1997

# Contributions to authentication logics and analysis of authentication protocols

Anish Mathuria
*University of Wollongong*


# Contributions to authentication logics and analysis of authentication protocols



A thesis submitted in fulfillment of the
requirements for the award of the degree

**Doctor of Philosophy**

from

UNIVERSITY OF WOLLONGONG

by

**Anish Mathuria, MSc (Hons)**

School of Information Technology & Computer Science
September 1997

*Dedicated to*

*my sisters: Pooja and Bhavna;*
*and my fiancee: Hemal*

# Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Anish Mathuria, MSc (Hons)
September 11, 1997

# Preface

An authentication protocol essentially provides a mechanism for verifying the identities of nodes in an insecure network, and for the safe distribution of secrets. The subject of authentication protocols is enormously subtle. It is surprisingly easy to design incorrect protocols. A typical authentication protocol consists of an exchange of just a few messages, may appear intuitively correct, and still not work as intended. It is common to find examples of published protocols in the literature which have subsequently been found to contain flaws. As a result, methods for verifying the correctness of protocols have proliferated. A pioneering work in this area is a modal logic of Burrows, Abadi and Needham. Their work has led to the development of a substantial number of logics of a similar kind, often referred to as "authentication logics". If authentication logics are to be used to verify the correctness of protocols, then there is a need to verify the correctness of the logics themselves. The latter is the metalogical problem of obtaining assurance about the soundness of a logic. A meaningful solution to this problem requires the development of an independently motivated semantics for the logic. However, as compared to the formalisms in which authentication logics are couched, the development of semantics for such logics has generally lagged behind. Indeed, despite some notable previous work in the latter direction, it is rare to find a rigorous proof of soundness for an existing authentication logic.

There are several other interesting areas in the study of protocols besides authentication logics. These include alternative methods for analyzing protocols, and models for analyzing protocol efficiency.

This thesis makes some contributions to the areas of authentication logics and protocol analysis; the contributions made are summarized below.

1. **Authentication logics**

   (a) A critical appraisal of an authentication logic of Gong, Needham and Yahalom:

It is shown that the above logic exhibits several undesirable features, including instances of unsoundness, incompleteness, and redundancy. These observations are used to highlight the need for a semantic basis for authentication logics.

(b) A modification to the logic of Gong, Needham and Yahalom for automatic analysis of protocols:

The proposed modification lends to a simple technique for automating deductions in the modified logic. Not only does the automation provided serve as an aid in analyzing protocols, but it also proves useful in confirming some of the difficulties in using the original logic.

(c) A computational model for authentication logics:

The proposed model decouples the syntax and semantics of notions that are central to existing authentication logics. The import of the resulting model is that it provides a solid foundation for devising such logics.

(d) An authentication logic and its proof of soundness:

The model developed above is used to devise a new authentication logic and to establish a soundness theorem for the logic in a rigorous manner.

2. **Protocol analysis**

A model for reasoning about lower bounds on rounds:

The proposed model is primarily motivated by the need to verify the correctness of some informal bounds found in the literature. It provides a precise definition of the metric *number of rounds* and a theorem which relates lower bounds on rounds with security requirements.

The thesis is organized as follows. Chapter 1 is an introductory survey on authentication logics. The Chapters 2, 3, 4, and 5 cover parts (1)(a)–(d) above, respectively. Chapter 6 covers part 2. Chapter 7 contains our conclusions.

# Acknowledgments

I owe very much to the support and guidance that I have received from my co-supervisors: Associate Professor Reihaneh Safavi-Naini and Dr. Peter Nickolas. Their unrelenting technical feedback has led to the timely completion of this work. Rei has been instrumental in securing my entry into the PhD program at the University of Wollongong. Much of the financial support that I needed at various times while this work was in progress has been arranged by her. I would also like to thank her for introducing me to the subject of authentication logics, and for insisting that I keep deadlines. Peter has been an excellent teacher to me. He has had a deep influence on my research outlook; from him I have learnt to appreciate the power of mathematical proofs and clear writing. His warm nature, integrity, and high standards have provided me with a great deal of inspiration.

I have been privileged enough to have found an expert collaborator in Dr. Colin Boyd. I have benefited a lot from many stimulating discussions with him on the subject of protocol security. I am grateful to Professor Jennifer Seberry for her advice and financial support. I would also like to acknowledge the financial support that I have received from the University of Wollongong in the form of a postgraduate award for the past two years. The University of Wollongong Research Office, the Department of Computer Science, and the Centre for Computer Security Research have also provided additional support to enable me to attend several conferences at various times. I thank the two anonymous referees for their comments and criticism.

I would like to thank all my friends for their support behind the scenes and for believing in me; in particular, Gopesh Rana (Bobby), Jyotindra Trivedi and Dhananjay Patel. I am indebted to Bobby for bailing me out of a financial difficulty during the course of this work. Finally, I am grateful to my parents and the rest of my family for their love and understanding.

# Mathematical preliminaries

This section contains a concise summary of some standard mathematical notations, definitions, and results used in this thesis.

## Integers

The notation $\mathbb{Z}$ denotes the set of all integers $\{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

## Sets

If $X$ is a finite set, the *rank* of $X$ is the number of elements in $X$. Let $X$ be some set (finite or infinite). The *powerset* of $X$, written $2^X$, is the set of all subsets of $X$. Let $Y$ and $Z$ be any two sets. The *cartesian product* of $Y$ and $Z$, written $Y \times Z$, is the set $\{(y, z) \mid y \in Y \text{ and } z \in Z\}$. A *partition* $\pi$ of $X$ is a set of non-empty subsets of $X$ such that: (1) the elements of $\pi$ are pairwise disjoint, and (2) the union of all elements of $\pi$ is the set $X$. An element of a partition is called a *block*.

## Relations

Let $X$ be some set (finite or infinite). A *binary relation* on $R$ is a subset of $X \times X$. If $R$ is a binary relation on $X$, we usually write $xRy$ instead of $(x, y) \in R$, for $x, y \in X$. A relation $R$ on $X$ is: *reflexive* if, for every $x \in X$, $xRx$; *transitive* if, for every $x, y, z \in X$, whenever $xRy$ and $yRz$, then $xRz$; *euclidean* if, for every $x, y, z \in X$, whenever $xRy$ and $xRz$, then $yRz$. A relation $R$ on $X$ is *irreflexive* if, for every $x \in X$, $(x, x) \notin X$; *anti-symmetric* if, for every $x, y \in X$, whenever $(x, y) \in R$, then $(y, x) \notin R$. The *reflexive transitive closure* of a binary relation $R$, written $R^*$, is the smallest reflexive transitive relation that includes $R$ as a subset. A *partial order* is a binary relation that is irreflexive, anti-symmetric, and transitive. Let $R$ be a relation on $X$, and let $X'$ be a subset of $X$. Define the relation $R'$ on $X'$ as $R' = R \cap (X' \times X')$; if $R$ is a partial order on $X$, then $R'$ is a partial order on $X'$. We normally use the symbol $\prec$ to denote an

arbitrary partial order. If $\prec$ is a partial order on $X$, the ordered pair $(X, \prec)$ is called a *partially ordered set*, or a *poset*. A partial order $R$ on $X$ is a *total order* if, for every $x, y \in X$, $xRy$ or $yRx$. We normally use the symbol $<$ to denote an arbitrary total order. If $<$ is a total order on $X$, the ordered pair $(X, <)$ is called a *totally ordered set*. If $R$ is partial order on $X$, then an *infinite descending chain* with respect to $R$ is an infinite sequence $x_1, x_2, \ldots$ of elements of $X$ such that $x_{n+1} R x_n$ for all $n$; $R$ is *well-founded* if there are no infinite descending chains with respect to $R$.

## Strings

An *alphabet* is a finite and non-empty set of symbols. If $\Sigma$ is an alphabet, the set of all finite strings of symbols from $\Sigma$ is written as $\Sigma^*$. We write the empty sequence as $()$. If $S$ is a finite sequence and $s \in \Sigma$, then $S \cdot s$ denotes the sequence obtained by extending $S$ by $s$. If $S_1$, $S_2$, ... is a (finite or infinite) sequence of finite sequences with the property that $S_i$ is an initial segment of $S_{i+1}$ for each $i = 1, 2, \ldots$, then call the shortest sequence of which all the $S_i$ are initial segments the *union* of the $S_i$.

## Graphs

A *digraph* is an ordered pair $G = (V, R)$ where $V$ is a set and $R$ is a binary relation on $V$. The elements of $V$ are called the *nodes* of $G$; the elements of $R$ are called the *edges* of $G$. An *edge* $e = (a, b)$ is said to *originate* at node $a$ and *terminate* at node $b$; the node $a$ is called the *initial node* of $e$ and the node $b$ is called the *terminal node* of $e$. The number of edges which originate (respectively, *terminate*) at a node $a$ is called the *outdegree* (respectively, *indegree*) of $a$. A finite or infinite sequence of edges is called a *path* if the terminal node of each edge in the sequence is the initial node of the next edge, if any, in the sequence. A path is said to *originate* in the initial node of the first edge and *end* in the terminal node of the last edge, if any, in the sequence. A path that originates from a node $a$ and ends in a node $b$ is called a *path from $a$ to $b$*. A path that originates and ends at the same node is called a *cycle*. A digraph that does not contain any cycles is called *acyclic*.

## Trees

A *tree* is a digraph with a nonempty set of nodes such that: (1) there is exactly one node, called the *root* of the tree, which has indegree 0; (2) every node other than the root has indegree 1; and (3) for every node $a$ of the tree, there is a path from the root

to $a$. A tree is called *finitely generated* if each node of the tree has a finite outdegree. A tree is called *finite* if it has only finitely many nodes; otherwise the tree is called *infinite*. A *branch* of a tree is a path that originates at the root of the tree.

## König's lemma

Every finitely generated tree with infinitely many nodes must contain at least one infinite branch.

# Contents

# List of Tables

# List of Figures

# Chapter 1

# A survey of BAN-like logics

This chapter surveys some prominent logics for reasoning about authentication protocols. The seminal work in this area is a modal logic of Burrows, Abadi and Needham [1], usually called, the BAN logic. This logic has been extremely influential in the authentication protocol literature; it has stimulated widespread interest in the formal analysis of protocols. The logics we will survey are more or less classified in the literature as *BAN-like* logics. This nomenclature is somewhat loose, yet sufficiently descriptive to convey our intent, viz: The BAN-like logics share the following two traits. (1) They appear to be somehow related to the original BAN logic. (2) They were suggested by others subsequent to the BAN logic and were motivated as extensions or improvements over this logic.

We follow the usual convention of naming BAN-like logics after their authors. In addition to the BAN logic itself, we will survey the BAN-like logics proposed by:

- Gong, Needham and Yahalom [2] (GNY);
- Gong [3] (G);
- Gaarder and Snekkenes [4] (GS);
- Kailar and Gligor [5] (KG);
- van Oorschot [6] (VO);
- Mao and Boyd [7] (MB);
- Abadi and Tuttle [8] (AT);
- Syverson and van Oorschot [9] (SVO);
- Wedel and Kessler [10] (WK).

We will survey these logics in the order presented above, which is more or less chronological, with one exception. The logic of Abadi and Tuttle appeared prior to that of Kailar and Gligor; however, we make the above rearrangement for the sake of convenience.

## 1.1 Introduction

We begin with some terminology which is commonly used to describe mechanisms referred to as: *authentication protocols*. Broadly speaking, an authentication protocol consists of a sequence of message exchanges designed to achieve some security objective using cryptographic functions. The design of authentication protocols usually makes the following two characteristic assumptions (cf. Needham and Schroeder [11]). (1) A protocol is subject to an adversarial environment: it is assumed that there is an enemy who can see and manipulate messages exchanged in the communication network at will, with the purpose of subverting the protocol objectives. (2) The cryptographic functions that underlie a protocol are assumed to be secure—for example, an encrypted message is considered to be impossible for anyone to decrypt without knowing the decryption key. The goals of authentication protocols can vary depending on application, but they broadly fall into the following two categories: *entity authentication*, in which the aim is to verify the identities of one or more communicating principals; and *authenticated key exchange*, in which the aim is to make available a shared key between some principals. Some of the earliest examples of authentication protocols can be found in the paper by Needham and Schroeder [11].

The terminology used to describe authentication protocols includes the following terms: *principals*, *keys*, and *nonces*. A *principal* is an entity which takes part in a protocol run. Typically, the *keys* used by a protocol are classified as: the *long-term keys* (also sometimes called *terminal keys*), which are cryptographic keys assumed to be available initially; the *session keys*, which are cryptographic keys to be securely obtained via the protocol itself. Usually, a session key obtained in one run of the protocol is deemed unsafe for use in subsequent runs of the protocol. In particular, a sound protocol should be robust against replay of session keys. In other words, if $A$ and $B$ are two principals wishing to establish a session key via a suitable protocol, then it should not be possible for an enemy to manipulate the protocol messages to make the principals believe that an old session key is a new one. A *nonce* is a quantity which is typically used for verifying the freshness of messages [11]. The simplest example

of a nonce is a random number: if $A$ generates a random number $r$ and sends a message containing $r$, then $A$ can be assured that any message which cannot be feasibly constructed without the knowledge of $r$ cannot possibly be made prior to the message in which $A$ originally sends $r$.

In the literature most protocols are schematically described by means of syntax representing a sequence of message exchanges between some principals. It is worth noting that a typical protocol description identifies the order in which the protocol messages are meant to be exchanged in a *successful* run of the protocol. Specifically, a message exchange of the form $A \rightarrow B : M$ means that, at the point where this exchange appears in the associated protocol, (1) principal $A$ is supposed to send a message $M$, and (2) that this message is supposed to be received by principal $B$. This exchange might be accompanied by additional checks which are performed by $B$ upon receipt, if any, of the message claimed as $M$; typically $B$ does not proceed with the rest of the protocol if the stipulated check is unsuccessful.

## 1.2 BAN logic

The BAN logic [1] is a logic for reasoning about authentication protocols in terms of belief statements. It provides a useful formalism which reflects at a high level of abstraction how authentication protocols are intuitively understood to work.

The syntax of the BAN logic distinguishes three types of primitive objects: *principals*, *keys*, and *nonces*. A protocol message is expressed as a formula of the logic. Let $P$, $Q$, $R$ range over principals; let $K$ range over keys; let $X$, $Y$, $X_1, X_2, \ldots$ range over formulas. The formulas of the logic along with their informal semantics can be given as follows.

| | |
|---|---|
| $P \models X$ | $P$ *believes* $X$; $P$ believes that $X$ is true. |
| $P \triangleleft X$ | $P$ *sees* $X$; $P$ has received a message from which it can read $X$. |
| $P \mid\sim X$ | $P$ *once said* $X$; $P$ has sent (or uttered) a message containing $X$. |
| $P \Rightarrow X$ | $P$ has *jurisdiction* over $X$; $P$ is trusted on the truth of $X$. |
| $\sharp(X)$ | $X$ is *fresh*; $X$ has not been sent previous to the current protocol run. |
| $P \overset{K}{\leftrightarrow} Q$ | $P$ and $Q$ share key $K$ which is *good* in the sense that it remains confidential to $P$, $Q$ and principals trusted by either $P$ or $Q$. |
| $\overset{K}{\mapsto} P$ | $P$ has $K$ as its *public key*. The corresponding private-key $K^{-1}$ remains confidential to $P$ and principals trusted by $P$. |
| $P \overset{X}{=\!=} Q$ | $P$ and $Q$ share *secret* $X$ in the sense that it remains |

confidential to $P$, $Q$ and principals trusted by either $P$ or $Q$.

$\{X\}_K$ $X$ encrypted with key $K$.

$\langle X \rangle_Y$ $X$ *combined* with $Y$; $Y$ serves as a proof of origin of $X$.

The notation used above is from the original presentation of the logic [1]. A latter presentation of the logic [12] uses a more verbose but mnemonic notation; for example, $P\,believes X$ instead of $P \models X$. However, we retain the original notation here. The logic also includes as formulas the following expressions: (1) $\{X\}_K$ *from* $P$, which means that the encrypted message identified originates from $P$; (2) $(X, Y)$, which means the conjunction of $X$ and $Y$ (the BAN logic uses ',' as the propositional conjunction operator). The logic treats conjunction as an operator on sets of formulas, leaving properties such as associativity and commutativity implicit.

Essentially, the inference rules of the logic reflect intuitive consequences of the semantics of the logical constructs. Typically, an inference rule is read, 'if formulas $X_1, \ldots, X_n$ hold then formula $Y$ holds', written more concisely as:

$$\frac{X_1, \ldots, X_n}{Y}$$

The main inference rules of the logic are [1]:

- *Message-meaning rules:*

$$\frac{P \models P \overset{K}{\leftrightarrow} Q,\ P \triangleleft \{X\}_K}{P \models Q \mid\sim X}$$

  This rule allows the identity of the sender of an encrypted message to be deduced from the encryption key used. It makes up one of the three message-meaning rules of the logic; the message-meaning rules for public-keys and shared secrets are given along similar lines.

- *Freshness rule:*

$$\frac{P \models \sharp(X)}{P \models \sharp(X, Y)}$$

  This rule allows the freshness of a message to be deduced from the freshness of a subpart of the message.

- *Nonce-verification rule:*

$$\frac{P \models \sharp(X),\ P \models Q \mid\sim X}{P \models Q \models X}$$

  This rule allows beliefs from freshly uttered messages to be derived.

- *Jurisdiction rule:*

$$\frac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$$

This rule allows beliefs based on jurisdiction to be derived.

There are several other inference rules which reflect other properties of the various logical constructs; however, the ones given above are central to the logic. For the sake of convenience we list the original set of inference rules in appendix A.

A protocol to be analyzed using the logic is first transformed into an idealized protocol, which is essentially the sequence of protocol message exchanges expressed in terms of a set of logical formulas. Roughly speaking, a protocol idealization reflects some intended interpretation of the protocol messages. An example helps convey the basic idea behind idealization. Consider the message exchange $S \rightarrow A : \{N_a, B, K_{ab}\}_{K_{as}}$, in which a trusted server $S$ distributes a session key $K_{ab}$ to be shared between $A$ and $B$. Here $K_{as}$ is a key shared by $A$ and $S$, and $N_a$ is $A$'s nonce, which is used by $A$ to verify that the message is not a replay. Suppose the above message exchange is part of some protocol, and that this message exchange implies that $S$ asserts $K_{ab}$ to be a good session key for $A$ and $B$. Then this assertion is typically reflected in the protocol idealization by the formula: (1) $A \triangleleft \{N_a, A \overset{K_{ab}}{\leftrightarrow} B\}_{K_{as}}$. To proceed with the analysis, some formulas which express initial assumptions about the protocol in question are also asserted, and the inference rules are then applied to determine whether the formulas expressing the goal of the protocol are derivable using the logic. For example, the above idealization can be accompanied by the following assumptions: (2) $A \models \sharp(N_a)$; and (3) $A \models S \Rightarrow A \overset{K_{ab}}{\leftrightarrow} B$. It is easy to see that we can derive the formula $A \models A \overset{K_{ab}}{\leftrightarrow} B$ from (1), (2) and (3) using the logic. The import of the logical analysis is that it forces us to make explicit the various assumptions needed to obtain the desired goals. If the desired goals do not follow from an application of the logic, the pre-conditions to the inference rules often provide a hint to further assumptions that might be needed. Intuitively, if an unreasonable assumption is found in the process, it suggests the presence of a protocol flaw. For example, the BAN logic has been used to verify a flaw in the Needham-Schroeder protocol pointed out by Denning and Sacco [13]. In analyzing the protocol using their logic, Burrows *et al.* [1] show that this flaw manifests as a dubious statement amounting to the *assumption* that one of the parties believes that the session key distributed via the protocol is fresh.

The logic has also been used by Burrows *et al.* [1] to analyze several other well-known protocols from the literature. As concerning their example protocol analyses using the logic it is worth rehashing an observation of van Oorschot [6]. Namely, that

the protocol analyses presented by them might give the impression that all assumptions needed in a protocol analysis are known *a priori*. However, it should be stressed that the assumptions included in their analyses are made beforehand only for the sake of appearance. The main idea in the using the logic is to detect various unobvious assumptions which might be needed in addition to those assumptions that are initially made at the outset.

As noted by its authors [1], [14], the validity of the inference rules of the logic depends on a number of subtle assumptions that are made outside of the logic. For example, the validity of the message-meaning rule above requires the assumption that $P$ has not sent the encrypted message $\{X\}_K$ himself. This is reflected simply by writing the encrypted message as '$\{X\}_K$ *from* $R$' in the original rule together with the side condition that $P \neq R$. Another important assumption concerns the nonce-verification rule above. There it is assumed that the message $X$ does not contain any encrypted subparts (i.e., any subformula of the form $\{Y\}_K$), since intuitively a principal may not necessarily believe in a message that it cannot read. If we suppose that this assumption is satisfied, the rule effectively implies that principals are *honest*, in the sense that all message parts which a principal can read from the messages it sends must be believed by that principal. However, Burrows *et al.* recommend a rather strong operational notion of honesty: they require that a principal believe every message he sends. This strong notion of honesty has drawn some criticism of the logic, despite the fact that the soundness of the nonce-verification rule does not strictly depend on this requirement. For example, Heintze and Tygar [15] note that such a requirement precludes a BAN logic analysis of the Needham-Schroeder protocol, since this protocol requires a principal to send an encrypted message he may not believe in (Message 3). However, it is fair to say that the strong version of honesty recommended by Burrows *et al.* appears to be an inadvertent slip; if we retain the weaker version of honesty, an analysis of the protocol using the logic still goes through as intended. Moreover, the original BAN logic analysis of the protocol does not conform to the notion of honesty in the strong sense. On the other hand, the weak notion of honesty appears to be a quite useful requirement to make in the logic. For example, Engberg [16] also enforces the same weak notion of honesty while applying the logic; he shows that absurd conclusions can be drawn using the logic otherwise.

The BAN logic also makes two general assumptions concerning encrypted messages. (a) It assumes message integrity in the sense that encrypted messages cannot be spliced or aggregated without destroying the message structure—for example, that

an encrypted message $\{M, M'\}_K$ cannot be obtained from the sole knowledge of the two encrypted messages $\{M\}_K$ and $\{M'\}_K$ (and vice-versa). However, there may be protocols where the cryptographic mechanisms used do not provide such guarantees (cf. Boyd [17] and Gligor *et al.* [18]). (b) It assumes that encrypted messages are verifiable in the sense that the result of a decryption can be somehow verified to be genuine. To put (a) and (b) another way, the logic cannot be used to detect the lack of message integrity or message verifiability in protocols.

Although the BAN logic is simple to use, proofs carried out in the logic require subtle interpretation. An early example of the underlying subtlety is provided by a protocol due to Nessett [19] (we omit Message 2 from the original protocol since it is not relevant to the present discussion):

$$\text{Message 1.} \quad A \rightarrow B: \ \{N_a, K_{ab}\}_{K_a^{-1}}$$

Here $B$ trusts $A$ to generate a session key $K_{ab}$ to be shared between them. The nonce $N_a$ is used by $A$ to convince $B$ that Message 1 is fresh. $K_a^{-1}$ is $A$'s private key for use in a suitable public-key system. The corresponding public-key $K_a$ is publicly known. Nessett [19] asserts some formulas meant to reflect the initial assumptions, which include amongst others the formula $A \models A \overset{K_{ab}}{\leftrightarrow} B$, and shows that the BAN logic can be used to sanction the protocol in the sense that the formula $B \models A \overset{K_{ab}}{\leftrightarrow} B$ can be derived from Message 1. The protocol of course is insecure, nevertheless, since everyone knows $K_a$ and can thus decrypt Message 1 using $K_a$ to obtain $K_{ab}$. Based on the above example protocol analysis, Nessett claims that the BAN logic is *flawed.* However, Burrows *et al.* [20] refute the grounds on which Nessett advances his claim. They object to the assumption $A \models A \overset{K_{ab}}{\leftrightarrow} B$ above as being unjustifiable since Message 1 contradicts this assumption. The essence of the controversy surrounding Nessett's point and the counterpoint of Burrows *et al.* is not novel: a 'proof' is only as good as the assumptions it makes. However, it does accurately point out a practical difficulty in appealing to proofs in the logic.

## 1.3 GNY logic

The logic of Gong, Needham and Yahalom [2] is one of the earliest BAN-like logics. Their logic largely modifies and adds to the BAN logic notation and rules in an attempt to provide more features than the logic on which it is framed.

Unlike the BAN logic, the syntax of the GNY logic distinguishes between messages and assertions about messages. The former are represented in the GNY logic

as *formulas* (which do not assume truth-values), whereas the latter are represented as *statements*. Essentially, this distinction precludes messages from being treated as believable expressions in the logic.

Let $X$, $Y$ range over formulas, and let $C$ range over statements. The formulas of the GNY logic include the following:

| | |
|---|---|
| $\{X\}_K$ | encryption of $X$ with shared key $K$. |
| $\{X\}_K^{-1}$ | decryption of $X$ with shared key $K$. |
| $\{X\}_{+K}$ | encryption of $X$ with public-key $+K$. |
| $\{X\}_{-K}$ | decryption of $X$ with private-key $-K$. |
| $H(X)$ | one-way hash of $X$. |
| $F(X,Y)$ | a many-to-one function of $X$ and $Y$ which is one-to-one and invertible when either $X$ or $Y$ is fixed; for example, $X \oplus Y$. |
| $X \rightsquigarrow C$ | $X$ with *extension* $C$; $C$ reflects some interpretation of $X$. |

Let $P$, $Q$, $R$ range over principals. Essentially, the new notions introduced in the GNY logic are expressed by the following statements:

| | |
|---|---|
| $P \triangleleft *X$ | $P$ has received a message from which it can read $X$ and $X$ is *not-originated-from* $P$ in the sense that $X$ has not been previously sent by $P$ in the current run. |
| $P \models \otimes(P)$ | $P$ can identify that certain messages *never-originated-from* him in the sense that they have not been sent by $P$ in any run. |
| $P \ni X$ | $P$ *possesses* $X$; $P$ has received $X$ or can compute $X$. |
| $P \models \phi(X)$ | $P$ believes $X$ is *recognizable*; $P$ recognizes $X$ in the sense that $P$ can forecast part or whole of the contents of $X$ without receiving $X$. |
| $P \models Q \Rightarrow Q \models *$ | $P$ believes $Q$ is *honest and competent* in the sense that $Q$ has jurisdiction over all his beliefs. |

The remaining statements of the logic are made along similar lines to the BAN logic, except that they are formed within the scope of the belief construct; for example, the GNY logic includes a statement of the form $P \models \sharp(X)$, but not $\sharp(X)$. The term *shared secrets* is used to encompass both encryption keys as well as other types of secrets in the GNY logic. Typically, the symbol $S$ ranges over secrets. The logic uses a single construct in place of the two BAN logic constructs $\leftrightarrow$ and $\rightleftharpoons$. However, the GNY logic notation for sharing of secrets is rather loose: $P \models P \overset{S}{\leftrightarrow} Q$. This notation is rather

restrictive since it does not admit of statements of the form $R \models P \overset{S}{\leftrightarrow} Q$ when the principal denoted by $R$ is distinct from the principals denoted by $P$ and $Q$. However, such statements are often needed in practice, as is evident from the protocol analyses given in the GNY logic paper; see, for example, the analysis of the Needham-Schroeder protocol there (p. 241).

The GNY logic has over forty inference rules, many of them quite complex in terms of the number of premises involved. For the sake of convenience we list them in appendix B. Some of these are rules which are absent from the BAN logic but which seem intuitively justifiable. For example, one part of the freshness rule F2,

$$\frac{P \models \sharp(X),\ P \ni K}{P \models \sharp(\{X\}_K),\ P \models \sharp(\{X\}_K^{-1})},$$

essentially captures an inference which did not seem to be required for the protocol analyses carried out by Burrows *et al.* [1] using their logic. A large number of rules reason about new notions introduced in the logic, for instance, those of possession and recognizability.

The rest of this section discusses the intended role of the following notions found in the logic: *not-originated-from* and *never-originated-from*, *possession*, *recognizability*, and *honesty*.

## Not-originated-from and Never-originated-from

The idea behind the notions of not-originated-from and never-originated-from is to capture the side condition to the BAN logic message-meaning rules for shared keys. Recall that the side condition reflects the assumption that a principal can tell whether an encrypted message was sent by himself or not. The informal semantics of the notion of *not-originated-from* is that, if $P$ receives $X$ at some point and $X$ is not-originated-from $P$ (written $P \triangleleft *X$), then $P$ has not sent $X$ since the start of the current protocol run up to that point. The GNY logic is accompanied by a parser algorithm which mechanically translates a protocol description into one with the not-originated-marker $*$. There are two distinct ways the GNY logic attempts to capture the original BAN logic side condition. Firstly, it reformulates the BAN logic message-meaning rule for shared keys by including premises of the form $P \triangleleft *\{X\}_K$ and $P \models \sharp(X)$ in place of $P \triangleleft \{X\}_K$ and the extra side condition; see the message interpretation rule I1 in appendix B.6. An alternative reformulation simply distinguishes the side condition by means of a premise of the form $P \models \otimes(P)$; see the never-originated-rule I1′ in appendix B.8. Despite their intuitive appeal, the usefulness of the above two notational

devices is rather limited. Since the parser algorithm only controls the current run of the protocol, the GNY logic reformulation involving the not-originated-from notion is of use only in the case of a protocol which requires a message $X$ to be conveyed by some principal $P$ and such that the same message $X$ is to be later told to $P$. The rule I1 then produces the desirable effect of blocking the derivation of beliefs for $P$ from $X$. The reformulation which makes use of the never-originated-from notion simply seems to be another way of writing the side condition. Since the GNY logic does not include any rules with statement of the form $P \models \otimes(P)$ as conclusions, it is debatable whether this notion represents an improvement over the BAN logic side condition.

## Possession

The notion of *possession* is a noteworthy addition introduced in the GNY logic. This notion underlies many rules of the BAN logic; for example, in the BAN logic message seeing rule for shared keys:

$$\frac{P \models P \overset{K}{\leftrightarrow} Q,\ P \triangleleft \{X\}_K}{P \triangleleft X},$$

it is implicitly assumed that $P$ possesses $K$. However, this assumption has the effect of conflating the two distinct notions of possession of a key $K$ and that of belief about $K$. The GNY logic make this distinction explicit; the above rule is reworked as:

$$\text{T3.}\quad \frac{P \triangleleft \{X\}_K,\ P \ni K}{P \triangleleft X}$$

## Recognizability

As noted in previous section, the BAN logic makes the implicit assumption that encrypted messages are verifiable. The GNY logic captures this assumption explicitly by reformulating the BAN logic message-meaning rules; for example, in the message interpretation rule I1 of the GNY logic,

$$\frac{P \triangleleft *\{X\}_K,\ P \ni K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models \phi(X),\ P \models \sharp(X,K)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K,\ P \models Q \ni K},$$

the premise $P \models \phi(X)$ essentially reflects the implicit BAN logic assumption that encrypted messages are verifiable. An example protocol analysis which evidently demonstrates the usefulness of the recognizability feature of the logic can be found in the GNY logic paper, where the logic is shown to reveal the lack of message verifiability in the enhanced Needham-Schroeder protocol. Specifically, an analysis of the protocol

using the GNY logic does not yield a certain desired protocol goal, in contrast to a BAN logic analysis. The difference is attributed to the fact that one of the message exchanges used by the protocol is: $Q$ sends to $P$ an encrypted message $\{N_q\}_K$, where $N_q$ is a nonce generated by $Q$ and $K$ is a session key between $P$ and $Q$. The lack of recognizability of this message relative to $P$ produces the desirable effect of blocking the derivation of the following statement using the logic: $P \models Q \models P \overset{K}{\leftrightarrow} Q$.

The GNY logic has several rules which allow derivation of statements of the form $P \models \phi(X)$; see the 'recognizability rules' in appendix B.5. While the notion of recognizability appears to be a useful addition to the logic, certain recognizability rules of the GNY logic are problematic; we will discuss some of the problems involved in the next chapter.

### Honesty

Recall that the BAN logic makes the implicit assumption that principals are honest. However, not all principals may be equally honest; in other words, the notion of honesty can be specified relative to a trusting principal. The GNY logic makes such a viewpoint explicit. Essentially, the BAN logic nonce-verification rule is reformulated in the logic as the jurisdiction rule J2,

$$\frac{P \models Q \Rightarrow Q \models *,\ P \models Q \mid\sim (X \rightsquigarrow C),\ P \models \sharp(X)}{P \models Q \models C},$$

where the premise $P \models Q \Rightarrow Q \models *$ means that $Q$ is trusted by $P$ over his beliefs.

The honesty requirement is enforced during a protocol analysis using the GNY logic by means of a *belief consistency* check: to idealize a protocol message exchange $P \rightarrow Q : X$ as $Q \triangleleft *X \rightsquigarrow C$ a precondition is that the statement $P \models C$ holds. Similarly, a *possession consistency* check is also carried out to enforce the intuitive requirement that a principal can only send possessed messages: another precondition to the above idealization step is that the statement $P \ni X$ holds.

## 1.4 G logic

The original GNY logic has been revised by Gong [3] in his doctoral thesis. The revised logic mostly expands the GNY logic set of rules concerning notions such as freshness and message interpretation. Essentially, the only new feature to be found in this logic is a notion called *eligibility*:

$P \propto X$ $P$ is *eligible* to convey $X$; $P$ holds the *relevant* possessions and beliefs to convey $X$.

Here the requirement that $P$ holds the 'relevant' possessions and beliefs for a formula $X \rightsquigarrow C$ is roughly understood to mean that the statements $P \ni X$ and $P \models C$ hold, among other things. In the revised logic, the consistency requirement given in the GNY logic paper is reflected by the rule T1 [3],

$$\frac{P \rightarrow Q : X,\ P \propto X}{Q \triangleleft X},$$

which essentially excludes message exchanges not satisfying the belief or possession consistency checks. Additionally, the logic includes several rules, called 'eligibility rules', which allow statements of the form $P \propto X$ to be derived. In effect, these rules define the set of formulas which a principal is eligible to convey. For example, the rule E1 [3],

$$\frac{P \ni X}{P \propto X},$$

says that a principal is eligible to convey any formula he possesses. It is worth noting that in the revised logic $P$'s eligibility to convey $\{X\}_K \rightsquigarrow C$ is not intended to mean the same thing as (1) $P$ possession of $X$ and $K$, and (2) $P$'s belief in $C$. This is apparent from the eligibility rule E5 [3],

$$\frac{P \propto X,\ P \ni K,\ P \models P \overset{K}{\leftrightarrows} Q,\ P \models C}{P \propto \{X\}_K \rightsquigarrow C},$$

where the premise $P \models P \overset{K}{\leftrightarrows} Q$ indicates a component of $P$'s 'relevant' beliefs which is independent of $P$'s belief in $C$. It is thus clear that the consistency requirement made in the revised logic is more stringent than that motivated in the GNY logic paper.

The significance of the belief and possession consistency requirements is further discussed by Gong [21]. He argues that their absence can lead to *infeasible* specifications in the sense that: (1) protocols which do not meet the possession consistency check cannot be realized, and (2) protocols which do not meet the belief consistency check may allow *non-causal* beliefs to be derived. Here non-causality of beliefs means that some statement of the form $P \models Q \models C$ holds, but that the statement $Q \models C$ does not hold.

## 1.5 GS logic

The logic of Gaarder and Snekkenes [4], [22] extends the BAN logic in two ways: (1) It reformulates the BAN logic notions and rules for public-key systems. (2) It introduces

new notions for reasoning about freshness mechanisms which lie outside the scope of the BAN logic. These two extensions are described below.

Recall that the notions of binding of a public-key and that of the secrecy of the corresponding private-key are represented by means of a single construct in the BAN logic: $\mapsto$. The GS logic distinguishes between these two notions using the following constructs:

| | |
|---|---|
| $\mathcal{PK}(K, P)$ | $K$ is $P$'s authentic public-key. |
| $\Pi(P)$ | $P$'s private-key is *good* in the sense that it is a secret known to $P$ alone. |

The above distinction is further reflected in the GS logic by the rule R13 [4],

$$\frac{P \models \mathcal{PK}(K, Q),\ P \models \Pi(Q),\ P \triangleleft \sigma(X, Q)}{P \models Q \mid\sim X},$$

which is essentially obtained by reformulating the BAN logic message-meaning rule for public-keys; here $\sigma(X, Q)$ replaces $\{X\}_{K^{-1}}$ used in the latter rule to represent $X$ signed with $Q$'s private-key. The GS logic also replaces the BAN logic message seeing rule for public-keys with the following rule [4]:

$$\text{R14.}\quad \frac{P \triangleleft \sigma(X, Q)}{P \triangleleft X}$$

Notice that the above rule makes the implicit assumption that a principal possesses every principal's public-key, including his own.

Essentially, the BAN logic nonce-verification rule tells us that no beliefs can be derived from a message sent during a protocol run if that message is *not fresh* (i.e. if that message has been sent previous to the current protocol run). However, some protocols make use of mechanisms which do not rely on the notion of freshness in the above sense and, yet, for which it is intuitively reasonable to establish the level of belief supported by the nonce-verification rule. The basic idea underlying such mechanisms is the use of a *duration-stamp* to indicate a time interval for which a message is claimed to be *good.* The above observations essentially motivate the remaining GS logic extensions that incorporate time into the logical syntax. In particular, the logic includes the following constructs to reflect the associated notions of *duration-stamp* and *good time interval.*

| | |
|---|---|
| $(\Theta(t_1, t_2), X)$ | $X$ tagged with duration-stamp $\Theta(t_1, t_2)$; $X$ *holds in the interval* $t_1, t_2$. |
| $\Delta(t_1, t_2)$ | $t_1, t_2$ denotes a *good time interval*; the current time lies in the interval between $t_1$ and $t_2$. |

The logic includes the following rule to reason about duration-stamps [4]:

$$\text{R15.}\quad \frac{P \models Q \models \Delta(t_1, t_2),\ P \models Q \mid\sim (\Theta(t_1, t_2), X)}{P \models Q \models X}$$

Here it is assumed that the validity of a time interval is ascertained by a principal relative to its own local clock. The rule R15 allows beliefs from uttered messages attested by duration-stamps to be derived; it is framed similar to the nonce-verification rule of the BAN logic.

The import of the above extensions is demonstrated in the GS logic paper [4] by means of a concrete protocol which has been fielded for use: the CCITT X.509 protocol. An analysis of this protocol using the extended logic is shown to compare favorably with a BAN logic analysis of the same protocol. The main improvement concerns the idealization of the certificates used to distribute public-keys in the protocol. A protocol analysis using the BAN logic requires the dubious assumption that the certificates are fresh, despite the fact that the actual working of the protocol does not make this assumption. Stated another way, the timing mechanism used to guarantee the validity of the certificates in the protocol can be captured in the GS logic, whereas in the BAN logic it cannot.

## 1.6 KG logic

Kailar and Gligor [5] have devised a BAN-like logic to extend the applicability of the original BAN logic. They argue that the BAN logic suffers from the following two limitations: (1) The BAN notion of key jurisdiction is dependent on key generation; that is, a principal who is authorized to *generate* a key only has jurisdiction over that key. However, there may be protocols which do not satisfy key jurisdiction in the above sense. Gligor *et al.* [18] sketch an example to motivate the restrictive nature of the BAN logic notion of key jurisdiction. Suppose that (a) $P$ trusts $Q$ to read and forward a key $K$ generated by $R$, and (b) $P$ trusts $Q$ to maintain the privacy of $K$. Although $Q$ does not have jurisdiction over $K$ in the above sense, $P$'s trust in $Q$ should allow $P$ to infer $K$ is a good key for use with $Q$. Nonetheless the BAN logic does not capture this line of reasoning. (2) The BAN logic allows derivation of non-causal beliefs in certain protocols. The KG logic is essentially aimed to address the above two concerns.

The syntax of the KG logic makes the ordering of protocol message exchanges explicit using the notion of a *message round* (also called, *message instance*). Specifically, a message round corresponds to a transfer of a message contents $X$ from a source principal $P$ to a destination principal $Q$, possibly via other principals, with the property

that either: (1) $X$ is signed with $P$'s private-key and encrypted with $Q$'s public-key, or (2) $X$ is encrypted with a key shared by $P$ and $Q$. The idea of message ordering can be explained by means of an example. Suppose we have a protocol which transfers data values $X$ and $Y$ from $S$ to $A$ and $B$, respectively, as follows:

$$\begin{aligned} &\text{Message 1.} \quad S \rightarrow A: \quad \{X, \{Y\}_{K_{bs}}\}_{K_{as}} \\ &\text{Message 2.} \quad A \rightarrow B: \quad \{Y\}_{K_{bs}} \end{aligned}$$

Here $K_{as}$ and $K_{bs}$ are keys shared by $A$ and $B$ with $S$, respectively. The above protocol consists of two message rounds: one which transfers $X$ from $S$ to $A$ and another one which transfers $Y$ from $S$ to $B$. The Messages 1 and 2 above are then explicitly represented in terms of message rounds using the following tuples, respectively:

$$\{M_1, S, A, (X, \{M_2, S, B, Y\})\} \text{ and } \{M_2, S, B, Y\},$$

where in general (1) a tuple of the form $\{M_i, P, Q, X\}$ denotes that $X$ is transferred from $P$ to $Q$ in message round $i$; and (2) the message rounds comprising the protocol are consecutively denoted as: $M_1$, $M_2$, ..., $M_l$ for a fixed positive integer $l$.

The additional notions introduced in the logic along with their informal semantics can be given as:

| | |
|---|---|
| $P \triangleright \{M_i, P, Q, X\}$ | $P$ *sends* a message with contents $X$ in round $i$ to $Q$. |
| $Q \triangleleft \{M_i, P, Q, X\}$ | $Q$ *sees* a message with contents $X$ in round $i$; $Q$ reads $X$ and knows $X$ originated from $P$. |
| $KS(X, M_i)$ | the *knowledge set* of $X$ at round $M_i$; the set of all principals who know $X$ when the message identified in $M_i$ is seen. |
| $Trust_X(P, Q)$ | $P$ *trusts* $Q$ on the *context* $X$. |

Apart from the BAN logic notation, some additional notation from predicate logic and set theory is used in formulating the set of inference rules of the logic; for example, the symbol $\forall$ is used to denote universal quantification and the symbol $\in$ is used to denote set membership. Below we list some of the rules of the KG logic [5].

- *Belief in the uniqueness of the message recipient:*

$$\frac{P \triangleright \{M_i, P, Q, X\}, \;\; P \models (R \triangleleft \{M_i, P, Q, X\})}{P \models (R = Q)}$$

Notice that this rule is cast from the perspective of a message originator instead of the intended recipient of the message; it essentially replaces the BAN logic message-meaning rules.

- *Set inclusion belief (1):*

$$\frac{P \notin KS(X, M_i),\ \ P \triangleleft \{M_{i+1}, Q, P, X\}}{P \models KS(X, M_{i+1}) = \{P, Q\}}$$

This rule allows belief in the knowledge set of a message content to be derived based on messages that are seen.

- *Set inclusion belief (2):*

$$\frac{P \models (Q \notin KS(X, M_i)),\ \ P \triangleright \{M_{i+1}, P, Q, X\}}{P \models KS(X, M_{i+1}) = KS(X, M_i) \cup \{Q\}}$$

This rule allows belief in the knowledge set of a message content to be derived based on messages that are sent; it reflects a notion of so-called 'eager' belief, not present in the BAN logic.

- *Belief in the freshness of message contents:*

$$\frac{P \models \sharp(X, M_k),\ \ P \triangleleft \{M_k, Q, P, (X, Y)\}}{P \models \sharp(Y, M_k)}$$

Here the premise $P \models \sharp(X, M_k)$ is taken to mean that $P$ believes $X$ *is fresh in round* $k$; and similarly for the rule's conclusion.[1]

- *Belief about another principal's knowledge set beliefs:*

$$\frac{P \triangleleft \{M_k, Q, P, X\},\ \ P \models \sharp(X, M_k)}{P \models Q \models (KS(X, M_k) = \{P, Q\})}$$

This rule essentially replaces the BAN logic nonce-verification rule.

- *First-level beliefs:*

$$\frac{P \models \{P, Q\} \subseteq KS(K, M_l),\ \ Trust_K(P, R)\ \forall R \in KS(K, M_l)}{P \models P \overset{K}{\leftrightarrows} Q}$$

Here: (1) $M_l$ is taken to mean the last round of the protocol being analyzed, and (2) the premise $Trust_K(P, R)\ \forall R \in KS(K, M_l)$ essentially says that $P$ trusts all principals who know $K$ to maintain its secrecy. In effect, the above rule allows beliefs about keys to be derived without the need for key jurisdiction in the sense of the BAN logic.

[1] Intuitively, this rule appears to be questionable. From the the BAN logic notion of freshness it is apparent that we cannot infer that $Y$ is fresh from the fact that $(X, Y)$ is fresh, since this would imply that if $Y$ was sent previous to the current protocol run then so are all messages of the form $(X, Y)$.

- *Second-level beliefs:*

$$\frac{P \models Q \models \{P, Q\} \subseteq KS(K, M_l),\ P \models Trust_K(Q, R)\ \forall R \in KS(K, M_l)}{P \models Q \models P \overset{K}{\leftrightarrows} Q}$$

Kailar and Gligor [5] give several examples of protocol analyses using their logic to demonstrate its potential advantages over the BAN logic. Here we sketch one of the example protocol analyses which they give to show that the BAN logic does not preserve belief ordering. Specifically, a ticket-forwarding protocol is analyzed using the BAN logic, based on the following idealization of the protocol [5, Subsection 5.3]:

$$\begin{array}{lll} \text{Message 2.} & TGS \rightarrow X: & \ldots, \{X \overset{K_{X/Y-S}}{\leftrightarrow} S\}_{K_{X-TGS}} \\ \text{Message 3.} & X \rightarrow Y: & \ldots, \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}_{K_{X-Y}} \\ \text{Message 4.} & Y \rightarrow S: & \ldots, \{X \overset{K_{X/Y-S}}{\longleftrightarrow} S\}_{K_{X/Y-S}} \end{array}$$

In the protocol analysis given, it is essentially shown that although $Y \models X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ is derived from Message 3, only $S \models X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ is derived from Message 4, instead of the desired $S \models Y \models X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$. Notice that the latter is still possible to derive using the BAN logic if we appeal to the weak notion of honesty discussed in Section 1.2. Accordingly, we cannot include the formula $X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ in the idealization of Message 4 above, since in their analysis $Y \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ does not hold. However, in accordance with the honesty requirement we can include the formula $X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ instead, since $Y \models X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ holds. The desired formula $S \models Y \models X \models X \overset{K_{X/Y-S}}{\longleftrightarrow} S$ is then easily derived.

## 1.7 VO logic

Van Oorschot [6] provides an extension of the BAN and GS logics to cater for *key agreement* protocols. The notion of *key agreement* used in such protocols can be described as follows. Two principals wishing to establish a common secret key individually generate a pair of keys consisting of a *public key-agreement key* and a *private key-agreement key.* As these names suggest, each principal keeps its private key-agreement key secret, but reveals its public key-agreement key. The common key is then obtained by each principal as some suitable function $f$ of its own private key-agreement key and the other principal's public key-agreement key, where $f$ is chosen in advance and made public. A characteristic feature of the above notion of key agreement is that the common key established is not obtained from any trusted principal and is exclusively obtained by

the two principals who derive the key jointly. However, the BAN logic notion of good keys cannot be used to distinguish this feature: recall that the BAN logic construct $P \overset{K}{\leftrightarrow} Q$ means that $K$ is *good* for $P$ and $Q$ in the sense that it is known only to $P$ or $Q$ or principals trusted by either of them. The above observation essentially motivates the VO logic refinements of the broader BAN logic notion of *good* shared keys. Specifically, the refinements include the following two constructs:

| | |
|---|---|
| $P \overset{K}{\leftrightarrow}\!\!\!\!\!\!- \; Q$ | $K$ is $P$'s *unconfirmed secret* for use with $Q$; $P$ possesses $K$ and knows that no other principal except $Q$ can possibly obtain $K$. |
| $P \overset{K}{\leftrightarrow}\!\!\!\!\!\!\ddagger \; Q$ | $K$ is $P$'s *confirmed secret* for use with $Q$; $P$ receives evidence to the effect that its unconfirmed secret meant for use with $Q$ is indeed possessed by $Q$. |

Here *possession* is treated as in the sense of the GNY logic. The VO logic includes the construct $P$ *has* $K$ in place of the less verbose GNY construct $P \ni K$. As in the GS logic, the notions of binding of a public-key and the secrecy of the corresponding private-key are treated distinctly. In the VO logic, further distinction is made between the asymmetric key pairs used for signature, encryption, and key agreement. However, since the notation for asymmetric encryption key pairs is not exploited in the logic, we omit it here.

| | |
|---|---|
| $PK_\sigma(P, K)$ | $K$ is the public signature-verification key *associated* with $P$. |
| $PK_\sigma^{-1}(P)$ | $P$'s *private signature key* $K^{-1}$ is good in the sense that it is known only to $P$. |
| $PK_\delta(P, K)$ | $K$ is the public key-agreement key *associated* with $P$. |
| $PK_\delta^{-1}(P)$ | $P$'s private key-agreement key $K^{-1}$ is *good* in the sense that it is known only to $P$. |
| $\{X\}_{sP}$ | $X$ signed with $P$'s private signature key. |
| $\{X\}_K$ | $X$ encrypted with shared key $K$. |
| *confirm*$(K)$ | *Current knowledge* of $K$ *has been demonstrated* in the sense that $K$ has been used to perform some cryptographic action such as encrypting or hashing. |

Notice that the constructs $PK_\sigma^{-1}(P)$ and $\{X\}_{sP}$ essentially replace their GS logic counterparts $\Pi(P)$ and $\sigma(X, P)$, respectively.

The logic introduces several rules for reasoning about key-agreement keys. For the sake of notational convenience, the following adjustments are made in presenting the rules there [6]:

- $PK_\delta(P)$ is written in place of $PK_\delta(P, K)$ when $K$ is evident from the context or is not explicitly referred to, and
- $PK_\delta(P)$ is used to denote the value of the public-key agreement key of $P$.

The rules for key-agreement keys are given as follows [6].

- *Unqualified key-agreement:*

$$\text{R30.} \quad \frac{P \textit{ has } PK_\delta^{-1}(P),\ \ P \textit{ has } PK_\delta(U)}{P \textit{ has } K}$$

Here $K = f(PK_\delta^{-1}(P), PK_\delta(U))$ is called an *unqualified* key for $P$, which is taken to mean that the identity of principal $U$ is not verified.

- *Qualified key-agreement:*

$$\text{R31.} \quad \frac{P \models PK_\delta^{-1}(P),\ \ P \models PK_\delta(Q),\ \ P \models PK_\delta^{-1}(Q)}{P \models P \overset{K-}{\leftrightarrow} Q}$$

Here $K-$ denotes that $K$ is a *qualified* key, which is taken to mean that $P$ knows that $K$ cannot be possessed by any other principal except $Q$.

- *Key confirmation:*

$$\text{R32.} \quad \frac{P \models P \overset{K-}{\leftrightarrow} Q,\ \ P \textit{ sees } *confirm(K)}{P \models P \overset{K+}{\leftrightarrow} Q}$$

Here $K+$ denotes that $K$ is a *confirmed* key, which is taken to mean that $P$ has obtained confirmation that $Q$ actually possesses $K$.

The VO logic paper contains analyses of three well-known key-agreement protocols using the extended logic: the STS protocol, the Goss protocol and the Günther protocol. (An informal description of the working of some other notable key-agreement protocols independent of any logical formalism can be found in a paper by Rueppel and van Oorschot [23].) The analyses are shown to reveal some subtle differences between the assumptions made and the goals reached by these protocols. The comparison are made on the basis of six generic goals captured using the logical syntax. For example, the two goals called *secure key establishment* (G3) and *key confirmation* (G4) are respectively expressed as follows [6]: $A \models A \overset{K-}{\leftrightarrow} B$ and $A \models A \overset{K+}{\leftrightarrow} B$ (and similarly for $B$), where $A$ and $B$ denote the two principals wishing to establish a common key $K$ via some suitable protocol. All the above three protocols are shown to attain G3, whereas only the STS protocol is shown to attain G4.

## 1.8 MB logic

Mao and Boyd [7] have devised a BAN-like logic to address some of their objections to the BAN logic. Their main criticism concerns the lack of well-defined rules for protocol idealization in using the BAN logic. The point is that this difficulty can lead to incorrect idealizations. A simplified version of the Otway-Rees protocol is used as an example of a protocol which can be sanctioned using the BAN logic and yet for which an attack is possible (cf. Boyd and Mao [24]). Their other criticisms of the BAN logic include the lack of typing and the absence of a notion of confidentiality. The former criticism concerns an oddity noted as early as by Burrows *et al.* [1] themselves, namely that their logic does not make any distinction between messages and formulas (truth-valued expressions); for example, the logical syntax allows as formulas expressions of the form $P \models N$, where $N$ is a nonce. The latter criticism is motivated by means of the flaw in Nessett's protocol: the flaw is traced to a failure of the protocol to maintain the confidentiality of the key distributed.

The syntax of the MB logic makes a distinction between messages and formulas by means of a typing mechanism. The logical syntax is divided into three syntactic classes: $\mathcal{P}$ (for *principals*), $\mathcal{M}$ (for *messages*), and $\mathcal{F}$ (for *formulas*). Typically,

- the letters $P$, $Q$, $R$, ...are used to denote elements of class $\mathcal{P}$;
- the letters $K$, $M$, $N$, ...are used to denote elements of class $\mathcal{M}$; and
- the letters $X$, $Y$, $Z$, ...are used to denote elements of class $\mathcal{F}$.

Additionally, the symbol $\mathcal{S}$ is used to denote a set of principals; $\mathcal{S}^C$ denotes the complement of the set of principals denoted by $\mathcal{S}$. A set of formation rules defines the class $\mathcal{M}$ of messages and the class $\mathcal{F}$ of formulas of the logic. For example, the *belief* formulas are formed as follows: $\_ \models \_ : \mathcal{P} \times \mathcal{F} \rightarrow \mathcal{F}$. The BAN logic constructs $\leftrightarrow$, $\mapsto$ and $\sharp$ are similarly reformulated. Further, '$\wedge$' replaces ',' used to represent the conjunction operator in the BAN logic. The additional notions introduced in the MB logic can be given as follows.

| | |
|---|---|
| $P \overset{K}{\mid\sim} M$ | *P said M using the encryption key K.* |
| $P \overset{K}{\triangleleft} M$ | *P sees M using the decryption key K.* |
| $sup(P)$ | $P$ is a *super-principal.* |
| $\mathcal{S}^C \not\triangleleft M$ | the principals in the set $\mathcal{S}^C$ *cannot see M.* |

Of these the first two are essentially refinements of the BAN logic notions of *said* and *seen* messages. The idea here is to make the key used to convey a message or see a message explicit; the original BAN logic constructs (without superscripts) are taken to mean that the key used is not of particular significance. The notion of a *super-principal* is used to capture *unconditional* trust in some principal. Notice that this notion is far less expressive when compared to the BAN logic notion of jurisdiction. The notion *cannot see* provides a basis to express confidentiality requirements.[2]

The terms *challenge*, *replied challenge* and *response* are used to capture the context-dependent role played by message elements. Typically, a nonce issued by some principal is called a *challenge* in a message where it is originally sent; it is called a *replied challenge* in a message where it is received by the originator of the challenge. A *response* is taken to be a primitive message which is combined with a replied challenge by the originator of the message containing the replied challenge. The above terminology is used to formulate rules for protocol idealization using two constructs, called 'message combinators': | and $\Re$. The first of these, '|', is used to associate challenges or responses; the second, '$\Re$', is used to associate responses with challenges, typically as *response* $\Re$ *challenge*.

Below we list the principal rules of the logic [7].

- *Authentication rule (A1):*

$$\frac{P \models P \overset{K}{\leftrightarrow} Q \wedge P \overset{K}{\triangleleft} M}{P \models Q \overset{K}{\mid\sim} M}$$

- *Confidentiality rule (C):*

$$\frac{P \models P \overset{K}{\leftrightarrow} Q \wedge P \models \mathcal{S}^C \blacktriangleleft M \wedge P \overset{K}{\mid\sim} M}{P \models (S \cup \{Q\})^C \blacktriangleleft M}$$

- *Nonce-verification rule (N):*

$$\frac{P \models \sharp(M) \wedge P \models Q \overset{K}{\mid\sim} M}{P \models Q \models P \overset{K}{\leftrightarrow} Q}$$

[2] We can formulate this notion more clearly, and without any loss of generality, as:

$\mathcal{S} \blacktriangleleft M$ the principals in the set $\mathcal{S}$ *cannot see* $M$.

However, we retain the more cumbersome formulation in deference to the original presentation of the logic.

- *Super-principal rule (S):*

$$\frac{P \models Q \models X \wedge P \models sup(Q)}{P \models X}$$

- *Fresh rule (F):*

$$\frac{P \models \sharp(M) \wedge P \triangleleft N \Re M}{P \models \sharp(N)}$$

- *Good-key rule (G1):*

$$\frac{P \models \{P, Q\}^C \lhd\!\!\sharp\, K \wedge P \models \sharp(K)}{P \models P \overset{K}{\leftrightarrows} Q}$$

A backward reasoning technique is recommended in analyzing protocols using the logic. The aim of this technique is to derive the minimal sets of assumptions needed to infer a fixed set of desired protocol goals using the logic. The reasoning technique is applied to the Nessett protocol and it is shown that an application of the confidentiality rule is needed to meet a specific protocol goal, which in turn requires an unreasonable assumption.

## 1.9 AT logic

The BAN-like logics that we have discussed so far rely heavily on syntax, with little apparent effort being made to define the semantics of logical expressions independently of the syntax. The work of Abadi and Tuttle [8] marks a turning point in this regard: it is one of the earliest works to make an attempt at providing such a semantics for a BAN-like logic and to suggest a soundness theorem for the proposed logic.

The AT logic can be thought of as a reformulated BAN logic with a revised semantics. As discussed by Abadi and Tuttle, the motivations for their logic include the following semantic issues related to the original logic:

- *The meaning of good keys:* They note that the secrecy property stipulated in the informal semantics of the BAN logic notion of *good keys* is not strictly necessary for the soundness of the message-meaning rules of the logic. This point is reflected in the formal semantics of the BAN logic, since there a good key $K$ is defined in terms of who *sends* messages encrypted with $K$. However, according to them, this definition is also quite strong: if $K$ is a good key between $P$ and $Q$ then anyone can send a message encrypted with $K$ as long as $P$ and $Q$ are the only principals *using* $K$ to encrypt messages.

- *Possession v/s Belief:* They argue that these two notions should be made distinct in the logic, for the sake of a proper semantics. This observation is motivated by the fact that the notion of possession is implicit in the BAN logic seeing rules, where it is assumed that belief in a key implies possession of that key.

- *Stability of beliefs:* In the BAN logic, it is assumed that formulas are *stable* in the sense that a formula remains true once it becomes true. In particular, the stability of belief formulas is critical to the soundness of the nonce-verification rule of the logic. However, this requirement can be removed by expressing the conclusion of the rule slightly differently, viz: if $P$ said $X$ and $X$ is fresh, then $P$ has *recently said* $X$. To carry this idea further, they suggest defining the notion of jurisdiction in terms of the notion *recently said* in place of belief. The BAN logic essentially takes the latter course because of the way the nonce-verification rule is designed to work.

The syntax of the AT logic is designed to exclude messages from being treated as formulas, unlike the BAN logic. Another difference concerns the fact that the AT logic relates more closely to traditional propositional modal logics of belief: it includes primitive propositions and the standard propositional connectives for *negation*, *disjunction*, *conjunction*, *implication*, and *equivalence*, respectively denoted as $\neg$, $\vee$, $\wedge$, $\supset$, and $\equiv$. Typically,

- the symbols $P$, $Q$, $R$, $S$ range over principals,

- the symbol $K$ ranges over keys,

- the symbol $X$ ranges over messages, and

- the symbols $\phi$, $\psi$ range over formulas.

The logic proper includes a set of axioms which essentially express the statements captured via inference rules in the BAN logic as formulas in the logic itself. Most of the logical axioms are formulated without the use of the belief operator. For example, one of the axioms for message-meaning is stated as:

$$\text{A5.} \quad P \overset{K}{\leftrightarrow} Q \wedge R \textit{ sees } \{X^S\}_K \supset Q \textit{ said } X$$

with the side condition that $P \neq S$. The axioms for belief are formulated separately; for example, one of the axioms for belief is given as:

$$\text{A1.} \quad P \textit{ believes } \phi \wedge P \textit{ believes } (\phi \supset \psi) \supset P \textit{ believes } \psi$$

The logic also includes two inference rules typically found in more traditional logics: *modus ponens* (R1) and *belief necessitation* (R2).

R1. From $\vdash \phi$ and $\vdash \phi \supset \psi$ infer $\vdash \psi$.

R2. From $\vdash \phi$ infer $\vdash P$ *believes* $\phi$.

Here $\vdash \phi$ means that $\phi$ is derivable in the logic.

A model of computation is given to define a semantics for the logic. The main idea of the model is to assign truth-values to formulas with respect to a *run* $r$ and a time $t$, where a run typically represents an execution of a given protocol. Each principal is assumed to be capable of performing the following actions:

- *send*$(m, Q)$: the action of sending of message $m$ to principal $Q$.
- *receive*$(m)$: the action of receiving of message $m$.
- *newkey*$(K)$: the action of generating key $K$.

Further, the two notions *history* and *key set* are associated with each principal. A principal's history in $r$ is taken to be the sequence of all actions $P$ performs in $r$; the key set is simply the set of keys the principal holds. The notion of a key set is essentially used to define two operations on messages: *seen-submsgs*$_{\mathcal{K}}(M)$ and *said-submsgs*$_{\mathcal{K}}(M)$. If $\mathcal{K}$ denotes the key set of some principal $P$, then roughly speaking: the first operation determines the messages *seen* by $P$ as a result of receiving $M$; the second operation determines the messages *said* by $P$ as a result of sending $M$. The notions sketched above suffice to give a flavor of the truth conditions defined in the AT logic paper. For example, the truth condition for $P$ *sees* $X$ is stated as,

$$(r, k) \models P \textit{ sees } X$$

iff, for some message $M$, at time $k$ in $r$:

1. *receive*$(M)$ appears in $P$'s local history, and
2. $X \in$ *seen-submsgs*$_{\mathcal{K}}(M)$, where $\mathcal{K}$ is $P$'s key set.

A notable aspect of the semantics of the AT logic is its treatment of the notion of belief. This notion is defined in terms of *possible worlds*, where a world is a pair $(r, k)$ consisting of a run $r$ and a time $k$: a principal $P$ believes a formula $\phi$ in $(r, k)$ if $\phi$ is true in all the worlds $P$ considers possible in $(r, k)$. This contrasts with the syntactic approach used in defining the notion of belief in the BAN logic.

The AT logic paper includes a theorem (cf. Theorem 1 of Abadi and Tuttle [8]) which states that the logical axiomatization is sound with respect to the semantics defined. However, no proof of this theorem has been published.

## 1.10 SVO logic

Syverson and van Oorschot [9] have devised a BAN-like logic to encompass selective features from the logics BAN, GNY, VO and AT. The syntax and semantics of their logic mostly follows the line of the AT logic with additional extensions. To deal with the demands that their aimed expansion seemingly brings to the logical syntax, they employ some notational short cuts. For example, the notation $\{X\}_K$ is used to denote encrypted (using a public-key encryption or a shared key encryption function) as well as signed messages. Additionally, the notation $F(X_1, \ldots, X_k)$ subsumes the previous notation and also the notation $(X_1, \ldots, X_k)$ used to denote concatenated messages. Notice that the authors of previous BAN-like logics have avoided such notational short cuts. The syntax of formulas of the SVO logic includes constructs to denote binding of public-keys for signature verification and public-keys for key agreement: $PK_\sigma(P, K)$ and $PK_\delta(P, K)$, which are essentially from the VO logic. Unlike the latter, however, no explicit constructs are defined in the syntax to denote the VO logic concepts related to goodness of corresponding private keys. The construct $P$ *receives* $X$ essentially replaces the construct $P$ *sees* $X$ of the AT logic. The latter is reserved for the notion of *possession*: $P$ *sees* $X$ is used to denote that $P$ possesses $X$. The notation $K^{-1}$ is used to denote the complement of key $K$. Most of the axioms and the inference rules found in the AT logic are included in a slightly different form in the SVO logic. For example, the message-meaning axiom (A5) of the AT logic is recast as:

$$3.\quad P \overset{K}{\leftrightarrow} Q \wedge R \textit{ received } \{X^Q\}_K \supset Q \textit{ said } X$$

The SVO logic includes several other axioms to reflect the intended extensions. For example, the following axiom is designed to reflect the VO logic notion of *key agreement*:

$$5.\quad PK_\delta(P, K_p) \wedge PK_\delta(Q, K_q) \supset P \overset{K}{\leftrightarrow} Q$$

where $K_{pq} = f(K_p, K_q^{-1}) = f(K_q, K_p^{-1})$ for some key agreement function $f$. Similarly, the axiom given below is designed to capture the possession rules of the GNY logic collectively, barring the rules P1 and P3. (The latter two are reflected by means of individual axioms.)

$$10.\quad P \textit{ sees } X_1 \wedge \cdots \wedge P \textit{ sees } X_n \supset P \textit{ sees } F(X_1, \ldots, X_n)$$

Perhaps the more interesting axioms are those that indirectly relate to the GNY logic notion of recognizability. These are the two axioms called *comprehending* axioms:

11. $P$ *believes* $(P$ *sees* $F(X)) \supset P$ *believes* $(P$ *sees* $X)$
12. $P$ *received* $F(X) \wedge P$ *believes* $(P$ *sees* $X) \supset P$ *believes* $(P$ *received* $F(X))$

Here the expression $P$ *believes* $(P$ *sees* $X)$ is taken to replace the GNY logic statement $P \models \phi(X)$.

The model of computation of the SVO logic is similar to that of the AT logic. Essentially, the former is obtained by modifying the latter to include additional notions. For example, the action of generating a key $K$, denoted *newkey*$(K)$ in the model of Abadi and Tuttle, is replaced by the more general action of generating a primitive message $X$, denoted *generate*$(X)$. This modification is used in defining the notion of *seen* messages, with the aim of capturing the GNY logic notion of possession: a set of seen messages is associated with a principal, which includes, amongst other things, the messages that are received or generated by that principal.

The SVO logic paper contains a soundness theorem for the proposed logic. However, the sketch of the proof given there leaves the soundness of most of the logical axioms implicit.

## 1.11 WK logic

The logic of Wedel and Kessler [10] is one of the latest BAN-like logics along the lines of the logics AT and SVO. One of the motivations underlying their logic is to allow analysis of protocols relying on various cryptographic mechanisms that cannot be adequately captured in these two logics. The authors of the WK logic take a middle ground between the notations of BAN-like logics that predate the AT logic and the notational short cuts introduced in the SVO logic. For example, the syntax of the WK logic distinguishes between encrypted and hashed messages, unlike the SVO logic. The WK logic notations for encryption, hashing and signing functions are respectively given as *enc*, $h$, and $\sigma$. However, the notation *enc* is variously used to cover symmetric or asymmetric encryption functions as well as signature functions with message recovery; the notation $\sigma$ is reserved for signature functions that do not provide message recovery. The notation $F$ is used to denote either of *enc*, $h$, or $\sigma$; the notation $F(M)$ is taken to mean the *structure* of the message computed by $F$ on $M$, not its *value*. A notion of message *localized towards* a principal is defined to capture what parts of a message structure can be verified by the principal. If $M$ is a message, the notation $M_P$ is read

*M localized towards P*; for example, if $P$ possesses $M$ then $(h(M))_P$ is defined to be equal to $h(M)$; this definition reflects the property that $P$ can verify the hash of any messages he possesses. A set of *generalized messages* $\mathcal{M}_P$ is defined to consist of all messages that can be constructed from the basic message items and with the additional property that it is closed under localization.

Unlike the logics AT and SVO, the syntax of formulas of the WK logic precludes formulas from being treated as messages. Most of the constructs found in the logics AT and SVO are also carried over to the WK logic with some adjustments. The notation $\longrightarrow$ is used to denote the propositional connective for implication. A primitive construct *P recognizes M* is used to denote the notion of recognizability; this contrasts with the SVO logic where recognizability is not defined as a primitive notion. The constructs $\epsilon \overset{K}{\mapsto} P$, $\sigma \overset{K}{\mapsto} P$, and $\alpha \overset{K}{\mapsto} P$ replace their SVO logic counterparts $PK_\psi(P, K)$, $PK_\sigma(P, K)$, and $PK_\delta(P, K)$, respectively. The WK logic includes axioms similar to those found in the logics on which it is framed. For example, the AT/SVO logic message-meaning axiom for shared keys is modified to capture the associated side condition in the logic itself:

$$\text{A1. } R \textit{ sees } F(K, X) \wedge P \overset{K}{\leftrightarrow} Q \wedge \neg P \textit{ said } F(K, X) \longrightarrow Q \textit{ said } (K, X)$$

Here $F(K, X)$ is taken to variously denote shared-key encryption as well as hashing of $X$ using $K$. The jurisdiction axiom found in the AT/SVO logics is modified to bring it closer to the original BAN logic rule, as follows:

$$\text{J. } P \textit{ controls } \phi \wedge P \textit{ believes } \phi \longrightarrow \phi$$

The logic also includes several additional axioms; for example, an axiom for recognizability is given as:

$$\text{R1. } P \textit{ recognizes } X_i \longrightarrow P \textit{ recognizes } (X_1, \ldots, X_k)$$

A noteworthy innovation of the authors of the WK logic concerns protocol idealization. Unlike the logics AT and SVO, protocol analyses using the WK logic are carried out without having to treat formulas as messages. The semantics of the WK logic is developed along essentially similar lines to the logics AT and SVO. The authors of the WK logic use their semantics to suggest instances of unsoundness in some earlier logics; for example, the GNY logic recognizability rule R6. (We will have occasion to return to this rule in the next chapter.) The proof of soundness of the WK logic follows the line of the SVO logic paper: it leaves the soundness of most of the logical axioms implicit.

# Chapter 2

# An informal proposal for rectifying some problematic features of the GNY logic

This chapter highlights some problematic features of the GNY logic. In particular, we will point out several classes of problems which arise in the GNY logic:

1. an unsound rule;
2. the possibility of drawing unsound conclusions by pairing rules;
3. the incompleteness of the set of rules; and
4. rules with redundant premises.

The notions of soundness and completeness of a logic are usually defined with respect to an independently motivated formal semantics for the logic. However, as we shall discuss in the following section, the GNY logic does not appear to have such a semantics. Our use of the terms "unsound" and "incomplete" in this chapter must therefore be understood informally. We will give specific instances of the above problems and suggest some solutions to rectify these informally, at the syntactic level. A formal justification for the suggested solutions, however, rests ultimately on provision of an independently motivated semantics for the logic. Our purpose here is not to find a semantic solution to the problems, but our observations clearly point out the need for such a solution. In a later chapter, we will build an independently motivated semantic model for BAN-like logics, which provides a step in the former direction.

(Parts of this chapter appeared in preliminary form elsewhere [25].)

## 2.1 Soundness troubles: recognizability and freshness rules

Informally, if a logic is sound then false conclusions cannot be inferred from true premises in the logic. A formal semantics for the logic provides a precise structure with respect to which soundness can be proved. However, in order to obtain any assurance about the soundness of the logic, the semantics itself must be sufficiently independent of the logical syntax. As emphasized by Syverson [26], [27], an independently motivated semantics can provide adequate assurance about the validity and power of the logic, by means of soundness and completeness proofs, respectively.

Gong, Needham and Yahalom [2], like Burrows, Abadi and Needham [1], provide an "operational" semantics for their logic, but as has been argued by others, most notably by Syverson [27] and by Tuttle [28], the original semantics of these logics is not independently motivated, as it takes its structure directly from the logical syntax. For example, the authors of the BAN logic define the semantics of the $\sharp$ operator to correspond directly to the freshness inference rule of their logic. A set of *fresh* formulas $\mathcal{F}$ is defined for each run under consideration, as follows: $\mathcal{F}$ contains all formulas $X$ such that $\sharp(X)$ holds as an initial assumption, and additionally $\mathcal{F}$ is taken to enjoy the closure property that, if $X \in \mathcal{F}$ and $X$ is a subformula of $Y$ then $Y \in \mathcal{F}$. Then $\sharp(X)$ is defined to be true in the corresponding run if $X \in \mathcal{F}$. (The GNY logic semantics of the $\sharp$ operator is also taken to be defined similarly.) The problem with such a semantics is that it does not provide us with any independent means to check the soundness of the inference rules themselves. Indeed, we give examples below of unsound conclusions derivable in the GNY logic.

### 2.1.1 Unsound rule

The GNY logic recognizability rule R6 states that if $P$ possesses the hash of $X$, then $P$ believes $X$ is recognizable. Note that recognizability of $X$ is intended to mean that $P$ has prior expectations about the contents of $X$ independent of the act of receiving it; this interpretation of recognizability is part of the informal semantics given by Gong, Needham and Yahalom [2, p. 236]. Surprisingly, the rule R6 enables the conclusion that $P$ believes $X$ is recognizable from the premise that $P$ possesses $X$: from $P \ni X$ it follows by rule P4 that $P \ni H(X)$, and therefore, by R6, that $P \models \phi(X)$. There is nothing wrong with P4; it just says that a principal is capable of computing the hash of a message he possesses. Seemingly, the problem lies with R6. The rule becomes

problematic when we take into account the fact that a principal $P$'s possessions include the following: (1) the messages received by $P$, and (2) the messages that can possibly be computed from $P$'s possessed messages. This fact is evident from the possession rules P1 through P8 of the logic. Since, by rule P1, every message $P$ receives is also possessed by $P$, the rule R6 in effect tells us that every message received by $P$ is also recognizable by $P$. However, this conclusion conflicts with a basic intuition underlying the notion of recognizability. For example, consider a protocol where $P$ generates a random value $N_p$, and sends it to $Q$. Here $N_p$ is not recognizable by $Q$, although $Q$ possesses it.

The problem with R6 can also be seen from another viewpoint. Recall that the notion of recognizability is meant to reflect the implicit BAN logic assumption that encrypted messages are verifiable. For instance, the rule I1, which is the GNY logic counterpart of the BAN logic message-meaning rule for shared keys stipulates a recognizability premise, $P \models \phi(X)$, to express the BAN logic assumption explicitly. This rule also has the following two premises: (a) $P \triangleleft *\{X\}_K$ and (b) $P \ni K$. Given R6, it is easy to see that the recognizability premise itself is derivable from these two premises using the logic: From (a) and rule T1, $P \triangleleft \{X\}_K$, and therefore, from (b) and T3, $P \triangleleft X$. Hence, by P1, $P \ni X$, and so, by P4 and R6, $P \models \phi(X)$. It is clear here that R6 really begs the recognizability feature of the logic.

Ironically enough, the unsoundness of R6 is perhaps best illustrated by appealing to the analysis of the enhanced Needham-Schroeder protocol in the GNY logic paper, which is used to promote the recognizability feature of the logic. As part of the protocol handshake, $Q$ sends to $P$ an encrypted message $\{N_q\}_K$, where $N_q$ is a nonce generated by $Q$ and $K$ is a session key known to $P$ and $Q$. In the protocol analysis given in the paper (p. 242), it is argued that this message is unrecognizable to $P$, since $N_q$ is unpredictable by $P$. Thus $P$ can only gain possession of $N_q$ (so that $P \ni N_q$), but not any beliefs from this message. The latter is essentially reflected in the logic by the recognizability premise in the rule I1. To make the above message recognizable to $P$, it is suggested that the message be modified to include $Q$'s identifier: $\{N_q, Q\}_K$; the modified version is seen to allow the expected belief for $P$ to be derived under the additional assumption that $P \models \phi(Q)$. However, since we have $P \ni N_q$, R6 allows us to infer that $P \models \phi(N_q)$, which does not require the extra assumption that $P \models \phi(Q)$. It should be emphasized that we are not claiming that the protocol modification suggested in the GNY logic paper is superfluous. The point of the above exercise is only to reinforce our claim that R6 is at odds with the original purpose of

adding recognizability to the logic.

### 2.1.2 Unsound conclusions from pairs of rules

We first recall an observation made by Anderson [29], namely that the freshness rules F2 and F7 of the GNY logic, when used together, imply a "strange result". Suppose that for principal $P$ all of the following conditions hold: (1) $P$ believes that formula $X$ is recognizable; (2) $P$ possesses a key $K$; (3) $P$ believes that $K$ is fresh. Then, by F7, $P \models \sharp(\{X\}_K)$, and therefore from F2 and the fact that $\{\{X\}_K\}_K^{-1} = X$, it follows that $P \models \sharp(X)$.

Curiously, although each of the rules F2 and F7 is plausible in itself, when used together as above these rules produce a suspect conclusion. For example, we can extend the analysis of the modified enhanced Needham-Schroeder protocol in the GNY logic paper, to derive the nonsensical conclusion that $P \models \sharp(Q)$, as follows. Observe that in the protocol analysis given there (p. 241), the following statements hold: (a) $P \models \phi(Q)$; and (b) $P \ni K$. ($K$ is a session key known to $P$ and $Q$.) The statement (a) holds by assumption, and the statement (b) holds from message 4 of the protocol in which $S$ sends to $P$ the following: $\{N_p, Q, K, \ldots\}_{K_{ps}}$. To derive the nonsensical conclusion using F2 and F7, we need the statement $P \models \sharp(K)$. This is a reasonable statement to obtain, since the session key $K$, which is generated by the server $S$ in the protocol, is normally expected to be a fresh quantity. We can capture this in the logic as follows. Firstly, we introduce two additional statements as assumptions: (c) $S \models \sharp(K)$; and (d) $P \models S \Rightarrow \sharp(K)$. Secondly, we modify the idealization of the above message to include the former statement in the extension attached to the message:

$$P \triangleleft *\{\ldots\}_{K_{ps}} \rightsquigarrow (S \models P \overset{K}{\leftrightarrow} Q, S \models \sharp(K))$$

We can now derive the statement $P \models S \models \sharp(K)$ in an essentially similar way to which the statement $P \models S \models P \overset{K}{\leftrightarrow} Q$ is derived from the original idealization in the GNY logic paper. The required statement $P \models \sharp(K)$ then immediately follows from (d) and the jurisdiction rule J1.

We note that the problem pointed out by Anderson is not only confined to the freshness rules F2 and F7 used together. There are several other pairs of freshness and recognizability rules which lead to essentially the same problem:

(i) R2 and F7;

(ii) F8 and F4;

(iii) F9 and F3;

(iv) R3 and F9;

(v) R4 and F8.

Note that with the pairs (iii) and (v) we assume public-key schemes where $\{\{X\}_{-K}\}_{+K} = X$; for example, RSA [30].

### 2.1.3 Side conditions

To tackle the above problem, we suggest side conditions to several of the freshness and recognizability rules of the logic. We begin by replacing each of the rules F2, F7, and R2 with two equivalent rules:

$$\text{F2}' \quad \frac{P \models \sharp(X),\ P \ni K}{P \models \sharp(\{X\}_K)}$$

$$\text{F2}'' \quad \frac{P \models \sharp(X),\ P \ni K}{P \models \sharp(\{X\}_K^{-1})}$$

$$\text{F7}' \quad \frac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K}{P \models \sharp(\{X\}_K)}$$

$$\text{F7}'' \quad \frac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K}{P \models \sharp(\{X\}_K^{-1})}$$

$$\text{R2}' \quad \frac{P \models \phi(X),\ P \ni K}{P \models \phi(\{X\}_K)}$$

$$\text{R2}'' \quad \frac{P \models \phi(X),\ P \ni K}{P \models \phi(\{X\}_K^{-1})}$$

We proceed to include the following side condition to the rule F2'': $X$ is not of the form $\{Y\}_K$. The intuition used to arrive at this side condition is as follows. Let us assume that the statements $P \models \sharp(\{Y\}_K)$ and $P \ni K$ hold. In the absence of the side condition, by F2' we can obtain the conclusion $P \models \sharp(Y)$. Now, the only way we could have established $P \models \sharp(\{Y\}_K)$ is by a prior application of either of the rules F2' or F7'. Observe that: (1) If F2' were applied, then the statement $P \models \sharp(Y)$ holds *a priori*; (2) If F7' were applied, then the statements $P \models \phi(Y)$ and $P \models \sharp(K)$ hold *a priori*. In the former case, since the statement $P \models \sharp(Y)$ holds already, deriving it through F2'' is of no use essentially. However, in the latter case, deriving the statement $P \models \sharp(Y)$

through F2″ is unsound since this statement does not necessarily hold. Intuitively then, by the side condition on F2″, we have only omitted the possibility of an unsound conclusion, without losing any useful derivations.

We can also argue for similar side conditions to each of the rules F3, F4, F7″, F8, F9, R2″, R3, and R4. We list below these rules along with their corresponding side conditions.

F3 $\dfrac{P \models \sharp(X),\ P \ni +K}{P \models \sharp(\{X\}_{+K})}$, $X$ is not of the form $\{Y\}_{-K}$

F4 $\dfrac{P \models \sharp(X),\ P \ni -K}{P \models \sharp(\{X\}_{-K})}$, $X$ is not of the form $\{Y\}_{+K}$

F7″ $\dfrac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K}{P \models \sharp(\{X\}_K^{-1})}$, $X$ is not of the form $\{Y\}_K$

F8 $\dfrac{P \models \phi(X),\ P \models \sharp(+K),\ P \ni +K}{P \models \sharp(\{X\}_{+K})}$, $X$ is not of the form $\{Y\}_{-K}$

F9 $\dfrac{P \models \phi(X),\ P \models \sharp(-K),\ P \ni -K}{P \models \sharp(\{X\}_{-K})}$, $X$ is not of the form $\{Y\}_{+K}$

R2″ $\dfrac{P \models \phi(X),\ P \ni K}{P \models \phi(\{X\}_K^{-1})}$, $X$ is not of the form $\{Y\}_K$

R3 $\dfrac{P \models \phi(X),\ P \ni +K}{P \models \phi(\{X\}_{+K})}$, $X$ is not of the form $\{Y\}_{-K}$

R4 $\dfrac{P \models \phi(X),\ P \ni -K}{P \models \phi(\{X\}_{-K})}$, $X$ is not of the form $\{Y\}_{+K}$

Similarly, we include the side condition: $X$ is not of the form $\{Y\}_K^{-1}$, to the rules F2′, F7′, and R2′, for conventional cryptosystems in which $\{\{X\}_K^{-1}\}_K = X$; for example, DES [31]. Note that the side conditions to F3 and F8 are only needed for public-key schemes in which $\{\{X\}_{-K}\}_{+K} = X$.

## 2.2 Completeness troubles: The Yahalom protocol

In this section we give an example of a non-trivial rule which is not captured by the GNY logic [2]. We find this rule to be essential for verifying the working of the Yahalom

protocol, which according to the GNY logic paper, is apparently within the scope of the logic (p. 243). While it has been suggested that in such logics rules may be added when needed (cf. Gong [3, p. 18]), because of the variety of cryptographic techniques possible, an independently motivated semantics is essential if we are to be able to obtain assurance about the soundness of the added rules.

Essentially, the message interpretation rules of the logic enable the derivation of beliefs from encrypted or hashed messages. As we shall see below, the GNY logic lacks a message interpretation rule to reason about the use of a shared secret in the Yahalom protocol. In analyzing this protocol, we use the protocol parsing scheme given by Gong [3], instead of the scheme given in the GNY logic paper. First, we clarify the reasons for not using the original parsing scheme.

### 2.2.1 Protocol parsing

The first step in analyzing a protocol described in the conventional notation is to generate a form suitable for manipulation in the logic. In the GNY logic, this task is performed by a protocol parser. For each protocol message $X$ received by a principal $P$ (written $P \triangleleft X$), the parser inserts symbolic information to distinguish those parts of $X$ which are not included in any message sent by $P$ up to the point of receiving $X$ in the current protocol run. Specifically, for every statement of the form $P \triangleleft X$, the parser inserts a *not-originated-here* marker, '$*$', in front of each complete subpart $Y$ of $X$, if $Y$ does not appear as a subpart of any message $P$ has sent previously in the current run (p. 238).

We observe that the only message interpretation rules with a formula of the form $P \triangleleft *X$ appearing as a premise are the rules I1, I2, and I3. In each of these rules, the not-originated-here marker is either prefixed to an encrypted formula (I1 and I2, respectively) or a hashed formula (I3). For the purpose of using the logic to derive beliefs from encrypted or hashed messages, it makes no significant difference whether the insertion of the not-originated-here marker is carried out for non-encrypted and non-hashed message parts or not; we choose not to. Not only does this simplify the parsing process, it also avoids a peculiar problem with original parsing scheme. In particular, we note that the original scheme precludes some legitimate applications of the message interpretation rules. For example, in the analysis of a voting protocol in the GNY logic paper, the statement $Q \models Q \overset{*S_i}{\leftrightarrow} P_i$ is clearly required to apply I3 to the second message of the protocol (p. 239). Presumably, we can derive this statement from the protocol assumption $Q \models Q \overset{S_i}{\leftrightarrow} P_i$, but there is nothing in the logic which

would enable us to do so. However, such a difficulty does not arise if we adopt the modified parsing scheme given by Gong [3].

### 2.2.2 The Yahalom protocol

The goal of the Yahalom protocol [1] is to distribute an authenticated session key to two principals $A$ and $B$ via a trusted third party known as the authentication server $S$. The following sequence of messages describes a successful run of the protocol (p. 30):

1. $A \rightarrow B : A, N_a$
2. $B \rightarrow S : B, \{A, N_a, N_b\}_{K_{bs}}$
3. $S \rightarrow A : \{B, K_{ab}, N_a, N_b\}_{K_{as}}, \{A, K_{ab}\}_{K_{bs}}$
4. $A \rightarrow B : \{A, K_{ab}\}_{K_{bs}}, \{N_b\}_{K_{ab}}$

As explained by Burrows *et al.* [1], this protocol makes use of an *uncertified key*: a key which is used before its validity is established.

In the sequel, we refer to the protocol initiator $A$ as 'Alice' and the other principal $B$ as 'Bob', following standard practice. Initially, Alice and Bob share keys $K_{as}$ and $K_{bs}$ with the authentication server $S$ respectively. Alice initiates the protocol by sending her identity and a nonce $N_a$ to Bob. In the second message, Bob sends to the server his own name and an encrypted part $\{A, N_a, N_b\}_{K_{bs}}$, where $N_b$ is Bob's nonce. In the third message, the server sends to Alice: $\{B, K_{ab}, N_a, N_b\}_{K_{as}}$, $\{A, K_{ab}\}_{K_{bs}}$. The first encrypted part tells Alice that $K_{ab}$ is a good session key for communicating with Bob, and also tells her Bob's nonce. The second encrypted part is intended for Bob. In the fourth message, Alice forwards this encrypted part to Bob, along with Bob's nonce encrypted with $K_{ab}$. Bob decrypts the first encrypted part of this message to get $K_{ab}$, and uses it to decrypt the second encrypted part. If the latter decryption yields Bob his nonce $N_b$, then he obtains assurance that $K_{ab}$ is a good session key for communicating with Alice.

### 2.2.3 Analyzing the Yahalom protocol using GNY logic

We begin the analysis by using the parsing scheme to produce a protocol description containing $*$'s in the appropriate places:

1. $B \triangleleft A, N_a$
2. $S \triangleleft B, *\{A, N_a, N_b\}_{K_{bs}}$
3. $A \triangleleft *\{B, K_{ab}, N_a, N_b\}_{K_{as}} \rightsquigarrow S \models A \overset{K_{ab}}{\leftrightarrow} B,$
   $*\{A, K_{ab}\}_{K_{bs}} \rightsquigarrow S \models A \overset{K_{ab}}{\leftrightarrow} B$
4. $B \triangleleft *\{A, K_{ab}\}_{K_{bs}} \rightsquigarrow S \models A \overset{K_{ab}}{\leftrightarrow} B,$
   $*\{N_b\} \rightsquigarrow A \models A \overset{K_{ab}}{\leftrightarrow} B$

In the above description, we have also added extensions which describe the beliefs held when the messages are sent. The following statements describe the initial protocol assumptions:

$A \ni K_{as};\ A \models A \overset{K_{as}}{\leftrightarrow} S;\ A \ni N_a;\ A \models \sharp(N_a);$
$A \models \phi(B)$

$B \ni K_{bs};\ B \models B \overset{K_{bs}}{\leftrightarrow} S;\ B \ni N_b;\ B \models \sharp(N_b);$
$B \models \phi(N_b);\ B \models A \overset{N_b}{\leftrightarrow} B$

$S \ni K_{as};\ S \models A \overset{K_{as}}{\leftrightarrow} S;\ S \ni K_{bs};\ S \models B \overset{K_{bs}}{\leftrightarrow} S;$
$S \ni K_{ab};\ S \models A \overset{K_{ab}}{\leftrightarrow} B$

That is, Alice possesses a secret $K_{as}$ and believes it is a secret between herself and $S$. Similarly, Bob possesses a secret $K_{bs}$ and believes it is a secret between himself and $S$. Each possesses a nonce and believes that it is fresh. Alice believes that the identifier $B$ is recognizable to her. Bob believes that $N_b$ is recognizable to him. Also, Bob believes that his nonce $N_b$ is a suitable secret with Alice. The server $S$ possesses valid keys $K_{as}$ and $K_{bs}$ with Alice and Bob, respectively. It also possesses a session key $K$ and believes $K$ is a suitable secret between Alice and Bob.

$A \models S \Rightarrow S \models *;\ A \models S \Rightarrow A \overset{K}{\leftrightarrow} B$

$B \models S \Rightarrow S \models *;\ B \models S \Rightarrow A \overset{K}{\leftrightarrow} B;$
$B \models A \Rightarrow A \models *$

Both Alice and Bob believe that $S$ is honest and competent. They also trust $S$ to invent a suitable secret key for them. Also, Bob believes that Alice is honest and competent.

For a run of the protocol, we apply the inference rules to the messages, as follows:

**Message 1:** From P1 we obtain $B \ni (A, N_a)$.

**Message 2:** From T2 and P1 we obtain $S \ni B$. From T2, T1, T3, and P1 we obtain $S \ni (A, N_a, N_b)$.

**Message 3:** The extension $S \models A \overset{K_{ab}}{\leftrightarrow} B$ attached to the two encrypted parts is valid because it holds by assumption. From T1, T3, T2, P1 and the first encrypted part, we obtain $A \ni K_{ab}$ and $A \ni N_b$.

From F1, R1, I1, J2, and J3 we obtain $A \models S \models A \overset{K_{ab}}{\leftrightarrow} B$. Hence, by J1, $A \models A \overset{K_{ab}}{\leftrightarrow} B$. We can thus include this statement in the extension attached to the second encrypted part of message 4.

**Message 4:** From T1, T3, T2, P1 and the first encrypted part we obtain $B \ni K_{ab}$. However, we cannot derive any beliefs from this part since the statement $B \models \sharp(\{A, K_{ab}\}_{K_{bs}})$ does not hold. In the actual working of the protocol, Bob deduces that $K_{ab}$ is shared with Alice if the decryption of the second encrypted part yields his nonce. By appealing to the GNY logic rules, we find that the only way we can proceed in the logic to reason in this manner is by first establishing that Bob believes the extension attached to the second encrypted part. However, none of the GNY logic rules enable this to be derived; the only applicable rule is I1 which cannot be applied, since it requires the recipient of an encrypted message to believe that the key used to encrypt the message is shared with another principal *a priori*.

### 2.2.4 Adding a new rule

The incompleteness revealed by the above analysis motivates us to propose the addition of the following new message interpretation rule to the logic:

$$\text{I8}\ \frac{P \triangleleft *\{X, <S>\}_K,\ P \ni K,\ P \models P \overset{S}{\leftrightarrow} Q,\ P \models \phi(X, S),\ P \models \sharp(X, S, K)}{P \models Q \sim (X, <S>),\ P \models Q \sim \{X, <S>\}_K, P \models Q \ni K}$$

That is, suppose that for principal $P$ all of the following conditions hold: (1) $P$ receives a formula consisting of $X$ concatenated with $S$, encrypted with key $K$ and marked with a not-originated here mark; (2) $P$ possesses $K$; (3) $P$ believes $S$ is a suitable secret for himself and $Q$; (4) $P$ believes that $X$ concatenated with $S$ is recognizable; (5) $P$ believes that at least one of $S$, $X$, or $K$ is fresh. Then $P$ is entitled to believe that: (1) $Q$ once conveyed the formula $X$ concatenated with $S$; (2) $Q$ once conveyed the formula $X$ concatenated with $S$ and encrypted with $K$; (3) $Q$ possesses $K$. (A similar rule can be added along previous lines to the set of "never-originated-here" rules of the

GNY logic.) The new rule I8 enables us to derive Bob's beliefs in the validity of $K_{ab}$, as follows:

**Message 4 (continued):** From I8 and the second encrypted part we obtain $B \models A \ni K_{ab}$, and $B \models A \mid\sim \{N_b\}_{K_{ab}} \rightsquigarrow A \models A \overset{K_{ab}}{\leftrightarrow} B$. From F2, J2, and J3 we obtain $B \models A \models A \overset{K_{ab}}{\leftrightarrow} B$. We can include the statement $A \models S \models A \overset{K_{ab}}{\leftrightarrow} B$ in the extension attached to the second encrypted part, since this statement holds from message 3. We also need an additional assumption which reflects Bob's trust in Alice to pass on the session key from the server: $B \models A \Rightarrow (S \models A \overset{K_{ab}}{\leftrightarrow} B)$. This assumption is the logical embodiment of a curious feature of the protocol: Alice can make Bob believe in a replayed session key. Notice that the statement of the assumption essentially amounts to Bob believing that this does not take place. The fact that we are forced to make the odd assumption explicit during the analysis provides a good example of the virtue of the logic. (The above protocol feature also emerges from the BAN logic analysis of the Yahalom protocol, cf. [1, p. 33].) From I8, J2, J3, and J1 we finally obtain $B \models A \overset{K_{ab}}{\leftrightarrow} B$.

To conclude our analysis of the Yahalom protocol, we list the final position attained:

$A \ni K_{ab}$; $A \models A \overset{K_{ab}}{\leftrightarrow} B$

$B \ni K_{ab}$; $B \models A \overset{K_{ab}}{\leftrightarrow} B$; $B \models A \ni K_{ab}$

$B \models A \models A \overset{K_{ab}}{\leftrightarrow} B$

Both Alice and Bob possess the session key and believe in it. In addition, Bob believes that Alice possesses the session key and believes in it.

The above analysis shows an interesting point: it hints at a possible redundancy in the last message of the protocol. The analysis tells us that no beliefs about $K_{ab}$ are derived for Bob from the encrypted part which Alice forwards him from the server in message 4: $\{A, K_{ab}\}_{K_{bs}}$. Also notice that Bob binds the identity claimed by Alice to his nonce, by concatenating them and encrypting with $K_{bs}$ in message 2. Apparently then, in the last message of the protocol, Bob's nonce not only assures him of the freshness of the encrypted half sent by Alice, but also guarantees that $K_{ab}$ is shared with Alice. Since Bob decrypts the encrypted part forwarded by Alice only to gain possession of $K_{ab}$, we can delete Alice's name from this part:

$$3'. \quad S \rightarrow A : \{B, K_{ab}, N_a, N_b\}_{K_{as}}, \{K_{ab}\}_{K_{bs}}$$
$$4'. \quad A \rightarrow B : \{K_{ab}\}_{K_{bs}}, \{N_b\}_{K_{ab}}$$

An analysis of the modified protocol using the logic confirms our above intuition: the same final position as the original protocol is achieved by the modified protocol.

## 2.3 Redundancy in the logic

In this section, we give an example of a rule which contains a redundant premise. Observe that the message interpretation rule I2 includes the following statements as premises: (1) $P \triangleleft *\{X, <S>\}_{+K}$; (2) $P \ni -K$; and (3) $P \ni S$. In the original rule, (2) and (3) are combined into one single premise using the conjunction operator ',': $P \ni (-K, S)$. We replace this premise by (2) and (3) only for the sake of convenience. It is easy to see that (3) follows from (1) and (2): From T1 and (1), $P \triangleleft \{X, <S>\}_{+K}$, and therefore, from (2) and T4, $P \triangleleft (X, <S>)$. Hence, by T2, $P \triangleleft S$, and so, by P1, $P \ni S$. Thus, we see that the premise (3) of the above rule is redundant. (The message interpretation rule I2′ exhibits a similar redundancy.)

# Chapter 3

# A modification of the GNY logic for automatic analysis of protocols

This chapter proposes a modified GNY logic, and describes the implementation of a protocol analysis tool based on that logic. The modifications are designed to allow the logical statements derivable from any protocol represented by a finite set of statements to be deduced in a finite number of steps, without losing any useful inferences. The tool can be used to automatically generate proofs of statements representing protocol goals.

(Parts of this chapter appeared in preliminary form elsewhere [32].)

## 3.1 Introduction

The BAN and GNY logics can be used to effectively explain the working of protocols. Very often a protocol analysis using the logics reveals missing assumptions or deficiencies in the protocol begin analyzed. This can lead to the assumptions or the original protocol being revised and the inference rules being reapplied to determine if the desired goal is then attainable. The process of applying and reapplying the inference rules, however, is in practice often tedious and error-prone to do by hand. Several tools which relieve the manual burden of carrying out this task for the BAN logic or modified versions of it can be found in the literature; see, for example, [16], [33], [34], [35]. The appeal of tools for mechanical validation is clear, but such tools can also assist in examining the role played by protocol messages and assumptions in attaining the desired goal. In addition, such tools can also be used to verify proofs of protocol goals which are obtained by manually applying the logic. Manual analysis of protocols using the GNY logic is particularly unwieldy, as the logic has more than forty inference rules. Moreover, the GNY logic operates at a finer level than its predecessor, so proofs of protocol goals in the logic typically work out to be much longer than their BAN logic counterparts.

Our main aim in automating the logic is to be able to mechanically determine whether one or more statements describing the goal of a protocol are derivable using the logic from some initial assumptions. Furthermore, it is also desirable to obtain all statements that are derivable from the initial assumptions. This allows us to analyze the state of the principals after the execution of each protocol step. We therefore use a forward-chaining strategy in automating the logic. This involves repeated application of the inference rules of the logic to the set of statements consisting of the idealized protocol, initial assumptions, and derived statements, until all statements derivable are obtained. However, many of the inference rules of the original GNY logic are unsuitable for forward-chaining. The problem is clear just from the freshness rule F1,

$$\frac{P \models \sharp(X)}{P \models \sharp(X,Y)},$$

which essentially says that if $X$ is a fresh message and $X$ is concatenated with any other message $Y$, then the resulting message is also fresh. (The rule has one more conclusion; however, the one shown suffices to illustrate the problem.) It is easy to see that this rule can be used to derive an infinite set of statements starting from a finite set of statements of the form $P \models \sharp(X)$. Although the inference expressed by this rule is intuitive and desirable, it is necessary to restrict the application of the rule for the purposes of forward-chaining.

We will show that the set of inference rules can be modified in such a way that the statements derivable from any protocol represented by a finite set of statements are also finite in number. Essentially, the point of our modifications is to convert the set of rules into a form which is directly amenable to forward-chaining. The modifications are designed to produce a restricted logic in the sense that the modified logic does not capture all inferences which are possible in the GNY logic. However, we will argue that the inferences lost by the modified logic do not affect our central aim in using the logic: that is, to reason about a principal's possessions and his beliefs about the statements conveyed by other principals, based on the messages received by the principal. In other words, we can still use the modified logic to analyze protocols with the same intended effect as the original GNY logic.

## 3.2 Modifying the GNY rule set

We now describe the modifications to the set of inference rules of the GNY logic, which we make in order to obtain finiteness of derivations. Additionally, we include several

new rules which are clearly required during protocol analyses using the logic, but which are nonetheless absent from the original GNY logic [2]. The resulting set of rules is given in appendix C, and a proof of finiteness of derivations for this rule set is given in the next section.

### 3.2.1 Adding new rules

We add three new rules all of which enable dropping of extensions attached to formulae:

$$\text{T7}\quad \frac{P \triangleleft X \rightsquigarrow C}{P \triangleleft X}$$

$$\text{I8}\quad \frac{P \models Q \mid\sim X \rightsquigarrow C}{P \models Q \mid\sim X}$$

$$\text{I9}\quad \frac{P \models Q \mid\sim X \rightsquigarrow (C, C')}{P \models Q \mid\sim X \rightsquigarrow C}$$

While the above rules capture rather trivial inferences, these rules are nevertheless required during protocol analyses. The first two rules, T7 and I8, are also present in an extended version of the GNY logic found in Gong's thesis [3]; the role played by these rules should be intuitively clear. Surprisingly, the rule I9 is absent from both the original GNY logic [2] and Gong's extension [3]. Essentially, I9 enables the splitting of message extensions which are conjunctions of two or more statements. The logical use of this rule's conclusion is made in the jurisdiction rule J2 where it appears as a premise (see appendix C.6). A handy example of the need for I9 can be seen from the analysis of the Yahalom protocol given in the previous chapter: there we tacitly made use of this rule in proceeding with the derivation of the statement $B \models A \overset{K_{ab}}{\leftrightarrow} B$ from the statement $B \models A \mid\sim \{N_b\}_{K_{ab}} \rightsquigarrow (A \models A \overset{K_{ab}}{\leftrightarrow} B, A \models S \models A \overset{K_{ab}}{\leftrightarrow} B)$.

### 3.2.2 Modifying existing rules

Like F1, several other freshness and recognizability rules cause problems by repeated application. We deal with this problem by modifying these rules into a form suitable for forward-chaining. Notice that every freshness and recognizability rule has a conclusion of the form $P \models \sharp(X)$ and $P \models \phi(X)$, respectively. Our modification introduces an additional premise of the form $P \ni X$ in each of these rules; the modified freshness and recognizability rules are listed in appendices C.3 and C.4, respectively. We now discuss the rationale behind the modifications to the freshness rules; the modifications to the recognizability rules are explained similarly.

The idea behind the modifications to the freshness rules is to limit the original rules to allow only those inferences which contribute to our purpose of reasoning about a principal's possessions and beliefs about the statements conveyed by other principals. Evidently, from the possession rules of the GNY logic we see that a principal $P$'s possessions do not depend on $P$'s beliefs. Therefore what we can conclude using a freshness rule for $P$ is of no use for the above purpose if it does not affect $P$'s beliefs about statements conveyed by others. Essentially, the rule which enables us to obtain such beliefs for $P$ is the jurisdiction rule J2, which has a premise of the form $P \models Q \mid\sim (X \rightsquigarrow C)$. This premise reflects the requirement that $P$ can only obtain beliefs from messages sent by some well-known principal $Q$, and appears as a conclusion of the message interpretation rules I1, I2, I3, I4, I1′, I2′, and I3′. Hence the statement $P \models \sharp(X)$ is of significance in deriving $P$'s beliefs in statements conveyed by others only if it appears as a premise in one of these rules. Out of these rules, only I1, I2, and I3 have a freshness premise. Further, each of these rules satisfies the following property: if $P \models \sharp(X_1, \ldots, X_m)$ is the freshness premise of the rule, then $P \ni X_i$ for $i = 1, \ldots, m$. For example, take I1; this rule has a freshness premise of the form $P \models \sharp(X, K)$. Since this premise is meant to *denote* $P \models \sharp(X)$ or $P \models \sharp(K)$ ([2], p. 245), we can replace I1 by the following two equivalent rules:

$$\text{I1}' \quad \frac{P \triangleleft *\{X\}_K,\ P \ni K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models \phi(X),\ P \models \sharp(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K,\ P \models Q \ni K}$$

$$\text{I1}'' \quad \frac{P \triangleleft *\{X\}_K,\ P \ni K,\ P \models P \overset{K}{\leftrightarrow} Q,\ P \models \phi(X),\ P \models \sharp(K)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K,\ P \models Q \ni K}$$

It is easy to see that in both I1' and I1", $P$ possesses the formula appearing in the corresponding freshness premise:

**I1′:** From the premise $P \triangleleft *\{X\}_K$ and rule T1, $P \triangleleft \{X\}_K$, and therefore, from the premise $P \ni K$ and rule T3, $P \triangleleft X$. So, by P1, $P \ni X$, as required.

**I1″:** Trivially, $P \ni K$ holds as a premise.

### 3.2.3 Deleting existing rules

We delete several possession rules of the logic: P2, P4, P6, P7 and P8. Each of these rules can be applied indefinitely to derive new possessions. For example, suppose the statements $P \ni X$ and $P \ni K$ hold. Then we can use the possession rule P6 for shared

keys,

$$\frac{P \ni K,\ P \ni X}{P \ni \{X\}_K,\ P \ni \{X\}_K^{-1}},$$

to derive the infinite set of statements: $P \ni \{X\}_K$, $P \ni \{\{X\}_K\}_K$, .... The above rules are evidently useful in enforcing the possession consistency check, but their role during protocol analyses otherwise is not so clear. Furthermore, we do not include this check in automating the logic, since it is intended to be performed outside of the logic. For our purposes, we simply find it convenient to delete these rules. Similarly, we delete the GNY logic rationality rule, which states that if $\frac{C_1}{C_2}$ is a rule, then so is $\frac{P \models C_1}{P \models C_2}$, for any principal $P$.

## 3.3 Finiteness of derivations

In this section, we prove that for the modified rule set given in appendix C:

> *The statements derivable from a finite set of idealized protocol steps and initial assumptions are finite in number, and are therefore derivable in a finite number of steps.*

Essentially, we will follow a a technique used by Engberg [16] to prove a similar property for a modified version of the BAN logic.

We begin with a set-theoretic formulation of the statement which we wish to prove. To this end, it is convenient to introduce the notation $(\mathcal{D}/E)$ to denote a generic inference rule, where $\mathcal{D}$ is the set consisting of the premises of the rule and $E$ is the conclusion of the rule; here we assume that rules with multiple conclusions are decomposed in the obvious way into separate rules, each with a single conclusion. Denote by $\mathcal{R}$ the modified set of rules. We define an operator $\rho$ on sets of statements, as follows: for any set of statements $\mathcal{S}$,

$$\rho(\mathcal{S}) = \mathcal{S} \cup \{E:\ \text{there exists } (\mathcal{D}/E) \in \mathcal{R} \text{ such that } \mathcal{D} \subseteq \mathcal{S}\}.$$

Thus $\rho$ returns $\mathcal{S}$ together with the statements derivable from $\mathcal{S}$ by applying the inference rules in $\mathcal{R}$ exactly once. The main idea of the proof is an argument showing that there exists an $n$ such that

$$\rho^n(\mathcal{S}) = \rho^\infty(\mathcal{S}),$$

where we use $\rho^\infty(\mathcal{S})$ to denote the infinite union $\cup_{m=0}^{\infty} \rho^m(\mathcal{S})$.

Recall that any statement of the logic is a statement of the form $P \triangleleft X$ or $P \ni X$ or $P \models\!\!\equiv C$. The key step in the argument is to construct a well-founded and finitary relation over the set of statements of the forms $P \ni X$, $P \triangleleft X$, and $P \models\!\!\equiv C$. We construct the required relation, denoted by $\prec$, in terms in terms of six subsidiary relations $\prec_{\triangleleft}$, $\prec^{\triangleleft}_{\ni}$, $\prec_{\ni}$, $\prec^{\ni}_{\models\!\!\equiv}$, $\prec^{\triangleleft}_{\models\!\!\equiv}$ and $\prec_{\models\!\!\equiv}$, as follows:

$$
\begin{array}{lll}
(1) & P \triangleleft X \prec P \triangleleft Y & \text{if } X \prec_{\triangleleft} Y \\
(2) & P \ni X \prec P \triangleleft Y & \text{if } X \prec^{\triangleleft}_{\ni} Y \\
(3) & P \ni X \prec P \ni Y & \text{if } X \prec_{\ni} Y \\
(4) & P \models\!\!\equiv C \prec P \ni X & \text{if } C \prec^{\ni}_{\models\!\!\equiv} X \\
(5) & P \models\!\!\equiv C \prec P \triangleleft X & \text{if } C \prec^{\triangleleft}_{\models\!\!\equiv} X \\
(6) & P \models\!\!\equiv C \prec P \models\!\!\equiv D & \text{if } C \prec_{\models\!\!\equiv} D
\end{array}
$$

The definitions of the subsidiary relations are derived from suitably chosen classes of rules and are given below.

The definition of $\prec_{\triangleleft}$ is read off the *being-told* rules T1, T2, T3, T4, T5, T6 and T7 (see appendix C.1), where T2 and T5 are used in their two symmetrical forms, giving clauses as follows:

$$
\begin{array}{ll}
(1) & X \prec_{\triangleleft} {*X} \\
(2)(\text{i}) & X \prec_{\triangleleft} (X, Y) \\
\quad (\text{ii}) & X \prec_{\triangleleft} (Y, X) \\
(3) & X \prec_{\triangleleft} \{X\}_K \\
(4) & X \prec_{\triangleleft} \{X\}_{+K} \\
(5)(\text{i}) & X \prec_{\triangleleft} F(X, Y) \\
\quad (\text{ii}) & X \prec_{\triangleleft} F(Y, X) \\
(6) & X \prec_{\triangleleft} \{X\}_{-K} \\
(7) & X \prec_{\triangleleft} X \rightsquigarrow C
\end{array}
$$

The definition of $\prec^{\triangleleft}_{\ni}$ consists of a single clause which is read off the *possession* rule P1 (see appendix C.2):

$$(1) \quad X \prec^{\triangleleft}_{\ni} X$$

The definition of $\prec_{\ni}$ is read off the *possession* rules P3 and P5 (see appendix C.2), where both the rules are used in their two symmetrical forms, giving clauses as follows:

$$
\begin{array}{ll}
(1)(\text{i}) & X \prec_{\ni} (X, Y) \\
\quad (\text{ii}) & X \prec_{\ni} (Y, X) \\
(2)(\text{i}) & X \prec_{\ni} F(X, Y) \\
\quad (\text{ii}) & X \prec_{\ni} F(Y, X)
\end{array}
$$

The definition of $\prec^{\ni}_{\models}$ consists of two clauses. The first clause is read off each of the *freshness* rules F1′, F1″, F2′, F2″, F3′, F4′, F5′, F6′, F7′, F7″, F8′, F9′, F10′, and F11′ (see appendix C.3). The second clause is read off each of the *recognizability* rules R1′, R1″, R2′, R2″, R3′, R4′, and R5′ (see appendix C.4).

$$(1)\quad \sharp(X) \prec^{\ni}_{\models} X$$
$$(2)\quad \phi(X) \prec^{\ni}_{\models} X$$

The definition of $\prec^{\triangleleft}_{\models}$ is read off: (A) the *message interpretation* rules I1, I2, I3, I4 and I5 (see appendix C.5), and (B) the rules for *never-originated-here* messages I1′, I2′ and I3′ (see appendix C.7). Each rule contributes as many clauses to the definition as the number of conclusions in the rule, giving clauses as follows:

$$\begin{array}{lrcl}
(1)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & *\{X\}_K \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash \{X\}_K \rightsquigarrow C & \prec^{\triangleleft}_{\models} & *\{X\}_K \rightsquigarrow C \\
(\mathrm{iii}) & Q \ni K & \prec^{\triangleleft}_{\models} & *\{X\}_K \rightsquigarrow C \\
(2)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & *\{X\}_{+K} \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash \{X\}_{+K} \rightsquigarrow C & \prec^{\triangleleft}_{\models} & *\{X\}_{+K} \rightsquigarrow C \\
(\mathrm{iii}) & Q \ni +K & \prec^{\triangleleft}_{\models} & *\{X\}_{+K} \rightsquigarrow C \\
(3)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & *H(X) \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash H(X) \rightsquigarrow C & \prec^{\triangleleft}_{\models} & *H(X) \rightsquigarrow C \\
(4)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & \{X\}_{-K} \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash \{X\}_{-K} \rightsquigarrow C & \prec^{\triangleleft}_{\models} & \{X\}_{-K} \rightsquigarrow C \\
(5)(\mathrm{i}) & Q \ni -K & \prec^{\triangleleft}_{\models} & \{X\}_{-K} \\
(\mathrm{ii}) & Q \ni X & \prec^{\triangleleft}_{\models} & \{X\}_{-K} \\
(6)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & \{X\}_K \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash \{X\}_K \rightsquigarrow C & \prec^{\triangleleft}_{\models} & \{X\}_K \rightsquigarrow C \\
(7)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & \{X\}_{+K} \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash \{X\}_{+K} \rightsquigarrow C & \prec^{\triangleleft}_{\models} & \{X\}_{+K} \rightsquigarrow C \\
(8)(\mathrm{i}) & Q \vdash X & \prec^{\triangleleft}_{\models} & H(X) \rightsquigarrow C \\
(\mathrm{ii}) & Q \vdash H(X) \rightsquigarrow C & \prec^{\triangleleft}_{\models} & H(X) \rightsquigarrow C
\end{array}$$

The definition of $\prec_{\models}$ is read off: (A) the *message interpretation* rules I6, I7, I8 and I9 (see appendix C.5), where I7 and I9 are used in their two symmetrical forms, and (B)

the *jurisdiction* rules J1, J2 and J3 (see appendix C.6), giving clauses as follows:

$$
\begin{array}{lrcl}
(1) & Q \ni X & \prec_{\models} & Q \mathrel{|\!\sim} X \\
(2)(\mathrm{i}) & Q \mathrel{|\!\sim} X & \prec_{\models} & Q \mathrel{|\!\sim} (X, Y) \\
\quad (\mathrm{ii}) & Q \mathrel{|\!\sim} X & \prec_{\models} & Q \mathrel{|\!\sim} (Y, X) \\
(3) & Q \mathrel{|\!\sim} X & \prec_{\models} & Q \mathrel{|\!\sim} X \rightsquigarrow C \\
(4)(\mathrm{i}) & Q \mathrel{|\!\sim} X \rightsquigarrow C & \prec_{\models} & Q \mathrel{|\!\sim} X \rightsquigarrow (C, C') \\
\quad (\mathrm{ii}) & Q \mathrel{|\!\sim} X \rightsquigarrow C & \prec_{\models} & Q \mathrel{|\!\sim} X \rightsquigarrow (C', C) \\
(5) & C & \prec_{\models} & Q \models C \\
(6) & Q \models C & \prec_{\models} & Q \mathrel{|\!\sim} (X \rightsquigarrow C) \\
(7) & Q \models C & \prec_{\models} & Q \models Q \models C
\end{array}
$$

This completes the definitions of the six subsidiary relations. Of the six relations, the most critical in the analysis are: $\prec_{\triangleleft}$, $\prec_{\ni}$, and $\prec_{\models}$. It is easy to see that each of these three relations is well-founded. Consider the definition of the first of the three, $\prec_{\triangleleft}$: the formula on the left in each clause is syntactically shorter than the formula on the right, so there cannot be infinite descending chains with respect to $\prec_{\triangleleft}$. Well-foundedness of $\prec_{\ni}$ and $\prec_{\models}$ is equally easily proved. (Well-foundedness of the other three subsidiary relations $\prec_{\ni}^{\triangleleft}$, $\prec_{\models}^{\ni}$ and $\prec_{\models}^{\triangleleft}$ is not required.)

We proceed to show that $\prec$ is also well-founded; that is, there are no infinite descending chains with respect to $\prec$. If we show that: (*) any infinite descending chain with respect to $\prec$ must contain an infinite chain of statements of one of the three forms $P \ni X$, $P \triangleleft X$, or $P \models C$, then the well-foundedness of $\prec$ follows from the well-foundedness of $\prec_{\triangleleft}$, $\prec_{\ni}$, and $\prec_{\models}$. It remains to show that (*) holds. So assume there is an infinite descending chain $\cdots \prec C_3 \prec C_2 \prec C_1$, where each $C_i$ is a statement of one of the three forms $P \triangleleft X$, $P \ni X$, or $P \models C$. We say that $\prec_{\models}$ occurs at $C_i$ if $C_{i+1}$ is obtained from an application of clause (6) in the definition of $\prec$; and similarly in the case of the relations $\prec_{\models}^{\triangleleft}$, $\prec_{\models}^{\ni}$, $\prec_{\ni}$, $\prec_{\ni}^{\triangleleft}$, and $\prec_{\triangleleft}$.

Case (A): Suppose that $\prec_{\models}$ occurs at $C_n$ for some $n$. It follows that $\prec_{\models}$ also occurs at $C_m$ for all $m \geq n$; that is, there is an infinite descending chain with respect to $\prec_{\models}$.

Case (B): Suppose that $\prec_{\models}$ does not occur at $C_i$ for all $i$. It follows that $\prec_{\models}^{\ni}$ and $\prec_{\models}^{\triangleleft}$ do not occur at $C_i$ for all $i$.

Case (i): Suppose that $\prec_{\ni}$ occurs at $C_n$ for some $n$. It follows that $\prec_{\ni}$ occurs at $C_m$ for all $m \geq n$; that is, there is an infinite descending chain with respect to $\prec_{\ni}$.

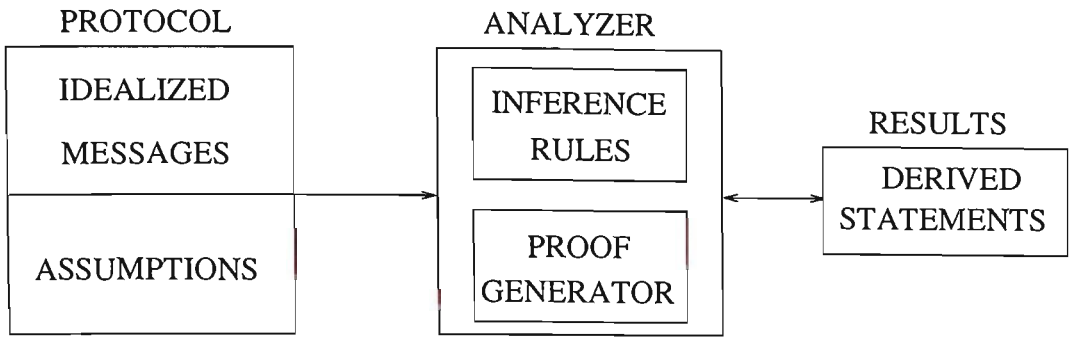


Figure 3.1: Protocol analysis

Case (ii): Suppose that $\prec_{\ni}$ does not occur at $C_i$ for all $i$. It follows that $\prec_{\ni}^{\triangleleft}$ does not occur at $C_i$ for all $i$. Therefore the only remaining case is that $\prec_{\triangleleft}$ occurs at $C_n$ for some $n$. It follows that $\prec_{\triangleleft}$ occurs at $C_m$ for all $m \geq n$; that is, there is an infinite descending chain with respect to $\prec_{\triangleleft}$.

A further property of all six subsidiary relations is that they are *finitary*; that is, given any statement $C$ of the form $P \ni X$ or $P \triangleleft X$ or $P \models C'$, the set of statements $\{D : D \prec C\}$ is finite. This is easy to see in the case of the relation $\prec_{\triangleleft}$, and is equally easily proved for the other five relations. It follows straightforwardly that $\prec$ is also finitary. Since $\prec$ is well-founded as well as finitary, it follows by König's lemma that for any $C$ the set of statements $\{D : D \prec^* C\}$, where $\prec^*$ denotes the transitive and reflexive closure of $\prec$, is finite as well.

Now the definitions of the subsidiary relations and of $\prec$ are constructed so as to give a straightforward guarantee that for each rule $(\mathcal{D}/E) \in \mathcal{R}$, there exists $C \in \mathcal{D}$ such that $E \prec C$; that is, in each rule the conclusion is smaller, with respect to $\prec$, than at least one of the premises. Therefore, if $\mathcal{S}$ is the set of idealized steps and initial assumptions of any protocol, then for every $C \in \rho^{\infty}(\mathcal{S})$, there exists $S \in \mathcal{S}$ such that $C \prec^* S$. Hence

$$\rho^{\infty}(\mathcal{S}) \subseteq \bigcup_{S \in \mathcal{S}} \{C : C \prec^* S\}.$$

Now the right-hand side above is a finite union, since by assumption the set $\mathcal{S}$ is finite. By the finitary property we established earlier, the set $\{C : C \prec^* S\}$, for any $S \in \mathcal{S}$, is finite as well. Therefore, we conclude that $\rho^{\infty}(\mathcal{S})$ is finite, as required.

## 3.4 Implementing the tool

We now outline an implementation of a tool based on the modified set of rules given in appendix C. The tool is implemented along similar lines as in our previous work on

| *Formula* | *Structure* |
|---|---|
| $(X,Y)$ | `[X,Y]` |
| $\{X\}_K$ | `encrypt(X, shared(K))` |
| $\{X\}_K^{-1}$ | `decrypt(X, shared(K))` |
| $\{X\}_{+K}$ | `encrypt(X, public(K))` |
| $\{X\}_{-K}$ | `decrypt(X, private(K))` |
| $H(X)$ | `h(X)` |
| $F(X_1,\ldots,X_n)$ | `f(X1, ..., Xn)` |
| $*X$ | `star(X)` |
| $X \rightsquigarrow C$ | `ext(X, C)` |
| $X$ | `ext(X, nil)` |

Table 3.1: Representing formulas

automating the BAN logic [35]. It consists of (1) an inference engine which produces the complete set of logical statements derivable from an input specification consisting of the idealized protocol and the initial assumptions, and (2) a routine to extract proofs from the database of derived statements. Since the modification we make to the original tool only concerns the representation of the logical syntax, we will mostly skip the details of the remaining parts of the implementation. Figure 3.1 gives an overall block diagram of how the tool is used in analyzing protocols [35]. As in the original tool, we use Prolog as an implementation language and represent the logical syntax in terms of Prolog structures.

### 3.4.1 Formulas and statements

In the logic, protocol messages are represented as formulas. The building blocks of messages are constants like principal names, keys, nonces, etc. We typically represent these constants by one or more lowercase letters. For example, a session key $K_{ab}$ for principals $A$ and $B$ is denoted by the Prolog atom `kab`. The remaining formulas like concatenation, encryption, functions, etc. are represented by Prolog structures chosen to represent their typographical counterparts wherever possible. We also use the structure `ext(X, nil)` to represent a formula $X$ without any extension. Table 3.1 shows how we represent the logical formulas by means of Prolog structures.

The statements of the logic are similarly represented by appropriately named Prolog structures as shown in Table 3.2. It is straightforward to translate any formula or statement in the logical syntax to its Prolog counterpart by looking up Tables 3.1 and 3.2.

| *Statement* | *Structure* |
|---|---|
| $P \triangleleft X$ | `told(P, X)` |
| $P \ni X$ | `possesses(P, X)` |
| $P \mid\sim X$ | `conveyed(P, X)` |
| $P \models \sharp(X)$ | `believes(P, fresh(X))` |
| $P \models \phi(X)$ | `believes(P, recognizes(X))` |
| $P \models Q \overset{S}{\leftrightarrow} R$ | `believes(P, secret(Q,S,R))` |
| $P \models \overset{+K}{\mapsto} Q$ | `believes(P, public(K,Q))` |
| $P \models C$ | `believes(P, C)` |
| $P \models Q \Rightarrow C$ | `believes(P, controls(Q,C))` |
| $P \models Q \Rightarrow Q \models *$ | `believes(P, honest(Q))` |
| $C_1, C_2$ | `[C1, C2]` |

Table 3.2: Representing statements

### 3.4.2 Derived statements

Apart from representing the logical constructs in Prolog syntax, we also need to maintain derivation information about statements obtained by applying the inference rules. The predicate `fact/3` which represents an inference step is used for this purpose. It takes the form:

```
fact(Index, Stat, reason(PremIs, Rule))
```

Here the integer argument `Index` is used to index instances of `fact/3`. The second argument `Stat` is bound to a derived statement. In the last argument, `PremIs` is a list containing the indices of premises used in deriving `Stat` by an application of rule `Rule`.

### 3.4.3 Logical rules

The representation of the inference rules is best explained by means of an example; the being-told rule T1 is defined by the following clause for `told/2`:

```
told(told(P, X), reason([I], 'T1')) :-
        fact(I, told(P, star(X)), _).
```

## 3.5 Using the tool: an example

We now demonstrate the use of the tool by means of one of the protocol analyzed in the GNY logic paper [2]: the voting protocol. Our aim here is to illustrate how the

tool confirms a problem with the parsing scheme, which we discussed in the previous chapter.

The idealized protocol is given as follows (p. 239):

1. $P_i \triangleleft *N_q$
2. $Q \triangleleft *P_i, *N_i, *V_i, *H(N_q, *<S_i>, V_i)$
3. $P_i \triangleleft *R, *H(N_i, <S_i>, R)$

Here $S_i$ is a secret between $P_i$ and $Q$, and $N_i$ and $N_q$ are nonces generated by $P_i$ and $Q$, respectively.

The following statements describe the protocol assumptions:

$$P_i \ni S_i; \quad P_i \ni N_i; \quad P_i \models Q \overset{S_i}{\leftrightarrow} P_i; \quad P_i \models \sharp(N_i)$$
$$Q \ni S_i; \quad Q \ni N_q; \quad Q \models Q \overset{S_i}{\leftrightarrow} P_i; \quad Q \models \sharp(N_q)$$

It is straightforward to convert the above statements into the syntax of the tool using Tables 1 and 2. For example, the second idealized message is represented by the Prolog fact:

```
fact(2, told(q, [star(pi), star(ni), star(vi), ext(star(h([nq, star(si), vi])),
nil)]), reason([], 'Step')).
```

The set of Prolog facts representing the idealized protocol and the initial assumptions is then loaded into the analyzer to obtain all the logical statements derivable:

```
?- analyze(voting).
Analyzed in 4 cycles
```

The database of facts can now simply be queried to determine whether a particular goal statement is attained or not. For example, according to the GNY paper, the statements $Q \models P_i \mid\sim V_i$ and $P_i \models Q \mid\sim R$ hold for the protocol (p. 239). The following queries can be used to verify this:

```
?- fact(I, believes(q,conveyed(pi,vi)), Rule).
no
?- fact(I, believes(pi,conveyed(q,r)), Rule).
I = 37
Rule = reason([33],I7);
yes
```

The output of the above queries show that $Q \models P_i \mid\sim V_i$ does not hold, whereas $P_i \models Q \mid\sim R$ holds. It is not difficult to explain the discrepancy behind this mismatch. Looking at the analysis sketched in the GNY paper, we see that the conclusion

$Q \models P_i \mid\sim V_i$ is obtained from the message interpretation rule I3 and the second message. It is easy to see that in the premises of the intended application of the rule, the secret $S_i$ appears prefixed with a $*$; for example, one of the premises works out to be: $Q \models Q \overset{*S_i}{\leftrightarrow} P_i$. Although we have the statement $Q \models Q \overset{S_i}{\leftrightarrow} P_i$ as an initial assumption, we see that the statement $Q \models Q \overset{*S_i}{\leftrightarrow} P_i$ does not follow from the logic. As we discussed in the previous chapter, this difficulty is best dealt with by modifying the parsing scheme so that non-encrypted and non-hashed message parts are not marked with $*$'s. The desired statement $Q \models P_i \mid\sim V_i$ is immediately derived once we alter the idealization of the second message to reflect this change:

```
fact(2, told(q, [pi, ni, vi, ext(star(h([nq, si, vi])), nil)]), reason([],
                                        'Step')).
```

```
| ?- fact(I, believes(q, conveyed(pi, vi)), Rule).
I = 37
Rule = reason([31],I7);
no
```

The proof explanation routine can be further used to obtain explicit representations of proofs of derived statements. For example, we obtain the following machine-generated proof of $Q \models P_i \mid\sim V_i$:

```
?- explain_proof(believes(q, conveyed(pi,vi))).
1. told(q,[pi,ni,vi,ext(star(h([nq,si,vi])),
   nil)]) {Step}
2. told(q,vi) {1, T2}
3. possesses(q,vi) {2, P1}
4. possesses(q,si) {Assumption}
5. possesses(q,nq) {Assumption}
6. believes(q,fresh(nq)) {Assumption}
7. believes(q,secret(q,si,pi)) {Assumption}
8. told(q,ext(star(h([nq,si,vi])),nil)) {1, T2}
9. believes(q,conveyed(pi,[nq,si,vi])) {8, 7,
   6, 5, 4, 3, I3}
10. believes(q,conveyed(pi,vi)) {9, I7}
```

# Chapter 4

# Semantic foundations for authentication logics

The motivation for this chapter is perhaps best described by the title of a note written by Tuttle [28]: "*Flaming in Franconia: Build models, not logics.*" Broadly speaking, the problem statement is as follows: To develop a model capable of providing a semantic basis for BAN-like logics, but which is essentially independent of any such logic itself.

In constructing a model in this chapter, we will attempt to isolate and formalize the semantics of some of the notions found in existing logics, without appealing too closely to the logical formalisms themselves. In the next chapter, we will devise a logic based on the model constructed here.

## 4.1 Informal groundwork

We begin by reviewing some of the notions which form the mainstay of existing models for BAN-like logics, namely the models due to Abadi and Tuttle [8], Syverson and van Oorschot [9], and Wedel and Kessler [10]. Our intention is to highlight some of the problems that arise in defining such notions semantically and to lay down some groundwork for the model which we will construct in the next section.

As in existing works, we are interested in modeling a system of communicating, message-passing principals. We assume that principals can perform some actions; for convenience we divide the class of actions into two: *communication actions*, and *message-construction actions*. For example: (1) The actions of *sending* and *receiving* a message belong to the former class; (2) The action of *constructing* a constant term such as the name of a principal belongs to the latter class.

The notions which are central to our model are those that associate various sets of messages with principals: *possessed messages*, *seen messages*, and *said messages*.

### 4.1.1 Possessed messages

The notion of *possessed messages* essentially reflects the following intuition: (1) if a principal $P$ receives or constructs a message $X$, then $X$ is possessed by $P$; (2) any message that can be possibly computed by $P$ from $P$'s possessed messages is also possessed by $P$. Motivated by the above intuition, we can proceed to construct a definition for the set of $P$'s possessed messages as follows: we form the set of all the messages which $P$ receives or constructs and close this set off under the operations that are available to $P$ within the system. We make the assumption that the available operations include: keyed encryption function $E_{\_}(\cdot)$, keyed hashing function $H_{\_}(\cdot)$, and concatenation function $\_ \mid \cdots \mid \_$. To continue the previous definition we can require, for example, that if a message $X$ and a key $K$ are in the set of $P$'s possessions, then so is the message $E_K(X)$; of course, this also implies that so are the messages $E_K(E_K(X))$, $E_K(E_K(E_K(X)))$, .... However, the unbounded nature of this definition makes the set of $P$'s possessed messages infinite; this means potentially all messages are possessed by $P$. Furthermore, it introduces arbitrary messages, which does not seem necessary for reasoning about messages that are actually constructed within the system. As in existing approaches, we fix the set of $P$'s possessed messages for each time $t$, but we do not allow this set to be infinite in our model. We will employ a limited notion of possession, which works as follows. In defining the set of possessed messages for a given time $t$, we restrict the closure operation to admit only those messages which *occur* in the system at that time. Intuitively, a message occurs at time $t$, if it was constructed by any principal at a time earlier than $t$. A characteristic property of the resulting definition is that the set of possessed messages is finite. As we shall see in the next section, our definition also has many other interesting properties which appear quite natural.

The notion of a message being constructed in the system also enables us to formulate the assumption that 'accidents' do not happen. That is, we treat what is highly improbable as impossible: we shall assume that a message can be constructed in the system in only one way. For example, if a message is constructed as an encryption then our assumption guarantees that the same message cannot be constructed as a concatenation. As another example, if a message is constructed as an encryption of $X$ using $K$, then the same message cannot be constructed as an encryption of $X'$ using $K'$, unless $X = X'$ and $K = K'$. The assumption which rules out chance equality between messages is crucial for our definitions to make sense. For example, one part of the closure operation that we will use in defining $P$'s set of possessed messages at

time $t$ roughly captures the following statement:

(*) if a message $Y$ is in this set and $Y = E_K(X)$ for some $K$, $X$ such that $K^{-1}$ is also in this set, then so is $X$, provided that some principal has encrypted $X$ using $K$, and thus constructed $E_K(X)$, at a time earlier than $t$.

Essentially, it is by virtue of the assumption which says that messages can be constructed in only one way that we can fix $X$ as intended in the above statement. Of course, this assumption cannot hold with certainty in the real world. However, it simply reflects an idealization and is not unrealistic to make for our purposes.

The statement (*) above reflects an example of how we capture decryptions in the model. The role of deconcatenations is captured similarly; we will give an example of this below.

### 4.1.2 Seen messages

The notion of *seen messages* is somewhat more restrictive than that of possessed messages. It essentially reflects what messages can be extracted by a principal from the messages it receives: (1) if a principal $P$ receives a message $X$, then $X$ is seen by $P$; (2) any message that can be possibly extracted from $P$'s seen messages, perhaps using keys possessed by $P$, is also seen by $P$. The idea behind (2) is expanded as follows: (2′) if a message $Y$ is seen by $P$ and $Y = E_K(X)$ for some $X$, $K$ such that $K^{-1}$ is possessed by $P$, then $X$ is seen by $P$; and (2″) if a message $Y$ is seen by $P$ and $Y = X_1 \mid \cdots \mid X_k$ for some $X_1, \ldots, X_k$, then $P$'s seen messages include $X_i$ for all $i$. As with the set of $P$'s possessed messages, we fix the set of $P$'s seen messages for each time $t$. The closure operation that we will use in defining the set of $P$'s seen messages at time $t$ has essentially the following two properties:

(**) if a message $Y$ is in this set and $Y = E_K(X)$ for some $X$, $K$ such that $K^{-1}$ is in the set of $P$'s possessed messages at time $t$, then $X$ is in the set of $P$'s seen messages at time $t$, provided that some principal has encrypted $X$ using $K$, and thus constructed $E_K(X)$, at a time earlier than $t$, and

(***) if a message $Y$ is in this set and $Y = X_1 \mid \cdots \mid X_k$ for some $X_1, \ldots, X_k$, then so are $X_1, \ldots, X_k$, provided that some principal has concatenated $X_1, \ldots, X_k$, and thus constructed $X_1 \mid \cdots \mid X_k$, at a time earlier than $t$.

Again our assumption which says that messages can be constructed in only one way is crucial to the intended meaning of the statements (**) and (***) above. Notice that (**) shows an example of how we capture deconcatenations in the model.

### 4.1.3 Said messages

The notion of *said messages* essentially reflects the following intuition: (1) if a principal $P$ sends a message $X$, then $X$ is said by $P$; (2) if $X$ is said by $P$, then so are the messages from which $X$ was immediately constructed, if those messages are possessed by $P$. Intuitively, if $X$ is a message then the messages from which $X$ was immediately constructed are those messages that allow $X$ to be obtained as the output of a single message-construction action. For example, to construct an encrypted message $E_K(X)$ the immediate messages that are needed are $X$ and $K$. We emphasize that the notion of 'immediate messages' is not inductive in nature: in the previous example $X$ could itself have been constructed as an encrypted message $E_{K'}(X')$; however, $X'$ and $K'$ are not amongst the immediate messages from which $E_K(X)$ was constructed. In our model, we will fix the set of $P$'s said messages for each time $t$. To define the set of said messages along the above lines, we need to capture the notion of immediate messages. This is done simply in terms of the notion of a message being constructed in the system. For example, one part of the closure operation that we will use in defining $P$'s set of said messages at time $t$ roughly captures the following statement:

(†) if a message $Y$ is in this set and $Y = E_K(X)$ for some $X$, $K$ such that $X$ and $K$ are in the set of $P$'s possessed messages at time $t$, then $X$ and $K$ are in the set of $P$'s said messages at time $t$, provided that some principal has encrypted $X$ using $K$, and thus constructed $E_K(X)$, at a time earlier than $t$.

## 4.2 A computational model of communicating principals

Let $\Sigma$ be a finite alphabet, and let $\mathcal{M} = \Sigma^*$ be the set of all messages. For simplicity, we take $\Sigma = \{0, 1\}$; the set $\mathcal{M}$ then consists of all binary strings of finite length. Let a finite set of *principal names* $\mathcal{P} \subseteq \mathcal{M}$ be fixed; henceforth we always refer to principal names simply as *principals*. Let a set of *nonces* $\mathcal{N} \subseteq \mathcal{M}$ be fixed. Let the set of all *possible keys* $\mathcal{K} \subseteq \mathcal{M}$ be fixed. For each key $K \in \mathcal{K}$, we assume a one-to-one function $E_K : \mathcal{M} \to \mathcal{M}$ is fixed, which we call a *keyed encryption function.* Assume a set $\mathcal{K}^{-1} \subseteq \mathcal{M}$ is fixed along with a one-to-one onto function $^{-1} : \mathcal{K} \to \mathcal{K}^{-1}$. For each $K \in \mathcal{K}$, we assume a non-invertible function $H_K : \mathcal{M} \to \mathcal{M}$ is fixed, which we call a *keyed hash function.* For each natural number $m > 1$, we use the symbol $|^m$ to represent $m$-fold concatenation function over $\Sigma^*$. If $X_1, \ldots, X_m \in \mathcal{M}$, we usually write

$|^m(X_1, \ldots, X_m)$ as $X_1 \mid \cdots \mid X_m$.

Informally, for each $K \in \mathcal{K}$ we use the value $K^{-1}$ to stand for the property that the inverse function of $E_K$ is accessible. The difficulty of decrypting a message encrypted under $K$ without the knowledge of $K^{-1}$ will be captured by the way we define certain sets of messages in the model later. The collision-free property of keyed hash functions will be captured as part of a restriction we will make on our model later.

We assume that there is a global notion of time which is linear and discrete; for convenience we think of time as ranging over the set of all integers $\mathbb{Z}$. We call our finite collection $\mathcal{P}$ of principals a *system (of principals)*. The actions a principal can perform are defined by the following:

1. *generate*$(m)$: This corresponds to generating a primitive term $m$.
2. *send*$(m)$: This corresponds to sending a message $m$.
3. *receive*$(m)$: This corresponds to receiving a message $m$.
4. *encrypt*$(m,k)$, and *hash*$(m,k)$: These correspond to encrypting, and keyed hashing, respectively, of a message $m$ using key $k$.
5. *concatenate*$(m_1, \ldots, m_k)$: This corresponds to concatenating messages $m_1, \ldots, m_k$.

We assume that at a given time a principal can perform at most one of the above actions. We also include a null action, denoted *null*, assumed to be performed precisely when none of the above actions is performed.

Fix a system: $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ for some positive integer $n$. Intuitively, the notion of a *run of the system* describes an execution of the system over time. We shall characterize a run $r$ of the system by means of the following components: (1) a time $t_{\text{first}}(r)$, called the *start time for* $r$, at which execution is assumed to begin; (2) for each $i$, a sequence $h(P_i, r)$, called the *total history of* $P_i$ *in* $r$, which describes all the actions $P_i$ performs in $r$.

**Definition 4.1** A *run* $r$ *of the system* is a tuple $(t_{\text{first}}(r), h(P_1, r), \ldots, h(P_n, r))$, where:

1. $t_{\text{first}}(r) \in \mathbb{Z}$, and
2. for each $i$, $h(P_i, r)$ is the union of the sequences $h(P_i, r, t_{\text{first}}(r))$, $h(P_i, r, t_{\text{first}}(r)+1)$, ..., which are determined as follows:

$$h(P_i, r, t) = \begin{cases} () & \text{if } t = t_{\text{first}}(r) \\ h(P_i, r, t-1) \cdot a & \text{if } t > t_{\text{first}}(r) \text{ and } a \text{ is the action performed by } P_i \text{ at time } t-1. \end{cases}$$

It is convenient to call the sequence $h(P_i, r, t)$, for each $i$, the *partial history of* $P_i$ *at* $t$ *in* $r$. We emphasize that $h(P_i, r, t)$ includes all actions $P_i$ has performed in $r$ up to, but not including, time $t$.

Let a principal $P$ and a run $r$ of the system be fixed. The key component of the model consists of the definition of several *message sets*. For convenience we define some auxiliary sets first.

**Definition 4.2** Let $a$ denote the action $P$ performs at time $t$ in $r$.

$$\text{(a) } \mathcal{S}_{genr}(P,r,t) = \begin{cases} \{X\} & \text{if } a = generate(X) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\text{(b) } \mathcal{S}_{recv}(P,r,t) = \begin{cases} \{X\} & \text{if } a = receive(X) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\text{(c) } \mathcal{S}_{poss}(P,r,t) = \begin{cases} \{X\} & \text{if } a = generate(X) \text{ or } a = receive(X) \\ \{E_K(X)\} & \text{if } a = encrypt(X,K) \\ \{H_K(X)\} & \text{if } a = hash(X,K) \\ \{(X_1 \mid \cdots \mid X_k)\} & \text{if } a = concatenate(X_1,\ldots,X_k) \\ \emptyset & \text{otherwise} \end{cases}$$

$$\text{(d) } \mathcal{S}_{said}(P,r,t) = \begin{cases} \{X\} & \text{if } a = send(X) \\ \emptyset & \text{otherwise} \end{cases}$$

The following lemma is easily proved from Definition 4.2.

**Lemma 4.1**

*(a)* $\mathcal{S}_{genr}(P,r,t) \subseteq \mathcal{S}_{poss}(P,r,t)$;

*(b)* $\mathcal{S}_{recv}(P,r,t) \subseteq \mathcal{S}_{poss}(P,r,t)$.

In preparation for the lengthy definition that will follow, we begin by discussing informally some of the sets to be defined there. For each time $t$, we will define the following message sets: $\mathcal{M}_{genr}(P,r,t)$, $\mathcal{M}_{recv}(P,r,t)$, $\mathcal{M}_{poss}(P,r,t)$, $\mathcal{M}_{seen}(P,r,t)$, and $\mathcal{M}_{said}(P,r,t)$. Informally, the set $\mathcal{M}_{genr}(P,r,t)$ (respectively, $\mathcal{M}_{recv}(P,r,t)$) consists of all the messages $P$ produces by means of the *generate*() (respectively, *receive*()) action at any time in $r$ up to, but not including, time $t$. In other words, for all $X$: (1) $X \in \mathcal{M}_{genr}(P,r,t)$ iff $generate(X)$ appears in $h(P,r,t)$; (2) $X \in \mathcal{M}_{recv}(P,r,t)$ iff $receive(X)$ appears in $h(P,r,t)$. The sets $\mathcal{M}_{poss}(P,r,t)$, $\mathcal{M}_{seen}(P,r,t)$, and $\mathcal{M}_{said}(P,r,t)$ are meant to model the intuitive notions of *possessed messages*, *seen messages*, and *said*

*messages*, respectively. As discussed in the previous section, in defining these sets we shall use the idea of a message 'occurring in the system'. Certain sets of tuples are useful for capturing this idea: $\mathcal{E}(P,r,t)$, $\mathcal{H}(P,r,t)$, and $\mathcal{C}(P,r,t)$. The set $\mathcal{E}(P,r,t)$ (respectively, $\mathcal{H}(P,r,t)$) consists of 2-tuples corresponding to message-key pairs; the set $\mathcal{C}(P,r,t)$ consists of $m$-tuples of messages for various natural numbers $m > 1$. Informally, we use the set $\mathcal{E}(P,r,t)$ to record all the pairs $(X,K)$ such that $P$ has performed the action *encrypt*$(X,K)$ at any time in $r$ up to, but not including, time $t$; and similarly for the sets $\mathcal{H}(P,r,t)$ and $\mathcal{C}(P,r,t)$, respectively. Notice that the union of the sets $\mathcal{E}(P_i,r,t)$ for all $i$ identifies all encryptions that are constructed by any principal in $r$ at any time earlier than $t$; and similarly for hashes and concatenations. For convenience we introduce the following additional sets to denote the respective unions: $\mathcal{E}(r,t) = \bigcup_{i=1}^{n} \mathcal{E}(P_i,r,t)$, $\mathcal{H}(r,t) = \bigcup_{i=1}^{n} \mathcal{H}(P_i,r,t)$, and $\mathcal{C}(r,t) = \bigcup_{i=1}^{n} \mathcal{C}(P_i,r,t)$.

To formulate the assumption that messages can be constructed in only one way, we distinguish sets of messages occurring in the system according to the type of action which gave rise to them. For example, we will define the set $\mathcal{M}_{genr}(r,t)$ as the set containing all the messages constructed by means of the *generate*() action by any principal at any time in $r$ up to, but not including, time $t$; and similarly the sets $\mathcal{M}_{encr}(r,t)$, $\mathcal{M}_{hash}(r,t)$, and $\mathcal{M}_{conc}(r,t)$ for encrypted, hashed and concatenated messages, respectively. The desired assumption is then stated in two parts: one part which says that the above sets are pairwise disjoint; another part which says that $E_{\_}(\_)$ and $H_{\_}(\_)$ (respectively, $\_\,|\cdots|\,\_$) are one-to-one functions when restricted to those message-key pairs (respectively, message-tuples) which occur in the system.

We will make use of the sets $\mathcal{E}(r,t)$, $\mathcal{H}(r,t)$, and $\mathcal{C}(r,t)$ in defining the closure operation that determines the set $\mathcal{M}_{poss}(P,r,t)$. For example, suppose that a message $X$ and a key $K$ are in this set. Then our definition implies that so is the encrypted message $E_K(X)$, but only if $(X,K) \in \mathcal{E}(r,t)$, i.e. if the encrypted message already occurs in the system at time $t$ in $r$. The sets $\mathcal{M}_{seen}(P,r,t)$, and $\mathcal{M}_{said}(P,r,t)$ will also be defined along similar lines.

The following definition brings together the above discussion and is central to our model. It proceeds in two parts: Each of the above sets is defined to be empty at $t = t_{\text{first}}(r)$. Then assuming all sets are defined at all times up to and including $t-1$, we define them at $t$. In parallel with these definitions, we restrict the actions that can be performed at a given time.

**Definition 4.3**

1. Let $t = t_{\text{first}}(r)$.

   (i) $\mathcal{M}genr(P,r,t) = \mathcal{M}recv(P,r,t) = \emptyset$

   (ii) $\mathcal{E}(P,r,t) = \mathcal{H}(P,r,t) = \mathcal{C}(P,r,t) = \emptyset$

   (iii) $\mathcal{E}(r,t) = \mathcal{H}(r,t) = \mathcal{C}(r,t) = \emptyset$

   (iv) $\mathcal{M}genr(r,t) = \mathcal{M}encr(r,t) = \mathcal{M}hash(r,t) = \mathcal{M}conc(r,t) = \emptyset$

   (v) $\mathcal{M}poss(P,r,t) = \mathcal{M}seen(P,r,t) = \mathcal{M}said(P,r,t) = \emptyset$

   RE0. The only action permitted is the *generate*() action.

2. Let $t > t_{\text{first}}(r)$.

   (i) $\mathcal{M}genr(P,r,t) = \mathcal{M}genr(P,r,t-1) \cup \mathcal{S}genr(P,r,t-1)$

   (ii) $\mathcal{M}recv(P,r,t) = \mathcal{M}recv(P,r,t-1) \cup \mathcal{S}recv(P,r,t-1)$

   (iii) $\mathcal{E}(P,r,t) = \mathcal{E}(P,r,t-1) \cup \mathcal{S}$, where

   $$\mathcal{S} = \begin{cases} \{(X,K)\} & \text{if } P \text{ performs } encrypt(X,K) \text{ at time } t-1 \\ \emptyset & \text{otherwise} \end{cases}$$

   (iv) $\mathcal{H}(P,r,t) = \mathcal{H}(P,r,t-1) \cup \mathcal{S}$, where

   $$\mathcal{S} = \begin{cases} \{(X,K)\} & \text{if } P \text{ performs } hash(X,K) \text{ at time } t-1 \\ \emptyset & \text{otherwise} \end{cases}$$

   (v) $\mathcal{C}(P,r,t) = \mathcal{C}(P,r,t-1) \cup \mathcal{S}$, where

   $$\mathcal{S} = \begin{cases} \{(X_1,\ldots,X_k)\} & \text{if } P \text{ performs } concatenate(X_1,\ldots,X_k) \text{ at time } t-1 \\ \emptyset & \text{otherwise} \end{cases}$$

(vi) $\mathcal{E}(r,t) = \bigcup_{i=1}^{n} \mathcal{E}(P_i,r,t)$

(vii) $\mathcal{H}(r,t) = \bigcup_{i=1}^{n} \mathcal{H}(P_i,r,t)$

(viii) $\mathcal{C}(r,t) = \bigcup_{i=1}^{n} \mathcal{C}(P_i,r,t)$

RE1. (a) If $(X,K),(X',K') \in \mathcal{E}(r,t)$ and $E_K(X) = E_{K'}(X')$, then $X = X'$ and $K = K'$.

(b) If $(X,K),(X',K') \in \mathcal{H}(r,t)$ and $H_K(X) = H_{K'}(X')$, then $X = X'$ and $K = K'$.

(c) If $(X_1,\ldots,X_k),(X'_1,\ldots,X'_{k'}) \in \mathcal{C}(r,t)$ and $X_1 \mid \cdots \mid X_k = X'_1 \mid \cdots \mid X'_{k'}$, then $k = k'$ and $X_1 = X'_1, \ldots, X_k = X'_{k'}$.

(ix) $\mathcal{M}_{genr}(r,t) = \bigcup_{i=1}^{n} \mathcal{M}_{genr}(P_i,r,t)$

(x) $\mathcal{M}_{encr}(r,t) = \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t)\}$

(xi) $\mathcal{M}_{hash}(r,t) = \{H_K(X) \mid (X,K) \in \mathcal{H}(r,t)\}$

(xii) $\mathcal{M}_{conc}(r,t) = \{(X_1 \mid \cdots \mid X_k) \mid (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}$

RE2. The sets $\mathcal{M}_{genr}(r,t)$, $\mathcal{M}_{encr}(r,t)$, $\mathcal{M}_{hash}(r,t)$, and $\mathcal{M}_{conc}(r,t)$ are pairwise disjoint.

(xiii) $\mathcal{M}_{poss}(P,r,t)$ is the smallest set of messages such that:

I. (Basis)

$$\mathcal{M}_{poss}(P,r,t-1) \cup \mathcal{S}_{poss}(P,r,t-1) \subset \mathcal{M}_{poss}(P,r,t)$$

II. (Induction)

| | | |
|---|---|---|
| (a) | $E_K(X) \in \mathcal{M}_{poss}(P,r,t)$ | if $X, K \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ |
| (b) | $H_K(X) \in \mathcal{M}_{poss}(P,r,t)$ | if $X, K \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{H}(r,t)$ |
| (c) | $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t)$ | if $X_1, \ldots, X_k \in \mathcal{M}_{poss}(P,r,t)$ and $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$ |
| (d) | $X \in \mathcal{M}_{poss}(P,r,t)$ | if $E_K(X), K^{-1} \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ |
| (e) | $X_i \in \mathcal{M}_{poss}(P,r,t)$ | if $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t)$ and $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$ |

(xiv) $\mathcal{M}_{seen}(P,r,t)$ is the smallest set of messages such that:

I. (Basis)

$$\mathcal{M}_{seen}(P,r,t-1) \cup \mathcal{S}_{recv}(P,r,t-1) \subset \mathcal{M}_{seen}(P,r,t)$$

II. (Induction)

| | | |
|---|---|---|
| (a) | $X \in \mathcal{M}_{seen}(P,r,t)$ | if $E_K(X) \in \mathcal{M}_{seen}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ and $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$ |
| (b) | $X_i \in \mathcal{M}_{seen}(P,r,t)$ | if $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{seen}(P,r,t)$ and $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$ |

(xv) $\mathcal{M}_{said}(P,r,t)$ is the smallest set of messages such that:

I. (Basis)

$$\mathcal{M}_{said}(P,r,t-1) \cup \mathcal{S}_{said}(P,r,t-1) \subset \mathcal{M}_{said}(P,r,t)$$

II. (Induction)

| | | |
|---|---|---|
| (a) | $X, K \in \mathcal{M}_{said}(P,r,t)$ | if $E_K(X) \in \mathcal{M}_{said}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ and $X, K \in \mathcal{M}_{poss}(P,r,t)$ |
| (b) | $X, K \in \mathcal{M}_{said}(P,r,t)$ | if $H_K(X) \in \mathcal{M}_{said}(P,r,t)$ and $(X,K) \in \mathcal{H}(r,t)$ and $X, K \in \mathcal{M}_{poss}(P,r,t)$ |
| (c) | $X_i \in \mathcal{M}_{said}(P,r,t)$ | if $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{said}(P,r,t)$ and $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$ |

RE3. If $P$ performs $receive(X)$ at $t$, then there exists a principal $Q$ which performs $send(X)$ at some $t' < t$.

RE4. If $P$ performs $send(X)$ at $t$, then $X \in \mathcal{M}_{poss}(P, r, t)$.

RE5. If $P$ performs $encrypt(X, K)$ or $hash(X, K)$ at $t$, then $X, K \in \mathcal{M}_{poss}(P, r, t)$.

RE6. If $P$ performs $concatenate(X_1, \ldots, X_k)$ at $t$, then $X_1, \ldots, X_k \in \mathcal{M}_{poss}(P, r, t)$.

(This completes Definition 4.3.)

The following lemmas are easily proved from Definition 4.3.

**Lemma 4.2** *For all $t, t'$ such that $t < t'$ the following holds:*

*(a)* $\mathcal{M}_{genr}(P, r, t) \subseteq \mathcal{M}_{genr}(P, r, t')$;

*(b)* $\mathcal{M}_{recv}(P, r, t) \subseteq \mathcal{M}_{recv}(P, r, t')$;

*(c)* $\mathcal{E}(P, r, t) \subseteq \mathcal{E}(P, r, t')$;

*(d)* $\mathcal{H}(P, r, t) \subseteq \mathcal{H}(P, r, t')$;

*(e)* $\mathcal{C}(P, r, t) \subseteq \mathcal{C}(P, r, t')$;

*(f)* $\mathcal{M}_{poss}(P, r, t) \subseteq \mathcal{M}_{poss}(P, r, t')$;

*(g)* $\mathcal{M}_{seen}(P, r, t) \subseteq \mathcal{M}_{seen}(P, r, t')$;

*(h)* $\mathcal{M}_{said}(P, r, t) \subseteq \mathcal{M}_{said}(P, r, t')$.

**Lemma 4.3** *For all $t$, $t'$ such that $t < t'$ the following holds:*

*(a)* $\mathcal{E}(r, t) \subseteq \mathcal{E}(r, t')$;

*(b)* $\mathcal{H}(r, t) \subseteq \mathcal{H}(r, t')$;

*(c)* $\mathcal{C}(r, t) \subseteq \mathcal{C}(r, t')$.

**Lemma 4.4** *For each time $t$, the following sets are finite:*

*(a)* $\mathcal{M}_{genr}(P, r, t)$, *and* $\mathcal{M}_{recv}(P, r, t)$;

*(b)* $\mathcal{E}(P,r,t)$, $\mathcal{H}(P,r,t)$, *and* $\mathcal{C}(P,r,t)$.

We will later prove the finiteness property for the sets $\mathcal{M}_{poss}(P,r,t)$, $\mathcal{M}_{seen}(P,r,t)$, and $\mathcal{M}_{said}(P,r,t)$.

**Lemma 4.5** *For each time $t$, the following sets are finite:*

$$\mathcal{E}(r,t), \mathcal{H}(r,t), \text{ and } \mathcal{C}(r,t).$$

**Lemma 4.6** *For all $t$, $t'$ such that $t < t'$ the following holds:*

*(a)* $\mathcal{M}_{genr}(r,t) \subseteq \mathcal{M}_{genr}(r,t')$;

*(b)* $\mathcal{M}_{encr}(r,t) \subseteq \mathcal{M}_{encr}(r,t')$;

*(c)* $\mathcal{M}_{hash}(r,t) \subseteq \mathcal{M}_{hash}(r,t')$;

*(d)* $\mathcal{M}_{conc}(r,t) \subseteq \mathcal{M}_{conc}(r,t')$.

**Lemma 4.7** *For each time $t$, the following sets are finite:*

$$\mathcal{M}_{genr}(r,t), \mathcal{M}_{encr}(r,t), \mathcal{M}_{hash}(r,t), \text{ and } \mathcal{M}_{conc}(r,t).$$

**Lemma 4.8**

*(a) If* $(X,K) \in \mathcal{E}(P,r,t)$ *then* $X, K \in \mathcal{M}_{poss}(P,r,t-1)$.

*(b) If* $(X,K) \in \mathcal{H}(P,r,t)$ *then* $X, K \in \mathcal{M}_{poss}(P,r,t-1)$.

*(c) If* $(X_1,\ldots,X_k) \in \mathcal{C}(P,r,t)$ *then* $X_1,\ldots,X_k \in \mathcal{M}_{poss}(P,r,t-1)$.

**Proof.** (By induction on $t$.) We only prove part (a); the remaining parts are proved similarly.

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3, $\mathcal{E}(P,r,t_{\text{first}}(r)) = \emptyset$. Therefore, the required statement holds vacuously.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: if $(X,K) \in \mathcal{E}(P,r,t)$ then $X, K \in \mathcal{M}_{poss}(P,r,t-1)$; and we show this implies that, if $(X,K) \in \mathcal{E}(P,r,t+1)$ then $X, K \in \mathcal{M}_{poss}(P,r,t)$.

   Suppose $(X,K) \in \mathcal{E}(P,r,t+1)$. By definition 4.3 we need to consider the following two cases:

Case (A): $(X, K) \in \mathcal{E}(P, r, t)$. The inductive hypothesis yields $X, K \in \mathcal{M}_{poss}(P, r, t-1)$. By Lemma 4.2(f) it follows that $X, K \in \mathcal{M}_{poss}(P, r, t)$.

Case (B): $P$ performs $encrypt(X, K)$ at $t$. RE5 yields $X, K \in \mathcal{M}_{poss}(P, r, t)$.

(This completes the proof of Lemma 4.8.) □

In the sequel, we shall make use of a proof technique which is vital to proving properties of the following inductively defined sets: $\mathcal{M}_{poss}(P, r, t)$, $\mathcal{M}_{seen}(P, r, t)$, and $\mathcal{M}_{said}(P, r, t)$. It suffices to explain this technique in context of the set $\mathcal{M}_{poss}(P, r, t)$, since it works similarly in other contexts. Essentially, the technique works as follows. We construct a sequence of sets $\mathcal{M}^i_{poss}(P, r, t)$ for $i = 0, 1, 2, \ldots$, with the following property: (1) the first set in the sequence is the basis set for $\mathcal{M}_{poss}(P, r, t)$, and (2) each of the remaining sets in the sequence is the union of the set which immediately precedes it and the set obtained from the preceding set by applying the closure operation exactly once. The point of the above construction is now obvious: for proving that a particular statement holds for $\mathcal{M}_{poss}(P, r, t)$, we use induction on $i$ to show that it holds for all sets $\mathcal{M}^i_{poss}(P, r, t)$.

**Definition 4.4** Let $i \geq 0$.

1. Let $t = t_{\text{first}}(r)$. Then $\mathcal{M}^i_{poss}(P, r, t) = \emptyset$ for all $i$.

2. Let $t > t_{\text{first}}(r)$. Then

$$\mathcal{M}^i_{poss}(P, r, t) = \begin{cases} \mathcal{M}_{poss}(P, r, t-1) \cup \mathcal{S}_{poss}(P, r, t-1) & \text{if } i = 0 \\ \mathcal{M}^{i-1}_{poss}(P, r, t) \cup \mathcal{S} & \text{if } i > 0 \end{cases}$$

where

$$\begin{aligned} \mathcal{S} \;=\; & \{E_K(X) \mid X, K \in \mathcal{M}^{i-1}_{poss}(P, r, t) \text{ and } (X, K) \in \mathcal{E}(r, t)\} \\ & \cup \{H_K(X) \mid X, K \in \mathcal{M}^{i-1}_{poss}(P, r, t) \text{ and } (X, K) \in \mathcal{H}(r, t)\} \\ & \cup \{(X_1 \mid \cdots \mid X_k) \mid X_1, \ldots, X_k \in \mathcal{M}^{i-1}_{poss}(P, r, t) \text{ and } (X_1, \ldots, X_k) \in \mathcal{C}(r, t)\} \\ & \cup \{X \mid E_K(X), K^{-1} \in \mathcal{M}^{i-1}_{poss}(P, r, t) \text{ and } (X, K) \in \mathcal{E}(r, t)\} \\ & \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^{i-1}_{poss}(P, r, t) \text{ and } (X_1, \ldots, X_k) \in \mathcal{C}(r, t)\}. \end{aligned}$$

The following lemma is easily proved from Definition 4.3 and Definition 4.4.

**Lemma 4.9**

$$\mathcal{M}^0_{poss}(P,r,t) \subseteq \mathcal{M}^1_{poss}(P,r,t) \subseteq \cdots \subseteq \bigcup_{i=0}^{\infty} \mathcal{M}^i_{poss}(P,r,t) = \mathcal{M}_{poss}(P,r,t)$$

**Lemma 4.10**

$$\mathcal{M}_{genr}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3,

   $$\mathcal{M}_{genr}(P,r,t_{\text{first}}(r)) = \mathcal{M}_{poss}(P,r,t_{\text{first}}(r)) = \emptyset.$$

   Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: $\mathcal{M}_{genr}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$; and we show this implies $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

   From Lemma 4.1(a) and the inductive hypothesis it follows that $\mathcal{M}_{genr}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t) \cup \mathcal{S}_{poss}(P,r,t)$. By definition 4.3, $\mathcal{M}_{genr}(P,r,t+1) = \mathcal{M}_{genr}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t)$, and, by definition 4.4, $\mathcal{M}^0_{poss}(P,r,t+1) = \mathcal{M}_{poss}(P,r,t) \cup \mathcal{S}_{poss}(P,r,t)$. Hence $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}^0_{poss}(P,r,t+1)$. By Lemma 4.9 it follows that $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

(This completes the proof of Lemma 4.10.) □

**Lemma 4.11**

$$\mathcal{M}_{poss}(P,r,t) \subseteq \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3, $\mathcal{M}_{poss}(P,r,t_{\text{first}}(r)) = \emptyset$ and $\mathcal{M}_{genr}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{encr}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{hash}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{conc}(r,t_{\text{first}}(r)) = \emptyset$. Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) for all $t' < t$, $\mathcal{M}_{poss}(P,r,t') \subseteq \mathcal{M}_{genr}(r,t') \cup \mathcal{M}_{encr}(r,t') \cup \mathcal{M}_{hash}(r,t') \cup \mathcal{M}_{conc}(r,t')$; and we show this implies $\mathcal{M}_{poss}(P,r,t) \subseteq \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

   By Lemma 4.9 it suffices to show that, for all $Y$ and for all $m$, if $Y \in \mathcal{M}^m_{poss}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$. This assertion is shown using induction on $m$:

I. (Basis) Let $m = 0$. Suppose $Y \in \mathcal{M}^0_{poss}(P,r,t)$. By definition 4.4, $Y \in \mathcal{M}_{poss}(P,r,t-1) \cup \mathcal{S}_{poss}(P,r,t-1)$.

Case (i): $Y \in \mathcal{M}_{poss}(P,r,t-1)$. HP1 yields $Y \in \mathcal{M}_{genr}(r,t-1) \cup \mathcal{M}_{encr}(r,t-1) \cup \mathcal{M}_{hash}(r,t-1) \cup \mathcal{M}_{conc}(r,t-1)$. By Lemma 4.6 it follows that $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (ii): $Y \in \mathcal{S}_{poss}(P,r,t-1)$.

Case (A): $P$ performs $generate(Y)$ at $t-1$.
By definition 4.3, $Y \in \mathcal{M}_{genr}(r,t)$.

Case (B): $P$ performs $receive(Y)$ at $t-1$.
RE3 yields: there exists a $Q$ which performs $send(Y)$ at some time $t' < t-1$. RE4 yields $Y \in \mathcal{M}_{poss}(Q,r,t')$. HP1 yields $Y \in \mathcal{M}_{genr}(r,t') \cup \mathcal{M}_{encr}(r,t') \cup \mathcal{M}_{hash}(r,t') \cup \mathcal{M}_{conc}(r,t')$. By Lemma 4.6 it follows that $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (C): $P$ performs $encrypt(X,K)$ at $t-1$ for some $X$ and some $K$, where $Y = E_K(X)$.
By definition 4.3, $E_K(X) \in \mathcal{M}_{encr}(r,t)$. Hence $Y \in \mathcal{M}_{encr}(r,t)$.

Case (D): $P$ performs $hash(X,K)$ at $t-1$ for some $X$ and some $K$, where $Y = H_K(X)$.
Similar to Case (C).

Case (E): $P$ performs $concatenate(X_1,\ldots,X_k)$ at $t-1$ for some $X_1,\ldots,X_k$, where $Y = X_1 \mid \cdots \mid X_k$.
Similar to Case (C).

II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis: (HP2) for all $Y$, if $Y \in \mathcal{M}^m_{poss}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$; and we show this implies that, for all $Y$, if $Y \in \mathcal{M}^{m+1}_{poss}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Suppose $Y \in \mathcal{M}^{m+1}_{poss}(P,r,t)$. By definition 4.4,

$$
\begin{aligned}
Y \in\ & \mathcal{M}^m_{poss}(P,r,t) \\
& \cup \{E_K(X) \mid X,K \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\} \\
& \cup \{H_K(X) \mid X,K \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{H}(r,t)\} \\
& \cup \{(X_1 \mid \cdots \mid X_k) \mid X_1,\ldots,X_k \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}
\end{aligned}
$$

$$\cup \{X \mid E_K(X), K^{-1} \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\}$$
$$\cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}.$$

Case (A): $Y \in \mathcal{M}^m_{poss}(P,r,t)$. HP2 yields $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (B): $Y \in \{E_K(X) \mid X, K \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\}$.
We have $Y = E_K(X)$ for some $X$ and some $K$ such that $(X,K) \in \mathcal{E}(r,t)$. By definition 4.3, $E_K(X) \in \mathcal{M}_{encr}(r,t)$. Hence $Y \in \mathcal{M}_{encr}(r,t)$.

Case (C): $Y \in \{H_K(X) \mid X, K \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{H}(r,t)\}$.
Similar to Case (B).

Case (D): $Y \in \{(X_1 \mid \cdots \mid X_k) \mid X_1,\ldots,X_k \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}$.
Similar to Case (B).

Case (E): $Y \in \{X \mid E_K(X), K^{-1} \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\}$.
We have $(Y,K) \in \mathcal{E}(r,t)$ for some $K$. By definition 4.3, $(Y,K) \in \mathcal{E}(Q,r,t)$ for some $Q$. By Lemma 4.8(a), $Y, K \in \mathcal{M}_{poss}(Q,r,t-1)$. HP1 yields $Y \in \mathcal{M}_{genr}(r,t-1) \cup \mathcal{M}_{encr}(r,t-1) \cup \mathcal{M}_{hash}(r,t-1) \cup \mathcal{M}_{conc}(r,t-1)$. By Lemma 4.6 it follows that $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (F): $Y \in \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}$.
Similar to Case (E).

(This completes the proof of Lemma 4.11.)

□

**Lemma 4.12**

*(a) Let $(X,K) \in \mathcal{E}(r,t)$. If $E_K(X) \in \mathcal{M}_{poss}(P,r,t')$ for some $P$ and for some $t' < t$, then $(X,K) \in \mathcal{E}(r,t')$.*

*(b) Let $(X,K) \in \mathcal{H}(r,t)$. If $H_K(X) \in \mathcal{M}_{poss}(P,r,t')$ for some $P$ and for some $t' < t$, then $(X,K) \in \mathcal{H}(r,t')$.*

*(c) Let $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$. If $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t')$ for some $P$ and for some $t' < t$, then $(X_1,\ldots,X_k) \in \mathcal{C}(r,t')$.*

**Proof.** We only prove part (a); the remaining parts are proved similarly. We have $(X,K) \in \mathcal{E}(r,t)$. Suppose $E_K(X) \in \mathcal{M}_{poss}(P,r,t')$ for some $P$ and for some $t' < t$. By

definition 4.3, $E_K(X) \in \mathcal{M}_{encr}(r,t)$. RE2 yields $E_K(X) \notin \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$. By Lemma 4.6 it follows that $E_K(X) \notin \mathcal{M}_{genr}(r,t') \cup \mathcal{M}_{hash}(r,t') \cup \mathcal{M}_{conc}(r,t')$. Since $E_K(X) \in \mathcal{M}_{poss}(P,r,t')$, it follows by Lemma 4.11 that $E_K(X) \in \mathcal{M}_{genr}(r,t') \cup \mathcal{M}_{encr}(r,t') \cup \mathcal{M}_{hash}(r,t') \cup \mathcal{M}_{conc}(r,t')$. Hence $E_K(X) \in \mathcal{M}_{encr}(r,t')$. By definition 4.3, $E_K(X) = E_{K'}(X')$ for some $(X',K') \in \mathcal{E}(r,t')$. We have $t > t'$. By Lemma 4.6 it follows that $(X',K') \in \mathcal{E}(r,t)$. RE1 yields $X = X'$ and $K = K'$. Therefore, $(X,K) \in \mathcal{E}(r,t')$. □

**Lemma 4.13**

*(a) If $X, K \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$, then $E_K(X) \in \mathcal{M}_{poss}(P,r,t)$.*

*(b) If $X, K \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{H}(r,t)$, then $H_K(X) \in \mathcal{M}_{poss}(P,r,t)$.*

*(c) If $X_1, \ldots, X_k \in \mathcal{M}_{poss}(P,r,t)$ and $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$, then $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t)$.*

*(d) If $E_K(X), K^{-1} \in \mathcal{M}_{poss}(P,r,t)$ for some $(X,K) \in \mathcal{E}(r,t)$, then $X \in \mathcal{M}_{poss}(P,r,t)$.*

*(e) If $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t)$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$, then $X_1, \ldots, X_k \in \mathcal{M}_{poss}(P,r,t)$.*

**Proof.** We only prove part (a); the remaining parts are proved similarly. Suppose $X, K \in \mathcal{M}_{poss}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$. By Lemma 4.9 it suffices to show that $E_K(X) \in \mathcal{M}^l_{poss}(P,r,t)$ for some $l$. Since $X, K \in \mathcal{M}_{poss}(P,r,t)$, it follows by Lemma 4.9 that $X, K \in \mathcal{M}^m_{poss}(P,r,t)$ for some $m$. By definition 4.4, $\mathcal{M}^{m+1}_{poss}(P,r,t) \supseteq \{E_K(X) \mid X, K \in \mathcal{M}^m_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\}$. Hence $E_K(X) \in \mathcal{M}^{m+1}_{poss}(P,r,t)$. □

**Definition 4.5** Let $i \geq 0$.

1. Let $t = t_{\text{first}}(r)$. Then $\mathcal{M}^i_{seen}(P,r,t) = \emptyset$ for all $i$.

2. Let $t > t_{\text{first}}(r)$. Then

$$\mathcal{M}^i_{seen}(P,r,t) = \begin{cases} \mathcal{M}_{seen}(P,r,t-1) \cup \mathcal{S}_{recv}(P,r,t-1) & \text{if } i = 0 \\ \mathcal{M}^{i-1}_{seen}(P,r,t) \cup \mathcal{S} & \text{if } i > 0 \end{cases}$$

where

$$\mathcal{S} = \{X \mid E_K(X) \in \mathcal{M}^{i-1}_{seen}(P,r,t) \text{ and } K^{-1} \in \mathcal{M}_{poss}(P,r,t) \text{ and } (X,K) \in \mathcal{E}(r,t)\} \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^{i-1}_{seen}(P,r,t) \text{ and } (X_1, \ldots, X_k) \in \mathcal{C}(r,t)\}.$$

The following lemma is easily proved from Definition 4.3 and Definition 4.5.

**Lemma 4.14**

$$\mathcal{M}^0_{seen}(P,r,t) \subseteq \mathcal{M}^1_{seen}(P,r,t) \subseteq \cdots \subseteq \bigcup_{i=0}^{\infty} \mathcal{M}^i_{seen}(P,r,t) = \mathcal{M}_{seen}(P,r,t)$$

**Lemma 4.15**

$$\mathcal{M}_{recv}(P,r,t) \subseteq \mathcal{M}_{seen}(P,r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3,

   $$\mathcal{M}_{recv}(P,r,t_{\text{first}}(r)) = \mathcal{M}_{seen}(P,r,t_{\text{first}}(r)) = \emptyset.$$

   Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: $\mathcal{M}_{recv}(P,r,t) \subseteq \mathcal{M}_{seen}(P,r,t)$; and we show this implies $\mathcal{M}_{recv}(P,r,t+1) \subseteq \mathcal{M}_{seen}(P,r,t+1)$.

   By the inductive hypothesis it follows that

   $$\mathcal{M}_{recv}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t) \subseteq \mathcal{M}_{seen}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t).$$

   By definition 4.3, $\mathcal{M}_{recv}(P,r,t+1) = \mathcal{M}_{recv}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t)$, and, by definition 4.5, $\mathcal{M}^0_{seen}(P,r,t+1) = \mathcal{M}_{seen}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t)$. Hence $\mathcal{M}_{recv}(P,r,t+1) \subseteq \mathcal{M}^0_{seen}(P,r,t+1)$. By Lemma 4.14 it follows that $\mathcal{M}_{recv}(P,r,t+1) \subseteq \mathcal{M}_{seen}(P,r,t+1)$.

(This completes the proof of Lemma 4.15.) □

**Lemma 4.16**

$$\mathcal{M}_{seen}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3,

   $$\mathcal{M}_{seen}(P,r,t_{\text{first}}(r)) = \mathcal{M}_{poss}(P,r,t_{\text{first}}(r)) = \emptyset.$$

   Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) $\mathcal{M}_{seen}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$; and we show this implies $\mathcal{M}_{seen}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

   By Lemma 4.14 it suffices to show that, for all $m$, $\mathcal{M}^m_{seen}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$. This assertion is shown using induction on $m$:

   I. (Basis) Let $m = 0$. From Lemma 4.1(b) and HP1 it follows that $\mathcal{M}_{seen}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t) \cup \mathcal{S}_{poss}(P,r,t)$. By definition 4.5, $\mathcal{M}^0_{seen}(P,r,t+1) = \mathcal{M}_{seen}(P,r,t) \cup \mathcal{S}_{recv}(P,r,t)$, and, by definition 4.4, $\mathcal{M}^0_{poss}(P,r,t+1) = \mathcal{M}_{poss}(P,r,t) \cup \mathcal{S}_{poss}(P,r,t)$. Hence $\mathcal{M}^0_{seen}(P,r,t+1) \subseteq \mathcal{M}^0_{poss}(P,r,t+1)$. By Lemma 4.9 it follows that $\mathcal{M}^0_{seen}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

   II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis: (HP2) $\mathcal{M}^m_{seen}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$; and we show this implies $\mathcal{M}^{m+1}_{seen}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

   It suffices to show that, for all $Y$, if $Y \in \mathcal{M}^{m+1}_{seen}(P,r,t+1)$ then $Y \in \mathcal{M}_{poss}(P,r,t+1)$. Suppose $Y \in \mathcal{M}^{m+1}_{seen}(P,r,t+1)$. By definition 4.5,

$$
\begin{aligned}
Y \in\ & \mathcal{M}^m_{seen}(P,r,t+1) \\
& \cup \{X \mid E_K(X) \in \mathcal{M}^m_{seen}(P,r,t+1) \text{ and } (X,K) \in \mathcal{E}(r,t+1) \text{ and} \\
& \quad K^{-1} \in \mathcal{M}_{poss}(P,r,t+1)\} \\
& \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{seen}(P,r,t+1) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)\}.
\end{aligned}
$$

   Case (i): $Y \in \mathcal{M}^m_{seen}(P,r,t+1)$. HP2 yields $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

   Case (ii): $Y \in \{X \mid E_K(X) \in \mathcal{M}^m_{seen}(P,r,t+1) \text{ and } (X,K) \in \mathcal{E}(r,t+1) \text{ and } K^{-1} \in \mathcal{M}_{poss}(P,r,t+1)\}$.
   We have, for some $K$, $E_K(Y) \in \mathcal{M}^m_{seen}(P,r,t+1)$, $(Y,K) \in \mathcal{E}(r,t+1)$, and $K^{-1} \in \mathcal{M}_{poss}(P,r,t+1)$. HP2 yields $E_K(Y) \in \mathcal{M}_{poss}(P,r,t+1)$. By Lemma 4.13(d) it follows that $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

   Case (iii): $Y \in \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{seen}(P,r,t+1) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)\}$.
   We have $Y = X_i$ for some $i$ and for some $X_1 \mid \cdots \mid X_k \in \mathcal{M}^m_{seen}(P,r,t+1)$ such that $(X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)$. HP2 yields $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(P,r,t+1)$. By Lemma 4.13(e) it follows that $X_1,\ldots,X_k \in \mathcal{M}_{poss}(P,r,t+1)$. But $Y = X_i$ for some $i$, so $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

(This completes the proof of Lemma 4.16.) □

**Lemma 4.17** *Let* $Y \in \mathcal{M}_{seen}(P,r,t)$. *Then*

*(a) if* $Y = E_K(X)$ *for some* $(X,K) \in \mathcal{E}(r,t)$ *and* $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$, *then* $X \in \mathcal{M}_{seen}(P,r,t)$, *and*

*(b) if* $Y = X_1 \mid \cdots \mid X_k$ *for some* $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, *then*

$$X_1,\ldots,X_k \in \mathcal{M}_{seen}(P,r,t).$$

**Proof.** We only prove part (a); part (b) is proved similarly. Since $Y \in \mathcal{M}_{seen}(P,r,t)$, it follows by Lemma 4.14 that $Y \in \mathcal{M}^m_{seen}(P,r,t)$ for some $m$. Suppose $Y = E_K(X)$ for some $(X,K) \in \mathcal{E}(r,t)$, and further suppose $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$. By definition 4.5, $\mathcal{M}^{m+1}_{seen}(P,r,t) \supseteq \{X \mid E_K(X) \in \mathcal{M}^m_{seen}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ and $K^{-1} \in \mathcal{M}_{poss}(P,r,t)\}$. Hence $X \in \mathcal{M}^{m+1}_{seen}(P,r,t)$. By Lemma 4.14 it follows that $X \in \mathcal{M}_{seen}(P,r,t)$.

□

**Definition 4.6** Let $i \geq 0$.

1. Let $t = t_{\text{first}}(r)$. Then $\mathcal{M}^i_{said}(P,r,t) = \emptyset$ for all $i$.

2. Let $t > t_{\text{first}}(r)$. Then

$$\mathcal{M}^i_{said}(P,r,t) = \begin{cases} \mathcal{M}_{said}(P,r,t-1) \cup \mathcal{S}_{said}(P,r,t-1) & \text{if } i = 0 \\ \mathcal{M}^{i-1}_{said}(P,r,t) \cup \mathcal{S} & \text{if } i > 0 \end{cases}$$

where

$$\begin{aligned} \mathcal{S} = \; & \{X,K \mid E_K(X) \in \mathcal{M}^{i-1}_{said}(P,r,t) \text{ and } X,K \in \mathcal{M}_{poss}(P,r,t) \text{ and} \\ & (X,K) \in \mathcal{E}(r,t)\} \\ & \cup \{X,K \mid H_K(X) \in \mathcal{M}^{i-1}_{said}(P,r,t) \text{ and } X,K \in \mathcal{M}_{poss}(P,r,t) \text{ and} \\ & (X,K) \in \mathcal{H}(r,t)\} \\ & \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^{i-1}_{said}(P,r,t) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t)\}. \end{aligned}$$

The following lemma is easily proved from Definition 4.3 and Definition 4.6.

**Lemma 4.18**

$$\mathcal{M}^0_{said}(P,r,t) \subseteq \mathcal{M}^1_{said}(P,r,t) \subseteq \cdots \subseteq \bigcup_{i=0}^{\infty} \mathcal{M}^i_{said}(P,r,t) = \mathcal{M}_{said}(P,r,t)$$

**Lemma 4.19**

$$\mathcal{M}_{said}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3,

   $$\mathcal{M}_{said}(P,r,t_{\text{first}}(r)) = \mathcal{M}_{poss}(P,r,t_{\text{first}}(r)) = \emptyset.$$

   Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) $\mathcal{M}_{said}(P,r,t) \subseteq \mathcal{M}_{poss}(P,r,t)$; and we show this implies $\mathcal{M}_{said}(P,r,t+1) \subseteq \mathcal{M}_{poss}(P,r,t+1)$.

   By Lemma 4.18, it suffices to show that, for all $Y$ and for all $m$, if $Y \in \mathcal{M}^m_{said}(P,r,t+1)$ then $Y \in \mathcal{M}_{poss}(P,r,t+1)$. This assertion is shown using induction on $m$:

   I. (Basis) Let $m = 0$. Suppose $Y \in \mathcal{M}^0_{said}(P,r,t+1)$. By definition 4.6, $Y \in \mathcal{M}_{said}(P,r,t) \cup \mathcal{S}_{said}(P,r,t)$.

   Case (A): $Y \in \mathcal{M}_{said}(P,r,t)$. HP1 yields $Y \in \mathcal{M}_{poss}(P,r,t)$. By Lemma 4.2(f) it follows that $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

   Case (B): $Y \in \mathcal{S}_{said}(P,r,t)$. By definition 4.2, $P$ performs $send(Y)$ at $t$. RE4 yields $Y \in \mathcal{M}_{poss}(P,r,t)$. By Lemma 4.2(f), $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

   II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis: (HP2) for all $Y$, if $Y \in \mathcal{M}^m_{said}(P,r,t+1)$ then $Y \in \mathcal{M}_{poss}(P,r,t+1)$; and we show this implies that, for all $Y$, if $Y \in \mathcal{M}^{m+1}_{said}(P,r,t+1)$ then $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

   Suppose $Y \in \mathcal{M}^{m+1}_{said}(P,r,t+1)$. By definition 4.6,

   $$\begin{aligned} Y \in\ & \mathcal{M}^m_{said}(P,r,t+1) \\ & \cup \{X,K \mid E_K(X) \in \mathcal{M}^m_{said}(P,r,t+1) \text{ and } X,K \in \mathcal{M}_{poss}(P,r,t+1) \text{ and} \\ & \quad (X,K) \in \mathcal{E}(r,t+1)\} \\ & \cup \{X,K \mid H_K(X) \in \mathcal{M}^m_{said}(P,r,t+1) \text{ and } X,K \in \mathcal{M}_{poss}(P,r,t+1) \text{ and} \\ & \quad (X,K) \in \mathcal{H}(r,t+1)\} \\ & \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{said}(P,r,t+1) \text{ and } (X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)\}. \end{aligned}$$

Case (i): $Y \in \mathcal{M}^m_{said}(P,r,t+1)$. HP2 yields $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

Case (ii): $Y \in \{X, K \mid E_K(X) \in \mathcal{M}^m_{said}(P,r,t+1)$ and $X, K \in \mathcal{M}_{poss}(P,r,t+1)$ and $(X,K) \in \mathcal{E}(r,t+1)\}$.
We have $Y = X$ or $Y = K$ for some $X, K$ such that $X, K \in \mathcal{M}_{poss}(P,r,t+1)$. Therefore, $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

Case (iii): $Y \in \{X, K \mid H_K(X) \in \mathcal{M}^m_{said}(P,r,t+1)$ and $X, K \in \mathcal{M}_{poss}(P,r,t+1)$ and $(X,K) \in \mathcal{H}(r,t+1)\}$.
Similar to Case (ii).

Case (iv): $Y \in \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{said}(P,r,t+1)$ and $(X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)\}$.
We have $Y = X_i$ for some $i$ and for some $X_1 \mid \cdots \mid X_k \in \mathcal{M}^m_{said}(P,r,t+1)$ such that $(X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)$. HP2 yields $(X_1,\ldots,X_k) \in \mathcal{M}_{poss}(P,r,t+1)$. By Lemma 4.13(e) it follows that $X_1,\ldots,X_k \in \mathcal{M}_{poss}(P,r,t+1)$. But $Y = X_i$ for some $i$, so $Y \in \mathcal{M}_{poss}(P,r,t+1)$.

(This completes the proof of Lemma 4.19.) □

**Lemma 4.20**

*(a) If $E_K(X) \in \mathcal{M}_{said}(P,r,t)$ for some $(X,K) \in \mathcal{E}(r,t)$ such that $X, K \in \mathcal{M}_{poss}(P,r,t)$, then $X, K \in \mathcal{M}_{said}(P,r,t)$.*

*(b) If $H_K(X) \in \mathcal{M}_{said}(P,r,t)$ for some $(X,K) \in \mathcal{H}(r,t)$ such that $X, K \in \mathcal{M}_{poss}(P,r,t)$, then $X, K \in \mathcal{M}_{said}(P,r,t)$.*

*(c) If $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{said}(P,r,t)$ for some $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, then $X_1,\ldots,X_k \in \mathcal{M}_{said}(P,r,t)$.*

**Proof.** We only prove part (a); the remaining parts are proved similarly. Suppose $E_K(X) \in \mathcal{M}_{said}(P,r,t)$ for some $(X,K) \in \mathcal{E}(r,t)$ such that $X, K \in \mathcal{M}_{poss}(P,r,t)$. By Lemma 4.18 it suffices to show that $X, K \in \mathcal{M}^l_{said}(P,r,t)$ for some $l$. Since $E_K(X) \in \mathcal{M}_{said}(P,r,t)$ it follows by Lemma 4.18 that $E_K(X) \in \mathcal{M}^m_{said}(P,r,t)$ for some $m$. By definition 4.6, $\mathcal{M}^{m+1}_{said}(P,r,t) \supseteq \{X, K \mid E_K(X) \in \mathcal{M}^m_{said}(P,r,t)$ and $(X,K) \in \mathcal{E}(r,t)$ and $X, K \in \mathcal{M}_{poss}(P,r,t)\}$. Hence $X, K \in \mathcal{M}^{m+1}_{said}(P,r,t)$.

(This completes the proof of Lemma 4.20.) □

**Proposition 4.1** *For each time $t$, the following sets are finite:*

(a) $\mathcal{M}_{poss}(P,r,t)$;

(b) $\mathcal{M}_{seen}(P,r,t)$;

(c) $\mathcal{M}_{said}(P,r,t)$.

**Proof.**

(a) Follows from Lemma 4.11 and Lemma 4.7.

(b) Follows from Lemma 4.16 and part (a).

(c) Follows from Lemma 4.19 and part (a).

□

The following corollary to Proposition 4.1 is easily proved.

**Corollary 4.1** *For each time $t$, the following holds:*

(a) $\mathcal{M}_{poss}(P,r,t) = \mathcal{M}^k_{poss}(P,r,t)$ *for some* $k$;

(b) $\mathcal{M}_{seen}(P,r,t) = \mathcal{M}^k_{seen}(P,r,t)$ *for some* $k$;

(c) $\mathcal{M}_{said}(P,r,t) = \mathcal{M}^k_{said}(P,r,t)$ *for some* $k$;

**Proposition 4.2**
*Let $Y \in \mathcal{M}^m_{poss}(P,r,t)$ for some $m$, and suppose that $Y \notin \mathcal{M}_{seen}(P,r,t)$.*

(a) *If $Y = E_K(X)$ for some $(X,K) \in \mathcal{E}(r,t)$, then $X, K \in \mathcal{M}^m_{poss}(P,r,t)$.*

(b) *If $Y = H_K(X)$ for some $(X,K) \in \mathcal{H}(r,t)$, then $X, K \in \mathcal{M}^m_{poss}(P,r,t)$.*

(c) *If $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, then*

$$X_1,\ldots,X_k \in \mathcal{M}^m_{poss}(P,r,t).$$

**Proof.** We prove parts (a), (b) and (c) simultaneously by induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.4, $\mathcal{M}^i_{poss}(P,r,t) = \emptyset$ for all $i$. Therefore, the required statement holds vacuously.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) for all $m$, if $Y \in \mathcal{M}^m_{poss}(P,r,t)$ and $Y \notin \mathcal{M}_{seen}(P,r,t)$ then

(a) if $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r,t)$ then $X, K \in \mathcal{M}^m_{poss}(P,r,t)$, and

(b) if $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r,t)$ then $X, K \in \mathcal{M}^m_{poss}(P,r,t)$, and

(c) if $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r,t)$ then $X_1, \ldots, X_k \in \mathcal{M}^m_{poss}(P,r,t)$.

We show the above hypothesis implies that, for all $m$, if $Y \in \mathcal{M}^m_{poss}(P,r,t+1)$ and $Y \notin \mathcal{M}_{seen}(P,r,t+1)$ then

(a) if $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r,t+1)$ then $X, K \in \mathcal{M}^m_{poss}(P,r,t+1)$, and

(b) if $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r,t+1)$ then $X, K \in \mathcal{M}^m_{poss}(P,r,t+1)$, and

(c) if $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r,t+1)$ then $X_1, \ldots, X_k \in \mathcal{M}^m_{poss}(P,r,t+1)$.

The above assertion is shown using induction on $m$:

I. (Basis) Let $m = 0$. Suppose $Y \in \mathcal{M}^0_{poss}(P,r,t+1)$ and $Y \notin \mathcal{M}_{seen}(P,r,t+1)$. By definition 4.4, $Y \in \mathcal{M}_{poss}(P,r,t) \cup \mathcal{S}_{poss}(P,r,t)$.

Case (i): $Y \in \mathcal{M}_{poss}(P,r,t)$. By Lemma 4.9 it follows that $Y \in \mathcal{M}^{m'}_{poss}(P,r,t)$ for some $m'$.

Case (a): $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r,t+1)$.
By Lemma 4.12(a) it follows that $(X, K) \in \mathcal{E}(r,t)$. Since $Y \notin \mathcal{M}_{seen}(P,r,t+1)$, it follows by Lemma 4.2(g) that $Y \notin \mathcal{M}_{seen}(P,r,t)$. HP1 yields $X, K \in \mathcal{M}^{m'}_{poss}(P,r,t)$. By Lemma 4.9 it follows that $X, K \in \mathcal{M}_{poss}(P,r,t)$. Hence, by definition 4.4, $X, K \in \mathcal{M}^0_{poss}(P,r,t+1)$.

Case (b): $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r,t+1)$.
Similar to Case (a).

Case (c): $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r,t+1)$.
Similar to Case (a).

Case (ii): $Y \in \mathcal{S}_{poss}(P,r,t)$.

Case (a): $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r,t+1)$.
By definition 4.3, $E_K(X) \in \mathcal{M}_{encr}(r,t+1)$. Hence $Y \in \mathcal{M}_{encr}(r,t+1)$.

Case (A): $P$ performs *generate*$(Y)$ at $t$.
By definition 4.3, $Y \in \mathcal{M}_{genr}(r, t+1)$, which is impossible by RE2.

Case (B): $P$ performs *receive*$(Y)$ at $t$.
By definition 4.3, $Y \in \mathcal{M}_{seen}(P, r, t+1)$, which is impossible by assumption.

Case (C): $P$ performs *encrypt*$(X', K')$ at $t$ for some $X'$ and some $K'$, where $Y = E_{K'}(X')$.
RE5 yields $X', K' \in \mathcal{M}_{poss}(P, r, t)$. By definition 4.3, $(X', K') \in \mathcal{E}(r, t+1)$. RE1 yields $X = X'$ and $K = K'$. Hence $X, K \in \mathcal{M}_{poss}(P, r, t)$. Therefore, by definition 4.4, $X, K \in \mathcal{M}^0_{poss}(P, r, t+1)$.

Case (D): $P$ performs *hash*$(X', K')$ at $t$ for some $X'$ and some $K'$, where $Y = H_{K'}(X')$.
By definition 4.3, $Y \in \mathcal{M}_{hash}(r, t+1)$, which is impossible by RE2.

Case (E): $P$ performs *concatenate*$(X'_1, \ldots, X'_k)$ at $t$ for some $X'_1, \ldots, X'_k$, where $Y = X'_1 \mid \cdots \mid X'_k$.
Similar to Case (D).

Case (b): $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r, t+1)$.
Similar to Case (a).

Case (c): $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)$.
Similar to Case (a).

II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis:
(HP2) for all $m$, if $Y \in \mathcal{M}^m_{poss}(P, r, t+1)$ and $Y \notin \mathcal{M}_{seen}(P, r, t+1)$ then

(a) if $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r, t+1)$ then $X, K \in \mathcal{M}^m_{poss}(P, r, t+1)$, and

(b) if $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r, t+1)$ then $X, K \in \mathcal{M}^m_{poss}(P, r, t+1)$, and

(c) if $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)$ then $X_1, \ldots, X_k \in \mathcal{M}^m_{poss}(P, r, t+1)$.

We show the above hypothesis implies that, if $Y \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$ and $Y \notin \mathcal{M}_{seen}(P, r, t+1)$ then

(a) if $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r, t+1)$ then $X, K \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$, and

(b) if $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r, t+1)$ then $X, K \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$, and

(c) if $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)$ then $X_1, \ldots, X_k \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$.

Suppose $Y \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$ and $Y \notin \mathcal{M}_{seen}(P, r, t+1)$. By definition 4.4,

$$
\begin{aligned}
Y \in\ & \mathcal{M}^{m}_{poss}(P, r, t+1) \\
& \cup \{E_K(X) \mid X, K \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X, K) \in \mathcal{E}(r, t+1)\} \\
& \cup \{H_K(X) \mid X, K \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X, K) \in \mathcal{H}(r, t+1)\} \\
& \cup \{(X_1 \mid \cdots \mid X_k) \mid X_1, \ldots, X_k \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } \\
& (X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)\} \\
& \cup \{X \mid E_K(X), K^{-1} \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X, K) \in \mathcal{E}(r, t+1)\} \\
& \cup \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)\}.
\end{aligned}
$$

Case (A): $Y \in \mathcal{M}^{m}_{poss}(P, r, t+1)$. The required statement follows from HP2 and Lemma 4.9.

Case (B): $Y \in \{E_K(X) \mid X, K \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X, K) \in \mathcal{E}(r, t+1)\}$.
We have $Y = E_{K'}(X')$ for some $X', K' \in \mathcal{M}^{m}_{poss}(P, r, t+1)$ such that $(X', K') \in \mathcal{E}(r, t+1)$. By definition 4.3, $E_{K'}(X') \in \mathcal{M}_{encr}(r, t+1)$, and therefore, $Y \in \mathcal{M}_{encr}(r, t+1)$.

Case (a): $Y = E_K(X)$ for some $(X, K) \in \mathcal{E}(r, t+1)$.
RE1 yields $X = X'$ and $K = K'$. Hence $X, K \in \mathcal{M}^{m}_{poss}(P, r, t+1)$.
By Lemma 4.9 it follows that $X, K \in \mathcal{M}^{m+1}_{poss}(P, r, t+1)$.

Case (b): $Y = H_K(X)$ for some $(X, K) \in \mathcal{H}(r, t+1)$.
By definition 4.3, $Y \in \mathcal{M}_{hash}(r, t+1)$, which is impossible by RE2.

Case (c): $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)$.
By definition 4.3, $Y \in \mathcal{M}_{conc}(r, t+1)$, which is impossible by RE2.

Case (C): $Y \in \{H_K(X) \mid X, K \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X, K) \in \mathcal{H}(r, t+1)\}$.
Similar to Case (B).

Case (D): $Y \in \{(X_1 \mid \cdots \mid X_k) \mid X_1, \ldots, X_k \in \mathcal{M}^{m}_{poss}(P, r, t+1) \text{ and } (X_1, \ldots, X_k) \in \mathcal{C}(r, t+1)\}$.
Similar to Case (B).

Case (E): $Y \in \{X \mid E_K(X), K^{-1} \in \mathcal{M}^m_{poss}(P,r,t+1)$ and $(X,K) \in \mathcal{E}(r,t+1)\}$.
We have $E_K(Y), K^{-1} \in \mathcal{M}^m_{poss}(P,r,t+1)$ for some $K$ such that $(Y,K) \in \mathcal{E}(r,t+1)$. By Lemma 4.9 it follows that $E_K(Y), K^{-1} \in \mathcal{M}_{poss}(P,r,t+1)$. Since $Y \notin \mathcal{M}_{seen}(P,r,t+1)$ it follows by the contrapositive of Lemma 4.17(a) that $E_K(Y) \notin \mathcal{M}_{seen}(P,r,t+1)$. HP2 yields $Y, K \in \mathcal{M}^m_{poss}(P,r,t+1)$. The required statement then follows from HP2 and Lemma 4.9.

Case (F): $Y \in \{X_i \mid (X_1 \mid \cdots \mid X_k) \in \mathcal{M}^m_{poss}(P,r,t+1)$ and $(X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)\}$.
We have $Y = X_i$ for some $i$ and for some $X_1 \mid \cdots \mid X_k \in \mathcal{M}^m_{poss}(P,r,t+1)$ such that $(X_1,\ldots,X_k) \in \mathcal{C}(r,t+1)$. Since $X_i \notin \mathcal{M}_{seen}(P,r,t+1)$ for some $i$, it follows by the contrapositive of Lemma 4.17(b) that $X_1 \mid \cdots \mid X_k \notin \mathcal{M}_{seen}(P,r,t+1)$. HP2 yields $X_1,\ldots,X_k \in \mathcal{M}^m_{poss}(P,r,t+1)$. But $Y = X_i$ for some $i$, so $Y \in \mathcal{M}^m_{poss}(P,r,t+1)$. The required statement then follows from HP2 and Lemma 4.9.

(This completes the proof of Proposition 4.2.) □

Looking back at the proof of Proposition 4.2, it is apparent that we could have proved a stronger statement. We can refine the hypothesis further to prove the following result, for example: Suppose $Y \in \mathcal{M}^m_{poss}(P,r,t)$ for some $m$, and suppose that $Y \notin \mathcal{M}_{seen}(P,r,t)$. If $Y = E_K(X)$ for some $(X,K) \in \mathcal{E}(r,t)$, then

$$X, K \in \begin{cases} \mathcal{M}_{poss}(P,r,t-1) & \text{if } m = 0 \text{ and } t > t_{\text{first}}(r) \\ \mathcal{M}^{m-1}_{poss}(P,r,t) & \text{if } m > 0. \end{cases}$$

However, the statement of Proposition 4.2 is less cumbersome and proves to be more direct for our purposes.

The following theorem is easily proved from Proposition 4.2 and Lemma 4.9.

**Theorem 4.1** *Let $Y \in \mathcal{M}_{poss}(P,r,t)$, and suppose that $Y \notin \mathcal{M}_{seen}(P,r,t)$.*

*(a) If $Y = E_K(X)$ for some $(X,K) \in \mathcal{E}(r,t)$, then $X, K \in \mathcal{M}_{poss}(P,r,t)$.*

*(b) If $Y = H_K(X)$ for some $(X,K) \in \mathcal{H}(r,t)$, then $X, K \in \mathcal{M}_{poss}(P,r,t)$.*

It is apparent that we have omitted the following case from the statement of Theorem 4.1: *(c) If $Y = X_1 \mid \cdots \mid X_k$ for some $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, then $X_1,\ldots,X_k \in \mathcal{M}_{poss}(P,r,t)$.* We do this simply because the omitted case is exactly part (e) of Lemma 4.13, which, however, does not require the extra hypothesis that $Y \notin \mathcal{M}_{seen}(P,r,t)$.

**Theorem 4.2** *If* $X \in \mathcal{M}_{seen}(P,r,t)$, *then* $X \in \mathcal{M}_{said}(Q,r,t')$ *for some* $Q$ *and for some* $t' < t$.

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3, $\mathcal{M}_{seen}(P,r,t_{\text{first}}(r)) = \emptyset$. Therefore, the required statement holds vacuously.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) for all $t' < t$, if $X \in \mathcal{M}_{seen}(P,r,t')$, then $X \in \mathcal{M}_{said}(Q,r,t'')$ for some $Q$ and for some $t'' < t'$; and show this implies that, if $X \in \mathcal{M}_{seen}(P,r,t)$ then $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$.

   By Lemma 4.14 it suffices to show that, for all $m$, if $X \in \mathcal{M}^m_{seen}(P,r,t)$, then $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$. We show this by induction on $m$:

   I. (Basis) Let $m = 0$. Suppose $X \in \mathcal{M}^0_{seen}(P,r,t)$. By definition 4.5, $X \in \mathcal{M}_{seen}(P,r,t-1) \cup \mathcal{S}_{recv}(P,r,t-1)$.

   Case (i): $X \in \mathcal{M}_{seen}(P,r,t-1)$.
   HP1 yields $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t-1$.

   Case (ii): $X \in \mathcal{S}_{recv}(P,r,t-1)$.
   RE3 yields: there exists a $Q$ which performs $send(X)$ at some $t' < t-1$. By definition 4.2, $X \in \mathcal{S}_{said}(Q,r,t')$, and therefore, by definition 4.6, $X \in \mathcal{M}^0_{said}(Q,r,t'+1)$. By Lemma 4.18 it follows that $X \in \mathcal{M}_{said}(Q,r,t'+1)$, which is as required, since $t'+1 < t$.

   II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis: (HP2) if $X \in \mathcal{M}^m_{seen}(P,r,t)$, then $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$; and we show this implies that, if $X \in \mathcal{M}^{m+1}_{seen}(P,r,t)$, then $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$. Suppose $X \in \mathcal{M}^{m+1}_{seen}(P,r,t)$. By definition 4.5,

$$\begin{aligned} X \in\ & \mathcal{M}^m_{seen}(P,r,t) \\ & \cup \{Y \mid E_K(Y) \in \mathcal{M}^m_{seen}(P,r,t) \text{ and } (Y,K) \in \mathcal{E}(r,t) \text{ and} \\ & K^{-1} \in \mathcal{M}_{poss}(P,r,t)\} \\ & \cup \{Y_i \mid (Y_1 \mid \cdots \mid Y_k) \in \mathcal{M}^m_{seen}(P,r,t) \text{ and } (Y_1,\ldots,Y_k) \in \mathcal{C}(r,t)\} \end{aligned}$$

   Case (A): $X \in \mathcal{M}^m_{seen}(P,r,t)$.
   HP2 yields $X \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$.

Case (B): $X \in \{Y \mid E_K(Y) \in \mathcal{M}^m_{seen}(P,r,t)$ and $(Y,K) \in \mathcal{E}(r,t)$ and $K^{-1} \in \mathcal{M}_{poss}(P,r,t)\}$.

We have, for some $K$, $E_K(X) \in \mathcal{M}^m_{seen}(P,r,t)$, $(X,K) \in \mathcal{E}(r,t)$, and $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$. HP2 yields $E_K(X) \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$. Consider the smallest $t' < t$ for which there exists $Q$ such that $E_K(X) \in \mathcal{M}_{said}(Q,r,t')$, and fix one such $Q$. Thus, for all $R$ and for all $t'' < t'$, $E_K(X) \notin \mathcal{M}_{said}(R,r,t'')$. By the contrapositive of the inductive hypothesis HP1 it follows that $E_K(X) \notin \mathcal{M}_{seen}(Q,r,t')$. Since $E_K(X) \in \mathcal{M}_{said}(Q,r,t')$, it follows by Lemma 4.19 that $E_K(X) \in \mathcal{M}_{poss}(Q,r,t')$. Also, we have $(X,K) \in \mathcal{E}(r,t)$ and $t' < t$. By Lemma 4.12 it follows that $(X,K) \in \mathcal{E}(r,t')$. Since $E_K(X) \in \mathcal{M}_{poss}(Q,r,t')$ and $E_K(X) \notin \mathcal{M}_{seen}(Q,r,t')$, it follows by Theorem 4.1 that $X$, $K \in \mathcal{M}_{poss}(Q,r,t')$. By Lemma 4.20(a) it follows that $X$, $K \in \mathcal{M}_{said}(Q,r,t')$.

Case (C): $X \in \{Y_i \mid (Y_1 \mid \cdots \mid Y_k) \in \mathcal{M}^m_{seen}(P,r,t)$ and $(Y_1,\ldots,Y_k) \in \mathcal{C}(r,t)\}$.

We have $X = Y_i$ for some $i$ such that $Y_1 \mid \cdots \mid Y_k \in \mathcal{M}^m_{seen}(P,r,t)$ and $(Y_1,\ldots,Y_k) \in \mathcal{C}(r,t)$. HP2 yields $Y_1 \mid \cdots \mid Y_k \in \mathcal{M}_{said}(Q,r,t')$ for some $Q$ and for some $t' < t$. By Lemma 4.19 it follows that $Y_1 \mid \cdots \mid Y_k \in \mathcal{M}_{poss}(Q,r,t')$. Also, we have $(Y_1,\ldots,Y_k) \in \mathcal{C}(r,t)$ and $t' < t$. By Lemma 4.12(c) it follows that $(Y_1,\ldots,Y_k) \in \mathcal{C}(r,t')$. By Lemma 4.20(c) it follows that $Y_1,\ldots,Y_k \in \mathcal{M}_{said}(Q,r,t')$. But $X = Y_i$ for some $i$, so $X \in \mathcal{M}_{said}(Q,r,t')$.

(This completes the proof of Theorem 4.2.) □

## 4.3 Related work

The semantic model developed in this chapter alleviates some major deficiencies of existing models for authentication logics proposed by Abadi and Tuttle [8], and Syverson and van Oorschot [9]. In particular, the problems it addresses include the following:

- A fundamental problem with existing models is that they reflect the syntax of the corresponding logics. As emphasized by Syverson [27], this makes the proof of soundness of such logics largely trivial and uninformative.

- A more compelling problem with existing models is that the definitions made as part of the models are generally not made sufficiently accurate. As a result, there

is often confusion about exactly what properties can be proved as a consequence of such definitions. For example, the AT logic paper claims that the logic proposed by its authors is sound with respect to the model defined in that paper. However, it has been subsequently reported that one of the axioms of the AT logic is unsound (cf. Syverson and van Oorschot [9]). Indeed, no detailed proofs of soundness of the logics AT and SVO have been published yet.

- Existing models leave implicit some critical assumptions that underlie authentication logics; for example, the assumption that messages can only be constructed in a unique way within the system, which is formally captured in our model as restrictions RE1 and RE2. It is difficult to see how proofs of properties which depend on such assumptions can be carried out formally in existing models.

Overall, existing models do not appear to enable proofs of desired properties to be carried out rigorously.

Although our model is motivated by notions found in previous works, it is essentially independent of any logical syntax. It formalizes various critical assumptions that underlie authentication logics, but which are nonetheless absent from existing models for such logics. In contrast to previous works, we have provided detailed and accurate proofs of the properties of our model. Our model is therefore a major advance as compared to the models of Abadi and Tuttle [8], and Syverson and van Oorschot [9].

# Chapter 5

# The soundness of a logic of authentication

This chapter presents a logic for analyzing authentication protocols. The logic presented here is motivated by the model developed in the previous chapter. The semantics we give for the logic is based on this model; thus our logic has an essentially independently motivated semantics. We demonstrate the virtue of this approach by giving a mathematically rigorous and intuitively convincing proof of soundness of the logic. While the syntax of the logic presented in this chapter is somewhat similar in appearance to that of the logics AT and SVO, there is a significant underlying difference nonetheless; namely, that the soundness of our logic is proved rigorously. As emphasized elsewhere in this thesis, claims regarding the soundness of the logics AT and SVO appear unsupported by published evidence.

## 5.1 Logic

### 5.1.1 Syntax

We begin by defining a formal language $\ell$. Although $\ell$ is defined without essential regard to the intended interpretation, its structure is motivated by that interpretation.

The symbols of $\ell$ are defined as follows.

1. *Logical symbols*

   $|\quad H \quad E \quad {}^{-1}$

   $\neg \quad \wedge \quad \vee \quad \Rightarrow \quad \Leftrightarrow$

   $\mathit{occurs\_encr} \quad \mathit{occurs\_hash} \quad \mathit{occurs\_conc}$

   $\mathit{fresh}$

   $\mathit{generates} \quad \mathit{received} \quad \mathit{sees} \quad \mathit{said} \quad \mathit{says} \quad \mathit{has} \quad \mathit{recognizes}$

   $\leftrightarrow$

*believes* *controls*

2. *Parameters*

$P_1$, $P_2$, ..., $P_n$ (for some fixed natural number $n$)

$K_1$, $K_2$, ...

$N_1$, $N_2$, ...

$q_1$, $q_2$, ...

The classification of the symbols into the above two classes is motivated by their intended interpretation: the logical symbols are the symbols whose interpretation will be fixed, whereas the interpretation of the parameters will be allowed to vary. However, this distinction plays no essential role in characterizing the language itself. The symbols $P_1, \ldots, P_n$ are called *principal symbols.* The symbols $K_i$ are called *key symbols.* The symbols $N_i$ are called *nonce symbols.* The symbols $q_i$ are called *propositional symbols.* The symbols $P_i$, $K_i$, and $N_i$ are called *primitive symbols.*

### Formation rules

We distinguish two classes of expressions in $\ell$: the *terms* and the *formulas.* The *terms* are the expressions which under their intended interpretation represent messages. The *formulas* are the expressions which under their intended interpretation represent assertions about messages.

The *terms* are defined as follows.

T1. Any primitive symbol is a term.

T2. For each fixed positive integer $k$ if $X_1, \ldots, X_k$ are terms, then $X_1 \mid \cdots \mid X_k$ is a term.

T3. If $X$ is a term and $K$ is a key symbol, then $E_K(X)$ and $H_K(X)$ are terms.

T4. If $K$ is a key symbol, then $K^{-1}$ is a term.

T5. No expression is a term unless it can be shown to be so from (T1)–(T4).

The *formulas* are defined as follows.

F1. Any propositional symbol is a formula.

F2. If $\phi$ is a formula, then so is $\neg\phi$.

F3. If $\phi$ and $\psi$ are formulas, then so are $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \Rightarrow \psi$, and $\phi \Leftrightarrow \psi$.

F4. If $X$ is a term and $K$ is a key symbol, then $occurs\_encr(X, K)$ and $occurs\_hash(X, K)$ are formulas.

F5. For each fixed positive integer $k$ if $X_1, \ldots, X_k$ are terms, then $occurs\_conc(X_1, \ldots, X_k)$ is a formula.

F6. If $X$ is a term, then $fresh(X)$ is a formula.

F7. If $P$ is a principal symbol and $X$ is a term, then $P$ *generates* $X$, $P$ *received* $X$, $P$ *sees* $X$, $P$ *said* $X$, $P$ *says* $X$, and $P$ *has* $X$ are formulas.

F8. If $K$ is a key symbol and $P$ and $Q$ are principal symbols, then $P \overset{K}{\leftrightarrow} Q$ is a formula.

F9. If $P$ is a principal symbol and $\phi$ is a formula, then $P$ *believes* $\phi$ and $P$ *controls* $\phi$ are formulas.

F10. No expression is a formula unless it can be shown to be so from (F1)–(F9).

## Formal system

We now define a formal system, called L, which consists of the language $\ell$ together with a deductive apparatus for $\ell$. The deductive apparatus is specified by defining the following: (1) a set of axioms; (2) a finite set of inference rules.

The axioms of L are divided into two classes: the *logical axioms* and the *proper axioms* (also called, *nonlogical axioms*). We shall fix a set of formulas as the logical axioms. The set of proper axioms consists of formulas which are protocol-specific, and is thus left unspecified. By an inference rule $\rho$ we mean a relation among formulas: if a set of formulas $\Gamma$ is in relation $\rho$ to a formula $\phi$, then we say that $\phi$ is a *direct consequence* of the formulas in $\Gamma$ by virtue of $\rho$.

The set of logical axioms and the set of inference rules is fixed as follows.

1. **Logical axioms**

   We define the set of logical axioms in terms of *axiom-schemas*, all instances of which are logical axioms. To give the axiom-schemas, we need several classes of metavariables. Let

   - $\phi$, $\chi$, $\psi$ be metavariables ranging over formulas,

- $P$, $Q$, $R$ be metavariables ranging over principal symbols, and
- $X$, $X_1$, $X_2$, ... be metavariables ranging over terms.

Let $k$ range over the set of all positive integers, and let $i$ range over the set $\{1, \ldots, k\}$ for each fixed $k$.

The following are the axiom-schemas of L:

A1. $\phi \Rightarrow (\chi \Rightarrow \phi)$

A2. $(\phi \Rightarrow (\chi \Rightarrow \psi)) \Rightarrow ((\phi \Rightarrow \chi) \Rightarrow (\phi \Rightarrow \psi))$

A3. $(\neg\chi \Rightarrow \neg\phi) \Rightarrow ((\neg\chi \Rightarrow \phi) \Rightarrow \chi)$

A4. $P$ *generates* $X \Rightarrow P$ *has* $X$

A5. $P$ *sees* $X \Rightarrow P$ *has* $X$

A6. $P$ *said* $X \Rightarrow P$ *has* $X$

A7. $P$ *has* $X \wedge P$ *has* $K \wedge$ *occurs_encr*$(X, K) \Rightarrow P$ *has* $E_K(X)$

A8. $P$ *has* $X \wedge P$ *has* $K \wedge$ *occurs_hash*$(X, K) \Rightarrow P$ *has* $H_K(X)$

A9. $P$ *has* $X_1 \wedge \cdots \wedge P$ *has* $X_k \wedge$ *occurs_conc*$(X_1, \ldots, X_k) \Rightarrow P$ *has* $X_1 \mid \cdots \mid X_k$

A10. $P$ *has* $E_K(X) \wedge$ *occurs_encr*$(X, K) \wedge P$ *has* $K^{-1} \Rightarrow P$ *has* $X$

A11. $P$ *has* $X_1 \mid \cdots \mid X_k \wedge$ *occurs_conc*$(X_1, \ldots, X_k) \Rightarrow P$ *has* $X_1 \wedge \cdots \wedge P$ *has* $X_k$

A12. $P$ *received* $X \Rightarrow P$ *sees* $X$

A13. $P$ *sees* $E_K(X) \wedge$ *occurs_encr*$(X, K) \wedge P$ *has* $K^{-1} \Rightarrow P$ *sees* $X$

A14. $P$ *sees* $X_1 \mid \cdots \mid X_k \wedge$ *occurs_conc*$(X_1, \ldots, X_k) \Rightarrow$
$P$ *sees* $X_1 \wedge \cdots \wedge P$ *sees* $X_k$

A15. $P$ *said* $E_K(X) \wedge$ *occurs_encr*$(X, K) \wedge P$ *has* $X \wedge P$ *has* $K \Rightarrow$
$P$ *said* $X \wedge P$ *said* $K$

A16. $P$ *said* $H_K(X) \wedge$ *occurs_hash*$(X, K) \wedge P$ *has* $X \wedge P$ *has* $K \Rightarrow$
$P$ *said* $X \wedge P$ *said* $K$

A17. $P$ *said* $X_1 \mid \cdots \mid X_k \wedge$ *occurs_conc*$(X_1, \ldots, X_k) \Rightarrow$
$P$ *said* $X_1 \wedge \cdots \wedge P$ *said* $X_k$

A18. $P$ *says* $X \Rightarrow P$ *said* $X$

A19. $P$ *said* $X \wedge$ *fresh*$(X) \Rightarrow P$ *says* $X$

A20. *fresh*$(X_i) \wedge$ *occurs_conc*$(X_1, \ldots, X_k) \Rightarrow$ *fresh*$(X_1 \mid \cdots \mid X_k)$

A21. $fresh(X) \wedge occurs\_encr(X, K) \Rightarrow fresh(E_K(X))$

A22. $fresh(K) \wedge occurs\_encr(X, K) \Rightarrow fresh(E_K(X))$

A23. $fresh(X) \wedge occurs\_hash(X, K) \Rightarrow fresh(H_K(X))$

A24. $fresh(K) \wedge occurs\_hash(X, K) \Rightarrow fresh(H_K(X))$

A25. $P \overset{K}{\leftrightarrow} Q \Leftrightarrow Q \overset{K}{\leftrightarrow} P$

A26. $P \overset{K}{\leftrightarrow} Q \wedge R \text{ sees } E_K(X) \wedge occurs\_encr(X, K) \Rightarrow$
$(P \text{ said } X \wedge P \text{ said } E_K(X) \wedge P \text{ has } K) \vee (Q \text{ said } X \wedge Q \text{ said } E_K(X) \wedge Q \text{ has } K)$

A27. $P \overset{K}{\leftrightarrow} Q \wedge R \text{ sees } H_K(X) \wedge occurs\_hash(X, K) \Rightarrow$
$(P \text{ said } X \wedge P \text{ said } H_K(X) \wedge P \text{ has } K) \vee$
$(Q \text{ said } X \wedge Q \text{ said } H_K(X) \wedge Q \text{ has } K)$

A28. $P \text{ believes } \phi \wedge P \text{ believes } (\phi \Rightarrow \psi) \Rightarrow P \text{ believes } \psi$

A29. $P \text{ believes } \phi \Rightarrow P \text{ believes } (P \text{ believes } \phi)$

A30. $\neg P \text{ believes } \phi \Rightarrow P \text{ believes } (\neg P \text{ believes } \phi)$

A31. $P \text{ controls } \phi \wedge P \text{ believes } \phi \Rightarrow \phi$

2. **Inference rules**

R1. (Modus Ponens) If $\phi$ and $\psi$ are any formulas, then $\psi$ is a direct consequence of $\phi$ and $\phi \Rightarrow \psi$.

R2. (Necessitation) If $\phi$ is any formula and $P$ any principal symbol, then $P \text{ believes } \phi$ is a direct consequence of $\phi$.

(This completes the definition of L.)

For the purpose of studying properties of L, we define some standard proof-theoretic notions: *proof in* L, *theorem of* L, and *deduction in* L *from a set of formulas.*

**Definition 5.1** A *proof in* L is a finite sequence of formulas $\phi_1, \ldots, \phi_k$ such that, for each $i$, either $\phi_i$ is an axiom, or $\phi_i$ is a direct consequence of some preceding formulas by a rule of inference.

**Definition 5.2** A formula $\phi$ is a *theorem of* L (written $\vdash_L \phi$) if $\phi$ is the last formula of a proof in L.

Notice that all axioms (logical or proper) of L are theorems of L.

**Definition 5.3** A *deduction in* L *from a set of formulas* $\Gamma$ is a finite sequence of formulas $\phi_1, \ldots, \phi_k$ such that, for each $i$, either $\phi_i$ is an axiom, or $\phi_i$ is an element of $\Gamma$, or $\phi_i$ is a direct consequence of some preceding formulas by a rule of inference.

**Definition 5.4** A formula $\phi$ is *deducible in* L *from a set of formulas* $\Gamma$ (written $\Gamma \vdash_L \phi$) if $\phi$ is the last formula of a deduction in L from $\Gamma$.

The following lemma is easily proven from the above definitions.

**Lemma 5.1** *Let $\phi$, $\psi$ be any formulas and $\Gamma$, $\Delta$ any sets of formulas. Let $P$ be any principal symbol.*

*(a) If $\Gamma$ is the empty set, then $\Gamma \vdash_L \phi$ iff $\vdash_L \phi$.*

*(b) If $\Gamma \vdash_L \phi$ then $\Gamma \cup \Delta \vdash_L \phi$.*

*(c) $\Gamma \vdash \phi$ iff there is a finite subset $\Sigma$ of $\Gamma$ such that $\Sigma \vdash_L \phi$.*

*(d) If $\Gamma \vdash_L \phi$ and $\Gamma \vdash_L \phi \Rightarrow \psi$, then $\Gamma \vdash_L \psi$.*

*(e) If $\Gamma \vdash_L \phi$ then $\Gamma \vdash_L P$ believes $\phi$.*

### 5.1.2 Semantics

We introduce a *possible worlds* framework. Fix a system, say, $P_1, \ldots, P_n$, where $n$ is the number of principal symbols in $\ell$. Intuitively, a *world* is an ordered pair $(r, t)$, which consists of a run $r$ of the system and a time $t$. Let $\mathcal{R}$ be the set of all runs of the system. If $R \subseteq \mathcal{R}$, call $\{(r, t) \mid r \in R \text{ and } t \geq t_{\text{first}}(r)\}$ the *set of worlds of* $R$, denoted $w(R)$. The semantics we define is of a model-theoretic nature; it rests on the usual notions: *interpretation*, *truth for an interpretation*, and *validity*. Roughly, an *interpretation* is a structure relative to which *truth* is defined. The class of structures we take as interpretations is essentially due to Kripke. (Since their invention *Kripke structures* have become a pervasive tool in giving semantics for modal logics.) For our purposes an interpretation consists of the following components: a set of runs $R \subseteq \mathcal{R}$, a truth assignment to the primitive propositions with respect to the set of worlds of $R$, $n$ binary relations (one for each principal) on the set of worlds of $R$, called *possibility relations*, and a function $f$ which maps terms of $\ell$ to messages in $\mathcal{M}$.

**Definition 5.5** Let $\Phi_0$ be the set of propositional symbols of $\ell$. An *interpretation of* $\ell$ is a tuple $I = (R, \pi, \sim_1, \ldots, \sim_n, f)$, where:

1. $R \subseteq \mathcal{R}$,

2. $\pi : \Phi_0 \to 2^{w(R)}$,

3. for each $i$, $\sim_i$ is a binary relation on $w(R)$ (so that $\sim_i \subseteq w(R) \times w(R)$) which is transitive and euclidean, and

4. (a) $f$ maps each principal symbol to a distinct element of $\mathcal{P}$ (the set of principal names);

   (b) $f$ maps each key symbol to an element of $\mathcal{K}$ (the set of keys);

   (c) $f$ maps each nonce symbol to an element of $\mathcal{N}$ (the set of nonces);

   (d) if $X_1, \ldots, X_k$ are terms, then $f(X_1 \mid \cdots \mid X_k) = f(X_1) \mid \cdots \mid f(X_k)$ (the concatenation of the strings $f(X_1), \ldots, f(X_k)$);

   (e) if $K$ is a key symbol and $X$ is a term, then $f(E_K(X)) = E_{f(K)}(f(X))$ (the symbol $E$ on the right-hand side is the semantic keyed encryption function defined in the model);

   (f) $f(H_K(X)) = H_{f(K)}(f(X))$ (the symbol $H$ on the right-hand side is the semantic keyed hash function defined in the model);

   (g) if $K$ is a key symbol, then $f(K^{-1}) = (f(K))^{-1}$ (the symbol $^{-1}$ on the right-hand side is the function from $\mathcal{K}$ to $\mathcal{K}^{-1}$ specified earlier).

Although the above definition fixes the possibility relations to be transitive and euclidean, there is considerable flexibility in choosing alternative properties. We follow the usual idea that a principal's possibility relation determines its *beliefs*, and that the properties of the possibility relation govern the properties of the notion of belief.

**Convention.** We normally suppress $f$; for example, instead of $f(K)$ we write $K$. Any resulting ambiguity is resolved from the context.

Fix an interpretation $I = (R, \pi, \sim_1, \ldots, \sim_n, f)$. If $(r,t) \in w(R)$, we say that $(r,t)$ *is in* $I$. We now define what it means for a *formula* $\phi$ *to be true for* $(r,t)$ *in* $I$ (written $\models^I_{(r,t)} \phi$). The definition proceeds by induction on the structure of $\phi$.

**Definition 5.6** For all $i, j \in \{1, \ldots, n\}$ and for all positive integers $l$:

1. $\models^I_{(r,t)} q_m$ iff $(r,t) \in \pi(q_m)$, for $m = 1, 2, \ldots$.

2. $\models^I_{(r,t)} \neg\phi$ iff not $\models^I_{(r,t)} \phi$.

3. $\models^I_{(r,t)} \phi \wedge \psi$ iff $\models^I_{(r,t)} \phi$ and $\models^I_{(r,t)} \psi$.

4. $\models^I_{(r,t)} \phi \vee \psi$ iff $\models^I_{(r,t)} \phi$ or $\models^I_{(r,t)} \psi$ or both.

5. $\models^I_{(r,t)} \phi \Rightarrow \psi$ iff either not $\models^I_{(r,t)} \phi$, or $\models^I_{(r,t)} \psi$, or both.

6. $\models^I_{(r,t)} \phi \Leftrightarrow \psi$ iff either $\models^I_{(r,t)} \phi$ and $\models^I_{(r,t)} \psi$, or not $\models^I_{(r,t)} \phi$ and not $\models^I_{(r,t)} \psi$.

7. $\models^I_{(r,t)} P_i$ *generates* $X$ iff $X \in \mathcal{M}genr(P_i, r, t)$.

8. $\models^I_{(r,t)} P_i$ *received* $X$ iff $X \in \mathcal{M}recv(P_i, r, t)$.

9. $\models^I_{(r,t)} P_i$ *sees* $X$ iff $X \in \mathcal{M}seen(P_i, r, t)$.

10. $\models^I_{(r,t)} P_i$ *said* $X$ iff $X \in \mathcal{M}_{said}(P_i, r, t)$.

11. $\models^I_{(r,t)} P_i$ *says* $X$ iff $X \in \mathcal{M}_{said}(P_i, r, t) \setminus \mathcal{M}_{said}(P_i, r, 0)$.

12. $\models^I_{(r,t)} P_i$ *has* $X$ iff $X \in \mathcal{M}poss(P_i, r, t)$.

13. $\models^I_{(r,t)}$ *occurs_encr*$(X, K)$ iff $(X, K) \in \mathcal{E}(r, t)$.

14. $\models^I_{(r,t)}$ *occurs_hash*$(X, K)$ iff $(X, K) \in \mathcal{H}(r, t)$.

15. $\models^I_{(r,t)}$ *occurs_conc*$(X_1, \ldots, X_l)$ iff $(X_1, \ldots, X_l) \in \mathcal{C}(r, t)$.

16. $\models^I_{(r,t)}$ *fresh*$(X)$ iff $X \notin \mathcal{M}_{said}(P_k, r, 0)$ for all $k = 1, \ldots, n$.

17. $\models^I_{(r,t)} P_i \overset{K}{\leftrightarrow} P_j$ iff for all $t' \leq t$, for all $X$, for all $k = 1, \ldots, n$:

    (a) if $E_K(X) \in \mathcal{M}_{said}(P_k, r, t')$ and $(X, K) \in \mathcal{E}(r, t')$, then $E_K(X) \in \mathcal{M}seen(P_k, r, t')$ or $P_k \in \{P_i, P_j\}$ or both, and

    (b) if $H_K(X) \in \mathcal{M}_{said}(P_k, r, t')$ and $(X, K) \in \mathcal{H}(r, t')$, then $H_K(X) \in \mathcal{M}seen(P_k, r, t')$ or $P_k \in \{P_i, P_j\}$ or both.

18. $\models^I_{(r,t)} P_i$ *believes* $\phi$ iff for all worlds $(r', t')$ in $I$, if $(r, t) \sim_i (r', t')$ then $\models^I_{(r',t')} \phi$.

19. $\models^I_{(r,t)} P_i$ *controls* $\phi$ iff $\models^I_{(r,t)} P_i$ *believes* $\phi$ implies $\models^I_{(r,t)} \phi$.

The truth conditions defined above need some explanation. Clause (1) reflects what has already been noted before: we fix the truth of propositional symbols by means of $\pi$. Clauses (2)–(6) reflect standard propositional truth assignments for $\neg$, $\wedge$, $\vee$, $\Rightarrow$, and $\Leftrightarrow$. Each of the clauses (7)–(15), with the seeming exception of clause (11), reflects notions that we have independently developed in the model of the previous

chapter. However, *says* is simply a derived notion: the truth condition for *says* is essentially that for *said* with an added restriction. A similar comment applies to the notion reflected by clause (16); the only novelty here is that we quantify over all sets of said messages for a fixed time of 0. Essentially, the truth condition for $\leftrightarrow$ captures the following intuition: a *key* $K$ *is shared between principal* $P$ *and* $Q$ iff $P$ and $Q$ are the only principals encrypting and hashing messages using $K$. Clause (18) reflects the standard possible worlds view of *belief*. Roughly, it says that a principal $P$ believes exactly those facts that are true in the worlds $P$ considers possible.

To this point, the notion of truth is defined relative to a given interpretation and a world in that interpretation. As usual, we extend this notion to truth with respect to a given interpretation and define validity in terms of truth for all interpretations.

**Definition 5.7** A formula $\phi$ is *true for an interpretation* $I$ (written $\models^I \phi$) iff $\phi$ is true for every world in $I$.

**Definition 5.8** A formula $\phi$ is *valid* (written $\models \phi$) iff $\phi$ is true for every interpretation.

The following proposition shows that the inference rules preserve truth with respect to interpretations.

**Proposition 5.1** *Let* $\phi$, $\psi$ *be any formulas and* $P$ *any principal symbol. For any interpretation* $I$*:*

*(a) If* $\models^I \phi$ *and* $\models^I \phi \Rightarrow \psi$, *then* $\models^I \psi$.

*(b) If* $\models^I \phi$, *then* $\models^I P$ *believes* $\phi$.

**Proof.**

(a) Suppose there is an interpretation $I$ such that $\models^I \phi$ and $\models^I \phi \Rightarrow \psi$. Then $\models^I_w \phi$ and $\models^I_w \phi \Rightarrow \psi$ for every $w$ in $I$. Therefore, by condition 5 of definition 5.6, $\models^I_w \psi$ for every $w$ in $I$; that is, $\models^I \psi$, as required.

(b) Suppose there is an interpretation $I$ such that $\models^I \phi$. Then $\models^I_w \phi$ for every $w$ in $I$. Therefore, by condition 18 of definition 5.6, $\models^I_w P$ *believes* $\phi$ for every $w$ in $I$; that is, $\models^I P$ *believes* $\phi$, as required.

(This completes the proof of Proposition 5.1.) □

**Corollary 5.1** *Let* $\phi$ *and* $\psi$ *be any formulas and* $P$ *any principal symbol.*

*(a) If* $\models \phi$ *and* $\models \phi \Rightarrow \psi$, *then* $\models \psi$.

*(b) If* $\models \phi$, *then* $\models P$ *believes* $\phi$.

Hereafter we write $\not\models^I \phi$ to mean not $\models^I \phi$; and similarly for the cases with or without subscripts and superscripts. If $P$ denotes a principal symbol, write $\sim_P$ to stand for the possibility relation of the principal denoted by $P$.

We now proceed to show that all the logical axioms of L are valid.

**Lemma 5.2** *The following formulas are valid:*

*(a)* $\phi \Rightarrow (\chi \Rightarrow \phi)$

*(b)* $(\phi \Rightarrow (\chi \Rightarrow \psi)) \Rightarrow ((\phi \Rightarrow \chi) \Rightarrow (\phi \Rightarrow \psi))$

*(c)* $(\neg\chi \Rightarrow \neg\phi) \Rightarrow ((\neg\chi \Rightarrow \phi) \Rightarrow \chi)$

**Proof.** We only prove part (a); the remaining parts are proved similarly. Take an arbitrary interpretation $I$ and an arbitrary world $w$ in $I$ such that $\models^I_w \phi$. From condition 5 of definition 5.6 and the fact that $\models^I_w \phi$, it follows that $\models^I_w \chi \Rightarrow \phi$, as required.

(This completes the proof of Lemma 5.2.) □

**Lemma 5.3** *The following formulas are valid:*

*(a) P generates X* $\Rightarrow$ *P has X*

*(b) P received X* $\Rightarrow$ *P sees X*

*(c) P sees X* $\Rightarrow$ *P has X*

*(d) P said X* $\Rightarrow$ *P has X*

**Proof.**

(a) Take an arbitrary interpretation $I$ and an arbitrary world $(r,t)$ in $I$ such that $\models^I_{(r,t)}$ *P generates X*. Then, by condition 7 of definition 5.6, $X \in \mathcal{M}_{genr}(P,r,t)$, and therefore, by Lemma 4.10, $X \in \mathcal{M}_{poss}(P,r,t)$. Hence, by condition 12, $\models^I_{(r,t)}$ *P has X*, as required.

(b) Follows similarly using Lemma 4.15.

(c) Follows similarly using Lemma 4.16.

(d) Follows similarly using Lemma 4.19.

(This completes the proof of Lemma 5.3.) □

**Lemma 5.4** *The following formulas are valid:*

*(a)* $P \text{ has } X \wedge P \text{ has } K \wedge \mathit{occurs\_encr}(X, K) \Rightarrow P \text{ has } E_K(X)$

*(b)* $P \text{ has } X \wedge P \text{ has } K \wedge \mathit{occurs\_hash}(X, K) \Rightarrow P \text{ has } H_K(X)$

*(c)* $P \text{ has } X_1 \wedge \cdots \wedge P \text{ has } X_k \wedge \mathit{occurs\_conc}(X_1, \ldots, X_k) \Rightarrow P \text{ has } X_1 \mid \cdots \mid X_k$

*(d)* $P \text{ has } E_K(X) \wedge \mathit{occurs\_encr}(X, K) \wedge P \text{ has } K^{-1} \Rightarrow P \text{ has } X$

*(e)* $P \text{ has } X_1 \mid \cdots \mid X_k \wedge \mathit{occurs\_conc}(X_1, \ldots, X_k) \Rightarrow P \text{ has } X_1 \wedge \cdots \wedge P \text{ has } X_k$

**Proof.** We only prove part (a); the remaining parts are proved similarly. Take an arbitrary interpretation $I$ and an arbitrary world $(r, t)$ in $I$ such that $\models^I_{(r,t)} P \text{ has } X \wedge P \text{ has } K \wedge \mathit{occurs\_encr}(X, K)$. Then, by conditions 3, 12, and 14 of definition 5.6, $X, K \in \mathcal{M}_{poss}(P, r, t)$ and $(X, K) \in \mathcal{E}(r, t)$, and therefore, by Lemma 4.13(a), $E_K(X) \in \mathcal{M}_{poss}(P, r, t)$. Hence, by condition 12, $\models^I_{(r,t)} P \text{ has } E_K(X)$, as required.

(This completes the proof of Lemma 5.4.) □

**Lemma 5.5** *The following formulas are valid:*

*(a)* $P \text{ sees } E_K(X) \wedge \mathit{occurs\_encr}(X, K) \wedge P \text{ has } K^{-1} \Rightarrow P \text{ sees } X$

*(b)* $P \text{ sees } X_1 \mid \cdots \mid X_k \wedge \mathit{occurs\_conc}(X_1, \ldots, X_k) \Rightarrow P \text{ sees } X_1 \wedge \cdots \wedge P \text{ sees } X_k$

**Proof.** We only prove part (a); the remaining part is proved similarly. Take an arbitrary interpretation $I$ and an arbitrary world $(r, t)$ in $I$ such that $\models^I_{(r,t)} P \text{ sees } E_K(X) \wedge \mathit{occurs\_encr}(X, K) \wedge P \text{ has } K^{-1}$. Then, by conditions 3, 9, 12, and 14 of definition 5.6, $E_K(X) \in \mathcal{M}_{seen}(P, r, t)$, $(X, K) \in \mathcal{E}(r, t)$, and $K^{-1} \in \mathcal{M}_{poss}(P, r, t)$, and therefore, by Lemma 4.17(a), $X \in \mathcal{M}_{seen}(P, r, t)$. Hence, by condition 9, $\models^I_{(r,t)} P \text{ sees } X$, as required.

(This completes the proof of Lemma 5.5.) □

**Lemma 5.6** *The following formulas are valid:*

*(a)* $P \text{ said } E_K(X) \wedge \mathit{occurs\_encr}(X, K) \wedge P \text{ has } X \wedge P \text{ has } K \Rightarrow P \text{ said } X \wedge P \text{ said } K$

*(b)* $P \text{ said } H_K(X) \wedge \mathit{occurs\_hash}(X, K) \wedge P \text{ has } X \wedge P \text{ has } K \Rightarrow P \text{ said } X \wedge P \text{ said } K$

(c) $P\ said\ X_1 \mid \cdots \mid X_k \wedge occurs\_conc(X_1, \ldots, X_k) \Rightarrow P\ said\ X_1 \wedge \cdots \wedge P\ said\ X_k$

**Proof.** We only prove part (a); the remaining parts are proved similarly. Take an arbitrary interpretation $I$ and an arbitrary world $(r,t)$ in $I$ such that $\models^I_{(r,t)} P\ said\ E_K(X) \wedge occurs\_encr(X,K) \wedge P\ has\ X \wedge P\ has\ K \Rightarrow P\ said\ X \wedge P\ said\ K$. Then, by conditions 3, 10, 12 and 13 of definition 5.6, $E_K(X) \in \mathcal{M}_{said}(P,r,t)$, $(X,K) \in \mathcal{E}(r,t)$, and $X, K \in \mathcal{M}_{poss}(P,r,t)$, and therefore, by Lemma 4.20(a), $X, K \in \mathcal{M}_{said}(P,r,t)$. Hence, by conditions 3 and 10, $\models^I_{(r,t)} P\ said\ X \wedge P\ said\ K$, as required.

(This completes the proof of Lemma 5.6.) □

**Lemma 5.7** *The following formulas are valid:*

(a) $P\ says\ X \Rightarrow P\ said\ X$

(b) $P\ said\ X \wedge fresh(X) \Rightarrow P\ says\ X$

**Proof.**

(a) Take an arbitrary interpretation $I$ and an arbitrary world $(r,t)$ in $I$ such that $\models^I_{(r,t)} P\ says\ X$. Then, by condition 11 of definition 5.6, $X \in \mathcal{M}_{said}(P,r,t) \setminus \mathcal{M}_{said}(P,r,0)$, and therefore, $X \in \mathcal{M}_{said}(P,r,t)$. Hence, by condition 11, $\models^I_{(r,t)} P\ said\ X$, as required.

(b) Take an arbitrary interpretation $I$ and an arbitrary world $(r,t)$ in $I$ such that $\models^I_{(r,t)} P\ said\ X \wedge fresh(X)$. Then, by conditions 3, 10 and 16 of definition 5.6, $X \in \mathcal{M}_{said}(P,r,t)$ and $X \notin \mathcal{M}_{said}(Q,r,0)$ for all $Q$; in particular, $X \notin \mathcal{M}_{said}(P,r,0)$, and therefore, $X \in \mathcal{M}_{said}(P,r,t) \setminus \mathcal{M}_{said}(P,r,0)$. Hence, by condition 11, $\models^I_{(r,t)} P\ says\ X$, as required.

(This completes the proof of Lemma 5.6.) □

**Lemma 5.8** *The following formula is valid:*

$$fresh(X_i) \wedge occurs\_conc(X_1, \ldots, X_k) \Rightarrow fresh(X_1 \mid \cdots \mid X_k)$$

**Proof.** (By contradiction.) Suppose there is an interpretation $I$ for which the formula $fresh(X_i) \wedge occurs\_conc(X_1, \ldots, X_k) \Rightarrow fresh(X_1 \mid \cdots \mid X_k)$ is not true. Then there exists a world $(r,t)$ in $I$ such that $\not\models^I_{(r,t)} fresh(X_i) \wedge occurs\_conc(X_1, \ldots, X_k) \Rightarrow fresh(X_1 \mid \cdots \mid X_k)$. By condition 5 of definition 5.6, $\models^I_{(r,t)} fresh(X_i) \wedge occurs\_conc(X_1, \ldots, X_k)$

and $\not\models^I_{(r,t)} \mathit{fresh}(X_1 \mid \cdots \mid X_k)$. By conditions 3, 15 and 16, $X_i \notin \mathcal{M}_{said}(P,r,0)$ for all $P$, $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, and $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{said}(Q,r,0)$ for some $Q$. If $(X_1,\ldots,X_k) \in \mathcal{C}(r,0)$, then, by Lemma 4.20(c), $X_1,\ldots,X_k \in \mathcal{M}_{said}(Q,r,0)$, and the required statement follows by contradiction. It remains to show that $(X_1,\ldots,X_k) \in \mathcal{C}(r,0)$.

Case (i): Let $t \leq 0$. From $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$, it follows that $(X_1,\ldots,X_k) \in \mathcal{C}(r,0)$, trivially when $t = 0$, and by Lemma 4.3 when $t < 0$.

Case (ii): Let $t > 0$. Since $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{said}(Q,r,0)$, it follows by Lemma 4.19 that $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{poss}(Q,r,0)$. Also, $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$. By Lemma 4.12(c) it follows that $(X_1,\ldots,X_k) \in \mathcal{C}(r,0)$.

(This completes the proof of Lemma 5.8.) □

**Lemma 5.9** *The following formulas are valid:*

*(a)* $\mathit{fresh}(X) \wedge \mathit{occurs\_encr}(X,K) \Rightarrow \mathit{fresh}(E_K(X))$

*(b)* $\mathit{fresh}(K) \wedge \mathit{occurs\_encr}(\dot{X},K) \Rightarrow \mathit{fresh}(E_K(X))$

*(c)* $\mathit{fresh}(X) \wedge \mathit{occurs\_hash}(X,K) \Rightarrow \mathit{fresh}(H_K(X))$

*(d)* $\mathit{fresh}(K) \wedge \mathit{occurs\_hash}(X,K) \Rightarrow \mathit{fresh}(H_K(X))$

**Proof.** (By contradiction.) We only prove part (a); the remaining parts are proved similarly. Suppose there is an interpretation $I$ for which the formula $\mathit{fresh}(X) \wedge \mathit{occurs\_encr}(X,K) \Rightarrow \mathit{fresh}(E_K(X))$ is not true. Then there exists a world $(r,t)$ in $I$ such that $\not\models^I_{(r,t)} \mathit{fresh}(X) \wedge \mathit{occurs\_encr}(X,K) \Rightarrow \mathit{fresh}(E_K(X))$. By condition 5 of definition 5.6, $\models^I_{(r,t)} \mathit{fresh}(X) \wedge \mathit{occurs\_encr}(X,K)$ and $\not\models^I_{(r,t)} \mathit{fresh}(E_K(X))$. By conditions 3, 13, and 16, $X \notin \mathcal{M}_{said}(P,r,0)$ for all $P$, $(X,K) \in \mathcal{E}(r,t)$, and $E_K(X) \in \mathcal{M}_{said}(Q,r,0)$ for some $Q$. Consider the smallest $t' \leq 0$ for which there exists $R$ such that $E_K(X) \in \mathcal{M}_{said}(R,r,t')$, and fix one such $R$. Thus, for all $R'$ and for all $t'' < t'$, $E_K(X) \notin \mathcal{M}_{said}(R',r,t'')$. The contrapositive of Theorem 4.2 yields $E_K(X) \notin \mathcal{M}_{seen}(R,r,t')$. Since $E_K(X) \in \mathcal{M}_{said}(R,r,t')$, it follows by Lemma 4.19 that $E_K(X) \in \mathcal{M}_{poss}(R,r,t')$. We now show that $(X,K) \in \mathcal{E}(r,t')$. Recall that $(X,K) \in \mathcal{E}(r,t)$.

Case (i): Let $t \leq t'$. It follows that $(X,K) \in \mathcal{E}(r,t')$, trivially when $t = t'$, and by Lemma 4.3 when $t < t'$.

Case (ii): Let $t > t'$. By Lemma 4.12(a) it follows that $(X, K) \in \mathcal{E}(r, t')$.

Thus, $(X, K) \in \mathcal{E}(r, t')$. By Theorem 4.1 it follows that $X, K \in \mathcal{M}_{poss}(R, r, t')$. Since $E_K(X) \in \mathcal{M}_{said}(R, r, t')$ it follows by Lemma 4.18 that $E_K(X) \in \mathcal{M}^m_{said}(R, r, t')$ for some $m$, and therefore, by definition 4.6, $X, K \in \mathcal{M}^{m+1}_{said}(R, r, t')$. By Lemma 4.18, $X, K \in \mathcal{M}_{said}(R, r, t')$, and therefore, by Lemma 4.2, $X, K \in \mathcal{M}_{said}(R, r, 0)$ since $t' \leq 0$, which contradicts the fact that $X \notin \mathcal{M}_{said}(P, r, 0)$ for all $P$.

(This completes the proof of Lemma 5.6.) □

**Lemma 5.10** *The following formula is valid:*

1. $P \overset{K}{\leftrightarrows} Q \Leftrightarrow Q \overset{K}{\leftrightarrows} P$

**Proof.** Obvious. □

**Lemma 5.11** *The following formulas are valid:*

(a) $P \overset{K}{\leftrightarrows} Q \wedge R \textit{ sees } E_K(X) \wedge \textit{occurs\_encr}(X, K) \Rightarrow$
$(P \textit{ said } X \wedge P \textit{ said } E_K(X) \wedge P \textit{ has } K) \vee (Q \textit{ said } X \wedge Q \textit{ said } E_K(X) \wedge Q \textit{ has } K)$

(b) $P \overset{K}{\leftrightarrows} Q \wedge R \textit{ sees } H_K(X) \wedge \textit{occurs\_hash}(X, K) \Rightarrow$
$(P \textit{ said } X \wedge P \textit{ said } H_K(X) \wedge P \textit{ has } K) \vee (Q \textit{ said } X \wedge Q \textit{ said } H_K(X) \wedge Q \textit{ has } K)$

**Proof.** We only prove part (a); the remaining part is proved similarly. Take an arbitrary interpretation $I$ and an arbitrary world $(r, t)$ in $I$ such that $\models^I_{(r,t)} P \overset{K}{\leftrightarrows} Q \wedge R \textit{ sees } E_K(X) \wedge \textit{occurs\_encr}(X, K)$. Then, by condition 3 of definition 5.6, $\models^I_{(r,t)} P \overset{K}{\leftrightarrows} Q$, $\models^I_{(r,t)} R \textit{ sees } E_K(X)$, and $\models^I_{(r,t)} \textit{occurs\_encr}(X, K)$. By condition 9, $E_K(X) \in \mathcal{M}_{seen}(R, r, t)$, and therefore, by Theorem 4.2, $E_K(X) \in \mathcal{M}_{said}(R', r, t')$ for some $R'$ and for some $t' < t$. Consider the smallest $t' < t$ for which there exists $R'$ such that $E_K(X) \in \mathcal{M}_{said}(R', r, t')$, and fix one such $R'$. Thus, for all $R''$ and for all $t'' < t'$, $E_K(X) \notin \mathcal{M}_{said}(R'', r, t'')$. By the contrapositive of Theorem 4.2 it follows that $E_K(X) \notin \mathcal{M}_{seen}(R', r, t')$. Since $E_K(X) \in \mathcal{M}_{said}(R', r, t')$, it follows by Lemma 4.19 that $E_K(X) \in \mathcal{M}_{poss}(R', r, t')$. By condition 14, $(X, K) \in \mathcal{E}(r, t)$, and therefore, by Lemma 4.12(a), $(X, K) \in \mathcal{E}(r, t')$ since $t' < t$. Hence, by condition 18(a), $R' \in \{P, Q\}$. Also, by Theorem 4.1, $X, K \in \mathcal{M}_{poss}(R', r, t')$. Since $E_K(X) \in \mathcal{M}_{said}(R', r, t')$, it follows by Lemma 4.18 that $E_K(X) \in \mathcal{M}^m_{said}(R', r, t')$ for some $m$, and therefore, by definition 4.6, $X, K \in \mathcal{M}^{m+1}_{said}(R', r, t')$. By Lemma 4.18, $X, K \in \mathcal{M}_{said}(R', r, t')$.

Hence, by Lemma 4.2, $E_K(X), X \in \mathcal{M}_{said}(R', r, t)$ and $K \in \mathcal{M}_{poss}(R', r, t)$ since $t' < t$. Hence, by conditions 3, 4, 10 and 12, $\models^I_{(r,t)}$ ($P$ *said* $X \wedge P$ *said* $E_K(X) \wedge P$ *has* $K$) $\vee$ ($Q$ *said* $X \wedge Q$ *said* $E_K(X) \wedge Q$ *has* $K$), as required.

(This completes the proof of Lemma 5.11.) □

**Lemma 5.12** *The following formulas are valid:*

*(a)* $P$ *believes* $\phi \wedge P$ *believes* $(\phi \Rightarrow \psi) \Rightarrow P$ *believes* $\psi$

*(b)* $P$ *believes* $\phi \Rightarrow P$ *believes* ($P$ *believes* $\phi$)

*(c)* $\neg P$ *believes* $\phi \Rightarrow P$ *believes* $\neg$($P$ *believes* $\phi$)

**Proof.**

(a) Take an arbitrary interpretation $I$ and an arbitrary world $w$ in $I$ such that $\models^I_w$ $P$ *believes* $\phi \wedge P$ *believes* $(\phi \Rightarrow \psi)$. Then, by conditions 3 and 18 of definition 5.6, $\models^I_{w'} \phi$ and $\models^I_{w'} \phi \Rightarrow \psi$ for every $w'$ in $I$ such that $w \sim_P w'$. Therefore, by condition 5, $\models_{w'} \phi$ for every $w'$ in $I$ such that $w \sim_P w'$. Hence, by condition 18, $\models^I_w P$ *believes* $\psi$, as required.

(b) Take an arbitrary interpretation $I$ and an arbitrary world $w$ in $I$ such that $\models^I_w$ $P$ *believes* $\phi$. Then, by condition 18 of definition 5.6, (*) for every $w'$ in $I$ such that $w \sim_P w'$, $\models^I_{w'} \phi$. We wish to show that $\models^I_w P$ *believes* ($P$ *believes* $\phi$). By condition 18 it suffices to show that for every $w'$ in $I$ such that $w \sim_P w'$, and for every $w''$ in $I$ such that $w' \sim_P w''$, $\models^I_{w''} \phi$. This statement clearly holds by the transitivity of $\sim_P$ and (*).

(c) Take an arbitrary interpretation $I$ and an arbitrary world $w$ in $I$ such that $\models^I_w$ $\neg P$ *believes* $\phi$. By conditions 2 and 18 of definition 5.6, (**) there exists a world $w_0$ in $I$ such that $w \sim_P w_0$ and $\models^I_{w_0} \neg\phi$. We wish to show that $\models^I_w P$ *believes* $\neg$($P$ *believes* $\phi$). By conditions 2 and 18 it suffices to show that for every $w'$ in $I$ such that $w \sim_P w'$ there is a $w''$ in $I$ such that $w' \sim_P w''$ and $\models^I_{w''} \neg\phi$. But if $w'$ is a world in $I$ such that $w \sim_P w'$, then from the euclideanness of $\sim_P$ and (**) it follows that $w_0$ is a world in $I$ such that $w' \sim_P w_0$ and $\models^I_{w_0} \neg\phi$, as required.

(This completes the proof of Lemma 5.12.) □

**Lemma 5.13** *The following formula is valid:*

$$P \text{ believes } \phi \wedge P \text{ controls } \phi \Rightarrow \phi$$

**Proof.** Take an arbitrary interpretation $I$ and an arbitrary world $(r,t)$ in $I$ such that $\models^I_{(r,t)} P$ *believes* $\phi \wedge P$ *controls* $\phi$. Then, by condition 3 of definition 5.6, $\models^I_{(r,t)} P$ *believes* $\phi$ and $\models^I_{(r,t)} P$ *controls* $\phi$. Hence, by condition 19, $\models^I_{(r,t)} \phi$, as required.

(This completes the proof of Lemma 5.13.) □

**Theorem 5.1** *Every logical axiom of* L *is valid.*

**Proof.**

1. Axioms (A1)–(A3) are valid, by Lemma 5.2.
2. Axioms (A4)–(A6), and axiom (A12) are valid, by Lemma 5.3.
3. Axioms (A7)–(A11) are valid, by Lemma 5.4.
4. Axioms (A13)–(A14) are valid, by Lemma 5.5.
5. Axioms (A15)–(A17) are valid, by Lemma 5.6.
6. Axioms (A18)–(A19) are valid, by Lemma 5.7.
7. Axiom (A20) is valid, by Lemma 5.8.
8. Axioms (A21)–(A24) are valid, by Lemma 5.9.
9. Axiom (A25) is valid, by Lemma 5.10.
10. Axioms (A26)–(A27) are valid, by Lemma 5.11.
11. Axioms (A28)–(A30) are valid, by Lemma 5.12.
12. Axiom (A31) is valid, by Lemma 5.13.

(This completes the proof of Theorem 5.1.) □

### Soundness theorems

Let $L_0$ be the system L with an empty set of proper axioms. Thus, the only axioms of $L_0$ are the logical axioms (A1) through (A32).

We are now ready to establish the main soundness theorem.

**Theorem 5.2** *Every theorem of* $L_0$ *is valid.*

**Proof.** Let $\phi$ be a theorem of $L_0$. The required statement is proved by induction on the length of a proof in $L_0$ of $\phi$.

Let $\phi_1, \ldots, \phi_n = \phi$ be the sequence of formulas of a proof in $L_0$ of $\phi$. We show, by induction on $i$, that $\models \phi_i$ for $1 \leq i \leq n$.

1. (Basis) Let $i = 1$. Then $\phi_1$ must be a logical axiom, and therefore, by Theorem 5.1, $\models \phi_1$.

2. (Induction) Let $i > 1$ be arbitrary. Assume the inductive hypothesis that, for all $j < i$, $\models \phi_j$.

Case (1): $\phi_i$ is a logical axiom. As in the basis step, $\models \phi_i$.

Case (2): $\phi_i$ follows by *modus ponens* from formulas $\phi_j$ and $\phi_m$, where $j < i$ and $m < i$, and $\phi_m$ is of the form $\phi_j \Rightarrow \phi_i$. By the inductive hypothesis, $\models \phi_j$ and $\models \phi_j \Rightarrow \phi_i$, and therefore, by Corollary 5.1(a), $\models \phi_i$.

Case (3): $\phi_i$ follows by *necessitation* from a formula $\phi_j$, where $j < i$, and $\phi_i$ is of the form $P$ *believes* $\phi_j$ for some principal symbol $P$. By the inductive hypothesis, $\models \phi_j$, and therefore, by Corollary 5.1(b), $\models P$ *believes* $\phi_j$, which is as required.

(This completes the proof of Theorem 5.2.) □

Note that the statement of Theorem 5.2 does not hold for L, since in general we allow L to contain proper axioms which can be arbitrary formulas. However, a modified form of the soundness theorem can still be obtained for L. Technically, when carrying out deductions in L we are only interested in those interpretations for which all the proper axioms are true. We can then prove soundness of L relative to such interpretations. (This idea is routinely used in the study of formal systems with proper axioms.)

**Definition 5.9** An interpretation $I$ is a *model of* L iff every axiom of L is true for $I$.

**Theorem 5.3** *Every theorem of* L *is true for any model of* L.

**Proof.** Suppose that $I$ is an interpretation for which all the axioms of L are true. Let $\phi$ be a theorem of L. The required statement is proved by induction on the length of a proof in L of $\phi$.

Let $\phi_1, \phi_2, \ldots, \phi_n = \phi$ be the sequence of formulas of a proof in L of $\phi$. We show, by induction on $i$, that $\models^I \phi_i$ for $1 \leq i \leq n$.

1. (Basis) Let $i = 1$. Then $\phi_1$ must be an axiom, and therefore, by supposition, $\models^I \phi_1$.

2. (Induction) Let $i > 1$ be arbitrary. Assume the inductive hypothesis that, for all $j < i$, $\models^I \phi_j$.

Case (1): $\phi_i$ is an axiom. As in the basis step, $\models^I \phi_i$.

Case (2): $\phi_i$ follows by *modus ponens* from formulas $\phi_j$ and $\phi_m$, where $j < i$ and $m < i$, and $\phi_m$ is of the form $\phi_j \Rightarrow \phi_i$. By the inductive hypothesis, $\models^I \phi_j$ and $\models^I \phi_j \Rightarrow \phi_i$, and therefore, by Proposition 5.1(a), $\models^I \phi_i$.

Case (3): $\phi_i$ follows by *necessitation* from a formula $\phi_j$, where $j < i$, and $\phi_i$ is of the form $P$ *believes* $\phi_j$ for some principal symbol $P$. By the inductive hypothesis, $\models^I \phi_j$, and therefore, by Proposition 5.1(b), $\models^I P$ *believes* $\phi_j$, which is as required since $\phi_i$ is of the form $P$ *believes* $\phi_j$.

(This completes the proof of Theorem 5.3.) □

# Chapter 6

# A model for reasoning about lower bounds on rounds

In this chapter we introduce a new model, which allows reasoning about lower bounds on rounds for a class of authentication protocols. This continues the theme of formal reasoning developed in the preceding chapters. The motivation for the model introduced here is a largely informal body of bounds arising from the work of Gong [36], [37], [38]. Our aim in developing the model is to provide a systematic means for deriving such bounds. In particular, we will show how some of the bounds intuitively obtained by Gong are formally derived in our model.

(Parts of this chapter appeared in preliminary form elsewhere [39].)

## 6.1 Introduction

An authentication protocol, in its barest form, consists of a sequence of message exchanges. The appeal of defining metrics for comparing authentication protocols is obvious. Of course, the most important aspect of a protocol is its correctness and there is a sizable amount of literature on this subject. However, the literature on metrics for authentication protocols is rather sparse. An essentially similar observation to the one made above motivates Gong [36], [37], [38] to study some efficiency metrics for authentication protocols. Specifically, he defines two efficiency metrics: *the number of messages* and *the number of rounds*. The former metric simply means the total number of message exchanges comprising a protocol. To define the latter metric, Gong uses the notion of *round*: a round consists of protocol messages that can be exchanged simultaneously—the number of rounds is then taken to mean the minimum number of rounds needed to complete the protocol. Notice that the notion of round reflects the concurrency inherent in a distributed protocol: multiple participants may simultaneously send or receive messages in one round. In his works, Gong [36], [37], [38] gives lower bounds on the above two metrics for some common protocol classes, in an

informal manner. Independently, Yahalom [40], [41] has devised a model for analyzing bounds on the number of messages for a class of secure asynchronous protocols. The model provides constructs for expressing security requirements using the notion of *verifiable causality*, which is related to Lamport's [42] *happened before* relation. Yahalom [40] employs the model to define a class of *secure data exchange* protocols, and derives a lower bound on the number of messages for this class. However, the metric of rounds is not addressed in his work.

Set against the above background, we introduce a model to formally derive bounds on rounds from security requirements. The idea behind our model can be sketched as follows. We adopt Yahalom's notion of verifiable causality between events as a means of specifying security requirements for asynchronous protocols. This allows us to define the notion of an abstract *protocol class* in terms of verifiable causality. A characteristic property of this notion is that it induces a partial order on an associated set of events; this partial order is a causal order in the sense of Lamport [42]. A round then precisely consists of a set of causally unordered events. The key upshot of the definitions we make to exploit this fact is that they lead us to a theorem for proving lower bounds on the number of rounds. The theorem gives rise to a simple graph-theoretic technique for finding bounds.

## 6.2 Basic model

We begin by recalling some of the notions described by Yahalom [40].

A *system* consists of a collection of nodes, also called *principals*, which communicate solely by *asynchronous message passing.* That is, we assume that: (1) the principals do not maintain synchronized clocks, and (2) the only means of communication between principals is via message exchanges. Each principal can generate a new pseudorandom value, called an *up-nonce*, which is unpredictable by others. It is assumed that principals may act maliciously, that is, they can see, modify, or replay any message exchanged within the system. Further, any principal can inject fake messages into the system.

An *event* is an action taken by a principal. The actions a principal can perform include the following: (i) sending a message $M$, denoted *send*($M$); (ii) receiving a message $M$, denoted *receive*($M$). Each node maintains its own local abstract clock. It is assumed that the local clock value at a principal is incremented at least once between two successive events at that principal. Each event $E$ is associated with the local clock

reading, $c(E)$, at the principal where that event occurs.

Following Lamport [42], we define a *happened before* relation, denoted $\rightarrow$, as the smallest binary relation on the set of events of a system satisfying the following conditions:

1. $E \rightarrow E'$ holds:

   (i) if $E$ and $E'$ are events occurring at the same principal such that $c(E) < c(E')$, or

   (ii) if $E = send(M)$ and $E' = receive(M)$ for any message $M$ exchanged between two principals, or

   (iii) if $E \rightarrow E''$ and $E'' \rightarrow E'$ for some $E''$.

2. $E \not\rightarrow E$ for all $E$.

The above definition essentially generalizes the following two basic observations about the order of events in a distributed system (cf. [43]): (a) A principal is a sequential process; that is, the events occurring at the same principal are totally ordered; (b) Whenever a message exchange takes place, the event of sending the message occurs before the event of receiving the message. It is easy to see that $\rightarrow$ is an irreflexive, transitive, anti-symmetric relation; that is, a partial order on the events of a system.

A basic property of $\rightarrow$ is concerned with a notion of *information flow* between events. If $E_i \rightarrow E_j$ for events $E_i$ and $E_j$ at two different principals $P_i$ and $P_j$, respectively, then the above definition implies that there exists a send event, $send(M)$, at $P_i$, and a receive event, $receive(M')$, at $P_j$, for some messages $M$ and $M'$, such that $send(M) \rightarrow receive(M')$. We then say that there is an information flow from $E_i$ to $E_j$.

Note that the *happened before* relation effectively captures the notion of *potential* causality: $E \rightarrow E'$ means $E$ *may* (but does not necessarily) causally affect $E'$. The basic idea underlying Yahalom's notion of *verifiable* causality is to capture *strict* causal dependence between events, in that the occurrence of one event is precluded without the occurrence of another event. This notion is relativised to principals, and causal dependence is further distinguished as *precedence* or *succession* between events, as follows.

**Definition 6.1 ([40])** An event $E_i$ of one principal $P_i$ verifiably-precedes an event $E_j$ of another principal $P_j$ if $P_i$ can establish that $E_j$ could not be generated without $P_j$ receiving some information derived from the occurrence of $E_i$ or from some event at $P_i$ that occurred after $E_i$.

**Definition 6.2** ([40]) An event $E_i$ of one principal $P_i$ verifiably-succeeds an event $E_j$ of another principal $P_j$ if, at the time it generates $E_i$, $P_i$ can establish that $E_j$ has occurred.

As noted by Yahalom [40], the notions of verifiable precedence and verifiable succession defined above are strictly independent: $E_i$ verifiably-precedes $E_j$ does not necessarily imply that $E_j$ verifiably-succeeds $E_i$ (and vice-versa).

The following propositions relate verifiable causality with potential causality.

**Proposition 6.1** ([40]) For any two events $E_j$ and $E_i$ that have occurred at different principals, if $E_i$ verifiably-precedes $E_j$ then $E_i \rightarrow E_j$.

**Proposition 6.2** ([40]) For any two events $E_j$ and $E_i$ that have occurred at different principals, if $E_i$ verifiably-succeeds $E_j$ then $E_j \rightarrow E_i$.

As noted by Yahalom [40], the two notions represented by $E_i$ verifiably-precedes $E_j$ and $E_j$ verifiably-succeeds $E_i$ are strictly stronger than $E_i \rightarrow E_j$. The converses of Propositions 6.1 and 6.2 do not hold.

The following definition is intended to capture the notion of an event at one principal occurring *relatively recently* with respect to an event at another principal.

**Definition 6.3** ([40]) An event $E_j$ of one principal $P_j$ $\triangle$-precedes an event $E_i$ of another principal $P_i$ if $P_i$ can establish that $E_j$ was generated at most $\triangle$ ticks (as measured by $P_i$ on its local site clock) before the generation of $E_i$.

In Yahalom's model, the notion of $\triangle$-precedence is central to capturing the security requirement that principals be able to determine that certain messages are fresh and not replays of earlier ones.

The following theorem (Theorem 1 of Yahalom [40]) gives necessary and sufficient conditions for $\triangle$-precedence.

**Theorem 6.1** An event $E_j$ of a principal $P_j$ at one site $\triangle$-precedes an event $E_i$ of principal $P_i$ at another site if and only if the following conditions hold:

1. There exists another event $E_i'$, generated by principal $P_i$, such that $E_i'$ verifiably-precedes $E_j$.
2. $E_i$ verifiably-succeeds $E_j$.
3. $c(E_i) - c(E_i') \leq \triangle$.

Note that the first condition above asserts that for a principal $P_i$ to establish that an event $E_j$ at a different principal $P_j$ $\triangle$-precedes an event $E_i$ at $P_i$, there must exist another event $E_i'$ at $P_i$, from which there is an information flow to $E_j$. This information flow implicitly includes a receive event (respectively, send event) of some message at $P_j$ (respectively, $P_i$). The received message at $P_j$ is referred to as a $\triangle$-*precedence establishing* ($\triangle$-pe) message by Yahalom [40].

Informally, a *protocol* defines a sequence of events at various principals. An *execution* of a protocol consists of a realization in which various protocol events take place at the principals involved. Each event is associated with the protocol execution where it occurs. Events that occur in different executions at the same principal are assumed to be unrelated, in that the clock values associated with such events are incomparable.

For the purpose of deriving bounds, the significance of the above model is that it allows us to deduce the information flows that are needed to satisfy some security requirements. Essentially, Yahalom [40] exploits this fact to obtain a lower bound on the number of messages for a particular class of protocols.

## 6.3 Extending the model: Rounds

For our purposes, we abstract a class of secure asynchronous protocols as a collection of protocols that achieve some goal defined using Yahalom's notions of verifiable causality. We represent such a goal in general by means of the following: (1) a finite set $\mathcal{E}_b$ of base events at various principals, and (2) a set $\mathcal{C}$ of verifiable causal relationships over $\mathcal{E}_b$ defined using verifiably-precedes, verifiably-succeeds, or $\triangle$-precedes. Clearly, $\mathcal{C}$ induces a partial order, defined by $\rightarrow$, on the set $\mathcal{E} = \mathcal{E}_b \cup \mathcal{E}_d$, where $\mathcal{E}_d$ is a possibly empty set of additional events induced by Theorem 6.1. We thus represent a protocol class formally as a partially ordered set $\Pi = (\mathcal{E}, \prec)$, where $\prec$ denotes the partial order associated with $\mathcal{E}$.

As an aside, we note that in light of the poset formulation for a protocol class, it appears natural to view an individual protocol of class $\Pi$ as a totally ordered set $(\mathcal{E}, <)$, where $<$ is a total order on $\mathcal{E}$ consistent with $\prec$; that is, such that $E \prec E'$ implies $E < E'$, for all $E, E' \in \mathcal{E}$. In other words, a protocol of class $\Pi$ may be thought of as a *topological sort* (cf. [44]) of the poset $(\mathcal{E}, \prec)$. However, we do not explore the notion of an individual protocol further, since the protocol class abstraction suffices here.

### 6.3.1 Rounds and causality

Gong [36] defines the metric *number of rounds* as follows:

> The number of rounds in a protocol is the total number of time units from the instant that the [protocol] originator sends the first message till the instant that the last message is received, under the best execution scenario. (p. 28)

Further, "A round consists of all messages that can be sent and received in parallel within one time unit" ([36], p. 27). For the sake of the above definition, Gong makes the following two idealized timing assumptions: (i) exactly one time unit elapses between sending and receiving of a message; and (ii) the processing time for any event is exactly zero time units. As Gong observes, the number of rounds gives a rough estimate on the execution time of a protocol.

For our purposes more precision is required than the definitions used by Gong. In our model, the notion of 'time' is captured by the *happened before* relation. We effectively use this relation to formulate below our counterparts to Gong's notions on rounds. First, we need to fix a message set associated with a protocol class.

**Definition 6.4** A *message* $M$ is a triple $(P, Q, m)$, $P \neq Q$, denoting that principals $P$ and $Q$ are the sender and recipient, respectively, of the message contents, $m$.

Thus, messages with the same contents but which are sent or received at different principals are distinct messages for our purposes. The case where a principal is meant to send the message contents to itself does not appear to be meaningful in our context. (Such messages do not serve to establish verifiable causality.) The side condition in the definition rules out this uninteresting case by excluding messages of the form $(P, P, m)$.

We fix a *message set* $\mathcal{M}$ on any protocol of class $\Pi = (\mathcal{E}, \prec)$ as the set of messages corresponding to the prescribed send (or alternatively, receive) events in $\mathcal{E}$:

$$\mathcal{M} = \{M \mid send(M) \in \mathcal{E}\}$$

Following Lamport [42], we say that events $E$ and $E'$ are *concurrent* if $E \not\rightarrow E'$ and $E' \not\rightarrow E$, and write this as $E \parallel E'$. We then define a round to consist of a subset of $\mathcal{M}$ for which the corresponding send events are concurrent.

**Definition 6.5** Let $\mathcal{M}' \subseteq \mathcal{M}$ be non-empty. Then $\mathcal{M}'$ is a *round* of $\mathcal{M}$, if $send(M) \parallel send(M')$ for all $M, M' \in \mathcal{M}'$.

Intuitively, the rounds comprising a protocol are mutually exclusive and exhaustive: each element of the message set belongs to exactly one round.

**Definition 6.6** A *round partition* of $\mathcal{M}$ is a partition $\pi$ of $\mathcal{M}$ such that every block of $\pi$ is a single round of $\mathcal{M}$.

We call a round partition linear if its blocks may be totally ordered to be consistent with $\prec$; this is intended to capture the idea that there is an execution order over that round partition.

**Definition 6.7** A round partition $\pi$ of $\mathcal{M}$ is *linear* if there exists a total order $<$ on $\pi$ satisfying the following restriction: for all $\pi_i, \pi_j \in \pi$, if there exist messages $M_i \in \pi_i$ and $M_j \in \pi_j$ such that $send(M_i) \prec send(M_j)$, then $\pi_i < \pi_j$.

A round partition may not necessarily be linear. For example, consider a hypothetical protocol class with:

$$\begin{aligned} \mathcal{M} &= \{M_1, M_2, M_3, M_4\} \\ \prec &= \{(send(M_1), send(M_2)), (send(M_3), send(M_4))\} \end{aligned}$$

where $M_1$, $M_2$, $M_3$, and $M_4$ are all distinct messages. In this example, the set $\{\{M_1, M_4\}, \{M_2, M_3\}\}$ is a round partition of $\mathcal{M}$ but not a linear round partition.

We can now define the number of rounds.

**Definition 6.8** The *number of rounds* for $\Pi$ is the rank of the smallest (having fewest blocks) linear round partition of $\mathcal{M}$.

Notice how our definition pins down the intended meaning of the phrase, "best execution scenario," seen in Gong's informal definition earlier.

### 6.3.2 Rounds and directed acyclic graphs

We now proceed to relate lower bounds on rounds with the structure of the poset defining a protocol class.

**Definition 6.9** Let $\Pi = (\mathcal{E}, \prec)$ be a protocol class. Define $\mathcal{E}_s \subseteq \mathcal{E}$ and $\prec_s \subseteq \prec$ such that:

1. $\mathcal{E}_s = \{E \mid E \in \mathcal{E}$ and E is a send event$\}$, and
2. $\prec_s = \prec \cap (\mathcal{E}_s \times \mathcal{E}_s)$.

Then the poset $(\mathcal{E}_s, \prec_s)$ is called the *send-poset* of $\Pi$.

Implicit in the above definition is the fact that $\prec_s$ is a partial order on $\mathcal{E}_s$. This fact follows directly from clauses 1 and 2. Technically, the send-poset of a protocol class $\Pi = (\mathcal{E}, \prec)$ is simply a restriction of the poset $(\mathcal{E}, \prec)$ to the send events in $\mathcal{E}$.

As before, let $\mathcal{M}$ denote the message set on $\Pi$.

**Lemma 6.1** *Let $n_1$ and $n_2$ be the ranks, respectively, of the smallest round partition and the smallest linear round partition of $\mathcal{M}$. Then $n_1 \leq n_2$.*

The proof of this lemma is immediate from the fact that the set of linear round partitions of $\mathcal{M}$ is a subset of the set of round partitions of $\mathcal{M}$.

We can now state our main theorem.

**Theorem 6.2** *Let $\Omega(\Pi)$ be the number of rounds for a protocol class $\Pi$ whose send-poset is $(\mathcal{E}_s, \prec_s)$. If there exist send events $send(M_1)$, $send(M_2)$, ..., $send(M_n) \in \mathcal{E}_s$ such that:*

$$send(M_1) \prec_s send(M_2) \prec_s \cdots \prec_s send(M_n),$$

*then $\Omega(\Pi) \geq n$.*

*Proof.* Assume that $send(M_i) \prec_s send(M_{i+1})$ for $i = 1, \ldots, n-1$. Since $\prec_s$ is irreflexive, we have $M_i \neq M_j$, when $i \neq j$. Therefore, the set $\mathcal{M}' = \{M_1, \ldots, M_n\}$ has exactly $n$ elements. Clearly, $\mathcal{M}' \subseteq \mathcal{M}$. Now, any subset of $\mathcal{M}$ containing two or more distinct elements of $\mathcal{M}'$ cannot be a round of $\mathcal{M}$. This follows from Definition 6.5, by the assumption: for all $M, M' \in \mathcal{M}'$, $M \neq M'$, we have $send(M) \not\parallel send(M')$. Then any round partition of $\mathcal{M}$ must contain at least $n$ blocks. Hence by Lemma 6.1 it follows that the smallest linear round partition of $\mathcal{M}$ must also contain at least $n$ blocks. □

To obtain the best lower bound implied by Theorem 6.2, we obviously need to find the longest chain of send events in $\mathcal{E}_s$. This is conveniently viewed in graph-theoretic terms: we can view the poset $(\mathcal{E}_s, \prec_s)$ as an acyclic digraph $G$, with $\mathcal{E}_s$ as the set of vertices and $\prec_s$ as the set of edges. The longest chain of send events in $\mathcal{E}_s$ then corresponds to the longest path between any pair of vertices in $G$.

## 6.4 Case study

We shall now demonstrate our model by deriving lower bounds on rounds for several classes of authenticated key exchange protocols informally analyzed by Gong [36].

The overall setting is as follows (cf. [36]): Two clients $A$ and $B$ share secret keys with a trusted server $S$. The protocol aim is to distribute a fresh temporary session key for use between the clients, followed by an optional handshake using the session key to verify the presence of clients. In particular, each client must be convinced that the message from where it gets the session key, as well as the message from which it confirms the presence of the other client, have not been replayed. (Hereafter, we refer to such messages as *session key message* and *handshake message*, respectively.) This is achieved using either nonces or timestamps as freshness identifiers, distinguished as nonce based – NB or timestamp based – TB. The session key goal is distinguished as AO – authentication only, or AH – authentication with handshake. The candidates for choosing the temporary key are distinguished as SO – server only, CO – one client only, or CC – both clients. In the CC case, the temporary session key is suitably derived from two individual partial key values respectively chosen by the clients.

The choice of the above setting parameters gives twelve protocol classes in all. We distinguish them using Gong's [36] shorthand notation: TB/NB + AO/AH + SO/CO/CC. (Examples of concrete protocols for each class can be found in Gong's paper [36].)

Since our model precludes synchronized clocks, it does not apply to the TB cases. The remaining six asynchronous (NB) cases, labeled Case 7–12 in Gong's paper [36], fit in with our model; we will consider each of these cases in turn below. First, we recall some general assumptions made by Gong [36, p. 28]:

H1 A client cannot send out a handshake message before it has received the temporary key. Thus, the last handshake message cannot be sent before all clients have received the temporary key.

H2 A client without a synchronized clock cannot accept a temporary key before it sends out a nonce.

H3 The protocol responder (client) or the server cannot send out any message (e.g., a nonce) before the protocol originator sends out a notification message.

(For convenience we have labeled Gong's assumptions above.) Further, client $A$ is designated as the protocol originator and client $B$ is called the protocol responder.

Some remarks on Gong's above assumptions are in order: (H1) implicitly reflects that knowledge of the temporary key is necessary to form the handshake message. (H2) is essentially captured in our model using Yahalom's Theorem 6.1. To see the

connection between the two, note that (H2) is informally based on the requirement that a client be able to verify the freshness of session key messages [36, p. 27]; the notion of $\Delta$-precedence allows us to express such requirements precisely. Observe that (H2) is simply a derived fact about the system, as implied by condition 1 of Yahalom's Theorem 1. We will directly capture (H1) and (H3) using the *happened before* relation. (H1) applies to the three AH cases, whereas (H3) is common to all six cases.

Without loss of generality, we assume in the following that the generation event of a message coincides with the send event of that message. For all protocol classes considered below, we make the following event definition:

| | |
|---|---|
| $e_{A,0}$ | send of protocol start message at $A$ |

In the remainder of this section, we prove lower bounds on rounds for the six protocol classes:

- NB+AO+SO
- NB+AH+SO
- NB+AO+CO
- NB+AH+CC
- NB+AO+CC
- NB+AH+CC

In our proofs, we make use of a Prolog procedure for maximal path finding in DAGs, which is shown in appendix D.

### 6.4.1 Protocol class NB+AO+SO

To specify this class, we define the following events:

| | |
|---|---|
| $e_{S,1}$ | send of session key message for $A$ at $S$ |
| $e_{S,2}$ | send of session key message for $B$ at $S$ |
| $e_{A,1}$ | receive of session key message at $A$ |
| $e_{B,1}$ | receive of session key message at $B$ |

and capture the session key goal as follows:

CR1 $e_{S,1}$ $\Delta$-precedes $e_{A,1}$

CR2 $e_{S,2}$ $\Delta$-precedes $e_{B,1}$

CR1 and CR2 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{A,2}$ and $e_{B,2}$, respectively, at $A$ and $B$ such that:

$$\begin{aligned} &\text{CR3} \quad e_{A,2} \rightarrow e_{S,1} \\ &\text{CR4} \quad e_{S,1} \rightarrow e_{A,1} \\ &\text{CR5} \quad e_{B,2} \rightarrow e_{S,2} \\ &\text{CR6} \quad e_{S,2} \rightarrow e_{B,1} \end{aligned}$$

To satisfy (H3), we stipulate the following constraints:

$$\begin{aligned} &\text{CR7} \quad e_{A,0} \rightarrow e_{S,1} \\ &\text{CR8} \quad e_{A,0} \rightarrow e_{B,2} \end{aligned}$$

We collect the above events and *happened before* relationships to form the required posets.

$(\mathcal{E}^7, \prec^7)$:

$$\begin{aligned} \mathcal{E}^7 &= \{e_{A,0}, e_{A,1}, e_{A,2}, e_{B,1}, e_{B,2}, e_{S,1}, e_{S,2}\} \\ \prec^7 &= \{(e_{A,2}, e_{S,1}), (e_{S,1}, e_{A,1}), (e_{B,2}, e_{S,2}), (e_{S,2}, e_{B,1}), (e_{A,0}, e_{S,1}), (e_{A,0}, e_{B,2})\} \end{aligned}$$

The partial order shown above does not explicitly include every pair of events which is ordered by $\rightarrow$, since the omitted pairs are deduced by the path-finding algorithm used later. We will tacitly follow this convention hereafter.

$(\mathcal{E}_s^7, \prec_s^7)$:

$$\begin{aligned} \mathcal{E}_s^7 &= \{e_{A,0}, e_{A,2}, e_{B,2}, e_{S,1}, e_{S,2}\} \\ \prec_s^7 &= \{(e_{A,2}, e_{S,1}), (e_{B,2}, e_{S,2}), (e_{A,0}, e_{S,1}), (e_{A,0}, e_{B,2})\} \end{aligned}$$

We now use the path finding program given in appendix D to obtain the best lower bound implied by Theorem 6.2. To save space, we only show the resulting output here:

```
MaxPath = [e(a,0),e(b,2),e(s,2)]
Bound = 3;
```

It is instructive to compare the maximal path found above with Gong's [36] informal proof:

> The responder [$B$] has to be notified before it can send out its nonce and later receive a fresh message; thus three rounds is a lower bound. (p. 30)

### 6.4.2 Protocol class NB+AH+SO

We define the following additional events:

| | |
|---|---|
| $e_{A,3}$ | send of handshake message for $B$ at $A$ |
| $e_{B,3}$ | send of handshake message for $A$ at $B$ |
| $e_{A,4}$ | receive of handshake message at $A$ |
| $e_{B,4}$ | receive of handshake message at $B$ |

and capture the handshake goal as follows:

$$\begin{array}{ll} \text{CR9} & e_{A,3} \triangle\text{-precedes } e_{B,4} \\ \text{CR10} & e_{B,3} \triangle\text{-precedes } e_{A,4} \end{array}$$

We capture (H1) using the following constraints:

$$\begin{array}{ll} \text{CR11} & e_{A,1} \rightarrow e_{A,3} \\ \text{CR12} & e_{B,1} \rightarrow e_{B,3} \end{array}$$

CR9 and CR10 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{B,5}$ and $e_{A,5}$, respectively, at $B$ and $A$ such that:

$$\text{CR13} \quad e_{B,5} \rightarrow e_{A,3}$$

$$\begin{array}{ll} \text{CR14} & e_{A,3} \rightarrow e_{B,4} \\ \text{CR15} & e_{A,5} \rightarrow e_{B,3} \\ \text{CR16} & e_{B,3} \rightarrow e_{A,4} \end{array}$$

To satisfy (H3), we stipulate the following additional constraint:

$$\text{CR17} \quad e_{A,0} \rightarrow e_{B,5}$$

$(\mathcal{E}^8, \prec^8)$:

$$\begin{aligned} \mathcal{E}^8 &= \mathcal{E}^7 \cup \{e_{A,3}, e_{A,4}, e_{A,5}, e_{B,3}, e_{B,4}, e_{B,5}\} \\ \prec^8 &= \prec^7 \cup \\ &\quad \{(e_{A,1}, e_{A,3}), (e_{B,1}, e_{B,3}), (e_{B,5}, e_{A,3}), (e_{A,3}, e_{B,4}), (e_{A,5}, e_{B,3}), (e_{B,3}, e_{A,4}), (e_{A,0}, e_{B,5})\} \end{aligned}$$

$(\mathcal{E}_s^8, \prec_s^8)$:

$$\begin{aligned} \mathcal{E}_s^8 &= \{e_{A,0}, e_{A,2}, e_{A,3}, e_{A,5}, e_{B,2}, e_{B,3}, e_{B,5}, e_{S,1}, e_{S,2}\} \\ \prec_s^8 &= \{(e_{A,2}, e_{S,1}), (e_{S,1}, e_{A,3}), (e_{B,2}, e_{S,2}), (e_{S,2}, e_{B,3}), (e_{A,0}, e_{S,1}), (e_{A,0}, e_{B,2}), \\ &\quad (e_{B,5}, e_{A,3}), (e_{A,5}, e_{B,3}), (e_{A,0}, e_{B,5})\} \end{aligned}$$

```
MaxPath = [e(a,0),e(b,2),e(s,2),e(b,3)]
Bound = 4;
```

The above path is simply an extension of the path found in the previous case. It is again instructive to compare with Gong's [36] informal proof:

> ... at least one more round is needed than in Case 7 [NB+AO+SO] to complete the handshake [after both clients have received the temporary key]; thus four rounds is a lower bound ... (p. 30)

### 6.4.3 Protocol class NB+AO+CO

Here we assume that the protocol responder chooses the session key. (The case where the protocol initiator chooses the session key can be similarly worked out.) To specify this class, we define the following events:

| | |
|---|---|
| $e_{S,1}$ | send of session key message for $A$ at $S$ |
| $e_{A,1}$ | receive of session key message at $A$ |
| $e_{B,1}$ | send of session key message for $S$ at $B$ |

and capture the session key goal as follows:

$$\text{CR1} \quad e_{S,1} \triangle\text{-precedes } e_{A,1}$$
$$\text{CR2} \quad e_{B,1} \triangle\text{-precedes } e_{A,1}$$
$$\text{CR3} \quad e_{B,1} \text{ verifiably-precedes } e_{S,1}$$

CR1 and CR2 respectively imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{A,2}$ and $e_{A,3}$ at $A$ such that:

$$\text{CR4} \quad e_{A,2} \rightarrow e_{S,1}$$
$$\text{CR5} \quad e_{S,1} \rightarrow e_{A,1}$$
$$\text{CR6} \quad e_{A,3} \rightarrow e_{B,1}$$
$$\text{CR7} \quad e_{B,1} \rightarrow e_{A,1}$$

CR3 implies by Proposition 6.1 that:

$$\text{CR8} \quad e_{B,1} \rightarrow e_{S,1}$$

To satisfy (H3), we stipulate the following constraints:

$$\text{CR9} \quad e_{A,0} \rightarrow e_{S,1}$$
$$\text{CR10} \quad e_{A,0} \rightarrow e_{B,1}$$

$(\mathcal{E}^9, \prec^9)$:

$$\mathcal{E}^9 = \{e_{A,0}, e_{A,1}, e_{A,2}, e_{A,3}, e_{B,1}, e_{S,1}\}$$
$$\prec^9 = \{(e_{A,2}, e_{S,1}), (e_{S,1}, e_{A,1}), (e_{A,3}, e_{B,1}), (e_{B,1}, e_{A,1}), (e_{B,1}, e_{S,1}), (e_{A,0}, e_{S,1}), (e_{A,0}, e_{B,1})\}$$

$(\mathcal{E}^9_s, \prec^9_s)$:

$$\mathcal{E}^9_s = \{e_{A,0}, e_{A,2}, e_{A,3}, e_{B,1}, e_{S,1}\}$$
$$\prec^9_s = \{(e_{A,2}, e_{S,1}), (e_{A,3}, e_{B,1}), (e_{B,1}, e_{S,1}), (e_{A,0}, e_{S,1}), (e_{A,0}, e_{B,1})\}$$

```
MaxPath = [e(a,3),e(b,1),e(s,1)]
Bound = 3;

MaxPath = [e(a,0),e(b,1),e(s,1)]
Bound = 3;
```

### 6.4.4 Protocol class NB+AH+CO

We introduce the following additional events:

| | |
|---|---|
| $e_{A,4}$ | send of handshake message for $B$ at $A$ |
| $e_{B,2}$ | send of handshake message for $A$ at $B$ |
| $e_{A,5}$ | receive of handshake message at $A$ |
| $e_{B,3}$ | receive of handshake message at $B$ |

and capture the handshake goal as follows:

CR11 $e_{A,4}$ $\triangle$-precedes $e_{B,3}$
CR12 $e_{B,2}$ $\triangle$-precedes $e_{A,5}$

We capture (H1) using the following constraint:

CR13 $e_{A,1} \rightarrow e_{A,4}$

CR11 and CR12 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{B,4}$ and $e_{A,6}$, respectively, at $B$ and $A$ such that:

CR14 $e_{B,4} \rightarrow e_{A,4}$
CR15 $e_{A,4} \rightarrow e_{B,3}$
CR16 $e_{A,6} \rightarrow e_{B,2}$
CR17 $e_{B,2} \rightarrow e_{A,5}$

To satisfy (H3), we stipulate the following additional constraints:

$$\text{CR18} \quad e_{A,0} \rightarrow e_{B,2}$$
$$\text{CR19} \quad e_{A,0} \rightarrow e_{B,4}$$

$(\mathcal{E}^{10}, \prec^{10})$:

$$\begin{aligned}\mathcal{E}^{10} &= \mathcal{E}^{9} \cup \{e_{A,4}, e_{A,5}, e_{A,6}, e_{B,2}, e_{B,3}, e_{B,4}\} \\ \prec^{10} &= \prec^{9} \cup \{(e_{A,1}, e_{A,4}), (e_{B,4}, e_{A,4}), (e_{A,4}, e_{B,3}), (e_{A,6}, e_{B,2}), (e_{B,2}, e_{A,5}), \\ &\quad (e_{A,0}, e_{B,2}), (e_{A,0}, e_{B,4})\}\end{aligned}$$

$(\mathcal{E}_s^{10}, \prec_s^{10})$:

$$\begin{aligned}\mathcal{E}_s^{10} &= \{e_{A,0}, e_{A,2}, e_{A,3}, e_{A,4}, e_{A,6}, e_{B,1}, e_{B,2}, e_{B,4}, e_{S,1}\} \\ \prec_s^{10} &= \{(e_{A,2}, e_{S,1}), (e_{S,1}, e_{A,4}), (e_{A,3}, e_{B,1}), (e_{B,1}, e_{A,4}), (e_{B,1}, e_{S,1}), (e_{A,0}, e_{S,1}), \\ &\quad (e_{A,0}, e_{B,1}), (e_{B,4}, e_{A,4}), (e_{A,6}, e_{B,2}), (e_{A,0}, e_{B,2}), (e_{A,0}, e_{B,4})\}\end{aligned}$$

```
MaxPath = [e(a,3),e(b,1),e(s,1),e(a,4)]
Bound = 4;

MaxPath = [e(a,0),e(b,1),e(s,1),e(a,4)]
Bound = 4;
```

### 6.4.5 Protocol class NB+AO+CC

To specify this class, we define the following events:

| | |
|---|---|
| $e_{S,1}$ | send of partial session key message for $A$ at $S$ |
| $e_{S,2}$ | send of partial session key message for $B$ at $S$ |
| $e_{A,1}$ | send of partial session key message for $S$ at $A$ |
| $e_{A,2}$ | receive of partial session key message at $A$ |
| $e_{B,1}$ | send of partial session key message for $S$ at $B$ |
| $e_{B,2}$ | receive of partial session key message at $B$ |

and capture the session key goal as follows:

$$\text{CR1} \quad e_{S,1} \triangle\text{-precedes } e_{A,2}$$
$$\text{CR2} \quad e_{S,2} \triangle\text{-precedes } e_{B,2}$$

$$\text{CR3} \quad e_{B,1} \triangle\text{-precedes } e_{A,2}$$
$$\text{CR4} \quad e_{A,1} \triangle\text{-precedes } e_{B,2}$$

CR5 $e_{B,1}$ verifiably-precedes $e_{S,1}$

CR6 $e_{A,1}$ verifiably-precedes $e_{S,2}$

CR1 and CR2 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{A,3}$ and $e_{B,3}$, respectively, at $A$ and $B$ such that:

CR7 $e_{A,3} \rightarrow e_{S,1}$

CR8 $e_{S,1} \rightarrow e_{A,2}$

CR9 $e_{B,3} \rightarrow e_{S,2}$

CR10 $e_{S,2} \rightarrow e_{B,2}$

CR3 and CR4 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{A,4}$ and $e_{B,4}$, respectively, at $A$ and $B$ such that:

CR11 $e_{A,4} \rightarrow e_{B,1}$

CR12 $e_{B,1} \rightarrow e_{A,2}$

CR13 $e_{B,4} \rightarrow e_{A,1}$

CR14 $e_{A,1} \rightarrow e_{B,2}$

CR5 and CR6 imply by Proposition 6.1 respectively the following:

CR15 $e_{B,1} \rightarrow e_{S,1}$

CR16 $e_{A,1} \rightarrow e_{S,2}$

To satisfy (H3), we stipulate the following constraints:

CR17 $e_{A,0} \rightarrow e_{B,1}$

CR18 $e_{A,0} \rightarrow e_{B,3}$

CR19 $e_{A,0} \rightarrow e_{B,4}$

$(\mathcal{E}^{11}, \prec^{11})$:

$$\mathcal{E}^{11} = \{e_{A,0}, e_{A,1}, e_{A,2}, e_{A,3}, e_{A,4}, e_{B,1}, e_{B,2}, e_{B,3}, e_{B,4}, e_{S,1}, e_{S,2}\}$$

$$\prec^{11} = \{(e_{A,3}, e_{S,1}), (e_{S,1}, e_{A,2}), (e_{B,3}, e_{S,2}), (e_{S,2}, e_{B,2}), (e_{A,4}, e_{B,1}), (e_{B,1}, e_{A,2}), (e_{B,4}, e_{A,1}), (e_{A,1}, e_{B,2}), (e_{B,1}, e_{S,1}), (e_{A,1}, e_{S,2}), (e_{A,0}, e_{B,1}), (e_{A,0}, e_{B,3}), (e_{A,0}, e_{B,4})\}$$

$(\mathcal{E}_s^{11}, \prec_s^{11})$:

$$\mathcal{E}_s^{11} = \{e_{A,0}, e_{A,1}, e_{A,3}, e_{A,4}, e_{B,1}, e_{B,3}, e_{B,4}, e_{S,1}, e_{S,2}\}$$

$$\prec_s^{11} = \{(e_{A,3}, e_{S,1}), (e_{B,3}, e_{S,2}), (e_{A,4}, e_{B,1}), (e_{B,4}, e_{A,1}), (e_{B,1}, e_{S,1}), (e_{A,1}, e_{S,2}), (e_{A,0}, e_{B,1}), (e_{A,0}, e_{B,3}), (e_{A,0}, e_{B,4})\}$$

```
MaxPath = [e(a,0),e(b,4),e(a,1),e(s,2)]
Bound = 4;
```

### 6.4.6 Protocol class NB+AH+CC

We introduce the following additional events:

| | |
|---|---|
| $e_{A,5}$ | send of handshake message for $B$ at $A$ |
| $e_{B,5}$ | send of handshake message for $A$ at $B$ |
| $e_{A,6}$ | receive of handshake message at $A$ |
| $e_{B,6}$ | receive of handshake message at $B$ |

and capture the handshake goal as follows:

$$\text{CR20} \quad e_{A,5} \triangle\text{-precedes } e_{B,6}$$
$$\text{CR21} \quad e_{B,5} \triangle\text{-precedes } e_{A,6}$$

We capture (H1) using the following constraints:

$$\text{CR22} \quad e_{A,2} \rightarrow e_{A,5}$$
$$\text{CR23} \quad e_{B,2} \rightarrow e_{B,5}$$

CR20 and CR21 imply by Theorem 6.1, Proposition 6.1, and Proposition 6.2 that there exist send events $e_{B,7}$ and $e_{A,7}$, respectively, at $B$ and $A$ such that:

$$\text{CR24} \quad e_{B,7} \rightarrow e_{A,5}$$
$$\text{CR25} \quad e_{A,5} \rightarrow e_{B,6}$$
$$\text{CR26} \quad e_{A,7} \rightarrow e_{B,5}$$
$$\text{CR27} \quad e_{B,5} \rightarrow e_{A,6}$$

To satisfy (H3), we stipulate the following additional constraints:

$$\text{CR28} \quad e_{A,0} \rightarrow e_{B,5}$$
$$\text{CR29} \quad e_{A,0} \rightarrow e_{B,7}$$

$(\mathcal{E}^{12}, \prec^{12})$:

$$\begin{aligned}
\mathcal{E}^{12} &= \mathcal{E}^{11} \cup \\
&\quad \{e_{A,5}, e_{A,6}, e_{A,7}, e_{B,5}, e_{B,6}, e_{B,7}\} \\
\prec^{12} &= \prec^{11} \cup \\
&\quad \{(e_{A,2}, e_{A,5}), (e_{B,2}, e_{B,5}), (e_{B,7}, e_{A,5}), (e_{A,5}, e_{B,6}), (e_{A,7}, e_{B,5}), (e_{B,5}, e_{A,6}) \\
&\quad (e_{A,0}, e_{B,5}), (e_{A,0}, e_{B,7})\}
\end{aligned}$$

$(\mathcal{E}_s^{12}, \prec_s^{12})$:

$$\begin{aligned}
\mathcal{E}_s^{12} &= \{e_{A,0}, e_{A,1}, e_{A,3}, e_{A,4}, e_{A,5}, e_{A,7}, e_{B,1}, e_{B,3}, e_{B,4}, e_{B,5}, e_{B,7}, e_{S,1}, e_{S,2}\} \\
\prec_s^{12} &= \{(e_{A,3}, e_{S,1}), (e_{S,1}, e_{A,5}), (e_{B,3}, e_{S,2}), (e_{S,2}, e_{B,5}), (e_{A,4}, e_{B,1}), (e_{B,4}, e_{A,1}), \\
&\quad (e_{B,1}, e_{S,1}), (e_{A,1}, e_{S,2}), (e_{A,0}, e_{B,1}), (e_{A,0}, e_{B,3}), (e_{A,0}, e_{B,4}), (e_{B,7}, e_{A,5}), \\
&\quad (e_{A,7}, e_{B,5}), (e_{A,0}, e_{B,5}), (e_{A,0}, e_{B,7})\}
\end{aligned}$$

```
MaxPath = [e(a,0),e(b,4),e(a,1),e(s,2),e(b,5)]
Bound = 5;
```

# Chapter 7

# Conclusions

The subtlety which underlies reasoning about authentication protocols is well-recognized in the literature. It is also recognized that both formal as well as informal methods are useful to tackle the underlying subtlety [61], [62]. Authentication logics constitute a significant class of formal methods for reasoning about protocols. This thesis lays some semantic foundations for such logics. It also contributes to reasoning about efficiency metrics for protocols. Appendix E illustrates the use of an existing informal method for protocol analysis and design due to Boyd and Mao [45]. We show how it can be heuristically used to explain flaws in several well-known protocols and to design new, improved protocols. Below we look back on the main developments of this thesis and suggest some directions for future work.

## 7.1 Summary

In Chapter 1 we review several existing authentication logics and discuss some of the motivations underlying their evolution. In Chapter 2 we stress the need for a semantic basis for authentication logics. We make our case by means of some convincing examples based on a well-known authentication logic of Gong, Needham and Yahalom; our intention is not criticize their logic but only to draw attention to the problematic nature of semantically unsupported syntactic definitions. In Chapter 3 we modify the logic of Gong, Needham and Yahalom to obtain a modified logic with the property that derivations in the logic are finite. This allows a direct automation of the modified logic using forward-chaining. In Chapter 4 we develop a model to explain some of the notions that existing logics attempt to capture, not in terms of any logical formalism but within a framework which we can appeal to on independent grounds. One of the virtues of our model is that it forces us to make explicit various assumptions that are needed to formally establish the properties which are usually associated with the above notions. In Chapter 5 we exploit the model developed earlier to help devise a new

authentication logic which is sound with respect to that model. The soundness theorem established there gives us confidence that our logic correctly models in syntactic terms the properties which we wish to capture. The conventional metalogical machinery we employ in carrying out the proof of the soundness theorem should enable comparisons of the logic with more traditional logics. We emphasize that the proposed logic is rather modest in regards to the number of features it offers for protocol analysis: it does not capture many interesting notions found in other logics. However, it stands out from these logics in a unique way—it is accompanied by a rigorous proof of soundness. It is our understanding that some notable researchers have lately expressed concern about the lack of solid foundations for authentication logics [63], [64]. We believe our work represents a positive step in this direction. Indeed, in the words of Tuttle [28],

> "...let's go back to basics and *concentrate on* [emphasis ours] meaningful models and definitions. Then let's see what new logics these definitions suggest."

In Chapter 6 we develop a general model for reasoning about the round complexity of authentication protocols. The model draws upon some existing notions of causality to build a definition of the metric number of rounds. The upshot of our definition is a key theorem that yields lower bounds on the number of rounds.

## 7.2 Future work

There are a number of directions to consider for future work. This includes modeling of the notion of recognizability using the computational model developed in Chapter 4. A preliminary attempt at this is documented in Appendix F. However, it remains to be seen how the notion of recognizability can be integrated into the logic developed in Chapter 5. It is not clear that the traditional possible worlds semantics for belief that we have adopted best fits our purposes. It would be worthwhile to find a more natural semantics for belief. An interesting problem is to investigate whether the notion of recognizability holds the key to defining a more natural semantics for belief.

Although we have used the model proposed in Chapter 6 to verify the correctness of some existing bounds on rounds from the literature, the model should also provide a means to investigate bounds for more complex protocol classes. It would also be desirable to make our model applicable to a synchronous setting. Such a move seems feasible since the definitions that we make to capture the notions related to rounds

are essentially independent of the assumption that the system is asynchronous. A theoretically stimulating direction is to provide a formal semantics for the notion of verifiable causality; this would compel us to develop a more solid foundation for the model proposed in Chapter 6.

# Bibliography


[1] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," Tech. Rep. 39, Systems Research Center, Digital Equipment Corporation, Palo Alto, California, Feb. 1989. Revised Feb. 1990.

[2] L. Gong, R. Needham, and R. Yahalom, "Reasoning about Belief in Cryptographic Protocols," in *Proc. IEEE Symposium on Security and Privacy*, (Los Alamitos, California), pp. 234–248, IEEE Computer Society Press, May 1990.

[3] L. Gong, *Cryptographic Protocols for Distributed Systems.* PhD thesis, Cambridge University, U.K., 1990.

[4] K. Gaarder and E. Snekkenes, "Applying a Formal Analysis Technique to the CCITT X.509 Strong Two-Way Authentication Protocol," *Journal of Cryptology*, vol. 3, pp. 81–98, 1991.

[5] R. Kailar and V. D. Gligor, "On Belief Evolution in Authentication Protocols," in *Proc. IEEE Computer Security Foundations Workshop IV*, (Los Alamitos, California), pp. 103–116, IEEE Computer Society Press, 1991.

[6] P. C. van Oorschot, "Extending Cryptographic Logics of Belief to Key Agreement Protocols (Extended Abstract)," in *Proc. First ACM Conference on Computer and Communications Security*, pp. 232–243, Nov. 1993.

[7] W. Mao and C. Boyd, "Towards Formal Analysis of Security Protocols," in *Proc. of Computer Security Foundations Workshop VI*, pp. 147–158, IEEE Computer Society Press, 1993.

[8] M. Abadi and M. R. Tuttle, "A Semantics for a Logic of Authentication," in *Proceedings of the Tenth ACM Symposium on Principles of Distributed Computing*, pp. 201–216, ACM Press, August 1991.

[9] P. F. Syverson and P. C. van Oorschot, "On Unifying Some Cryptographic Protocol Logics," in *Proc. IEEE Symposium on Security and Privacy*, pp. 14–28, May 1994.

[10] G. Wedel and V. Kessler, "Formal Semantics for Authentication Logics," in *Computer Security - ESORICS 96* (E. Bertino, ed.), vol. 1146 of *Lecture Notes in Computer Science*, pp. 219–241, Springer-Verlag, 1996.

[11] R. M. Needham and M. D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, vol. 21, pp. 993–999, Dec. 1978.

[12] M. Burrows, M. Abadi, and R. Needham, "A Logic of Authentication," *ACM Trans. on Computer Systems*, vol. 8, pp. 18–36, Feb. 1990.

[13] D. E. Denning and G. M. Sacco, "Timestamps in Key Distribution Protocols," *Communications of the ACM*, vol. 24, pp. 533–536, Aug. 1981.

[14] M. Burrows, M. Abadi, and R. Needham, "The Scope of a Logic of Authentication," in *Distributed Computing and Cryptography* (J. Feigenbaum, ed.), no. 2 in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, pp. 119–126, AMS & ACM Press, 1991.

[15] N. Heintze and J. D. Tygar, "Timed Models for Protocol Security," Tech. Rep. CMU-CS-92-100, Carnegie Mellon University, School of Computer Science, Pittsburgh, PA 15213, Jan. 1992.

[16] U. Engberg, "Analyzing Authentication Protocols," Tech. Rep. TR DAIMI IR-97, Aarhus University, Denmark, 1990.

[17] C. A. Boyd, "Hidden Assumptions in Cryptographic Protocols," in *Proc. IEE*, no. 6, pp. 433–436, Nov. 1990.

[18] V. D. Gligor, R. Kailar, S. Stubblebine, and L. Gong, "Logics for Cryptographic Protocols - Virtues and Limitations," in *Proc. IEEE Computer Security Foundations Workshop IV*, (Los Alamitos, California), pp. 219–226, IEEE Computer Society Press, 1991.

[19] D. M. Nessett, "A Critique of The Burrows, Abadi and Needham Logic," *ACM Operating Systems Review*, vol. 24, pp. 35–38, Apr. 1990.

[20] M. Burrows, M. Abadi, and R. Needham, "Rejoinder to Nessett," *ACM Operating Systems Review*, vol. 24, pp. 39–40, Apr. 1990.

[21] L. Gong, "Variations on the Themes of Message Freshness and Replay," in *Proceedings of the Computer Security Foundations Workshop VI*, (Los Alamitos, California), pp. 131–136, IEEE Computer Society Press, 1993.

[22] K. Gaarder and E. Snekkenes, "On The Formal Analysis of PKCS Authentication Protocols," in *Advances in Cryptology - Auscrypt'90* (J. Seberry and J. Pieprzyk, eds.), vol. 453 of *Lecture Notes in Computer Science*, pp. 106–121, Springer Verlag, 1990.

[23] R. A. Rueppel and P. C. van Oorschot, "Modern key agreement techniques," *Computer Communications*, vol. 17, pp. 458–465, July 1994.

[24] C. Boyd and W. Mao, "On a Limitation of BAN Logic," in *Advances in Cryptology - EUROCRYPT '93* (T. Helleseth, ed.), no. 765 in Lecture Notes in Computer Science, pp. 240–247, Springer Verlag, 1993.

[25] A. Mathuria, R. Safavi-Naini, and P. Nickolas, "Some Remarks on the Logic of Gong, Needham and Yahalom," in *Proceedings of the 1994 International Computer Symposium*, (National Chiao Tung University, Taiwan), pp. 303–308, December 1994.

[26] P. F. Syverson, "The Use of Logic in the Analysis of Cryptographic Protocols," in *Proc. IEEE Symposium on Security and Privacy*, pp. 156–170, June 1991.

[27] P. F. Syverson, "Knowledge, Belief, and Semantics in the Analysis of Cryptographic Protocols," *Journal of Computer Security*, vol. 1, no. 3, pp. 317–334, 1992.

[28] M. R. Tuttle, "Flaming in Franconia: Build Models, not logics." Note on the panel discussion on the Use of Formal Methods in the Analysis of Cryptographic Protocols, Computer Security Foundations Workshop V, June 1992.

[29] R. J. Anderson, "UEPS - A Second Generation Electronic Wallet," in *Computer Security - ESORICS 92* (Y. Deswarte, G. Eizenberg, and J.-J. Quisquater, eds.), pp. 411–418, Springer-Verlag, 1992.

[30] R. Rivest, A. Shamir, and L. Adleman, "A Method for Obtaining Digital Signatures and Public-key Cryptosystems," *Communications of the ACM*, vol. 21, pp. 120–126, Feb. 1978.

[31] "National Bureau of Standards. Data Encryption Standard." Federal Information Processing Standards, Pub. 46, Washington, D.C., Jan. 1977.

[32] A. Mathuria, R. Safavi-Naini, and P. Nickolas, "On the Automation of GNY Logic," in *Proceedings of the Eighteenth Australasian Computer Science Conference (ACSC '95)* (R. Kotagiri, ed.), vol. 17:(1) of *Australian Computer Science Communications*, pp. 370–379, February 1995.

[33] E. A. Campbell and R. Safavi-Naini, "On Automating The BAN Logic of Authentication," in *Proc. 15th Australian Computer Science Conference (ACSC-15)*, 1992.

[34] R. C. Hauser and E. S. Lee, "Verification and Modelling of Authentication Protocols," in *Computer Security - ESORICS 92* (Y. Deswarte, G. Eizenberg, and J.-J. Quisquater, eds.), no. 648 in Lecture Notes in Computer Science, pp. 141–154, Springer-Verlag, 1992.

[35] A. Mathuria, "Automating BAN Logic," Master's thesis, University of Wollongong, Department of Computer Science, 1994.

[36] L. Gong, "Lower Bounds on Messages and Rounds for Network Authentication Protocols," in *Proc. First ACM Conference on Computer and Communications Security*, pp. 26–37, Nov. 1993.

[37] L. Gong, "Efficient network authentication protocols: Lower bounds and optimal implementations," Tech. Rep. 94-15, SRI Computer Science Laboratory, CA 94025, U. S. A, Oct. 1994.

[38] L. Gong, "Efficient network authentication protocols: lower bounds and optimal implementations," *Distributed Computing*, vol. 9, no. 3, pp. 131–145, 1995.

[39] A. Mathuria, R. Safavi-Naini, and P. Nickolas, "Causality, partial orders and lower bounds on rounds for a class of authentication protocols," in *Proceedings of the Twentieth Australasian Computer Science Conference (ACSC'97)* (M. Patel, ed.), vol. 19:(1) of *Australian Computer Science Communications*, pp. 27–36, February 1997.

[40] R. Yahalom, "Optimality of Asynchronous Two–Party Secure Data-Exchange Protocols," *Journal of Computer Security*, vol. 2, no. 2–3, pp. 191–209, 1993.

[41] R. Yahalom, "Optimality of Multi-Domain Protocols," in *Proc. First ACM Conference on Computer and Communications Security*, pp. 38–48, Nov. 1993.

[42] L. Lamport, "Time, Clocks, and the Ordering of Events in a Distributed System," *Communications of the ACM*, vol. 21, pp. 558–565, July 1978.

[43] G. Coulouris, J. Dollimore, and T. Kindberg, *Distributed Systems Concepts and Design*, ch. 10. Reading, Massachusetts: Addison Wesley, second ed., 1994.

[44] D. F. Stanat and D. F. McAllister, *Discrete Mathematics in Computer Science.* Englewood Cliffs, New Jersey: Prentice-Hall, Inc., 1977.

[45] C. Boyd and W. Mao, "Designing Secure Key Exchange Protocols," in *Computer Security – ESORICS 94* (D. Gollmann, ed.), vol. 875 of *Lecture Notes in Computer Science*, pp. 93–105, Springer-Verlag, 1994.

[46] C. Boyd and A. Mathuria, "Systematic design of key establishment protocols based on one-way functions," *IEE Proceedings - Computers and Digital Techniques*, vol. 144, pp. 93–99, Mar. 1997.

[47] A. Mathuria, "Addressing weaknesses in two cryptographic protocols of Bull, Gong and Sollins," *Electronics Letters*, vol. 31, pp. 1543–1544, Aug. 1995.

[48] L. Gong, "Using One-Way Functions for Authentication," *Computer Communication Review*, vol. 19, pp. 8–11, Oct. 1989.

[49] R. Molva, G. Tsudik, E. V. Herreweghen, and S. Zatti, "KryptoKnight Authentication and Key Distribution System," in *Computer Security - ESORICS 92* (Y. Deswarte, G. Eizenberg, and J.-J. Quisquater, eds.), vol. 648 of *Lecture Notes in Computer Science*, pp. 155–174, Springer-Verlag, 1992.

[50] R. Bird, I. Gopal, A. Herzberg, P. Janson, S. Kutten, R. Molva, and M. Yung, "The KryptoKnight Family of Light-Weight Protocols for Authentication and Key Distribution," *IEEE/ACM Transactions on Networking*, vol. 3, pp. 31–41, Feb. 1995.

[51] R. C. Merkle, "A Fast Software One-Way Hash Function," *Journal of Cryptology*, vol. 3, pp. 43–58, Sept. 1990.

[52] R. Anderson, "The Classification of Hash Functions," in *Fourth IMA Conference on Coding and Cryptography*, pp. 83–93, 1994.

[53] T. A. Berson, L. Gong, and T. M. A. Lomas, "Secure, Keyed, and Collisionful Hash Functions." Included in technical report SRI-CSL-94-08, Computer Science Laboratory, SRI International, Menlo Park, California, May 1994.

[54] B. Preneel, R. Govaerts, and J. Vandewalle, "Hash Functions for Information Authentication," in *Proceedings of the 6th Annual European Computer Conference (CompEuro'92) - Computer Systems and Software Engineering* (P. Dewilde and J. Vandewalle, eds.), pp. 475–480, IEEE Computer Society Press, 1992.

[55] R. L. Rivest, "The MD5 message-digest algorithm." Request for Comments (RFC) 1320, Internet Activities Board, Internet Privacy Task Force, April 1992.

[56] C. Boyd and W. Mao, "Design and Analysis of Key Exchange Protocols via Secure Channel Identification," in *Advances in Cryptology - ASIACRYPT '94* (J. Pieprzyk and R. Safavi-Naini, eds.), vol. 917 of *Lecture Notes in Computer Science*, pp. 171–181, Springer-Verlag, 1995.

[57] D. Denning, *Cryptography and Data Security.* Reading, Mass.: Addison-Wesley, 1982.

[58] M. Abadi and R. Needham, "Prudent Engineering Practice for Cryptographic Protocols," in *Proceedings of the 1994 IEEE Symposium on Security and Privacy*, (Los Alamitos, California), pp. 122–136, IEEE Computer Society Press, may 1994.

[59] J. A. Bull, L. Gong, and K. R. Sollins, "Towards Security in an Open Systems Federation," in *Computer Security - ESORICS 92* (Y. Deswarte, G. Eizenberg, and J.-J. Quisquater, eds.), vol. 648 of *Lecture Notes in Computer Science*, pp. 3–20, Springer-Verlag, 1992.

[60] C. Mitchell, "Limitations of Challenge-Response Entity Authentication," *Electronic Letters*, vol. 25, pp. 1195–1196, August 1989.

[61] M. Abadi and R. Needham, "Prudent engineering practice for cryptographic protocols," *IEEE Transactions on Software Engineering*, vol. 22, pp. 6–15, Jan. 1996.

[62] R. Anderson and R. Needham, "Programming Satan's Computer," in *Computer Science Today: Recent Trends and Developments* (J. van Leeuwen, ed.), vol. 1000 of *Lecture Notes in Computer Science*, pp. 426–440, Springer-Verlag, 1995.

[63] L. Paulson and R. Needham, "Authentication Logics: New Theory and Implementations." Computer Laboratory, University of Cambridge, EPSRC research proposal GR/K77051. http://www.cl.cam.ac.uk/users/lcp/Auth.

[64] L. Paulson, "Proving Properties of Security Protocols by Induction," Tech. Rep. 409, University of Cambridge, Computer Laboratory, Dec. 1996.

[65] I. Bratko, *Prolog programming for artificial intelligence.* Addison-Wesley Publishers Ltd., Second ed., 1990.

# Appendix A

# BAN logic rules

## A.1 Message-meaning rules

$$\frac{P \models Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P \models Q \mid\sim X}$$

$$\frac{P \models \overset{K}{\mapsto} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P \models Q \mid\sim X}$$

$$\frac{P \models Q \overset{Y}{\rightleftharpoons} P,\ P \triangleleft \langle X \rangle_Y}{P \models Q \mid\sim X}$$

## A.2 Nonce-verification rule

$$\frac{P \models \sharp(X),\ P \models Q \mid\sim X}{P \models Q \models X}$$

## A.3 Jurisdiction rule

$$\frac{P \models Q \Rightarrow X,\ P \models Q \models X}{P \models X}$$

## A.4 Belief rules

$$\frac{P \models X,\ P \models Y}{P \models (X,Y)} \qquad \frac{P \models (X,Y)}{P \models X} \qquad \frac{P \models Q \models (X,Y)}{P \models Q \models X}$$

## A.5 Utterance rule

$$\frac{P \models Q \mid\sim (X,Y)}{P \models Q \mid\sim X}$$

## A.6 Message seeing rules

$$\frac{P \triangleleft (X, Y)}{P \triangleleft X}$$

$$\frac{P \triangleleft \langle X \rangle_Y}{P \triangleleft X}$$

$$\frac{P \models Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}$$

$$\frac{P \models \overset{K}{\rightarrow} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}$$

$$\frac{P \models \overset{K}{\rightarrow} Q,\ P \triangleleft \{X\}_{K^{-1}}}{P \triangleleft X}$$

## A.7 Freshness rule

$$\frac{P \models \sharp(X)}{P \models \sharp(X, Y)}$$

## A.8 Shared key and shared secret rules

$$\frac{P \models R \overset{K}{\leftrightarrow} R'}{P \models R' \overset{K}{\leftrightarrow} R} \qquad \frac{P \models Q \models R \overset{K}{\leftrightarrow} R'}{P \models Q \models R' \overset{K}{\leftrightarrow} R}$$

$$\frac{P \models R \overset{K}{\rightleftharpoons} R'}{P \models R' \overset{K}{\rightleftharpoons} R} \qquad \frac{P \models Q \models R \overset{K}{\rightleftharpoons} R'}{P \models Q \models R' \overset{K}{\rightleftharpoons} R}$$

## A.9 Supplementary rules

$$\frac{P \models R \mid\sim Q \overset{K}{\leftrightarrow} P,\ P \triangleleft \{X\}_K}{P \triangleleft X}$$

$$\frac{P \models Q \mid\sim H(X),\ P \triangleleft X}{P \models Q \mid\sim X}$$

$$\frac{P \models Q \mid\sim H(X_1, \ldots, X_k),\ P \triangleleft X_1, \ldots, P \triangleleft X_k}{P \models Q \mid\sim (X_1, \ldots, X_k)}$$

# Appendix B

# GNY logic rules

## B.1 Rationality rule

If $\frac{C_1}{C_2}$ is a rule, then for any principal $P$, so is $\frac{P \models C_1}{P \models C_2}$.

## B.2 Being-told rules

T1 $\dfrac{P \triangleleft *X}{P \triangleleft X}$

T2 $\dfrac{P \triangleleft (X,Y)}{P \triangleleft X}$

T3 $\dfrac{P \triangleleft \{X\}_K,\ P \ni K}{P \triangleleft X}$

T4 $\dfrac{P \triangleleft \{X\}_{+K},\ P \ni -K}{P \triangleleft X}$

T5 $\dfrac{P \triangleleft F(X,Y),\ P \ni X}{P \triangleleft Y}$

T6 $\dfrac{P \triangleleft \{X\}_{-K},\ P \ni +K}{P \triangleleft X}$

## B.3 Possession rules

P1 $\dfrac{P \triangleleft X}{P \ni X}$

P2 $\dfrac{P \ni X,\ P \ni Y}{P \ni (X,Y),\ P \ni F(X,Y)}$

P3 $\dfrac{P \ni (X,Y)}{P \ni X}$

P4 $$\frac{P \ni X}{P \ni H(X)}$$

P5 $$\frac{P \ni F(X,Y),\ P \ni X}{P \ni Y}$$

P6 $$\frac{P \ni K,\ P \ni X}{P \ni \{X\}_K,\ P \ni \{X\}_K^{-1}}$$

P7 $$\frac{P \ni +K,\ P \ni X}{P \ni \{X\}_{+K}}$$

P8 $$\frac{P \ni -K,\ P \ni X}{P \ni \{X\}_{-K}}$$

## B.4 Freshness rules

F1 $$\frac{P \models \sharp(X)}{P \models \sharp(X,Y),\ P \models \sharp(F(X))}$$

F2 $$\frac{P \models \sharp(X),\ P \ni K}{P \models \sharp(\{X\}_K),\ P \models \sharp(\{X\}_K^{-1})}$$

F3 $$\frac{P \models \sharp(X),\ P \ni +K}{P \models \sharp(\{X\}_{+K})}$$

F4 $$\frac{P \models \sharp(X),\ P \ni -K}{P \models \sharp(\{X\}_{-K})}$$

F5 $$\frac{P \models \sharp(+K)}{P \models \sharp(-K)}$$

F6 $$\frac{P \models \sharp(-K)}{P \models \sharp(+K)}$$

F7 $$\frac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K}{P \models \sharp(\{X\}_K),\ P \models \sharp(\{X\}_K^{-1})}$$

F8 $$\frac{P \models \phi(X),\ P \models \sharp(+K),\ P \ni +K}{P \models \sharp(\{X\}_{+K})}$$

F9 $$\frac{P \models \phi(X),\ P \models \sharp(-K),\ P \ni -K}{P \models \sharp(\{X\}_{-K})}$$

F10 $$\frac{P \models \sharp(X),\ P \ni X}{P \models \sharp(H(X))}$$

F11 $$\frac{P \models \sharp(H(X)),\ P \ni H(X)}{P \models \sharp(X)}$$

## B.5 Recognizability rules

R1 $$\frac{P \models \phi(X)}{P \models \phi(X,Y),\ P \models \phi(F(X))}$$

R2 $$\frac{P \models \phi(X),\ P \ni K}{P \models \phi(\{X\}_K),\ P \models \phi(\{X\}_K^{-1})}$$

R3 $$\frac{P \models \phi(X),\ P \ni +K}{P \models \phi(\{X\}_{+K})}$$

R4 $$\frac{P \models \phi(X),\ P \ni -K}{P \models \phi(\{X\}_{-K})}$$

R5 $$\frac{P \models \phi(X),\ P \ni X}{P \models \phi(H(X))}$$

R6 $$\frac{P \ni H(X)}{P \models \phi(X)}$$

## B.6 Message interpretation rules

I1 $$\frac{P \triangleleft *\{X\}_K,\ P \ni K,\ P \models P \overset{K}{\leftrightarrows} Q,\ P \models \phi(X),\ P \models \sharp(X,K)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K,\ P \models Q \ni K}$$

I2 $$\frac{\begin{array}{c}P \triangleleft *\{X,<S>\}_{+K},\ P \ni (-K,S),\ P \models^{\pm K}\!\!\mapsto P,\\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X,S),\ P \models \sharp(X,S,+K)\end{array}}{P \models Q \mid\sim (X,<S>),\ P \models Q \mid\sim \{X,<S>\}_{+K},\ P \models Q \ni +K}$$

I3 $$\frac{P \triangleleft *H(X,<S>),\ P \ni (X,S),\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \sharp(X,S)}{P \models Q \mid\sim (X,<S>),\ P \models Q \mid\sim H(X,<S>)}$$

I4 $$\frac{P \triangleleft \{X\}_{-K},\ P \ni +K,\ P \models^{\pm K}\!\!\mapsto Q,\ P \models \phi(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_{-K}}$$

I5 $$\frac{P \triangleleft \{X\}_{-K},\ P \ni +K,\ P \models^{\pm K}\!\!\mapsto Q,\ P \models \phi(X),\ P \models \sharp(X,+K)}{P \models Q \ni (-K,X)}$$

I6 $$\frac{P \models Q \mid\sim X,\ P \models \sharp(X)}{P \models Q \ni X}$$

I7 $$\frac{P \models Q \mid\sim (X, Y)}{P \models Q \mid\sim X}$$

## B.7 Jurisdiction rules

J1 $$\frac{P \models Q \models\!\!\!\Rightarrow C,\ P \models Q \models C}{P \models C}$$

J2 $$\frac{P \models Q \models\!\!\!\Rightarrow Q \models *,\ P \models Q \mid\sim (X \rightsquigarrow C),\ P \models \sharp(X)}{P \models Q \models C}$$

J3 $$\frac{P \models Q \models\!\!\!\Rightarrow Q \models *,\ P \models Q \models Q \models C}{P \models Q \models C}$$

## B.8 Never-originated-here rules

I1′ $$\frac{P \triangleleft \{X\}_K,\ P \ni K,\ P \models P \overset{K}{\leftrightarrows} Q,\ P \models \phi(X),\ P \models \otimes(P)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K}$$

I2′ $$\frac{\begin{array}{c} P \triangleleft \{X, <S>\}_{+K},\ P \ni (S, -K),\ P \models^{+K}\!\!\!\mapsto P, \\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X, S),\ P \models \otimes(P) \end{array}}{P \models Q \mid\sim (X, <S>),\ P \models Q \mid\sim \{X, <S>\}_{+K}}$$

I3′ $$\frac{P \triangleleft H(X, <S>),\ P \ni (X, S),\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X, S),\ P \models \otimes(P)}{P \models Q \mid\sim (X, <S>),\ P \models Q \mid\sim H(X, <S>)}$$

# Appendix C

# Modified GNY logic

## C.1 Being-told rules

T1 $\dfrac{P \triangleleft *X}{P \triangleleft X}$

T2 $\dfrac{P \triangleleft (X,Y)}{P \triangleleft X}$

T3 $\dfrac{P \triangleleft \{X\}_K,\ P \ni K}{P \triangleleft X}$

T4 $\dfrac{P \triangleleft \{X\}_{+K},\ P \ni -K}{P \triangleleft X}$

T5 $\dfrac{P \triangleleft F(X,Y),\ P \ni X}{P \triangleleft Y}$

T6 $\dfrac{P \triangleleft \{X\}_{-K},\ P \ni +K}{P \triangleleft X}$

T7 $\dfrac{P \triangleleft X \rightsquigarrow C}{P \triangleleft X}$

## C.2 Possession rules

P1 $\dfrac{P \triangleleft X}{P \ni X}$

P3 $\dfrac{P \ni (X,Y)}{P \ni X}$

P5 $\dfrac{P \ni F(X,Y),\ P \ni X}{P \ni Y}$

## C.3 Freshness rules

F1′ $$\frac{P \models \sharp(X),\ P \ni (X,Y)}{P \models \sharp(X,Y)}$$

F1″ $$\frac{P \models \sharp(X),\ P \ni F(X)}{P \models \sharp(F(X))}$$

F2′ $$\frac{P \models \sharp(X),\ P \ni K,\ P \ni \{X\}_K}{P \models \sharp(\{X\}_K)}$$

F2″ $$\frac{P \models \sharp(X),\ P \ni K,\ P \ni \{X\}_K^{-1}}{P \models \sharp(\{X\}_K^{-1})}$$

F3′ $$\frac{P \models \sharp(X),\ P \ni +K,\ P \ni \{X\}_{+K}}{P \models \sharp(\{X\}_{+K})}$$

F4′ $$\frac{P \models \sharp(X),\ P \ni -K,\ P \ni \{X\}_{-K}}{P \models \sharp(\{X\}_{-K})}$$

F5′ $$\frac{P \models \sharp(+K),\ P \ni -K}{P \models \sharp(-K)}$$

F6′ $$\frac{P \models \sharp(-K),\ P \ni +K}{P \models \sharp(+K)}$$

F7′ $$\frac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K,\ P \ni \{X\}_K}{P \models \sharp(\{X\}_K)}$$

F7″ $$\frac{P \models \phi(X),\ P \models \sharp(K),\ P \ni K,\ P \ni \{X\}_K^{-1}}{P \models \sharp(\{X\}_K^{-1})}$$

F8′ $$\frac{P \models \phi(X),\ P \models \sharp(+K),\ P \ni +K,\ P \ni \{X\}_{+K}}{P \models \sharp(\{X\}_{+K})}$$

F9′ $$\frac{P \models \phi(X),\ P \models \sharp(-K),\ P \ni -K,\ P \ni \{X\}_{-K}}{P \models \sharp(\{X\}_{-K})}$$

F10′ $$\frac{P \models \sharp(X),\ P \ni X,\ P \ni H(X)}{P \models \sharp(H(X))}$$

F11′ $$\frac{P \models \sharp(H(X)),\ P \ni H(X),\ P \ni X}{P \models \sharp(X)}$$

## C.4 Recognizability rules

R1′ $$\frac{P \models \phi(X),\ P \ni (X,Y)}{P \models \phi(X,Y)}$$

R1″ $$\frac{P \models \phi(X),\ P \ni F(X)}{P \models \phi(F(X))}$$

R2′ $$\frac{P \models \phi(X),\ P \ni K,\ P \ni \{X\}_K}{P \models \phi(\{X\}_K)}$$

R2″ $$\frac{P \models \phi(X),\ P \ni K,\ P \ni \{X\}_K^{-1}}{P \models \phi(\{X\}_K^{-1})}$$

R3′ $$\frac{P \models \phi(X),\ P \ni +K,\ P \ni \{X\}_{+K}}{P \models \phi(\{X\}_{+K})}$$

R4′ $$\frac{P \models \phi(X),\ P \ni -K,\ P \ni \{X\}_{-K}}{P \models \phi(\{X\}_{-K})}$$

R5′ $$\frac{P \models \phi(X),\ P \ni X,\ P \ni H(X)}{P \models \phi(H(X))}$$

## C.5 Message interpretation rules

I1 $$\frac{P \triangleleft *\{X\}_K \rightsquigarrow C,\ P \ni K,\ P \models P \overset{K}{\leftrightarrows} Q,\ P \models \phi(X),\ P \models \sharp(X,K)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K \rightsquigarrow C,\ P \models Q \ni K}$$

I2 $$\frac{\begin{array}{c}P \triangleleft *\{X,<S>\}_{+K} \rightsquigarrow C,\ P \ni (-K,S),\ P \models^{+K}\!\!\mapsto P,\\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X,S),\ P \models \sharp(X,S,+K)\end{array}}{P \models Q \mid\sim (X,<S>),\ P \models Q \mid\sim \{X,<S>\}_{+K} \rightsquigarrow C,\ P \models Q \ni +K}$$

I3 $$\frac{P \triangleleft *H(X,<S>) \rightsquigarrow C,\ P \ni (X,S),\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \sharp(X,S)}{P \models Q \mid\sim (X,<S>),\ P \models Q \mid\sim H(X,<S>) \rightsquigarrow C}$$

I4 $$\frac{P \triangleleft \{X\}_{-K} \rightsquigarrow C,\ P \ni +K,\ P \models^{+K}\!\!\mapsto Q,\ P \models \phi(X)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_{-K} \rightsquigarrow C}$$

I5 $$\frac{P \triangleleft \{X\}_{-K},\ P \ni +K,\ P \models^{+K}\!\!\mapsto Q,\ P \models \phi(X),\ P \models \sharp(X,+K)}{P \models Q \ni (-K,X)}$$

I6 $$\frac{P \models Q \mid\sim X,\ P \models \sharp(X)}{P \models Q \ni X}$$

I7 $$\frac{P \models Q \mid\sim (X, Y)}{P \models Q \mid\sim X}$$

I8 $$\frac{P \models Q \mid\sim X \rightsquigarrow C}{P \models Q \mid\sim X}$$

I9 $$\frac{P \models Q \mid\sim X \rightsquigarrow (C, C')}{P \models Q \mid\sim X \rightsquigarrow C}$$

## C.6 Jurisdiction rules

J1 $$\frac{P \models Q \Mapsto C,\ P \models Q \models C}{P \models C}$$

J2 $$\frac{P \models Q \Mapsto Q \models *,\ P \models Q \mid\sim (X \rightsquigarrow C),\ P \models \sharp(X)}{P \models Q \models C}$$

J3 $$\frac{P \models Q \Mapsto Q \models *,\ P \models Q \models Q \models C}{P \models Q \models C}$$

## C.7 Never-Originated-Here rules

I1′ $$\frac{P \triangleleft \{X\}_K \rightsquigarrow C,\ P \ni K,\ P \models P \overset{K}{\leftrightarrows} Q,\ P \models \phi(X),\ P \models \otimes(P)}{P \models Q \mid\sim X,\ P \models Q \mid\sim \{X\}_K \rightsquigarrow C}$$

I2′ $$\frac{\begin{array}{c} P \triangleleft \{X, <S>\}_{+K} \rightsquigarrow C,\ P \ni (S, -K),\ P \models \overset{+K}{\leftrightarrows} P, \\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X, S),\ P \models \otimes(P) \end{array}}{P \models Q \mid\sim (X, <S>),\ P \models Q \mid\sim \{X, <S>\}_{+K} \rightsquigarrow C}$$

I3′ $$\frac{P \triangleleft H(X, <S>) \rightsquigarrow C,\ P \ni (X, S),\ P \models P \overset{S}{\leftrightarrows} Q,\ P \models \phi(X, S),\ P \models \otimes(P)}{P \models Q \mid\sim (X, <S>),\ P \models Q \mid\sim H(X, <S>) \rightsquigarrow C}$$

# Appendix D

## Path finding program

The following Prolog procedure for path finding in DAGs is adopted from the text by Bratko [65] with slight simplifications. It employs a brute force technique to determine maximal paths, and is thus highly inefficient. We nonetheless use it for the sake of simplicity.

```
% lbr(Digraph, MaxPath, Bound):
% MaxPath is the longest path between
% any pair of vertices in Digraph

lbr(Digraph, MaxPath, Bound) :-
    path(_, _, Digraph, MaxPath, Bound),
    not((path(_, _, Digraph, _, Cost),
                    Cost > Bound)).

% path(A, Z, Digraph, Path, Cost):
% Digraph is represented as
% digraph(Nodes, Edges), where
% Nodes is a list of vertices and
% Edges is a list of edges in Digraph
% ;
% Path is an acyclic path with
% cost Cost from A to Z in Digraph
% p(X, Y) means there is an edge
% from X to Y in Digraph


path(A, Z, Digraph, Path, Cost) :-
```

```
    path1(A, [Z], 1, Digraph, Path, Cost).

path1(A, [A|Path1], Cost1, _,
                  [A|Path1], Cost1).
path1(A, [Y|Path1], Cost1,
    digraph(Nodes, Edges), Path, Cost) :-
    member(p(X, Y), Edges),
    Cost2 is Cost1 + 1,
    path1(A, [X, Y|Path1], Cost2,
      digraph(Nodes, Edges), Path, Cost).
```

# Appendix E

# An informal approach to the analysis and design of some key exchange protocols

In this appendix, we investigate the security of several existing key exchange protocols using a methodology proposed by Boyd and Mao [45]. The main idea behind this methodology is to view the security of key exchange protocols in terms of two design principles based on confidentiality and authenticity properties. The purpose of this appendix is to demonstrate the effectiveness of the above view by means of case studies of some published protocols. Specifically, we will analyze several notable key exchange protocols from the literature that are based on one-way functions. The analyses we carry out provide valuable insight into the working of the protocols and reveal security weaknesses in some of the protocols. Alternative protocols will be devised that can not only be shown to be secure in a specific sense, but which are also simple and elegant when compared with the protocols analyzed.

(The contents of this appendix are based on a recent work co-authored with Colin Boyd [46], and an earlier work by the author [47]. Colin Boyd provided an unpublished manuscript to the author, which formed a substantial basis for the joint work with the author.)

## E.1 Introduction

Key exchange protocols involve an exchange of messages between two or more users with the aim of establishing a shared key among the users. Such protocols employ cryptographic functions to provide confidentiality and authenticity of the distributed keys. A variety of such functions are available in practice, and it is important to select them judiciously while designing protocols. Although the majority of key exchange protocols found in the literature use either symmetric cryptosystems or public key cryptosystems, such protocols can equally be designed using one-way hash functions. The idea of using one-way hash functions as a basis for key exchange protocols appears

to be due to Gong [48]. It has also been adopted by IBM in their KryptoKnight authentication and key distribution system [49], [50].

A one-way hash function $f$ can be characterized as follows (cf., e.g., Merkle [51]): (1) Given $x$ it is easy to calculate the hash value $f(x)$; and (2) Given a hash value $y$, it is computationally infeasible to find a value $x$ such that $f(x) = y$. Moreover, the function produces a fixed length output, but allows an input value of arbitrary length.

As noted by Anderson [52] and by Berson *et al.* [53], the above characterization of a one-way hash function is not adequate for the security of key exchange protocols of the type suggested by Gong and other similar protocols found in the literature. In particular, such protocols make use of a secret value in the input to the function, in a keyed manner, so there are additional constraints governing the use of the function that do not follow from the above definition, and which must therefore be made explicit. The desired functions are commonly labeled as *keyed hash functions* or *message authentication codes (MACs)* [54]. The exact properties required of a keyed hash function may well be application specific; however, for the protocols we are concerned with here it appears suitable to assume the properties of a *Secure Keyed One-Way Hash Function* (SKOWHF) defined by Berson *et al.* [53]. For convenience we recall their definition below.

A function $g()$ that maps a key $k$ and a second bit string $x$ to a string of a fixed length is a SKOWHF if it satisfies five additional properties:

1. Given $k$ and $x$, it is easy to compute $g(k, x)$;
2. Given $k$ and $g(k, x)$, it is hard to compute $x$;
3. Given $k$ it is hard to find two values $x$ and $y$ such that $g(k, x) = g(k, y)$, but $x \neq y$;
4. Given (possibly many) pairs $x$ and $g(k, x)$, it is hard to compute $k$;
5. Without knowledge of $k$, it is hard to compute $g(k, x)$ for any $x$.

We also assume that the mapping from input to output has the property that it is impossible to predict any portion of the output, other than by computing the function.

It is possible to construct keyed hash functions using conventional unkeyed hash functions such as MD5 [55]. There are some potential advantages of using one-way hash functions instead of conventional cryptosystems in designing key exchange protocols. Namely, that hash function implementations may have less export restrictions than conventional cryptosystems and may also be faster as compared to such cryptosystems.

## E.2 Channels for secure key exchange

We begin by briefly reviewing the methodology due to Boyd and Mao [45].

Cryptographic transformations can broadly be viewed to provide the following two primitive security services:

- **Confidentiality** of a message guarantees that only the authorized users will be able to read it.
- **Authenticity** of a message guarantees that only an authorized user could have created it.

The authorized users here are defined by their possession of the required cryptographic keys. The above two properties form the basis for the notion of abstract channels of confidentiality or authentication that may be used to characterize a secure key exchange protocol. The notation

$$S \xrightarrow{c} A : m$$

denotes that $m$ is sent by $S$ over a confidentiality channel to $A$. It implies $S$ knows no one except $A$ could possibly read $m$. The notation

$$A \xleftarrow{a} S : m$$

denotes that $m$ is received by $A$ over an authentication channel from $S$. It implies $A$ knows no one except $S$ could have possibly sent $m$. The above notations differ fundamentally from the conventional notation

$$S \rightarrow A : m$$

which only indicates that $m$ is meant to be received by $A$ supposedly from $S$. It does not imply that $m$ remains confidential to $A$ or that $S$ has actually sent $m$.

The basic goal of a key exchange protocol is to establish a shared key between two or more users for a subsequent session. We recall below two principles for secure key exchange (cf. Boyd and Mao [45]):

**Key confidentiality** The key must not be divulged to any unauthorized user. In other words, there must exist a *confidentiality channel* from the generator of the key to each recipient of the key.

**Key authenticity** Each recipient of the key must be sure that the key comes from an authorized user and is a new key for use with the stated users. In other words, there must exist an *authentication channel* from the generator of the key to each recipient of the key.

The first principle suggests that the new shared key is all that needs to be sent along a confidentiality channel from the key originator to a key recipient. The second principle suggests that this key must also be sent along an authentication channel from the key originator to the recipient together with a freshness identifier and the names of each recipient of the key. Typically, the freshness identifier used is an unpredictable nonce previously sent by the recipient.

It is easy to show that adherence to the above two principles suffices to guarantee the security of the resulting protocol in the following sense [45]: The key recipients know that the key must have newly originated from an authorized user and they also know who else this key is shared with. This security guarantee is demonstrable in a simple manner, without appealing to specific attacks.

In practice, a variety of concrete protocols may be designed by defining the required confidentiality and authentication channels in various ways using the available cryptographic functions. On the other hand, existing protocols can also be analyzed by investigating how these channels are possibly realized in the protocols, even when they might not have been specifically identified by the authors of the protocols. For a sample application of such an approach on several existing protocols employing conventional cryptosystems, cf. Boyd and Mao [56].

In the protocols considered in the following sections we shall focus only on key exchange. Some of the existing protocols we analyze using the above approach appear to include an additional feature that allows users to mutually confirm their receipt of the session key. However, as explained later in the appendix this feature lies outside the scope of the analysis approach. Therefore, we do not attempt to address key confirmation while using the analysis approach.

For the sake of uniformity in presenting the protocols below, we make slight adjustments to the original notation used by the protocol authors. The notation ',' usually denotes concatenation. Following standard practice, we extend the notation $f(k, x)$ and write $f(k, x_1, x_2, \ldots, x_n)$ to mean that the second argument of $f$ is the concatenation of $x_1, x_2, \ldots, x_n$.

## E.3 Gong's Protocols

The first protocol suggested by Gong [48] is both novel and ingenious, and appears to represent the original idea of using a one-way function as the basis for the security of a key exchange protocol. The scenario is a typical one for such protocols; a server $S$ is trusted by a pair of users to distribute a session key for use in a subsequent session between the users. The server initially shares a secret $P_U$ with each user $U$. The messages exchanged in a successful run of the protocol between $A$ and $B$ are as follows [48, p. 9]:

1. $A \rightarrow B: \quad A, B, n_A$
2. $B \rightarrow S: \quad A, B, n_A, n_B$
3. $S \rightarrow B: \quad n_S, f(P_B, n_S, n_B, A) \oplus (k, h_A, h_B), g(P_B, k, h_A, h_B)$
4. $B \rightarrow A: \quad n_S, h_B$
5. $A \rightarrow B: \quad h_A$

Here $f$ and $g$ are publicly known keyed one-way (hash) functions. The values $n_A$ and $n_B$ are random values chosen for a one-time use (nonces) by $A$ and $B$ respectively. If, for example, $B$ receives a message containing $n_B$, then $B$ can be sure that the message is new. The value $n_S$ is similarly a nonce chosen by $S$, but as we shall explain below, it is for the purpose of confidentiality and not authentication. The values $k$ (the shared session key), $h_A$, and $h_B$ are extracted using the following equation:

$$(k, h_A, h_B) = f(P_A, n_S, n_A, B)$$

Here it is assumed that the procedure for extracting the fields $k$, $h_A$, and $h_B$ from the value computed as $f(P_A, n_S, n_A, B)$ is known in advance.

It is immediately apparent from the above equation that the protocol is highly asymmetrical with respect to $A$ and $B$. Both $S$ and $A$ contribute to the value of the session key via $n_S$ and $n_A$, respectively, but $B$ has no influence on it. (Although $B$'s name appears in the above equation, it remains fixed in every run.)

We now isolate the confidentiality and authentication channels used to deliver $k$ to $A$ and $B$ respectively.

**Channels from $S$ to $A$** We first note that $k$ is generated by $A$ and $S$ jointly using their shared secret $P_A$. We can regard the value $n_S$ relayed by $B$ to $A$ (from $S$) as analogous to $k$ encrypted with $P_A$. We draw this analogy essentially by observing that

$P_A$ is a secret value that is required to recover $k$ using $n_S$. Thus we may consider the confidentiality channel from $S$ to $A$:

$$S \xrightarrow{c} A : k$$

to be defined as

$$B \rightarrow A : n_S$$

where $f(P_A, n_S, \ldots) = (k, \ldots)$. Notice that the channel definition is quite restrictive in the choice of the session key. The server $S$ cannot choose the key value independently of $A$. Furthermore, this value is the result of an application of a one-way function. This implies that the key cannot be chosen to have specific structure.

Authentication of $k$ to $A$ is provided by the value $h_B$ relayed by $B$ to $A$. The properties we assumed of $f$ imply that it is feasible to derive the value $h_B$ only with the knowledge of $P_A$. Furthermore, since only $A$ and $S$ share $P_A$, $A$ may be sure that $S$ must have originally sent $h_B$, as $A$ itself does not send it in the protocol. Additionally, $A$ may be sure that $k$ and $h_B$ must be new since both are obtained as a function of $n_A$. Thus we may consider the authentication channel from $S$ to $A$:

$$A \xleftarrow{a} S : k, n_A, B$$

to be defined as

$$B \rightarrow A : h_B$$

where $f(P_A, \ldots, n_A, B) = (k, \ldots, h_B)$.

It is now apparent that the confidentiality and authentication functions are tied together, and this appears to make the analysis complex.

**Channels from $S$ to $B$** The authentication and confidentiality channels from $S$ to $B$ are rather different.

We may consider the confidentiality channel from $S$ to $B$:

$$S \xrightarrow{c} B : k$$

to be defined as

$$S \rightarrow B : f(P_B, n_S, \ldots) \oplus k$$

Since $n_S$ is randomly chosen each time by $S$, we can consider $f(P_B, n_S, \ldots)$ to be essentially random. Moreover, it is infeasible to form $f(P_B, \ldots)$ without knowledge of $P_B$. So the confidentiality channel can be simply viewed as analogous to the Vernam cipher with a non-repeating random ciphering key, known as the *one-time pad* (cf. [57]).

We may consider the authentication channel from $S$ to $B$:

$$B \xleftarrow{a} S : k, n_B, A$$

to be defined as

$$S \to B : f(P_B, \ldots, n_B, A) \oplus (k, h_A, h_B), g(P_B, k, h_A, h_B)$$

We observe a curious feature of the channel definition. The authentication function is coupled with key delivery in such a way that $B$ has no real assurance that $k$ is new. Indeed, it is possible for $A$ to circumvent this channel. Suppose that $A$ knows an old session key $k'$ from a previous run of the protocol between some user $X$ and $B$. Then $A$ can force $B$ to accept $k'$ for a new session with $A$, as follows. In the attacking run, the first two protocol messages are exchanged as in a normal run.

1. $A \to B$ : $A, B, n_A$
2. $B \to S$ : $A, B, n_A, n_B$

The next message from $S$ which is actually meant for $B$ is intercepted by A.

3. $S \to A$ : $n_S, f(P_B, n_S, n_B, A) \oplus (k, h_A, h_B), g(P_B, k, h_A, h_B)$

Now $A$ computes the value $(k, h_A, h_B) = f(P_A, n_S, n_A, B)$, and computes the exclusive-or (XOR) of this value with the intercepted value $f(P_B, n_S, n_B, A) \oplus (k, h_A, h_B)$, to extract $f(P_B, n_S, n_B, A)$. Then $A$ pretends to be $S$ and sends the following message to $B$:

3'. $A \to B$ : $n_S, f(P_B, n_S, n_B, A) \oplus (k', h'_X, h'_B), g(P_B, k', h'_X, h'_B)$

Here we assume that $h'_X$, $h'_B$, and $g(P_B, k', h'_X, h'_B)$ were recorded by $A$ from the previous run between $X$ and $B$. The rest of of the protocol is then successfully completed as follows.

4. $B \to A$ : $n_S, h'_B$
5. $A \to B$ : $h'_X$

The above attack is rather unconventional, because here $A$ itself purportedly defeats the security of the subsequent session with $B$. It is easily precluded under the assumption that $B$ trusts $A$ to let a session key between the two to faithfully pass from $S$ to $B$. Nonetheless, such an assumption appears to be only implicit in Gong's discussion [48] and may be viewed as a potential weakness. In suggesting general design guidelines for cryptographic protocols, Abadi and Needham [58] caution that such trust assumptions may not always apply and should be adjudged carefully. The particular assumption seems to arise in Gong's protocol not so much as a genuine requirement, but rather as a result of a misplaced authentication channel.

### E.3.1 Gong's alternative protocol

In the same paper [48], Gong suggests an alternative protocol in which the responsibility for key generation rests solely with $S$. The server $S$ now randomly chooses all of $k$, $h_A$ and $h_B$ to be of the appropriate size. And the message it sends is symmetric with respect to $A$ and $B$ [48, p. 10]:

$$\begin{aligned}
3.\ S \to B:\ & n_S, f(P_A, n_S, n_A, B) \oplus (k, h_A, h_B), g(P_A, k, h_A, h_B), \\
& f(P_B, n_S, n_B, A) \oplus (k, h_A, h_B), g(P_B, k, h_A, h_B) \\
4.\ B \to A:\ & n_S, f(P_A, n_S, n_A, B) \oplus (k, h_A, h_B), g(P_A, k, h_A, h_B), h_B
\end{aligned}$$

The rest of the messages remain the same as in the previous protocol, and are omitted for the sake of brevity.

The confidentiality and authentication channels to $A$ and $B$ are now essentially the same as those to $B$ in the original protocol. So the curious feature applies to both $A$ and $B$; each of them can make the other accept a previously shared old key. Again it is crucial to make this assumption explicit.

## E.4 A protocol of Bull, Gong and Sollins

We now explain how the analysis technique enables us to discover the cause of a flaw in a protocol due to Bull *et al.* [59]. In this protocol, the message sent by $S$ is somewhat similar to the one in Gong's alternative protocol. A successful run of the protocol between $A$ and $B$ can be given as follows.

$$\begin{aligned}
1.\ A \to B:\ & A, f(P_A, B), n_A \\
2.\ B \to S:\ & A, B, f(P_B, S, A, f(P_A, B), n_A), n_A, n_B \\
3.\ S \to B:\ & f(P_B, A, n_B) \oplus k, f(P_B, A, n_B, k), \\
& f(P_A, B, n_A) \oplus k, f(P_A, B, n_A, k) \\
4.\ B \to A:\ & f(P_A, B, n_A) \oplus k, f(P_A, B, n_A, k)
\end{aligned}$$

It is easy to see that in this protocol the session key $k$ is not sent over a confidentiality channel from $S$. For note that $S$ cannot possibly be sure that $n_A$ or $n_B$ is new. As a result there is no guarantee that the session key is XORed with a new random value each time. So we can regard the particular channels used by $S$ as analogous to the Vernam cipher with a possibly repeating ciphering key (cf. [57]). Thus in contrast to Gong's protocols, the channels here no longer provide a confidentiality service. For example, suppose each of $k'$ (an old session key) and $k$ (a new session key) is XORed

with the same ciphering key $x$, giving the ciphertexts $x \oplus k'$ and $x \oplus k$, respectively. Then given $k'$ we can easily break the ciphertext $x \oplus k$ by computing $(x \oplus k') \oplus (x \oplus k) \oplus k'$, to reveal $k$.

In more concrete terms, suppose that in the above protocol an attacker $E$ knows an old session key $k'$, and that she has also recorded $n'_A$, $f(P_A, B)$ and $f(P_A, B, n'_A) \oplus k'$, all from the corresponding run. We also make the reasonable assumption that $S$ and $B$ ignore replays of nonces not generated by them. An attacking run on the protocol proceeds as follows, with $E$ masquerading as $A$ [47]:

1. $E \rightarrow B$ : $A, f(P_A, B), n'_A$
2. $B \rightarrow S$ : $A, B, f(P_B, S, A, f(P_A, B), n'_A), n'_A, n_B$
3. $S \rightarrow B$ : $f(P_B, A, n_B) \oplus k, f(P_B, A, n_B, k)$
   $f(P_A, B, n'_A) \oplus k, f(P_A, B, n'_A, k)$
4. $B \rightarrow E$ : $f(P_A, B, n'_A) \oplus k, f(P_A, B, n'_A, k)$

Although $k$ is intended to be a new session key $A$ and $B$, an attacker $E$ can easily compute

$$(f(P_A, B, n'_A) \oplus k) \oplus (f(P_A, B, n'_A) \oplus k') \oplus k'$$

to obtain $k$. At the end of the attacking run $B$ believes $k$ is shared with $A$, whereas in fact it is shared with an impostor $E$; it is easy to construct a similar attack where $E$ masquerades as $B$ to $A$. This concludes the modus operandi of our attack on the protocol. However, the gist of the above attack is that the protocol makes a fundamentally wrong use of a cryptographic algorithm.

In the same work [47] where the above attack was first published by us, we also suggested the following improved protocol to counter this attack:

1. $A \rightarrow B$ : $A, n_A$
2. $B \rightarrow S$ : $A, B, n_A, n_B$
3. $S \rightarrow B$ : $n_S, f(P_B, A, n_B, n_S) \oplus k, f(P_B, A, n_B, k),$
   $f(P_A, B, n_A, n_S) \oplus k, f(P_A, B, n_A, k)$
4. $B \rightarrow A$ : $n_S, f(P_A, B, n_A, n_S) \oplus k, f(P_A, B, n_A, k)$

Ironically, we later discovered that our improved protocol suffers from essentially the same curious feature found in Gong's protocols. Observe that in this protocol $k$ *is* sent over a confidentiality channel from $S$ to $B$. So the improved protocol does indeed represent a marked improvement over the protocol of Bull *et al.* However, the confidentiality channel used in this protocol is still not quite in its simplest form. In particular, it is unnecessary to include $A$'s name and $n_B$ in defining this channel, since

these fields are already included in the authentication channel from $S$ to $B$. In fact, their inclusion in the confidentiality channel is not only superfluous but, as explained below, also serves to potentially undermine the authentication channel.

Consider the confidentiality channel definition, inclusive of the superfluous elements:

$$S \rightarrow B : f(P_B, A, n_B, n_S) \oplus k$$

And consider the authentication channel definition, which is actually in its adequate form:

$$S \rightarrow B : f(P_B, A, n_B, k)$$

It becomes apparent that the inclusion of redundant fields in the confidentiality channel results in a striking similarity between the formats of the hashed components used in the two channel definitions. This symmetry can be exploited to construct essentially the same type of attack we demonstrated on Gong's protocols earlier. For instance, $A$ can force $B$ to accept $n_S$ as a session key between the two by intercepting message 3 and replacing it with message 3′, as follows.

$$\begin{aligned} 3.\quad & S \rightarrow A : \quad n_S, f(P_B, A, n_B, n_S) \oplus k, f(P_B, A, n_B, k), \\ & \qquad\qquad f(P_A, B, n_A, n_S) \oplus k, f(P_A, B, n_A, k) \\ 3'.\quad & A \rightarrow B : \quad n_S, f(P_B, A, n_B, n_S) \oplus n_S, f(P_B, A, n_B, n_S), \ldots \end{aligned}$$

Undoubtedly, it is possible to assume away such an attack by putting side conditions on the protocol. For example, we can require that $k$ and $n_S$ be somehow made distinct by a protocol implementation. Or, as in Gong's protocols, we can make a trust assumption on $B$'s side about $A$'s actions. Alternatively, here we can even eliminate such an assumption, by requiring $B$ to perform an additional check. Still further, the attack can be avoided by constraining protocol implementations to follow a particular ordering on the fields before hashing. (Such countermeasures are by no means exhaustive.) In principle, however, such measures do very little to address the unnecessary confusion of the confidentiality and authentication channels. (An essentially similar discussion applies to the channels from $S$ to $A$.)

## E.5 KryptoKnight protocols

KryptoKnight [49], [50] is an authentication and key distribution system developed by IBM. The KryptoKnight protocols have been implemented as part of IBM's NetSP (Network Security Program) system.

## E.5.1 Initial version

The original KryptoKnight mechanism described by Molva *et al.* [49] enables a user $A$ to obtain a session key generated by $S$ for use between $A$ and $B$, as follows:

1. $A \rightarrow S: \; n_A$
2. $S \rightarrow A: \; n_A, n_S, B, T, f(P_A, n_S \oplus B, n_A, n_S \oplus S, T) \oplus k$

(A similar exchange also essentially takes place between $S$ and $B$.) Here $T$ is the duration for which the session key $k$ is meant to remain valid. In contrast to Gong's protocols, $n_S$ is not chosen at random by $S$, but is obtained as the encryption of $n_A$ under $k$ using a non-reversible encryption function $E$:

$$n_S = E(k, n_A)$$

($E$ may be considered to have the same properties as $f$.) However, $n_S$ can be considered as essentially random, since $k$ is randomly chosen by $S$. Thus the confidentiality channel from $S$ to $A$ is similar to that in Gong's alternative protocol.

The authentication channel from $S$ to $A$ is essentially based on the binding between $k$ and $n_S$, albeit in a highly convoluted fashion. We may consider the authentication channel:

$$A \xleftarrow{a} S : k, n_A, B$$

to be defined as

$$S \rightarrow A : n_S, f(P_A, n_S \oplus B, \ldots) \oplus k$$

where $n_S = E(k, n_A)$. Notice this channel definition appears rather peculiar when compared with its counterpart from the protocol of Bull *et al.* Relatedly, the simplicity associated with the latter definition is no longer preserved.

## E.5.2 Recent version

The above key exchange mechanism appears to have been simplified by Bird *et al.* [50] to derive some recent protocols of the KryptoKnight family. Although our analysis of the original KryptoKnight mechanism did not reveal any specific weaknesses, we find surprising failures in the recent protocols.

Let us consider a specific instance of the basic key exchange protocol of Bird *et al.* [50, p. 35]:

1. $A \rightarrow S: \; B, n_A$
2. $S \rightarrow A: \; T, f(P_A, B, n_A, T) \oplus k$

It is apparent that this protocol provides neither session key confidentiality nor authenticity. The cause for loss of confidentiality is similar to that in the protocol of Bull *et al.*, which we discussed in the previous section. On the other hand, there is no authentication channel from $S$ to $A$; $A$ cannot be sure that the session key it supposedly recovers upon a protocol execution is indeed from $S$.

We note that several key exchange protocols proposed by Bird *et al.* ([50], pp. 36–38) are meant to cover the above instance as well, although the specific protocols proposed there employ the following message format:

$$S \rightarrow A : n_S, T, f(P_A, n_A, n_S, S, B, T) \oplus k, \ldots$$

Unlike the original KryptoKnight mechanism, here $n_S$ is randomly chosen by $S$, independently of $n_A$; $n_A$ itself is randomly chosen by $A$. Surprisingly, Bird *et al.* suggest that the nonce $n_S$ is of no particular value in their protocols:

> ...the use of $N_k$ $[n_S]$ in the tickets does not serve any particular purpose. $N_k$ $[n_S]$ is used here simply to preserve some homogeneity between ticket format in all scenarios, but for no other significant purpose. ([50], p. 38)

Nevertheless, it is easy to see that $n_S$ is crucial for maintaining session key confidentiality in their protocols. Indeed, if we act on the above suggestion of Bird *et al.* and omit $n_S$ from their protocols, then an attack similar to the one that we demonstrated on the protocol of Bull *et al.* in the previous section follows immediately.

On the other hand, Bird *et al.* admittedly allow loss of session key integrity. They note that $B$ can change the session key, without $A$'s knowledge ([50], p. 37). However, the resulting situation appears rather dubious with the protocols of Bird *et al.*, when compared with Gong's protocols. For now $B$ can even arrange that $A$ and $C$ share the same session key, and thus authenticate each other, although each of them may be purportedly authenticating $B$. Below we demonstrate an attack on one of their proposed protocols: the A-B-K ticket distribution protocol (expanded version). A successful run of the protocol between $A$ and $B$ can be given as follows [50, pp. 37]:

1. $A \rightarrow B$ : $A, n_A$
2. $B \rightarrow S$ : $n_A, n_B, A, B$
3. $S \rightarrow B$ : $n_S, T, f(P_A, n_A, n_S, S, B, T) \oplus k,$
   $f(P_B, n_B, n_S, S, A, T) \oplus k$
4. $B \rightarrow A$ : $n_S, T, f(P_A, n_A, n_S, S, B, T) \oplus k$

For simplicity of presentation, we have omitted certain message elements from the original protocol since they do not affect our attack. In the following 'triangle' attack

on the above protocol, $B$ engages in two parallel runs of the protocol, one with $A$ and the other with $C$. $S$ generates $k$ and $k'$ as the session keys for use between $A$ and $B$, and $C$ and $B$, respectively; $k_x$ is a value chosen by $B$.

$$
\begin{aligned}
&1. \quad A \rightarrow B: \quad A, n_A \\
&2. \quad B \rightarrow S: \quad n_A, n_B, A, B \\
&3. \quad S \rightarrow B: \quad n_S, T, f(P_A, n_A, n_S, S, B, T) \oplus k, \\
&\qquad\qquad\qquad\qquad f(P_B, n_B, n_S, S, A, T) \oplus k \\
&4. \quad B \rightarrow A: \quad n_S, T, f(P_A, n_A, n_S, S, B, T) \oplus k_x \\
\\
&1'. \quad C \rightarrow B: \quad C, n_C \\
&2'. \quad B \rightarrow S: \quad n_C, n'_B, C, B \\
&3'. \quad S \rightarrow B: \quad n'_S, T, f(P_C, n_C, n'_S, S, B, T) \oplus k', \\
&\qquad\qquad\qquad\qquad f(P_B, n'_B, n'_S, S, C, T) \oplus k' \\
&4'. \quad B \rightarrow C: \quad n'_S, T, f(P_C, n_C, n'_S, S, B, T) \oplus k_x
\end{aligned}
$$

Now $A$ and $C$ unexpectedly end up sharing $k_x$, although they did not directly participate in a mutual run with each other.

## E.6 Alternative designs using secure channels

The protocols examined in the previous sections reflect a mix-up of confidentiality and authentication channels. In particular, these protocols exhibit confusion about the purpose of the message fields and the use of cryptographic transformations. It is tempting to speculate that this confusion might have even been the root of flaws or unusual features in some of the protocols. We can easily avoid such defects by addressing the desired channels explicitly at the design stage itself. In fact, the same technique we used to analyze existing protocols can be applied equally well to design new protocols that can be shown to be secure. A key exchange protocol is designed by simply defining the required channels from the key originator to the key recipients. We illustrate this concept below by designing two concrete protocols using one-way functions.

### E.6.1 Three-party key exchange

Consider a conventional key exchange scenario where a shared key $k$ needs to be established between two principals $A$ and $B$ via a trusted server $S$. The server $S$ is assumed

to share secrets $P_A$ and $P_B$ with $A$ and $B$, respectively. The desired channels from $S$ to $A$ are specified as [45]:

$$S \xrightarrow{c} A: \ k$$
$$A \xleftarrow{a} S: \ k, A, B, N$$

where $N$ is a nonce used to convince $A$ that $k$ is new. The channels from $S$ to $B$ are similarly specified. We can look upon the above specification as a generic key exchange protocol. A concrete protocol is derived from the specification by implementing the required channels using the available cryptographic functions.

We shall define the required confidentiality channel from $S$ to $A$ as:

$$S \rightarrow A : f(P_A, n_S) \oplus k$$

where $n_S$ is a random value chosen by $S$. And we shall define the required authentication channel from $S$ to $A$ as:

$$S \rightarrow A : f(P_A, k, B, n_A)$$

where $n_A$ is a random value chosen by $A$. Recall that the hash value $f(P_A, \ldots)$ cannot be formed without the knowledge of $P_A$. Furthermore, $P_A$ is a shared secret between $A$ and $S$. We can thus regard $A$'s name as being implicitly included in the use of $P_A$, and thereby omit it from the actual definition. (The desired confidentiality and authentication channels from $S$ to $B$ are similarly defined.)

We assume that the ordering of messages is irrelevant, except for the constraint that certain messages must necessarily precede others. A protocol that makes the desired confidentiality and authentication channels concrete is now easily constructed as follows:

1. $A \rightarrow B: \ A, B, n_A$
2. $B \rightarrow S: \ A, B, n_A, n_B$
3. $S \rightarrow B: \ n_S, f(P_B, n_S) \oplus k, f(P_B, k, A, n_B), f(P_A, n_S) \oplus k, f(P_A, k, B, n_A)$
4. $B \rightarrow A: \ n_S, f(P_A, n_S) \oplus k, f(P_A, k, B, n_A)$

It is clear that the precise formulation of the channel requirements enables us to optimize the design by using exactly what is needed in each channel. The protocol is conceptually simple and elegant—the confidentiality and authentication channels now only contain those elements that are relevant to the function of each channel. Consequently, the channels are now transparent, which makes the purpose of the protocol messages quite clear.

### E.6.2 Conference key exchange

The above design extends straightforwardly to a conference key protocol. Assume that there are $n$ different participants from a set $\mathcal{U} = \{U_1, U_2, \ldots, U_n\}$, and that each $U_i \in \mathcal{U}$ shares a secret $P_{U_i}$ with $S$ initially. Let $k$ denote a conference key chosen by $S$ to be shared among the participants contained in $\mathcal{U}$. Then each $U_i \in \mathcal{U}$ generates its own nonce, $n_{U_i}$, and carries out the following exchange with $S$:

1. $U_i \rightarrow S: \ \mathcal{U}, n_{U_i}$
2. $S \rightarrow U_i: \ n_S, f(P_{U_i}, n_S) \oplus k, f(P_{U_i}, k, \mathcal{U} \setminus U_i, n_{U_i})$

('\' denotes the set difference operator.)

## E.7 Discussion

Throughout this appendix we have employed a methodology of Boyd and Mao [45] to pinpoint problems of varying seriousness in several existing key exchange protocols based on one-way functions. Furthermore, we used the insight gained from the analyses carried out to design a simplified protocol which we claim is as secure as any published protocol of its type, and still enjoys a transparent and elegant design. The simplicity of this approach has further enabled us to design a new conference key protocol as an obvious extension.

It may be argued that our protocols are susceptible to guessing attacks on the long-term secrets assumed initially if these secrets were user chosen passwords. However, all the previous protocols we considered also have the same feature. Although we make guessing infeasible by simply assuming that the initial secrets are well-chosen, it may be desirable to relax this assumption and investigate alternative designs that allow passwords to be used.

As noted by Boyd and Mao [45], their approach is targeted at key exchange only. We emphasize that it does not directly apply to the analysis of protocol properties that are essentially independent of key exchange. We illustrate this scope limitation by means of an example here. Consider the following protocol due to Bull *et al.* [59]:

1. $A \rightarrow S: \ n_A$
2. $S \rightarrow A: \ n_S, k \oplus f(P_A, n_A, n_S), f(k, n_S), \bar{n}_S$
3. $A \rightarrow S: \ f(k, \bar{n}_S)$
4. $S \rightarrow A: \ f(k, n_A)$

Here $S$ generates two nonces $n_S$ and $\bar{n}_S$. The first two messages constitute the key exchange phase of the protocol: these messages are used to transfer a new session key $k$ from $S$ to $A$ for use between them. It is not difficult to isolate the confidentiality and authentication channels used to transfer $k$ from $S$ to $A$. Indeed, the reason we are giving this example is not because we are concerned with the key exchange phase here. It is to show that the working of the subsequent phase lies outside the scope the analysis approach that we have used so far. The phase consisting of the last two messages essentially constitutes a handshake using the session key $k$. The intended interpretation of this handshake is as follows: the third message is used to confirm $A$'s receipt of the session key; and the fourth message is used to confirm $S$'s receipt of $A$'s confirmation. However, the latter message provides no such guarantee, as shown by an attack we found on this protocol earlier [47].

In the attacking run, an attacker $E$ copies the opening message from $A$:

$$1.\quad A \to S:\quad n_A \text{ (copied by } E)$$

The next message from $S$, which is actually meant for $B$, is intercepted by $E$:

$$2.\quad S \to E:\quad n_S, k \oplus f(P_A, n_A, n_S), f(k, n_S), \bar{n}_S$$

Now $E$ simply replaces $\bar{n}_S$ with $n_A$ in the above message. She also sends the resulting message to $A$ pretending to be $S$:

$$2'.\quad E \to A:\quad n_S, k \oplus f(P_A, n_A, n_S), f(k, n_S), n_A$$

She then prevents $A$'s response from reaching $S$ and instead plays it back to $A$:

$$3.\quad A \to E:\quad f(k, n_A)$$
$$4.\quad E \to A:\quad f(k, n_A)$$

At the end of the attacking run, $A$ wrongly believes that $S$ has responded to its handshake message, although $S$ in fact did not participate in the handshake.

The above attack essentially rests on the inability of a principal to detect a replay of one of its own messages. Such attacks are not new; similar attacks have been addressed in the past (cf., e.g., Mitchell [60]). For example, if the hash used as a handshake message also includes the name of the originator then the above attack is easily averted:

$$3.\quad A \to S:\quad f(k, \bar{n}_S, A)$$
$$4.\quad S \to A:\quad f(k, n_A, S)$$

Other possible solutions include the use of direction bits or of different hash functions in the handshake messages. For a discussion on the use of similar countermeasures in a more general setting, cf. Gong [21].

# Appendix F

## Modeling of recognizability

Treat this section as though it followed on directly from the end of Chapter 4.

The notion of *recognizable messages* essentially reflects the following intuition: (1) if a principal $P$ generates a message $X$, then $X$ is recognizable by $P$; (2) any message that can be possibly verified on the basis of $P$'s recognizable messages, perhaps using keys possessed by $P$, is also recognizable by $P$. The idea behind (2) is expanded as follows: (2′) if a message $X$ is recognizable by $P$ and $Y = E_K(X)$ for some $K$ such that $K^{-1}$ is possessed by $P$, then $Y$ is recognizable by $P$ (since $P$ can decrypt $Y$ using $K$ to reveal the recognizable message $X$); (2″) if a message $X$ is recognizable by $P$ and $Y = E_X(K)$ for some $K$ such that $K$ is possessed by $P$, then $Y$ is recognizable by $P$ (since $P$ can encrypt the recognizable message $X$ using $K$ to obtain $Y$); (2‴) if a message $X$ is recognizable by $P$ and $Y = H_K X$ for some $K$ such that $K$ is possessed by $P$, then $Y$ is recognizable by $P$ (since $P$ can hash the recognizable message $X$ using $K$ to obtain $Y$); and (6) if a message $X$ is recognizable by $P$ and $Y = Y_1 \mid \cdots \mid Y_k$ for some $Y_1, \ldots, Y_k$ such that $X = Y_i$ for some $i$, then $Y$ is recognizable by $P$ (since $P$ can reveal from $Y$ the recognizable message $X$). We fix the set of $P$'s recognizable messages for each time. The closure operation that we use in defining $P$'s set of recognizable messages at time $t$ roughly captures amongst others the following statement:

(*) if a message $X$ is in this set and $Y = E_K(X)$ for some $K$ such that $K^{-1}$ is in the set of $P$'s possessed messages at $t$, then $Y$ is in the set of $P$'s recognizable messages at $t$, provided that some principal has encrypted $X$ using $K$, and thus constructed $E_K(X)$, at a time earlier than $t$.

For each time $t$, we define the message set $\mathcal{M}_{recg}(P, r, t)$ to model the intuitive notion of recognizable messages.

**Definition F.1**

1. Let $t = t_{\text{first}}(r)$. Then $\mathcal{M}_{recg}(P, r, t) = \emptyset$.

2. Let $t > t_{\text{first}}(r)$. Then $\mathcal{M}_{recg}(P,r,t)$ is the smallest set of messages such that:

   I. (Basis)

$$\mathcal{M}_{recg}(P,r,t-1) \cup \mathcal{S}_{genr}(P,r,t-1) \subset \mathcal{M}_{recg}(P,r,t)$$

   II. (Induction)

   (a) $E_K(X) \in \mathcal{M}_{recg}(P,r,t)$ if $(X,K) \in \mathcal{E}(r,t)$ and $X \in \mathcal{M}_{recg}(P,r,t)$ and $K \in \mathcal{M}_{poss}(P,r,t)$

   (b) $E_K(X) \in \mathcal{M}_{recg}(P,r,t)$ if $(X,K) \in \mathcal{E}(r,t)$ and $X \in \mathcal{M}_{recg}(P,r,t)$ and $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$

   (c) $H_K(X) \in \mathcal{M}_{recg}(P,r,t)$ if $(X,K) \in \mathcal{H}(r,t)$ and $X \in \mathcal{M}_{recg}(P,r,t)$ and $K \in \mathcal{M}_{poss}(P,r,t)$

   (d) $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{recg}(P,r,t)$ if $(X_1,\ldots,X_k) \in \mathcal{C}(r,t)$ and $X_i \in \mathcal{M}_{recg}(P,r,t)$ for some $i$

**Lemma F.1** *For all $t$, $t'$ such that $t < t'$ the following holds:*

$$\mathcal{M}_{recg}(P,r,t) \subseteq \mathcal{M}_{recg}(P,r,t')$$

**Definition F.2** Let $i \geq 0$.

1. Let $t = t_{\text{first}}(r)$. Then $\mathcal{M}^i_{recg}(P,r,t) = \emptyset$ for all $i$.

2. Let $t > t_{\text{first}}(r)$. Then

$$\mathcal{M}^i_{recg}(P,r,t) = \begin{cases} \mathcal{M}_{recg}(P,r,t-1) \cup \mathcal{S}_{genr}(P,r,t-1) & \text{if } i = 0 \\ \mathcal{M}^{i-1}_{recg}(P,r,t) \cup \mathcal{S} & \text{if } i > 0 \end{cases}$$

   where

$$\begin{aligned} \mathcal{S} \;=\; & \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^{i-1}_{recg}(P,r,t) \text{ and} \\ & K \in \mathcal{M}_{poss}(P,r,t)\} \\ & \cup \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^{i-1}_{recg}(P,r,t) \text{ and} \\ & K^{-1} \in \mathcal{M}_{poss}(P,r,t)\} \\ & \cup \{H_K(X) \mid (X,K) \in \mathcal{H}(r,t) \text{ and } X \in \mathcal{M}^{i-1}_{recg}(P,r,t) \text{ and} \\ & K \in \mathcal{M}_{poss}(P,r,t)\} \\ & \cup \{(X_1 \mid \cdots \mid X_k) \mid (X_1,\ldots,X_k) \in \mathcal{C}(r,t) \text{ and} \\ & X_j \in \mathcal{M}^{i-1}_{recg}(P,r,t) \text{ for some } j\}. \end{aligned}$$

The following lemma is easily proved from Definition F.1 and Definition F.2.

**Lemma F.2**

$$\mathcal{M}^0_{recg}(P,r,t) \subseteq \mathcal{M}^1_{recg}(P,r,t) \subseteq \cdots \subseteq \bigcup_{i=0}^{\infty} \mathcal{M}^i_{recg}(P,r,t) = \mathcal{M}_{recg}(P,r,t)$$

**Lemma F.3**

$$\mathcal{M}_{genr}(P,r,t) \subseteq \mathcal{M}_{recg}(P,r,t)$$

**Proof.** By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3 and definition F.1, $\mathcal{M}_{genr}(P,r,t_{\text{first}}(r)) = \mathcal{M}_{recg}(P,r,t_{\text{first}}(r)) = \emptyset$. Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: $\mathcal{M}_{genr}(P,r,t) \subseteq \mathcal{M}_{recg}(P,r,t)$; and we show this implies $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}_{recg}(P,r,t+1)$.

   By the inductive hypothesis it follows that $\mathcal{M}_{genr}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t) \subseteq \mathcal{M}_{recg}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t)$. By definition 4.3, $\mathcal{M}_{genr}(P,r,t+1) = \mathcal{M}_{genr}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t)$, and, by definition F.2, $\mathcal{M}^0_{recg}(P,r,t+1) = \mathcal{M}_{recg}(P,r,t) \cup \mathcal{S}_{genr}(P,r,t)$. Hence $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}^0_{recg}(P,r,t+1)$. By Lemma F.2 it follows that $\mathcal{M}_{genr}(P,r,t+1) \subseteq \mathcal{M}_{recg}(P,r,t+1)$.

(This completes the proof of Lemma F.3.) □

**Lemma F.4**

$$\mathcal{M}_{recg}(P,r,t) \subseteq \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$$

**Proof.** (Similar to proof of Lemma 4.11.) By induction on $t$:

1. (Basis) Let $t = t_{\text{first}}(r)$. By definition 4.3 and definition F.1, $\mathcal{M}_{recg}(P,r,t_{\text{first}}(r)) = \emptyset$ and $\mathcal{M}_{genr}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{encr}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{hash}(r,t_{\text{first}}(r)) \cup \mathcal{M}_{conc}(r,t_{\text{first}}(r)) = \emptyset$. Therefore, the required statement holds.

2. (Induction) Let $t > t_{\text{first}}(r)$ be arbitrary. We assume the inductive hypothesis: (HP1) for all $t' < t$, $\mathcal{M}_{recg}(P,r,t') \subseteq \mathcal{M}_{genr}(r,t') \cup \mathcal{M}_{encr}(r,t') \cup \mathcal{M}_{hash}(r,t') \cup \mathcal{M}_{conc}(r,t')$; and we show this implies $\mathcal{M}_{recg}(P,r,t) \subseteq \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

   By Lemma F.2 it suffices to show that, for all $Y$ and for all $m$, if $Y \in \mathcal{M}^m_{recg}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$. This assertion is shown using induction on $m$:

I. (Basis) Let $m = 0$. Suppose $Y \in \mathcal{M}^0_{recg}(P,r,t)$. By definition F.2, $Y \in \mathcal{M}_{recg}(P,r,t-1) \cup \mathcal{S}_{genr}(P,r,t-1)$.

Case (i): $Y \in \mathcal{M}_{recg}(P,r,t-1)$. HP1 yields $Y \in \mathcal{M}_{genr}(r,t-1) \cup \mathcal{M}_{encr}(r,t-1) \cup \mathcal{M}_{hash}(r,t-1) \cup \mathcal{M}_{conc}(r,t-1)$. By Lemma 4.6 it follows that $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (ii): $Y \in \mathcal{S}_{genr}(P,r,t-1)$.

By definition 4.2, $P$ performs $generate(Y)$ at $t-1$, and therefore, by definition 4.3, $Y \in \mathcal{M}_{genr}(r,t)$.

II. (Induction) Let $m > 0$ be arbitrary. We assume the inductive hypothesis: (HP2) for all $Y$, if $Y \in \mathcal{M}^m_{recg}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$; and we show this implies that, for all $Y$, if $Y \in \mathcal{M}^{m+1}_{recg}(P,r,t)$ then $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Suppose $Y \in \mathcal{M}^{m+1}_{recg}(P,r,t)$. By definition F.2,

$$
\begin{aligned}
Y \in\ & \mathcal{M}^m_{recg}(P,r,t) \\
& \cup \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^m_{recg}(P,r,t) \text{ and} \\
& K \in \mathcal{M}_{poss}(P,r,t)\} \\
& \cup \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^m_{recg}(P,r,t) \text{ and} \\
& K^{-1} \in \mathcal{M}_{poss}(P,r,t)\} \\
& \cup \{H_K(X) \mid (X,K) \in \mathcal{H}(r,t) \text{ and } X \in \mathcal{M}^m_{recg}(P,r,t) \text{ and} \\
& K \in \mathcal{M}_{poss}(P,r,t)\} \\
& \cup \{(X_1 \mid \cdots \mid X_k) \mid (X_1,\ldots,X_k) \in \mathcal{C}(r,t) \text{ and} \\
& X_j \in \mathcal{M}^m_{recg}(P,r,t) \text{ for some } j\}.
\end{aligned}
$$

Case (A): $Y \in \mathcal{M}^m_{recg}(P,r,t)$. HP2 yields $Y \in \mathcal{M}_{genr}(r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

Case (B): $Y \in \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^m_{recg}(P,r,t) \text{ and } K \in \mathcal{M}_{poss}(P,r,t)\}$.

We have $Y = E_K(X)$ for some $X$ and some $K$ such that $(X,K) \in \mathcal{E}(r,t)$ and $X \in \mathcal{M}^m_{recg}(P,r,t)$ and $K \in \mathcal{M}_{poss}(P,r,t)$. By definition 4.3, $E_K(X) \in \mathcal{M}_{encr}(r,t)$. Hence $Y \in \mathcal{M}_{encr}(r,t)$.

Case (C): $Y \in \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t) \text{ and } X \in \mathcal{M}^m_{recg}(P,r,t) \text{ and } K^{-1} \in \mathcal{M}_{poss}(P,r,t)\}$.

Similar to Case (B).

Case (D): $Y \in \{H_K(X) \mid (X,K) \in \mathcal{H}(r,t)$ and $X \in \mathcal{M}^m_{recg}(P,r,t)$ and $K \in \mathcal{M}_{poss}(P,r,t)\}$.
Similar to Case (B).

Case (E): $Y \in \{(X_1 \mid \cdots \mid X_k) \mid (X_1,\ldots,X_k) \in \mathcal{C}(r,t)$ and $X_j \in \mathcal{M}^m_{recg}(P,r,t)$ for some $j\}$.
Similar to Case (B).

(This completes the proof of Lemma F.4.) □

The statement of Lemma F.4 is rather weak; it is apparent from the proof of this lemma that we can also obtain the following stronger statement: $\mathcal{M}_{recg}(P,r,t) \subseteq \mathcal{M}_{genr}(P,r,t) \cup \mathcal{M}_{encr}(r,t) \cup \mathcal{M}_{hash}(r,t) \cup \mathcal{M}_{conc}(r,t)$.

**Lemma F.5** *Let $X \in \mathcal{M}_{recg}(P,r,t)$. Then*

*(a) if $(X,K) \in \mathcal{E}(r,t)$ for some $K$ such that $K \in \mathcal{M}_{poss}(P,r,t)$, then $E_K(X) \in \mathcal{M}_{recg}(P,r,t)$, and*

*(b) if $(X,K) \in \mathcal{E}(r,t)$ for some $K$ such that $K^{-1} \in \mathcal{M}_{poss}(P,r,t)$, then $E_K(X) \in \mathcal{M}_{recg}(P,r,t)$, and*

*(c) if $(X,K) \in \mathcal{H}(r,t)$ for some $K$ such that $K \in \mathcal{M}_{poss}(P,r,t)$, then $H_K(X) \in \mathcal{M}_{recg}(P,r,t)$, and*

*(d) if $X_1 \mid \cdots \mid X_k \in \mathcal{C}(r,t)$ and $X = X_i$ for some $i$, then $X_1 \mid \cdots \mid X_k \in \mathcal{M}_{recg}(P,r,t)$.*

**Proof.** We only prove part (a); the remaining parts are proved similarly. Suppose $(X,K) \in \mathcal{E}(r,t)$ for some $K$ such that $K \in \mathcal{M}_{poss}(P,r,t)$. By Lemma F.2 it suffices to show that $E_K(X) \in \mathcal{M}^l_{recg}(P,r,t)$ for some $l$. Since $X \in \mathcal{M}_{recg}(P,r,t)$, it follows by Lemma F.2 that $X \in \mathcal{M}^m_{recg}(P,r,t)$ for some $m$. By definition F.2, $\mathcal{M}^{m+1}_{recg}(P,r,t) \supseteq \{E_K(X) \mid (X,K) \in \mathcal{E}(r,t)$ and $X \in \mathcal{M}^m_{recg}(P,r,t)$ and $K \in \mathcal{M}_{poss}(P,r,t)$. Hence $E_K(X) \in \mathcal{M}^{m+1}_{recg}(P,r,t)$. □

**Proposition F.1** *For each time $t$, the set $\mathcal{M}_{recg}(P,r,t)$ is finite.*

**Proof.** Follows from Lemma F.4 and Lemma 4.7. □

**Corollary F.1** *For each time $t$, $\mathcal{M}_{recg}(P,r,t) = \mathcal{M}^k_{recg}(P,r,t)$ for some $k$.*