Flaws and Remedies of the MIDD Approach

Xiaofeng Luo\(^1\), Seng Wang\(^2\), and Wanbo Luo\(^3,\ast\)

\(^1\)Dept. of Computer Science and Technology, Sichuan Police College, Luzhou City, Sichuan, China
\(^2\)Chengdu ICT Co., Ltd. China Mobile, Chengdu, China
\(^3\)Computer College, Sichuan University, Chengdu, China

Abstract. We found that the Multi-data-type Interval Decision Diagrams (MIDD) approach proposed by Ngo et al. has some flaws: the way in which the critical mark of an attribute is represented does not retain the critical information, definition of the internal node is not entirely correct, and there are bugs in the corresponding algorithms. The first flaw could lead to potential missing attribute attacks, the second one wastes resources, and the last one makes the decision diagrams inconsistent. Two solutions were proposed to fix the first flaw. The first solution is to expand the scope of representation of internal nodes from only normal attributes of subject, object, operations, and environmental conditions to other required elements so that the critical attribute marks, obligations and/or advices, etc. are also represented by nodes. The second solution is to bind the critical mark of an attribute to its value. For the second flaw, corresponding definition was revised. For the third problem, a debug suggestion was made. The corresponding decision diagrams showed that the proposed schemes can overcome these problems of the MIDD approach.

1 Introduction

Access control is the most basic information security technology, and access control policy is the basis for decision-making. In the numerous researches on access control policy evaluation and management, the Multi-data-type Interval Decision Diagram (MIDD) approach proposed by Ngo et al., which contains both the analysis of the XACML logic and practical implementation mechanisms, worked quite well.

Unfortunately, some problems were found with the MIDD approach, which make it impossible to retain the critical marks of attributes. In addition, its basic definition has minor errors, and related algorithms also have bugs.

Through the in-depth study, three defects and deficiencies that affect the practical applications of the MIDD approach were singled out, and corresponding solutions were proposed.

2 Problems and analysis

An abbreviated access control policy example, which is shown in Listing 1 and was used in (Ngo et al., 2015), is still used in this article for the convenience of presentation and discussion.
Listing 1. Access policy example (Ngo et al., 2015)

Policy { id: \( P_1 \); combine_algo: po; 
target: \((vol \geq 100) \land (vol \leq 500)\); 
children: \{\( R_1, R_2 \) \}
}

Rule { id: \( R_1 \); effect: Permit; 
target: 
\[
[(100 \leq vol \leq 150) \land (12 \leq t \leq 17) \land (3 \leq p \leq 4)] \lor 
[(300 \leq vol \leq 500) \land (1 \leq p \leq 2)] \lor 
[(100 \leq vol \leq 500) \land (6 \leq t \leq 9) \land (1 \leq p \leq 2)]
\]; 
obligations: \{O_1, Permit\} 
}

Rule { id: \( R_2 \); effect: Deny; 
target: 
\[
[(vol = 100) \land (t = 17)] \lor 
[(100 \leq vol \leq 300) \land (t = 9)] \lor 
[(vol = 500) \land (t \geq 12)]
\]; 
obligations: \{O_2, Deny\} 
}

In Listing 1, combine_algo is the abbreviation of combining-algorithm, and po stands for combining-algorithm permit-overrides.

In the rest of this article, if not specified, symbols defined in (Ngo et al., 2015) are used in the same way, and XACML refers to XACML-v3.0-Errata01-complete.

2.1 Problem 1

In the rule R1, a vol (volume attribute) is underlined as critical. The remaining vols, as well as all t (time attribute) and p (price attribute), are not underlined, indicating that they are not critical attributes.

According to the Definition 3 in (Ngo et al., 2015), for an internal node of the MIDD, if x is marked as critical, then the state value \( s = \text{IN} \) (“indeterminate”); otherwise \( s = \text{F} \) (“false” or “No-matched”).

Figure 1 is drawn as defined. Figure 1a corresponds to
\[
100 \leq vol \leq 150 \land 12 \leq t \leq 17 \land 3 \leq p \leq 4,
\] (1)
and Figure 1b to
\[
300 \leq vol \leq 500 \land 1 \leq p \leq 2,
\] (2)
and Figure 1c to
\[
100 \leq vol \leq 150 \land 6 \leq t \leq 9 \land 1 \leq p \leq 2.
\] (3)
Figure 1. MIDDs of $R_1$ target expressions (Ngo et al., 2015).

Node vol(IN) of Figure 1a indicates that vol is a critical attribute; Nodes vol(F) in both Figure 1b and 1c indicate that they are not critical.

Figure 2 is the MIDD of the Rule $R_1$ after combining the MIDDs of Eq.(1), (2) and (3). Note that in Figure 2 the vol node is vol(F), and the critical information of the attribute vol in Figure 1a has been lost.

Figure 2. MIDD of the rule $R_1$.

Obviously, the way to represent the critical mark of an attribute in the MIDD loses the critical information.

Why is this happening after combining the MIDDs?

According to (Ngo et al., 2015), when the node vol(IN) is joined by another node vol(F), the lattice order rule $F \leq \text{IN} \leq T$ is applied so that a low-order vol (F) node is formed, so that the critical mark information is lost. That is what happens in Figure 2.

Figure 3 is the X-MIDDs of rules $R_1$ and $R_2$, where N in the nodes means “Not Applicable”. It is similar to the MIDDs that the critical mark is also missed in the joined X-MIDDs.
Figure 3. X-MIDDs of the rules $R_1$ and $R_2$ (Ngo et al., 2015).

The reason for this problem in X-MIDD is completely same to MIDD. Actually, in XACML MustBePresent = “True” is used to indicate that an attribute is critical. The attribute MustBePresent is a mandatory property. Losing it could affect the correctness and completeness of XACML, and in serious cases, could lead to potential missing attribute attacks (Ngo et al., 2015; 2013; XACML-v3.0-Errata01-complete, 2017).

Therefore, for the practical applications of access control, losing critical mark is a serious problem.

2.2 Problem 2

According to Definition 4 in (Ngo et al., 2015), an internal node of X-MIDD is defined as a tuple $(x, s, O, C)$. Among them, the state $s \in \mathcal{VR}:=\{P, D, N, INP, IND, INPD\}$; $O$ contains list of obligations and advices matching $s$ if $s \in \{P, D\}$, otherwise it is empty.

Note that $P$ means "Permit"; $D$ means "Deny"; $N$ means "NotApplicable"; $INP$ means that it cannot be determined and may be assessed as "Permit" rather than "Deny"; $IND$ means that it cannot be determined and may be assessed as "Deny" rather than "Permit"; $INPD$ means it cannot be determined and may be assessed as "Permit" or "Deny".

Here $s \in \mathcal{VR}$ is wrong. It only satisfies $s:=\{N, INP, IND, INPD\}$, and $O$ is redundant.

A brief proof is given as below.

Assume that the internal node variable is $x$, the state of the MIDD node is $s_m$, and the state of the X-MIDD node is $s_x$.

According to the Definition 3 and 4 in (Ngo et al., 2015), $s_m \in \{F, IN\}$, $s_x \in \mathcal{VR}:=\{P, D, N, INP, IND, INPD\}$.

Suppose that we have $s_x=P$, according to the rules of MIDD to X-MIDD transformation in section 5.3 (Ngo et al., 2015), for each internal node $m=(x, s_m, C)$, the state $s_m=\{F, IN\}$ is mapped to $\mathcal{VR}$ as follows:

- If $s_m=F$, the new state $s_x$ is $N$.
- If $s_m=IN$, the new state $s_x$ is $INe$ with $e:=\{P, D\}$, where $e$ is the policy's effect.
- Since $s_x=P$, it requires $s_m \notin \{F, IN\}$, which contradicts the above transformation rules.
- Similarly, $s_x=D$ requires $s_m \notin \{F, IN\}$, which also contradicts the above rules.
- Therefore $s_x \notin \{P, D\}$.

Conclusion: Definition of $s_x \in \mathcal{VR}:=\{P, D, N, INP, IND, INPD\}$ is wrong. And the element $O$ of each X-MIDD internal node is redundant because it is always empty.

In practical applications, redundant components, which are included in the nodes, result in the waste of storage and computation resources, therefore reducing efficiency.
2.3 Problem 3

The X-MIDD formation algorithm has bugs that make the X-MIDD internal nodes inconsistent.

The analysis in Problem 2 explains that the node \( s \notin \{P, D\} \) in X-MIDD, and element O is always empty. Since the empty elements are excluded, the \((s, O)\) of each X-MIDD internal node in Figure 3 is reduced to \((N)\), and \(N\) is the actual state.

Figure 4 shows the X-MIDD of policy P1. Note that both the left price node and the right price node in Figure 4 are represented as \((D, O_2)\), and the remaining internal nodes, including the middle two price nodes, are \((N)\). The inconsistency between the internal nodes is obvious.

The inconsistency among the internal nodes also existed in the original document (Ngo et al., 2013).

Figure 4. X-MIDD of the policy \(P_1\) (Ngo et al., 2015).

The inconsistency demonstrated in Figure 4 is a problem that may happen either in the MIDD formation process, or in the MIDD to X-MIDD transformation process. Since the original publication did not provide the MIDD of the policy P1, it is impossible to determine which part of the algorithms has bugs.

3 Solutions

As analyzed above, Problem 1 is serious and must be fixed. Problem 2 wastes resources in practical applications and it can be solved by simply revising the X-MIDD definition appropriately. Problem 3 is related to the algorithm bugs and can be fixed by debugging, so it is not discussed herein.

3.1 Internal node expansion as a solution to problem 1

The internal nodes in both MIDD and X-MIDD represent the attributes of subjects, objects (resources), operations, and environment conditions. For the purpose of narration and distinction, these attributes are called normal attributes hereafter. Other attributes or elements, such as MustBePresent in XACML (the critical mark in R1), are a sort of
supplements to the aforementioned normal attributes, and they can neither be represented in MIDD nor in iX-MIDD.

In addition, elements such as Obligations, Advices, and combine-algorithm, etc. do not count as attributes as well.

The basic idea is that the internal nodes of MIDD and X-MIDD, which express the normal attributes, are extended to express the elements. In this way, MustBePresent and obligations and/or advices are also represented as internal nodes. However, combining-algorithm is an exception because it is already used when forming MIDD and X-MIDD.

Since the internal node is extended, it is necessary to identify its kind. Therefore, the node type must be a member of the internal node tuple.

For MIDD internal nodes, the tuples (x, s, C) is expanded to (x, s, k, C), and for the X-MIDD internal nodes, the tuples (x, s, O, C) expanded to (x, s, O, k, C). Here, k:={ Normal, Supplement, Obligation&Advice, ...}, or k:={1, 2,3, ...}

For access policy example, it is assumed that the critical mark of vol in Eq. (1) is represented by the attribute flag. According to this scheme, the improved approach treats $100 \leq \text{vol} \leq 150$ as $(100 \leq \text{vol} \leq 150 \land \text{vol. flag} = \text{True})$, or abbreviated as $(100 \leq \text{vol} \leq 150 \land \text{flag} = \text{True})$. And $100 \leq \text{vol} \leq 150$ is equivalent to $100 \leq \text{vol} \leq 150 \land \text{vol. flag} = \text{False}$. Usually vol.flag = False can be omitted.

The improved MIDDs (iMIDDs) of the rule R1 target expressions and the iMIDD of the rule R1 are shown in Figure 5 and Figure 6, respectively. The letters and numbers enclosed in parentheses in the figures correspond to s and k respectively.

Figure 5. iMIDDs of the rule $R_1$ target expressions.

Figure 6. iMIDD of the rule $R_1$. 

136
Comparing with Figure 2, in Figure 6 the critical attribute information of vol in R1 is completely retained.

Due to the addition of members, the processing of match-evaluation and decision-evaluation should also be modified appropriately. The details are not described here.

### 3.2 Solution to Problem 2

To solve this problem, it is necessary to correct the errors and remove the redundant components.

Regarding to Scheme 1, the corresponding definition of the internal node in X-MIDD is modified as follows:

An internal node of the improved X-MIDD (iX-MIDD) is defined by the tuple \((x, s, k, C)\), where \(x\) and \(C\) are the same as X-MIDD; \(S \in \{N, INP, IND, INPD\}\); \(k\) is the node category, and \(k \in \{\text{Normal, Supplement, Obligation&Advice, ...}\}\), corresponding to normal attribute node, supplement attribute node, responsibility and advice node, etc. respectively.

Since obligation and advice are also represented by internal nodes, the external node of X-MIDD is also modified as follows:

An external node of iX-MIDD contains the effect value \(e \in \{P, D\}\) of the rules, policies, and policysets. The external node is also called the decision-leaf node.

Similarly, the processing of decision-evaluation should also be modified appropriately.

The iX-MIDD of the policy \(P1\) is shown in Figure 7. The letters and numbers enclosed in the brackets correspond to \(s\) and \(k\) respectively, and the numbers above the variable names are the serial numbers of the internal nodes for convenience.

Compared with Figure 4, the biggest change in Figure 7 is that the critical mark information is retained. In Figure 7, such information is stored in the following node-edge paths:

1. \(\text{vol} \rightarrow [100] \rightarrow \text{flag} \rightarrow [\text{True}] \rightarrow 6 \text{ time}\),
2. \(\text{vol} \rightarrow (100,150) \rightarrow \text{flag} \rightarrow [\text{True}] \rightarrow 6 \text{ time}\),
3. \(\text{vol} \rightarrow [150] \rightarrow \text{flag} \rightarrow [\text{True}] \rightarrow 8 \text{ time}\).

![Figure 7. iX-MIDD of the policy \(P_1\).](image)

### 3.3 Another solution to problem 1

Another solution to Problem 1 is to bind critical mark to value.

137
It is necessary to revise the tuple \((p, c)\) of \(C\), which is a member of the MIDD internal node tuple \((x, s, C)\), to be a tuple \((p, \text{flag}, c)\), and if \(x\) is marked, \(\text{flag} = "True"\), else \(\text{flag} = "False"\). The rest remains unchanged.

When making match-evaluation or decision-evaluation, it only needs to compare with \(p\), and note the information of \(\text{flag} = "True"\).

The X-MIDD definition needs to be modified similarly. The iX-MIDD of the policy \(P1\) is shown in Figure 8. Since the scheme of section II.A is retained in dealing with Obligation and Advice, \(k\) is also added to the internal node tuple. Similarly, the letters and numbers enclosed by brackets in Figure 8 correspond to \(s\) and \(k\), and the numbers above the variable names are the internal node serial numbers.

In Figure 8, the critical mark information is stored in the following node-edge paths:

1. \(vol \rightarrow [100] \rightarrow 6\) time,
2. \(vol \rightarrow (100,150) \rightarrow 6\) time,
3. \(vol \rightarrow [150] \rightarrow 8\) time.

It is noteworthy that this scheme only works to solve problem 1.

\[\text{Figure 8. Another iX-MIDD represents the policy } P_1.\]

4 Conclusion

In this research we investigated some problems with the MIDD approach and provided feasible solutions. We found that the MIDD approach cannot preserve critical mark information. In addition, the definition of internal nodes in X-MIDD was found incomplete and lack of consistency.

For the first problem, two solutions were proposed. In solution 1, the internal node is extended to represent both element and normal attribute. MustBePresent, Obligation and advice, etc., all can be represented by nodes. This method successfully solves the first problem, and could be applied to general attribute-based access control (ABAC). In solution 2 the attribute critical mark is bound to its value, so the critical mark information is retained. We demonstrated with both iMIDDs and iX-MIDDs that both solutions can improve the MIDD approach.

We revised the X-MIDD internal node definition to solve the second problem. For the third one, debugging is the only solution.
References

1. Ngo, C., Demchenko, Y., & Laat, C. D. (2015). Decision diagrams for xacml policy evaluation and management. Computers & Security, 49(5), 1-16.

2. Ngo C., Makkes MX, Demchenko Y, Laat C. D. (2013). Multi-data-types interval decision diagrams for xacml evaluation engine. In: Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on. IEEE; 2013. pp. 257-266.

3. XACML-v3.0-Errata01-complete.(2017).eXtensible Access Control Markup Language (XACML) Version 3.0 Plus Errata 01. Edited by Erik Rissanen. 12 July 2017. OASIS Standard incorporating Approved Errata.

4. http://docs.oasisopen.org/xacml/3.0/errata01/os/xacml-3.0-core-spec-errata01-os-complete.html. Latest version:

5. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-en.html.