

信息流图的分割算法^{*}

钱 炜 鞠时光 王昌达 宋香梅

(江苏大学计算机科学与通信工程学院 镇江 212013)

摘 要 一个多级安全系统的信息传递关系是搜索隐通道的重要依据。由于一个大型多安全级软件系统的信息传递关系非常复杂,对其直接进行信息流分析并搜索隐通道的工作量非常大,因此为进一步提高搜索和分析隐通道的效率,提出了一种将信息流图分割成信息流森林的算法。该算法首先选定一棵父树,然后通过边割集中的孤置换父树中的树枝得到新树,并能从理论上保证分割后得到的信息流森林正确继承原图的信息,即原图中信息流既未被割裂也未丢失。通过对每棵信息流树表示的信息流进行分析就可取代对整个信息流图的分析,从而有效降低了分析问题的规模。

关键词 信息流,隐通道,多级安全系统

Divide Algorithm Based on Information-flow Chart Build Forest

QIAN Wei JU Shi-guang WANG Chang-da SONG Xiang-mei

(School of Computer Science and Telecommunications Engineering, Jiangsu University, Zhenjiang 212013, China)

Abstract Information-flows of mains and objects in a Multi-level security system could be used to describe an information-flow chart. This information-flow chart is the important basis of covert-channel search. But the information-flow charts in big system are intricately. Therefore, we proposed a new segmentation algorithm based on information-flow chart to build forest. Segmentation algorithm needs to ensure that information-flow forest could inherit correct information from information-flow chart. First, arbitrary full path in information-flow chart is not fragmented. Second, arbitrary full path in information-flow could not miss. Last, algorithm is able to handle the cycle flow of information. Analyzing the information flow, we can replace the entire information-flow chart. So it could effectively lower the analysis of the complexity of the problem. The new model could be used in the field of covert-channel search analysis, and improve the efficiency.

Keywords Information-flow, Covert channel, Multi-level security system

1 问题的提出

隐通道利用系统中原本不是设计为传输信息的方式来传导信息,因此隐通道不能被安全模型所侦测,这对系统的安全造成了巨大的威胁。国际上对于具有一定可信等级的系统都要求通过相应强度的隐通道检测。提出适用于我国的隐通道检测技术也是目前迫切需要解决的问题。

迄今为止,国际上提出了许多隐通道检测方法,如信息流分析法^[1]、共享资源矩阵法^[2]、无干扰分析法^[3]等。其中,信息流分析法通过检查系统中的信息流,找出违反安全策略的信息流动,从而检测出系统中的隐通道^[1]。早期的信息流分析法都是基于语法的,在使用这一类信息流分析方法时,首先要将信息流附加在每条语句之后,以赋值语句 $a=b$ 为例,产生由 b 到 a 的信息流,用 $a \leftarrow b$ 表示;然后定义安全信息流策略,如果有信息流 $a \leftarrow b$,则 a 的安全级必支配 b 的安全级;接下来将策略应用于系统的形式化顶级描述或源代码,生成信息流公式,例如 $a=b$ 语句的流公式为 $SL(a) \geq SL(b)$,其中 $SL(a)$ 和 $SL(b)$ 分别表示变量 a 和 b 的安全级;最后证明信息流公式的正确性,如果无法证明某个流公式的正确性,则需要进一步对语句进行语义分析,来判断该信息流是否能够产生

真实的隐通道,而不只是潜在的伪隐通道^[1]。这种基于语法的信息流分析方法,要求对每个变量或客体都赋予特定的安全级,这将产生大量的伪非法流^[1],对于不能通过正确性证明的信息流,还需要辅助手工语义分析来消除伪隐通道,增加了很大的工作量;此外,这种方法在描述信息流时并没有考虑各信息流之间的关系,也就不能获得系统的信息流路径,很难构造出隐通道发生的场景,增加了确定隐通道位置和放置隐通道处理代码的难度。针对前一问题, Tsai 等人^[4]对信息流法作了重大改进,提出了一种新的语义信息流分析方法,通过增加语义分析,该方法可以发现大量的伪非法流。对于后一问题,则需要提供能准确描述系统中各信息流间关系的信息流模型,如信息流图^[5]。其中,信息流图是有向图,根据信息流语义以及信息流传递性和依赖性分析建立得到。在图中描述了系统中错综复杂的信息流向,图中的任意一条路径都代表了系统中的一条信息流通路。一个多级安全系统中的信息传递关系可以用一张信息流图表示。该信息流图是检查信息流向来搜索隐通道的重要依据^[6]。由此可以发现要分析系统中的隐通道首先要将系统的信息流分析清楚,包括显式和隐式的信息流。多条最小的信息流构成信息流通道^[7],以形式语义学为背景^[5],信息流通道的集合可以建立与待分析程序相

^{*} 本文受到国家自然科学基金(60573046,60773049),江苏省高校自然科学研究计划(07KJB520016)和江苏大学高级人才项目(07JDG053)的资助。

对应的有穷有向信息流网,其中描述主客体之间信息流动关系的信息流网称为信息流图。在信息流图中节点是信息传导的主体或者客体,有向弧表示在有向弧的前驱节点有信息传导到有向弧的后继节点。信息流图中两个节点以及联系他们的有向弧组成了最小的信息传导途径。

多级安全软件系统分析后得到的信息流图往往非常复杂,人工不易判读,这使得各种隐通道检测方法不能在信息流图的层面上直接运用于大中型系统。目前已提出的隐通道搜索方法在进行隐通道分析之前,仍需辅之以大量的人工分析信息流的工作^[8,9]。以一个十万行系统为例,其中一般会存在数百个信息流节点,即信息传导的主体或者客体,这就会造成非常复杂的信息流,包括很多循环信息流。直接在信息流图的层面上进行信息流分析,以其操作复杂度而言几乎是不可能完成的。因此,必须将分析信息流工作的复杂度降低,故需要将信息流图化简后使其适用于成熟的隐通道检测方法。因此,开发一个信息流图分割成信息流森林的算法就显得非常必要,当信息流图被分割成信息流森林后,单棵信息流树上的信息流节点比较少,并且没有循环信息流,故在信息流树的层面上分析信息流是可行的。

以图1为例,信息流图被分割为由3棵信息流树所组成的信息流森林。在单棵信息流树中从根节点到叶子节点的一条完整路径描述了图中一个完整的信息流。这样,信息流森林中叶子节点的总数目就是原信息流图中完整信息流的数目。在信息流树的层面上分析原系统隐通道时只需分析由信息流树根节点至信息流树叶子节点的完整信息流,如果某条完整信息流不能通过相应隐通道检测,则将此完整信息流单独提出详细分析即可。目前已有的图生成树技术并不适用于信息流图生成信息流森林。因为信息流图的分割有以下要求:①信息流图中任意信息流的完整路径不能被割裂。②信息流图中的任意信息流不能缺失。③算法要能处理循环信息流。本文主要开发了一种信息流图生成信息流森林的算法。

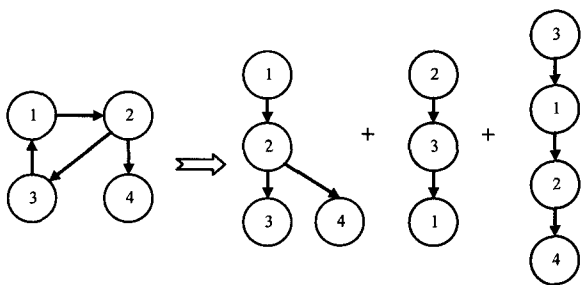


图1 信息流图分割成信息流森林

2 相关定义及符号

为了提供有效的信息流描述方法,可将程序系统中的信息传导机制看成节点,把节点间的信息传导关系表示成有向弧,这样就可以将程序系统抽象成一个有向有限的信息流脉络网,即信息流图^[5]。通过该方法所确定的信息流图能准确描述程序系统中信息流的走向。信息流图忽略了系统的实现细节,因而能够更加清晰地反映出系统中的信息流走向。在本文的研究中,信息流图表示为 G ,由信息流图得到的单棵树用 t 表示,树集即用 T 表示。

定义1 设 t_1 和 t_2 为图 G 中两棵不同的生成树,在 t_1 且

不在 t_2 中树枝的数目为 t_1 到 t_2 的距离。记为 $d(t_1, t_2)$ 。

定义2 设 S 是图 G 中关于 G 的生成树 t 的基本割集^[10], e 是 S 中所包含的 t 中的树枝,该基本割集用 $S_e(T)$ 表示。

定义3 设 t_0 和 t_i 为算法开始时和算法过程中图 G 的生成树。用属于 a_K 在 G 中关于 t_i 和 t_0 基本割集的交集中一条不为 a_K 的边 p 替换 a_K 后,由上一代树 $T_{a_1 a_2 \dots a_{K-1}}$ 生成新一代树 $T_{a_1 a_2 \dots a_K}$,引入更一般的符号表示为:

$$T_{a_1 a_2 \dots a_K} = \{t_i \oplus \{a_K, p\} \mid p \in S_{a_K}(t_i) \cap S_{a_K}(t_0), t_i \in T_{a_1 a_2 \dots a_{K-1}}, p \neq a_K\}$$

其中 t_i 属于上一代树。在特殊情况下,即由 t_0 生成第一代树时有: $T_a = \{t_0 \oplus \{a, p\} \mid p \in S_a(t_0), p \neq a\}$ 。

为了描述方便,本文使用如下一些符号:

- $tdeg^-(v), tdeg^+(v)$, 分别表示节点 v 在生成树 t 中的出度和入度;
- $Gdeg^-(v), Gdeg^+(v)$, 分别表示节点 v 在有向图 G 中的出度和入度;
- $tbf(a), taft(a)$, 分别表示有向弧 a 在生成树 t 中的前驱节点和后继节点;
- $Gbf(a), Gaft(a)$ 分别表示有向弧 a 在有向图 G 中的前驱节点和后继节点;
- 设 A 为一集合,获得 A 集合中的原子项的操作用 $ori(A)$ 表示。

3 信息流图分割成信息流森林的算法

结合相关的连通图和生成树的知识,本文通过用信息流图中树枝在信息流图中相应条件下基本割集里的弧替换后得到不同的信息流树,并设置一定的选择条件将信息流图分割成若干棵树,且尽量减小森林对图覆盖的冗余,实现分割算法。一个完整系统的信息流图一定可以表示为一个弱连通图。如果整个系统由若干个独立子系统构成,则单独考虑每个独立子系统,在信息流图的层面上仍表示为单个弱联通子图。对弱连通图实施下面的算法。

对于弱联通图 G 。 m 和 n 分别为图 G 中有向边的条数和顶点数。

- 利用传统方法任意找到图 G 的一棵生成树 t_0 。
- 依次选取一棵生成树 t_i 上的树枝 $e_j (j=1 \dots n-1)$,生成其在图中关于 t_i 和 t_0 的基本边割集 $S_{e_j}(t_i)$ 和 $S_{e_j}(t_0)$,依次用两个基本边割集交集集中的有向边替换 e_j ,得到一个生成树集合。其中替换的有向边要满足:用该边替换后的新生成树中,该边的后继节点入度为1,且新生成树与已有的生成树不能重复。对于原生成树集合中的每棵树,顺序执行替换操作,此时新生成的所有生成树组成新一代树。

• 以新一代树为基础继续生成新一代树,直到不能生成新一代树为止,或直到生成 $m-n+1$ 代树,其中 m 和 n 分别为原始弱联通图 G 中的有向边的条数和顶点数。事实上很多情况下生成的最后一代树中所有树的两个基本边割集交集为空,因此,生成新一代树的过程会在还没有生成到第 $m-n+1$ 代树时就已停止。另外因为所有生成树在没有剪枝前都拥有 $n-1$ 条边,而对 t_0 而言,遗漏了图 G 中的 $m-n+1$ 条边,同时考虑到相邻两代树中父子之间的距离为1,因此最多只需要考虑 $m-n+1$ 代树即可。

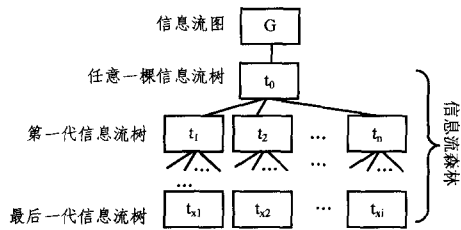


图2 生成关系树

• 为了记录以上步骤中产生的生成树的顺序以及各生成树之间的产生的(父子)关系,以便于第三部分的操作,将生成树集合构建一个生成关系树(见图2):令生成关系树的根节点为 t_0 ,生成关系树的中间节点为生成树 t_i ,由 t_i 生成的子树记为该节点的儿子节点。在图2中由树到树的直线都表示用基本边割集交集的边替换树枝后得到新生代树,图中最后生成到第 x 代树。

• 裁减生成树冗余的树枝。在生成关系树中从叶子节点回溯至根节点。子节点与其父节点相比较,删除其中与父树中相同的树枝。

上述算法可以概括为:首先任意确定一棵图的生成树;通过使用边割集中的弧在前代树的基础上生成后代树,直到不能生成新生代树或者生成到 $m-n+1$ 代树;在生成的森林中从最后一代树开始向前剪枝,减少森林中信息流路径的冗余度。算法的具体描述如下:

输入:弱联通图 $G(V, E)$;图 G 的节点数 n ;图 G 的边数 m 。

输出:树集 T 。

过程:

```

(1)  $t[0]$ 为图 $G$ 的任意一棵生成树;
(2)  $p=1$ ;
(3) for ( $k=0; k \leq m-n+1; k++$ ) {
    for ( $s=1; s \leq p; s++$ ) { /* 在第 $k$ 代树集  $T_{a_1 a_2 \dots a_k}$  中顺序取树  $t[s]$  */
        if ( $k=0$ ) then  $t[s]=t[0]$ ; /* 初始状态下令  $t[s]=t[0]$  */
        else  $t[s]$  = 第 $k$ 代树集  $T_{a_1 a_2 \dots a_k}$  中的第 $s$ 棵树;
        for ( $j=1; j \leq n-1; j++$ ) { /* 在  $t[s]$  中顺序取树枝  $a[j]$  */
             $a[j]$  = 树  $t[s]$  中的第 $j$ 个树枝;
             $B = \{b | b \in S_{a[j]}(t[s]) \cap b \in S_{a[j]}(t[0]) \cap b \neq a[j] \cap \text{tdeg}^-(\text{taft}(b)) < 1\}$ ;
             $u$  = 集合 $B$ 中元素的个数;
            for ( $c=1; c \leq u; c++$ ) {
                 $b[c]$  = 集合 $B$ 中的第 $c$ 个元素;
                 $T_{a_1 a_2 \dots a_{k+1}} = T_{a_1 a_2 \dots a_k} + (t[s] \oplus \{a[j], b[c]\})$ ;
                /* 由 $k$ 代树集中的  $t[s]$  生成新的一棵树加入第 $k+1$ 代树集中 */
            }
        }
        if ( $T_{a_1 a_2 \dots a_{k+1}} = \phi$ ) then break;
         $p$  = 第 $k+1$ 代树集  $T_{a_1 a_2 \dots a_{k+1}}$  中树的数目;
    }
}
(4) if ( $k=m-n+1$ ) then  $x=k$ ;
else  $x=k-1$ ; /* 设最后一代树集为第 $x$ 代树 */
(5) for ( $y=x; y \geq 1; y--$ ) { /* 从这一步起开始砍树 */
    for ( $s=1; s \leq p; s++$ ) { /* 在第 $y$ 代树集  $T_{a_1 a_2 \dots a_y}$  中顺序取树  $t[s]$  */

```

```

 $t[s]$  = 第 $y$ 代树集  $T_{a_1 a_2 \dots a_y}$  中的第 $s$ 棵树;
 $t[s]$  与上代中的父树比较,删除  $t[s]$  中与其父树相同的树枝;
}
(6)  $T = \{t[0], \text{ori}(T_{a_1}), \dots, \text{ori}(T_{a_1 a_2 \dots a_x})\}$ ;
}

```

4 算法性质及理论证明

在前文描述中算法需要达到3点要求,在这里通过对以下的3个性质的数学证明保证上述算法具有前文所讨论的3点要求。其中①、②点要求是直接由性质2和性质1及其证明保证。第③点要求将在性质2的证明过程中被讨论。

性质1 由算法生成的信息流树集 T 覆盖了算法中的信息流图 $G(V, E)$ 。

证明1:

设图 $G(V, E), \forall a \in E$ 。

如果, $a \in t_0$, 证毕。

讨论 $a \notin t_0$ 。

一定有 $G_{bef}(a) \in t_0$ 和 $G_{aft}(a) \in t_0$ 。

作边集 $S = \{r | G_{bef}(r) = G_{bef}(a) \cup G_{aft}(r) = G_{bef}(a), r \neq a\}$ 。

如果 $S = \phi$, 表明 $G_{bef}(a)$ 只连接有 a 一条边, 则必有 $a \in t_0$ 。证毕。

如果 $S \neq \phi$, 则 S 中一定有一项属于 t_0 , 不妨设为 r_i , 即: $S = \{r_1 \dots r_i \dots r_n\}$, 有 $r_i \in t_0$ 。

构成边集 $S' = S + \{a\}$, 有 S' 为 G 的割集。

这样, $S' = S' - r_i = S + \{a\} - \{r_i\}$ 为 G 的关于 t_0 的基本割集。这样, $a \in T_{r_i}$ 。

性质2 算法中信息流图 $G(V, E)$ 的每条完整的信息流在算法生成的生成树集 T 的一棵生成树上完整出现。

证明2:

\exists 图 $G(V, E)$, 其中 G 有 n 个点, m 条边, 有信息流如下: $a_1, a_2, \dots, a_k, a_i \in E$ 。

数学归纳法证明如下:

1) $k=1$, 则见证明1, 得证。

2) $1 < k \leq n-1$,

取此 k 条边和所有顶点, 扩展成生成树 t_k 。

不妨设 t_0 中有 t_k 的边数为 x , t_0 中没有 t_k 的边数为 y , 则有:

$$d(t_0 t_k) = d(t_k t_0) = n - 1 - x.$$

对于 G 的 t_0 , 需要余留 $m-n+1$ 条边, 因此, 有 $x+y = k$ 。 $y \leq m-n+1$

于是得到:

$$d(t_0 t_k) = d(t_k t_0) = n - 1 - (k - y).$$

最坏情况为:

$$d(t_0 t_k) |_{\text{MAX}(y)} = d(t_k t_0) |_{\text{MAX}(y)} = m - n + 1$$

此时, t_k 在生成树集 T 中。

3) $k > n-1$, 此信息流一定成环。

不妨设有信息流如下: $a_1, a_2, \dots, a_k, a_1$

环 $a_1, a_2, \dots, a_k, a_1$ 看成 k 个信息流如下:

$$a_2, a_3 \dots a_k, a_1, a_3 \dots a_k \dots a_1, a_2, \dots, a_{k-1}$$

这 k 条信息流, 根据上述证明可知, 它们将出现在 k 棵生成树上。同时这 k 条信息流的总体, 体现了信息流环的所有可能的节点间的信息交互。因此, 这样的信息流环的信息流含义也没有被破坏。

性质3 算法的生成树集 T 无重复项。

承,所以,第二个复杂度是不可以降低的。但是,在信息流图上寻找满足文章开始时三个要求的信息流是非常复杂的,而信息流图中的完整信息流在信息流森林中就是每个根节点到叶子节点的路径。因此,运用本文提出的算法后,第一个复杂度将会被大大降低。

信息流图生成信息流森林后,原图中的任意完整路径不能被割裂,我们对于算法第二个性质的证明不仅保证了信息流的完整性,还保证了每条完整的信息流一定会在某一棵生成树上完整地出现。对于信息流图中的任意信息流不能缺失这点。算法的第一个性质保证了信息流森林覆盖了原来的信息流图。这两条性质保证了算法的正确性。

在隐通道的检测中对于循环信息流的研究是必要的,因此在性质2的第3证明部分被详细讨论,最后得到的信息流森林正确继承了原信息流图中循环信息流所有可能的信息交互。为了减少后期分析的复杂度。笔者首先利用性质3及其证明保证了信息流森林中不会有相同的生成树。进一步我们提出了剪枝算法,这样大幅度减少了信息流森林的整体树枝数目。

运用上面给出的算法,我们可以实现对于信息流图信息的正确分割。上述的算法除了适用于信息流图分割成信息流森林外,还能很好地运用于程序的控制流分解等希望完整保留图中信息路径的场合。

参 考 文 献

[1] Denning D. A Lattice Model of Secure Information-flow[J]. Comm. of the ACM,1976,19(5):236-246
 [2] Kemmerer R A. Shared resource matrix methodology: An approach to identifying storage and timing channels. ACM Trans. on Computer Systems[J],1983:256-277
 [3] Goguen J A, Meseguer J. Security policies and security models//

Proc. of the IEEE Symp. on Security and Privacy. 1982;11-20
 [4] Tsai C R, Gligor V D. A formal method for the identification of covert storage channels in secure code. IEEE trans. on Software Engineering, 1990;569-580
 [5] 鞠时光,宋香梅.用于信息流分析的信息流树结构.江苏大学学报:自然科学版,2005,5
 [6] Eckmann S T. Eliminating Formal Flows in Automated Information-flow Analysis. IEEE,1994
 [7] 郭殿春,鞠时光,余春堂,等.隐通道及其搜索方法.计算机工程,2003(17)
 [8] 郭殿春,鞠时光,余春堂,等.隐通道及其搜索方法[J].计算机工程,2003,29(17):73-75
 [9] 郭殿春.隐通道搜索方法研究与实现[D].硕士学位论文.2004
 [10] 王朝瑞.图论.工业出版社,1987,6
 [11] Goguen J A, Meseguer J. Unwinding and Inference Control// Proceedings of the IEEE Symposium on Security and Privacy. Oakland, California, April 1984;75-86
 [12] Haigh J T, Kemmerer R A, McHugh J, et al. An Experience Using Two Covert Channel Analysis Techniques on a Real System Design. IEEE Transactions on Software Engineering, February 1987,13(2):157-168
 [13] Haberman A N. Synchronization of Communicating Processes. Communications of the ACM, July 1972,12(7):171-176
 [14] Kemmerer R A. Covert Flow Trees; A Visual Approach to Analyzing Covert Storage Channels. IEEE Transactions on Software Engineering[J], November 1991,17(11):1166-1185
 [15] McHugh J. Covert Channel Analysis; A Chapter of the Handbook for the Computer Security Certification of Trusted Systems[M]. Portland State University, December 1995
 [16] Kemmerer R A. Covert Flow Trees; A Visual Approach to Analyzing Covert Storage Channels[J]. IEEE Transactions on Software Engineering, 1991,17(11)

(上接第125页)

综上所述,管理角色的集合AR的定义为 $\{SuperAdmin, HRAdmin\} \cup \{L, M | l \in L\}$,之所以这样定义,是基于两个基本的安全原则:职责分离原则和最小权限原则,三个类型的管理角色各自行使自己的管理职能,超级管理角色能为图层管理角色和人力资源角色指派用户,但其自身不具有图层管理角色和人力资源角色的权限;图层管理角色可以管理特定特征类的权限、角色及受限角色,但其自身不具有这些角色、受限角色的权限,即不能对特征类所对应的特征进行任何形式的修改;而人力资源角色可以为受限角色指派用户,但其自身也不能查看和修改任何特征。通过实现职责分离和最小权限原则能最大限度地提高系统的安全。

结束语 本文提出了一个新的基于特征的空间数据访问控制模型,该模型由基本授权模型与授权约束模块两大部分组成,基本授权模型采用了基于角色访问控制模型中用户、角色和权限等概念,针对空间数据模型的具体特点,对权限和权限之间的依赖关系进行了详细的定义,并给出了由权限之间的依赖关系引出的隐式授权的定义及算法,基本授权模型中所关注的是角色所具有的权限,比如可以操作哪些特征类的哪些属性,其关注点是指派给角色的用户能够获得足够的权限以完成特定的业务功能;授权约束部分所提出的受限角色继承基本授权模型中角色的权限,但其关注点是将角色的权限限制在特定的地理区域内或特定的特征集合内,以控制角色权限的行使范围;通过这两个层次的授权以实现灵活的、细粒度的访问控制。文章给出了授权管理模型,通过定义3种

类型的管理员角色,对访问控制模型进行分布式管理。

参 考 文 献

[1] Bertino E, Jajodia S, Samarati P. A Flexible Authorization Mechanism for Data Management Systems. ACM Trans. on Information Systems, 1999,17(2):101-140
 [2] Bertino E, Samarati P, Jajodia S. An Extended Authorization Model for Relational Databases. IEEE Trans. on Knowledge and Data Engineering, 1997,9(1):85-101
 [3] Atluri V, Mazzoleni P. A uniform indexing scheme for geo-spatial data and authorizations//Proc. of the Sixteen Conf. on Data and Application Security. 2002;207-218
 [4] Bertino E, Damiani M L, Momini D. An access control system for a web map management service//Proc. of the 14th Int. Workshop on Research Issues on Data Engineering; Web Services for E-Commerce and E-Government Applications. 2004;33-39
 [5] Bertino E, Damiani M L. A controlled access to spatial data on Web//Proc. of the 7th AGILE Conf. on Geographic Information Science. 2004
 [6] OpenGIS Consortium, OpenGIS simple Features Specification for SQL, OGC 99-049, 1999
 [7] Belussi A, Bertino E, Catania B, et al. An authorization model for geographical maps//Proc. of the 12th annual ACM international workshop on Geographic information systems. 2004;82-91
 [8] Belussi A, Catania B, Bertino E. A reference framework for integrating multiple representations of geographical maps//Proc. of ACM GIS. 2003;33-34
 [9] Sandhu R, Coyne E J, Feinstein H L, et al. Role Based Access Control Models. IEEE Computer, 1996,29(2):870-881