通用计算电路的不可区分混淆自动化构造方法

朱率率[1,2],韩益亮[1*],李 鱼[1,2]
(1. 中国人民武装警察部队工程大学密码工程学院,陕西西安 710086;
2. 网络与信息安全武警部队重点实验室,陕西西安 710086)

摘 要: 不可区分混淆(indistinguishability obfuscation,$i\mathcal{O}$)的构造问题是多年来一直困扰密码学研究的一个难题. 现有的基于多线性映射、函数加密、全同态加密等密码学原语的$i\mathcal{O}$构造均存在不同程度的安全性问题,且存在构造过程不易实现、电路扩展效率不高等缺陷. 本文从电路的自动化搜索的全新角度审视$i\mathcal{O}$的设计问题,将电路设计映射到图神经网络构造问题中,基于图神经网络的自动演化技术,探索了一种可以实现限定性满足不可区分性和功能保持性的通用$i\mathcal{O}$构造方法:AG$i\mathcal{O}$(Adversarial Graphweualietwork based $i\mathcal{O}$). 该$i\mathcal{O}$的基本架构基于对偶的对抗性图神经网络架构,针对任意给定输入电路,通过图枚举得到备用的电路样本集合,然后使用以子电路为粒度的差分演化算法分别独立优化上述对偶的图神经网络,当自动化判定模型从统计上不能有效识别不同的输出电路时,达到所需不可区分的状态. 测试结果表明,该AG$i\mathcal{O}$架构简单,易于实现,较好地实现了输入电路的通用性和统计上的不可区分性.

关键词: 不可区分混淆;公钥密码;图神经网络;生成式对抗网络;可证明安全
基金项目: 陕西省自然科学基础研究基金(No.2021JM-252);武警工程大学科研创新团队基金(No.KYTD201805)
中图分类号: TP309.7 **文献标识码:** A **文章编号:** 0372-2112(2024)01-0144-13
电子学报URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20211099

An Automatic Construction Method of Indistinguishable Obfuscation for Generic Computing Circuits

ZHU Shuai-shuai[1,2], HAN Yi-liang[1*], LI Yu[1,2]
(1. *Engineering College of Cryptography, Engineering University of the APF, Xi'an, Shaanxi* 710086, *China*;
2. *Network and Information Security Key Laboratory of APF, Xi'an, Shaanxi* 710086, *China*)

Abstract: The construction of indistinguishability obfuscation ($i\mathcal{O}$) is a long-term concern confusing the researchers. The existing $i\mathcal{O}$ constructions are based on primitives of multi-linear map, functional encryption, fully homomorphic encryption. These routines naturally inherent the shortages appeared in security, efficiency and generic abilities. To explore new approaches satisfying better generic functioning and indistinguishability from the angle of automatic searching, circuit design problems are mapped to the construction of graph neural network. In this paper, we present an $i\mathcal{O}$ framework called AG$i\mathcal{O}$ (Adversarial Graphweualietwork based $i\mathcal{O}$), which is based on dual adversarial neural network and can automatically generate sub-optimal $i\mathcal{O}$ with functional equivalence and generic circuit obfuscation. The AG$i\mathcal{O}$ achieved indistinguishability by circuit garbling which is a natural tool in constructing obfuscation. Then we design the graph-based automatic evolvement, which can well achieve sub-optimal circuit generalization. Through our test, the AG$i\mathcal{O}$ is simple to deploy and implement, while the efficiency is acceptable in achieving generalization and statistical indistinguishability.

Key words: indistinguishability obfuscation; public key; graph neural network; generative adversarial network; provable security

Foundation Item(s): Natural Science Foundation of Shaanxi Province (No. 2021JM-252); Innovative Research Team Project of Engineering University of APF (No.KYTD201805)

收稿日期:2021-08-16;修回日期:2022-03-14;责任编辑:王天慧
*通讯作者:韩益亮

1 引言

不可区分混淆器(indistinguishability obfuscation, $i\mathcal{O}$)[1]是近年来密码设计中的一个重要概念,最初由实现程序代码的版权保护机制发展而来,目的是保留代码的功能而不泄露执行过程. 出于相似的目的,$i\mathcal{O}$可以对密码算法的功能和执行过程加密,以保护密码算法的执行. 从广义上讲,使用安全、高效的$i\mathcal{O}$可以便捷地实现非对称加密算法、身份基加密算法和各类密钥封装算法,使数据加密过程不再受限于密码原语的选择和密钥的处理. $i\mathcal{O}$如此强大的构造能力吸引了众多密码研究人员的关注[2]. 同时,经过十多年的研究,$i\mathcal{O}$设计理论虽然呼之欲出,但在解决安全性、可靠性和实用性等关键问题上依然若隐若现,目前没有一个令人满意的结果.

$i\mathcal{O}$在设计上的两个基本要求是实现通用运算功能的保持和电路的不可区分. 实际上,有很多其他的密码构造也可以实现这两个基本要求,例如,混淆电路[3]和函数加密[4,5]等密码原语. 这些构造都可以实现对加密过程的保护,同时不影响程序的输出正确性,因此,在概念上与$i\mathcal{O}$非常相似. 事实上有许多工作专注于它们之间的相互构造,并取得了重要的成果[6]. 另外,$i\mathcal{O}$理论与其他构造的一个明显的区别是凌驾于电路、密钥和输入数据之上的通用性要求,目前这一点还难以实现.

因此,归纳当前$i\mathcal{O}$构造理论中有待克服如下两个方面的问题.

首先,$i\mathcal{O}$在电路混淆形式化的通用构造问题困扰了研究者很长时间. 根据文献[1]对不可区分混淆的形式化定义,构造规模相同、功能相同,并通过严格证明不可区分的密码结构是较难实现的. Garg等人[7]首先利用多线性映射构造通用的$i\mathcal{O}$,并将不可区分性规约到多线性映射困难问题上. Bitansky等人[8]利用函数加密构造了新的$i\mathcal{O}$方案. 但目前多线性映射类密码方案饱受攻击,从而影响了其在$i\mathcal{O}$构造中的应用;函数加密方案的构造基础还没有打牢,许多新型函数加密方案没有被充分认可,因此,用函数加密构造的$i\mathcal{O}$也是仅停留在理论探索阶段.

其次,通用$i\mathcal{O}$实现的复杂性影响了$i\mathcal{O}$理论的进一步应用. 为了实现Barak等人[1]定义的$i\mathcal{O}$形式化语义安全,目前$i\mathcal{O}$的所有构造方式均直接或间接地依托现有的密码原语,从而将不可区分的困难性规约到已知的困难问题上. 按照这样的思路所构造的$i\mathcal{O}$再用于进一步构造更高级密码协议,将会使新的密码协议在复杂程度、运行效率等方面的障碍而难以有效实现.

本文为了尝试解决$i\mathcal{O}$存在的上述问题,探索通过限定$i\mathcal{O}$的构造条件,实现自动化构造满足通用电路功能保持且次优不可区分特性的$i\mathcal{O}$. 这里的“次优”一方面指的是攻击敌手通过自动化搜索攻击$i\mathcal{O}$时,在限定时间内从统计达到足够的不可区分特性,从而取代现有的通过语义安全分析得到理论上的不可区分困难性;另一方面指的是在通用电路构造时,使用给定的随机选择的电路作为$i\mathcal{O}$的输入,取代语义上任意电路作为输入. 本文第2节和第3节分别介绍了$i\mathcal{O}$的研究现状和本文用到的重要概念或理论;第4节详细介绍了不可区分混淆AG$i\mathcal{O}$(Adversarial Graphweualietwork based $i\mathcal{O}$),的自动化构造过程;第5节和第6节分别从理论和实现角度分析了AG$i\mathcal{O}$的可行性、安全性等基本特性.

2 相关工作

Barak等人[1]提出了电路混淆的严格安全定义,即虚拟黑盒安全,并提出了安全上弱化的一种虚拟黑盒,即不可区分混淆器($i\mathcal{O}$),在现代密码学中具有重要的应用前景[9,10]. 然而,构造既能够实现混淆的不可区分性,同时又能够实现通用的功能,一直以来是构造通用$i\mathcal{O}$难以逾越的障碍,因此,构造满足条件的$i\mathcal{O}$成为密码学中一个重要的问题.

文献[11~13]提出了多线性映射的构造(multilinear maps),该构造具有非常好的函数映射通用性. Garg等人[7]利用多线性映射构造了首个通用的$i\mathcal{O}$,并实现了对于相同的输入,其输出电路在计算上满足不可区分性. 文献[14~16]等利用基于多线性映射构造的$i\mathcal{O}$,构造了语义上安全的单密钥公钥密码体制、多方安全计算、密钥交换协议等多个密码学应用. 其中,Sahai等人[14]证明大多数的密码单向计算应用均可以通过$i\mathcal{O}$实现新的构造. 然而,随着深入的研究,针对多线性映射的安全问题暴露出来. Cheon等人[17]分析了基于整数的多线性映射中采用的全新的安全假设. Chen等人[18]成功攻击了文献[7,19,20]等利用多线性映射构造的$i\mathcal{O}$. 为了弥补多线性映射构造$i\mathcal{O}$的安全缺陷和效率问题,文献[21]采用弱多线性映射构造了新的$i\mathcal{O}$,避开了多线性映射中已知的缺陷,文献[22]中使用固定编码梯度的多线性映射,文献[23]中沿用相同的思路,证明了梯度为3的三线性映射上的计算困难性,并相应构造了$i\mathcal{O}$. 文献[24,25]成功构造了基于双线性映射的$i\mathcal{O}$,使其安全性建立在成熟的DDH问题上. 但$i\mathcal{O}$应该是密码学上具有安全完备性的基础原语,因此,一般认为仅仅在构造工具上的修补对实现$i\mathcal{O}$的完备性没有本质上的帮助.

近年来出现了许多基于新的密码原语或困难问题的$i\mathcal{O}$构造方法,如基于函数加密、全同态加密等. 2018年,文献[26]基于有限域上的矩阵分支运算构造了程

序的混淆方法. 2019 年, Agrawal[27]构造了一种基于带噪声的线性函数加密原语的 $i\mathcal{O}$, 且不使用任何映射结构, 并在 2020 年对这种构造存在的安全问题进行了攻击分析和改进[28]. 2020 年, Brakerski 等人[29]利用全同态加密原语构造了新的 $i\mathcal{O}$, 使用了一种全同态加密的变体(split FHE)构造了全新的 split $i\mathcal{O}$, 使 $i\mathcal{O}$ 的不可恢复和不可区分性建立在 LWE 经典困难问题.

另外, 与不可区分混淆有着天然相似性的混淆电路也独立的发展, 并在多个方面有着重要的应用. 混淆电路最初由 Yao[3]提出用于解决多方安全计算问题, 主要用电路设计的技巧来隐藏运算电路. 之后, Lindell 等人[30]和 Bellare 等人[31]分别将混淆电路进行了完整的形式化定义, 使之成为独立的抽象密码工具, 用于密码方案的分析和构造, 并逐渐演变成了白盒密码理论. 但从已有工作来看, 高效的白盒密码对具体的密码方案实现都采用专用的电路, 且仅对当前的对称密码算法有效, 从而缺少通用白盒密码的可重用构造. 2013 年, Goldwasser 等人[32]首次利用全同态加密原语构造了可重用的混淆电路, 但该方案效率不高, 使其仅具有理论意义. 2019 年, Zhang 等人[2]指出从构造条件和过程来看, 使用混淆电路构造不可区分混淆具有较大的探索意义, 并指出构造 $i\mathcal{O}$ 其实质是构造一个不需要私钥的混淆电路方案, 同时在解决 $i\mathcal{O}$ 效率问题时, 混淆电路也具有一定的优势, 这一点也是本文有待探究的一个重要问题.

3 预备知识

这一节对本文用到的一些定义和结论做简要的介绍.

定义 1 布尔电路: 具有 n 个输入节点、m 个输出节点和 q 个中间节点的有向无环图, 表示为 $C(n,m,q)$, 其中 $n,m,q\in N$, 且满足:

(1)每个中间节点可以执行逻辑运算与 $\wedge$、或 $\vee$、非 $\neg$ 中的一种;

(2)C 的电路规模指节点数, 记为 $|C|$;

(3)因为非运算可以转化为与或运算, 故中间节点的逻辑运算为 $f:\{0,1\}\times\{0,1\}\to\{0,1\}$;

(4)电路的运算表示为 $F: x\to C(x)$, 其中 $x\in\{0,1\}^n$, 任意电路总可以分解为若干个子电路的形式, 故而若仅考虑输出顶点为 1 的情况, 则电路运算可表示为 $F:\{0,1\}^n\to\{0,1\}$, 且 F 函数是 f 函数在所有顶点递归的结果.

定义 2 不可区分混淆 $i\mathcal{O}$[31]: 给定多项式规模的电路 C, 对任意合法的输入 x, 构造电路 C', 对任意多项式时间的判别器 D, 满足 $\Pr\{D(C',C)=1|C'(x)\leftarrow i\mathcal{O}\{C(x)\}, C'(x)=C(x), |C'|=|C|\}\leqslant \mathrm{negl}(\lambda)$.

$i\mathcal{O}$ 包含两个算法, 即电路混淆算法 obfu 和电路计算算法 eval, 分别用于对给定的电路进行混淆和计算混淆后电路的结果.

定义 3 $i\mathcal{O}$ 的不可区分性游戏:

(1)敌手 A 选择电路 C_0, C_1 发送给挑战者, 满足 $|C_0|=|C_1|$, 对任意输入 x, $C_0(x)=C_1(x)$;

(2)挑战者 C 掷一枚硬币, 结果为 $b\in\{0,1\}$, 计算 $C'=i\mathcal{O}(C_b)$, 并将 C' 发送给敌手;

(3)敌手 A 对任意 x 作为输入, 计算 $C'(x)$;

(4)敌手 A 输出 b' 作为对 b 的猜测值.

定义 4 $i\mathcal{O}$ 的自适应性不可区分性安全: 给定安全参数 λ, 对多项式时间计算能力的 A, 若在定义 3 的游戏中的优势满足 $\mathrm{Adv}_{A,i\mathcal{O}}^{\text{Adaptive-IND}}=\Pr\{b'=b\}-1/2\leqslant\mathrm{negl}(\lambda)$, 则 $i\mathcal{O}$ 满足自适应性不可区分性安全.

根据 Lindell 等人[30]和 Bellare 等人[31]的形式化定义和功能描述, 本文得出混淆电路如下简洁定义.

定义 5 混淆电路[30,31]: 给定任意电路 C, 通过电路等效替换、集成、电路加密等方式, 重新构造 $\tilde{C}$, 对任意合法输入 x, 满足 $\tilde{C}(x)=C(x)$.

$i\mathcal{O}$ 要求加密后的电路保持相同的功能和规模, 且实现电路的通用性构造, 即对任意电路 C, 都可以实现安全的不可区分混淆, 而不依赖特定的电路结构或密钥. 因此, 构造 $i\mathcal{O}$ 的实质为构造满足自适应不可区分性安全的混淆电路方案, 且在电路评估阶段不需要私钥的输入.

定义 6 图神经网络 GNN[33,34]: GNN 是用图结构表示输入输出或中间连接状态而构造的一类神经网络, 用于完成高维度特征的表达、学习和演化.

定义 7 差分演化[35]: 给定一个群体 $G=\{x_0,x_1,\cdots,x_n\}$ 的初始值, 以及定义在 G 上的某约束 $F(G)$, 通过演化算法 $\mathrm{DE}(\cdot)$, 求解 $\tilde{G}=\mathrm{DE}\{x_0,x_1,\cdots,x_n\}$ 满足 $F(G)$ 的最佳组合关系, 即 $\mathrm{argmin}_\theta\{F(\cdot)\leftarrow \mathrm{DE}_\theta(G)\}$, 其中 θ 为 $\mathrm{DE}(\cdot)$ 的代价参数.

定义 8 图枚举算法: 设 μ 为关于 λ 的一个多项式, 给定深度为 d, 输入规模为 I, 总规模为 N 的电路 C: $\mathrm{DAG}(g_0,g_1,\cdots,g_N)$, 通过搜索空间 $\{0,1\}^N$, 输出规模为 1 的电路 $\tilde{C}$, 满足对于任意输入 x, 有 $C(x)=\tilde{C}(x)$, 则该算法的最大步骤数为 $2^I\times\sum_{i=1}^{d-1}2^i\binom{2}{2^{i-1}}$, 即计算复杂度为 $\mathcal{O}\left(\mu(\lambda)\left(\frac{d}{2}!\right)2^d\right)$.

根据定义 6、定义 7、定义 8, 本文构造了基于图神经网络的差分演化模型, 模型中假定所有电路拓扑中的节点都是入度为 2 且出度为 1 的与门和或门, 即任意节

点运算为 g:$\{0,1\}\times\{0,1\}\to\{0,1\}$. 在实现自动化搜索某个子电路 g 的等效电路过程最直接的方法是图枚举算法,即通过枚举搜索具备等效真值表的功能可替代子电路. 为了减小算法复杂度,在图枚举的输出为 $g(x)\in\{0,1\}$.

对给定的电路拓扑,图枚举算法可以搜索得到功能相同、拓扑不同的等效电路. 但该算法是一个计算复杂度较高的指数级算法,为了快速求得等效电路的拓扑,通常将输入电路 C 划分为若干个子电路 $\{g_0,g_1,\cdots,g_n\}$,其中 $n=N/|g_i|$,分别并行的执行图枚举算法,以提高效率. 在子电路 g_i 中,我们称其最大深度为子电路阶数 q. 给定 q,任何一个电路 C 都可以划分为若干个拓扑不重叠子电路,进而在调用图枚举算法时避开阶数高的子电路,大幅降低算法复杂度. 同时,算法对 i 进行抽样,在保证功能的同时,使算法在多项式时间内得到次优的结果.

引理 1　非完全覆盖的图枚举[36,37]:对于最大深度为 d 的电路 C 及其阶数为 q 的划分 $\{g_0,g_1,\cdots,g_n\}$,存在关于安全参数 λ 的多项式 $\mu(\lambda)$,使在 C 上的图枚举算法计算复杂度为 $\mathcal{O}\left(\mu(\lambda)\dfrac{N}{q}2^{3q-2}\right)$.

从电路的拓扑构造上,阶数越高,等效电路搜索的复杂度越高;另外,从电路的输入规模上,规模越大,等效电路存在的可能性越小. 因此,在调用图枚举算法时,尽可能选择阶数小的拓扑划分,实际上,第 4 节令 $q\leqslant 5$,而且有效的等效电路多集中在 $q\leqslant 4$,所以在非完全覆盖的图枚举场景中,引理 1 具有很好的适用性.

定理 1　对于输入电路 C 和任意子电路划分方法,基于图枚举算法的差分演化输出满足门类型及其连接关系的统计不可区分性. 具体来说,对于多项式时间电路攻击算法 $\mathcal{A}$:

(1)对于任意一个规模为 n 的子电路 $g\in C$ 及其任意两个不同的演化输出 $\tilde{C}_0$ 和 $\tilde{C}_1$,在多项式时间内,有 $\Pr\{D(\tilde{C}_0,\tilde{C}_1)=1\}\leqslant\mathrm{negl}(n)$;

(2)给定安全参数 λ 和在多项式时间内获得的有限个演化结果 $\{\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N\}$,对攻击算法 $\mathcal{A}$,有 $\Pr\{C\leftarrow\mathcal{A}(1^\lambda,\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N)\}\leqslant\mathrm{negl}(\lambda)$.

证明:假定存在自适应的攻击算法 $\mathcal{A}$ 对输入电路 $g=\{g_0,g_1,\cdots,g_n|T_i\}$ 的等效电路进行区分,其中 T_i 为 g 的拓扑,满足对任意 x,有 $g(x)=T_i(x)$. 令两个相互独立的差分演化的实例为 $\tilde{C}_0\leftarrow\mathrm{GE}_0(g,[q]_0),\tilde{C}_1\leftarrow\mathrm{GE}_1(g,[q]_1)$ 下面通过等效电路搜索和 i 抽样的独立性,证明 $\tilde{C}_0$ 和 $\tilde{C}_1$ 的随机拓扑连接是由搜索复杂度决定的独立同分布的两个不同抽样,进而,彼此的区分成功的概率为对应两个分布的抽样自由度的多项式函数.

对于给定的电路规模 n,拓扑的枚举空间为 Λ:$T_i^*\times\{0,1\}^n$,因此,攻击算法区分 $\tilde{C}_0$ 和 $\tilde{C}_1$ 成功的概率即为从 Λ 中枚举 T_i^*,满足 $\tilde{C}_0(x)=T_i^*(x)$ 或 $\tilde{C}_1(x)=T_i^*(x)$ 的概率.

设 p 为 g 上所有可能的拓扑,则 $\Lambda=\{T_0,T_1,\cdots,T_p\}\times\{0,1\}^n$,则有 $p\leqslant(2n)!$,因此,一次枚举操作中失败的概率为 $1-\left(\dfrac{2}{(2n)!}\right)^2$,对于 $\tilde{C}_0$ 和 $\tilde{C}_1$ 的枚举只需任意一个成功即可,因此,攻击算法成功的概率为

$$\begin{aligned}\Pr\{D(\tilde{C}_0,\tilde{C}_1)=1\}&\leqslant 1-\left(1-\left(\frac{2}{(2n)!}\right)^2\right)^2\\&\leqslant\left(\frac{2}{(2n)!}\right)^2\left(2-\left(\frac{2}{(2n)!}\right)^2\right)\\&\leqslant\frac{8}{(2n)!}.\end{aligned}$$

因此,有 $\Pr\{D(\tilde{C}_0,\tilde{C}_1)=1\}\leqslant\mathrm{negl}(n)$.

对于恢复 C 的攻击,即做图枚举和逐层抽样的逆操作,则搜索空间为 $\{0,1\}^{\mu(\lambda)\frac{N}{q}2^{3q-2}}$,根据上述结果,在该空间中有

$$\begin{aligned}&\Pr\{C\leftarrow\mathcal{A}(1^\lambda,\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N)\}\\&=\prod_{i=0}^{N}\Pr\{g_i\leftarrow\mathcal{A}(\tilde{g}_0,\tilde{g}_1,\cdots,\tilde{g}_N)\}\\&\leqslant\prod_{i=0}^{N}\Pr\{D(g_i,\tilde{g}_i)=1\}=\prod_{i=0}^{N}\frac{8}{(2n)!}.\end{aligned}$$

N 容易满足 $N\leqslant\dfrac{\lambda}{q}$,于是 $\mathcal{A}$ 恢复 C 的拓扑的概率小于 $\prod\limits_{i=0}^{\frac{\lambda}{q}}\dfrac{8}{\left(\frac{2\lambda}{q}\right)!}$,因此,$\Pr\{C\leftarrow\mathcal{A}(1^\lambda,\tilde{C}_0,\tilde{C}_1,\cdots,\tilde{C}_N)\}\leqslant\mathrm{negl}(\lambda)$.

接下来,我们在上述结论的基础上定义次优的不可区分性,以及相应的次优的不可区分混淆 $i\mathcal{O}$. 设由算法 $\mathrm{DE}(\lambda,C)$ 在电路 C 对应的 GNN 上得到的结果为 C'. 敌手构造为了区分 C',C 而构造的区分器为 $D_{\mathrm{GNN}}(C',C)$,其准确度为 $\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C))$,且

$$\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C))=\frac{N\{b'=b\}}{N\{b'=b\}+N\{b'\neq b\}}.$$

其中,$N\{b'=b\},N\{b'\neq b\}$ 为定义 3 对应游戏实例所进行的结果统计,且 $N\{b'=b\}+N\{b'\neq b\}\leqslant 2^\lambda$,则定义次优的不可区分性安全如下.

定义 9　次优的自适应性不可区分性安全:设挑战者使用等效电路演化的方法构造的 $i\mathcal{O}$ 作为定义 3 的挑战结构,令 $\mathrm{negl}(\lambda)=(1-\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C)))^q$,

$\mathrm{Adv}_{A,i\mathcal{O}}^{\text{Adaptive - IND}}=\mathrm{Adv}_{A,D}^{\text{Adaptive - IND}}$，则当定义3中敌手优势满足 $\mathrm{Adv}_{A,i\mathcal{O}}^{\text{Adaptive - IND}}\leqslant(1-\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C)))^{q}$ 时，所挑战的 $i\mathcal{O}$ 满足次优的自适应性不可区分性安全.

定义 10 次优的不可区分混淆 $i\mathcal{O}$：任意输入电路 C 的可区分性概率满足：$\Pr\{D(C',C)=1|C'(x)\leftarrow i\mathcal{O}\{C(x)\},|C'|=|C|\}\leqslant\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C))$ 的 $i\mathcal{O}$，且对任意输入电路 C，$\mathrm{Acc}(\lambda,D_{\mathrm{GNN}}(C',C))$ 存在上确界 $\mathrm{Adv}_{A,D}^{\text{Adaptive - IND}}$.

为了保护电路拓扑，本文借鉴了混淆电路的设计方法，对门电路及其连接关系实现混淆，从而隐藏电路 C 的电路拓扑，同时可以使不同的输出电路 $\tilde{C}_0$ 和 $\tilde{C}_1$ 满足统计上次优的不可区分性.

4 不可区分混淆 AG$i\mathcal{O}$ 的自动化构造

4.1 系统架构

为了构造满足 $i\mathcal{O}$ 必备特性且易于实现的结构，我们借助具有自动化搜索和生成能力的对抗式演化结构. 该架构包括3个部分：带抽样功能的图枚举算法、对偶的两个图神经网络演化模型和全局的电路判别器. 带抽样功能的图枚举算法中，对于给定的功能 F，批量产生参与演化的电路样本，从中再次随机抽样得到 C_0 和 C_1，作为 G_0 和 G_1 的输入；在输出编码和判别优势的监督下，对偶的图神经网络演化模型采用差分演化算法更新模型 G_0 和 G_1；判别器 D 将 G_0 和 G_1 的图输出作为待判定的输入，经过基于图神经网络的学习得到当前输入的判别优势 Adv，由 Adv 作为该架构分支或终止的条件. 该架构以任意功能的电路 $C(x)$ 作为输入，使用对输出电路 Adv 的监督作为 G_0 和 G_1 演化的全局策略，并使用以随机选择 x 的电路编码输出作为功能监督，以最大置信保持了 G_0 和 G_1 中间电路等效的结构演化. G_0 和 G_1 独立决定了各自演化方向，并使 D 达到对二者输出的可忽略 Adv，使局部电路呈现统计上的最大不可区分性. 同时，为了优化全局的执行效率，根据输入电路 $C(x)$ 的规模，架构中设置了判别器的静态淘汰阈值，用于决定演化初始电路 C_0 和 C_1 是否具备快速达到预定的不可区分程度.

如图1所示，不可区分混淆 AG$i\mathcal{O}$ 架构可以表示为 $\mathrm{AG}i\mathcal{O}\leftarrow\{(G_0,G_1)\leftarrow\mathrm{DE}[(C_0,C_1)\leftarrow\mathrm{GE}(C)]\|\arg\min_{\lambda,\eta}\{\mathrm{Adv}\}\leftarrow D(G_0,G_1)\}$，其中，$\mathrm{DE}(\cdot)$ 为差分演化部件，$\mathrm{GE}(\cdot)$ 为图枚举部件，$D(\cdot)$ 为图判别器，G_0 或 G_1 为输入电路 C 的混淆. 若 $\mathrm{Adv}\leqslant\mathrm{negl}(\lambda)$，则上述模型的输出满足不可区分性.

4.2 功能电路枚举

电路的阶数 q 定义为一个子电路 g 中包含与门和或门的个数. 阶数为 q 的电路 g，其等效电路 $\tilde{g}$ 定义为满

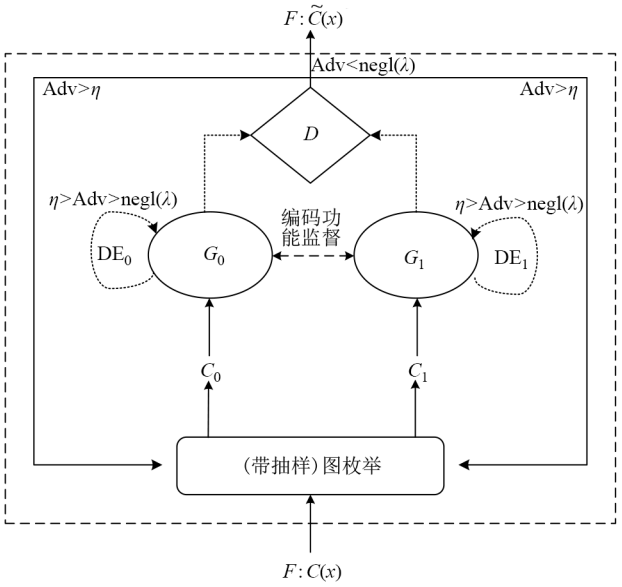


图1 基于图神经网络演化的不可区分混淆器 AG$i\mathcal{O}$ 架构

足 $|g|=\tilde{g}$，且对于任意输入 x，有 $g(x)=\tilde{g}(x)$，即从输入输出编码的角度讲，二者有相同功能表达. 如果把 $i\mathcal{O}$ 看作虚拟黑盒，对于给定的任意电路 C，它首先要识别该电路的功能，然后通过其内部状态的转换，得到混淆后的电路 C. 对于输入编码长度为 n 的电路，这个识别的步骤需要搜索功能相同的电路 $\{C_0,C_1,\cdots\}$ 作为备用的潜在输出电路，但搜索规模是关于 n 的指数级规模. 因此，设计多项式级的功能电路枚举是设计 $i\mathcal{O}$ 的基本步骤，其本质问题是给定任意电路规模（例如，C 中包含 N 个门，连接结构不限定，电路功能是实现一个函数 $F:y=C(x)$），通过抽样枚举可能的子电路构造空间，满足在所在阶数上子电路的功能等效，从而得到集合 $S_C=\{C_0,C_1,\cdots\}$，对任意 x，满足 $C_i(x)=C(x)$，且 $|C_i|=|C|$.

算法1描述了从给定的任意电路 C 出发，通过枚举局部的子电路 $\tilde{g}$，产生具有相同电路功能和电路规模的部分混淆电路集合 S_C，用于产生统计上不可区分的目标电路 $\tilde{C}$. 该算法的基本思想是，从带输入节点的子电路 g 开始，逐一枚举真值表相同的等效子电路 $\tilde{g}$. 根据 q 的值，由简单到复杂，产生满足保持电路功能的潜在混淆电路. 实际电路 C 的规模远远超出了多项式时间内可枚举的范围，需要采用局部抽样的方法进行枚举. 算法1中采用了均匀的随机对子电路编号进行抽样的方式降低算法复杂度. 针对规模较大的输入电路，应按照实际的电路结构分布进行抽样. 一种方法是，按照电路的阶数 q，统计门电路的类型，并估计门电路跳数的分布函数；然后，带输入节点的子电路 g 依照该分布的位置选取，采用局部等效替换的方法枚举有效的备选电路.

算法1　等效功能电路枚举算法

输入:C,T.

/*输入的电路和全局的最大时间限制*/

输出:$S_C=\{C_0,C_1,\cdots,C_s\}$

1. $t\leftarrow t_0$

/*设置枚举时间,t_0为开始时间,t为当前时间*/

2. **ForEach** q in $[3,|C|]$:

/*遍历等效电路的阶数*/

3. **While** $|S_C|<s$:

/*得到的备选电路添加到S_C中*/

4. **ForEach** randonly pick g in C:

/*枚举在C中抽样得到的阶数为q的且包含输入节点的子电路*/

5. **While do**:

6. ENUM $\tilde{g}$

/*枚举一个阶为q的电路*/

7. **If** $|\tilde{g}|=|g|$且$\tilde{g}\neq g$且$\mathrm{EC}(g,\tilde{g})=1$ **Then**:

/*判断$\tilde{g}$是否是等效子电路,若是则进行局部的电路替换*/

8. $g\leftarrow C$

9. **Else** Continue

10. **If** $g=\perp$或$\tilde{g}=\perp$ **Then**:

/*已穷尽当前所有子电路*/

11. Break

12. $S_C\leftarrow C$

/*添加一个新找到的等效电路*/

13. **If** $t>t_0+T$ **Then**:

/*超时退出*/

14. Quit

4.3　基于图神经网络的电路差分演化

G_0和G_1的图演化是产生不可区分混淆的驱动部分,基本方法是按照子电路的阶数逐层对输入的节点类型和连接关系进行局部最大化差分演化,达到对C_0和C_1不可区分重构的目的. 演化过程使用判别器D的优势Adv作为图神经网络全局的半监督,并使用$G_0(x)=G_1(x)$作为演化的中断性监督,其中x随机选自子电路真值表的合法输入. 中断性监督仅是一个必要的校验步骤,并不能保证结果满足演化的功能等价性. 该演化过程分为如下3个基本步骤:

(1)将C_0和C_1按照阶数分别划分为两个子电路的列表$L_0=\{c_{00},c_{01},\cdots,c_{0u}\}$和$L_1=\{c_{10},c_{11},\cdots,c_{1u}\}$,其中$u=\lfloor|C|/q\rfloor$,在每一次$G_0$和$G_1$的演化中,$L_0$和$L_1$重新随机划分;

(2)对于$i,j\in\{0,1,\cdots,|C|\}$,以子电路为单位输入演化算子$\nabla(c_{0i},c_{1j})=\mathrm{Pr}\cdot\mathrm{SW}(c_{0i})+(1-\mathrm{Pr})\cdot\mathrm{SW}(c_{1j})$,其中,$\mathrm{SW}(\cdot)$为图中等效电路所代表的节点替换,$\mathrm{Pr}=\{[c_{0i}(0)-c_{0j}(0)]+[c_{0i}(1)-c_{0j}(1)]\}/(2q)$,即当前等效电路中相同位置门电路结构重复的概率;

(3)对G_0和G_1进行演化算子的迭代,并将每一代的结果送入判别器D,直到D达到满足要求的判定优势$\mathrm{Adv}\leqslant\mathrm{negl}(\lambda)$.

算法2详细实现了上述步骤.

算法2　电路差分演化算法

输入:C,S_C,λ,η,T

/*输入的电路,安全参数和全局的最大时间限制*/

输出:$\widetilde{C_0},\widetilde{C_1},\mathrm{Adv}$

/*输出混淆电路和电路的区分优势*/

1. $t\leftarrow t_0,q\in\{3,4,\cdots,|C|\},\mathrm{negl}(\lambda)\leftarrow 2^{-\lambda}$;

/*设置迭代初始时间和子电路阶数*/

2. 初始化G_0,G_1,D,L_0,L_1

3. **While** $\eta>\mathrm{Adv}>\mathrm{negl}(\lambda)$

/*驱动G_0,G_1的电路差分演化*/

4. 随机重排顺序(L_0,L_1)

/*一轮迭代前,将存储子电路的列表乱序排列*/

5. **ForEach** (c_{0i},c_{1j}) in (L_0,L_1):

/*遍历相同阶数的子电路*/

6. $\mathrm{Pr}\leftarrow\{[\mathrm{c}_{1i}(0)-\mathrm{c}_{0j}(0)]+[\mathrm{c}_{1i}(1)-\mathrm{c}_{0j}(1)]\}/(2q)$

7. $\mathrm{Pr}:c_{0i}\leftarrow\mathrm{Algorithm}_1(c_{0i}),\ 1-\mathrm{Pr}:c_{0j}\leftarrow\mathrm{Algorithm}_1(c_{0j})$

/*分别以概率Pr,1−Pr调用等效电路替换算法*/

8. **If** $\mathrm{Adv}\leqslant D(G_0,G_1)$ **Then**:

/*判别器优势未达到减少不可区分优势,则裁剪该子电路,并重新选择等效电路*/

9. $i=i-1,j=j-1$

10. Continue

11. $\mathrm{Adv}\leftarrow D(G_0,G_1)$

12. **If** $t>t_0+T$ or $\mathrm{Adv}<\mathrm{negl}(\lambda)$ **Then**:

/*超时或满足性能要求后退出*/

13. Quit

14. **If** $\mathrm{Adv}>\eta$ **Then**:

/*输入电路不满足要求,重新生成样本集S_C*/

15. $S_C\leftarrow\mathrm{Algorithm}_1(C)$

/*调用等效电路替换算法*/

16. **Goto** Line 2

4.4　判别器的设计

判别器D是实现对G_0和G_1演化有效监督的另一个关键部件,它由一个图自动二分类的分类器CL组成,对输入的一个电路对$\{G_b,C\},b\in\{0,1\}$,能够尽最大准确度实现对b的判定. 对于电路结构,该自动分类器的首选结构是使用卷积图神经网络分类器GCN,训练的样本为矩阵化的$\{G_0,C\}$和$\{G_1,C\}$,并使用二元交叉熵(Binary Cross-Entropy, BCE)损失函数进行训练监督. 在AG$i\mathcal{O}$运行的初期,二分类的分类器,如SVM、逻辑回归、随机森林等经典工具均可满足,但随着子电路阶数的增加,深度GCN结构更具有准确性优势. 本

文选择用全连接的图卷积网络,每一层的输出特征用$f_i^l=\sigma\left(\sum_{j\in n_i} f_j^{l-1} w_{T_j}^{l-1}\right)$表示,其中,$w$为特征权重,$T_j$为第$j$个节点的门类型. 根据数据规模,也可在层输入特征值上增加归一化因子和偏置,以提高计算效率和准确性.

判别器D的目的是在统计上尽可能的得到高精度的准确率,因此,在实现环节中可以设置m种自动判别分类器,在演化迭代中采用并行判定的方式,取各个算法的最高准确率作为最终判别结果$\tilde{b}$的计算根据,即$\text{Accuracy}_{D,\tilde{b}}=\max_{\text{Accuracy max}}(\text{CL}_0,\text{CL}_1,\cdots,\text{CL}_m)$. 基于智能算法的二分类工具的构造技术比较成熟,此处不再赘述.

在AG$i\mathcal{O}$启动时,判别器延后于G_0和G_1启动,并以0.5的概率随机读取G_0和G_1的,并输出判定结果$\tilde{b}$,然后计算该判别器累积的判别准确度为$\text{Accuracy}_{D,\tilde{b}}$,则$\text{Adv}=\text{Accuracy}_{D,\tilde{b}}-0.5$代表了判定$G_0$和$G_1$输出的优势.

5 AG$i\mathcal{O}$的构造分析

本节分别从AG$i\mathcal{O}$功能的通用性和构造的不可区分性两个方面分析其性能. AG$i\mathcal{O}$的通用性体现在,图枚举部件、演化部件和对抗性判别器对于任意输入电路C的功能、拓扑、规模不做任何限定,中间处理过程仅与当前的枚举结果和判别器反馈的优势有关. AG$i\mathcal{O}$的不可区分性通过多层迭代的拓扑差分演化保证. 具体分两个方面:一方面从多层阶数子电路替换了原有的子电路,隐藏了输入电路的拓扑结构;另一方面,对每一层的随机抽样决定了等效电路替换操作的不可逆.

5.1 不可区分特性分析

从定性上讲,AG$i\mathcal{O}$的不可区分性,一方面建立在直接在输入C上的子电路等效替换上,使用低阶数替换后的结果$\{C_0,C_1,\cdots,C_s\}$作为执行进一步通用演化的样本;另一方面建立在对不同样本的演化上,更进一步地自动混淆不同样本间的电路拓扑特征. 两个方面的区别在于:前者会尽可能覆盖以固定q划分的所有子电路,效率较高;而后者在输入的C_0和C_1上,按照q对$\{g_{i0},g_{i1},\cdots,g_{in}\}$进行逐层抽样,在反馈的Adv监督下迭代演化出新的拓扑,满足对高阶子电路的覆盖. 当通过判别器无法对GE_0和GE_1输出的电路特征产生明显的判别优势时,即达到了次优的不可区分性.

根据定义9,我们用定理2来计算输出结果的区分概率.

定理2 对任意选择的输入多项式规模为$\mu(\lambda)$的电路C和输入变量x,AG$i\mathcal{O}$满足$\Pr\{D(\tilde{C}_0\leftarrow\text{AG}i\mathcal{O}(C),\tilde{C}_1\leftarrow\text{AG}i\mathcal{O}(\text{C}))=1\}\leqslant\text{negl}(\lambda)$.

证明:假定存在敌手可以控制判别器D,并运行攻击算法$\mathcal{A}$用于分析AG$i\mathcal{O}$,通过下面的攻击游戏实现对AG$i\mathcal{O}$输出结果的区分攻击.

(1)生成阶段:初始化AG$i\mathcal{O}$的结构,设置输入规模$|x|=\mu(\lambda)$,调用初始图枚举部件并输出样本库$S_C=\{C_0,C_1,\cdots,C_s\}$.

(2)抽样阶段:由GE_0和GE_1从S中随机抽取样本索引$\{i,j\}$,令$C_0\leftarrow S[i],C_1\leftarrow S[j]$.

(3)演化阶段:对于电路阶数$q\geqslant 2$,判别器计算判别优势$\text{Adv}\leftarrow D(\tilde{C}_0,\tilde{C}_1)$,并反馈给$\text{GE}_0$和$\text{GE}_1$,由$\text{GE}_0$和$\text{GE}_1$运行演化算法得到稳定的挑战电路$\tilde{C}_0^*\leftarrow\text{GE}_0(C_0),\tilde{C}_1^*\leftarrow\text{GE}_1(C_1)$.

(4)攻击阶段:敌手控制判别器,并自适应请求运行$\tilde{C}\leftarrow\text{AG}i\mathcal{O}(\text{GE}_0,\text{GE}_1,D)$.

(5)输出阶段:敌手输出$b\in\{0,1\}$,若b满足$\tilde{C}_b^*\leftarrow\text{GE}_b(C_b)$,则攻击成功.

下面计算敌手获得成功的概率:

$$\Pr\{b=0|\tilde{C}_0\leftarrow\text{GE}_0(C_0)\quad\text{or}\quad b=1|\tilde{C}_1\leftarrow\text{GE}_1(C_1)\}$$
$$=1-\prod_{i=0}^{n}\Pr\{D(g_{0i},g_{0i^*})=0\}\times\prod_{i=0}^{n}\Pr\{D(g_{1i},g_{1i^*})=0\},$$

根据定理1,上式小于等于$\prod_{i=0}^{n}\Pr\{D(g_{i0}^*,g_{i1}^*)=1\}$,小于等于$\prod_{i=0}^{n}\frac{8}{(2n)!}$,输入规模容易满足$n\leqslant\frac{\lambda}{q}$,故上式小于等于$\prod_{i=0}^{\frac{\lambda}{q}}\frac{8}{\left(\frac{2\lambda}{q}\right)!}$.

5.2 通用性分析

AG$i\mathcal{O}$的通用性体现在通过架构的可重用性,实现对任意输入电路的运算功能保持. 根据定义2和定义10,AG$i\mathcal{O}$的通用性由定理3来定量描述.

定理3 对任意输入规模为$\mu(\lambda)$的电路C和输入变量x,AG$i\mathcal{O}$满足$\Pr\{\tilde{C}_0(x)=\tilde{C}_1(x)=C(x)|\Pr\{D(\tilde{C}_0\leftarrow\text{AG}i\mathcal{O}(C),\tilde{C}_1\leftarrow\text{AG}i\mathcal{O}(C))=1\}\leqslant\text{negl}_0(\lambda)\}\leqslant 1-\text{negl}_1(\lambda)$.

证明:即证当D的优势$\text{Adv}\{D(\tilde{C}_0,C)=1\}\leqslant\text{negl}(\lambda)$时,对任意输入$x$,有$\tilde{C}_0=\tilde{C}_1=C(x)$.

此时根据定理2,在GE_0的第i轮迭代满足$\Pr\{D(\tilde{C}_0,\tilde{C}_1)=1\}\leqslant\text{negl}_0(\lambda)$.

当$i=0$时,根据等效电路枚举中的替换规则,有$\tilde{C}_0=C_0(x),\tilde{C}_0=C_0(x)$,显然有$\Pr\{\tilde{C}_0(x)=\tilde{C}_1(x)=$

$C(x)\|(\tilde{C}_0,\tilde{C}_1)\leftarrow \mathrm{AG}i\mathcal{O}\}\rightarrow 1-\mathrm{negl}_1(\lambda)$.

当 $i\geqslant 1$,且 $q\geqslant 2$ 时,

$\Pr\{\tilde{C}_0=\tilde{C}_1=C(x)\}$,

$\Pr\{\tilde{g}_0(x)=g_0(x),\tilde{g}_1(x)=g_1(x),\cdots,\tilde{g}_n(x)=g_n(x)\}$,

又因为对 $\{g_0,g_1,\cdots,g_n\}$ 演化是在每一轮对 GE_i 完成随机划分后独立替换的,故

$$\begin{aligned}&\Pr\{\tilde{C}_0=\tilde{C}_1=C(x)\}\\&=\prod_{j=1}^{n}\Pr\{\tilde{g}_j(x)=g_j(x)\}\\&=\prod_{j=1}^{n}\left(1-\left(\mu(\lambda)\frac{n}{q}2^{3q-2}\right)\right)^{-1}\\&\geqslant\left(1-\mu(\lambda)\lambda 2^{\frac{3\lambda}{n}-2}\right)^{n}\\&=\sum_{k=0}^{n}\binom{k}{n}\left(-\mu(\lambda)\lambda\cdot 2^{\frac{3\lambda}{n}-2}\right)^{n-k}\\&\geqslant 1-(n-1)(\mu(\lambda)\lambda\cdot 2^{\frac{3\lambda}{n}-2})^{n}.\end{aligned}$$

综上,存在可以忽略的变量 $\mathrm{negl}_0(\lambda)$ 和 $\mathrm{negl}_1(\lambda)$,当 $\Pr\{D((\tilde{C}_0),\tilde{C}_1)\leftarrow \mathrm{AG}i\mathcal{O}(C(x))=1\}=\mathrm{negl}_0(\lambda)$ 时,有 $\Pr\{\tilde{C}_0=\tilde{C}_1=C(x)\}=1-\mathrm{negl}_1(\lambda)$.

6　AG$i\mathcal{O}$性能测试

本节根据第4节的基本框架和主要算法实现了一个原型本版的AG$i\mathcal{O}$,初步摸索了AG$i\mathcal{O}$训练中所需要的运行环境和参数设置,同时结合AG$i\mathcal{O}$的设计初衷,对得到的部分结果做了性能、效率等方面的分析.

6.1　环境设置

原型版本的AG$i\mathcal{O}$使用Pytorch在Python3 64bit环境下编写,在E5218@2.3 GHz 32core CPU,128 GB内存和P2200GPU加速下运行. 原型AG$i\mathcal{O}$目前仅支持单比特输出的电路,且由于计算复杂度原因,GE的图演化中要求 $q\leqslant 10$. 在判别器的实现上,本文参考了SDNE网络[36]嵌入的方法对输入的图样本进行特征提取. AG$i\mathcal{O}$实例中所需要的基本参数见表1~3,其中表1为AG$i\mathcal{O}$的输入 C 和输出 C',以及由二者枚举出的样本电路所需要的参数,表2为差分演化部件DE的参数设置,表3为判别器D的参数设置.

表1　样本参数

参数名称	值	注释
$\lvert S_C\rvert$	1 000	样本容量
N	2~1 023	输入电路的规模
q	{2,3,4,5,6,7,8,9,10}	子电路的阶数(电路深度)
η	0.25	GE重新抽样的优势

表2　差分演化部件DE参数

参数名称	值	注释
$\lvert S_C\rvert$	1 000	样本容量
N	2~1 023	输入电路的规模
Input	S_C	输入样本
Output	G_0, G_1	输出结果
q	{2,3,4,5,6,7,8,9,10}	子电路的阶数(电路深度)
η	0.25	GE重新抽样的优势

表3　判别器D训练参数

参数名称	值	注释
CL	SDNE基本模型	图神经网络分类器
Input Width	1 023	输入宽度
Depth	6	图神经网络深度
Input	G_0, G_1	输入样本
Output	{0,1}	输出结果
$\sigma(\cdot)$	sigmoid($\cdot$)	层内特征激活函数
$\lvert s\rvert$	$\leqslant\lvert S\rvert$	可供训练的样本量
negl(λ)	$(1-\mathrm{Acc}(2^{10},\mathrm{CL}_{\mathrm{SDNE}}))^q$	D的不可区分精度(终止条件)

6.2　AG$i\mathcal{O}$的运行实例

6.2.1　输入输出电路结构

由表1可知,当输入电路 C 的规模最大时,q 最大为10,则 $n=2^{q-1}=512$,$|C|=\sum_{i=0}^{q-1}2^i=1\,023$. C 可表示为节点入度均为2且出度均为1的有向无环图(Directed Acyclic Graph,DAG),其中每个中间顶点均为一个随机生成的门(Random Gate从二元门And,Or,Nand,Nor中随机选取),输出电路 C' 的结构与 C 相同,如图2所示. 通过将 C 输入枚举算法,可以产生一系列与 C 等效等规模的电路样本,例如当 $q=2$ 时,DAG中任意两个相邻的节点构成的子电路均存在对称的等效等规模电路,如图3(a)所示. 当 $q\geqslant 3$ 时,枚举算法通过枚举等效子电路的方法

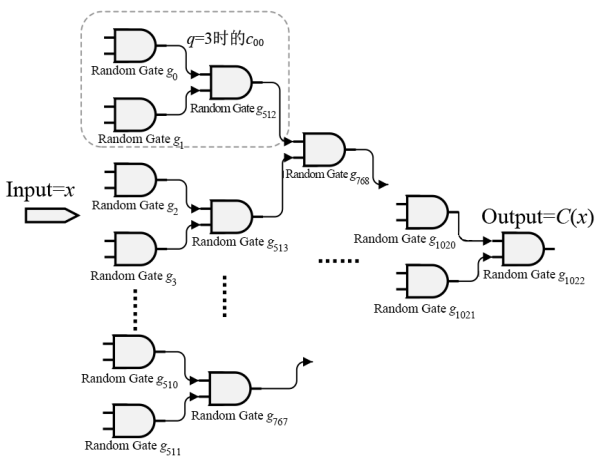
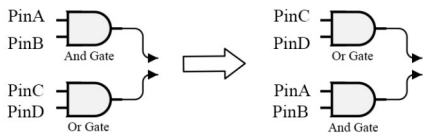
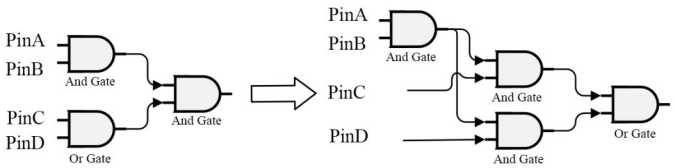


图2　输入电路C的结构

产生新的电路样本集合 S_C，且 $|S_C| \leqslant \sum_{q=2}^{10} \frac{|C|}{q}$，并对其中的子电路分别做等效替换的标记和随机生成的 C_0 和 C_1 类别标记，已替换的子电路不再重复枚举，例如，当 q=3时，等效替换子电路如图3(b)所示．由于布尔运算均满足分配律，故而扩展和缩减的等效子电路需要成对出现才能保证样本电路规模的恒定，可以通过在DAG全局设置电路规模的实时计数器实现．



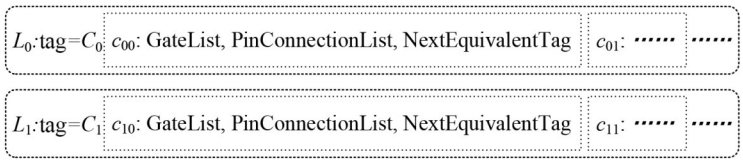
(a) q=2时的等效电路之一



(b) q=3时的等效电路之一

图3 等效电路的不同形态

电路结构在实现上有两种表达方式，如图4所示．一种是将逻辑门和连接关系分别存储在列表中独立存储，并标记各自的位置和子电路的位置，方便算法1对子电路进行结构遍历和功能枚举，如图4(a)所示；另一种是将逻辑门和连接关系转化为DAG邻接矩阵，并将枚举的结果作为子电路输出节点的属性，用于执行枚举算法之后的操作，如图4(b)所示．两种表达方式可以随时根据需要相互切换．



(a) 样本集合SC列表存储

	g_0	g_1	g_2	·····	$g_{1\,023}$
g_0	Na	Na	1	·····	0
g_1	0	Na	1	·····	0
g_2	0	1	Na	·····	0
·····	·····	·····	·····	·····	·····
$g_{1\,023}$	0	0	0	·····	Na

注：1表示有连接，0表示无连接，Na表示无意义，值由枚举结果决定．

(b) 样本集合SC中用DAG邻接矩阵表示的电路

图4 样本集合SC的存储方式

(2)G_0和G_1

G_0和G_1分别用于存储枚举算法产生的样本电路，同时也负责管理DE部件的输入和输出．当G_0和G_1为DE部件输入时，样本电路转换为DAG邻接矩阵的形式；当需要监督电路功能时，样本电路转换为列表形式．

(3)DE部件

该部件按照3个步骤迭代执行．

首先，在G_0和G_1中，以q=2为例，C_0和C_1按照子电路的标签分别划分为两个子电路的列表L_0=$\{c_{00}, c_{01}, \cdots, c_{0u}\}$和$L_1=\{c_{10}, c_{11}, \cdots, c_{1u}\}$，其中$u=\lfloor |C|/q \rfloor$=511．在每一次执行算法2前，$L_0$和$L_1$重新随机划分，并验证$C_0(x)=C_1(x)$，且$|C_0|=|C_1|$，$x$为512 bit随机输入．

然后，对G_0和G_1进行演化算子的迭代．对于$i,j \in \{0, 1, \cdots, 1\,022\}$，以子电路$c_{0i}$和$c_{1j}$为单位执行演化算子$\nabla(c_{0i}, c_{1j})=\mathrm{Pr}\cdot \mathrm{SW}(c_{0i})+(1-\mathrm{Pr})\cdot \mathrm{SW}(c_{1j})$，用于决定是否对$c_{0i}$和$c_{1j}$进行等效电路替换，其中，SW($\cdot$)为$S_C$中最邻近的等效子电路替换，Pr为当前等效电路中相同位置门电路结构重复的概率，当q=2时，c_{0i}和c_{1j}分别只包含2个门电路，因此Pr=0.25.

最后,将每一代的G_0和G_1中的结果送入判别器D,直到D达到满足要求的判定优势$\mathrm{Adv}\leqslant \mathrm{negl}(\lambda)$,Adv由$D$动态更新.

(4)D部件

该部件由表3中的超参构造的SDNE图神经网络[34,36]构成,由7层神经元组成,包含输入层、4层图卷积编码层、中间层、输出层,且不再需要解码层,其优化过程见文献[36]的算法1. 在DE部件开始执行之前,D部件的训练集输入为S_C,在DE部件开始执行后,其输入为G_0和G_1中的DAG邻接矩阵,输出为判定结果$\{0,1\}$. 判别器D的判定优势Adv是动态更新的,且由定义3可知,

$$\mathrm{Adv}_{i\mathcal{O}}^{D}\approx \frac{\text{判别正确的次数}}{\text{总次数}}-\frac{1}{2},$$

若$q\leqslant 5$,且在S_C上得到的$\mathrm{CL}_{\mathrm{SDNE}}(C_0,C_1)$准确率为0.9,则$\mathrm{negl}(\lambda)=10^{-5}$. 当满足$\mathrm{Adv}_{i\mathcal{O}}^{\mathrm{D}}\leqslant 10^{-5}$时,终止AG$i\mathcal{O}$的运行,并将$C_0$作为AG$i\mathcal{O}$的输出电路;当$\mathrm{Adv}_{i\mathcal{O}}^{D}>\eta=0.25$时,由枚举算法重新构造样本集合$S_C$.

6.3　结果分析

现有的$i\mathcal{O}$构造方案中最成熟的是借助基于多线性映射的全同态加密方案构造的$i\mathcal{O}$,绝大多数停留在理论阶段的原因是计算效率低,系统开销大. 公开的文献中实现的实例非常少,如文献[38]中小于100个门电路的$i\mathcal{O}$执行时间在小时级别,且其性能经过了多年并没有提升. 本文围绕构造次优$i\mathcal{O}$的目的就是在保证基本特性的前提下,通过折中不可区分特性的实现方法和衡量方法,大幅提高$i\mathcal{O}$执行的效率. 在同等CPU核心数情况下,表4对比了AG$i\mathcal{O}$与目前最成熟的$i\mathcal{O}$构造方式之间的性能对比. 为了统一对比标准,表4使用了算法实例的效率进行算法性能的对比,从中可以看出,AG$i\mathcal{O}$在混淆速度和评估速度上具有明显的提升.

表4　时间开销对比

方法	电路规模	Obfu时间/s	Eval时间/s	算法实例效率
方案1[39]	255	1 007.40	57.85	0.239 4
方案2[2]	63	8 473.00	953.00	0.006 7
方案3[38]	31	249.80	5.81	0.121 3
AG$i\mathcal{O}$	1 023	536.05	0.05	1.908 2

注:算法实例效率=电路规模/(Obfu时间+Eval时间)

图5和图6分别显示了AG$i\mathcal{O}$在G_0和G_1电路规模小于1 024时的存储开销和时间开销,其中,1 023个门电路的输入,混淆时间小于600 s,运行内存开销约为6 GB,在执行效率上有了实质性的提高. 同时,我们对GE部件是否对子电路进行抽样做了对比测试. 可以看出,经过抽样的算法实例随着门电路规模增长呈现远低于指数级的开销增加,而没有经过抽样的算法实例则呈现出指数级开销增长. 对不同阶数的子电路抽样之后再进行演化是AG$i\mathcal{O}$实现效率提升的关键.

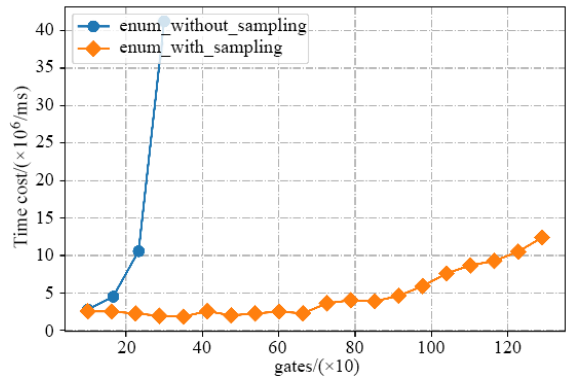


图5　AG$i\mathcal{O}$空间开销

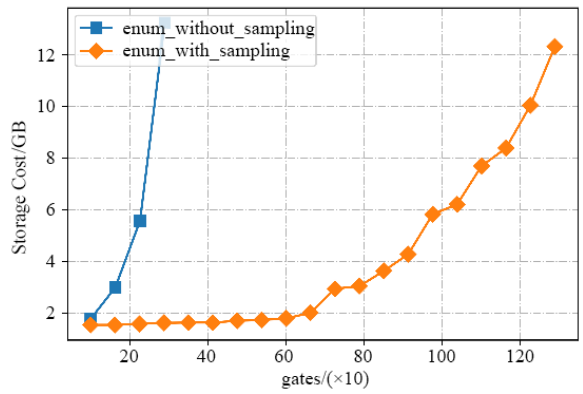


图6　AG$i\mathcal{O}$时间开销变化

另外,为了直观地体现输入电路C的特征,本文分析了GE子电路的抽样中,采用不同的q时等效子电路的覆盖情况,如图7所示. 显然当q越小时,在计算上越容易进行等效替换,反之,则需要更多轮的迭代尝试. 当q越小时,进行的等效电路替换仍然能够泄露电路局部的特征,例如,$q=2$时,自动搜索到的等效电路一般是两个门电路的对偶置换,并不能隐藏对电路局部的统计特征. 这种情况下,就需要对GE中的电路进行再次抽样和更高阶数的演化. 另外,局部的等效电路演化并不能完全保证电路规模的恒定,因此,需要设定额外的规模浮动变量$\Delta(G_i)=||G_i|-|C_i||$,当一轮迭代结束时,满足$\Delta(G_i)=0$. 我们用$n/N$作为门电路覆盖情况的衡量参数,其中$n$为等效替换操作的次数,$N$为输入规模,当该值逼近1时,意味着几乎所有的计算门都参与了电路演化. 针对不同的覆盖情况,在AG$i\mathcal{O}$的训练中适当的对q进行选择,可以进一步提高生成$\tilde{C}$的效率.

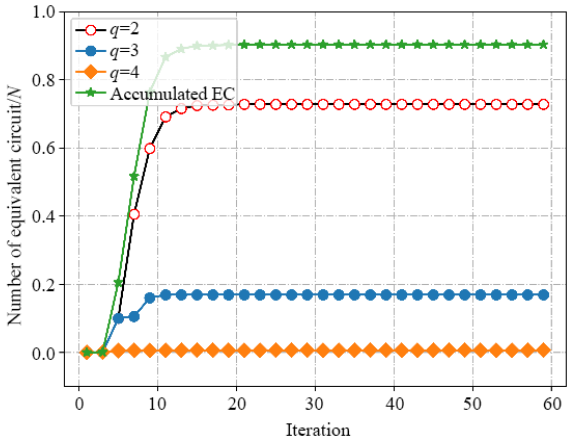


图7 G_0和G_1中差分演化情况(等效电路的覆盖率)

对于G_0和G_1的输出,判别器的会快速收敛,且所需Epoches不需要太大,当Epoches大于10时,判别器所使用的图神经网络均会稳定收敛,如图8所示. AG$i\mathcal{O}$的不可区分特性主要通过G_0和G_1中的差分演化实现,其收敛情况由判别器D输出的判定优势Adv体现,当判定优势Adv依λ可忽略时,我们认为达到了次优的不可区分性. 如图9所示,当迭代次数大于20时,图样本的特征迅速减少,Adv随着GE迭代次数的增加而迅速降低.

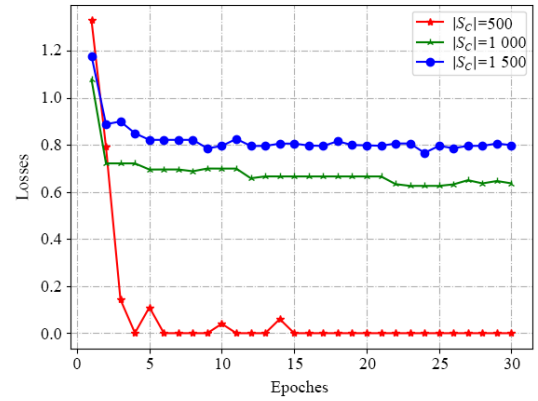


图8 D中的图神经网络收敛情况

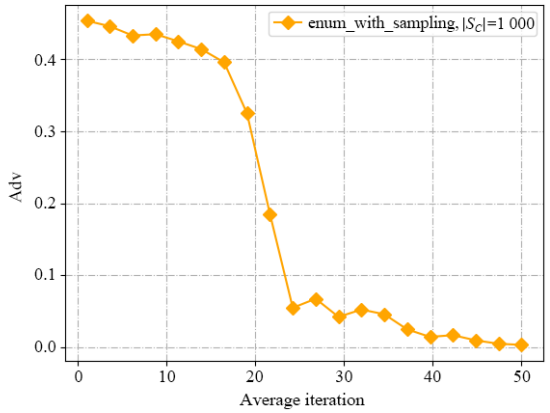


图9 判定优势收敛情况

7 结论

本文围绕$i\mathcal{O}$的构造问题,提出了一种新型的构造思想和技术方法,用于克服现有$i\mathcal{O}$构造中存在的方法和性能的瓶颈. AG$i\mathcal{O}$架构中,采用等效电路自动化搜索的全新方法解决通用$i\mathcal{O}$构造问题,即将通用电路生成映射到图神经网络构造问题中,基于图神经网络的自动演化技术,探索了一种满足统计上次优的不可区分性和功能保持性的通用$i\mathcal{O}$构造方法. 该$i\mathcal{O}$的基本架构是基于对偶的对抗性图神经网络架构,对于任意功能的输入电路C,通过图枚举得到备用的替代电路样本集合,然后使用差分演化算法分别独立优化上述图神经网络,并使用一个全局的判别优势监督图神经网络演化的方向,最终达到判别器对该图神经网络不可区分的状态.

AG$i\mathcal{O}$的构造中依然存在诸多不确定的因素,会导致潜在的安全问题. 图枚举部件、演化部件和对抗性判别器对于任意输入电路C的功能、拓扑、规模不做任何限定,中间处理结果仅与当前的枚举结果和判别器反馈的优势有关,因此可以达到通用电路处理的目的,但中间处理过程与输入电路规模有直接关系,因此,AG$i\mathcal{O}$会泄露出计算执行的资源耗费信息. 另外,AG$i\mathcal{O}$中对子电路的抽样方式直接决定了演化路径,从而对整个结构的安全性有重要影响,但本文尚未发现可以准确描述这种制约关系的方法. 总之,AG$i\mathcal{O}$的原型架构还存在诸多改进之处以及结构优化的余地.

参考文献

[1] BOAZ B, ODED G, RUSSELL I, et al. On the (im)possibility of obfuscating programs[EB/OL]. (2001) [2021]. https://eprint.iacr.org /2001/69.pdf.

[2] ZHANG F G, ZHANG Z. Garbled circuits and indistinguishability obfuscation[J]. Journal of Cryptologic Research, 2019, 6(5): 541.

[3] YAO A C C. How to generate and exchange secrets[C]// 27th Annual Symposium on Foundations of Computer Science (SFCS 1986). Piscataway: IEEE, 1986: 162-167.

[4] DAN BONEH, SAHAI AMIT, AND WATERS BRENT. FunctioNal encryption: Definitions and challenges[EB/OL]. (2010)[2021]. https://eprint.iacr.org/2010/543.pdf.

[5] BRAKERSKI Z, SEGEV G. Function-private functional encryption in the private-key setting[J]. Journal of Cryptology, 2018, 31(1): 202-225.

[6] LIU X M, ZHAO Q S, ZENG Q K. Verifiable computation using re-randomizable garbled circuits[J]. Journal of Software, 2019, 30(2): 399.

[7] GARG S, GENTRY C, HALEVI S, et al. Candidate indis-

tinguishability obfuscation and functional encryption for all circuits[J]. SIAM Journal on Computing, 2016, 45(3): 882-929.

[8] BITANSKY N, VAIKUNTANATHAN V. Indistinguishability obfuscation from functional encryption[C]//2015 IEEE 56th Annual Symposium on Foundations of Computer Science. Piscataway: IEEE, 2015: 1-37.

[9] 戚珉, 陈明 . 标准模型下基于身份的混淆乐观公平交换方案[J]. 电子学报, 2020, 48(8): 1516-1527.
QI M, CHEN M. ID-based ambiguous optimistic fair exchange in the standard model[J]. Acta Electronica Sinica, 2020, 48(8): 1516-1527. (in Chinese)

[10] 宋秀丽, 周道洋, 文爱君 . d维(t,n)门限量子同态加密算法的设计与仿真[J]. 电子学报, 2020, 48(5): 846-853.
SONG X L, ZHOU D Y, WEN A J. Design and simulation of d dimensional (t, n) threshold quantum homomorphic encryption algorithm[J]. Acta Electronica Sinica, 2020, 48(5): 846-853. (in Chinese)

[11] GARG S, GENTRY C, HALEVI S. Candidate multilinear maps from ideal lattices[C]//Advances in Cryptology-EUROCRYPT 2013 (Lecture Notes in Computer Science). Berlin: Springer, 2013: 1-17.

[12] JEAN-SBASTIEN C, TANCRDE L, MEHDI T. Practical multilinear maps over the integers[C]//Advances in Cryptolog-CRYPTO 2013 (Lecture Notes in Computer Science). Berlin: Springer, 2013: 476-493.

[13] GENTRY C, GORBUNOV S, HALEVI S. Graph-Induced multilinear maps from Lattices[C]//Theory of Cryptography. Heidelberg: Springer, 2015: 498-527.

[14] SAHAI A, WATERS B. How to use indistinguishability obfuscation: Deniable encryption, and more[C]//Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing. New York: ACM, 2014: 475-484.

[15] GARG SANJAM, GENTRY CRAIG, HALEVI SHAI, et al. Two-round secure mpc from indistinguishability obfuscation[C]//Theory of Cryptography. Berlin: Springer, 2014: 74-94.

[16] BONEH D, ZHANDRY M. Multiparty key exchange, efficient traitor tracing, and more from indistinguishability obfuscation[J]. Algorithmica, 2017, 79(4): 1233-1285.

[17] CHEON J H, HAN K, LEE C M, et al. CryptaNalysis of the multilinear map over the integers[C]//Advances in Cryptology- EUROCRYPT 2015. Berlin: Springer, 2015: 3-12.

[18] CHEN Y L, GENTRY C, HALEVI S. CryptaNalyses of candidate branching program obfuscators[C]//Advances in Cryptology-EUROCRYPT 2017. Berlin: Springer, 2017: 278-307.

[19] BRAKERSKI Z, ROTHBLUM G N. Virtual black-box obfuscation for all circuits via generic graded encoding[M]// Theory of Cryptography. Berlin: Springer, 2014: 1-25.

[20] BOAZ B, SANJAM G, YAEL T K, et al. Protecting obfuscation against algebraic attacks[C]//Advances in Cryptology - EUROCRYPT 2014. Berlin: Springer, 2014: 221-238.

[21] GARG S, MILES E, MUKHERJEE P, et al. Secure obfuscation in a weak multilinear map model[C]//Proceedings, Part II, of the 14th International Conference on Theory of Cryptography - Volume 9986. New York: ACM, 2016: 241-268.

[22] LIN H J. Indistinguishability obfuscation from constant-degree graded encoding schemes[C]//Advances in Cryptology-EUROCRYPT 2016 (Lecture Notes in Computer Science). Berlin: Springer, 2016: 28-57.

[23] LIN H J, TESSARO S. Indistinguishability obfuscation from trilinear maps and block-wise local PRGS[C]//Advances in Cryptology-CRYPTO 2017 (Lecture Notes in Computer Science). Cham: Springer, 2017: 630-660.

[24] ANANTH P, JAIN A, LIN H J, et al. Indistinguishability obfuscation without multilinear maps: New paradigms via low degree weak pseudorandomness and security amplification[C]//Advances in Cryptology- CRYPTO 2019 (Lecture Notes in Computer Science). Cham: Springer, 2019: 284-332.

[25] JAIN A, LIN H J, MATT C, et al. How to leverage hardness of constant-degree expanding polynomials over R to build iO[C]//Advances in Cryptology-EUROCRYPT 2019. Cham: Springer, 2019: 251-281.

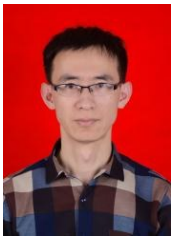
[26] DÍEZ J, DEL COZ J J, LUACES O, et al. Using tensor products to detect unconditional label dependence in multilabel classifications[J]. Information Sciences: An International Journal, 2016, 329(1): 20-32.

[27] AGRAWAL S. Indistinguishability obfuscation without multilinear maps: New methods for bootstrapping and instantiation[C]//Advances in Cryptology-EUROCRYPT 2019. Cham: Springer International Publishing, 2019: 191-225.

[28] AGRAWAL S, PELLET-MARY A. Indistinguishability obfuscation without maps: Attacks and fixes for noisy linear FE[C]//Advances in Cryptology-EUROCRYPT 2020. Cham: Springer International Publishing, 2020: 110-140.

[29] BRAKERSKI Z, DÖTTLING N, GARG S, et al. Candidate iO from homomorphic encryption schemes[J]. Journal of Cryptology, 2023, 36(3): 1-41.
[30] LINDELL Y, PINKAS B. A proof of security of Yao's protocol for two-party computation[J]. Journal of Cryptology, 2009, 22(2): 161-188.
[31] BELLARE M, HOANG V T, ROGAWAY P. Foundations of garbled circuits[C]//Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York: ACM, 2012: 784-796.
[32] GOLDWASSER S, KALAI Y, POPA R A, et al. Reusable garbled circuits and succinct functional encryption [C]//Proceedings of the Forty-Fifth Annual ACM Symposium on Theory of Computing. New York: ACM, 2013: 555-564.
[33] WU Z H, PAN S R, CHEN F W, et al. A comprehensive survey on graph neural networks[J]. IEEE Transactions on Neural Networks and Learning Systems, 2021, 32(1): 4-24.
[34] 车向北, 康文倩, 邓彬, 等. 一种基于图神经网络的SDN路由性能测试模型[J]. 电子学报, 2021, 49(3): 484-491.
CHE X B, KANG W Q, DENG B, et al. A prediction model of SDN routing performance based on graph neural network[J]. Acta Electronica Sinica, 2021, 49(3): 484-491.
[35] STORN R, PRICE K. Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces[J]. Journal of Global Optimization, 1997, 11 (4): 341-359.
[36] WANG D X, CUI P, ZHU W W. Structural deep network embedding[C]//Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM, 2016: 1225-1234.
[37] WANG J S. A general algorithm for enumerating some subgraphs of a simple graph[J]. Journal of Computers, 1986, 1: 39-45.
[38] HALEVI S, HALEVI T, SHOUP V, et al. Implementing BP-obfuscation using graph-induced encoding[C]//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM, 2017: 783-798.
[39] APON D, HUANG Y, KATZ J, et al. Implementing cryptographic program obfuscation[EB/OL]. (2014) [2021]. https://eprint.iacr.org/2014/779.pdf.

作者简介



朱率率 男,1985年出生,山东高青人. 博士. 武警工程大学密码工程学院. 副教授. 主要研究方向为密码算法设计、信息安全.
E-mail: zsseupap@163.com

韩益亮 男,1977年出生,甘肃会宁人. 武警工程大学密码工程学院教授,博士生导师. 主要研究方向为公钥密码学、人工智能、算法设计.
E-mail: Yilianghan@hotmail.com

李 鱼 男,1995年出生,重庆人. 武警工程大学密码工程学院博士研究生. 主要研究方向为密码协议,信息安全.