Noname manuscript No.
(will be inserted by the editor)

Hypergraphs of Multiparty Secrets

Sara Miner More · Pavel Naumov

the date of receipt and acceptance should be inserted later

Abstract The article 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 properties of interdependencies are expressed through a multi-argument relation called *independence*, which is a generalization of a binary relation also known as nondeducibility. The main result is a complete and decidable logical system that describes interdependencies that may exist on a fixed hypergraph. Additionally, the axioms and inference rules in this system are shown to be independent in the standard logical sense.

Keywords information flow, nondeducibility, independence, hypergraph

1 Introduction

In this article, 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.

1.0.1 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

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

multi-valued dependency. A different logical framework for reasoning about dependence was proposed by Väänänen [4].

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 [5] and is also known as *nondeducibility* in the study of information flow. Halpern and O'Neill [6] 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 [7]. Essentially the same axioms were shown by Geiger, Paz, and Pearl [8] 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 [9].

1.0.2 Secrets in Networks

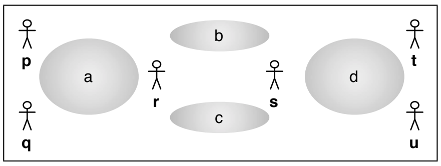

Fig. 1 Collaboration network H_0.

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.

In this article, 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[1] secret a; parties r and s

[1] In this article, the "sharing of a secret" between parties means that all parties know the entire secret in question; this is not to be confused with cryptographic secret-sharing [10].

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 article, 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 [11], 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 [12], we developed a complete logical system that describes the properties of the functional dependence relation $A \rhd B$ between sets of secrets over hypergraph 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 article, 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 and inference rules for the relation $[a_1,a_2,\ldots,a_n]$. We also prove the independence (in the standard logical sense) of the axioms and rules of this system. In terms of the proof of completeness, the most significant difference between the earlier work [11] and this one is in the construction of the parity protocol in Section 7.1.

1.1 Data Streams and Collaboration Networks

In this section, we will consider a more sophisticated example of collaboration network from network coding theory. Network coding studies methods of attaining maximum information flow in a network where channels have limited throughput. A standard example of network coding is given in terms of the butterfly network [13] depicted in Figure 2 as H_1. Suppose that parties p and q generate streams of 1-bit messages $a_1, a_2, \dots$ and $b_1, b_2, \dots$, respectively, with rate one message per second. They need to transmit both sequences of messages to both s and t using only the available communication channels. Each channel's throughput is one bit per second. Note that any protocol over H_1 that attempts to independently transmit streams of messages $\{a_i\}_i$ and $\{b_i\}_i$ will fail due to the limited combined capacity of the three channels connecting parties p, q, and r, with parties s, t, and u.

The desired result, however, can be easily achieved by a "network coding" protocol that combines the two streams. Under this protocol, at time 1, party p transmits bit a_1 to both s and r. At the same time, party q transmits bit b_1 to both t and r. At time 2, party r already possesses bits a_1 and b_1, so can compute the bit $a_1 + b_1 \mod 2$ and send it to u. At time 3, party u forwards this bit to s and t. Note that party s received bit a_1 directly from party p, and after receiving $a_1 + b_1 \mod 2$ from u one second later, s can reconstruct the value of b_1, since

$$a_1 + (a_1 + b_1) \equiv a_1 \mod 2.$$

Similarly, party t receives b_1 directly from q, and can reconstruct the Boolean value a_1 after receiving the sum from u. For each time $i > 1$, the propagation of bits a_i and b_i is carried out in a similar fashion.

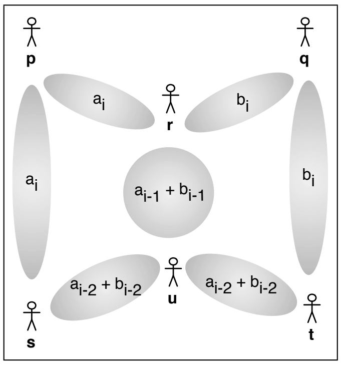

Fig. 2 Butterfly network H_1.

The coding protocol described above can be viewed as a protocol over a collaboration network if the whole stream of messages sent over a single channel in the coding network is interpreted as a single message in the collaboration network. The computation rules of the coding protocol are viewed as the local conditions of the collaboration network. For example, if the notation m_{xy} denotes the entire secret value shared between parties x and y, and $\langle m_{xy} \rangle_i$ denotes its i-th bit, then, for example, the local condition at party r can be described as

$$\forall i \geq 1 \; (\langle m_{ru} \rangle_{i+1} \equiv \langle m_{pr} \rangle_i + \langle m_{qr} \rangle_i \mod 2).$$

For the network coding protocol described above, any possible data stream between parties p and q is consistent with any possible data stream between parties r and u. Thus, in our notation, $[m_{ps}, m_{ru}]$. On the other hand, $\neg[m_{ps}, m_{ru}, m_{qt}]$, because data stream m_{ru} can be reconstructed from streams m_{ps} and m_{qt}. At the same time, $[\langle m_{ps} \rangle_i, \langle m_{ru} \rangle_i, \langle m_{qt} \rangle_i]$ for any $i > 1$.

Other network protocols that deal with data streams, such as, for example, the alternating bit protocol [14], can similarly be interpreted in terms of collaboration networks.

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 article.

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 [15, 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 [15, 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 $\mathscr{P} = \langle Val, Loc \rangle$ 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 $Loc_v(r(e_1), \dots, r(e_k))$ is a true statement.

Definition 7 A protocol is any semi-protocol that has at least one run.

The set of all runs of a protocol $\mathscr{P}$ is denoted by $\mathscr{R}(\mathscr{P})$.

Definition 8 A protocol $\mathscr{P} = \langle Val, Loc \rangle$ 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 [11] for standard graphs.

Definition 9 A set of edges $Q = \{q_1, \dots, q_k\}$ is independent under protocol $\mathscr{P}$ if for any runs $r_1, \dots, r_k \in \mathscr{R}(\mathscr{P})$ there is a run $r \in \mathscr{R}(\mathscr{P})$ 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 $\perp$ 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 $\perp$.

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

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

1. $\mathscr{P} \nvDash \perp$,
2. $\mathscr{P} \vDash [A]$ if the set of edges A is independent under protocol $\mathscr{P}$,
3. $\mathscr{P} \vDash \phi_1 \to \phi_2$ if $\mathscr{P} \nvDash \phi_1$ or $\mathscr{P} \vDash \phi_2$.

In this article, we study the formulas $\phi \in \Phi(H)$ that are true under *every* protocol $\mathscr{P}$ 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 [16,11] and by Kelvey, More, Naumov, and Sapp [9], 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 article is focused on only one such system, here we call it simply *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 is demonstrated in Section 6, and the logical independence of these principles is established in Section 8.

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 Examples of Formal Proofs

Our first example refers to hypergraph H_2 in Figure 3. 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.

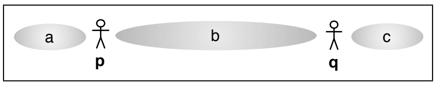

Fig. 3 Hypergraph H_2.

Theorem 2 $H_2 \vdash [a,b] \rightarrow [a,c]$.

Proof Set $\{b\}$ is a gateway between sets $\{a\}$ and $\{c\}$. Thus, by the Gateway axiom, $H_2 \vdash [a,b] \rightarrow ([c] \rightarrow [a,c])$. At the same time, $H_2 \vdash [c]$, by the Small Set axiom. Therefore, $H_2 \vdash [a,b] \rightarrow [a,c]$. □

Our second example deals with the collaboration network defined by hypergraph H_3 on Figure 4. Here, parties p, q, and r have individual secrets a,b,c, and together share secret d.

Theorem 3 $H_3 \vdash [a,d] \rightarrow ([b,d] \rightarrow [a,b,c])$.

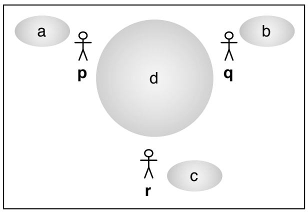

Fig. 4 Hypergraph H_3.

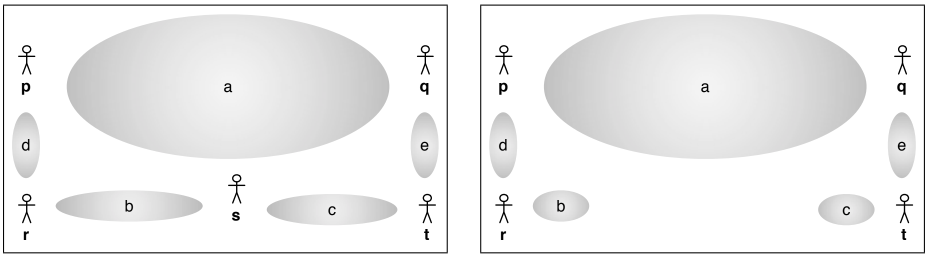

Fig. 5 Hypergraphs H_4 (left) and H'_4 (right).

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

$$H_3 \vdash [a,d] \to ([b,d] \to [a,b,d]). \tag{1}$$

Next, observe that set $\{d\}$ is a gateway between sets $\{a,b\}$ and $\{c\}$. Thus, by the Gateway axiom, $H_3 \vdash [a,b,d] \to ([c] \to [a,b,c])$. By the Small Set axiom, $H_3 \vdash [c]$. Hence,

$$H_3 \vdash [a,b,d] \to [a,b,c]. \tag{2}$$

From statements (1) and (2), it follows that $H_3 \vdash [a,d] \to ([b,d] \to [a,b,c])$. □

Our third and final example refers to hypergraph H_4 and hypergraph H'_4 obtained from H_4 by the truncation of set $\{s\}$. These graphs are depicted in Figure 5.

Theorem 4 $H_4 \vdash [a,b,c] \to ([d,a] \to [d,e])$.

Proof In hypergraph H'_4 set $\{a\}$ is a gateway between sets $\{d\}$ and $\{e\}$. Hence, by the Gateway axiom,

$$H'_4 \vdash [d,a] \to ([e] \to [d,e]).$$

By the Small Set axiom, $H'_4 \vdash [e]$. Thus,

$$H'_4 \vdash [d,a] \to [d,e].$$

By the Truncation inference rule,

$$H_4 \vdash [a,b,c] \to ([d,a] \to [d,e]).$$

□

6 Soundness

The proof of soundness, particularly for the Gateway axiom and Truncation rule, is nontrivial. 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 $\mathscr{P}$ is an arbitrary protocol over H, then $\mathscr{P} \vDash [A]$.*

Proof If $A = \varnothing$, then $\mathscr{P} \vDash [A]$ follows from the existence of at least one run of any protocol (see Definition 7). If $A = \{a_1\}$, consider any run $r_1 \in \mathscr{R}(\mathscr{P})$. 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 $\mathscr{P} \vDash [A,G]$, $\mathscr{P} \vDash [B]$, and $A \cap G = \varnothing$, then $\mathscr{P} \vDash [A,B]$.*

Proof Assume $\mathscr{P} \vDash [A,G]$, $\mathscr{P} \vDash [B]$, 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 $r \in \mathscr{R}(\mathscr{P})$ 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 $\mathscr{P} \vDash [B]$, there is $r_b \in \mathscr{R}(\mathscr{P})$ such that

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

By the assumptions $\mathscr{P} \vDash [A,G]$ 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 $\mathscr{P}$. 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 $i \le n$. 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 $i \le k$. 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 $\mathscr{P}' \vDash \phi$ for any protocol $\mathscr{P}'$ over hypergraph H', then $\mathscr{P} \vDash [Out(X)] \to \phi$ for any protocol $\mathscr{P}$ over hypergraph H.*

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

Let $\mathscr{P} = \langle Val, Loc\rangle$. 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 $\mathscr{P}$. To construct protocol $\mathscr{P}' = \langle Val', Loc'\rangle$ 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 $\mathscr{P}$:

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

The local condition Loc'_v at any vertex v of hypergraph H' is the same as under protocol $\mathscr{P}$. To show that protocol $\mathscr{P}'$ has at least one run, notice that the restriction of any run of $\mathscr{P}$ to edges in H' constitutes a valid run of $\mathscr{P}'$.

Lemma 1 *For any run $r' \in \mathscr{R}(\mathscr{P}')$ there is a run $r \in \mathscr{R}(\mathscr{P})$ such that $r(e) = r'(e)$ for each edge e in hypergraph H'.*

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

We will now construct a combined run $r \in \mathscr{R}(\mathscr{P})$ 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 *$\mathscr{P} \vDash [Q]$ if and only if $\mathscr{P}' \vDash [Q]$, for any set of edges Q in H'.*

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

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

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

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 $\mathscr{P} \vDash \phi$ for all finite protocols $\mathscr{P}$ 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 $\{\mathscr{P}_A\}_A$ defined below.

7.1 Parity Protocol $\mathscr{P}_A$

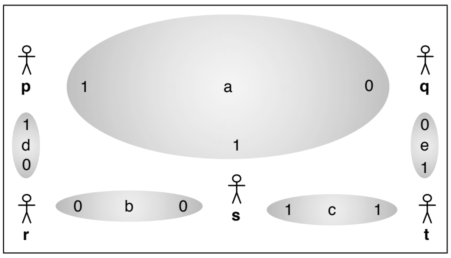

Fig. 6 Parity protocol run on graph H_4.

Let $H = \langle V, E\rangle$ be a hypergraph and A be a subset of E. We define the "parity protocol" $\mathscr{P}_A$ 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 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 6 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:

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

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 $\mathscr{P}_A$.

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:

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

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 $\mathscr{P}_A$ 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 $\mathscr{P}_A$,*

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

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

$$\sum_{e\in A}\sum_{v\in e} r(e,v) = \sum_{e\in E}\sum_{v\in e} r(e,v) - \sum_{e\notin A}\sum_{v\in e} r(e,v) \equiv$$
$$\equiv \sum_{v\in V}\sum_{e\in E(v)} r(e,v) - \sum_{e\notin A} 0 = \sum_{v\in V} 0 - 0 \equiv 0 \mod 2.$$

□

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

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

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

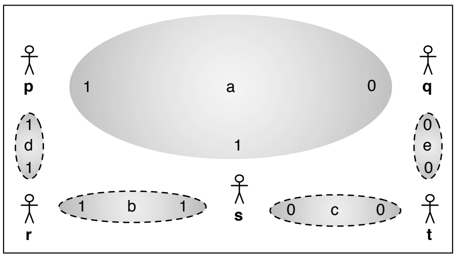

Fig. 7 Run r_π.

Informally, r_π is obtained from r by "flipping" the boolean value at each end along path π. For example, Figure 7 depicts the "flipped" run r_π, where π is d, r, b, s, c, t, e, and run r is the run from Figure 6. The edges along path π are indicated with dashed lines in Figure 7.

Theorem 10 *For any $r \in \mathscr{P}_A$ 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 $r_\pi \in \mathscr{R}(\mathscr{P}_A)$.*

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 $r \in \mathscr{R}(\mathscr{P}_A)$ such that $\sum_{v\in e} r(e,v) \equiv g \mod 2$.*

Proof Each protocol has at least one run. Let r be a run of the protocol $\mathscr{P}_A$. Suppose that $\sum_{v\in e} r(e,v) \not\equiv g \mod 2$. 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

$$\sum_{v\in e} r_\pi(e,v) = \sum_{v\in e} r(e,v) + 1 \equiv g \mod 2.$$

□

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

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

$$\sum_{v \in a_1} r(a_1, v) + \dots + \sum_{v \in a_k} r(a_k, v) = \sum_{v \in a_1} r_1(a_1, v) + \dots + \sum_{v \in a_k} r_k(a_k, v) \equiv g_1 + \dots + g_k \equiv 1 \mod 2.$$

This contradicts Theorem 9. □

Theorem 13 *If A and B are 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 $\mathscr{P}_A \vDash [B]$.*

Proof Let $B = \{b_1, \dots, b_k\}$. Consider any runs $r_1, \dots, r_k \in \mathscr{R}(\mathscr{P}_A)$. We will prove that there is a run $r \in \mathscr{R}(\mathscr{P}_A)$ such that $r(b_i, v) = r_i(b_i, v)$ for any $v \in b_i$ and any $i \leq k$. Indeed, protocol $\mathscr{P}_A$ 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 $\pi = e_0, v_1, e_1, v_2, e_2, \dots, e_n, v, b_i$ in hypergraph H. By Theorem 10, $\hat{r}_\pi \in \mathscr{R}(\mathscr{P}_A)$. 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),

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

   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 $\langle V, E \setminus B \rangle$. Thus, by the assumption of the theorem, hypergraph $\langle V, E \setminus B \rangle$ contains a path $\pi_u = a_u, \dots, e_u$ that connects an edge $a_u \in A$ with an edge e_u containing end u. By the same assumption, hypergraph $\langle V, E \setminus B \rangle$ must also contain a path $\pi_v = e_v, \dots, a_v$ 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, $\hat{r}_\pi \in \mathscr{R}(\mathscr{P}_A)$. 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 $r(b_i, v) = \hat{r}(b_i, v) = r_i(b_i, v)$ 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 $\mathscr{P}'$ on H', there is a finite protocol $\mathscr{P}$ on H such that $\mathscr{P} \vDash [Q]$ if and only if $\mathscr{P}' \vDash [Q \cap E']$, for any set $Q \subseteq E$.*

Proof To define protocol $\mathscr{P}$, 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 $\mathscr{P}'$. Otherwise, $Val(c) = \{\varepsilon\}$, where ε is an arbitrary element. The local conditions for vertices in $V \setminus X$ are the same as in protocol $\mathscr{P}'$, and the local conditions for vertices not in X are equal to the boolean constant $True$. This completes the definition of $\mathscr{P}$. Clearly, $\mathscr{P}$ has at least one run r_0 since protocol $\mathscr{P}'$ has a run.

$(\Rightarrow)$: Suppose that $Q \cap E' = \{q_1, \dots, q_k\}$. Consider any $r'_1, \dots, r'_k \in \mathscr{R}(\mathscr{P}')$. 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, $r_i \in \mathscr{R}(\mathscr{P})$ for any $i \in \{1, \dots, k\}$. Since $\mathscr{P} \vDash [Q]$, there is a run $r \in \mathscr{R}(\mathscr{P})$ 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 $\mathscr{P}'$. Thus, $r' \in \mathscr{R}(\mathscr{P}')$. 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 $r_1, \dots, r_k \in \mathscr{R}(\mathscr{P})$, 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 $\mathscr{P}'$ at any node of hypergraph H', we can conclude that $r'_1, \dots, r'_k \in \mathscr{R}(\mathscr{P}')$. By the assumption that $\mathscr{P}' \vDash [Q \cap E']$, there is a run $r' \in \mathscr{R}(\mathscr{P}')$ 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 $\mathscr{P}$ at all nodes. Thus, $r \in \mathscr{R}(\mathscr{P})$. 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 $\mathscr{P}$ over H such that $\mathscr{P} \nvDash [A]$ and $\mathscr{P} \vDash [B_i]$ for all $i \le n$.

Proof Induction on the size of V.
Case 1. If $|A| \leq 1$, then, by the Small Set axiom, $H \vdash [A]$. Hence, $H \vdash \bigwedge_{1\leq i\leq n}[B_i] \to [A]$, 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 $H \nvdash \bigwedge_{1\leq i\leq n}[B_i] \to [A]$, either

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

or

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

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

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

By the Small Set axiom,

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

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,

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

By the Induction Hypothesis, there is a protocol $\mathscr{P}'$ on H' such that $\mathscr{P}' \nvDash [A\cap X]$ and $\mathscr{P}' \vDash [B_i \cap X]$, for any $i \leq n$. Therefore, by Theorem 14, there is a protocol $\mathscr{P}$ on H such that $\mathscr{P} \nvDash [A]$ and $\mathscr{P} \vDash [B_i]$ for any $i \leq n$.
Case 3. Suppose there is $i_0 \in \{1,\ldots,n\}$ 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,

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

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 $\mathscr{P}'$ on H' such that $\mathscr{P}' \nvDash [A \setminus In(V_Y)]$ and $\mathscr{P}' \vDash [B_i \setminus In(V_Y)]$ for any $i \le n$. Therefore, by Theorem 14, there is a protocol $\mathscr{P}$ on H such that $\mathscr{P} \nvDash [A]$ and $\mathscr{P} \vDash [B_i]$ 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 $\mathscr{P}_A$ over H. By Theorem 12, $\mathscr{P}_A \nvDash [A]$. By Theorem 13, $\mathscr{P}_A \vDash [B_i]$ for any $i \in \{1,\dots,n\}$. □

7.3 Completeness: Final Steps

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

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

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

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

To show that $\mathscr{P}$ is a protocol, we need to show that it has at least one run. Let $r^1,\dots,r^n$ be runs of $\mathscr{P}^1,\dots,\mathscr{P}^n$. Define $r(c)$ to be $\langle r^1(c),\dots,r^n(c)\rangle$. It is easy to see that r satisfies the local conditions Loc_v for any vertex v of the hypergraph H. Thus, $r \in \mathscr{R}(\mathscr{P})$.

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,*

$$\mathscr{P} \vDash [Q] \quad \textit{if and only if} \quad \forall i\,(\mathscr{P}_i \vDash [Q]).$$

Proof Let $Q = \{q_1, \ldots, q_\ell\}$.

$(\Rightarrow)$: Assume $\mathscr{P} \vDash [Q]$ and pick any $i_0 \in \{1, \ldots, n\}$. We will show that $\mathscr{P}_{i_0} \vDash [Q]$. Pick any runs $r'_1, \ldots, r'_\ell \in \mathscr{R}(\mathscr{P}_{i_0})$. For each $i \in \{1, \ldots, i_0 - 1, i_0 + 1, \ldots, n\}$, select an arbitrary run $r^i \in \mathscr{R}(\mathscr{P}_i)$. 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 $\mathscr{R}(\mathscr{P}_i)$, the composed runs are themselves members of $\mathscr{R}(\mathscr{P})$. By our assumption, $\mathscr{P} \vDash [Q]$, thus there is $r \in \mathscr{R}(\mathscr{P})$ 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, $r^* \in \mathscr{R}(\mathscr{P}_{i_0})$, and it matches the original $r'_1, \ldots, r'_\ell \in \mathscr{R}(\mathscr{P}_{i_0})$ on edges $q_1, \ldots, q_\ell$, respectively. Hence, we have shown that $\mathscr{P}_{i_0} \vDash [Q]$.

$(\Leftarrow)$: Assume $\forall i\,(\mathscr{P}_i \vDash [Q])$. We will show that $\mathscr{P} \vDash [Q]$. Pick any runs $r_1, \ldots, r_\ell \in \mathscr{R}(\mathscr{P})$. For each $i \in \{1, \ldots, n\}$, each $j \in \{1, \ldots, \ell\}$, and each edge c, let $r^i_j(c) = \{r_j(c)\}_i$. That is, for each c, define a run r^i_j 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^i_j is a run in $\mathscr{R}(\mathscr{P}_i)$. Next, for each $i \in \{1, \ldots, n\}$, we use the fact that $\mathscr{P}_i \vDash [Q]$ to construct a run $r^i \in \mathscr{R}(\mathscr{P}_i)$ 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 $r \in \mathscr{R}(\mathscr{P})$. We note that the value of each edge q_j on r matches the the value of q_j in run $r_j \in \mathscr{R}(\mathscr{P})$, demonstrating that $\mathscr{P} \vDash [Q]$. □

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 $\mathscr{P}$ such that $\mathscr{P} \nvDash [A_i]$ for any $i \le n$ and $\mathscr{P} \vDash [B_j]$ for any $j \le k$.

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

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

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. □

8 Logical Independence of the Axioms and the Inference Rule

Theorem 17 *The Small Set axiom is not derivable from the Gateway axiom and the Truncation inference rule.*

Proof Consider a new semantics for the language of secrets under which statement $[A]$ is always false. The set of all statements of the form $G \vdash \phi$ that are true under this semantics (i) includes all instances of the Gateway axiom since these formulas contain the assumption $[A,G]$, and (ii) is closed with respect to the Truncation inference rule since the conclusion of this rule contains the assumption $[Out(X)]$. However, this set does not include a single instance of the Small Set axiom. □

Theorem 18 *The Gateway axiom is not derivable from the Small Set axiom and the Truncation inference rule.*

Proof Consider a new semantics for the language of secrets under which statement $[A]$ means that $|A| < 2$. We will show that the set of statements of the form $H \vdash \phi$ that are true under this semantics includes all instances of the Small Set axiom, is closed with respect to Truncation inference rule, and, at the same time, does not contain all instances of the Gateway axiom.

1. All instances of the Small Set axiom are trivially true under this semantics simply by the definition of the semantics.
2. Let us now show that the set of true sentences is closed under the Truncation inference rule. Indeed, note that if hypergraph H' is obtained from hypergraph H by a truncation and $[A] \in \Phi(H')$, then $[A]$ is true on H' if and only if $[A]$ is true on H. Thus, by induction on structural complexity of ϕ, one can show that if $\phi \in \Phi(H')$, then ϕ is true on H' if and only if ϕ is true on H. Therefore, if ϕ is true on H', then $[C] \to \phi$ is true on H for any set of edges C.
3. Finally, we will show that the set does not include at least some instances of the Gateway axiom by giving a specific example of a hypergraph H and sets of edges A, G, and B for which Gateway axiom is not true under the new semantics. Let H be a hypergraph consisting of two disconnected edges a and b. Let $A = \{a\}$, $G = \varnothing$, and $B = \{b\}$. By the definition of the semantics, statement $[A,G]$ and $[B]$ are true on hypergraph H. At the same time, statement $[A,B]$ is false since set $A \cup B$ contains more than one edge. Therefore, Gateway axiom $[A,G] \to ([B] \to [A,B])$ is false on the hypergraph H. □

Theorem 19 *The Truncation inference rule is not admissible in the logical system that consists of the Small Set axiom, the Gateway axiom, propositional tautologies, and the Modus Ponens inference rule.*

Proof We first will show that statement

$$H_5 \vdash [a,b] \to ([b,c] \to ([c,a] \to [a,b,c])), \tag{11}$$

where H_5 is the hypergraph in Figure 8, is provable in the Logic of Secrets (with the Truncation inference rule). Indeed, we first consider hypergraph H_5' obtained from H_5 by the truncation of set $\{p\}$. Hypergraph H_5' is also depicted in Figure 8. Note that in hypergraph H_5', set $\{c\}$ is a gateway between sets of edges $\{b\}$ and $\{c,a\}$. Thus, by the Gateway axiom,

$$H_5' \vdash [b,c] \to ([c,a] \to [a,b,c])$$

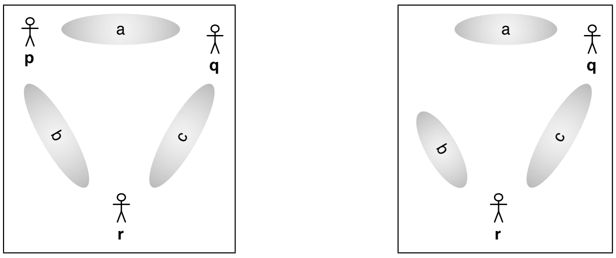

Fig. 8 Hypergraph H_5 (left) and truncated hypergraph H_5' (right).

By the Truncation inference rule,

$$H_5 \vdash [a,b] \to ([b,c] \to ([c,a] \to [a,b,c])).$$

Now it will be sufficient to establish that the same statement (11) is not provable in the logical system without the Truncation inference rule. To do this, we will describe a new semantics under which the Small Set and the Gateway axioms are true and formula $[a,b] \to ([b,c] \to ([c,a] \to [a,b,c]))$ is false. Note that, in the absence of the Truncation rule, our logical system does not modify hypergraph and, thus, it will be sufficient to construct new semantics for hypergraph H_5 only.

Consider a semantics under which, for any subset A of edges of hypergraph H_5, statement $[A]$ is true if and only if $|A| < 3$. The Small Set axiom is clearly true, and formula $[a,b] \to ([b,c] \to ([c,a] \to [a,b,c]))$ is clearly false under this semantics. The fact that the Gateway axiom is true under this semantics follows from the lemma below.

Lemma 5 *For any subsets of edges A, G, and B of the hypergraph H_5, such that $A \cap G = \varnothing$ and G is a gateway between set A and B, if $|A \cup G| < 3$ and $|B| < 3$, then $|A \cup B| < 3$.*

Proof Assume that $|A \cup B| = 3$. Thus, $A \cup B = \{a,b,c\}$. We will consider the following three cases separately:
Case 1: $|A| = 3$. This contradicts the assumption that $|A \cup G| < 3$.
Case 2: $|B| = 3$. This contradicts the assumption that $|B| < 3$.
Case 3: $|A| < 3$ and $|B| < 3$. Recall our assumption that $|A \cup B| = 3$. Thus, sets $A \setminus B$ and $B \setminus A$ are not empty. Let $x \in A \setminus B$ and $y \in B \setminus A$. Note that any two edges in hypergraph H_5 are adjacent. Thus, x and y are adjacent. Consider the path x,y in hypergraph H_5. It connects edge $x \in A$ with edge $y \in B$. Thus, this path must contain an edge from the gateway G. By the assumption that $A \cap G = \varnothing$, edge x can not be in G. Hence, $y \in G$.

Since $x \in A \setminus B$ and $y \in B \setminus A$, edges x and y are two different edges in hypergraph H_5. Let z be the remaining third edge of this hypergraph. Since $|A \cup B| = 3$, edge z must belong to at least one of the sets A and B. If $z \in A$, then $A \cup G \supseteq \{x,z\} \cup \{y\}$, which contradicts the assumption that $|A \cup G| < 3$. Thus, $z \in B$. In this, case, however, x,z is a path connecting $x \in A$ and $z \in B$. Thus, it must contain an edge from gateway G. Due to the assumption $A \cap G = \varnothing$, edge x can not be in G. Thus, $z \in G$. Therefore, $A \cup G \supseteq \{x\} \cup \{z,y\}$, which contradicts the assumption $|A \cup G| < 3$. □

This concludes the proof of Theorem 19. □

9 Conclusion

In this article, we extended our previous work [11] from graphs to hypergraphs. In both settings, we assumed that all ends of an edge "collaborate" together to produce the secret value of that edge. Another possible direction for extension of the original work is to consider collaboration networks formed by directed graphs, and, later, directed hypergraphs [17].

When considering directed graphs, the simplest case to analyze is the directed acyclic graph (DAG) setting. In a DAG, each secret has a single active end, the sender of the message, and a single passive end, the recipient of the message. The distinction between active and passive ends can be captured formally by adding a "continuability" condition to the definition of a protocol. A protocol is continuable at a vertex v if, for any assignment of values to the *incoming* edges of vertex v, there is an extension of this assignment on the *outgoing* edges of vertex v that satisfies local condition Loc_v. An entire protocol over a DAG is continuable if it is continuable at each vertex of the DAG. Then, the logic of secrets over a directed acyclic graph would be the set of all formulas that are true under any continuable protocol over the DAG.

Since each continuable protocol is a protocol in our original sense, all theorems of the logic of secrets for undirected graphs are still true for DAGs. In addition, some new properties become true. For example, consider directed acyclic graph G shown in Figure 9. Since vertices p and r have no way to coordinate their choices of values of a and b, it can be shown that the values of these secrets must be independent in G. In fact, this result follows from a more general property of directed graphs: if DAG G' is obtained from DAG G by the elimination of one of the sinks in the graph, then $G' \vdash \phi$ implies $G \vdash \phi$. A complete description of the properties of independence on directed acyclic graphs remains an open question.

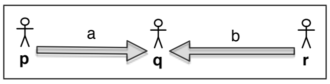

Fig. 9 Collaboration DAG G.

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. Väänänen, J.: Dependence logic. Volume 70 of London Mathematical Society Student Texts. Cambridge University Press, Cambridge (2007) A new approach to independence friendly logic.
5. Sutherland, D.: A model of information. In: Proceedings of Ninth National Computer Security Conference. (1986) 175–183
6. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. **12**(1) (2008) 1–47
7. Miner More, S., Naumov, P.: An independence relation for sets of secrets. Studia Logica **94**(1) (2010) 73–85
8. Geiger, D., Paz, A., Pearl, J.: Axioms and algorithms for inferences involving probabilistic independence. Inform. and Comput. **91**(1) (1991) 128–141
9. 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
10. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11) (November 1979) 612–613

11. 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
12. Miner More, S., Naumov, P.: The functional dependence relation on hypergraphs of secrets. In: 12th International Workshop on Computational Logic in Multi-Agent Systems CLIMA XI (Barcelona, Spain), Springer (2011) (to appear).
13. Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: Network information flow. IEEE Trans. Inform. Theory **46**(4) (2000) 1204–1216
14. Bartlett, K.A., Scantlebury, R.A., Wilkinson, P.T.: A note on reliable full-duplex transmission over half-duplex links. Commun. ACM **12**(5) (1969) 260–261
15. Berge, C.: Hypergraphs. Volume 45 of North-Holland Mathematical Library. North-Holland Publishing Co., Amsterdam (1989) Combinatorics of finite sets, Translated from the French.
16. 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
17. Gallo, G., Longo, G., Pallottino, S., Nguyen, S.: Directed hypergraphs and applications. Discrete Appl. Math. **42**(2-3) (1993) 177–201 Combinatorial structures and algorithms.