Hypergraphs of Multiparty Secrets

Sara Miner More and Pavel Naumov

Department of Mathematics and Computer Science
McDaniel College, Westminster, Maryland 21157, USA
{smore,pnaumov}@mcdaniel.edu

Abstract. The paper considers interdependencies between secrets in a multiparty system. Each secret is assumed to be known only to a certain fixed set of parties. These sets can be viewed as edges of a hypergraph whose vertices are the parties of the system. The main result is a complete and decidable logical system that describes interdependencies that may exist on a fixed hypergraph. The properties of interdependencies are defined through a multi-argument relation called *independence*, which is a generalization of a binary relation also known as nondeducibility.

1 Introduction

In this paper, we study properties of interdependencies between pieces of information. We call these pieces *secrets* to emphasize the fact that they might be known to some parties and unknown to others. Below, we first describe two relations for expressing interdependencies between secrets. Next, we discuss these relations in the context of collaboration networks which specify the available communication channels for the parties establishing the secrets.

Relations on Secrets. One of the simplest relations between two secrets is *functional dependence*, which we denote by $a \rhd b$. It means that the value of secret a reveals the value of secret b. This relation is reflexive and transitive. A more general and less trivial form of functional dependence is functional dependence between sets of secrets. If A and B are two sets of secrets, then $A \rhd B$ means that, together, the values of all secrets in A reveal the values of all secrets in B. Armstrong [1] presented a sound and complete set of axioms for this relation.

These axioms are known in database literature as Armstrong's axioms [2, p. 81]. Beeri, Fagin, and Howard [3] suggested a variation of Armstrong's axioms that describe properties of multi-valued dependency.

Not all dependencies between two secrets are functional. For example, if secret a is a pair $\langle x, y\rangle$ and secret b is a pair $\langle y, z\rangle$, then there is an interdependence between these secrets in the sense that not every value of secret a is compatible with every value of secret b. However, neither $a \rhd b$ nor $b \rhd a$ is necessarily true. If there is no interdependence at all between two secrets, then we will say that the two secrets are *independent*. In other words, secrets a and b are independent if any possible value of secret a is compatible with any possible value of secret b. We denote this relation between two secrets by $[a, b]$. This relation was introduced by Sutherland [4] and is also known as *nondeducibility* in the study of information flow. Halpern and O'Neill [5] proposed a closely related notion called f-secrecy.

Like functional dependence, independence also can be generalized to relate two sets of secrets. If A and B are two such sets, then $[A, B]$ means that any consistent combination of values of the secrets in A is compatible with any consistent combination of values of the secrets in B. Note that "consistent combination" is an important condition here, since some interdependence may exist between secrets in set A even while the entire set of secrets A is independent from the secrets in set B. The following is an example of a non-trivial property expressible in this language:

$$[A \cup B\,,\, C] \rightarrow ([A\,,\, B] \rightarrow [A\,,\, B \cup C]).$$

A sound and complete axiomatization of all such properties was given by More and Naumov [6]. Essentially the same axioms were shown by Geiger, Paz, and Pearl [7] to provide a complete axiomatization of the independence relation between sets of random variables in probability theory. A complete logical system that combines independence and functional dependence predicates for *single* secrets was described by Kelvey, More, Naumov, and Sapp [8].

Secrets in Networks. So far, we have assumed that the values of secrets are determined a priori. In the physical world, however, secret values are often generated, or at least disseminated, via interaction between several parties. Quite often such interactions happen over a network with fixed topology. For example, in social networks, interaction between nodes happens along connections formed by friendship, kinship, financial relationship, etc. In distributed computer systems, interaction happens over computer networks. Exchange of genetic information happens along the edges of the genealogical tree. Corporate secrets normally flow over an organization chart. In cryptographic protocols, it is often assumed that values are transmitted over well-defined channels. On social networking websites, information is shared between "friends". Messages between objects on an UML interaction diagram are sent along connections defined by associations between the classes of the objects.

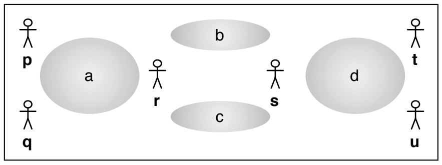

Fig. 1. Collaboration network H_0.

In this paper, we will use the notion of *collaboration network* to refer to the topological structure that specifies which secrets are known to which parties. An example of such network is given on Figure 1. In this network, parties p, q and r share secret a; parties r and s share secrets b and c; and parties s, t and u share secret d. If different secrets are established completely independently, 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 secrets may enforce some desired interdependence between the values of these secrets. These "local" interdependencies between secrets known to a single party may result in a "global" interdependence 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.

We will say that the local interdependencies define a *protocol*. For the collaboration network H_0 depicted in Figure 1, for example, we can imagine the following protocol. Parties p, q and r together pick a random value a from set $\{0, 1\}$. Next, party r chooses values b and c from $\{0, 1\}$ in such a way that $a = b + c \mod 2$ and sends both of these values to party s. Party s computes $d = b + c \mod 2$ and shares value d with parties t and u. In this protocol, it is clear that the values of a and d will always match. Hence, for this specific protocol, we can say that $a \rhd d$ and $d \rhd a$, but at the same time, $[a, b]$ and $[a, c]$.

The functional dependence and independence examples above are for a single protocol, subject to a particular set of local interdependencies between secrets. If the network remains fixed, but the protocol is changed, then secrets which were previously interdependent may no longer be so, and vice versa. For example, for network H_0 above, the claim $a \rhd d$ will no longer be true if, say, party s switches from enforcing the local condition $d = b + c \mod 2$ to enforcing the local condition $d = b$. In this paper, we study properties of relations between secrets that follow from the topological structure of the collaboration network, no matter which specific protocol is used. Examples of such properties for network H_0 are $a \rhd d \rightarrow b, c \rhd d$ and $[\{a\}, \{b, c\}] \rightarrow [a, d]$.

A special case of the collaboration network is an undirected graph collaboration network in which any secret is shared between at most two parties. In an earlier work [9], we considered this special case and gave a complete axiomatic system for the independence relation between single secrets in that setting. In fact, we axiomatized a slightly more general relation $[a_1, a_2, \ldots, a_n]$ between multiple *single* secrets, which means that any possible values of secrets $a_1, \ldots, a_n$ can occur together.

In a more recent work, currently under review, we developed a complete logical system that describes the properties of the functional dependence relation $A \rhd B$ between sets of secrets over graph collaboration networks. This system includes Armstrong's axioms and a new Gateway axiom that captures properties of functional dependence specific to the topology of the collaboration network.

In the current paper, we focus on independence and generalize our results from collaboration networks defined by standard graphs to those defined by hypergraphs. That is, we examine networks where, as in Figure 1, a secret can be shared between more than two parties. In this setting, we give a complete and decidable system of axioms for the relation $[a_1, a_2, \ldots, a_n]$. In terms of the proof of completeness, the most significant difference between the earlier work [9] and this one is in the construction of the parity protocol in Section 7.1.

2 Hypergraphs

A collaboration network where a single secret can be shared between multiple parties can be described mathematically as a hypergraph in which vertices are parties and (hyper)edges are secrets. In this section, we will introduce the hypergraph terminology that is used later in the paper.

Definition 1. *A hypergraph is pair $H = \langle V, E \rangle$, where*

1. *V is a finite set, whose elements are called "vertices".*
2. *E is a finite multiset of non-empty subsets of V. Elements of E are called "edges". Elements of an edge are called the "ends" of the edge.*

Note that we use "mulitisets" in the above definition to allow for multiple edges between the same set of ends. Also note that, as is common in hypergraph literature [10, p. 1], we exclude empty edges from consideration.

Definition 2. *For any set of vertices V' of a hypergraph H, by $Out(V')$ we mean the set of edges in H that contain ends from both set V' and the complement of V'. By $In(V')$ we mean the set of edges in H that contain only ends from V'.*

From the collaboration network perspective, V' is a group of parties, $Out(V')$ is the public interface of this group (secrets that the group members share with non-members) and $In(V')$ is the set of secrets only known within group V'. For example, for the collaboration network defined by hypergraph H_0 on Figure 1, if $V' = \{r, s\}$, then $Out(V') = \{a, d\}$ and $In(V') = \{b, c\}$.

A *path* in a hypergraph is an alternating sequence of edges and vertices in which adjacent elements are incident. It will be convenient to assume that paths start and end with edges rather than with vertices. Paths will be assumed to be simple, in the sense that no edge or vertex is repeated in the path, with the exception that the last edge in the path may be the same as the first. In this case, the path is called cyclic. For example, a, r, b, s, c is a path in H_0 of Figure 1.

Definition 3. *A gateway between sets of edges A and B is a set of edges G such that every path from A to B contains at least one edge from G.*

For instance, set $\{b, c\}$ is a gateway between single-element sets $\{a\}$ and $\{d\}$ on the hypergraph H_0 from Figure 1. Note also that in the definition above, 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 hypergraph that are not connected one to another.

Definition 4. *If X is an arbitrary set of vertices of a hypergraph $H = \langle V, E \rangle$, then the truncation of set X from H is a hypergraph $H' = \langle V \setminus X, E' \rangle$, where*

$$E' = \{e \setminus X \mid e \in E \text{ and } e \setminus X \neq \varnothing\}.$$

Truncated hypergraph H' is also commonly [10, p. 3] referred to as the subhypergraph of H induced by the set of vertices $V \setminus X$.

3 Protocol: A Formal Definition

Definition 5. *A semi-protocol over a hypergraph $H = \langle V, E\rangle$ is a pair $\mathcal{P} = \langle Val, Loc\rangle$ such that*

1. *$Val(e)$ is an arbitrary set of "values" for each edge $e \in E$,*
2. *$Loc = \{Loc_v\}_{v\in V}$ is a family of relations, indexed by vertices (parties) of the hypergraph H, which we call "local conditions". If $e_1, \dots e_k$ is the list of all edges incident with vertex v, then $Loc_v \subseteq Val(e_1) \times \cdots \times Val(e_k)$.*

Definition 6. *A run of a semi-protocol $\langle Val, Loc\rangle$ is a function r such that*

1. *$r(e) \in Val(e)$ for any edge $e \in E$,*
2. *If $e_1, \dots e_k$ is the list of all edges incident with vertex $v \in V$, then the statement $Loc_v(r(e_1), \dots, r(e_k))$ is true.*

Definition 7. *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 $\mathcal{R}(\mathcal{P})$.

Definition 8. *A protocol $\mathcal{P} = \langle Val, Loc\rangle$ is called finite if the set $Val(e)$ is finite for every edge e of the hypergraph.*

The following definition of independence is identical to the one given earlier [9] for standard graphs.

Definition 9. *A set of edges $Q = \{q_1, \dots, q_k\}$ is 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 $i \in \{1, \dots, k\}$.*

4 Language of Secrets

By $\Phi(H)$, we denote the set of all collaboration network properties specified by hypergraph H that are expressible through the independence predicate. More formally, $\Phi(H)$ is a minimal set of formulas defined recursively as follows: (i) for any finite subset A of the set of edges of hypergraph H, formula $[A]$ is in $\Phi(H)$, (ii) the false constant $\bot$ is in set $\Phi(H)$, and (iii) for any formulas ϕ and $\psi \in \Phi(H)$, the implication $\phi \to \psi$ is in $\Phi(H)$. As usual, we assume that conjunction, disjunction, and negation are defined through $\to$ and $\bot$.

Next, we define a relation $\vDash$ between a protocol and a formula from $\Phi(H)$. Informally, $\mathcal{P} \vDash \phi$ means that formula ϕ is true under protocol $\mathcal{P}$.

Definition 10. *For any protocol $\mathcal{P}$ over a hypergraph H, and any formula $\phi \in \Phi(H)$, we define the relation $\mathcal{P} \vDash \phi$ recursively as follows:*

1. *$\mathcal{P} \nvDash \bot$,*
2. *$\mathcal{P} \vDash [A]$ if the set of edges A 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 paper, we study the formulas $\phi \in \Phi(H)$ that are true under *any* protocol $\mathcal{P}$ over a fixed hypergraph H. Below we describe a formal logical system for such formulas. This system, like earlier systems defined by Armstrong [1], More and Naumov [11, 9] and by Kelvey, More, Naumov, and Sapp [8], belongs to the set of deductive systems that capture properties of secrets. In general, we refer to such systems as *logics of secrets*. Since this paper is focused on only one such system, here we call it *the logic of secrets* of hypergraph H.

5 Logic of Secrets

In this section we will define a formal deductive system for the logic of secrets and give examples of proofs in this system. The soundness, completeness, and decidability of this system will be shown in the next two sections.

5.1 Formal System: Axioms and Rules

For any hypergraph $H = \langle V, E\rangle$, we will write $H \vdash \phi$ to state that formula $\phi \in \Phi(H)$ is provable in the logic of secrets of hypergraph H. 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. $H \vdash [A]$, where $A \subseteq E$ and $|A| < 2$.

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

Truncation Rule. If $H' \vdash \phi$, then $H \vdash [Out(X)] \rightarrow \phi$, where H' is obtained from H by the truncation of set X.

The soundness of this system will be demonstrated in Section 6.

Theorem 1 (monotonicity). $H \vdash [A] \rightarrow [B]$*, for any hypergraph* H *and any subset* B *of a set of edges* A *of hypergraph* H*.*

Proof. Consider sets B and $\varnothing$. Since there are no paths connecting these sets, any set of edges 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, $H \vdash [B, A \setminus B] \rightarrow ([\varnothing] \rightarrow [B])$. By the Small Set axiom, $H \vdash [B, A \setminus B] \rightarrow [B]$. By assumption $B \subseteq A$, we get $H \vdash [A] \rightarrow [B]$. □

5.2 Proof Examples

Our first example refers to hypergraph H_1 in Figure 2. It shows parties p and q that have secrets a and c, respectively, that they do not share with each other, and secret b that they both know.

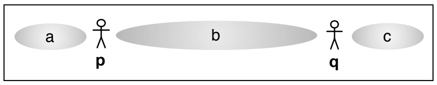

Fig. 2. Hypergraph H_1.

Theorem 2. $H_1 \vdash [a, b] \to [a, c]$.

Proof. Set $\{b\}$ is a gateway between sets $\{a\}$ and $\{c\}$. Thus, by the Gateway axiom, $H_1 \vdash [a, b] \to ([c] \to [a, c])$. At the same time, $H_1 \vdash [c]$, by the Small Set axiom. Therefore, $H_1 \vdash [a, b] \to [a, c]$. □

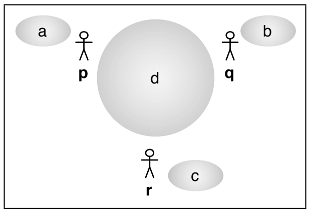

Fig. 3. Hypergraph H_2.

Our second example deals with the collaboration network defined by hypergraph H_2 on Figure 3. Here, parties p, q, and r have individual secrets a, b, c, and together share secret d.

Theorem 3. $H_2 \vdash [a, d] \to ([b, d] \to [a, b, c])$.

Proof. Note that set $\{d\}$ is a gateway between sets $\{a\}$ and $\{b, d\}$. Thus, by the Gateway axiom,

$$H_2 \vdash [a, d] \to ([b, d] \to [a, b, d]). \tag{1}$$

Next, observe that set $\{d\}$ is a gateway between sets $\{a, b\}$ and $\{c\}$. Thus, by the Gateway axiom, $H_2 \vdash [a, b, d] \to ([c] \to [a, b, c])$. By the Small Set axiom, $H_2 \vdash [c]$. Hence,

$$H_2 \vdash [a, b, d] \to [a, b, c]. \tag{2}$$

From statements (1) and (2), it follows that $H_2 \vdash [a, d] \to ([b, d] \to [a, b, c])$. □

Our third and final example refers to hypergraph H_3 depicted in Figure 4. In the proof we will also refer to hypergraph H_3', shown in the same figure, which is the result of the truncation of set $\{q, r, u, v\}$ from hypergraph H_3.

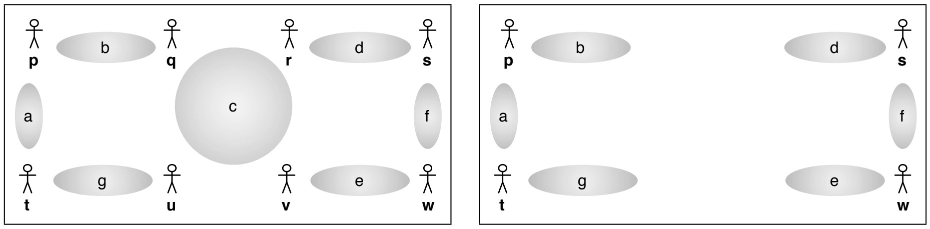

Fig. 4. Hypergraphs H_3 (left) and H_3' (right).

Theorem 4. $H_3 \vdash [b, d, g, e] \to [a, f]$.

Proof. Note that in the truncated hypergraph H_3', the empty set is a gateway between the single element sets $\{a\}$ and $\{f\}$. Thus, by the Gateway axiom, $H_3' \vdash [a] \to ([f] \to [a, f])$. By the Small Set axiom, $H_3' \vdash [a]$ and $H_3' \vdash [f]$. Hence, $H_3' \vdash [a, f]$. By the Truncation rule, $H_3 \vdash [Out(q, r, u, v)] \to [a, f]$. Since $Out(q, r, u, v) = \{b, d, g, e\}$, we get $H_3 \vdash [b, d, g, e] \to [a, f]$. □

6 Soundness

The proof of soundness, particularly for the Gateway axiom and 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 hypergraph $H = \langle V, E\rangle$ and any set of edges A that has at most one element, if $\mathcal{P}$ is an arbitrary protocol over H, 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 (see Definition 7). 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 hypergraph $H = \langle V, E\rangle$, and any gateway G between sets of edges A and B, 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 $A = \{a_1, \dots, a_n\}$ and $B = \{b_1, \dots, b_k\}$. Consider any $r_1, \dots, r_{n+k}$. It will be sufficient to show that there is $r \in \mathcal{R}(\mathcal{P})$ such that $r(a_i) = r_i(a_i)$ for any $i \le n$ and $r(b_i) = r_{n+i}(b_i)$ for any $i \le k$. By the assumption $\mathcal{P} \vDash [B]$, there is $r_b \in \mathcal{R}(\mathcal{P})$ such that

$$r_b(b_i) = r_{n+i}(b_i) \qquad \text{for any } i \le k. \tag{3}$$

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

$$r_a(c) = \begin{cases} r_i(c) \text{ if } c = a_i \text{ for } i \le n, \\ r_b(c) \text{ if } c \in G. \end{cases} \tag{4}$$

Next, consider hypergraph $H' = \langle V, E \setminus G\rangle$. By the definition of a gateway, no single connected component of hypergraph H' can contain edges from set A and set $B \setminus G$ at the same time. Let us divide all connected components of H' into two subhypergraphs H'_a and H'_b such that H'_a contains no edges from $B \setminus G$ and H'_b contains no edges from A. Components that do not contain edges from either A or $B \setminus G$ can be arbitrarily assigned to either H'_a or H'_b.

By definition (4), runs r_a and r_b agree on each edge of the 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,

$$r(c) = \begin{cases} r_a(c) & \text{if } c \in H_a, \\ r_a(c) = r_b(c) & \text{if } c \in G, \\ r_b(c) & \text{if } c \in H_b. \end{cases} \tag{5}$$

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 Loc_v at every vertex v. Without loss of generality, assume that $v \in H'_a$. Hence, on all edges incident with v, run r agrees with run r_a. Thus, run r satisfies Loc_v 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 (4) and (5), $r(a_i) = r_a(a_i) = r_i(a_i)$. Finally, we will need to show that $r(b_i) = r_{n+i}(b_i)$ for any $i \leq k$. This, however, trivially follows from equation (3) and equation (5). □

Theorem 7 (Truncation). *Assume that hypergraph H' is obtained from H by the truncation of set X and that $\phi \in \Phi(H')$. If $\mathcal{P}' \vDash \phi$ for any protocol $\mathcal{P}'$ over hypergraph H', then $\mathcal{P} \vDash [Out(X)] \to \phi$ for any protocol $\mathcal{P}$ over hypergraph H.*

Proof. Suppose that there is a protocol $\mathcal{P}$ over H such that $\mathcal{P} \vDash [Out(X)]$, but $\mathcal{P} \nvDash \phi$. We will construct a protocol $\mathcal{P}'$ over H' such that $\mathcal{P}' \nvDash \phi$.

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

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

The local condition Loc'_v at any vertex v of hypergraph H' 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 edges in H' 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 edge e in hypergraph H'.*

Proof. Consider any run $r' \in \mathcal{R}(\mathcal{P}')$. By definition of Val', for any $e \in Out(X)$ there is a run $r_e \in \mathcal{R}(\mathcal{P})$ such that $r'(e) = r_e(e)$. Since $\mathcal{P} \vDash [Out(X)]$, there is a run $r_X \in \mathcal{R}(\mathcal{P})$ such that $r_X(e) = r_e(e) = r'(e)$ for any $e \in Out(X)$.

We will now construct a combined run $r \in \mathcal{R}(\mathcal{P})$ by "sewing" together r_X and r' with the "stitches" placed in set $Out(X)$. Formally,

$$r(e) = \begin{cases} r_X(e) & \text{if } e \in In(X), \\ r_X(e) = r'(e) & \text{if } e \in Out(X), \\ r'(e) & \text{otherwise.} \end{cases}$$

We just need to show that r satisfies Loc_v at every vertex v of hypergraph H. Indeed, if $v \in X$, then run r is equal to r_X on all edges incident with v. Thus, it satisfies the local condition because run r_X does. Alternatively, if $v \notin X$, then run r is equal to run r' on all edges incident with v. Since r' satisfies local condition Loc'_v and, by definition, $Loc'_v \equiv Loc_v$, we can conclude that r again satisfies condition Loc_v.

Lemma 2. *$\mathcal{P} \vDash [Q]$ if and only if $\mathcal{P}' \vDash [Q]$, for any set of edges Q in H'.*

Proof. Assume first that $\mathcal{P} \vDash [Q]$ and consider any runs $r'_1, \dots, r'_n \in \mathcal{R}(\mathcal{P}')$. We will construct a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q_i) = r'_i(q_i)$ for every $i \in \{1, \dots, n\}$. Indeed, by Lemma 1, there are runs $r_1, \dots, r_n \in \mathcal{R}(\mathcal{P})$ that match runs $r'_1, \dots, r'_n$

on all edges in H'. 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 a restriction of run r to the edges in H'. 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 edges in H'. 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 H'. Therefore, $r(q_i) = r'(q_i) = r_i(q_i)$ for all $i \in \{1, \ldots, n\}$.

Lemma 3. *For any formula $\psi \in \Phi(H')$, $\mathcal{P} \vDash \psi$ 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. □

7 Completeness

Our main result is the following completeness theorem for the logic of secrets:

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

We prove this theorem by contrapositive. At the core of this 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 comes from the family of "parity" protocols $\{\mathcal{P}_A\}_A$ defined below.

7.1 Parity Protocol $\mathcal{P}_A$

Let $H = \langle V, E \rangle$ be a hypergraph and A be a subset of E. We define the "parity protocol" $\mathcal{P}_A$ over H as follows. The set of values of any edge e in hypergraph H is $\{0,1\}^e$, or the set of boolean functions on e. Thus, a run r of the protocol will be a function that maps an edge into a function from the ends of this edge into boolean values: $r(e)(v) \in \{0,1\}$, where e is an edge

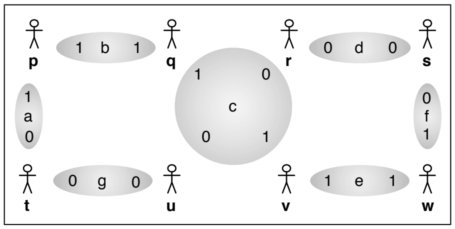

Fig. 5. Parity protocol run on graph H_3.

and v is an end of e. It will be more convenient, however, to think about a run as a two-argument function $r(e, v) \in \{0, 1\}$. We will graphically represent this function by placing boolean values at each end of each edge of the hypergraph. See Figure 5 for an example.

Not all assignments of boolean values to the ends of an edge e will be permitted in the parity protocol. Namely, if $e \notin A$, then the sum of all values assigned to the ends of e must be equal to zero modulo 2:

$$\sum_{v \in e} r(e, v) = 0 \mod 2. \tag{6}$$

However, if $e \in A$, then no restriction on the assignment of boolean values to the ends of e will be imposed. This defines the set of values $Val(e)$ for each edge e under the protocol $\mathcal{P}_A$.

The second restriction on the runs will require that the sum of all values assigned to ends incident with any vertex v is also equal to zero modulo 2:

$$\sum_{e \in E(v)} r(e, v) = 0 \mod 2, \tag{7}$$

where $E(v)$ is the set of all edges incident with v. The latter restriction specifies the local condition Loc_v for each vertex v. The protocol $\mathcal{P}_A$ is now completely defined. We just need to prove the existence of at least one run that satisfies all local conditions. Indeed, consider the run r such that $r(e, v) = 0$ for any end v of any edge e. This run clearly satisfies restrictions (6) and (7).

Theorem 9. *For any run r of the parity protocol $\mathcal{P}_A$,*

$$\sum_{e \in A} \sum_{v \in e} r(e, v) = 0 \mod 2.$$

Proof. Let $H = \langle V, E \rangle$. Using equations (7) and (6),

$$\begin{aligned} \sum_{e \in A} \sum_{v \in e} r(e, v) &= \sum_{e \in E} \sum_{v \in e} r(e, v) - \sum_{e \notin A} \sum_{v \in e} r(e, v) = \\ &= \sum_{v \in V} \sum_{e \in E(v)} r(e, v) - \sum_{e \notin A} 0 = \sum_{v \in V} 0 - 0 = 0 \mod 2. \end{aligned}$$ □

Recall that we defined a path to start and end with edges rather than vertices.

Definition 11. *For any path $\pi = e_0, v_1, e_1, \ldots, e_n$ in a hypergraph H and any run r of the parity protocol $\mathcal{P}_A$, we define r_π as*

$$r_\pi(e, v) = \begin{cases} 1 - r(e, v) & \textit{if } e = e_i, v = v_{i+1} \textit{ or } v = v_i, e = e_{i+1} \textit{ for some } i < n, \\ r(e, v) & \textit{otherwise.} \end{cases}$$

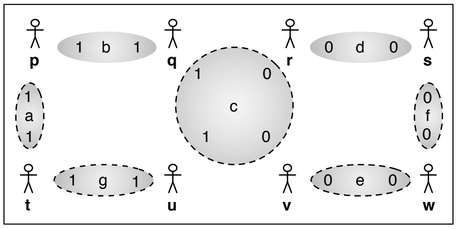

Fig. 6. Run r_π.

Informally, r_π is obtained from r by "flipping" the boolean value at each end along path π. For example, Figure 6 depicts the "flipped" run r_π, where π is $a, t, g, u, c, v, e, w, f$, and run r is the run from Figure 5. The edges along path π are indicated with dashed lines in Figure 6.

Theorem 10. *For any $r \in \mathcal{P}_A$ and any path π in a hypergraph H, if π is a cycle or starts and ends with edges that belong to set A, then $r_\pi \in \mathcal{R}(\mathcal{P}_A)$.*

Proof. Run r_π satisfies condition (6) because r_π is different from r at exactly two ends of any non-terminal edge of path π. The same run r_π satisfies condition (7) at every vertex v of the hypergraph, because path π includes either zero or two ends of edges incident at vertex v. □

Theorem 11. *If $|A| > 1$ and hypergraph H is connected, then for any $e \in A$ and any $g \in \{0, 1\}$ there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $\sum_{v \in e} r(e, v) = g \mod 2$.*

Proof. Each protocol has at least one run. Let r be a run of the protocol $\mathcal{P}_A$. Suppose that $\sum_{v \in e} r(e, v) \neq g \mod 2$. Since $|A| > 1$ and hypergraph H is connected, there is a path π that connects edge e with an edge $a \in A$ such that $a \neq e$. Notice that $\sum_{v \in e} r_\pi(e, v) = \sum_{v \in e} r(e, v) + 1 = g \mod 2$. □

Theorem 12. *If $|A| > 1$ and hypergraph H is connected, then $\mathcal{P}_A \nvDash [A]$.*

Proof. Let $A = \{a_1, \dots, a_k\}$. Pick any boolean values $g_1, \dots, g_k$ such that $g_1 + \dots + g_k = 1 \mod 2$. By Theorem 11, there are runs $r_1, \dots, r_k \in \mathcal{R}(\mathcal{P}_A)$ such that $\sum_{v \in a_i} r_i(a_i, v) = g_i \mod 2$ for any $i \leq k$. If $\mathcal{P}_A \vDash [A]$, then there is a run $r \in \mathcal{R}(\mathcal{P}_A)$ such that $r(a_i, v) = r_i(a_i, v)$ for any $v \in a_i$ and any $i \leq k$. Therefore, $\sum_{v \in a_1} r(a_1, v) + \dots + \sum_{v \in a_k} r(a_k, v) = \sum_{v \in a_1} r_1(a_1, v) + \dots + \sum_{v \in a_k} r_k(a_k, v) = g_1 + \dots + g_k = 1 \mod 2$. This contradicts Theorem 9. □

Theorem 13. *If A and B are two sets of edges of a hypergraph $H = \langle V, E \rangle$, such that each connected component of hypergraph $\langle V, E \setminus B \rangle$ contains at least one edge from A, then $\mathcal{P}_A \vDash [B]$.*

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

1. If $b_i \in A$, then, by the assumption of the theorem, there must be a path $e_0, v_1, e_1, v_2, e_2 \dots, e_n$ in the hypergraph $\langle V, E \setminus B \rangle$ such that $e_0 \in A$, and

$v \in e_n$. Consider path $\pi = e_0, v_1, e_1, v_2, e_2 \ldots, e_n, v, b_i$ in hypergraph H. By Theorem 10, $\hat{r}_\pi \in \mathcal{R}(\mathcal{P}_A)$. Note also that $\hat{r}_\pi(b_j, u) = \hat{r}(b_j, u)$ for all j and all $u \in b_j$ with the exception of $j = i$ and $u = v$. In the case that $j = i$ and $u = v$, we have $\hat{r}_\pi(b_j, u) = 1 - \hat{r}(b_j, u) = r_i(b_i, u)$. Pick $\hat{r}_\pi$ to be the new $\hat{r}$.

2. If $b_i \notin A$, then, by (6),

$$\sum_{v \in b_i} \hat{r}(b_i, v) = 0 = \sum_{v \in b_i} r_i(b_i, v) \mod 2.$$

At the same time, by our assumption, $\hat{r}(b_i, v) \neq r_i(b_i, v)$. Thus there must be $u \in b_i$ such that $u \neq v$ and $\hat{r}(b_i, u) \neq r_i(b_i, u)$. Note that vertices u and v could belong either to the same connected component or to two different connected components of hypergraph $\langle V, E \setminus B \rangle$. We will consider these two subcases separately.

(a) Suppose u and v belong to the same connected component of hypergraph $\langle V, E \setminus B \rangle$. Thus, there must be a path π' in that hypergraph which connects an edge containing vertex u with an edge containing v. Consider now a cyclic path in hypergraph $H = \langle V, E \rangle$ that starts at edge b_i, via vertex u get on the path π', goes through the whole path π', and via vertex v gets back to b_i. Call this cyclic path π.

(b) Suppose u and v belong to different connected components of hypergraph $\langle V, E \backslash B \rangle$. Thus, by the assumption of the theorem, hypergraph $\langle V, E \backslash B \rangle$ contains a path $\pi_u = a_u, \ldots, e_u$ that connects an edge $a_u \in A$ with an edge e_u containing end u. By the same assumption, hypergraph $\langle V, E \backslash B \rangle$ must also contain a path $\pi_v = e_v, \ldots, a_v$ that connects an edge e_v, containing end v, with an edge $a_v \in A$. Let $\pi = \pi_u, u, b_i, v, \pi_v$.

By Theorem 10, $\hat{r}_\pi \in \mathcal{R}(\mathcal{P}_A)$. Note also that $\hat{r}_\pi(b_j, w) = \hat{r}(b_j, w)$ for all j and all $w \in b_j$ with the exception of $j = i$ and $w \in \{u, v\}$. In the case that $j = i$ and $w \in \{u, v\}$, we have $\hat{r}_\pi(b_j, w) = 1 - \hat{r}(b_j, w) = r_i(b_i, w)$. Pick $\hat{r}_\pi$ to be the new $\hat{r}$.

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

7.2 Generalized Parity Protocol

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

Theorem 14 (protocol extension). *Let $H = \langle V, E \rangle$ be any hypergraph, X be a set of vertices in H and $H' = \langle V', E' \rangle$ be the result of the truncation of X from H. For any finite protocol $\mathcal{P}'$ on H', there is a finite protocol $\mathcal{P}$ on H such that $\mathcal{P} \vDash [Q]$ if and only if $\mathcal{P}' \vDash [Q \cap E']$, for any set $Q \subseteq E$.*

Proof. To define protocol $\mathcal{P}$, we need to specify a set of values $Val(c)$ for each edge $c \in E$ and the set of local conditions Loc_v for each vertex v in hypergraph H. If $c \in E'$, then let $Val(c)$ be the same as in protocol $\mathcal{P}'$. Otherwise, $Val(c) = \{\epsilon\}$, where ϵ is an arbitrary element. The local conditions for vertices in $V \setminus X$ are the same as in protocol $\mathcal{P}'$, and the local conditions for vertices not in X are equal to the boolean constant $True$. This completes the definition of $\mathcal{P}$. Clearly, $\mathcal{P}$ has at least one run r_0 since protocol $\mathcal{P}'$ has a run.

$(\Rightarrow)$: Suppose that $Q \cap E' = \{q_1, \dots, q_k\}$. Consider any $r'_1, \dots, r'_k \in \mathcal{R}(\mathcal{P}')$. Define runs $r_1, \dots, r_k$ as follows, for any $c \in E$:

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

Note that runs r_i and r'_i, by definition, are equal on any edge incident with any vertex in hypergraph H'. Thus, r_i satisfies the local conditions at any such vertex. Hence, $r_i \in \mathcal{R}(\mathcal{P})$ for any $i \in \{1, \dots, k\}$. Since $\mathcal{P} \vDash [Q]$, there is a run $r \in \mathcal{R}(\mathcal{P})$ such that

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

Define r' to be a restriction of r on hypergraph H'. Note that r' satisfies all local conditions of $\mathcal{P}'$. Thus, $r' \in \mathcal{R}(\mathcal{P}')$. At the same time, $r'(q_i) = r_i(q_i) = r'_i(q_i)$ for each $q_i \in Q \cap E'$.

$(\Leftarrow)$: Suppose that $Q = \{q_1, \dots, q_k\}$. Consider any $r_1, \dots, r_k \in \mathcal{R}(\mathcal{P})$, and let $r'_1, \dots, r'_k$ be their respective restrictions to hypergraph H'. Since, for any $i \in \{1, \dots, k\}$, run r'_i satisfies the local conditions of $\mathcal{P}'$ at any node of hypergraph H', we can conclude that $r'_1, \dots, r'_k \in \mathcal{R}(\mathcal{P}')$. By the assumption that $\mathcal{P}' \vDash [Q \cap E']$, there is a run $r' \in \mathcal{R}(\mathcal{P}')$ such that $r'(q) = r'_i(q)$ for any $q \in Q \cap E'$. In addition, $r'(q) = \varepsilon = r'_i(q)$ for any $q \in Q \backslash E'$. Hence, $r'(q_i) = r'_i(q_i)$ for any $i \in \{1, \dots, k\}$. Define run r as follows:

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

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 $q_i \in Q$. □

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

Theorem 15. *For any hypergraph $H = \langle V, E \rangle$ and any sets $A, B_1, \dots, B_n \subseteq E$, if $H \nvdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, then there is a finite protocol $\mathcal{P}$ over H such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for all $i \le n$.*

Proof. Induction on the size of V.
Case 1. If $|A| \le 1$, then, by the Small Set axiom, $H \vdash [A]$. Hence, $H \vdash \bigwedge_{1 \le i \le n} [B_i] \to [A]$, which is a contradiction.
Case 2. Suppose that the edges of hypergraph H can be divided into two non-trivial disconnected sets X and Y. Thus, the empty set is a gateway between $A \cap X$ and $A \cap Y$. By the Gateway axiom,

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

Thus, taking into account the assumption $H \nvDash \bigwedge_{1\le i\le n}[B_i] \to [A]$, either

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

or

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

Without loss of generality, we will assume the former. By Theorem 1,

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

By the Small Set axiom,

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

Consider the set V_Y of all vertices in component Y. Let H' be the result of the truncation of graph H that removes V_Y from H. Note that $Out(V_Y)=\varnothing$, since sets X and Y are disconnected. Thus, by the Truncation rule,

$$H' \nvDash \bigwedge_{1\le i\le n}[B_i\cap X] \to [A\cap X].$$

By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on H' such that $\mathcal{P}' \nvDash [A\cap X]$ and $\mathcal{P}' \vDash [B_i \cap X]$, for any $i \le n$. Therefore, by Theorem 14, there is a protocol $\mathcal{P}$ on H such that $\mathcal{P} \nvDash [A]$ and $\mathcal{P} \vDash [B_i]$ for any $i\le n$.

Case 3. Suppose there is $i_0 \in \{1,\dots,n\}$ such that at least one connected component of hypergraph $\langle V, E\setminus B_{i_0}\rangle$ does not contain an element of A. We will call this connected component Y. Let V_Y be the set of all vertices in this component. Note that $Out(V_Y)$ is a gateway between $In(V_Y)$ and the complement of $In(V_Y)$. Hence, $Out(V_Y)$ is also a gateway between $A\cap In(V_Y)$ and $A\setminus In(V_Y)$. Therefore, by the Gateway axiom, taking into account that $In(V_Y)\cap Out(V_Y)=\varnothing$,

$$H \vdash [A\cap In(V_Y), Out(V_Y)] \to ([A\setminus In(V_Y))] \to [A]). \tag{8}$$

Recall now that by the assumption of this case, component Y of graph $\langle V, E\setminus B_{i_0}\rangle$ does not contain any elements of A. Hence, $A\cap In(V_Y) \subseteq B_{i_0}$. At the same time, $Out(V_Y)\subseteq B_{i_0}$ by the definition of set V_Y. Thus, from statement (8) and Theorem 1,

$$H \vdash [B_{i_0}] \to ([A\setminus In(V_Y))] \to [A]). \tag{9}$$

By the assumption of the theorem,

$$H \nvDash \bigwedge_{1\le i\le n}[B_i] \to [A]. \tag{10}$$

From statements (9) and (10),

$$H \nvDash \bigwedge_{1\le i\le n}[B_i] \to [A \setminus In(V_Y)].$$

By the laws of propositional logic,

$$H \nvDash [B_{i_0}] \to (\bigwedge_{1\le i\le n}[B_i] \to [A \setminus In(V_Y)]).$$

Since $Out(V_Y) \subseteq B_{i_0}$, by Theorem 1,

$$H \nvDash [Out(V_Y)] \to (\bigwedge_{1\le i\le n}[B_i] \to [A \setminus In(V_Y)]).$$

Again by Theorem 1,

$$H \nvDash [Out(V_Y)] \to (\bigwedge_{1\le i\le n}[B_i \setminus In(V_Y)] \to [A \setminus In(V_Y)]).$$

Let H' be the result of the truncation of set V_Y from hypergraph H. By the Truncation rule,

$$H' \nvDash \bigwedge_{1\le i\le n}[B_i \setminus In(V_Y)] \to [A \setminus In(V_Y)].$$

By the Induction Hypothesis, there is a protocol $\mathcal{P}'$ on H' such that $\mathcal{P}' \nvDash [A \setminus In(V_Y)]$ and $\mathcal{P}' \vDash [B_i \setminus In(V_Y)]$ for any $i \le n$. Therefore, by Theorem 14, there is a protocol $\mathcal{P}$ on H 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) hypergraph H is connected, and (iii) for any $i \in \{1,\dots,n\}$, each connected component of hypergraph $\langle V, E \setminus B_{i_0}\rangle$ contains at least one element of A. Consider the parity protocol $\mathcal{P}_A$ over H. By Theorem 12, $\mathcal{P}_A \nvDash [A]$. By Theorem 13, $\mathcal{P}_A \vDash [B_i]$ for any $i \in \{1,\dots,n\}$. □

7.3 Completeness: final steps

Theorem 16. *For any $n \ge 0$ and any finite protocols $\mathcal{P}_1,\dots,\mathcal{P}_n$ over a hypergraph H there is a finite protocol $\mathcal{P}$ over H such that for any set of edges Q of this hypergraph, $\mathcal{P} \vDash [Q]$ if and only if $\mathcal{P}_i \vDash [Q]$ for any $i \le n$.*

Proof. First, consider the case where $n = 0$. Pick any symbol ϵ and define $\mathcal{P}$ to be $\langle Val, Loc\rangle$ such that $Val(c) = \{\epsilon\}$ for any $c \in E$, and local condition Loc_v to be the constant $True$ at every vertex v. By Definition 9, $\mathcal{P} \vDash [C]$ for any $C \subseteq E$.

We will now assume that $n > 0$ and define the composition of protocols $\mathcal{P}_1,\dots,\mathcal{P}_n$. Informally, composition is the result of several protocols run over the same hypergraph without any interaction between the protocols. Formally, suppose that $\mathcal{P}_1 = \langle Val^1, Loc^1\rangle,\dots,\mathcal{P}_n = \langle Val^n, Loc^n\rangle$ and define protocol $\mathcal{P} = \langle Val, Loc\rangle$ as follows:

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

To show that $\mathcal{P}$ is a protocol, we need to show that it has at least one run. 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 Loc_v for any vertex v of the hypergraph H. Thus, $r \in \mathcal{R}(\mathcal{P})$.

We will use notation $\{r(c)\}_i$ to denote the ith component of the value of $r(c)$.

Lemma 4. *For any set of edges 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 edge $c \in E$. 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 $c \in E$. That is, we let the value of r^* on c be the i_o-th component of $r(c)$. By 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 edges $q_1, \ldots, q_\ell$, 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 edge c, let $r_j^i(c) = \{r_j(c)\}_i$. That is, for each c, define a run r_j^i whose value on edge c equals the ith component of $r_j(c)$. Note that by the definition of composition, for each i and each j, r_j^i 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 $r^i(q_j) = r_j^i(q_j)$. 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 edge q_j on r matches the the value of q_j in run $r_j \in \mathcal{R}(\mathcal{P})$, demonstrating that $\mathcal{P} \vDash [Q]$. □

This concludes the proof of Theorem 16. □

We are now ready to prove Theorem 8.

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

By Theorem 15, there are finite protocols $\mathcal{P}^1, \dots, \mathcal{P}^n$ such that $\mathcal{P}^i \nvDash [A_i]$ and $\mathcal{P}^i \vDash [B_j]$ for all $i \leq n$ and $j \leq k$. By Theorem 16, there is a protocol $\mathcal{P}$ such that $\mathcal{P} \nvDash [A_i]$ for any $i \leq n$ and $\mathcal{P} \vDash [B_j]$ for any $j \leq k$.

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

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

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

References

1. Armstrong, W.W.: Dependency structures of data base relationships. In: Information processing 74 (Proc. IFIP Congress, Stockholm, 1974). North-Holland, Amsterdam (1974) 580–583
2. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Second edn. Prentice-Hall (2009)
3. Beeri, C., Fagin, R., Howard, J.H.: A complete axiomatization for functional and multivalued dependencies in database relations. In: SIGMOD '77: Proceedings of the 1977 ACM SIGMOD international conference on Management of data, New York, NY, USA, ACM (1977) 47–61
4. Sutherland, D.: A model of information. In: Proceedings of Ninth National Computer Security Conference. (1986) 175–183
5. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. **12**(1) (2008) 1–47 (originally appeared as [12]).
6. Miner More, S., Naumov, P.: An independence relation for sets of secrets. Studia Logica **94**(1) (2010) 73–85 (originally appeared as [11]).
7. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inform. and Comput. **91**(1) (1991) 128–141
8. Kelvey, R., Miner More, S., Naumov, P., Sapp, B.: Independence and functional dependence relations on secrets. In: Proceedings of 12th International Conference on the Principles of Knowledge Representation and Reasoning (Toronto, 2010), AAAI (2010) 528–533
9. Miner More, S., Naumov, P.: On interdependence of secrets in collaboration networks. In: Proceedings of 12th Conference on Theoretical Aspects of Rationality and Knowledge (Stanford University, 2009). (2009) 208–217
10. Berge, C.: Hypergraphs. Volume 45 of North-Holland Mathematical Library. North-Holland Publishing Co., Amsterdam (1989) Combinatorics of finite sets, Translated from the French.
11. Miner More, S., Naumov, P.: An independence relation for sets of secrets. In Ono, H., Kanazawa, M., de Queiroz, R., eds.: Proceedings of 16th Workshop on Logic, Language, Information and Computation (Tokyo, 2009), LNAI 5514, Springer (2009) 296–304
12. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. In: Proceedings of the Fifteenth IEEE Computer Security Foundations Workshop. (2002) 32–46