基于符号 ADD 和线性多分支程序的分类算法安全评估

古天龙,何仲春,常 亮,徐周波
(桂林电子科技大学广西可信软件重点实验室,广西桂林 541004)

摘 要: 分类算法是机器学习和数据分析中重要的算法.当需要对分类算法本身以及算法的输入数据进行隐私保护时,就出现了分类算法安全评估问题.针对现有的分类算法安全评估协议效率较低的问题,文章给出了一种基于代数决策图和线性多分支程序的解决方案.首先,设计了基于代数决策图的安全函数评估协议,用以安全评估决策函数;其次,引入了线性多分支程序的概念,用其对分类算法进行表示.最后,借助线性多分支程序和基于代数决策图的安全函数评估协议,给出了一个私有线性多分支程序的安全评估协议.对新的协议的正确性和安全性进行了分析和证明.实验数据表明,与原有的解决方案相比,新的协议在效率上有明显的提高.

关键词: 安全评估; 分类算法; 代数决策图; 线性多分支程序
中图分类号: TN918; TP309 **文献标识码:** A **文章编号:** 0372-2112 (2014)05-0940-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.05.016

Secure Evaluation of Classification Algorithms Based on Symbolic ADD and Linear Multi-Branching Program

GU Tian-long, HE Zhong-chun, CHANG Liang, XU Zhou-bo
(*Guangxi Key Laboratory of Trusted Software, Guilin University of Electronic Technology, Guilin, Guangxi* 541004, *China*)

Abstract: Classification algorithms are widely used in the areas of machine learning and data mining. It is an important task to evaluate the classification algorithms securely when both the classification algorithm and the input data of the algorithm are private. In order to improve the efficiency of existing secure evaluation protocols, a solution based on both the algebraic decision diagram and the linear multi-branching program was presented. Firstly, a secure function evaluation protocol based on algebraic decision diagram was designed for evaluating decision functions securely. Secondly, a structure named linear multi-branching program was proposed to represent the classification algorithms. Based on both the secure function evaluation protocol and the structure of linear multi-branching program, a protocol for securely evaluating the private linear multi-branching programs was constructed. Both the correctness and the security of the new protocol were analyzed. Experimental results show that the new protocol is more efficient than the existing solutions.

Key words: secure evaluation; classification algorithm; algebraic decision diagram; linear multi-branching program

1 引言

分类算法旨在根据样本数据的特殊属性将未知类别的样本映射到给定类别中的某一类,是机器学习和数据挖掘中的最重要的技术之一[1],在垃圾邮件过滤[2]、远程故障诊断[3]、医疗诊断专家系统[4]等许多领域有着广泛的应用.

分类算法一般涉及到两个参与主体:提供样本数据的客户端,以及提供分类算法的服务端.传统的 C/S 模型下对分类算法的计算(客户端将样本数据提交给服务端,由服务端计算;或者服务端将分类算法提供给客户端,由客户端计算)面临着一个重要的安全问题:无法保护数据和算法的隐私.然而随着信息技术尤其是网络技术的发展,保护个人隐私和知识产权的要求越来越强烈.因此,出现了分类算法的安全评估问题[5,6],即能否在对分类算法的评估中,既能够保护客户端数据的隐私性,同时保证服务端的分类算法不被客户端获知.

安全评估最早由 Yao 提出[7],Yao 在介绍安全函数评估(任意函数的安全评估)的概念时,通过"百万富翁问题"进行了形象的说明,即如何通过某个协议,让两位百万富翁计算出谁更富有,且同时协议执行过程中两位百万富翁互不知晓对方的资产数量.

继文献[7]的开创性工作之后,Yao 提出了混淆电路的方法来解决安全函数评估问题[8],即用布尔电路表

收稿日期:2013-01-06;修回日期:2013-05-14;责任编辑:李勇锋
基金项目:国家自然科学基金(No.60963010,No.60903079,No.61100025,No.61262030,No.61363030);广西省自然科学基金(No.2012GXNSFBA053169)

示安全函数评估问题中待评估的函数.

2004 年,Malkhi 等人在文献[8]的研究工作基础上设计了一个两方安全函数评估软件 Fairplay[9].除了使用布尔电路表示待评估的函数之外,2006 年,Kruger 等人设计了基于符号 OBDD 的安全函数评估协议,并通过实验数据证明了用 OBDD 表示比较函数时比用布尔电路表示协议效率更高[10].

为降低基于布尔电路的安全函数评估协议的混淆电路规模,2010 年,Henecka 等人将同态加密技术与文献[11]中能有效减少加密电路规模的优化方法结合起来,开发了软件 TASTY[12],用于两方安全评估.

对分类算法的安全评估,可借助可编程的通用电路,将函数 f 的描述 P_f 和数据 x 作为可编程通用电路的输入然后计算 $f(x)$ 并输出[13].但是,分类算法作为一类特殊的函数形式使用布尔电路表示,效率不够理想.因此,2007 年 Brickell 等人使用分支程序 BP(Branching Program)表示分类算法,并提出了一个 BP 的安全评估协议[5],以及 Mohassel 等人基于一回合不经意传输技术提出的 BP 的安全评估协议[14].然而 BP 在描述分类算法时还存在一定的不足.为此,2009 年 Barni 等人在文献[5]的基础上提出了线性分支程序 LBP(Linear Branching Program)的概念,用其表示分类算法并给出了 LBP 的安全评估协议[6].

由于分类算法中的决策函数多为比较函数,且符号技术比布尔电路在表示非线性函数时空间复杂度更低[10],同时专为解决安全比较问题提出的两方安全比较协议[15]无法用于除比较函数之外其他函数的安全评估.因此,本文利用符号 ADD 在对伪布尔函数表示方面的优越性,提出了一个基于 ADD 的安全函数评估协议,用于分类算法中决策函数的安全评估.另外,为了拓展分类算法的描述形式,本文在 LBP[6]的基础上提出线性多分支程序 LMBP (Linear Multi-Branching Program)的概念,并给出了 LMBP 的安全评估协议.

2　预备知识

不经意传输协议(OT)

1-out-of-2 不经意传输协议是一个两方协议.例如,其中一方(Alice)输入 m 对 l-bit 的字符串 $S_i = < s_i^0, s_i^1 >, s_i^0, s_i^1 \in \{0,1\}^l, i = 1,2,\cdots,m$.另一方(Bob)输入 m bits $b_i \in \{0,1\}$,执行协议后的输出为:Bob 获得 $s_i^{b_i}$ 但是对 $s_i^{1-b_i}$ 值一无所知,并且 Alice 对 b_i 的值也是一无所知.本文中使用的不经意传输协议可实例化为文献[16,17]中一种或其他的高效解决方案.

同态加密(HE)

本文中使用一种语义安全的加同态公钥加密机制,加密机制的语义安全是指无法从特定密文中提取任何明文信息.加同态加密是指两个明文加法运算的结果加密等同这两个明文各自密文的乘法运算结果,例如$[\![a + b]\!] = [\![a]\!][\![b]\!]$.由此性质则可以做如下推导:$[\![c \cdot a]\!] = [\![a]\!]^c$,其中 c 为常数.

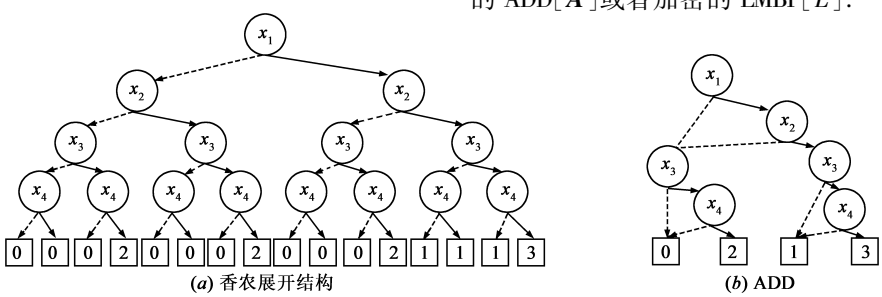
本文中对同态加密的实例化使用的是文献[18]中的加密机制,其明文空间表示为$\mathbb{Z}_N$ 对应的密文空间表示为$\mathbb{Z}_{N^2}^*$.关于加解密的更多细节可参考文献[19].

代数决策图(ADD)

一个 ADD 就是表示一簇伪布尔函数 $f_i: \{0,1\}^n \to S$(代数结构的有限值域)的有一个直接根结点的有向无环图,所有的有限个结点被分为终结点(叶子节点)和非终结点两类.每个终结点都被标有 S 中的一个元素,并且没有射出边.而每个非终结点都有两条射出边,即 0-边和 1-边.变量的一组赋值决定由根结点到一个终结点的一条路径分支,该分支的终结点所标识的值就是变量在这组赋值下所对应的函数值.

例:$f = x_1x_2 + 2x_3x_4$,其中变量序为 $x_1 < x_2 < x_3 < x_4$,它的完全二叉树及 ADD 如图 1 所示.由图 1 可见,函数用 ADD 表示只需要 10 个结点,占用空间较小,这就便于在函数上进行其它操作.

本文中提到的 x^l 表示 l-bit 的无符号整数.$\mathrm{Gen}(1^T)$ 表示的是文献[18]中同态加密的密钥生成算法.$[\![x^l]\!]$ 表示 l-bit 的明文消息 x^l($\mathbb{Z}_N$ 用公钥 pk 加密后的密文,其中$\mathbb{Z}_N$ 是公钥 pk 对应的明文空间.如果对象上方标记一个波浪符号则表示该对象已被混淆,意指打乱主体原本的顺序,例如获得 $\widetilde{\boldsymbol{w}}^i$ 的客户端无法推断混淆值 $\widetilde{\boldsymbol{w}}^i$ 中 i 的真实值,但是能够用这个混淆值去正确解密加密的 ADD$[\widetilde{\boldsymbol{A}}]$或者加密的 LMBP$[\tilde{L}]$.



(a) 香农展开结构　　(b) ADD

图1　伪布尔函数$f=x_1x_2+2x_3x_4$的表示

3 基于 ADD 的安全函数评估协议

本文在 Kruger[10]等人的基础上提出基于 ADD 的安全函数评估协议.全文中一概假设参与安全评估的双方为 Alice 和 Bob,协议具体内容如下:

协议 1 基于 ADD 的安全函数评估协议

输入:表示布尔函数 $f(x_1,x_2,\cdots,x_n)$的 ADD(f),其中变量序为 $x_1<x_2<\cdots<x_n$.Alice 的输入 $\boldsymbol{x}_a=(a_1,a_2,\cdots,a_k)$对应 ADD($f$)中的 k 个变量,Bob 的输入 $\boldsymbol{x}_b=(b_1,b_2,\cdots,b_m)$对应 ADD($f$)中的 m 个变量,其中 $k+m=n$.

输出:$C=f(\boldsymbol{x}_a,\boldsymbol{x}_b)$.

Step1 Alice 给 ADD(f)加上伪节点得到 ADD(f_{full}).

Step2 Alice 用 x_a 约束 ADD(f_{full})得到 ADD($f_{\text{full}|x_a}$).

Step3 Alice 为 ADD($f_{\text{full}|x_a}$)添加混淆节点.

Step4 Alice 加密 ADD($f_{\text{full}|x_a}$)和 **Step3** 添加的混淆节点,并将密文发送给 Bob.

Step5 Bob 通过 m 次 1-out-of-2 不经意传输获得对应密钥,然后解密 **Step4** 的密文得到 C,即 $f(\boldsymbol{x}_a,\boldsymbol{x}_b)$.

3.1 伪节点及混淆节点的添加

在 ADD 图中可能存在节点跳级的情况,会导致约束 ADD 的参与者输入信息的泄露,如图 1(b)所示.因此,在用 ADD 表示待评估函数后需要为 ADD 添加伪节点[10].除了节点的跳级,Alice 用不同的输入约束 ADD 出现规模不一致的情况,也会导致输入信息的泄露.

例如,图 2(a)以及图 2(b)表示图 1(b)中的 ADD 在填充了伪节点后,Alice 用不同 x_1,x_3 输入约束后的 ADD.可见,图 2(a)中的 ADD 有三个非终结节点,而图 2(b)中的 ADD 只有两个非终结节点.Kruger 等人通过保留全部非输入约束的节点方法来解决此问题[10].

例如,图 2(c)中的 ADD 是使用 x_1,x_3 约束图 1(b)时,保留全部 x_2,x_4 节点的 ADD.此时,不管 x_1,x_3 用什么样的输入来约束,约束后的 ADD 始终拥有六个非终结节点.

本文提出增加混淆节点的方法来解决不同约束导致 ADD 规模不一致的安全问题,具体思想如下:对给定函数的 ADD 表示,假设使用 $\boldsymbol{x}_a$ 约束时,约束后的内部节点的数量最大,并假设其值为 Max,则在使用其他输入约束时,将约束后的内部节点数量通过加入混淆节点的方法,使其数量增加到 Max.相比 Kruger 等人的方法,节约了离线加密次数以及在线通信量.

图 2(d)是在图 2(b)的 ADD 中增加了一个混淆节点,此时,其与图 2(a)中的 ADD 拥有同样数量的非终结节点,从而使得 Bob 无法从 ADD 的密文规模来判断 Alice 的输入.混淆节点必须具备的性质有:

(1)混淆节点须对应 Bob 的输入.

(2)混淆节点无法由约束后的图的根节点到达.

(3)混淆节点具有子节点并可为任意节点.

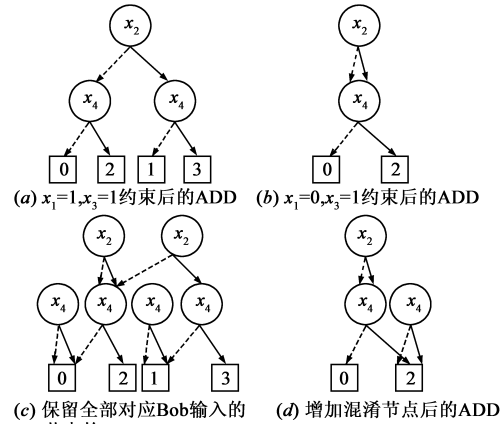


图2 约束ADD以及增加混淆节点

3.2 ADD 的加密和解密

协议 1 中需要对 ADD 进行混淆加密,以及对密文正确解密.ADD 的加密算法及解密算法如下.

算法 1 加密 ADD

输入:表示函数 f 的 ADD A 有 m 个变量,变量序为 $x_1<x_2<\cdots<x_m$,其中有 d 个非终结节点 $P_i(1\leqslant i\leqslant d)$,$z-d$ 个叶子节点 $P_j(d<j\leqslant z)$,每个非终结节点对应一个值 $level(P_i)$,表示节点标注的变量在变量序中的位置,P_1 为根节点,$level(P_1)=1$,每个叶子节点对应一个值 C_j,表示变量在一组赋值下所对应的函数值.

输出:密文$[\tilde{A}]=\{<\tilde{P}_1,\tilde{P}_2,\cdots,\tilde{P}_d>\}$;$m$ 对值 $\boldsymbol{W}_1,\boldsymbol{W}_2,\cdots,\boldsymbol{W}_m$.

1: choose a random permutation Π of the set $1,2,\cdots,d$ with $\Pi[1]=1$

2: choose key $\Delta_1=0^t$, random keys $\Delta_i\in_R\{0,1\}^t, 1<i\leqslant d$

3: choose m pairs of random value

$\boldsymbol{W}_k=\{\boldsymbol{w}_k^0=<s_k^0,\pi_k>,\boldsymbol{w}_k^1=<s_k^1,1-\pi_k>\}, 1\leqslant k\leqslant m, s_k\in\{0,1\}^t, \pi_k\in\{0,1\}$

4: for $i=1$ to d do

5: let permuted index $\tilde{i}=\Pi[i]$

6: let 0 successor $i_0=Low[i]$

7: if $i_0\leqslant d$ then {P_{i_0} is a Non-terminal node}

8: let $\tilde{i}_0=\Pi[i_0], m^{\tilde{i},0}=<\tilde{i}_0,\Delta_{i_0}>$

9: else{P_{i_0} is a Leaf node}

10: let $m^{\tilde{i},0}=<C_{i_0}>$

11: end if

12: let 1 successor $i_1=High[i]$,

13: if $i_1\leqslant d$ then

14: let $\tilde{i}_1=\Pi[i_1], m^{\tilde{i},1}=<\tilde{i}_1,\Delta_{\tilde{i}_1}>$

15: else let $m^{\tilde{i},1}=<C_{i_1}>$

16: end if

17: let $\tilde{P}_{\tilde{i}}=<\text{Enc}^{\tilde{i},0}_{\Delta_{\tilde{i}}\oplus s^{\pi_{level(P_i)}}_{level(P_i)}}(m^{\tilde{i},\pi_{level(P_i)}})$,

$$\mathrm{Enc}^{\tilde{i},1}_{\Delta_{\tilde{i}}\oplus s^{1-\pi_{level(P_i)}}_{level(P_i)}}(m^{\tilde{i},1-\pi_{level(P_i)}})>$$

18: end for

19: return $[\tilde{A}]=\{<\tilde{P}_1,\tilde{P}_2,\cdots,\tilde{P}_d>\};\boldsymbol{W}_1,\boldsymbol{W}_2,\cdots,\boldsymbol{W}_m$

在获得密文 ADD,以及 m 个解密密钥 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_m$ 后,通过算法 2 解密密文 ADD,得到某一叶子节点值 C.

算法 2　加密 ADD 的解密

输入:密文 $[\tilde{A}]=\{<\tilde{P}_1,\tilde{P}_2,\cdots,\tilde{P}_d>\}$; m 组值 $\boldsymbol{w}_1,\boldsymbol{w}_2,\cdots,\boldsymbol{w}_m$.

输出:一个叶子节点对应的值 C.

1: let $\tilde{i}=1;\Delta_{\tilde{i}}=0^t$(start at root node)

2: for $k=1$ to $m-1$ do

3:　let $<s_k,\pi_k>=\boldsymbol{w}_k;<c^0_{\tilde{i}},c^1_{\tilde{i}}>=\tilde{P}_{\tilde{i}}$;

$<\tilde{i},\Delta_{\tilde{i}}>=\mathrm{Dec}^{\tilde{i},\pi_k}_{\Delta_{\tilde{i}}\oplus s_k}(c^{\pi_k}_{\tilde{i}})$

4: end for

5: let $<s_m,\pi_m>=\boldsymbol{w}_m$;

$<c^0_{\tilde{i}},c^1_{\tilde{i}}>=\tilde{P}_{\tilde{i}}$;

$<C>=\mathrm{Dec}^{\tilde{i},\pi_k}_{\Delta_{\tilde{i}}\oplus s_m}(c^{\pi_k}_{\tilde{i}})$

6: return C

算法 1 和 2 中使用的语义安全的对称加密机制实例化为 $Enc^s_k(m)=m\oplus H(k\parallel s)=Dec^s_k(m)$,其中 $H(k\parallel s)$为安全散列算法 SHA256.

容易看出算法 1 和算法 2 的主要计算开销为安全散列算法 SHA256,并且在算法中 SHA256 的输入长度为 $t+\lfloor\log_2 d\rfloor$. SHA256 算法的明文空间为 2^{64}-bit,其输入按 512-bit 分组进行处理,产生 256-bit 的摘要;在输入不足 448-bit 时,则无需分组且填充输入至 512-bit.

在算法 1 中密钥长度 t 的取值为 80、96、112、128 四种[12];因此,当 $t=128$ 并且 SHA256 无需分组处理的前提下,算法 1 输入的 ADD 的非终节点 d 最大数量可达 2^{320}个,已远远超出实际处理的 ADD 规模;而在非终结节点数量不超过最大值的情况下,算法中的 SHA256 计算开销不变.因此,算法的输入无法影响到 SHA256 的计算开销;从而算法 1 和算法 2 在输入的 ADD 有 m 个变量,d 个非终节点时,其计算开销为 $2d+m$ 次 SHA256 计算.

算法 1 的空间复杂度(密文规模)取决于 $m^i(1\leqslant i\leqslant d)$,令算法 1 输入的 ADD,其叶子节点对应的值 C 的长度小于$(\lfloor\log_2 d\rfloor+t)$bits,则算法 1 在输入的 ADD 有 d 个非终节点,密钥长度为 t 时,密文规模为 $2d(\lfloor\log_2 d\rfloor+t)$bits.

4　私有 LMBP 的安全评估协议

4.1　线性多分支程序

定义 1　一个线性多分支程序(LMBP)是一个二元组 $<\{P_1,P_2,\cdots,P_z\},\{R_1,R_2,\cdots,R_d\}>$,其中 $1\leqslant d\leqslant z$,并且:

(1) $P_1,P_2,\cdots,P_d$ 为决策节点(即非终结节点),$P_{d+1},P_{d+2},\cdots,P_z$ 为分类节点(即终结节点).

(2) R_i 是决策点 P_i 的子节点索引函数(其中 $1\leqslant i\leqslant d$);令决策点 P_i 拥有的分支数量为 $mul(i)$,则 $R_i(t)$(其中 $1\leqslant t\leqslant mul(i)$)为 P_i 的第 t 个分支节点的索引.

(3)每个决策节点 $P_i(1\leqslant i\leqslant d)$对应于一个三元组 $<\boldsymbol{a}^l_i,\boldsymbol{y}^{l'}_i,\boldsymbol{f}_i>$,其中:$\boldsymbol{a}^l_i=(a^l_{i,1},a^l_{i,2},\cdots,a^l_{i,n})$是由 n 个 l-bit 的无符号整数组成的线性组合向量;$\boldsymbol{y}^{l'}_i=(y^{l'}_{i,1},y^{l'}_{i,2},\cdots,y^{l'}_{i,u})$是由 u 个 l'-bit 的无符号整数组成的决策向量;f_i 为决策函数,令 LMBP 的输入向量为 $\boldsymbol{x}^l=(x^l_1,x^l_2,\cdots,x^l_n)$,则函数 f_i 将输入 $y^{l'}_i$ 和 $\boldsymbol{a}^l_i\odot\boldsymbol{x}^l=\sum_{j=1}^{n}a^l_{i,j}x^l_j$ 映射为集合$\{R_i(1),R_i(2),\cdots,R_i(mul(i))\}$中的某个元素.

(4)每个分类点 $P_j(d<j\leqslant z)$对应于一个分类值 c_j.

本文引入的 LMBP 具有良好的兼容性;文献中的 LBP、BP 和 OBDD 都可以看作 LMBP 的特殊形式.具体来说,如果将每个决策点 $P_i(1\leqslant i\leqslant d)$的分支数量都限定为 $mul(i)=2$,将向量 $\boldsymbol{y}^{l'}_i$ 限定为一个无符号整数 $y^{l'}_i$,并且令函数 f_i 在 $\boldsymbol{a}^l_i\odot\boldsymbol{x}^l\leqslant y^{l'}_i$ 时输出 $R_1(i)$,其它情况输出 $R_2(i)$,则本文定义的 LMBP 将等同于文献[6]中定义的 LBP. 又由于 BP 和 OBDD 都是 LBP 的特殊形式[6],因此 LMBP 也兼容了 BP 和 OBDD.

4.2　LMBP 的加解密

下面给出 4.1 小节定义的 LMBP 的加解密算法.加密算法将一个 LMBP 中各个决策节点的子节点信息混淆加密,并将原有节点的顺序混淆(决策节点对应的三元组相应的改变了原来的序列),具体算法如下.

算法 3　加密 LMBP

输入:一个 LMBP $L=<\{P_1,P_2,\cdots,P_z\},\{R_1,R_2,\cdots,R_d\}>$,其中有 d 个决策节点 $P_i(1\leqslant i\leqslant d)$,$z-d$ 个分类节点 $P_j(d<j\leqslant z)$.

输出:密文 $[\tilde{L}]=\{<\tilde{P}_1,\tilde{P}_2,\cdots,\tilde{P}_d>\}$; d 对混淆值 $\widetilde{\boldsymbol{W}}_1,\widetilde{\boldsymbol{W}}_2,\cdots,\widetilde{\boldsymbol{W}}_d$;改变了序列的线性组合向量 $\tilde{\boldsymbol{a}}^l_1,\tilde{\boldsymbol{a}}^l_2,\cdots,\tilde{\boldsymbol{a}}^l_d$;改变了序列的决策向量 $\tilde{\boldsymbol{y}}^{l'}_1,\tilde{\boldsymbol{y}}^{l'}_2,\cdots,\tilde{\boldsymbol{y}}^{l'}_d$;改变了序列的函数 $\tilde{f}_1,\tilde{f}_2,\cdots,\tilde{f}_d$.

1: choose a random permutation Π of the set $1,2,\cdots,d$ with $\Pi[1]=1$

2: choose key $\Delta_1=0^t$, random keys $\Delta_i\in_R\{0,1\}^t,1<i\leqslant d$

3: for $i=1$ to d do $\{P_i$ is a decision node $<\boldsymbol{a}^l_i,\boldsymbol{y}^{l'}_i,f_i>\}$

4:　let permuted index $\tilde{i}=\Pi[i]$

5:　set permuted linear combination vector $\tilde{\boldsymbol{a}}^l_{\tilde{i}}=\boldsymbol{a}^l_i$;

permuted vector $\tilde{\boldsymbol{y}}^{l'}_{\tilde{i}}=\boldsymbol{y}^{l'}_i$;

permuted function $\tilde{\boldsymbol{f}}_{\tilde{i}}=\boldsymbol{f}_i$

6:　choose a random permutation Π_i of the set $1,2,\cdots,\mathrm{mul}(i)$

7:　choose a random garbled value

$\widetilde{\boldsymbol{W}}_i = \{\widetilde{w}_{i\,i}^{\Pi_i[1]} = < k_{i\,i}^{\Pi_i[1]}, 1 >, \widetilde{w}_{i\,i}^{\Pi_i[2]} = < k_{i\,i}^{\Pi_i[2]}, 2 >, \cdots, \widetilde{w}_{i\,i}^{\Pi_i[mul(i)]} = < k_{i\,i}^{\Pi_i[mul(i)]}, mul(i) >\}$

8: for $1 \leqslant j \leqslant mul(i)$ do

9: let the j-th successor $i_j = R_i[j]$

10: if $i_j \leqslant d$ then {P_{i_j} is a decision node}

11: let $\tilde{i}_j = \Pi[i_j]$, $m^{\tilde{i},j} = <$"decision", $\tilde{i}_j, \Delta_{i_j} >$

12: else {$P_{i_j} = < c_{i_j} >$ is a classification node}

13: let $m^{\tilde{i},j} = <$"classification", $c_{i_j} >$

14: end if

15: end for

16: let $\tilde{P}_i = < \mathrm{Enc}_{\Delta_i \oplus k_i^{\Pi_i[1]}}^{\tilde{i},1}(m^{\tilde{i},\Pi_i[1]}), \cdots, \mathrm{Enc}_{\Delta_i \oplus k_i^{\Pi_i[mul(i)]}}^{\tilde{i},mul(i)}(m^{\tilde{i},\Pi_i[mul(i)]}) >$

17: end for

18: return $[\tilde{L}] = \{< \tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_d >\}$;
$\widetilde{\boldsymbol{W}}_1, \widetilde{\boldsymbol{W}}_2, \cdots, \widetilde{\boldsymbol{W}}_d; \tilde{\boldsymbol{a}}_1^l, \tilde{\boldsymbol{a}}_2^l, \cdots, \tilde{\boldsymbol{a}}_d^l;$
$\tilde{\boldsymbol{y}}_1^{l'}, \tilde{\boldsymbol{y}}_2^{l'}, \cdots, \tilde{\boldsymbol{y}}_d^{l'}; \tilde{f}_1, \tilde{f}_2, \cdots, \tilde{f}_d$

在拥有密文$[\tilde{L}]$,以及解密密钥后(获得解密密钥的步骤将在 4.3 节中说明),便可对 LMBP 的密文进行解密,得到在一组输入下 LMBP 的输出.解密算法如下.

算法 4 加密 LMBP 的解密

输入:密文$[\tilde{L}] = \{< \tilde{P}_1, \tilde{P}_2, \cdots, \tilde{P}_d >\}$;
d 组混淆值$\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, \cdots, \widetilde{\boldsymbol{w}}_d$.

输出:一个分类节点对应的分类值 c.

1: let $\tilde{i} = 1$;
$\Delta_{\tilde{i}} = 0^t$(start at root)

2: while true do

3: let $< k_{\tilde{i}}, h_{\tilde{i}} > = \widetilde{\boldsymbol{w}}_{\tilde{i}}$;
$< c_{\tilde{i}}^1, c_{\tilde{i}}^{mul(\tilde{i})} > = \tilde{P}_{\tilde{i}}$;
$< type_{\tilde{i}}, data_{\tilde{i}} > = \mathrm{Dec}_{k_{\tilde{i}} \oplus h_{\tilde{i}} \oplus \Delta_{\tilde{i}}}^{\tilde{i},h_{\tilde{i}}}(c_{\tilde{i}}^{h_{\tilde{i}}})$

4: if $type_{\tilde{i}}$ = "decision" then

5: let $< \tilde{i}, \Delta_{\tilde{i}} > = data_{\tilde{i}}$

6: else

7: let $< c > = data_{\tilde{i}}$

8: return c

9: end if

10: end while

本节算法 3 和算法 4 中使用的加密机制与算法 1 和 2 中的加密机制相同.因此,参照 3.2 小节的分析.本节算法 3、4 在输入的 LMBP 有 d 个节点,其中每个节点拥有的分支数为 $mul(i)(1 \leqslant i \leqslant d)$时,其计算开销最坏情况下为$(\sum_{i=1}^{d} mul(i) + d)$次 SHA256 计算,算法 3 输出的密文规模为 $\sum_{i=1}^{d} mul(i)(|\log_2 d| + t)$bits.

4.3 协议流程

私有 LMBP 的安全评估协议建立在 C/S 模型上,服务端 Alice 拥有私有的 LMBP,客户端 Bob 拥有私人数据 $\boldsymbol{x}^l = (x_1^l, x_2^l, \cdots, x_n^l)$.在 4.2 小节的算法基础上,协议的具体内容如下.

协议 2 私有 LMBP 的安全评估协议

输入:LMBP L;向量 $\boldsymbol{x}^l = (x_1^l, x_2^l, \cdots, x_n^l)$.

输出:$c = L(\boldsymbol{x}^l)$.

Step1 Alice 加密 L,并将$[\tilde{L}]$发送给 Bob.

Step2 Bob 生成加同态加密机制的密钥$(sk, pk) = Gen(1^T)$,然后加密 $x_1^l, x_2^l, \cdots, x_n^l$,并将密文$[\![x_1^l]\!], [\![x_2^l]\!], \cdots, [\![x_n^l]\!]$和公钥 pk 发送给 Alice.

Step3 收到 Bob 发送来的$[\![x_1^l]\!], [\![x_2^l]\!], \cdots, [\![x_n^l]\!]$及 pk 后,Alice 计算$[\![z_i^{l''}]\!] = [\![\tilde{\boldsymbol{a}}_i^l \odot \boldsymbol{x}^l]\!] = \prod_{j=1}^{n} [\![x_j^l]\!]^{\tilde{a}_{i,j}^l}, (i = 1, 2, \cdots, d)$.

Step4 Alice 随机生成 d 个无符号数 $C_1^{l''}, C_2^{l''}, \cdots, C_d^{l''}$,计算密文$[\![z_i^{l''}]\!][\![C_i^{l''}]\!]$,并将密文打包发送给 Bob,$(i = 1, 2, \cdots, d)$.

Step5 Alice 构造 $\tilde{f}_i(\tilde{\boldsymbol{a}}_i^l \odot \boldsymbol{x}^l + C_i^{l''}, \tilde{y}_i^{l'})$的 $\mathrm{ADD}(\tilde{f}_i)$,将 $\mathrm{ADD}(\tilde{f}_i)$的叶子节点的值用 $\widetilde{W}_i$ 中的值替换$(i = 1, 2, \cdots, d)$.

Step6 Bob 用私钥 sk 解密 **Step4** 发送来的打包密文,得到 $\gamma_1, \gamma_2, \cdots, \gamma_{dl''}$.

Step7 Alice 和 Bob 执行 d 次协议 1,Bob 得到 $\widetilde{\boldsymbol{w}}_1, \widetilde{\boldsymbol{w}}_2, \cdots, \widetilde{\boldsymbol{w}}_d$,其中协议 1 的输入为 **Step5** 和 **Step6** 的结果.

Step8 Bob 解密$[\tilde{L}]$得到 $c = L(\boldsymbol{x}^l)$.

协议 2 中使用的加法 $\tilde{\boldsymbol{a}}_i^l \odot \boldsymbol{x}^l + C_i^{l''}$ 不计最高位进位,即输出仍为 l''-bits 位.因为通常情况下 $z_i^{l''}$ 的长度 l'' 要远远小于密钥的长度 T,因此将密文打包再发送给解密方能有效的节约通信上的开支和解密次数.

协议 2 在打包的同时给每个 $z_i^{l''}$ 加上一个 l''-bits 的随机值 $C_i^{l''}$ 能防止 $z_i^{l''}$ 的值在解密的过程中泄露.假设一次打包 d' 个密文,打包后的密文的明文空间需满足 $d' l'' \leqslant T$,即 $d' \leqslant \left| \frac{T}{l''} \right|$.解密所有的密文包后得到 dl'' 个 bit 值.关于密文打包和解包的详细步骤可结合本段参考文献[5,6].

5 正确性及安全性分析

首先对协议 1 的正确性进行分析.假设节点 $\tilde{P}_{\tilde{i}}$ 的密文中第一条密文为 c_0,第二条密文为 c_1.Bob 通过 1-out-of-2 不经意传输协议获得与其输入对应的 $w_1, w_2, \cdots, w_m$(Bob 输入 0 获得 w_k^0;Bob 输入 1 获得 $w_k^1, 1 \leqslant k \leqslant m$)后,根据 w_k 中的 π_k 值,选择 c_0 和 c_1 其中一条解密($\pi_k = 0$ 选择 c_0,$\pi_k = 1$ 选择 c_1).

假设 Bob 获得 $w_k^0, k = level(P_i)$,并通过上一个节点的解密获得了正确的 $\tilde{i}$ 和 $\Delta_{\tilde{i}}$,则当 $\pi_k = 0$ 时,s_k^0 解密 c_0 得 $m^{\tilde{i},\pi_k}$即 $m^{\tilde{i},0}$,当 $\pi_k = 1$ 时,s_k^0 解密 c_1 得 $m^{\tilde{i},1-\pi_k}$即

$m^{\tilde{i},0}$,故 s_k^0 总能正确解密出 $m^{\tilde{i},0}$,得到正确解密下一个节点的 $\tilde{i}_0$ 和 $\Delta_{\tilde{i}_0}$;Bob 获得 w_k^1 时同理.因此,协议 1 是正确的.

其次,对协议 2 的正确性进行分析.Bob 通过协议 1 得到 $\tilde{\boldsymbol{w}}_1,\tilde{\boldsymbol{w}}_2,\cdots,\tilde{\boldsymbol{w}}_d$,由协议 1 的正确性可知,若 $\tilde{f}_i(\tilde{\boldsymbol{a}}_i^l \odot \boldsymbol{x}^l,\tilde{\boldsymbol{y}}_i^{l'}) = R_j[i]$、$\tilde{i} = \Pi[i]$,则有 $\tilde{\boldsymbol{w}}_{\tilde{i}} = \tilde{\boldsymbol{w}}_i^j = <k_i^j, h_{\tilde{i}}>$、$h_{\tilde{i}} \in [1, mul(i)]$、$\Pi_i[h_{\tilde{i}}] = j$.令节点 $\tilde{P}_{\tilde{i}}$ 的密文中第一条密文为 c_1,第二条密文为 c_2,第 $h_{\tilde{i}}$ 条密文为 $c_{h_{\tilde{i}}}$;Bob 通过上一个节点的解密获得了正确的 $\tilde{i}$ 和 $\Delta_{\tilde{i}}$,根据 $\tilde{\boldsymbol{w}}_{\tilde{i}}^j$ 中 $h_{\tilde{i}}$ 的值解密 $c_{h_{\tilde{i}}}$ 得到 $m^{\tilde{i},\Pi_i[h_{\tilde{i}}]}$ 即 $m^{\tilde{i},j}$,其包含正确解密下一个节点的 $\tilde{i}_j$ 和 $\Delta_{\tilde{i}_j}$.因此协议 2 是正确的.

接下来,对协议 1 的安全性进行分析.本文协议 1 与文献[10]的安全函数评估协议主要有三处不同:

(1)本文的协议 1 中将 ADD 的叶子节点的信息全部包含在上一层的内部节点的密文中,而在加密算法中使用的加密机制为语义安全的加密机制的默认假设下.节点的密文是安全的,即叶子节点的信息是安全的.

(2)本文的协议 1 中使用了增加混淆节点的方法代替文献[10]保留所有节点的方法,而在第三节已经说明两种方法都能达到同样的目的.

(3)本文的协议 1 中,Alice 在 w_k 中加入随机数 π_k 来混淆节点中密文的顺序,而在伪随机数的安全性基础上,密文的顺序是不可猜测的.因此,本文协议 1 的安全性与文献[10]无异.对于本文协议 1 安全性的详细证明过程可基于本段的分析参考文献[18]的附录.

最后,对协议 2 的安全性进行分析.本文协议 2 与文献[6]中的协议不同点在于:

(1)在保护 LMBP 的决策参数时,本文协议 2 中使用本文协议 1 对 LMBP 中的决策点内的函数进行安全评估,协议 1 的安全性已于上段分析.

(2)在隐藏 LMBP 的结构时,LMBP 中的决策节点有不确定数量的子节点,即多分支,对于多分支安全性的处理,我们使用伪随机序列来混淆,并且对应节点每条分支的解密密钥分布在该节点对应的 ADD 的叶子节点中,基于上段分析可知,LMBP 的多分支信息是安全的.因此,本文协议 2 的安全性与文献[6]无异.对于本文协议 2 安全性的详细证明过程,可基于本段的分析参考文献[6]的完整版附录 C.

6　效率分析

在本文中 LMBP 的功能也可由 LBP 实现,但是 LBP 将需要更多的节点来完成,如图 3 所示.若 LMBP 中有 d_1 个决策点,每个决策点都有 m 条分支,则用 LBP 表示需要 $d_2 = (m-1)\ d_1$ 个决策点.

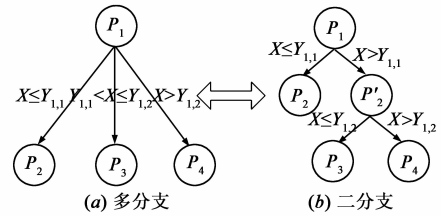


图3　LMBP与LBP的相互转化

分类算法中较为常见的为比较函数,因此,本文协议 1(表中用 ADD Based 表示)在安全评估函数 $n$cmp16($n = 2,3,4$)时——比较 n 个 16bits 整数的大小,与文献[9]中的 Fairplay、文献[12]中的 TASTY 从函数表示的规模以及 SHA256 执行的次数的角度进行比较,如表 2 所示.

令函数 $n$cmp16 的 n 个输入中 Alice 拥有 $n-1$ 个,表中 ADD 的 size 指函数的 ADD 表示用 Alice 的输入约束,并且加入伪节点及混淆节点后,ADD 拥有的非终结节点数量;Fairplay 以及 TASTY 的 size 分别表示 Fairplay、TASTY 输出的布尔电路表示函数时布尔电路门的数量.

表 1　协议 1 与 Fairplay[9]、TASTY[12]的比较

functions			2cmp16	3cmp16	4cmp16
ADD Based	size (nodes)		45	52	98
	execute SHA256 (times)		106	120	212
Fairplay[9]	size (gates)	3-input	14	36	42
		2-input	32	28	41
	execute SHA256 (times)		289	467	583
TASTY[12]	size (gates)	3-input	15	30	45
		2-input non-XOR	1	2	3
	execute SHA256 (times)		112	197	286

表 2 列出了本文协议 2 与 LBP 安全评估协议[6]在决策点数量、密文大小(bit)、最坏情况下需要执行 SHA256 的次数等三个方面的比较.其中 LMBP 的决策节点数量为 64、节点分支数 m 分别为 2、3 和 4、密钥长度 $t = 80$.

从表 1 的数据对比,可以看出基于 ADD 的安全函数评估在评估 LMBP 中常见的几种多值比较函数时效率较高.同时表 2 的实验数据表明,在多分支的情况下,LMBP 的安全评估协议的通信复杂度以及计算复杂度均优于 LBP 的安全评估协议[6].结合第 5 节的安全性和正确性分析可以得出,本文协议可以正确高效地对包含多函数多分支的分类算法进行安全评估.

表 2 协议 2 与 LBP[6]的安全评估协议的比较

Branching of decision node		$m=2$	$m=3$	$m=4$
LMBP	d_1	64		
	size(bits)	11136	16704	22272
	execute SHA256 (times)	192	256	320
LBP[6]	d_2	64	128	192
	size(bits)	11136	22528	34017
	execute SHA256 (times)	192	384	576

7 结束语

本文针对分类算法的安全评估问题提出了私有 LMBP 的安全评估协议,同时提出了基于 ADD 的安全函数评估协议,用其对 LMBP 中的决策函数进行安全评估,拓展了分类算法的描述形式,提高了分类算法安全评估协议的效率,并适用于其他能用 LMBP 描述的算法或函数的安全评估问题.

本文给出的私有 LMBP 的安全评估协议仅考虑了两方安全评估的情况,在下一步的工作中,我们将对多方安全评估协议进行研究.除此之外,本文私有 LMBP 的安全评估协议的对象主要为分类算法,在下一步的工作中,将从安全评估问题的一般性出发进行研究.

参考文献

[1] Ruggieri S. Efficient C4.5[J]. IEEE Trans on Knowledge and Data Engineering, 2002, 14(2): 438 – 444.

[2] Delany S J, Cunningham P, Doyle D, Zamolotskikh A. Generating estimates of classification confidence for a case-based spam filter[A]. Proceedings of the 6th International Conference on Case-Based Reasoning[C]. Chicago: Springer, 2005. Volume 3620 of LNCS: 177 – 190.

[3] Ha J, Rossbach C J, Davis J V, et al. Improved error reporting for software that uses black-box components[A]. Proceedings of the 2007 ACM SIGPLAN Conference on Programming Language Design and Implementation[C]. New York: ACM, 2007. 101 – 111.

[4] Rodriguez J, Goni A, Illarramendi A. Real-time classification of ECGs on a PDA[J]. IEEE Trans on Information Technology in Biomedicine, 2005, 9(1): 23 – 34.

[5] Brickell J, Porter D E, Shmatikov V, et al. Privacy-preserving remote diagnostics[A]. Proceedings of the 14th ACM Conference on Computer and Communications Security[C]. New York: ACM, 2007. 498 – 507.

[6] Barni M, Failla P, Kolesnikov V, et al. Secure evaluation of private linear branching programs with medical applications[A]. Proceedings of the 14th European Symposium on Research in Computer Security[C]. Saint-Malo: Springer, 2009. Volume 5789 of LNCS: 424 – 439.

[7] Yao A C. Protocols for secure computation[A]. Proceedings of the 23th IEEE Symposium On Foundations of Computer Science[C]. Chicago: IEEE, 1982. 160 – 164.

[8] Yao A C. How to generate and exchange secrets[A]. Proceedings of the 27th IEEE Symposium On Foundations of Computer Science[C]. Toronto: IEEE, 1986. 162 – 167.

[9] Malkhi D, Nisan N, Pinkas B, et al. Fairplay-a secure two-party computation system[A]. Proceedings of the 13th Conference on USENIX Security Symposium[C]. California: USENIX, 2004. 287 – 302.

[10] Kruger L, Jha S, Goh E J, et al. Secure function evaluation with ordered binary decision diagrams[A]. Proceedings of the 13th ACM Conference on Computer and Communications Security[C]. New York: ACM, 2006. 410 – 420.

[11] Pinkas B, Schneider T, Smart N P, et al. Secure two-party computations is practical[A]. Proceedings of the 15th International Conference on the Theory and Application of Cryptology and Information Security[C]. Tokyo: Springer, 2009. Volume 5912 of LNCS: 250 – 267.

[12] Henecka W, Kogl S, Sadeghi A R, et al. TASTY: tool for automating secure two-party computations[A]. Proceedings of the 17th ACM Conference on Computer and Communications Security[C]. New York: ACM, 2010. 451 – 462.

[13] Kolesnikov V, Schneider T. A practical universal circuit construction and secure evaluation of private functions[A]. Proceedings of the 12th International Conference on Financial Cryptography and Data Security[C]. Cozumel: Springer, 2008. Volume 5143 of LNCS: 83 – 97.

[14] Mohassel P, Niksefat S. Oblivious decision programs from oblivious transfer: efficient reductions[A]. Proceedings of the 16th International Conference on the Financial Cryptography and Data Security[C]. Bonaire: Springer, 2012. Volume 7397 of LNCS: 269 – 284.

[15] 李顺东,戴一奇,游启友.姚氏百万富翁问题的高效解决方案[J].电子学报,2005,33(5):769 – 773.
Li Shun-dong, Dai Yi-qi, You Qi-you. An efficient solution to yao′s millionaires′problem[J]. Acta Electronica Sinica, 2005, 33(5): 769 – 773. (in Chinese)

[16] Lipmaa H. Verifiable homomorphic oblivious transfer and private equality test[A]. Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security[C]. Taiwan: Springer, 2003. Volume 2894 of LNCS: 416 – 433.

[17] Ishai Y, Kilian J, Nissim K, et al. Extending oblivious transfers efficiently[A]. Proceedings of the 23rd Annual International Cryptology Conference[C]. California: Springer, 2003. Volume 2729 of LNCS: 145 – 161.

[18] Paillier P. Public-key cryptosystems based on composite de-

gree residuosity classes[A]. Proceedings of 17th International Conference on the Theory and Application of Cryptographic Techniques Prague[C]. Czech: Springer, 1999. Volume 1592 of LNCS: 223 – 238.

[19] Damgard I, Jurik M. A generalisation, a simplification and some applications of Paillier's probabilistic public-key system [A]. Proceedings of the 4th International Workshop on Practice and Theory in Public Key Cryptosystems[C]. Cheju Island: Springer, 2001. Volume 1992 of LNCS: 119 – 136.

作者简介

古天龙　男,1964 年 10 月出生,山西芮城人.分别于 1984 年、1986 年、1996 年在太原理工大学、西安电子科技大学、浙江大学获工学学士、硕士、博士学位.现为桂林电子科技大学副校长、博士生导师.主要研究方向为软件工程与形式化方法、移动计算与协议工程、符号计算与知识工程.

E-mail: cctlgu@guet.edu.cn

何仲春　男,1988 年 2 月出生,湖南永州人.现为桂林电子科技大学硕士研究生.研究方向为网络安全计算.

常　亮　男,1980 年 6 月出生,贵州赫章人.2008 年 7 月毕业于中国科学院计算技术研究所获工学博士学位.现为桂林电子科技大学教授、硕士生导师.研究方向为知识表示与推理、智能规划、形式化方法.

徐周波　女,1976 年 10 月出生,浙江奉化人.2012 年 1 月毕业于西安电子科技大学获工学博士学位.现为桂林电子科技大学副教授、硕士生导师.研究方向为符号计算、智能规划.