Better Concrete Security for Half-Gates Garbling (in the Multi-Instance Setting)

Chun Guo, Jonathan Katz, Xiao Wang, Chenkai Weng, Yu Yu

All widely used GCs have a birthday-bound security

- GC based on fix-key block cipher -> $O(tC/2^n)$

Explicit attack

267 machine-month to break a GC with 80-bit labels, ~~ 3500$

- Those based on standard PRFs: C hybrids in the proof, each with a PRF game -> $O(tC/2^n)$

No proof with optimal security (but also no attack)

- Exceptions: some RO based protocols

Slow

Attack in the multi-instance setting

- An adversary, with n garbled circuits (each garbled **independently**), can break one of them with probability ~$tC/2^n$
  - t: running time
  - C: **sum of all circuit sizes**
- In means that switching free-XOR Delta does NOT help!

Our New Abstraction for better security

- A weaker version of Tweakable correlation robust hash
  - Tweakable, but there is a explicit bound how frequently each tweak will be used.
  - Bound = 2 for Garbling and OT extension.
- Hash function H is secure if $F_k(x, i) = H(k \oplus x, i)$ is a pseudorandom function with a ***bounded-query*** adversary.

Construction

- $TMMO(x, i) = E_i(\sigma(x)) \oplus \sigma(x)$
  - Friendly to batch
  - $\sigma(x)$ is orthomorphism if $\sigma(x)$ and $\sigma(x) \oplus x$ are all permutations

- Proven secure if E is an ideal cipher
  - Adv's advantage is bounded by $O(u(p+q)/2^n)$, where u is maximum number of oracle calls for any tweak

Practical performance

Hash function	NI support?	k	Comp. sec. (bits)	100 Mbps	2 Gbps	localhost
Zahur et al.	Y	128	89	0.4	7.8	23
SHA-3	N	128	125	0.27	0.27	0.28
SHA-256	N	128	125	0.4	1.1	1.2
SHA-256	Y	128	125	0.4	2.1	2.45
$\widehat{\text{MMO}}^{E}$	Y	128	125	0.4	7.8	15
$\widehat{\text{MMO}}^{E}$	Y	88	86	0.63	12	15

Improved to 24 since then

Table 1: **Performance of different hash functions in the half-gates scheme.** All reported numbers are in 10^6 AND gates per second. "NI support" indicates whether the implementation utilizes hardware-level instructions (i.e., AES-NI or SHA-NI), and "comp. sec." refers to the computational security bound assuming $C < 2^{40}$. The length of the wire labels is k.

Implementation suggestion

- Always use TMMO regardless of semi-honest or malicious security
- Always randomize the start point of the tweak
- Code?