s, B, K_{ab}\}_{K_{as}}$

Message 1″ $\quad Z(A) \rightarrow S : \ A, \{t'_s, B, K_{ab}\}_{K_{as}}$

Message 2″ $\quad S \rightarrow Z(B) : \ \{t''_s, A, K_{ab}\}_{K_{bs}}$

and now Z is in a position to replay appropriate messages to A and B.

Message i $\quad Z(S) \rightarrow A : \ \{t'_s, B, K_{ab}\}_{K_{as}}$

Message ii $\quad Z(S) \rightarrow B : \ \{t''_s, A, K_{ab}\}_{K_{bs}}$

The attack is successful since S doesn't recognize a request (e.g. receipt of Message 1′ or receipt of Message 1″) as having been a response it previously generated (e.g. in Message 2 or Message 2′ respectively).

Recognizing one's own messages can be achieved by having a different type of message for requests and responses. Even without changing the message types, principals can determine if received message is fresh by checking that they both never received them before and they never sent them before within a time interval, δ_B. This can be expressed as

$$B \ \ \textit{believes}_{t_4} \ \ (\forall t' : t_4 - \delta_B \leq t' < t_4) \neg B \ \ \textit{said}_{t'} \ \ B, \{t_B, A, K_{ab}\}_{K_{bs}} \text{ , and}$$

$$B \ \ \textit{believes}_{t_4} \ \ (\forall t' : t_4 - \delta_B \leq t' < t_4) \neg B \ \ \textit{received}_{t'} \ \ \{t_s, A, A \stackrel{K_{AB}}{\leftrightarrow}{}_{t_s, S} \ B\}_{K_{as}}.$$

Again, unlike the BAN logic, we can specify notions that messages haven't recently been sent before or that they are old.

4 Conclusions

We have presented a logic for reasoning about cryptographic protocols. Our logic supports reasoning about several important properties underlying practical cryptographic protocols: synchronization, recency, and revocation. In the following appendices, we also present a semantics for our logic and prove the logic sound with respect to the semantics.

In our logic, we represent security policies as protocol-specific inference rules. In the course of reasoning, the security policies advance the flow of reasoning provided the preconditions relating to the property are satisfied. In this way, the analysis yields information about the protocol together with the specified security policies. By evaluating security objectives with respect to the context of particular principals and their policies, one determines whether assumptions of the correctness proof are within the constraints of the policies of the principals.

In practice, bounds on clock synchronization can have a significant impact on the correctness of cryptographic protocols for distributed systems. We demonstrate how our logic exposes assumptions about trusted synchronization paths and synchronization bounds between principals. This establishes requirements for evaluating the adequacy of the underlying clock synchronization algorithm. Also, it provides a metric for comparing protocol designs. By modifying synchronization assumptions to reflect synchronization failures the designer can assess the extent of vulnerability due to a synchronization failure. Finally, exposing synchronization assumptions is a step towards engineering security architectures to minimize risk due to inconsistent knowledge. No synchronization assumptions are inherent to our logic or model. Like other security-relevant features, protocol-specific assumptions about synchronization are included in the premise set for analysis.

In this paper, we illustrated some recency policies and gave examples for reasoning about recency. We have shown how proving recency properties is fundamental to proving the correctness of practical security mechanisms such as replay detection. Authentication subject to recency properties can also prove the continued presence of an entity. Authentication based on recency of beliefs (such as beliefs about the goodness of keys) or recent-secure authentication [Stu95] is fundamental to managing risk due to the inherent infeasibility of obtaining concurrent knowledge in distributed systems. Recency policies place constraints on the recency of statements, such as statements about the goodness of keys and jurisdiction for issuing keys. In effect, recency policies constrain the window of inconsistent knowledge as the protocol progresses.

Revocation—referencing and negating previously made statements—is an important part of practical systems. Our logic provides a general method for expressing revocation policies and for reasoning about revocation. We illustrated some revocation policies and gave an example for reasoning about the revocation of keys.

Acknowledgements

It is a pleasure to thank Paul Syverson and Martin Strauss for many interesting and useful discussions. We also thank the anonymous reviewers for their helpful comments.

References

[ABLP93] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. A calculus for access control in distributed systems. *ACM Trans. Program. Lang. Syst.*, 15(4):706–734, 1993.

[AG99] M. Abadi and A. Gordon. A calculus for cryptographic protocols: The spi calculus. *Information and Computation*, 148(1):1–70, 1999.

[AT91] Martín Abadi and Mark R. Tuttle. A semantics for a logic of authentication. In *Proc. 10th Annual ACM Symposium on Principles of Distributed Computing*, pages 201–218, August 1991.

[BAN89] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. Technical Report SRC Research Report 39, Digital Equipment Corporation, February 1989. Revised February 1990.

[BM97] A. Bleeker and L. Meertens. A semantics for BAN logic. In *DIMACS Workshop on Design and Formal Verification of Security Protocols*, September 1997.

[CCI93] CCITT. The directory-authentication framework. Technical Report X.509, International Telegraph and Telephone Consultative Committee, February 1993.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature: Version, 1997.

[DHR+97] S. Dusse, P. Hoffman, B. Ramsdell, L. Lundblade, and L. Repka. S/MIME version 2 message specification. *Internet Draft*, November 1997.

[DHRW97] S. Dusse, P. Hoffman, B. Ramsdell, and J. Weinstein. S/MIME version 2 certificate handling. *Internet Draft*, November 1997.

[FHMV95] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. Reasoning about knowledge. *The MIT Press, Cambridge, Massachusetts*, 1995.

[FKK96] A. Freir, P. Karlton, and P. Kocher. The SSL protocol version 3.0. *Internet Draft*, November 1996.

[GNY90] Li Gong, Roger Needham, and Raphael Yahalom. Reasoning about belief in cryptographic protocols. In *Proc. 14th IEEE Symposium on Research in Security and Privacy*, pages 234–248, May 1990.

[GS91] K. Gaarder and E. Snekkenes. Applying a formal analysis technique to the CCITT X.509 strong two-way authentication protocol. *Journal of Cryptology*, 3(2):81–98, 1991.

[KN93] J. Kohl and C. Neuman. The Kerberos network authentication service (V5). *IETF Networking Working Group, RFC 1510*, September 1993.

[LABW92] B. Lampson, M. Abadi, M. Burrows, and E. Wobber. Authentication in distributed systems: Theory and practice. *ACM Trans. on Comp. Systs.*, 10(4):265–310, 1992.

[Mao95] W. Mao. An augmentation of BAN-like logics. In *8th IEEE Computer Security Foundations Workshop, IEEE Computer Society Press*, pages 44–56, June 1995.

[MB95] Y. Moses and B. Bloom. Knowledge, timed precedence and clocks. Technical Report 95-21, Weizmann Institute, July 1995.

[NT94] B. Clifford Neuman and Theodore Ts'o. Kerberos: An authentication service for computer networks. *IEEE Communications*, 32(9), 1994.

[Stu95] Stuart G. Stubblebine. Recent-secure authentication: Enforcing revocation in distributed systems. In *Proc. 19th IEEE Symposium on Security and Privacy*, pages 224–235, May 1995.

[SW96] S. G. Stubblebine and R. N. Wright. An authentication logic supporting synchronization, revocation, and recency. In *Proc. 3rd ACM Conference on Computer and Communications Security*, pages 95–105, March 1996.

[Syv93] Paul F. Syverson. Adding time to a logic of authentication. In *Proc. 1st Conference on Computer and Communication Security*, pages 97–101, November 1993.

[SvO96] Paul F. Syverson and Paul C. van Oorschot. A unified cryptographic protocol logic. Preprint. An earlier version of this work appears as "On Unifying Some Cryptographic Protocol Logics" in *Proc. 18th IEEE Symposium on Research in Security and Privacy*, May 1994, pp. 14–28., 1996.

[WK96] G. Wedel and V. Kessler. Formal semantics for authentication logics. In *European Symposium on Research in Computer Security - ESORICS-96*, September 1996.

A Model of Computation

In this section, we provide a semantics for our logic. Specifically, we describe our model of computation and define the truth of all formulas of our logic in this model. An important feature of our model is the description of relevant timing of events that can play a central role in authentication and security policies. Our model, which builds upon the works of [AT91, FHMV95, MB95, SvO96], is quite general since we make no assumptions about the synchronization of principals' clocks.

We consider a setting in which a finite set of *principals* $\mathcal{P} = \{P_1, \ldots, P_n\}$ communicate by sending messages to each other. In addition, there is a special *environment principal* P_e that represents relevant aspects of the external environment. Each principal has a clock that represents its local time. P_e's clock represents the global time. The global time is not in general known to the principals. Times are represented by natural numbers. As defined below, the behavior of the principals is described by a run; a system is a set of possible runs.

We define several *basic events* that may take place for principals.

- send(X): send message X
- receive(X): receive message X, and
- generate(X): generate message X.

A *timestamped event* is a pair (v, t) where v is a basic event and t is a time. A *history* $\mathcal{H}$ is a set of timestamped events.

Associated with each principal P_i is a local state that contains the all the information P_i has access to, namely, P_i's identity, the current local time, keys P_i can use, and a history of events that have taken place for P_i. Formally, a *local state* is a tuple $s_i = (i, t_i, \mathcal{K}_i^s, \mathcal{K}_i^e, \mathcal{H}_i)$, where t_i is a time, $\mathcal{K}_i^s$ and $\mathcal{K}_i^e$ are sets of keys, and $\mathcal{H}_i$ is a history.

The *environment*, P_e, captures relevant aspects of the system not captured in the local states of the principals, including message modification, replay, loss, and delay. The environment state contains

1. a time t_e, called the *real time*,
2. a sequential history $\mathcal{H}_e$ at t_e, and
3. for $1 \leq i \leq n$, a *message buffer* b_i for P_i containing the set of messages sent to P_i but not yet delivered.

The *global state* of a system with n principals is a $(n+1)$-tuple of the form $(s_e, s_1, \ldots, s_n)$, where s_e is the environment state and each s_i is the local state of principal P_i. A *run* is a function $r : \Re \rightarrow \mathcal{G}$, associating a global state with every point of real time. Given a run r and a time t, we denote the local state of P_i in the global state $r(t)$ by $r_i(t)$. We sometimes write (r, t) to denote $r(t)$, and call (r, t) a *point*.

Given a run r, a time t, and a principal P_i, $\text{clock}_{P_i}(r, t)$ is defined to be P_i's local time, i.e. the time in P_i's local state. Similarly, $\text{history}_{P_i}(r, t)$ is defined to be the history in the local state $r_i(t)$, $\text{skeys}_{P_i}(r, t)$ is defined to be the starting key set $\mathcal{K}_i^s$ in the local state $r_i(t)$, and $\text{ekeys}_{P_i}(r, t)$ is defined to be the ending key set $\mathcal{K}_i^e$ in the local state $r_i(t)$. We define

$$\text{rcvdmsgs}_P(r, t) = \{M : (\text{receive}(M), t') \in \text{history}_P(r, t) \text{ for some } t'\},$$

so $\text{rcvdmsgs}_P(r, t)$ is the set of messages received by P at or before time t in run r. Also, we define

$$\text{sentmsgs}_P(r, t) = \{M : (\text{send}(M), t') \in \text{history}_P(r, t) \text{ for some } t'\},$$

so $\text{sentmsgs}_P(r, t)$ is the set of messages sent by P at or before time t in run r.

Formulas, messages, and other basic values are defined as in Section 2.1. Given a set $\mathcal{K}$ of keys and a set $\mathcal{M}$ of messages, we denote the set of messages derivable from $\mathcal{M}$ using keys in $\mathcal{K}$ (and keys in $\mathcal{M}$ itself) by $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Formally, let $\mathcal{M}$ be a set of messages. Then $\text{submsgs}_{\mathcal{K}}(\mathcal{M}) = \bigcup_{M \in \mathcal{M}} \text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$, where $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$ is defined as the union of $\{M\}$ and

1. $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X_1) \cup \ldots \cup \text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X_n)$ if $M = (X_1, \ldots, X_n)$,

2. $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ if $M = \lfloor X \rfloor_{K^{-1}}$,

3. $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ if $M = \{X\}_K$ and $K^{-1} \in \mathcal{K}$,

4. $\text{submsgs}_{\mathcal{K}\cup\{K\}}(\mathcal{M})$ if $M = K$ is a key and $K \notin \mathcal{K}$.

Note that when a key is found, the new key is applied to the entire original set of messages. We define $\text{subfmls}_{\mathcal{K}}(\mathcal{M})$ to be the set defined by $\{\varphi : \varphi \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and φ is a formula$\}$.

Analogously, we define $\text{recsubmsgs}_{\mathcal{K}}(\mathcal{M})$, the set of messages that can be recognized from a set $\mathcal{M}$ of messages and a set $\mathcal{K}$ of keys, as follows: $\text{recsubmsgs}_{\mathcal{K}}(\mathcal{M}) = \cup_{M\in\mathcal{M}} \text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$, where $\text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$ is defined as

1. $\{M\} \cup \text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(X_1) \cup \ldots \cup \text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(X_n)$ if $M = (X_1, \ldots, X_n)$,

2. $\{M\} \cup \text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ if $M = \lfloor X \rfloor_{K^{-1}}$,

3. $\{M\} \cup \text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ if $M = \{X\}_K$ and $K^{-1} \in \mathcal{K}$,

4. $\{M\} \cup \text{recsubmsgs}_{\mathcal{K}\cup\{K\}}(\mathcal{M})$ if $M = K$ is a key and $K \notin \mathcal{K}$.

The difference between $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and $\text{recsubmsgs}_{\mathcal{K}}(\mathcal{M})$ is subtle but important: a message M will only be added to $\text{recpartsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$ if one of the four recursive cases applies, while M will always be added to $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(M)$. In particular, received encrypted messages are always in $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$, but are only in $\text{recsubmsgs}_{\mathcal{K}}(\mathcal{M})$ if the decryption key is known.

Since P_i's local time t_i may be the same for a period of real time in the run r, the set $\{t : \text{clock}_{P_i}(r,t) = t_i\}$ generally contains more than a single element. We denote this set by $\text{realtime}_{P_i}(r, t_i)$. We write $\text{start}_{P_i}(r, t_i)$ and $\text{end}_{P_i}(r, t_i)$ to denote the minimum and maximum element, respectively, of this set. We denote the set $\{t : t_1 \leq \text{clock}_{P_i}(r,t) \leq t_2\}$ by $\text{realtime}_{P_i}(r, [t_1, t_2])$.

A run r is legal if certain monotonicity and consistency conditions hold. Namely, r is *legal* if for every principal P,

1. $t \leq t'$ iff $\text{clock}_P(r,t) \leq \text{clock}_P(r,t')$,

2. if $t \leq t'$, then $\text{history}_P(r,t) \subseteq \text{history}_P(r,t')$,

3. if $(v, t_i) \in \text{history}_P(r,t)$, then $t_i \leq \text{clock}_P(r,t)$,

4. if $t < t'$, then $\text{ekeys}_P(r,t) \subseteq \text{skeys}_P(r,t')$,

5. if K is a key and $(\text{generate}(K), t) \in \text{history}_P(r,t)\}$, then $K \in \text{skeys}_P(r,t)$,

6. $\text{ekeys}_P(r,t) = \text{skeys}_P(r,t) \cup \{K \in \text{submsgs}_{\text{skeys}_P(r,t)}(\text{rcvdmsgs}_P(r,t)) : K \text{ is a key}\}$,

7. if $(\text{receive}(X), t') \in \text{history}_P(r,t)$, then $(\text{send}(X), t'') \in \text{history}_Q(r,t)$ for some Q and t'' such that $\text{end}_Q(r,t'') \leq \text{start}_P(r,t')$,

Note that this last condition need not imply that principals are synchronized. It can be achieved, for example, by having all participant's clocks tick faster than the shortest possible message delivery time.

Given a time t and a principal P, we denote by $\text{prevtick}_P(r,t)$ the value of $\text{clock}_P(r,t')$ for the largest t' such that $\text{clock}_P(r,t') < \text{clock}_P(r,t)$. Thus, in a legal run, $\text{prevtick}_P(r,t)$ denotes the time on P's clock before P's clock ticks t.

A *system* $\mathcal{R}$ is a set of legal runs. Typically, we will be interested in a set of runs consisting of relevant events occurring before the start of a protocol followed by possible executions of that protocol. In particular, in order for a system to correspond to the set of executions being analyzed by our logic, the system should consist of the possible runs of the system in which the assumptions of the analysis are satisfied (e.g. by being considered as received messages before the start of the protocol execution).

A.1 Semantics

In this section, we inductively define the truth of formulas in our model. An *interpretation* π is a function $\pi(\mathcal{G}, p)$ assigning truth values to primitive propositions at each global state. A pair $\mathcal{I} = (\mathcal{R}, \pi)$ is an *interpreted system*. The truth of formulas is evaluated with respect to a run r in an interpreted system $\mathcal{I}$. We write $(\mathcal{I}, r) \models \varphi$ to indicate that φ is true in the run r of $\mathcal{I}$, and $(\mathcal{I}, r) \not\models \varphi$ to indicate that φ is not true in run r of $\mathcal{I}$. Where $\mathcal{I}$ is clear from context, we write simply $r \models \varphi$ and $r \not\models \varphi$.

We define our truth conditions in such a way that for nonnegated basic formulas, only formulas about the past can be true. We do this in order to maintain a certain kind of stability of formulas that is important to the soundness of the logic. In particular, we only define statements that can be revoked, such as statements about belief, control, and keys, to be true at a given time on a principal's clock if the principal's clock has changed from that time, since otherwise it would be possible to construct runs in which the same statement is both true and false at the same time.

Fix an interpreted system $\mathcal{I}$ and a run r of $\mathcal{I}$. We define the truth of a formula in run r as follows.

Primitive Propositions

$r \models p \;\; at_Q \;\; t$ iff there exists $t' \in \text{realtime}_Q(r, t)$ such that $\pi((r, t'), p) = \text{T}$.

$r \models p \;\; at_Q \;\; [t_1, t_2]$ iff $r \models p \;\; at_Q \;\; t$ for every $t_1 \le t \le t_2$.

$r \models p \;\; at_Q \;\; \langle t_1, t_2 \rangle$ iff $r \models p \;\; at_Q \;\; t$ for some $t_1 \le t \le t_2$.

Logical Connectives

$r \models \varphi \wedge \psi$ iff $r \models \varphi$ and $r \models \psi$.

$r \models \neg\varphi$ iff $r \not\models \varphi$.

$r \models (\exists t : t_1 \le t \le t_2)\varphi$ iff $r \models \varphi$ for some t such that $t_1 \le t \le t_2$.

$r \models (\exists t_1 \ge t_2)\varphi$ iff $r \models \varphi$ for some t_1 such that $t_1 \ge t_2$.

$r \models (\exists t_1 \le t_2)\varphi$ iff $r \models \varphi$ for some t_1 such that $t_1 \le t_2$.

$r \models (\forall t : t_1 \le t \le t_2)\varphi$ iff $r \models \varphi$ for every t such that $t_1 \le t \le t_2$.

$r \models (\forall t_1 \ge t_2)\varphi$ iff $r \models \varphi$ for every t_1 such that $t_1 \ge t_2$.

$r \models (\forall t_1 \le t_2)\varphi$ iff $r \models \varphi$ for every t_1 such that $t_1 \le t_2$.

Relations

$r \models t_1 \le t_2$ iff $t_1 \le t_2$.

$r \models t_1 \ge t_2$ iff $t_1 \ge t_2$.

Received P derives messages from the set of messages P has received and P's current key set.

$r \models P \;\; received_{t,S} \;\; X$ iff $\exists t' \in \text{realtime}_S(r, t)$ such that $X \in \text{submsgs}_{\text{skeys}_P(r,t')}(\text{rcvdmsgs}_P(r, t'))$.

$r \models P \;\; received_{[t_1,t_2],S} \;\; X$ iff $r \models P \;\; received_{t,S} \;\; X$ for every $t_1 \le t \le t_2$.

$r \models P \;\; received_{\langle t_1,t_2 \rangle,S} \;\; X$ iff $r \models P \;\; received_{t,S} \;\; X$ for some $t_1 \le t \le t_2$.

Has

$r \models P \;\; has_{t,S} \;\; K$ iff $\exists t' \in \text{realtime}_S(r, t)$ such that $K \in \text{ekeys}_P(r, t')$.

$r \models P \;\; has_{[t_1,t_2],S} \;\; K$ iff $r \models P \;\; has_{t,S} \;\; K$ for every $t_1 \le t \le t_2$.

$r \models P \;\; has_{\langle t_1,t_2 \rangle,S} \;\; K$ iff $r \models P \;\; has_{t,S} \;\; K$ for some $t_1 \le t \le t_2$.

Said

$r \models P \ \ \mathit{said}_{t,S} \ \ X$ iff $\exists t' \in \mathrm{realtime}_S(r,t)$ such that $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{sentmsgs}_P(r,t'))$.

$r \models P \ \ \mathit{said}_{[t_1,t_2],S} \ \ X$ iff $r \models P \ \ \mathit{said}_{t,S} \ \ X$ for every $t_1 \leq t \leq t_2$.

$r \models P \ \ \mathit{said}_{\langle t_1,t_2\rangle,S} \ \ X$ iff $r \models P \ \ \mathit{said}_{t,S} \ \ X$ for some $t_1 \leq t \leq t_2$.

Says A principal says a message at a particular time if the message is derivable from messages sent at that time.

$r \models P \ \ \mathit{says}_{t,S} \ \ X$ iff there is some message M and some $t' \in \mathrm{realtime}_S(r,t)$ such that $(\mathrm{send}(M), t') \in \mathrm{history}_P(r,t')$ and $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\{M\})$.

$r \models P \ \ \mathit{says}_{[t_1,t_2],S} \ \ X$ iff $r \models P \ \ \mathit{says}_{t,S} \ \ X$ for every $t_1 \leq t \leq t_2$.

$r \models P \ \ \mathit{says}_{\langle t_1,t_2\rangle,S} \ \ X$ iff $r \models P \ \ \mathit{says}_{t,S} \ \ X$ for some $t_1 \leq t \leq t_2$.

Controls Principals neither lie about formulas they control, nor make contradictory statements about formulas they control using the same timestamp.

$r \models P \ \ \mathit{controls}_{t,S} \ \ \varphi$ iff $r \models P \ \ \mathit{says}_{t,S} \ \ \varphi$ implies $r \models \varphi$.

$r \models P \ \ \mathit{controls}_{[t_1,t_2],S} \ \ \varphi$ iff $r \models P \ \ \mathit{controls}_{t,S} \ \ \varphi$ for every $t_1 \leq t \leq t_2$.

$r \models P \ \ \mathit{controls}_{\langle t_1,t_2\rangle,S} \ \ \varphi$ iff $r \models P \ \ \mathit{controls}_{t,S} \ \ \varphi$ for some $t_1 \leq t \leq t_2$.

Freshness A message is fresh if it has not been said before in the run.

$r \models \mathit{fresh}_{t,S} \ \ X$ iff $r \not\models P \ \ \mathit{said}_{t,S} \ \ X$ for all $P \in \mathcal{P}$.

$r \models \mathit{fresh}_{[t_1,t_2],S} \ \ X$ iff $r \models \mathit{fresh}_{t,S} \ \ X$ for every $t_1 \leq t \leq t_2$.

$r \models \mathit{fresh}_{\langle t_1,t_2\rangle,S} \ \ X$ iff $r \models \mathit{fresh}_{t,S} \ \ X$ for some $t_1 \leq t \leq t_2$.

Shared Keys Shared keys provide (partial) originator identification.

$r \models P \overset{K}{\leftrightarrow}_{t,S} Q$ iff

$r \models V \ \ \mathit{received}_{t,S} \ \ \{X\}_K$ implies

1. $r \models T \ \ \mathit{said}_{t,S} \ \ \{X\}_K$,
2. $r \models T \ \ \mathit{said}_{t,S} \ \ X$, and
3. $r \models T \ \ \mathit{has}_{t,S} \ \ K$

for some $T \in \{P, Q\}$

$r \models P \overset{K}{\leftrightarrow}_{[t_1,t_2],S} Q$ iff $r \models P \overset{K}{\leftrightarrow}_{t,S} Q$ for all $t_1 \leq t \leq t_2$.

$r \models P \overset{K}{\leftrightarrow}_{\langle t_1,t_2\rangle,S} Q$ iff $r \models P \overset{K}{\leftrightarrow}_{t,S} Q$ for some $t_1 \leq t \leq t_2$.

Public Signature Verification Keys Signature checking keys are good if they properly identify signatures.

$r \models (\overset{K}{\Rightarrow}_{t,S} P)$ iff $r \models Q \ \ \mathit{received}_{t,S} \ \ \lfloor X \rfloor_{K^{-1}}$ implies (a) $r \models P \ \ \mathit{said}_{t,S} \ \ \lfloor X \rfloor_{K^{-1}}$, (b) $r \models P \ \ \mathit{said}_{t,S} \ \ X$, and (c) $r \models P \ \ \mathit{has}_{t,S} \ \ K$.

$r \models (\overset{K}{\Rightarrow}_{[t_1,t_2],S} P)$ iff $r \models (\overset{K}{\Rightarrow}_{t,S} P)$ for all $t_1 \leq t \leq t_2$.

$r \models (\overset{K}{\Rightarrow}_{\langle t_1,t_2\rangle,S} P)$ iff $r \models (\overset{K}{\Rightarrow}_{t,S} P)$ for some $t_1 \leq t \leq t_2$.

Public Encryption Keys An encryption key is good provided at the points in the run where the encryption key is said to be good, the decryption can be performed by the principal whose key it is.

$r \models (\stackrel{K}{\mapsto}_{t,S} P)$ iff $r \models P$ *received*$_{t,S}$ $\{X\}_K$, implies $r \models P$ *received*$_{t,S}$ X.

$r \models (\stackrel{K}{\mapsto}_{[t_1,t_2],S} P)$ iff $r \models (\stackrel{K}{\mapsto}_{t,S} P)$ for all $t_1 \leq t \leq t_2$.

$r \models (\stackrel{K}{\mapsto}_{\langle t_1,t_2\rangle,S} P)$ iff $r \models (\stackrel{K}{\mapsto}_{t,S} P)$ for some $t_1 \leq t \leq t_2$.

Note that encryption keys only implicitly provide confidentiality, in that one cannot conclude that other principals receiving $\{X\}_K$ also receive X. While it would be desirable to more explicitly state a confidentiality property (i.e. that other principals do *not* receive X), it appears difficult to correctly do so in a BAN-like logical framework (because other principals may receive X by some other means). Hence, as in previous BAN-like logics, confidentiality remains out of the scope of our logic.

Believes We define an equivalence relation $\sim_P$ between points that relates states if they appear the same to P. For a principal P and a point (r,t), we say $(r,t) \sim_P (r',t')$ iff all of the following hold:

1. $\mathrm{recsubmsgs}_{\mathrm{skeys}_P(r,t)}(\mathrm{rcvdmsgs}_P(r,t)) = \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r',t')}(\mathrm{rcvdmsgs}_P(r',t'))$,
2. $\mathrm{recsubmsgs}_{\mathrm{skeys}_P(r,t)}(\mathrm{sentmsgs}_P(r,t)) = \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r',t')}(\mathrm{sentmsgs}_P(r',t'))$,
3. $\mathrm{skeys}_P(r,t) = \mathrm{skeys}_P(r',t')$,
4. $\mathrm{generate}(X) \in \mathrm{history}_P(r,t)$ iff $\mathrm{generate}(X) \in \mathrm{history}_P(r',t')$, and
5. $\mathrm{clock}_P(r,t) = \mathrm{clock}_P(r',t')$.

We then define belief with respect to this relation.

$r \models P$ *believes*$_{t,S}$ φ iff $r' \models \varphi$ for all r' such that $(r,s) \sim_P (r',s')$ for every $s \in \mathrm{realtime}_S(r,t)$ and $s' \in \mathrm{realtime}_S(r',t)$.

$r \models P$ *believes*$_{[t_1,t_2],S}$ φ iff $r \models P$ *believes*$_{t,S}$ φ for every $t_1 \leq t \leq t_2$.

$r \models P$ *believes*$_{\langle t_1,t_2\rangle,S}$ φ iff $r \models P$ *believes*$_{t,S}$ φ for some $t_1 \leq t \leq t_2$.

Synchronization Principals' clocks are not more than ϵ time units from each other. In addition, in order to say two principals' clocks are synchronized, we first require that the run is itself a synchronous run. We do this in order to exclude the possibility, in the extreme case, that two principals' clocks happen to read the same value at only a single point of real time, but are arbitrarily far apart before and after that point.

$r \models \epsilon$-*sync*$_{t,S}$ (P,Q) iff

1. $\forall t' \in \mathrm{realtime}_S(r,t)$, $|\,\mathrm{clock}_P(r,t') - \mathrm{clock}_Q(r,t')| \leq \epsilon$,
2. there is some k such that $\forall\widehat{t}$, $|\{s \in \mathrm{realtime}_P(r,\widehat{t})\}| = |\{s \in \mathrm{realtime}_Q(r,\widehat{t})\}| = k$
3. $\forall\widehat{t}$, $\widehat{t} - \mathrm{prevtick}_P(r,\widehat{t}) \leq \epsilon$
4. $\forall\widehat{t}$, $\widehat{t} - \mathrm{prevtick}_Q(r,\widehat{t}) \leq \epsilon$

$r \models \epsilon$-*sync*$_{[t_1,t_2],S}$ (P,Q) iff $r \models \epsilon$-*sync*$_{t,S}$ (P,Q) for every $t_1 \leq t \leq t_2$.

$r \models \epsilon$-*sync*$_{\langle t_1,t_2\rangle,S}$ (P,Q) iff $r \models \epsilon$-*sync*$_{t,S}$ (P,Q) for some $t_1 \leq t \leq t_2$.

The k in the second condition of the *sync* definition in some sense says that the units of P's clock are the same as the units of Q's clock.

B Soundness

In this section, we show that our logic is sound, in the sense that any derivation allowed by the logic corresponds to a truth in the computational model. Furthermore, we show if a derivation starts with assumptions that are "well-behaved" with regard to time (in a sense we will make precise in Appendix B.1), the results will also be well-behaved with regard to time. More specifically, we define *strong well-timedness* and *weak well-timedness*. Strong well-timedness is more intuitive, but only a subset of the logic can be proven sound with respect to it. On the other hand, the entire logic can be proven sound with respect to weak well-timedness, but the definition does not have as clear an intuitive interpretation.

Our soundness results are formalized below in Theorem B.3. Most of the details of the proof are in the following three subsections. In Appendix B.1, we present the definition of a well-timed message. In Appendix B.2, we show some general results that will be helpful in proving the soundness of our axioms. Finally, we show the soundness of all the logical axioms case by case in Appendix B.3.

For any formula φ and set Γ of formulas, we write $r \models \Gamma$ if $r \models \varphi$ for every $\varphi \in \Gamma$. We write $\Gamma \models \varphi$ if $r \models \varphi$ for all legal runs r such that $r \models \Gamma$. We write $\models \varphi$ if $r \models \varphi$ for every r. Similarly, we say Γ is strongly (respectively, weakly) well-timed with respect to r if φ is strongly (respectively, weakly) well-timed with respect to r for all $\varphi \in \Gamma$ and we say Γ is strongly (respectively, weakly) well-timed if Γ is strongly (respectively, weakly) well-timed with respect to every r.

Lemma B.1 *For every axiom $\varphi \supset \psi$ of our logic and every set Γ of formulas, if $\Gamma \models \varphi$, then $\Gamma \models \psi$. Furthermore, if $\Gamma \models \varphi$ and φ is weakly well-timed with respect to a legal run r, then ψ is weakly well-timed with respect to r.*

Proof: This can be proven by showing that for every axiom in our logic of the form $\varphi \supset \psi$ and every legal run r, that if $r \models \varphi$ then $r \models \psi$, and that if $r \models \varphi$ and φ is weakly well-timed with respect to r, then ψ is weakly well-timed with respect to r. This is shown case by case in Appendix B.3. ■

Let $\mathcal{L}$ be the subset of our logic consisting of both inference rules and all the axioms except A5a–A5g and A16. We write $\Gamma \vdash_{\mathcal{L}} \varphi$ if from the set Γ of formulas, it is possible to derive φ using the axioms and inference rules of $\mathcal{L}$.

Lemma B.2 *For every axiom $\varphi \supset \psi$ of $\mathcal{L}$ and every set Γ of formulas, if $\Gamma \models \varphi$ then $\Gamma \models \psi$. Furthermore, if $\Gamma \models \varphi$ and φ is strongly well-timed with respect to a legal run r, then ψ is strongly well-timed with respect to r.*

Proof: As above, this can be proven by showing that for every axiom in $\mathcal{L}$ of the form $\varphi \supset \psi$ and every legal run r, that if $r \models \varphi$ then $r \models \psi$, and that if $r \models \varphi$ and φ is strongly well-timed with respect to r, then ψ is strongly well-timed with respect to r. This is shown case by case in Appendix B.3. ■

Theorem B.3 *Let Γ be a set of formulas, let φ be a formula, and let r be a legal run. If $\Gamma \vdash \varphi$, then $\Gamma \models \varphi$. Furthermore, if $\Gamma \vdash \varphi$ and Γ is weakly well-timed with respect to r, then φ is weakly well-timed with respect to r. If $\Gamma \vdash_{\mathcal{L}} \varphi$ and Γ is strongly well-timed with respect to r, then φ is strongly well-timed with respect to r.*

Proof: Let Γ be a set of formulas, let φ be a formula, and assume $\Gamma \vdash \varphi$. We show by induction that $\Gamma \models \varphi$. If φ is a member of Γ, then by definition $\Gamma \models \varphi$. Otherwise, φ was derived by applying one of the inference rules: necessitation or modus ponens. If φ was obtained by necessitation, then it was obtained from some ψ such that $\vdash \psi$ and $\varphi = P$ *believes*$_{t_P}$ ψ for some P and t_P. Since ψ is a theorem, it is derivable from the axioms alone. Hence, it follows from Lemma B.1 that $\models \psi$ and that ψ is well-timed. By the truth conditions for *believes*, it follows that P *believes*$_{t_P}$ ψ. If, instead, φ was obtained by modus ponens, then it was obtained from some formula ψ such that $\Gamma \vdash \psi$ and $\Gamma \vdash \psi \supset \varphi$. As an inductive hypothesis, assume that $\Gamma \models \psi$ and $\Gamma \models \psi \supset \varphi$. Hence, it follows from the above and from Lemma B.1 that $\Gamma \models \varphi$.

Now suppose that $\Gamma \vdash \varphi$ and Γ is weakly well-timed with respect to a legal run r. By the above, $\Gamma \models \varphi$, so it follows from Lemma B.1 that φ is weakly well-timed with respect to r.

Finally, if instead $\Gamma \vdash_{\mathcal{L}} \varphi$ and Γ is strongly well-timed with respect to a legal run r, then it follows by the definition of $\vdash_{\mathcal{L}}$ that $\Gamma \vdash \varphi$. Hence, again $\Gamma \models \varphi$, so it follows from Lemma B.2 that φ is strongly well-timed with respect to r. ■

B.1 Well-timed Messages

Since at a given point in time, it is not in general possible for a principal to determine the truth of statements about the future, we will be particularly interested in formulas that don't "refer to the future", except perhaps in tautological or irrelevant ways. In order to formalize this concept, we introduce the notions of *weakly well-timed* messages and *strongly well-timed* messages.

We start by defining the set $\mathrm{time}_r(X)$, for any message X and run r, of times "relevant" to the message X in the run r. Recall that $\mathcal{T}$ is the set of primitive terms. The set $\mathrm{time}_r(X)$ is defined recursively as follows. If φ is a theorem, then $\mathrm{time}_r(\varphi) = \emptyset$. Otherwise,

- if $X \in \mathcal{T}$, then $\mathrm{time}_r(X) = \emptyset$
- $\mathrm{time}_r(\{X\}_K) = \mathrm{time}_r(X)$
- $\mathrm{time}_r(\lfloor X \rfloor_K) = \mathrm{time}_r(X)$
- $\mathrm{time}_r((X_1, \dots, X_n)) = \bigcup_i \mathrm{time}_r(X_i)$
- $\mathrm{time}_r(p \;\; \mathit{at}_Q \;\; t) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(p \;\; \mathit{at}_Q \;\; [t_1, t_2]) = \mathrm{time}_r(p \;\; \mathit{at}_Q \;\; \langle t_1, t_2 \rangle) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(\neg\varphi) = \mathrm{time}_r(\varphi)$
  $\mathrm{time}_r(\varphi \wedge \psi) = \mathrm{time}_r(\varphi) \cup \mathrm{time}_r(\psi)$
- $\mathrm{time}_r(t_1 \leq t_2) = \mathrm{time}_r(t_1 \geq t_2) = \emptyset$
- $\mathrm{time}_r(P \;\; \mathit{believes}_{t,Q} \;\; \varphi) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{believes}_{[t_1,t_2],Q} \;\; \varphi) = \mathrm{time}_r(P \;\; \mathit{believes}_{\langle t_1,t_2 \rangle,Q} \;\; \varphi) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \;\; \mathit{controls}_{t,Q} \;\; \varphi) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{controls}_{[t_1,t_2],Q} \;\; \varphi) = \mathrm{time}_r(P \;\; \mathit{controls}_{\langle t_1,t_2 \rangle,Q} \;\; \varphi) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \;\; \mathit{received}_{t,Q} \;\; X) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{received}_{[t_1,t_2],Q} \;\; X) = \mathrm{time}_r(P \;\; \mathit{received}_{\langle t_1,t_2 \rangle,Q} \;\; X) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \;\; \mathit{said}_{t,Q} \;\; X) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{said}_{[t_1,t_2],Q} \;\; X) = \mathrm{time}_r(P \;\; \mathit{said}_{\langle t_1,t_2 \rangle,Q} \;\; X) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \;\; \mathit{says}_{t,Q} \;\; X) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{says}_{[t_1,t_2],Q} \;\; X) = \mathrm{time}_r(P \;\; \mathit{says}_{\langle t_1,t_2 \rangle,Q} \;\; X) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \overset{K}{\leftrightarrow}_{t,R} Q) = \mathrm{realtime}_R(r, t)$
  $\mathrm{time}_r(P \overset{K}{\leftrightarrow}_{[t_1,t_2],R} Q) = \mathrm{time}_r(P \overset{K}{\leftrightarrow}_{\langle t_1,t_2 \rangle,R} Q) = \mathrm{realtime}_R(r, [t_1, t_2])$
- $\mathrm{time}_r(\overset{K}{\Rightarrow}_{t,Q} P) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(\overset{K}{\Rightarrow}_{[t_1,t_2],Q} P) = \mathrm{time}_r(\overset{K}{\Rightarrow}_{\langle t_1,t_2 \rangle,Q} P) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(\overset{K}{\mapsto}_{t,Q} P) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(\overset{K}{\mapsto}_{[t_1,t_2],Q} P) = \mathrm{time}_r(\overset{K}{\mapsto}_{\langle t_1,t_2 \rangle,Q} P) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(P \;\; \mathit{has}_{t,Q} \;\; K) = \mathrm{realtime}_Q(r, t)$
  $\mathrm{time}_r(P \;\; \mathit{has}_{[t_1,t_2],Q} \;\; K) = \mathrm{time}_r(P \;\; \mathit{has}_{\langle t_1,t_2 \rangle,Q} \;\; K) = \mathrm{realtime}_Q(r, [t_1, t_2])$
- $\mathrm{time}_r(\mathit{fresh}_{t,P} \;\; X) = \mathrm{realtime}_P(r, t)$
  $\mathrm{time}_r(\mathit{fresh}_{[t_1,t_2],P} \;\; X) = \mathrm{time}_r(\mathit{fresh}_{\langle t_1,t_2 \rangle,P} \;\; X) = \mathrm{realtime}_P(r, [t_1, t_2])$
- $\mathrm{time}_r(\epsilon\text{-}\mathit{sync}_{t,R} \;\; (P, Q)) = \mathrm{realtime}_R(r, t)$
  $\mathrm{time}_r(\epsilon\text{-}\mathit{sync}_{[t_1,t_2],R} \;\; (P, Q)) = \mathrm{time}_r(\epsilon\text{-}\mathit{sync}_{\langle t_1,t_2 \rangle,R} \;\; (P, Q)) = \mathrm{realtime}_R(r, [t_1, t_2])$

- $\mathrm{time}_r((\exists t : t_1 \le t \le t_2)\varphi) = \mathrm{time}_r((\exists t \ge t')\varphi) = \mathrm{time}_r((\exists t \le t')\varphi) = \mathrm{time}_r(\varphi)$
- $\mathrm{time}_r((\forall t : t_1 \le t \le t_2)\varphi) = \mathrm{time}_r((\forall t \ge t')\varphi) = \mathrm{time}_r((\forall t \le t')\varphi) = \mathrm{time}_r(\varphi)$

We define two partial orders $\trianglerighteq_s$ and $\trianglerighteq_w$ on sets of times as follows. If R and R' are nonempty sets of times, then $R \trianglerighteq_s R'$ if $\min_{q\in R} q \ge \max_{q\in R'} q$ and $R \trianglerighteq_w R'$ if $\min_{q\in R} q \ge \min_{q\in R'} q$ and $\max_{q\in R} q \ge \max_{q\in R'} q$. Additionally, $R \trianglerighteq_s \emptyset$ and $R \trianglerighteq_w \emptyset$ for any set R of times.

We are now ready to formalize the concept that a message does not "refer to the future". As discussed in Section B, this intuition is better captured by *strong well-timedness*, but at the cost that only a (large) subset of the logic is sound with respect to that definition. We also provide a less intuitive notion of *weak well-timedness*, and are able to show the entire logic sound with respect to it.

Fix a run r. A *strongly well-timed* message with respect to r is defined recursively as follows. Where r is clear from context, we often simply say a message is strongly well-timed. If φ is a theorem, then φ is strongly well-timed (with respect to r). Otherwise,

- if $X \in \mathcal{T}$, then X is strongly well-timed (with respect to r)
- $\{X\}_K$ is strongly well-timed if X is strongly well-timed
- $\lfloor X \rfloor_K$ is strongly well-timed if X is strongly well-timed
- $(X_1, \dots, X_n)$ is strongly well-timed if X_i is strongly well-timed for $1 \le i \le n$
- $p\ \mathit{at}_Q\ t$, $p\ \mathit{at}_Q\ [t_1, t_2]$, and $p\ \mathit{at}_Q\ \langle t_1, t_2\rangle$ are strongly well-timed
- $\neg\varphi$ is strongly well-timed if φ is strongly well-timed
  $\varphi \wedge \psi$ is strongly well-timed if φ and ψ are strongly well-timed
- $t_1 \le t_2$ is strongly well-timed if $t_1 \le t_2$, and $t_1 \ge t_2$ is strongly well-timed if $t_1 \ge t_2$
- $r_1 \le r_2$ is strongly well-timed if $r_1 \le r_2$, and $r_1 \ge r_2$ is strongly well-timed if $r_1 \ge r_2$
- $P\ \mathit{believes}_{t,Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,t) \trianglerighteq_s \mathrm{time}_r(\varphi)$
  $P\ \mathit{believes}_{[t_1,t_2],Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(\varphi)$
  $P\ \mathit{believes}_{\langle t_1,t_2\rangle,Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(\varphi)$
- $P\ \mathit{controls}_{t,Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,t) \trianglerighteq_s \mathrm{time}_r(\varphi)$
  $P\ \mathit{controls}_{[t_1,t_2],Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(\varphi)$
  $P\ \mathit{controls}_{\langle t_1,t_2\rangle,Q}\ \varphi$ is strongly well-timed if φ is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(\varphi)$
- $P\ \mathit{received}_{t,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,t) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{received}_{[t_1,t_2],Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{received}_{\langle t_1,t_2\rangle,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
- $P\ \mathit{said}_{t,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,t) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{said}_{[t_1,t_2],Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{said}_{\langle t_1,t_2\rangle,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
- $P\ \mathit{says}_{t,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,t) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{says}_{[t_1,t_2],Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
  $P\ \mathit{says}_{\langle t_1,t_2\rangle,Q}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_Q(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
- $P \overset{K}{\leftrightarrow}_{t,R} Q$, $P \overset{K}{\leftrightarrow}_{[t_1,t_2],R} Q$, and $P \overset{K}{\leftrightarrow}_{\langle t_1,t_2\rangle,R} Q$ are strongly well-timed

- $\overset{K}{\Rightarrow}_{t,Q} P$, $\overset{K}{\Rightarrow}_{[t_1,t_2],Q} P$, and $\overset{K}{\Rightarrow}_{\langle t_1,t_2\rangle,Q} P$ are strongly well-timed
- $\overset{K}{\mapsto}_{t,Q} P$, $\overset{K}{\mapsto}_{[t_1,t_2],Q} P$, and $\overset{K}{\mapsto}_{\langle t_1,t_2\rangle,Q} P$ are strongly well-timed
- $P\ \mathit{has}_{t,Q}\ K$, $P\ \mathit{has}_{[t_1,t_2],Q}\ K$, and $P\ \mathit{has}_{\langle t_1,t_2\rangle,Q}\ K$ are strongly well-timed
- $\mathit{fresh}_{t,P}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$
  $\mathit{fresh}_{[t_1,t_2],P}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_P(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
  $\mathit{fresh}_{\langle t_1,t_2\rangle,P}\ X$ is strongly well-timed if X is strongly well-timed and $\mathrm{realtime}_P(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$
- $\epsilon\text{-}\mathit{sync}_{t,R}\ (P,Q)$, $\epsilon\text{-}\mathit{sync}_{[t_1,t_2],R}\ (P,Q)$, and $\epsilon\text{-}\mathit{sync}_{\langle t_1,t_2\rangle,R}\ (P,Q)$ are strongly well-timed
- $(\exists t : t_1 \le t \le t_2)\varphi$ is strongly well-timed if φ is strongly well-timed for every $t_1 \le t \le t_2$
  $(\exists t \ge t')\varphi$ is strongly well-timed if φ is strongly well-timed for every $t \ge t'$
  $(\exists t \le t')\varphi$ is strongly well-timed if φ is strongly well-timed for every $t \le t'$
- $(\forall t : t_1 \le t \le t_2)\varphi$ is strongly well-timed if φ is strongly well-timed for every $t_1 \le t \le t_2$
  $(\forall t \ge t')\varphi$ is strongly well-timed if φ is strongly well-timed for every $t \ge t'$
  $(\forall t \le t')\varphi$ is strongly well-timed if φ is strongly well-timed for every $t \le t'$

We define weak well-timedness with respect to a run r exactly as above except that every occurrence of "strongly well-timed" is replaced by "weakly well-timed", every occurrence of $\trianglerighteq_s$ is replaced by $\trianglerighteq_w$, and the following additional changes are made:

- $P\ \mathit{believes}_{\langle t_1,t_2\rangle,Q}\ \varphi$ is weakly well-timed if φ is weakly well-timed and there exists t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_Q(r,t) \trianglerighteq_w \mathrm{time}_r(\varphi)$
- $P\ \mathit{controls}_{\langle t_1,t_2\rangle,Q}\ \varphi$ is weakly well-timed if φ is weakly well-timed and there exists t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_Q(r,t) \trianglerighteq_w \mathrm{time}_r(\varphi)$
- $P\ \mathit{received}_{\langle t_1,t_2\rangle,Q}\ X$ is weakly well-timed if X is weakly well-timed and there exists t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_Q(r,t) \trianglerighteq_w \mathrm{time}_r(X)$
- $P\ \mathit{said}_{\langle t_1,t_2\rangle,Q}\ X$ is weakly well-timed if X is weakly well-timed and there exists t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_Q(r,t) \trianglerighteq_w \mathrm{time}_r(X)$
- $P\ \mathit{says}_{\langle t_1,t_2\rangle,Q}\ X$ is weakly well-timed if X is weakly well-timed and there exists t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_Q(r,t) \trianglerighteq_w \mathrm{time}_r(X)$
- $\mathit{fresh}_{\langle t_1,t_2\rangle,P}\ X$ is weakly well-timed if X is weakly well-timed and there exists a t such that $t_1 \le t \le t_2$ and $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$
- $(\exists t : t_1 \le t \le t_2)\varphi$ is weakly well-timed if φ is weakly well-timed for some $t_1 \le t \le t_2$
  $(\exists t \ge t')\varphi$ is weakly well-timed if φ is weakly well-timed for some $t \ge t'$
  $(\exists t \le t')\varphi$ is weakly well-timed if φ is weakly well-timed for some $t \le t'$

As with strong well-timedness, where r is clear from context, we often simply say a message is weakly well-timed if it is weakly well-timed with respect to r.

B.2 Three Useful Lemmas

We construct a tree, called a *derivation tree*, corresponding to the derivation of $\mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$ for any set of keys $\mathcal{K}$ and set of messages $\mathcal{M}$, as follows. Nodes of the tree are recursive calls to $\mathrm{submsgs}(\cdot)$; edges are labelled by messages or are unlabelled. For any node $\mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$, there is an edge labelled X to a child $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ for each $X \in \mathcal{M}$. Any node $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}((X_1,\ldots,X_n))$ has an edge labelled X_i to a child $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X_i)$ for $1 \le i \le n$. Any node $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\lfloor X \rfloor_{K^{-1}})$ has an edge labelled X to a child $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$. Any node $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\{X\}_K)$ where $K \in \mathcal{K}$ has an edge labelled X to a child $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$. Any node $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(K)$ where K is a key and $K \notin \mathcal{K}$ has an unlabelled edge to a child $\text{submsgs}_{\mathcal{K}\cup\{K\}}(\mathcal{M})$. (We call this kind of edge a *key-finding edge*.) All other nodes have no children.

It is immediate from the definitions that $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ is equal to the set of labels of the edges of the derivation tree rooted at $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and that $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ is equal to the set of labels of the edges of the derivation tree rooted at $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$. Note also that each node of the derivation tree rooted at $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ is of the form $\text{submsgs}_{\mathcal{K}'}(\mathcal{M})$ or $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(M)$, where $\mathcal{K}' \supseteq \mathcal{K}$ and M is a message.

Lemma B.4 *Let $\mathcal{K}$ and $\mathcal{K}'$ be sets of keys and $\mathcal{M}$ and $\mathcal{M}'$ be sets of messages. If $\mathcal{K} \subseteq \mathcal{K}'$ and $\mathcal{M} \subseteq \mathcal{M}'$, then* $\text{submsgs}_{\mathcal{K}}(\mathcal{M}) \subseteq \text{submsgs}_{\mathcal{K}'}(\mathcal{M}')$.

Proof: Let $\mathcal{K}$ and $\mathcal{K}'$ be sets of keys and $\mathcal{M}$ and $\mathcal{M}'$ be sets of messages and suppose that $\mathcal{K} \subseteq \mathcal{K}'$ and $\mathcal{M} \subseteq \mathcal{M}'$. We start by showing that the derivation tree T of $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and the derivation tree T' of $\text{submsgs}_{\mathcal{K}'}(\mathcal{M}')$ have the following property: given any node $\text{submsgs}_{\mathcal{L}}(\mathcal{N})$in T, there is a corresponding node $\text{submsgs}_{\mathcal{L}'}(\mathcal{N}')$ in T' such that $\mathcal{L} \subseteq \mathcal{L}'$, $\mathcal{N} \subseteq \mathcal{N}'$, and if there is an incoming edge to $\text{submsgs}_{\mathcal{L}}(\mathcal{N})$ labelled with the message X, then there is an incoming edge to $\text{submsgs}_{\mathcal{L}'}(\mathcal{N}')$ labelled with the message X. Similarly, given any node $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(M)$in T, there is a corresponding node $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(M)$ in T' such that $\mathcal{L} \subseteq \mathcal{L}'$, $\mathcal{N} \subseteq \mathcal{N}'$, and if there is an incoming edge to $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(M)$ labelled with the message X, then there is an incoming edge to $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(M)$ labelled with the message X. We prove this by induction on the depth of nodes in T.

As a base case, the root $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ of T and the derivation tree $\text{submsgs}_{\mathcal{K}'}(\mathcal{M}')$ of T' trivially satisfy the property since $\mathcal{K} \subseteq \mathcal{K}'$, $\mathcal{M} \subseteq \mathcal{M}'$, and there are no incoming edges. For the induction step, suppose that the property holds for a node v in T of depth d. We will show the property holds for all children of v. If v has no children, then we are done. Otherwise, v is one of the five forms described above and by the induction hypothesis has a corresponding node v' in T'.

- If $v = \text{submsgs}_{\mathcal{L}}(\mathcal{N})$, then $v' = \text{submsgs}_{\mathcal{L}'}(\mathcal{N}')$ such that $\mathcal{L} \subseteq \mathcal{L}'$ and $\mathcal{N} \subseteq \mathcal{N}'$. Then any child, $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(X)$, of v has an incoming edge labelled X and satisfies $X \in \mathcal{N}$. Thus $X \in \mathcal{N}'$ and therefore v' has a corresponding child $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(X)$ also with an incoming edge labelled X.

- If $v = \text{partsubmsgs}_{\mathcal{L},\mathcal{N}}((X_1, \ldots, X_n))$, then $v' = \text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}((X_1, \ldots, X_n))$, and hence any child $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(X_i)$ with an incoming edge labelled X_i corresponds to a child $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(X_i)$ of v' with an incoming edge labelled X_i.

- If $v = \text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(\lfloor X \rfloor_{K^{-1}})$, then $v' = \text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(\lfloor X \rfloor_{K^{-1}})$, and hence the only child $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(X)$ with an incoming edge labelled X corresponds to a child $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(X)$ of v' with an incoming edge labelled X.

- If $v = \text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(\{X\}_K)$ and $K \in \mathcal{L}$, then $v' = \text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(\{X\}_K)$ and $K \in \mathcal{L}'$ since $\mathcal{L} \subseteq \mathcal{L}'$. Hence the only child $\text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(X)$ with an incoming edge labelled X corresponds to a child $\text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(X)$ of v' with an incoming edge labelled X.

- If $v = \text{partsubmsgs}_{\mathcal{L},\mathcal{N}}(K)$ and $K \notin \mathcal{L}$, then $v' = \text{partsubmsgs}_{\mathcal{L}',\mathcal{N}'}(K)$. The only child of v is $w = \text{submsgs}_{\mathcal{L}\cup\{K\}}(\mathcal{N})$. If $K \notin \mathcal{L}'$, then v' has a child $\text{submsgs}_{\mathcal{L}'\cup\{K\}}(\mathcal{N}')$ corresponding to w. Otherwise, $K \in \mathcal{L}'$. Consider the path in T' from v' to the root, then the previous submsgs() node before v' on the path must $\text{submsgs}_{\mathcal{L}'}(\mathcal{N}')$ (and such a node must exist because the root itself is a submsgs() node). Since $\mathcal{L} \subseteq \mathcal{L}'$ and $K \in \mathcal{L}'$, it follows that $\mathcal{L} \cup \{K\} \subseteq \mathcal{L}'$, and hence this node corresponds to w.

Hence, the desired correspondence between T and T' holds. Thus, for every labelled edge in T, there is an edge with the same label in T'. It follows that $\text{submsgs}_{\mathcal{K}}(\mathcal{M}) \subseteq \text{submsgs}_{\mathcal{K}'}(\mathcal{M}')$, as desired. ∎

Lemma B.5 *Let $\mathcal{K}$ be a set of keys, $\mathcal{M}$ be a set of messages, and X be a message. If $\{X\}_K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and $K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$, then $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.*

Proof: Suppose $\{X\}_K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ and $K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Because $\{X\}_K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$, there is a path p from the node $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ to a node $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(\{X\}_K)$ such that $\mathcal{K}' \supseteq \mathcal{K}$. If $K \in \mathcal{K}$, it follows that $K \in \mathcal{K}'$, so this node has an edge labelled X to the node $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$. It follows that $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.

Otherwise, $K \notin \mathcal{K}$. Because $K \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$, there is a path from the node $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ to some node $\text{partsubmsgs}_{\mathcal{K}'',\mathcal{M}}(K)$ such that $\mathcal{K}'' \supseteq \mathcal{K}$ and $K \notin \mathcal{K}''$. (Note that it may also be possible that some such nodes exist where $K \in \mathcal{K}''$, but that since $K \notin \mathcal{K}$, there will always be some such node with $K \notin \mathcal{K}''$.) This node has a child $\text{submsgs}_{\mathcal{K}''\cup\{K\}}(\mathcal{M})$. This node, in turn, has a path to $\text{partsubmsgs}_{\mathcal{K}''\cup\mathcal{K}'\cup\{K\},\mathcal{M}}(\{X\}_K)$ that corresponds to the path p above except that the key sets possibly contain additional keys from $\mathcal{K}''\cup\{K\}$. Since $\text{partsubmsgs}_{\mathcal{K}''\cup\mathcal{K}'\cup\{K\},\mathcal{M}}(X)$ has an outgoing edge labelled X, it follows that $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. ■

Lemma B.6 *Let $\mathcal{K}$ be a set of keys, $\mathcal{M}$ be a set of messages, and X be a message. If $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$, then $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.*

Proof: If $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$, then there is an edge labelled X to a child $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$ in the derivation tree rooted at $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ for some $\mathcal{K}' \supseteq \mathcal{K}$. It follows that $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.

We now show three facts. First, we show $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X) \subseteq \text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X) \cup \text{submsgs}_{\mathcal{K}'}(\mathcal{M})$. This holds because $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X) \subseteq \text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$ unless there is key-finding edge in the derivation tree for $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X)$ that is not in the derivation tree for $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$. In this exceptional case, the node the key-finding leads to must be $\text{submsgs}_{\mathcal{K}\cup\{K\}}(\mathcal{M})$ for a key $K \in \mathcal{K}' \setminus \mathcal{K}$. Then, since $\mathcal{K} \subseteq \mathcal{K}'$ and $K \in \mathcal{K}'$, we have $\mathcal{K} \cup \{K\} \subseteq \mathcal{K}'$, so by Lemma B.4, $\text{submsgs}_{\mathcal{K}\cup\{K\}}(\mathcal{M}) \subseteq \text{submsgs}_{\mathcal{K}'}(\mathcal{M})$. Hence, in every case, $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X) \subseteq \text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X) \cup \text{submsgs}_{\mathcal{K}'}(\mathcal{M})$.

Second, since $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$ is a node in the derivation tree of $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$, it follows that $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.

Third, we show that $\text{submsgs}_{\mathcal{K}'}(\mathcal{M}) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Since $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$ is a node in the derivation tree of $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$, it follows that $\text{submsgs}_{\mathcal{K}'}(\mathcal{M})$ must appear somewhere on the path from the root $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$ to the node $\text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X)$. Hence, $\text{submsgs}_{\mathcal{K}'}(\mathcal{M})$ is itself a node in the derivation tree of $\text{submsgs}_{\mathcal{K}}(\mathcal{M})$, so $\text{submsgs}_{\mathcal{K}'}(\mathcal{M}) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$.

Putting these three facts together,

$$\begin{aligned}
\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(X) &\subseteq \text{partsubmsgs}_{\mathcal{K}',\mathcal{M}}(X) \cup \text{submsgs}_{\mathcal{K}'}(\mathcal{M}) \\
&\subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M}) \cup \text{submsgs}_{\mathcal{K}'}(\mathcal{M}) \\
&= \text{submsgs}_{\mathcal{K}}(\mathcal{M})
\end{aligned}$$

as desired. ■

B.3 Soundness of Axioms

For each axiom, we show soundness with respect to truth (if $r \models \varphi$, then $r \models \psi$) and with respect to weak well-timedness (if φ is weakly well-timed with respect to r then ψ is weakly well-timed with respect to r). In addition, for axioms in $\mathcal{L}$, we show soundness with respect to strong well-timedness (if φ is strongly well-timed with respect to r then ψ is strongly well-timed with respect to r). In most cases, we are able to show the well-timed and the truth requirements for each axiom separately. The only exceptions are some of the parts of axiom A5, in which the truth assumption $r \models \varphi$ is also needed in order to prove the well-timed requirement for ψ. For ease of presentation, we divide the proofs into these two or three parts.

Belief

A1. Belief and logical connective axioms.

a. P *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ $(\varphi \supset \psi) \supset P$ *believes*$_t$ ψ

Proof: Note that in our primitive notation $\varphi \supset \psi$ is $\neg(\varphi \wedge \neg\psi)$.
Truth: Suppose $r \models P$ *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ $\neg(\varphi \wedge \neg\psi)$. Then, by definition, $r \models P$ *believes*$_t$ φ and $r \models P$ *believes*$_t$ $\neg(\varphi \wedge \neg\psi)$. Let r' be such that $(r,s) \sim_P (r',s')$ for every $s \in \text{realtime}_S(r,t)$ and

$s' \in \text{realtime}_S(r',t)$. Then it follows from the semantics of *believes* that $r' \models \varphi$ and $r' \models \neg(\varphi \wedge \neg\psi)$. Hence $r' \not\models \varphi \wedge \neg\psi$, so either $r' \not\models \varphi$ or $r' \not\models \neg\psi$. Since in fact $r' \models \varphi$, it follows that $r' \not\models \neg\psi$, so $r' \models \psi$ (since if $r' \not\models \psi$ then $r' \models \neg\psi$, a contradiction). Hence, P *believes*$_t$ ψ.

Well-timedness: Suppose P *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ $\neg(\varphi \wedge \neg\psi)$ is strongly (resp. weakly) well-timed with respect to r. Then P *believes*$_t$ φ is strongly (resp. weakly) well-timed and P *believes*$_t$ $\neg(\varphi \wedge \neg\psi)$ is strongly (resp. weakly) well-timed. It follows that φ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\varphi)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\varphi)$), and $\neg(\varphi \wedge \neg\psi)$ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\neg(\varphi \wedge \neg\psi))$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\neg(\varphi \wedge \neg\psi))$). It further follows that ψ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\psi)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\psi)$), and therefore P *believes*$_t$ ψ is strongly (resp. weakly) well-timed. ■

b. P *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ $\psi \supset P$ *believes*$_t$ $(\varphi \wedge \psi)$

Proof: *Truth:* Suppose $r \models P$ *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ ψ. Then $r \models P$ *believes*$_t$ φ and $r \models P$ *believes*$_t$ ψ. This implies $r' \models \varphi$ and $r' \models \psi$ for all r' as specified in the truth conditions for *believes*. Thus, $r' \models \varphi \wedge \psi$ for all such r', so $r \models P$ *believes*$_t$ $(\varphi \wedge \psi)$.

Well-timedness: If P *believes*$_t$ $\varphi \wedge P$ *believes*$_t$ ψ is strongly (resp. weakly) well-timed with respect to r, then P *believes*$_t$ φ is strongly (resp. weakly) well-timed and P *believes*$_t$ ψ is strongly (resp. weakly) well-timed. Hence, φ and ψ are strongly (resp. weakly) well-timed, $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\varphi)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\varphi)$), and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\psi)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\psi)$). It follows that $\varphi \wedge \psi$ is strongly (resp. weakly) well-timed and that $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\varphi) \cup \text{time}_r(\psi) = \text{time}_r(\varphi \wedge \psi)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\varphi) \cup \text{time}_r(\psi) = \text{time}_r(\varphi \wedge \psi)$). Therefore, P *believes*$_t$ $(\varphi \wedge \psi)$ is strongly (resp. weakly) well-timed. ■

A2. P *believes*$_t$ $\varphi \equiv P$ *believes*$_t$ P *believes*$_t$ φ

Proof: *Truth:* Let $\mathcal{R}' = \{r' : \forall s \in \text{realtime}_P(r,t), s' \in \text{realtime}_P(r',t), (r,s) \sim_P (r',s')\}$, and let $\mathcal{R}'' = \{r'' : \forall r' \in \mathcal{R}', s' \in \text{realtime}_P(r',t), s'' \in \text{realtime}_P(r'',t)(r',s') \sim_P (r'',s'')\}$. Then, in fact $\mathcal{R}' = \mathcal{R}''$, so

$$\begin{aligned}
& r \models P \textit{ believes}_t\ \varphi \\
\iff & r' \models \varphi \text{ for all } r' \in \mathcal{R}' \\
\iff & r'' \models \varphi \text{ for all } r'' \in \mathcal{R}'' \\
\iff & r' \models P \textit{ believes}_t\ \varphi \text{ for all } r' \in \mathcal{R}' \\
\iff & r \models P \textit{ believes}_t\ P \textit{ believes}_t\ \varphi
\end{aligned}$$

as desired.

Well-timedness: Note that $\text{realtime}_P(r,t) = \text{time}_r(P$ *believes*$_t$ $\varphi)$. Hence, it follows that $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(P$ *believes*$_t$ $\varphi)$ and $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(P$ *believes*$_t$ $\varphi)$. Thus by the definition of strongly (resp. weakly) well-timed, P *believes*$_t$ φ is strongly (resp. weakly) well-timed iff P *believes*$_t$ P *believes*$_t$ φ is strongly (resp. weakly) well-timed. ■

Time and Synchronization

A3. Monotonicity axioms

a. P *received*$_t$ $X \wedge t' \geq t \supset P$ *received*$_{t'}$ X

Proof: *Truth:* Suppose $r \models P$ *received*$_t$ $X \wedge t' \geq t$ for a legal run r. Then $r \models P$ *received*$_t$ X and $r \models t' \geq t$. Hence, there exists $\hat{t} \in \text{realtime}_P(r,t)$ such that $X \in \text{submsgs}_{\text{skeys}_P(r,\hat{t})}(\text{rcvdmsgs}_P(r,\hat{t}))$. Let $\hat{t}' \in \text{realtime}_P(r,t')$. Then $\text{clock}_P(r,\hat{t}) = t$ and $\text{clock}_P(r,\hat{t}') = t'$. Since r is legal, and $r \models t' \geq t$, it follows that $\hat{t} \leq \hat{t}'$. It further follows from the legality of r that $\text{skeys}_P(r,\hat{t}) \subseteq \text{skeys}_P(r,\hat{t}')$ and $\text{rcvdmsgs}_P(r,\hat{t}) \subseteq \text{rcvdmsgs}_P(r,\hat{t}')$. Thus by Lemma B.4, $\text{submsgs}_{\text{skeys}_P(r,\hat{t})}(\text{rcvdmsgs}_P(r,\hat{t})) \subseteq \text{submsgs}_{\text{skeys}_P(r,\hat{t}')}(\text{rcvdmsgs}_P(r,\hat{t}'))$. Hence $r \models P$ *received*$_{t'}$ X.

Well-timedness: Suppose P *received*$_t$ $X \wedge t' \geq t$ is strongly (resp. weakly) well-timed with respect to r. Then P *received*$_t$ X is strongly (resp. weakly) well-timed and $t' \geq t$ is strongly (resp. weakly) well-timed. Thus X is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Also, $t' \geq t$, so $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{realtime}_P(r,t)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{realtime}_P(r,t)$). It follows that $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{time}_r(X)$), so P *received*$_{t'}$ X is strongly (resp. weakly) well-timed. ■

b. P *said*$_t$ $X \wedge t' \geq t \supset P$ *said*$_{t'}$ X

Proof: *Truth:* Suppose $r \models P$ *said*$_t$ $X \wedge t' \geq t$ for a legal run r. Then $r \models P$ *said*$_t$ X and $r \models t' \geq t$. Hence, there exists $\hat{t} \in \mathrm{realtime}_P(r,t)$ such that $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,\hat{t})}(\mathrm{sentmsgs}_P(r,\hat{t}))$. Let $\hat{t}' \in \mathrm{realtime}_P(r,t')$. Then $\mathrm{clock}_P(r,\hat{t}) = t$ and $\mathrm{clock}_P(r,\hat{t}') = t'$. Since r is legal, and $r \models t' \geq t$, it follows that $\hat{t} \leq \hat{t}'$. It further follows from the legality of r that $\mathrm{skeys}_P(r,\hat{t}) \subseteq \mathrm{skeys}_P(r,\hat{t}')$ and $\mathrm{sentmsgs}_P(r,\hat{t}) \subseteq \mathrm{sentmsgs}_P(r,\hat{t}')$. Furthermore, $\mathrm{submsgs}_{\mathrm{skeys}_P(r,\hat{t})}(\mathrm{sentmsgs}_P(r,\hat{t})) \subseteq \mathrm{submsgs}_{\mathrm{skeys}_P(r,\hat{t}')}(\mathrm{sentmsgs}_P(r,\hat{t}'))$ By Lemma B.4. Hence $r \models P$ *said*$_{t'}$ X.

Well-timedness: Suppose that P *said*$_t$ $X \wedge t' \geq t$ is strongly (resp. weakly) well-timed with respect to r. Then P *said*$_t$ X is strongly (resp. weakly) well-timed and $t' \geq t$ is strongly (resp. weakly) well-timed. Thus X is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Also, $t' \geq t$, so $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{realtime}_P(r,t)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{realtime}_P(r,t)$). It follows that $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{time}_r(X)$), so P *said*$_{t'}$ X is strongly (resp. weakly) well-timed. ■

c. P *has*$_t$ $K \wedge t \leq t' \supset P$ *has*$_{t'}$ K

Proof: *Truth:* Suppose $r \models P$ *has*$_t$ $K \wedge t' \geq t$ for a legal run r. Then $r \models P$ *has*$_t$ K and $r \models t' \geq t$. Hence, there exists $\hat{t} \in \mathrm{realtime}_P(r,t)$ such that $X \in \mathrm{ekeys}_P(r,\hat{t})$. Let $\hat{t}' \in \mathrm{realtime}_P(r,t')$. Then $\mathrm{clock}_P(r,\hat{t}) = t$ and $\mathrm{clock}_P(r,\hat{t}') = t'$. Since r is legal, and $r \models t' \geq t$, it follows that $\hat{t} \leq \hat{t}'$. It further follows from the legality of r that $\mathrm{ekeys}_P(r,\hat{t}) \subseteq \mathrm{ekeys}_P(r,\hat{t}')$. Hence $r \models P$ *has*$_{t'}$ K.

Well-timedness: P *has*$_{t'}$ K is strongly and weakly well-timed. ■

d. *fresh*$_{t,P}$ $X \wedge t' \leq t \supset$ *fresh*$_{t',P}$ X

Proof: *Truth:* Suppose that $r \models$ *fresh*$_{t,P}$ $X \wedge t' \leq t$. Then $r \models$ *fresh*$_{t,P}$ X and $r \models t' \leq t$. It follows that $r \not\models Q$ *said*$_{t,P}$ X for all $Q \in \mathcal{P}$. By the soundness of A3, part (b), above, it follows that $r \not\models Q$ *said*$_{t',P}$ X for all $Q \in \mathcal{P}$. Therefore, $r \models$ *fresh*$_{t',P}$ X.

Well-timedness: Suppose that *fresh*$_{t,P}$ $X \wedge t' \geq t$ is strongly (resp. weakly) well-timed with respect to r. Then *fresh*$_{t,P}$ X is strongly (resp. weakly) well-timed and $t' \geq t$ is strongly (resp. weakly) well-timed. Thus X is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Also, $t' \geq t$, so $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{realtime}_P(r,t)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{realtime}_P(r,t)$). It follows that $\mathrm{realtime}_P(r,t') \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t') \trianglerighteq_w \mathrm{time}_r(X)$), so *fresh*$_{t',P}$ X is strongly (resp. weakly) well-timed. ■

e. ϵ-*sync*$_t$ $(P,Q) \wedge \epsilon \leq \epsilon' \supset \epsilon'$-*sync*$_t$ (P,Q)

Proof: *Truth:* Suppose $r \models \epsilon$-*sync*$_{t,S}$ (P,Q). Then

(a) $\forall t' \in \mathrm{realtime}_S(r,t)$, $|\mathrm{clock}_P(r,t') - \mathrm{clock}_Q(r,t')| \leq \epsilon$,

(b) there is some k such that $\forall \widehat{t}$, $|\{s \in \mathrm{realtime}_P(r,\widehat{t})\}| = |\{s \in \mathrm{realtime}_Q(r,\widehat{t})\}| = k$

(c) $\forall \widehat{t}$, $\widehat{t} - \mathrm{prevtick}_P(r,\widehat{t}) \leq \epsilon$

(d) $\forall \widehat{t}$, $\widehat{t} - \mathrm{prevtick}_Q(r,\widehat{t}) \leq \epsilon$

Since $\epsilon \leq \epsilon'$, it follows that

(a′) $\forall t' \in \mathrm{realtime}_S(r,t)$, $|\mathrm{clock}_P(r,t') - \mathrm{clock}_Q(r,t')| \leq \epsilon'$,

(c′) $\forall \widehat{t}$, $\widehat{t} - \mathrm{prevtick}_P(r,\widehat{t}) \leq \epsilon'$

(d′) $\forall \widehat{t}$, $\widehat{t} - \mathrm{prevtick}_Q(r,\widehat{t}) \leq \epsilon'$

Hence $r \models \epsilon'\text{-}sync_t\ (P, Q)$.

Well-timedness: $\epsilon'\text{-}sync_t\ (P, Q)$ is strongly and weakly well-timed. ■

A4. Synchronization axioms

a. $\epsilon\text{-}sync_t\ (P, Q) \supset \epsilon\text{-}sync_{\langle t-\epsilon, t+\epsilon\rangle}\ (Q, P)$

Proof: *Truth:* Suppose $r \models \epsilon\text{-}sync_t\ (P, Q)$. We wish to show that $r \models \epsilon\text{-}sync_{\langle t-\epsilon,t+\epsilon\rangle}\ (Q, P)$. Since conditions (2)-(4) of ϵ-*sync* are properties of r (that don't depend on t) and are symmetric with regards to P and Q, we need only show that condition (1) is satisfied. That is, it suffices to show that there is some t_q such that $t - \epsilon \le t_q \le t + \epsilon$ and $r \models \epsilon\text{-}sync_{t_q}\ (Q, P)$.

To this end, let $t' \in \text{realtime}_P(r, t)$, let $t_q = \text{clock}_Q(r, t')$, and let $s_q \in \text{realtime}_Q(r, t_q)$. Then $|\,\text{clock}_P(r, t') - \text{clock}_Q(r, t')| \le \epsilon$, so $t - \epsilon \le t_q \le t + \epsilon$. By conditions (2)-(4), we can assume without loss of generality $t_q \le t$ and $\text{clock}_P(r, s_q) \le t$. Hence $t - \epsilon \le t_q \le t$ and $t - \epsilon \le \text{clock}_P(r, s_q) \le t$. It follows that $|t_q - \text{clock}_P(r, s_q)| \le \epsilon$, so $r \models \epsilon\text{-}sync_{t_q}\ (Q, P)$.

Well-timedness: $\epsilon\text{-}sync_{\langle t-\epsilon,t+\epsilon\rangle}\ (Q, P)$ is strongly and weakly well-timed. ■

b. $\epsilon_1\text{-}sync_{t,S}\ (P, Q) \wedge \epsilon_2\text{-}sync_{t,S}\ (Q, R) \supset (\epsilon_1 + \epsilon_2)\text{-}sync_{t,S}\ (P, R)$

Proof: *Truth:* Suppose $r \models \epsilon_1\text{-}sync_{t,S}\ (P, Q)$ and $r \models \epsilon_2\text{-}sync_{t,S}\ (Q, R)$. Then

(a) $\forall t' \in \text{realtime}_S(r, t)$, $|\,\text{clock}_P(r, t') - \text{clock}_Q(r, t')| \le \epsilon_1$,

(b) there is some k such that $\forall \hat{t}$, $|\{s \in \text{realtime}_P(r, \hat{t})\}| = |\{s \in \text{realtime}_Q(r, \hat{t})\}| = k$

(c) $\forall \hat{t}$, $\hat{t} - \text{prevtick}_P(r, \hat{t}) \le \epsilon_1$

(d) $\forall \hat{t}$, $\hat{t} - \text{prevtick}_Q(r, \hat{t}) \le \epsilon_1$

and

(a$'$) $\forall t'' \in \text{realtime}_S(r, t)$, $|\,\text{clock}_Q(r, t'') - \text{clock}_R(r, t'')| \le \epsilon_2$,

(b$'$) there is some k' such that $\forall \hat{t}$, $|\{s \in \text{realtime}_Q(r, \hat{t})\}| = |\{s \in \text{realtime}_R(r, \hat{t})\}| = k'$

(c$'$) $\forall \hat{t}$, $\hat{t} - \text{prevtick}_Q(r, \hat{t}) \le \epsilon_2$

(d$'$) $\forall \hat{t}$, $\hat{t} - \text{prevtick}_R(r, \hat{t}) \le \epsilon_2$

We wish to show that the analogous four conditions hold for $\epsilon_1 + \epsilon_2$, P, and R. First, by (a) and (a$'$), it follows that $\forall t'' \in \text{realtime}_S(r, t)$, $|\,\text{clock}_Q(r, t'') - \text{clock}_R(r, t'')| \le \epsilon_1 + \epsilon_2$. Second, by (b) and (b'), it follows that $k = k'$. Hence, k satisfies the desired requirement: $\forall \hat{t}$, $|\{s \in \text{realtime}_P(r, \hat{t})\}| = |\{s \in \text{realtime}_R(r, \hat{t})\}| = k$. Finally, (c) implies $\forall \hat{t}$, $\hat{t} - \text{prevtick}_P(r, \hat{t}) \le \epsilon_1 \le \epsilon_1 + \epsilon_2$, and (d') implies $\forall \hat{t}$, $\hat{t} - \text{prevtick}_R(r, \hat{t}) \le \epsilon_2 \le \epsilon_1 + \epsilon_2$. Therefore, $r \models (\epsilon_1 + \epsilon_2)\text{-}sync_{t,S}\ (P, R)$.

Well-timedness: $(\epsilon_1 + \epsilon_2)\text{-}sync_{t,S}\ (P, R)$ is strongly and weakly well-timed. ■

A5. Time Conversion Axioms

a. $P\ \ received_{t,S}\ X \wedge \epsilon\text{-}sync_t\ (S, R) \supset P\ \ received_{\langle t-\epsilon,t+\epsilon\rangle,R}\ X$

Proof: *Truth:* Suppose $r \models P\ \ received_{t,S}\ X \wedge \epsilon\text{-}sync_t\ (S, R)$. Then $r \models P\ \ received_{t,S}\ X$ and $r \models \epsilon\text{-}sync_t\ (S, R)$. Thus, there exists $t' \in \text{realtime}_S(r, t)$ such that $X \in \text{submsgs}_{\text{skeys}_P(r,t')}(\text{rcvdmsgs}_P(r, t'))$. It follows that $r \models P\ \ received_{\text{clock}_R(r,t'),R}\ X$. In addition, $|\,\text{clock}_S(r, t') - \text{clock}_R(r, t')| \le \epsilon$, so $t - \epsilon \le \text{clock}_R(r, t') \le t + \epsilon$. Hence, $r \models P\ \ received_{\langle t+\epsilon,t-\epsilon\rangle,R}\ X$.

Weak well-timedness: Suppose that $P\ \ received_{t,S}\ X \wedge \epsilon\text{-}sync_t\ (S, R)$ is weakly well-timed with respect to r, and further suppose that $r \models P\ \ received_{t,S}\ X \wedge \epsilon\text{-}sync_t\ (S, R)$. Then, in particular, $P\ \ received_{t,S}$ X is weakly well-timed with respect to r and $r \models \epsilon\text{-}sync_t\ (S, R)$. It follows that X is weakly well-timed and $\text{realtime}_S(r, t) \trianglerighteq_w \text{time}_r(X)$. Take $t' = \max_u : u \in \text{realtime}_S(r, t)$. Then, since $r \models \epsilon\text{-}sync_t\ (S, R)$, it follows that $t - \epsilon \le \text{clock}_R(r, t') \le t + \epsilon$ and $\text{realtime}_R(r, \text{clock}_R(r, t')) \trianglerighteq_w \text{realtime}_S(r, t)$. Hence $\text{realtime}_R(r, \text{clock}_R(r, t')) \trianglerighteq_w \text{time}_r(X)$, and it follows that $P\ \ received_{\langle t-\epsilon,t+\epsilon\rangle,R}\ X$ is weakly well-timed with respect to r. ■

b. $P\ \ says_{t,S}\ X \wedge \epsilon\text{-}sync_t\ (S, R) \supset P\ \ says_{\langle t-\epsilon,t+\epsilon\rangle,R}\ X$

Proof: *Truth:* Suppose $r \models P\ \ says_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then $r \models P\ \ says_{t,S}\ \ X$ and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. Thus, there is some message M and some $t' \in \mathrm{realtime}_S(r,t)$ such that $(\mathrm{send}(M), t') \in \mathrm{history}_P(r,t')$ and $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\{M\})$. Hence, $r \models P\ \ says_{\mathrm{clock}_R(r,t'),R}\ \ X$. In addition, $|\,\mathrm{clock}_S(r,t') - \mathrm{clock}_R(r,t')| \leq \epsilon$, so $t - \epsilon \leq \mathrm{clock}_R(r,t') \leq t + \epsilon$. Hence, $r \models P\ \ says_{\langle t+\epsilon, t-\epsilon\rangle,R}\ \ X$.

Weak well-timedness: Suppose that $P\ \ says_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$ is weakly well-timed with respect to r, and further suppose that $r \models P\ \ says_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then, in particular, $P\ \ says_{t,S}\ \ X$ is weakly well-timed with respect to r and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. It follows that X is weakly well-timed and $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(X)$. Take $t' = \max_u : u \in \mathrm{realtime}_S(r,t)$. Then, since $r \models \epsilon\text{-}sync_t\ \ (S,R)$, it follows that $t - \epsilon \leq \mathrm{clock}_R(r,t') \leq t + \epsilon$ and $\mathrm{realtime}_R(r, \mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{realtime}_S(r,t)$. Hence $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{time}_r(X)$, and it follows that $P\ \ says_{\langle t-\epsilon, t+\epsilon\rangle,R}\ \ X$ is weakly well-timed with respect to r. ■

c. $P\ \ said_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R) \supset P\ \ said_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ X$

Proof: *Truth:* Suppose that $r \models P\ \ said_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then $r \models P\ \ said_{t,S}\ \ X$ and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. Thus there exists $t' \in \mathrm{realtime}_S(r,t)$ such that $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{sentmsgs}_P(r,t'))$. It follows that $r \models P\ \ said_{\mathrm{clock}_R(r,t'),R}\ \ X$. In addition, $|\,\mathrm{clock}_S(r,t') - \mathrm{clock}_R(r,t')| \leq \epsilon$, so $t - \epsilon \leq \mathrm{clock}_R(r,t') \leq t + \epsilon$. Hence, $r \models P\ \ said_{\langle t+\epsilon,t-\epsilon\rangle,R}\ \ X$.

Weak well-timedness: Suppose that $P\ \ said_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$ is weakly well-timed with respect to r, and further suppose that $r \models P\ \ said_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then, in particular, $P\ \ said_{t,S}\ \ X$ is weakly well-timed with respect to r and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. It follows that X is weakly well-timed and $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(X)$. Take $t' = \max_u : u \in \mathrm{realtime}_S(r,t)$. Then, since $r \models \epsilon\text{-}sync_t\ \ (S,R)$, it follows that $t - \epsilon \leq \mathrm{clock}_R(r,t') \leq t + \epsilon$ and $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{realtime}_S(r,t)$. Hence $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{time}_r(X)$, and it follows that $P\ \ said_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ X$ is weakly well-timed with respect to r. ■

d. $p\ \ at_S\ \ t \wedge \epsilon\text{-}sync_t\ \ (S,R) \supset p\ \ at_R\ \ \langle t-\epsilon, t+\epsilon\rangle$

Proof: *Truth:* Suppose $r \models p\ \ at_S\ \ t \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then $r \models p\ \ at_S\ \ t$ and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. By the truth conditions for *at*, this implies that there exists $t' \in \mathrm{realtime}_S(r,t)$ such that $\pi((r,t'),p) = \mathrm{T}$. Then, since by definition $t' \in \mathrm{realtime}_R(r,\mathrm{clock}_R(r,t'))$, it follows that $r \models p\ \ at_R\ \ \mathrm{clock}_R(r,t')$. In addition, the truth conditions for *sync* imply that $|\,\mathrm{clock}_S(r,t') - \mathrm{clock}_R(r,t')| \leq \epsilon$. Therefore, $t-\epsilon \leq \mathrm{clock}_R(r,t') \leq t+\epsilon$, so it follows that $r \models p\ \ at_R\ \ \langle t+\epsilon, t-\epsilon\rangle$.

Well-timedness: $p\ \ at_R\ \ \langle t-\epsilon, t+\epsilon\rangle$ is strongly and weakly well-timed. ■

e. $P\ \ has_{t,S}\ \ K \wedge \epsilon\text{-}sync_t\ \ (S,R) \supset P\ \ has_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ K$

Proof: *Truth:* Suppose $r \models P\ \ has_{t,S}\ \ X \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then $r \models P\ \ has_{t,S}\ \ X$ and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. Thus, there exists $t' \in \mathrm{realtime}_S(r,t)$ such that $K \in \mathrm{ekeys}_P(r,t')$. Hence, $r \models P\ \ has_{\mathrm{clock}_R(r,t'),R}\ \ X$. In addition, $|\,\mathrm{clock}_S(r,t') - \mathrm{clock}_R(r,t')| \leq \epsilon$, so $t-\epsilon \leq \mathrm{clock}_R(r,t') \leq t+\epsilon$. Hence, $r \models P\ \ has_{\langle t+\epsilon,t-\epsilon\rangle,R}\ \ X$.

Well-timedness: $P\ \ has_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ K$ is strongly and weakly well-timed. ■

f. $P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi \wedge \epsilon\text{-}sync_t\ \ (S,R) \supset P\ \ believes_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ \varphi$

Proof: *Truth:* Suppose that $r \models P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then $r \models P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi$ and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. By the semantics of *sync*, it follows that there exists s such that $\mathrm{realtime}_R(r,s) \subseteq \mathrm{realtime}_S(r,t) \cup \mathrm{realtime}_S(r,\mathrm{prevtick}_S(r,t))$. Furthermore, $|t-s| \leq \epsilon$. It follows from the semantics of *believes* that $P\ \ believes_{s,R}\ \ \varphi$, and therefore $P\ \ believes_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ \varphi$.

Weak well-timedness: Suppose that $P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi \wedge \epsilon\text{-}sync_t\ \ (S,R)$ is weakly well-timed with respect to r, and further suppose that $r \models P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi \wedge \epsilon\text{-}sync_t\ \ (S,R)$. Then, in particular, $P\ \ believes_{[t-\epsilon,t],S}\ \ \varphi$ is weakly well-timed with respect to r and $r \models \epsilon\text{-}sync_t\ \ (S,R)$. It follows that X is weakly well-timed and $\mathrm{realtime}_S(r,[t-\epsilon,t]) \trianglerighteq_w \mathrm{time}_r(X)$. Hence also $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(X)$. Take $t' = \max_u : u \in \mathrm{realtime}_S(r,t)$. Then, since $r \models \epsilon\text{-}sync_t\ \ (S,R)$, it follows that $t-\epsilon \leq \mathrm{clock}_R(r,t') \leq t+\epsilon$ and $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{realtime}_S(r,t)$. Hence $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{time}_r(X)$, and it follows that $P\ \ believes_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \ \varphi$ is weakly well-timed with respect to r. ■

g. $P \ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R) \supset P\ \mathit{controls}_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \varphi$

Proof: *Truth:* Suppose that $r \models P\ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$. Then $r \models P\ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi$ and $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$. By the semantics of *sync*, it follows that there exists s such that $\mathrm{realtime}_R(r,s) \subseteq \mathrm{realtime}_S(r,t) \cup \mathrm{realtime}_S(r,\mathrm{prevtick}_S(r,t))$. Furthermore, $|t-s| \leq \epsilon$. It follows from the semantics of *controls* that $P\ \mathit{controls}_{s,R}\ \varphi$, and therefore $P\ \mathit{controls}_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \varphi$.

Weak well-timedness: Suppose that $P\ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$ is weakly well-timed with respect to r, and further suppose that $r \models P\ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$. Then, in particular, $P\ \mathit{controls}_{[t-\epsilon,t],S}\ \varphi$ is weakly well-timed with respect to r and $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$. It follows that X is weakly well-timed and $\mathrm{realtime}_S(r,[t-\epsilon,t]) \trianglerighteq_w \mathrm{time}_r(X)$. Hence also $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(X)$. Take $t' = \max_u : u \in \mathrm{realtime}_S(r,t)$. Then, since $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$, it follows that $t-\epsilon \leq \mathrm{clock}_R(r,t') \leq t+\epsilon$ and $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{realtime}_S(r,t)$. Hence $\mathrm{realtime}_R(r,\mathrm{clock}_R(r,t')) \trianglerighteq_w \mathrm{time}_r(X)$, and it follows that $P\ \mathit{controls}_{\langle t-\epsilon,t+\epsilon\rangle,R}\ \varphi$ is weakly well-timed with respect to r. ■

h. $P \overset{K}{\leftrightarrow}_{[t-\epsilon,t],S} Q \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R) \supset P \overset{K}{\leftrightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} Q$

Proof: *Truth:* Suppose that $r \models P \overset{K}{\leftrightarrow}_{[t-\epsilon,t],S} Q \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$. Then $r \models P \overset{K}{\leftrightarrow}_{[t-\epsilon,t],S} Q$ and $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$. By the semantics of *sync*, it follows that there exists an s such that $\mathrm{realtime}_R(r,s) \subseteq \mathrm{realtime}_S(r,t) \cup \mathrm{realtime}_S(r,\mathrm{prevtick}_S(r,t))$. Furthermore, $|t-s| \leq \epsilon$. It follows from the semantics of $\leftrightarrow$ that $P \overset{K}{\leftrightarrow}_{s,R} Q$, and therefore $P \overset{K}{\leftrightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} Q$.

Well-timedness: $P \overset{K}{\leftrightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} Q$ is strongly and weakly well-timed. ■

i. $\overset{K}{\Rightarrow}_{[t-\epsilon,t],S} P \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R) \supset \overset{K}{\Rightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$

Proof: *Truth:* Suppose that $r \models \overset{K}{\Rightarrow}_{[t-\epsilon,t],S} P \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$. Then $r \models \overset{K}{\Rightarrow}_{[t-\epsilon,t],S} P$ and $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$. By the semantics of *sync*, it follows that there exists an s such that $\mathrm{realtime}_R(r,s) \subseteq \mathrm{realtime}_S(r,t) \cup \mathrm{realtime}_S(r,\mathrm{prevtick}_S(r,t))$. Furthermore, $|t-s| \leq \epsilon$. It follows from the semantics of $\Rightarrow$ that $\overset{K}{\Rightarrow}_{s,R} P$, and therefore $\overset{K}{\Rightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$.

Well-timedness: $\overset{K}{\Rightarrow}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$ is strongly and weakly well-timed. ■

j. $\overset{K}{\mapsto}_{[t-\epsilon,t],S} P \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R) \supset \overset{K}{\mapsto}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$

Proof: *Truth:* Suppose that $r \models \overset{K}{\mapsto}_{[t-\epsilon,t],S} P \wedge \epsilon\text{-}\mathit{sync}_t\ (S,R)$. Then $r \models \overset{K}{\mapsto}_{[t-\epsilon,t],S} P$ and $r \models \epsilon\text{-}\mathit{sync}_t\ (S,R)$. By the semantics of *sync*, it follows that there exists an s such that $\mathrm{realtime}_R(r,s) \subseteq \mathrm{realtime}_S(r,t) \cup \mathrm{realtime}_S(r,\mathrm{prevtick}_S(r,t))$. Furthermore, $|t-s| \leq \epsilon$. It follows from the semantics of $\mapsto$ that $\overset{K}{\mapsto}_{s,R} P$, and therefore $\overset{K}{\mapsto}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$.

Well-timedness: $\overset{K}{\mapsto}_{\langle t-\epsilon,t+\epsilon\rangle,R} P$ is strongly and weakly well-timed. ■

Originator Identification

A6. $\overset{K}{\Rightarrow}_{t,S} Q \wedge R\ \mathit{received}_{t,S}\ \lfloor X \rfloor_{K^{-1}} \supset Q\ \mathit{said}_{t,S}\ X \wedge Q\ \mathit{said}_{t,S}\ \lfloor X \rfloor_{K^{-1}} \wedge Q\ \mathit{has}_{t,S}\ K^{-1}$

Proof: *Truth:* Suppose that $r \models \overset{K}{\Rightarrow}_{t,S} Q \wedge R\ \mathit{received}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$. Then $r \models \overset{K}{\Rightarrow}_{t,S} Q$ and $r \models R\ \mathit{received}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$. By the truth conditions for signature verification keys, it follows that $r \models Q\ \mathit{said}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$, $r \models Q\ \mathit{said}_{t,S}\ X$, and $r \models Q\ \mathit{has}_{t,S}\ K^{-1}$, and therefore $r \models Q\ \mathit{said}_{t,S}\ \lfloor X \rfloor_{K^{-1}} \wedge Q\ \mathit{said}_{t,S}\ X \wedge Q\ \mathit{has}_{t,S}\ K^{-1}$.

Well-timedness: Suppose $\overset{K}{\Rightarrow}_{t,S} Q \wedge R\ \mathit{received}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed with respect to r. Then $\overset{K}{\Rightarrow}_{t,S} Q$ and $R\ \mathit{received}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$ are strongly (resp. weakly) well-timed with respect to r. It follows that $\mathrm{realtime}_S(r,t) \trianglerighteq_s \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$ (resp. $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$). Thus, $Q\ \mathit{said}_{t,S}\ X$, $Q\ \mathit{said}_{t,S}\ \lfloor X \rfloor_{K^{-1}}$, and $Q\ \mathit{has}_{t,S}\ K^{-1}$ are strongly (resp. weakly) well-timed with respect to r, so $Q\ \mathit{said}_{t,S}\ X \wedge Q\ \mathit{said}_{t,S}\ \lfloor X \rfloor_{K^{-1}} \wedge Q\ \mathit{has}_{t,S}\ K^{-1}$ is strongly (resp. weakly) well-timed with respect to r. ■

A7. $P \overset{K}{\leftrightarrow}_{t,S} Q \wedge R$ $received_{t,S}$ $\{X\}_{K^{-1}} \wedge \neg P$ $said_{t,S}$ $\{X\}_{K^{-1}} \supset Q$ $said_{t,S}$ $X \wedge Q$ $said_{t,S}$ $\{X\}_{K^{-1}} \wedge Q$ $has_{t,S}$ K^{-1}

Proof: Note that since K is a shared key, $K = K^{-1}$. We use only K throughout this proof.

Truth: By the truth conditions of $\wedge$ and $\neg$, it suffices to show that if $r \models P \overset{K}{\leftrightarrow}_{t,S} Q$, $r \models R$ $received_{t,S}$ $\{X\}_K$ and $r \not\models P$ $said_{t,S}$ $\{X\}_K$, then $r \models Q$ $said_{t,S}$ X, $r \models Q$ $said_{t,S}$ $\{X\}_K$ and $r \models Q$ $has_{t,S}$ K. To this end, we assume $r \models P \overset{K}{\leftrightarrow}_{t,S} Q$, $r \models R$ $received_{t,S}$ $\{X\}_K$ and $r \not\models P$ $said_{t,S}$ $\{X\}_K$.

By the truth condition for shared keys, it follows that $r \models R$ $received_{t,S}$ $\{X\}_K$ implies $r \models T$ $said_{t,S}$ $\{X\}_K$, $r \models T$ $said_{t,S}$ X, and $r \models T$ $has_{t,S}$ K, for some $T \in \{P, Q\}$.

Thus, since $r \models R$ $received_{t,S}$ $\{X\}_K$, it follows that $r \models T$ $said_{t,S}$ $\{X\}_K$, $r \models T$ $said_{t,S}$ X, and $r \models T$ $has_{t,S}$ K, for some $T \in \{P, Q\}$. Further, since $r \not\models P$ $said_{t,S}$ $\{X\}_K$, this implies that $r \models Q$ $said_{t,S}$ $\{X\}_K$, $r \models Q$ $said_{t,S}$ X, and $r \models Q$ $has_{t,S}$ K.

Well-timedness: Suppose $P \overset{K}{\leftrightarrow}_{t,S} Q \wedge R$ $received_{t,S}$ $\{X\}_K \wedge \neg P$ $said_{t,S}$ $\{X\}_K$ is strongly (resp. weakly) well-timed with respect to r. Then $P \overset{K}{\leftrightarrow}_{t,S} Q$, R $received_{t,S}$ $\{X\}_K$, and P $said_{t,S}$ $\{X\}_K$ are strongly (resp. weakly) well-timed with respect to r. It follows that $\mathrm{realtime}_S(r,t) \trianglerighteq_s \mathrm{time}_r(\{X\}_K)$ (resp. $\mathrm{realtime}_S(r,t) \trianglerighteq_w \mathrm{time}_r(\{X\}_K)$). Thus, Q $said_{t,S}$ X, Q $said_{t,S}$ $\{X\}_K$, and Q $has_{t,S}$ K are strongly (resp. weakly) well-timed with respect to r, so Q $said_{t,S}$ $X \wedge Q$ $said_{t,S}$ $\{X\}_K \wedge Q$ $has_{t,S}$ K is strongly (resp. weakly) well-timed with respect to r. ■

Receiving and Having

A8. P $received_t$ $(X_1, \ldots, X_n) \supset P$ $received_t$ X_i

Proof: *Truth:* Suppose that $r \models P$ $received_t$ $(X_1, \ldots, X_n)$. Then there exists $t' \in \mathrm{realtime}_P(r,t)$ such that $(X_1, \ldots, X_n) \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{rcvdmsgs}_P(r,t'))$. Let $\mathcal{K} = \mathrm{skeys}_P(r,t')$ and $\mathcal{M} = \mathrm{rcvdmsgs}_P(r,t')$. Then by Lemma B.6, it follows that $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}((X_1, \ldots, X_n)) \subseteq \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$. Since, in addition, $X_i \in \mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}((X_1, \ldots, X_n))$, it follows that $X_i \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$. Hence, $r \models P$ $received_t$ X_i.

Well-timedness: Suppose P $received_t$ $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed with respect to a run r. Then $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r((X_1, \ldots, X_n))$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r((X_1, \ldots, X_n))$). Since $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed, it follows that X_i is strongly (resp. weakly) well-timed. Further, since $\mathrm{time}_r((X_1, \ldots, X_n)) = \bigcup_i \mathrm{time}_r(X_i)$, it follows that $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X_i)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X_i)$). Therefore, P $received_t$ X_i is strongly (resp. weakly) well-timed. ■

A9. P $received_t$ $K \supset P$ has_t K

Proof: *Truth:* Suppose $r \models P$ $received_t$ K for a legal run r. Then there exists $t' \in \mathrm{realtime}_P(r,t)$ such that

$$K \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{rcvdmsgs}_P(r,t'))$$

Since r is legal, it follows that $K \in \mathrm{ekeys}_P(r,t')$. Therefore $r \models P$ has_t K.

Well-timedness: P has_t K is strongly and weakly well-timed. ■

A10. P $received_t$ $\{X\}_K \wedge P$ has_t $K^{-1} \supset P$ $received_t$ X

Proof: *Truth:* Suppose $r \models P$ $received_t$ $\{X\}_K \wedge P$ has_t K^{-1} for a legal run r. Let $t' \in \mathrm{realtime}_P(r,t)$, let $\mathcal{M} = \mathrm{rcvdmsgs}_P(r,t')$, let $\mathcal{K} = \mathrm{skeys}_P(r,t')$, and let $\mathcal{K}' = \mathrm{ekeys}_P(r,t')$. Thus $\{X\}_K \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$ and $K \in \mathcal{K}'$. Since r is legal, $\mathcal{K}' = \mathcal{K} \cup \{K \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M}) : K \text{ is a key}\}$. Therefore, $K \in \mathcal{K}$ or $K \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$. If $K \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$, then it follows by Lemma B.5 that $X \in \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$. Otherwise, $K \in \mathcal{K}$, so $X \in \mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\{X\}_K)$. Furthermore, by Lemma B.6 we have $\mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\{X\}_K) \subseteq$

$\text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Hence $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Since $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ in either case, it follows that $r \models P\ \mathit{received}_t\ X$.

Well-timedness: Suppose $P\ \mathit{received}_t\ \{X\}_K \wedge P\ \mathit{has}_t\ K^{-1}$ is strongly (resp. weakly) well-timed with respect to r. Then $P\ \mathit{received}_t\ \{X\}_K$ is strongly (resp. weakly) well-timed with respect to r. Since $\text{time}_r(\{X\}_K) = \text{time}_r(X)$, it follows that $P\ \mathit{received}_t\ X$ is strongly (resp. weakly) well-timed with respect to r. ∎

A11. $P\ \mathit{received}_t\ \lfloor X \rfloor_{K^{-1}} \supset P\ \mathit{received}_t\ X$

Proof: *Truth:* Suppose $r \models P\ \mathit{received}_t\ \lfloor X \rfloor_{K^{-1}}$. Let $t' \in \text{realtime}_P(r,t)$, let $\mathcal{K} = \text{skeys}_P(r,t')$, and let $\mathcal{M} = \text{rcvdmsgs}_P(r,t')$. Then $\lfloor X \rfloor_{K^{-1}} \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Then by Lemma B.6, $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\lfloor X \rfloor_{K^{-1}}) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Since $X \in \text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\lfloor X \rfloor_{K^{-1}})$, $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Hence, $r \models P\ \mathit{received}_t\ X$.

Well-timedness: Suppose $P\ \mathit{received}_t\ \lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed with respect to r. Then $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(\lfloor X \rfloor_{K^{-1}})$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(\lfloor X \rfloor_{K^{-1}})$). Since $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed, it follows that X is strongly (resp. weakly) well-timed. Further, since $\text{time}_r(\lfloor X \rfloor_{K^{-1}}) = \text{time}_r(X)$, it follows that $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(X)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(X)$). Therefore, $P\ \mathit{received}_t\ X$ is strongly (resp. weakly) well-timed with respect to r. ∎

Saying

A12. $P\ \mathit{said}_t\ (X_1, \ldots, X_n) \supset P\ \mathit{said}_t\ X_i$

Proof: *Truth:* Suppose $r \models P\ \mathit{said}_t\ (X_1, \ldots, X_n)$. Then there exists $t' \in \text{realtime}_P(r,t)$ such that $(X_1, \ldots, X_n) \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ for $\mathcal{K} = \text{skeys}_P(r,t')$ and $\mathcal{M} = \text{sentmsgs}_P(r,t')$. By Lemma B.6, it follows that $\text{partsubmsgs}_{\mathcal{K},\mathcal{M}}((X_1, \ldots, X_n)) \subseteq \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Since $X_i \in \text{partsubmsgs}_{\mathcal{K},\mathcal{M}}((X_1, \ldots, X_n))$ by definition, it follows that $X_i \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Hence, $r \models P\ \mathit{said}_t\ X_i$.

Well-timedness: Suppose that $P\ \mathit{said}_t\ (X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed with respect to r. Then $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r((X_1, \ldots, X_n))$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r((X_1, \ldots, X_n))$). Since $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed, it follows that X_i is strongly (resp. weakly) well-timed. Furthermore, since $\text{time}_r((X_1, \ldots, X_n)) = \bigcup_i \text{time}_r(X_i)$, it follows that $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(X_i)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(X_i)$). Therefore, $P\ \mathit{said}_t\ X_i$ is strongly (resp. weakly) well-timed with respect to r. ∎

A13. $P\ \mathit{says}_t\ (X_1, \ldots, X_n) \supset P\ \mathit{says}_t\ X_i$

Proof: *Truth:* Suppose that $r \models P\ \mathit{says}_t\ (X_1, \ldots, X_n)$. Then there is some message M and some $t' \in \text{realtime}_P(r,t)$ such that $(\text{send}(M), t') \in \text{history}_P(r,t')$ and $(X_1, \ldots, X_n) \in \text{submsgs}_{\mathcal{K}}(\{M\})$, where $\mathcal{K} = \text{skeys}_P(r,t')$. As above, $X_i \in \text{partsubmsgs}_{\mathcal{K},\{M\}}((X_1, \ldots, X_n))$. Thus by Lemma B.6, $X_i \in \text{submsgs}_{\mathcal{K}}(\{M\})$. Hence $r \models P\ \mathit{says}_t\ X_i$.

Well-timedness: Suppose that $P\ \mathit{says}_t\ (X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed with respect to r. Then $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed and $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r((X_1, \ldots, X_n))$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r((X_1, \ldots, X_n))$). Since $(X_1, \ldots, X_n)$ is strongly (resp. weakly) well-timed, it follows that X_i is strongly (resp. weakly) well-timed. Furthermore, since $\text{time}_r((X_1, \ldots, X_n)) = \bigcup_i \text{time}_r(X_i)$, it follows that $\text{realtime}_P(r,t) \trianglerighteq_s \text{time}_r(X_i)$ (resp. $\text{realtime}_P(r,t) \trianglerighteq_w \text{time}_r(X_i)$). Therefore, $P\ \mathit{says}_t\ X_i$ is strongly (resp. weakly) well-timed with respect to r. ∎

A14. $P\ \mathit{said}_t\ \lfloor X \rfloor_{K^{-1}} \supset P\ \mathit{said}_t\ X$

Proof: *Truth:* Suppose $r \models P\ \mathit{said}_t\ \lfloor X \rfloor_{K^{-1}}$. Then there exists $t' \in \text{realtime}_P(r,t)$ such that $\lfloor X \rfloor_{K^{-1}} \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$ for $\mathcal{K} = \text{skeys}_P(r,t')$ and $\mathcal{M} = \text{sentmsgs}_P(r,t')$. Also, $X \in \text{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\lfloor X \rfloor_{K^{-1}})$, so it follows by Lemma B.6 that $X \in \text{submsgs}_{\mathcal{K}}(\mathcal{M})$. Hence, $r \models P\ \mathit{said}_t\ X$.

Well-timedness: Suppose $P\ \ said_t\ \lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed with respect to r. Then $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$). Since $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed, it follows that X is strongly (resp. weakly) well-timed. Furthermore, since $\mathrm{time}_r(\lfloor X \rfloor_{K^{-1}}) = \mathrm{time}_r(X)$, it follows that $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Thus, $P\ \ said_t\ X$ is strongly (resp. weakly) well-timed with respect to r. ■

A15. $P\ \ says_t\ \lfloor X \rfloor_{K^{-1}} \supset P\ \ says_t\ X$

Proof: *Truth:* Suppose $r \models P\ \ says_t\ \lfloor X \rfloor_{K^{-1}}$. Then there is some message M and some $t' \in \mathrm{realtime}_P(r,t)$ such that $(\mathrm{send}(M), t') \in \mathrm{history}_P(r,t')$ and $\lfloor X \rfloor_{K^{-1}} \in \mathrm{submsgs}_{\mathcal{K}}(\{M\})$, where $\mathcal{K} = \mathrm{skeys}_P(r,t')$. As above, $X \in \mathrm{partsubmsgs}_{\mathcal{K},\{M\}}(\lfloor X \rfloor_{K^{-1}})$. Thus by Lemma B.6, $X \in \mathrm{submsgs}_{\mathcal{K}}(\{M\})$. Hence $r \models P\ \ says_t\ X$.

Well-timedness: Suppose $P\ \ says_t\ \lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed with respect to r. Then $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(\lfloor X \rfloor_{K^{-1}})$). Since $\lfloor X \rfloor_{K^{-1}}$ is strongly (resp. weakly) well-timed, it follows that X is strongly (resp. weakly) well-timed. Furthermore, since $\mathrm{time}_r(\lfloor X \rfloor_{K^{-1}}) = \mathrm{time}_r(X)$, it follows that $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Thus, $P\ \ says_t\ X$ is strongly (resp. weakly) well-timed with respect to r. ■

A16. $P\ \ said_t\ X \supset (\exists t' \le t) P\ \ says_{t'}\ X$

Proof: *Truth:* Suppose that $r \models P\ \ said_t\ X$ for a legal run r. Then it follows that $\exists s \in \mathrm{realtime}_P(r,t)$ such that $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,s)}(\mathrm{sentmsgs}_P(r,s))$. It further follows from the legality of r and the definition of $\mathrm{submsgs}_{\mathrm{skeys}_P(r,s)}(\mathrm{sentmsgs}_P(r,s))$ that there is some $s' \le s$ and some M such that $(\mathrm{send}(M), s') \in \mathrm{history}_P(r,s')$ and $X \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,s')}(\{M\})$. Let $t' = \mathrm{clock}_P(r,s')$. Then again by the legality of r, $t' \le t$. Hence, $r \models (\exists t' \le t) P\ \ says_{t'}\ X$.

Weak well-timedness: Suppose that $P\ \ said_t\ X$ is weakly well-timed with respect to r. Then X is weakly well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$. Thus $P\ \ says_t\ X$ is weakly well-timed. Considering $t' = t$, it follows that $(\exists t' \le t) P\ \ says_{t'}\ X$ is weakly well-timed. ■

A17. $P\ \ says_t\ X \supset P\ \ said_t\ X$

Proof: *Truth:* Suppose $r \models P\ \ says_t\ X$. Then there is some message M and some $t' \in \mathrm{realtime}_P(r,t)$ such that $(\mathrm{send}(M), t') \in \mathrm{history}_P(r,t')$ and $X \in \mathrm{submsgs}_{\mathcal{K}}(\{M\})$, where $\mathcal{K} = \mathrm{skeys}_P(r,t')$. Then $\{M\} \subseteq \mathrm{sentmsgs}_P(r,t')$, so by Lemma B.4, $\mathrm{submsgs}_{\mathcal{K}}(\{M\}) \subseteq \mathrm{submsgs}_{\mathcal{K}}(\mathrm{sentmsgs}_P(r,t'))$. It follows that $X \in \mathrm{submsgs}_{\mathcal{K}}(\mathrm{sentmsgs}_P(r,t'))$, so $r \models P\ \ said_t\ r$.

Well-timedness: If $P\ \ says_t\ X$ is strongly (resp. weakly) well-timed with respect to r then X is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$), so $P\ \ said_t\ X$ is strongly (resp. weakly) well-timed with respect to r. ■

Recognition

A18. $P\ \ received_t\ \{X\}_K \wedge P\ \ has_t\ K^{-1} \supset P\ \ believes_t\ P\ \ received_t\ \{X\}_K$

Proof: *Truth:* Suppose $r \models P\ \ received_t\ \{X\}_K \wedge P\ \ has_t\ K^{-1}$ for a legal run r. Then $r \models P\ \ received_t\ \{X\}_K$ and $r \models P\ \ has_t\ K^{-1}$. Hence, it follows that there exists $u \in \mathrm{realtime}_P(r,t)$ such that $\{X\}_K \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,u)}(\mathrm{rcvdmsgs}_P(r,u))$ and there exists $u' \in \mathrm{realtime}_P(r,t)$ such that $K^{-1} \in \mathrm{ekeys}_P(r,u')$. Since $\mathrm{submsgs}_{\mathrm{skeys}_P(r,t)}(\mathrm{rcvdmsgs}_P(r,t))$ and $\mathrm{ekeys}_P(r,t)$ are both monotonically increasing throughout any legal run as t increases, we can assume without loss of generality that $u \ge u'$, and therefore $K^{-1} \in \mathrm{ekeys}_P(r,u)$.

Let r' be a run such that $(r,s) \sim_P (r',s')$ for every $s \in \mathrm{realtime}_P(r,t)$ and $s' \in \mathrm{realtime}_P(r',t)$, and fix $s' \in \mathrm{realtime}_P(r',t)$. Then, in particular, $(r,u) \sim_P (r',s')$, so

1. $\mathrm{recsubmsgs}_{\mathrm{skeys}_P(r,u)}(\mathrm{rcvdmsgs}_P(r,u)) = \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r',s')}(\mathrm{rcvdmsgs}_P(r',s'))$,
2. $\mathrm{recsubmsgs}_{\mathrm{skeys}_P(r,u)}(\mathrm{sentmsgs}_P(r,u)) = \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r',s')}(\mathrm{sentmsgs}_P(r',s'))$,
3. $\mathrm{skeys}_P(r,u) = \mathrm{skeys}_P(r',s')$,
4. $\mathrm{generate}(X) \in \mathrm{history}_P(r,u)$ iff $\mathrm{generate}(X) \in \mathrm{history}_P(r',s')$, and
5. $\mathrm{clock}_P(r,u) = \mathrm{clock}_P(r',s')$.

Recall that $\{X\}_K \in \mathrm{submsgs}_{\mathrm{skeys}_P(r,u)}(\mathrm{rcvdmsgs}_P(r,u))$ and $K^{-1} \in \mathrm{ekeys}_P(r,u)$. It follows that $\{X\}_K \in \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r,u)}(\mathrm{rcvdmsgs}_P(r,u))$. Then, by (1), $\{X\}_K \in \mathrm{recsubmsgs}_{\mathrm{skeys}_P(r',s')}(\mathrm{rcvdmsgs}_P(r',s'))$. Since $\mathrm{recsubmsgs}_{\mathcal{K}}(\mathcal{M}) \subseteq \mathrm{submsgs}_{\mathcal{K}}(\mathcal{M})$ for any set $\mathcal{K}$ of keys and $\mathcal{M}$ of messages, it follows that $\{X\}_K \in \mathrm{submsgs}_{\mathrm{skeys}_P(r',s')}(\mathrm{rcvdmsgs}_P(r',s'))$. Hence, $r' \models P \;\mathit{received}_t\; \{X\}_K$. By the choice of r', it follows that $r \models P \;\mathit{believes}_t\; P \;\mathit{received}_t\; \{X\}_K$.

Well-timedness: Suppose $P \;\mathit{received}_t\; \{X\}_K \wedge P \;\mathit{has}_t\; K^{-1}$ is strongly (resp. weakly) well-timed with respect to r. Then, $P \;\mathit{received}_t\; \{X\}_K$ is strongly (resp. weakly) well-timed. By definition, $\mathrm{time}_r(P \;\mathit{believes}_t\; P \;\mathit{received}_t\; \{X\}_K) = \mathrm{realtime}_P(r,t)$ and $\mathrm{time}_r(P \;\mathit{received}_t\; \{X\}_K) = \mathrm{realtime}_P(r,t)$. It follows that $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(P \;\mathit{received}_t\; \{X\}_K)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(P \;\mathit{received}_t\; \{X\}_K)$). Therefore, $P \;\mathit{believes}_t\; P \;\mathit{received}_t\; \{X\}_K$ is strongly (resp. weakly) well-timed. ■

Freshness

A19. $\mathit{fresh}_t\; X \supset \mathit{fresh}_t\; \lfloor X \rfloor_K$

Proof: *Truth:* Suppose that $r \models \mathit{fresh}_t\; X$. Then $r \not\models P \;\mathit{said}_{t,S}\; X$ for all $P \in \mathcal{P}$. That is, for any $P \in \mathcal{P}$ and $t' \in \mathrm{realtime}_S(r,t)$, it follows that $X \notin \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{sentmsgs}_P(r,t'))$. Then, since $X \in \mathrm{partsubmsgs}_{\mathcal{K},\mathcal{M}}(\{\lfloor X \rfloor_K\})$ for any $\mathcal{K}$ and $\mathcal{M}$, it follows by Lemma B.6 that for any $P \in \mathcal{P}$ and $t' \in \mathrm{realtime}_S(r,t)$, we have $\lfloor X \rfloor_K \notin \mathrm{submsgs}_{\mathrm{skeys}_P(r,t')}(\mathrm{sentmsgs}_P(r,t'))$. Therefore $r \models \mathit{fresh}_t\; \lfloor X \rfloor_K$.

Well-timedness: Suppose that $\mathit{fresh}_t\; X$ is strongly (resp. weakly) well-timed with respect to r. Then X is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(X)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(X)$). Hence, since $\mathrm{time}_r(\lfloor X \rfloor_K) = \mathrm{time}_r(X)$, it follows that $\lfloor X \rfloor_K$ is strongly (resp. weakly) well-timed and $\mathrm{realtime}_P(r,t) \trianglerighteq_s \mathrm{time}_r(\lfloor X \rfloor_K)$ (resp. $\mathrm{realtime}_P(r,t) \trianglerighteq_w \mathrm{time}_r(\lfloor X \rfloor_K)$). Hence, $\mathit{fresh}_t\; \lfloor X \rfloor_K$ is strongly (resp. weakly) well-timed. ■

Nonce-Verification

A20. $\mathit{fresh}_{t_1,P}\; X \wedge P \;\mathit{said}_{t_2}\; X \supset P \;\mathit{says}_{\langle t_1,t_2 \rangle}\; X$

Proof: *Truth:* Suppose that $r \models \mathit{fresh}_{t_1,P}\; X \wedge P \;\mathit{said}_{t_2}\; X$. Then $r \models \mathit{fresh}_{t_1,P}\; X$ and $r \models P \;\mathit{said}_{t_2}\; X$. Then, in particular, $r \not\models P \;\mathit{said}_{t_1}\; X$. Hence, by the soundness of A3, part (b), it follows that for any $t \leq t_1$, $r \not\models P \;\mathit{said}_t\; X$. Then by the soundness of A17, for any $t \leq t_1$, $r \not\models P \;\mathit{says}_t\; X$. Thus, $r \not\models (\exists t \leq t_1) P \;\mathit{says}_t\; X$. Furthermore, since $r \models P \;\mathit{said}_{t_2}\; X$, it follows from the soundness of A16 that $r \models (\exists t' \leq t_2) P \;\mathit{says}_{t'}\; X$. Hence, it follows that $r \models (\exists t : t_1 \leq t \leq t_2) P \;\mathit{says}_t\; X$, so $r \models P \;\mathit{says}_{\langle t_1,t_2 \rangle}\; X$.

Strong well-timedness: Suppose $\mathit{fresh}_{t_1,P}\; X \wedge P \;\mathit{said}_{t_2}\; X$ is strongly well-timed with respect to a legal run r. Then $\mathit{fresh}_{t_1,P}\; X$ is strongly well-timed with respect to r, so X is strongly well-timed and $\mathrm{realtime}_P(r,t_1) \trianglerighteq_s \mathrm{time}_r(X)$. Thus $\min_{s \in \mathrm{realtime}_P(r,t_1)} s \geq \max_{s \in \mathrm{time}_r(X)} s$. Since r is a legal run, $\min_{s \in \mathrm{realtime}_P(r,t_1)} s = \min_{s \in \mathrm{realtime}_P(r,[t_1,t_2])} s$. It follows that $\mathrm{realtime}_P(r,[t_1,t_2]) \trianglerighteq_s \mathrm{time}_r(X)$, so $P \;\mathit{says}_{\langle t_1,t_2 \rangle}\; X$ is strongly well-timed with respect to r.

Weak well-timedness: Suppose that $\mathit{fresh}_{t_1,P}\; X \wedge P \;\mathit{said}_{t_2}\; X$ is weakly well-timed with respect to a legal run r. Then $\mathit{fresh}_{t_1,P}\; X$ and $P \;\mathit{said}_{t_2}\; X$ are weakly well-timed. Hence X is weakly well-timed, $\mathrm{realtime}_P(r,t_1) \trianglerighteq_w \mathrm{time}_r(X)$, and $\mathrm{realtime}_P(r,t_2) \trianglerighteq_w \mathrm{time}_r(X)$. It follows that $\min_{s \in \mathrm{realtime}_P(r,t_1)} s \geq \min_{s \in \mathrm{time}_r(X)} s$ and $\max_{s \in \mathrm{realtime}_P(r,t_2)} s \geq \max_{s \in \mathrm{time}_r(X)} s$. Since r is a legal run, $\min_{s \in \mathrm{realtime}_P(r,t_1)} s = \min_{s \in \mathrm{realtime}_P(r,[t_1,t_2])} s$ and $\max_{s \in \mathrm{realtime}_P(r,t_2)} s = \max_{s \in \mathrm{realtime}_P(r,[t_1,t_2])} s$. Then $\mathrm{realtime}_P(r,[t_1,t_2]) \trianglerighteq_w \mathrm{time}_r(X)$, so $P \;\mathit{says}_{\langle t_1,t_2 \rangle}\; X$ is weakly well-timed with respect to r. ■

Jurisdiction

A21. P *controls*$_t$ $\varphi \wedge P$ *says*$_t$ $\varphi \supset \varphi$

Proof: *Truth:* Suppose $r \models P$ *controls*$_t$ $\varphi \wedge P$ *says*$_t$ φ. Then $r \models P$ *controls*$_t$ φ and $r \models P$ *says*$_t$ φ. It follows immediately from the truth conditions for *controls* that $r \models \varphi$.

Well-timedness: Suppose P *controls*$_t$ $\varphi \wedge P$ *says*$_t$ φ is strongly (resp. weakly) well-timed with respect to r. Then, in particular, P *controls*$_t$ φ is strongly (resp. weakly) well-timed. It follows that φ is also strongly (resp. weakly) well-timed. ■

Symmetry

A22. $P \overset{K}{\leftrightarrow}_{t,R} Q \equiv Q \overset{K}{\leftrightarrow}_{t,R} P$

Proof: *Truth:* If $r \models P \overset{K}{\leftrightarrow}_t Q$ then $r \models Q \overset{K}{\leftrightarrow}_t P$ because the truth conditions for shared keys refer to P and Q only in the set $\{P, Q\}$, which is identical to the set $\{Q, P\}$.

Well-timedness: Both $P \overset{K}{\leftrightarrow}_{t,R} Q$ and $Q \overset{K}{\leftrightarrow}_{t,R} P$ are strongly and weakly well-timed. ■

A23. ϵ-*sync*$_{t,R}$ $(P, Q) \equiv \epsilon$-*sync*$_{t,R}$ (Q, P)

Proof: *Truth:* If $r \models \epsilon$-*sync*$_{t,R}$ (P, Q) then $r \models \epsilon$-*sync*$_{t,R}$ (Q, P) because the truth conditions for synchronization are symmetric with respect to P and Q.

Well-timedness: Both ϵ-*sync*$_{t,R}$ (P, Q) and ϵ-*sync*$_{t,R}$ (Q, P) are strongly and weakly well-timed. ■