A Bound on Attacks on Authentication Protocols

Scott D. Stoller*
Computer Science Dept., SUNY at Stony Brook, Stony Brook, NY 11794-4400

December 18, 2001

Abstract

Authentication protocols are designed to work correctly in the presence of an adversary that can prompt honest principals to engage in an unbounded number of concurrent executions of the protocol. The amount of local state used in a single execution of a typical authentication protocol is bounded. This suggests that there is a bound on the number of protocol executions that could be useful in attacks. Such bounds clarify the nature of attacks on and provide a rigorous basis for automated verification of authentication protocols. This paper establishes such a bound for a large class of protocols, which contains versions of some well-known authentication protocols, including the Yahalom, Otway-Rees, and Needham-Schroeder-Lowe protocols.

1 Introduction

Many protocols are designed to work correctly in the presence of an adversary—hereafter called a penetrator—that can prompt honest principals to engage in an unbounded number of concurrent executions of the protocol. This includes some protocols for authentication (including key establishment) [DvOW92, MvOV97], Byzantine Agreement [GLR95], and electronic payment [OPT97]. This paper focuses on authentication. Authentication protocols should satisfy at least two kinds of correctness requirements: *secrecy*, which states that certain values are not obtained by the penetrator, and *agreement*, which states, *e.g.*, that a principal's conclusion about the identity of a principal with whom it is communicating is never incorrect.

Authentication protocols are short and look deceptively simple, but numerous flawed or weak protocols have been published; some examples are described in [DS81, BAN90, WL94, AN95, AN96, Low96, Aba97, LR97, THG98c]. This attests to the importance of rigorous verification.

Allowing an unbounded number of concurrent protocol executions makes the number of reachable states unbounded, so automated verification using state-space exploration is not directly applicable. The case studies in [MCF87, Ros95, HTWW96, DK97, LR97, MMS97, MCJ97, MSS98, Bol98, DNL99] show that state-space exploration of authentication protocols and similar cryptographic protocols is feasible when small upper bounds are imposed on the size of messages and the number of protocol executions. However, in most of those case studies, the bounds were not rigorously justified, so the results do not prove correctness

*The author gratefully acknowledges the support of NSF under Grant CCR-9876058 and the support of ONR under Grants N00014-99-1-0358 and N00014-01-1-0109. This work was started while the author was at Indiana University in Bloomington.

of the protocols. Reduction theorems are needed, which show that if a protocol is correct in a system with certain bounds on these parameters, then the protocol is correct in the unbounded system as well.

1.1 Contribution

This paper presents a reduction for a large class of protocols. Our results are formulated in the strand space model [THG98c, THG98a, THG98b].[1] In this model, a regular strand can be regarded as a thread that runs the program corresponding to one role (*e.g.*, initiator or responder) of the protocol and then terminates; thus, a regular strand corresponds to one execution of one role. Our reduction imposes three significant restrictions on protocols.

Shallow ciphertext restriction: the protocol does not use nested ciphertexts.[2] This is easily checked by static analysis of the program, so we call it a *static restriction*.

Bounded Support Restriction (BSR): in every history (*i.e.*, every possible behavior) of the system, each regular strand depends on at most a given number of regular strands. Our notion of dependence between nodes, embodied in the definition of support, is a variant of Lamport's happened-before relation [Lam78], modified to treat nonces and session keys—collectively called *generated values*, or *genvals* for short—appropriately. For example, if a genval g generated on a strand s_1 appears in messages received by strand s_2 but only in contexts in which it could be replaced with a value generated by the penetrator, then g's presence in those messages does not cause s_2 to depend on s_1. Intuitively, correct authentication protocols are designed to involve only a small number of participants and hence typically satisfy BSR.

Revealed Genval Restriction (RGR): every genval revealed to the penetrator is revealed "directly"; roughly, this means that the penetrator needs to perform at most one decryption to obtain the genval.

It seems difficult to develop static analyses to check BSR and RGR, so we call them *dynamic restrictions* and propose to check them during state-space exploration. With static restrictions alone, it seems difficult to find restrictions that are both strong enough to justify a reduction and weak enough to be satisfied by well-known protocols. Dynamic restrictions and correctness requirements are properties of histories; the difference is that restrictions come with the reduction, while correctness requirements come with the protocol.

In order to check the dynamic restrictions during state-space exploration, we need reductions for them as well as for the correctness requirements. We prove: if a protocol satisfies the dynamic restrictions and correctness requirements when appropriate bounds are imposed on the number of regular strands in a history, then the protocol also satisfies the dynamic restrictions and correctness requirements without those bounds. We prove the contrapositive of this statement, by supposing that some history of the unbounded system violates a dynamic restriction or correctness requirement and constructing a history violating the same property and containing at most the specified number of regular strands. That history is constructed by starting from an earliest node (in the strand space model, events are usually called "nodes") that causes a violation of the property and finding the set of nodes on which that node depends. Roughly speaking, that set of nodes, augmented with appropriate actions by the penetrator, is the desired history.

[1] Our results are relatively model-independent. We proved a similar reduction in a variant of Woo and Lam's semantic model of authentication protocols [WL93].

[2] This restriction can be relaxed, as discussed at the end of Section 4.

1.2 Related Work

Few reductions applicable to authentication protocols are known. Most existing techniques for automated analysis of systems with unbounded numbers of processes, such as [CGJ95, KM95, EN96, AJ98], are not applicable to authentication protocols, because they assume the set of values (equivalently, the set of local states of each process) is independent of the number of processes, whereas authentication protocols generate fresh nonces and session keys, so the set of values grows as the number of processes (equivalently, the number of regular strands) increases.

Dolev and Yao's algorithms for verifying secrecy requirements of cryptographic protocols [DY83] are efficient but limited. They do not handle agreement requirements, and they apply to a severely restricted class of protocols that excludes almost all well-known authentication protocols (*e.g.*, the Otway-Rees [OR87] and Yahalom [BAN90] protocols) and is strictly included in the class of protocols handled by our reduction.

Roscoe and Broadfoot use data-independence techniques to bound the number of nonces that could be useful in attacks [Ros98, RB99]. That result assumes that each trustworthy principal participates in at most a given number of protocol executions at a time. Our reduction does not require such assumptions; indeed, its purpose is to justify such assumptions.

Lowe proved a reduction for a corrected version of the Needham-Schroeder public-key protocol, hereafter called the NSL protocol [Low96]. Lowe subsequently proved a reduction for a class of protocols [Low99]. The main limitations of the reduction in [Low99] are that it does not handle agreement requirements or known-key attacks[3] and does not apply to the Otway-Rees, Yahalom, and NSL protocols, due to various restrictions.

The reduction embodied in Theorems 2 and 3 handles secrecy and agreement requirements, allows known-key attacks, and applies to some well-known protocols, including the Otway-Rees, Yahalom, and NSL protocols, after the Otway-Rees and Yahalom protocols have been modified slightly to eliminate forwarding of ciphertexts. Our reduction provides an upper bound on the number of regular strands that could be useful in attacks. From this bound and the shallow ciphertext restriction, it is easy to obtain upper bounds on the number of nonces and the number of penetrator strands that could be useful in attacks.

Athena [Son99] is a model checker based on symbolic backwards state-space exploration. Athena can efficiently verify many authentication protocols. Work on Athena does not provide much general insight into the number of protocol executions that could be useful in attacks, because there is no discussion in [Son99] of conditions under which Athena terminates.

2 Model of Authentication Protocols

We adopt the strand space model [THG98c], with minor modifications. We introduce simple languages for expressing authentication protocols and correctness requirements.

[3] *Known-key attacks* are scenarios in which genvals generated in a previous protocol execution are somehow (not necessarily through a weakness of the protocol) obtained by the penetrator and used in attacks on subsequent protocol executions [MvOV97, p. 496].

2.1 Term, Signed Term, and Trace

The set *Prim* of *primitive terms* is the union of the following sets, which are assumed to be disjoint:[4]

- *Text*: a set of miscellaneous non-cryptographic values. *Name* is a distinguished subset of *Text* containing names of principals.
- *Nonce*: a set of nonces.
- $\mathit{Key}_{\mathit{sess}}$: a set of session keys.
- $\mathit{Key}_{\mathit{sym}} = \{\mathit{key}(x,y) \mid x,y \in \mathit{Name}\}$: a set of long-term symmetric keys. Informally, $\mathit{key}(x,y)$ is intended to be shared by x and y.
- $\mathit{Key}_{\mathit{asym}} = \{\mathit{pubkey}(x) \mid x \in \mathit{Name}\} \cup \{\mathit{pvtkey}(x) \mid x \in \mathit{Name}\}$: a set of long-term asymmetric keys. $\mathit{pubkey}(x)$ and $\mathit{pvtkey}(x)$ represent x's public and private keys, respectively.

Let $\mathit{Key} = \mathit{Key}_{\mathit{sym}} \cup \mathit{Key}_{\mathit{asym}} \cup \mathit{Key}_{\mathit{sess}}$.

The set *Term* of terms is defined inductively as follows.

1. $\mathit{Prim} \subseteq \mathit{Term}$.
2. If $t \in \mathit{Term}$ and $k \in \mathit{Key}$, then $\mathit{encr}(t,k) \in \mathit{Term}$. $\{t\}_k$ represents encryption of t with k and is usually written as $\{t\}_k$.
3. If $t_1 \in \mathit{Term}$ and $t_2 \in \mathit{Term}$, then $\mathit{pair}(t_1,t_2) \in \mathit{Term}$. $\mathit{pair}(t_1,t_2)$ is usually written as $t_1 \cdot t_2$.

The function $\mathrm{inv} \in \mathit{Key} \to \mathit{Key}$ maps each key to its inverse: decrypting $\{t\}_k$ with $\mathrm{inv}(k)$ yields t. For a symmetric key k, $\mathrm{inv}(k) = k$. We usually write $\mathrm{inv}(k)$ as k^{-1}. We assume perfect encryption, *i.e.*, $\{t\}_k = \{t'\}_{k'}$ iff $t = t'$ and $k = k'$. Distinct primitive terms are assumed to represent distinct values (*e.g.*, $\mathit{key}(A,B)$ and $\mathit{key}(A,S)$ represent different keys). Recall that elements of $\mathit{Nonce} \cup \mathit{Key}_{\mathit{sess}}$ are called *generated values*, or *genvals* for short.

For $S \subseteq \mathit{Term}$, the set of genvals that occur in S is

$$\mathrm{genvals}(S) = \{g \in \mathit{Nonce} \cup \mathit{Key}_{\mathit{sess}} \mid \exists t \in S : g \text{ occurs in } t\}. \tag{1}$$

A *ciphertext* is a term whose outermost operator is *encr*. A term t' *occurs in the clear* in a term t if there is an occurrence of t' in t that is not in the scope of *encr*. For example, in the term $\{A\}_{k_1} \cdot \{\{B\}_{k_1}\}_{k_2}$, the term $\{A\}_{k_1}$ occurs in the clear; the term $\{B\}_{k_1}$ does not.

Let $\mathrm{size}(S)$ denote the size of a set S. Let $\mathrm{dom}(f)$ denote the domain of a function f. A sequence is a function from a finite prefix of the natural numbers to elements. Let $\mathrm{len}(\sigma)$ denote the length of a sequence σ. $\langle\!\langle a, b, \ldots \rangle\!\rangle$ denotes a sequence σ with $\sigma(0) = a$, $\sigma(1) = b$, and so on.

A *signed term* is $+t$ or $-t$, where t is a term.[5] Positive and negative terms represent sending and receiving messages, respectively. Let $\pm \mathit{Term}$ denote the set of signed terms. For a signed term $t \in \pm \mathit{Term}$, the absolute value of t, denoted $\mathrm{abs}(t)$, is t without its sign; for example $\mathrm{abs}(-A) = A$. For $S \subseteq \pm \mathit{Term}$, let

[4] Allowing these sets to be different for different systems presents no difficulties. It would require only that we include the desired sets in the description of each system. To avoid clutter, we do not do this.

[5] This piece of strand space terminology can be confusing. A signed term is *not* a term with a digital signature.

$\mathrm{abs}(S) = \{\mathrm{abs}(t) \mid t \in S\}$. For brevity, we often refer to signed terms as terms and treat them as terms, for instance as having subterms.

A *trace* is a finite sequence of signed terms. Let $(\pm \mathit{Term})^*$ denote the set of traces.

2.2 Strand Space

A *strand space* is a function $tr \in \mathrm{dom}(tr) \to (\pm \mathit{Term})^*$, where $\mathrm{dom}(tr)$ is an arbitrary set whose elements are called *strands*.[6]

A *node* of tr is a pair $\langle s, i\rangle$ with $s \in \mathrm{dom}(tr)$ and $0 \leq i < \mathrm{len}(tr(s))$. Let $\mathcal{N}_{tr}$ denote the set of nodes of tr. We say that node $\langle s, i\rangle$ is on strand s. Let $\mathrm{nodes}_{tr}(s)$ denote the set of nodes on strand s in tr. Let $\mathrm{strand}(\langle s, i\rangle) = s$, $\mathrm{index}(\langle s, i\rangle) = i$, and $\mathrm{term}_{tr}(\langle s, i\rangle) = tr(s)(i)$. For $S \subseteq \mathcal{N}_{tr}$, let $\mathrm{strand}(S) = \{\mathrm{strand}(n) \mid n \in S\}$ and $\mathrm{term}_{tr}(S) = \{\mathrm{term}_{tr}(n) \mid n \in S\}$. If $\mathrm{term}_{tr}(n)$ is positive (or negative), we say that n is positive (or negative).

The local dependence relation on nodes is defined by: $n_1 \Rightarrow n_2$ iff $\mathrm{strand}(n_1) = \mathrm{strand}(n_2)$ and $\mathrm{index}(n_2) = \mathrm{index}(n_1) + 1$.

A term t *originates* from a node $\langle s, i\rangle$ in tr iff $\langle s, i\rangle$ is positive, t is a subterm of $\mathrm{term}_{tr}(\langle s, i\rangle)$,[7] and t is not a subterm of $\mathrm{term}_{tr}(\langle s, 0\rangle), \mathrm{term}_{tr}(\langle s, 1\rangle), \ldots, \mathrm{term}_{tr}(\langle s, i-1\rangle)$.

A term t *uniquely originates* from a node n in tr iff t originates from n in tr and not from any other node in tr. This is the strand space way of expressing freshness of genvals.

For symbols subscripted by the strand space, we elide the subscript when the strand space is evident from context.

2.3 Role

Let *Param* be a set of parameters. The set *pTerm* of *parameterized terms* is defined in the same way as *Term* in Section 2.1 but with the addition of a fourth item: $\mathit{Param} \subseteq \mathit{pTerm}$.

A *role* is a sequence of signed parameterized terms, with a type—*i.e.*, a set of allowed values—associated with each parameter, and with a subset of the parameters designated as uniquely-originated.[8] Informally, parameters that represent genvals generated by the role (and hence that first occur in the role in a positive term) are so designated, to indicate that values of those parameters must be uniquely-originated. In examples, uniquely-originated parameters are underlined in the parameter list. Let $r.x$ denote parameter x of role r.

For example, consider the NSL protocol [Low96].

$$\begin{array}{l} 1.\ A \to B : \{n_A \cdot A\}_{pubkey(B)} \\ 2.\ B \to A : \{n_A \cdot n_B \cdot B\}_{pubkey(A)} \\ 3.\ A \to B : \{n_B\}_{pubkey(B)} \end{array} \tag{2}$$

[6] Thayer *et al.* [THG98c] use the symbol Σ for $\mathrm{dom}(tr)$.

[7] We use the standard notion of subterm, rather than the modified subterm relation $\sqsubseteq$ defined in [THG98c], in which k is not necessarily a subterm of $\{t\}_k$. Our version induces a stronger notion of uniquely-originates that corresponds more closely to the standard notion of freshness.

[8] This is essentially the same notion of role as in [Son99] and [CDL+00].

The roles for the initiator and responder are

$$\begin{array}{l} \mathrm{Init}_{NSL}(i : \mathit{Name} \setminus \{P\}, r : \mathit{Name}, \underline{ni} : \mathit{Nonce}, nr : \mathit{Nonce}) = \\ \langle\!\langle +\{ni \cdot i\}_{pubkey(r)}, \\ \quad -\{ni \cdot nr \cdot r\}_{pubkey(i)}, \\ \quad +\{nr\}_{pubkey(r)} \rangle\!\rangle \\ \mathrm{Resp}_{NSL}(i : \mathit{Name}, r : \mathit{Name} \setminus \{P\}, ni : \mathit{Nonce}, \underline{nr} : \mathit{Nonce}) = \\ \langle\!\langle -\{ni \cdot i\}_{pubkey(r)}, \\ \quad +\{ni \cdot nr \cdot r\}_{pubkey(i)}, \\ \quad -\{nr\}_{pubkey(r)} \rangle\!\rangle. \end{array} \tag{3}$$

In both roles, parameters i and r hold the names of the initiator and responder, respectively. We exclude P from the type of $\mathrm{Init}_{NSL}.i$ and $\mathrm{Resp}_{NSL}.r$, because we interpret P as the name of a dishonest principal (the penetrator), and we interpret $\mathrm{Init}_{NSL}.i$ and $\mathrm{Resp}_{NSL}.r$ as the name of the principal executing the role, and all actions of the penetrator are represented by traces for penetrator roles, described in Section 2.5. Parameters $\mathrm{Init}_{NSL}.ni$ and $\mathrm{Resp}_{NSL}.nr$ are uniquely-originated, because they represent nonces generated by their respective roles.

A role is *well-formed* if

1. The type of each parameter is Key_{sess}, Nonce, Key, or a subset of Text.

2. The type of each uniquely-originated parameter is Key_{sess} or Nonce.

3. Genvals do not occur in roles, except in the parameter types.

4. For every role r, for every parameter x of r of type Key_{sess} or Nonce, x is uniquely-originated iff the first occurrence of x in r is in a positive term.

Hereafter, all roles are assumed to be well-formed unless explicitly noted otherwise. The only non-well-formed roles we consider are Msg in Section 2.5 and Src in Section 2.7.

The first well-formedness condition is violated by roles that receive and forward ciphertexts without decrypting them. Typically, roles that forward ciphertexts do not encrypt the forwarded ciphertexts, so modifying the protocol to eliminate such forwarding has no impact on the correctness of the protocol, so for our purposes, it suffices to analyze the modified protocol. This transformation is also used in [Low99, RB99, HL99].

A *trace for role r* is a prefix of a trace obtained by substituting for each parameter x of r a term in the type of x. For example, two traces for Init_{NSL} are

$$\begin{array}{rcl} \sigma_0 & = & \langle\!\langle +\{ni_0 \cdot B\}_{pubkey(A)} \rangle\!\rangle \\ \sigma_1 & = & \langle\!\langle +\{ni_0 \cdot A\}_{pubkey(B)}, -\{ni_0 \cdot nr_0 \cdot B\}_{pubkey(A)}, +\{nr_0\}_{pubkey(B)} \rangle\!\rangle. \end{array}$$

The requirement that parameters be instantiated with terms of the specified types is sometimes called the *strong typing assumption*. This assumption is common in protocol analysis, but ensuring that it provides a reasonable abstraction of a given implementation is non-trivial.

$$\begin{array}{l}
\mathrm{Init}_Y(i : \mathit{Name} \setminus \{P\},\, r : \mathit{Name},\, \underline{ni} : \mathit{Nonce},\, nr : \mathit{Nonce},\, k : \mathit{Key}_{sess}) = \\
\quad \langle\!\langle +i \cdot ni, \\
\qquad -\{r \cdot k \cdot ni \cdot nr\}_{key(i,S)}, \\
\qquad +\{nr\}_k \rangle\!\rangle
\end{array}$$

$$\begin{array}{l}
\mathrm{Resp}_Y(i : \mathit{Name},\, r : \mathit{Name} \setminus \{P\},\, ni : \mathit{Nonce},\, \underline{nr} : \mathit{Nonce},\, k : \mathit{Key}_{sess}) = \\
\quad \langle\!\langle -i \cdot ni, \\
\qquad +r \cdot \{i \cdot ni \cdot nr\}_{key(r,S)}, \\
\qquad -\{i \cdot k\}_{key(r,S)} \\
\qquad -\{nr\}_k \rangle\!\rangle
\end{array}$$

$$\begin{array}{l}
\mathrm{Srvr}_Y(i : \mathit{Name},\, r : \mathit{Name},\, ni : \mathit{Nonce},\, nr : \mathit{Nonce},\, \underline{k} : \mathit{Key}_{sess}) = \\
\quad \langle\!\langle -r \cdot \{i \cdot ni \cdot nr\}_{key(r,S)}, \\
\qquad +\{r \cdot k \cdot ni \cdot nr\}_{key(i,S)}, \\
\qquad +\{i \cdot k\}_{key(r,S)} \rangle\!\rangle
\end{array}$$

Figure 1: Yahalom protocol Π_Y. Forwarding of ciphertexts has been eliminated.

A role r and a trace σ for r uniquely determine a mapping, denoted $\mathit{args}(r, \sigma)$, from the set of parameters of r that appear in $r(0), r(1), \ldots, r(\mathrm{len}(\sigma) - 1)$ to *Term*. For example, $\mathrm{dom}(\mathit{args}(\mathrm{Init}_{NSL}, \sigma_0)) = \{i, r, ni\}$, $\mathit{args}(\mathrm{Init}_{NSL}, \sigma_0)(i) = B$, $\mathrm{dom}(\mathit{args}(\mathrm{Init}_{NSL}, \sigma_1)) = \{i, r, ni, nr\}$, and $\mathit{args}(\mathrm{Init}_{NSL}, \sigma_1)(i) = A$.

2.4 Protocol

A *protocol* is a set of roles. For example, the NSL protocol is $\Pi_{NSL} = \{\mathrm{Init}_{NSL}, \mathrm{Resp}_{NSL}\}$, where the roles are defined in (3).

For an example of a symmetric-key protocol, consider the Yahalom protocol [BAN90].

$$\begin{array}{l}
1.\ A \to B : A \cdot n_A \\
2.\ B \to S : B \cdot \{A \cdot n_A \cdot n_B\}_{key(B,S)} \\
3.\ S \to A : \{B \cdot k \cdot n_A \cdot n_B\}_{key(A,S)} \cdot \{A \cdot k\}_{key(B,S)} \\
4.\ A \to B : \{A \cdot k\}_{key(B,S)} \cdot \{n_B\}_k
\end{array} \tag{4}$$

The forwarding of $\{A \cdot k\}_{key(B,S)}$ by A is easily eliminated without affecting the correctness of the protocol. The modified protocol Π_Y appears in Figure 1; the roles are well-formed. In the original version of the Yahalom protocol, the third and fourth messages of Resp are combined, as are the second and third messages of Srvr. We split these messages so that the protocol can run to completion by itself. Without the splitting, the protocol can run to completion only with help from the penetrator (to split and combine pairs); this is a side-effect of elimination of forwarding of ciphertexts.

As another example, consider the Otway-Rees protocol [OR87].

$$\begin{array}{l}
1.\ A \to B : m \cdot A \cdot B \cdot \{n_A \cdot m \cdot A \cdot B\}_{key(A,S)} \\
2.\ B \to S : m \cdot A \cdot B \cdot \{n_A \cdot m \cdot A \cdot B\}_{key(A,S)} \cdot \{n_B \cdot m \cdot A \cdot B\}_{key(B,S)} \\
3.\ S \to B : m \cdot \{n_A \cdot k\}_{key(A,S)} \cdot \{n_B \cdot k\}_{key(B,S)} \\
4.\ B \to A : m \cdot \{n_A \cdot k\}_{key(A,S)}
\end{array} \tag{5}$$

After elimination of forwarding of ciphertexts, it can be expressed as the protocol Π_{OR} in Figure 2.

$$\begin{aligned}
&\mathrm{Init}_{OR}(i : \mathit{Name} \setminus \{P\},\ r : \mathit{Name},\ \underline{ni} : \mathit{Nonce},\ \underline{m} : \mathit{Nonce},\ k : \mathit{Key}_{sess}) = \\
&\quad \langle\!\langle +m\cdot i\cdot r\cdot \{ni\cdot m\cdot i\cdot r\}_{key(i,S)}, \\
&\quad\ \ -m\cdot \{ni\cdot k\}_{key(i,S)}\rangle\!\rangle \\
&\mathrm{Resp}_{OR}(i : \mathit{Name},\ r : \mathit{Name} \setminus \{P\},\ \underline{nr} : \mathit{Nonce},\ m : \mathit{Nonce},\ k : \mathit{Key}_{sess}) = \\
&\quad \langle\!\langle -m\cdot i\cdot r, \\
&\quad\ \ +m\cdot i\cdot r\cdot \{nr\cdot m\cdot i\cdot r\}_{key(r,S)}, \\
&\quad\ \ -m\cdot \{nr\cdot k\}_{key(r,S)}\rangle\!\rangle \\
&\mathrm{Srvr}_{OR}(i : \mathit{Name},\ r : \mathit{Name},\ ni : \mathit{Nonce},\ nr : \mathit{Nonce},\ m : \mathit{Nonce},\ \underline{k} : \mathit{Key}_{sess}) = \\
&\quad \langle\!\langle -m\cdot i\cdot r\cdot \{ni\cdot m\cdot i\cdot r\}_{key(i,S)}, \\
&\quad\ \ -m\cdot i\cdot r\cdot \{nr\cdot m\cdot i\cdot r\}_{key(r,S)} \\
&\quad\ \ +m\cdot \{ni\cdot k\}_{key(i,S)}, \\
&\quad\ \ +m\cdot \{nr\cdot k\}_{key(r,S)}\rangle\!\rangle
\end{aligned}$$

Figure 2: Otway-Rees Protocol Π_{OR}. Forwarding of ciphertexts has been eliminated.

There is no constraint on the relationship between the roles in a protocol, so one can consider a protocol that contains roles from multiple "protocols" (in the usual sense of a set of roles designed to work together); strand spaces for such protocols are sometimes called *mixed strand spaces* [THG99].

2.5 Penetrator

We consider a penetrator with roughly the same capabilities as in [THG98c]. A *penetrator model* for a protocol Π is a pair $\langle \mathit{pik}, \mathit{compr}\rangle$, where $\mathit{pik} \subseteq \mathit{Term}$ and compr is a compromised trace policy for Π, as defined below.

2.5.1 Penetrator's Initial Knowledge and Penetrator Roles

The penetrator model is parameterized by a set $\mathit{pik} \subseteq \mathit{Term}$, called the *penetrator's initial knowledge.* Typically, we assume there is a single dishonest principal, named P, and take $\mathit{pik} \supseteq \mathit{pik}_{keys}$, where

$$\begin{aligned}\mathit{pik}_{keys} = \{\mathit{pvtkey}(P)\} \cup \{\mathit{pubkey}(x) \mid x \in \mathit{Name}\} \\ \cup \{\mathit{key}(P,x), \mathit{key}(x,P) \mid x \in \mathit{Name} \setminus \{P\}\}.\end{aligned} \tag{6}$$

Known-key attacks are modeled by including in pik the absolute values of terms appearing in some executions of the protocol and the genvals generated during those executions. For example, for the NSL protocol, one might take the penetrator's initial knowledge pik_{NSL} to include pik_{keys}, absolute values of the three terms in a trace $\mathrm{Init}_{NSL}(A, B, n_0, n_1)$, absolute values of the three terms in a trace $\mathrm{Resp}_{NSL}(A, B, n_0, n_1)$ (which happen to be the same as the terms in the trace for Init_{NSL}), the genvals n_0 and n_1, and analogous terms from an execution in which B is the initiator and A is the responder. pik_{OR} and pik_{Y} are defined similarly.

$\Pi_P(\mathit{pik})$, the set of *penetrator roles* for a penetrator with initial knowledge pik, contains the following roles.

$$\begin{aligned}
&\mathrm{Msg}(x : \mathit{Text} \cup \mathit{Nonce} \cup \mathit{Key}_{sess} \cup \mathit{pik}) = \langle\!\langle +x \rangle\!\rangle\\
&\mathrm{Pair}(x_1 : \mathit{Term}, x_2 : \mathit{Term}) = \langle\!\langle -x_1,\ -x_2,\ +x_1 \cdot x_2 \rangle\!\rangle\\
&\mathrm{Sep}_1(x_1 : \mathit{Term}, x_2 : \mathit{Term}) = \langle\!\langle -x_1 \cdot x_2,\ +x_1 \rangle\!\rangle\\
&\mathrm{Sep}_2(x_1 : \mathit{Term}, x_2 : \mathit{Term}) = \langle\!\langle -x_1 \cdot x_2,\ +x_2 \rangle\!\rangle\\
&\mathrm{Enc}(k : \mathit{Key}, x : \mathit{Term}) = \langle\!\langle -k,\ -x,\ +\{x\}_k \rangle\!\rangle\\
&\mathrm{Dec}(k : \mathit{Key}, x : \mathit{Term}) = \langle\!\langle -k^{-1},\ -\{x\}_k,\ +x \rangle\!\rangle
\end{aligned} \tag{7}$$

The Msg role is instantiated only with terms known to, or originated by, the penetrator. Msg is not well-formed, because, *e.g.*, *pik* may contain non-primitive terms. No parameters of penetrator roles are uniquely-originated. For convenience, we assume $\Pi \cap \Pi_P(\mathit{pik}) = \emptyset$ for all protocols Π and all *pik*.

2.5.2 Compromised-Trace Policy

Informally, a compromised trace is a trace running the protocol with the penetrator as a partner. For example, in the Yahalom protocol, a trace σ for Resp_Y with $\mathit{args}(\mathrm{Resp}_Y, \sigma)(i) = P$ is compromised. We formalize this notion in terms of a function that provides a general way of indicating when the penetrator is involved in a protocol execution.

Let params(Π) denote the set of parameters of roles of Π. For example, $\mathrm{params}(\Pi_{NSL}) = \{r.x \mid r \in \Pi_{NSL} \wedge x \in \{i, r, ni, nr\}\}$.

Let $\mathit{Set}(S)$ denote the powerset of a set S.

A *compromised-trace policy* for a protocol Π is a function $\mathit{compr} \in \mathrm{params}(\Pi) \to \mathit{Set}(\mathit{Text} \cup \mathit{Key}_{sym} \cup \mathit{Key}_{asym})$. For example,

$$\mathit{compr}_{NSL}(x) = \begin{cases} \{P\} & \text{if } x \in \{\mathrm{Init}_{NSL}.r, \mathrm{Resp}_{NSL}.i\}\\ \emptyset & \text{otherwise} \end{cases} \tag{8}$$

$$\mathit{compr}_{Y}(x) = \begin{cases} \{P\} & \text{if } x \in \{\mathrm{Init}_{Y}.r, \mathrm{Resp}_{Y}.i, \mathrm{Srvr}_Y.i, \mathrm{Srvr}_Y.r\}\\ \emptyset & \text{otherwise.} \end{cases} \tag{9}$$

compr_{OR} is similar to compr_Y, except with Init_Y replaced with Init_{OR}, *etc.*

A trace σ for a role r is *compromised* with respect to a compromised-trace policy *compr* if there exists $x \in \mathrm{dom}(\mathit{args}(r, \sigma))$ such that $\mathit{args}(r, \sigma)(x) \in \mathit{compr}(r.x)$. If a trace is not compromised, we say that it is *uncompromised.*

2.6 System and History

A *system* is a pair $\langle \Pi, \mathit{pen} \rangle$, where Π is a protocol and *pen* is a penetrator model for Π. For example,

$$\begin{aligned}
\mathcal{M}_{NSL} &= \langle \Pi_{NSL}, \langle \mathit{pik}_{NSL}, \mathit{compr}_{NSL} \rangle \rangle\\
\mathcal{M}_{Y} &= \langle \Pi_{Y}, \langle \mathit{pik}_{Y}, \mathit{compr}_{Y} \rangle \rangle\\
\mathcal{M}_{OR} &= \langle \Pi_{OR}, \langle \mathit{pik}_{OR}, \mathit{compr}_{OR} \rangle \rangle.
\end{aligned} \tag{10}$$

A *history* of a system $\langle \Pi, \langle \mathit{pik}, \mathit{compr} \rangle \rangle$ is a tuple $h = \langle \mathit{tr}, \to, \mathit{role} \rangle$, where *tr* is a strand space, $\to$ is a binary relation on $\mathcal{N}_{tr}$, and $\mathit{role} \in \mathrm{dom}(\mathit{tr}) \to (\Pi \cup \Pi_P(\mathit{pik}))$ such that

$$\begin{tikzcd}
\underline{s_I : \mathrm{Init}_{NSL}} & \underline{s_R : \mathrm{Resp}_{NSL}} \\
+\{ni_0 \cdot A\}_{pubkey(B)} \arrow[r] \arrow[d, Rightarrow] & -\{ni_0 \cdot A\}_{pubkey(B)} \arrow[d, Rightarrow] \\
-\{ni_0 \cdot nr_0 \cdot B\}_{pubkey(A)} \arrow[d, Rightarrow] & +\{ni_0 \cdot nr_0 \cdot B\}_{pubkey(A)} \arrow[l] \arrow[d, Rightarrow] \\
+\{nr_0\}_{pubkey(B)} \arrow[r] & -\{nr_0\}_{pubkey(B)}
\end{tikzcd}$$

Figure 3: A history of $\mathcal{M}_{NSL}$. The underlined text above each trace has the form *strand* : *role*. Vertical and horizontal arrows represent ⇒ and →, respectively.

1. For all $n_1, n_2 \in \mathcal{N}_{tr}$, if $n_1 \to n_2$, then there exists $t \in \mathit{Term}$ such that $\mathrm{term}_{tr}(n_1) = +t$ and $\mathrm{term}_{tr}(n_2) = -t$. This represents that n_1 sends t, and n_2 receives t.

2. For all $n_1 \in \mathcal{N}_{tr}$, if $\mathrm{term}_{tr}(n_1)$ is negative, then there exists exactly one $n_2 \in \mathcal{N}_{tr}$ such that $n_2 \to n_1$.

3. $\preceq_h$ is well-founded (*i.e.*, does not have infinite descending chains) and acyclic, where $\preceq_h$ is the reflexive and transitive closure of $(\to \cup \Rightarrow)$. Note that $\preceq_{tr}$ is a partial order, originally defined by Lamport, who called it happened-before [Lam78].

4. For all $s \in \mathrm{dom}(tr)$, $tr(s)$ is a trace for $\mathit{role}(s)$

5. For all $s \in \mathrm{dom}(tr)$, for all $x \in \mathrm{dom}(\mathit{args}(\mathit{role}(s), tr(s)))$, if parameter x is uniquely-originated and $tr(s)$ is uncompromised with respect to compr, then $\mathit{args}(\mathit{role}(s), tr(s))(x)$ is not in $\mathrm{genvals}(\mathit{pik})$ and uniquely originates from $\langle s, i\rangle$, where i is the index of the first term in r that contains x.[9]

Let $\mathrm{Hist}(\mathcal{M})$ denote the set of histories of a system $\mathcal{M}$.

We say that a strand s in a history $\langle tr, \to, \mathit{role}\rangle$ of a system $\langle \Pi, \langle \mathit{pik}, \mathit{compr}\rangle\rangle$ is compromised iff trace $tr(s)$ for $\mathit{role}(s)$ is compromised with respect to compr.

In condition 5, it would be reasonable to omit the antecedent that s is uncompromised; in other words, it would be reasonable to require that values of uniquely-originated parameters of compromised strands are uniquely-originated. Including this antecedent simplifies the reduction, because dealing with uniquely-originated (fresh) genvals is the trickiest aspect (*cf.* the first paragraph of Section 1.2), so life is easier when fewer genvals are required to be uniquely-originated. Including this antecedent is safe in the sense that it provides a slightly more hostile environment for the protocol (*e.g.*, because the penetrator has immediate access to genvals that originate from compromised strands) and therefore a slightly stricter notion of correctness of a system (correctness requirements are discussed in Section 2.8).

If $\mathit{role}(s) \in \Pi$, then s is called a *regular strand* for $\mathit{role}(s)$. If $\mathit{role}(s) \in \Pi_P(\mathit{pik})$, then s is called a *penetrator strand* for $\mathit{role}(s)$. Nodes on regular strands are called *regular nodes*; nodes on penetrator strands are called *penetrator nodes*.

A history of $\mathcal{M}_{NSL}$ is illustrated in Figure 3. It contains no penetrator strands; an example of a history containing penetrator strands appears in Figure 4 (see Section 3.1).

[9] The requirement that $\mathit{args}(\mathit{role}(s), tr(s))(x) \notin \mathrm{genvals}(\mathit{pik})$ is discussed in Appendix A.

For convenience, we sometimes use a history instead of a strand space as a subscript. For example, for a history $h = \langle \mathit{tr}, \rightarrow, \mathit{role}\rangle$, we sometimes write $\mathcal{N}_h$ when we mean $\mathcal{N}_{\mathit{tr}}$.

The set of predecessors of a node n in a history h is $\mathrm{preds}_h(n) = \{n' \in \mathcal{N}_h \mid n' \preceq_h n \ \wedge\ n' \neq n\}$.

A set S of nodes is *backwards-closed* with respect to a binary relation R iff, for all nodes n_1 and n_2, if $n_2 \in S$ and $n_1 \mathrel{R} n_2$, then $n_1 \in S$.

Given a history $h = \langle \mathit{tr}, \rightarrow, \mathit{role}\rangle$ of a system $\mathcal{M}$, a set S of nodes that is backward-closed with respect to $\preceq_h$ can be regarded as a history, denoted $\mathrm{nodesToHist}_h^{\mathcal{M}}(S)$, in a natural way. Specifically, $\mathrm{nodesToHist}_h^{\mathcal{M}}(S)$ is $\langle \mathit{tr}_1, \rightarrow_1, \mathit{role}_1\rangle$, where $\mathcal{N}_{\mathit{tr}_1} = S$, $\rightarrow_1 = \rightarrow \cap (S \times S)$, $\mathrm{term}_{\mathit{tr}_1}(n) = \mathrm{term}_{\mathit{tr}}(n)$ for all $n \in S$, and $\mathit{role}_1(s) = \mathit{role}(s)$ for all $s \in \mathrm{strand}(S)$.

2.7 Derivability

Informally, a term t is derivable (by the penetrator) from a set S of nodes if the penetrator can compute t from $\mathrm{term}(S)$ and *pik*. A formal definition follows.

For a genval g that uniquely originates in a history h, let $\mathrm{origin}_h(g)$ denote the node from which g originates in h.

For a set S of nodes of a history $h = \langle \mathit{tr}, \rightarrow, \mathit{role}\rangle$ of a system $\mathcal{M} = \langle \Pi, \mathit{pen}\rangle$, let $\mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S)$ denote the set of genvals g that are required to be uniquely originated in h by item 5 in the definition of history and that originate from a node in S; formally, this is the set of genvals g such that there exists an uncompromised strand $s \in \mathrm{dom}(\mathit{tr})$ and a uniquely-originated parameter x of $\mathit{role}(s)$ such that $\mathit{args}(\mathit{role}(s), \mathit{tr}(s))(x) = g$ and $\mathrm{origin}_h(g) \in S$.

For a set T of terms, the (possibly non-well-formed) role Src_T is defined by $\mathrm{Src}_T(x : T) = \langle\!\langle +x \rangle\!\rangle$. When the subscript on Src is clear from context, we elide it.

A term t is *derivable* from a set S of nodes of a history h of a system $\mathcal{M} = \langle \Pi, \langle \mathit{pik}, \mathit{compr}\rangle\rangle$, denoted $S \vdash_h^{\mathcal{M}} t$, if there exists a history $h' = \langle \mathit{tr}', \rightarrow', \mathit{role}'\rangle$ of the system $\langle \{\mathrm{Src}_{\mathrm{abs}(\mathrm{term}_h(S))}\}, \langle \mathit{pik}, \mathit{compr}_d\rangle\rangle$, where $\mathit{compr}_d(\mathrm{Src}_{\mathrm{abs}(\mathrm{term}_h(S))}.x) = \emptyset$, such that

1. Arguments of strands for Msg in h' are not in $\mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S)$; that is, for all $s \in \mathrm{dom}(\mathit{tr}')$, if $\mathit{role}'(s) = \mathrm{Msg}$ and $x \in \mathrm{dom}(\mathit{args}(\mathrm{Msg}, \mathit{tr}'(s')))$, then $\mathit{args}(\mathrm{Msg}, \mathit{tr}'(s'))(x) \notin \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S)$.

2. There exists a node $n \in \mathcal{N}_{\mathit{tr}'}$ with $\mathrm{term}_{\mathit{tr}'}(n) = +t$.

This is essentially the derivability relation considered by Clarke *et al.* [CJM98]. Similar relations or functions have been considered by other researchers, *e.g.*, Paulson's analz and synth functions [Pau96].

2.8 Correctness Requirements

We consider the following correctness requirements, which are based on [WL93]. For a correctness requirement ϕ, we say that a system $\mathcal{M}$ satisfies ϕ iff every history of $\mathcal{M}$ satisfies ϕ. A *genval parameter* is a parameter with type $\mathit{Key}_{\mathit{sess}}$ or *Nonce*.

Genval Secrecy. Informally, genval secrecy says: the values of specified genval parameters are not revealed to the penetrator. Formally, a genval secrecy requirement for a system $\langle \Pi, \mathit{pen}\rangle$ is specified by a set of uniquely-originated genval parameters of Π.[10] A history $h = \langle tr, \rightarrow, role\rangle$ of a system $\mathcal{M}$ satisfies a genval secrecy requirement G iff, for every $r.x \in G$, for every uncompromised regular strand s for r, if $x \in \mathrm{dom}(args(role(s), tr(s)))$, then $\mathcal{N}_{tr} \not\vdash_h^{\mathcal{M}} args(role(s), tr(s))(x)$.

Agreement. Informally, agreement says: if some uncompromised strand executes a certain role to a certain point with certain arguments, then some strand must have executed a certain role to a certain point with certain arguments. An agreement requirement for a protocol Π has the form "$\langle r_1, len_1, xs_1\rangle$ precedes $\langle r_2, len_2, xs_2\rangle$", where $r_1 \in \Pi$, $r_2 \in \Pi$, and xs_1 and xs_2 are sequences of parameters of r_1 and r_2, respectively, such that $\mathrm{len}(xs_1) = \mathrm{len}(xs_2)$ and for $j \in \{1, 2\}$, every parameter in xs_j occurs in $r_j(0), r_j(1), \ldots, r_j(len_j - 1)$. A history $\langle tr, \rightarrow, role\rangle$ of a system $\langle \Pi, pen\rangle$ satisfies that agreement requirement iff, if tr contains an uncompromised strand s_2 such that $role(s_2) = r_2$ and $\mathrm{len}(tr(s_2)) \geq len_2$, then tr contains a strand s_1 such that $role(s_1) = r_1$ and $\mathrm{len}(tr(s_1)) \geq len_1$ and

$$(\forall i \in \mathrm{dom}(xs_1) : args(r_1, tr(s_1))(xs_1(i)) = args(r_2, tr(s_2))(xs_2(i)). \tag{11}$$

For example, $\mathcal{M}_Y$ might be expected to satisfy the genval secrecy requirement $\{\mathrm{Init}_Y.nr, \mathrm{Init}_Y.k, \mathrm{Resp}_Y.nr, \mathrm{Resp}_Y.k, \mathrm{Srvr}_Y.nr, \mathrm{Srvr}_Y.k\}$ and the agreement requirements

$$\langle \mathrm{Resp}_Y, 2, \langle\!\langle i, r, ni, nr\rangle\!\rangle\rangle \text{ precedes } \langle \mathrm{Init}_Y, 2, \langle\!\langle i, r, ni, nr\rangle\!\rangle\rangle$$
$$\langle \mathrm{Init}_Y, 3, \langle\!\langle i, r, ni, nr, k\rangle\!\rangle\rangle \text{ precedes } \langle \mathrm{Resp}_Y, 4, \langle\!\langle i, r, ni, nr, k\rangle\!\rangle\rangle.$$

3 Restrictions

Hereafter, we consider only systems $\langle \Pi, \langle pik, compr\rangle\rangle$ that satisfy the following static restrictions:

Shallow Ciphertext Restriction: In every parameterized term of every role in Π and in every term in pik, $encr$ does not occur in the scope of $encr$. This property also holds for every term in every trace for a well-formed role, because arguments of well-formed roles cannot be ciphertexts.

Unsent Long-Term Keys Restriction: In every parameterized term in every role of Π and in every term in $pik \setminus (Key_{sym} \cup Key_{asym})$, the operators key, $pubkey$, and $pvtkey$ occur only in the second argument of $encr$. This implies that long-term keys not in pik are not sent in messages.

If a system does not satisfy the shallow ciphertext restriction, applying a transformation that removes some encryptions might help [HL99].

3.1 Weak Support

Informally, a set S' of nodes supports a set S of nodes if S' contains all of the nodes in S and all of the regular nodes on which nodes in S depend. Note that S and S' cannot easily be regarded as histories, because S might lack some necessary regular strands and penetrator strands, and S' might lack some necessary penetrator strands. A formal definition of support follows.

[10] Including genval parameters that are not uniquely-originated would be pointless, because a genval secrecy requirement containing one would be violated by all systems, because values of such parameters are available to the penetrator from strands for Msg.

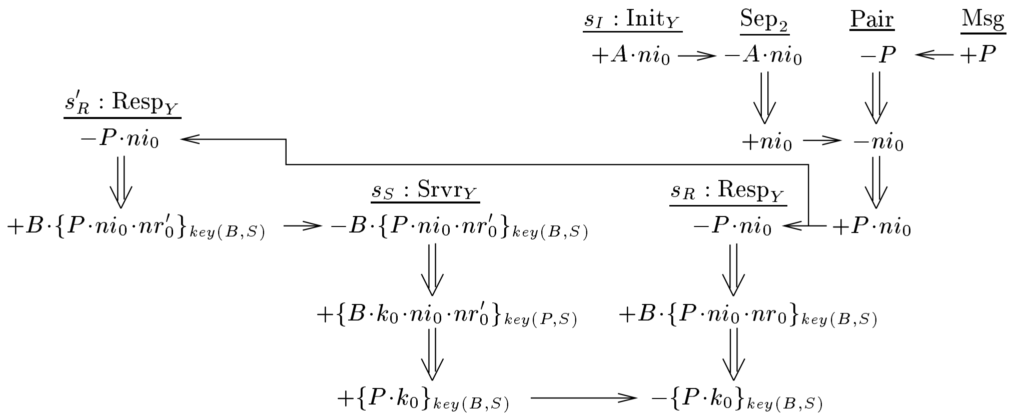
s_I : Init$_Y$
Sep$_2$
Pair
Msg
$+A\cdot ni_0$
$-A\cdot ni_0$
$-P$
$+P$
s'_R : Resp$_Y$
$-P\cdot ni_0$
$+ni_0$
$-ni_0$
s_S : Srvr$_Y$
s_R : Resp$_Y$
$+B\cdot\{P\cdot ni_0\cdot nr'_0\}_{key(B,S)}$
$-B\cdot\{P\cdot ni_0\cdot nr'_0\}_{key(B,S)}$
$-P\cdot ni_0$
$+P\cdot ni_0$
$+\{B\cdot k_0\cdot ni_0\cdot nr'_0\}_{key(P,S)}$
$+B\cdot\{P\cdot ni_0\cdot nr_0\}_{key(B,S)}$
$+\{P\cdot k_0\}_{key(B,S)}$
$-\{P\cdot k_0\}_{key(B,S)}$

Figure 4: A history of $\mathcal{M}_Y$. Strand names for penetrator strands are elided.

Let $\mathcal{RN}_h^{\mathcal{M}}$ denote the set of regular nodes in history h of system $\mathcal{M}$; formally,

$$\mathcal{RN}_{\langle tr,\rightarrow,role\rangle}^{\langle \Pi,pen\rangle} = \{n \in \mathcal{N}_{tr} \mid role(\mathrm{strand}(n)) \in \Pi\}. \tag{12}$$

A set S' of nodes is a *weak support* for a set S of nodes in a history h of a system $\mathcal{M}$ if

WkSupp1. $\mathcal{N}_h \supseteq S' \supseteq S$.

WkSupp2. S' is backwards-closed with respect to $\Rightarrow$.

WkSupp3. For all negative nodes n in S', $\mathrm{preds}_h(n) \cap S' \cap \mathcal{RN}_h^{\mathcal{M}} \vdash_h^{\mathcal{M}} \mathrm{term}_h(n)$.

For example, in the history h_Y of $\mathcal{M}_Y$ in Figure 4,

$$S_Y = \{\langle s'_R, 0\rangle, \langle s'_R, 1\rangle, \langle s_S, 0\rangle, \langle s_S, 1\rangle, \langle s_S, 2\rangle, \langle s_R, 0\rangle, \langle s_R, 1\rangle, \langle s_R, 2\rangle\} \tag{13}$$

is a weak support for $\{\langle s_R, 2\rangle\}$. $\langle s_R, 2\rangle$ does not depend on $\langle s_I, 0\rangle$, in the sense that $\langle s_I, 0\rangle \notin S_Y$, even though $\langle s_I, 0\rangle \preceq_{h_Y} \langle s_R, 2\rangle$. Informally, the dependence represented by $\preceq_{h_Y}$ can be ignored here because nodes in S_Y receive ni_0 from nodes outside S_Y only in contexts in which ni_0 can be replaced with a value generated by the penetrator. The careful treatment of unique origination in the definition of derivability allows such inessential dependencies to be ignored.

The following lemma says, roughly speaking, that a weak support can be transformed into a history by adding only penetrator nodes, not adding or changing regular nodes. This shows that the notion of weak support captures all essential dependencies between regular nodes.

Given a strand space tr, a strand $s \in \mathrm{dom}(tr)$, and a set S of nodes of tr that is backwards-closed with respect to $\Rightarrow$, S contains nodes on a prefix of $tr(s)$; let $\mathrm{prefix}_{tr}(s, S)$ denote that prefix. For example, for the history in Figure 4, $\mathrm{prefix}_{tr}(s_R, \{\langle s_R, 0\rangle, \langle s_I, 0\rangle\}) = \langle\!\langle -P\cdot ni_0\rangle\!\rangle$.

Lemma 1. If S' is a weak support for S in a history $h = \langle tr, \rightarrow, role\rangle$ of a system $\mathcal{M} = \langle \Pi, \langle pik, compr\rangle\rangle$, then there exists a history $h' = \langle \mathit{tr}', \rightarrow', \mathit{role}' \rangle$ of $\mathcal{M}$ such that

$$\begin{array}{l}(\forall s \in \mathrm{strand}(S') : s \in \mathrm{dom}(\mathit{tr}') \ \wedge\ \mathit{tr}'(s) = \mathrm{prefix}_{\mathit{tr}}(s, S') \ \wedge\ \mathit{role}'(s) = \mathit{role}(s)) \\ \wedge\, (\forall s \in \mathrm{dom}(\mathit{tr}') \setminus \mathrm{strand}(S') : \mathit{role}'(s) \in \Pi_P(\mathit{pik}))\end{array} \tag{14}$$

Proof: A witness h' can be constructed as follows. Let S'_{neg} denote the set of negative nodes in S'. For each $n \in S'_{neg}$, let $h_n = \langle \mathit{tr}_n, \rightarrow_n, \mathit{role}_n \rangle$ be a history that witnesses the truth of $\mathrm{preds}_h(n) \cap S' \cap \mathcal{RN}_h^{\mathcal{M}} \vdash_h^{\mathcal{M}} \mathrm{term}_{\mathit{tr}}(n)$. Let out_n be a node in $\mathcal{N}_{\mathit{tr}_n}$ such that $\mathrm{term}_{\mathit{tr}_n}(\mathit{out}_n) = +\mathrm{abs}(\mathrm{term}_{\mathit{tr}}(n))$. For each strand s for Src in h_n, let $\mathrm{witness}_n(s)$ be a node in $\mathrm{preds}_h(n) \cap S' \cap \mathcal{RN}_h^{\mathcal{M}}$ such that $\mathrm{abs}(\mathrm{term}_{\mathit{tr}}(\mathrm{witness}_n(s))) = \mathit{args}(\mathrm{Src}, \mathit{tr}_n(s))(x)$; the definitions of weak support and derivability together imply that such a node exists. Rename strands, if necessary, so that $\mathrm{dom}(\mathit{tr}_n) \cap \mathrm{dom}(\mathit{tr}) = \emptyset$ and $\mathrm{dom}(\mathit{tr}_{n_1}) \cap \mathrm{dom}(\mathit{tr}_{n_2}) = \emptyset$ for $n_1 \neq n_2$. We define h' in two steps. The first step merges S' and all the h_n (for n in S'_{neg}), yielding $\langle \mathit{tr}_1, \mathit{role}_1, \rightarrow_1 \rangle$. The second step eliminates strands for Src, yielding h'.

$$\begin{array}{rcl}
\mathrm{dom}(\mathit{tr}_1) & = & \mathrm{strand}(S') \cup \displaystyle\bigcup_{n \in S'_{neg}} \mathrm{dom}(\mathit{tr}_n) \\
\mathit{tr}_1(s) & = & \begin{cases} \mathrm{prefix}_{\mathit{tr}}(s, S') & \text{if } s \in \mathrm{strand}(S') \\ \mathit{tr}_n(s) & \text{if } s \in \mathrm{dom}(\mathit{tr}_n) \text{ for some } n \in S'_{neg} \end{cases} \\
\mathit{role}_1(s) & = & \begin{cases} \mathit{role}(s) & \text{if } s \in \mathrm{strand}(S') \\ \mathit{role}_n(s) & \text{if } s \in \mathrm{dom}(\mathit{tr}_n) \text{ for some } n \in S'_{neg} \end{cases} \\
\rightarrow_1 & = & \rightarrow \cap\, (S' \times S') \cup \displaystyle\bigcup_{n \in S'_{neg}} \rightarrow_n \cup \{\langle \mathit{out}_n, n \rangle\} \\
\mathrm{dom}(\mathit{tr}') & = & \{s \in \mathrm{dom}(\mathit{tr}_1) \mid \mathit{role}_1(s) \neq \mathrm{Src}\} \\
\mathit{tr}'(s) & = & \mathit{tr}_1(s) \\
\mathit{role}'(s) & = & \mathit{role}_1(s) \\
\mathrm{source}_n(s) & = & \begin{cases} \mathrm{witness}_n(s) & \text{if } \mathrm{term}_{\mathit{tr}}(\mathrm{witness}_n(s)) \text{ is positive} \\ \mathit{out}_{\mathrm{witness}_n(s)} & \text{if } \mathrm{term}_{\mathit{tr}}(\mathrm{witness}_n(s)) \text{ is negative} \end{cases} \\
\rightarrow' & = & \rightarrow_1 \cap\, (\mathcal{N}_{\mathit{tr}'} \times \mathcal{N}_{\mathit{tr}'}) \\
 & & \cup \bigcup_{n \in S'_{neg}} \{\langle \mathrm{source}_n(s), n_1 \rangle \mid s \in \mathrm{dom}(\mathit{tr}_1) \wedge \mathit{role}_1(s) = \mathrm{Src} \wedge \langle s, 0 \rangle \rightarrow_1 n_1\}
\end{array}$$

$\langle \mathit{tr}', \rightarrow', \mathit{role}' \rangle$ being a history of $\mathcal{M}$ follows from h and all the h_n being histories of the appropriate systems and from the following observations. For acyclicity of $\rightarrow'$, note that $\mathrm{witness}_n(s) \rightarrow n$. For the unique-origination condition, we need to show that for every regular strand $s \in \mathrm{strand}(S')$, for every parameter $x \in \mathrm{dom}(\mathit{args}(\mathit{role}'(s), \mathit{tr}'(s)))$, if x is uniquely-originated and s is uncompromised, then the genval $g = \mathit{args}(\mathit{role}'(s), \mathit{tr}'(s))(x)$ does not occur in pik and uniquely originates from $\langle s, i \rangle$, where i is the index of the first term in r that contains x. Note that $\mathit{role}'(s) = \mathit{role}(s)$, $\mathit{tr}'(s) = \mathrm{prefix}_{\mathit{tr}}(s, S')$, $g \in \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S')$, and $\langle s, i \rangle \in S'$. h is a history of $\mathcal{M}$, so g does not occur in pik and does not originate from any node other than $\langle s, i \rangle$ in S'. It remains to show that g does not originate from any penetrator node in tr'. By inspection of $\Pi_P(\mathit{pik})$, a genval that originates from a penetrator node must originate from a strand for Msg. The definitions of weak support and derivability imply that tr' does not contain strands for Msg from which genvals in $\mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S')$ originate. ∎

Continuing the above example, a history of $\mathcal{M}_Y$ that witnesses that S_Y is a weak support for $\{\langle s_R, 2\rangle\}$ in the history in Figure 4 can be obtained from the history in Figure 4 by deleting s_I and the strand for Sep_2 and adding a strand for Msg with argument ni_0. This witness can be produced by the construction in the proof of Lemma 1; we say "can be" because the construction depends on the choice of the histories h_n, which are not uniquely determined.

3.2 Support

Weak supports are not compositional, in the sense that Lemma 2 (below) does not hold for weak supports. We introduce a stronger notion that is compositional.

A set S' of nodes is a *support* for a set S of nodes in a history h of a system $\mathcal{M}$ if

Supp1. S' is a weak support of S in history h of $\mathcal{M}$.

Supp2. For all $g \in \mathrm{genvals}(\mathrm{term}_h(S')) \cap (\mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(\mathcal{N}_h) \setminus \mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'))$, g occurs in the clear in $\mathrm{term}_h(\mathrm{origin}_h(g))$.

If S' is a support for S, we say that S' supports S. For a strand s, if S' supports $\mathrm{nodes}(s)$, we say that S' supports s.

Lemma 2. If S'_0 and S'_1 support S_0 and S_1, respectively, in a history $h = \langle tr, \to, role\rangle$ of a system $\mathcal{M}$, then $S'_0 \cup S'_1$ supports $S_0 \cup S_1$ in history h of $\mathcal{M}$.

Proof: We need to show that Supp1 (equivalently, WkSupp1–WkSupp3) and Supp2 are satisfied. WkSupp1, WkSupp2, and Supp2 follow easily from the corresponding condition holding in the hypotheses. For WkSupp3, consider $i \in \{0, 1\}$, and consider a negative node n in S'_i. Let $h' = \langle tr', \to', role'\rangle$ be a history that witnesses $\mathrm{preds}_h(n) \cap S'_i \cap \mathcal{RN}^{\mathcal{M}}_h \vdash^{\mathcal{M}}_h \mathrm{term}_h(n)$. Suppose $\mathrm{term}_h(n)$ does not contain a genval in $\mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_{(i+1)\%2}) \setminus \mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_i)$; then h' also witnesses $\mathrm{preds}_h(n) \cap (S'_0 \cup S'_1) \cap \mathcal{RN}^{\mathcal{M}}_h \vdash^{\mathcal{M}}_h \mathrm{term}_h(n)$. Suppose $\mathrm{term}_h(n)$ contains a genval g in $\mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_{(i+1)\%2}) \setminus \mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_i)$; then h' might not witness $\mathrm{preds}_h(n) \cap (S'_0 \cup S'_1) \cap \mathcal{RN}^{\mathcal{M}}_h \vdash^{\mathcal{M}}_h \mathrm{term}_h(n)$, because h' might contain strands for Msg with argument g, violating the first condition in the definition of derivable. Note that $\mathrm{origin}_h(g) \in S'_{(i+1)\%2}$, and $\mathrm{origin}_h(g) \preceq_h n$ (because g uniquely originates from $\mathrm{origin}_h(g)$ in h, and $\mathrm{term}_h(n)$ contains g), and $\mathrm{origin}_h(g) \in \mathcal{RN}^{\mathcal{M}}_h$ (because penetrator roles do not have uniquely-originated parameters). Thus, we can construct a history $h'' = \langle tr'', \to'', role''\rangle$ that witnesses $\mathrm{preds}_h(n) \cap (S'_0 \cup S'_1) \cap \mathcal{RN}^{\mathcal{M}}_h \vdash^{\mathcal{M}}_h \mathrm{term}_h(n)$ by starting with h' and, for each $i \in \{0, 1\}$ and each genval g in $\mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_{(i+1)\%2}) \setminus \mathrm{uniqOrigRqrd}^{\mathcal{M}}_h(S'_i)$,

1. Add a strand s_{Src} for Src with argument $\mathrm{abs}(\mathrm{term}_h(\mathrm{origin}_h(g)))$. Note that $\mathrm{term}_{h''}(\langle s_{\mathrm{Src}}, 0\rangle) = \mathrm{term}_h(\mathrm{origin}_h(g))$.

2. Add strands for Sep_1 and Sep_2, if necessary, to select g from $\mathrm{term}_{h''}(\langle s_{\mathrm{Src}}, 0\rangle)$, so h'' contains a node n_g with $\mathrm{term}(n_g) = +g$. This selection is possible because Supp2 implies that g occurs in the clear in $\mathrm{term}_h(\mathrm{origin}_h(g))$.

3. For each strand s_{Msg} for Msg with argument g, delete s_{Msg} from h'' and, for each negative node n_{neg} such that $\langle s_{\mathrm{Msg}}, 0\rangle \to' n_{neg}$, let $n_g \to'' n_{neg}$. This does not create cycles in $\preceq_{h''}$; the main point is that s_{Src} contains no negative nodes, so $\langle s_{\mathrm{Src}}, 0\rangle$ is $\preceq_{h''}$-minimal. ∎

Continuing the example in Section 3.1, S_Y is a support for $\{\langle s_R, 2\rangle\}$ in the history in Figure 4, because ni_0 occurs in the clear in $\mathrm{term}(\langle s_I, 0\rangle)$.

3.3 Bounded Support Restriction

A *strand count* for a protocol Π is a function from Π to the natural numbers. The strand count of a set S of nodes of a history $h = \langle tr, \rightarrow, role\rangle$ of a system $\langle \Pi, pen\rangle$, denoted $\mathrm{SC}_h^{\langle \Pi, pen\rangle}(S)$, is defined by: $\mathrm{SC}_h^{\langle \Pi, pen\rangle}(S)(r) = \mathrm{size}(\{s \in \mathrm{strand}(S) \mid role(s) = r\})$ for $r \in \Pi$. We say that a history h of a system $\mathcal{M}$ has strand count $\mathrm{SC}_h^{\mathcal{M}}(\mathcal{N}_h)$.

Define a partial order on strand counts for a protocol Π: $f_1 \preceq_{SC} f_2$ iff $(\forall r \in \mathrm{dom}(f_1) : f_1(r) \leq f_2(r))$.

Given a strand count f for Π, a history h of a system $\mathcal{M} = \langle \Pi, pen\rangle$ satisfies the *bounded support restriction* for strand count f, abbreviated $\mathrm{BSR}(f)$, iff for every regular strand s in h, there exists a support S for s in history h of $\mathcal{M}$ such that $\mathrm{SC}_h^{\mathcal{M}}(S) \preceq_{SC} f$. A system satisfies $\mathrm{BSR}(f)$ iff all of its histories do.

Table (15) lists some systems and, for each system, a strand count f for which the system satisfies $\mathrm{BSR}(f)$. In principle, using Theorem 2 in Section 5, these results can be obtained automatically through state-space exploration of histories with bounded strand counts. In practice, using the bounds in Section 7, this is feasible for some of the protocols but perhaps not all of them. The results in (15) were proven by hand. The proof for the NSL protocol appears in Appendix B. The proofs for the other protocols involve similar patterns of reasoning.

System	f(Init)	f(Resp)	f(Srvr)
$\mathcal{M}_{NSL}$ (Needham-Schroeder-Lowe)	1	1	none
$\mathcal{M}_Y$ (Yahalom)	1	2	1
$\mathcal{M}_{OR}$ (Otway-Rees)	1	1	1

(15)

The history of $\mathcal{M}_Y$ in Figure 4 illustrates why $f(\mathrm{Resp}) > 1$ for Yahalom; every support for s_R contains nodes from s_R and s'_R.

There are systems that do not satisfy $BSR(f)$ for any f. An example is the system $\mathcal{M}_{us} = \langle \Pi_{us}, \langle pik_{keys}, compr_{us}\rangle\rangle$, where $\Pi_{us} = \{I, R\}$ and

$$\begin{array}{lcl} I(\underline{n} : Nonce) & = & \langle\!\langle +\{n\}_{key(A,B)}\rangle\!\rangle \\ R(n : Nonce, \underline{n'} : Nonce) & = & \langle\!\langle -\{n\}_{key(A,B)}, +\{n'\}_{key(A,B)}\rangle\!\rangle \end{array} \qquad (16)$$

and $compr_{us}(x) = \emptyset$ for $x \in \mathrm{params}(\Pi_{us})$. Let $role(s_0) = I$ and $tr(s_0) = I(n_0)$. For $i > 0$, let $role(s_i) = R$ and $tr(s_i) = R(n_{i-1}, n_i)$. Let $\rightarrow = \{\langle\langle s_0, 0\rangle, \langle s_1, 0\rangle\rangle\} \cup \{\langle\langle s_i, 1\rangle, \langle s_{i+1}, 0\rangle\rangle \mid i > 0\}$. It is easy to see that $h = \langle tr, \rightarrow, role\rangle$ is a history of $\mathcal{M}_{us}$. Every support for s_i in h contains $\Omega(i)$ nodes.

3.4 Revealed Genval Restriction

Informally, the revealed genval restriction (RGR) says: the penetrator learns a genval g that uniquely originates on a regular strand only if the protocol "directly reveals" g. RGR prevents genvals from being revealed to the penetrator indirectly, *e.g.*, by encrypting one genval with another and then directly revealing the latter genval. Without RGR, obtaining a simple static bound on the dependence width (see Section 4) would be difficult.

A node n *directly reveals* a term t in a history h of a system $\mathcal{M}$ iff n is a positive regular node and $\{n\} \vdash_h^{\mathcal{M}} t$.

Formally, a history $\langle tr, \to, role\rangle$ of a system $\mathcal{M}$ satisfies RGR iff for every regular strand $s \in \mathrm{dom}(tr)$ and every parameter $x \in \mathrm{dom}(args(role(s), tr(s)))$ of $role(s)$, if x is uniquely-originated, s is uncompromised, and $\mathcal{N}_{tr} \vdash_h^{\mathcal{M}} args(role(s), tr(s))(x)$, then $\mathcal{N}_{tr}$ contains a node that directly reveals $args(role(s), tr(s))(x)$. A system satisfies RGR iff all of its histories do.

The systems in (10) satisfy RGR. As in Section 3.3, these results were proven by hand (the proofs are straightforward), although using Theorem 2 in Section 5, they can in principle be obtained automatically through state-space exploration of histories with bounded strand counts.

4 Dependence Width

Informally, the dependence width of a negative parameterized term $r(i)$ in a role r of a system $\mathcal{M}$ is the maximum number of "additional" positive regular nodes needed in any history h of $\mathcal{M}$ to provide the penetrator with enough knowledge to produce the term received by any node $\langle s, i\rangle$ of h such that $role(s) = r$. "Additional" here means "beyond those needed for the penetrator to produce negative terms that occur earlier in the same strand". The concept of dependence width is used in the proof of Theorem 2 (in Section 5) to bound the number of strands involved in a violation of the bounded support restriction. A formal definition of dependence width follows.

For a set S of numbers, let $\min(S)$ and $\max(S)$ denote the minimum and maximum element of S, respectively. We define $\min(\emptyset) = 0$ and $\max(\emptyset) = 0$.

A *revealing set* for a term t at a node n in a history h of a system $\mathcal{M}$ is a set R of positive regular nodes of tr such that $R \subseteq \mathrm{preds}_h(n)$ and $R \vdash_h^{\mathcal{M}} t$. Intuitively, the main difference between "revealing set for term(n) at n" and "weak support for $\{n\}$" is, roughly, that the former considers only one step of dependency, while the latter implicitly considers a transitive closure of dependencies.

The *revealing set min-size* of a term t at a node n in a history h of a system $\mathcal{M}$ is

$$\begin{aligned}\mathrm{rvlSetMinSz}(t, n, h, \mathcal{M}) = \min(\{&\mathrm{size}(R \setminus \mathrm{nodes}_h(\mathrm{strand}(n))) \mid \\ &R \text{ is a revealing set for } t \text{ at } n \text{ in } h \text{ of } \mathcal{M}\})\end{aligned} \tag{17}$$

Nodes in R that are on strand(n) are not counted in the revealing set min-size (and hence not in the dependence width, defined by (18)), because in the proof of Theorem 2—specifically, in equation (27)—those nodes appear in $\mathrm{support}_{h_0}^{\mathcal{M}}(s_0)$ and hence are excluded from the index set of the rightmost union, and the dependence width is designed to bound the size of that index set.

Note that, if there are no revealing sets for t at n in history h of system $\mathcal{M}$ (*i.e.*, t is not known to the penetrator at that point), then $\mathrm{rvlSetMinSz}(t, n, h, \mathcal{M}) = 0$.

Let r be a role in (the protocol in) a system $\mathcal{M}$, and let i be the index of a negative term in r. The *dependence width* of $\langle r, i\rangle$ in $\mathcal{M}$ is

$$\begin{aligned}\mathrm{DW}(\langle r, i\rangle, \mathcal{M}) = \max(\{&\mathrm{rvlSetMinSz}(\mathrm{term}_{tr}(\langle s, i\rangle), \langle s, i\rangle, \langle tr, \to, role\rangle, \mathcal{M}) \mid \\ &\langle tr, \to, role\rangle \in \mathrm{Hist}(\mathcal{M}) \wedge \langle s, i\rangle \in \mathcal{N}_{tr} \wedge role(s) = r\})\end{aligned} \tag{18}$$

The *dependence width* of a system $\langle \Pi, \mathit{pen} \rangle$ is

$$\mathrm{DW}(\langle \Pi, \mathit{pen} \rangle) = \max(\{\mathrm{DW}(\langle r, i \rangle, \langle \Pi, \mathit{pen} \rangle) \mid r \in \Pi \wedge r(i) \text{ is negative}\}) \tag{19}$$

The proof of Theorem 2 relies on an upper bound on the dependence width of a system. It is convenient to obtain this bound based on the syntactic structure of the protocol. This is difficult if a protocol sends terms of the forms $\{g\}_{k_1}$, $\{k_1\}_{k_2}$, $\{k_2\}_{k_3}$, …, $\{k_{i-1}\}_{k_i}$, k_i; in this case, a minimum-size revealing set for g might contain $i + 1$ nodes. RGR prohibits such behavior.

The *RGR dependence width* of $\langle r, i \rangle$ in $\mathcal{M}$, denoted $\mathrm{DW}_{\mathrm{RGR}}(\langle r, i \rangle, \mathcal{M})$, is defined by (18) with DW replaced with $\mathrm{DW}_{\mathrm{RGR}}$ and $\mathrm{Hist}(\mathcal{M})$ replaced with $\{h \in \mathrm{Hist}(\mathcal{M}) \mid h \text{ satisfies RGR}\}$.

The *RGR dependence width* of a system $\mathcal{M}$, denoted $\mathrm{DW}_{\mathrm{RGR}}(\mathcal{M})$, is defined by (19) with DW replaced with $\mathrm{DW}_{\mathrm{RGR}}$.

For a role r and $i \in \mathrm{dom}(r)$, let $\mathrm{genPrm}(r, i)$ be the set of genval parameters of r that occur in $r(i)$. Let

$$\begin{aligned} \mathrm{genPrmClr}(r, i) &= \{x \in \mathrm{genPrm}(r, i) \mid x \text{ occurs in the clear in } r(i)\} & (20)\\ \mathrm{genPrmClr}(r, 0..i) &= \bigcup_{j \in [0..i]} \mathrm{genPrmClr}(r, j). & (21) \end{aligned}$$

Theorem 1. Let $\mathcal{M} = \langle \Pi, \langle \mathit{pik}, \mathit{compr} \rangle \rangle$ be a system satisfying the shallow ciphertext and unsent long-term keys restrictions. Let $r \in \Pi$. If $r(i)$ is negative and contains at most one occurrence of *encr*, then

$$\begin{aligned} \mathrm{DW}_{\mathrm{RGR}}(\langle r, i \rangle, \mathcal{M}) \leq \max(\{&\mathrm{size}(\mathrm{genPrm}(r, i) \setminus \mathrm{genPrmClr}(r, 0..i-1)),\\ &\mathrm{size}(\mathrm{genPrmClr}(r, i) \setminus \mathrm{genPrmClr}(r, 0..i-1)) + 1\}) \end{aligned} \tag{22}$$

Proof: Consider a history $h = \langle \mathit{tr}, \rightarrow, \mathit{role} \rangle$ for $\langle \Pi, \langle \mathit{pik}, \mathit{compr} \rangle \rangle$. Consider a node $\langle s, i \rangle$ such that $\mathit{role}(s) = r$. We bound the size of a revealing set for $\mathrm{term}(\langle s, i \rangle)$ at $\langle s, i \rangle$ in history h (of $\mathcal{M}$). By hypothesis, $r(i)$ contains at most one occurrence of *encr*, and arguments of well-formed roles do not contain ciphertexts, so $\mathrm{term}(\langle s, i \rangle)$ contains at most one occurrence of *encr*.

Suppose $\mathrm{term}(\langle s, i \rangle)$ contains one occurrence of *encr* and hence one ciphertext $\{t_e\}_k$. Let E be the set of primitive terms that occur in t_e. Let C be the set of primitive terms that occur in the clear in $\mathrm{term}(\langle s, i \rangle)$.

Some observations: (O1) The definition of history implies that all primitive terms are available to the penetrator from strands for Msg except long-term keys not in *pik* and genvals in $\mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h)$. (O2) The unsent long-term keys restriction implies that E and C do not contain long-term keys not in *pik*.

Consider cases based on where $\{t_e\}_k$ originates.

case 1: $\{t_e\}_k$ originates from a penetrator node in $\mathrm{preds}_h(\langle s, i \rangle)$. In other words, the penetrator can perform an encryption that produces $\{t_e\}_k$. Let

$$S = \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h) \cap (C \cup E \cup \{k\}). \tag{23}$$

The unsent long-term keys restriction implies that the penetrator does not learn long-term keys, so if k is a long-term key, then $k \in \mathit{pik}$, and $\emptyset$ is a revealing set for k at $\langle s, i \rangle$ in h. This, together with observations (O1) and (O2), implies that the union of revealing sets at $\langle s, i \rangle$ in h for the genvals in S is a revealing set for $\text{term}(\langle s,i\rangle)$ at $\langle s,i\rangle$ in h. r is assumed to be well-formed, so $\text{size}(S) \le \text{size}(\text{genPrm}(r,i))$.

case 2: $\{t_e\}_k$ does not originate from a penetrator node in $\text{preds}_h(n)$. Then $\{t_e\}_k$ originates from some regular node $n_0 \in \text{preds}_h(\langle s,i\rangle)$. The shallow ciphertext restriction implies $\{n_0\}$ is a revealing set for $\{t_e\}_k$ at $\langle s,i\rangle$ in h. Let

$$S = \text{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h) \cap C. \tag{24}$$

Observations (O1) and (O2) imply that the union of $\{n_0\}$ and revealing sets at $\langle s,i\rangle$ in h for the genvals in S is a revealing set for $\text{term}(\langle s,i\rangle)$ at $\langle s,i\rangle$ in h. r is assumed to be well-formed, so $\text{size}(S) \le \text{size}(\text{genPrmClr}(r,i))$.

For each genval $g \in S$, note that g is known to the penetrator at $\langle s,i\rangle$ in h, *i.e.*, $\text{preds}_h(\langle s,i\rangle) \vdash_h^{\mathcal{M}} \mathit{args}(\mathit{role}(s), \mathit{tr}(s))(x)$ (in case 1, for genvals in $S \cap (E \cup \{k\})$, this follows from the fact that the ciphertext originates from a penetrator node), so RGR applied to $\text{nodesToHist}_h^{\mathcal{M}}(\text{preds}_h(\langle s,i\rangle))$ implies that $\text{preds}_h(\langle s,i\rangle)$ contains a regular node that directly reveals g, so g has a revealing set of size 1 at $\langle s,i\rangle$ in h.

A genval that occurs in the clear in a term in $\text{term}_{tr}(\{\langle s,0\rangle,\ldots,\langle s,i-1\rangle\})$ does not contribute to the RGR dependence width of $\text{term}(\langle s,i\rangle)$ at $\langle s,i\rangle$ in h, because nodes on the same strand as $\langle s,i\rangle$ are not counted in (17). This justifies excluding genval parameters in $\text{genPrmClr}(r,0..i-1)$ from S. Thus, in case 1, the RGR dependence width of $\text{term}(\langle s,i\rangle)$ at $\langle s,i\rangle$ in h is at most $\text{size}(\text{genPrm}(r,i) \setminus \text{genPrmClr}(r,0..i-1))$, and in case 2, it is at most $\text{size}(\text{genPrmClr}(r,i) \setminus \text{genPrmClr}(r,0..i-1)) + 1$.

Suppose $\text{term}_h(\langle s,i\rangle)$ contains 0 occurrences of *encr*. The proof is the same as above, except there is no contribution from a ciphertext, so the RGR dependence width of $\text{term}(\langle s,i\rangle)$ at $\langle s,i\rangle$ in h is at most $\text{size}(\text{genPrm}(r,i) \setminus \text{genPrmClr}(r,0..i-1))$. ∎

Applying Theorem 1 to the systems in (10) yields

$$\text{DW}_{\text{RGR}}(\mathcal{M}_{NSL}) \le 2 \qquad \text{DW}_{\text{RGR}}(\mathcal{M}_{Y}) \le 2 \qquad \text{DW}_{\text{RGR}}(\mathcal{M}_{OR}) \le 2 \tag{25}$$

For example, the bound on the RGR dependence width of $\langle \text{Init}_Y, 1\rangle$ is

$$\max(\{\text{size}(\{k, ni, nr\} \setminus \{ni\}), \text{size}(\emptyset \setminus \{ni\}) + 1\}),$$

which simplifies to 2.

Theorem 1 applies only to terms containing at most one ciphertext. Generalizing it to apply to terms containing multiple ciphertexts (in protocols that satisfy the shallow ciphertext restriction) requires considering cases corresponding to which subset of the ciphertexts originate from penetrator nodes. This is not conceptually difficult, but it complicates the counting, since genval parameters that occur in multiple ciphertexts should be counted at most once.

Generalizing Theorem 1 to eliminate the shallow ciphertext restriction is also possible. This would entirely eliminate the need for this restriction, which is not used directly in proofs of other theorems. This requires extending the proof of Theorem 1 to consider values that are revealed by sequences of decryptions applied to nested ciphertexts. The resulting bounds on RGR dependence width of protocols that do not satisfy the shallow ciphertext restriction are larger and hence less useful in practice.

5 Reduction for Dynamic Restrictions

The following lemma says, roughly, that constructing a history h' from a support S' of a set S of nodes of a history h does not create new supports for S.

Lemma 3. Suppose S_0 supports S in a history h of a system $\mathcal{M}$. Let h' be a history of $\mathcal{M}$ whose existence is implied by Lemma 1 applied to S_0. Suppose S_1 supports S in history h' of $\mathcal{M}$. Then $S_1 \cap \mathcal{RN}_h^{\mathcal{M}}$ supports S in history h of $\mathcal{M}$.

Proof: It is easy to show that WkSupp1, WkSupp2, and Supp2 hold. For WkSupp3, we need to show that for every negative node n in S_1, $\mathrm{preds}_h(n) \cap S_1 \cap \mathcal{RN}_h^{\mathcal{M}} \vdash_h^{\mathcal{M}} \mathrm{term}_h(n)$. S_1 supports S in h', so $\mathrm{preds}_{h'}(n) \cap S_1 \cap \mathcal{RN}_{h'}^{\mathcal{M}} \vdash_{h'}^{\mathcal{M}} \mathrm{term}_{h'}(n)$. By definition of h', $\mathrm{term}_{h'}(n_1) = \mathrm{term}_h(n_1)$ for all $n_1 \in S_1$, and $\mathrm{preds}_{h'}(n) \subseteq \mathrm{preds}_h(n)$, and $\mathcal{RN}_{h'}^{\mathcal{M}} \subseteq \mathcal{RN}_h^{\mathcal{M}}$, so $\mathrm{preds}_{h'}(n) \cap S_1 \cap \mathcal{RN}_{h'}^{\mathcal{M}} \subseteq \mathrm{preds}_h(n) \cap S_1 \cap \mathcal{RN}_h^{\mathcal{M}}$. Derivability is monotonic (with respect to $\subseteq$) in its leftmost argument. Thus, $\mathrm{preds}_h(n) \cap S_1 \cap \mathcal{RN}_h^{\mathcal{M}} \vdash_h^{\mathcal{M}} \mathrm{term}_h(n)$. ▮

For a strand count f and a system $\mathcal{M}$, define a strand count $\beta(f, \mathcal{M})$ by

$$\beta(f, \mathcal{M})(r) = \max(\{\mathrm{DW}_{\mathrm{RGR}}(\mathcal{M}) + 1, 3\}) f(r). \qquad (26)$$

Theorem 2. Let $\mathcal{M} = \langle \Pi, \mathit{pen} \rangle$ be a system satisfying the shallow ciphertext and unsent long-term keys restrictions. Let f be a strand count for Π. $\mathcal{M}$ satisfies BSR(f) and RGR iff all histories of $\mathcal{M}$ with strand count $\beta(f, \mathcal{M})$ do.

Proof: The forward direction ($\Rightarrow$) of the "iff" follows immediately from the definitions. For the reverse direction ($\Leftarrow$), we prove the contrapositive, *i.e.*, we suppose there exists a history h of $\mathcal{M}$ that violates BSR(f) or RGR, and we construct a history of $\mathcal{M}$ with strand count at most $\beta(f, \mathcal{M})$ that violates the same property.

BSR(f) and RGR are safety properties [AS85] satisfied by histories with zero nodes, and $\preceq_h$ is well-founded, so there exists a $\preceq_h$-minimal node n_0 such that

- $\mathrm{nodesToHist}_h^{\mathcal{M}}(\mathrm{preds}_h(n_0))$ satisfies BSR(f) and RGR.
- $\mathrm{nodesToHist}_h^{\mathcal{M}}(\mathrm{preds}_h(n_0) \cup \{n_0\})$ violates BSR(f) or RGR.

The definitions of BSR and RGR imply that n_0 is a regular node. Let $h_0 = \mathrm{nodesToHist}_h^{\mathcal{M}}(\mathrm{preds}_h(n_0))$. Let $s_0 = \mathrm{strand}(n_0)$ and $i_0 = \mathrm{index}(n_0)$. Note that $n_0 \notin \mathcal{N}_{h_0}$.

For a history h' of $\mathcal{M}$ that satisfies $BSR(f)$, for a regular strand s of h', let $\mathrm{support}_{h'}^{\mathcal{M}}(s)$ denote a support for s in h' that has strand count at most f and contains no penetrator nodes (the intersection with $\mathcal{RN}_h^{\mathcal{M}}$ in WkSupp3 implies that if S' supports s in h', then so does $S' \cap \mathcal{RN}_{h'}^{\mathcal{M}}$).

Consider cases based on the sign of n_0.

case 1: n_0 is a negative node. n_0 cannot cause a violation of RGR, so n_0 causes a violation of BSR(f) in h. Suppose $i_0 > 0$. n_0 directly depends on $\langle s_0, i_0 - 1 \rangle$ and on a revealing set R for $\mathrm{term}(n_0)$ at n_0; more precisely, for all S', if S' supports $\{\langle s_0, i_0 - 1 \rangle\} \cup R$ in h, then $S' \cup \{n_0\}$ supports $\{n_0\}$ in h. Let

$$S_1 = \{n_0\} \cup \mathrm{support}_{h_0}^{\mathcal{M}}(s_0) \cup \bigcup_{n \in R \setminus \mathrm{nodes}_{tr_0}(s_0)} \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(n)), \qquad (27)$$

h_0 satisfies RGR, so Theorem 1 implies $\mathrm{size}(R \setminus \mathrm{nodes}_{h_0}(s_0)) \leq \mathrm{DW}_{\mathrm{RGR}}(\mathcal{M})$. h_0 satisfies $\mathrm{BSR}(f)$, so each support in (27) has strand count at most f. n_0 is on s_0, so it does not increase the strand count of S_1. Thus, S_1 has strand count at most $\beta(f, \mathcal{M})$.

Lemma 2 implies that $S_1 \setminus \{n_0\}$ supports $\{\langle s_0, i_0 - 1\rangle\} \cup R$ in h; thus, S_1 supports $\{n_0\}$ in h. Lemma 1 implies that S_1 can be transformed into a history h_1 of $\mathcal{M}$ by adding penetrator nodes. Adding penetrator nodes does not affect the strand count, so h_1 has strand count at most $\beta(f, \mathcal{M})$. We show by contradiction that n_0 also causes a violation of $\mathrm{BSR}(f)$ in h_1. Suppose n_0 does not cause such a violation. Then there exists a support S' for $\{n_0\}$ in h_1 with strand count at most f. Lemma 3 implies that $S' \cap \mathcal{RN}_h^{\mathcal{M}}$ is a support for $\{n_0\}$ in h with strand count at most f, a contradiction.

Suppose $i_0 = 0$. The proof in this case is similar to the case $i_0 > 0$, except n_0 directly depends only on R, so we omit $\mathrm{support}_{h_0}^{\mathcal{M}}(s_0)$ from the definition of S_1, and Lemma 2 implies that $S_1 \setminus \{n_0\}$ supports R in h.

case 2: n_0 is a positive node. n_0 cannot cause a violation of $\mathrm{BSR}(f)$, so n_0 causes a violation of RGR in h. Let gs be the set of genvals g such that $g \in \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h)$, $\mathrm{preds}_h(n_0) \nvdash_h^{\mathcal{M}} g$, $\mathrm{preds}_h(n_0) \cup \{n_0\} \vdash_h^{\mathcal{M}} g$, and no node in $\mathrm{preds}_h(n_0) \cup \{n_0\}$ directly reveals g. $\mathrm{nodesToHist}_h^{\mathcal{M}}(\mathrm{preds}_h(n_0) \cup \{n_0\})$ violates RGR, so gs is non-empty. There exists a g_0 in gs that can be computed by the penetrator before the other elements of gs; more precisely, for some g_0 in gs, there is a history h' that witnesses $\mathrm{preds}_h(n_0) \cup \{n_0\} \vdash_h^{\mathcal{M}} g_0$ and that does not contain any strand for Dec (the penetrator's decryption role) whose first argument is in gs. If such a g_0 and h' did not exist, then there would be circular constraints on the order in which elements of gs can be computed by the penetrator, and none of the elements of gs would be derivable from $\mathrm{preds}_h(n_0) \cup \{n_0\}$, a contradiction. No node in h directly reveals g_0, so g_0 does not occur in the clear in $\mathrm{term}_h(n_0)$. Thus, every history $h_d = \langle tr_d, \rightarrow_d, role_d\rangle$ that witnesses derivability of g_0 from $\mathrm{preds}_h(n_0) \cup \{n_0\}$ contains a strand s_d for Dec with arguments $k_d = args(\mathrm{Dec}, tr_d(s_d))(k)$ and $t_d = args(\mathrm{Dec}, tr_d(s_d))(x)$ such that g_0 occurs in the clear in t_d, and $\{t_d\}_{k_d}$ originates from a node on a strand for Src in h_d and from a regular node $n_c \in \mathrm{preds}_h(n_0) \cup \{n_0\}$ in h. Let h_d be a history that witnesses derivability of g_0 from $\mathrm{preds}_h(n_0) \cup \{n_0\}$ such that the corresponding k_d is not in gs; the above choice of g_0 ensures that such a history exists. In all of the following cases, a set S_1 of nodes is defined that satisfies $\mathrm{SC}_h^{\mathcal{M}}(S_1) \preceq_{SC} \beta(f, \mathcal{M})$ and $S_1 \vdash_h^{\mathcal{M}} g_0$ and $\mathrm{origin}_h(g_0) \in S_1$.

case 2.1: k_d is a long-term key. The unsent long-term keys restriction implies $k_d \in pik$. This implies $\{n_c\} \vdash_h^{\mathcal{M}} g_0$. Let $S_1 = \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(n_c)) \cup \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(\mathrm{origin}_h(g_0)))$.

case 2.2: k_d is not a long-term key. Then $k_d \in Key_{sess}$.

case 2.2.1: $k_d \notin \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h)$. k_d is available to the penetrator from strands for Msg, so $\{n_c\} \vdash_h^{\mathcal{M}} g_0$. Let $S_1 = \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(n_c)) \cup \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(\mathrm{origin}_h(g_0)))$.

case 2.2.2: $k_d \in \mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(\mathcal{N}_h)$.

case 2.2.2.1: $\mathrm{preds}_h(n_0) \vdash_h^{\mathcal{M}} k_d$. Then $\mathcal{N}_{h_0} \vdash_{h_0}^{\mathcal{M}} k_d$. h_0 satisfies RGR, so there is a node n_k that directly reveals k_d. Let $S_1 = \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(n_c)) \cup \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(\mathrm{origin}_h(g_0))) \cup \mathrm{support}_{h_0}^{\mathcal{M}}(\mathrm{strand}(n_k))$.

case 2.2.2.2: $\mathrm{preds}_h(n_0) \nvdash_h^{\mathcal{M}} k_d$. The existence of strand s_d in h_d implies $\mathrm{preds}_h(n_0) \cup \{n_0\} \vdash_h^{\mathcal{M}} k_d$. If k_d is not directly revealed by n_0, then k_d is in gs (because it satisfies all of the conditions), a contradiction; thus, k_d is directly revealed by n_0. Let $S_1 = \mathrm{support}^{\mathcal{M}}_{h_0}(\mathrm{strand}(n_c)) \cup \mathrm{support}^{\mathcal{M}}_{h_0}(\mathrm{strand}(\mathrm{origin}_h(g_0))) \cup \mathrm{support}^{\mathcal{M}}_{h_0}(\mathrm{strand}(n_0)) \cup \{n_0\}$.

In all the above cases (within case 2) except 2.2.2.2, S_1 is a union of supports (of various strands), so Lemma 2 implies that S_1 is a support of some set of nodes (*e.g.*, itself), so Lemma 1 implies that S_1 can be transformed into a history h_1 of $\mathcal{M}$ by adding penetrator nodes. In case 2.2.2.2, this reasoning applies to $S_1 \setminus \{n_0\}$, and it is easy to see (because n_0 is positive) that adding n_0 yields a history of $\mathcal{M}$, so in this case, too, S_1 can be transformed into a history h_1 of $\mathcal{M}$ by adding penetrator nodes. Adding penetrator nodes does not affect the strand count, so h_1 has strand count at most $\beta(f, \mathcal{M})$. By construction, $\mathrm{origin}_h(g_0) \in S_1$, so $g_0 \in \mathrm{uniqOrigRqrd}^{\mathcal{M}}_{h_1}(\mathcal{N}_{h_1})$. By construction, $S_1 \vdash^{\mathcal{M}}_{h} g_0$, which implies $\mathcal{N}_{h_1} \vdash^{\mathcal{M}}_{h_1} g_0$. Removing nodes in $\mathcal{N}_h \setminus \mathcal{N}_{h_1}$ and adding penetrator nodes preserves the fact that no node directly reveals g_0. Thus, n_0 causes a violation of RGR in h_1. $\blacksquare$

6 Reduction for Correctness Requirements

Given a strand count f for a protocol Π, define a strand count $\mathrm{dbl}(f)$ for Π by: $\mathrm{dbl}(f)(r) = 2f(r)$.

Theorem 3. Let $\mathcal{M} = \langle \Pi, \mathit{pen} \rangle$ be a system satisfying the shallow ciphertext and unsent long-term keys restrictions. Let f be a strand count for Π. Let ϕ be a genval secrecy or agreement requirement. Suppose all histories of $\mathcal{M}$ with strand count $\beta(f, \mathcal{M})$ satisfy BSR(f) and RGR. $\mathcal{M}$ satisfies ϕ iff all histories of $\mathcal{M}$ with strand count $\mathrm{dbl}(f)$ do.

Proof: The forward direction ($\Rightarrow$) of the "iff" follows immediately from the definitions. For the reverse direction ($\Leftarrow$), we prove the contrapositive, *i.e.*, we suppose there exists a history $h = \langle \mathit{tr}, \rightarrow, \mathit{role} \rangle$ of $\mathcal{M}$ that violates ϕ, and we construct a history of $\mathcal{M}$ with strand count at most $\mathrm{dbl}(f)$ that violates ϕ.

Genval secrecy and agreement are safety properties [AS85] satisfied by histories with zero nodes, and $\preceq_h$ is well-founded, so there exists a $\preceq_h$-minimal node n_0 such that

- $\mathrm{nodesToHist}^{\mathcal{M}}_{h}(\mathrm{preds}_h(n_0))$ satisfies ϕ.
- $\mathrm{nodesToHist}^{\mathcal{M}}_{h}(\mathrm{preds}_h(n_0) \cup \{n_0\})$ violates ϕ.

By hypothesis, all histories of $\mathcal{M}$ with strand count $\beta(f, \mathcal{M})$ satisfy BSR(f) and RGR, so Theorem 2 implies that $\mathcal{M}$ satisfies BSR(f) and RGR. Thus, there exists a support S_0 for strand(n_0) in history h of $\mathcal{M}$ with strand count at most f.

Suppose ϕ is a genval secrecy requirement G. n_0 causes a violation of ϕ, so there exists a genval g such that g is the value of some parameter in G for some strand s_g and $g \in \mathrm{uniqOrigRqrd}^{\mathcal{M}}_{h}(\mathrm{preds}_h(n_0))$ (because genval secrecy requirements contain only uniquely-originated parameters) and $\mathrm{preds}_h(n_0) \nvdash^{\mathcal{M}}_{h} g$ and $\mathrm{preds}_h(n_0) \cup \{n_0\} \vdash^{\mathcal{M}}_{h} g$. RGR applied to $\mathrm{nodesToHist}^{\mathcal{M}}_{h}(\mathrm{preds}_h(n_0) \cup \{n_0\})$ implies that there is a node in $\mathrm{preds}_h(n_0) \cup \{n_0\}$ that directly reveals g. No node in $\mathrm{preds}_h(n_0)$ directly reveals g (because $\mathrm{preds}_h(n_0) \nvdash^{\mathcal{M}}_{h} g$), so n_0 directly reveals g. $\mathcal{M}$ satisfies BSR(f), so there exists a support S_g for s_g in history h of $\mathcal{M}$ with strand count at most f. Lemma 2 implies that $S_0 \cup S_g$ is a support for $\mathrm{nodes}_h(\mathrm{strand}(n_0)) \cup \mathrm{nodes}_h(s_g)$ with strand count at most $\mathrm{dbl}(f)$. Lemma 1 implies that $S_0 \cup S_g$ can be transformed into a history h_0 of $\mathcal{M}$ by adding penetrator nodes. $\mathrm{origin}_h(g)$ is in $S_0 \cup S_g$, so $g \in \mathrm{uniqOrigRqrd}^{\mathcal{M}}_{h_0}(\mathcal{N}_{h_0})$. Thus, n_0 directly reveals g in h_0 and thereby causes a violation of ϕ in h_0.

Suppose ϕ is an agreement requirement: $\langle r_1, len_1, xs_1\rangle$ precedes $\langle r_2, len_2, xs_2\rangle$. n_0 causes a violation of ϕ in h, so strand(n_0) is an uncompromised strand for r_2, and index$(n_0) = len_2 - 1$. Lemma 1 implies that S_0 can be transformed into a history h_0 of $\mathcal{M}$ by adding penetrator nodes. Note that $n_0 \in \mathcal{N}_{h_0}$. Removing nodes in $\mathcal{N}_h \setminus \mathcal{N}_{h_0}$ and adding penetrator nodes preserve strand(n_0) being uncompromised and preserves the lack of a node $\langle s_1, len_1\rangle$ such that $role(s_1) = r_1$ and the equality (11) (with s_2 replaced with s_0) holds. Thus, h_0 violates ϕ. ∎

7 Bounds for Sample Protocols

Based on Theorems 2 and 3, bounds on the strand counts of histories that need to be explored to check correctness of the systems in (10) can be computed from (15), (25), and (26). The results are

System	Init	Resp	Srvr
$\mathcal{M}_{NSL}$ (Needham-Schroeder-Lowe)	3	3	none
$\mathcal{M}_{Y}$ (Yahalom)	3	6	3
$\mathcal{M}_{OR}$ (Otway-Rees)	3	3	3

(28)

For a role consisting only of negative terms followed by positive terms, there is no need to allow instances of that role to be concurrent. Server roles typically have this form. For example, when model-checking the Yahalom protocol, the three instances of the server role can be represented by a single process that executes a loop that iterates three times, where the body of the loop is the server role. This reduces the number of global states.

Our restrictions are designed to hold for correct systems but hold for some flawed systems as well. For example, the Otway-Rees protocol does not ensure key agreement [THG98a] but satisfies our restrictions, so our reduction can be used with state-space exploration to verify that it satisfies some weaker agreement properties.

8 Computing Small Supports

Define a partial order on sets of nodes of a history h of a system $\mathcal{M}$: $S_1 \preceq_{SC}^{\mathcal{M},h} S_2$ iff $\mathrm{SC}_h^{\mathcal{M}}(S_1) \preceq_{SC} \mathrm{SC}_h^{\mathcal{M}}(S_2)$.

To check the bounded support restriction automatically, small supports need to be computed during state-space exploration. It is easy to devise a brute-force algorithm that computes a $\preceq_{SC}^{\mathcal{M},h}$-minimal support of a given set S of nodes in a given history h of a system $\mathcal{M}$ by testing whether each subset of $\mathcal{N}_h$ is a support for S. A faster algorithm that computes sufficiently small supports for protocols of interest is preferable, even if it does not always compute $\preceq_{SC}^{\mathcal{M},h}$-minimal supports.

We describe a simple polynomial-time algorithm that computes sufficiently small supports for the systems we have considered; *e.g.*, for all regular strands in all histories of $\mathcal{M}_{NSL}$, it computes a support containing at most one strand for each role. The algorithm assumes the protocol satisfies the shallow ciphertext restriction and the unsent long-term keys restriction, and that the history h satisfies RGR (otherwise, the hypotheses of the reduction are violated, so computing a support is unnecessary).

The discussion in the proof of Theorem 1 implies that, if $\mathcal{N}_h \supseteq S' \supseteq S$ and S' is backward closed with respect to $\Rightarrow$ and S' is not a weak support for S, then there exist a node $n_u \in S'$, a node $n_d \in \mathcal{N}_h$, and a subterm t_u of $\mathrm{term}_h(n)$ such that $\mathrm{preds}_h(n_u) \cap S' \cap \mathcal{RN}_h^{\mathcal{M}} \not\vdash_h^{\mathcal{M}} t_u$ and n_d directly reveals t_u in h (if t_u is a genval in $\mathrm{uniqOrigRqrd}_h^{\mathcal{M}}(S')$, this follows from RGR; if t_u is a ciphertext, this follows from the shallow ciphertext restriction). For such an S', let $\mathrm{need}_h^{\mathcal{M}}(S')$ denote a function call that returns a $\preceq_h$-minimal such node n_d (n_u and t_u are not needed in the algorithm). For the systems we have considered, sufficiently small supports are obtained regardless of which n_u, t_u, and n_d are chosen. In fact, for each n_u and t_u, there is in most cases only a single possible n_d (intuitively, this is because authentication protocols are designed to use unambiguous messages), and in such cases, the order in which different n_u and t_u are considered is insignificant.

For a set S of nodes, let $\mathrm{backClosure}(S)$ denote the least superset of S that is backward-closed with respect to $\Rightarrow$.

```
$S' := \mathrm{backClosure}(S)$
**while** $\neg(S'$ is a weak support for $S$ in $h)$
    $S' := \mathrm{backClosure}(S' \cup \{\mathrm{need}_h^{\mathcal{M}}(S')\})$                    (29)
**if** $(S'$ is a support for $S$ in $h)$ **return** $S'$
**else abort**;
```

Checking whether a given set of nodes is a weak support or a support for another given set of nodes can be done in polynomial time; the only non-trivial aspect is checking derivability of terms, which can be done using the approach in [CJM98]. The simple treatment in (29) of Supp2 (*i.e.*, abort if Supp2 is not satisfied) is sufficient for the systems we have considered.

For example, consider using algorithm (29) to compute a support for $\{\langle s_R, 2\rangle\}$ in the history of $\mathcal{M}_Y$ in Figure 4. The algorithm starts with $S' = \mathrm{backClosure}(\{\langle s_R, 2\rangle\})$, which evaluates to $S' = \{\langle s_R, 0\rangle, \langle s_R, 1\rangle, \langle s_R, 2\rangle\}$. S' is not a weak support for S, due to $n_u = \langle s_R, 2\rangle$, $t_u = \{P \cdot k_0\}_{key(B,S)}$, $n_d = \langle s_S, 2\rangle$. Inserting $\langle s_S, 2\rangle$ in S' and taking the backward closure yields $S' = \{\langle s_S, 0\rangle, \langle s_S, 1\rangle, \langle s_S, 2\rangle, \langle s_R, 0\rangle, \langle s_R, 1\rangle, \langle s_R, 2\rangle\}$. S' is not a weak support for S, due to $n_u = \langle s_S, 0\rangle$, $t_u = \{P \cdot ni_0 \cdot nr'_0\}_{key(B,S)}$, $n_d = \langle s'_R, 1\rangle$. Inserting $\langle s'_R, 1\rangle$ in S' and taking the backward closure yields S_Y, which is a weak support and a support for $\{\langle s_R, 2\rangle\}$, so the algorithm returns S_Y.

The efficiency of algorithm (29) can be improved by maintaining auxiliary data structures. The primary goal of this paper is to bound the number of strands and thereby limit exponential state-space explosion, so optimizations to polynomial-time algorithms are not explored here.

9 Discussion of Strand Counts

We do not have formal guidelines for choosing a strand count f for verifying a given system. Let f_i denote the strand count such that $f_i(r) = i$ for every role r. In principle, one could do an iterative search, starting with $f = f_1$ and repeatedly increasing f until state-space exploration of histories with strand count $\beta(f, \mathcal{M})$ either finds an attack or shows that the protocol satisfies $BSR(f)$ and RGR. This is a semi-decision procedure, because it diverges for systems like $\mathcal{M}_{us}$ in Section 3.3. This semi-decision procedure is more powerful than a semi-decision procedure that simply searches for attacks using more and more strands (in the manner of iterative deepening), because the former can both find attacks and verify correctness, while the latter diverges for all correct protocols and hence cannot verify correctness. In practice, iterative search for an appropriate f seems largely unnecessary, because it appears that correct protocols of interest here satisfy

BSR(f_2).

Acknowledgments. I am grateful to John Mitchell and the anonymous referees for suggestions that helped simplify this work and improve its presentation.

References

[Aba97] Martín Abadi. Explicit communication revisited: Two new attacks on authentication protocols. *IEEE Transactions on Software Engineering*, 23(3):185–186, March 1997.

[AJ98] Parosh Aziz Abdulla and Bengt Jonsson. Verifying networks of timed processes. In *Proc. 4th Intl. Conf. on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*. Springer-Verlag, 1998.

[AN95] Ross Anderson and Roger Needham. Robustness principles for public key protocols. In *Proc. Int'l. Conference on Advances in Cryptology (CRYPTO 95)*, volume 963 of *Lecture Notes in Computer Science*, pages 236–247. Springer-Verlag, 1995.

[AN96] Martín Abadi and Roger Needham. Prudent engineering practice for cryptographic protocols. *IEEE Transactions on Software Engineering*, 22(1):6–15, January 1996.

[AS85] Bowen Alpern and Fred B. Schneider. Defining liveness. *Information Processing Letters*, 24(4):181–185, October 1985.

[BAN90] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, February 1990.

[Bol98] Dominique Bolignano. Integrating proof-based and model-checking techniques for the formal verification of cryptographic protocols. In Alan J. Hu and Moshe Y. Vardi, editors, *Proc. Tenth Int'l. Conference on Computer-Aided Verification (CAV)*, volume 1427 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[CDL$^+$00] Iliano Cervesato, Nancy Durgin, Patrick Lincoln, John Mitchell, and Andre Scedrov. Relating strands and multiset rewriting for security protocol analysis. In Paul Syverson, editor, *Proc. 13th IEEE Computer Security Foundations Workshop (CSFW)*, pages 35–51. IEEE Computer Society Press, July 2000.

[CGJ95] Edmund M. Clarke, Orna Grumberg, and Somesh Jha. Verifying parameterized networks using abstractions and regular languages. In *Proc. Sixth Int'l. Conference on Concurrency Theory (CONCUR)*, 1995.

[CJM98] E.M. Clarke, S. Jha, and W. Marrero. Using state space exploration and a natural deduction style message derivation engine to verify security protocols. In *Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET)*, June 1998.

[DK97] Z. Dang and R. A. Kemmerer. Using the ASTRAL model checker for cryptographic protocol analysis. In *Proc. DIMACS Workshop on Design and Formal Verification of Security Protocols*, September 1997.

[DNL99] Ben Donovan, Paul Norris, and Gavin Lowe. Analyzing a library of security protocols using Casper and FDR. In *Proc. 1999 Workshop on Formal Methods and Security Protocols*, July 1999. Available via http://cm.bell-labs.com/cm/cs/who/nch/fmsp99/.

[DS81] D. E. Denning and G. M. Sacco. Timestamps in key distribution protocols. *Communications of the ACM*, 24(8):198–208, 1981.

[DvOW92] Whitfield Diffie, Paul C. van Oorschot, and Michael J. Wiener. Authentication and authenticated key exchanges. *Designs, Codes and Cryptography*, 2:107–125, 1992.

[DY83] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, IT-29(2):198–208, March 1983.

[EN96] E. Allen Emerson and Kedar S. Namjoshi. Automated verification of parameterized synchronous systems. In *Proc. 8th Int'l. Conference on Computer-Aided Verification (CAV)*, 1996.

[GLR95] Li Gong, Patrick Lincoln, and John Rushby. Byzantine agreement with authentication: Observations and applications in tolerating hybrid and link faults. In *Dependable Computing for Critical Applications—5*, pages 79–90. IFIP WG 10.4, preliminary proceedings, 1995.

[HL99] Mei Lin Hui and Gavin Lowe. Simplifying transformations for security protocols. In *Proc. 12th IEEE Computer Security Foundations Workshop (CSFW)*, 1999. To appear in Journal of Computer Security.

[HTWW96] Nevin Heintze, J. D. Tygar, Jeannette Wing, and Hao-Chi Wong. Model checking electronic commerce protocols. In *Proc. USENIX 1996 Workshop on Electronic Commerce*, November 1996.

[KM95] R. P. Kurshan and K. L. McMillan. A structural induction theorem for processes. *Information and Computation*, 117(1):1–11, 1995.

[Lam78] Leslie Lamport. Time, clocks, and the ordering of events in a distributed system. *Communications of the ACM*, 21(7):558–564, 1978.

[Low96] Gavin Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. Workshop on Tools and Algorithms for the Construction and Analysis of Systems (TACAS)*, volume 1055 of *Lecture Notes in Computer Science*, pages 147–166. Springer-Verlag, 1996.

[Low99] Gavin Lowe. Towards a completeness result for model checking of security protocols. *The Journal of Computer Security*, 7(2/3):89–146, 1999.

[LR97] Gavin Lowe and Bill Roscoe. Using CSP to detect errors in the TMN protocol. *IEEE Transactions on Software Engineering*, 23(10):659–669, 1997.

[MCF87] Jonathan K. Millen, Sidney C. Clark, and Sheryl B. Freedman. The Interrogator: protocol security analysis. *IEEE Transactions on Software Engineering*, SE-13(2):274–288, February 1987.

[MCJ97] Will Marrero, Edmund Clarke, and Somesh Jha. A model checker for authentication protocols. In *Proc. DIMACS Workshop on Design and Formal Verification of Security Protocols*, 1997.

[MMS97] John C. Mitchell, Mark Mitchell, and Ulrich Stern. Automated analysis of cryptographic protocols using Murϕ. In *Proc. 18th IEEE Symposium on Research in Security and Privacy*, pages 141–153. IEEE Computer Society Press, 1997.

[MSS98] John C. Mitchell, Vitaly Shmatikov, and Ulrich Stern. Finite-state analysis of SSL 3.0. In *Seventh USENIX Security Symposium*, pages 201–216, 1998.

[MvOV97] Alfred J. Menezes, Paul C. van Oorschot, and Scott A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1997.

[OPT97] Donal O'Mahony, Michael Peirce, and Hitesh Tewari. *Electronic Payment Systems.* Artech House, Boston, 1997.

[OR87] Dave Otway and Owen Rees. Efficient and timely mutual authentication. *Operating Systems Review*, 21(1):8–10, January 1987.

[Pau96] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *The Journal of Computer Security*, 6(1/2):85–128, 1996.

[RB99] A. W. Roscoe and P. J. Broadfoot. Proving security protocols with model checkers by data independence techniques. *The Journal of Computer Security*, 7(2/3), 1999.

[Ros95] A. W. Roscoe. Modelling and verifying key exchange protocols using CSP and FDR. In *Proc. 8th IEEE Computer Security Foundations Workshop (CSFW)*, pages 98–107. IEEE Computer Society Press, 1995.

[Ros98] A. W. Roscoe. Proving security protocols with model checkers by data independence techniques. In *Proc. 11th IEEE Computer Security Foundations Workshop (CSFW).* IEEE Computer Society Press, 1998.

[Son99] Dawn Xiaodong Song. Athena: A new efficient automatic checker for security protocol analysis. In *Proc. 12th IEEE Computer Security Foundations Workshop (CSFW).* IEEE Computer Society Press, 1999.

[THG98a] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Honest ideals on strand spaces. In *Proc. 11th IEEE Computer Security Foundations Workshop (CSFW).* IEEE Computer Society Press, June 1998.

[THG98b] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand space pictures. In *Workshop on Formal Methods and Security Protocols*, June 1998. Available via http://www.cs.bell-labs.com/who/nch/fmsp/index.html.

[THG98c] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Why is a security protocol correct? In *Proc. 18th IEEE Symposium on Research in Security and Privacy.* IEEE Computer Society Press, 1998.

[THG99] F. Javier Thayer Fábrega, Jonathan C. Herzog, and Joshua D. Guttman. Mixed strand spaces. In *Proc. 12th IEEE Computer Security Foundations Workshop (CSFW)*, pages 72–82. IEEE Computer Society Press, June 1999.

[WL93] Thomas Y. C. Woo and Simon S. Lam. A semantic model for authentication protocols. In *Proc. 14th IEEE Symposium on Research in Security and Privacy*, pages 178–194. IEEE Computer Society Press, May 1993.

[WL94] Thomas Y.C. Woo and Simon S. Lam. A lesson in authentication protocol design. *ACM Operating Systems Review*, 28(3):24–37, July 1994.

A Uniquely-Originated Genvals in *pik*

The definition of history requires that values of uniquely-originated parameters for uncompromised strands are not in genvals(pik). This reflects the intuition that uniquely-originated genvals are fresh. However, it might seem surprising to require that values of uniquely-originated parameters in a history $\langle tr, \rightarrow, role\rangle$ satisfy a (slightly) stronger requirement than that they be uniquely-originated in tr. Consider a modified definition obtained by omitting the requirement that values of uniquely-originated parameters for uncompromised strands are not in genvals(pik); this defines a *weak history.* We show that the two definitions are equivalent from the perspective of correctness of systems.

Lemma 4. Let $h = \langle tr, \rightarrow, role\rangle$ be a weak history of a system $\langle \Pi, \langle pik, compr\rangle\rangle$. Suppose there is an uncompromised regular strand s in h and a uniquely-originated parameter $x \in \mathrm{dom}(args(role(s), tr(s)))$

such that the genval $g = \mathit{args}(\mathit{role}(s), \mathit{tr}(s))(x)$ is in genvals(pik). Let h_1 be obtained from h by replacing all occurrences of g with a genval g_1 that is not in genvals$(\mathrm{term}_h(\mathcal{N}_h) \cup \mathit{pik})$. Then h_1 is a weak history of $\langle \Pi, \langle \mathit{pik}, \mathit{compr} \rangle \rangle$.

Proof: All of the conditions on weak histories are trivially preserved by the replacement except the strong typing assumption for strands for Msg and regular strands (note that compromisedness is preserved because genvals are excluded from the range of compromised-trace policies). Regarding strands for Msg, the unique-origination condition in the definition of weak history implies that h does not contain a strand s_{Msg} for Msg such that g occurs in $\mathrm{term}_h(\langle s_{\mathrm{Msg}}, 0 \rangle)$, so the replacement does not change the arguments of strands for Msg, so the strong typing assumption holds for all strands for Msg in h_1. For all regular strands s and all $x \in \mathrm{dom}(\mathit{args}(\mathit{role}(s), \mathit{tr}(s)))$, we need to show that replacing g with g_1 in $\mathit{args}(\mathit{role}(s), \mathit{tr}(s))(x)$ yields a term in the type of x. The roles of Π are assumed to be well-formed, and the first well-formedness condition on roles implies that the type of every parameter of a well-formed role is closed under substitutions that replace one genval with another genval. ∎

Lemma 5. For every system $\mathcal{M}$ and every correctness requirement ϕ, all weak histories of $\mathcal{M}$ satisfy ϕ iff all histories of $\mathcal{M}$ satisfy ϕ.

Proof: The forward direction ($\Rightarrow$) of the "iff" follows immediately from the fact that the set of histories of $\mathcal{M}$ is a subset of the set of weak histories of $\mathcal{M}$. For the reverse direction ($\Leftarrow$), we show the contrapositive, *i.e.*, we suppose $\mathcal{M}$ has a weak history h that violates ϕ and show that $\mathcal{M}$ has a history h_1 that violates ϕ. h_1 is obtained from h by performing the replacement described in Lemma 4 to every genval to which it is applicable. It is easy to show using Lemma 4 that h_1 is a history of $\mathcal{M}$. It remains to show that h_1 violates ϕ. This is straightforward if ϕ is an agreement requirement. Suppose ϕ is a genval secrecy requirement, and h violates ϕ by revealing a genval g. If g was replaced with a different genval g_1 during construction of h_1, then h_1 reveals g_1 and thereby violates ϕ; otherwise, h_1 reveals g and thereby violates ϕ. ∎

We adopt the stronger definition of history, because it simplifies some proofs (by factoring out the above reasoning). For example, Lemma 1 were expressed using weak histories, then for a weak history $\langle \mathit{tr}, \rightarrow, \mathit{role} \rangle$ that satisfies the hypotheses of Lemma 4 for some uncompromised strand s and some uniquely-originated parameter x of $\mathit{role}(s)$, the proof of Lemma 1 would need to consider the possibility that some tr_n contains a strand for Msg from which the genval $\mathit{args}(\mathit{role}(s), \mathit{tr}(s))(x)$ originates, which would cause a violation of the unique-origination condition in tr'.

B Proof that Needham-Schroeder-Lowe Protocol Satisfies BSR(f_1)

Lemma 6. Every regular strand s in every history $h = \langle \mathit{tr}, \rightarrow, \mathit{role} \rangle$ for $\mathcal{M}_{NSL}$ has a weak support in h with strand count at most f_1, where $f_1(r) = 1$ for all $r \in \Pi_{NSL}$.

Proof: We assume $\mathrm{len}(\mathit{tr}(s)) = \mathrm{len}(\mathit{role}(s))$, because shorter traces have smaller supports. Consider cases based on $\mathit{role}(s)$.

case 1: $\mathit{role}(s) = \mathrm{Init}_{NSL}$. Let i_I, r_I, ni_I, nr_I denote the arguments of Init_{NSL} in s (*i.e.*, $i_I = \mathit{args}(\mathrm{Init}_{NSL}, \mathit{tr}(s))(i)$, and so on).

case 1.1: s is compromised, *i.e.*, $r_I = P$. Then nodes$_h(s)$ is a weak support for s with strand count at most f_1, because $\langle s, 1\rangle$ is the only negative node on s, nr_I and ni_I are not in uniqOrigRqrd$_h^{\mathcal{M}_{NSL}}$(nodes$_h(s)$), and the other subterms of term$_{h'}(\langle s, 1\rangle)$ are always available to the penetrator from strands for Msg.

case 1.2: s is uncompromised. Let S be a $\subseteq$-minimal weak support for s in h. Let $h' = \langle tr', \rightarrow', role'\rangle$ be a history whose existence is implied by Lemma 1 applied to S. Note that $role'(s) = role(s)$ for all regular strands in h'. $\langle s, 1\rangle$ is the only negative node on s. Consider cases based on whether the ciphertext received by node $\langle s, 1\rangle$ originates from a regular node in h'. That ciphertext has the form $\{t_1 \cdot t_2 \cdot t_3\}_{t_4}$, where the t_i are primitive terms.

case 1.2.1: there exists a positive regular node $\langle s_R, j_R\rangle \in$ preds$_{h'}(\langle s, 1\rangle)$ with abs(term$_{h'}(\langle s_R, j_R\rangle)$) = abs(term$_{h'}(\langle s, 1\rangle)$). Traces for Init$_{NSL}$ do not contain positive terms with subterms of the form $\{t_1 \cdot t_2 \cdot t_3\}_{t_4}$, so $role'(s_R) =$ Resp$_{NSL}$ and $j_R = 1$. Let i_R, r_R, ni_R, nr_R denote the arguments of Resp$_{NSL}$ in s_R. The equality abs(term$_{h'}(\langle s_R, j_R\rangle)$) = abs(term$_{h'}(\langle s, 1\rangle)$) implies $ni_I = ni_R \wedge nr_I = nr_R \wedge r_I = r_R$. S and $\mathcal{N}_{tr'}$ contain nodes$_h(s) \cup \{\langle s_R, 0\rangle, \langle s_R, 1\rangle\}$ (they also contain a weak support for $\langle s_R, 0\rangle$). Consider cases based on whether the ciphertext received by $\langle s_R, 0\rangle$ originates from a regular node in h'.

case 1.2.1.1: there exists a positive regular node $\langle s', j'\rangle \in$ preds$_{h'}(\langle s_R, 0\rangle)$ with abs(term$_{h'}(\langle s', j'\rangle)$) = abs(term$_{h'}(\langle s_R, 0\rangle)$). Traces for Resp$_{NSL}$ do not contain positive terms with subterms of the form $\{t_1 \cdot t_2\}_{t_3}$, where the t_i are primitive terms, so $role'(s') =$ Init$_{NSL}$ and $j' = 0$. Let i', r', ni', nr' denote the arguments of Init$_{NSL}$ in s'. The equality abs(term$_{h'}(\langle s', j'\rangle)$) = abs(term$_{h'}(\langle s_R, 0\rangle)$) implies $ni' = ni_R \wedge i' = i_R \wedge r' = r_R$. $ni' = ni_R$ and $ni_I = ni_R$ together imply $ni' = ni_I$. Init$_{NSL}.ni$ is uniquely-originated, so $s' = s$ or s and s' are compromised. In case 1.2, s is uncompromised, so $s' = s$. S is $\subseteq$-minimal, so len$(tr'(s_R)) = j_R$, so $\langle s_R, 2\rangle \notin \mathcal{N}_{h'}$. Thus, $S =$ nodes$_h(s) \cup \{\langle s_R, 0\rangle, \langle s_R, 1\rangle\}$ is a weak support for s in h with strand count at most f_1.

case 1.2.1.2: there does not exist a positive regular node $\langle s', j'\rangle \in$ preds$_{h'}(\langle s_R, 0\rangle)$ with abs(term$_{h'}(\langle s', j'\rangle)$) = abs(term$_{h'}(\langle s_R, 0\rangle)$). Then there exists a positive penetrator node $\langle s', j'\rangle \in$ preds$_{h'}(\langle s_R, 0\rangle)$ such that abs(term$_{h'}(\langle s', j'\rangle)$) = abs(term$_{h'}(\langle s_R, 0\rangle)$), $role'(s') =$ Enc, and $tr'(s') =$ Enc$(ni_R \cdot i_R, pubkey(r_R))$. We show that this case is impossible. $\langle s', 0\rangle$ is a negative penetrator node in which ni_R occurs in the clear. $ni_R = ni_I$, so $\mathcal{N}_{h'} \vdash_{h'}^{\mathcal{M}_{NSL}} ni_I$. $\mathcal{M}_{NSL}$ is known to satisfy the genval secrecy requirement $\{$Init$_{NSL}.ni$, Init$_{NSL}.nr$, Resp$_{NSL}.ni$, Resp$_{NSL}.nr\}$, so s is compromised, contradicting the hypothesis of case 1.2.

case 1.2.2: there does not exist a positive regular node $\langle s_R, j_R\rangle \in$ preds$_{h'}(\langle s, 1\rangle)$ with abs(term$_{h'}(\langle s_R, j_R\rangle)$) = abs(term$_{h'}(\langle s, 1\rangle)$). Then there exists a positive penetrator node $\langle s', j'\rangle \in$ preds$_{h'}(\langle s, 1\rangle)$ such that abs(term$_{h'}(\langle s', j'\rangle)$) = abs(term$_{h'}(\langle s, 1\rangle)$), $role'(s') =$ Enc, and $tr'(s') =$ Enc$(ni_I \cdot nr_I \cdot r_I, pubkey(i_I))$. We show that this case is impossible. As in case 1.2.1.2, genval secrecy implies s is compromised, a contradiction.

case 2: $role(s) =$ Resp$_{NSL}$. Let i_R, r_R, ni_R, nr_R denote the arguments of Resp$_{NSL}$ in s.

case 2.1: s is compromised, *i.e.*, $i_R = P$. By reasoning similar to that in case 1.1, nodes$_h(s)$ is a weak support for s.

case 2.2: s is uncompromised. Consider cases based on whether the ciphertext received by $\langle s, 2\rangle$ originates from a regular node.

case 2.2.1: there exists a positive regular node $\langle s_I, j_I\rangle \in \text{preds}_{h'}(\langle s, 2\rangle)$ with $\text{abs}(\text{term}_{h'}(\langle s_I, j_I\rangle)) = \text{abs}(\text{term}_{h'}(\langle s, 2\rangle))$. Traces for Resp_{NSL} do not contain positive terms with subterms of the form $\{t_1\}_{t_2}$, where the t_i are primitive terms, so $\mathit{role}'(s_I) = \text{Init}_{NSL}$ and $j_I = 1$. Let i_I, r_I, ni_I, nr_I denote the arguments of Init_{NSL} in s_I. The equality $\text{abs}(\text{term}_{h'}(\langle s_I, j_I\rangle)) = \text{abs}(\text{term}_{h'}(\langle s, 2\rangle))$ implies $nr_I = nr_R \wedge r_I = r_R$. If s_I were compromised (*i.e.*, $r_I = P$), then $r_R = P$, which is impossible, because the type of $\text{Resp}_{NSL}.r$ does not contain P. Thus, s_I is uncompromised.

Let S be a $\subseteq$-minimal weak support for s_I. Consider cases based on whether the ciphertext received by $\langle s_I, 1\rangle$ originates from a regular node in h'.

case 2.2.1.1: there exists a positive regular node $\langle s', j'\rangle \in \text{preds}_{h'}(\langle s_I, 1\rangle)$ with $\text{abs}(\text{term}_{h'}(\langle s', j'\rangle)) = \text{abs}(\text{term}_{h'}(\langle s_I, 1\rangle))$. Then s_I falls in case 1.2.1, so (as argued there) $\mathit{role}'(s') = \text{Resp}_{NSL}$ and $j' = 1$. Let i', r', ni', nr' denote the arguments of Resp_{NSL} in s'. Case 1.2.1.2 is impossible, so s_I falls in case 1.2.1.1, *i.e.*, the ciphertext received by $\langle s', 0\rangle$ originates from a regular node (namely, $\langle s_I, 0\rangle$) in h'. The equality $\text{abs}(\text{term}_{h'}(\langle s', j'\rangle)) = \text{abs}(\text{term}_{h'}(\langle s_I, 1\rangle))$ implies $nr' = nr_I$. $nr' = nr_I$ and $nr_I = nr_R$ together imply $nr' = nr_R$. $\text{Resp}_{NSL}.nr$ is uniquely-originated, so $s' = s$ or s and s' are compromised. In case 2.2, s is uncompromised, so $s' = s$. It is easy to show that $\text{nodes}_h(s_I) \cup \text{nodes}_h(s)$ is a weak support for both s_I and s in h with strand count f_1.

case 2.2.1.2: there does not exist a positive regular node $\langle s', j'\rangle \in \text{preds}_{h'}(\langle s_I, 1\rangle)$ with $\text{abs}(\text{term}_{h'}(\langle s', j'\rangle)) = \text{abs}(\text{term}_{h'}(\langle s_I, 1\rangle))$. This case is impossible, because s_I falls in case 1.2.2, which is impossible.

case 2.2.2: there does not exist a positive regular node $\langle s_I, j_I\rangle \in \text{preds}_{h'}(\langle s, 2\rangle)$ with $\text{abs}(\text{term}_{h'}(\langle s_I, j_I\rangle)) = \text{abs}(\text{term}_{h'}(\langle s, 2\rangle))$. We show that this case is impossible. By reasoning similar to that in case 1.2.1.2, genval secrecy implies s is compromised, a contradiction. ∎

Lemma 7. $\mathcal{M}_{NSL}$ satisfies $BSR(f_1)$.

Proof: This follows from Lemma 6 and the observation that, the set over which g ranges in condition Supp2 is empty, mainly because no genvals that are required to be uniquely-originated are revealed to the penetrator, *i.e.*, $\mathcal{M}_{NSL}$ satisfies the genval secrecy requirement $\{\text{Init}_{NSL}.ni, \text{Resp}_{NSL}.nr\}$. ∎