Strand Spaces: Proving Security Protocols Correct*

F. Javier Thayer Fábrega Jonathan C. Herzog
Joshua D. Guttman
The MITRE Corporation
202 Burlington Rd., MS A150
Bedford, MA 01730 USA
{jt, jherzog, guttman}@mitre.org

Abstract

A *strand* is a sequence of events; it represents either an execution by a legitimate party in a security protocol or else a sequence of actions by a penetrator. A *strand space* is a collection of strands, equipped with a graph structure generated by causal interaction. In this framework, protocol correctness claims may be expressed in terms of the connections between strands of different kinds.

Preparing for a first example, the Needham-Schroeder-Lowe protocol, we prove a lemma that gives a bound on the abilities of the penetrator in any protocol. Our analysis of the example gives a detailed view of the conditions under which it achieves authentication and protects the secrecy of the values exchanged. We also use our proof methods to explain why the original Needham-Schroeder protocol fails.

Before turning to a second example, we introduce *ideals* as a method to prove additional bounds on the abilities of the penetrator. We can then prove a number of correctness properties of the Otway-Rees protocol, and we clarify its limitations.

We believe that our approach is distinguished from other work by the simplicity of the model, the precision of the results it produces, and the ease of developing intelligible and reliable proofs even without automated support.

*Appears in *Journal of Computer Security*, 7 (1999), pages 191–230.

Contents

1 Introduction

A security protocol is an exchange of messages between two or more parties in which encryption is used to provide authentication or to distribute cryptographic keys for new conversations [20]. Even when security protocols have been developed carefully by experts and reviewed carefully by other experts, they are often found later to have flaws that make them unusable (see, for example, [6, 12]). In many cases, the attacks do not presuppose any weakness in the cryptosystem being used, and would be just as harmful with an ideal cryptosystem. In other cases, characteristics of the cryptosystem and characteristics of the protocol combine to cause protocol failure [19, 5, 21].

Analyzing security protocols consists mainly of two complementary activities. The first is to find flaws in those protocols that are not correct, and the second is to establish convincingly the correctness of those that are. These activities are interrelated, because the discovery of a flaw may suggest an altered protocol that we may wish to prove correct, and because a failure to prove the correctness of a protocol may suggest a particular flaw.

In this paper, however, we focus on the second activity, proving the correctness of protocols when they are in fact correct. Moreover, at this stage, we will study protocol correctness assuming ideal cryptography.

Much work both recently (for instance, [1, 25, 31]) and of an earlier vintage (such as [7, 3]) has proposed techniques for proving protocols correct. We believe that the approach presented here has several advantages.

- Our approach gives a clear semantics to the assumption that certain data items, such as nonces and session keys, are fresh, and never arise in more than one protocol run.

- We work with an explicit model of the possible behaviors of a system penetrator; this allows us to develop general theorems that bound the abilities of the penetrator, independent of the protocol under study. (A simple example is presented below in Section 3.2; a more powerful method is introduced in Section 6.)

- Our method allows various notions of correctness, involving both secrecy and authentication, to be stated and proved.

- In our opinion, the approach leads to detailed insight into the reasons why the protocol is correct, and the assumptions required. Proofs are simple and informative: they are easily developed by hand, and they help to identify more exact conditions under which we can rely on the protocol.

Our basic contribution is the *strand space*. A strand space is a set of *strands*, and a strand is a sequence of events that a single principal may engage in. Each individual strand is a sequence of message transmissions and receptions, with specific values of all data such as keys and nonces. It is thus a sequential process that exhibits neither internal nor external choice [11].

For a legitimate principal, a strand represents the actions of that party (but of that party only, not its presumed interlocutor) in one particular run of the protocol. If that party may be involved in more than one run of the protocol during a period of time, they are represented by other strands. The activities of different parties are represented by different strands.

A strand for a penetrator is a sequence of message transmissions and receptions that model a basic capability a penetrator should be assumed to possess. Examples of penetrator strands include such activities as:

- receiving a symmetric key and a message encrypted using that key, and then sending the result of decrypting the message;
- receiving two messages and sending the result of concatenating them;
- sending out a data item such as a name that the penetrator may know.

Useful penetrator actions may be modeled by connecting a number of penetrator strands.

A *strand space* is a set of strands, consisting of strands for the various legitimate protocol parties, together with penetrator strands. One may think of a strand space as containing all the legitimate executions of the protocol expected within its useful lifetime, together with all the actions that a penetrator might apply to the messages contained in those executions.

A *bundle* is a portion of a strand space. It consists of a number of strands—legitimate or otherwise—hooked together where one strand sends a message and another strand receives that same message. Typically, for a protocol to be *correct*, each such bundle must contain one strand for each of the legitimate principals apparently participating in this session, all agreeing on the principals, nonces, and session keys [15, 26, 32]. Penetrator strands or stray legitimate strands may also be entangled in a bundle, even in a correct protocol, but they should not prevent the legitimate parties from agreeing on the data values, or from maintaining the secrecy of the values chosen.

One may think of a bundle as collecting all of the activities that were relevant to one run of a protocol, although the definition that we give allows a bundle to contain additional events that need not have been strictly relevant.

A strand is a linear structure, a sequence of one principal's message transmissions and receptions. A bundle is a graph-structured entity, representing the communication among a number of strands.

Protocol correctness typically depends essentially on the *freshness* of data items such as nonces and session keys. For this reason, the strand spaces that concern us are not full, in the sense that they do not contain all the strands that would arise if all possible data items were used. Presumably, the useful lifetime of a protocol is much shorter than the length of time that would be needed for the principals to use every possible session key or random value, and indeed we may reasonably assume that values of these kinds will be invented only once during the lifetime of the protocol.

A strand space models the assumption that some values occur only freshly by including only one strand *originating* that data item by initially sending a

message containing it. Many strands, by contrast, may stand ready to combine with the originating strand by receiving the message and processing its contents further. A strand space will also model the assumption that some values are impossible for a penetrator to guess; in essence, the space simply lacks any penetrator strand in which this value is sent without having first been received.

In this paper, we will develop the basic machinery of strand spaces (Section 2). This machinery includes a partial order that models causal contribution, and justifies an induction-like proof method (Section 2.2). We then develop our model of the penetrator (Section 3), including a simple but useful theorem that gives a general bound on what the penetrator can do, regardless of the protocol being modeled (Section 3.2). Section 4 describes notions of correctness that may be easily expressed.

In Section 5, we study the Needham-Schroeder-Lowe public key protocol [20, 12, 13] as an example, proving both authentication results (Sections 5.2 and 5.5) and secrecy results (Section 5.4).

In Section 6 we develop some more sophisticated machinery for reasoning about protocols, based on a notion of *ideal*. We use this concept (in Section 6.3) to state more powerful bounds on the penetrator than the straightforward theorem of Section 3.2. We then turn to the Otway-Rees protocol as a case study to show the utility of these results. In contrast to the Needham-Schroeder-Lowe protocol, the Otway-Rees protocol uses secret-key cryptography; the results of Section 6 are particularly useful for secret-key protocols.

In each case study, we discover detailed (and unexpected) information on the exact conditions under which the protocol is correct.

2 Strand Spaces

In this section, we will introduce strand spaces and related notions (Section 2.1). A *bundle* (Section 2.2) is a portion of a strand space large enough to represent at least a full protocol exchange; it has a natural causal precedence relation relative to which inductive arguments may be carried out. The set of messages that we will consider in the present paper are described in Section 2.3. In [27], we develop a less restrictive treatment that supports all of the reasoning we develop in this paper; however, the details presented there would merely distract from the main points in this exposition.

2.1 Basic Notions

Consider a set A, the elements of which are the possible messages that can be exchanged between principals in a protocol. We will refer to the elements of A as *terms*. We will later (Section 2.3) impose more algebraic structure on the set A, but in this section we assume only that a *subterm* relation is defined on A. $t_0 \sqsubset t_1$ means t_0 is a subterm of t_1.

In a protocol, principals can either send or receive terms. We will represent transmission of a term as the occurrence of that term with positive sign, and

reception of a term as its occurrence with a negative sign.

Definition 2.1 *A* signed term *is a pair* $\langle \sigma, a \rangle$ *with* $a \in \mathsf{A}$ *and* σ *one of the symbols* $+, -$. *We will write a signed term as* $+t$ *or* $-t$. $(\pm\mathsf{A})^*$ *is the set of finite sequences of signed terms. We will denote a typical element of* $(\pm\mathsf{A})^*$ *by* $\langle \langle \sigma_1, a_1 \rangle, \ldots, \langle \sigma_n, a_n \rangle \rangle$.

By abuse of language, we will still treat signed terms as ordinary terms. For instance, we shall refer to subterms of signed terms.

Definition 2.2 *A* strand space *over* A *is a set* Σ *together with a trace mapping* $\mathrm{tr} : \Sigma \to (\pm\mathsf{A})^*$.

We will usually represent a strand space by its underlying set of strands Σ. In particular applications of the theory, the trace mapping need not be injective. We may want to distinguish between various instances of the same trace; for example, we may need to distinguish identical traces occurring at different times to model replay attacks.

Definition 2.3 Fix a strand space Σ

1. A *node* is a pair $\langle s, i \rangle$, with $s \in \Sigma$ and i an integer satisying $1 \leq i \leq \mathrm{length}(\mathrm{tr}(s))$. The set of nodes is denoted by $\mathcal{N}$. We will say the node $\langle s, i \rangle$ belongs to the strand s. Clearly, every node belongs to a unique strand.

2. If $n = \langle s, i \rangle \in \mathcal{N}$ then $\mathrm{index}(n) = i$ and $\mathrm{strand}(n) = s$. Define $\mathrm{term}(n)$ to be $(\mathrm{tr}(s))_i$, i.e. the ith signed term in the trace of s. Similarly, $\mathrm{uns\_term}(n)$ is $((\mathrm{tr}(s))_i)_2$, i.e. the unsigned part of the ith signed term in the trace of s.

3. There is an edge $n_1 \rightarrow n_2$ if and only if $\mathrm{term}(n_1) = +a$ and $\mathrm{term}(n_2) = -a$ for some $a \in \mathsf{A}$. Intuitively, the edge means that node n_1 sends the message a, which is received by n_2, recording a potential causal link between those strands.

4. When $n_1 = \langle s, i \rangle$ and $n_2 = \langle s, i+1 \rangle$ are members of $\mathcal{N}$, there is an edge $n_1 \Rightarrow n_2$. Intuitively, the edge expresses that n_1 is an immediate causal predecessor of n_2 on the strand s. We write $n' \Rightarrow^+ n$ to mean that n' precedes n (not necessarily immediately) on the same strand.

5. An unsigned term t *occurs in* $n \in \mathcal{N}$ iff $t \sqsubset \mathrm{term}(n)$.

6. Suppose I is a set of unsigned terms. The node $n \in \mathcal{N}$ is an *entry point* for I iff $\mathrm{term}(n) = +t$ for some $t \in I$, and whenever $n' \Rightarrow^+ n$, $\mathrm{term}(n') \notin I$.

7. An unsigned term t *originates* on $n \in \mathcal{N}$ iff n is an entry point for the set $I = \{t' : t \sqsubset t'\}$.

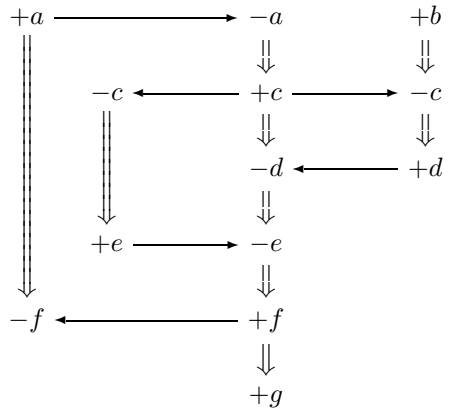


Figure 1: A Bundle

8. An unsigned term t is *uniquely originating* iff t originates on a unique $n \in \mathcal{N}$.

If a term t originates uniquely in a particular strand space, then it can play the role of a nonce or session key in that structure.

$\mathcal{N}$ together with both sets of edges $n_1 \rightarrow n_2$ and $n_1 \Rightarrow n_2$ is a directed graph $\langle \mathcal{N}, (\rightarrow \cup \Rightarrow) \rangle$.

2.2 Bundles and Causal Precedence

A *bundle* is a finite subgraph of this graph, for which we can regard the edges as expressing the causal dependencies of the nodes. Figure 1 illustrates a particular bundle.

Definition 2.4 *Suppose* $\rightarrow_{\mathcal{C}} \subset \rightarrow$*; suppose* $\Rightarrow_{\mathcal{C}} \subset \Rightarrow$*; and suppose* $\mathcal{C} = \langle \mathcal{N}_{\mathcal{C}}, (\rightarrow_{\mathcal{C}} \cup \Rightarrow_{\mathcal{C}}) \rangle$ *is a subgraph of* $\langle \mathcal{N}, (\rightarrow \cup \Rightarrow) \rangle$*.* $\mathcal{C}$ *is a bundle if:*

1. $\mathcal{C}$ *is finite.*
2. *If* $n_2 \in \mathcal{N}_{\mathcal{C}}$ *and* $term(n_2)$ *is negative, then there is a unique* n_1 *such that* $n_1 \rightarrow_{\mathcal{C}} n_2$*.*
3. *If* $n_2 \in \mathcal{N}_{\mathcal{C}}$ *and* $n_1 \Rightarrow n_2$ *then* $n_1 \Rightarrow_{\mathcal{C}} n_2$*.*
4. $\mathcal{C}$ *is acyclic.*

In conditions 2 and 3, it follows that $n_1 \in \mathcal{N}_{\mathcal{C}}$, because $\mathcal{C}$ is a graph.

For our purposes, it does not matter whether communication is regarded as a synchronizing event or as an asynchronous activity. This definition formalizes a process communication model with three properties:

- A strand (process) may send or receive a message, but not both at the same time;
- When a strand receives a message m, there is a unique node transmitting m from which the message was immediately received;
- When a strand transmits a message m, many strands may immediately receive m.

Notational Convention 2.5 *A node n is in a bundle $\mathcal{C} = \langle \mathcal{N}_\mathcal{C}, \rightarrow_\mathcal{C}, \Rightarrow_\mathcal{C} \rangle$, written $n \in \mathcal{C}$, if $n \in \mathcal{N}_\mathcal{C}$; a strand s is in $\mathcal{C}$ if all of its nodes are in $\mathcal{N}_\mathcal{C}$.*

If $\mathcal{C}$ is a bundle, then the $\mathcal{C}$-height *of a strand s is the largest i such that $\langle s, i \rangle \in \mathcal{C}$. $\mathcal{C}$-trace$(s) = \langle tr(s)(1), \ldots, tr(s)(m) \rangle$, where $m = \mathcal{C}$-height(s).*

Definition 2.6 *If $\mathcal{S}$ is a set of edges, i.e. $\mathcal{S} \subset \rightarrow \cup \Rightarrow$, then $\prec_\mathcal{S}$ is the transitive closure of $\mathcal{S}$, and $\preceq_\mathcal{S}$ is the reflexive, transitive closure of $\mathcal{S}$.*

The relations $\prec_\mathcal{S}$ and $\preceq_\mathcal{S}$ are each subsets of $\mathcal{N}_\mathcal{S} \times \mathcal{N}_\mathcal{S}$, where $\mathcal{N}_\mathcal{S}$ is the set of nodes incident with any edge in $\mathcal{S}$.

Lemma 2.7 *Suppose $\mathcal{C}$ is a bundle. Then $\preceq_\mathcal{C}$ is a partial order, i.e. a reflexive, antisymmetric, transitive relation. Every non-empty subset of the nodes in $\mathcal{C}$ has $\preceq_\mathcal{C}$-minimal members.*

We regard $\preceq_\mathcal{C}$ as expressing causal precedence, because $n \prec_\mathcal{S} n'$ holds only when n's occurrence causally contributes to the occurrence of n'. When a bundle $\mathcal{C}$ is understood, we will simply write $\preceq$. Similarly, "minimal" will mean $\preceq_\mathcal{C}$-minimal.

The existence of minimal members in non-empty sets serves as an induction principle, an observation that clarifies the relation of our approach to Paulson's and Schneider's [25, 31].

Most of our arguments turn on the $\preceq_\mathcal{C}$-minimal elements in some set of nodes. These arguments are motivated by the question, "What did he know, and when did he know it?"

Lemma 2.8 *Suppose $\mathcal{C}$ is a bundle, and $S \subseteq \mathcal{C}$ is a set of nodes such that*

$$\forall m, m' \,.\, \mathit{uns\_term}(m) = \mathit{uns\_term}(m') \quad \mathit{implies} \quad (m \in S \quad \mathit{iff} \quad m' \in S)$$

If n is a $\preceq_\mathcal{C}$-minimal member of S, then the sign of n is positive.

Proof. If term(n) were negative, then by the bundle property, $n' \rightarrow n$ for some $n' \in \mathcal{C}$ and uns_term(n) = uns_term(n'). Hence, $n' \in S$, violating the minimality property of n. ■

Lemma 2.9 *Suppose $\mathcal{C}$ is a bundle, $t \in \mathsf{A}$ and $n \in \mathcal{C}$ is a $\preceq_\mathcal{C}$-minimal element of $\{m \in \mathcal{C} : t \sqsubset \mathit{term}(m)\}$. The node n is an originating occurrence for t.*

Proof. Because n is a member, $t \sqsubset$ term(n). By Lemma 2.8, the sign of n is positive. If $n' \Rightarrow^+ n$, then applying Definition 2.4, Clause 3 as many times as necessary, $n' \in \mathcal{C}$. Hence by the minimality property of n, $t \not\sqsubset$ term(n'). Thus n is originating for t. ■

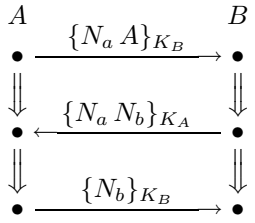


Figure 2: Needham-Schroeder

2.3 Terms and Encryption

We will now specialize the set of terms A. In particular we will assume given:

- A set $\mathsf{T} \subseteq \mathsf{A}$ of texts (representing the atomic messages).
- A set $\mathsf{K} \subseteq \mathsf{A}$ of cryptographic keys disjoint from T, equipped with a unary operator $\mathsf{inv} : \mathsf{K} \to \mathsf{K}$.

  We assume that inv is injective; that it maps each member of a key pair for an asymmetric cryptosystem to the other; and that it maps a symmetric key to itself.
- Two binary operators

$$\mathsf{encr} : \mathsf{K} \times \mathsf{A} \to \mathsf{A}$$
$$\mathsf{join} : \mathsf{A} \times \mathsf{A} \to \mathsf{A}$$

We will follow custom and write $\mathsf{inv}(K)$ as K^{-1}, $\mathsf{encr}(K, m)$ as $\{m\}_K$, and $\mathsf{join}(a, b)$ as $a\, b$. If k is a set of keys, k^{-1} denotes the set of inverses of elements of k.

We illustrate this notation in Figure 2, which shows the bundle containing the intended behavior of the Needham-Schroeder public key protocol [20]. The column below A represents the strand consisting of the initiator's activity during the exchange, while the column below B represents the strand of the respondent's activity. In the form we discuss it here, the protocol assumes that each participant has somehow acquired the other's public key. One party, the initiator A, generates a number randomly (a "nonce"); he joins this to his name and encrypts it with the intended respondent's public key. The latter generates a nonce of his own, sending it and the initiator's nonce back, encrypted with the initiator's public key. He has thus answered the initiator's challenge by showing that he could read the first message. Finally, the initiator returns the respondent's nonce encrypted with the respondent's public key.

The intended result of this protocol is that the two participants should come to share access to the values N_a and N_b, each associating these values with the other participant, and no other party should be in possession of them. The protocol might be used in a context where the two values are hashed together

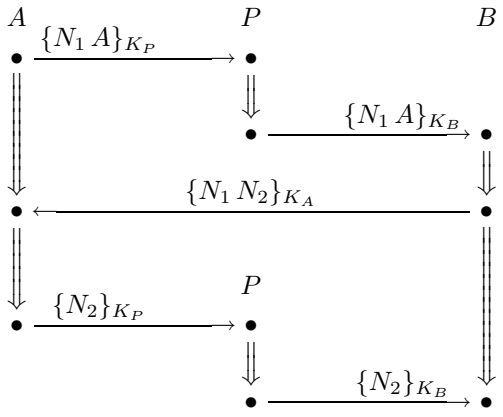


Figure 3: Needham-Schroeder Infiltrated

to yield a shared symmetric key for an encrypted session, for instance. In fact, it does not achieve this goal [12]; Figure 3 displays a bundle that serves a counterexample and illustrates what can go wrong in this protocol. In this figure, the penetrator P has two periods of activity, each represented here by a short strand. The initiator A intends to have a session with P or some principal whose key P controls; P exploits this opportunity to impersonate A to B. Figure 4 will show in more detail how this behavior could be achieved.

2.4 Freeness Assumptions

The proofs in this paper use an assumption we will call the assumption of free encryption; many other authors (e.g. [14, 18, 24]) have made similar assumptions, dating back to Dolev and Yao [7], although not all have [8]. It stipulates that a ciphertext can be regarded as a ciphertext in just one way:

Axiom 1 *For $m, m' \in \mathsf{A}$ and $K, K' \in \mathsf{K}$,*

$$\{m\}_K = \{m'\}_{K'} \Longrightarrow m = m' \wedge K = K'$$

For clarity of exposition we make a stronger assumption in this paper, namely that A is the algebra freely generated from T and K by the two operators encr and join, as embodied in Axiom 2.

Axiom 2 *For $m_0, m_0', m_1, m_1' \in \mathsf{A}$ and $K, K' \in \mathsf{K}$,*

1. $m_0\, m_1 = m_0'\, m_1' \Longrightarrow m_0 = m_0' \wedge m_1 = m_1'$
2. $m_0\, m_1 \neq \{m_0'\}_{K'}$
3. $m_0\, m_1 \notin \mathsf{K} \cup \mathsf{T}$

4. $\{m_0\}_K \notin \mathsf{K} \cup \mathsf{T}$

This is more than is needed for our method but it leads to the simplest exposition of the main points. In [27] we showed how to weaken this assumption considerably, accounting for the possibility that the join operator is associative, for instance.

Given Axiom 2, we may define the width of terms:

Definition 2.10 *If $m \in \mathsf{K} \cup \mathsf{T}$ or if $m = \{m_0\}_K$, then width$(m) = 1$. If $m = m_0\, m_1$, then width$(m) =$ width$(m_0) +$ width(m_1).*

Attacks that might exist if there are terms that may be "read" as having more than one form are referred to as *type flaw attacks* [4]. Some type flaw attacks seem implausible—in the sense that most implementations would not be vulnerable to them—while others are more troublesome.

Type flaw attacks are an example of a more general issue in protocol analysis. In real protocols, the algebra of messages has many more relations—that is, identities holding among terms—than we allow in our model. For instance, message composition is usually an associative operator as implemented. More seriously, contrary to Axiom 1, in real cryptosystems there are non-trivial identities of the form $\{m\}_K = \{m'\}_{K'}$. In what sense then can we say that our techniques provide useful information about protocols which use real cryptography?

For any encryption algebra A there is a free encryption algebra A' and a surjective algebra morphism $\pi : \mathsf{A}' \to \mathsf{A}$. Moreover, π and A' are unique to within isomorphism, this being effectively the definition of free algebra in the theory of universal algebras. In this paper we have shown protocol correctness results for strand spaces over the free algebra A'. Now it is easy to see that if a protocol property fails for strands over A', then the same protocol property fails for A. However, the converse is not true, since protocol failures may exploit relations in the algebra A that cannot be lifted to A'. Nevertheless, much useful information can be obtained by considering the free message algebra, since we are thereby excluding vulnerabilities based on the structure of the protocol itself, rather than on particular properties of the message algebra.

The problem remains to determine which relations among the elements of the free algebra A' will preserve a protocol correctness result. This is a hard problem, which will doubtless require much future work exploring different approaches; Maneki has considered one aspect of this problem [16].

Since we have assumed that our message algebra A is freely generated, we can use a simple inductive definition of the subterm relation.

Definition 2.11 *The subterm relation $\sqsubset$ is defined inductively, as the smallest relation such that:*

- $a \sqsubset a$;
- $a \sqsubset \{g\}_K$ *if* $a \sqsubset g$;
- $a \sqsubset g\, h$ *if* $a \sqsubset g$ *or* $a \sqsubset h$.

We should emphasize that, for $K \in \mathsf{K}$, $K \sqsubset \{g\}_K$ only if $K \sqsubset g$ already. Restricting subterms in this way reflects an assumption about the penetrator's capabilities: that keys can be obtained from ciphertext only if they are embedded in the text that was encrypted. This might not always be the case—for instance, if a dictionary attack is possible—but it is the assumption we will make here.

This notion of subterm does not always mesh perfectly with the definition of origination and unique origination, which refers to the subterm relation (Definition 2.3, Clauses 7 and 8). In some cases, it might be more natural to use a notion of origination referring to a larger relation $\sqsubset'$; that relation would be defined so that

$$a \sqsubset' \{g\}_K \quad \text{iff} \quad a \sqsubset' g \ \vee \ a = K \ \vee \ a = \{g\}_K$$

Definition 7.1, Clause 3, for instance, contains a condition on what key a server may choose that would be unnecessary with the alternative notion $\sqsubset'$. In this paper, however, we will make do with $\sqsubset$.

An immediate consequence of the freeness assumption and the inductive definition of subterm is:

Proposition 2.12 *Suppose $K \neq K'$ and $\{h'\}_{K'} \sqsubset \{h\}_K$. Then $\{h'\}_{K'} \sqsubset h$.*

3 The Penetrator

The penetrator's powers are characterized by two ingredients, namely a set of keys known initially to the penetrator and a set of penetrator strands that allow the penetrator to generate new messages from messages he intercepts.

A *penetrator set* consists of a set of keys $\mathsf{K_P}$. It contains the keys initially known to the penetrator. Typically it would contain: all public keys; all private keys held by the penetrator or his accomplices; and all symmetric keys K_{px}, K_{xp} initially shared between the penetrator and principals playing by the protocol rules. It may also contain "lost keys" that became known to the penetrator previously, perhaps because he succeeded in some cryptanalysis.

3.1 Penetrator Strands

The atomic actions available to the penetrator are encoded in a set of *penetrator traces*. They summarize his ability to discard messages, generate well known messages, piece messages together, and apply cryptographic operations using keys that become available to him. A protocol attack typically requires hooking together several of these atomic actions.

Definition 3.1 *A* penetrator trace *is one of the following:*

M. *Text message:* $\langle +t \rangle$ *where* $t \in \mathsf{T}$

F. *Flushing:* $\langle -g \rangle$

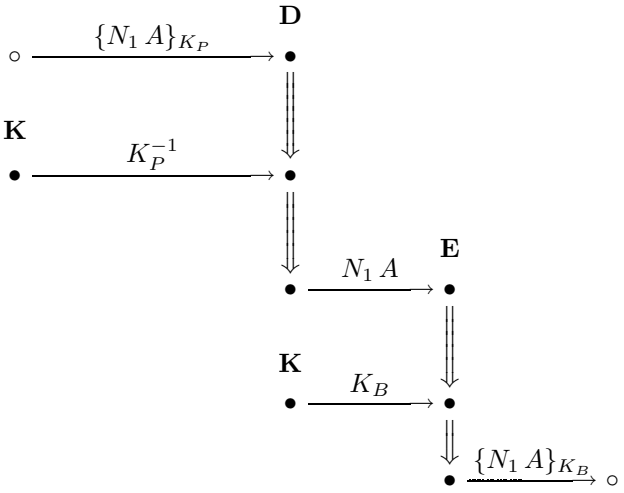


Figure 4: Needham-Schroeder: Penetrator's First Step

T. *Tee:* $\langle -g,\ +g,\ +g \rangle$

C. *Concatenation:* $\langle -g,\ -h,\ +g\, h \rangle$

S. *Separation into components:* $\langle -g\, h,\ +g,\ +h \rangle$

K. *Key:* $\langle +K \rangle$ *where* $K \in \mathsf{K_P}$.

E. *Encryption:* $\langle -K,\ -h,\ +\{h\}_K \rangle$.

D. *Decryption:* $\langle -K^{-1},\ -\{h\}_K,\ +h \rangle$.

This set of penetrator traces gives the penetrator powers similar to those in other approaches, e.g. [14, 24]. They ensure that the values that may be emitted by the penetrator are closed under joining, encryption, and the relevant "inverses." Figure 4 shows an example, illustrating how these penetrator strands can be hooked together to provide the behavior assumed in Figure 3. The open circles ($\circ$) here show the two points at which this diagram meshes with the first nodes of A and B's strands at the top of Figure 3. The label above each of the four strands shows which kind of strand it is, following Definition 3.1.

It is also possible to extend the set of penetrator traces given here if it is desired to model some special ability of the penetrator. That requires no essential change to our overall framework, although the proofs in this paper would then need to be modified to take account of the additional penetrator traces. Our theorems characterize a penetrator with just the powers we have described; a penetrator with additional computational or cryptanalytic abilities may not be subject to the same limitations.

One example of an extended penetrator would be a penetrator who can cryptanalyze old session keys, and thus benefit from some kinds of replay attacks [6]; the penetrator we formalize here does not have this ability.

Definition 3.2 *An* infiltrated strand space *is a pair* $(\Sigma, \mathcal{P})$ *with* Σ *a strand space and* $\mathcal{P} \subseteq \Sigma$ *such that* $tr(p)$ *is a penetrator trace for all* $p \in \mathcal{P}$.

A strand $s \in \Sigma$ *is a* penetrator strand *if it belongs to* $\mathcal{P}$*, and a node is a* penetrator node *if the strand it lies on is a penetrator strand. Otherwise we will call it a* non-penetrator *or* regular *strand or node.*

A node n *is an* ***M****,* ***F****, etc. node if* n *lies on a penetrator strand with a trace of kind* ***M****,* ***F****, etc.*

We would not expect an infiltrated strand space to realize all of the penetrator traces of type **M**. In that case, the space could not model unguessable nonces. It is usually assumed that the space Σ lacks **M**-strands for many text values, which regular participants can use for fresh nonces.

In the remainder of this paper, we will examine infiltrated strand spaces in which the regular strands all belong to a single protocol. In [30], we examine the case in which the regular strands may belong to more than one protocol.

3.2 A Bound on the Penetrator

Because the powers of the penetrator are defined by the penetrator keys and the penetrator strands, they are independent of the choice of a particular protocol to be proved correct. We can accordingly prove general facts about the penetrator's powers, re-using them whenever we become interested in a new protocol. In Section 6.3, we develop several powerful theorems about the penetrator, which are used to prove results about various protocols. Here, we will prove a simple theorem that is useful in the example we will turn to next, namely the Needham-Schroeder-Lowe protocol.

The proof of this theorem is typical of how we use Lemma 2.7. By "$S \setminus T$" we mean the set difference of S and T.

Proposition 3.3 *Let* $\mathcal{C}$ *be a bundle, and let* $K \in \mathsf{K} \setminus \mathsf{K_P}$.

If K *never originates on a regular node, then* $K \not\sqsubset term(n)$ *for any node* $n \in \mathcal{C}$*. In particular, for any penetrator node* $p \in \mathcal{C}$*,* $K \not\sqsubset term(p)$.

Proof. Consider the set $S = \{n \in \mathcal{C} : K \sqsubset \text{term}(n)\}$. Suppose (to derive a contradiction) that S is non-empty. Then S has members that are minimal relative to $\preceq_{\mathcal{C}}$ (Lemma 2.7). By Lemma 2.9, any $\preceq_{\mathcal{C}}$-minimal members of S are originating occurrences of K. Hence, by the assumption, they are all penetrator nodes. By Lemma 2.8, they are all positive nodes. We will now examine the possible cases for positive penetrator nodes.

M. The strand has the form $\langle +t \rangle$ where $t \in \mathsf{T}$, but $K \not\sqsubset t$.

F. The strand has the form $\langle -g \rangle$, and thus lacks any positive nodes.

T. The strand has the form $\langle -g,\ +g,\ +g \rangle$, so no value originates on the positive nodes.

C. The strand has the form $\langle -g,\ -h,\ +g\, h \rangle$; by the freeness of the algebra no key is a subterm of the positive node unless it was a subterm of a previous node.

S. The strand has the form $\langle -g\, h,\ +g,\ +h \rangle$, so no value originates on the positive nodes.

K. The strand has the form $\langle +K_0 \rangle$ where $K_0 \in \mathsf{K_P}$. But $K \sqsubset K_0$ iff $K = K_0$, contrary to the assumption that $K \in \mathsf{K} \setminus \mathsf{K_P}$.

E. The strand has the form $\langle -K_0,\ -h,\ +\{h\}_{K_0} \rangle$. By the definition of $\sqsubset$, $a \sqsubset \{h\}_{K_0}$ iff $a \sqsubset h$ or $a = \{h\}_{K_0}$. But because our algebra is freely generated, $K \neq \{h\}_{K_0}$. Hence, no key can occur in the positive node without having occurred in a previous node.

D. The strand has the form $\langle -K_0^{-1},\ -\{h\}_{K_0},\ +h \rangle$. By the definition of $\sqsubset$, $a \sqsubset h$ only if $a \sqsubset \{h\}_{K_0}$, so no key can occur in the positive node without having occurred in a previous node.

Hence S is in fact empty. But if S is empty, then $K \not\sqsubset \mathrm{term}(n)$ for any $n \in \mathcal{C}$, hence certainly $K \not\sqsubset \mathrm{term}(p)$ for penetrator nodes $p \in \mathcal{C}$. ■

This proof method is characteristic: it successively considers the minimal elements in a set, considers whether they are regular nodes or penetrator nodes, and finally takes cases on the different forms of penetrator strands. Proposition 3.3 is an instance of a fact we will establish later as Corollary 6.12. We have proved it separately here because it is useful in Section 5 and because we wanted to illustrate a straightforward use of this characteristic proof method.

4 Notions of Correctness

Gavin Lowe studies a range of authentication properties in [15]; strand spaces are a natural model for stating and proving his *agreement* properties, which are akin to the *correspondence* properties of Woo and Lam [32].

A protocol guarantees agreement to a participant B (say, as the responder) for certain data items $\vec{x}$ if:

> each time a principal B completes a run of the protocol as responder using $\vec{x}$, which to B appears to be a run with A, then there is a unique run of the protocol with the principal A as initiator using $\vec{x}$, which to A appears to be a run with B.

For a regular strand in an authentication protocol, the principal engaging in that strand, as well as the apparent interlocutor, can be inferred from the contents of the terms occurring in the strand.

A weaker non-injective agreement does not ensure uniqueness, but requires only:

each time a principal B completes a run of the protocol as responder using $\vec{x}$, apparently with A, then there is a run of the protocol with the principal A as initiator using $\vec{x}$, apparently with B.

Non-injective agreement is weaker because it does not prevent the other party A from being duped into executing multiple runs matching a single run by B.

We can prove non-injective agreement by establishing that, whenever a bundle $\mathcal{C}$ contains a strand representing a responder run using $\vec{x}$, then $\mathcal{C}$ also contains a strand representing an initiator run that corresponds in the sense that it also uses $\vec{x}$. We can establish agreement by showing that $\mathcal{C}$ contains a *unique* initiator strand using $\vec{x}$. We will illustrate these properties in Propositions 5.2 and 5.8, and also in Propositions 7.8 and 7.9.

We will also state a simple notion of secrecy for a data value x, which will be sufficient for our purposes here. A value x is secret in a bundle $\mathcal{C}$ if for every $n \in \mathcal{C}$, $\mathrm{term}(n) \neq x$. Propositions 5.10 and 7.4 illustrate this property.

This notion of secrecy concerns only what is "said on the wire." In this sense, a value is secret if the regular strands never emit it, and the penetrator can never emit it. Regular protocol participants may "know" a secret value in the sense of carrying out computations that depend on it, so long as their behavior in the protocol does not include disclosing it in public.

Moreover, if we prove that the penetrator never emits a value, it follows that he can never derive it from values he receives: for if he derived it, then he would be capable of emitting it. The penetrator strands defined in Definition 3.1 correspond to the ways that a penetrator would derive new values from those he already possesses. For instance, if the penetrator received a value $g\,x$, then an **S**-strand would lead to the penetrator emitting the supposed secret x.

More stringent notions of secrecy are also possible, as for instance information flow security properties, and may be fruitfully applied to security protocols [9].

5 The Needham-Schroeder-Lowe Protocol

The Needham-Schroeder-Lowe protocol was proposed by Gavin Lowe [13] as a way to fix the public-key protocol proposed by Needham and Schroeder [20], which he had discovered to be flawed [12]. In the form Lowe considers, the protocol assumes that each participant has somehow discovered the other's public key.

1. $A \longrightarrow B$: $\{N_a\, A\}_{K_B}$
2. $B \longrightarrow A$: $\{N_a\, N_b\, B\}_{K_A}$
3. $A \longrightarrow B$: $\{N_b\}_{K_B}$

This protocol differs from the original Needham-Schroeder public key protocol only in message 2; in the original protocol, B's name is not included.

In [13], Lowe proves the correctness of the revised protocol, showing that any attack against the revised protocol could be realized using just two runs of the protocol. The FDR model checker discloses that no attack exists on such a small system; this result is confirmed by examining the possible forms of an attack. In this section we will give a different proof using the strand space approach.

We specialize the term algebra somewhat, equipping it with:

- A set of names $\mathsf{T}_{\text{name}} \subseteq \mathsf{T}$. We will use variables such as A, B to range over T_{name}.
- A mapping $K : \mathsf{T}_{\text{name}} \to \mathsf{K}$. This is the mapping that associates a public key with each principal. We will follow tradition by writing $K(A)$ in the form K_A. We will assume that this function is injective, so that if $K_A = K_B$, then $A = B$.

The protocol does not achieve its authentication goals unless the mapping K is injective.

5.1 NSL Strand Spaces

Definition 5.1 *An infiltrated strand space $\Sigma, \mathcal{P}$ is an* NSL *space if Σ is the union of three kinds of strands:*

1. *Penetrator strands $s \in \mathcal{P}$;*
2. *"Initiator strands" $s \in \text{Init}[A, B, N_a, N_b]$ with trace:*

$$\langle +\{N_a\, A\}_{K_B}, \quad -\{N_a\, N_b\, B\}_{K_A}, \quad +\{N_b\}_{K_B} \rangle$$

   where $A, B \in \mathsf{T}_{name}$, $N_a, N_b \in \mathsf{T}$ but $N_a \notin \mathsf{T}_{name}$. $\text{Init}[A, B, N_a, N_b]$ will denote the set of all strands with the trace shown. The principal associated with this strand is A.

3. *Complementary "responder strands" $s \in \text{Resp}[A, B, N_a, N_b]$ with trace:*

$$\langle -\{N_a\, A\}_{K_B}, \quad +\{N_a\, N_b\, B\}_{K_A}, \quad -\{N_b\}_{K_B} \rangle$$

   where $A, B \in \mathsf{T}_{name}$, $N_a, N_b \in \mathsf{T}$ but $N_b \notin \mathsf{T}_{name}$. $\text{Resp}[A, B, N_a, N_b]$ will denote the set of all strands with the trace shown. The principal associated with this strand is B.

If $s \in \text{Init}[A, B, N_a, N_b]$ or $s \in \text{Resp}[A, B, N_a, N_b]$ is a regular strand, then we refer to A and B as the initiator and the responder of s (respectively), and to N_a and N_b as the initiator's value and responder's value (respectively). The intention is that these values should be nonces, in the sense of texts uniquely originating in Σ. Although all the initiator and responder strands in the strand space are complete, in the sense that they contain all three nodes, particular bundles may contain only the first one or two nodes on some strand.

Given any strand s in Σ, we can uniquely classify it as a penetrator strand, an initiator's strand, or a respondent's strand just by the form of its trace. In particular, given an NSL space Σ, we can read off which strands are penetrator strands, so that $(\Sigma, \mathcal{P})$ is uniquely determined. Hence we can omit $\mathcal{P}$ safely.

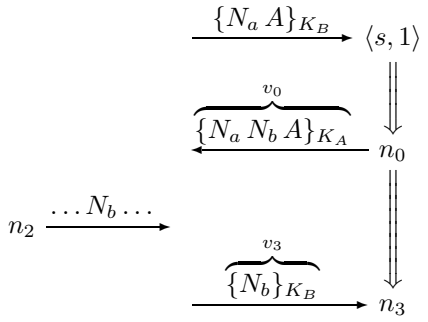


Figure 5: Regular Node n_2: Minimal in S

5.2 Agreement: The Responder's Guarantee

Proposition 5.2 *Suppose:*

1. Σ *is an NSL space,* $\mathcal{C}$ *is a bundle in* Σ*, and* s *is a responder strand in* $\mathrm{Resp}[A, B, N_a, N_b]$ *with* $\mathcal{C}$*-height 3;*
2. $K_A^{-1} \notin \mathsf{K_P}$*; and*
3. $N_a \neq N_b$ *and* N_b *is uniquely originating in* Σ*.*

Then $\mathcal{C}$ *contains an initiator's strand* $t \in \mathrm{Init}[A, B, N_a, N_b]$ *with* $\mathcal{C}$*-height 3.*

We will prove this using a sequence of lemmas. Throughout the remainder of this section, we will fix an arbitrary Σ, $\mathcal{C}$, s, A, B, N_a, and N_b satisfying the hypotheses of Proposition 5.2. The node $\langle s, 2\rangle$ outputs the value $\{N_a\, N_b\, B\}_{K_A}$; for convenience we will refer to this node as n_0, and to its term as v_0. The node $\langle s, 3\rangle$ receives the value $\{N_b\}_{K_B}$; we will refer to this node as n_3 and its term as v_3. We will identify two additional nodes n_1 and n_2 during the course of the proof, such that $n_0 \prec n_1 \prec n_2 \prec n_3$.

Lemma 5.3 N_b *originates at* n_0*.*

Proof. By the assumptions, $N_b \sqsubset v_0$, and the sign of n_0 is positive. Thus, we need only check that $N_b \not\sqsubset n'$, where n' is the node $\langle s, 1\rangle$ preceding n_0 on the same strand. Since $\mathrm{term}(n') = \{N_a\, A\}_{K_B}$, we need to check that $N_b \neq N_a$, which is a hypothesis, and $N_b \neq A$, which follows from the stipulation—in Definition 5.1 Clause 3—that the responder's value not be in $\mathsf{T}_{\mathrm{name}}$. ■

Next comes the main lemma, which establishes that the crucial step is taken by a regular strand and not a penetrator strand. As usual, it considers the $\preceq$-minimal members of a set of nodes. The content of the lemma is represented in Figure 5.

Lemma 5.4 *The set* $S = \{n \in \mathcal{C} : N_b \sqsubset \mathit{term}(n) \wedge v_0 \not\sqsubset \mathit{term}(n)\}$ *has a* $\preceq$*-minimal node* n_2*. The node* n_2 *is regular, and the sign of* n_2 *is positive.*

PROOF. Because $n_3 \in \mathcal{C}$, and n_3 contains N_b but not v_0, S is non-empty. Hence S has at least one $\preceq$-minimal element n_2 by Lemma 2.7. The sign of n_2 is positive by Lemma 2.8.

Can n_2 lie on a penetrator strand p? Let us examine the possible cases for positive penetrator nodes, according to the form of the trace of p. We will consider case **S** last.

M. The trace $\mathrm{tr}(p)$ has the form $\langle +t \rangle$ where $t \in \mathsf{T}$; so we must have $t = N_b$. In this case N_b originates on this strand. But that is impossible, as N_b originates uniquely on the regular node n_0 (Lemma 5.3).

F. The trace $\mathrm{tr}(p)$ has the form $\langle -g \rangle$, and thus lacks any positive nodes.

T. The trace $\mathrm{tr}(p)$ has the form $\langle -g,\ +g,\ +g \rangle$, so the positive nodes are not minimal occurrences.

C. The trace $\mathrm{tr}(p)$ has the form $\langle -g,\ -h,\ +g\,h \rangle$, so the positive node is not a minimal occurrence.

K. The trace $\mathrm{tr}(p)$ has the form $\langle +K_0 \rangle$ where $K_0 \in \mathsf{K_P}$. But $N_b \not\sqsubset K_0$, so this case does not apply.

E. The trace $\mathrm{tr}(p)$ has the form $\langle -K_0,\ -h,\ +\{h\}_{K_0} \rangle$. Suppose $N_b \sqsubset \{h\}_{K_0} \wedge v_0 \not\sqsubset \{h\}_{K_0}$. Since $N_b \neq \{h\}_{K_0}$, $N_b \sqsubset h$. Moreover, $v_0 \not\sqsubset h$, so the positive node is not minimal in S.

D. The trace $\mathrm{tr}(p)$ has the form $\langle -K_0^{-1},\ -\{h\}_{K_0},\ +h \rangle$. If the positive node is minimal in S, then $v_0 \not\sqsubset h$ but $v_0 \sqsubset \{h\}_{K_0}$. Hence (using the assumption of free encryption) $h = N_a\, N_b\, B$ and $K_0 = K_A$. Thus, there exists a node m (the first on this strand) with $\mathrm{term}(m) = K_A^{-1}$. Since by assumption, $K_A^{-1} \notin \mathsf{K_P}$, we may apply Proposition 3.3 to infer that K_A^{-1} originates on a regular node. However, no initiator strand or responder strand originates K_A^{-1}.

S. The trace $\mathrm{tr}(p)$ has the form $\langle -g\,h,\ +g,\ +h \rangle$. Assume $\mathrm{term}(n_2) = g$; there is a symmetrical case if $\mathrm{term}(n_2) = h$.

Because $n_2 \in S$, $N_b \sqsubset g$ and $v_0 \not\sqsubset g$. Observe that $v_0 \sqsubset h$ in this situation: by the minimality of n_2, we know $v_0 \sqsubset g\,h$. However, $v_0 \neq g\,h$, hence $v_0 \sqsubset h$.

Let $T = \{m \in \mathcal{C} : m \prec n_2 \wedge g\,h \sqsubset \mathrm{term}(m)\}$. Every member of T is a penetrator node, because no regular node contains a subterm $g\,h$ where h contains any encrypted subterm.

T is non-empty because $\langle p, 1 \rangle \in T$. Hence T has a minimal member m by Lemma 2.7, which is of positive sign by Lemma 2.8. Let us consider what kind of strand m can lie on.

M, F, T, K. Clearly a minimal member of T cannot lie on these strands.

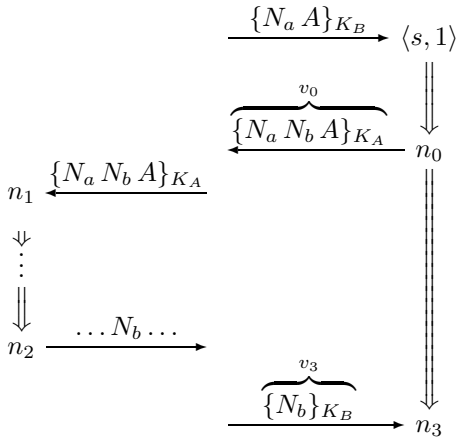


Figure 6: Node n_1 Contains v_0

S. If $g\, h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **S**, then $g\, h \sqsubset \text{term}(\langle p', 1\rangle)$. Moreover, $\langle p', 1\rangle \prec m$, contradicting the minimality of m in T.

E. If $g\, h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **E**, then $g\, h \sqsubset \text{term}(\langle p', 2\rangle)$. Moreover, $\langle p', 2\rangle \prec m$, contradicting the minimality of m in T.

D. If $g\, h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **D**, then $g\, h \sqsubset \text{term}(\langle p', 2\rangle)$. Moreover, $\langle p', 2\rangle \prec m$, contradicting the minimality of m in T.

C. Suppose $g\, h \sqsubset \text{term}(m)$, where m is a positive node on a strand p' of kind **C**, and m is minimal in T. Then $g\, h = \text{term}(m)$, and p' has trace $\langle -g,\ -h,\ +g\, h\rangle$. Hence, $\text{term}(\langle p', 1\rangle) = \text{term}(n_2)$ and $\langle p', 1\rangle \prec n_2$, contradicting the minimality of n_2 in S.

Therefore n_2 does not lie on a penetrator strand, but must lie on a regular strand instead. ■

Definition 5.5 *Fix some n_2 that is $\preceq$-minimal in $S = \{n \in \mathcal{C} : N_b \sqsubset \text{term}(n) \wedge v_0 \not\sqsubset \text{term}(n)\}$, and is therefore regular and of positive sign.*

Let t be the strand on which n_2 lies. We show next that t also has a node in which v_0 ($= \{N_a\, N_b\, B\}_{K_A}$) occurs. This lemma is illustrated in Figure 6.

Lemma 5.6 *A node n_1 precedes n_2 on t, and $\text{term}(n_1) = \{N_a\, N_b\, B\}_{K_A}$.*

Proof. N_b originates at n_0 (Lemma 5.3), and originates uniquely in Σ (Assumption 3). Moreover, $n_2 \neq n_0$, because $v_0 \sqsubset \text{term}(n_0)$ while $v_0 \not\sqsubset \text{term}(n_2)$. Hence, N_b does not originate at n_2. So there is a node n_1 preceding n_2 on

the same strand such that $N_b \sqsubset \text{term}(n_1)$. By the minimality property of n_2, $v_0 = \{N_a\, N_b\, B\}_{K_A} \sqsubset \text{term}(n_1)$. However, as no regular node contains an encrypted term as a proper subterm, $\{N_a\, N_b\, B\}_{K_A} = \text{term}(n_1)$. ■

Lemma 5.7 *The regular strand t containing n_1 and n_2 is an initiator strand, and is contained in $\mathcal{C}$.*

Proof. Node n_2 is a positive regular node and comes after a node (namely n_1) of the form $\{x\, y\, z\}_K$. Hence t is an initiator strand; if it were a responder strand, it would contain only a negative node after one of that form. Thus, n_1 and n_2 are the second and third nodes of t respectively. Since the last node of t is contained in $\mathcal{C}$, it must have $\mathcal{C}$-height of 3. ■

Proof of Proposition 5.2. Proposition 5.2 now follows immediately from Lemmas 5.6 and 5.7. ■

We have now proved the non-injective agreement property for the NSL responder. Injectivity follows easily on the assumption that the initiator chooses his value N_a so that it uniquely originates. If N_a is not uniquely originating, then the injectivity property is clearly false.

Proposition 5.8 *If Σ is an NSL space, and N_a is uniquely originating in Σ, then there is at most one strand $t \in \text{Init}[A, B, N_a, N_b]$ for any A, B, and N_b.*

Proof. If $t \in \text{Init}[A, B, N_a, N_b]$ for any A, B, and N_b, then $\langle t, 1\rangle$ is positive, $N_a \sqsubset \text{term}\langle t, 1\rangle$, and N_a cannot possibly occur earlier on t. So N_a originates at node $\langle t, 1\rangle$. Hence, if N_a originates uniquely in Σ, there can be at most one such t. ■

The requirement in Proposition 5.2 that N_a and N_b be distinct is a peculiarity of our approach. Without this assumption, the proposition is false. The responder strand

$$\langle -\{N_a\, A\}_{K_B}, \quad +\{N_a\, N_a\, B\}_{K_A}, \quad -\{N_a\}_{K_B}\rangle$$

can be embedded in a bundle $\mathcal{C}$ in which N_a and A originate on **M**-nodes, and the final term $\{N_a\}_{K_B}$ is generated by the penetrator on the "off chance" that B will reuse the given nonce N_a. The responder's nonce N_b $(= N_a)$ does originate uniquely then; however, not on the responder's strand, but on an **M**-strand.

In a probabilistic model, we would assume that the choice of N_b is independent of the value of N_a. In this case, the penetrator's strategy will succeed sometimes, but no more frequently than randomly generating the bits to encrypt to make up the last message. Hence, this strategy may be safely ignored.

Thus, our strand space model can be more stringent than a faithful probabilistic model. An implementor can justify "cutting corners," for instance by not programming the check for $N_b = N_a$, by showing in the probabilistic model that an exploitation strategy has negligible probability of success, despite existing in the strand space model.

5.3 The Original Needham-Schroeder Protocol

This analysis also sheds light on why the original Needham-Schroeder protocol would be vulnerable. The analysis is exactly parallel except that the Lemma corresponding to Lemma 5.6 would read:

Lemma 5.9 *In the original Needham-Schroeder protocol, a node n_1 precedes n_2 on the same regular strand t, and $term(n_1) = \{N_a\, N_b\}_{K_A}$.*

With this weaker information, we can not conclude that $t \in \mathrm{Init}[A, B, N_a, N_b]$, because the responder's identity is not determined by the term $\{N_a\, N_b\}_{K_A}$, which is all that we know s and t agree on. We can only infer that $t \in \mathrm{Init}[A, C, N_a, N_b]$ for some C. This is exactly the weakness that Lowe's attack exploits.

5.4 Secrecy: The Responder's Nonce

We may use the same methods to show that the responder's nonce N_b remains secret in the protocol. For this result, we also need to assume that the responder's private key is not compromised. If it were, the penetrator could read N_b directly from the last message of the exchange.

Proposition 5.10 *Suppose:*

1. *Σ is an NSL space, $\mathcal{C}$ is a bundle in Σ, and s a responder's strand in* $\mathrm{Resp}[A, B, N_a, N_b]$*;*
2. *$K_A^{-1} \notin \mathsf{K_P}$ and $K_B^{-1} \notin \mathsf{K_P}$; and*
3. *$N_a \neq N_b$ and N_b is uniquely originating in Σ.*

Then for all nodes $m \in \mathcal{C}$ such that $N_b \sqsubset term(m)$, either $\{N_a\, N_b\, B\}_{K_A} \sqsubset term(m)$ or $\{N_b\}_{K_B} \sqsubset term(m)$. In particular, $N_b \neq term(m)$.

Proof. Let Σ, $\mathcal{C}$, s, A, B, N_a, and N_b satisfy the hypotheses, and, as in Proposition 5.2, we will again refer to $\langle s, 2\rangle$ as n_0, and to its term $\{N_a\, N_b\, B\}_{K_A}$ as v_0. The node $\langle s, 3\rangle$ receives the value $\{N_b\}_{K_B}$; we will refer to this node as n_3 and its term as v_3. Consider the set:

$$\begin{aligned} S = \{n \in \mathcal{C} \quad &: \quad N_b \sqsubset \mathrm{term}(n) \\ &\wedge \quad v_0 \not\sqsubset \mathrm{term}(n) \ \wedge\ v_3 \not\sqsubset \mathrm{term}(n)\} \end{aligned}$$

If S is non-empty, then it has at least one $\preceq$-minimal element. We show first (Lemma 5.11) that such nodes are not regular. We next show (Lemma 5.12) that they are not penetrator nodes. Therefore S is empty, and the theorem holds.

Lemma 5.11 *No minimal member of S is a regular node.*

PROOF. Suppose instead that $m \in S$ is minimal and a regular node. The sign of m is positive by Lemma 2.8.

Node m cannot lie on s: Only n_0 is positive, and $v_0 = \text{term}(n_0)$, so n_0 is not in S.

Nor can m lie on a responder's strand $s' \neq s$. In that case, $m = \langle s', 2 \rangle$, so $\text{term}(m) = \{N, N', C\}_{K_D}$. Since $N_b \sqsubset \text{term}(m)$, either $N_b = N$ or $N_b = N'$.

- If $N_b = N$, $N_b \sqsubset \text{term}(\langle s', 1 \rangle)$, because the first node $\langle s', 1 \rangle$ is $\{N, D\}_{K_C} = \{N_b, D\}_{K_C}$. Moreover, $v_0 \not\sqsubset \{N_b, D\}_{K_C}$ and $v_3 \not\sqsubset \{N_b, D\}_{K_C}$. Hence $\langle s', 1 \rangle \in S$. Since $\langle s', 1 \rangle \prec m$, this contradicts the minimality of m.
- If $N_b \neq N$ and $N_b = N'$, then N_b originates at m, contradicting the assumption that N_b originates uniquely on n_0.

Suppose next that m lies on an initiator strand s'. It must be either the first or third node.

- If $m = \langle s', 1 \rangle$, then since $N_b \sqsubset \text{term}(m)$, N_b originates at m, contradicting the assumption that N_b originates uniquely on n_0.
- If $m = \langle s', 3 \rangle$, then $\text{term}(m) = \{N_b\}_{K_C}$. So the second node $\langle s', 2 \rangle$ is of the form $\{x\, N_b\, C\}_K$. However, $C \neq B$, because otherwise $v_3 = \text{term}(m)$. Hence $\langle s', 2 \rangle \prec m$ is in S, contradicting the minimality of m. ■

Lemma 5.12 *No minimal member of S is a penetrator node.*

PROOF SKETCH. The proof is almost identical to the proof of Lemma 5.4. The only significant difference is that when the penetrator strand is of type **D**, we must consider two cases. In one case, $h = N_a\, N_b\, B$ and $K_0 = K_A$, which are the plaintext and key that produce v_0. In the other case, $h = N_b$ and $K_0 = K_B$, which are the plaintext and key that produce v_3. Hence, we must apply Proposition 3.3 to each of the two private keys, which explains the need to assume both uncompromised. ■

5.5 The Initiator's Guarantees: Secrecy and Agreement

The proof of the secrecy of the initiator's nonce N_a is very similar to the proof we have just given.

Proposition 5.13 *Suppose:*

1. *Σ is an NSL space, $\mathcal{C}$ is a bundle in Σ, and s an initiator's strand in* $\text{Init}[A, B, N_a, N_b]$ *with $\mathcal{C}$-height 3;*
2. *$K_A^{-1} \notin \mathsf{K_P}$ and $K_B^{-1} \notin \mathsf{K_P}$; and*
3. *N_a is uniquely originating in Σ.*

Then for all nodes $m \in \mathcal{C}$ such that $N_a \sqsubset \text{term}(m)$, either $\{N_a\, A\}_{K_B} \sqsubset \text{term}(m)$ or $\{N_a\, N_b\, B\}_{K_A} \sqsubset \text{term}(m)$. In particular, $N_a \neq \text{term}(m)$.

By contrast, the initiator's guarantee of agreement is essentially different. In particular, it requires a stronger hypothesis than Proposition 5.2, namely that both private keys K_A^{-1} and K_B^{-1} are uncompromised. Not surprisingly, if $K_B^{-1} \in \mathsf{K_P}$, then the penetrator can complete the entire exchange with no activity on B's part.

Somewhat more surprising is this: If $K_A^{-1} \in \mathsf{K_P}$, then the penetrator can read B's reply $\{N_a\, N_b\, B\}_{K_A}$, substituting a different reply $\{N_a\, N'\, B\}_{K_A}$. This attack prevents us from proving agreement for the initiator assuming only that the responder's private key is uncompromised. Indeed, a proof approach based on an analogy with Proposition 5.2 fails.

However, we can prove an agreement theorem using the secrecy of N_a as a lemma.

Proposition 5.14 *Suppose:*

1. Σ *is an NSL space,* $\mathcal{C}$ *is a bundle in* Σ*, and* s *an initiator's strand in* $\mathrm{Init}[A, B, N_a, N_b]$ *with* $\mathcal{C}$*-height 3;*
2. $K_A^{-1} \notin \mathsf{K_P}$ *and* $K_B^{-1} \notin \mathsf{K_P}$*; and*
3. N_a *is uniquely originating in* Σ*.*

Then there exists a responder's strand $t \in \mathrm{Resp}[A, B, N_a, N_b]$ *where* t *has* $\mathcal{C}$*-height 2.*

Proof Sketch. Consider the set $\{m \in \mathcal{C} : \{N_a\, N_b\, B\}_{K_A} \sqsubset \mathrm{term}(m)\}$. It is non-empty because it contains $\langle s, 2\rangle$. So it contains a minimal member m_0. If m_0 lies on a regular strand t, then we can show that $t \in \mathrm{Resp}[A, B, N_a, N_b]$, and that t has two nodes (at least) in $\mathcal{C}$.

If instead m_0 lies on a penetrator strand t, then t can be shown to be an **E**-strand with trace

$$\langle -K_A, \quad -N_a\, N_b\, B, \quad +\{N_a\, N_b\, B\}_{K_A}\rangle$$

But this contradicts Proposition 5.13, which implies that N_a does not appear in the form shown in node $\langle t, 2\rangle$. ■

A uniqueness result corresponding to 5.8 is easy to establish; it requires the assumption that $N_a \neq N_b$.

Propositions 5.2, 5.8, 5.10, 5.13, and 5.14 give a detailed insight into the conditions under which the Needham-Schroeder-Lowe protocol achieves its authentication and secrecy goals.

6 Ideals and Honesty

We now introduce the concept of *ideal* to formulate general facts about the penetrator's capabilities.

6.1 Ideals

Definition 6.1 *If* $\mathsf{k} \subseteq \mathsf{K}$*, a* k*-ideal of* A *is a subset* I *of* A *such that for all* $h \in I$*,* $g \in \mathsf{A}$ *and* $K \in \mathsf{k}$

1. $h\,g, g\,h \in I$.
2. $\{h\}_K \in I$.

The smallest k*-ideal containing* h *is denoted* $I_{\mathsf{k}}[h]$.

It follows immediately from this definition and Definition 2.11 that $g \sqsubset h$ if and only if $h \in I_{\mathsf{K}}[g]$.

Definition 6.2 *If* $S \subseteq \mathsf{A}$*,* $I_{\mathsf{k}}[S]$ *is the smallest* k*-ideal containing* S.

The ideal structure is very simple:

Proposition 6.3 *If* $S \subseteq \mathsf{A}$*,* $I_{\mathsf{k}}[S] = \bigcup_{x \in S} I_{\mathsf{k}}[x]$.

Proof. The property of being a k-ideal is equivalent to closure under the mappings $x \mapsto x\,a$, $x \mapsto a\,x$ and $x \mapsto \{x\}_k$ for $k \in \mathsf{k}$. Thus the union of k-ideals is a k-ideal. Thus $\bigcup_{x \in S} I_{\mathsf{k}}[x]$ is a k-ideal which contains S. Clearly $\bigcup_{x \in S} I_{\mathsf{k}}[x] \subseteq I_{\mathsf{k}}[S]$. ■

Lemma 6.4 *Let* $S_0 = S$*,* $S_{i+1} = \{\{g\}_K : g \in I_\emptyset[S_i], K \in \mathsf{k}\}$*. Then* $I_{\mathsf{k}}[S] = \bigcup_i I_\emptyset[S_i]$.

Proof. By induction, $S_i \subseteq I_{\mathsf{k}}[S]$, so $\bigcup_i I_\emptyset[S_i] \subseteq I_{\mathsf{k}}[S]$. In the other direction, $\bigcup_i I_\emptyset[S_i]$ is clearly a k-ideal which contains S. ■

Definition 6.5 *A term is* simple *iff it is not of the form* $a\,b$ *for* $a, b \in \mathsf{A}$.

Alternatively, a term is simple iff it is either an element of T, an element of K or is of the form $\{h\}_K$.

Proposition 6.6 *Suppose* $K \in \mathsf{K}$*;* $S \subseteq \mathsf{A}$*; and for every* $s \in S$*,* s *is simple and is not of the form* $\{g\}_K$*. If* $\{h\}_K \in I_{\mathsf{k}}[S]$*, then* $h \in I_{\mathsf{k}}[S]$.

Note that S may contain a term $\{g\}_{K'}$ where $K' \neq K$ and g contains subterms encrypted in K.
Proof. Assume $K \in \mathsf{K}$, $\{h\}_K \in I_{\mathsf{k}}[S]$ and $h \notin I_{\mathsf{k}}[S]$. Let I' be the set difference $I_{\mathsf{k}}[S] \setminus \{\{h\}_K\}$. Clearly $S \subseteq I'$, since S does not contain anything encrypted with outermost key K. Moreover I' is a k-ideal: Since $I_{\mathsf{k}}[S]$ is already an ideal and $\{h\}_K$ is not of the form $a\,b$, I' clearly satisfies the join closure condition for ideals. If $\{h\}_K = \{h_1\}_{K'}$ for $h_1 \in I'$, then by Axiom 1 (free encryption), $h = h_1 \in I' \subseteq I_{\mathsf{k}}[S]$ a contradiction. Thus I' is an ideal which contains S. This contradicts the definition of $I_{\mathsf{k}}[S]$ as the smallest ideal which contains S. ■

Proposition 6.7 *Suppose* $K \in \mathsf{K}$*;* $S \subseteq \mathsf{A}$*; and every* $s \in S$ *is simple and is not of the form* $\{g\}_K$*. If* $\{h\}_K \in I_{\mathsf{k}}[S]$ *for* $K \in \mathsf{K}$*, then* $K \in \mathsf{k}$.

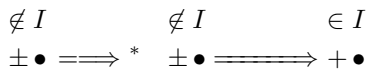


Figure 7: Entry Point for I

The proof is similar to the proof of Proposition 6.6.
PROOF. Assume $K \in \mathsf{K}$, $\{h\}_K \in I_{\mathsf{k}}[S]$ and $K \notin \mathsf{k}$. As in the preceding proposition, let $I' = I_{\mathsf{k}}[S] \setminus \{\{h\}_K\}$. For the same reason as before, $S \subseteq I'$ and I' satisfies the join closure condition for ideals. Moreover, by free encryption, $\{h\}_K$ is not of the form $\{h'\}_{K'}$ for any $K' \in \mathsf{k}$. Thus I' is an ideal which contains S. This contradicts the definition of $I_{\mathsf{k}}[S]$. ■

Proposition 6.8 *Suppose $S \subseteq \mathsf{A}$, and every $s \in S$ is simple. If $g\,h \in I_{\mathsf{k}}[S]$ then either $g \in I_{\mathsf{k}}[S]$ or $h \in I_{\mathsf{k}}[S]$.*

PROOF. In virtue of Lemma 6.4, $g\,h \in I_\emptyset[S_i]$ for some i. By Proposition 6.3, $g\,h \in I_\emptyset[x]$ for some $x \in S_i$. This x is simple, as either $i = 0$, in which case $S_i = S$, or else $i = j + 1$, in which case each $x \in S_i$ is of the form $\{h\}_K$, and hence simple. We claim either $g \in I_\emptyset[x]$ or $h \in I_\emptyset[x]$. Otherwise, consider the set $I_\emptyset[x] \setminus \{g\,h\}$. By the freeness assumption, it is an $\emptyset$-ideal which contains x, contradicting minimality. ■

6.2 Entry Points and Honesty

Recall from Definition 2.3, Clause 6, that a node n is an *entry point* for $I \subseteq \mathsf{A}$ if and only if $\mathrm{term}(n) = +t$ for some $t \in I$ and for all nodes n' such that $n' \Rightarrow^+ n$, $\mathrm{term}(n') \notin I$, as shown in Figure 7.

Proposition 6.9 *Suppose $\mathcal{C}$ is a bundle over A. If m is minimal in $\{m \in \mathcal{C} : term(m) \in I\}$, then m is an entry point for I.*

PROOF. If $\mathrm{term}(m) = -h$, then by Definition 2.4 Clause 2, there is a node $m' \in \mathcal{C}$ with $\mathrm{term}(m') = +h$, violating minimality. If $m' \Rightarrow^+ m$ and $\mathrm{term}(m') \in I$, then using Definition 2.4 Clause 3 repeatedly, $m' \in \mathcal{C}$, again contradicting minimality. ■

Definition 6.10 *A set $I \subseteq A$ is* honest *relative to a bundle $\mathcal{C}$ if and only if whenever a penetrator node p is an entry point for I, p is an **M** node or a **K** node.*

Thus, I is honest relative to $\mathcal{C}$ if the penetrator can achieve entry into I only by a lucky guess: either he utters the right nonce or other text in a lucky **M** node, or he utters the right key in a lucky **K** node. He does not deduce it via his abilities to decrypt and encrypt, or to concatenate and separate.

6.3 More Bounds on the Penetrator

Our main theorem interrelates the structure of ideals with the possible cases for a penetrator strand.

Theorem 6.11 *Suppose $\mathcal{C}$ is a bundle over A; $S \subseteq \mathsf{T} \cup \mathsf{K}$; $\mathsf{k} \subseteq \mathsf{K}$; and $\mathsf{K} \subseteq S \cup \mathsf{k}^{-1}$. Then $I_{\mathsf{k}}[S]$ is honest.*

Proof. Let $I = I_{\mathsf{k}}[S]$. Because $I \cap \mathsf{K} = S \cap \mathsf{K}$, we may infer $\mathsf{K} \setminus I = \mathsf{K} \setminus S \subseteq \mathsf{k}^{-1}$. Also, since $S \subseteq \mathsf{T} \cup \mathsf{K}$, the set S contains nothing encrypted and no concatenations, so Propositions 6.8 and 6.6 can be applied.

Suppose m is a penetrator node and an entry point for I. We now consider the various kinds of strands on which a penetrator node can occur. By the definition of entry point, m cannot be on a strand of kind **F** or kind **T**. Consider now the remaining cases:

C. m is on a strand with trace $\langle -g, -h, +h\, g\rangle$. Since $h\, g \in I$, by Proposition 6.8, one of g, h must be in I, contradicting the definition of entry point.

S. m is on a strand with trace $\langle -h\, g, +h, +g\rangle$. Since term$(m)$ must be positive, m is either the second or third node of the strand, so either $h \in I$ or $g \in I$. By the ideal property, $h\, g \in I$, contradicting the definition of entry point.

D. m belongs to a strand with trace $\langle -K_0^{-1}, -\{h\}_{K_0}, +h\rangle$. By the assumption that m is an entry point for I, $K_0^{-1} \notin I$. Hence, $K_0^{-1} \notin S$. However, $\mathsf{K} \subseteq S \cup \mathsf{k}^{-1}$. Therefore $K_0^{-1} \in \mathsf{k}^{-1}$, so $K_0 \in \mathsf{k}$. By the k-ideal property of I, $\{h\}_{K_0} \in I$, contradicting the definition of entry point.

E. m belongs to a strand with trace $\langle -K', -h, +\{h\}_{K'}\rangle$. By assumption $\{h\}_{K'} \in I$. By Proposition 6.6, $h \in I$, contradicting the definition of entry point.

The only remaining possibilities are that m is on a strand of kind **M** or of kind **K** as asserted. ■

In our analysis of Otway-Rees in Section 7, we use two corollaries of this main result. The first allows us to conclude (in some situations) that if a key that is not originally known to the penetrator is transmitted, then a regular (i.e. non-penetrator) node has provided the entry point.

Corollary 6.12 *Suppose $\mathcal{C}$ is a bundle, $\mathsf{K} = S \cup \mathsf{k}^{-1}$ and $S \cap \mathsf{K}_{\mathsf{P}} = \emptyset$. If there exists a node $m \in \mathcal{C}$ such that term$(m) \in I_{\mathsf{k}}[S]$, then there exists a regular node $n \in \mathcal{C}$ such that n is an entry point for $I_{\mathsf{k}}[S]$.*

Proof. Assume that no regular node is an entry point for $I_{\mathsf{k}}[S]$. By hypothesis, $\{n \in \mathcal{C} : \text{term}(n) \in I_{\mathsf{k}}[S]\}$ is non-empty and therefore contains a minimal element m. By Proposition 6.9, m is an entry point for $I_{\mathsf{k}}[S]$. m cannot be regular, and so must be a penetrator node. Theorem 6.11 implies m is either a penetrator node of kind **M** or of kind **K**.

However, since $\mathsf{K} = S \cup \mathsf{k}^{-1}$, $S \subseteq \mathsf{K}$. Hence $I_{\mathsf{k}}[S] \cap \mathsf{T} = \emptyset$, so m is not of kind **M**. Because $S \cap \mathsf{K_P} = \emptyset$, m is not of kind **K**. ■

Proposition 3.3 is a special case of this, in which $\{K\}$ is chosen as S and K is chosen as k.

The second corollary gives a condition under which encryption guarantees a non-penetrator origin.

Corollary 6.13 *Suppose $\mathcal{C}$ is a bundle; $\mathsf{K} = S \cup \mathsf{k}^{-1}$; $S \cap \mathsf{K_P} = \emptyset$; and no regular node $\in \mathcal{C}$ is an entry point for $I_{\mathsf{k}}[S]$. Then any term of the form $\{g\}_K$ for $K \in S$ does not originate on a penetrator strand.*

PROOF. By Corollary 6.12, for every node $m \in \mathcal{C}$, $\mathrm{term}(m) \notin I = I_{\mathsf{k}}[S]$. Suppose $t_1 = \{g\}_K$ for $K \in S$ originates on a penetrator strand m. By inspection, m cannot occur on a penetrator strand of kind **F**, **T**, **K**, **M**, **C** or **S**. Consider the remaining cases:

E. m occurs on a strand with trace $\langle -K_0, -h, +\{h\}_{K_0} \rangle$. Now $K_0 \notin I$ and so $K_0 \neq K$. Since $\{g\}_K \sqsubset \{h\}_{K_0}$, Proposition 2.12 implies $\{g\}_K \sqsubset h$, contradicting the definition of entry point.

D. m belongs to a strand with trace $\langle - {K_0}^{-1}, - \{h\}_{K_0}, +h \rangle$. If $\{g\}_K \sqsubset h$, then $\{g\}_K \sqsubset \{h\}_{K_0}$, contradicting the definition of entry point. ■

Although, as we will illustrate in the next section, these theorems about ideals are frequently quite useful, not all protocol correctness assertions fit this particular mold. In fact, the Needham-Schroeder-Lowe protocol is a counterexample. In Lemma 5.4, we proved that the minimal members of the set

$$S = \{n \in \mathcal{C} : N_b \sqsubset \mathrm{term}(n) \wedge \{N_a\, N_b\, B\}_{K_A} \not\sqsubset \mathrm{term}(n)\}$$

are regular nodes. This set S is not an ideal; instead, it is formed from the difference of two ideals:

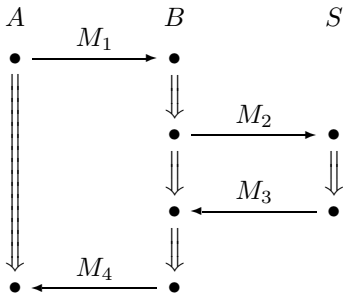
$$S = \{n \in \mathcal{C} : \mathrm{term}(n) \in I_{\mathsf{K}}[N_b] \setminus I_{\mathsf{K}}[\{N_a\, N_b\, B\}_{K_A}]\}$$

The crucial, authenticating step in which the initiator demonstrates his identity to the respondent is this one; if his private key is uncompromised only he can extract N_b and emit a term containing N_b but not $\{N_a\, N_b\, B\}_{K_A}$. Thus, in this case, we need to be able to reason about the entry points into this difference of ideals.

However, we have used these results about ideals to prove facts about the Yahalom protocol and (in [30]) the Neuman-Stubblebine protocol. Maneki has used corresponding results to prove a version of the TMN protocol [16]. Thus, they seem to be quite widely useful, especially to reason about shared secrets.

7 The Otway-Rees Protocol

In this section, we will illustrate the machinery of ideals and honesty by applying it to analyze the Otway-Rees protocol.



$M_1 = M\,A\,B\,\{N_a\,M\,A\,B\}_{K_{AS}}$

$M_2 = M\,A\,B\,\{N_a\,M\,A\,B\}_{K_{AS}}\,\{N_b\,M\,A\,B\}_{K_{BS}}$

$M_3 = M\,\{N_a\,K_{AB}\}_{K_{AS}}\,\{N_b\,K_{AB}\}_{K_{BS}}$

$M_4 = M\,\{N_a\,K_{AB}\}_{K_{AS}}$

Figure 8: Message Exchange in Otway-Rees

7.1 The Otway-Rees Protocol Itself

This protocol has three roles: initiator, responder, and server. The goal of the protocol is to authenticate initiator and responder mutually and to distribute a session key generated by a server. See Figure 8.

To provide a mathematical model of this protocol, we refine the algebra A in ways similar to those in Section 5:

- A set $\mathsf{T}_{\text{name}} \subseteq \mathsf{T}$ of names.
- A mapping $K : \mathsf{T}_{\text{name}} \to \mathsf{K}$. This is intended to denote the mapping which associates to each principal the key it shares with the server. In the literature on this protocol this mapping is usually written using subscripts: $K(A) = K_{AS}$. We assume the mapping $A \mapsto K_{AS}$ is injective. We also assume $K_{AS} = K_{AS}^{-1}$, i.e. that the protocol is using symmetric cryptography.

We will adopt some conventions on variables for the remainder of this section:

- Variables A, B range over T_{name};
- Variables K, K' range over K;
- Variables N, M (or the same letters decorated with subscripts) range over $\mathsf{T} \setminus \mathsf{T}_{\text{name}}$, i.e. those texts that are not names.

Other letters such as G and H range over all of A. We would emphasize that N_a is just a variable, having no reliable connection to A, whereas K_{AS} is the

result of applying the function K to the argument A. Thus, the latter reliably refers to the long term key shared between A and S.

Definition 7.1 *1.* $\mathrm{Init}[A, B, N, M, K]$ *is the set of strands* $s \in \Sigma$ *whose trace is*

$$\langle + M\ A\ B\ \{N\ M\ A\ B\}_{K_{AS}}, - M\ \{N\ K\}_{K_{AS}} \rangle$$

The principal associated with a strand $s \in \mathrm{Init}[A, B, N, M, K]$ *is* A.

2. $\mathrm{Resp}[A, B, N, M, K, H, H']$ *is defined when* $N \not\sqsubset H$*; its value then is the set of strands in* Σ *whose trace is*

$$\begin{aligned} \langle & - M\ A\ B\ H, \\ & + M\ A\ B\ H\ \{N\ M\ A\ B\}_{K_{BS}}, \\ & - M\ H'\ \{N\ K\}_{K_{BS}}, \\ & + M\ H' \rangle \end{aligned}$$

The principal associated with a strand $s \in \mathrm{Resp}[A, B, N, M, K, H, H']$ *is* B.

3. $\mathrm{Serv}[A, B, N_a, N_b, M, K]$ *is defined if* $K \notin \mathsf{K_P}$*,* $K \notin \{K_{AS} : A \in \mathsf{T}_{name}\}$ *and* $K = K^{-1}$*; its value then is the set of strands in* Σ *whose trace is:*

$$\begin{aligned} \langle & - M\ A\ B\ \{N_a\ M\ A\ B\}_{K_{AS}}\ \{N_b\ M\ A\ B\}_{K_{BS}}, \\ & + M\ \{N_a\ K\}_{K_{AS}}\ \{N_b\ K\}_{K_{BS}} \rangle \end{aligned}$$

The principal associated with a strand $s \in \mathrm{Serv}[A, B, N_a, N_b, M, K]$ *is a fixed server* S_0.

In the definition of the responder traces, the condition $N \not\sqsubset H$ implies N originates on each strand in $\mathrm{Resp}[A, B, N, M, K, H, H']$. A protocol participant cannot inspect the contents of H to enforce this condition, since under normal operation of the protocol, H is ciphertext inaccessible to the participant. Rather, we are assuming that this condition is enforced by a probabilistic mechanism.

We sometimes find it convenient to use the $*$ to indicate union over some indices. Thus for instance $\mathrm{Resp}[A, B, N_b, M, K, *, *] =$

$$\bigcup_{H,H'} \mathrm{Resp}[A, B, N_b, M, K, H, H']$$

In the extreme case in which all the parameters are $*$, we omit them; for instance, $\mathrm{Init} = \mathrm{Init}[*, *, *, *, *]$.

Lemma 7.2 *The sets* $\mathrm{Serv}, \mathrm{Init}, \mathrm{Resp}$ *are pairwise disjoint.*

Proof. It suffices to prove the sets of traces are disjoint. Originator traces begin with a positive term. The second term of of a responder trace has width (Definition 2.10) at least 4, whereas for a server trace the width is exactly 3.

Definition 7.3 *An* Otway-Rees strand space *is an infiltrated strand space* Σ *such that* $\Sigma = \text{Serv} \cup \text{Init} \cup \text{Resp} \cup \mathcal{P}$.

This union is disjoint, by Lemma 7.2 and the observation that $\mathcal{P}$ contains no strands of the same form as Serv $\cup$ Init $\cup$ Resp.

Fix an Otway-Rees strand space Σ over A.

7.2 Otway-Rees: Secrecy

We first prove that session keys distributed by the server cannot be disclosed unless the penetrator possesses one of the long-term keys used in the run. We show that a session key can never occur in a form in which it is not encrypted by the participants' long-term keys.

Theorem 7.4 *Suppose* $\mathcal{C}$ *is a bundle in* Σ*;* $A, B \in \mathsf{T}_{name}$*;* K *is uniquely originating;* $K_{AS}, K_{BS} \notin \mathsf{K_P}$*; and* $s_{serv} \in \text{Serv}[A, B, N_a, N_b, M, K]$*. Let* $S = \{K_{AS}, K_{BS}, K\}$ *and* $\mathsf{k} = \mathsf{K} \setminus S$.

For every node $m \in \mathcal{C}$*,* term$(m) \notin I_{\mathsf{k}}[K]$.

Proof. By Proposition 6.3, it suffices to prove the stronger statement that for every node m, term$(m) \notin I_{\mathsf{k}}[S]$. Since $S \cap \mathsf{K_P} = \emptyset$, $\mathsf{k} = \mathsf{k}^{-1}$ and $\mathsf{K} = \mathsf{k} \cup S$, by Corollary 6.12 it suffices to show that no regular node m is an entry point for $I_{\mathsf{k}}[S]$.

We will argue by contradiction and assume m is a regular node which is an entry point for $I_{\mathsf{k}}[S]$. Since m is an entry point for $I_{\mathsf{k}}[S]$, by the definitions, it follows that term(m) is an element of $I_{\mathsf{k}}[S]$. By 6.3, this implies that one of the keys K, K_{AS}, K_{BS} is a subterm of term(m). Now no regular node contains any key of the form K_{XS} as a subterm. In fact, the only keys which occur as subterms of term(m) (for m regular) are the session keys emanating from a server. But by the definition of server strands, the set of such keys is disjoint from the set of keys of the form K_{XS}. It thus follows K must be a subterm of term(m).

If m is a positive regular node on a strand s, then $K \sqsubset \text{term}(m)$ implies either:

1. $s \in \text{Serv}$ and $m = \langle s, 2 \rangle$, in which case K is the session key of s; or
2. $s \in \text{Resp}[*, *, *, *, *, H, *]$, $m = \langle s, 2 \rangle$, and $K \sqsubset H$.

In case 2, m is not an entry point for $I_{\mathsf{k}}[S]$, because $H \sqsubset \langle s, 1 \rangle$, which is a preceding negative node.

So consider case 1. By the unique origination of K, $s = s_{\text{serv}}$, so term$(m) = M\, \{N_a\, K\}_{K_{AS}}\, \{N_b\, K\}_{K_{BS}}$. By Proposition 6.8, either

1. $M \in I_{\mathsf{k}}[S]$, or
2. $\{N_a\, K\}_{K_{AS}} \in I_{\mathsf{k}}[S]$, or
3. $\{N_b\, K\}_{K_{BS}} \in I_{\mathsf{k}}[S]$.

But the first is impossible by Definition 6.1; the second and third are impossible by Proposition 6.7. ■

7.3 Otway-Rees: Authentication

In this subsection we will prove the authentication guarantees that Otway-Rees provides to its initiator and responder. It is also possible to prove that the protocol provides authentication guarantees to the server, but we will not do so here. We first "import" the consequence of Corollary 6.13 that we will need.

Proposition 7.5 *Consider a bundle $\mathcal{C}$ in Σ. Suppose $X \in \mathsf{T}_{name}$ is such that $K_{XS} \notin \mathsf{K_P}$. Then no term of the form $\{g\}_{K_{XS}}$ for $X \in \mathsf{T}_{name}$ can originate on a penetrator node in $\mathcal{C}$.*

Proof. Let $S = \{K_{XS}\}$ and $\mathsf{k} = \mathsf{K}$. To apply Corollary 6.13, we must check that no regular node is an entry point for $I_{\mathsf{K}}[S]$, or equivalently, that K_{XS} does not originate on any regular node.

A key K originates on a regular node only if it is a session key K originating on a server strand $s \in \mathrm{Serv}[*, *, *, *, K, *, *]$. However, by the definition of Serv, the session key K is never a long term key K_{XS}.

Hence, we may apply Corollary 6.13 to $I_{\mathsf{K}}[S]$, so any term $\{g\}_{K_{XS}}$ can only originate on a regular node. ■

Proposition 7.6 *If $\{H\}_{K_{XS}}$ originates on a regular strand s, then:*

1. *If $s \in \mathrm{Serv}$, then $H = N\,K$ for $N \in \mathsf{A}$ and $K \in \mathsf{K}$.*
2. *If $s \in \mathrm{Init}$, then $H = N\,M\,X\,C$ for $N \in \mathsf{A}$, $M \in \mathsf{T}$ and $X, C \in \mathsf{T}_{name}$.*
3. *If $s \in \mathrm{Resp}$, then $H = N\,M\,C\,X$ for $N \in \mathsf{A}$, $M \in \mathsf{T}$ and $X, C \in \mathsf{T}_{name}$.*

Proof. By the definition of originating (Definition 2.3, Clause 7), if the term $\{H\}_{K_{XS}}$ originates on m, then m is positive.

If $s \in \mathrm{Init}$ then $m = \langle s, 1\rangle$. Thus $\mathrm{term}(m)$ is of the form $M\,A\,B\,\{N\,M\,A\,B\}_{K_{AS}}$. The only encrypted subterm of this term, $\{N\,M\,A\,B\}_{K_{AS}}$, is of form 2.

If $s \in \mathrm{Resp}$, then the positive nodes of s are $\langle s, 2\rangle$ and $\langle s, 4\rangle$. The encrypted subterms of $\langle s, 2\rangle$ have plaintext of forms 2 and 3 respectively, while the encrypted subterm of $\langle s, 4\rangle$ has form 1 which is not originating.

A similar argument holds if $s \in \mathrm{Serv}$. ■

Corollary 7.7 *Suppose s is a regular strand of Σ.*

1. *If $\{N\,K\}_{K_{XS}}$ originates on s, then either*
   - $s \in \mathrm{Serv}[A, X, N', N, M, K]$
   - $s \in \mathrm{Serv}[X, B, N, N', M, K]$

   for some A, B, N', M. In either case the term originates on the node $\langle s, 2\rangle$ and K originates on s.
2. *If $\{N\,M\,A\,B\}_{K_{AS}}$ originates on s, with $A \neq B$ then*
   - $s \in \mathrm{Init}[A, B, N, M, K]$

for some K. The term originates on the node $\langle s, 1\rangle$ and N originates on s.

3. *If $\{N\, M\, A\, B\}_{K_{BS}}$ originates on s, with $A \neq B$ then*

   - $s \in \mathrm{Resp}[A, B, N, M, K, H, H']$

   for some K, H, H'. The term originates on the node $\langle s, 2\rangle$ and N originates on s.

PROOF. Since s is regular, $s \in \mathrm{Serv} \cup \mathrm{Init} \cup \mathrm{Resp}$. Apply Proposition 7.6. ■

7.3.1 Initiator's Guarantee

The following theorem asserts that if a bundle contains a strand $s \in \mathrm{Init}$, then under reasonable assumptions, there are regular strands $s_{\mathrm{resp}} \in \mathrm{Resp}$ and $s_{\mathrm{serv}} \in \mathrm{Serv}$ which agree on the initiator, responder, and M values.

Theorem 7.8 *Suppose $\mathcal{C}$ is a bundle in Σ; $A \neq B$; N_a is uniquely originating in $\mathcal{C}$; and $K_{AS}, K_{BS} \notin \mathsf{K_P}$.*

If $s \in \mathrm{Init}[A, B, N_a, M, K]$ has $\mathcal{C}$-height 2, then for some $N_b \in \mathsf{T}$ there are regular strands

- *$s_{resp} \in \mathrm{Resp}[A, B, N_b, M, *, *, *]$ of $\mathcal{C}$-height at least 2.*
- *$s_{serv} \in \mathrm{Serv}[A, B, N_a, N_b, M, K]$ of $\mathcal{C}$-height 2.*

PROOF. The assumption of the theorem means

$$\langle +\ M\, A\, B\, \{N_a\, M\, A\, B\}_{K_{AS}},\\ -\ M\, \{N_a\, K\}_{K_{AS}} \rangle$$

is the $\mathcal{C}$-trace of a strand s.

Since $K_{AS} \notin \mathsf{K_P}$, by Proposition 7.5, $\{N_a\, K\}_{K_{AS}}$ originates on a regular node in $\mathcal{C}$. By Corollary 7.7, this node belongs to a strand s_{serv} which satisfies one of the conditions:

1. $s_{\mathrm{serv}} \in \mathrm{Serv}[A, X, N_a, N_b, M_1, K]$, or
2. $s_{\mathrm{serv}} \in \mathrm{Serv}[X, A, N_b, N_a, M_1, K]$

where $X \in \mathsf{T}_{\mathrm{name}}$, and $N_b, M_1 \in \mathsf{T}$. Since $\langle s_{\mathrm{serv}}, 2\rangle \in \mathcal{C}$, s_{serv} has $\mathcal{C}$-height 2.

If condition 1 holds, $\{N_a\, M_1\, A\, X\}_{K_{AS}} \sqsubset \mathrm{term}(\langle s_{\mathrm{serv}}, 1\rangle)$. By Proposition 7.5, $\{N_a\, M_1\, A\, X\}_{K_{AS}}$ originates on a regular strand s_1, and by Corollary 7.7, N_a originates on the same strand s_1. By the unique origination of N_a, $s = s_1$. Thus $M_1 = M$ and $X = B$, and $s_{\mathrm{serv}} \in \mathrm{Serv}[A, B, N_a, N_b, M, K]$.

By Proposition 7.5, $\{N_b\, M\, A\, B\}_{K_{BS}}$ originates on a regular node in $\mathcal{C}$. By Corollary 7.7, this node is the second on a strand $s_{\mathrm{resp}} \in \mathrm{Resp}[A, B, N_b, M, *, *, *]$. Since $\langle s_{\mathrm{resp}}, 2\rangle \in \mathcal{C}$, it follows s_{resp} has $\mathcal{C}$-height at least 2.

Suppose that condition 2 holds instead. Then $\{N_a\, M_1\, X\, A\}_{K_{AS}}$ is a subterm of term$(\langle s_{\text{serv}}, 1\rangle)$. By Proposition 7.5, $\{N_a\, M_1\, X\, A\}_{K_{AS}}$ originates on a regular strand s_1, and by Corollary 7.7, N_a originates on the same strand s_1. By the unique origination of N_a, $s = s_1$. Hence by Corollary 7.7, $X = A = B$, contradicting an assumption. ■

Remarks. Even though the intention of the protocol design is to have B receive $H = \{N_a\, M\, A\, B\}_{K_{AS}}$ from A there is no way to prevent a penetrator from replacing $\{N_a\, M\, A\, B\}_{K_{AS}}$ with garbage. Moreover a penetrator can prevent the output of the server from reaching B. Thus, we cannot show that B has $\mathcal{C}$-height > 2.

7.3.2 Responder's Guarantee

The responder can rest assured that if a bundle contains a strand $s \in \text{Resp}$, then under familiar assumptions there are regular strands $s_{\text{init}} \in \text{Init}$ and $s_{\text{serv}} \in \text{Serv}$ which agree on the initiator, responder, and M values. Its proof is very similar to the proof of Theorem 7.8.

Theorem 7.9 *Suppose $\mathcal{C}$ is a bundle in Σ; $A \neq B$; N_b is uniquely originating in $\mathcal{C}$; and $K_{AS}, K_{BS} \notin \mathsf{K_P}$.*

If $s \in \text{Resp}[A, B, N_b, M, K, H, H']$ has $\mathcal{C}$-height at least 3, then there are regular strands

- *$s_{init} \in \text{Init}[A, B, *, M, *]$ of $\mathcal{C}$-height at least 1.*
- *$s_{serv} \in \text{Serv}[A, B, *, N_b, M, K]$ of $\mathcal{C}$-height 2.*

Proof. The assumption of the proposition means the $\mathcal{C}$-trace of s contains at least:

$$\begin{aligned}\langle &- M\, A\, B\, H,\\ &+ M\, A\, B\, H\, \{N_b\, M\, A\, B\}_{K_{BS}},\\ &- M\, H'\, \{N_b\, K\}_{K_{BS}}\rangle\end{aligned}$$

Since $K_{BS} \notin \mathsf{K_P}$, by Proposition 7.5, $\{N_b\, K\}_{K_{BS}}$ originates on a regular node in $\mathcal{C}$. By Corollary 7.7, this node belongs to a strand s_{serv} which satisfies one of the following two conditions:

1. $s_{\text{serv}} \in \text{Serv}[X, B, N_b, N, M_1, K]$, or
2. $s_{\text{serv}} \in \text{Serv}[B, X, N, N_b, M_1, K]$

where $X \in \mathsf{T}_{\text{name}}$, and $N, M_1 \in \mathsf{T}$. Since $\langle s_{\text{serv}}, 2\rangle \in \mathcal{C}$, s_{serv} has $\mathcal{C}$-height 2.

If condition 1 holds, then $\{N_b\, M_1\, X\, B\}_{K_{BS}} \sqsubset \langle s_{\text{serv}}, 1\rangle$. By Proposition 7.5, $\{N_b\, M_1\, X\, B\}_{K_{BS}}$ originates on a regular strand s_1, and by Corollary 7.7, N_b originates on the strand s_1. By the unique origination of N_b, $s = s_1$. Hence $X = B = A$, contradicting an assumption.

Suppose that condition 2 holds instead. Again, $\{N_b\, M_1\, X\, B\}_{K_{BS}} \sqsubset \langle s_{\text{serv}}, 1\rangle$. By Proposition 7.5, $\{N_b\, M_1\, X\, B\}_{K_{BS}}$ originates on a regular strand s_1, and by

Corollary 7.7, N_b originates on the strand s_1. By the unique origination of N_b, $s = s_1$. Thus, $M_1 = M$ and $X = A$, and $s_{\text{serv}} \in \text{Serv}[A, B, N, N_b, M, K]$.

By Proposition 7.5, $\{N\, M\, A\, B\}_{K_{AS}}$ originates on a regular node in $\mathcal{C}$. By Corollary 7.7, this node belongs to a strand $s_{\text{init}} \in \text{Init}[A, B, N, M, *]$. s_{init} has $\mathcal{C}$-height at least 1. ■

Remarks. As in the previous theorem there are some penetrator behaviors that cannot be prevented. For instance the penetrator could take the encrypted session key that B is supposed to pass on to A and throw it away. Hence, we can not show that the initiator's strand has $\mathcal{C}$-height > 1.

More significantly, the above argument makes vividly clear why the BAN modification to Otway-Rees [3, Section 4] might fail, as was shown by Mao and Boyd [17]. In that modification the nonce N_b is outside the encryption. Though it is still true, when condition 2 holds, that the term $\{M_1\, X\, B\}_{K_{BS}}$ originates on a regular strand s_1, this term does not contain N_b. Hence, s_1 may not be an origination point for N_b, and we can no longer conclude that $s_1 = s$.

Indeed, the BAN modification also requires a weakening of Theorem 7.8, as we can no longer infer that the responder and the server strands will agree on the responder's nonce N_b.

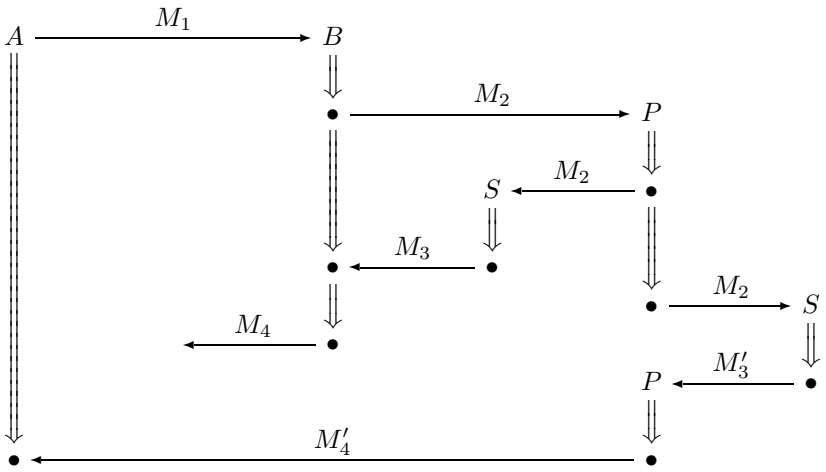
7.3.3 A Missing Guarantee

The authentication theorems do not establish something that we had expected they would, namely that if a bundle $\mathcal{C}$ contains complete initiator and responder strands, then they agree on the session key distributed.

That is, one cannot strengthen Theorem 7.8 by replacing the asterisk by K to obtain $s_{\text{resp}} \in \text{Resp}[A, B, N_b, M, K, *, *]$. Nor can one strengthen Theorem 7.9 by replacing an asterisk by K to obtain $s_{\text{init}} \in \text{Init}[A, B, *, M, K]$. The reason is that there is a counterexample, a bundle $\mathcal{C}$ (illustrated in Figure 9) in which each player has a complete strand in $\mathcal{C}$, and they agree on A, B, and M, but they do not agree on K.

Although this protocol has been studied very carefully in the past (e.g. [3, 17, 24]), this weakness appears not to be explicit in the literature. For instance, the BAN authors [3, Section 4] suggest the contrary, that the two participants at the end each believe of a (single) key K_{AB} that it is a good shared key for A and B. The authors comment that neither principal can know whether the key is known to the other, but this is presumably because neither principal knows whether the other has completed his strand. Paulson [24], despite his very detailed argument, does not comment on this point.

Presumably this protocol weakness is not serious, as no shared keys are disclosed. However, it serves to illustrate the subtleties that remain poorly understood even in very familiar protocols.



Where

1. $M_1 = M\ A\ B\ \{N_a\ M\ A\ B\}_{K_{AS}}$.
2. $M_2 = M\ A\ B\ \{N_a\ M\ A\ B\}_{K_{AS}}\ \{N_b\ M\ A\ B\}_{K_{BS}}$.
3. $M_3 = M\ \{N_a\ K_{AB}\}_{K_{AS}}\ \{N_b\ K_{AB}\}_{K_{BS}}$.
4. $M'_3 = M\ \{N_a\ K'_{AB}\}_{K_{AS}}\ \{N_b\ K'_{AB}\}_{K_{BS}}$.
5. $M_4 = M\ \{N_a\ K_{AB}\}_{K_{AS}}$.
6. $M'_4 = M\ \{N_a\ K'_{AB}\}_{K_{AS}}$.

Figure 9: An Otway-Rees Weakness: Mismatched Keys

8 Conclusion

8.1 Discussion

In this paper, we have developed the idea of strand spaces for proving the correctness of cryptographic protocols. We have also developed some algebraic machinery—the notion of ideal—to supplement the strand space idea, and to prove general, re-usable bounds on the penetrator (Section 6). Our methods exploit two partial orderings, namely the subterm relation $\sqsubset$ between terms and the $\preceq$ relation between nodes. Inductive characteristics of these orderings are formulated via the notion of an ideal in the case of $\sqsubset$, and via a least element principle in the case of $\preceq$.

Our work is closely related to Paulson's inductive approach [24, 23, 25]. Paulson models a protocol as a set of rules for extending a sequence of events; some of these rules represent actions by legitimate participants, while others represent actions by the penetrator. A sequence of events generated by these rules corresponds roughly to a bundle. Paulson expresses authentication goals and secrecy goals as properties of these sequences, which he can then prove by induction on the way that the sequence is generated. The general-purpose theorem-proving system Isabelle [22] provides mechanical support for the reasoning.

By contrast, our approach uses a partially ordered structure, the bundle. As we mentioned, Lemma 2.7 is in effect an induction principle on the partial order $\preceq_{\mathcal{C}}$. The nodes in the bundle are organized into strands. Naturally, every bundle may be linearized into an event sequence in at least one way, while any event sequence determines a bundle.

However, we think there are two advantages to our approach.

- The bundle contains exactly the causally relevant information. There is no ordering relation between two nodes unless the causality determined by the basic relations $\rightarrow$ and $\Rightarrow$ requires one, and this simplifies inductive arguments.

- The strand captures a great deal of information. A particular strand may be known to have nodes in a bundle (e.g. because a value originates uniquely on it). From this we can identify the whole sequence of relevant actions for that participant, which aids in isolating the exact agreement properties the protocol satisfies. We believe this is why our results are somewhat sharper than others in the literature.

The strand space framework can also be used in other ways, apart from being used simply to prove a protocol correct. For instance, it could be used to give an alternate semantics for belief logics, whether applied to cryptographic protocols [3, 2] or distributed systems more broadly [10], in contrast to the more usual semantical approaches based on sequences of events or states. The localization that the notion of strand offers should help to refine and sharpen such models. Alternatively, results about authentication protocols proved in a strand space context can be imported into the more usual linear models by linearizing the bundles.

The specific algebraic properties we have considered are still elementary. They are applied under assumptions (such as "free encryption") that are still restrictive. However, as recent work suggests [16], it is likely that the approach can be used in the case of message algebras with less restrictive assumptions.

8.2 The Goals of Protocols

We have proved a variety of specific results about protocols; each of them formalizes a protocol goal. They include:

- Secrecy results for both protocols (Theorems 5.10, 5.13, and 7.4);
- Agreement properties for Needham-Schroeder-Lowe (Theorems 5.2 and 5.14) and Otway-Rees (Theorems 7.8 and 7.9);
- A uniqueness ("injectiveness") result for Needham-Schroeder-Lowe (Theorem 5.8), but not for Otway-Rees (Figure 9 gives a counterexample).

However, each of these results is a little different. Theorems 5.2 and 5.14 differ in which keys must be uncompromised and in the $\mathcal{C}$-height of the corresponding strand. Theorem 5.2 has the additional assumption that the initiator and the respondent use different nonces.

Thus, despite the fact that our proof methods are fairly tightly organized around induction (Lemma 2.7) and the bounds on the penetrator (Theorems 3.3 and 6.11, along with the corollaries of the latter), the protocol goals to be proved are hand-crafted so as to fit each specific protocol and to express optimally what it achieves. To what extent is this variability a real fact of life, and to what extent is it a drawback of our method?

In the following paragraphs we will argue that each of secrecy and authentication should not be thought of as a single property, dictating a rigidly formulated theorem to be proved about protocols. Rather, each is a logical form that suggests a kind of theorem to be considered. One of the main things to be learnt from protocol analysis is exactly which theorems of these kinds are true of a protocol.

What is Secrecy? One kind of correctness property for protocols is secrecy. This means that some value, usually a key, never falls into the wrong hands. Exactly what this means for a particular protocol may vary, although there is a logical form common to all.

A secrecy theorem, stating that none of the values in the set S can be disclosed, takes the form of a sentence $\mathcal{S}(\phi, S)$:

$$\forall \mathcal{C}\ \forall s\ \forall n\ .\ \phi(s)\ \wedge\ n \in \mathcal{C}\ \Longrightarrow\ \mathrm{term}(n) \notin I_{\mathsf{k}}[S]$$

in which we write k for $(\mathsf{K} \setminus S)^{-1}$, and we let $\mathcal{C}$ range over bundles, s over strands, and n over nodes. The hypothesis $\phi(s)$ frequently contains assumptions about keys used in s being uncompromised; it frequently contains assumptions that

values sent or received are uniquely originating; and it frequently stipulates what kind of strand s is, such as an initiator strand or a responder strand.

To see that $\mathcal{S}(\phi, S)$ really expresses secrecy, first suppose that $K \in \mathsf{K_P} \cap S$. Then there exist bundles $\mathcal{C}$ containing penetrator **K**-nodes emitting K. So unless the antecedent of the conditional is always false, $\mathcal{S}(\phi, S)$ cannot be true.

Moreover, for a value in S to be disclosed, it is necessary that it occur either unencrypted or else encrypted only using keys whose inverse the penetrator could obtain. The set k includes all keys whose inverse the penetrator can obtain before obtaining any value in S. Hence, so long as $\mathrm{term}(n) \notin I_{\mathsf{k}}[S]$ for all $n \in \mathcal{C}$, the penetrator cannot derive any term containing a value in S in a form that he can decrypt.

This form of secrecy for a set of data values is formally different than the corresponding notion discussed in section 4. The notion proposed there states that a set S is secret if nothing in S is ever received or transmitted in the clear by anybody. The formally stronger notion proposed here says that S is secret if nothing in $I_{\mathsf{k}}[S]$ is ever received or transmitted by anybody. However, if for some node $n \in \mathcal{C}$, $\mathrm{term}(n) \in I_{\mathsf{k}}[S]$, then by adding penetrator strands, we can construct a bundle $\mathcal{C}'$ such that that for some penetrator node $n \in \mathcal{C}'$, $\mathrm{term}(n) \in S$.

We prefer the form we have given here, as an explication of secrecy, because it matches the strong methods we have developed in Section 6 for proving secrecy results. Although for expository reasons Theorems 5.10 and 5.13 were not stated in the form $\mathcal{S}(\phi, S)$, they amount to theorems of this form taking $S = \{K_A, K_B, N\}$ where $N = N_b$ or $N = N_a$ respectively.

What is Authentication? According to one view, an authentication goal is one that establishes the identity of one principal ("entity") to another principal. Unfortunately, this concept of authentication seems vague and naïve: It does not say which actions may be traced back to that principal, or during what period of time its identity remains unchanged.

The authentication properties we have proved above are motivated by the desire—shared with other authors [15, 26, 32]—to replace this naïve view of authentication by a more meaningful concept. When we prove an authentication property, we prove that a bundle contains a regular strand of a given $\mathcal{C}$-height (subject to some assumptions). Thus we have shown that a well defined sequence of events has been performed by the principal associated with that regular strand. A result of this form seems to us to extract a core of precise meaning from the traditional notion of entity authentication.

All of our authentication results have a logical form in common. They all take the form of a sentence $\mathcal{A}(i, j, \phi, \psi)$:

$$\forall \mathcal{C}\ \forall s\ \exists s'\ .\ \mathcal{C}\text{-height}(s) = i\ \wedge\ \phi(s)\ \Longrightarrow\ \mathcal{C}\text{-height}(s') = j\ \wedge\ \psi(s, s')$$

The hypothesis $\phi(s)$ typically says what type of strand s is, such as an initiator strand or a responder strand. It typically contains assumptions that certain values are uniquely originating and that certain keys are uncompromised. It

may also assume that values are distinct (e.g. $N_a \neq N_b$ in Theorem 5.2, and $A \neq B$ in Theorem 7.8). The conclusion $\psi(s, s')$ typically says what type of strand s' is, such as a responder strand or an initiator strand, and always entails that s' is regular. It will also say which data values must be shared between s and s'. It may also impose uniqueness by asserting that if s'' is any strand satisfying the previous conditions, then $s'' = s'$.

Analyzing the authentication guarantee offered by a protocol means in effect determining which values of i, j, ϕ, and ψ yield true instances of $\mathcal{A}(i, j, \phi, \psi)$.

What actually happens in a specific authentication protocol may be complex. It is a good question to ask, of a given protocol, "What kind of authentication are we talking about?" That way we can decide whether the protocol is capable of achieving some goal with a real-world significance. And this, we think, is the purpose of protocol analysis.

Acknowledgements. The National Security Agency supported this work through US Army CECOM contract DAAB 07-96-C-E601. This paper integrates the contents of [29, 27].

We are grateful to Sylvan Pinsky, Al Maneki, and their colleagues at NSA for support, encouragement, and discussions. Shim Berkovits, Marion Michaud, and John Vasak patiently helped us improve the presentation. Peter Ryan taught us a great deal about the field, and provided the initial impetus for doing the work. Martín Abadi and several anonymous referees made shrewd and useful comments.

References

[1] Martín Abadi and Andrew D. Gordon. Reasoning about cryptographic protocols in the spi calculus. In *CONCUR 97*, Lecture Notes in Computer Science, pages 59–73. Springer-Verlag, July 1997.

[2] Martìn Abadi and Mark R. Tuttle. A semantics for a logic of authentication. In *Proceedings of the 10th ACM Symposium on Principles of Distributed Computing*, pages 201–216, August 1991.

[3] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *Proceedings of the Royal Society*, Series A, 426(1871):233–271, December 1989. Also appeared as SRC Research Report 39 and, in a shortened form, in ACM Transactions on Computer Systems 8, 1 (February 1990), 18-36.

[4] Ulf Carlsen. Cryptographic protocol flaws. In *Proceedings 7th IEEE Computer Security Foundations Workshop*, pages 192–200. IEEE Computer Society, 1994.

[5] John Clark and Jeremy Jacob. On the security of recent protocols. *Information Processing Letters*, 56(3):151–155, November 1995.

[6] Dorothy Denning and G. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8), August 1981.

[7] D. Dolev and A. Yao. On the security of public-key protocols. *IEEE Transactions on Information Theory*, 29:198–208, 1983.

[8] Shimon Even, Oded Goldreich, and Adi Shamir. On the security of ping-pong protocols when implemented using the RSA. In *Advances in Cryptology*—CRYPTO *'85*, LNCS, pages 58–72. Springer Verlag, 1985.

[9] Riccardo Focardi and Roberto Gorrieri. The compositional security checker: A tool for the verification of information flow security properties. *IEEE Transactions on Software Engineering*, 23(9), September 1997.

[10] Joseph Y. Halpern. Reasoning about knowledge: A survey. In D. Gabbay, C. J. Hogger, and J. A Robinson, editors, *Handbook of Logic in Artificial Intelligence and Logic Programming*, volume 4, pages 1–34. Oxford University Press, 1995.

[11] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice-Hall International, Englewood Cliffs, New Jersey, 1985.

[12] Gavin Lowe. An attack on the Needham-Schroeder public key authentication protocol. *Information Processing Letters*, 56(3):131–136, November 1995.

[13] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proceeedings of* TACAS, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer Verlag, 1996.

[14] Gavin Lowe. Casper: A compiler for the analysis of security protocols. In *10th Computer Security Foundations Workshop Proceedings*, pages 18–30. IEEE Computer Society Press, 1997.

[15] Gavin Lowe. A heirarchy of authentication specifications. In *10th Computer Security Foundations Workshop Proceedings*, pages 31–43. IEEE Computer Society Press, 1997.

[16] Al Maneki. Honest functions and their application to the analysis of cryptographic protocols. In *Proceedings of the 12th Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1999.

[17] Wenbo Mao and Colin Boyd. Towards the formal analysis of security protocols. In *Proceedings of the Computer Security Foundations Workshop VI*, pages 147–158. IEEE Computer Society Press, 1993.

[18] Will Marrero, Edmund Clarke, and Somesh Jha. A model checker for authentication protocols. In Cathy Meadows and Hilary Orman, editors, *Proceedings of the DIMACS Workshop on Design and Verification of Security Protocols*. DIMACS, Rutgers University, September 1997.

[19] Judy H. Moore. Protocol failures in cryptosystems. *Proceedings of the IEEE*, 76(5), May 1988.

[20] Roger Needham and Michael Schroeder. Using encryption for authentication in large networks of computers. *Communications of the ACM*, 21(12), December 1978.

[21] Sarvar Patel. Number theoretic attacks on secure password schemes. In *Proceedings of the 1997 IEEE Symposium on Security and Privacy*, pages 236–247. IEEE Computer Society Press, May 1997.

[22] L. C. Paulson. *Isabelle: A Generic Theorem Prover*. Number 828 in LNCS. Springer Verlag, 1994.

[23] Lawrence C. Paulson. Mechanized proofs of a recursive authentication protocol. In *10th IEEE Computer Security Foundations Workshop*, pages 84–94. IEEE Computer Society Press, 1997.

[24] Lawrence C. Paulson. Proving properties of security protocols by induction. In *10th IEEE Computer Security Foundations Workshop*, pages 70–83. IEEE Computer Society Press, 1997.

[25] Lawrence C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 1998. Also Report 443, Cambridge University Computer Lab.

[26] A. W. Roscoe. Intensional specifications of security protocols. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pages 28–38, 1996.

[27] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Honest ideals on strand spaces. In *Proceedings of the 11th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1998.

[28] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand space pictures. Presented at the LICS Workshop on Formal Methods and Security Protocols, June 1998.

[29] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Why is a security protocol correct? In *1998 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 1998.

[30] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Mixing protocols. In *Proceedings of the 12th IEEE Computer Security Foundations Workshop*. IEEE Computer Society Press, June 1999.

[31] Steve Schneider. Verifying authentication protocols with CSP. In *Proceedings of the 10th IEEE Computer Security Foundations Workshop*, pages 3–17. IEEE Computer Society Press, 1997.

[32] Thomas Y. C. Woo and Simon S. Lam. Verifying authentication protocols: Methodology and example. In *Proc. Int. Conference on Network Protocols*, October 1993.