Constraint-Aware Role Mining via Extended Boolean Matrix Decomposition

Haibing Lu, *Member*, *IEEE Computer Society*, Jaideep Vaidya, *Member*, *IEEE Computer Society*, Vijayalakshmi Atluri, *Senior Member*, *IEEE Computer Society*, and Yuan Hong

Abstract—The role mining problem has received considerable attention recently. Among the many solutions proposed, the Boolean matrix decomposition (BMD) formulation has stood out, which essentially discovers roles by decomposing the binary matrix representing user-to-permission assignment (UPA) into two matrices—user-to-role assignment (UA) and permission-to-role assignment (PA). However, supporting certain embedded constraints, such as separation of duty (SoD) and exceptions, is critical to the role mining process. Otherwise, the mined roles may not capture the inherent constraints of the access control policies of the organization. None of the previously proposed role mining solutions, including BMD, take into account these underlying constraints while mining. In this paper, we extend the BMD so that it reflects such embedded constraints by proposing to allow negative permissions in roles or negative role assignments for users. Specifically, by allowing negative permissions in roles, we are often able to use less roles to reconstruct the same given user-permission assignments. Moreover, from the resultant roles we can discover underlying constraints such as separation of duty constraints. This feature is not supported by any existing role mining approaches. Hence, we call the role mining problem with negative authorizations the *constraint-aware role mining problem* (CRM). We also explore other interesting variants of the CRM, which may occur in real situations. To enable CRM and its variants, we propose a novel approach, extended Boolean matrix decomposition (EBMD), which addresses the ineffectiveness of BMD in its ability of capturing underlying constraints. We analyze the computational complexity for each of CRM variants and present heuristics for problems that are proven to be NP-hard.

Index Terms—RBAC, constraint-aware role mining, EBMD.

1 Introduction

Role-based access control (RBAC) has proven to be a very successful model for access control. Its flexibility and cost efficiency have resulted in it being widely adopted by current commercial systems. Indeed, it has been the model of choice for migration in the case of enterprises still employing traditional access control schemes. However, for such enterprises, the first and indeed the most crucial step is to design a good set of roles and assign them appropriately to each user. This process of designing roles is called role engineering [7]. While top-down techniques have been proposed for role engineering, for large-scale enterprises with more than thousands of users and permissions, bottom-up role extraction, called *role mining*, which mines roles purely from the existing user-to-permission assignment (UPA), may be more effective.

However, most of the existing role mining approaches only strive to find roles to best reconstitute existing user-to-permission assignments without worrying about the policy implications. They ignore the fact that the RBAC policy that best suits the business needs of an organization, may incorporate some constraints such as separation of duty or exceptions, which are usually supported in practice. In such cases, a robust role mining approach should be able to take the existing constraints into account within the role mining process. By doing so, the mined role set can well reconstitute the existing user-to-permission assignments and also reflect the real needs of an organization. To achieve this goal, in this paper, we propose a novel *constraint-aware role mining* (CRM) approach, which would enable discovery of roles along with the underlying constraints. To understand the intuition behind our approach, we first examine the exceptions and separation of duty constraints that typically may occur in organizations.

Exceptions. Exceptions are inherent to any real-world access control policy that uses some notion of abstraction in the authorization. Since role-based access control is the de facto access control model used in industry today, we restrict our attention to it. Suppose the RBAC policy states that any user with the role "manager" is allowed to access the file "project A." However, assume there exists an exception to this policy stating that all users except John (who can play the role of the manager) is not allowed to access "project A" due to certain conflict of interest requirements. Such exceptions are quite common to real-world policies. Under a typical RBAC policy, this is supported through a negative authorization as it does not make sense to create a new role specifically to John alone. It is important to realize that supporting negative authorizations sometimes may result in conflicting authorizations (in this case due to permission inheritance through role hierarchy). These can be handled by

- H. Lu is with the Department of Operations Management and Information System, The Leavey School of Business, Santa Clara University, 500 El Camino Real, Santa Clara, CA 95053. E-mail: hlu@scu.edu.
- J. Vaidya, V. Atluri, and Y. Hong are with the Department of Management Science and Information Systems, Rutgers University, 1 Washington Park, Newark, NJ 07102. E-mail: jsvaidya@rbs.rutgers.edu, atluri@rutgers.edu, yhong@cimic.rutgers.edu.

Manuscript received 8 Apr. 2011; revised 25 Sept. 2011; accepted 10 Jan. 2012; published online 30 Jan. 2012.
For information on obtaining reprints of this article, please send e-mail to: tdsc@computer.org, and reference IEEECS Log Number TDSCSI-2011-04-0099.
Digital Object Identifier no. 10.1109/TDSC.2012.21.

user	p
Alice	1
Bob	1
Cathy	1
Dave	1
Eve	1
John	-1

Fig. 1. UPA of the example above.

implementing conflict resolution policies (in this example, negatives take precedence). Assume other users assigned to "manager" are Alice, Bob, Cathy, Dave, and Eve, and the permission to access "project A" is p, the corresponding UPA of this example would be as shown in Fig. 1. Traditional role mining approaches attempt to mine roles that have the same permission sets. In this case, two roles will be mined, first comprising of Alice, Bob, Cathy, Dave, and Eve, and the second with John alone. Our proposed role mining approach in this paper attempts to capture the underlying rules of such exceptions and eliminates mining of such incorrect role sets. Note that similar exceptions can be found with other abstractions of authorizations. An example of such a policy would be "John is allowed to access all project reports except the report of project A." Our approach can elegantly handle exceptions of these kinds as well.

Separation of duty constraints. SoD constraints are an integral part of RBAC, as stated in the definition of $RBAC_2$ [25]. These help to limit exploitation of privileges and limit fraud. Consider the following toy example. Assume the following four permissions of a company: "Purchasing," "Auditing," "Marketing," and "Sales." A person can assume multiple permissions. Suppose that the same person is in charge of purchasing and sales. Hence, these two permissions are grouped together as a role, which can be represented by a Boolean vector as $\{1, 0, 0, 1\}^T$. To prevent fraud, the company has a policy stating that a person cannot assume both "Purchasing" and "Auditing" permissions. Simply representing a role as a Boolean vector cannot reflect this constraint. Even though the "auditing" permission is not included in $\{1, 0, 0, 1\}^T$, a person who has been assigned this role, can obtain the "Auditing" permission by acquiring other roles, which is perfectly valid in the BMD model. We, however, would like to recognize such constraints as part of the mining process itself.

To address this ineffectiveness of the BMD model in capturing underlying rules, we propose introducing *negative permissions* or *negative user-to-role assignment*, which can cleverly resolve both of the above issues.

As distinct from regular permissions, negative permissions mean that once a permission is assigned to a user negatively, this user can never exercise that permission. Thus, negative permissions have higher priority than positive permissions. Indeed, if the user is already assigned the permission positively through another role or even through the hierarchy, this assignment is automatically revoked. If the user is assigned the permission positively in the future, it still does not become effective. Thus, negative permissions yield a great power and can effectively model both SoD constraints and exceptions.

SoD constraints can be modeled through introducing negative permissions in roles. Consider again the "Purchasing" and "Auditing" example. To enforce the SoD constraint on them, for any role containing one of them, we add the negative permission of the other. Hence, the role of $\{1, 0, 0, 1\}^T$ is changed to $\{1, -1, 0, 1\}^T$, where the cell of -1 denotes the negative "Auditing" permission. As a result any employee assuming that role can never have the "Auditing" permission, unless the role assignment is revoked. We denote such roles, allowing negative permissions, as *rich roles*.

	p_1	p_2	p_3	p_4
u_1	1	0	1	1
u_2	1	0	1	1
u_3	1	1	0	1
u_4	0	1	0	1

Fig. 2. Existing user-to-permission assignments UPA.

Exceptions can be modeled through introducing negative user-to-role assignments. Negative user-to-role assignments mean that if a role is assigned to a user, the user cannot have access to any permission of that role. The negative user-to-role assignment is superior to the positive (or regular) user-to-role assignment. Revisiting the "Manager" example of John. To forbid him from accessing "project A," we only need to assign the "manager" role negatively to him. We call such user-to-role assignments, which include both positive and negative assignments, as *rich role assignments*.

We observe that in addition to increasing administration flexibility, negative authorizations can help discover underlying existing user-to-permission assignments during the role mining process. Consider the example of existing user-to-permission assignments UPA as shown in Fig. 2, where $\{u_1, u_2, u_3, u_4\}$ denote users and $\{p_1, p_2, p_3, p_4\}$ denote permissions.

One optimal solution of the conventional role mining problem, minimizing required roles, is as shown in Fig. 3, where $\{r_1, r_2, r_3\}$ denote roles. The first Boolean matrix gives user-to-role assignments UA and the second Boolean matrix represents permission-role assignments PA.

If we allow negative permissions in roles, a role r_i would consist of two parts, positive permissions P_i^+ and negative permissions P_i^-. Hence, a role can be represented as a vector in $\{-1, 0, 1\}$. For example, a vector $(-1, 0, 1)^T$ denotes a role with the negative authorization for the first permission and the positive authorization for the third permission. Assigning r_i to a user means that the user can never have any permission of P_i^- unless r_i is revoked and the user can have a permission of P_i^+ if he is not assigned any role consisting of its negation.

Now, let us do the same thing as the conventional role mining problem, minimizing the number of required roles. The only difference is that negative permissions are allowed this time. As we expect it to discover underlying constraints, we call it the *constraint-aware role mining problem*. For the same user-to-permission assignments as above, the resultant optimal solution is as shown in Fig. 4.

The first impression on the result is that with negative authorization for permissions we need only two roles to

	r_1	r_2	r_3
u_1	1	0	1
u_2	1	0	1
u_3	1	1	0
u_4	0	1	0

(a) UA

	p_1	p_2	p_3	p_4
r_1	1	0	0	1
r_2	0	1	0	1
r_3	0	0	1	0

(b) PA

Fig. 3. Conventional role mining.

	r_1	r_2
u_1	1	0
u_2	1	0
u_3	1	1
u_4	0	1

(a) UA

	p_1	p_2	p_3	p_4
r_1	1	0	1	1
r_2	0	1	-1	1

(b) PA

Fig. 4. Constraint-aware role mining with negative permission.

reconstruct the same existing user-to-permission assignments. Further, by taking a close look, you can find more information. First, $r_2 : \{0, 1, -1, 1\}$ shows that if r_2 is assigned to a user, he can never have the privilege of p_3. It implies that p_3 might be exclusive from p_2 and p_4. Second, $r_1 : \{1, 0, 1, 1\}$ shows that p_2 and p_4 can be existent in one role. It leaves one plausible explanation that there is a separation of duty constraint on p_2 and p_3. So, the real fact might be there are indeed only two roles in the system, $\underline{r_1 : \{1, 0, 1, 1\}}$ and $\underline{r_2 : \{0, 1, 0, 1\}}$. The reason that u_3 does not get p_3 even though he is assigned r_2, is the separation of duty constraint on p_2 and p_3. However, the conventional role mining approach is not able to discover such underlying rules. Compared to the result in Fig. 3, the result in Fig. 4 seems more plausible.

This toy example demonstrates the ability of negative authorization on discovering underlying constraints. To perform constraint-aware role mining, we propose introducing a new approach, extended Boolean matrix decomposition (EBMD). As its name indicates, EBMD is an extended version of BMD, which allows -1 in one of the decomposed matrices. Thus, EBMD is to decompose a Boolean matrix into one Boolean matrix and one matrix in $\{-1, 0, 1\}$.

From the technical perspective, constraint-aware role mining is like finding a good EBMD solution of the Boolean matrix corresponding to given user-to-permission assignments. However, it is more complicated than that. The particular role mining context has to be incorporated in the matrix decomposition process. In Section 3, we will present some variants of constraint-aware role mining and illustrate their real implications.

Note that although negative authorizations are not stated in the NIST standard for RBAC [9], negative authorizations are integral part of many access control systems. Negative authorizations have been actively studied in the RBAC literature as an effective means to implement SoD constraints and exceptions since the introduction of RBAC. Important work in this direction includes [1], [2], [3], [14]. Quoted from the work of Bertino et al. [3], introducing negative authorizations has many advantages: It enables a temporary suspension of a permission from a user without having to revoke it (revoking a permission sometimes may have a cascading effect), allows exceptions to be specified, and prevents a user from being able to exercise a privilege.

The main contribution of this paper is threefold. First, we identify the limitation that current role mining approaches are not able to reflect embedded RBAC constraints such as separation of duty or exceptions. Second, to address the limitation, we propose a novel approach, constraint-aware role mining, which takes the constraints into account while performing role mining. Third, to perform constraint-aware role mining, we model it as an EBMD problem. An additional benefit of EBMD is that it may result in a concise representation of roles. The computational complexity of the EBMD problem and its variants is carefully studied. Efficient and effective algorithms are proposed to solve those problems and extensive experiments on both synthetic and real data sets are conducted to evaluate the effectiveness of our constraint-aware role mining approach.

2 Extended Boolean Matrix Decomposition

In this section, we will introduce a novel matrix decomposition method EBMD. To help better understand the function of EBMD, we will first briefly introduce Boolean matrix multiplication and Boolean matrix decomposition.

Definition 1 (Boolean matrix multiplication). *A Boolean matrix multiplication between Boolean matrices $B \in \{0,1\}^{m \times k}$ and $C \in \{0,1\}^{k \times n}$ is $B \otimes C = A$ where A is in space $\{0,1\}^{m \times n}$ and*

$$a_{ij} = \bigvee_{l=1}^{k} (b_{il} \wedge c_{lj}).$$

Definition 2 (Boolean matrix decomposition). *If $A = B \otimes C$, where A, B, and C are Boolean matrices, $B \otimes C$ is called a decomposition of A.*

BMD is a data analytic method for binary data. It is essential to discover a set of discrete concepts and use them to describe each observed Boolean record as a union of some discrete concepts. The key advantage of BMD is to provide much interpretability to decomposition solutions.

Look at the binary matrix in Fig. 2 again. Its conventional role mining solution in Fig. 3 can be represented in the BMD form as below

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{pmatrix} \otimes \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

BMD does provide much interpretability to matrix decomposition solutions. However, it is only able to represent the set union operation. In reality, some data scenarios require the representation of the set difference operation as well. For example, in the access control setting, a role could be negatively assigned to a user, such that any permissions belonging to the role can never be assigned to the user. The advantage of negative assignments has been illustrated in the introduction.

To enable BMD to capture the set difference operation, we introduce a new concept extended Boolean matrix decomposition also called EBMD, which allows -1 in the combination matrix and uses it to represent the set difference operation.

To illustrate, see the following example:

$$\begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \\ 1 & 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}.$$

In the matrix on the left side of the equation, each row is a record, each column is an attribute and an element of

1 shows the corresponding record contains the corresponding attribute. With introducing elements of -1 (the set difference operations) in the combination matrix (the first decomposed matrix), we only need two concepts to describe those three records. As a result, the third record, (1 1 0 0), is represented by concept 1, (1 1 1 0), excluding concept 2, (0 0 1 1).

As illustrated, EBMD is to able to describe a set of observed records with a small set of concepts, such that each record can be represented as inclusion of one subset of concepts with exclusion of another subset of concepts. If a record includes one concept, that record should contain all elements of that concept; if a record excludes one concept, that record should not contain any element of that record. As is natural in set operations, exclusion overrides inclusion. In other words, if a record excludes one concept, any element in that concept is not included in the reconstructed record, even if it is present in any other concept that is included in that record.

The essential task of EBMD is to find a set of concepts and the way of reconstructing the input Boolean matrix with those concepts. Similar to BMD, a concept is represented by a Boolean vector. In BMD, the combinations are represented by a Boolean matrix, where an element of 1 for a record denotes that the corresponding concept is included, otherwise not. To reflect the set difference operation, we introduce elements of -1. So, an EBMD solution of a Boolean matrix $A_{m\times n}$ is in a form of $\{B_{m\times k}, C_{k\times n}\}$, where the concept matrix C is a Boolean matrix and the combination matrix B is in $\{-1,0,1\}$ where $b_{ij}=1$ denotes the jth record includes the ith concept and $b_{ij}=-1$ denotes the jth record excludes the ith concept. In contrast to BMD, we denote EBMD as $A = B \odot C$. The following is the formal set-theoretic EMBD definition:

Definition 3 (EBMD). *$\{B \in \{-1,0,1\}, C \in \{0,1\}\}$ is called an EBMD solution of $A \in \{0,1\}$, denoted by $A = B \odot C$, if $A_i = \cup_{b_{ij}=1} C_j \setminus \cup_{b_{ij}=-1} C_j$, where A_i denotes the item subset corresponding to elements of 1 in the jth row of A and C_j denotes similarly.*

Although the definition of EBMD is intuitive, the $\odot$ operator cannot be directly executed as the $\otimes$ operator of BMD. So, we give the following definition of the $\odot$ operator based on logic arithmetic.

Definition 4 ($\odot$ operator). *The $\odot$ operator operates over a matrix $B_{m\times k} \in \{-1,0,1\}^{m\times k}$ and a matrix $C_{k\times n} \in \{0,1\}^{k\times n}$. If $A_{m\times n} = B_{n\times k} \odot C_{k\times n}$, we have*

$$\begin{cases} a_{ij=1} & if\ (\exists t_1)(c_{i,t_1}=1\ AND\ b_{t_1,j}=1) \\ & AND\ (\neg\exists t_2)(c_{i,t_2}=1\ AND\ b_{t_2,j}=-1) \\ a_{ij=0} & if\ (\neg\exists t_1)\ (c_{i,t_1}=1\ AND\ b_{t_1,j}=1) \\ & OR\ (\exists t_2)\ (c_{i,t_2}=1\ AND\ b_{t_2,j}=-1), \end{cases}$$

where $i \in [1,m]$ and $j \in [1,n]$.

Note that $\odot$ and $\otimes$ operators are equivalent when all entries in B are in $\{0,1\}$. The EBMD operator has the commutative property described as follows:

Property 1 (Commutativity).

$$(B_{m\times k} \odot C_{k\times n})^T = C^T_{k\times n} \odot B^T_{m\times k}.$$

The commutativity implies that if $A = C \odot B$ where $C \in \{-1,0,1\}$ and $B \in \{0,1\}$, we have $A^T = B^T \odot C^T$ as well. So, EBMD essentially decomposes one Boolean matrix into one Boolean matrix and one matrix in $\{-1,0,1\}$. The order of these two decomposed matrices does not matter.

Such a commutative property of EBMD well suites the needs of constraint-aware role mining. Look at the previous example. Negative elements appear in the first decomposed matrix which is at the position of the user-to-role assignment matrix. Those negative elements then represent negative role assignments.

Consider the Fig. 4 illustrated in Section 1, in which r_2 contains a negative permission assignment. Such a constraint-aware role mining result can be represented by the following EBMD solution with negative elements appearing in the second decomposed matrix

$$\begin{pmatrix} 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & -1 & 1 \end{pmatrix}.$$

3 Problems

In this section, we formulate the problems of discovering a good role set along with its potentially embedded constraints.

As we have illustrated, negative authorizations can help identify embedded constraints and discover a succinct role-based access control system. Negative authorizations can be negative permissions in a role or negative role assignments. However, it does not make sense to have both in a system as it would be difficult to interpret a negative assignment of a role that includes a negative permission. So, in this paper, we limit negative authorizations to be only one kind, either negative role assignments or negative permissions.

We call roles and role assignments with negative authorizations as rich roles and rich role assignments, respectively.

Definition 5 (Rich role). *A rich role r_i is a role consisting of positive permissions P_i^+ and negative permissions P_i^-.*

Definition 6 (Rich role assignment). *A rich role assignment c_i consists of positive role assignments R_i^+ and negative role assignments R_i^-.*

A permission can be assigned to a user both positively and negatively. To resolve such a conflict, we require that a negative permission assignment always overrides a positive permission assignment. In other words, if a permission p_i is negatively assigned to a user, the user can never have that permission, unless the negative assignment of the permission p_i is revoked.

3.1 Primary Problems

The goal of role mining is to discover a good set of roles to meet the needs of an organization. The goodness of a set of roles is usually evaluated by the number of roles and its resultant assignment errors. So, we formulate the constraint-aware role mining problem.

Problem 1 (Constraint-aware role mining). *Given existing user-to-permission assignments $UPA_{m\times n}$ and a positive* *number k, discover a set of rich roles (or regular roles) $UA_{m\times k}$ and identify the corresponding regular role assignments (or rich role assignments) $PA_{k\times n}$, which minimize the reconstruction errors.*

Note that in the above definition either rich roles or rich role assignments exist, not both. In other words, either UA or PA is allowed to contain negative elements. So, the CRM is equivalent to finding an EBMD solution $\{UA, PA\}$ of the Boolean matrix UPA. Mathematically, the CRM can be described as the following optimization problem:

$$minimize\|UPA_{m\times n} - UA_{m\times k} \odot PA_{k\times n}\|_1. \tag{1}$$

$\|.\|_1$ norm is used to calculate the dissimilarity between two matrices, defined as $\|X_{m\times n}\|_1 = \sum_{i=1}^{m}\sum_{j=1}^{n}|x_{ij}|$.

If one believes that exception constraints are embedded in the given permission-based access control policy and should be reflected in the RBAC policy to be implemented, then user-to-permission assignments UA are allowed to have negative authorizations. In other words, the RBAC policy to be implemented will use rich role assignments to realize the exception constraints. If one believes that SoD constraints are embedded and should be reflected, rich roles, which include negative permissions in roles, will be employed in the RBAC policy to enforce those SoD constraints. From the EBMD perspective, those two cases are the same, as both of them decompose a binary matrix into one binary matrix and one matrix in $\{-1, 0, 1\}$.

When $UA \odot PA$ approximates UPA, there might be two types of reconstruction errors, *over assignments* and *under assignments*. In the under assignment case, the user can always call help desk to correct errors. However, over assignments may cause serious security problems, because users may misuse assignments that are not supposed to be granted. So, some organizations may forbid over assignment errors in the RBAC implementation phase. Due to the importance of over assignment errors, we propose and study the conservative CRM.

Problem 2 (Conservative CRM). *Same as the CRM, except that no over assignment errors are allowed.*

From the optimization perspective, the conservative CRM can be described as the following:

$$\begin{aligned} minimize\ &\|UPA_{m\times n} - UA_{m\times k} \odot PA_{k\times n}\|_1 \\ s.t.\ &(UA \odot PA)_{ij} = 0\ if\ UPA_{ij} = 0. \end{aligned} \tag{2}$$

3.2 Subproblems

The CRM and its conservative version need to determine two variable matrices UA and PA. It is difficult to determine two variable matrices at the same time, while it is relatively easier to determine one of them with the other one being fixed. Inspired from this observation, we propose an alternating minimization algorithm for the two problems.

Alternating minimization is one of the most important heuristics. It has been widely and successfully used to solve hard problems occurring in data analysis, including the k-means algorithm [15] in clustering and the iterative closest pair algorithm [4] in model registration.

Here, we give a brief introduction to the alternating minimization algorithm. Consider a minimization problem of $minf(p, q)$, where p and q are two disjoint groups of variables. Suppose that it is difficult to solve $minf(p, q)$ in whole, while it is relatively easier to solve $minf(p, q)$ with either p or q being fixed. Alternating minimization is to start from an initial solution of p and q and then repeat a two-step procedure: 1) minimizing $f(p, q)$ with p being fixed and updating q to the optimal solution; 2) minimizing $f(p, q)$ with q being fixed and then updating p to the optimal solution, until some termination condition is met. Note that alternating minimization is a heuristic and does not guarantee a global optimum. Only a portion of feasible solutions is usually traversed and a local optimum returned. Therefore, the performance of alternating minimization is affected by the initial solution choice, which will be studied in Section 6.

The sketch of the alternating minimization algorithm for the CRM is stated in Algorithm 1.

Algorithm 1. Sketch of Alternating Minimization for the CRM

```
Input: $A \in \{0,1\}^{m\times n}$
Output: $B \in \{-1,0,1\}^{m\times k}$ and $C \in \{0,1\}^{k\times n}$
Define an initial value of $\{B^0, C^0\}$;
$B^{curr} = B^0$, $C^{curr} = C^0$;
$B^{next} = arg\ min_B\|A - B \odot C^{curr}\|_1$
$C^{next} = arg\ min_C\|A - B^{curr} \odot C\|_1$
while $(B^{curr}! = B^{next})\|(C^{curr}\ ! = C^{next})$ do
    $B^{curr} = B^{next}$, $C^{curr} = C^{next}$;
    $B^{next} = arg\ min_B\|A - B \odot C^{curr}\|_1$
    $C^{next} = arg\ min_C\|A - B^{curr} \odot C\|_1$
end while
```

Each iteration of the algorithm is a two-step procedure. It leads to two subprograms. One is given the binary decomposed matrix to find the other decomposed matrix in (−1, 0, 1). The other one is the opposite. We call these two subproblems partial CRM I and partial CRM II. In the language of EBMD, they can be described as follows:

Problem 3 (Partial CRM I). *Given* $A \in \{0,1\}^{m\times n}$ *and* $C \in \{0,1\}^{k\times n}$, *find* $B \in \{-1,0,1\}^{m\times k}$ *minimizing* $\|A - B \odot C\|_1$.

Problem 4 (Partial CRM II). *Given* $A \in \{0,1\}^{m\times n}$ *and* $B \in \{-1,0,1\}^{m\times k}$, *find* $C \in \{0,1\}^{k\times n}$ *minimizing* $\|A - B \odot C\|_1$.

Apart from being the CRM's subproblems, they can also arise on their own in the RBAC setting. As for the partial CRM I, one realistic scenario is that: when regular roles (the decomposed binary matrix C) are given, the system administrator may want to reduce administrative work by employing rich role assignments (the other decomposed matrix B in {−1, 0, 1}) so that fewer role assignments are required. As for the partial CRM II, consider this case. Suppose that SoD polices have been enforced in an organization and the reflective rich roles (the decomposed matrix B in {−1,0,1}) are given. The system administrator needs to determine appropriate role assignments (the binary decomposed matrix C) so that each user gets necessary permissions.

An alternating minimization algorithm for the conservative CRM can be obtained by modifying Algorithm 1 such

TABLE 1
Comparison on CRM Variants with the Same Objective of Minimizing $\|A - B \odot C\|_1$

	Constants	Variables	Constraint
CRM	$A \in \{0,1\}^{m \times n}$	$B \in \{-1,0,1\}^{m \times k}, C \in \{0,1\}^{k \times n}$	none
Conservative CRM	$A \in \{0,1\}^{m \times n}$	$B \in \{-1,0,1\}^{m \times k}, C \in \{0,1\}^{k \times n}$	$(B \odot C)_{ij} = 0 \; if \; A_{ij} = 0$
Partial CRM I	$A \in \{0,1\}^{m \times n}$, $C \in \{0,1\}^{m \times k}$	$B \in \{-1,0,1\}^{k \times n}$	none
Conservative Partial CRM I	$A \in \{0,1\}^{m \times n}$, $C \in \{0,1\}^{m \times k}$	$B \in \{-1,0,1\}^{k \times n}$	$(B \odot C)_{ij} = 0 \; if \; A_{ij} = 0$
Partial CRM II	$A \in \{0,1\}^{m \times n}$, $B \in \{-1,0,1\}^{m \times k}$	$C \in \{0,1\}^{k \times n}$	none
Conservative Partial CRM II	$A \in \{0,1\}^{m \times n}$, $B \in \{-1,0,1\}^{m \times k}$	$C \in \{0,1\}^{k \times n}$	$(B \odot C)_{ij} = 0 \; if \; A_{ij} = 0$

that no over assignment errors are allowed at any step of the algorithm. The requirement can be achieved by adding a constraint of $(UA \odot PA)_{ij} = 0, \; if \; UPA_{ij} = 0$ to each subproblem occurring in the algorithm. As a result, the added constraint generates two new subproblems. We call them the conservative partial CRM I and the conservative partial CRM II. They are stated as the follows:

Problem 5 (Conservative partial CRM I). *Same as the partial CRM I, except that no over assignment errors are allowed.*

Problem 6 (Conservative partial CRM II). *Same as the partial CRM II, except that no over assignment errors are allowed.*

To facilitate the understanding of those CRM variants, we compare them in Table 1. Due to their structure, the effectiveness of the alternating minimization algorithms largely depends on the effectiveness of the algorithms for their subproblems. Therefore, the remaining paper focuses on those subproblems.

4 Computational Complexity Analysis

In this section, we will study NP-hardness of the partial CRM I, the conservative partial CRM I, the partial CRM II, and the conservative CRM II. We start by looking at the partial CRM I. Its decision version problem is NP-complete, which can be proven by a reduction to a known NP-complete problem, the decision BU problem [20].

Problem 7 (Decision BU). *Given binary matrices $A_{m \times n}$ and $C_{k \times n}$ and a nonnegative integer t, is there a binary matrix $B_{m \times k}$ such that $\|A - B \otimes C\|_1 \leq t$?*

Theorem 1. *The decision partial CRM I is NP-complete.*

Proof. An instance of the decision partial CRM I is a triplet of $\{A \in \{0,1\}^{m \times n}, C \in \{0,1\}^{k \times n}, t\}$, where t is a positive integer. Give a matrix $B \in \{-1,0,1\}^{m \times k}$, we can determine if $\|A - B \odot C\|_1 \leq t$ is true in a time linearly proportional to the size of the input data. So, the decision partial CRM I belongs to NP.

For every decision BU instance of $\{A'_{m \times n}, C'_{k \times n}, t'\}$, we can construct an equivalent decision partial CRM I instance of $\{A, C, t\}$ as follows: 1) A is a $m \times (n + 2t')$ binary matrix where the first $2t'$ columns containing all $1'$s and the remaining n columns are A'; 2) C is a $k \times (n + 2t')$ binary matrix where the first $2t'$ columns contain all $1'$s and the remaining n columns are C', and 3) t is equal to t'.

If the (i,j) cell of B is -1, the first $2t'$ elements at the ith row of $(B \odot C)$ must be $0'$s, which makes $\|A - B \odot C\|_1 \geq 2t'$. So, in order for the constructed decision partial CRM I instance to be true, the elements of B can only be 0 or 1. Therefore, the decision BU instance is equivalent to the constructed partial CRM I instance. □

Before studying the conservative partial CRM I, we introduce a known NP-hard problem, the red-blue set cover problem (RBSC) [5].

Problem 8 (RBSC). *Given a finite set of red elements R and a finite set of blue elements B and a family $\mathcal{S} = \{S_1, \ldots, S_n\} \in 2^{R \cup B}$, find a subfamily $\mathcal{C} \in \mathcal{S}$, the union of which covers all blue elements and the minimum possible number of red elements.*

The conservative partial CRM I is equivalent to a special variant of the RBSC, which we call extended RBSC I. Details on their mapping are provided in the next section. So, for convenience, instead of studying the conservative partial CRM I, we consider the extended RBSC I.

Problem 9 (Extended RBSC I). *Given a collection $\mathcal{C}$ of subsets of red-blue elements $\{\mathcal{B} \cup \mathcal{R}\}$, find two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $(\cup \mathcal{C}_1) \backslash (\cup \mathcal{C}_2)$ maximizes #(covered blue elements), while no red elements are covered.*

Theorem 2. *The decision extended RBSC I is NP-complete.*

Proof. A decision extended RBSC I instance is $\{R, B, \mathcal{S}, t\}$. Given a solution of $\{\mathcal{C}_1, \mathcal{C}_2\}$, we can determine if it is true in a polynomial time. So, the problem belongs to NP.

For any decision RBSC instance $\{R, B, \mathcal{S}, t\}$, we can create an equivalent decision extended RBSC I instance $\{R', B', \mathcal{S}', t'\}$, such that

- for each blue element $blue_i$ in B, create a corresponding red element red'_i and include it in R'. Hence, $|B| = |R'|$.
- For each red element red_i in R, create a corresponding blue element $blue'_i$ and include it in B'.
- In addition, we create k more blue elements, $\{blue'_{|R|+1}, \ldots, blue'_{|R|+k}\}$, where $k \gg |R| + |B|$.
- For each $s_i \in \mathcal{S}$, create s'_i, such that for each $blue_i$ in s_i, include the corresponding red'_i in s'_i and for each red_i in s_i, include the corresponding $blue'_i$ in s'_i. Include s'_i in $\mathcal{S}'$.
- Create a subset of $s'_{|\mathcal{S}|+1}$, such that it contains all blue and red elements. In other words, $s'_{|\mathcal{S}|+1} = R \cup B \cup \{blue'_{|R|+1}, \ldots, blue'_{|R|+k}\}$.
- Let $\mathcal{S}' = \{s'_1, \ldots, s'_{|\mathcal{S}|}\} \cup s_{|\mathcal{S}|+1}$.
- Let t$' = t$.

Because $s'_{|\mathcal{S}|+1}$ contains k new blue elements, which do not belong to any subset, and $k \gg |R| + |B|$, the optimal solution should be that $s'_{|\mathcal{S}|+1}$ excludes a subcollection of $\{s'_1, \ldots, s'_{|\mathcal{S}|}\}$, the union of which covers all red elements and the minimum blue elements. As the blue elements in $\mathcal{S}'$ correspond to the red elements in $\mathcal{S}$ and the red elements in $\mathcal{S}'$ correspond to the blue elements in $\mathcal{S}$, so the decision RBSC instance is true, if and only if the constructed decision extended RBSC I instance is true. □

The partial CRM II is NP-hard as well, which can be proven from a reduction from the BU.

Theorem 3. *The decision partial CRM II is NP-complete.*

Proof. An instance of the decision partial CRM II is a triplet of $\{A \in \{-1,0,1\}^{m\times n}, B \in \{-1,0,1\}^{m\times k}, t\}$. Given a solution $C \in \{0,1\}^{k\times n}$, it is easy to determine whether $\|A - B \odot C\|_1 \leq t$ is true. So, the decision partial CRM II belongs to NP.

For every instance of the decision BU $\{A', B', t'\}$, we can create an equivalent decision partial CRM II instance $\{A, B, t\}$, such that $A = A'$, $B = B'$, and $t = t'$. As B has no negative elements, so $B \otimes C$ is the same as $B \odot C$. Therefore, the constructed decision partial CRM II instance is true if and only if the decision BU instance is true. □

Before we study the conservative partial CRM II, we introduce a known NP-hard problem, Positive-Negative Partial Set Cover (±PSC) [20] and its variant Equal ±PSC.

Problem 10 (±PSC). *Given disjoint sets P and N of positive and negative elements, respectively, and a collection $\mathcal{S}$ of subsets of $P \bigcup N$, find a subcollection $\mathcal{C} \in \mathcal{S}$ minimizing $|P\backslash(\cup\mathcal{C}|) + |N \cap (\cup\mathcal{C}|)$.*

Problem 11 (Equal ±PSC). *Given disjoint sets P and N of positive and negative elements, respectively, where $|P| = |N|$, and a collection $\mathcal{S}$ of subsets of $P \bigcup N$, find a subcollection $\mathcal{C} \in \mathcal{S}$ minimizing $|P\backslash(\cup\mathcal{C})| + |N \cap (\cup\mathcal{C})|$.*

Theorem 4. *The decision equal ±PSC is NP-complete.*

Proof. Equal ±PSC is a special case of ±PSC. Obviously, it belongs to NP. Next, we will show that for every instance of decision ±PSC, we can find a corresponding decision equal ±PSC instance. Given a decision ±PSC instance as $\{P, N, \mathcal{S}, t\}$, we create a corresponding equal ±PSC instance $\{P', N', \mathcal{S}', t'\}$ such that

- if $|P| < |N|$
  - Introduce $|N| - |P|$ new positive elements, $\{p'_{|P|+1}, \ldots, p'_{|N|}\}$. Let $P' = P \cup \{p'_{|P|+1}, \ldots, p'_{|N|}\}$.
  - Let $N' = N$.
  - For every subset $s_i \in \mathcal{S}$, create a subset s'_i such that $s'_i = s_i \cup \{p'_{|P|+1}, \ldots, p'_{|N|}\}$ and include it in $\mathcal{S}'$. So, $\mathcal{S}' = \{s'_1, \ldots, s'_{|\mathcal{S}|}\}$.
  - $t' = t$.
- else if $|P| > |N|$
  - Introduce $|P| - |N|$ new negative elements, $\{n'_{|N|+1}, \ldots, n'_{|P|}\}$. Let $N' = N \cup \{n'_{|N|+1}, \ldots, n'_{|P|}\}$.
  - Let $P' = P$.
  - For every subset $s_i \in \mathcal{S}$, create a subset s'_i such that $s'_i = s_i \cup \{n'_{|N|+1}, \ldots, n'_{|P|}\}$ and include it in $\mathcal{S}'$. So, $\mathcal{S}' = \{s'_1, \ldots, s'_{|\mathcal{S}|}\}$.
  - $t' = t + (|N| - |P|)$.
- else
  - $P' = P$; $N' = N$; $\mathcal{S}' = \mathcal{S}$; $t' = t$.

Consider the case of $|P| < |N|$. If the ±PSC instance, $\{P, N, \mathcal{S}, t\}$, is true, there exists a subcollection $\mathcal{C} \in \mathcal{S}$ such that $|P\backslash(\cup\mathcal{C})| + |N \cap (\cup\mathcal{C})| < t$. We can find a subcollection $\mathcal{C}' \in \mathcal{S}'$ corresponding to $\mathcal{C} \in \mathcal{S}$. As $\{p'_{|P|+1}, \ldots, p'_{|N|}\}$ belong to any subset in $\mathcal{C}'$, we have $|P'\backslash(\cup\mathcal{C}')| = |P\backslash(\cup\mathcal{C})|$. It is obviously true that $|N' \cap (\cup\mathcal{C}')| = |N \cap (\cup\mathcal{C})|$. So, we have $|P'\backslash(\cup\mathcal{C}')| + |N' \cap (\cup\mathcal{C}')| < t$. In the other way, if the decision equal ±PSC instance is true, the ±PSC instance must be true.

For $|P| > |N|$, as new negative elements $\{n'_{|N|+1}, \ldots, n'_{|P|}\}$ are added for each subset, we have $|N' \cap (\cup\mathcal{C}')| = |N \cap (\cup\mathcal{C}) + (|N| - |P|)|$. It is true that $\{p'_{|P|+1}, \ldots, p'_{|N|}\} = \{p_{|P|+1}, \ldots, p_{|N|}\}$. Hence, we have $|P\backslash(\cup\mathcal{C})| + |N \cap (\cup\mathcal{C})| = |P\backslash(\cup\mathcal{C})| + |N \cap (\cup\mathcal{C})| + (|N| - |P|)$. Therefore, the decision ±PSC instance is true if and only if the equal ±PSC instance is true. □

We will prove the decision conservative partial CRM II is NP-complete by relating it to the decision equal ±PSC.

Theorem 5. *The decision conservative partial CRM II is NP-complete.*

Proof. The conservative partial CRM II is given user-permission assignments $A_{m\times n}$ and rich roles $B \in \{-1,0,1\}^{m\times k}$ to determine role assignments $C \in \{0,1\}^{k\times n}$. Given a solution C, it is easy to determine if $\|A - C \odot B\|_1 \leq t$ is true,[1] where t is a constant. So, the decision conservative partial CRM II belongs to NP.

For any decision ±PSC instance $\{P, N, \mathcal{S}, t\}$, we can create an equivalent decision conservative partial CRM II instance $\{A \in \{0,1\}^{m\times n}, B \in \{-1,0,1\}^{m\times k}, t'\}$, such that

- Let m be 1 and n be $|P|$, the number of positive elements.
- Let A be a single row with all elements being 1.
- For each subset s_i of $\mathcal{S}$, create a rich role (a row) in B, such that
  - if s_i contains n_j, $B(i,j) = -1$.
  - if s_i contains p_j and excludes n_j, $B(i,j) = 1$.
  - if s_j has neither p_j, nor n_j, $B(i,j) = 0$.
- Let t′ = t.

We use an example to illustrate the instance construction. Suppose a ±PSC instance is $\{\{p_1, p_2, n_1\}, \{p_2, n_2\}, \{p_1, n_2\}\}$ with two positive elements and two negative elements. Then, the corresponding conservative partial CRM II instance is $A = (1, 1)$ and

$$B = \begin{pmatrix} -1 & 0 \\ 0 & -1 \\ 1 & -1 \end{pmatrix}.$$

Suppose a subset of rich roles are selected, which gives role assignments C. Denote $\mathcal{C}$ to be the subcollection of $\mathcal{S}$ in the ±PSC instance corresponding to the role assignments C in the decision conservative partial CRM II instance. Recall the definition of rich roles. If a user is

[1] Due to the commutative property of the $\odot$ operator, $C \odot B$ is equal to $B \odot C$.

assigned to a role with negative permission i, he can never have that permission, even though he is assigned to other roles containing positive permission i. So, $\|A - C \odot B\|_1$ can be reformulated as the following:

$$
\begin{aligned}
&|Permissions| - (\#(covered\ positive\ permissions)\\
&\quad - \#(covered\ negative\ permissions))\\
&= |Permissions| - \#(covered\ positive\ permissions)\\
&\quad + \#(covered\ negative\ permissions)\\
&= \#(uncovered\ positive\ permissions)\\
&\quad + \#(covered\ negative\ permissions)\\
&= |P \backslash (\cup \mathcal{C})| + |N \cap (\cup \mathcal{C})|.
\end{aligned}
$$

It shows that the equal ±PSC instance is true if and only if the constructed partial CRM II instance is true. □

5 Algorithms

In this section, we will provide heuristics for partial CRM I, partial CRM II, and their conservative versions.

5.1 Partial CRM I

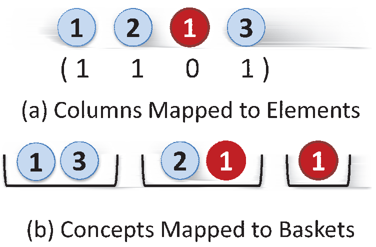
In the language of EBMD, the partial CRM I problem is given a matrix $A \in \{0,1\}^{m \times n}$ and a matrix $C \in \{0,1\}^{k \times n}$ to find the matrix $B \in \{-1,0,1\}^{m \times k}$ such that $\|A - B \odot C\|_1$ is minimized. As $\|A - B \odot C\|_1 = \sum_i \|A_i - B_i \odot C\|_1$, where A_i and B_i denote the ith row of A and B, respectively, a partial CRM I problem can be divided into a set of subproblems with each row of A as an input. So, without loss of generality, we consider A to be a Boolean row vector. Such a reduction allows us to describe the partial CRM I as a variant of the RBSC problem as the following:

- *Given a collection $\mathcal{C}$ of subsets of red-blue elements $\{\mathcal{B} \cup \mathcal{R}\}$, find two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $(\cup \mathcal{C}_1) \backslash (\cup \mathcal{C}_2)$ maximizes $\#(covered\ blue\ elements) - \#(covered\ red\ elements)$.*

To illustrate the mapping, consider the following partial CRM I problem, where the variables $\{b_1, b_2, b_3\}$ need to be determined

$$
\left\{
\begin{array}{l}
A : (1 \quad 1 \quad 0 \quad 1)\\
C : \begin{pmatrix} 1 & 0 & 0 & 1\\ 0 & 1 & 1 & 0\\ 0 & 0 & 1 & 0 \end{pmatrix}.\\
B : \{b_1, b_2, b_3\}
\end{array}
\right. \tag{3}
$$

Let each column of A correspond to a distinct red or blue element. In particular, the columns at which the elements of A are 1 correspond to blue elements and the other columns correspond to red elements. Thus, in the example the first, second, and fourth columns are mapped to blue elements $\{blue_1, blue_2, blue_3\}$, respectively, and the third column is mapped to the red element $\{red_1\}$, as illustrated in Fig. 5a.[2] Consequently, the matrix C can be mapped to a collection $\mathcal{C}$ of subsets of red-blue elements $\{\mathcal{B} \cup \mathcal{R}\}$, where $\mathcal{B}$ and $\mathcal{R}$ denote the blue element set and the red element set, respectively, such that $\{\{blue_1, blue_3\}$, $\{blue_2, red_1\}$,$\{red_1\}\}$, as illustrated in Fig. 5b. Denote baskets from left to right in Fig. 5b to be c_1, c_2, and c_3, respectively. In the partial CRM I, (b_1, b_2, b_3) are allowed to be -1, 0, or 1, where 1 corresponds to the set union operation and -1 corresponds to the set difference operation. Therefore, determining the optimal values of (b_1, b_2, b_3) is equivalent to finding two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $(\cup \mathcal{C}_1) \backslash (\cup \mathcal{C}_2)$ maximizes $\#(covered\ blue\ elements) - \#(covered\ red\ elements)$. It is not difficult to see that the optimal solution in this toy example is $(c_1 \cup c_2) \backslash c_3$. Therefore, $\{b_1, b_2, b_3\} = (1, 1, -1)$.



(a) Columns Mapped to Elements

(b) Concepts Mapped to Baskets

Fig. 5. Mapping Illustration I.

As proven in the previous section, the decision partial CRM I problem is NP-complete. So, we propose a greedy heuristic. We first divide the given subset collection $\mathcal{C}$ into three groups $\{\mathcal{C}^{\mathcal{B}}, \mathcal{C}^{\mathcal{R}}, \mathcal{C}^{\mathcal{B},\mathcal{R}}\}$, where $\mathcal{C}^{\mathcal{B}}$ includes subsets containing blue elements only, $\mathcal{C}^{\mathcal{R}}$ consists of red elements only and the subsets in $\mathcal{C}^{\mathcal{B},\mathcal{R}}$ have both red and blue elements. Obviously, including $\mathcal{C}^{\mathcal{B}}$ in $\mathcal{C}_1$ and $\mathcal{C}^{\mathcal{R}}$ in $\mathcal{C}_2$ does not introduce any covering error. Since $\mathcal{C}^{\mathcal{R}}$ has been included in $\mathcal{C}_2$, assigning any subset of $\mathcal{C}_2$, in which all contained red elements belong to $\mathcal{C}^{\mathcal{R}}$, to $\mathcal{C}_1$ does not introduce covering error either. Next, we need to assign the remaining subsets of $\mathcal{C}^{\mathcal{B},\mathcal{R}}$ to either $\mathcal{C}_1$ or $\mathcal{C}_2$. We will do it in an iterative fashion. At each step, we select a subset c from the remaining $\mathcal{C}^{\mathcal{B},\mathcal{R}}$ and put it in $\mathcal{C}_1$. The selection criterion is based on the following function:

$$
f_1 = \frac{\#(Newly\ Covered\ Blue)}{\#(Newly\ Covered\ Red)}. \tag{4}
$$

The numerator part in the criterion function denotes the number of newly covered blue elements when including a new subset c, while the denominator part denotes the number of newly covered red elements. We iteratively select the subset with the greatest measure till all blue elements are covered. The complete algorithm is as described in Algorithm 2.

Algorithm 2. Partial CRM I
Input: A collection $\mathcal{C}$ of subsets of $\{\mathcal{B} \cup \mathcal{R}\}$.
Output: Two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$.

```
Divide $\mathcal{C}$ into $\{\mathcal{C}^{\mathcal{B}}, \mathcal{C}^{\mathcal{R}}, \mathcal{C}^{\mathcal{B},\mathcal{R}}\}$
Include $\mathcal{C}^{\mathcal{B}}$ in $\mathcal{C}_1$, and $\mathcal{C}^{\mathcal{R}}$ in $\mathcal{C}_2$;
Update $\mathcal{C}^{\mathcal{B},\mathcal{R}}$ by deleting elements contained in $\mathcal{C}^{\mathcal{B}}$
    and $\mathcal{C}^{\mathcal{R}}$;
Set $\mathcal{B}' = \cup \mathcal{C}^{\mathcal{B},\mathcal{R}} \cap \mathcal{B}$ and $\mathcal{R}' = \emptyset$;
while The objective value can be further improved. do
    Select the subset $c \in \mathcal{C}^{\mathcal{B},\mathcal{R}}$ with the largest $f_1$ value
        and include it in $\mathcal{C}_1$;
    Update $\mathcal{B}'$ as $\mathcal{B}' \backslash (c \cap \mathcal{B})$, and $\mathcal{R}'$ as $\mathcal{R}' \cup (c \cap \mathcal{R})$.
end while
```

Notice that at the beginning of Algorithm 2, the whole $\mathcal{C}^{\mathcal{R}}$ is included in $\mathcal{C}_2$. The purpose of doing it is to allow as

[2] If printed in black and white, the light shade is blue and the dark shade is red.

many subsets in $C^{\mathcal{B},\mathcal{R}}$ to be included as possible. However, at the end of the algorithm, some baskets in $C^{\mathcal{R}}$ may not be utilized at all. In other words, these baskets do not differentiate any red elements included in C_1. So, from the perspective of role assignment effectiveness, these negative role assignments should be removed, although they do not improve the assignment accuracy.

To illustrate Algorithm 2, consider the example as shown in Figs. 5a and 5b. According to Algorithm 2, to cover $blue_1$, $blue_2$, and $blue_3$, first the three baskets in Fig. 5b are divided into three groups with the first basket in $\mathcal{C}^B$, the second basket in $\mathcal{C}^{B,R}$, and the third basket in $\mathcal{C}^R$. Then, the first basket is included in $\mathcal{C}_1$ as it contains blue elements only and the third basket is included in $\mathcal{C}_2$ as it contains red elements only. Finally, we consider the second basket which contains $blue_2$ and red_1. According to Algorithm 2, it is included in $\mathcal{C}_1$ because $blue_2$ will be covered and red_1 in the second basket will be canceled out by the same element in the third basket which is included in $\mathcal{C}_2$. Therefore, the resultant solution of (b_1, b_2, b_3) is $(1, 1, -1)$, where the first two elements of 1 correspond to the fact that the first and second baskets are included in $\mathcal{C}_2$ and the third element of -1 corresponds to the fact that the third basket is included in $\mathcal{C}_2$.

5.2 Conservative Partial CRM I

The conservative partial CRM I problem requires no 0-becoming-1 errors. In the red-blue set cover problem setting, it can be described as follows:

- *Given a collection $\mathcal{C}$ of subsets of red-blue elements $\{\mathcal{B} \cup \mathcal{R}\}$, find two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$ such that $(\cup\mathcal{C}_1)\backslash(\cup\mathcal{C}_2)$ maximizes $\#(covered\ blue\ elements)$, while no red elements are covered.*

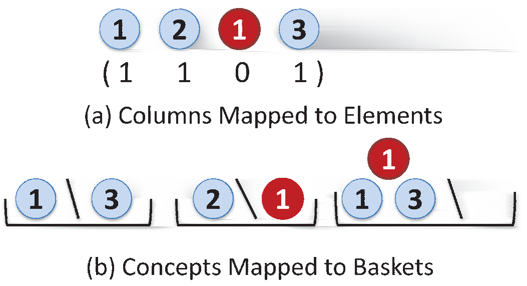
Its decision problem has been proven to be NP-complete. So, a greedy heuristic is proposed and stated in Algorithm 3. Notice that the first two steps are the same as that for the partial CRM I problem. After the first two steps, only subsets in $\mathcal{C}^{\{\mathcal{B},\mathcal{R}\}}$ remain. For each remaining subset c, if its contained elements are already included in $\mathcal{C}_2$, we include it in $\mathcal{C}_1$, as it will not introduce any red element in the final solution.

Algorithm 3. Conservative Partial CRM I
Input: A collection $\mathcal{C}$ of subsets of $\{\mathcal{B} \cup R\}$.
Output: Two subcollections $\mathcal{C}_1$ and $\mathcal{C}_2$.

```
1: Divide $\mathcal{C}$ into $\{\mathcal{C}^B, \mathcal{C}^R, \mathcal{C}^{\mathcal{B},\mathcal{R}}\}$
2: Include $\mathcal{C}^{\mathcal{B}}$ in $\mathcal{C}_1$ and $\mathcal{C}^R$ in $\mathcal{C}_2$
3: for each $c \in \mathcal{C}^{\mathcal{B},\mathcal{R}}$ do
4:     if $c \cap \mathcal{R} \in \mathcal{C}^{\mathcal{R}}$ then
5:         Include $c$ in $\mathcal{C}_1$.
6:     end if
7: end for
```

5.3 Partial CRM II

The partial CRM II is given a matrix $A \in \{0,1\}^{m\times n}$ and a decomposed matrix B in $\{-1, 0, 1\}^{m\times k}$, to find the other decomposed matrix $C \in \{0,1\}^{k\times n}$ to minimize the reconstruction errors. As the $\odot$ operator has the commutative property, for convenience we state the partial CRM II as minimizing $\|A - C \odot B\|_1$. As $\|A - C \odot B\|_1 = \sum_i(\|A_i - C_i \odot B\|_1)$, the original problem can be divided into a set of subproblems with each row of A as the input data.



(a) Columns Mapped to Elements

(b) Concepts Mapped to Baskets

Fig. 6. Mapping Illustration II.

Therefore, without loss of generality, we consider A as a Boolean row vector

$$\begin{pmatrix} 1 & 1 & 0 & 1 \end{pmatrix} = \begin{pmatrix} b_1 & b_2 & b_3 \end{pmatrix} \odot \begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix}. \quad (5)$$

For ease of explaining our algorithm later, we first look at an example as shown in (5). The row vector on the left and the matrix on the right are the input data. $\{b_1, b_2, b_3\}$ are Boolean variables to be determined. This partial CRM II problem can be also viewed as a variant of the red-blue set cover problem. First, we map columns to red-blue elements. The mapping policy is the same as what we did for the partial CRM I problem. The mapping result is as shown in Fig. 6. Now, we will map each row vector in the concept matrix on the right side of (5) to a basket of red-blue elements. Notice that each row vector may contain three different component values $\{-1,\ 0,\ 1\}$. The value of 1 corresponds to the set union operation, while the value of -1 corresponds to the set different operation. To reflect that, we map each row vector to a special red-blue element basket in the form of $c^+\backslash c^-$, where both c^+ and c^+ are red-blue element subsets. Based on this mapping rule, the row vectors in the example of (5) are mapped to baskets as illustrated in Fig. 6b, where the symbol of $\backslash$ denotes the set difference operator. Therefore, the partial CRM II problem can be described as follows:

- *Given a basket set of $\{\{c_1^+\backslash c_1^-\}, \ldots, \{c_k^+\backslash c_k^-\}\}$, where $\{c_i^+, c_i^-\}$ are red-blue element subsets, select a basket subset $\mathcal{S}$ such that $(\cup_{i\in\mathcal{S}} c_i^+)\backslash(\cup_{i\in\mathcal{S}} c_i^-)$ maximizes*

$$\#(covered\ blue\ elements) - \#(covered\ red\ elements).$$

For the example of Fig. 6, it is not difficult to see that the optimal solution is to select the second and third baskets. The result covers all blue elements without introducing one red element.

The partial CRM II has been proven to be NP-hard. So, we propose a greedy heuristic. Its basic idea is to iteratively select the best remaining basket based on some selection criterion. We observe that four cases may occur when including a basket into the solution:

1. new blue elements being covered;
2. new red elements being covered;
3. new blue elements being excluded; and
4. new red elements being excluded.

Obviously, the first and the fourth cases are desired while the other two cases are disliked. So, our selection criterion is based on the function in (6) and the greedy heuristic is described in Algorithm 4

$$f_2 = \frac{\#(Newly\ Covered\ Blue) + \#(Newly\ Excluded\ Red)}{\#(Newly\ Covered\ Red) + \#(Newly\ Excluded\ Blue)}. \tag{6}$$

Algorithm 4. Partial CRM II

Input: A red-blue element basket set of $\{\{c_1^+\backslash c_1^-\},\ldots,\{c_k^+\backslash c_k^-\}\}$.

Output: A red-blue element basket subset $\mathcal{S}$.

```
Iteratively include the basket with the greatest $f_2$
    value into $\mathcal{S}$ till the objective value cannot be
    improved.
```

5.4 Conservative Partial CRM II

The conservative partial CRM II does not allow over assignment errors. It can also be studied in the setting of the red-blue set cover problem as follows:

- *Given a basket set of $\{\{c_1^+\backslash c_1^-\},\ldots,\{c_k^+\backslash c_k^-\}\}$, where $\{c_i^+, c_i^-\}$ are red-blue element subsets, select a basket subset $\mathcal{S}$ such that $(\cup_{i\in\mathcal{S}} c_i^+)\backslash(\cup_{i\in\mathcal{S}} c_i^-)$ maximizes $\#(covered\ blue\ elements)$ while no red elements are covered.*

As the objective is to cover as many blue elements as possible, for simplicity we only consider baskets $\{c_i^+\backslash c_i^1\}$ with $c_i^1 \subseteq R$ only. The intuition behind is not to exclude any blue elements in the final solution. However, if we select all such baskets, red elements may be included. To eliminate red elements, we remove troubling baskets in an iterative fashion. In particular, at each step we delete the basket which reduces the number of covered red elements the most. The complete description is provided in Algorithm 5.

Algorithm 5. Conservative Partial CRM II

Input: A red-blue element basket set of $\{\{c_1^+\backslash c_1^-\},\ldots,\{c_k^+\backslash c_k^-\}\}$.

Output: A red-blue element basket subset $\mathcal{S}$.

```
For each basket $\{c_i^+\backslash c_i^-\}$, if $c_i^- \subseteq R$, include it into $\mathcal{S}$.
Iteratively remove the basket $\{c_i^+\backslash c_i^-\}$ from $\mathcal{S}$, which
    reduces the number of covered elements the most, till
    no red elements being covered.
```

6 Experimental Study

In this section, we will conduct experiments to compare our algorithms with other algorithms including the $Loc\&IterX$ algorithm for the discrete basis problem proposed by Pauli [20], the $FastMiner$ algorithm for the conventional role mining problem proposed by Vaidya et al. [30], the MAC algorithm for multiassignment clustering for Boolean data proposed by Streich et al. [27], and singular vector decomposition (SVD) used for cleaning access control data sets by Molloy et al. [23]. All experiments are implemented in Matlab and are conducted on a Dell desktop with a 2.8 GHz dual core processor and 4 GB memory.

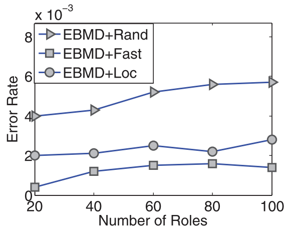


Fig. 7. Comparison on different initial solutions.

6.1 Synthetic Data

A synthetic binary matrix $A_{m\times n}$ is generated as follows: First, generate a random binary matrix $C_{k\times n}$, such that each entry has the probability ρ_1 of being 1 and there is no redundant row vector. Second, generate a random matrix $B_{m\times k}$, such that each entry has the probability ρ_2 of being 1, the probability ρ_3 of being -1, and the probability $1-\rho_2-\rho_3$ of being 0. Let A be $B\odot C$. To reflect data noise in reality, we randomly select δ ratio of entries in A and flip their values. If the resultant A has some zero row vector, we repeat the whole data generation process till every row vector in A contains at least one 1's entry.

Experiment 1. The first experiment is to test the impact of the initial solution to the performance of the alternating minimization algorithm. Given A, to discover its original EBMD solution of B and C, we use three different initialization strategies for our alternating minimization heuristic. The first is to randomly generate a binary matrix C and then use it as the initial solution to feed the algorithm, which we call $EBMD + Rand$. The second one is running the $FastMiner$ algorithm to discover roles and using the discovered roles as the initial solution for C, which we call $EBMD + Fast$. The third one is running the $Loc\&IterX$ algorithm, which is based on association rule mining, to discover centroids of the given binary matrix A and using the discovered centroids as the initial solution of C, which we call $EBMD + Loc$.

The synthetic data are generated according to the above described data generator. The parameter settings are: $m = 100$, $n = 50$, $\rho_1 = 0.3$, $\rho_2 = 0.4$, $\rho_3 = 0.1$, and $\delta = 0$. We generate five binary matrices with the setting of k, ranging from 4 to 20. We run our alternating minimization heuristic with the three different initialization strategies on the synthetic data sets with the assumption that k is known. We compare different initialization strategies in terms of their ability in reconstructing the original binary matrix A. To facilitate the comparison, we used the following error ratio metric:

$$\|A - A'\|_1/size(A), \tag{7}$$

where A is the input matrix and A' is the reconstructed matrix. The results are plotted in Fig. 7. An instant observation is that $EBMD + Fast$ and $EBMD + Loc$ perform much better than $EBMD + Rand$. Therefore, random initialization is not a good choice, although it is simple.

Experiment 2. The second experiment is to test the effect of the value of k with respect to the error rate. We use the same parameter settings as the previous experiment. For each size of k ranging from 4 to 20, we generate five matrices. Reported results are the mean value and the

TABLE 2
Error Ratio w.r.t Role Number for Synthetic Data

k	4		8		12		16		20	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
FastMiner	0.013	0.001	0.039	0.001	0.051	0.003	0.052	0.002	0.062	0.004
EBMD+Fast	0.013	0.001	0.036	0.001	0.042	0.002	0.046	0.002	0.051	0.003
Conservative EBMD +Fast	0.013	0.001	0.036	0.002	0.044	0.003	0.045	0.003	0.055	0.004
Loc & IterX	0.040	0.001	0.091	0.002	0.099	0.002	0.110	0.003	0.126	0.003
EBMD+Loc	0.039	0.001	0.076	0.002	0.093	0.002	0.105	0.002	0.120	0.003
Conservative EBMD + Loc	0.040	0.001	0.079	0.003	0.093	0.003	0.105	0.002	0.120	0.003

TABLE 3
Error Ratio w.r.t Noise for Synthetic Data

δ	0		0.1		0.2		0.3		0.4	
	mean	STD	mean	STD	mean	STD	mean	STD	mean	STD
FastMiner	0.036	0.002	0.096	0.003	0.128	0.005	0.146	0.007	0.154	0.009
EBMD+Fast	0.029	0.002	0.092	0.002	0.128	0.005	0.143	0.008	0.154	0.007
Conservative EBMD +Fast	0.030	0.002	0.096	0.002	0.128	0.004	0.146	0.007	0.154	0.008
Loc & IterX	0.091	0.003	0.120	0.003	0.151	0.005	0.168	0.007	0.172	0.008
EBMD+Loc	0.082	0.002	0.112	0.003	0.149	0.004	0.168	0.007	0.172	0.006
Conservative EBMD + Loc	0.083	0.002	0.112	0.003	0.151	0.005	0.168	0.007	0.172	0.007

standard deviation over those five matrices. When running an algorithm on a binary matrix $A_{100\times 50}$, we assumed its real decomposition size k is known and thus decompose $A_{100\times 50}$ into $B_{100\times k}$ and $C_{k\times 50}$. The experimental results are reported in Table 2, where STD stands for standard deviation.

The result shows that our EBMD model performs better than $FastMiner$ and $Loc\&IterX$ with respect to the error rate. In other words, the EBMD model can better describe the input data than the BMD model, which effectively validates the usefulness of our model. Another observation is that there is no much difference between EBMD and conservative EBMD in terms of the error rate. Theoretically, as conservative EBMD carries a constraint that zero elements cannot become one, regular EBMD should be able to better reconstruct the same binary matrix than conservative EBMD. However, in Table 2, for most cases the performances of EBMD and conservative EBMD are comparable. The underlying reason is that both the EBMD algorithm and the conservative EBMD algorithm are heuristics.

Experiment 3. The third experiment tests the effect of the noise parameter δ with respect to error rate. All parameter settings are the same as previous experiments, except that k is fixed to 10 and δ varies from 0 to 0.4. For each configuration, we generated five matrices and reported results are mean values over these five matrices and their standard deviations, which help tell the significance of the difference between mean values. We assumed the value of k is known and run the FastMiner algorithm and the $Loc\&InterX$ algorithm, respectively, for each generated matrix. Then, we took their solutions in our EMBD and conservative EBMD algorithms to obtain EBMD solutions. The results are reported in Table 3

The first observation is that when there is no noise, the performance of EBMD is significantly better than that of $FastMiner$ and $Loc\&IterX$. However, when the noise ratio increases, the advantage of our EBMD-based algorithms over $FastMiner$ and $Loc\&IterX$ decreases. As Table 3 shows, when δ is greater than 0.2, the performances of the EBMD and BMD algorithms become comparable. Note that when the δ value increases, the reconstruction error rate increases accordingly. This is because increased errors disrupt the internal structure of the binary matrix and therefore more roles are required to describe the noisy signals.

Experiment 4. The fourth experiment is to compare our approach with the MAC algorithm, which is presented for multiassignment clustering for Boolean data in [27]. As the EBMD product has the commutative property, so without loss of generality, we assume that the synthetic user-to-permission assignments A is embedded with exception constraints. In other words, the RBAC policy to be implemented consists of regular roles and rich role assignments. We compare our approach with the MAC algorithm in terms of the ability in recovering the real regular roles and the ability in reconstructing the original data.

We generate five sets of binary matrices with k ranging from 4 to 20 while the other parameter settings are the same as the previous experiments. For each size k, we generate five matrices. So, all results reported later are mean values. We apply both our EBMD approach and the MAC algorithm to discover the roles C and assume that the number of roles k is known to both approaches. The code for the MAC algorithm is available at the author's website.[3] Their code needs to set up the number of the maximum roles a user can take. We use the default value in their code. In fact, we tried some larger values and the resultant computing time was too expensive given that the algorithm itself is already time consuming.

To compare the similarity of discovered roles C' with real roles C, we use the following formula to measure the similarity between two sets of roles:

$$S(C', C) = \frac{\sum_{i=1}^{k} max_j S(C'_i, C_j)}{k}$$

where C'_i and C_j denote the ith role of C' and the jth role of C, respectively, and $S(C'_i, C_j)$ computes the similarity between two roles. The formula of $S(C'_i, C_j)$ is based on the Jaccard similarly coefficient [28] and is presented as the following: $S(C'_i, C_j) = \frac{\#(t|C'_i(t)=1 \,\&\&\, C_j(t)=1)}{\#(t|C'_i(t)=1 \,\|\, C_j(t)=1)}$. The results are

[3] http://www.inf.ethz.ch/personal/astreich/index.html.

TABLE 4
Comparison with MAC w.r.t Role Similarity

k	4	8	12	16	20
EBMD+Fast	0.6437	0.6124	0.5567	0.4943	0.4148
MAC	0.6392	0.5319	0.4065	0.3783	0.3582

TABLE 5
Comparison with MAC w.r.t Error Rate

k	4	8	12	16	20
EBMD+Fast	0.1115	0.1654	0.0937	0.0779	0.0863
MAC	0.0986	0.1698	0.1027	0.0806	0.1116

TABLE 6
Robustness of Alternating Minimization Heuristic

k	4	8	12	16	20
Noisy Data	0.082	0.093	0.101	0.125	0.130
Cleaned Data	0.081	0.095	0.113	0.117	0.127

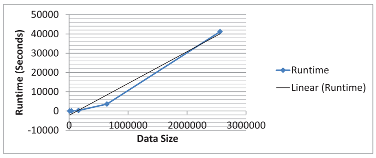


Fig. 8. Computing time.

TABLE 7
Reconstruction Error Comparison for $HealthCare$

Number of Roles	2	4	6	8	10	12	14
FastMiner	133	73	40	25	15	8	4
EBMD+Fast	133	73	40	25	15	8	4
Conservative EBMD+Fast	133	73	40	25	15	8	4
Loc&IterX	699	463	163	116	106	106	73
EBMD + Loc	491	325	115	115	92	75	51
Conservative BMD + Loc	491	325	115	115	92	75	51

reported in Table 4. The observation is that when k is 4, the performances of our EBMD approach and MAC are comparable, while when k is a large number, our EBMD approach performs significantly better than MAC.

We also compare our algorithm with MAC with respect to the error rate metric of (7). The results are reported in Table 5. We observe that when k increases, the advantage of our approach over the MAC becomes more evident. However, when k is a small number, the performance of our approach and the MAC is comparable. One possible explanation is that when the k is small, the internal structure of the data set is simple and hence a complex data description model has no advantage. Note that while the results of our approach are better in terms of quality as compared to the MAC approach, this comparison may not be completely unbiased, since the solution domain for MAC is different than EBMD, as well as the way in which the data are generated. However, theoretically, our EBMD model should always better describe a binary data set than the MAC model, since it is more general (i.e., it can also employ negative authorizations), and therefore has more descriptive power. Of course, in practice the security administrator would have to separately judge whether the negative authorizations returned by our model correctly reflect the underlying access control policy based on the semantics of the data set.

Experiment 5. The fifth experiment is to test the robustness of our alternating minimization heuristic by comparing it with some prior data clearing technique for access control data studied in [23]. According to [23], the singular value decomposition approximation has good performance in cleaning data noise. The experimental setting is as follows: We generate five binary matrices A according to the above-described data generation procedures with k ranging from 4 to 12 and add noise by randomly flipping 10 percent of their entries. To test the robustness of our approach, the $EBMD + Fast$ algorithm is employed to decompose the noisy binary matrix and then reconstruct the original binary matrix with the decomposed EBMD solution. Then, the same procedure is performed once again, while this time the noisy binary matrix is replaced with its rank k approximation matrix obtained through singular value decomposition. Note that a rank-k SVD approximation does not necessarily reconstruct the original binary matrix, as the EBMD rank is not equal to the real rank. There has been some study with respect to the relations of the BMD rank and the real rank in [24]. They showed that the BMD rank could be much larger than the real rank. As the BMD product is a special case of the EBMD product, so the EBMD rank could be much larger than the real rank as well. We use the same error rate metric as in (7). Experimental results are recorded in Table 6. We observe that the results on the noisy data and the cleaned data are comparable. This shows that our alternating minimization approach is robust to noise.

Experiment 6. The sixth experiment studies the scalability of our algorithms. We consider the representative $EBMD + Fast$ algorithm. Five binary matrices are generated. We let m and n be equal and vary them from 100 to 1,600. So, the data size varies from 10,000 to 2,560,000. k varies from 20 to 320. The other parameter settings are the same as the first experiment. The results are reported in Fig. 8. We observe that the computing time grows almost linearly with the size of the matrix. However, the algorithm takes about 11 hours to analyze a matrix of size $1{,}600 \times 1{,}600$, while in reality the data size could be much larger. Parallelization should help with this, as should the use of techniques recently developed to make role engineering more scalable[31].

6.2 Real Data

Three real data sets $HealthCare$, $Fire1$ and apj are studied. All of them are collected by Ene et al. [8]. $HealthCare$ is 46×46 and $Fire1$ is 365×709. The total permission assignments in $HealthCare$ are 1,486. The total permission assignments in $Fire1$ are 31,951. apj is $2{,}044 \times 1{,}164$. The total permission assignments are 6,841.

Consider $HealthCare$ first. We ran the $FastMiner$ algorithm and the $Loc\&IterX$ algorithm by varying the number of roles from 2 to 14, and obtain their solutions. Then, we take those solutions as the starting solutions for our EBMD and conservative EBMD algorithms. Their assignment errors are reported in Tables 7 and 8. The $HealthCare$ results in Table 7 show that the performance of $FastMiner$ is better than that of the $Loc\&IterX$ algorithm. So, we only looked at $FastMiner$, $EBMD + Fast$, and $ConservativeEBMD + Fast$. It turned out that their reconstruction errors are the same. We took a

TABLE 8
Reconstruction Errors w.r.t Role Number for $Fire1$

Number of Roles	5	15	25	35
FastMiner	2184	437	185	94
EBMD+Fast	2076	416	185	91
Conservative EBMD+Fast	2076	416	185	91
Loc&IterX	7692	1632	716	201
EBMD + Loc	7320	1332	656	201
Conservative BMD + Loc	7549	1332	656	201

TABLE 9
Reconstruction Errors w.r.t Role Number for apj

Number of Roles	100	200	300	400
FastMiner	1496	1007	650	264
EBMD+Fast	1424	976	620	233
Conservative EBMD+Fast	1424	976	620	233
Loc&IterX	1847	1232	893	294
EBMD + Loc	1640	1129	876	287
Conservative BMD + Loc	1640	1129	876	287

further look at their respective decomposed matrices. They are also the same. This implies that the $HealthCare$ data set does not really require negative assignments.

For $Fire1$, we vary the role number from 5 to 35. The results are reported in Table 8. Again the performance of the $Loc\&InterX$ algorithm is not as good as other approaches. Let us look at $FastMiner$, $EBMD+Fast$, and $ConservativeEBMD+Fast$. Reconstruction errors of $EBMD+Fast$ and $ConservativeEBMD+Fast$ are less than those of $FastMiner$. In other words, negative assignments are utilized in both cases to reduce reconstruction errors. When the role number is five, reconstruction errors are around 2,000, which is a large number of errors as opposed to the total permission assignments of 31,951. So, the EBMD solution with the role number of five are not very useful or convincible. However, when the role number is 35, errors have been reduced to less than 100, which is negligible as opposed to 31,951. We check the $EBMD+Fast$ and $ConservativeEBMD+Fast$ solutions with the role number of 35. In both of which, negative assignments are utilized. There is a high potential that such those two EBMD solutions suggest some underlying data rules and deserve some further examination. Unfortunately, the $Fire1$ data set contains only the user-to-permission assignments without business information on permissions and users, so we are not able to make further examination. But this real example demonstrates the usage of the EBMD model in discovering roles and SoD or exception constraints in a role mining process.

Finally, we study apj and vary the role number from 100 to 400. The results are reported in Table 9. It again confirms the conclusion that the EBMD model can better describe the binary data set than the BMD model since it has less reconstruction errors.

7 Related Work

One of the major challenges in implementing RBAC is to define a complete and correct set of roles, known as *role engineering* [7], which has been identified as one of the costliest components in realizing RBAC [12]. There have been several attempts to propose good bottom-up techniques to finding roles. Kuhlmann et al. [16] present a clustering technique similar to the well known k-means clustering, which requires predefining the number of clusters. In [26], Schlegelmilch and Steffens propose an agglomerative clustering-based approach to role mining (called ORCA). Vaidya et al. [30] propose an approach based on subset enumeration, called RoleMiner. Molloy et al. [23] propose to clean noisy data before executing role mining. Streich et al. [27] present a multiassignment clustering approach for Boolean data. It is a probabilistic approach and can be applied to the role mining problem. Some role mining approaches like [6], [11] suggest to discover roles by taking into account of available business information on permissions and users.

Several criteria or metrics have been used to evaluate the goodness of a candidate role set in the literature. In [30], Vaidya et al. suggest three criteria. The first one is known as the basic role mining problem, which is to minimize the number of roles required to cover the whole existing permission assignments. The second one relaxes the first one by allowing certain amount of errors. The third one is given the number of roles to minimize the number of roles. In [30], Vaidya et al. introduce another criteria, which is to minimize the administrative cost of the resultant RBAC system. In [21], Molloy et al. propose a notion called weighted structural complexity that combines some of the above metrics and assigns them some weights. In [17], Lu et al. even formulate some role mining variants through mixed integer program, which greatly simplifies the role mining work. Several works including [10], [30], and [17] allow errors in the role mining solutions. In those works, different error types, overassignments and underassignments are treated equally. Some recent works such as [22] point out that it is safer to under assign permissions than over assign permissions.

The BMD problem originally comes from the discrete database problem [24]. The tilling database problem [13] can be formulated as the BMD problem as well. Vaidya et al. [29] proves the equivalence of the RMP, the discrete database problem, and the tiling database problem. One-dimensional BMD is studied by Lu et al. in [19]. In [18], we have first presented the "Extended Boolean Matrix Decomposition" to solve the text mining problem. In this paper, we have extended this work to address the novel problem of constraint-aware role mining.

8 Conclusions

In this paper, we proposed and studied a novel problem, constraint-aware role mining, which takes the awareness of constraints into account in the role mining process. To deal with the constraint-aware role mining problem, we introduced a new model, EBMD. It extends the concept of BMD by allowing negative elements in one of the decomposed matrices. A good EBMD decomposition solution can discover a set of roles, which well constitutes the original permission assignments and also reveals embedded SoD or exception constraints to better meet the business needs of an organization. Complexity of the EBMD problem and its subproblems is carefully studied and efficient and effective algorithms are proposed for them. Experimental results validate the effectiveness of our approach.

REFERENCES

[1] M.A. Al-Kahtani and R. Sandhu, "Rule-Based RBAC with Negative Authorization," *Proc. 20th Ann. Computer Security Applications Conf. (ACSAC '04),* pp. 405-415, 2004.

[2] E. Bertino, P. Samarati, and S. Jajodia, "Authorizations in Relational Database Management Systems," *Proc. First ACM Conf. Computer and Comm. Security,* pp. 130-139, 1993.

[3] E. Bertino, P. Samarati, and S. Jajodia, "An Extended Authorization Model for Relational Databases," *IEEE Trans. Knowledge and Data Eng.,* vol. 9, no. 1, pp. 85-101, Jan./Feb. 1997.

[4] P.J. Besl and H.D. McKay, "A Method for Registration of 3-d Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 2, pp. 239-256, Feb. 1992.

[5] R.D. Carr, S. Doddi, G. Konjevod, and M. Marathe, "On the Red-Blue Set Cover Problem," *SODA '00: Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms,* pp. 345-353, 2000.

[6] A. Colantonio, R. Di Pietro, A. Ocello, and N. Vincenzo Verde, "A Formal Framework to Elicit Roles with Business Meaning in RBAC Systems," *Proc. 14th ACM Symp. Access Control Models and Technologies (SACMAT '09),* pp. 85-94, 2009.

[7] E.J. Coyne, "Role Engineering," *RBAC '95: Proc. First ACM Workshop Role-Based Access Control,* p. 4, 1996.

[8] A. Ene, W. Horne, N. Milosavljevic, P. Rao, R. Schreiber, and R.E. Tarjan, "Fast Exact and Heuristic Methods for Role Minimization Problems," *SACMAT '08: Proc. 13th ACM Symp. Access Control Models and Technologies,* pp. 1-10, 2008.

[9] D.F. Ferraiolo, R. Sandhu, S. Gavrila, D. Richard Kuhn, and R. Chandramouli, "Proposed Nist Standard for Role-Based Access Control," *ACM Trans. Information and System Security,* vol. 4, pp. 224-274, Aug. 2001.

[10] M. Frank, D. Basin, and J.M. Buhmann, "A Class of Probabilistic Models for Role Engineering," *Proc. 15th ACM Conf. Computer and Comm. Security,* 2008.

[11] M. Frank, A.P. Streich, D. Basin, and J.M. Buhmann, "A Probabilistic Approach to Hybrid Role Mining," *Proc. 16th ACM Conf. Computer and Comm. Security (CCS '09),* pp. 101-111, 2009.

[12] M.P. Gallagher, A. O'Connor, and B. Kropp, "The Economic Impact of Role-Based Access Control," Planning Report 02-1, Nat'l Inst. of Standards and Technology, Mar. 2002.

[13] F. Geerts, B. Goethals, and T. Mielikainen, "Tiling Databases," *Proc. Int'l Conf. Discovery Science,* pp. 278-289, 2004.

[14] S. Jajodia, P. Samarati, M. Luisa Sapino, and V.S. Subrahmanian, "Flexible Support for Multiple Access Control Policies," *ACM Trans. Database Systems,* vol. 26, pp. 214-260, June 2001.

[15] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu, "An Efficient k-Means Clustering Algorithm: Analysis and Implementation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 7, pp. 881-892, July 2002.

[16] M. Kuhlmann, D. Shohat, and G. Schimpf, "Role Mining—Revealing Business Roles for Security Administration Using Data Mining Technology," *SACMAT '03: Proc. Eighth ACM Symp. Access Control Models and Technologies,* pp. 179-186, 2003.

[17] H. Lu, J. Vaidya, and V. Atluri, "Optimal Boolean Matrix Decomposition: Application to Role Engineering," *Proc. IEEE 24th Int'l Conf. Data Eng.,* pp. 297-306, 2008.

[18] H. Lu, J. Vaidya, V. Atluri, and Y. Hong, "Extended Boolean Matrix Decomposition," *Proc. IEEE Int'l Conf. Data Mining,* 2009.

[19] H. Lu, J. Vaidya, V. Atluri, H. Shin, and L. Jiang, "Weighted Rank-One Binary Matrix Factorization," *Proc. SIAM Int'l Conf. Data Mining (SDM),* pp. 283-294, 2011.

[20] P. Miettinen, "The Boolean Column and Column-Row Matrix Decompositions," *Data Mining Knowledge Discovery,* vol. 17, no. 1, pp. 39-56, 2008.

[21] I. Molloy, H. Chen, T. Li, Q. Wang, N. Li, E. Bertino, S. Calo, and J. Lobo, "Mining Roles with Semantic Meanings," *SACMAT '08: Proc. 13th ACM Symp. Access Control Models and Technologies,* pp. 21-30, 2008.

[22] I. Molloy, N. Li, T. Li, Z. Mao, Q. Wang, and J. Lobo, "Evaluating Role Mining Algorithms," *Proc. SACMAT '09: 14th ACM Symp. Access Control Models and Technologies,* pp. 95-104, 2009.

[23] I. Molloy, N. Li, Y. (Alan) Qi, J. Lobo, and L. Dickens, "Mining Roles with Noisy Data," *Proc. 15th ACM Symp. Access Control Models and Technologies (SACMAT '10),* pp. 45-54, 2010.

[24] M. Pauli, M. Taneli, G. Aristides, D. Gautam, and M. Heikki, "The Discrete Basis Problem," *IEEE Trans. Knowledge and Data Eng.,* vol. 20, no. 10, pp. 1348-1362, Oct. 2008.

[25] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," *Computer,* vol. 29, no. 2, pp. 38-47, Feb. 1996.

[26] J. Schlegelmilch and U. Steffens, "Role Mining with Orca," *SACMAT '05: Proc. 10th ACM Symp. Access Control Models and Technologies,* pp. 168-176, 2005.

[27] A.P. Streich, M. Frank, D. Basin, and J.M. Buhmann, "Multi-Assignment Clustering for Boolean Data," *Proc. 26th Ann. Int'l Conf. Machine Learning (ICML '09),* pp. 969-976, 2009.

[28] P.-N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining,* first ed. Addison-Wesley Longman Publishing Co., Inc., 2005.

[29] J. Vaidya, V. Atluri, and Q. Guo, "The Role Mining Problem: Finding a Minimal Descriptive Set of Roles," *Proc. ACM Symp. Access Control Models and Technologies (SACMAT),* pp. 175-184, 2007.

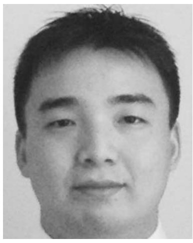
[30] J. Vaidya, V. Atluri, and J. Warner, "Roleminer: Mining Roles Using Subset Enumeration," *Proc. 13th ACM Conf. Computer and Comm. Security,* pp. 144-153, 2006.

[31] N. Verde, J. Vaidya, V. Atluri, and A. Colantonio, "Role Engineering: From Theory to Practice," *Proc. Second ACM Conf. Data and Application Security and Privacy,* 2012.

Haibing Lu received the undergraduate and master's degrees in mathematics from Xi'an Jiaotong University, China, in 2002 and 2005, respectively, and the doctoral degree in management from Rutgers University in 2011. He is currently an assistant professor in the Department of Operations and Management Information Systems, Santa Clara University. His research interests include data mining, privacy, and security. He is a member of the ACM and the IEEE Computer Society.

Jaideep Vaidya received the bachelor's degree at the University of Mumbai and master's and PhD degrees at Purdue University. He is an associate professor at Rutgers University. His research interests are in privacy, security, data mining, and data management. He has published more than 60 papers in international conferences and archival journals, and has received three best paper awards from the premier conferences in data mining, databases, and digital government. He is also the recipient of US National Science Foundation (NSF) Career Award and a Rutgers Board of Trustees Research Fellowship for Scholarly Excellence. He is a member of the IEEE Computer Society.

Vijayalakshmi Atluri is a professor of computer information systems in the MSIS Department, and research director for the Center for Information Management, Integration and Connectivity (CIMIC) at Rutgers University. She is currently a program director at US National Science Foundation (NSF) in the Information and Intelligent Systems division. Her research interests include information security, privacy, databases, workflow management, spatial databases, multimedia and distributed systems. She has published extensively in such journals and conferences as the *ACM Transactions on Information Systems Security, IEEE Transactions on Knowledge and Data Engineering (TKDE), IEEE Transactions on Dependable and Secure Computing (TDSC), The VLDB Journal,* ACM Conference on Computer and Communication Security, IEEE Symposium on Security and Privacy, IEEE Conference on Data Engineering. She currently serves as the vice-chair for the ACM Special Interest Group on Security Audit and Control (SIGSAC), and the chair of the IFIP WG11.3 Working Conference on Database Security. Currently, she is on the editorial board of *IEEE TDSC*, *Journal of Computer Security, Computers & Security, International Journal on Digital Libraries and International Journal of Information and Computer Security*. In the past, she served as the associate editor for the *IEEE TKDE*. She was the recipient of the National Science Foundation CAREER Award, and the Rutgers University Research Award for untenured faculty for outstanding research contributions. She is a senior member of the IEEE Computer Society and a member of the ACM.



Yuan Hong received the BSc degree in management information systems from Beijing Institute of Technology, China in 2004, and the MSc degree in computer science from Concordia University, Canada in 2008. He is currently working toward the PhD degree in information technology in the Department of Management Science and Information Systems, Rutgers University. His research interests include data privacy, secure distributed computation, data mining and optimization.