Logic of Secrets in Collaboration Networks

Sara Miner More

Department of Mathematics and Computer Science
McDaniel College, Westminster, Maryland 21157, USA

Pavel Naumov

Department of Mathematics and Computer Science
McDaniel College, Westminster, Maryland 21157, USA

Abstract

The article proposes *Logic of Secrets in Collaboration Networks*, a formal logical system for reasoning about a set of secrets established over a fixed configuration of communication channels. The system's key feature, a multi-channel relation called *independence*, is a generalization of a two-channel relation known in the literature as *nondeducibility*. The main result is the completeness of the proposed system with respect to a semantics of secrets.

Keywords: information flow, nondeducibility, independence, axiomatization

1. Introduction

Suppose several parties are connected by communication channels that form a network with a fixed topology. In this setting, which we call a *collaboration network*, a pair of parties connected by a channel uses this channel to establish a secret. If the pairs of parties establish their secrets completely independently from other pairs, then possession of one or several of these secrets reveals no information about the other secrets. Assume, however, that secrets are not picked completely independently. Instead, each party with access to multiple channels may enforce some desired interdependency between the secrets it shares with other parties. These "local" interdependencies between secrets known to a single party may result in a "global" interdependency between several secrets, not all of which are known to any single party. Given the fixed topology of the collaboration network, we study what global interdependencies between secrets may exist in the system.

Consider, for example, the collaboration network depicted in Figure 1. Suppose that the parties collaborate according to the following protocol. Party P

Email addresses: smore@mcdaniel.edu (Sara Miner More), pnaumov@mcdaniel.edu (Pavel Naumov)

picks a random value a from $\{0,1\}$ and sends it to party Q. Party Q picks values b and c from $\{0,1\}$ in such a way that $a = b + c \mod 2$ and sends both of these values to R. Party R computes $d = b + c \mod 2$ and sends value d to party S. In this protocol, it is clear that the values of a and d will always match. We view a, b, c, and d as secrets, conditions $a = b + c \mod 2$ and $d = b + c \mod 2$ as local interdependencies, and condition $a = d \mod 2$ as a global interdependency.

Note that in the above example, all channels transmit messages in one direction and, thus, the channel network forms a directed graph. However, in the more general setting, two parties might establish the value of a secret through a dialog over their communication channel, with messages traveling in both directions. Thus, in general, we will not assume any specific direction on a channel.

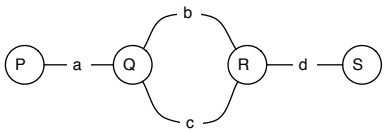

Figure 1: Collaboration network N_1.

If two or more secrets are not interdependent, then we will say that they are *independent* (a formal definition of independence will be given in Definition 4). In the logical system presented in this article we use independence, not interdependence, as the basic notion simply because it produces a slightly more elegant system. Another way to define independence is to say that secrets are independent if any values of these secrets that can occur in the protocol can also occur simultaneously. For example, secrets a and b in the above protocol are independent, but secrets a and d are not. Furthermore, although secrets a, b, c in the above protocol are all pairwise independent, the three secrets considered together are not independent.

The independence examples that we have given so far are for a single protocol, subject to a particular set of local interdependencies between secrets. If the topology remains fixed, but the protocol is changed, then secrets which were previously independent could become interdependent, and vice versa. In this article, however, we study the independence of secrets that follow from the topological structure of the network of channels, no matter which specific protocol is used.

For example, it is relatively easy to see that for collaboration network N_2 in Figure 2, if secrets a and b are independent, then secrets a and c are also independent, regardless of the protocol used. This is a property of the network topology, not of the protocol. We say that $[a, b] \to [a, c]$ is true on topology N_2, where $[a, b]$ is our notation for the independence of secrets a and b. Another less obvious property of independence is true for collaboration network N_1, which defines the network topology in Figure 1. Namely, if channels a, b, and c are independent, then channels a and d are independent: that is, $[a, b, c] \to [a, d]$ is true on N_1. As a final ex-

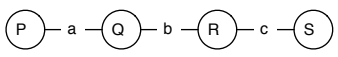

Figure 2: $[a, b] \to [a, c]$ holds on N_2.

ample, consider collaboration network N_3 in Figure 3, where the property $[b, c] \rightarrow ([a, e] \rightarrow [a, d])$ holds. In Section 6, we will prove each of these claims.

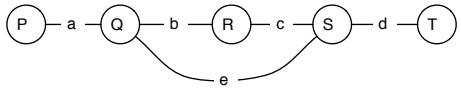

Figure 3: $[b, c] \rightarrow ([a, e] \rightarrow [a, d])$ holds on N_3.

In this article, we present a logic that describes the independence properties of any network topology. The deductive system for this logic operates with binary relation $N \vdash \phi$, where N is a collaboration network that specifies a network topology, and ϕ is a propositional statement about secret independence. Our main results are the soundness (see Theorems 5-7) and completeness (see Theorem 8) of this deductive system with respect to the intended protocol semantics. It is interesting to note that one of the inference rules of this deductive system modifies not only a formula ϕ, but the network N as well. The formulas in this logic capture properties of a fixed topology, but the logic itself modifies the topology as part of a derivation. This makes our formal system very different from the traditional deductive systems in mathematical logic.

Our work is related to the study of information flow. Most of the literature in this area, however, studies information flow from the language-based [1, 2] or probabilistic [3, 4] points of view. Historically ([5], page 185), one of the first attempts to capture independence in our sense was undertaken by Goguen and Meseguer [6] through their notion of *noninterference* between two computing devices. Later, Sutherland [7] introduced a *no information flow* relation, which is essentially our independence relation restricted to two-element sets. This relation has since become known in the literature as *nondeducibility*. Cohen [8] presented a related notion called *strong dependence*. Unlike nondeducibility, however, the strong dependence relation is not symmetric. More recently, Halpern and O'Neill [3, 4] introduced f-secrecy to reason about multiparty protocols. The f-secrecy predicate is a version of nondeducibility that can refer to a value of a certain function of the secret rather than the secret itself. However, all of these works focus on the application of the independence relation in the analysis of secure protocols, whereas the main focus of our work is on logical properties of the relation itself. This article is a significant revision of an earlier conference paper [9].

2. Protocol: A Formal Definition

Throughout this article, we assume a fixed infinite alphabet of variables $a, b, \ldots$, that we refer to as "secret variables". By a network topology we mean a collaboration network whose edges, or "channels", are labeled by secret variables. We allow multiple edges and loops. The set of all channels of collaboration network N will be denoted by $Ch(N)$. One channel may have several labels, but the same label can be assigned to only one channel. Given this, we will often informally refer to "the channel labeled with a" as simply "channel a".

Definition 1. *A semi-protocol over a collaboration network N is a pair $\langle V, L\rangle$ such that*

1. *$V(c)$ is an arbitrary set of "values" for each channel $c \in Ch(N)$,*
2. *$L = \{L_p\}_{p\in P}$ is a family of predicates, indexed by parties of N, which we call "local conditions". If $c_1, \dots c_k$ is the list of all channels incident with party p, then L_p is a predicate on $V(c_1) \times \cdots \times V(c_k)$.*

Definition 2. *A run of a semi-protocol $\langle V, L\rangle$ is a function r such that*

1. *$r(c) \in V(c)$ for any channel $c \in Ch(N)$,*
2. *If $c_1, \dots c_k$ is the list of all secrets incident with party $p \in P$, then predicate $L_p(r(c_1), \dots, r(c_k))$ is true.*

Definition 3. *A protocol is any semi-protocol that has at least one run.*

The set of all runs of a protocol $\mathcal{P}$ is denoted by $\mathcal{R}(\mathcal{P})$. We conclude this section with the key definition of this article. It is a multi-argument version of Sutherland's binary nondeducibility predicate that we call *independence*.

Definition 4. *A set of channels $Q = \{q_1, \dots, q_k\}$ is called independent under protocol $\mathcal{P}$ if for any runs $r_1, \dots, r_k \in \mathcal{R}(\mathcal{P})$ there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(q_i) = r_i(q_i)$ for any $i \in \{1, \dots, k\}$.*

Definition 5. *A protocol $\mathcal{P} = \langle V, L\rangle$ is called finite if the set $V(c)$ is finite for every $c \in Ch(N)$.*

3. Language of Secrets

Informally, by $\Phi(N)$, we denote the set of all properties of secrets in collaboration network N. Formally, $\Phi(N)$ is a minimal set defined recursively as follows: (i) for any finite set of secret variables $\{a_1, \dots, a_n\} \subseteq Ch(N)$, formula $[a_1, \dots, a_n]$ belongs to set $\Phi(N)$, (ii) the false constant $\bot$ belongs to $\Phi(N)$, and (iii) for any formulas ϕ and $\psi \in \Phi(N)$, the implication $\phi \to \psi$ also belongs to $\Phi(N)$. As usual, we assume that conjunction, disjunction, and negation are defined through $\to$ and $\bot$.

Next, we define relation $\mathcal{P} \vDash \phi$. Informally, it means that formula ϕ is true under protocol $\mathcal{P}$.

Definition 6. *For any protocol $\mathcal{P}$ over a collaboration network N, and any formula $\phi \in \Phi(N)$, we define the relation $\mathcal{P} \vDash \phi$ recursively as follows:*

1. *$\mathcal{P} \nvDash \bot$,*
2. *$\mathcal{P} \vDash [a_1, \dots, a_n]$ if the set of channels $\{a_1, \dots, a_n\}$ is independent under protocol $\mathcal{P}$,*
3. *$\mathcal{P} \vDash \phi_1 \to \phi_2$ if $\mathcal{P} \nvDash \phi_1$ or $\mathcal{P} \vDash \phi_2$.*

In this article, we study the set of formulas that are true under *any* protocol $\mathcal{P}$ as long as collaboration network N remains fixed. The set of all such formulas will be captured by the *Logic of Secrets in Collaboration Networks*. Below, we will list axioms and inference rules for this logic and prove their soundness and completeness.

4. Graph Notation

In preparation for the presentation of an inference rule used in our system, we introduce a graph operation called *truncation*. As usual, a *cut* of a graph is a disjoint partitioning of the nodes of the graph into two sets. A *crossing edge* in a cut is an edge whose ends belong to different sets of the partition. For any set of nodes X of a graph N we use $E(X)$ to denote the set of edges of N whose ends both belong to X.

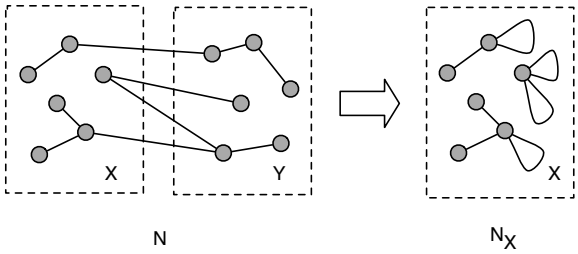

Figure 4: Graph truncation.

Definition 7. *Let N be an arbitrary graph and (X, Y) be an arbitrary cut of N (See Figure 4). We define the "truncation" graph N_X of graph N as follows:*

1. *The nodes of graph N_X are the nodes of set X.*
2. *The edges of N_X are all of the edges from $E(X)$ plus the crossing edges of the cut (X, Y) modified in the following way: if in graph N, a crossing edge c connects node $n \in X$ with node $m \in Y$, then in graph N_X, edge c loops from n back into n.*

Each edge e in N_X corresponds to a unique edge in N. Although the two corresponding edges might connect different nodes in their respective graphs, we will refer to both of them as edge e. From context, it will be clear to which of the two edges we are referring.

To close this section, we define the concept of a gateway between two sets of edges in a graph, which is used in an axiom introduced in the following section.

Definition 8. *A gateway between sets of edges A and B in a graph N is a set of edges G such that every path from A to B contains at least one edge from G.*

Note that sets A, B, and G are not necessarily disjoint. Thus, for example, for any set of edges A, set A is a gateway between A and itself. Also, note that the empty set is a gateway between any two components of the graph that are not connected one to another.

5. Formal System: Axioms and Rules

We are now ready to describe the *Logic of Secrets in Collaboration Networks*. We will write $N \vdash \phi$ to state that formula $\phi \in \Phi(N)$ is provable in this logic. Everywhere below by X, Y means union of sets X and Y. The deductive system for this logic, in addition to propositional tautologies and Modus Ponens

inference rule, consists of the *Small Set* axiom, the *Gateway* axiom, and the *Truncation* inference rule, defined below:

Small Set Axiom. Any set that contains less than two elements is independent: $N \vdash [A]$, where $A \subseteq Ch(N)$ and $|A| < 2$.

Gateway Axiom. $N \vdash [A, G] \rightarrow ([B] \rightarrow [A, B])$, where G is a gateway between sets of channels A and B in collaboration network N such that $A \cap G = \varnothing$.

Truncation Rule. Let $C \subseteq Ch(N)$ be the set of all crossing channels of a cut (X, Y) of collaboration network N and $\phi \in \Phi(N_X)$. If $N_X \vdash \phi$, then $N \vdash [C] \rightarrow \phi$.

The soundness of this system will be demonstrated in Section 7.

6. Examples of Proofs

In this section we provide examples of proofs in the *Logic of Secrets in Collaboration Networks*.

Theorem 1. *$N_2 \vdash [a, b] \rightarrow [a, c]$, where N_2 is shown in Figure 2.*

Proof. Note that the single-element set $\{b\}$ is a gateway between sets $\{a\}$ and $\{c\}$. Thus, by the *Gateway* axiom, $N_2 \vdash [a, b] \rightarrow ([c] \rightarrow [a, c])$. By the *Small Set* axiom, $N_2 \vdash [c]$. Therefore, $N_2 \vdash [a, b] \rightarrow [a, c]$. □

Theorem 2. *$N_1 \vdash [a, b, c] \rightarrow [a, d]$, where N_1 is shown in Figure 1.*

Proof. Note that set $\{b, c\}$ is a gateway between sets $\{a\}$ and $\{d\}$. Thus, by the *Gateway* axiom, $N_1 \vdash [a, b, c] \rightarrow ([d] \rightarrow [a, d])$. By the *Small Set* axiom, $N_1 \vdash [d]$. Therefore, $N_1 \vdash [a, b, c] \rightarrow [a, d]$. □

Theorem 3. *$N_3 \vdash [b, c] \rightarrow ([a, e] \rightarrow [a, d])$, where N_3 is shown in Figure 3.*

Proof. The cut $(\{P, Q, S, T\}, \{R\})$ of collaboration network N_3 has crossing edges b and c. A truncation along this cut yields collaboration network N_3' (see Figure 5). In N_3', set $\{e\}$ is a gateway between sets $\{a\}$ and $\{d\}$. Thus, by the *Gateway* axiom, we have $N_3' \vdash [a, e] \rightarrow ([d] \rightarrow [a, d])$. By the *Small Set* axiom, $N_3' \vdash [a, e] \rightarrow [a, d]$. Lastly, by the *Truncation* inference rule, we conclude that $N_3 \vdash [b, c] \rightarrow ([a, e] \rightarrow [a, d])$. □

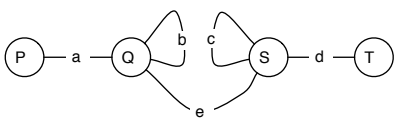

Figure 5: Collaboration network N_3' (shown) is a truncation of N_3 from Figure 3.

Finally, we present a general result to which we will refer during the proof of completeness in Section 8.

Theorem 4 (monotonicity). *$N \vdash [A] \to [B]$, for any collaboration network N and any subsets $B \subseteq A \subseteq Ch(N)$.*

Proof. Consider sets B and $\varnothing$. Since there are no paths connecting these sets, any set of channels is a gateway between these sets. In particular $(A \setminus B)$ is such a gateway. Taking into account that sets B and $(A \setminus B)$ are disjoint, by the *Gateway* axiom, $N \vdash [B, (A \setminus B)] \to ([\varnothing] \to [B])$. By the *Small Set* axiom, $N \vdash [B, (A \setminus B)] \to [B]$. By the assumption $B \subseteq A$, we conclude that $N \vdash [A] \to [B]$. □

7. Soundness

The proof of soundness, particularly the soundness of the *Gateway* axiom and the *Truncation* rule, is non-trivial. For each axiom and inference rule, we provide its justification as a separate theorem.

Theorem 5 (Small Set). *For any collaboration network N, if $\mathcal{P}$ is an arbitrary protocol over N and any $A \subseteq Ch(N)$ has at most one element, then $\mathcal{P} \vDash [A]$.*

Proof. If $A = \varnothing$, then $\mathcal{P} \vDash [A]$ follows from the existence of at least one run of any protocol. If $A = \{a_1\}$, consider any run $r_1 \in \mathcal{R}(\mathcal{P})$. Pick r to be r_1. This guarantees that $r(a_1) = r_1(a_1)$. □

Theorem 6 (Gateway). *For any collaboration network $N = \langle V, E \rangle$, and any gateway G between sets of channels A and B in N, if $\mathcal{P} \vDash [A, G]$, $\mathcal{P} \vDash [B]$, and $A \cap G = \varnothing$, then $\mathcal{P} \vDash [A, B]$.*

Proof. Assume $\mathcal{P} \vDash [A, G]$, $\mathcal{P} \vDash [B]$, and $A \cap G = \varnothing$. Let $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_k\}$. Consider any $r_1, \ldots, r_{n+k}$. It will be sufficient to show that there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(a_i) = r_i(a_i)$ for any $i \leq n$ and $r(b_i) = r_{n+i}(b_i)$ for any $i \leq k$. By the assumption $\mathcal{P} \vDash [B]$, there is a run $r_B \in \mathcal{R}(\mathcal{P})$ such that

$$r_B(b_i) = r_{n+i}(b_i) \qquad \text{for every } i \leq k. \tag{1}$$

By assumptions $\mathcal{P} \vDash [A, G]$ and $A \cap G = \varnothing$, there must be a run r_A such that

$$r_A(c) = \begin{cases} r_i(c) & \text{if } c = a_i \text{ for } i \leq n, \\ r_B(c) & \text{if } c \in G. \end{cases} \tag{2}$$

Next, consider collaboration network N' obtained from N by the removal of all channels in G. By the definition of a gateway, no single connected component

of network N' can contain both a channel from set A and a channel from set $(B \setminus G)$. Let us divide all connected components of N' into two subgraphs N'_A and N'_B such that N'_A contains no channels from $(B \setminus G)$ and N'_B contains no channels from A. Components that do not contain channels from either A or $(B \setminus G)$ can be arbitrarily assigned to either N'_A or N'_B.

By equation (2), runs r_A and r_B on N agree on each channel of gateway G. We will now construct a combined run r by "sewing together" portions of r_A and r_B with the "stitches" placed along gateway G. Formally,

$$r(c) = \begin{cases} r_A(c) & \text{if } c \in N'_A, \\ r_A(c) = r_B(c) & \text{if } c \in G, \\ r_B(c) & \text{if } c \in N'_B. \end{cases} \tag{3}$$

Let us first prove that r is a valid run of the protocol $\mathcal{P}$. For this, we need to prove that it satisfies local conditions L_p at every party p. Without loss of generality, assume that $p \in N'_A$. Hence, on all channels incident with p, run r agrees with run r_A. Thus, run r satisfies L_p simply because r_A does.

Next, we will show that $r(a_i) = r_i(a_i)$ for any $i \leq n$. Indeed, by equations (2) and (3), $r(a_i) = r_A(a_i) = r_i(a_i)$. Finally, we will need to show that $r(b_i) = r_{n+i}(b_i)$ for any $i \leq k$. This, however, follows easily from equations (1) and (3). □

Theorem 7 (Truncation). *Assume that (X, Y) is a cut of collaboration network N, set C is the set of all crossing channels of this cut, and ϕ is a formula in $\Phi(N_X)$. If $\mathcal{P}' \vDash \phi$ for every protocol $\mathcal{P}'$ over truncation N_X, then $\mathcal{P} \vDash [C] \to \phi$ for every protocol $\mathcal{P}$ over network N.*

Proof. Suppose that there is a protocol $\mathcal{P}$ over N such that $\mathcal{P} \vDash [C]$, but $\mathcal{P} \nvDash \phi$. We will construct a protocol $\mathcal{P}'$ over N_X such that $\mathcal{P}' \nvDash \phi$.

Let $\mathcal{P} = \langle V, L \rangle$. Note that, for any channel e, not all values from $V(e)$ may actually be used in the runs of this protocol. Some values might be excluded by the particular local conditions of $\mathcal{P}$. To construct protocol $\mathcal{P}' = \langle V', L' \rangle$ over truncation N_X, for any channel e of N_X we first define $V'(e)$ as the set of values that are actually used by at least one run of protocol $\mathcal{P}$:

$$V'(e) = \{r(e) \mid r \in \mathcal{R}(\mathcal{P})\}.$$

The local condition L'_p at any party p of truncation N_X is the same as under protocol $\mathcal{P}$. To show that protocol $\mathcal{P}'$ has at least one run, notice that the restriction of any run of $\mathcal{P}$ to channels in N_X constitutes a valid run of $\mathcal{P}'$.

Lemma 1. *For any run $r' \in \mathcal{R}(\mathcal{P}')$ there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(e) = r'(e)$ for each channel e in truncation N_X.*

Proof. Consider any run $r' \in \mathcal{R}(\mathcal{P}')$. By the definition of V', for any e in cut C there is a run $r_e \in \mathcal{R}(\mathcal{P})$ such that $r'(e) = r_e(e)$. Since $\mathcal{P} \vDash [C]$, there is a run $r_Y \in \mathcal{R}(\mathcal{P})$ such that $r_Y(e) = r_e(e) = r'(e)$ for any $e \in C$.

We will now construct a combined run $r \in \mathcal{R}(\mathcal{P})$ by "sewing" together r_Y and r' with the "stitches" placed in set C. Recall that we use the notation $E(X)$ to denote channels whose ends are both in set X. Formally, let

$$r(e) = \begin{cases} r'(e) & \text{if } e \in E(X), \\ r'(e) = r_Y(e) & \text{if } e \in C, \\ r_Y(e) & \text{if } e \in E(Y). \end{cases}$$

We just need to show that r satisfies L_p at every party p of collaboration network N. Indeed, if $p \in Y$, then run r is equal to r_Y on all channels incident with p. Thus, it satisfies the local condition because run r_Y does. Alternatively, if $p \in X$, then run r is equal to run r' on all channels incident with p. Since r' satisfies local condition L'_p and, by definition, $L'_p \equiv L_p$, we can conclude that r again satisfies condition L_p. □

Lemma 2. *For any set of channels $Q = \{q_1, \ldots, q_n\}$ in collaboration network N_X,*

$$\mathcal{P} \vDash [Q] \quad \textit{iff} \quad \mathcal{P}' \vDash [Q].$$

Proof. Assume first that $\mathcal{P} \vDash [Q]$ and consider any runs $r'_1, \ldots, r'_n \in \mathcal{R}(\mathcal{P}')$. We will construct a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q_i) = r'_i(q_i)$ for every $i \in \{1, \ldots, n\}$. Indeed, by Lemma 1, there are runs $r_1, \ldots, r_n \in \mathcal{R}(\mathcal{P})$ that match runs $r'_1, \ldots, r'_n$ on all channels in N_X. By the assumption that $\mathcal{P} \vDash [Q]$, there must be a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(q_i) = r_i(q_i)$ for all $i \in \{1, \ldots, n\}$. Hence, $r(q_i) = r_i(q_i) = r'_i(q_i)$ for all $i \in \{1, \ldots, n\}$. Let r' be the restriction of run r to the channels in N_X. Since the local conditions of protocols $\mathcal{P}$ and $\mathcal{P}'$ are the same, $r' \in \mathcal{R}(\mathcal{P}')$. Finally, we notice that $r'(q_i) = r(q_i) = r'_i(q_i)$ for any $i \in \{1, \ldots, k\}$.

Next, assume that $\mathcal{P}' \vDash [Q]$ and consider any runs $r_1, \ldots, r_n \in \mathcal{R}(\mathcal{P})$. We will show that there is a run $r \in \mathcal{R}(\mathcal{P})$ such that $r(q_i) = r_i(q_i)$ for all $i \in \{1, \ldots, n\}$. Indeed, let $r'_1, \ldots, r'_n$ be the restrictions of runs $r_1, \ldots, r_n$ to the channels in N_X. Since the local conditions of these two protocols are the same, $r'_1, \ldots, r'_n \in \mathcal{R}(\mathcal{P}')$. By the assumption that $\mathcal{P}' \vDash [Q]$, there is a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q_i) = r'_i(q_i) = r_i(q_i)$ for all $i \in \{1, \ldots, n\}$. By Lemma 1, there is a run $r \in \mathcal{R}(\mathcal{P})$ that matches r' everywhere in N_X. Therefore, $r(q_i) = r'(q_i) = r_i(q_i)$ for all $i \in \{1, \ldots, n\}$. □

Lemma 3. *For any formula $\psi \in \Phi(N_X)$, $\mathcal{P} \vDash \psi$ if and only if $\mathcal{P}' \vDash \psi$.*

Proof. We use induction on the complexity of ψ. The base case follows from Lemma 2, and the induction step is trivial. □

The statement of Theorem 7 immediately follows from Lemma 3. □

8. Completeness

Our main result is the following completeness theorem for the *Logic of Secrets in Collaboration Networks*:

Theorem 8. *For any collaboration network N, if $\mathcal{P} \vDash \phi$ for all finite protocols $\mathcal{P}$ over N, then $N \vdash \phi$.*

At the core of the proof is the construction of a finite protocol. This protocol will be formed as a composition of several simpler protocols, where each of the simpler protocols is defined recursively. The base case of this recursive definition is the parity protocol defined below.

8.1. Parity Protocol

Let N be a collaboration network and A be a subset of $Ch(N)$. We define the "parity protocol" $\mathcal{P}_A$ over N as follows. The set of values of any channel c in collaboration network N is a set of pairs such that

$$V(c) = \begin{cases} \{\langle b_1, b_2\rangle \mid b_1, b_2 \in \{0,1\}\} & \text{if } c \in A \\ \{\langle b, b\rangle \mid b \in \{0,1\}\} & \text{if } c \notin A \end{cases}$$

This means that under each run $r \in \mathcal{P}_A$, the value of each channel will be a pair. We identify each of the components of such a pair with one of the two ends of the channel. If channel c connects party p with party q and r is a run, then by the projection $pr_p(r(c))$ we mean the component of the pair associated with p, and by $pr_q(r(c))$, the component associated with q. Now we are ready to specify the local condition predicates L_p. If $c_1, \ldots c_n$ is the list of all channels incident with p, then L_p is the statement

$$pr_p(r(c_1)) + \ldots pr_p(r(c_n)) = 0 \mod 2.$$

This concludes the definition of the parity protocol $\mathcal{P}_A$.

Theorem 9. *$\mathcal{P}_A$ is a finite protocol.*

Proof. We need to prove the existence of a run that satisfies all local conditions. Indeed, consider the run r_0 such that $r_0(c) = \langle 0, 0\rangle$ for any channel c. □

Definition 9. *For any run r, if $r(c) = \langle b_1, b_2\rangle$, let $\oplus(r(c))$ denote $b_1 + b_2 \mod 2$.*

Theorem 10. *For any run r of the parity protocol $\mathcal{P}_A$,*

$$\sum_{c \in A} \oplus(r(c)) = 0 \mod 2.$$

Proof. Let P be the set of all parties in collaboration network N. If we let $Inc(p)$ denote the set of all channels incident with party p, then

$$\begin{aligned}\sum_{c\in A}\oplus(r(c)) &= \sum_{c\in Ch(N)}\oplus(r(c)) - \sum_{c\notin A}\oplus(r(c)) = \\ &= \sum_{p\in P}\sum_{c\in Inc(p)} pr_p(r(c)) - \sum_{c\notin A} 0 = \sum_{p\in P} 0 - 0 = 0 \mod 2.\end{aligned}$$

□

Definition 10. *Assume that π is a path in network N such that either:*

1. *$\pi = a, c_1, c_2, \ldots, c_n, b$ is a simple path, where $a, b \in A$ and $a \neq b$, or*
2. *$\pi = c_1, c_2, \ldots, c_n, c_1$ is a simple cyclic path.*

For any run r of the parity protocol $\mathcal{P}_A$ and path π in N, we introduce a function called $flip(r, \pi)$ that assigns a value from $V(c)$ to each channel c of N as follows. For any $x \in Ch(N)$, let $r(x) = \langle x_1, x_2\rangle$, and define:

$$flip(r,\pi)(x) = \begin{cases} \langle x_1, \neg x_2\rangle & \text{if } x = a, \\ \langle \neg x_1, \neg x\rangle & \text{if } x \in \{c_1, \ldots, c_n\}, \\ \langle \neg x_1, x_2\rangle & \text{if } x = b, \\ \langle x_1, x_2\rangle & \text{if } x \notin \pi. \end{cases}$$

Theorem 11. *$flip(r, \pi) \in \mathcal{R}(\mathcal{P}_A)$ for any $r \in \mathcal{P}_A$ and path π in N.*

Proof. The flip operation preserves the local conditions of protocol $\mathcal{P}_A$. □

Theorem 12. *If $|A| > 1$ and collaboration network N is connected, then for any $a \in A$ and any $v \in \{0, 1\}$, there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $\oplus(r(a)) = v$.*

Proof. By Theorem 9, there is a run r of protocol $\mathcal{P}_A$. Suppose that $\oplus(r(a)) \neq v$. Since $|A| > 1$ and collaboration network N is connected, there is a simple path π that connects channel a with channel $b \in A$ such that $b \neq a$. Consider run $r' = flip(r, \pi)$ and notice that $\oplus(r'(a)) = v$. □

Theorem 13. *If $|A| > 1$ and network N is connected, then $\mathcal{P}_A \not\models [A]$.*

Proof. Let $A = \{a_1, \ldots, a_k\}$. Pick any values $v_1, \ldots, v_k$ such that $v_1 + \cdots + v_k = 1 \mod 2$. By Theorem 12, there are runs $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P}_A)$ such that $\oplus(r_i(a_i)) = v_i$ for any $i \in \{1, \ldots, k\}$. If $\mathcal{P}_A \models [A]$, then there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $r(a_i) = r_i(a_i)$ for any $i \in \{1, \ldots, k\}$. Therefore, $\oplus(r(a_1)) + \cdots + \oplus(r(a_k)) = \oplus(r_1(a_1)) + \cdots + \oplus(r_k(a_k)) = v_1 + \cdots + v_k = 1 \mod 2$. This contradicts Theorem 10. □

Theorem 14. *Let A and B be subsets of $Ch(N)$ and let N' be the collaboration network N with all channels in B removed. If each connected component of N' contains at least one channel from A, then $\mathcal{P}_A \vDash [B]$.*

Proof. Let $B = \{b_1, \ldots, b_k\}$. Consider any runs $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P}_A)$. We will prove that there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $r(b_i) = r_i(b_i)$ for every $v \in b_i$. Indeed, protocol $\mathcal{P}_A$ has at least one run. Call it $\hat{r}$. We will modify run $\hat{r}$ to satisfy the condition $\hat{r}(b_i) = r_i(b_i)$ for any $i \leq k$. Our modification will consist of repeating the following procedure for each $i \leq k$ and each end p of channel b_i such that $pr_p(\hat{r}(b_i)) \neq pr_p(r_i(b_i))$:

1. Suppose $b_i \in A$. Let N'_p be the connected component of collaboration network N' that contains party p. By the assumption of the theorem, there must be a path π' in N'_p connecting party p with a channel in $(A \setminus B)$. Consider the path in N that starts with channel b_i and then follows path π' in N'_p.
   Let f denote the run $flip(\hat{r}, \pi)$. By Theorem 11, $f \in \mathcal{R}(\mathcal{P}_A)$. Note that $pr_p(f(b_i)) = 1 - pr_p(\hat{r}(b_j)) = pr_p(r_i(b_i))$, as desired. Additionally, run f matches $\hat{r}$ everywhere except path π, and π contains only a single end of one channel from set B. Specifically, it contains end p of channel b_i. Thus, it is clear that for each end q of each channel b_j other than b_i, $pr_q(f(b_j)) = pr_q(\hat{r}(b_j))$. Furthermore, for the end q of channel b_i where $q \neq p$, $pr_q(f(b_i) = pr_q(\hat{r}(b_i))$ as well. Let run f be the new $\hat{r}$.
2. If $b_i \notin A$, then, by definition of $\mathcal{P}_A$, for any run $r \in \mathcal{P}_A$ both components of pair $r(b_i)$ must be equal. At the same time, by our assumption, $pr_p(\hat{r}(b_i)) \neq pr_p(r_i(b_i))$. Thus $pr_q(\hat{r}(b_i)) \neq pr_q(r_i(b_i))$, where q is the end of channel b_i different from p. Note that parties p and q may belong either to the same connected component or to two different connected components of collaboration network N'. We will consider these two subcases separately.
   (a) Suppose p and q belong to the same connected component of N'. Thus, there must be a path π' in N' which connects parties p and q. Consider now a cyclic path in collaboration network N that starts at channel b_i, follows path π', and comes back to b_i. Call this cyclic path π.
   (b) Suppose p and q belong to different connected components of N'. Thus, by the assumption of the theorem, N' contains a path π_p that connects party p with an channel in $(A \setminus B)$. By the same assumption, N' must also contain a path π_q that connects party q with a channel in $(A \setminus B)$. Let path π be composed by attaching paths π_p and π_q to channel b_i at ends p and q, respectively.

   Again, let f denote the run $flip(\hat{r}, \pi)$, which is in $\mathcal{R}(\mathcal{P}_A)$ by Theorem 11. Note also that $f(b_j) = \hat{r}(b_j)$ for all j where $j \neq i$. When $j = i$, the two ends of $f(b_j)$ have values which are equal to each other, but opposite that on the two equal ends of $\hat{r}(b_j)$. Thus, $f(b_j) = r_i(b_i)$. Let f be the new $\hat{r}$.

Let r be $\hat{r}$ with all the modifications described above. These modifications guarantee that $r(b_i) = \hat{r}(b_i) = r_i(b_i)$ for any $i \leq k$. □

8.2. Recursive Construction

In this section we will generalize the parity protocol through a recursive construction. First, however, we will establish a technical result that we will need for this construction.

Theorem 15 (protocol extension). *For any cut (X, Y) of collaboration network N and any finite protocol $\mathcal{P}'$ on truncation N_X, there is a finite protocol $\mathcal{P}$ on N such that for any set $Q \subseteq Ch(N)$,*

$$\mathcal{P} \vDash [Q] \qquad \textit{iff} \qquad \mathcal{P}' \vDash [Q \cap E(N_X)]$$

Proof. To define protocol $\mathcal{P}$ we need to specify a set of values $V(c)$ for each channel $c \in Ch(N)$ and the set of local conditions for each party p in collaboration network N. If $c \in Ch(N_X)$, then let $V(c)$ be the same as in protocol $\mathcal{P}$. Otherwise, $V(c) = \{\epsilon\}$, where ϵ is an arbitrary element. The local conditions at the parties in X are the same as in protocol $\mathcal{P}'$, and the local conditions at the parties in Y are equal to the boolean constant $True$. This completes the definition of $\mathcal{P}$. Clearly, $\mathcal{P}$ has at least one run as long as $\mathcal{P}'$ has a run.
($\Rightarrow$) : Suppose that $Q \cap E(N_X) = \{q_1, \ldots, q_k\}$. Consider any $r'_1, \ldots, r'_k \in \mathcal{R}(\mathcal{P}')$. Define runs $r_1, \ldots, r_k$ as follows:

$$r_i(c) = \begin{cases} r'_i(c) & \text{if } c \in Ch(N_X), \\ \varepsilon & \text{if } c \notin Ch(N_X). \end{cases}$$

Note that runs r_i and r'_i, by definition, are equal on any channel incident with any party in collaboration network N_X. Thus, r_i satisfies the local conditions at any such party. Hence, $r_i \in \mathcal{R}(\mathcal{P})$ for any $i \in \{1, \ldots, k\}$. By Definition 3, there must be at least one run of protocol $\mathcal{P}$ (even if $k = 0$). Call this run r_0. By assumption $\mathcal{P} \vDash [Q]$, there is a run $r \in \mathcal{R}(\mathcal{P})$ such that

$$r(c) = \begin{cases} r_i(c) & \text{if } c = q_i, \\ r_0(c) & \text{if } c \in Q \setminus E(N_X). \end{cases}$$

Define r' to be a restriction of r on collaboration network N_X. Note that r' satisfies all local conditions of $\mathcal{P}'$. Thus, $r' \in \mathcal{R}(\mathcal{P}')$. At the same time, $r'(q_i) = r_i(q_i) = r'_i(q_i)$.
($\Leftarrow$) : Suppose that $Q = \{q_1, \ldots, q_k\}$. Consider any $r_1, \ldots, r_k \in \mathcal{R}(\mathcal{P})$, and let $r'_1, \ldots, r'_k$ be their respective restrictions to collaboration network N_X. Since, for any $i \in \{1, \ldots, k\}$, run r'_i satisfies the local conditions of $\mathcal{P}'$ at any node of N_X, we can conclude that $r'_1, \ldots, r'_k \in \mathcal{R}(\mathcal{P}')$. By the assumption that $\mathcal{P}' \vDash [Q \cap E(N_X)]$, there is a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q) = r'_i(q)$ for any $q \in Q \cap E(N_X)$. In addition, $r'(q) = \varepsilon = r'_i(q)$ for any $q \in Q \setminus E(N_X)$. Hence,

$r'(q_i) = r'_i(q_i)$ for any $i \in \{1, \ldots, k\}$. Define run r as follows:

$$r(c) = \begin{cases} r'(c) & \text{if } c \in Ch(N_X), \\ \varepsilon & \text{if } c \notin Ch(N_X). \end{cases}$$

Note that r satisfies the local conditions of $\mathcal{P}$ at all nodes. Thus, $r \in \mathcal{R}(\mathcal{P})$. In addition, $r(q_i) = r'(q_i) = r'_i(q_i)$ for all $i \in \{1, \ldots, k\}$. □

We will now prove another key theorem in our construction. The proof of this theorem recursively defines a generalization of the parity protocol.

Theorem 16. *For any sets $A, B_1, \ldots, B_n \subseteq Ch(N)$, if $N \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, then there is a finite protocol $\mathcal{P}$ over N such that $\mathcal{P} \vDash [B_i]$ for all $1 \le i \le n$ and $\mathcal{P} \nvDash [A]$.*

Proof. We use induction on the number of parties in collaboration network N.
Case 1. If $|A| \le 1$, then, by the *Small Set* axiom, $N \vdash [A]$. Hence,

$$N \vdash \bigwedge_{1 \le i \le n} [B_i] \to [A],$$

which is a contradiction.
Case 2. Suppose that the channels of collaboration network N can be partitioned into two non-trivial disconnected sets X and Y. That is, no channel in X is incident with a channel in Y. Thus, the empty set is a gateway between $A \cap X$ and $A \cap Y$. By the *Gateway* axiom,

$$N \vdash [A \cap X] \to ([A \cap Y] \to [A]).$$

Hence, taking into account the assumption $N \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, either

$$N \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A \cap X]$$

or

$$N \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A \cap Y].$$

Without loss of generality, we will assume the former. By Theorem 4,

$$N \nvdash \bigwedge_{1 \le i \le n} [B_i \cap X] \to [A \cap X].$$

By the *Small Set* axiom,

$$N \nvdash [\varnothing] \to (\bigwedge_{1 \le i \le n} [B_i \cap X] \to [A \cap X]).$$

Consider the sets P_X and P_Y of all parties in components X and Y respectfully. Note that (P_X, P_Y) is a cut of N that has no crossing channels. Let N_X be the

result of the truncation of N along this cut. By the *Truncation* rule,

$$N_X \nvdash \bigwedge_{1\leq i\leq n} [B_i \cap X] \to [A \cap X].$$

By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on N_X such that $\mathcal{P}' \not\models [A\cap X]$ and $\mathcal{P}' \models [B_i \cap X]$, for any $i \leq n$. Therefore, by Theorem 15, there is a protocol $\mathcal{P}$ on N such that $\mathcal{P} \not\models [A]$ and $\mathcal{P} \models [B_i]$ for any $i \leq n$.

Case 3. Suppose there is $i_0 \in \{1, \ldots, n\}$ such that if all channels in B_{i_0} are removed from collaboration network N, then at least one connected component of the resulting network N' does not contain an element of A. We will denote this connected component by Q. Recall that $E(Q)$ denotes the set of all channels in N that begin and end in Q. Let $Out(Q)$ be the set of channels in N that connect a party from Q with a party not in Q. Any path connecting a channel in $E(Q)$ with a channel not in $E(Q)$ will have to contain a channel from $Out(Q)$. In other words, $Out(Q)$ is a gateway between $E(Q)$ and the complement of $E(Q)$ in N. Hence, $Out(Q)$ is also a gateway between $A \cap E(Q)$ and $A \setminus E(Q)$. Therefore, by the *Gateway* axiom, taking into account that $(A\cap E(Q))\cap Out(Q) \subseteq E(Q)\cap Out(Q) = \varnothing$,

$$N \vdash [A \cap E(Q), Out(Q)] \to ([A \setminus E(Q)] \to [A]). \tag{4}$$

Recall now that by the assumption of this case, component Q of collaboration network N' does not contain any elements of A. Hence, $A \cap E(Q) \subseteq B_{i_0}$. At the same time, $Out(Q) \subseteq B_{i_0}$ by the definition of Q. Thus, from statement (4) and Theorem 4,

$$N \vdash [B_{i_0}] \to ([A \setminus E(Q))] \to [A]). \tag{5}$$

By the assumption of the theorem,

$$N \nvdash \bigwedge_{1\leq i\leq n} [B_i] \to [A]. \tag{6}$$

From statements (5) and (6),

$$N \nvdash \bigwedge_{1\leq i\leq n} [B_i] \to [A \setminus E(Q))].$$

By the laws of propositional logic,

$$N \nvdash [B_{i_0}] \to (\bigwedge_{1\leq i\leq n} [B_i] \to [A \setminus E(Q)]).$$

Note that if $\overline{Q}$ is the complement of set Q, then $(\overline{Q}, Q)$ is a cut of collaboration network N and $Out(Q)$ is the set of all crossing channels of this cut. Since Q is

a separate component in N', we have $Out(Q) \subseteq B_{i_0}$. Thus, by Theorem 4,

$$N \nvdash [Out(Q)] \to (\bigwedge_{1 \le i \le n} [B_i] \to [A \setminus E(Q)]).$$

Again by Theorem 4,

$$N \nvdash [Out(Q)] \to (\bigwedge_{1 \le i \le n} [B_i \setminus E(Q)] \to [A \setminus E(Q)]).$$

Let $N_{\overline{Q}}$ be the result of the truncation of network N along the cut $(\overline{Q}, Q)$. By the *Truncation* rule,

$$N_{\overline{Q}} \nvdash \bigwedge_{1 \le i \le n} [B_i \setminus E(Q)] \to [A \setminus E(Q)].$$

By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on $N_{\overline{Q}}$ such that $\mathcal{P}' \nvDash [A \setminus E(Q)]$ and $\mathcal{P}' \vDash [B_i \setminus E(Q)]$ for any $i \le n$. Therefore, by Theorem 15, there is a protocol $\mathcal{P}$ on N such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for any $i \le n$.
Case 4. Assume now that (i) $|A| > 1$, (ii) collaboration network N is connected, and (iii) collaboration network N' is the network obtained from N by the removal of all channels in B_i and for any $i \le n$, each connected component of N' contains at least one element of A. Consider the parity protocol $\mathcal{P}_A$ over N. By Theorem 13, $\mathcal{P}_A \nvDash [A]$. By Theorem 14, $\mathcal{P}_A \vDash [B_i]$ for any $i \le n$. □

8.3. Protocol Composition

In the previous section we defined protocol $\mathcal{P}_A$. In this section, we begin by defining the composition of several protocols. Later, we use this operation to combine protocols $\mathcal{P}_A$ for different values of A in two a single protocol in order to finish the proof of the completeness theorem.

Definition 11. *For any protocols $\mathcal{P}^1 = (V^1, L^1), \ldots, \mathcal{P}^n = (V^n, L^n)$ over a collaboration network N, we define the Cartesian composition $\mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ to be a pair (V, L) such that*

1. $V(c) = V^1(c) \times \cdots \times V^n(c)$,
2. $L_p(\langle c_1^1, \ldots, c_1^n \rangle, \ldots, \langle c_k^1, \ldots, c_k^n \rangle) = \bigwedge_{1 \le i \le n} L_p^i(c_1^i, \ldots, c_k^i)$,

For each composition $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$, we let $\{r(c)\}_i$ denote the ith component of the value of secret c over run r.

Theorem 17. *For any $n > 0$ and any finite protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$ over a collaboration network N, $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ is a finite protocol over N.*

Proof. We need to show that $\mathcal{P}$ has at least one run. Indeed, let $r^1, \ldots, r^n$ be runs of $\mathcal{P}^1, \ldots, \mathcal{P}^n$. Define $r(c)$ to be $\langle r^1(c), \ldots, r^n(c) \rangle$. It is easy to see

that r satisfies the local conditions L_p for any party p of network N. Thus, $r \in \mathcal{R}(\mathcal{P})$. □

Theorem 18. *For any $n > 0$, for any protocol $\mathcal{P} = \mathcal{P}^1 \times \mathcal{P}^2 \times \cdots \times \mathcal{P}^n$ over a collaboration network N, and for any set of channels Q,*

$$\mathcal{P} \vDash [Q] \quad \textit{if and only if} \quad \forall i\ (\mathcal{P}^i \vDash [Q]).$$

Proof. Let $Q = \{q_1, \ldots, q_\ell\}$.
($\Rightarrow$) : Assume $\mathcal{P} \vDash [Q]$ and pick any $i_0 \in \{1, \ldots, n\}$. We will show that $\mathcal{P}^{i_0} \vDash [Q]$. Pick any runs $r'_1, \ldots, r'_\ell \in \mathcal{R}(\mathcal{P}^{i_0})$. For each $i \in \{1, \ldots, i_0 - 1, i_0 + 1, \ldots, n\}$, select an arbitrary run $r^i \in \mathcal{R}(\mathcal{P}^i)$. We then define a series of composed runs r_j for $j \in \{1, \ldots, \ell\}$ by

$$r_j(c) = \langle r^1(c), \ldots, r^{i_0-1}(c), r'_j(c), r^{i_0+1}(c), \ldots, r^n(c) \rangle,$$

for each secret $c \in Ch(N)$. Since the component parts of each r_j belong in their respective sets $\mathcal{R}(\mathcal{P}^i)$, the composed runs are themselves members of $\mathcal{R}(\mathcal{P})$. By our assumption, $\mathcal{P} \vDash [Q]$, thus there is $r \in \mathcal{R}(\mathcal{P})$ such that $r(q_i) = r_i(q_i)$ for any $i_0 \in \{1, \ldots, \ell\}$. Finally, we consider the run r^*, where $r^*(c) = \{r(c)\}_{i_0}$ for each $c \in Ch(N)$. That is, we let the value of r^* on c be the i_o^{th} component of $r(c)$. By the definition of composition, $r^* \in \mathcal{R}(\mathcal{P}^{i_0})$, and it matches the original $r'_1, \ldots, r'_\ell \in \mathcal{R}(\mathcal{P}^{i_0})$ on channels $q_1, \ldots, q_\ell$, respectively. Hence, we have shown that $\mathcal{P}^{i_0} \vDash [Q]$.
($\Leftarrow$) : Assume $\forall i\ (\mathcal{P}^i \vDash [Q])$. We will show that $\mathcal{P} \vDash [Q]$. Pick any runs $r_1, \ldots, r_\ell \in \mathcal{R}(\mathcal{P})$. For each $i \in \{1, \ldots, n\}$, each $j \in \{1, \ldots, \ell\}$, and each channel c, let $r^i_j(c) = \{r_j(c)\}_i$. That is, for each c, define a run r^i_j whose value on channel c equals the ith component of $r_j(c)$. Note that by the definition of composition, for each i and each j, r^i_j is a run in $\mathcal{R}(\mathcal{P}^i)$. Next, for each $i \in \{1, \ldots, n\}$, we use the fact that $\mathcal{P}^i \vDash [Q]$ to construct a run $r^i \in \mathcal{R}(\mathcal{P}^i)$ such that $r^i(q_j) = r^i_j(q_j)$. Finally, we compose these n runs $r^1, \ldots, r^n$ to get run $r \in \mathcal{R}(\mathcal{P})$. We note that the value of each channel q_j on r matches the the value of q_j in run $r_j \in \mathcal{R}(\mathcal{P})$, demonstrating that $\mathcal{P} \vDash [Q]$. □

We are now ready to prove the completeness theorem, which appeared earlier as Theorem 8:

Theorem *For any collaboration network N, if $\mathcal{P} \vDash \phi$ for all finite protocols $\mathcal{P}$ over N, then $N \vdash \phi$.*

Proof. We give a proof by contradiction. Let X be a maximal consistent set of formulas from $\Phi(N)$ that contains $\neg\phi$. Let $\{A_1, \ldots, A_n\} = \{A \subseteq Ch(N) \mid N \nvdash [A]\}$ and $\{B_1, \ldots, B_k\} = \{B \subseteq Ch(N) \mid N \vdash [B]\}$. Thus, $N \nvdash \bigwedge_{1 \le j \le k} [B_j] \to [A_i]$, for any $i \in \{1, \ldots, n\}$. We will construct a protocol $\mathcal{P}$ such that $\mathcal{P} \nvDash [A_i]$ for any $i \in \{1, \ldots, n\}$ and $\mathcal{P} \vDash [B_j]$ for any $j \in \{1, \ldots, k\}$.

First consider the case where $n = 0$. Pick any symbol ϵ and define $\mathcal{P}$ to be $\langle V, L\rangle$ such that $V(c) = \{\epsilon\}$ for any $c \in Ch(N)$ and local condition L_p to be the constant $True$ at any party p. By Definition 4, $\mathcal{P} \vDash [C]$ for any $C \subseteq Ch(N)$.

We will assume now that $n > 0$. By Theorem 16, there are finite protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$ such that $\mathcal{P}^i \nvDash [A_i]$ and $\mathcal{P}^i \vDash [B_j]$ for all $j \in \{1, \ldots, k\}$. Consider the composition $\mathcal{P}$ of protocols $\mathcal{P}^1, \ldots, \mathcal{P}^n$. By Theorem 18, $\mathcal{P} \nvDash [A_i]$ for any $i \in \{1, \ldots, n\}$ and $\mathcal{P} \vDash [B_j]$ for any $j \in \{1, \ldots, j\}$.

By induction on the structural complexity of any formula $\psi \in \Phi(N)$, one can show now that $N \vdash \psi$ if and only if $\psi \in X$. Thus, $\mathcal{P} \vDash \neg\phi$. Therefore, $\mathcal{P} \nvDash \phi$, which is a contradiction. □

Corollary 1. *The set* $\{(N, \phi) \mid N \vdash \phi\}$ *is decidable.*

Proof. The complement of this set is recursively enumerable due to the completeness of the system with respect to finite protocols. □

9. Conclusions

We have presented a formal logical system for reasoning about an independence relation and proved the completeness of this system with respect to a semantics of secrets. As an extension, one could study a natural generalization of this result to secrets shared by more than two parties. In that setting, a collaboration network is a hypergraph whose edges (channels) may connect an arbitrary number of nodes (parties).

References

[1] A. Sabelfeld, A. C. Myers, Language-based information-flow security, IEEE Journal on Selected Areas in Communications 21 (1) (2003) 5–19.

[2] T. Amtoft, A. Banerjee, A logic for information flow analysis with an application to forward slicing of simple imperative programs, Sci. Comput. Program. 64 (1) (2007) 3–28.

[3] J. Y. Halpern, K. R. O'Neill, Secrecy in multiagent systems, in: Proceedings of the Fifteenth IEEE Computer Security Foundations Workshop, 2002, pp. 32–46.

[4] J. Y. Halpern, K. R. O'Neill, Secrecy in multiagent systems, ACM Trans. Inf. Syst. Secur. 12 (1) (2008) 1–47.

[5] D. MacKenzie, Mechanizing Proof: Computing, Risk, and Trust, MIT Press, 2004.

[6] J. A. Goguen, J. Meseguer, Security policies and security models, in: Proceedings of IEEE Symposium on Security and Privacy, 1982, pp. 11–20.

[7] D. Sutherland, A model of information, in: Proceedings of Ninth National Computer Security Conference, 1986, pp. 175–183.

[8] E. Cohen, Information transmission in computational systems, in: Proceedings of Sixth ACM Symposium on Operating Systems Principles, Association for Computing Machinery, 1977, pp. 113–139.

[9] S. Miner More, P. Naumov, On interdependence of secrets in collaboration networks, in: Proceedings of 12th Conference on Theoretical Aspects of Rationality and Knowledge (Stanford University, 2009), 2009, pp. 208–217.