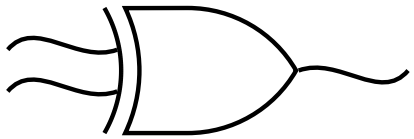
fleXOR: flexible garbling for XOR gates that beats free-XOR



▷ Vladimir Kolesnikov » Alcatel·Lucent

▷ Payman Mohassel » University of Calgary

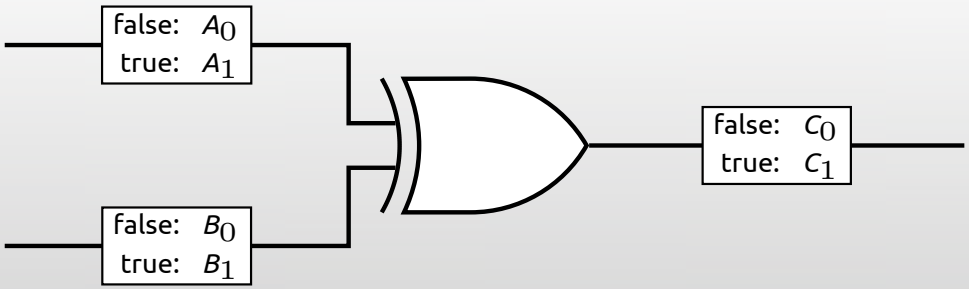
▶ Mike Rosulek » Oregon State University OSU

Abstract: Most implementations of Yao's garbled circuit approach for 2-party secure computation use the *free-XOR* optimization of Kolesnikov & Schneider (ICALP 2008). We introduce an alternative technique called *flexible-XOR* (fleXOR) that generalizes free-XOR and offers several advantages. First, fleXOR can be instantiated under a weaker hardness assumption on the underlying cipher/hash function (related-key security only, compared to related-key and circular security required for free-XOR). Second, even though XOR gates are not always "free" in our approach, the other (non-XOR) gates can be optimized more heavily than what is possible when using free-XOR. For many circuits of cryptographic interest, fleXOR can yield a significantly (over 30%) smaller garbled circuit than any other known techniques (including free-XOR) or their combinations.

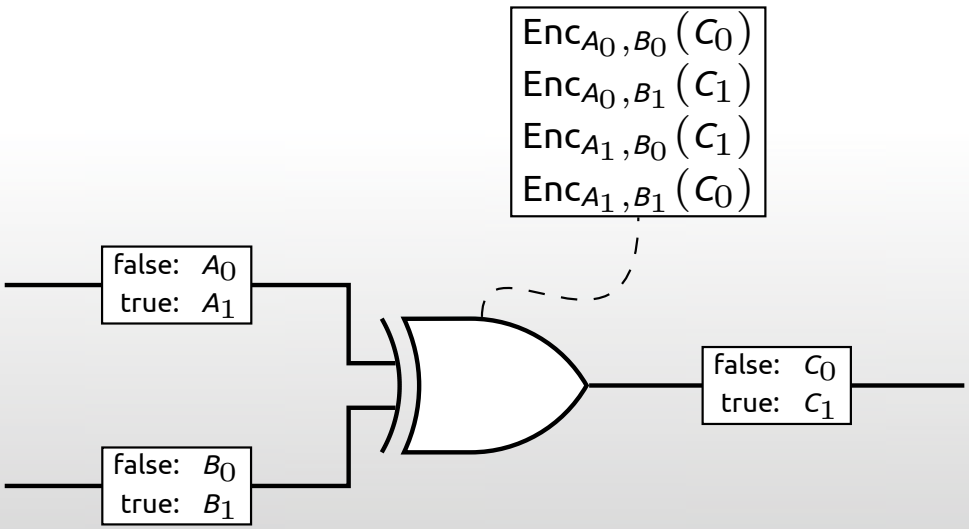
Our garbling schemes can be directly applied in the semi-honest model, as well as compiled into the malicious setting using any existing technique.

background: garbled circuit

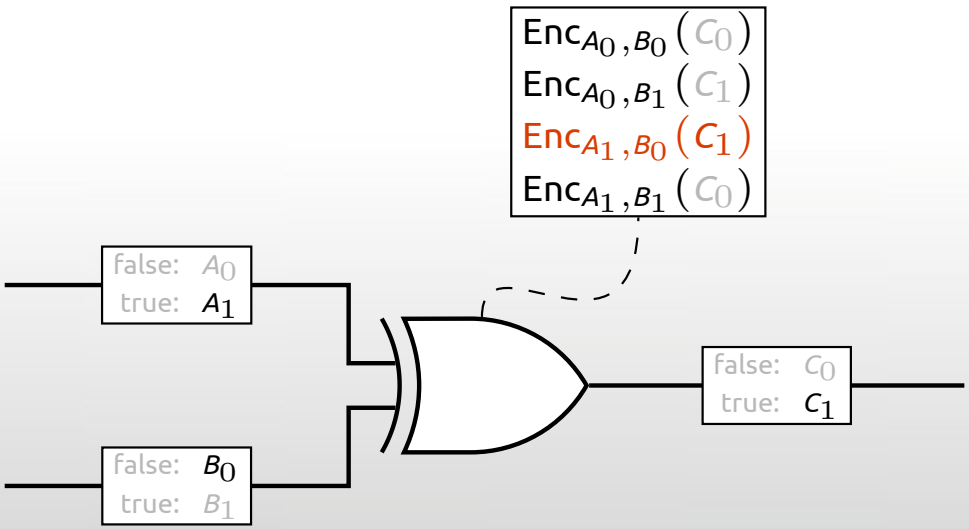
background: garbled circuit



background: garbled circuit

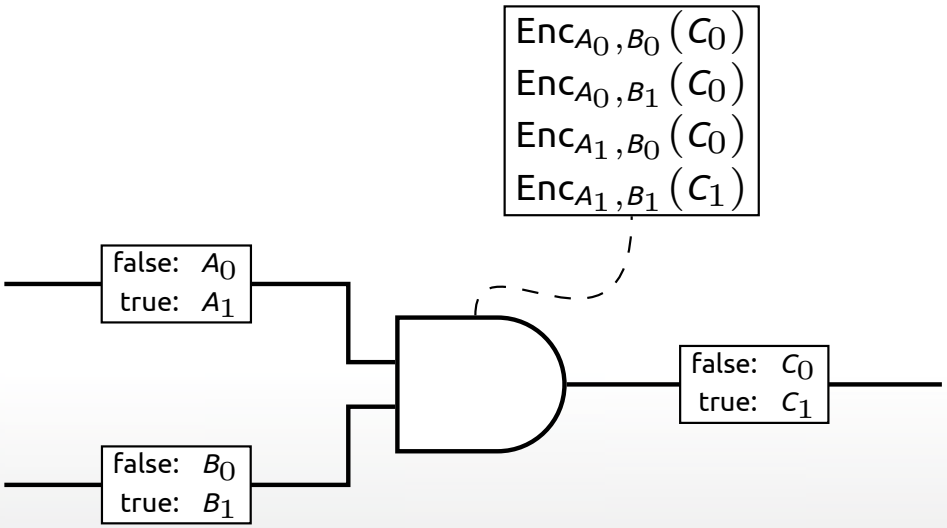


background: garbled circuit



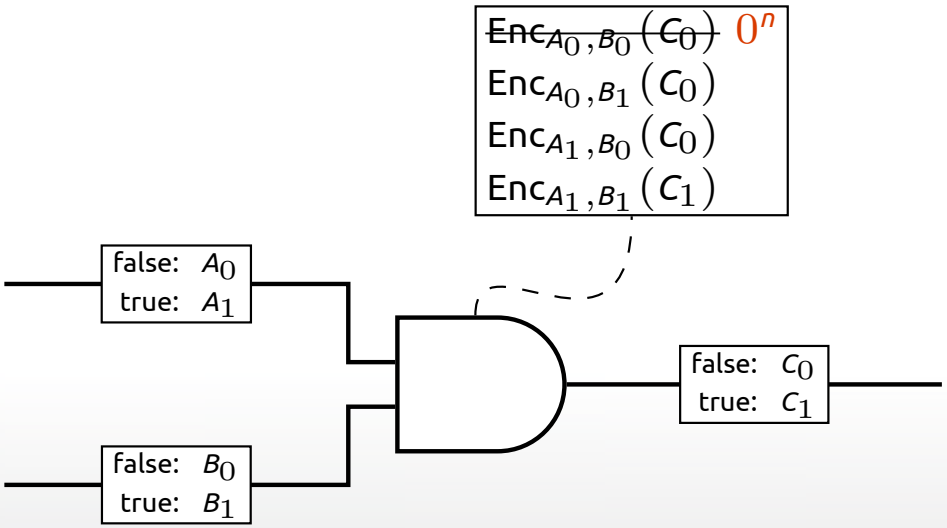
background: row reduction

background: row reduction



Garbled row reduction [NaorPinkasSumner99,PinkasSchneiderSmartWilliams09]

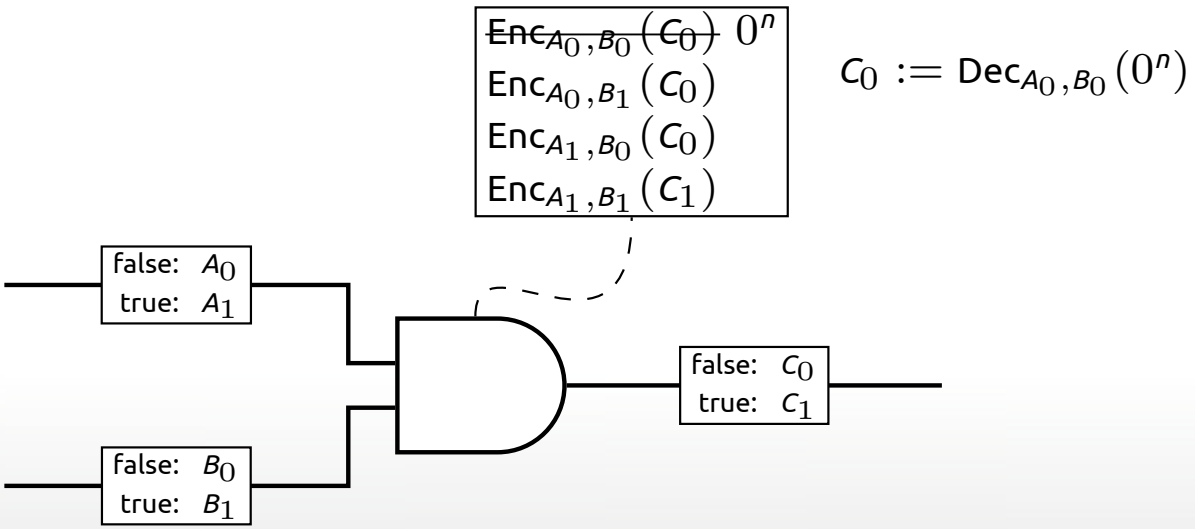
background: row reduction



Garbled row reduction [NaorPinkasSumner99,PinkasSchneiderSmartWilliams09]

- Fix one of the ciphertexts to be all zeroes

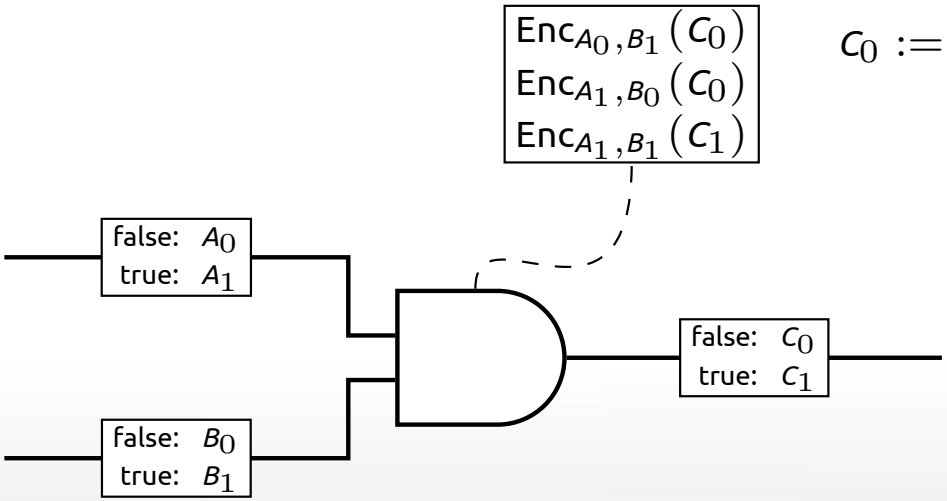
background: row reduction



Garbled row reduction [NaorPinkasSumner99,PinkasSchneiderSmartWilliams09]

- Fix one of the ciphertexts to be all zeroes
- Corresponding wire label must be $\mathrm{Dec}(0^n)$, not uniform

background: row reduction

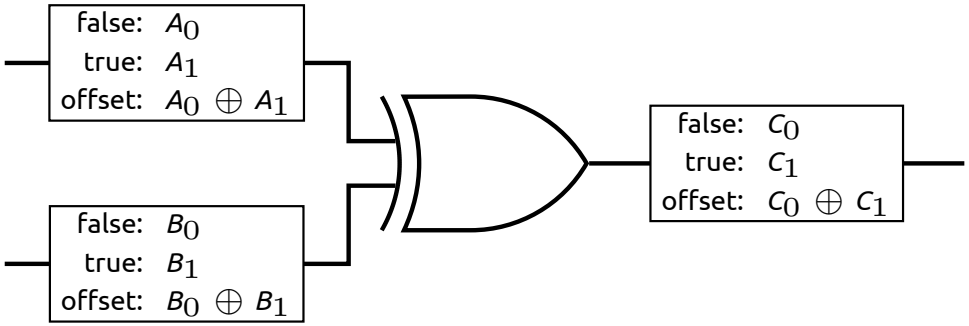


Garbled row reduction [NaorPinkasSumner99,PinkasSchneiderSmartWilliams09]

- Fix one of the ciphertexts to be all zeroes
- Corresponding wire label must be $\mathrm{Dec}(0^n)$, not uniform
- Only 3 ciphertexts needed for garbled gate
- More advanced technique reduces size to 2 ciphertexts

background: offsets & free XOR

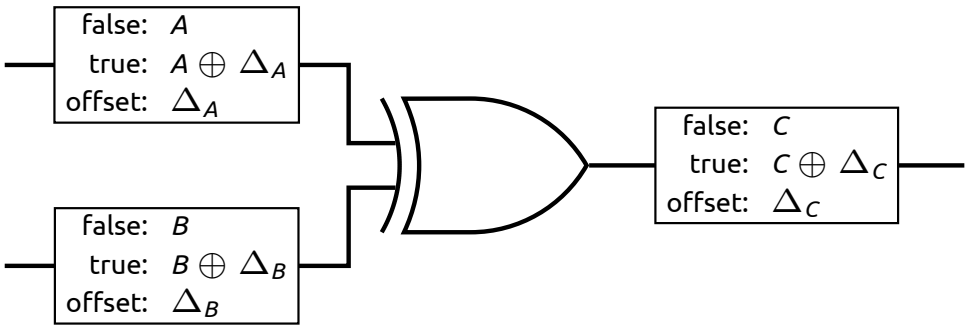
background: offsets & free XOR



Definition

Offset of a wire = XOR of its two wire labels

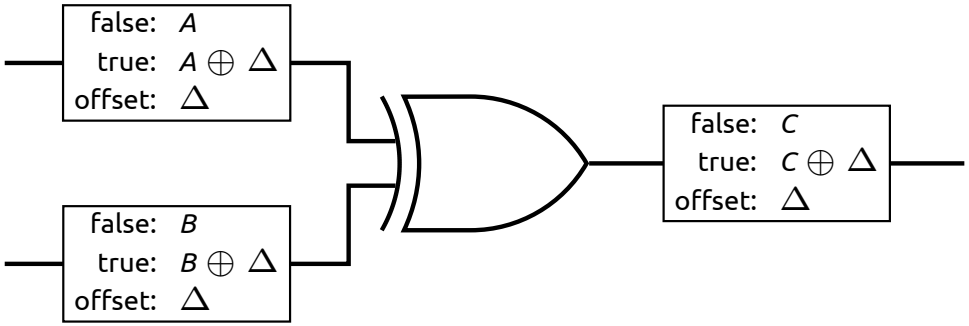
background: offsets & free XOR



Definition

Offset of a wire = XOR of its two wire labels

background: offsets & free XOR



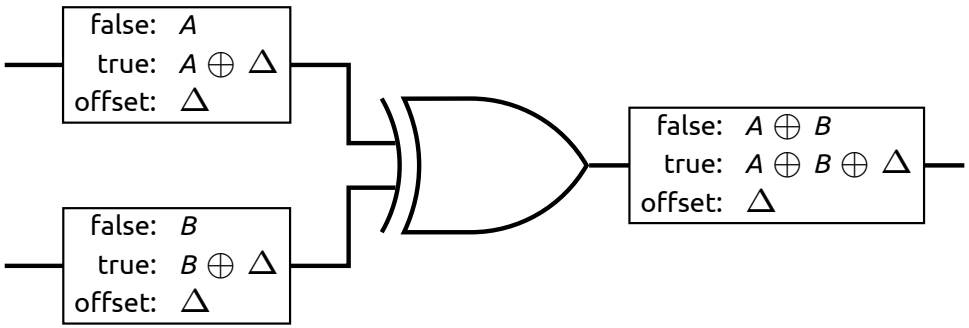
Definition

Offset of a wire = XOR of its two wire labels

Free XOR optimization [KolesnikovSchneider08]:

- all wires have *same* (secret) offset Δ

background: offsets & free XOR



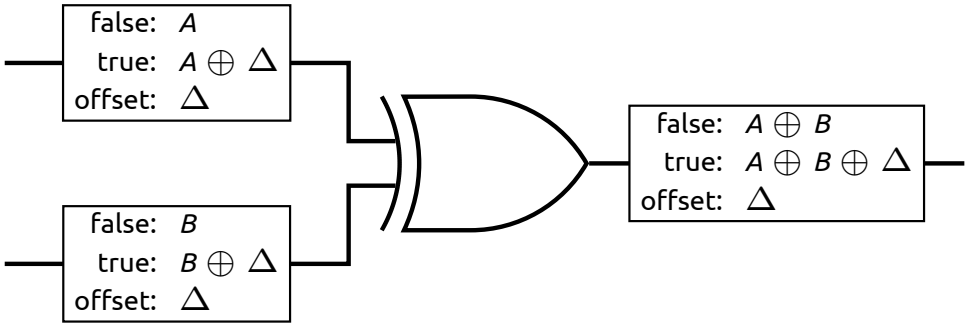
Definition

Offset of a wire = XOR of its two wire labels

Free XOR optimization [KolesnikovSchneider08]:

- all wires have *same* (secret) offset Δ
- wire *labels* for XOR gate satisfy $C = A \oplus B$

background: offsets & free XOR



Definition

Offset of a wire = XOR of its two wire labels

Free XOR optimization [KolesnikovSchneider08]:

- all wires have *same* (secret) offset Δ
- wire *labels* for XOR gate satisfy $C = A \oplus B$
- compute output wire label by XOR'ing input wire labels (no crypto!)

free XOR

Free XOR limitations:

1. Requires strong circularity hardness assumption [ChoiKatzKumaresanZhou12]
2. Incompatible with 4-to-2 row reduction [PinkasSchneiderSmartWilliams09]

free XOR

Free XOR limitations:

1. Requires strong circularity hardness assumption [ChoiKatzKumaresanZhou12]
2. Incompatible with 4-to-2 row reduction [PinkasSchneiderSmartWilliams09]

Motivating Question

Can we overcome these limitations, while retaining Free XOR's benefits (as much as possible)?

our results

	Free XOR:	**FleXOR:**
Hardness assumption:	circularity [CKKZ12]	
Compatible with 2-row-reduction?	No! non-XOR gates cost 3κ bits	

Garbled circuit size (ciphertexts per gate)

scheme	assump	AES	DES	SHA1	SHA2	HamDst	IntMult
classical	**OWF**	2.00	**2.00**	2.00	**2.00**	2.00	2.00
Free XOR	circular	**0.64**	2.79	**1.82**	2.05	**0.50**	**0.90**

our results

	Free XOR:	**FleXOR:**
Hardness assumption:	circularity [CKKZ12]	circularity, or related-key only slight increase in GC size
Compatible with 2-row-reduction?	No! non-XOR gates cost 3κ bits	

Garbled circuit size (ciphertexts per gate)

scheme	assump	AES	DES	SHA1	SHA2	HamDst	IntMult
classical	**OWF**	2.00	**2.00**	2.00	**2.00**	2.00	2.00
Free XOR	circular	**0.64**	2.79	**1.82**	2.05	**0.50**	**0.90**
FleXOR	related-key	0.76	2.84	2.02	2.26	0.67	1.15

our results

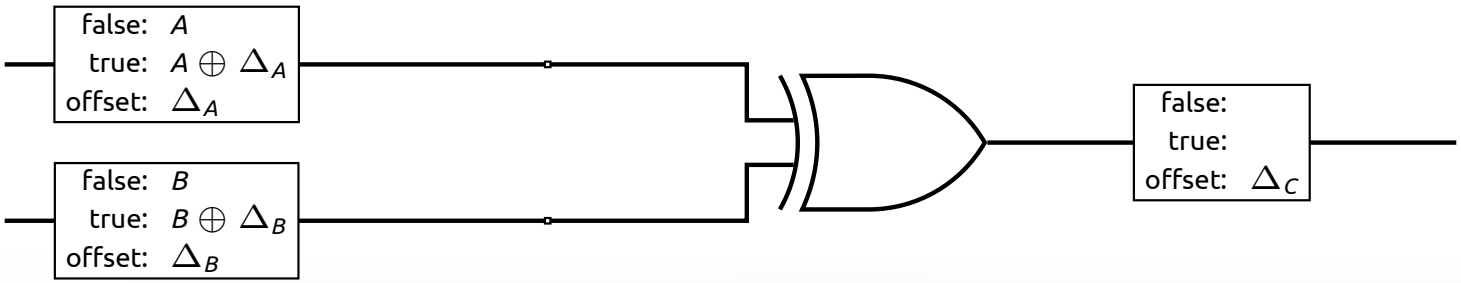
	Free XOR:	**FleXOR:**
Hardness assumption:	circularity [CKKZ12]	circularity, or related-key only slight increase in GC size
Compatible with 2-row-reduction?	No! non-XOR gates cost 3κ bits	Yes! up to 30% smaller GC overall

Garbled circuit size (ciphertexts per gate)

scheme	assump	AES	DES	SHA1	SHA2	HamDst	IntMult
classical	**OWF**	2.00	2.00	2.00	2.00	2.00	2.00
Free XOR	circular	**0.64**	2.79	1.82	2.05	**0.50**	**0.90**
FleXOR	related-key	0.76	2.84	2.02	2.26	0.67	1.15
FleXOR	circular	0.72	**1.89**	**1.39**	**1.56**	**0.50**	0.94

fleXOR garbling

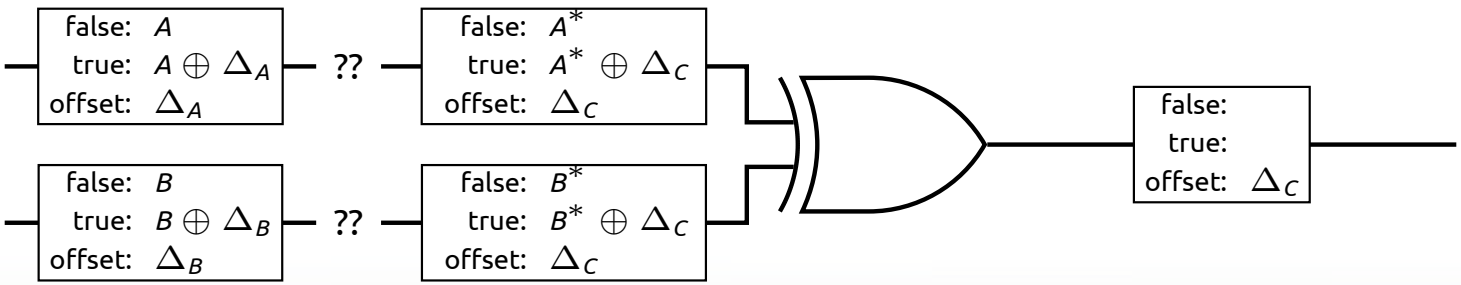
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C

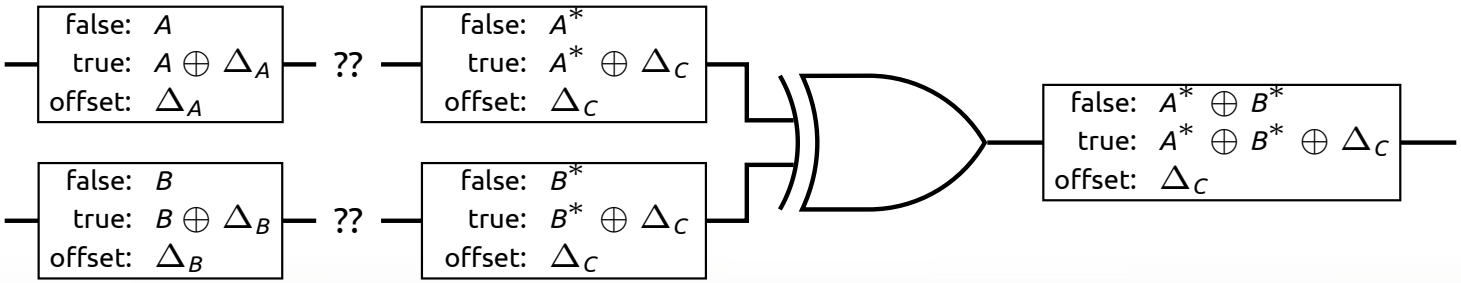
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C

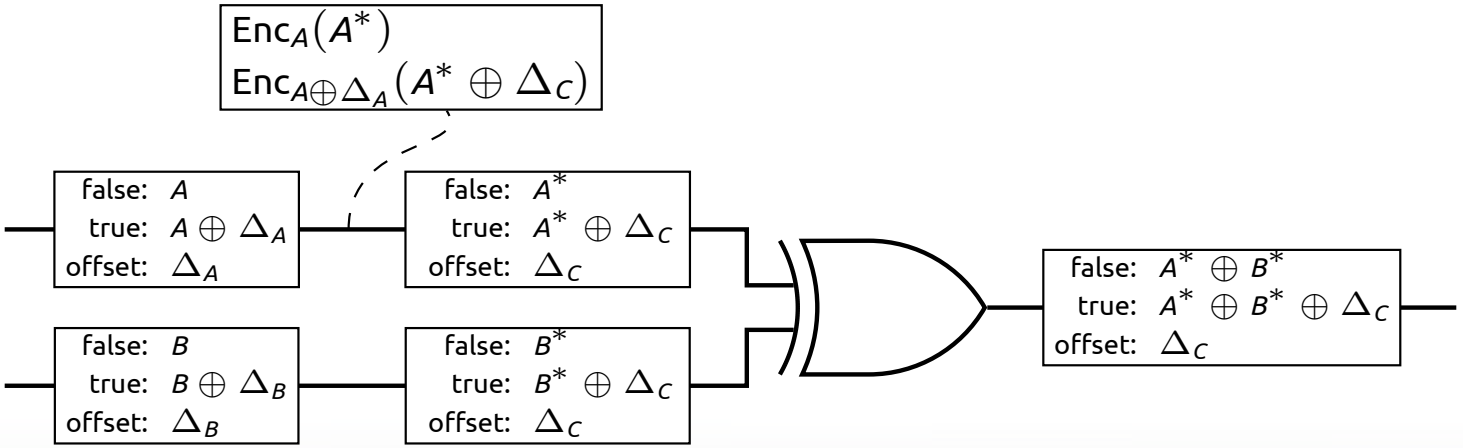
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR

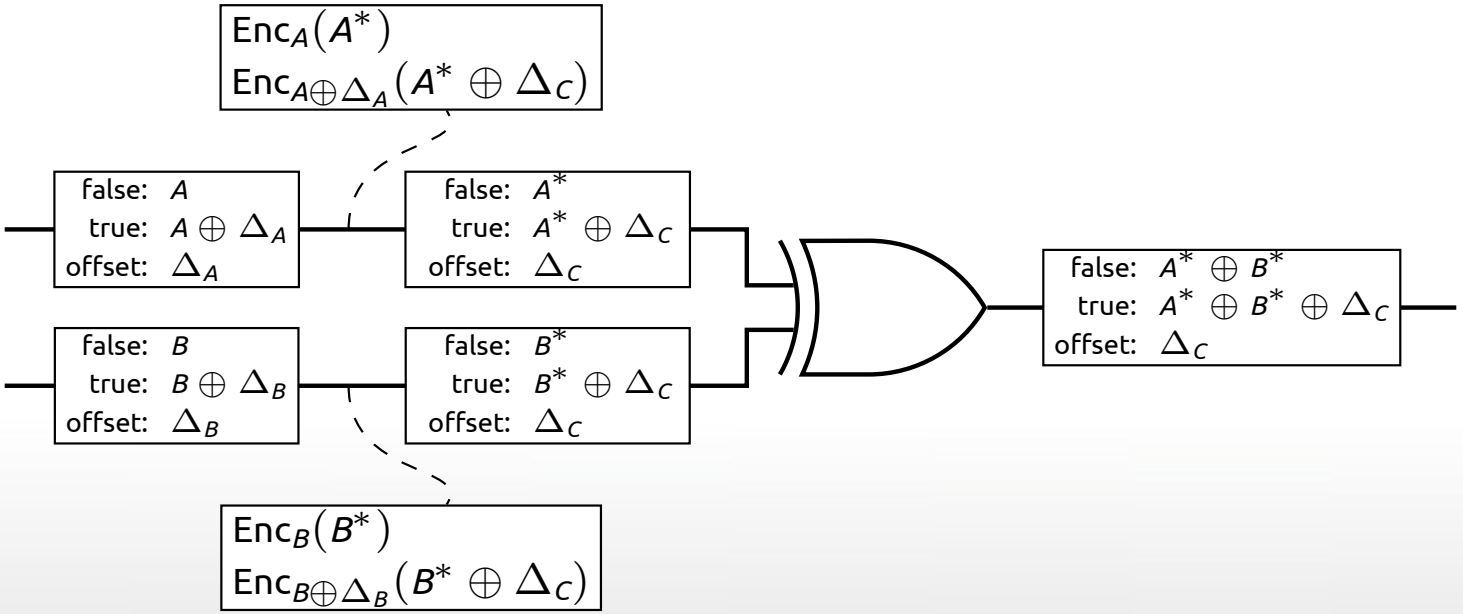
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR

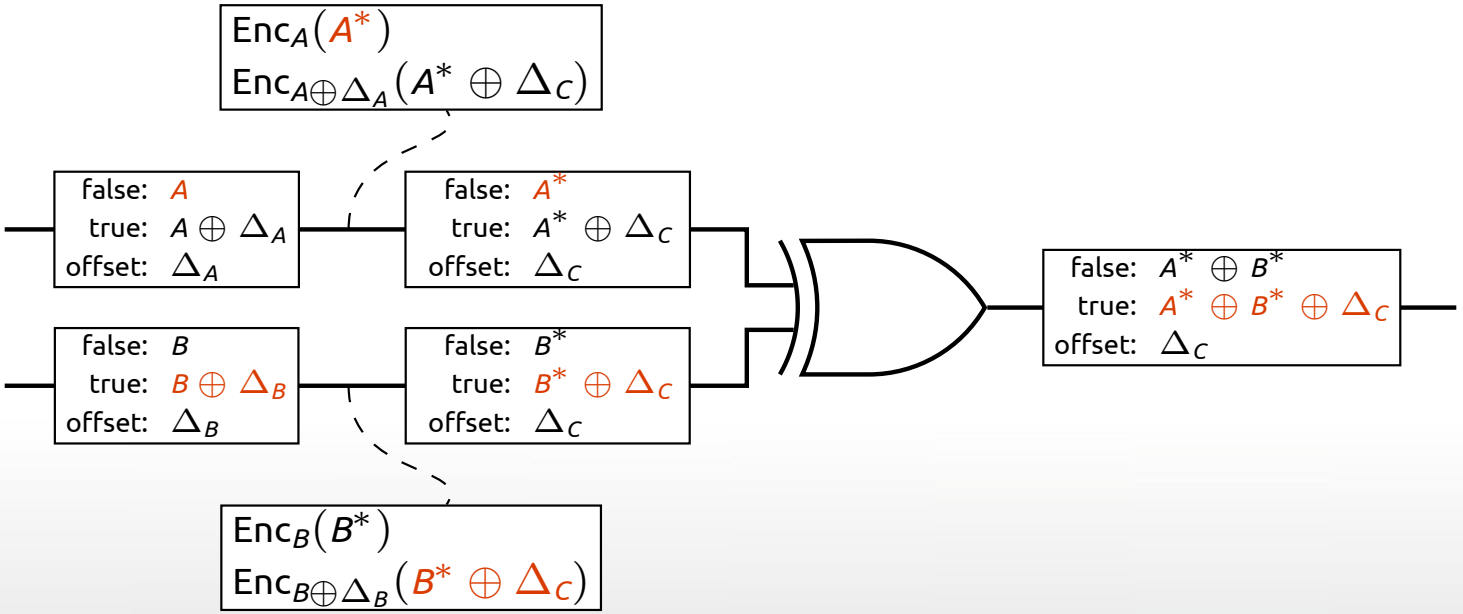
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR

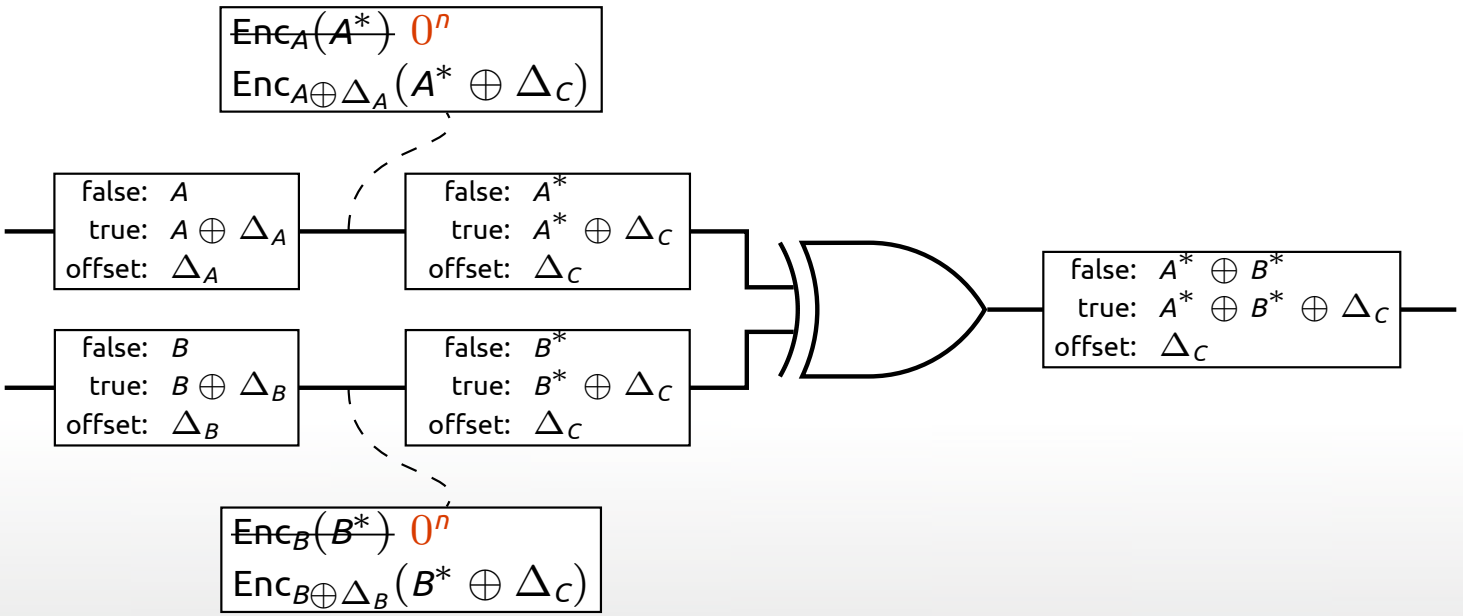
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR

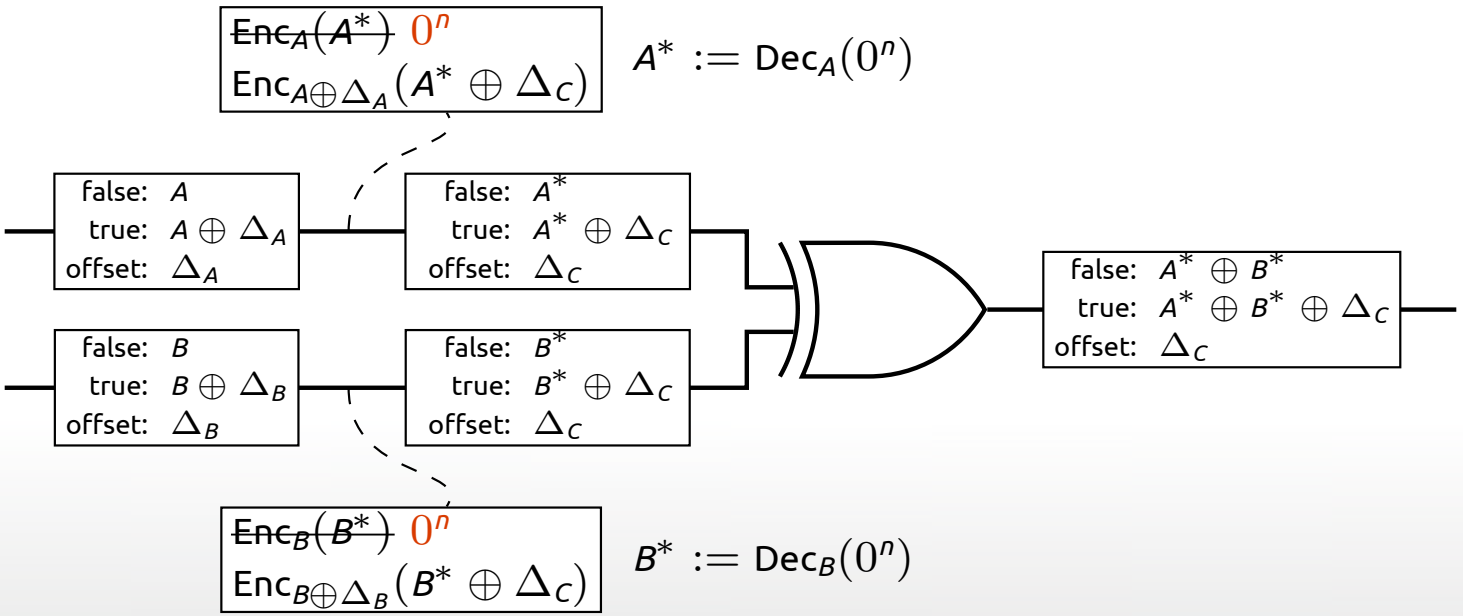
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR
- apply row reduction

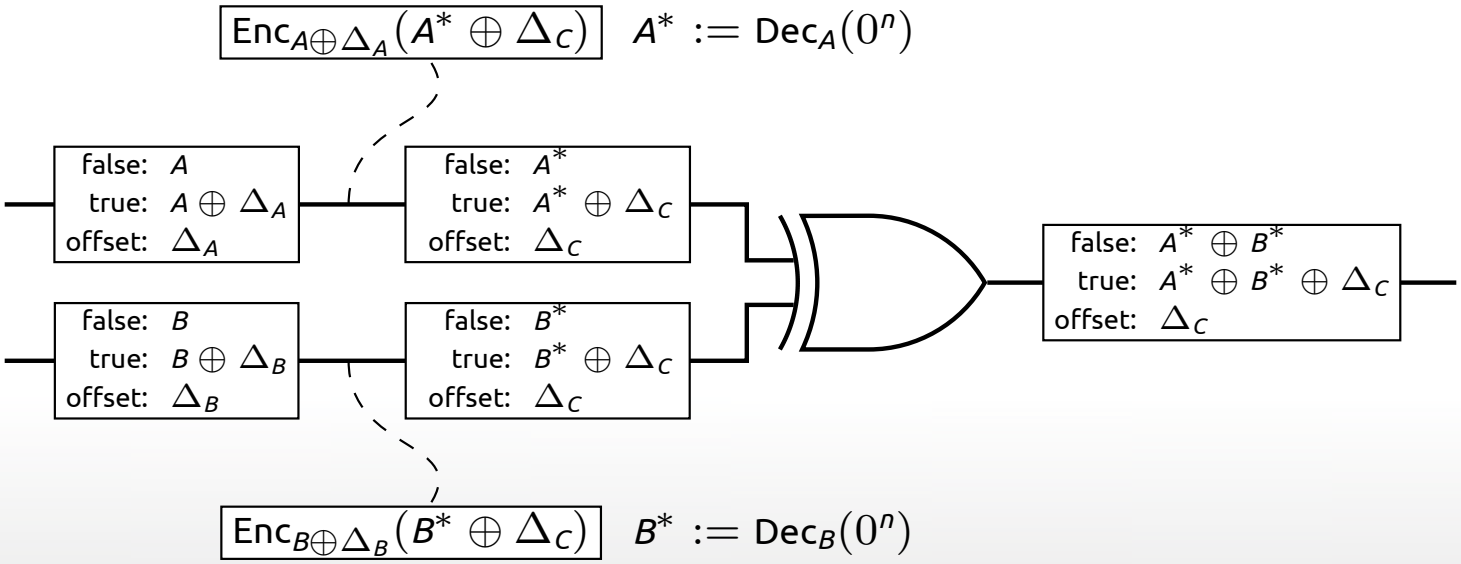
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR
- apply row reduction

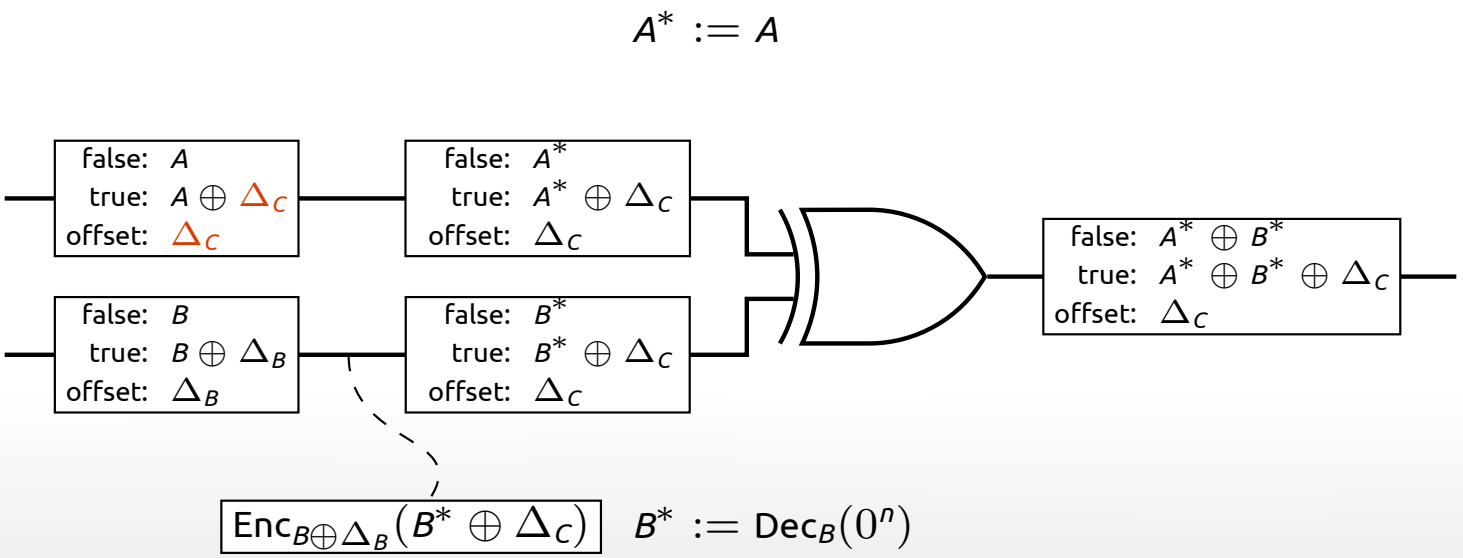
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR
- apply row reduction: each "adjustment" requires 1 ciphertext

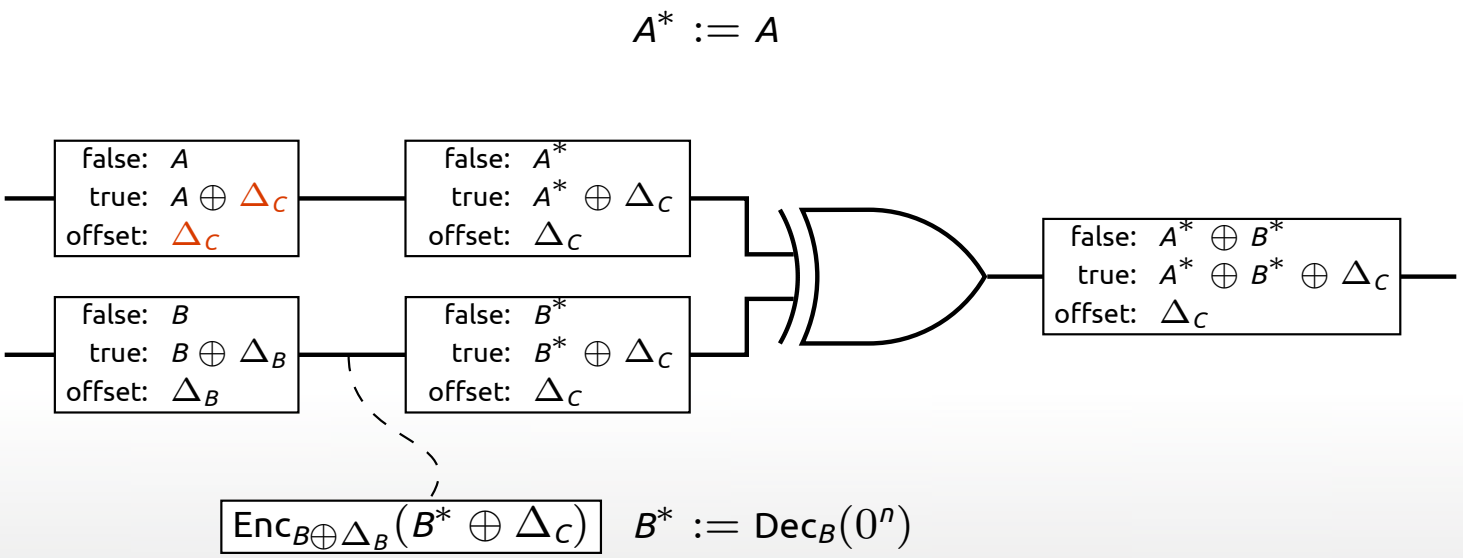
fleXOR garbling



Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR
- apply row reduction: each "adjustment" requires 1 ciphertext
- if $\Delta_A = \Delta_C$, no need to "adjust" first wire at all!

fleXOR garbling



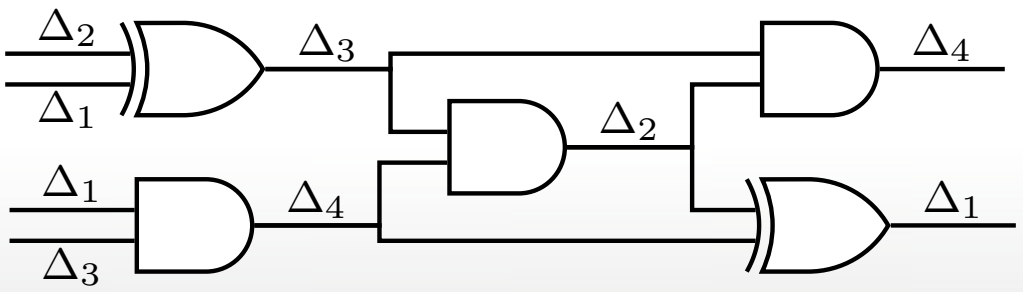
Flexible XOR (fleXOR) technique [this work]:

- "adjust" offsets of both input wires to Δ_C, then use free XOR
- apply row reduction: each "adjustment" requires 1 ciphertext
- if $\Delta_A = \Delta_C$, no need to "adjust" first wire at all!
- **garble XOR gate using 0, 1, or 2 ciphertexts**
  - $\cdots$ depending on how many of $\{\Delta_A, \Delta_B, \Delta_C\}$ are distinct

wire orderings

Wire ordering:

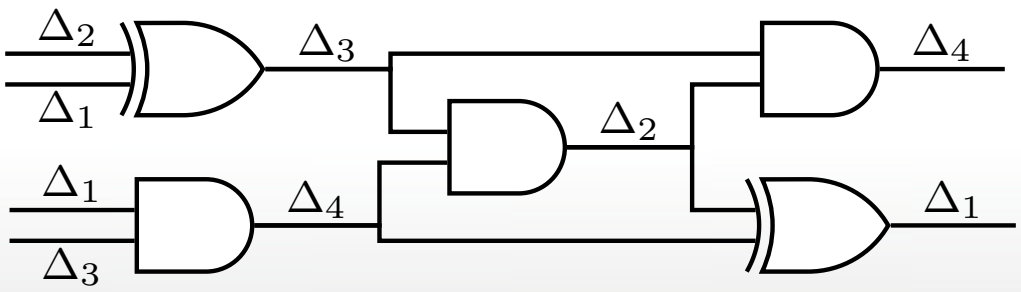
Group circuit's **wires** into equivalence classes (same class $\Leftrightarrow$ same offset)



wire orderings

Wire ordering:

Group circuit's **wires** into equivalence classes (same class $\Leftrightarrow$ same offset)

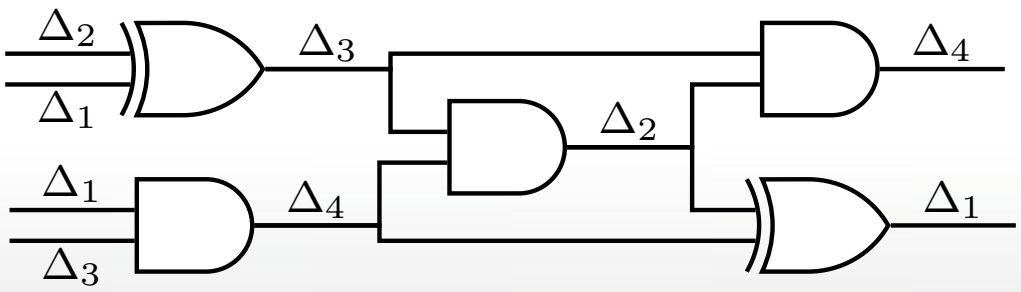


How should we choose wire orderings to minimize total cost of garbling XOR gates?

wire orderings

Wire ordering:

Group circuit's **wires** into equivalence classes (same class $\Leftrightarrow$ same offset)



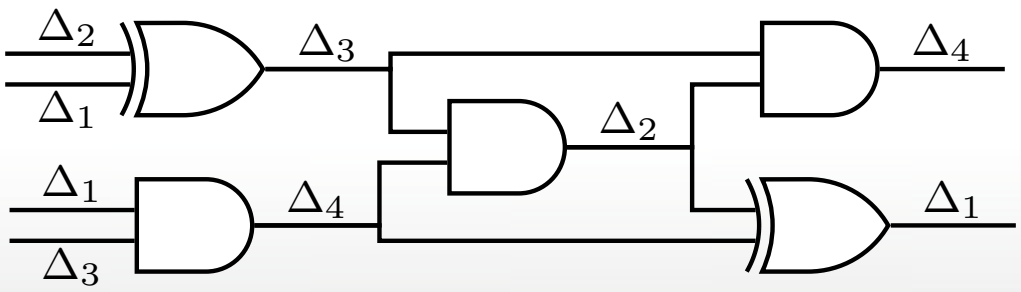
How should we choose wire orderings to minimize total cost of garbling XOR gates?

- while avoiding circularity problem of Free-XOR?
- while retaining compatibility with 2-row-reduction for non-XOR gates?

wire orderings

Wire ordering:

Group circuit's **wires** into equivalence classes (same class $\Leftrightarrow$ same offset)



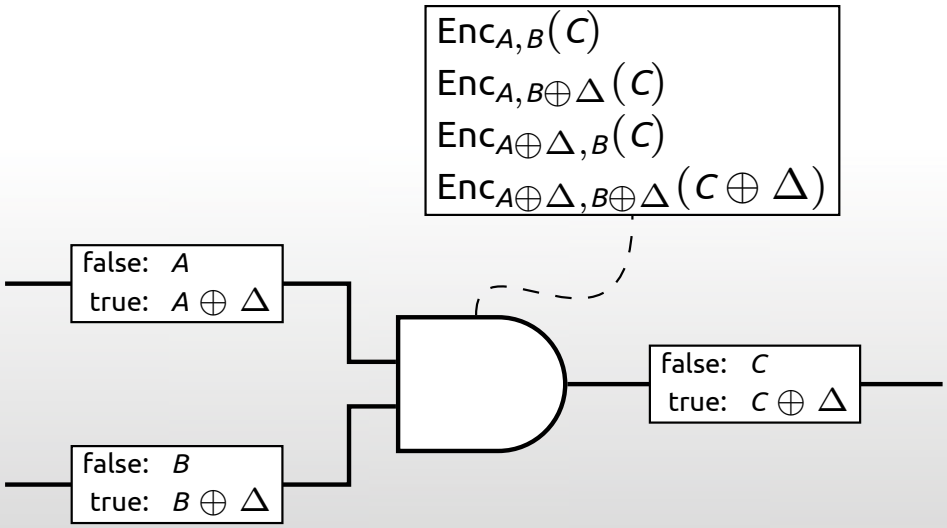
How should we choose wire orderings to minimize total cost of garbling XOR gates?

- while avoiding circularity problem of Free-XOR?
- while retaining compatibility with 2-row-reduction for non-XOR gates?

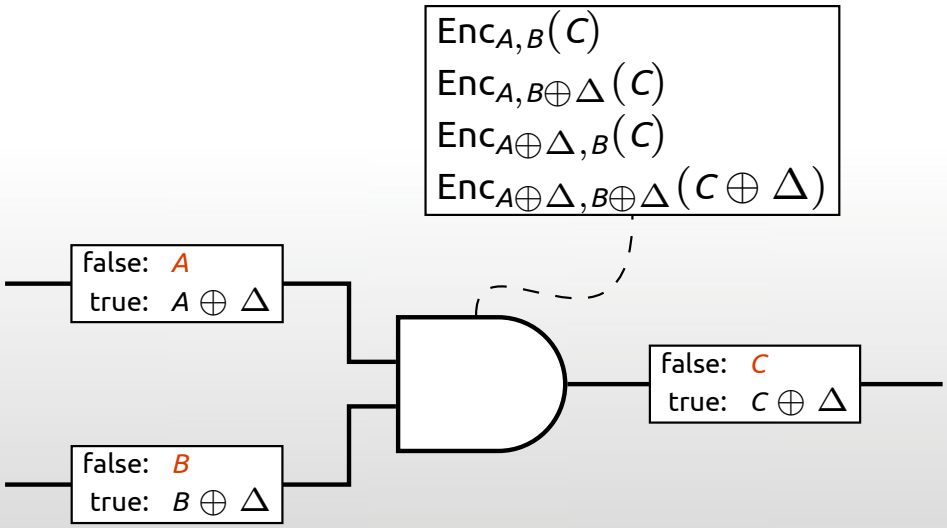
combinatorial properties of wire ordering

removing circularity

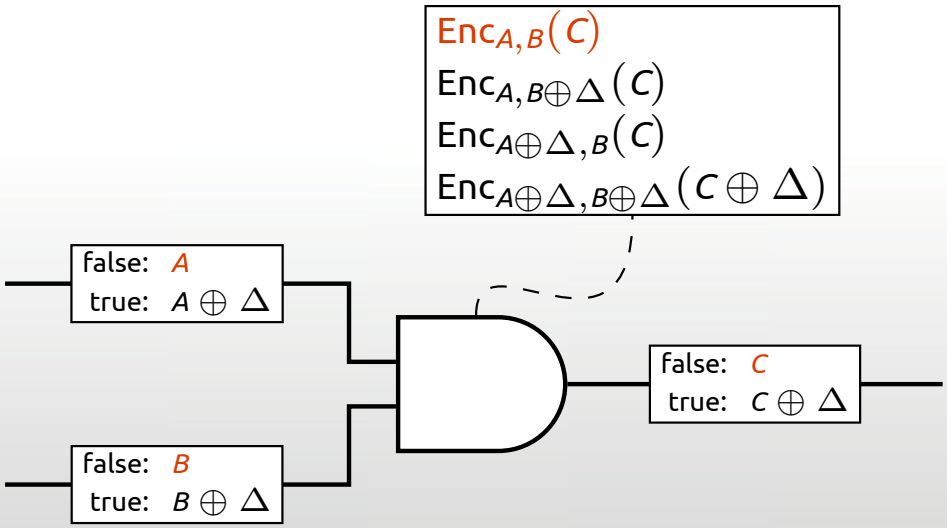
why does free-XOR require circularity assumption?



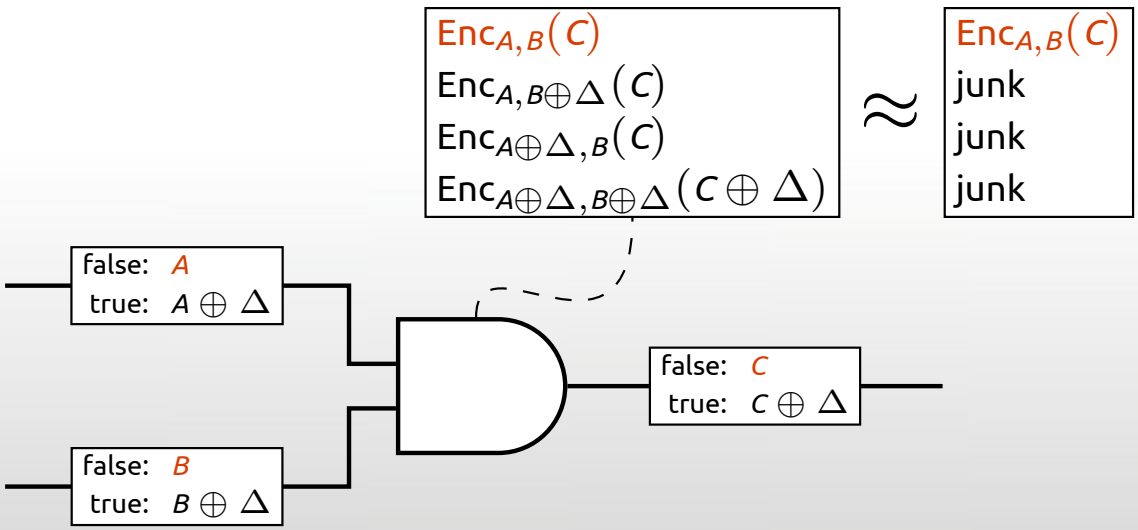
why does free-XOR require circularity assumption?



why does free-XOR require circularity assumption?

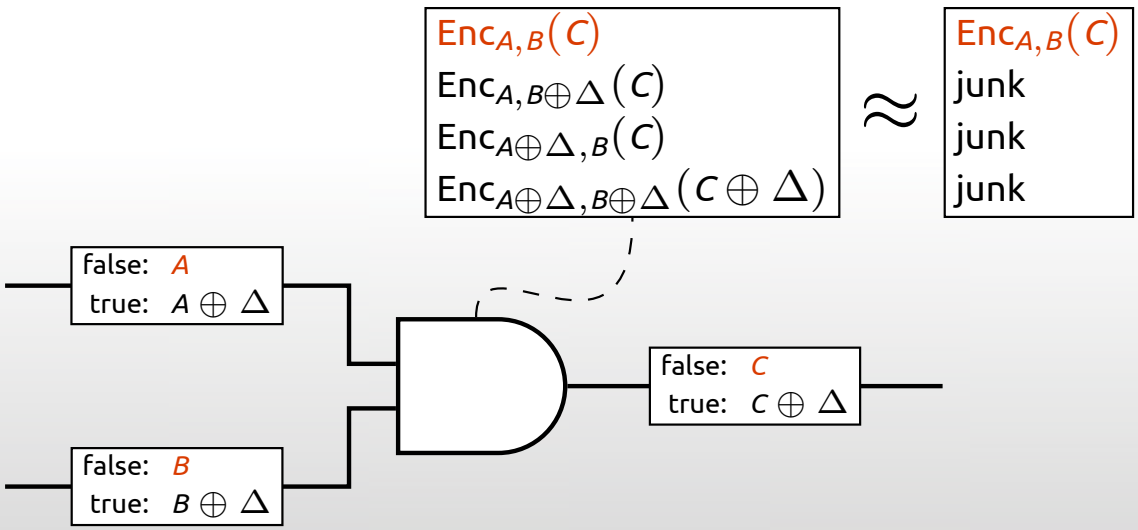


why does free-XOR require circularity assumption?



$\mathsf{Enc}_{A\oplus\Delta,B\oplus\Delta}(C\oplus\Delta) \approx$ junk,

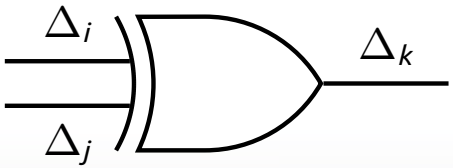
why does free-XOR require circularity assumption?

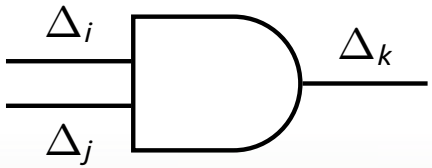


$\mathsf{Enc}_{A\oplus\Delta,B\oplus\Delta}(C\oplus\Delta) \approx$ junk,

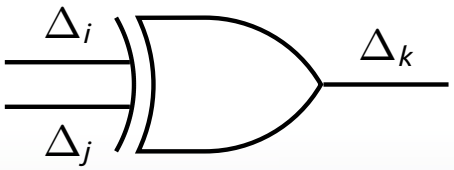
- **Key cycle:** same secret Δ in key and message!

circularity in FleXOR?

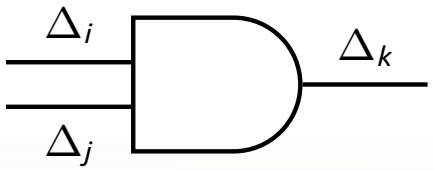




circularity in FleXOR?

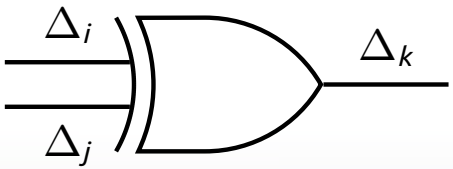


- Δ_k appears in message

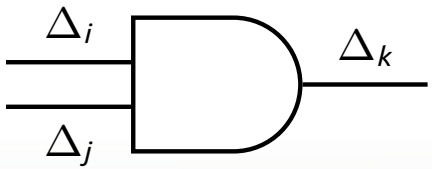


- Δ_k appears in message

circularity in FleXOR?

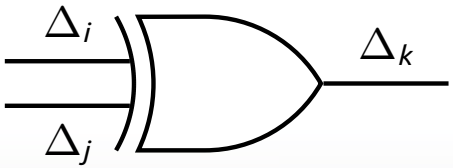


- Δ_k appears in message
- Δ_i, Δ_j appear in keys

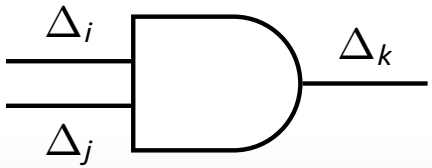


- Δ_k appears in message
- Δ_i, Δ_j appear in keys

circularity in FleXOR?



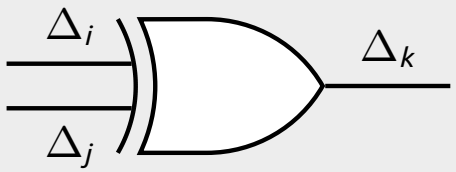
- Δ_k appears in message
- Δ_i, Δ_j appear in keys
- only when $i \neq k, j \neq k$



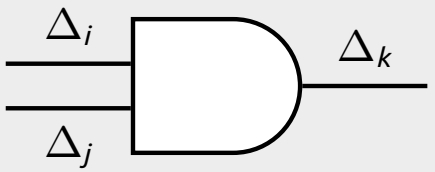
- Δ_k appears in message
- Δ_i, Δ_j appear in keys

removing circularity in fleXOR

Definition: monotone wire ordering



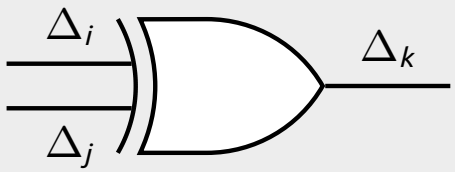
$\Rightarrow k \geq \max\{i, j\}$



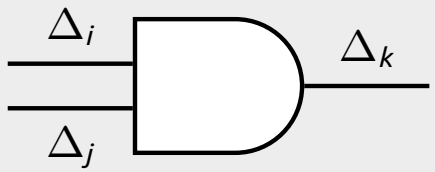
$\Rightarrow k > \max\{i, j\}$

removing circularity in fleXOR

Definition: monotone wire ordering



$\Rightarrow k \geq \max\{i, j\}$



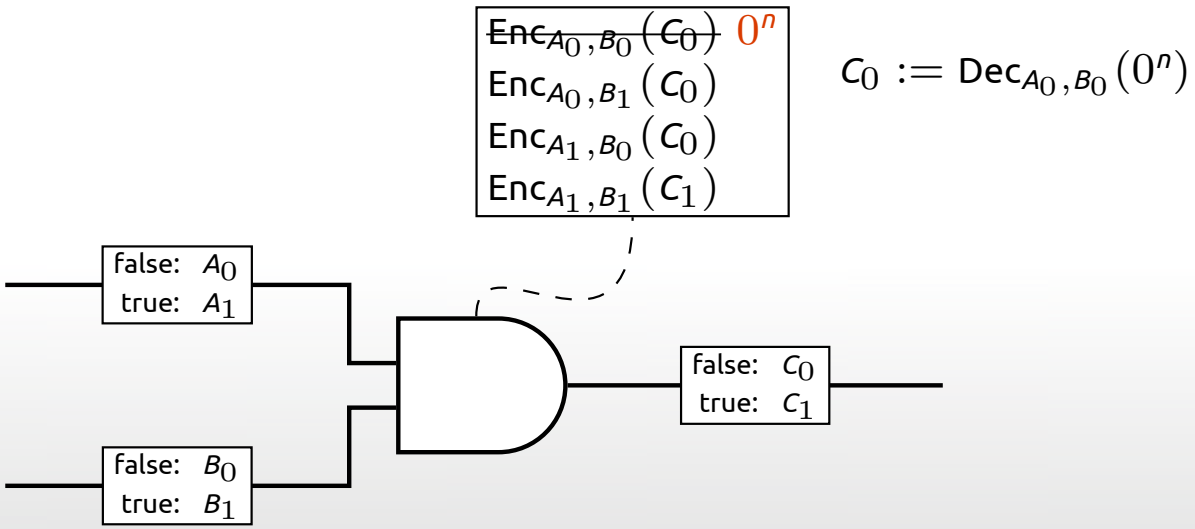
$\Rightarrow k > \max\{i, j\}$

Results

1. Can garble using FleXOR without circularity assumption, when offsets chosen via **monotone** wire ordering
   - Same assumption required for OT-extension [Ishai+03]
2. NP-hard to find optimal monotone wire ordering
3. We suggest heuristics that seem to find good monotone orderings

row-reduction compatibility

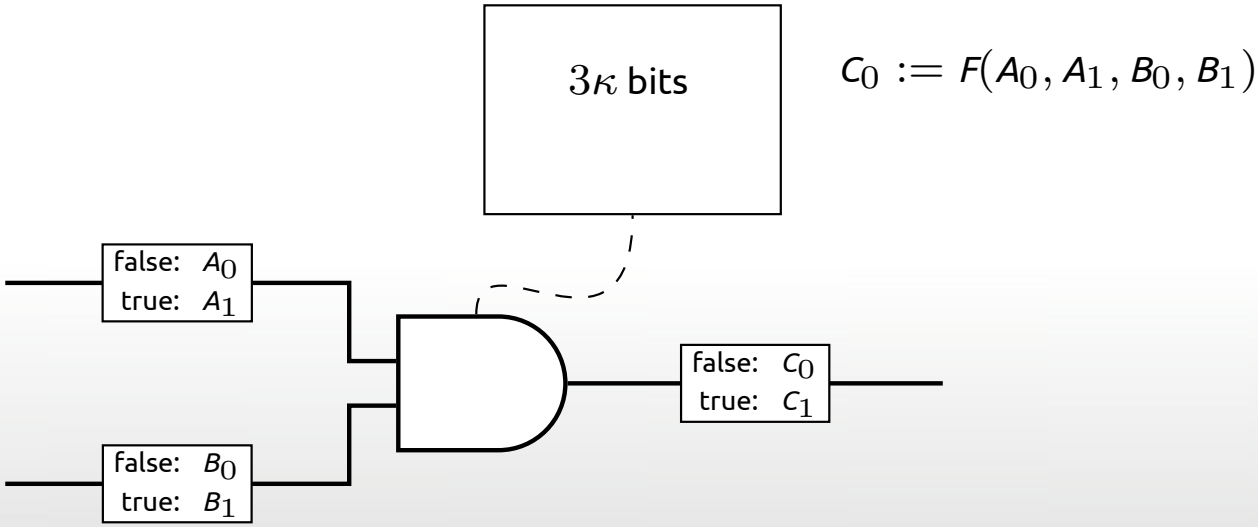
why is free-XOR incompatible with $4 \longrightarrow 2$-row-reduction?



Row reductions

- $4 \rightarrow 3$ reduction

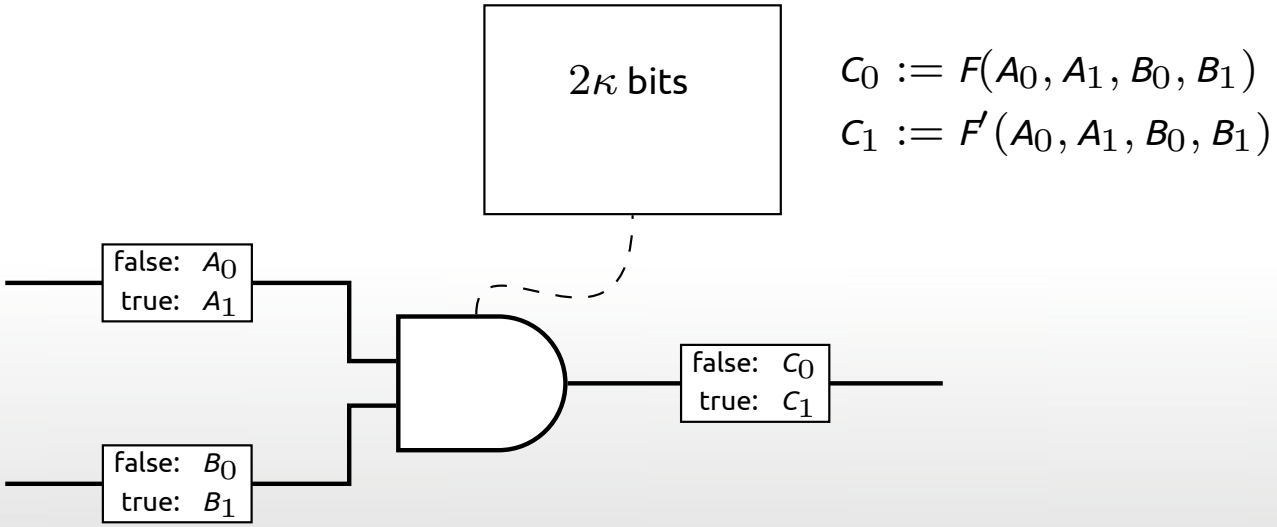
why is free-XOR incompatible with $4 \longrightarrow 2$-row-reduction?



Row reductions

- $4 \rightarrow 3$ reduction: C_0 set implicitly

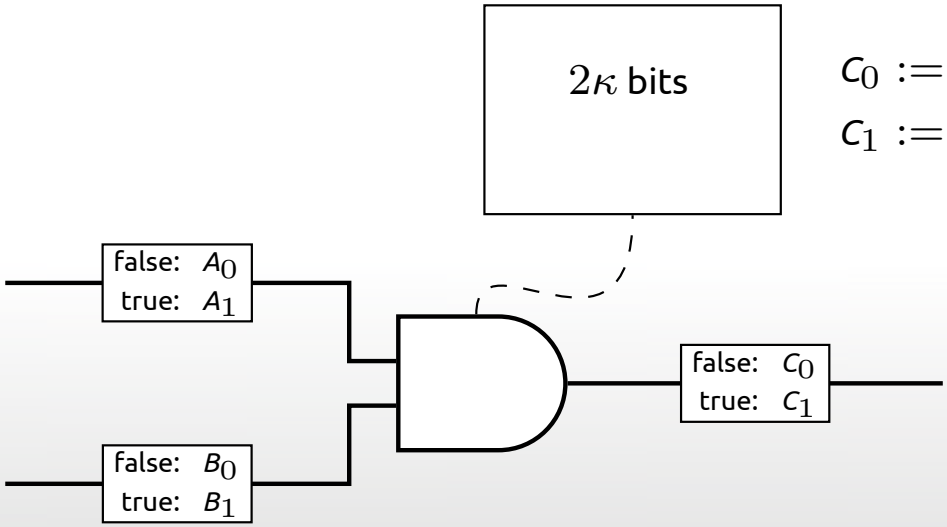
why is free-XOR incompatible with $4 \longrightarrow 2$-row-reduction?



Row reductions

- $4 \to 3$ reduction: C_0 set implicitly
- $4 \to 2$ reduction: both C_0, C_1 set implicitly

why is free-XOR incompatible with $4 \longrightarrow 2$-row-reduction?



$$C_0 := F(A_0, A_1, B_0, B_1)$$
$$C_1 := F'(A_0, A_1, B_0, B_1)$$

Row reductions

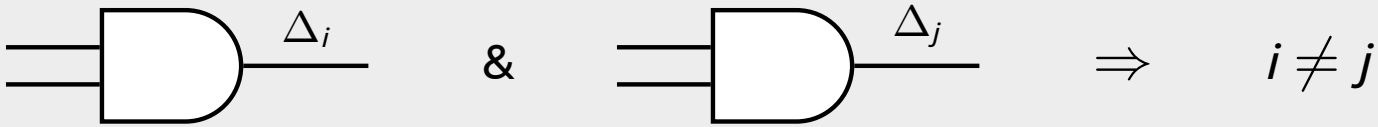
- $4 \to 3$ reduction: C_0 set implicitly
- $4 \to 2$ reduction: both C_0, C_1 set implicitly

$\Rightarrow$ can't ensure offset $C_0 \oplus C_1$ equals global offset Δ!

compatibility with fleXOR

$4 \rightarrow 2$ row reduction would set output wire's offset implicitly.

Definition: safe wire ordering



compatibility with fleXOR

$4 \to 2$ row reduction would set output wire's offset implicitly.

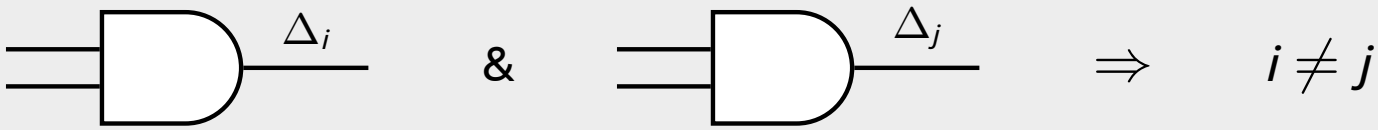
Definition: safe wire ordering

... plus some fine print

compatibility with fleXOR

$4 \to 2$ row reduction would set output wire's offset implicitly.

Definition: safe wire ordering



... plus some fine print

Results

1. Can garble using FleXOR + $4 \to 2$ row reduction, when offsets chosen via **safe** wire ordering
   - XOR gates cost 0, 1, or 2; other gates cost 2
2. We suggest a simple heuristic that finds good safe orderings

summary

FleXOR = **Fle**xible **XOR**!

- New way to garble XOR gates: costs 0, 1, or 2 ciphertexts per gate
- Get results competitive with Free-XOR, from weaker assumption
- Get results often better than Free-XOR, by leveraging $4 \to 2$ row-reduction

Garbled circuit size (ciphertexts per gate)

scheme	assump	AES	DES	SHA1	SHA2	HamDst	IntMult
classical	**OWF**	2.00	2.00	2.00	2.00	2.00	2.00
Free XOR	circular	**0.64**	2.79	1.82	2.05	**0.50**	**0.90**
FleXOR	related-key	0.76	2.84	2.02	2.26	0.67	1.15
FleXOR	circular	0.72	**1.89**	**1.39**	**1.56**	**0.50**	0.94


The End