Information Flow under Budget Constraints

Pavel Naumov[1] and Jia Tao[2]

[1] Vassar College, Poughkeepsie NY, United States
[2] The College of New Jersey, Ewing NJ, United States

Abstract. Although first proposed in the database theory as properties of functional dependencies between attributes, Armstrong's axioms capture general principles of information flow by describing properties of dependencies between sets of pieces of information. This paper generalizes Armstrong's axioms to a setting in which there is a cost associated with information. The proposed logical system captures general principles of dependencies between pieces of information constrained by a given budget.

1 Introduction

1.1 Functional Dependency

Armstrong [4] introduced a system of three axioms describing the properties of functional dependencies between sets of attributes in a database. The applicability of these axioms goes far beyond the domain of databases. They capture the properties of functional dependency between any two sets of pieces of information. To describe this setting informally, one can think of an agent that has knowledge of some of the pieces of information and is interested in uncovering some other pieces. For example, knowing a cyphertext c and the decryption key k, one can determine the plain text message m. We write this as $c, k \rhd m$. Yet, one cannot determine the original message from the cyphertext alone without the encryption key, and thus, $\neg(c \rhd m)$. Keeping the intended epistemic interpretation in mind, we refer to the pieces of information as *secrets*.

The property $c, k \rhd m$ is valid when secrets c, k, and m are a cyphertext, a decryption key, and the corresponding plain text message. However, it may not be valid under some other interpretation of these secrets. Armstrong's axioms capture the most general properties of functional dependencies that are valid in all settings. These axioms are:

(A1) *Reflexivity*: $A \rhd B$, if $B \subseteq A$,
(A2) *Augmentation*: $A \rhd B \to A, C \rhd B, C$,
(A3) *Transitivity*: $A \rhd B \to (B \rhd C \to A \rhd C)$,

where A, B denotes the union of sets of secrets A and B, and $\varphi \to \psi$ denotes the logical implication. Armstrong [4] proved the soundness and the completeness of this logical system with respect to a database semantics.

The above axioms became known in database literature as Armstrong's axioms, see Garcia-Molina, Ullman, and Widom [8, p. 81]. Beeri, Fagin, and Howard [5] suggested a variation of Armstrong's axioms that describes properties of multi-valued dependency. Hartmann, Link, and Schewe [10] investigated a "weak" version of functional dependency. Väänänen [18] proposed a first order version of these principles. Naumov and Nicholls [14] developed a similar set of axioms for what they called the *rationally* functional dependency.

1.2 Approximate Dependency

There have been two different approaches to extending Armstrong's axioms to handle approximate reasoning. Bělohlávek and Vychodil [6] described a complete logical system that formally captures the relation *approximate values of secrets in set A functionally determine approximate values of secrets in set B*. In his upcoming work [19], Väänänen considered the relation *secrets in set A determine secrets in set B with exception of p fraction of possible combinations of values of all secrets*. We denote this relation by $A \rhd_p B$. For example, $A \rhd_{0.05} B$ means that secrets in set A determine secrets in set B in all but 5% of the possible combinations. Väänänen [19] proposed a complete axiomatic system for this relation, consisting of the following principles for all real numbers $p, q \in [0, 1]$:

1. Reflexivity: $A \rhd_0 B$, where $B \subseteq A$,
2. Totality: $A \rhd_1 B$,
3. Weakening: $A \rhd_p C, D \to A, B \rhd_p C$,
4. Augmentation: $A \rhd_p B \to A, C \rhd_p B, C$,
5. Transitivity: $A \rhd_p B \to (B \rhd_q C \to A \rhd_{p+q} C)$, where $p + q \leq 1$,
6. Monotonicity: $A \rhd_p B \to A \rhd_q B$, where $p \leq q$.

Note that Väänänen's relation $A \rhd_p B$, when $p = 0$, is exactly the original Armstrong's functional dependency relation. In the case of an arbitrary p, relation $A \rhd_p B$ could be considered as a "weaker" form of functional dependency, which might hold even in the cases where the functional dependency does not hold.

1.3 Budget-Constrained Dependency

In this paper we propose another interpretation of atomic predicate $A \rhd_p B$ that we call *the budget-constrained dependency*. Just like Väänänen's approximate dependency, the budget-constrained dependency is a weaker form of the original Armstrong's functional dependency relation. Intuitively, $A \rhd_p B$ means that *an agent who already knows secrets in set A can recover secrets in set B at cost no more than p*. More formally, we assume that a non-negative cost is assigned to each secret and that $A \rhd_p B$ means that there is a way to add several secrets with the total cost no more than p to set A in such a way that the extended set of secrets functionally determines all secrets in set B.

One example of such a setting is fees associated with information access: criminal background check fees, court records obtaining fees, etc. Another example is geological explorations, where learning about deposits of mineral resources often requires costly drilling. Although it is convenient to think about a budget constraint as a financial one, a budget constraint can also refer to a limit on time, space, or some other resource.

In this paper we introduce a sound and complete logical system for the budget-constrained dependency which is based on the following three principles that generalize Armstrong's axioms:

1. *Reflexivity*: $A \rhd_p B$, if $B \subseteq A$,
2. *Augmentation*: $A \rhd_p B \to A, C \rhd_p B, C$,
3. *Transitivity*: $A \rhd_p B \to (B \rhd_q C \to A \rhd_{p+q} C)$.

1.4 Functional vs. Budget-Constrained Dependencies

Armstrong's axioms of functional dependency as well as our axioms of budget-constrained functional dependency can be formulated into two different ways using languages with different expressive power.

One approach is to allow only statements in our language that have the form $A \rhd_p B$ and not allow Boolean combinations of such statements. In this case, Armstrong's axioms should be stated as inference rules that allow to derive statements of the form $A \rhd_p B$ from other statements of the same form.

The other approach is to include Boolean connectives into the language. In this case, statements of the form $A \rhd_p B$ become atomic statements in the language. Then, Armstrong's axioms can be stated as actual axioms that would be used in the logical system along with the propositional tautologies and Modus Ponens inference rule.

The second approach clearly yields a more expressive language. While the proofs of the completeness of original Armstrong's axioms of functional dependency are surprisingly similar for these two cases, the situation is different when it comes to budget-constrained dependency. The more expressive language requires a significantly more sophisticated argument to prove the completeness theorem. In this paper we only consider the more expressive language. In the rest of this section we look at several examples to compare challenges raised by the proofs of the completeness for Armstrong's functional dependency and our budget-constrained dependency.

As the first example, consider the formula $a \rhd b \to b \rhd a$ in the language without budget constraints. To construct a counterexample for this formula we need to describe a model in which secret a functionally determines secret b but not vice versa. Informally, to construct this model, imagine a and b to be two paper folders. Let folder a contain copies of two different (and unrelated to each other) documents: X and Y, and let folder b contain only a copy of document Y. In this case, an agent can recover the content of folder b based on folder a but not vice versa.

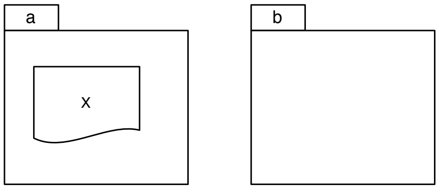

Fig. 1. Formula $a \rhd b$ is true, but formula $b \rhd a$ is false.

There is even a simpler counterexample for formula $a \rhd b \to b \rhd a$. Namely, consider a model in which folder a stores a copy of document X and folder b is empty, see Figure 1. In this model, based on the content of folder a one can vacuously recover the content of empty folder b. At the same time, based on the content of empty folder b one cannot recover the content of folder a. Thus, in this model formula $a \rhd b \to b \rhd a$ is false.

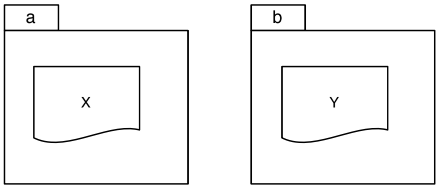

Fig. 2. Formulas $a \rhd b$ and $b \rhd a$ are both false.

Now consider formula $a \rhd b \vee b \rhd a$. To construct its counterexample, one can consider a model in which folders a and b containing copies of two different (and unrelated to each other) documents X and Y respectively, see Figure 2.

To construct counterexamples for more complicated formulas, one can consider models with multiple folders containing copies of multiple documents. An example of such a model is depicted in Figure 3. In this model $a, b \rhd c$ is true because anyone with access to folders a and b knows the content of folder c. The folder/document model informally described here is sufficiently general to create a counterexample for each formula unprovable from Armstrong's axioms.

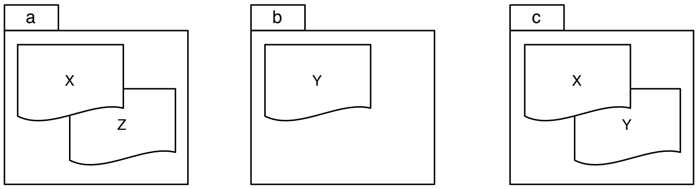

Fig. 3. Formula $a, b \rhd c$ is true.

In fact, the original Armstrong's proof of the completeness for his rule-based system and the proof of the completeness for the corresponding axiom-based system [11] could be viewed as formalizations of this folder/document construction.

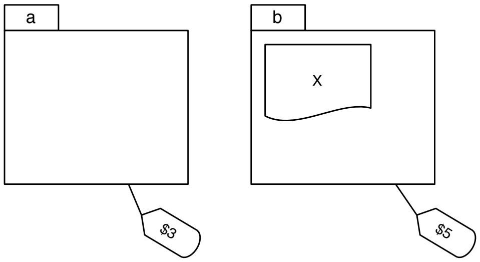

Fig. 4. Formula $a \rhd_4 b$ is false.

The situation becomes significantly more complicated once the cost of information is added to the language. Let us start with a very simple example. If we want to construct a counterexample for formula $a \rhd_4 b$, then we can consider a model depicted in Figure 4 with two folders: a and b, priced at \$3 and \$5, respectively. The first folder is empty and the second contains a copy of the document X. It is clear that in this model anyone who knows the content of folder a still needs to spend \$5 to learn the content of folder b. Thus, budget-constrained dependency $a \rhd_p b$ is not satisfied in this model for each $p < 5$.

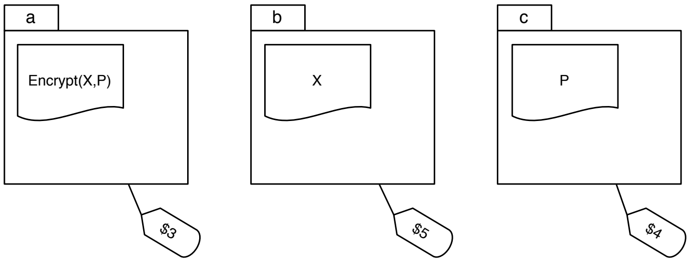

Fig. 5. Formula $a \rhd_4 b \to \varnothing \rhd_4 b$ is false.

Let us now consider a more interesting example. Suppose that we want to construct a counterexample for the formula $a \rhd_4 b \to \varnothing \rhd_4 b$. That is, we want to construct a model where anyone who knows the content of folder a can reconstruct the content of folder b after spending at most \$4. Yet, the same cannot be done without access to folder a. To construct such a model we use the cryptographic tool called one-time encryption pad[3]. Our model consists of three folders a, b, and c priced at \$3, \$5, and \$4, respectively, see Figure 5. Let folder b contain a copy of a document X, folder c contain an encryption pad P, and folder a contain the encrypted version of the document. In this model, $\varnothing \rhd_4 b$ is false because \$4 buys either access to the encryption pad in folder c or access to the encrypted text in folder a, but not both. However, formula $a \rhd_4 b$ is true in

[3] The one-time encryption pad is not the only way to construct a counterexample for formula $a \rhd_4 b \to \varnothing \rhd_4 b$. We introduce one-time pads to prepare readers for the general proof of the completeness presented later in this paper.

the same model because anyone who knows encrypted text $Encrypt(X, P)$ can spend $4 on pad P, decode message X, and thus, learn the content of folder b.

The one-time pad encryption is known in cryptography as a symmetric-key algorithm because the same key (i.e. the one-time pad) could be used to encrypt and to decrypt the text. As a result, in the model depicted in Figure 5, not only formula $a \rhd_4 b$ is true, but formula $b \rhd_4 a$ is true as well.

For the next example, we construct a counterexample for formula

$$a \rhd_4 b \rightarrow (\varnothing \rhd_4 b \vee b \rhd_4 a).$$

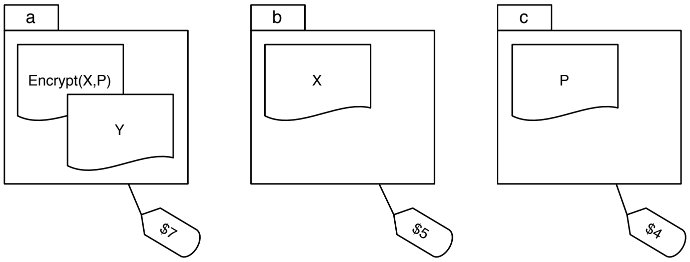

Fig. 6. Formula $a \rhd_4 b \rightarrow (\varnothing \rhd_4 b \vee b \rhd_4 a)$ is false.

This is an easier task than one might think because one just needs to modify the previous model by adding to the folder a some extra document not related to the document X and to raise the price of this folder, see Figure 6. This guarantees that the only way to learn all the content of folder a is to buy folder a directly.

The situation becomes much more complicated if we want (i) the value of secret a to be recoverable from the value of secret b and (ii) the value of secret b to be recoverable from the value of secret a, but at a different price. For instance, if we want to construct a counterexample for the following formula:

$$a \rhd_1 b \wedge b \rhd_5 a \rightarrow (\varnothing \rhd_5 a \vee \varnothing \rhd_1 b \vee b \rhd_4 a). \tag{1}$$

At first glance, this goal could be achieved using asymmetric key cryptography, commonly used in the public-key encryption. For instance, suppose that folder a contains a document X and folder b contains the same document encrypted with an encryption key k_e, see Figure 7. To obtain the content of folder b based on the content of folder a, one only needs to know the encryption key k_e. To restore the content of folder a based on folder b one needs to know the value of the decryption[4] key k_d. If the encryption key and the decryption key are priced at $1 and $5 respectively, the formula $b \rhd_4 a$ is not satisfied from the cryptographic point of view. Since folders a and b are priced in this model at

[4] In public-key cryptography, an encryption key is known as the public key and a decryption key as the private key. We do not use these terms here because in our setting neither of the keys is public in the sense that both of them have associated costs.

\$100 each, formulas $\varnothing \rhd_5 a$ and $\varnothing \rhd_1 b$ are not satisfied either. Thus, the entire formula (1) is not satisfied from the cryptographic point of view.

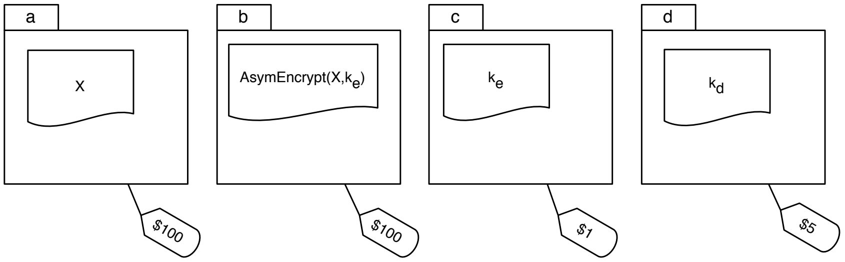

Fig. 7. Formula $a \rhd_1 b \wedge b \rhd_5 a \rightarrow (\varnothing \rhd_5 a \vee \varnothing \rhd_1 b \vee b \rhd_4 a)$ is false.

Note, however, that cryptographic asymmetric-key algorithms are only polynomial time secure and the proof of polynomial time security requires an appropriate computational hardness assumption [13, Ch. 2]. In other words, in public-key cryptography, the encrypted text can be decrypted using only the public encryption key if one has exponential time for the decryption. Neither Armstrong's [4] definition of functional dependency nor our definition of budget-constrained functional dependency, given in Definition 6 below, assumes any upper bound on the computability of the functional dependency. From our point of view, one would be able to eventually restore the content of folder a based on folder b by spending \$1 on the content of folder c. Thus, in the above setting, without polynomial restriction on computability, not only formula $b \rhd_4 a$ is true, but formula $b \rhd_1 a$ is true as well.

Figure 8 shows a counterexample for statement (1) that uses non-computable functional dependency. Assume that folders a and b contain copies of unrelated documents X and Y, folder c contains an infinite supply of one-time encryption pads $P_1, P_2, P_3, \dots$ and folder d contains another infinite set of one-time encryption pads $Q_1, Q_2, Q_3, \dots$. First, encrypt document Y with one-time pad P_1 and place a copy of the resulting cyphertext $\mathit{Encrypt}(Y, P_1)$ into folder a. Next, encrypt $\mathit{Encrypt}(Y, P_1)$ with pad Q_2 and place a copy of the resulting cyphertext $\mathit{Encrypt}(\mathit{Encrypt}(Y, P_1), Q_2)$ into folder b. Then, use pad P_3 to encrypt $\mathit{Encrypt}(\mathit{Encrypt}(Y, P_1), Q_2)$ and place a copy of the resulting cyphertext $\mathit{Encrypt}(\mathit{Encrypt}(\mathit{Encrypt}(Y, P_1), Q_2), P_3)$ into folder a, and so on ad infinitum. Perform similar steps with the document X, as shown in Figure 8.

To show that the model depicted in Figure 8 is a counterexample for formula (1), we need to prove that both formulas $a \rhd_1 b$ and $b \rhd_5 a$ are satisfied in this model and each of the formulas $\varnothing \rhd_5 a$, $\varnothing \rhd_1 b$, and $b \rhd_4 a$ is not satisfied. First, notice that formula $a \rhd_1 b$ is satisfied because folder a contains all documents in folder b encrypted with one-time pads $P_1, P_2, \dots$ and that all these pads could be acquired for \$1 by buying folder c. Second, formula $b \rhd_5 a$ is satisfied for a similar reason using pads $Q_1, Q_2, \dots$. Third, formula $\varnothing \rhd_5 a$ is not satisfied because for \$5 one can only buy either folder c or folder d, both containing only one-time pads. In the absence of folder b, one-time encryption pads can not be used to

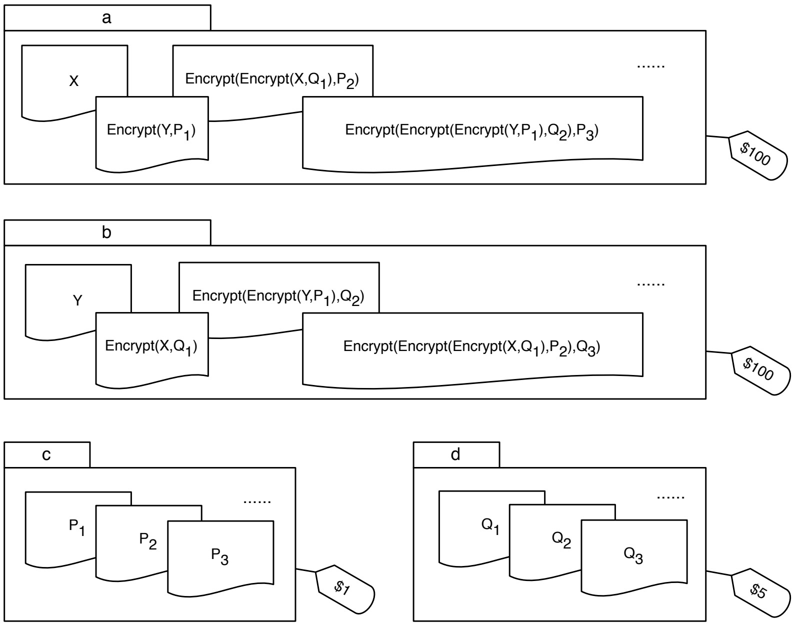
a
X
Encrypt(Encrypt(X,Q_1),P_2)
......
Encrypt(Y,P_1)
Encrypt(Encrypt(Encrypt(Y,P_1),Q_2),P_3)
$100
b
Y
Encrypt(Encrypt(Y,P_1),Q_2)
......
Encrypt(X,Q_1)
Encrypt(Encrypt(Encrypt(X,Q_1),P_2),Q_3)
$100
c
P_1
P_2
P_3
......
$1
d
Q_1
Q_2
Q_3
......
$5

Fig. 8. Formula $a \rhd_1 b \wedge b \rhd_5 a \to (\varnothing \rhd_5 a \vee \varnothing \rhd_1 b \vee b \rhd_4 a)$ is false.

recover document X stored in folder a. Formula $\varnothing \rhd_1 b$ is not satisfied for a similar reason. Finally, $b \rhd_4 a$ is not satisfied because \$4 is not enough to buy the content of folder d. This amount of money can only be used to buy pads $P_1, P_2, \ldots$ in folder c. Knowing the content of folder b and one-time pads $P_1, P_2, \ldots$, one can not recover document X contained in folder a. The counterexample described above produces non-computable functional dependency because the number of folders is infinite.

In the full version [15] of this paper we prove the completeness of our logical system. At the core of this proof is a generalized version of the construction presented in Figure 8.

As we have seen in the examples above, the complexity of constructing counterexamples arises from the formulas that contain disjunctions in the conclusions. Such formulas cannot be expressed in the logical systems that have no Boolean connectives and express Armstrong's axioms as inference rules. Although we did not work out the details, we believe that, in the case of such a less expressive logical system, the proof of the completeness presented in the full version [15] of this paper could be significantly simplified.

1.5 Related Literature

The axiomatic system proposed in this paper is related to other logical systems for reasoning about bounded resources. The classical logical system for reasoning about resources is the linear logic of Girard [9]. Alechina and Logan [1] presented a family of logical systems for reasoning about beliefs of a perfect reasoner that can only derive consequences of her beliefs after some time delay. This approach has been further developed into the multi-agent Timed Reasoning Logic in [3]. Bulling and Farwer [7] proposed Resource-Bounded Tree Logics for reasoning about resource-bounded computations and obtained preliminary results on the complexity and decidability of model checking for these logics. Alechina, Logan, Nga, and Rakib [2] incorporated resource requirements into Coalition logic and gave a sound and complete axiomatization of the resulting system. Another logical system for reasoning about knowledge under bounded resources was proposed by Jamroga and Tabatabaei [12]. Their paper focuses on the expressive power of the language of the system and the model checking algorithm. Naumov and Tao introduced sound and complete modal logics for reasoning about budget-constrained knowledge [16] and cost of privacy [17]. Unlike our current system all of the above logics do not provide a language for expressing functional dependencies.

1.6 Outline

The rest of the paper is organized as follows. In Section 2 we formally define the language of our logical system and its informational semantics. In Section 3 we list the axioms of the system that have already been discussed in the introduction. In Section 4 we give several examples of formal proofs in our logical system. In Section 5 we prove the soundness of our axioms with respect to the informational semantics. Section 6 states the completeness theorem. The proof of the completeness theorem can be found in the full version of this paper [15].

2 Syntax and Semantics

In this section we introduce the language of our system and formally describe its intended semantics that we call *informational semantics*.

Definition 1. *For any set of "secrets" $\mathcal{S}$, let language $\Phi(\mathcal{S})$ be the minimum set of formulas such that*

1. $A \rhd_p B \in \Phi(\mathcal{S})$ *for all finite sets* $A, B \subseteq \mathcal{S}$ *and all real numbers* $p \geq 0$,
2. *if* $\varphi \in \Phi(\mathcal{S})$, *then* $\neg\varphi \in \Phi(\mathcal{S})$,
3. *if* $\varphi, \psi \in \Phi(\mathcal{S})$, *then* $\varphi \to \psi \in \Phi(\mathcal{S})$.

Next, we introduce the formal informational semantics of our logical system. The only significant difference between our semantics and the one used by Armstrong [4] is the costs function $\|\cdot\|$ that assigns a non-negative cost to each secret. Note that we assume that the cost is assigned to a secret, not to its value. For example, if we assign a certain cost to a folder with documents, then this cost is uniform and does not depend on the content of the documents in this folder.

Definition 2. *An informational model is a tuple* $\langle \mathcal{S}, \{D_a\}_{a\in\mathcal{S}}, \|\cdot\|, \mathcal{L}\rangle$, *where*

1. $\mathcal{S}$ *is an arbitrary set of "secrets",*

2. *D_a is a set representing the domain of secret $a \in \mathcal{S}$,*
3. *$\|\cdot\|$ is a cost function that maps each secret $a \in \mathcal{S}$ into a non-negative real number or infinity $+\infty$,*
4. *$\mathcal{L} \subseteq \prod_{a\in\mathcal{S}} D_a$ is the set of vectors of values of secrets that satisfy the constraints imposed by the informational model.*

Note that we allow infinite attribute costs in our semantics captured in Definition 2 to include the possibility of attributes that cannot be bought. For the same reason, we do *not* allow infinite costs in our syntax given in Definition 1. In the example depicted in Figure 8, folders are secrets and the information stored in the documents contained in a folder is a value of such a secret. The set of all possible values of a secret is its domain. The cost of different secrets is specified explicitly in Figure 8. Note that there is a certain dependency between the plaintext, one-time encryption pads, and the cyphertext. In other words, not all combinations of values of different secrets are possible. The set $\mathcal{L}$ is the set of all possible combinations of these values.

We allow the cost $\|a\|$ of a secret a to be infinity. Informally, one can interpret this as secret a not being available for purchase at any cost. If all secrets are available for sale, then we say that the informational model is *finite cost*.

As an example, the setting described in Figure 4 could be formally captured in informational model $I_0 = \langle \{a, b\}, \{D_x\}_{x\in\{a,b\}}, \|\cdot\|, \mathcal{L}\rangle$, where the domain (range of values) D_a of the empty folder a is a single element set: $\{null\}$, the domain D_b of secret b is the set of all, say binary, strings $\{0, 1\}^*$, cost $\|a\|$ of secret a is 3, cost $\|b\|$ of secret b is 5, and set of vectors $\mathcal{L}$ is the set of all pairs in set $D_a \times D_b = \{\langle null, s\rangle \mid s \in \{0, 1\}^*\}$.

Definition 3. *Informational model $\langle \mathcal{S}, \{D_a\}_{a\in\mathcal{S}}, \|\cdot\|, \mathcal{L}\rangle$ is finite cost if $\|a\| < +\infty$ for each $a \in \mathcal{S}$.*

For example, informational model I_0 for the setting described in Figure 4 is a finite cost model, because $\|a\| = 3 < \infty$ and $\|b\| = 5 < \infty$.

Definition 4. *For any vector $\ell_1 = \langle f^1_a\rangle_{a\in\mathcal{S}} \in \mathcal{L}$, any vector $\ell_2 = \langle f^2_a\rangle_{a\in\mathcal{S}} \in \mathcal{L}$, and any set $A \subseteq \mathcal{S}$, let $\ell_1 =_A \ell_2$ if $f^1_a = f^2_a$ for each secret $a \in A$.*

For example, $\langle null, 01\rangle =_{\{a\}} \langle null, 1011\rangle$ and $\langle null, 01\rangle \neq_{\{a,b\}} \langle null, 1011\rangle$ for model I_0.

Definition 5. *For each finite set $A \subseteq \mathcal{S}$, let $\|A\| = \sum_{a\in A} \|a\|$.*

Thus, $\|\{a, b\}\| = \|a\| + \|b\| = 3 + 5 = 8$ for model I_0.

The next definition is the key definition of this section. It specifies the formal semantics of our logical system. Item 1. of this definition provides the exact meaning of the budget-constrained dependency. In this definition and throughout the rest of the paper, by A, B we denote the union of sets A and B.

Definition 6. *For each informational model $I = \langle \mathcal{S}, \{D_a\}_{a\in\mathcal{S}}, \|\cdot\|, \mathcal{L}\rangle$ and each formula $\varphi \in \Phi(\mathcal{S})$, the satisfiability relation $I \vDash \varphi$ is defined as follows:*

1. *$I \vDash A \rhd_p B$ when there is a finite set $C \subseteq \mathcal{S}$ such that $\|C\| \le p$ and for each pair of vectors $\ell_1, \ell_2 \in \mathcal{L}$, if $\ell_1 =_{A,C} \ell_2$, then $\ell_1 =_B \ell_2$,*
2. *$I \vDash \neg\psi$ if $I \nvDash \psi$,*
3. *$I \vDash \psi \to \chi$ if $I \nvDash \psi$ or $I \vDash \chi$.*

Then, $I_0 \Vdash \varnothing \rhd_0 a$ because $\ell_1 =_a \ell_2$ for any two vectors $\ell_1, \ell_2 \in \{nill\} \times \{0,1\}^*$.

3 Axioms

For any set of secrets $\mathcal{S}$, our logical system, in addition to the propositional tautologies in language $\Phi(\mathcal{S})$ and the Modus Ponens inference rule, contains the following axioms:

1. Reflexivity: $A \rhd_p B$, where $B \subseteq A$,
2. Augmentation: $A \rhd_p B \to A, C \rhd_p B, C$,
3. Transitivity: $A \rhd_p B \to (B \rhd_q C \to A \rhd_{p+q} C)$.

We write $\vdash \varphi$ if formula φ is derivable in our system. Also, we write $X \vdash \varphi$ if formula φ is derivable in our system extended by the set of additional axioms X.

4 Examples of Proofs

We prove the soundness of our logical system in the next section. Here we provide several examples of formal proofs in this system. We start by showing that Väänänen's Weakening and Monotonicity axioms [19] are derivable in our system.

Proposition 1 (Weakening). *$\vdash A \rhd_p C, D \to A, B \rhd_p C$.*

Proof. By Augmentation axiom,

$$\vdash A \rhd_p C, D \to A, B \rhd_p B, C, D. \tag{2}$$

By Reflexivity axiom,

$$\vdash B, C, D \rhd_0 C. \tag{3}$$

By Transitivity axiom,

$$A, B \rhd_p B, C, D \to (B, C, D \rhd_0 C \to A, B \rhd_p C). \tag{4}$$

Finally, from (2), (3), and (4), by the laws of propositional logic,

$$\vdash A \rhd_p C, D \to A, B \rhd_p C.$$

□

Proposition 2 (Monotonicity). *$\vdash A \rhd_p B \to A \rhd_q B$, where $p \le q$.*

Proof. By Reflexivity axiom,

$$\vdash B \rhd_{q-p} B. \tag{5}$$

By Transitivity axiom,

$$\vdash A \rhd_p B \to (B \rhd_{q-p} B \to A \rhd_q B). \tag{6}$$

Finally, from (5) and (6), by the laws of propositional logic,

$$\vdash A \rhd_p B \to A \rhd_q B.$$

□

As our last example, we prove a generalized version of Augmentation axiom.

Proposition 3. $\vdash A \rhd_p B \to (C \rhd_q D \to A, C \rhd_{p+q} B, D)$.

Proof. By Augmentation axiom,

$$\vdash A \rhd_p B \to A, C \rhd_p B, C \tag{7}$$

and

$$\vdash C \rhd_q D \to B, C \rhd_q B, D. \tag{8}$$

At the same time, by Transitivity axiom,

$$\vdash A, C \rhd_p B, C \to (B, C \rhd_q B, D \to A, C \rhd_{p+q} B, D). \tag{9}$$

Finally, from (7), (8), and (9), by the laws of propositional logic,

$$\vdash A \rhd_p B \to (C \rhd_q D \to A, C \rhd_{p+q} B, D).$$

□

5 Soundness

In this section we prove the soundness of our logical system.

Theorem 1. *If $\varphi \in \Phi(\mathcal{S})$ and $\vdash \varphi$, then $I \vDash \varphi$ for each informational model $I = \langle \mathcal{S}, \{D_a\}_{a \in \mathcal{S}}, \|\cdot\|, \mathcal{L} \rangle$.*

The soundness of propositional tautologies and the Modus Ponens inference rule follows from Definition 6 in the standard way. Below we prove the soundness of the remaining axioms as separate lemmas.

Lemma 1. *For all finite sets $A, B \subseteq \mathcal{S}$, if $B \subseteq A$, then $I \vDash A \rhd_p B$.*

Proof. Let $C = \varnothing$. Thus, $\|C\| = \|\varnothing\| = 0 \leq p$. Consider any two vectors $\ell_1, \ell_2 \in \mathcal{L}$ such that $\ell_1 =_{A,C} \ell_2$. It suffices to show that $\ell_1 =_B \ell_2$, which is true due to Definition 4 and the assumption $B \subseteq A$. □

Lemma 2. *For all finite sets $A, B, C \subseteq \mathcal{S}$, if $I \vDash A \rhd_p B$, then $I \vDash A, C \rhd_p B, C$.*

Proof. By Definition 6, assumption $I \vDash A \rhd_p B$ implies that there is a set $D \subseteq \mathcal{S}$ such that (i) $\|D\| \le p$ and (ii) for each $\ell_1, \ell_2 \in \mathcal{L}$, if $\ell_1 =_{A,D} \ell_2$, then $\ell_1 =_B \ell_2$.

Consider now $\ell_1, \ell_2 \in \mathcal{L}$ such that $\ell_1 =_{A,C,D} \ell_2$. It suffices to show that $\ell_1 =_{B,C} \ell_2$. Note that assumption $\ell_1 =_{A,C,D} \ell_2$ implies that $\ell_1 =_{A,D} \ell_2$ and $\ell_1 =_C \ell_2$ by Definition 4. Due to condition (ii) above, the former implies that $\ell_1 =_B \ell_2$. Finally, statements $\ell_1 =_B \ell_2$ and $\ell_1 =_C \ell_2$ together imply that $\ell_1 =_{B,C} \ell_2$. □

Lemma 3. *For all finite sets $A, B, C \subseteq \mathcal{S}$, if $I \vDash A \rhd_p B$ and $I \vDash B \rhd_q C$, then $I \vDash A \rhd_{p+q} C$.*

Proof. By Definition 6, assumption $I \vDash A \rhd_p B$ implies that there is $D_1 \subseteq \mathcal{S}$ such that (i) $\|D_1\| \le p$ and (ii) for each $\ell_1, \ell_2 \in \mathcal{L}$, if $\ell_1 =_{A,D_1} \ell_2$, then $\ell_1 =_B \ell_2$.

Similarly, assumption $I \vDash B \rhd_q C$ implies that there is $D_2 \subseteq \mathcal{S}$ such that (iii) $\|D_2\| \le q$ and (iv) for each $\ell_1, \ell_2 \in \mathcal{L}$, if $\ell_1 =_{B,D_2} \ell_2$, then $\ell_1 =_C \ell_2$.

Let $D = D_1, D_2$. By Definition 5, $\|D\| \le \|D_1\| + \|D_2\|$. Taking into account statements (i) and (iii) above, we conclude that $\|D\| \le p + q$. Consider any two vectors $\ell_1, \ell_2 \in \mathcal{L}$ such that $\ell_1 =_{A,D} \ell_2$. It suffices to show that $\ell_1 =_C \ell_2$. Indeed, by Definition 4, assumption $\ell_1 =_{A,D} \ell_2$ implies that $\ell_1 =_{A,D_1} \ell_2$. Hence, $\ell_1 =_B \ell_2$ due to condition (ii). At the same time, assumption $\ell_1 =_{A,D} \ell_2$ also implies that $\ell_1 =_{D_2} \ell_2$ by Definition 4. Thus, $\ell_1 =_{B,D_2} \ell_2$ by Definition 4. Therefore, $\ell_1 =_C \ell_2$ due to condition (iv). □

This concludes the proof of Theorem 1.

6 On the Completeness Theorem

The main result of this paper is a completeness theorem for our logical system with respect to the informational semantics. The completeness could be stated in different non-equivalent forms that we discuss and compare in this section.

Informally, a completeness theorem states that if a formula φ is not provable in our system, then there is an informational model I such that $I \nvDash \varphi$. To state the theorem formally, we need to decide if model I must use only secrets explicitly mentioned in formula φ or a set of secrets of model I could be a superset of the set of secrets used in formula φ.

This distinction applies not only to our system, but to other logical systems as well. For example, formulas in first order logic can have constants. When we prove the completeness of the first order logic, we allow universes that have more elements than the number of constants. This is significant because, for example, formula

$$\forall x(c_1 = c_2 \vee x = c_1 \vee x = c_2) \tag{10}$$

is not provable in the first order logic, but it is true in any model with a universe consisting of only elements that are interpretations of c_1 and c_2. Thus, to construct a counterexample for this formula one needs to consider first order models with more than two elements in the universe.

The situation with our logical system is similar. To prove the completeness of the system we often need to introduce additional secrets not explicitly mentioned in the formula. An analog of formula (10) is, for example, formula

$$\neg(a \rhd_1 b) \to (\neg(b \rhd_1 a) \to \neg(\varnothing \rhd_2 a, b)). \tag{11}$$

This formula is true in any informational model that has only two secrets explicitly mentioned in the formula: secret a and secret b. Indeed, the assumption $\neg(a \rhd_1 b)$ implies that costs of secret b is more than 1. Similarly, assumption $\neg(b \rhd_1 a)$ implies that costs of secret a is more than 1. So, given a budget of only 2, one can buy at most one of secrets a and b. Without loss of generality, assume that secret a is bought. After that purchase, the amount left is less than 1. Per assumption $\neg(a \rhd_1 b)$, the value of b is not attainable on this budget.

At the same time, formula (11) is not true in the information model that has three secrets: a, b, and c, all priced at 1.5, where values of a and b are unrelated and c is pair $\langle a, b\rangle$. One can think about this example as a formalization of "buy one, get one free" marketing.

In this paper we study the most general logical principles of budget-constrained dependency. Thus, we do not include principles like formula (11), that are true only for a specific set of secrets. In other words, when constructing a counterexample for the completeness theorem, we allow additional secrets that are not explicitly mentioned in the original formula. The completeness theorem is stated below.

Theorem 2. *For each formula $\varphi \in \Phi(\mathcal{S})$, if $\nvdash \varphi$, then there is a finite informational model $I = \langle \mathcal{S}, \{D_a\}_{a\in\mathcal{S}}, \|\cdot\|, \mathcal{L}\rangle$ such that $I \nvDash \varphi$.*

The proof of this theorem can be found in the full version of this paper [15].

7 Conclusion

In this paper we have introduced a notion of budget-constrained dependency that generalizes the notion of functional dependency previously studied by Armstrong [4]. We propose a sound and complete axiomatization that captures the properties of the budget-constrained dependency. Although the axioms of our system are generalizations of Armstrong's original axioms, the proof of the completeness for our system is significantly more complicated than Armstrong's counterpart.

References

1. Alechina, N., Logan, B.: Ascribing beliefs to resource bounded agents. In: Proceedings of the First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2002). vol. 2, pp. 881–888. ACM Press, Bologna (July 2002)

2. Alechina, N., Logan, B., Nga, N.H., Rakib, A.: Logic for coalitions with bounded resources. Journal of Logic and Computation 21(6), 907–937 (December 2011)
3. Alechina, N., Logan, B., Whitsey, M.: A complete and decidable logic for resource-bounded agents. In: Jennings, N.R., Sierra, C., Sonenberg, L., Tambe, M. (eds.) Proceedings of the Third International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS 2004). pp. 606–613. ACM Press, New York (July 2004)
4. Armstrong, W.W.: Dependency structures of data base relationships. In: Information Processing 74 (Proc. IFIP Congress, Stockholm, 1974), pp. 580–583. North-Holland, Amsterdam (1974)
5. 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. pp. 47–61. ACM, New York, NY, USA (1977)
6. Bělohlávek, R., Vychodil, V.: Data tables with similarity relations: functional dependencies, complete rules and non-redundant bases. In: Database Systems for Advanced Applications. pp. 644–658. Springer (2006)
7. Bulling, N., Farwer, B.: Expressing properties of resource-bounded systems: The logics RTL* and RTL. In: Computational Logic in Multi-Agent Systems, pp. 22–45. Springer (2010)
8. Garcia-Molina, H., Ullman, J., Widom, J.: Database Systems: The Complete Book. Prentice-Hall, second edn. (2009)
9. Girard, J.Y.: Linear logic. Theoretical computer science 50, 1–102 (1987)
10. Hartmann, S., Link, S., Schewe, K.: Weak functional dependencies in higher-order datamodels. In: International Symposium on Foundations of Information and Knowledge Systems. pp. 116–133. Springer, Berlin Heidelberg (2004)
11. Heckle, Z., Naumov, P.: Common knowledge semantics of Armstrong's axioms. In: Proceedings of 21st Workshop on Logic, Language, Information and Computation (WoLLIC), September 1st to 4th, 2014, Valparaiso, Chile. pp. 181–194. Springer (2014)
12. Jamroga, W., Tabatabaei, M.: Accumulative knowledge under bounded resources. In: Leite, J., Son, T., Torroni, P., van der Torre, L., Woltran, S. (eds.) Computational Logic in Multi-Agent Systems, Lecture Notes in Computer Science, vol. 8143, pp. 206–222. Springer Berlin Heidelberg (2013)
13. Katz, J.: Digital Signatures. Springer Science & Business Media (2010)
14. Naumov, P., Nicholls, B.: Rationally functional dependence. Journal of Philosophical Logic 43(2-3), 603–616 (2014)
15. Naumov, P., Tao, J.: The budget-constrained functional dependency. arXiv preprint arXiv:1507.05964 (2015)
16. Naumov, P., Tao, J.: Budget-constrained knowledge in multiagent systems. In: Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems. pp. 219–226. International Foundation for Autonomous Agents and Multiagent Systems (2015)
17. Naumov, P., Tao, J.: Price of privacy. In: 12th Conference on Logic and the Foundations of Game and Decision Theory (LOFT), Maastricht, the Netherlands (2016)
18. Väänänen, J.: Dependence logic: A new approach to independence friendly logic, vol. 70. Cambridge University Press (2007)
19. Väänänen, J.: The logic of approximate dependence. arXiv preprint arXiv:1408.4437 (2014)