Computer Security Symposium 2025
27-31 October 2025

Free-XORを含むガーブルド回路における入力ゲートの効率化

小野 知樹[1,a)]　渡邉 洋平[1,2]　岩本 貢[1]

概要：秘密計算の代表的な手法であるガーブルド回路において，通信量の削減は重要な課題である．XORゲートに関しては，通信を伴わずに実現する手法が得られており，この手法はfree-XORと呼ばれている．Rosulek–Royは，free-XORを利用する構成において，それ以外のゲートを1.5 λビット（λはセキュリティパラメータ）で実現可能であることを示した．また，Kempkaらは，入力ゲートに限ってはλビットで実現できる手法を提案している．しかし，Kempkaらの手法を用いて入力ゲートを構成した場合，回路内のXORゲートにfree-XORを適用できないという問題点があった．本研究では，回路中のXORゲートにもfree-XORを適用可能であり，かつ入力ゲートの通信量をλビットに抑えられる手法を提案する．

キーワード：秘密計算，ガーブルド回路

Efficient Input Gates in Garbled Circuits with Free-XOR

Tomoki Ono[1,a)]　Yohei Watanabe[1,2]　Mitsugu Iwamoto[1]

Abstract: Reducing communication cost is a critical challenge in garbled circuits, which are a representative technique for secure computation. For XOR gates, a method known as free-XOR enables evaluation without any communication. Rosulek and Roy showed that, in a garbling scheme utilizing free-XOR, non-XOR gates can be implemented using 1.5 λ bits of communication, where λ denotes the security parameter. Kempka et al. proposed a method that allows input gates to be realized using only λ bits of communication. However, when input gates are constructed using their method, free-XOR cannot be applied to XOR gates inside the circuit, which limits compatibility. In this work, we propose a new construction that allows free-XOR to be applied to XOR gates in the circuit while keeping the communication cost for input gates at λ bits.

Keywords: secure computation, garbled circuit

1. はじめに

1.1 背景

秘密計算はプレイヤの入力を隠したまま，その入力を用いた計算を行う技術である．秘密計算の代表的な実現手法の一つとしてガーブルド回路がある [12]．ガーブルド回路では通常 Generator と Evaluator の二人のプレイヤが秘密計算を行う状況を考える（今後 Gen, Eval と表す）．ガーブルド回路は，通信回数が少ないという点において優れているものの，通信量が大きいという問題があるため，高速化のためには通信量の削減が重要な課題である．Yao によって最初に提案されたガーブルド回路では，ゲート一つ当たり 4λ ビット（λ はセキュリティパラメータ）が必要であるが [12]，XOR ゲートに関しては Kolesnikov–Schneider によって，通信を伴わずに実現する，free-XOR と呼ばれる手法が提案されている [7]．さらに，Rosulek–Roy は，free-XOR を利用する構成において，それ以外のゲートを 1.5λ ビットで実現可能であることを示した．さらに Xu ら [11]，Baek–Kim [1]，Januzelli ら [5] は，XOR ゲートに

[1] 電気通信大学
The University of Electro-Communications
[2] 国立研究開発法人 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology
[a)] onotom@uec.ac.jp

free-XOR を利用し，回路中の任意の AND ゲートに適用可能である，という前提のもとで 1.5λ ビットが通信量の下界であることを示している．

さらに，[2], [6], [10] では，入力における AND ゲートを λ ビットで実現できる手法を提案している[*1]．しかし，これらの手法を用いて入力ゲートを構成した場合，回路内の XOR ゲートに free-XOR を適用できない，という問題点がある．この問題点が生じる理由は，これらの手法が $GF(2^\lambda)$ と異なる体の上で計算を行っているからである．

1.2 本研究の貢献とアイデア

本論文では，入力における AND ゲートの通信量を λ ビットに抑えながら，回路中の XOR ゲートに free-XOR を適用可能な新しいガーブルド回路構成法を提案する．

前節で述べた通り，先行研究では，$GF(2^\lambda)$ と異なる体の上で計算を行うことが free-XOR との共存への障壁となっていた．そこで本研究ではアプローチを変え，ゲートあたりの通信量が 2λ ビットで free-XOR と共存可能なハーフゲート [13] と呼ばれる手法を，入力における AND ゲートに適用して改良することで，この問題を回避する事に成功した．

先行研究の各方式と提案方式の比較を表 1 に示す．

1.3 本論文の構成

次節以降の構成は以下の通りである．2 節では記号の定理などの準備を行い，3 節では，提案方式の元となるプロトコルであるハーフゲートを紹介する．4 節では提案方式について述べ，5 節で free-XOR と共存可能でありながら 1.5λ の通信量下界を回避できた理由を述べる．最後に 6 節でまとめを行う．

2. 準備

2.1 記法

ハッシュ関数を $H : \{0,1\}^\lambda \to \{0,1\}^\lambda$ とする．二つのビット列 A, B に対して $A\|B$ で A と B の連接を表し，$A \oplus B$ はビットごとの排他的論理和を表す．$a \le b$ を満たす自然数 a, b に対して，$[a] \coloneqq \{1, 2, \ldots, a\}$, $[a : b] \coloneqq \{a, a+1, \ldots, b\}$ とする．ビット列 A の最下位ビットを取り出す関数を $\mathrm{lsb}(A)$ と書く．n ビット列の集合 $\{0,1\}^n$ に

$B \in \{0,1\}^a 1$ は $B' \in \{0,1\}^a$ のとき，$B \coloneqq B'\|1$ を表す．

本論文では，回路はループしないことを仮定し，論理回路 f を (n, m, q, L, R, g) で表す．各要素の定義は以下の通りである．$n = 2, m = 1, q = 3$ の例を図 1 に示す．

入力ビット数，出力ビット数，ゲート数をそれぞれ n, m, q (≥ 1) とする．このとき，論理回路のワイヤは全部で $n + q$ 本になるので，その集合を $W \coloneqq [1 : n + q]$ とする．ここで，回路の入力ワイヤの集合を $W_{input} \coloneqq [1 : n]$ とすると，ゲートの出力ワイヤの集合は $[n + 1 : n + q]$ になる．各ゲートは 2 入力 1 出力なので，ゲートとその出力ワイヤは一対一に対応する．そこで，各ゲートの番号をそのゲートの出力ワイヤの番号と同一視し，ゲートの番号の集合を $W_{gate} \coloneqq [n + 1 : n + q]$ と書く．論理回路 f は m 個の出力を持つので，論理回路 f の出力ワイヤ集合を $W_{output} = [n + q - m + 1 : n + q] \subseteq W_{gate}$ として一般性を失わない．

任意のゲート $i \in W_{gate}$ に対して，i を入力とし，ゲート i への二つの入力ワイヤ $\ell, r \in [1 : n + q]$ を返す関数をそれぞれ $L, R : W_{gate} \to W$ とする．すなわち，$L(i) = \ell$, $R(i) = r$ である．さらに，ゲート i が実現する論理演算を $g_i : \{0,1\}^2 \to \{0,1\}$ と書く．このとき，写像 $g : W_{gate} \times \{0,1\}^2 \to \{0,1\}$ を $g(i, \cdot, \cdot) \coloneqq g_i(\cdot, \cdot), i \in W_{gate}$ で定義する．

論理回路 f はループしないことを仮定しているので，任意の $i \in W_{gate}$ に対して $L(i) \le R(i) < i$ が成り立つとして一般性を失わない．図 1 はこの条件を満たしている．

例 1. 図 1 は，入力ビット数 $n = 2$，出力ビット数 $m = 1$，ゲート数 $q = 3$ の論理回路 $f = (2, 1, 3, L, R, g)$ の例である．ここで，入力ワイヤには 1, 2 と番号が付けられており，ゲートには 3, 4, 5 と番号が付けられている．このとき，ゲート 3 に対して $L(3) = 1, R(3) = 2$，ゲート 4 に対して $L(4) = 3, R(4) = 2$ である．ゲート 5 に対して $L(5) = 3, R(5) = 4$ である．例では，3, 4 のゲートが AND ゲートであり，5 のゲートが XOR ゲートであるため，$g_3 = (0, 0, 0, 1)$, $g_4 = (0, 0, 0, 1)$, $g_5 = (0, 1, 0, 1)$ となる．さらに，3, 4 のゲートは回路への入力をゲートへの入力として持つため，この二つのゲートは入力ゲートである．□

本論文では，簡単のため，回路 f のすべてのゲートは XOR もしくは AND ゲートで構成されると仮定するが，AND 以外のゲートでも同様に構成できる．XOR ゲートの集合を $\mathrm{XorGates}(f) \subseteq W_{gate}$ とする．回路の入力ワイヤが少なくとも一つ接続されたゲートを，入力ゲートと呼ぶ．提案方式では，AND を計算する入力ゲートに対する効率化を行う．AND を計算する入力ゲートを firstAND ゲートと呼び，その集合を $\mathrm{fAgate}(f)$ と書く．

ゲート $c \in W_{gate}$ へ到達するまでの全ての入力とゲートを cone と呼び，$\mathrm{cone}_c(f)$ と表す．例えば図 1 では $\mathrm{cone}_4(f) = \{1, 2, 3\}$ である．Cone 全体を A_f と書く．すなわち，$A_f \coloneqq \{\mathrm{cone}_c \mid c \in W_{gate}\}$ である．ゲート $c \in \mathrm{fAgate}(f)$ に対し，c の入力のうち，回路の入力ワイヤが一つ以上存在するので，そのうち一つを固定して $\mathrm{fAinw}_c^L(f)$ と書く．ここで，$\mathrm{fAinw}_c^L(f) = L(c)$ となるように L および R を定めておく．これらの入力の集合として

[*1] 正確には，[2] は，幾つかの条件を満たした回路内の AND ゲートに適用でき，入力ビット数より多い数の AND ゲートを λ ビットで実現できる．しかし，この手法も free-XOR と併用できない．

表 1　提案方式と先行研究の効率比較．λ はセキュリティパラメータ
Table 1　Comparison of efficiency between the proposed approach and prior works. λ denotes the security parameter

	入力 AND ゲートの通信量	適用可能なゲートの位置	free-XOR と組み合わせ
ハーフゲート [13]	2λ	任意	可能
Rosulek–Roy [9]	1.5λ	任意	可能
Kempka et al. [6]	λ	入力ゲート	不可能
Ball et al. [2] + GGR2 [8]	λ	一部 (入力ゲートを含む)	不可能
Wang–Malluhi [10]	λ	入力ゲート	不可能
提案方式	λ	入力ゲート	可能

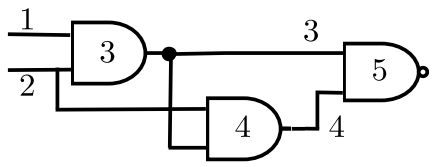

図 1　論理回路の例
Fig. 1　Example of Boolean circuit

$\mathrm{fAinw}^L(f) \coloneqq \{\mathrm{fAinw}_c^L(f) \mid c \in \mathrm{fAgate}(f)\}$ と定める．

2.2　ガーブリングスキーム

ガーブルド回路の定式化のために，Bellare ら [3] はブリングスキーム（garbling scheme）を導入した．以下に示すのは Rosulek–Roy によるガーブリングスキームの定義である．

定義 1 (ガーブリングスキーム)．セキュリティパラメータ λ および論理回路 f に対するガーブリングスキーム $(\mathsf{Gb}, \mathsf{En}, \mathsf{Ev}, \mathsf{De})$ は，以下の 4 つのアルゴリズムで定義される．Gb は確率的アルゴリズムであり，$\mathsf{En}, \mathsf{Ev}, \mathsf{De}$ は確定的アルゴリズムである．

- Gb: 入力 $(1^k, f)$ に対して，(F, e, d) を出力する．F をガーブルド回路，e をエンコード情報，d をデコード情報と呼ぶ．
- En: 入力 (e, x) に対して，X を出力する．X を入力ワイヤ x のラベルと呼ぶ．
- Ev: 入力 (F, X) に対して，Y を出力する．Y を出力ワイヤのラベルと呼ぶ．
- De: 入力 (d, Y) に対して，$y \in \{0,1\}^m$ または $\perp$ を出力する．□

ガーブリングスキームは次の正当性を満たす必要がある．

定義 2 (正当性)．任意の論理回路 f と任意の入力 $x \in \{0,1\}$ に対する出力 $(F, e, d) \leftarrow \mathsf{Gb}(1^\lambda, f)$ に対して，

$$\mathsf{De}\big(d, \mathsf{Ev}\big(F, \mathsf{En}(e, x)\big)\big) = f(x)$$

が無視できる確率を除いて成り立つ．□

ガーブルド回路の安全性を以下で定義する：

定義 3 (安全性)．

Privacy: $(1^\lambda, f, f(x))$ を入力とし，定義 1 における (F, X, d) と識別不可能な出力を生成する，シミュレータ $\mathrm{prv.sim}_S$ が存在する．

Obliviousness: $(1^\lambda, f)$ を入力とし，定義 1 における (F, X) と識別不可能な出力を生成する，シミュレータ $\mathrm{obv.sim}_S$ が存在する．

Authenticity: 入力 (F, d, X) が与えられた場合に，任意の攻撃者が $Y' \neq \mathsf{Ev}(F, X)$ かつ $\mathsf{De}(d, Y') \in \{f(x), \perp\}$ となる Y' を生成する確率が無視できる．□

ガーブリングスキームのセキュリティは Privacy を基本とし，Privacy に加えて，Obliviousness と Authenticity を満たしていれば，プライベートかつ検証可能なアウトソース計算に応用可能である [3]．

3.　ハーフゲート

3.1　構成法

提案手法の説明のためにガーブルド回路の既存技術であるハーフゲート [13] を紹介する．ハーフゲートの概要を図 2 に示し，手順をアルゴリズム 3 に示す．ハーフゲートでは，AND ゲートを生成者用ハーフゲートと評価者用ハーフゲートに分け，それの出力を XOR ゲートに入力する．以下で示す通り，生成者用ハーフゲートと評価者用ハーフゲートはそれぞれ λ ビットの通信を必要とすること，XOR ゲートを free-XOR で実現すれば通信を必要としないことから，ハーフゲートは合計で 2λ ビットの通信で実現できる．

以下では，$v_r, v_\ell \in \{0,1\}$ を AND ゲートの入力，$v_c \in \{0,1\}$ を出力とした場合のハーフゲート構成のアイデアを説明する．まず，$v_c = v_r \wedge v_\ell$ を任意のビット $v_a \in \{0,1\}$ を用いて次のように変形する．

$$\begin{aligned} v_c &= v_r \wedge v_\ell \\ &= v_r \wedge (v_a \oplus v_a \oplus v_\ell) \\ &= (v_r \wedge v_a) \oplus (v_r \wedge (v_\ell \oplus v_a)) \end{aligned} \tag{1}$$

ハーフゲートで式 (1) を計算するとき，変数 v_a を Gen がランダムに生成し，$v_r \wedge v_a$ と $v_r \wedge (v_\ell \oplus v_a)$ を生成者用ハーフゲートと評価者用ハーフゲートでそれぞれ計算する．以

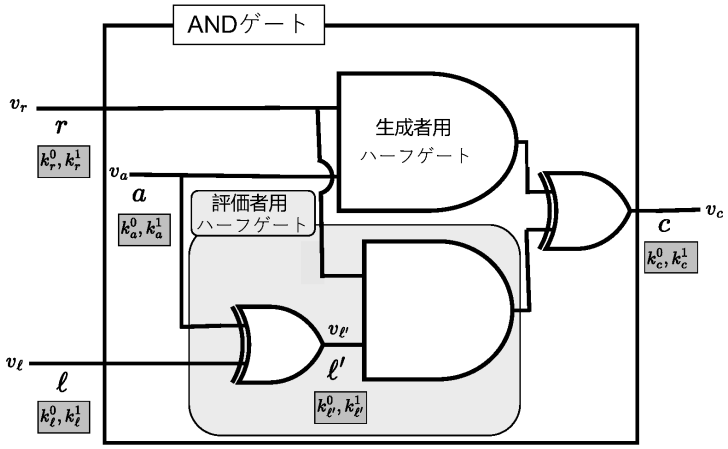

図 2 ハーフゲート
Fig. 2 Half gate

下では，それぞれの計算手法について述べる．

各ワイヤ $i \in W$ にワイヤの値 $0,1$ に対応する乱数 $k_i^0, k_i^1 \in \{0,1\}^\lambda$ を割り当て，その乱数をワイヤラベルと呼ぶ．図 2 においては，ゲートの入力ワイヤと出力ワイヤは $r, \ell \in W$ なので，これらに対応する値はそれぞれ $v_r, v_\ell \in \{0,1\}$ であり，対応するラベルはそれぞれ $k_r^{v_r}, k_\ell^{v_\ell} \in \{0,1\}^\lambda$ となる．

Free XOR を使用するためには，全てのラベルが以下の式を満たす必要がある．Δ を共通差分と呼ぶ．

$$\exists \Delta \in \{0,1\}^{\lambda-1}1, \forall i \in W,\ k_i^0 \oplus k_i^1 = \Delta \tag{2}$$

生成者用ハーフゲート： Gen が変数 v_a の値を選び，変数 $k_c^0 \in \{0,1\}^\lambda$ と $\Delta \in \{0,1\}^{\lambda-1}1$ を適切に選ぶことで，$r \wedge a$ を表す二つの暗号文

$$H(k_r^0, j) \oplus k_c^0 \tag{3}$$
$$H(k_r^1, j) \oplus k_c^0 \oplus v_a\Delta \tag{4}$$

を Eval に送る．ここで j は通常，ゲートの番号としておけば十分であるが，一つの AND ゲート内であってもハーフゲートの方式では生成者用と評価者用のハーフゲートに対して，異なる値を用いることに注意する．このとき，$v_a = v_r = 1$ ならば Eval は二つの暗号文から $k_c^1 = k_c^0 \oplus \Delta$ を，それ以外の場合，Eval は k_c^0 を得ることが確認できる．さらに，式 (2) が成り立つことも確認できる．

ここで k_c^0 は任意であるため，$k_c^0 = H(k_r^0, j)$ もしくは $k_c^0 = H(k_r^1, j) \oplus v_a\Delta$ とすることで，k_r^0, j の値にかかわらず，どちらか片方の暗号文を 0^λ とすることができる．したがって，非ゼロの暗号文一つのみを送信することで $r \wedge a$ のラベルを計算することができる．

評価者用ハーフゲート： Gen は v_r, v_ℓ の値を知らないまま，暗号文を生成する．そのため，通常の方法でガーブルド回路を作成すると 4λ ビットの暗号文が必要であり，2λ ビットの暗号文で計算を行うためには工夫が必要である．Gen は v_a の値を知っているため，これを利用して，v_ℓ の値を隠しつつ，v_ℓ の値によって暗号化の方法を変更する．そのため，$v_{\ell'} = v_\ell \oplus v_a$ と定義して，

Gen は暗号文

$$H(k_{\ell'}^0, j') \oplus k_c^0 \tag{5}$$
$$H(k_{\ell'}^1, j') \oplus k_c^0 \oplus k_r^0 \tag{6}$$

を Eval に送る．

$v_{\ell'} = 1$ に対して，Gen が式 (6) の暗号文を作成して，$v_{\ell'} = 0$ に対して，Gen が式 (5) の暗号文を作成する．

Eval が $v_{\ell'} := v_\ell \oplus v_a$ と $k_{\ell'}^{v_{\ell'}}$ の値を何らかの方法で得ていると仮定すると，Eval が $v_{\ell'}$ によって異なる方法を用いてラベルを得られる．具体的な手順は，$v_{\ell'} = 1$ の場合，Eval が式 (6) の暗号文から $k_c^0 \oplus k_r^0$ を得て，入力 r 側のラベルである $k_r^0 \oplus v_r\Delta$ との排他的論理和をとり，$k_c^0 \oplus v_r\Delta$ を得る．そうでない場合は，Eval が式 (5) の暗号文から k_c^0 を得る．このとき，$v'_\ell = v_r = 1$ ならば Eval が $k_c^1 = k_c^0 \oplus \Delta$ を得て，それ以外の場合，Eval が k_c^0 を得ることが確認できる．さらに，式 (2) が成り立つことも確認できる．

ここで，k_c^0 は任意であるため，$k_c^0 = H(k_{\ell'}^0, j')$ とすることで，$k_{\ell'}^0$, j' の値にかかわらず，式 (5) の暗号文 $H(k_{\ell'}^0, j') \oplus k_c^0$ を 0^λ とすることが出来る．したがって，非ゼロの暗号文一つのみを送信することで $r \wedge \ell'$ のラベルを計算することができる．

ここで，$v_{\ell'}$ の値を Eval が得ていると仮定しているため，どちらの暗号文を送信したかを Eval に隠す必要がないため，Gen は常に Eval に式 (6) の暗号文 $H(k_{\ell'}^0 \oplus \Delta, j') \oplus k_c^0 \oplus k_r^0$ を送信しても良い．

評価者がワイヤの値を得ないことを表すため，ワイヤの値を消して，ワイヤラベルを $k_r, k_\ell \in \{0,1\}^\lambda$ と書く．Gen

が $v_{\ell'}$ を得ないまま，通信量を増やさずに Eval が $v_{\ell'}$ 及び $k_{\ell'}$ を得る方法について述べる．$v_a = \mathrm{lsb}(k_\ell^0)$ として Gen が v_a を決める．このとき，$k_\ell^0 \in \{0,1\}^\lambda$ はランダムに選ばれているので，このように v_a を決めても問題ない．

- $v_a = 0$ のとき，Gen は $k_{\ell'}^{v_{\ell'}} = k_\ell^{v_\ell}$ としてガーブルド回路を作成する．
- $v_a = 1$ のとき，Gen は $k_{\ell'}^{v_{\ell'}} = k_\ell^{(v_\ell \oplus 1)}$ としてガーブルド回路を作成する．

Eval は得たラベル $k_\ell^{v_\ell} \in \{0,1\}^\lambda$ を ℓ' のラベル $k_{\ell'} \in \{0,1\}^\lambda$ とする．このようにすることで，$v_{\ell'} = \mathrm{lsb}(k_{\ell'}^{v_{\ell'}})$ となるため，Eval は $\mathrm{lsb}(k_{\ell'})$ を計算して，$v_{\ell'}$ を得る．

$v_{\ell'}$ の値を Eval が得ると仮定しても良い理由は，v_a の値は Gen がランダムに選択しているため，Eval が $v_{\ell'}$ から v_ℓ を導出できないためである．

3.2 Circular Correlation Robust Hashes

提案方式では free-XOR を用いる．プロトコル内で free-XOR を用いるには，プロトコル内で用いるハッシュ関数 H が CCR (Circular Correlation Robustness) という性質を持てば十分である．CCR の定義は Choi ら [4] で導入されたが，以下では，[13] の記法を用いる．

定義 4. H をハッシュ関数とし，二つのオラクル

$$\mathsf{Circ}\Delta(x,i,b) \coloneqq H(x \oplus \Delta, i) \oplus b\Delta,$$
$$\Delta \in \{0,1\}^{\lambda-1}1, b \in \{0,1\}$$
$$\mathsf{Rand}(x,i,b) : \lambda\text{-bit 出力を持つランダム関数}$$

に対して，異なる $b, b' \in \{0,1\}$ に対して同じ (x,i) をクエリしてはいけない，という制約を設ける．このとき，任意の多項式時間攻撃者 A に対して、

$$\left|\Pr\left[A^{\mathsf{Circ}\Delta}(1^\lambda) = 1\right] - \Pr\left[A^{\mathsf{Rand}}(1^\lambda) = 1\right]\right|$$

が無視できるほど小さいとき，H は *circular correlation robust* であるという．

4. 提案方式

4.1 アイデア

提案方式は，効率的なガーブルド回路の構成法であるハーフゲートをもとにしている．ハーフゲート中の評価者用ハーフゲートでは $r \wedge (\ell \oplus a)$ を計算するために次の二つの暗号文

$$H(k_{\ell'}^0, j') \oplus k_c^0 \tag{7}$$
$$H(k_{\ell'}^1, j') \oplus k_c^0 \oplus k_r^0 \tag{8}$$

のうち，式 (7) の暗号文を送信せずに式 (8) の暗号文だけを Gen から Eval に送信して計算を行う [13]．具体的には，ハーフゲート中の評価者用ハーフゲートでは k_c^0 が任意であることを用いて，$k_c^0 = H(k_{\ell'}^0, j')$ とすることで，式 (7) の暗号文を 0^λ として，式 (7) の暗号文を Gen から Eval に送信せずに計算を行うことができる．

提案方式では，式 (7) の暗号文だけでなく，式 (8) の暗号文も送信せずに評価者用ハーフゲートの計算を行う．具体的には，k_r^0 が任意であることを用いて，$k_r^0 = H(k_{\ell'}^1, j') \oplus k_c^0$ とすることで AND ゲートの入力ラベルに相関を持たせれば，式 (8) の暗号文も 0^λ とすることができる．この処理によって，式 (8) の暗号文も Gen から Eval に送信せずに評価者用ハーフゲートの計算を行うことができ，AND ゲートに必要な通信量を合計 λ ビットに削減することができる．このような暗号文に相関をもたせる処理は入力ゲートにのみ適用できることに注意する．[*2]

4.2 手順

4.1 節のようにラベルの値を設定したと仮定して，評価者用ハーフゲートの計算手順を述べる．$v'_\ell = 0$ の場合の手順はハーフゲートと同じである．ハーフゲートでは $v'_\ell = 1$ の場合 Eval は Gen から送られてきた暗号文を用いて次の計算を行う．

$$\begin{aligned}&(H(k_{\ell'}^1, j') \oplus k_c^0 \oplus k_r^0) \oplus H(k_{\ell'}^1, j') \oplus (k_r^0 \oplus v_r\Delta)\\ &= k_c^0 \oplus v_r\Delta\end{aligned} \tag{9}$$

すなわち，Gen から送られた暗号文 $H(k_{\ell'}^1, j') \oplus k_c^0 \oplus k_r^0$ と ℓ のラベルのハッシュを取ったもの $H(k_{\ell'}^1, j')$ と r のラベル $k_r^0 \oplus r\Delta$ の排他的論理和を Eval がとることで計算を行う．一方で，提案方式では $\ell' = 1$ の場合，Eval が ℓ のラベルのハッシュを取ったもの $H(k_{\ell'}^1, j')$ と r のラベル $k_r^0 \oplus v_r\Delta$ の排他的論理和をとることで計算を行う．具体的には，

$$\begin{aligned}&H(k_{\ell'}^1, j') \oplus (k_r^0 \oplus v_r\Delta)\\ =&H(k_{\ell'}^1, j') \oplus (H(k_{\ell'}^1, j') \oplus k_c^0 \oplus v_r\Delta)\\ =&k_c^0 \oplus v_r\Delta\end{aligned} \tag{10}$$

である．このように $k_r^0 = H(k_{\ell'}^1, j') \oplus k_c^0$ とすることで，暗号文を送信せずにハーフゲート中の評価者用ハーフゲートと同じ出力を得ることができる．ハーフゲートでは式 (2) が成り立つため，提案方式においてもこの式が成り立つ．

4.3 提案手法の構成方針

ガーブルド回路で free-XOR を適用するためには，全てのラベルが式 (2) を満たす必要がある．すなわち，AND ゲートの入力ワイヤを r, ℓ とし，出力ワイヤを c とすると，

[*2] 提案方式が入力ゲートにのみ適用可能な理由は，AND ゲートの入力ラベルに相関を持たせるためには AND ゲートの片側の入力のラベルを自由に決定する必要があるからである．さらに提案方式を適用する回路の入力は枝分かれしないものに限り，枝分かれしている場合は入力一つにつき AND ゲート一つを選んで提案方式を適用する．回路の入力が枝分かれしている場合，その入力ワイヤを入力にとるゲートのうち，ゲート番号が最も小さいものを選んで提案方式を適用するとして一般性を失わない．

入力ラベルに対して $k_r^0 \oplus k_r^1 = \Delta$, $k_l^0 \oplus k_l^1 = \Delta$ であって，出力ラベルに対しても

$$k_c^0 \oplus k_r^1 = \Delta \tag{11}$$

でなければならない．Eval が v_c の値を得ずに，式 (11) を満たす $k_c^{v_c}$ を得るプロトコルを構成することが目標となる．.

これを実現するために，Eval が v_c を得ずに，出力に対応するラベルとして，$\mathrm{lsb}(k_\ell^0) = 0$ のとき，$k_c^{v_c} = H(k_r^0) \oplus H(k_\ell^0) \oplus v_c\Delta$ を $\mathrm{lsb}(k_\ell^0) = 1$ のとき，$a = 0$ ならば $k_c^{v_c} = H(k_r^0) \oplus H(k_\ell^1) \oplus v_c\Delta$，$a = 1$ ならば，$k_c^{v_c} = H(k_r^0) \oplus H(k_\ell^1) \oplus (1 \oplus v_c)\Delta$ を Eval が得るように Gen がガーブルド回路を構成する．

4.4 提案方式のガーブリングスキーム

提案方式のガーブリングスキームはアルゴリズム 1–6 である．このうち提案方式のアルゴリズムは主に 4 であり，アルゴリズム 1, 2, 6 は提案方式を適用するため，ハーフゲートのアルゴリズム [13] を少し変更している．アルゴリズム 3, 5 は [13] と同一である．

Rosulek–Roy の方式との併用: アルゴリズム 1–6 中のハーフゲートを Rosulek–Roy の方式に置き換え可能である．Rosulek–Roy 方式と提案方式の併用では，Rosulek–Roy の方式の入出力の形式が提案方式の入出力の形式と異なるため，変換が必要である．提案方式の出力ラベルを Rosulek–Roy の方式の入力ラベルに変換するには次のようにする．ビット列の前半 $2^{\lambda/2}$ ビットを取り出す関数を **Front**()，後半 $2^{\lambda/2}$ ビットを取り出す関数を **Back**()，ビット列の並びを逆順にする関数を inv() と定義する．この場合，Rosulek–Roy の方式の形式に合わせて共通差分 Δ を変換すると

$$\Delta_{RR} = \begin{bmatrix} \mathbf{Front}(\mathbf{inv}(\Delta)) \\ \mathbf{Back}(\mathbf{inv}(\Delta)) \end{bmatrix} \in \begin{bmatrix} GF(2^{\lambda/2}) \\ GF(2^{\lambda/2}) \end{bmatrix} \tag{12}$$

であり，ワイヤ i に対するラベルを Rosulek–Roy の方式に合わせて変換すると，

$$\begin{bmatrix} \mathbf{Front}(\mathbf{inv}(k_i^{v_i})) \\ \mathbf{Back}(\mathbf{inv}(k_i^{v_i})) \end{bmatrix} \in \begin{bmatrix} GF(2^{\lambda/2}) \\ GF(2^{\lambda/2}) \end{bmatrix} \tag{13}$$

である．point-and-permute 用の乱数は $\pi_i = \mathrm{lsb}(k_i^{v_i})$ である．Rosulek–Roy の方式の出力の形式を提案方式の入力の形式に変換するには式 (12), (13) の逆変換を行えば良い．

正当性: ハーフゲートが正当性を満たし，提案方式がそれと同じ出力をすることから確認できる．本稿では紙面の都合上省略する．

安全性: 直感的にはハーフゲートが安全であることと，ハッシュを取った値はランダムであることから成り立つ．安全性の証明は本稿では紙面の都合上省略する[*3]．

[*3] Authenticity を満たすためにはアルゴリズムを少し書き換える必要がある（アルゴリズムに合わせて安全性のシュミレータも書き換える）がこれも紙面の都合上省略する．

5. ガーブルド回路の通信量の下界について

著者らが知る限り，ガーブルド回路の通信量の下界について考察した先行研究として [1], [5], [11], [13] がある．本節ではこれらの論文の通信量の下界を提案方式が回避できた理由を述べる．これらの論文ではセキュリティパラメータを λ として，それぞれ 2λ, 1.5λ, 1.5λ, 1.5λ ビットを通信量の下界としている．提案方式が下界を回避できるのはこれらの論文のモデルから除外されている手法を用いているからである．

ハーフゲートの提案論文 [13] における下界 2λ ビットや Xu らの論文 [11] と Beak らの論文 [1] における下界 1.5λ ビットを回避できている理由は，設定におけるラベルの作り方にある．ハーフゲートを提案した論文ではハッシュ関数へクエリを設定する前にラベルを全て決定している．これに対して，提案方式ではハッシュ関数へクエリを入力し，出力を得たあとに入力のラベルを決定している．このように対象とするモデルが異なるため 2λ または 1.5λ ビットの下界を回避できる．

Januzelli らの論文 [5] における下界 1.5λ ビットを回避できている理由は，彼らが入力ラベルを自由に選べない前提をおいているからである．これに対して，我々の提案方式では入力ゲートでは入力ラベルを自由に決定できることを前提としている．このように対象とするモデルが異なるため 1.5λ ビットの下界を回避できる．

6. おわりに

ガーブルド回路の既存技術であるハーフゲート [13] をもとに Gen が AND ゲートの入力ラベル（厳密には片側の入力ラベルともう片方の入力ラベルのハッシュ）にある相関を持たせることで，評価者用ハーフゲートの計算に必要な通信をなくす．これによって，評価者用ハーフゲートを通信なしで計算可能にし，回路中の XOR ゲートに free-XOR [7] を適用可能なまま，入力ゲートである AND ゲートを 1 ゲートあたり λ ビットで計算できるようにした．結果として，free-XOR や Rosulek–Roy の方式と組み合わせることで，ガーブルド回路の通信量を削減できる．

謝辞 本研究は JSPS 科研費 JP25KJ1294, JP23H00468, JP23H00479, JP23K17455, JP23K21644, JP23K24846 の助成，および JST CREST JPMJCR23M2 の支援を受けたものです．

参考文献

[1] Baek, C. and Kim, T.: Can we beat three halves lower bound? (Im)possibility of reducing communication cost for garbled circuits, *Des. Codes Cryptogr.*, Vol. 93, No. 6, pp. 2073–2105 (2025).

Algorithm 1 Gb – Garbling the Circuit

procedure $\textsc{Gb}(1^\lambda, f)$
  $F_0 \leftarrow f$
  $\Delta \leftarrow \{0,1\}^{(\lambda-1)}1$
  $(A_f, \mathrm{fAgate}(f)), \mathrm{fAinw}^L(f) \leftarrow \mathrm{firstANDcones}(f)$
  for $\forall c \in \mathrm{fAgate}(f)$ in topological order **do**
    for $i \in W_{\mathrm{input}} \wedge \forall c' < c, i \notin \mathrm{cone}_{c'}(f) \wedge i \in \mathrm{cone}_c(f) \wedge i \notin \mathrm{fAinw}_c^L(f)$ **do**
      $k_i^0 \leftarrow \{0,1\}^\lambda$
      $k_i^1 \leftarrow k_i^0 \oplus \Delta$
    end for
    for $i \in W_{\mathrm{gate}} \wedge \forall c' < c, i \notin \mathrm{cone}_{c'}(f) \wedge i \in \mathrm{cone}_c(f) \wedge L(i) \notin \mathrm{fAinw}^L(f)$ **do**)
      $r \leftarrow R(i)$
      $\ell \leftarrow L(i)$
      if $i \in \mathrm{XorGates}(f)$ **then**
        $k_i^0 \leftarrow k_r^0 \oplus k_\ell^0$
        $k_i^1 \leftarrow k_i^0 \oplus \Delta$
      else
        $(k_i^0, C_{G_i}, C_{E_i}) \leftarrow \mathrm{GbAnd}(k_r^0, k_\ell^0, k_r^1, k_\ell^1)$
        $F_i \leftarrow (C_{G_i}, C_{E_i})$
        $k_i^1 \leftarrow k_i^0 \oplus \Delta$
      end if
    end for
    $L(\mathrm{fAgate}_c(f)) = \ell, R(\mathrm{fAgate}_c(f) = r,\ k_\ell^0 = H(k_r^0) \oplus H(k_r^1)$
    $(k_i^0, C_{g_i}) \leftarrow \mathrm{GbfirstAnd}(k_r^0, k_\ell^0, k_r^1, k_\ell^1)$
    $F_i \leftarrow (C_{g_i})$
  end for
  for $i \notin A_f \wedge i \in W_{\mathrm{input}}$ **do**
    $k_i^0 \leftarrow \{0,1\}^\lambda$
    $k_i^1 \leftarrow k_i^0 \oplus \Delta$
  end for
  for $i \notin A_f \wedge i \in W_{\mathrm{gate}}$ **do**
    $r \leftarrow R(i)$
    $\ell \leftarrow L(i)$
    if $i \in \mathrm{XorGates}(f)$ **then**
      $k_i^0 \leftarrow k_r^0 \oplus k_\ell^0$
      $k_i^1 \leftarrow k_i^0 \oplus \Delta$
    else
      $(k_i^0, C_{G_i}, C_{E_i}) \leftarrow \mathrm{GbAnd}(k_r^0, k_\ell^0, k_r^1, k_\ell^1)$
      $F_i \leftarrow (C_{G_i}, C_{E_i})$
      $k_i^1 \leftarrow k_i^0 \oplus \Delta$
    end if
  end for
  for $i \in W_{\mathrm{output}}$ **do**
    $d_i \leftarrow \mathrm{lsb}(k_i^0)$
  end for
  for $i \in W_{\mathrm{input}}$ **do**
    $e_i \leftarrow k_i^0$
  end for
  return $(\hat{F}, \hat{e}, \hat{d})$
end procedure

Algorithm 2 firstANDcones

procedure $\textsc{firstANDcones}(f)$
  return $A_f, \mathrm{fAgate}(f), \mathrm{fAinw}^L(f)$
end procedure

Algorithm 3 GbAnd – Half-Gate Garbling for AND

procedure $\textsc{GbAnd}(k_r^0, k_\ell^0, k_r^1, k_\ell^1)$
  $p_\ell^0 \leftarrow \mathrm{lsb}(k_\ell^0)$; $p_r^0 \leftarrow \mathrm{lsb}(k_r^0)$
  $j \leftarrow \mathrm{NextIndex}()$; $j' \leftarrow \mathrm{NextIndex}()$
  $C_G \leftarrow H(k_\ell^0, j) \oplus H(k_\ell^1, j) \oplus p_r^0 \Delta$
  $k_{\mathrm{ga}}^0 \leftarrow H(k_\ell^0, j) \oplus p_\ell^0 C_G$
  $C_E \leftarrow H(k_r^0, j') \oplus H(k_r^1, j') \oplus k_\ell^0$
  $k_{\mathrm{ev}}^0 \leftarrow H(k_r^0, j') \oplus p_r^0(C_E \oplus k_\ell^0)$
  $k_i^0 \leftarrow k_{\mathrm{ga}}^0 \oplus k_{\mathrm{ev}}^0$
  return (k_i^0, C_G, C_E)
end procedure

Algorithm 4 GbfirstAnd –One ciphertext garbling for AND

procedure $\textsc{GbfirstAnd}(k_r^0, k_\ell^0, k_r^1, k_\ell^1)$
  $p_\ell^0 \leftarrow \mathrm{lsb}(k_\ell^0)$; $p_r^0 \leftarrow \mathrm{lsb}(k_r^0)$
  $j \leftarrow \mathrm{NextIndex}()$; $j' \leftarrow \mathrm{NextIndex}()$
  $C_g \leftarrow H(k_\ell^0, j) \oplus H(k_\ell^1, j) \oplus p_r^0 \Delta$
  $k_{\mathrm{ga}}^0 \leftarrow H(k_\ell^0, j) \oplus p_\ell^0 C_G$
  $k_{\mathrm{ev}}^0 \leftarrow H(k_r^0, j') \oplus p_r^0 k_\ell^0$
  $k_i^0 \leftarrow k_{\mathrm{ga}}^0 \oplus k_{\mathrm{ev}}^0$
  return (k_i^0, C_g)
end procedure

Algorithm 5 Encode and Decode

procedure $\textsc{En}(\hat{e}, \hat{x})$
  for each $e_i \in \hat{e}$ **do**
    $X_i \leftarrow e_i \oplus x_i \Delta$
  end for
  return $\hat{X}$
end procedure
procedure $\textsc{De}(\hat{d}, \hat{Y})$
  for each $d_i \in \hat{d}$ **do**
    $y_i \leftarrow d_i \oplus \mathrm{lsb}(Y_i)$
  end for
  return $\hat{y}$
end procedure

Algorithm 6 Ev – Evaluation

procedure $\text{Ev}(\hat{F}, \hat{X})$
  for $i \in W_{\text{Input}}$ **do**
    $k_i \leftarrow X_i$
  end for
  for $i \notin W_{\text{input}}$ in topological order **do**
    $r \leftarrow R(i)$
    $\ell \leftarrow L(i)$
    if $i \in \text{XorGates}(F_0)$ **then**
      $k_i \leftarrow k_\ell \oplus k_r$
    else if $i \notin \text{fAgate}(F_0)$ **then**
      $p_\ell \leftarrow \text{lsb}(k_\ell)$; $p_r \leftarrow \text{lsb}(k_r)$
      $j \leftarrow \text{NextIndex}()$; $j' \leftarrow \text{NextIndex}()$
      $(C_{G_i}, C_{E_i}) \leftarrow F_i$
      $k_{i_{ga}} \leftarrow H(k_\ell, j) \oplus p_\ell C_{g_i}$
      $k_{i_{ev}} \leftarrow H(k_r, j') \oplus p_r(C_{E_i} \oplus k_\ell)$
      $k_i \leftarrow k_{i_{ga}} \oplus k_{i_{ev}}$
    else
      $p_\ell \leftarrow \text{lsb}(k_\ell)$; $p_r \leftarrow \text{lsb}(k_r)$
      $j \leftarrow \text{NextIndex}()$; $j' \leftarrow \text{NextIndex}()$
      $(C_{g_i}) \leftarrow F_i$
      $k_{i_{ga}} \leftarrow H(k_\ell, j) \oplus p_\ell C_{g_i}$
      $k_{i_{ev}} \leftarrow H(k_r, j') \oplus p_r k_\ell$
      $k_i \leftarrow k_{i_{ga}} \oplus k_{i_{ev}}$
    end if
  end for
  for $i \in \text{Outputs}(F_0)$ **do**
    $Y_i \leftarrow k_i$
  end for
  return Y
end procedure

[2] Ball, M., Malkin, T. and Rosulek, M.: Garbling Gadgets for Boolean and Arithmetic Circuits, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pp. 565–577 (2016).

[3] Bellare, M., Hoang, V. T. and Rogaway, P.: Foundations of garbled circuits, *the ACM Conference on Computer and Communications Security, CCS'12, Raleigh, NC, USA, October 16-18, 2012* (Yu, T., Danezis, G. and Gligor, V. D., eds.), ACM, pp. 784–796 (2012).

[4] Choi, S. G., Katz, J., Kumaresan, R. and Zhou, H.: On the Security of the "Free-XOR" Technique, *Theory of Cryptography - 9th Theory of Cryptography Conference, TCC 2012, Taormina, Sicily, Italy, March 19-21, 2012. Proceedings* (Cramer, R., ed.), Lecture Notes in Computer Science, Vol. 7194, pp. 39–53 (2012).

[5] Januzelli, J., Rosulek, M. and Roy, L.: Lower Bounds for Garbled Circuits from Shannon-Type Information Inequalities, *IACR Cryptol. ePrint Arch.*, p. 876 (2025).

[6] Kempka, C., Kikuchi, R. and Suzuki, K.: How to Circumvent the Two-Ciphertext Lower Bound for Linear Garbling Schemes, *Advances in Cryptology - ASIACRYPT 2016 - 22nd International Conference on the Theory and Application of Cryptology and Information Security, Hanoi, Vietnam, December 4-8, 2016, Proceedings, Part II*, pp. 967–997 (2016).

[7] Kolesnikov, V. and Schneider, T.: Improved Garbled Circuit: Free XOR Gates and Applications, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, pp. 486–498 (2008).

[8] Pinkas, B., Schneider, T., Smart, N. P. and Williams, S. C.: Secure Two-Party Computation Is Practical, *Advances in Cryptology - ASIACRYPT 2009, 15th International Conference on the Theory and Application of Cryptology and Information Security, Tokyo, Japan, December 6-10, 2009. Proceedings*, pp. 250–267 (2009).

[9] Rosulek, M. and Roy, L.: Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits, *Advances in Cryptology - CRYPTO 2021 - 41st Annual International Cryptology Conference, CRYPTO 2021, Virtual Event, August 16-20, 2021, Proceedings, Part I*, pp. 94–124 (2021).

[10] Wang, Y. and Malluhi, Q. M.: Reducing Garbled Circuit Size While Preserving Circuit Gate Privacy, *Progress in Cryptology - AFRICACRYPT 2024 - 15th International Conference on Cryptology in Africa, Douala, Cameroon, July 10-12, 2024, Proceedings*, pp. 149–173 (2024).

[11] Xu, F., Hu, H. and Xu, C.: Bitwise Garbling Schemes - A Model with $\frac{3}{2}\kappa$-bit Lower Bound of Ciphertexts, *IACR Cryptol. ePrint Arch.*, p. 1532 (2024).

[12] Yao, A. C.: How to Generate and Exchange Secrets (Extended Abstract), *27th Annual Symposium on Foundations of Computer Science, Toronto, Canada, 27-29 October 1986*, pp. 162–167 (1986).

[13] Zahur, S., Rosulek, M. and Evans, D.: Two Halves Make a Whole - Reducing Data Transfer in Garbled Circuits Using Half Gates, *Advances in Cryptology - EUROCRYPT 2015 - 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pp. 220–250 (2015).