# Reducing Garbled Circuit Size While Preserving Circuit Gate Privacy

Yongge Wang[1] and Qutaibah m. Malluhi[2]

[1]UNC Charlotte

[2]Qatar University

July 10-12, 2024 / Douala, Cameroon

# Outline

# Yao's Garbled Circuit for gate $z = g(x, y)$

- Assigns two random values $k_x^0, k_x^1, k_y^0, k_y^1, k_z^0, k_z^1$ to each wire corresponding to 0 and 1 values of the wire.
- Let the garbled gate be:

$$
\begin{aligned}
k_z^{g(1,1)} &\oplus H_g(k_x^1 || k_y^1) \\
k_z^{g(1,0)} &\oplus H_g(k_x^1 || k_y^0) \\
k_z^{g(0,1)} &\oplus H_g(k_x^0 || k_y^1) \\
k_z^{g(0,0)} &\oplus H_g(k_x^0 || k_y^0)
\end{aligned}
$$

- Given $k_x^{b_x}, k_y^{b_y}$, one can get $k_z^{g(b_x, b_y)}$
- But... which entry to use? one needs to know the value of $b_x, b_y$. This is addressed using a random mapping

# Yao's Garbled Circuits

- A secret random permutation $\pi_x$ over $\{0,1\}$ for each wire.
- The garbled values for the wire $x$: $k_x^0||\pi_x(0)$ and $k_x^1||\pi_x(1)$
- For any $b \in \{0,1\}$, we have $b = \pi_x(b) \oplus \pi_x(0)$.
- For a gate $z = g(x,y)$, the garbled gate $\tilde{g}$ consists of four ciphertexts that are ordered using the external index $\pi_x(b_x)||\pi_y(b_y)$. Assume that $\pi_x(0) = \pi_y(0) = 1$ and $\pi_x(1) = \pi_y(1) = 0$, then $\tilde{g}$ is:

$$\pi_x(1)||\pi_y(1) : (k_z^{g(1,1)}||\pi_z(g(1,1))) \oplus H_g(k_x^1 \circ k_y^1)$$
$$\pi_x(1)||\pi_y(0) : (k_z^{g(1,0)}||\pi_z(g(1,0))) \oplus H_g(k_x^1 \circ k_y^0)$$
$$\pi_x(0)||\pi_y(1) : (k_z^{g(0,1)}||\pi_z(g(0,1))) \oplus H_g(k_x^0 \circ k_y^1)$$
$$\pi_x(0)||\pi_y(0) : (k_z^{g(0,0)}||\pi_z(g(0,0))) \oplus H_g(k_x^0 \circ k_y^0)$$

# GRR3: Naor, Pinkas, and Sumner 1999

Naor, Pinkas, and Sumner observed that one can choose a randomly fixed pair $(b_x, b_y) \in \{0,1\}^2$ and let

$$k_z^{g(b_x,b_y)} || \pi(g(b_x, b_y)) = H_g(k_x^{b_x} \circ k_y^{b_y}).$$

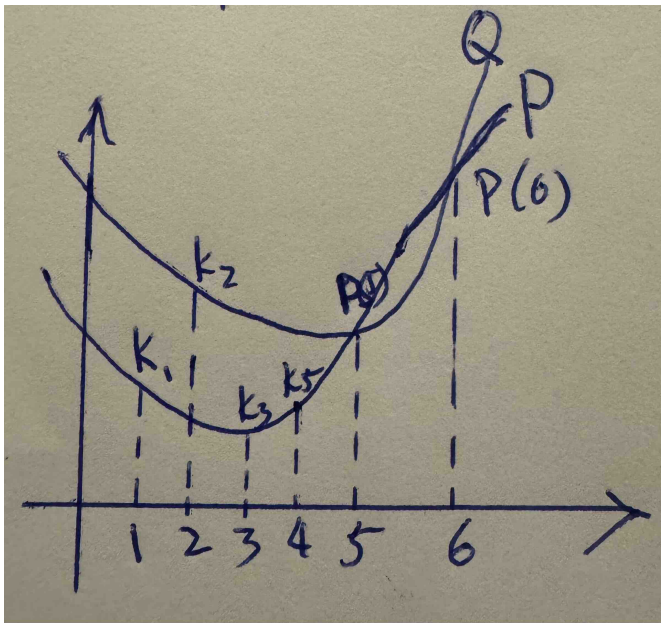
Then the corresponding ciphertext for the row $(\pi(b_x), \pi(b_y))$ is a zero string and one does not need to store it.

# Pinkas et al's Garbled Row Reduction GRR2

- gate is odd/even if its truth table has an odd/even number of '1': AND are OR are odd gates and XOR is even gate
- For an odd gate $g$, assume that the three ciphertexts $C_1, C_3, C_4$ encrypt the same wire label $k_z^b||\pi_z(b)$ via $C_i = (k_z^b||\pi_z(b)) \oplus (K_i||M_i)$ (for $i = 1, 3, 4$) and the ciphertext $C_2$ encrypts the wire label $k_z^{1-b}||\pi_z(1-b)$ via $C_2 = (k_z^{1-b}||\pi_z(1-b)) \oplus (K_2||M_2)$, where $b, M_i \in \{0, 1\}$ for $i = 1, 2, 3, 4$.

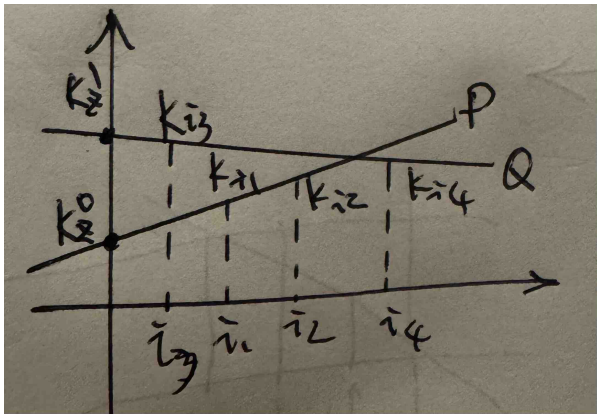
# Pinkas et al's Garbled Row Reduction GRR2

- Let $P(X)$ be a degree two polynomial passing through points $(1, K_1), (3, K_3)$, and $(4, K_4)$. Let $Q(X)$ be another degree two polynomial over $\mathbb{F}_{2^t}$ passing through points $(5, P(5)), (6, P(6))$, and $(2, K_2)$.
- Setting $k_z^b = P(0)$ and $k_z^{1-b} = Q(0)$. The garbled table is $\langle P(5), P(6), c_1, c_2, c_3, c_4 \rangle$



# Pinkas et al's Garbled Row Reduction GRR2

- For an even gate, $g$, assume that $K_{i_1}, K_{i_2}$ encrypt the wire label $k_z^0||\pi_z(0)$ and $K_{i_3}, K_{i_4}$ encrypt the wire label $k_z^1||\pi_z(1)$
- Let $P(X)$ be a linear polynomial passing through $(i_1, K_{i_1})$ and $(i_2, K_{i_2})$, and $Q(X)$ passing through $(i_3, K_{i_3})$ and $(i_4, K_{i_4})$.
- Define $k_z^0 = P(0)$ and $k_z^1 = Q(0)$. The garbled gate is $\langle P(5), Q(5), c_1, c_2, c_3, c_4 \rangle$. Otherwise, as $\langle Q(5), P(5), c_1, c_2, c_3, c_4 \rangle$.



# Zahur, Rosulek, and Evans's half-gates; free-XOR

- odd gate: two ciphertexts; even gate: free.
- A global offset $\Delta$ such that $k_x^1 = k_x^0 \oplus \Delta$ for all wire $x$
- $z = x \wedge y$ is garbled into two ciphertexts:

$$H(k_x^0) \oplus H(k_x^1) \oplus r_y \Delta; \quad H(k_y^0) \oplus H(k_y^1) \oplus k_x^0$$

- $z = x \vee y$ is garbled into two ciphertexts:

$$H(k_x^0) \oplus H(k_x^1) \oplus (1 - r_y)\Delta; \quad H(k_y^0) \oplus H(k_y^1) \oplus k_x^1$$

- The garbled labels for the output wire $z$ is

$$k_z^0 = H(k_x^{r_x}) \oplus g(r_x, r_y)\Delta \oplus H(k_y^{r_y});$$

# Gate Privacy preserving Garbled Row Reduction GPGRR2

- Pinkas et al's GRR2 garbling scheme leaks the number and positions of even/odd gate types. For example, an evaluator evaluates a garbled odd gate using degree two polynomial interpolation and evaluates a garbled even gate using linear interpolation.
- The free-XOR techniques proposed by Kolesnikov and Schneider leaks the number and positions of XOR gates and the half-gates techniques by Zahur, Rosulek, and Evans leaks the number and positions of XOR gates also.
- Our solution: propose a gate privacy preserving garbled row reduction GPGRR2 technique to garble circuits with security for $\Phi_{\mathtt{topo}}$.

# Proposed GPGRR2

- The circuit $C$ will be garbled in such a way that for all wires $x$, the garbled values $k_x^0||\pi_x(0)$ and $k_x^1||\pi_x(1)$ for the wire $x$ satisfy the following invariance property:

$$k_x^1 = k_x^0 + \Delta \mod 2^t \tag{1}$$

# Proposed GPGRR2

Let $k_x^0||\pi_x(0)$, $k_x^1||\pi_x(1)$, $k_y^0||\pi_y(0)$, and $k_y^1||\pi_y(1)$ be the garbled input wire values for the wires $x$ and $y$ respectively. Let $k_z^0||\pi_z(0)$, $k_z^1||\pi_z(1)$ be the garbled output wire values for the output wire $z = g(x, y)$ that will be defined. Define the operator $\circ$ as the integer addition modulo $2^t$. Then we have

$$\begin{aligned} k_x^0 \circ k_y^0 &= k_x^0 + k_y^0 = \bar{x}_1 \mod 2^t \\ k_x^0 \circ k_y^1 &= k_x^1 \circ k_y^0 = k_x^0 + k_y^0 + \Delta = \bar{x}_1 + \Delta \mod 2^t \\ k_x^1 \circ k_y^1 &= k_x^0 + k_y^0 + 2\Delta = \bar{x}_1 + 2\Delta \mod 2^t \end{aligned} \tag{2}$$

For these garbled input wire values, we have

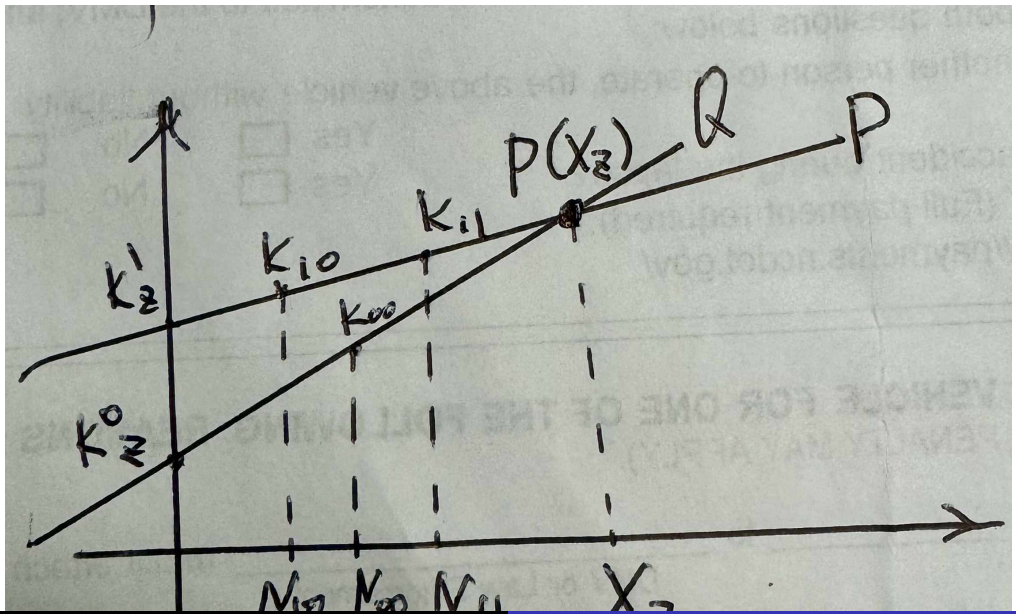
$$\begin{aligned} K_{00}||M_{00}||N_{00} &= H_g(k_x^0 \circ k_y^0) = H_g(\bar{x}_1 \mod 2^t) \\ K_{01}||M_{01}||N_{01} &= H_g(k_x^0 \circ k_y^1) = H_g(\bar{x}_1 + \Delta \mod 2^t) \\ K_{10}||M_{10}||N_{10} &= H_g(k_x^1 \circ k_y^0) = H_g(\bar{x}_1 + \Delta \mod 2^t) \\ K_{11}||M_{11}||N_{11} &= H_g(k_x^1 \circ k_y^1) = H_g(\bar{x}_1 + 2\Delta \mod 2^t) \end{aligned} \tag{3}$$

# Proposed linear polynomial GPGRR2

## Garbling an odd gate *g* OR

.

- $P(X)$ passes through: $(N_{10}, K_{10})$ and $(N_{11}, K_{11})$. $k_z^1 = P(0)$ and $k_z^0 = k_z^1 - \Delta \mod 2^t$.
- $Q(X)$ passes through $(0, k_z^0)$ and $(N_{00}, K_{00})$. $X_z$ be a solution for $P(X) = Q(X)$. The garbled table for gate $g$ is: $\langle X_z, P(X_z), c_1, c_2, c_3, c_4 \rangle = \langle X_z, Q(X_z), c_1, c_2, c_3, c_4 \rangle$



# Proposed linear polynomial GPGRR2

## Garbling an odd gate $g$ (AND)

.

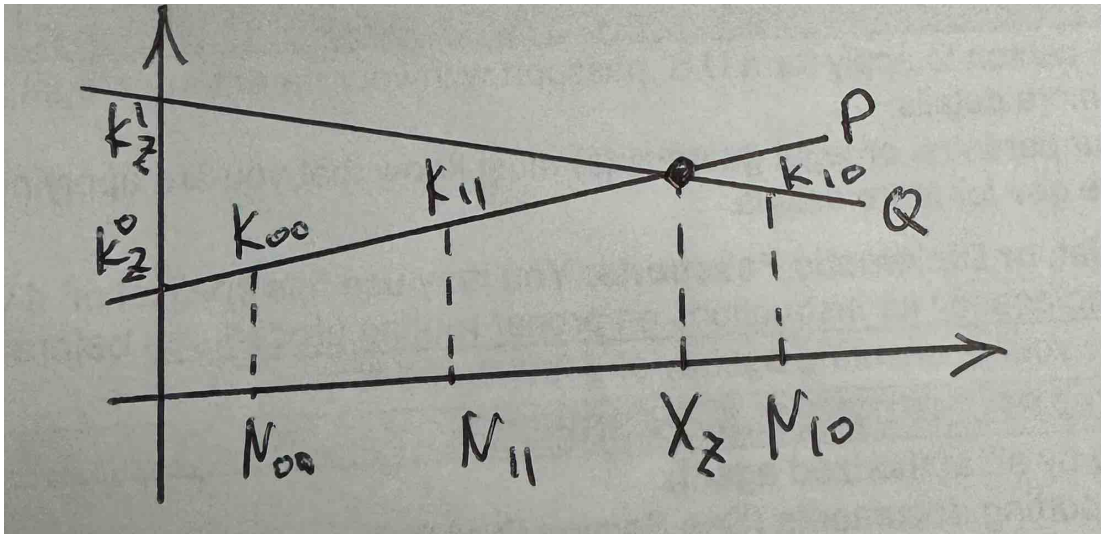
- For an AND gate, start by selecting a linear polynomial, denoted as $P(X)$, that passes through the points $(N_{10}, K_{10})$ and $(N_{00}, K_{00})$. Define $k_z^0$ as $P(0)$ and $k_z^1$ as $k_z^0 + \Delta \mod 2^t$. Next, construct another linear polynomial, denoted as $Q(X)$, passing through the points $(0, k_z^1)$ and $(N_{11}, K_{11})$. Essentially, $P(X)$ is interpolated based on when the output of the AND gate is 0, while $Q(X)$ is interpolated for an output of 1. The subsequent steps remain unchanged.

# Proposed linear polynomial GPGRR2

## Garbling an even gate $g$ XOR

.

- $P(X)$ passes through: $(N_{00}, K_{00})$ and $(N_{11}, K_{11})$. Set $k_z^0 = P(0)$ and $k_z^1 = k_z^0 + \Delta \mod 2^t$.
- $Q(X)$ passes through $(0, k_z^1)$ and $(N_{10}, K_{10})$.
- Find $X_z$, a solution of $P(X) = Q(X)$ over $\mathbb{F}_{2^t}$.
- The garbled table for gate $g$ is: $\langle X_z, P(X_z), c_1, c_2, c_3, c_4 \rangle$



# Private function evaluation

In a two party PFE protocol, participant $P_1$ has a string $x$, participant $P_2$ has a function $f$ and the outcome of the protocol is that $P_2$ learns $f(x)$ and nothing about $x$ (beyond its length), while $P_1$ learns nothing about $f$ (beyond side information we are willing to leak, such as the number of gates in the circuit $f$). Similarly, the outcome of the two party PFE protocol could be that $P_1$ learns $f(x)$ and nothing about $f$, while $P_2$ learns nothing about $x$. For the general case that $P_2$ has a private input $x_2$ himself, one can include the value of $x_2$ in the circuit computing $f$ itself.

# PFE in semi-honest security model: `semiPFE`

1. Given the pair $(n, l)$, $P_1$ generates a sequence of $n$ gates.
2. $P_2$ obliviously connects these gates to form a circuit $C_f$ using a singly homomorphic encryption scheme.
3. $P_1$ produces a garbled circuit corresponding to the circuit $C_f$ by garbling the $n$ gates independently (which are connected obliviously).
4. $P_1$ gives an encoded version of the input $x$ to $P_2$ and $P_2$ evaluates the garbled circuit to obtain the circuit output $C_f(x) = f(x)$.

# PFE protocols against malicious participants

- The protocol in the previous slide is insecure against active adversaries.
- In protocol `semiPFE`, $P_2$ may generate the wires $\texttt{encG}_i$ in a malicious way to learn $P_1$'s private input $x$.
- Zero-knowledge proofs could be used to make the protocol `semiPFE` secure against active malicious participants.

# Other PFE protocols based on our garbled circuits

- Circuit private PFE protocols with malicious $P_1$
- Secure PFE protocols against two malicious participants

# Questions

Questions?