AICR · OPEN PROBLEM

An Efficiently Simulatable Parity-Tolerant Circuit

t parities of the wires, simulated from t parities of the input and output, in time $\mathrm{poly}(t)$

Status. Statement: AI-written from Yuval Ishai and Yifan Song, *Leakage-Tolerant Circuits*, IACR ePrint 2024/332. Not yet checked by a human. Feasibility is settled: every circuit can be compiled into a t-parity-tolerant one. What is open is the running time of the simulator, which in the source's construction is $2^{O(t)}$; the source gives partial evidence, under LPN, that inefficient simulation may be inherent for a related relaxed notion.

Category. *Side-Channel Countermeasures.*

Conjecture (informal). *A leakage-*resilient *circuit protects a computation by encoding its input and output, so that leakage on the internal wires reveals nothing. A leakage-*tolerant *circuit has no such luxury: the input and output sit in the clear, so leakage on the wires necessarily reveals something, and the requirement instead is that whatever it reveals could have been obtained by leaking on the input and output alone. The source shows every circuit can be made tolerant against t parities of its wires. But its simulator — the algorithm translating a wire-leakage query into an input/output-leakage query — runs in time $2^{O(t)}$, which makes the guarantee vacuous for the security-parameter-sized t that applications want. The source asks whether a t-parity-tolerant circuit with a $\mathrm{poly}(t)$-time simulator exists, and reports what stands in the way: its simulator has to find a short vector in a linear code determined by the parity queries.*

1 The setting

Leakage-resilient circuits, following Ishai, Sahai and Wagner [ISW03] and, for global leakage classes, Goyal et al. [GIM+16], come as a triple (I, C, O): a randomized input encoder, a randomized circuit on encoded values, and an output decoder. Because x and y are protected by the encoding, one can ask for the strong guarantee

that leakage on C's wires be statistically independent of x. Leakage *tolerance*, generalizing t-probing tolerance [ISW03, AIS18, GIS22], drops the encoders: C maps a cleartext input to a cleartext output, so leakage on the wires can always be answered by leakage on (x, y) — and the definition demands exactly that, via a simulator.

The source's main positive result is that every circuit C_f can be compiled into an $\mathcal{L}$-tolerant circuit for the class $\mathcal{L}$ of leakage functions outputting t parities, or t disjunctions (equivalently conjunctions) of any number of wires or their negations. Concretely, its Corollary 2 gives, for every $f : \{0,1\}^n \to \{0,1\}^m$ with a circuit of size s and depth h, a t-parity-tolerant circuit of size $\widetilde{O}(s) + \mathrm{poly}(n, m, h, t, \kappa)$, with error negligible in the statistical security parameter κ.

For the parity class, however, the source's simulator runs in time exponential in t. The reason is visible in its analysis: to answer t wire-parity queries $W_1, \ldots, W_t$ jointly, it is not enough to handle each separately, because a subset-XOR $\bigoplus_{i \in S} W_i$ may touch few enough small-bias encodings to be informative even when each W_i individually does not. The simulator therefore computes $\bigcup_{S \subseteq [t]} V(\bigoplus_{i \in S} W_i)$, a union over all 2^t subsets — equivalently, as the source's own summary puts it, it must find a short vector in the linear code the queries span. The resulting set is small (the source bounds $|V| \le kt$); it is finding it that costs $2^{O(t)}$.

The source is explicit that it does not know whether this is necessary. In its "Future directions" it asks: "Is there a t-parity-tolerant circuit with a $\mathrm{poly}(t)$-time simulator? Our simulator needs to find a short vector in a linear code defined by the parity queries, and we are only able to show that this is inherent for a related encoding problem." The related problem is the relaxed notion of (t, t')-parity-tolerant *functions* it studies in its Appendix H, where any t parities of the function's output must be simulatable from t' parities of its input; there it exhibits, under standard LPN, an instance requiring super-polynomial simulation time. That is evidence and not a proof for circuits, and the source says so.

The question is not idle bookkeeping. The source's own application — compiling leakage-tolerant circuits into *stateful* leakage-resilient circuits that withstand continuous parity leakage — gets a $\mathrm{poly}(t)$-time simulator by a trick specific to the stateful setting (a control bit in the initial state that lets the simulator substitute

an encoding of the public output), and works "regardless of the efficiency of the LTC simulator". So the stateful application is already saved; the open question is about the stateless primitive itself, where efficiency of simulation is what decides whether $t = \Omega(\lambda)$ is meaningful.

2 Notation and parameters

- $f : \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ is the (possibly randomized) function being implemented; C is a randomized circuit computing it.
- $\tau(C,x)$ denotes the vector of wire values of C on input x, a random variable over C's randomness; $|C|$ is its number of wires.
- $\mathcal{L}$ is a class of leakage functions, assumed closed under restrictions. $(\mathcal{L}')^{\otimes t}$ is the t-fold class: all $L(x) = (L'_1(x), \ldots, L'_t(x))$ with $L'_j \in \mathcal{L}'$.
- The *parity* class contains, for each $S \subseteq [n]$, the function $L(x) = \bigoplus_{i \in S} x_i$; the t-parity class is its t-fold version.
- ε is the simulation error in statistical distance; t is the number of leakage queries; κ is a statistical security parameter, and ε negligible in κ is what the source's compiler achieves (at $k = O(t(t+\kappa))$ probes in its intermediate parity-to-probing step).

3 The conjecture

Definition 1 (Leakage-tolerant circuit, [IS24, Definition 2]). For a possibly randomized $f : \{0,1\}^{n_i} \to \{0,1\}^{n_o}$, a randomized circuit C mapping $x \in \{0,1\}^{n_i}$ to $y \in \{0,1\}^{n_o}$ is an $(\mathcal{L},\varepsilon)$-*leakage-tolerant implementation* of f if

- (correctness) for every x, the distributions $f(x)$ and $C(x)$ are identical; and
- (leakage tolerance) there is a simulator $\mathsf{Sim} = (\mathsf{Sim}_1, \mathsf{Sim}_2)$, where Sim_1 takes $L \in \mathcal{L}$ of input size $|C|$ and outputs a state st together with $L' \in \mathcal{L}$ of input size $n_i + n_o$, and Sim_2 takes st and the value of L' on the input and output of f, such that for every $x \in \{0,1\}^{n_i}$

and every $L \in \mathcal{L}$,

$$\big(L(\tau(C,x)),\, C(x)\big) \;\approx_{\varepsilon}\; \big(\mathsf{Sim}_2(st, L'(x,y)),\, y\big),$$

where $y \leftarrow f(x)$ and $(st, L') \leftarrow \mathsf{Sim}_1(L)$.
We write (t,ε)-*parity tolerance* for $(\mathcal{L},\varepsilon)$-leakage tolerance where $\mathcal{L}$ is the t-parity class.

Conjecture 1 (An efficiently simulatable parity-tolerant circuit). *There is a polynomial* $\mathrm{poly}(\cdot)$ *and a compiler that, given any Boolean circuit* C_f *for* $f : \{0,1\}^{n_i} \to \{0,1\}^{n_o}$ *and any* t *and any statistical security parameter* κ*, outputs a randomized circuit* C *that is* (t,ε)*-parity-tolerant (Definition 1) with* ε *negligible in* κ *and of size polynomial in* $|C_f|, t$ *and* κ*, together with a simulator* $(\mathsf{Sim}_1, \mathsf{Sim}_2)$ *for it whose running time is* $\mathrm{poly}(|C_f|, t, \kappa)$ *— in particular polynomial, not exponential, in* t*.*

Remark 1 (Both directions are open). The source poses this as a question, and either answer resolves it: a construction with a $\mathrm{poly}(t)$-time simulator, or a proof that no t-parity-tolerant circuit admits one. For the negative direction, the source's Appendix H already supplies the shape of an argument — a (t,t')-parity-tolerant *function* whose simulation is super-polynomial under LPN — and what is missing is the step from that relaxed notion to circuits. A conditional negative answer (under LPN or another standard assumption) would count.

Remark 2 (What may not be changed to make it easier). Three parts of the statement are load-bearing and are the source's, not this statement's choices. First, the leakage class is t *parities* of arbitrarily many wires, not t probes: for probing leakage, efficient simulation is classical [ISW03]. Second, tolerance, not resilience: with an input encoder and output decoder the source can already do better, and its own compiler to stateful leakage-*resilient* circuits achieves $\mathrm{poly}(t)$-time simulation whatever the tolerant circuit's simulator costs. Third, the simulator must be efficient in t for a single stateless evaluation; efficiency amortized over a stateful sequence of invocations is the case the source has settled.

Remark 3 (The neighbouring question about leakage classes).

The source's other stated open direction — obtaining leakage-tolerant circuits for classes beyond depth-1 AC^0 and parity, in particular for functions returning the Hamming weight of a subset of wires, or for depth-2 AC^0 — is a separate statement about which classes are achievable at all, not about the cost of simulation within the parity class, and is not covered here.

4 Bibliography

[IS24] Yuval Ishai and Yifan Song. *Leakage-tolerant circuits.* IACR ePrint 2024/332. The source. Definition 2 is on page 14; the open question is in the "Future directions" list, page 3; the LPN-based evidence is Appendix H, discussed on page 13.

[ISW03] Yuval Ishai, Amit Sahai and David Wagner. *Private circuits: securing hardware against probing attacks.* CRYPTO 2003. The probing model and the additive-sharing compiler the source's counterexample is stated against.

[GIM+16] Vipul Goyal, Yuval Ishai, Hemanta K. Maji, Amit Sahai and Alexander A. Sherstov. *Bounded-communication leakage resilience via parity-resilient circuits.* 57th Annual Symposium on Foundations of Computer Science (FOCS), 2016, pages 1–10. The leakage-resilient-circuit definition the source borrows, and the parity-resilient construction its parity-to-probing analysis builds on.

[AIS18] Prabhanjan Ananth, Yuval Ishai and Amit Sahai. *Private circuits: a modular approach.* CRYPTO 2018. One of the probing-tolerance notions the source generalizes.

[GIS22] Vipul Goyal, Yuval Ishai and Yifan Song. *Private circuits with quasilinear randomness.* EUROCRYPT 2022, pages 192–221. The other probing-tolerance notion the source generalizes.

[GR15] Shafi Goldwasser and Guy N. Rothblum. *How to compute in the presence of leakage.* SIAM Journal on Computing, 2015. Implies the stateful leakage-resilient circuits the source's application reproves, as its own erratum notes.