For each wire in the circuit, associate a pair of keys:

\[
\begin{array}{c|c|c}
\Pi_1 & k_1^* & k_1^* \\
\Pi_2 & k_2^* & k_2^* \\
\Pi_3 & k_3^* & k_3^* \\
\end{array}
\]

Label of key \( k_i^b \) is \( b \oplus \Pi_i \).

Garbled table:

<table>
<thead>
<tr>
<th></th>
<th>( \text{Enc}<em>{k_1^b}(\text{Enc}</em>{k_2^b}(b_3 \oplus \Pi_3, k_3^{b_3 \oplus \Pi_3})) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

where \( b_3 = G(\Pi_1, \Pi_2) \)

Optimization:

Free-XOR technique (KS '08)

1. Pick global random value \( R \)
2. For every wire:
   - Choose \( k_i^* \) uniform
   - \( k_i^* = k_i^b \oplus R \)
Now evaluation can compare:

\[ k_3^o = k_1^o \oplus k_2^o \]

\[ k_3^o = (k_1^o \oplus j_1 \cdot R) \oplus (k_2^o \oplus j_2 \cdot R) \]

\[ k_3^o = k_1^o \oplus k_2^o \oplus (j_1 \oplus j_2) \cdot R \]

[POSSW '09] - Reduction circuits to minimize non-XOR gates

- prove security in random oracle (RO) model
- or more complicated assumption
**Gambled-row reduction (GRR)**

\[ \Pi_1 \quad k_1^* \quad k_1^* \quad \Pi_2 \quad F: \text{pseudorandom function} \]

\[ \begin{array}{c}
k_3^* G (\Pi_1, \Pi_2) \oplus G (0, 0) = F_{k_1^*} (0) \oplus F_{k_2^*} (0) \\
 k_3^* G (\Pi_1, \Pi_2) \equiv k_3^* (\Pi_1, \Pi_2) \oplus R
\end{array} \]

This technique will cut down computations by 25%.

Further optimization:

Let \( P \) be degree-2 polynomial interpolating \((a, K_a), (b, K_b), (c, K_c)\) define corresponding key on output wire as \( P(a) \) include \( P(s), P(b) \) in gambled table.

Let \( Q \) be degree-2 poly interpolating \((5, P(5)), (6, P(6)) \) and \((d, k_d)\) define corresponding output-wire key to be \( Q(d) \).
1) $\text{Enc}_{k_1}(\text{Enc}_{k_2}(k_3))$

2) $H(k_1, k_2) \oplus GID \oplus k_3$

3) $F_{k_1}(GID \mid 00) \oplus F_{k_2}(GID \mid 00) \oplus k_3$

Bellare et al., '13
- use AES with hardcoded key $k$
- model $AES_k(\cdot)$ as an ideal permutation

$AES_k(S) \oplus S \oplus k_3$
where $S = 2k_1 \oplus 4k_2 \oplus \text{GID}$

$\Rightarrow 23$ cycles/gate for evaluation
$56$ cycles/gate for garbling
$0.32$ ns/cycle

(several orders of magnitude better)