The Functional Dependence Relation on Hypergraphs of 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. In previous work, the authors investigated properties of interdependencies that are expressible through a multi-argument relation called *independence*, which is a generalization of a binary relation also known as nondeducibility. This work studies properties expressible through functional dependence. The main result is a complete and decidable logical system that describes interdependencies on a fixed hypergraph.

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 the 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 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 [1] and is also known as *nondeducibility* in the study of information flow. Halpern and O'Neill [2] proposed a closely-related notion called f-secrecy. In earlier work [3, 4], we generalized independence to a relation $[a_1, \dots, a_n]$ between an arbitrary set of secrets.

Another natural relation 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. 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 [5] presented the following sound and complete axiomatization of this relation:

1. *Reflexivity*: $A \rhd B$, if $A \supseteq B$,
2. *Augmentation*: $A \rhd B \to A, C \rhd B, C$,
3. *Transitivity*: $A \rhd B \to (B \rhd C \to A \rhd C)$,

where here and everywhere below A, B denotes the union of sets A and B. The above axioms are known in database literature as Armstrong's axioms [6, p. 81]. Beeri, Fagin, and Howard [7] suggested a variation of Armstrong's axioms that describe properties of multi-valued dependence. A logical system that combines independence and functional dependence predicates was described by Kelvey, More, Naumov, and Sapp [8].

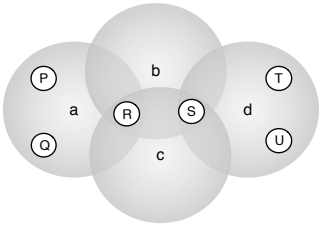

Fig. 1. Network H_0.

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.

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 in Figure 1. In this network, parties P, Q and R share[1] 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

[1] In this paper, 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 [9].

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 $a, b \rhd c, d$, 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 $a \rhd d \to b, c \rhd d$ and $[a, b, c] \to [a, d]$.

In our previous CLIMA paper [4], we gave a complete axiomatization of all properties of independence between sets of secrets over an arbitrary collaboration network. In this work, we give a similar axiomatization for the properties of functional dependence. It consists of the above-mentioned Armstrong axioms and an additional *Gateway* axiom that captures properties specific to the topology of the collaboration network.

Although the proposed logical system captures properties of functional dependence that are not specific to any protocol, this logic could potentially be used as a framework for reasoning about specific protocols in the same way, for example, as the first order logic is used for reasoning about specific mathematical theories.

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 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 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.

A *path* in a hypergraph is a sequence of edges in which adjacent edges share at least one end. Paths will be assumed to be simple, in the sense that no edge is repeated in a path.

Definition 2. *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.

3 Protocol: A Formal Definition

Definition 3. *A 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 $Inc(v)$ is the set of all edges incident with vertex v, then $Loc_v \subseteq \prod_{e\in Inc(v)} Val(e)$.*

Definition 4. *A run of a protocol $\langle Val, Loc\rangle$ is a function r such that*

1. *$r(e) \in Val(e)$ for any edge $e \in E$,*
2. *$\langle r(e)\rangle_{e\in Inc(v)} \in Loc(v)$.*

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

Definition 5. *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.*

We conclude this section with the key definition of this paper. It is the definition of functional dependence between sets of edges.

Definition 6. *A set of edges A functionally determines a set of edges B, with respect to a fixed protocol $\mathcal{P}$, if*

$$\forall r, r' \in \mathcal{R}(\mathcal{P}) \left(\bigwedge_{a\in A} r(a) = r'(a) \to \bigwedge_{b\in B} r(b) = r'(b) \right).$$

We find it convenient to use the notation $f \equiv_X g$ if functions f and g are equal on every argument from set X. Using this notation, we can say that a set of edges A functionally determines a set of edges B if

$$\forall r, r' \in \mathcal{R}(\mathcal{P})\ (r \equiv_A r' \to r \equiv_B r').$$

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 functional dependence predicate. More formally, $\Phi(H)$ is the minimal set of formulas defined recursively as follows: (i) for any finite subsets A and B of the set of all edges of hypergraph H, formula $A \rhd B$ is in $\Phi(H)$, (ii) the false constant $\bot$ is in $\Phi(H)$, 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 7. *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. $\mathcal{P} \vDash A \rhd B$ *if the set of edges A functionally determines set of edges B under protocol $\mathcal{P}$,*
3. $\mathcal{P} \vDash \phi_1 \to \phi_2$ *if* $\mathcal{P} \nvDash \phi_1$ *or* $\mathcal{P} \vDash \phi_2$.

In this article, we study the formulas $\phi \in \Phi(H)$ that are true under *every* 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 [5], More and Naumov [10, 3, 4] 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 article is focused on only one such system, here we call it simply the *Logic of Secrets* of hypergraph H.

5 Axioms

For a fixed hypergraph H, the Logic of Secrets, in addition to propositional tautologies and the Modus Ponens inference rule, contains the following axioms:

1. *Reflexivity*: $A \rhd B$, if $A \supseteq B$,
2. *Augmentation*: $A \rhd B \to A, C \rhd B, C$,
3. *Transitivity*: $A \rhd B \to (B \rhd C \to A \rhd C)$,
4. *Gateway* : $A \rhd B \to G \rhd B$, if G is a gateway between sets A and B in hypergraph H.

Recall that the first three of these axioms were introduced by Armstong [5]. The soundness of all four axioms will be shown in Section 7. We use the notation $X \vdash_H \Phi$ to state that formula Φ is derivable from the set of formulas X in the Logic of Secrets for hypergraph H.

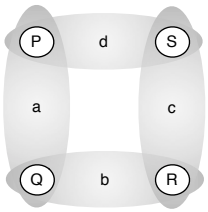

Fig. 2. Hypergraph H_1.

6 Examples of Proofs

In this section, we give four examples of proofs in the Logic of Secrets. Our first example refers to the square hypergraph H_1 depicted in Figure 2.

Proposition 1. $\vdash_{H_1} (a \rhd c) \wedge (b \rhd d) \rightarrow (a \rhd d) \wedge (b \rhd c)$.

Proof. Due to the symmetry of the hypergraph, it is sufficient to show that $(a \rhd c) \wedge (b \rhd d) \rightarrow a \rhd d$. Note that $\{a, c\}$ is a gateway between sets $\{b\}$ and $\{d\}$. Thus, by the Gateway axiom, $b \rhd d$ implies $(a, c \rhd d)$. On the other hand, by the Augmentation axiom, the assumption $a \rhd c$ yields $(a \rhd a, c)$. By the Transitivity axiom, $(a \rhd a, c)$ and $(a, c \rhd d)$ imply $a \rhd d$. □

For the second example, consider the linear hypergraph H_2 shown in Figure 3.

Fig. 3. Hypergraph H_2.

Proposition 2. $\vdash_{H_2} (a \rhd d) \wedge (e \rhd c) \rightarrow b \rhd c$.

Proof. We begin with the assumption that $e \rhd c$. Since $\{d\}$ is a gateway between sets $\{e\}$ and $\{c\}$, by the Gateway axiom, $d \rhd c$. Next, using the assumption that $a \rhd d$, the Transitivity axiom yields $a \rhd c$. Finally, we note that $\{b\}$ is a gateway between $\{a\}$ and $\{c\}$, and apply the Gateway axiom once again to conclude that $b \rhd c$. □

Note that the second hypothesis in the example above is significant. Indeed, imagine a protocol on H_2 where $V(d) = \{0\}$, the set of values allowed on all other edges is $\{0, 1\}$, and the local condition at each vertex v is always true, or, formally, $L(v) \equiv \prod_{e \in Inc(v)} Val(e)$. Under this protocol, $a \rhd d$ since the value of a on any run trivially determines the (constant) value of d. However, the value of b is of no help in determining the value of c, so the conclusion $b \rhd c$ does not hold.

Next, consider the "olympic rings" hypergraph H_3 shown in Figure 4.

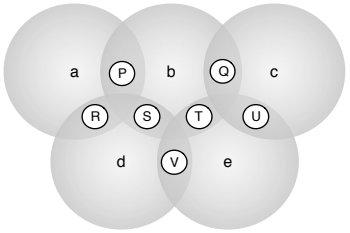

Fig. 4. Hypergraph H_3.

Proposition 3. $\vdash_{H_3} (a \rhd e) \wedge (c \rhd d) \rightarrow b, c \rhd e$.

Proof. Assume that $a \rhd e$ and $c \rhd d$. Note that set $\{d, b\}$ is a gateway between sets $\{a\}$ and $\{e\}$. Thus, by the Gateway axiom, from assumption $a \rhd e$ we can conclude that $d, b \rhd e$. The assumption $c \rhd d$, by the Augmentation axiom, implies that $b, c \rhd b, d$. Therefore, by the Transitivity axiom, $b, c \rhd e$. □

As our final example, we prove a property of the hexagonal hypergraph H_3 shown in Figure 5.

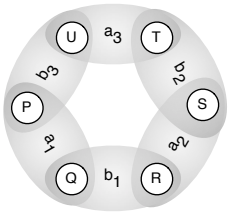

Fig. 5. Hypergraph H_4.

Proposition 4. $\vdash_{H_4} (a_1, a_2 \rhd a_3) \wedge (a_2, a_3 \rhd a_1) \wedge (a_3, a_1 \rhd a_2) \rightarrow b_1, b_2, b_3 \rhd a_1, a_2, a_3$.

Proof. Note that $\{b_1, b_3\}$ is a gateway between sets $\{a_2, a_3\}$ and $\{a_1\}$. Thus, by the Gateway axiom, $a_2, a_3 \rhd a_1 \rightarrow b_1, b_3 \rhd a_1$. Hence, by the assumption, $a_2, a_3 \rhd a_1$, we have that $b_1, b_3 \rhd a_1$. Similarly one can show that $b_1, b_2 \rhd a_2$ and $b_2, b_3 \rhd a_3$ using the assumptions $a_3, a_1 \rhd a_2$ and $a_1, a_2 \rhd a_3$.

Consider statements $b_1, b_3 \rhd a_1$ and $b_1, b_2 \rhd a_2$. By the Augmentation axiom, they, respectively, imply that $b_1, b_2, b_3 \rhd a_1, b_1, b_2$ and $a_1, b_1, b_2 \rhd a_1, a_2$. Thus, by the Transitivity axiom, $b_1, b_2, b_3 \rhd a_1, a_2$.

Now consider $b_1, b_2, b_3 \rhd a_1, a_2$ and statement $b_2, b_3 \rhd a_3$, established earlier. By the Augmentation axiom, they, respectively, imply that $b_1, b_2, b_3 \rhd a_1, a_2, b_2, b_3$

and $a_1, a_2, b_2, b_3 \triangleright a_1, a_2, a_3$. Thus, by the Transitivity axiom, $b_1, b_2, b_3 \triangleright a_1, a_2, a_3$. □

7 Soundness

In this section, we demonstrate the soundness of each of the four axioms in the Logic of Secrets.

Theorem 1 (Reflexivity). *$\mathcal{P} \vDash A \triangleright B$, for any protocol $\mathcal{P}$ and any $B \subseteq A$.*

Proof. Consider any two runs $r, r' \in \mathcal{R}(\mathcal{P})$ such that $r \equiv_A r'$. Thus $r \equiv_B r'$ for any $B \subseteq A$. □

Theorem 2 (Augmentation). *$\mathcal{P} \vDash A \triangleright B \rightarrow A, C \triangleright B, C$, for any protocol $\mathcal{P}$ and any sets of edges A, B, and C.*

Proof. Assume $\mathcal{P} \vDash A \triangleright B$ and consider any two runs $r, r' \in \mathcal{R}(\mathcal{P})$ such that $r \equiv_{A,C} r'$. By our assumption, $r \equiv_B r'$. Therefore, $r \equiv_{B,C} r'$. □

Theorem 3 (Transitivity). *$\mathcal{P} \vDash A \triangleright B \rightarrow (B \triangleright C \rightarrow A \triangleright C)$, for any protocol $\mathcal{P}$ and any sets of edges A, B, and C.*

Proof. Assume $\mathcal{P} \vDash A \triangleright B$ and $\mathcal{P} \vDash B \triangleright C$. Consider any two runs $r, r' \in \mathcal{R}(\mathcal{P})$ such that $r \equiv_A r'$. By the first assumption, $r \equiv_B r'$. By the second assumption, $r \equiv_C r'$. □

Theorem 4 (Gateway). *$\mathcal{P} \vDash A \triangleright B \rightarrow G \triangleright B$, for any protocol $\mathcal{P}$ and any gateway G between sets A and B.*

Proof. Assume $\mathcal{P} \vDash A \triangleright B$ and consider any two runs $r_1, r_2 \in \mathcal{R}(\mathcal{P})$ such that $r_1 \equiv_G r_2$. We will show that $r_1 \equiv_B r_2$. Consider the hypergraph H' obtained from H by removal of all edges in set G. By the definition of a gateway, no single connected component of hypergraph H' can contain edges from set $A \setminus G$ and set $B \setminus G$ at the same time. Let us divide all connected components of H' into two subgraphs H'_A and H'_B such that H'_A contains no edges from $B \setminus G$ and H'_B contains no edges from $A \setminus G$. Components that do not contain edges from either $A \setminus G$ or $B \setminus G$ can be arbitrarily assigned to either H'_A or H'_B.

Next, define a function r on each $c \in E$ as follows:

$$r(c) = \begin{cases} r_1(c) & \text{if } c \in H'_A, \\ r_1(c) = r_2(c) & \text{if } c \in G, \\ r_2(c) & \text{if } c \in H'_B. \end{cases}$$

We will prove that r is a run of protocol $\mathcal{P}$. We need to show that r satisfies the local conditions of protocol $\mathcal{P}$ at each vertex v. The connected component of H' containing a vertex v either belongs to H'_A or H'_B. Without loss of generality, assume that it belongs to H'_A. Thus, $Inc(v)$, the set of all edges in H incident with vertex v, is a subset of $H'_A \cup G$. Hence, $r \equiv_{Inc(v)} r_1$. Therefore, r satisfies the local condition at vertex v simply because r_1 does.

By the definition of r, we have $r \equiv_A r_1$ and $r \equiv_B r_2$. Together, the first of these statements and the assumption that $\mathcal{P} \vDash A \triangleright B$ imply that $r \equiv_B r_1$. Thus, due to the second statement, $r_1 \equiv_B r \equiv_B r_2$. □

8 Completeness

In this section, we demonstrate that the Logic of Secrets is complete with respect to the semantics defined above. To do so, we first describe the construction of a protocol called $\mathcal{P}_0$, which is implicitly parameterized by a hypergraph and a set of formulas.

8.1 Protocol $\mathcal{P}_0$

Throughout this section, we will assume that $H = \langle V, E \rangle$ is a fixed hypergraph, and $X \subseteq \Phi(H)$ is a fixed set of formulas.

Definition 8. *For any $A \subseteq E$, we define A^* to be the set of all edges $c \in E$ such that $X \vdash_H A \rhd c$.*

Theorem 5. *$A \subseteq A^*$, for any $A \subseteq E$.*

Proof. Let $a \in A$. By the Reflexivity axiom, $\vdash_H A \rhd a$. Hence, $a \in A^*$. □

Theorem 6. *$X \vdash_H A \rhd A^*$, for any $A \subseteq E$.*

Proof. Let $A^* = \{a_1, \ldots, a_n\}$. By the definition of A^*, $X \vdash_H A \rhd a_i$, for any $i \leq n$. We will prove, by induction on k, that $X \vdash_H (A \rhd a_1, \ldots, a_k)$ for any $0 \leq k \leq n$.
Base Case: $X \vdash_H A \rhd \varnothing$ by the Reflexivity axiom.
Induction Step: Assume that $X \vdash_H (A \rhd a_1, \ldots, a_k)$. By the Augmentation axiom,

$$X \vdash_H A, a_{k+1} \rhd a_1, \ldots, a_k, a_{k+1}. \tag{1}$$

Recall that $X \vdash_H A \rhd a_{k+1}$. Again by the Augmentation axiom, $X \vdash_H (A \rhd A, a_{k+1})$. Hence, $X \vdash_H (A \rhd a_1, \ldots, a_k, a_{k+1})$, by (1) and the Transitivity axiom. □

We now proceed to define our protocol $\mathcal{P}_0$. We will first specify the set of values $Val(c)$ for each edge $c \in E$. In this construction, the value of each edge c on a particular run will be a function from the set 2^E into the set $\{0, 1\}$. Thus, for any $c \in E$ and any $F \subseteq E$, we have $r(c)(F) \in \{0, 1\}$. We will find it more convenient, however, to think about r as a two-argument boolean function: $r(c, F) \in \{0, 1\}$.

Furthermore, we will not allow the value of a edge on a particular run to be just *any* function from the set 2^E into $\{0, 1\}$. Instead, for any edge c, we will restrict set $Val(c)$ so that, for any run r, if $c \in F^*$, then $r(c, F) = 0$.

To complete the description of protocol $\mathcal{P}_0$, we will specify the local conditions for each vertex in the hypergraph. At each vertex v, we define the local condition $Loc(v)$ in such away that run $r(c, F)$ satisfies $Loc(v)$ if and only if

$$\forall F \subseteq E \; \forall c, d \in (Inc(v) \setminus F^*) \; (r(c, F) = r(d, F)) \, .$$

That is, when two edges are incident with a vertex v and neither edge is in F^*, the values of the functions assigned to those edges on argument F must match on any given run.

Now that the definition of protocol $\mathcal{P}_0$ is complete, we make the following two claims about its relationship to the given set of formulas X.

Theorem 7. *If $\mathcal{P}_0 \vDash A \rhd B$, then $X \vdash_H A \rhd B$.*

Proof. Assume $\mathcal{P}_0 \vDash A \rhd B$ and consider two specific runs of $\mathcal{P}_0$. The first of these two runs will be the constant run $r_1(c, F) = 0$. The second run is defined as

$$r_2(c, F) = \begin{cases} 1 \text{ if } c \notin A^* \text{ and } F = A, \\ 0 \text{ if } c \in A^* \text{ or } F \neq A. \end{cases} \tag{2}$$

Run r_1 trivially satisfies the local condition at every vertex v. To show that r_2 satisfies the local condition at a vertex v, consider any $F \subseteq E$ and any $c, d \in Inc(v) \setminus F^*$. If $F \neq A$, then $r_2(c, F) = 0 = r_2(d, F)$. If $F = A$, then, since $c, d \in Inc(v) \setminus F^*$, we have $c, d \notin A^*$. Thus, $r_2(c, F) = 1 = r_2(d, F)$. Therefore, r_2 is a run of protocol $\mathcal{P}_0$.

Notice that by Theorem 5, $A \subseteq A^*$. Thus, by equality (2), $r_2(a, F) = 0$ for any $a \in A$ and any $F \subseteq E$. Hence, $r_1(a, F) = 0 = r_2(a, F)$ for any $a \in A$ and $F \subseteq E$. Thus, by the assumption that $\mathcal{P}_0 \vDash A \rhd B$, we have $r_1(b, F) = r_2(b, F)$ for any $b \in B$ and $F \subseteq E$. In particular, $r_1(b, A) = r_2(b, A)$ for any $b \in B$. Since, by definition, $r_1(b, A) = 0$, we get $r_2(b, A) = 0$ for any $b \in B$. By the definition of r_2, this means that $B \subseteq A^*$. By the Reflexivity axiom, $\vdash_H A^* \rhd B$. By Theorem 6 and the Transitivity axiom, $X \vdash_H A \rhd B$. □

Theorem 8. *If $X \vdash_H A \rhd B$, then $\mathcal{P}_0 \vDash A \rhd B$.*

Proof. Assume that $X \vdash_H A \rhd B$, but $\mathcal{P}_0 \nvDash A \rhd B$. Thus, there are runs r_1 and r_2 of $\mathcal{P}_0$ such that $r_1(a, F) = r_2(a, F)$ for any $a \in A$ and any $F \subseteq E$, yet there are $b_0 \in B$ and $F_0 \subseteq E$ such that

$$r_1(b_0, F_0) \neq r_2(b_0, F_0). \tag{3}$$

First, assume that hypergraph H', obtained from H by the removal of all edges in set F_0^*, contains a path π connecting edge b_0 with a edge $a_0 \in A$. This case implicitly assumes that $b_0, a_0 \notin F_0^*$. Let functions f_1 and f_2 on the edges of hypergraph H be defined as $f_1(c) = r_1(c, F_0)$ and $f_2(c) = r_2(c, F_0)$. Due to the local conditions of protocol $\mathcal{P}_0$, all edges along path π must have the same value of function f_1. The same is also true about function f_2. Therefore, $r_1(b_0, F_0) = f_1(b_0) = f_1(a_0) = r_1(a_0, F_0) = r_2(a_0, F_0) = f_2(a_0) = f_2(b_0) = r_2(b_0, F_0)$. This is a contradiction with statement (3).

Next, suppose that there is no path in H' connecting b_0 with a edge in A. Thus, set F_0^* is a gateway between sets A and $\{b_0\}$. By the Gateway axiom,

$$\vdash_H A \rhd b_0 \to F_0^* \rhd b_0. \tag{4}$$

By the Reflexivity axiom, $\vdash_H B \rhd b_0$. Recall the assumption $X \vdash_H A \rhd B$. Thus, by the Transitivity axiom, $X \vdash_H A \rhd b_0$. Taking into account (4), $X \vdash_H F_0^* \rhd b_0$. By Theorem 6, $\vdash F_0 \rhd F_0^*$. Hence, again by Transitivity, $X \vdash_H F_0 \rhd b_0$. Thus, by Definition 8, $b_0 \in F_0^*$. Hence, by the definition of protocol $\mathcal{P}_0$, $r(b_0, F_0)$ has value 0 for any run r. Therefore, $r_1(b_0, F_0) = 0 = r_2(b_0, F_0)$. This is a contradiction with statement (3). □

8.2 Main Result

Now, we are ready to finish the proof of completeness.

Theorem 9. *If $\nvdash_H \phi$, then there is a finite protocol $\mathcal{P}$ such that $\mathcal{P} \not\models \phi$.*

Proof. Assume $\nvdash_H \phi$. Let X be a maximal consistent set of formulas such that $\neg\phi \in X$. Consider the finite protocol $\mathcal{P}_0$ parameterized by hypergraph H and set of formulas X. For any formula ψ, we will show that $X \vdash_H \psi$ if and only if $\mathcal{P}_0 \models \psi$. The proof is by induction on the structural complexity of formula ψ. The base case follows from Theorems 7 and 8. The induction case follows from the maximality and consistency of set X. To finish the proof of the theorem, select ψ to be $\neg\phi$. □

Corollary 1. *Binary relation $\vdash_H \phi$ is decidable.*

Proof. This statement follows from the completeness of the Logic of Secrets with respect to *finite* protocols and the recursive enumerability of all theorems in the logic. □

9 Conclusion

We have presented a complete axiomatization of the properties of the functional dependence relation over secrets on hypergraphs. In light of previous results capturing properties of the independence relation in the same setting [4], it would be interesting to describe properties that connect these two predicates on hypergraphs.

An example of such a property for the hypergraph H_6 in Figure 6 is given in the following theorem.

Fig. 6. Hypergraph H_6.

Theorem 10. *For any protocol $\mathcal{P}$ over hypergraph H_6,*

$$\mathcal{P} \models (a, b \rhd c) \wedge [a, b] \rightarrow b \rhd c.$$

Proof. For any two runs $r_1, r_2 \in \mathcal{R}(\mathcal{P})$ where $r_1(b) = r_2(b)$, we must show that $r_1(c) = r_2(c)$. The assumption $[a, b]$ guarantees that values $r_1(a)$ and $r_2(b)$ coexist in some run in $\mathcal{R}(\mathcal{P})$; call this run r_3. Thus, we have $r_3(a) = r_1(a)$ and $r_3(b) = r_2(b)$.

Next, we create a new function r_4 which "glues" together runs r_3 and r_2 at vertex q. Formally, we define r_4 as

$$r_4(x) = \begin{cases} r_3(x) & \text{if } x = a, \\ r_2(x) & \text{if } x \in \{b, c\}. \end{cases}$$

We claim that function r_4 satisfies the local conditions of protocol $\mathcal{P}$, since at each vertex in H_5, it behaves locally like an existing run. Indeed, at vertex p, r_4 matches run r_3, and at parties r and s, r_4 matches run r_2. At vertex q, r_4 matches r_2 exactly, since $r_4(b) = r_2(b)$. Thus, $r_4 \in \mathcal{R}(\mathcal{P})$. To complete the proof, we note that $r_1(a) = r_3(a) = r_4(a)$ and $r_1(b) = r_2(b) = r_4(b)$. By the assumption that $(a, b \rhd c)$, we have $r_1(c) = r_4(c)$. The definition of r_4 is such that $r_4(c) = r_2(c)$, so $r_1(c) = r_2(c)$, as desired. □

A complete axiomatization of properties that connect the functional dependence relation and the independence relation between secrets on a hypergraph remains an open problem.

10 Acknowledgment

The authors would like to thank Andrea Mills and Benjamin Sapp for discussions of the functional dependence relation on sets of secrets during earlier stages of this work.

References

1. Sutherland, D.: A model of information. In: Proceedings of Ninth National Computer Security Conference. (1986) 175–183
2. Halpern, J.Y., O'Neill, K.R.: Secrecy in multiagent systems. ACM Trans. Inf. Syst. Secur. **12**(1) (2008) 1–47
3. 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
4. Miner More, S., Naumov, P.: Hypergraphs of multiparty secrets. In: 11th International Workshop on Computational Logic in Multi-Agent Systems CLIMA XI (Lisbon, Portugal), LNAI 6245, Springer (2010) 15–32
5. Armstrong, W.W.: Dependency structures of data base relationships. In: Information processing 74 (Proc. IFIP Congress, Stockholm, 1974). North-Holland, Amsterdam (1974) 580–583
6. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Second edn. Prentice-Hall (2009)

7. 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
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. Shamir, A.: How to share a secret. Communications of the Association for Computing Machinery **22**(11) (November 1979) 612–613
10. 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