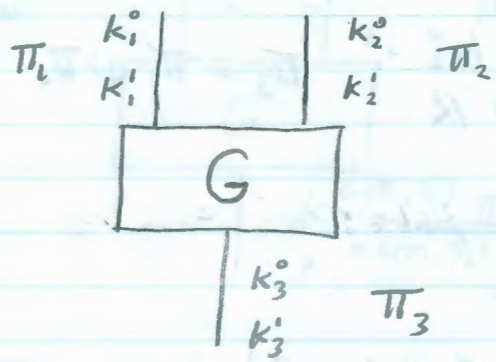
09/23/2013

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



Label of key k_i^b is $b \oplus \pi$;

Garbled table:

0	0	$\text{Enc}_{k_1^{\pi_1}}\left(\text{Enc}_{k_2^{\pi_2}}\left(b_3 \oplus \pi_3, k_3^{b_3 \oplus \pi_3}\right)\right)$
0	1	⋮
1	0	⋮
1	1	⋮

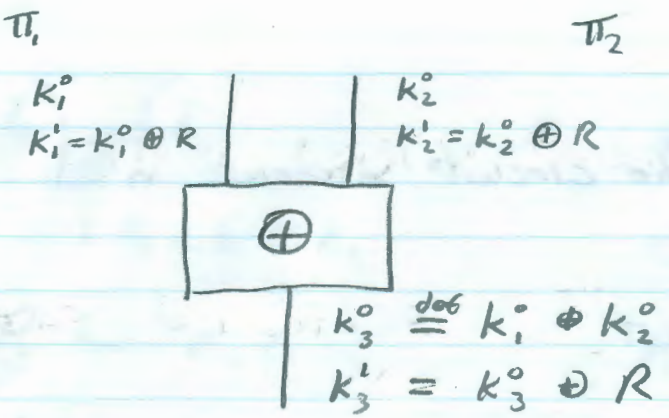
where $b_3 = G(\pi_1, \pi_2)$

Optimization:

Free-XOR technique (KS '08)

1) pick global random value R
for every wire:
choose k_i^0 uniform
$k_i^1 = k_i^0 \oplus R$

2)



$\Pi_3 = \Pi_1 \oplus \Pi_2$

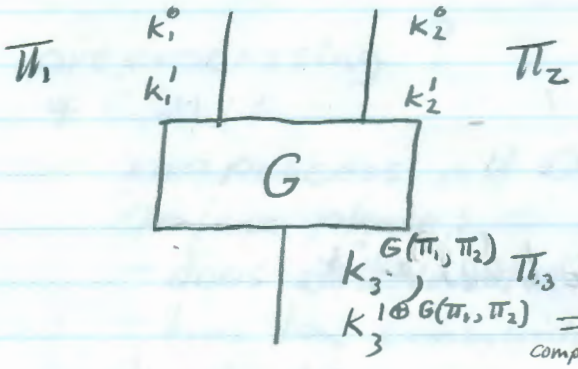
Now evaluator can compare:

$$K_3^{j} = k_1^{j_1} \oplus k_2^{j_2}$$
$$= (k_1^0 \oplus j_1 \cdot R) \oplus (k_2^0 \oplus j_2 \cdot R)$$
$$= k_1^0 \oplus k_2^0 \oplus (j_1 \oplus j_2) \cdot R$$
$$= k_3^{j_1 \oplus j_2}$$

[PSSW '09] – Reduction circuits to minimize num. XOR gates

- prove security in random oracle (RO) model
- or more complicated assumption

Garbled-row reduction (GRR)



F: pseudorandom function

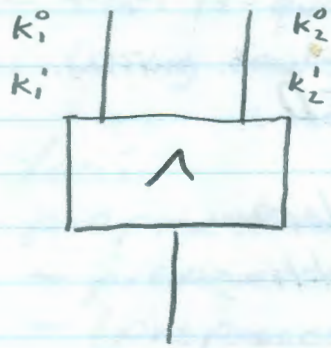
$$k_3^{G(\pi_1,\pi_2)}, \pi_3 \oplus G(0,0) = F_{k_1}^{\pi_1}(0) \oplus F_{k_2}^{\pi_2}(0)$$

$$k_3^{1 \oplus G(\pi_1,\pi_2)} = k_3^{G(\pi_1,\pi_2)} \oplus R$$

comparable

- This technique will cut down computations by 25%.

Further optimization:



$K_{b_1,b_2} = H(k_1^{b_1}, k_2^{b_2}, G_{id}) = F_{k_1}^{b_1}(G_{id}) \oplus F_{k_2}^{b_2}(G_{id})$

gate identifier

get values (a, K_a)
(b, K_b)
(c, K_c)

should all allow evaluator to recover one key.
(d, K_d)
should allow recovery of the other key

Let P be degree-2 polynomial interpolating (a, K_a), (b, K_b), (c, K_c) define corresponding key on output wire as $P(0)$ include $P(s)$, $P(G)$ in garbled table.

Let Q be degree-2 poly interpolating $(s, P(s))$, $(G, P(G))$ and (d, K_d) define corresponding output-wire key to be $Q(0)$

1) $Enc_{k_1}(Enc_{k_2}(k_3))$

2) $H(k_1 \| k_2) G_{id}) \oplus k_3$

3) $F_{k_1}(G_{id} \| 00) \oplus F_{k_2}(G_{id} \| 00) \oplus k_3$

Bellare et al. '13
- use AES w/o hard coded key k
- model $AES_k(\cdot)$ as an ideal permutation

$AES_k(s) \oplus s \oplus k_3$
where $s = 2k_1 \oplus 4k_2 \oplus GID$

⇒ 23 cycles / gate for evaluation
56 cycles / gate for garbling
0.32 ns / cycle

(several orders of magnitude better)