标准PRF假设下基于立体几何变换的轻量级混淆电路协议

谭振华 宁婧宇
（东北大学软件学院 沈阳 110819）

摘 要 混淆电路(Garbled Circuit, GC)是安全两方计算(Secure Two-Party Computation, S2PC)的重要基础协议.为保证安全性,GC协议需要调用加密算法对电路中的门信号进行加密混淆.当前,GC协议构造每个二元门(如与门)需调用4次加密算法,标准伪随机函数(Pseudorandom Function, PRF)假设下,每个二元门的混淆表至少包含2个密文.如何有效降低加密算法调用次数与混淆表规模,是GC协议提升性能的主要研究问题.本文在标准PRF假设下,提出了一种基于立体几何变换的轻量级混淆电路协议SGT-GC,根据每类二元门信号逻辑设计了专门的立体几何变换,并替代传统的加密算法实现混淆门的构造.其中,对于每个二元混淆与门(AND Gate),首先将其4种可能的输入组合(00,01,10,11)转换为三维空间中不共圆的4个点坐标P_{00}、P_{01}、P_{10}、P_{11},经过逻辑值为FALSE的三个点(P_{00},P_{01},P_{10})构造圆,然后在经过圆心的圆平面法线上取任意点C_i,并满足该点到P_{00}、P_{01}、P_{10}的距离相等且不同于到逻辑值为TRUE的点P_{11}的距离.则该随机点C_i即可作为二元与门混淆表中的交换信息,其通信成本变成1,且不再需要额外的加密算法调用.对于二元混淆异或门以及一元非门,本文也进行了专门的设计并给出了详细的协议过程与数学论证.本文所提出的SGT-GC协议中,每个混淆表中仅需1个共享交换信息,且不需调用任何额外加密算法,避免了多次调用复杂的加密算法所造成的计算成本及传输混淆表中多条密文所造成的通信成本.安全性证明表明,本文所提协议在半诚实模型下满足隐私性、不经意性和可认证性.

关键词 混淆电路;安全两方计算;立体几何变换;标准伪随机函数假设;安全协议
中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2023.02240

Lightweight Garbled Circuit Protocol Based on Solid Geometry Transformation under Standard PRF Assumption

TAN Zhen-Hua NING Jing-Yu
(*Software College*, *Northeastern University*, *Shenyang* 110819)

Abstract Garbled Circuit (GC) plays a critical role in Secure Two-Party Computation (S2PC). Since Boolean circuits reveal the input information of participants, Yao proposed the first Garbled Circuit, which could protect the security and privacy during communication. After Yao's, how to optimize GC has attracted great attention in decades, from the perspectives of computation or communication. To ensure the security, GC protocols consistently encrypt the gates in the circuits by encryption algorithms. Each gate invokes the encryption algorithm 4 times. Under standard pseudorandom function (PRF) assumption, the garbled table of each AND gate consists of 2 ciphertexts. Despite the secure requirement is liberalized to non-standard assumption, the garbled table of each AND gate consists of 1.5 ciphertexts. Invoking encryption algorithms and transferring garbled tables leads to prohibitively high cost of computation and communication. In

收稿日期:2023-01-11;在线发布日期:2023-07-10.本课题得到国家重点研发计划基金资助项目(No. 2019YFB1405803)、中央高校基本科研业务费专项资金(N2217001)、国家自然科学基金资助项目(No. 61772125)资助.**谭振华**(通信作者),博士,教授,中国计算机学会(CCF)高级会员,主要研究领域为安全多方计算与隐私保护. E-mail: tanzh@mail.neu.edu.cn. **宁婧宇**,博士研究生,主要研究领域为安全多方计算构造块与隐私保护.

view of this, this paper proposes a lightweight garbled circuit based on solid geometry transformation under standard PRF assumption, called SGT-GC. Without invoking any encryption algorithms, SGT-GC satisfies the security requirements during the garbling process with special solid geometry transformations. Each wire has two options, namely TRUE and FALSE, thus each binary gate has four input options (00, 01, 10, 11). Using distinct solid geometry transformations, four input options are converted into coordinates of four points in three dimensions (P_{00}, P_{01}, P_{10}, P_{11}), from which the Generator can calculate the coordinates of a specific point that is regarded as the shared information in garbled table. For each garbling AND gate in SGT-GC, the input options (00, 01, 10) output FALSE and (11) outputs TRUE. We select a random point C_i on the normal of the circle surface passing through the center of the circle which constructed by points P_{00}, P_{01} and P_{10}. Such a C_i has the same distance to points P_{00}, P_{01} and P_{10}, but has the different distance to P_{11}. As a result, we use the random C_i as the only shared information between the Generator and the Evaluator, and let the distances between C_i and the points be the output garbled values. For each garbling XOR gate, the input options (00, 11) output FALSE and (01, 10) output TRUE. We construct two perpendicular bisection planes of the line segment with endpoints (P_{01}, P_{10}) and the line segment with endpoints (P_{00}, P_{11}), respectively, and select a random point C_i on the intersection line of the two perpendicular bisection planes as the shared information. The point C_i has the same distance to endpoints of a same line segment, but has different distances to endpoints of different line segments, thus let the two distances be the output garbled values. Consequently, each gate in SGT-GC has only one shared information in the garbled table, and requires no calls to encryption, which avoids the computation cost caused by the complex encryption algorithms and the communication cost caused by the transmission of multiple ciphertexts. In the evaluation process, each binary gate only needs to calculate the Euclidean distance once, enhancing the computation efficiency of the Evaluator. Consequently, the proposed protocol is more applicable to the situation where the Evaluator exhibits low computation performance while the Generator exhibits high computation performance. The security proof demonstrates that the proposed SGT-GC satisfies the security requirements of privacy, obliviousness and authenticity against semi-honest attackers.

Keywords　garbled circuit; secure two-party computation; solid geometry transformation; standard pseudorandom function assumption; secure protocol

1 引　言

在信息时代，对数据的计算不再局限于单一参与方，越来越多的应用场景需要两个参与方进行联合计算，为保证联合计算过程中双方隐私不被泄露，且双方均能得到正确的计算结果，安全两方计算(Secure Two-Party Computation, S2PC)[1,2]成为了隐私保护领域的重要研究课题.

在S2PC中，任意函数可被表示为由门和导线组成的布尔电路形式. 由于布尔电路泄露了两参与方的输入信息，姚期智院士在1986年首次提出了著名的混淆电路(Garbled Circuit, GC)协议[3]，以保证电路计算过程中的安全性[4]. GC的基本思想是使用混淆值表示导线的布尔值，使用加密算法实现门函数. 两参与方分别为Generator和Evaluator. 具体来说，Generator随机为电路中的每根导线选择两个混淆值，分别用来表示0和1. GC中的每个二元门有2根输入线，因此存在4种可能的输入混淆值组合，分别以上述4种组合为密钥，以对应的输出混淆值为明文，执行加密算法，可得4个密文，即混淆表. Generator将混淆表发送给Evaluator，由于Evaluator只掌握每个门的一种输入混淆值组合，因此可通过解密得到唯一的输出混淆值.

在之后的几十年中，产生了一系列方法以提升GC的计算和通信性能. 一些方法通过减少对加密

算法的调用次数来提高GC的计算性能，例如Point-Permute[5]，Free-XOR[6]，FleXOR[7]，Half-Gates[8]，GLNP[9]和Three Half-Gates[10]. 一些方法通过减少混淆表中的密文数量来提高GC的通信性能，例如4-3 GRR[11]，4-2 GRR[2]，Free-XOR[6]，FleXOR[7]，Half-Gates[8]，GLNP[9]和Three Half-Gates[10]. 还有一些文献构建了可重用的混淆电路[12-15]，通过并行实现的方法提升协议效率. GC具有重要的理论和实践意义，已被广泛应用于多种S2PC协议的构建[1,16-20].

现存所有GC方案都需要调用加密算法，如哈希函数、密钥派生函数和对称加密函数，以获取混淆表中的密文. 虽然高效对称加密函数[9-10,21-25]的应用在一定程度上降低了计算成本，但GC协议中的大部分计算开销仍来自加密和解密操作. 并且在标准伪随机函数(Pseudorandom Function, PRF)假设下，GC中每个门的实现需要Generator向Evaluator传输至少2个密文，占用了大部分通信成本.

本文的核心动机是构造一种不依赖于加密算法的GC协议，消除由加密算法导致的计算和通信成本，并保证其安全性与基于加密算法的GC相同. 鉴于此，本文在标准PRF假设下，提出了一种基于立体几何变换的轻量级混淆电路协议SGT-GC，不需调用任何加密算法，但可通过特殊立体几何变换保证GC的安全需求.

对于一个二元与门，共有4种可能的输入组合，仅当两个输入参数都为TRUE时返回TRUE，当一个或两个参数为FALSE时返回FALSE. 因此本文设计了一种混淆与门，通过4种可能的输入组合表示2种输出结果. 其中3种包含FALSE的输入组合(<0,0>, <0,1>, <1,0>)的输出为FALSE，1种仅包含TRUE的输入组合(<1,1>)的输出为TRUE. 根据与门的以上性质，在SGT-GC中，Generator首先将4种可能的输入组合转换为三维空间中四个不共圆的点P_{00}，P_{01}，P_{10}和P_{11}，其中P_{00}，P_{01}和P_{10}对应输入中存在FALSE的组合，P_{11}对应两个输入值均为TRUE的组合. 若P_{00}，P_{01}，P_{10}三点不共线，则可确定唯一的圆. 设O_i为此圆的圆心，则过O_i的圆的法线上任意一点C_i到P_{00}，P_{01}，P_{10}三点的距离均相等，到P_{11}的距离不同. 因此将C_i到P_{00}，P_{01}，P_{10}的距离作为输出FALSE的混淆值，将C_i到P_{11}的距离作为输出TRUE的混淆值. 然后将C_i作为共享信息放入混淆表. Evaluator只掌握4种输入组合中的1种，因此可计算出P_{00}，P_{01}，P_{10}和P_{11}中的一个点P_i，并获得P_i到C_i的距离，作为该与门的输出混淆值. 这种立体几何变换可以满足混淆与门输入输出之间的对应关系，并且Generator生成的混淆表中只有一个信息C_i共享给Evaluator. 本文将所提出的混淆与门命名为SGT-AND，详细过程将在第4节介绍.

异或门在2种输入值不同的组合下输出TRUE，在2种输入值相同的组合下输出FALSE. 因此本文设计了一种混淆异或门，通过4种可能的输入组合表示2种输出结果. 其中2种输入值相同的组合(<0,0>, <1,1>)的输出为FALSE，2种输入值不同的组合(<0,1>, <1,0>)的输出为TRUE. 因此在SGT-GC中，Generator首先将4种可能的输入组合转换为三维空间中四个点P_{00}，P_{01}，P_{10}和P_{11}，其中P_{00}和P_{11}对应输出为FALSE组合，P_{01}和P_{10}对应输出为TRUE组合，线段$P_{00}P_{11}$与线段$P_{01}P_{10}$不平行. 设直线L_i为线段$P_{00}P_{11}$垂直平分面与线段$P_{01}P_{10}$垂直平分面的交线，C_i为L_i上任意一点，则点C_i到同一线段两端点的距离相同，但到不同线段端点的距离不同. 因此，将C_i到P_{00}和P_{11}的距离设为FALSE所对应的输出混淆值，将C_i到P_{01}和P_{10}的距离设为TRUE所对应的输出混淆值，并将C_i作为共享信息放入混淆表. Evaluator根据所掌握的输入组合可计算出一个点P_i，并获得P_i到C_i的距离，作为该异或门的输出混淆值. 以上立体几何变换可以满足混淆异或门输入输出之间的对应关系，而且Generator共享给Evaluator的混淆表中也只有一条信息C_i. 本文将所提出的混淆异或门命名为SGT-XOR. SGT-XOR的详细过程将在第4节介绍.

本文的主要贡献如下：

(1)首次提出基于立体几何变换的混淆电路协议SGT-GC，以立体几何变换替代传统加密算法，在Evaluator的求值过程中，每个二元门仅需计算一次欧氏距离，避免了调用复杂的加密算法所导致的计算成本，提升了参与方Evaluator的计算性能.

(2)根据每类二元门信号逻辑设计了专门的立体几何变换，以一个随机点作为二元门混淆表中的交换信息，将每个二元门混淆表中的共享信息数降低为1，提升了协议的通信性能.

安全证明结果表明，在标准PRF假设下，本文协议是半诚实安全的，满足隐私性、不经意性和可认证性的安全需求.

2　相关工作

自GC出现以来，研究者提出了多种方法以提高GC的性能. 本节将概述现有的高效GC协议，并将重点放在计算成本和通信成本的分析上. 由于非门、与门和异或门是构成电路的基本元素，且目前所有GC协议对非门的处理方法相同，均为对换表示0和1的混淆值，因此本节不对非门进行分析，只介绍与门和异或门.

现存GC协议主要可分为两类. 第一类基于标准PRF假设[2,3,5,9,11]，第二类基于循环相关鲁棒性哈希函数假设(Circular Correlation Robust hash function，CCR)[6-8,10]，满足PRF假设的GC安全性强于满足CCR假设的GC. 下面分别阐述.

标准的PRF假设下，混淆异或门和混淆与门的计算均需要调用加密算法. 在文献[3]中，Generator生成每个混淆门需要调用4次加密算法，分别以4种可能的输入混淆值为密钥，对输出混淆值进行加密，因此每个门的混淆表中有4个密文. 为了避免密文的排列顺序泄露门的信息，Generator将混淆表中的密文随机排列. Evaluator掌握1种输入混淆值，可以对4个密文的其中1个密文进行解密，但由于密文进行了随机排列，Evaluator无法准确选择与其混淆值对应的密文. 因此，Evaluator需要以其掌握的输入混淆值为密钥，对4个密文进行逐一解密，再通过解密结果进行判定. 最好的情况是解密的第1个密文与其混淆值对应，最坏的情况是解密的第4个密文与其混淆值对应，因此Evaluator平均进行2.5次解密操作. 针对文献[3]无法准确选择对应密文的问题，Point-permute技术[5]在每个混淆值后附加一个置换比特，每根导线上的两个混淆值具有相反的置换比特. 置换比特和逻辑真值之间没有对应关系. 混淆表中的密文根据该门输入线的置换比特进行排列，因此Evaluator可以通过置换比特选择对应的密文，以此将解密操作降低为1次. 4-3 GRR技术[11]不再随机选取输出线路的混淆值，而是通过选取特定的混淆值使得混淆表的第一个密文为全零比特串. 因此Generator不需发送第一个密文，从而将Generator在每个门的密文发送数量从4个减少为3个. 4-2 GRR技术[2]基于Shamir秘密共享算法[26]，分别用1次多项式和2次多项式构造混淆异或门和与门，将密文数量减少到2个. 文献[9]同样实现了密文数为2个的混淆与门，用2个密文的异或结果作为第3个密文. 对于混淆异或门，为输入和输出线设置了共同偏置，这个偏置对应加密后的混淆值，因此每个门的偏置不同. 该技术将混淆异或门的密文数降低为1个.

在CCR假设[27]下，仅混淆与门的计算需要调用加密算法. Free-XOR[6]在构造混淆异或门时，直接将两个输入混淆值进行异或，得出输出线上的混淆值，因此不需要加密操作和传输混淆表. Free-XOR的混淆与门构造方法与4-3 GRR相同，因此其每个混淆与门的混淆表中有3个密文. FleXOR方案[7]允许参与者对偏置进行不同设置，根据不同的偏置设计方法，混淆异或门密文个数可以为0、1或2，在密文数为0时，方案等价于Free-XOR. FleXOR可以与4-2 GRR的混淆与门兼容. Half-gates方案[8]保留了Free-XOR对混淆异或门的构造方法，对混淆与门进行了改进. 将混淆与门分解为两个与门异或的形式，称这两个与门为半门，分别调用Garbler-half-gate和Evaluator-half-gate算法计算两个半门，最后将两个半门的计算结果相异或得到输出混淆值. Half-gates将混淆与门密文数减少为2个. Three Half-Gates方案[10]对Half-gates进行了改进，仍然使用Free-XOR进行混淆异或操作，但在计算混淆与门时，设计了slicing和dicing方法，使用三个半门，将密文降低为1.5个.

本文在以上新近GC协议的启发下，拟设计一种不调用任何加密算法的混淆门构造方法，充分利用立体几何的变换原理，以立体几何的随机变换替代传统的加密算法实现混淆门的构造，将每个二元门混淆表的信息数降低为1个，并满足标准PRF假设下的隐私性、不经意性和可认证性的安全需求.

3　预备知识

3.1　混淆电路协议的基础算法

本文沿用文献[28]对混淆电路的相关定义.

定义1.　混淆电路协议中包括4种基础算法.

(1)混淆算法：$(F,e,d)\leftarrow Garble(1^{\kappa},f)$. 混淆算法的作用是根据布尔电路生成对应的混淆电路. 其中，算法的输入为安全参数1^{κ}和双方进行联合计算的布尔电路f，输出为混淆电路F、编码信息e和解码信息d.

(2)编码算法：$X\leftarrow Encode(e,x)$. 编码算法的作用是将双方的真实输入转换为输入混淆值. 算法

输入为编码信息e和双方的真实输入x,输出为真实输入对应的混淆值X.

(3)求值算法:$Y\leftarrow Eval(F,X)$. 算法的作用是通过混淆电路F和输入混淆值X求出输出混淆值Y. 布尔电路f和混淆电路F的门、导线及其连接方式相同,但计算方式不同.

(4)解码算法:$y\leftarrow Decode(Y,d)$. 算法的作用是根据解码信息d将输出混淆值Y解码为真实输出y.

以上4种算法之间的关系如图1所示. 首先,布尔电路f经混淆算法$Garble(\cdot)$生成混淆电路、编码信息e和解码信息d. 然后,编码算法$Encode(\cdot)$根据编码信息将双方的真实输入编码为输入混淆值. 之后,求值算法$Eval(\cdot)$通过混淆电路F和输入混淆值X求出输出混淆值Y. 最后,解码算法$Decode(\cdot)$根据解码信息将输出混淆值Y解码为真实输出y.

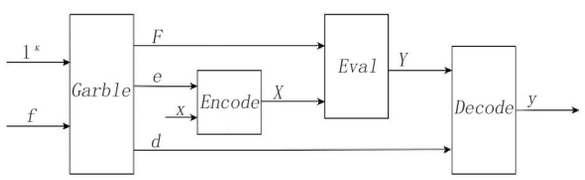

图1 混淆电路协议的基础算法

定义2. 正确性. 对于所有布尔电路f和任意输入x,在执行混淆算法$(F,e,d)\leftarrow Garble(1^{\kappa},f)$后等式(1)成立,则混淆电路协议满足正确性需求.

$$Decode(Eval(F,Encode(e,x)),d)=f(x) \quad (1)$$

其中$f(x)$为不使用混淆电路协议时,布尔电路的正常计算结果. 也就是说,若参与方均遵循协议,则能够计算得到正确的输出结果.

3.2 安全需求

本文协议是半诚实安全的,其安全性满足以下3种安全需求[9].

定义3. 隐私性. 对于任意布尔电路f和真实输入x,参与方生成的混淆分布(F,X,d)与概率多项式时间模拟器$\mathcal{S}(1^{\kappa},f,f(x))$生成的随机混淆结构$(F,X,d)$不可区分. 图2为构造上述$(F,X,d)$的两种方法,其中,$Simulate(\cdot)$为模拟器$\mathcal{S}$对布尔电路$f$的模拟函数.

$(F,e,d)\leftarrow Garble(1^{\kappa},f)$ $X\leftarrow Encode(e,x)$ return (F,X,d)	$(F,X,d)\leftarrow Simulate(1^{\kappa},f,f(x))$ return (F,X,d)

图2 隐私性的判别参数生成方法

定义4. 不经意性. 对于任意布尔电路f和真实输入x,参与方生成的混淆分布(F,X)与概率多项式时间模拟器$\mathcal{S}(1^{\kappa},f)$生成的随机混淆结构$(F,X)$不可区分. 图3为上述两种构造$(F,X)$的方法,其中,$Simulate(\cdot)$为模拟器$\mathcal{S}$对布尔电路$f$的模拟函数.

$(F,e,d)\leftarrow Garble(1^{\kappa},f)$ $X\leftarrow Encode(e,x)$ return (F,X)	$(F,X)\leftarrow Simulate(1^{\kappa},f)$ return (F,X)

图3 不经意性的判别参数生成方法

定义5. 可认证性. 在只给定(F,X)的情况下,除了可忽略的概率,任何概率多项式时间敌手$\mathcal{A}$无法生成$\tilde{Y}\neq Eval(F,X)$,使得$Decode(\tilde{Y},d)\neq\perp$.

3.3 伪随机函数

定义6. 伪随机函数[29] (Pseudorandom Function, PRF). A和B为有限集合,令$\mathcal{F}=\{F:A\rightarrow B\}$为一个函数簇,连带一个有效的抽样分布(即$\mathcal{F},A,B$由安全参数$\kappa$进行索引). 若下述2个交互是计算不可区分的,则称$\mathcal{F}$是一个伪随机函数簇.

1. 选取函数$F\leftarrow\mathcal{F}$,给予敌手对$F(\cdot)$适应性的oracle预言访问权.

2. 选取一致随机函数$U:A\rightarrow B$,给予敌手对$U(\cdot)$适应性oracle预言访问权.

4 基于立体几何变换的GC协议

SGT-GC针对混淆与门和混淆异或门,分别设计了不同的立体几何变换规则,生成了SGT-AND和SGT-XOR. 协议参数如表1所示.

表1 协议参数

参数	含义
w_i	编号为i的导线
v_i	w_i的真值,$v_i\in\{0,1\}$
g_i	编号为i的门
n	f(或F)中全部导线条数
m	f(或F)中输入线条数
l	f(或F)中输出线条数
p	f(或F)中门的个数
W_i^0,W_i^1	w_i上0和1分别对应的混淆值
$e[i,0],e[i,1]$	w_i上0和1分别对应的编码信息
$d[i,0],d[i,1]$	w_i上0和1分别对应的解码信息
C_i	g_i混淆表中的共享信息

定义7.　电路的输入线和输出线.　电路的输入线指，只作为某些门输入线而不作为任何门输出线的导线. 电路的输出线指，只作为某些门输出线而不作为任何门输入线的导线.

设f中的导线共n条，分别为$w_1,\cdots,w_n$，其中输入线有m条为$w_1,\cdots,w_m$，输出线有l条为$w_{n-l+1},\cdots,w_n$，其中$m+l<n$. 每个门的编号与其输出线编号相同，且满足每个门的输出线编号值大于输入线. 设f中的门有p个，分别为$g_{m+1},\cdots,g_n$，其中$m+p=n$.

每个二元门g_i有两根输入线和一根输出线，如图4所示，导线w_a和w_b为门的输入线，导线w_i为门的输出线. 每条导线都有0或1两个可能的布尔值，将其编码为两个混淆值W_i^0和W_i^1，其中上标表示布尔值，下标表示导线的编号.

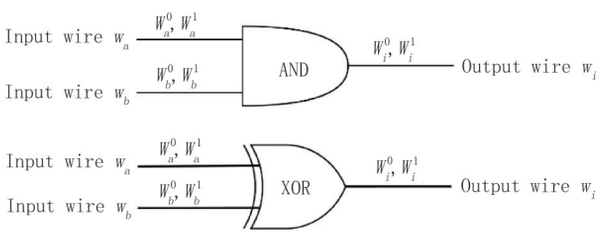

图4　混淆与门和混淆异或门

本文协议的两个参与方分别为Generator和Evaluator. 首先，Generator执行混淆算法$Garbled(\cdot)$和编码算法$Encode(\cdot)$，其中编码算法$Encode(\cdot)$的执行过程中需要与Evaluator进行不经意传输，细节将在4.2节详细描述. 之后Generator将混淆电路F和输入混淆值X发送给Evaluator. Evaluator执行求值算法$Eval(\cdot)$，并将输出混淆值Y发送给Generator. 最后，Generator对输出混淆值Y进行解码，得到电路的真正输出. 图5为SGT-GC时序图.

4.1　混淆算法

混淆算法是混淆电路协议的核心算法，由Generator实现，目的是将布尔电路转换为混淆电路，并生成相应的编码信息和解码信息.

首先，Generator初始化电路的所有输入线. 对于每条输入线i，随机选择两个长度为κ的混淆值W_i^0和W_i^1，分别表示0和1. 编码信息e记录每条输入线上0和1分别对应的混淆值，即令$e[i,0]=W_i^0$和$e[i,1]=W_i^1$，编码信息的结构如表2所示. 初始化算法$Initialize(\cdot)$如算法1所示.

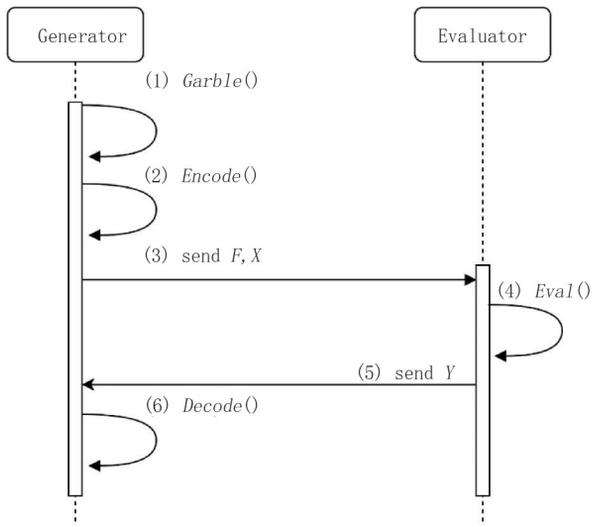

图5　SGT-GC时序图

表2　编码信息

输入线编号	真值	混淆值
1	0	$e[1,0]$
	1	$e[1,1]$
2	0	$e[2,0]$
	1	$e[2,1]$
⋮	⋮	⋮
	⋮	⋮
m	0	$e[m,0]$
	1	$e[m,1]$

算法1.　初始化算法$Initialize(\cdot)$.

输入：安全参数1^κ，布尔电路f

输出：f的输入线混淆值$W_i^0, W_i^1(i\in[1,m])$，编码信息e

1. FOR $i=1:m$
2. $\quad W_i^0\leftarrow\{0,1\}^\kappa$
3. $\quad W_i^1\leftarrow\{0,1\}^\kappa$
4. $\quad e[i,0]\leftarrow W_i^0$
5. $\quad e[i,1]\leftarrow W_i^1$
6. END FOR

然后，按照门编号由小到大的顺序，依次对每个门进行混淆操作. 由于每个门的输出线编号值大于输入线，因此在计算每个二元门g_i时，已知输入线w_a,w_b上的输入混淆值$W_a^0, W_a^1, W_b^0, W_b^1$，需要求得输出线$w_i$上的输出混淆值$W_i^0, W_i^1$和混淆表信息$C_i$.

当对与门进行混淆操作时，Generator首先将输入线上混淆值的4种组合$(W_a^{v_a}, W_b^{v_b})$转换为4个三维笛卡尔坐标系中的点坐标$P_{v_av_b}$：$<x_{v_av_b}, y_{v_av_b}, z_{v_av_b}>$，$v_a,v_b\in\{0,1\}$. 具体方法为：Generator将$W_a^{v_a}$和$W_b^{v_b}$按位异或，得到比特串$E_{v_av_b}=W_a^{v_a}\oplus W_b^{v_b}$，然后按照

平均分割的原则将$E_{v_av_b}$划分为3份，分别作为$P_{v_av_b}$的坐标$<x_{v_av_b}, y_{v_av_b}, z_{v_av_b}>$，如图6所示. 由于$E_{v_av_b}$的长度为$\kappa$，为了实现平均分割，本文要求$\kappa$为3的整数倍，即$3|\kappa$.

坐标获取算法*GetCoordinate*(·)的具体描述如算法2所示. 其中*Length*(·)函数的输入为比特串，输出为该比特串的长度. 指令*SHR*(a,b)将目的操作数a逻辑右移b位，并返回移位后的结果. 首先通过按位与操作获取$E_{v_av_b}$的最低$length(E_{v_av_b})/3$位，作为$z_{v_av_b}$. 然后将$E_{v_av_b}$右移$length(E_{v_av_b})/3$位得到E_{temp}，通过按位与操作获取E_{temp}的最低$length(E_{v_av_b})/3$位，作为$y_{v_av_b}$. 最后通过将E_{temp}右移$length(E_{v_av_b})/3$位得到$x_{v_av_b}$.

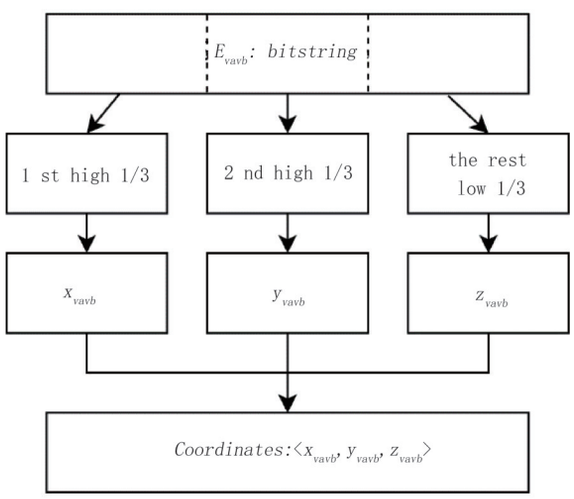

图6 比特串转换为三维坐标的方法

算法2. 坐标获取算法*GetCoordinate*(·).

输入：混淆值组合($W_a^{v_a}$, $W_b^{v_b}$)

输出：三维坐标$P_{v_av_b}$: $<x_{v_av_b}, y_{v_av_b}, z_{v_av_b}>$

1. $E_{v_av_b} \leftarrow W_a^{v_a} \oplus W_b^{v_b}$
2. $z_{v_av_b} \leftarrow E_{v_av_b}$ AND$(2^{length(E_{v_av_b})/3}-1)$
3. $E_{temp} \leftarrow SHR(E_{v_av_b}, length(E_{v_av_b})/3)$
4. $y_{v_av_b} \leftarrow E_{temp}$ AND$(2^{length(E_{v_av_b})/3}-1)$
5. $x_{v_av_b} \leftarrow SHR(E_{temp}, length(E_{v_av_b})/3)$

在得到4个三维点坐标P_{00},P_{01},P_{10}和P_{11}后进行判断，如果P_{00},P_{01}和P_{10}共线，当前程序终止，由初始化算法*Initialize*(·)重新进行计算. 除了可忽略的概率，P_{00},P_{01},P_{10}三点不会共线，本文计算得出P_{00},P_{01},P_{10}三点共线的概率小于$2^{-2\kappa/3}$，下面给出计算过程.

由于P_{00},P_{01},P_{10}和P_{11}四个点的坐标均为$2^{\kappa/3}$长的比特串，因此可以将每个点映射到三维笛卡尔坐标系中，坐标的取值范围为0到$2^{\kappa/3}-1$的整数，即三维空间中0到$2^{\kappa/3}-1$的整数格. 取值范围内共有2^{κ}个格，排列成边长为$2^{\kappa/3}$的正方体. 任意直线所经过的格最多为$2^{\kappa/3}$个. 当P_{00},P_{01},P_{10}三点中两点位于一条直线，第三点可能的位置共有2^{κ}个，其中与前两点共线的位置最多有$2^{\kappa/3}$个，此时三点共线的概率为$2^{-2\kappa/3}$. 当前两点所在直线经过的整数点数小于$2^{\kappa/3}$时，三点共线的概率将小于$2^{-2\kappa/3}$. 综上所述，P_{00},P_{01},P_{10}三点共线的概率小于$2^{-2\kappa/3}$，在计算上为可忽略的概率. 设P_{00}和P_{01}所确定的直线的方向向量为(a_1, b_1, c_1)，P_{00}和P_{10}所确定的直线的方向向量为(a_2, b_2, c_2)，则共线判定算法如算法3所示.

算法3. 共线判定算法*Collinear*(·).

输入：方向向量(a_1, b_1, c_1),(a_2, b_2, c_2)

输出：比特值*Colli*

1. IF a_1: b_1: c_1 = = a_2: b_2: c_2 THEN
2. *Colli* = 1
3. ELSE
4. *Colli* = 0
5. END IF

P_{00},P_{01},P_{10}三点在同一平面上，设该平面的法向量为(a, b, c)，则有：

$$\begin{aligned} ax_{00}+by_{00}+cz_{00}&=1\\ ax_{01}+by_{01}+cz_{01}&=1\\ ax_{10}+by_{10}+cz_{10}&=1 \end{aligned} \tag{2}$$

令：

$$\boldsymbol{A}=\begin{bmatrix} x_{00} & y_{00} & z_{00}\\ x_{01} & y_{01} & z_{01}\\ x_{10} & y_{10} & z_{10} \end{bmatrix} \tag{3}$$

则：

$$\begin{cases} a=\dfrac{1}{\det(\boldsymbol{A})}\begin{vmatrix} 1 & y_{00} & z_{00}\\ 1 & y_{01} & z_{01}\\ 1 & y_{10} & z_{10} \end{vmatrix}\\ b=\dfrac{1}{\det(\boldsymbol{A})}\begin{vmatrix} x_{00} & 1 & z_{00}\\ x_{01} & 1 & z_{01}\\ x_{10} & 1 & z_{10} \end{vmatrix}\\ c=\dfrac{1}{\det(\boldsymbol{A})}\begin{vmatrix} x_{00} & y_{00} & 1\\ x_{01} & y_{01} & 1\\ x_{10} & y_{10} & 1 \end{vmatrix} \end{cases} \tag{4}$$

由此可得P_{00},P_{01},P_{10}三点所在平面方程为

$$ax+by+cz=1 \tag{5}$$

经过P_{00},P_{01},P_{10}三点的圆有且只有一个，设O_i: $<x_o, y_o, z_o>$为该圆的圆心，设R_i为半径. 有：

$$\begin{cases}(x_{00}-x_o)^2+(y_{00}-y_o)^2+(z_{00}-z_o)^2=R_i^2\\(x_{01}-x_o)^2+(y_{01}-y_o)^2+(z_{01}-z_o)^2=R_i^2\\(x_{10}-x_o)^2+(y_{10}-y_o)^2+(z_{10}-z_o)^2=R_i^2\end{cases}\tag{6}$$

可得：

$$\begin{cases}(x_{01}-x_{00})x_o+(y_{01}-y_{00})y_o+(z_{01}-z_{00})z_o=\\\dfrac{x_{01}^2+y_{01}^2+z_{01}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\\(x_{10}-x_{00})x_o+(y_{10}-y_{00})y_o+(z_{10}-z_{00})z_o=\\\dfrac{x_{10}^2+y_{10}^2+z_{10}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\end{cases}\tag{7}$$

由于圆心也在P_{00},P_{01},P_{10}三点所在的平面上，由公式(5)和公式(7)可得：

$$\begin{bmatrix}a & b & c\\(x_{01}-x_{00}) & (y_{01}-y_{00}) & (z_{01}-z_{00})\\(x_{10}-x_{00}) & (y_{10}-y_{00}) & (z_{10}-z_{00})\end{bmatrix}\begin{bmatrix}x_o\\y_o\\z_o\end{bmatrix}=\begin{bmatrix}1\\\dfrac{x_{01}^2+y_{01}^2+z_{01}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\\\dfrac{x_{10}^2+y_{10}^2+z_{10}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\end{bmatrix}\tag{8}$$

令：

$$\boldsymbol{D}=\begin{bmatrix}a & b & c\\(x_{01}-x_{00}) & (y_{01}-y_{00}) & (z_{01}-z_{00})\\(x_{10}-x_{00}) & (y_{10}-y_{00}) & (z_{10}-z_{00})\end{bmatrix}\tag{9}$$

$$\boldsymbol{B}=\begin{bmatrix}1\\\dfrac{x_{01}^2+y_{01}^2+z_{01}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\\\dfrac{x_{10}^2+y_{10}^2+z_{10}^2-(x_{00}^2+y_{00}^2+z_{00}^2)}{2}\end{bmatrix}\tag{10}$$

根据克莱默法则(Cramer′s Rule)，当$\det(\boldsymbol{D})\neq 0$时，公式(8)有唯一解. 由于$P_{00}$,$P_{01}$,$P_{10}$三点不共线，因此向量$(x_{01}-x_{00},y_{01}-y_{00},z_{01}-z_{00})$与向量$(x_{10}-x_{00},y_{10}-y_{00},z_{10}-z_{00})$不平行. 向量$(a,b,c)$是$P_{00}$,$P_{01}$,$P_{10}$三点所在平面法向量，因此向量$(a,b,c)$与向量$(x_{01}-x_{00},y_{01}-y_{00},z_{01}-z_{00})$、向量$(x_{01}-x_{00},y_{01}-y_{00},z_{01}-z_{00})$均不平行. 因此$\det(\boldsymbol{D})\neq 0$. 则有：

$$x_o=\frac{\det(\boldsymbol{D}_1)}{\det(\boldsymbol{D})},y_o=\frac{\det(\boldsymbol{D}_2)}{\det(\boldsymbol{D})},z_o=\frac{\det(\boldsymbol{D}_3)}{\det(\boldsymbol{D})}\tag{11}$$

其中$\boldsymbol{D}_i$为将矩阵$\boldsymbol{D}$的第i列替换为列向量$\boldsymbol{B}$之后所得的矩阵. 由公式(11)可得圆心O_i的坐标$<x_o,y_o,z_o>$.

设直线L_i为过点O_i的圆面的法线，P:$<x,y,z>$为法线L_i上任意一点，则有：

$$\begin{cases}(P-O_i)\cdot(P_{00}-P_{01})=0\\(P-O_i)\cdot(P_{00}-P_{10})=0\end{cases}\tag{12}$$

其中，符号“·”表示数量积. 将坐标值带入公式(12)，可得法线方程：

$$\begin{cases}(x-x_o)(x_{00}-x_{01})+(y-y_o)(y_{00}-y_{01})+\\(z-z_o)(z_{00}-z_{01})=0\\(x-x_o)(x_{00}-x_{10})+(y-y_o)(y_{00}-y_{10})+\\(z-z_o)(z_{00}-z_{10})=0\end{cases}\tag{13}$$

设C_i:$<x_i,y_i,z_i>$为法线L_i上任意一点，则C_i到P_{00},P_{01},P_{10}三点的距离均相等，设这个距离为输出线w_i上表示0的混淆值W_i^0，即：

$$W_i^0=Dis(P_{00},C_i)=Dis(P_{01},C_i)=Dis(P_{10},C_i)\tag{14}$$

如图7所示. 设C_i到P_{11}的距离为输出线w_i上表示1的混淆值$W_i^1=Dis(P_{11},C_i)$. 设门g_i混淆表中的唯一共享信息为C_i. 除了可忽略的概率，P_{00},P_{01},P_{10}和P_{11}四点不共圆，本文计算得出四点共圆的概率小于$2^{-\kappa/3}$，下面给出计算过程.

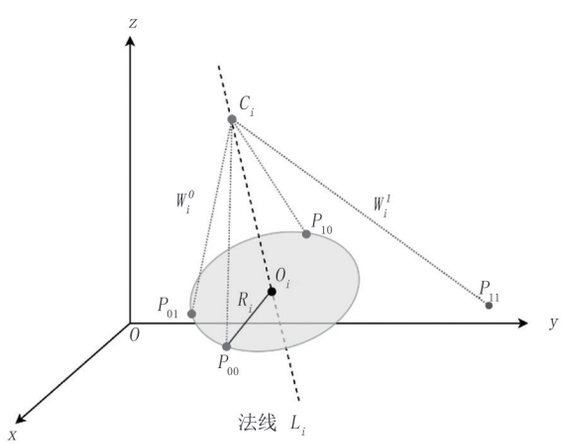

图7　与门的混淆过程

由于P_{00},P_{01},P_{10}和P_{11}四个点的坐标均为$2^{\kappa/3}$长的比特串，因此可以将每个点映射到三维笛卡尔坐标系中，坐标的取值范围为0到$2^{\kappa/3}-1$的整数，即三维空间中0到$2^{\kappa/3}-1$的整数格. 取值范围内共有2^{κ}个格，排列成边长为$2^{\kappa/3}$的正方体. 任意平面所经过的格最多为$2^{2\kappa/3}$个. P_{00},P_{01},P_{10}和P_{11}中，任意三点必然共面，这个平面经过的格最多为$2^{2\kappa/3}$个. 另一点共有2^{κ}个格可取，若与前三点共面则最多有$2^{2\kappa/3}$个格可取. 因此四点共面的概率小于$2^{-\kappa/3}$，则四点共圆的概率也小于$2^{-\kappa/3}$. 因此除了在可以忽略的概率下，$W_i^0\neq W_i^1$. 与门混淆算法$SGT\text{-}AND(\cdot)$如

算法4所示.

算法4. 与门混淆算法$SGT\text{-}AND(\cdot)$.

输入:输入线混淆值$(W_a^0, W_a^1, W_b^0, W_b^1)$

输出:输出线混淆值(W_i^0, W_i^1),共享信息C_i

1. FOR $v_a=0:1$
2. FOR $v_b=0:1$
3. $P_{v_av_b} \leftarrow GetCoordinate(W_a^{v_a}, W_b^{v_b})$
4. END FOR
5. END FOR
6. IF not $Collinear(\cdot)$ THEN
7. $L_i \leftarrow (P_{00}, P_{01}, P_{10})$
8. $C_i \leftarrow L_i$
9. $W_i^0 \leftarrow Dis(P_{00}, C_i)$
10. $W_i^1 \leftarrow Dis(P_{11}, C_i)$
11. END IF

当对异或门g_i进行混淆操作时,Generator首先将输入线上混淆值的4种组合$(W_a^{v_a}, W_b^{v_b})$转换为4个三维点坐标$P_{v_av_b}$: $<x_{v_av_b}, y_{v_av_b}, z_{v_av_b}>$, $v_a, v_b \in \{0,1\}$,具体方法与算法2相同. 在得到4个三维点坐标P_{00},P_{01},P_{10}和P_{11}后进行判断,若线段$P_{00}P_{11}$与线段$P_{01}P_{10}$平行,当前程序终止,由初始化算法$Initialize(\cdot)$重新进行计算. 除了可忽略的概率,线段$P_{00}P_{11}$与线段$P_{01}P_{10}$不会平行,本文计算得出线段$P_{00}P_{11}$与线段$P_{01}P_{10}$平行的概率小于$2^{-2\kappa/3}$,下面给出计算过程. 同与门中计算三点共圆概率的方法类似,由于在三维笛卡尔坐标系中,P_{00},P_{01},P_{10}和P_{11}的坐标均在三维空间中0到$2^{\kappa/3}-1$的整数格上,在这个范围内的直线所经过的整数点最多为$2^{\kappa/3}$个. 当其中三点位置确定时,一条线段已经确定,第二条线段只确定一点,另一点可能的位置共有2^{κ}个,其中能构成线段$P_{00}P_{11}$与线段$P_{01}P_{10}$平行的位置最多有$2^{\kappa/3}$个,因此线段$P_{00}P_{11}$与线段$P_{01}P_{10}$平行的概率小于$2^{-2\kappa/3}$,在计算上为可忽略的概率. 设P_{00}和P_{11}所确定的直线的方向向量为(a_1, b_1, c_1),P_{01}和P_{10}所确定的直线的方向向量为(a_2, b_2, c_2),则平行判定算法如算法5所示.

算法5. 平行判定算法$Parallel(\cdot)$.

输入:方向向量(a_1, b_1, c_1),(a_2, b_2, c_2)

输出:比特值$Para$

1. IF $a_1:b_1:c_1 = = a_2:b_2:c_2$ THEN
2. $Para=1$
3. ELSE
4. $Para=0$
5. END IF

设S_1与S_2分别为线段$P_{00}P_{11}$与线段$P_{01}P_{10}$的中点,即:

$$\begin{cases} S_1 = < \dfrac{x_{00}+x_{11}}{2}, \dfrac{y_{00}+y_{11}}{2}, \dfrac{z_{00}+z_{11}}{2} > \\ S_2 = < \dfrac{x_{01}+x_{10}}{2}, \dfrac{y_{01}+y_{10}}{2}, \dfrac{z_{01}+z_{10}}{2} > \end{cases} \tag{15}$$

设$Plane$-1为过S_1且垂直于$P_{00}P_{11}$的平面,$Plane$-2为过S_2且垂直于$P_{01}P_{10}$的平面,根据垂直关系,可得平面$Plane$-1和$Plane$-2的交线L_i方程为:

$$\begin{cases} (x_{11}-x_{00})x+(y_{11}-y_{00})y+(z_{11}-z_{00})z+d_1=0 \\ (x_{10}-x_{01})x+(y_{10}-y_{01})y+(z_{10}-z_{01})z+d_2=0 \end{cases} \tag{16}$$

将S_1与S_2带入方程(16),可解得参数d_1和d_2:

$$\begin{cases} d_1=\dfrac{1}{2}(x_{00}^2+y_{00}^2+z_{00}^2-(x_{11}^2+y_{11}^2+z_{11}^2)) \\ d_2=\dfrac{1}{2}(x_{01}^2+y_{01}^2+z_{01}^2-(x_{10}^2+y_{10}^2+z_{10}^2)) \end{cases} \tag{17}$$

如图8,由于$Plane$-1是线段$P_{00}P_{11}$的垂直平分面,因此$Plane$-1上任意一点到点P_{00}和P_{11}的距离相等,同理$Plane$-2上任意一点到点P_{01}和P_{10}的距离相等.

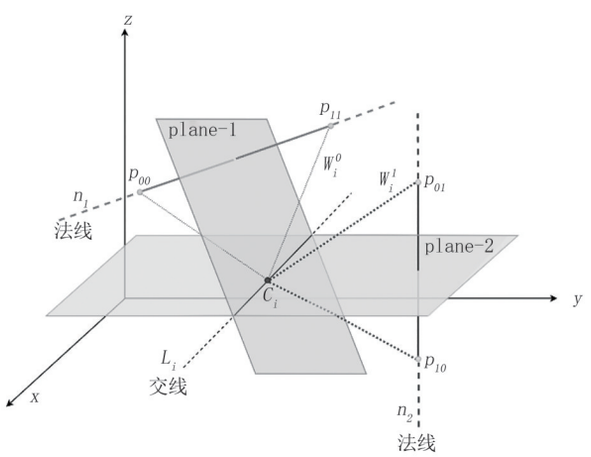

图8 异或门的混淆过程

设C_i为L_i上任意一点,由于L_i为平面$Plane$-1和$Plane$-2的交线,则C_i到点P_{00}和P_{11}的距离相等,设这个距离为$W_i^0 = Dis(P_{00}, C_i) = Dis(P_{11}, C_i)$. 同理,$C_i$到点$P_{01}$和$P_{10}$的距离相等,设此距离为$W_i^1 = Dis(P_{01}, C_i) = Dis(P_{10}, C_i)$. 设门$g_i$混淆表中的唯一共享信息为$C_i$,异或门混淆操作完成. 异或门混淆算法$SGT\text{-}XOR(\cdot)$如算法6所示.

算法6. 异或门混淆算法$SGT\text{-}XOR(\cdot)$.

输入:输入线混淆值$(W_a^0, W_a^1, W_b^0, W_b^1)$

输出:输出线混淆值(W_i^0, W_i^1),共享信息C_i

1. FOR $v_a=0:1$
2. FOR $v_b=0:1$
3. $P_{v_a v_b} \leftarrow GetCoordinate(W_a^{v_a}, W_b^{v_b})$
4. END FOR
5. END FOR
6. IF not $Para()$ THEN
7. $L_i \leftarrow (P_{00}, P_{01}, P_{10}, P_{11})$
8. $C_i \leftarrow L_i$
9. $W_i^0 \leftarrow Dis(P_{00}, C_i)$
10. $W_i^1 \leftarrow Dis(P_{01}, C_i)$
11. END IF

对非门的混淆操作不需共享信息，只将混淆值进行对调，具体算法如算法7所示.

算法7.　非门混淆算法$SGT\text{-}NOT(\cdot)$.

输入：输入线混淆值(W_a^0, W_a^1)

输出：输出线混淆值(W_i^0, W_i^1)

1. $W_i^0=W_a^1$
2. $W_i^1=W_a^0$

通过单个与门、异或门、非门可以构造完整的电路，对于导线连接的两个相邻的门，一个门的输出线为另一个门的输入线. 图9为一个混淆电路示意图，其中门i为异或门，输入为混淆值$W_a^0, W_a^1, W_b^0, W_b^1$，这四个混淆值由算法1所描述的初始化算法$Initialize(\cdot)$生成. 门$i$的输出为混淆值$W_i^0, W_i^1$，由异或门混淆算法$SGT\text{-}XOR(\cdot)$生成. 门$j$为与门，输入为混淆值$W_b^0, W_b^1, W_c^0, W_c^1$，由算法1所描述的初始化算法$Initialize(\cdot)$生成，输出为混淆值$W_j^0, W_j^1$，由与门混淆算法$SGT\text{-}AND(\cdot)$生成. 门$k$为与门，输入为混淆值$W_i^0, W_i^1, W_j^0, W_j^1$，分别为门$i$和门$j$的输出，门$k$的输出为混淆值$W_k^0, W_k^1$，由与门混淆算法$SGT\text{-}AND(\cdot)$生成.

混淆算法$Garble(\cdot)$如算法8所示. 通过上述方法将混淆门算法$SGT\text{-}AND(\cdot)$、$SGT\text{-}XOR(\cdot)$和$SGT\text{-}NOT(\cdot)$扩展为整个混淆电路，由此可以构造完整的混淆算法$Garble(\cdot)$. 首先对电路进行初始化，获取电路所有输入线$w_1, \cdots, w_m$的混淆值$W_i^0, W_i^1 (i\in[1, m])$. 然后按照门编号从小到大的顺序，逐一对门$g_{m+1}, \cdots, g_n$进行混淆计算. 最后将电路每条输出线$w_{n-l+1}, \cdots, w_n$上表示0和1的混淆值，分别赋值给解码信息$d$，如表3所示.

算法8.　混淆算法$Garble(\cdot)$.

输入：安全参数1^κ，布尔电路f

输出：混淆电路F，编码信息e，解码信息d

1. $(W_i^0, W_i^1, e) \leftarrow Initialize(1^\kappa, f)$ //$(i\in[1, m])$
2. FOR $i=(m+1):1:n$
3. IF g_i是与门 THEN
4. $(W_i^0, W_i^1, C_i) \leftarrow SGT-AND(W_a^0, W_a^1, W_b^0, W_b^1)$
5. ELSE IF g_i是异或门
6. $(W_i^0, W_i^1, C_i) \leftarrow SGT-XOR(W_a^0, W_a^1, W_b^0, W_b^1)$
7. ELSEIF g_i是非门
8. $(W_i^0, W_i^1) \leftarrow SGT-NOT(W_a^0, W_a^1)$
9. END IF
10. END FOR
11. $F \leftarrow C$
12. FOR $i=(n-l+1):1:n$
13. $d[i, 0] \leftarrow W_i^0$
14. $d[i, 1] \leftarrow W_i^1$
15. END FOR

表3　解码信息

输出线编号（电路后l条导线）	真值	混淆值
n	0	$d[n,0]$
	1	$d[n,1]$
$n-1$	0	$d[n-1,0]$
	1	$d[n-1,1]$
⋮	⋮	⋮
	⋮	⋮
$n-l+1$	0	$d[n-l+1,0]$
	1	$d[n-l+1,1]$

4.2　编码算法

编码算法$Encode(\cdot)$由Generator和Evaluator共同实现，目的是根据编码信息e将双方的真实输入x编码为对应的输入混淆值X，作为混淆电路F的输入. Generator和Evaluator分别持有对电路f的部分真实输入. 编码信息e由Generator持有，4.1节算法1描述了Generator生成编码信息e的方法.

对于导线i，编码信息包括两条，分别为$e[i, 0]=W_i^0$和$e[i, 1]=W_i^1$. 因此Generator对所持有的每个输入$x_i\in x_G$，可直接进行编码，即当$x_i=0$时，令$X_i=e[i, 0]$，当$x_i=1$时，$X_i=e[i, 1]$. 最后，将所得X_G发送给Evaluator.

Evaluator持有输入x_E，而不掌握编码信息，因此需要与Generator进行不经意传输[30-33]（Oblivious Transfer，OT）. 对于每个输入$x_i\in x_E$，Evaluator提供x_i，Generator提供$e[i, 0]$和$e[i, 1]$，通过OT，Evaluator得到$e[i, x_i]$，而Generator无法获得任何信息. 最后Evaluator获得输入值x_E所对应的输入混淆值X_E. 编码算法执行过程如图10所示.

通过编码算法，Evaluator获得真实输入x对应

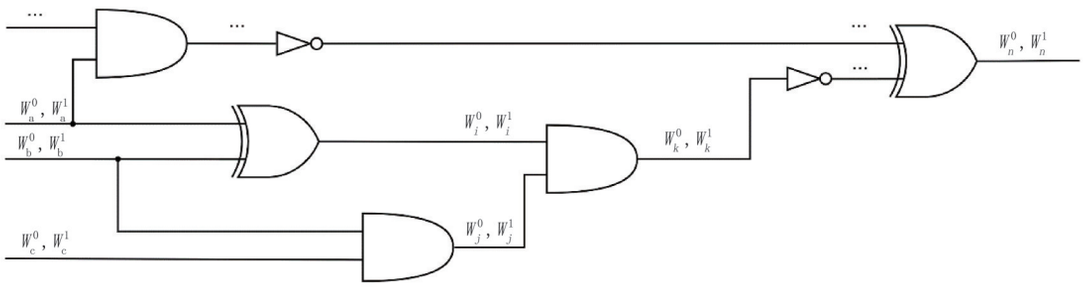

图9 由混淆门组成的混淆电路

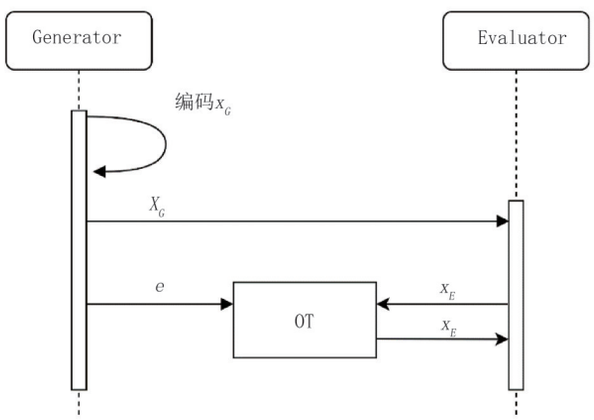

图10 编码算法执行过程

的输入混淆值$X=X_G\cup X_E$. 编码算法如算法9所示.

算法9. 编码算法$Encode(\cdot)$.

输入:编码信息e,双方真实输入x

输出:输入混淆值X

1. FOR $i=1:m$
2. IF $x_i\in x_G$ THEN
3. $X_i\leftarrow e[i,x_i]$
4. ELSE IF $x_i\in x_E$
5. $X_i\leftarrow OT(e,x_i)$
6. END IF
7. END FOR

4.3 求值算法

求值算法$Eval(\cdot)$由Evaluator实现,目的是根据混淆电路F和输入混淆值X计算输出混淆值Y. 输入混淆值X提供了混淆电路F每条输入线上的一个混淆值,Evaluator按照门编号由小到大对门进行计算. 对于每个二元门g_i,Evaluator可以获取共享信息C_i,然后根据算法2,通过两条输入线上的混淆值W_a, W_b计算一个三维点坐标P_i,并计算C_i和P_i之间的距离,作为输出混淆值W_i. 对于非门,令输出混淆值等于输入混淆值.

在计算所有门之后,将F的所有输出线的混淆值作为输出混淆值Y. 具体算法如算法10所示.

算法10. 求值算法$Eval(\cdot)$.

输入:混淆电路F,输入混淆值X

输出:输出混淆值Y

1. FOR $i=1:m$
2. $W_i\leftarrow X_i$
3. END FOR
4. $C\leftarrow F$
5. FOR $i=(m+1):1:n$
6. IF g_i是二元门 THEN
7. $P_i\leftarrow GetCoordinate(W_a, W_b)$
8. $W_i\leftarrow Dis(C_i, P_i)$
9. ELSEIF g_i是非门
10. $W_i\leftarrow W_a$
11. END IF
12. END FOR
13. FOR $i=(n-l+1):1:n$
14. $Y_i\leftarrow W_i$
15. END FOR

4.4 解码算法

解码算法$Decode(\cdot)$由Generator实现,目的是根据解码信息d,将Evaluator发送的输出混淆值Y解码为真实输出y. 对于导线i,解码信息包括两条,分别为$d[i,0]=W_i^0$和$d[i,1]=W_i^1$,如果混淆值$Y_i=d[i,0]$,则真实输出$y_i=0$,如果混淆值$Y_i=d[i,1]$,则真实输出$y_i=1$. 因此Generator可以根据获得其所对应的真实输出y. 解码算法$Decode(\cdot)$如算法11所示.

算法11. 解码算法$Decode(\cdot)$.

输入:解码信息d,输出混淆值Y

输出:真实输出y

1. FOR $i=(n-l+1):1:n$
2. IF $Y_i=d[i,0]$ THEN
3. $y_i\leftarrow 0$
4. ELSE IF $Y_i=d[i,1]$
5. $y_i\leftarrow 1$

6. END IF

7. END FOR

4.5 正确性证明

本文在定义2中明确了正确性的定义，本节证明所提SGT-GC协议满足正确性需求.

定理1. SGT-GC协议满足正确性需求. 即当$(F,e,d)\leftarrow Garble(1^{\kappa},f)$时，对于所有$x$，$Decode(Eval(F,Encode(e,x)),d)=f(x)$成立. 其中$f(x)$为不使用混淆电路协议时，布尔电路的正常计算结果.

证明. 对于一个布尔电路f，经混淆电路计算的结果应与直接计算布尔电路的结果相同. 本文将针对电路中的与门、异或门和非门三种基础门进行分别讨论.

对于与门，根据算法4，Generator得到输出线混淆值(W_i^0,W_i^1)和共享信息C_i，其中混淆值$W_i^0=Dis(P_{00},C_i)=Dis(P_{01},C_i)=Dis(P_{10},C_i)$，混淆值$W_i^1=Dis(P_{11},C_i)$. 由于在实际应用中，共享信息$C_i$可能为无理数，因此Generator需要对$C_i$进行精度限制. 同时，$W_i^0$和$W_i^1$需要精确到整数. 在上述条件下，有可能出现等式$[Dis(P_{00},C_i)]=[Dis(P_{01},C_i)]=[Dis(P_{10},C_i)]$不成立的异常情况，其中符号[ ]表示对数值进行四舍五入取整. 本文研究发现，当C_i的精度为$10^{-\kappa}$时，这种异常情况出现的概率小于$9\times10^{-2\kappa}$. 下面给出计算方法.

当对C_i: $<x_i,y_i,z_i>$的三个坐标进行精度为$10^{-\kappa}$的四舍五入操作后，每个坐标值的变化量小于$\frac{1}{2}\times10^{-\kappa}$，因此$C_i$点的位移小于$\sqrt{(\frac{1}{2}\times10^{-\kappa})^2+(\frac{1}{2}\times10^{-\kappa})^2+(\frac{1}{2}\times10^{-\kappa})^2}=\frac{\sqrt{3}}{2}\times10^{-\kappa}$. C_i点的位移会导致C_i到圆环上点的距离发生改变，距离变化量同样小于$\frac{\sqrt{3}}{2}\times10^{-\kappa}$. 由于距离计算结果保留整数部分，因此这个变化量在多数情况下不会对计算结果产生影响，除了一种特殊情况，即C_i点的位移导致C_i到圆环上两点的距离在保留整数后不相等，例如，当$Dis(P_{00},C_i)=\frac{1}{2}+10^{-\kappa}$且$Dis(P_{01},C_i)=\frac{1}{2}-10^{-\kappa}$时，$Dis(P_{00},C_i)$四舍五入的结果为1，而$Dis(P_{01},C_i)$四舍五入的结果为0. 当$[Dis(P_{00},C_i)]$，$[Dis(P_{01},C_i)]$，$[Dis(P_{10},C_i)]$中的两个值分别小于和大于x.5时，其中x是整数，会出现这种情况，概率为

$$\begin{aligned}&\Pr[\text{NotEqual}]\\&=(C_3^1\times2\times\frac{\sqrt{3}}{2}\times10^{-\kappa})\times(C_2^1\times\frac{\sqrt{3}}{2}\times10^{-\kappa})\\&=9\times10^{-2\kappa}\end{aligned}\tag{18}$$

其中C_n^m表示从n个不同元素中任取m个元素的组合数，这种小概率异常在实际应用中是可以接受的.

Evaluator可以获取共享信息C_i，根据两条输入线上的混淆值W_a, W_b可计算一个三维点坐标P_i，并计算两点距离得到输出混淆值$W_i=Dis(P_i,C_i)$. 当$(a,b)\in\{(0,0),(0,1),(1,0)\}$，即$P_i\in\{P_{00},P_{01},P_{10}\}$时，输出混淆值表示0，即$W_i=W_i^0$. 当$(a,b)=(1,1)$，即$P_i=P_{11}$时，输出混淆值表示1，即$W_i=W_i^1$. 当导线$w_i$为电路的输出线，解码信息为$d[i,0]=W_i^0$，$d[i,1]=W_i^1$.

当$(a,b)\in\{(0,0),(0,1),(1,0)\}$时，布尔电路的输出为$f(x)_i=0$，其中$f(x)_i$为$w_i$上的真实输出比特值. 混淆电路中$W_i=d[i,0]$，因此混淆电路的解码结果为$y_i=0$，可得$y_i=f(x)_i$. 当$(a,b)=(1,1)$时，布尔电路的输出为$f(x)_i=1$，混淆电路中$W_i=d[i,1]$，因此混淆电路的解码结果为$y_i=1$，可得$y_i=f(x)_i$.

对于每个异或门，根据算法6所述，Generator可计算得到输出线混淆值(W_i^0,W_i^1)和共享信息C_i，其中混淆值$W_i^0=Dis(P_{00},C_i)=Dis(P_{11},C_i)$，混淆值$W_i^1=Dis(P_{01},C_i)=Dis(P_{10},C_i)$. 和与门相同，在实际应用中，Generator需要对C_i进行精度限制. 同时，W_i^0和W_i^1需要精确到整数. 当C_i的精度为$10^{-\kappa}$时，由精度所产生的异常情况出现的概率小于$6\times10^{-2\kappa}$. 下面给出计算方法.

当对C_i: $<x_i,y_i,z_i>$的三个坐标进行精度为$10^{-\kappa}$的四舍五入操作后，每个坐标值的变化量小于$\frac{1}{2}\times10^{-\kappa}$，因此$C_i$点的位移小于$\sqrt{(\frac{1}{2}\times10^{-\kappa})^2+(\frac{1}{2}\times10^{-\kappa})^2+(\frac{1}{2}\times10^{-\kappa})^2}=\frac{\sqrt{3}}{2}\times10^{-\kappa}$. C_i点的位移会导致C_i到线段端点的距离发生改变，距离变化量同样小于$\frac{\sqrt{3}}{2}\times10^{-\kappa}$. 由于距离计算结果保留整数部分，因此这个变化量在多数情况下不会对计算结果产生影响，除了一种特殊情况，即C_i点的位移导致C_i到同一线段两端点的距离在保留

整数后不相等，例如，当$Dis(P_{00},C_i)=\frac{1}{2}+10^{-\kappa}$且$Dis(P_{11},C_i)=\frac{1}{2}-10^{-\kappa}$时，$Dis(P_{00},C_i)$四舍五入的结果为1，而$Dis(P_{11},C_i)$四舍五入的结果为0. 当$[Dis(P_{00},C_i)]$和$[Dis(P_{11},C_i)]$中的两个值分别小于和大于x.5，或者$[Dis(P_{01},C_i)]$和$[Dis(P_{10},C_i)]$中的两个值分别小于和大于x.5时，其中x是整数，会出现异常，概率为

$$\begin{aligned}&\Pr[\text{NotEqual}]\\&=C_2^1\times(C_2^1\times 2\times\frac{\sqrt{3}}{2}\times10^{-\kappa})\times(\frac{\sqrt{3}}{2}\times10^{-\kappa})\\&=6\times10^{-2\kappa}\end{aligned}\tag{19}$$

在实际应用中是可以接受的.

Evaluator可以获取共享信息C_i，根据两条输入线上的混淆值W_a, W_b计算一个三维点坐标P_i，并计算输出混淆值$W_i=Dis(P_i,C_i)$. 当$(a,b)\in\{(0,0),(1,1)\}$时，$P_i\in\{P_{00},P_{11}\}$，则$W_i=W_i^0$，当$(a,b)\in\{(0,1),(1,0)\}$时，$P_i\in\{P_{01},P_{10}\}$，则$W_i=W_i^1$. 对于电路的输出线，解码信息$d[i,0]=W_i^0$，$d[i,1]=W_i^1$. 当$(a,b)\in\{(0,0),(1,1)\}$时，布尔电路的输出为$f(x)_i=0$，混淆电路中$W_i=d[i,0]$，因此混淆电路的解码结果为$y_i=0$，可得$y_i=f(x)_i$. 当$P_i\in\{P_{01},P_{10}\}$时，布尔电路的输出为$f(x)_i=1$，混淆电路中$W_i=d[i,1]$，因此混淆电路的解码结果为$y_i=1$，可得$y_i=f(x)_i$.

对于非门，根据算法7，Generator将输入比特0和1对应的混淆值进行对调，因此Evaluator可直接将输入混淆值作为输出混淆值，在解码后会得到与输入比特相反的结果. 此时实际输出与布尔电路的输出相同.

综上所述，SGT-GC协议中的三种门，均满足$y=f(x)$. 根据定义1，$y=Decode(Eval(F,Encode(e,x)),d)$，因此可得$Decode(Eval(F,Encode(e,x)),d)=f(x)$. SGT-GC协议满足正确性需求. 证毕.

5 安全性证明

本文协议是半诚实安全的，其安全性需满足隐私性、不经意性和可认证性. 本节分别进行证明.

5.1 隐私性、不经意性证明

定理2. SGT-GC协议满足隐私性需求. 即对于任意布尔电路f和真实输入x，参与方生成的混淆分布(F,X,d)与概率多项式时间模拟器$\mathcal{S}(1^\kappa,f,f(x))$生成的随机混淆结构$(F,X,d)$不可区分.

证明. 对于布尔电路f中每条输入线w_i，$\mathcal{S}$随机生成输入混淆值X_i，令$W_i=X_i$. 然后按照门编号由小到大对每个门g_i进行混淆计算. 当g_i为二元门时，随机生成共享信息C_i，计算点坐标$P_i=GetCoordinate(W_a,W_b)$，并计算$W_i=Dis(P_i,C_i)$. 当$g_i$为非门时，令$W_i=W_a$. 最后，为电路$f$的所有输出线构造解码信息，令$d[i,f(x)_i]=W_i$，$d[i,\overline{f(x)_i}]=\{0,1\}^\kappa$，其中$\overline{f(x)_i}$为与$f(x)_i$相反的比特值，即对电路的正确输出$f(x)_i$，解码结果为特定的混淆值$W_i$，而对非正确的输出$\overline{f(x)_i}$，解码结果为随机生成的混淆值. 模拟器$\mathcal{S}$的模拟算法$Simulate(\cdot)$如算法12所示.

算法12. 模拟算法$Simulate(\cdot)$.

输入：安全参数1^κ，布尔电路f，双方正确输出$f(x)$

输出：混淆电路F，输入混淆值X，解码信息d

1. FOR $i=1:m$
2. $X_i\leftarrow\{0,1\}^\kappa$
3. $W_i\leftarrow X_i$
4. END FOR
5. FOR $i=(m+1):1:n$
6. IF g_i是二元门 THEN
7. $C_i\leftarrow\{0,1\}^\kappa$
8. $P_i\leftarrow GetCoordinate(W_a,W_b)$
9. $W_i\leftarrow Dis(C_i,P_i)$
10. ELSE IF g_i是非门
11. $W_i\leftarrow W_a$
12. END IF
13. END FOR
14. $F\leftarrow C$
15. FOR $i=(n-l+1):1:n$
16. $d[i,f(x)_i]=W_i$
17. $d[i,\overline{f(x)_i}]=\{0,1\}^\kappa$
18. END FOR

电路中门的个数为p，分别为$g_{m+1},\cdots,g_{m+p}$，其中$m+p=n$. 本文通过构造以下$p+1$个混合电路证明敌手$\mathcal{A}$不能区分(F,X,d)是由$\mathcal{S}$构造，还是由真实参与方构造.

设$M_j(1^\kappa,f,x)$，$0\leqslant j\leqslant p$为$p+1$个混合电路，其中$M_j(1^\kappa,f,x)$中的前j个门由模拟器$\mathcal{S}$按照算法12生成. 第$j+1,\cdots,p$个门按照混淆算法$SGT\text{-}AND(\,)$、$SGT\text{-}XOR(\,)$和$SGT\text{-}NOT(\,)$所述规则，由真实的混淆电路产生. 混淆算法要求每条输入线提

供0和1分别对应混淆值，而模拟器$\mathcal{S}$生成的每条导线上只有一个混淆值，因此本文在生成第$j+1,\cdots,p$个门之前首先检验其输入线是否包含两个混淆值，若输入线仅包含一个混淆值，则随机生成另一个混淆值. 在生成解码信息时，进行同样检验，若电路的输入线仅包含一个混淆值，则随机生成另一个混淆值. 显然，$M_0(1^\kappa,f,x)$是一个真实的混淆电路，$M_p(1^\kappa,f,x)$是算法12所介绍的模拟电路. $M_j(1^\kappa,f,x)$的具体实现算法如算法13所示.

算法13.　混合电路算法$M_j(\cdot)$.

输入：安全参数1^κ，布尔电路f，双方真实输入x

输出：混淆电路F，输入混淆值X，解码信息d

1. FOR $i=1:m$
2. $X_i\leftarrow\{0,1\}^\kappa$
3. $W_i\leftarrow X_i$
4. END FOR
5. IF $j>0$ THEN
6. FOR $i=(m+1):1:(m+j)$
7. IF g_i是二元门 THEN
8. $C_i\leftarrow\{0,1\}^\kappa$
9. $P_i\leftarrow GetCoordinate(W_a,W_b)$
10. $W_i\leftarrow Dis(C_i,P_i)$
11. ELSE IF g_i是非门
12. $W_i\leftarrow W_a$
13. END IF
14. END FOR
15. END IF
16. IF $j<p$ THEN
17. FOR $i=(m+j+1):1:n$
18. IF g_i是二元门 THEN
19. IF $v_a=0$ THEN
20. $W_a^0\leftarrow W_a$
21. $W_a^1\leftarrow\{0,1\}^\kappa$
22. ELSE
23. $W_a^0\leftarrow\{0,1\}^\kappa$
24. $W_a^1\leftarrow W_a$
25. END IF
26. IF $v_b=0$ THEN
27. $W_b^0\leftarrow W_b$
28. $W_b^1\leftarrow\{0,1\}^\kappa$
29. ELSE
30. $W_b^0\leftarrow\{0,1\}^\kappa$
31. $W_b^1\leftarrow W_b$
32. END IF
33. IF g_i是与门 THEN
34. $(W_i^0,W_i^1,C_i)\leftarrow SGT-AND(W_a^0,W_a^1,W_b^0,W_b^1)$
35. ELSE IF g_i是异或门
36. $(W_i^0,W_i^1,C_i)\leftarrow$SGT$-XOR(W_a^0,W_a^1,W_b^0,W_b^1)$
37. END IF
38. END IF
39. IF g_i是非门 THEN
40. IF $v_a=0$ THEN
41. $W_a^0\leftarrow W_a$
42. $W_a^1\leftarrow\{0,1\}^\kappa$
43. ELSE
44. $W_a^0\leftarrow\{0,1\}^\kappa$
45. $W_a^1\leftarrow W_a$
46. END IF
47. $(W_i^0,W_i^1)\leftarrow$SGT$-NOT(W_a^0,W_a^1)$
48. END IF
49. END FOR
50. END IF
51. $F\leftarrow C$
52. FOR $i=(n-l+1):1:n$
53. IF $W_i^0\neq$null &&. $W_i^1\neq$null THEN
54. $d[i,0]\leftarrow W_i^0$
55. $d[i,1]\leftarrow W_i^1$
56. ELSE
57. $d[i,f(x)_i]=W_i$
58. $d[i,\overline{f(x)_i}]=\{0,1\}^\kappa$
59. END IF
60. END FOR

混合电路M_{j-1}和M_j的区别在于门g_{m+j}的构造，在M_{j-1}中g_{m+j}由真实混淆电路生成，在M_j中g_{m+j}由模拟电路生成. 当g_{m+j}是非门时，无论真实电路还是模拟电路，敌手$\mathcal{A}$仅掌握g_{m+j}的输入混淆值，因此无法区分M_{j-1}和M_j. 当g_{m+j}是二元门时，$\mathcal{A}$掌握的输入混淆值仍相同，因此无法从输入混淆值区分M_{j-1}和M_j. 但在真实电路生成的门g_{m+j}中，共享信息C_{m+j}是由Generator通过混淆门算法计算得到，而模拟电路生成的门g_{m+j}中，共享信息C_{m+j}是由模拟器$\mathcal{S}$随机生成的. 因此，只有敌手$\mathcal{A}$能够区分以上两种不同方法生成的C_{m+j}，才能区分M_{j-1}和M_j.

当g_{m+j}为混淆与门时，共享信息C_{m+j}由与门混淆算法$SGT\text{-}AND(\cdot)$生成. C_{m+j}为法线L_{m+j}上任意一点，L_{m+j}通过P_{00}，P_{01}，P_{10}三点生成. 由于敌手只掌握一组输入混淆值，因此最多掌握P_{00}，P_{01}，P_{10}三点中的一点，在不确定另外两点的情况下，确定法线L_{m+j}的难度等同于在已知圆上一点的情况下，求解圆的法线方程. 因此对敌手来说，C_{m+j}与随机值不可区分，在混淆与门中C_{m+j}的生成方式符合伪随

机函数的定义要求.

当g_{m+j}为混淆异或门时,共享信息C_{m+j}由异或门混淆算法$SGT\text{-}XOR(\cdot)$生成.C_{m+j}为交线L_{m+j}上任意一点,L_{m+j}通过$P_{00},P_{01},P_{10},P_{11}$四点生成.由于敌手只掌握一组输入混淆值,因此最多掌握$P_{00},P_{01},P_{10},P_{11}$四点中的一点,在不确定另外三点的情况下,确定交线$L_{m+j}$的难度等同于仅已知两条线段其中一条的一个端点,求解两线段垂直平分面的交线.因此对敌手来说,C_{m+j}与随机值不可区分,在混淆异或门中C_{m+j}的生成方式符合伪随机函数的定义要求.

综上所述,敌手$\mathcal{A}$无法区分门g_{m+j}由真实混淆电路生成,还是由模拟电路生成,因此在$\mathcal{A}$的视角下有$M_{j-1}\equiv M_j$.由此可推出

$$M_0\equiv\ldots\equiv M_{j-1}\equiv M_j\equiv\ldots\equiv M_p \tag{20}$$

由于M_0是真实的混淆电路,M_p是模拟电路,因此敌手$\mathcal{A}$不能区分(F,X,d)是由$\mathcal{S}$构造,还是由真实参与方构造,SGT-GC协议满足隐私性需求. 证毕.

定理3. SGT-GC协议满足不经意性需求. 即对于任意布尔电路f和真实输入x,参与方生成的混淆分布(F,X)与概率多项式时间模拟器$\mathcal{S}(1^\kappa,f)$生成的随机混淆结构(F,X)不可区分.

由于不经意性需求中,模拟器$\mathcal{S}$和真实参与方的输入与隐私性需求相同,区别在于隐私性需求的输出为(F,X,d),不经意性需求的输出中不包含解码信息d.因此本定理的证明仍可通过构造模拟器算法实现,构造方法与隐私性证明中所述算法13相同,仅在输出过程省去了构造解码信息d的操作.

5.2 可认证性证明

定理4. SGT-GC协议满足可认证性需求. 即在只给定(F,X)的情况下,概率多项式时间敌手$\mathcal{A}$无法生成$\tilde{Y}\neq Eval(F,X)$,使得$Decode(\tilde{Y},d)\notin\perp$.

证明. 混淆电路F中,每条输出线上有两种可能的输出混淆值,但敌手$\mathcal{A}$只能通过$Y\leftarrow Eval(F,X)$计算出其中的一个混淆值,并且这些混淆值可以解码出正确的输出$y\leftarrow Decode(Y,d)$.

由于非门不涉及对混淆值的运算,因此只考虑二元门的情况.当电路F的输出线w_i为与门g_i的输出线时,由求值算法$Eval(\cdot)$可知,敌手$\mathcal{A}$得到的混淆值W_i是其掌握的点P_i到共享信息C_i的距离.在$W_i=W_i^0$的情况下,敌手$\mathcal{A}$破解W_i^1的难度等同于在三维空间中,仅已知一条线段其中一个端点的情况下,求解线段长度.在$W_i=W_i^1$的情况下,敌手$\mathcal{A}$破解W_i^0的难度等同于在仅已知圆心的情况下,求解圆的半径.因此敌手在已知一个混淆值的情况下,破解另一个混淆值的概率为$2^{-\kappa}$,因此概率多项式时间敌手$\mathcal{A}$无法生成$\tilde{Y}\neq Eval(F,X)$,使得$Decode(\tilde{Y},d)\notin\perp$. SGT-GC协议满足可认证性需求. 证毕.

6 性能分析

混淆电路的性能指标包括通信性能和计算性能两方面,为了验证本文协议性能的优势,将其与当前最先进的混淆电路协议进行对比.

表4给出了不同协议的通信成本,其中$|x_G|$和$|x_E|$分别为电路中Generator和Evaluator的输入长度,单位均为bit.在电路相同的情况下,Evaluator获取输入混淆值所产生的通信量$|x_E||OT|+|x_G|\kappa$与其传输输出混淆值所产生的通信量$l\kappa$是每个协议都包含且完全相等的.导致不同协议通信量不同的原因在于,每个协议传输的混淆表大小不同.

表4 混淆电路协议通信性能比较

GC协议	通信量(位)	安全假设
Yao's[3]	$\|x_E\|\|OT\|+\|x_G\|\kappa+4p_1\kappa+4p_2\kappa+l\kappa$	PRF
Point-Permute[5]	$\|x_E\|\|OT\|+\|x_G\|\kappa+4p_1\kappa+4p_2\kappa+l\kappa$	PRF
4-3GRR[11]	$\|x_E\|\|OT\|+\|x_G\|\kappa+3p_1\kappa+3p_2\kappa+l\kappa$	PRF
4-2GRR[2]	$\|x_E\|\|OT\|+\|x_G\|\kappa+2p_1\kappa+2p_2\kappa+l\kappa$	PRF
GLNP[9]	$\|x_E\|\|OT\|+\|x_G\|\kappa+p_1\kappa+2p_2\kappa+l\kappa$	PRF
本文**SGT-GC**	$\|\boldsymbol{x_E}\|\|\boldsymbol{OT}\|+\|\boldsymbol{x_G}\|\kappa+\boldsymbol{p_1}\kappa+\boldsymbol{p_2}\kappa+l\kappa$	**PRF**
Free-XOR[6]	$\|x_E\|\|OT\|+\|x_G\|\kappa+3p_2\kappa+l\kappa$	CCR
FleXOR[7]	$\|x_E\|\|OT\|+\|x_G\|\kappa+(0\,or\,1\,or\,2)\,p_1\kappa+2p_2\kappa+l\kappa$	CCR
Half-Gates[8]	$\|x_E\|\|OT\|+\|x_G\|\kappa+2p_2\kappa+l\kappa$	CCR
Three Half-Gates[10]	$\|x_E\|\|OT\|+\|x_G\|\kappa+1.5p_2\kappa+2l\kappa$	CCR

本文的安全性基于标准的PRF假设，当前最先进的PRF假设下的混淆电路协议为GLNP[9]，本文协议中，处理异或门所需的通信量为$p_1\kappa$，与GLNP[9]相同，而处理与门所需的通信量为$p_2\kappa$，比GLNP[9]减少了50%. 基于CCR假设的Free-XOR[6]、FleXOR[7]、Half-Gates[8]和Three Half-Gates[10]协议均使用了Free-XOR的方法，通过放宽安全标准实现了无需传输混淆表的异或门. 但CCR假设下的与门构造仍需双方进行通信. 目前CCR假设下最优的GC协议是2021年提出的Three Half-Gates[10]协议，其中与门通信需1.5条信息，是SGT-GC与门通信量的1.5倍. 因此在通信性能方面，SGT-GC优于现存PRF假设下的GC协议，且与门的混淆表小于现存CCR假设下的GC协议.

表5　混淆电路协议计算性能比较

GC协议	计算量		安全假设
	Generator	Evaluator	
Yao's[3]	$\|Encode\|+4p_1H+4p_2H+\|Decode\|$	$2.5p_1H+2.5p_2H$	PRF
Point-Permute[5]	$\|Encode\|+4p_1H+4p_2H+\|Decode\|$	p_1H+p_2H	PRF
4-3GRR[11]	$\|Encode\|+4p_1H+4p_2H+\|Decode\|$	p_1H+p_2H	PRF
4-2GRR[2]	$\|Encode\|+4p_1H+4p_2H+\|Decode\|$	p_1H+p_2H	PRF
GLNP[9]	$\|Encode\|+3p_1H+4p_2H+\|Decode\|$	$1.5p_1H+2p_2H$	PRF
本文**SGT-GC**	$\|Encode\|+p_1\|SGTXOR\|+p_2\|SGTAND\|+\|Decode\|$	$p_1\|Dis\|+p_2\|Dis\|$	**PRF**
Free-XOR[6]	$\|Encode\|+4p_2H+\|Decode\|$	p_2H	CCR
FleXOR[7]	$\|Encode\|+(0\ or\ 2\ or\ 4)\,p_1H+4p_2H+\|Decode\|$	$(0\ or\ 1\ or\ 2)\,p_1H+p_2H$	CCR
Half-Gates[8]	$\|Encode\|+4p_2H+\|Decode\|$	$2p_2H$	CCR
Three Half-Gates[10]	$\leqslant\|Encode\|+6p_2H+\|Decode\|$	$\leqslant 3p_2H$	CCR

表5给出了不同协议的计算成本，其中$|Encode|$和$|Decode|$分别为编码和解码所需的计算量，在电路相同的情况下，不同协议的编码和解码计算量是相同的. p_1和p_2分别为电路中异或门和与门的数量，H为协议中调用一次加密算法所需计算量，不同的加密算法计算量不同，为方便描述，本文统一用H表示. $|SGTXOR|$和$|SGTAND|$分别为混淆异或门和与门所需的计算量，$|Dis|$为计算三维坐标系中两点间距离的计算量. 为了分析本文的计算性能，统计了当前最先进的PRF假设下的混淆电路协议4-2GRR[2]和GLNP[9]，在对异或门和与门进行混淆和求值操作的时间，其中加密算法为AES. 不同协议的编码和解码计算量是相同的，因此本文不考虑$|Encode|$和$|Decode|$，仅统计计算两种二元门时，不同协议的计算时间.

本文运行环境为VMware workstation下Ubuntu 20.04虚拟机，内存为4 GB，处理器数量为1，内核数量为2. 统计结果如表6所示，在Generator生成混淆电路的过程中，混淆异或门和与门的时间大于4-2GRR[2]和GLNP[9]. 但在Evaluator对混淆电路求值的过程中，异或门和与门的计算时间较现存方法分别提升了82%和86%. 因此，本文降低了GC中Evaluator的计算量. 在实际应用中，更适用于Generator计算性能较强而Evaluator计算性能较弱的环境中。

表6　混淆电路协议中每个门的计算时间　　毫秒

GC协议	Generator		Evaluator	
	异或门	与门	异或门	与门
4-2GRR[2]	0.2331	0.2331	0.1762	0.1762
GLNP[9]	**0.1656**	**0.2208**	0.0828	0.1104
本文**SGT-GC**	0.2022	0.4041	**0.0148**	**0.0148**

7　总　结

本文提出了一种标准PRF假设下基于立体几何变换的轻量级混淆电路协议SGT-GC，首次实现了不调用任何加密算法的GC协议，在Evaluator的求值过程中，每个二元门仅需计算一次欧氏距离，避免了多次调用加密算法所造成的计算成本. 根据每类二元门信号逻辑设计了专门的立体几何变换，并替代传统的加密算法实现了混淆门的构造，使混淆表中仅包含一条共享信息，降低了协议的通信成本. 安全性证明表明，在半诚实模型下，所提协议满足隐私性、不经意性和可认证性. 在未来工作中，将在本文协议的基础上，继续探索降低本文协议计算

性能的方法，并尝试从优化OT协议的角度对本文方案进行进一步优化.

参 考 文 献

[1] Shelat A., Shen C.-h. Two-Output Secure Computation with Malicious Adversaries// Advances in Cryptology--EUROCRYPT 2011. Berlin, Germany, 2011: 386-405

[2] Pinkas B., Schneider T., Smart N. P., Williams S. C. Secure Two-Party Computation Is Practical// Advances in Cryptology - ASIACRYPT 2009. Berlin, Germany, 2009: 250-267

[3] Yao A. C.-C. How to generate and exchange secrets// 27th Annual Symposium on Foundations of Computer Science. Toronto, Canada, 1986: 162-167

[4] Lindell Y., Pinkas B. A Proof of Security of Yao's Protocol for Two-Party Computation. Journal of Cryptology., 2009, 22(2): 161-188

[5] Beaver D., Micali S., Rogaway P. The Round Complexity of Secure Protocols// Proceedings of the 22nd Annual ACM Symposium on Theory of Computing. New York, USA, 1990: 503-513

[6] Kolesnikov V., Schneider T. Improved Garbled Circuit: Free-XOR Gates and Applications// Proceedings of the 35th International Colloquium on Automata, Languages and Programming, Part II. Berlin, Germany, 2008: 486-498

[7] Kolesnikov V., Mohassel P., Rosulek M. FleXOR: Flexible garbling for XOR gates that beats free-XOR//Annual Cryptology Conference. Berlin, Germany, 2014: 440-457

[8] Zahur S., Rosulek M., Evans D. Two halves make a whole// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Germany, 2015: 220-250

[9] Gueron S., Lindell Y., Nof A., Pinkas B. Fast Garbling of Circuits Under Standard Assumptions. J. Cryptol., 2018, 31(3): 798 - 844

[10] Rosulek M., Roy L. Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits// Advances in Cryptology--CRYPTO 2021. 2021: 94-124

[11] Naor M., Pinkas B., Sumner R. Privacy Preserving Auctions and Mechanism Design// Proceedings of the 1st ACM Conference on Electronic Commerce. New York, USA, 1999: 129-139

[12] Agrawal S. Stronger Security for Reusable Garbled Circuits, General Definitions and Attacks//Advances in Cryptology-CRYPTO 2017. Santa Barbara, USA, 2017: 3-35

[13] Goldwasser S., Kalai Y., Popa R. A., Vaikuntanathan V., Zeldovich N. Reusable garbled circuits and succinct functional encryption//Proceedings of the 45th Annual ACM Symposium on Theory of Computing. Palo Alto, USA, 2013: 555-564

[14] Qingsong Z., Ximeng L., Huanliang X., Yanbin L. Practical reusable garbled circuits with parallel updates. Computer Standards & Interfaces, 2023, 86(0920-5489): 103721

[15] Harth-Kitzerow C., Carle G., Fei F., Luckow A., Klepsch J. CRGC--A Practical Framework for Constructing Reusable Garbled Circuits. arXiv, 2022

[16] Lindell Y. Fast Cut-and-Choose Based Protocols for Malicious and Covert Adversaries. Journal of Cryptology, 2016, 29(2): 456-490

[17] Lindell Y., Pinkas B. Secure Two-Party Computation via Cut-and-Choose Oblivious Transfer. Journal of Cryptology, 2011, 25(4): 329-346

[18] Zhang Zong-Yang, Liu Xiang-Yu, Li Wei-Han, Chen Lao. Efficient and cooperative secure two-party computation based on authenticated garbled circuit. Chinese Journal of Computers, 2022, 45(11): 2433-2455(in Chinese)
(张宗洋，刘翔宇，李威翰，陈劳. 基于可验证混淆电路的合作式安全两方计算协议. 计算机学报，2022，45(11)：2433-2455)

[19] Ben-Efraim A., Cong K., Omri E., Orsini E., Smart N. P., Soria-Vazquez E. Large Scale, Actively Secure Computation from LPN and Free-XOR Garbled Circuits//Advances in Cryptology--EUROCRYPT 2021. Zagreb, Croatia, 2021: 33-63

[20] Wei Li-Fei, Liu Ji-Hai, Zhang Lei, Wang Qin, He Chong-De. Survey of privacy preserving oriented set intersection computation. Journal of Computer Research and Development, 2022, 59(08): 1782-1799(in Chinese)
(魏立斐，刘纪海，张蕾，王勤，贺崇德. 面向隐私保护的集合交集计算综述. 计算机研究与发展，2022，59(08)：1782-1799)

[21] Bellare M., Hoang V. T., Keelveedhi S., Rogaway P. Efficient Garbling from a Fixed-Key Blockcipher//2013 IEEE Symposium on Security and Privacy. Berkeley, USA, 2013: 478-492

[22] Guo C., Katz J., Wang X., Weng C., Yu Y. Better Concrete Security for Half-Gates Garbling (in the Multi-Instance Setting)//Advances in Cryptology-CRYPTO 2020. Santa Barbara, USA, 2020: 793-822

[23] Black J. The Ideal-Cipher Model, Revisited: An Uninstantiable Blockcipher-Based Hash Function//Fast Software Encryption. FSE 2006. Berlin, Germany, 2006: 328-340

[24] Biryukov A., Khovratovich D., Nikolić I. Distinguisher and Related-Key Attack on the Full AES-256// Advances in Cryptology-CRYPTO 2009. Berlin, Germany, 2009: 231-249

[25] Gueron S. Intel's New AES Instructions for Enhanced Performance and Security// Fast Software Encryption. Berlin, Germany, 2009: 51-66

[26] Shamir A. How to Share a Secret. Commun. ACM, 1979, 22(11): 612-613

[27] Choi S. G., Katz J., Kumaresan R., Zhou H.-S. On the Security of the "Free-XOR" Technique//Theory of Cryptography. Berlin, Germany, 2012: 39-53

[28] Bellare M., Hoang V. T., Rogaway P. Foundations of Garbled Circuits//Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York, USA, 2012: 784-796

[29] Banerjee A., Peikert C., Rosen A. Pseudorandom Functions and Lattices//Theory and Application of Cryptographic Techniques. Berlin, Germany, 2011: 719-737

[30] Ishai Y., Kilian J., Nissim K., Petrank E. Extending Oblivious Transfers Efficiently// Advances in Cryptology-CRYPTO 2003. Berlin, Germany, 2003: 145-161

[31] Kolesnikov V., Kumaresan R. Improved OT Extension for Transferring Short Secrets//Advances in Cryptology- CRYPTO 2013. Berlin, Germany, 2013: 54-70

[32] Song Jia-Shou, Li Zhen-Zhen, Ding Hai-Yang, Li Zi-Chen. Efficient and fully simulated oblivious transfer protocol on elliptic curve. Chinese Journal of Network and Information Security, 2023, 9(01): 158-166(in Chinese)

(宋佳烁，李祯祯，丁海洋，李子臣. 椭圆曲线上高效可完全模拟的不经意传输协议. 网络与信息安全学报，2023，9(01)：158-166)

[33] Yongjun W., Kun X., He T., Jing Z., Xixi Y. Secure Two-Party Computation Based on Fast Cut-and-Choose Bilateral Oblivious Transfer. Security and Communication Networks, 2022, 2022(1939-0114): 1-10

Tan Zhen-Hua, Ph. D., professor. His current research interests include, multi-party computation and privacy-preserving.

Ning Jing-Yu, Ph. D. candidate. Her current research interests include multi-party computation blocks and privacy-preserving.

Background

Garbled Circuit (GC) is a critical building block of constant-round secure two-party computation (S2PC) where a function can be represented in the form of a Boolean circuit with gates and wires. Since Boolean circuits reveal the input information of participants, Yao proposed the first and famous GC, which could protect the security and privacy during communication. The basic idea of GC is to represent the bit values of wires by garbled values and implement the gate functions by encryption algorithms. Specifically, a Generator randomly selects two garbled values, TRUE and FALSE, for each wire in the circuit. The GC encrypts each possible output of a gate with related garbled input options as keys. Each garbled binary gate has four ciphertexts since the two wires have 4 possible input options. During GC, a Generator provides a garbled truth table for each gate where each combination of input wire labels is used to encrypt the corresponding output wire label. Consequently, the Generator generates four ciphertexts one for each input combination to the gate, and send them to the Evaluator, who only knows one label for each input wire and can only open one of them by decryption and decoding.

After Yao's, how to optimize GC has attracted great attention in decades, from the perspectives of computation or communication. Some focus on reducing the number of calls to the encryption algorithms to improve computation performance. Some focus on reducing the number of ciphertexts of each gate to improve communication performance. Up to now, the communication performance has improved greatly. For a general AND gate, the number of ciphertexts can be reduced to 2 under standard assumption of pseudorandom function (PRF), and to 1.5 under circular correlation robust hash function (CCR) assumption. The computation performance of GC has also improved greatly in the Evaluator, but still needs at least 4 times calls to encryption algorithm in the Generator.

This paper proposes a lightweight garbled circuit based on solid geometry transformation under standard assumption, named as SGT-GC. Each gate in SGT-GC has only one shared information in the garbled table, and requires no calls to encryption, which avoids the computation cost caused by the complex encryption algorithms and the communication cost caused by the transmission of multiple ciphertexts.

This work was funded by the National Key Research and Development Program of China under Grant No. 2019YFB1405803, the Fundamental Research Funds for the Central Universities No. N2217001 and the National Natural Science Foundation of China No. 61772125.