Tamper resilient circuits: The Adversary at the Gates

Yiannis Tselekounis
Joint work with Aggelos Kiayias

University of Athens

Asiacrypt 2013 - December 4, 2013

Attacking a cryptographic implementation

Adversary

x

$C_{\mathbf{s}}(x)$

Cryptographic device containing circuit C with private memory $\mathbf{s}$

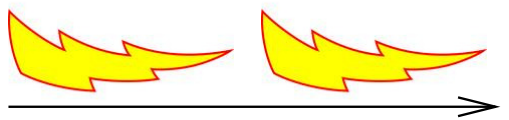
The adversary, having black-box access to $C_{\mathbf{s}}$, repeatedly supplies it with input x of her choice and receives $C_{\mathbf{s}}(x)$.
In reality though, the adversary can be much more inventive.

Introduction

Real world attacks

Tampering Adversary

x



$C'_{\mathbf{s}}(x)$

Cryptographic device containing circuit C with private memory $\mathbf{s}$

Physical *active* attacks against the implementation:

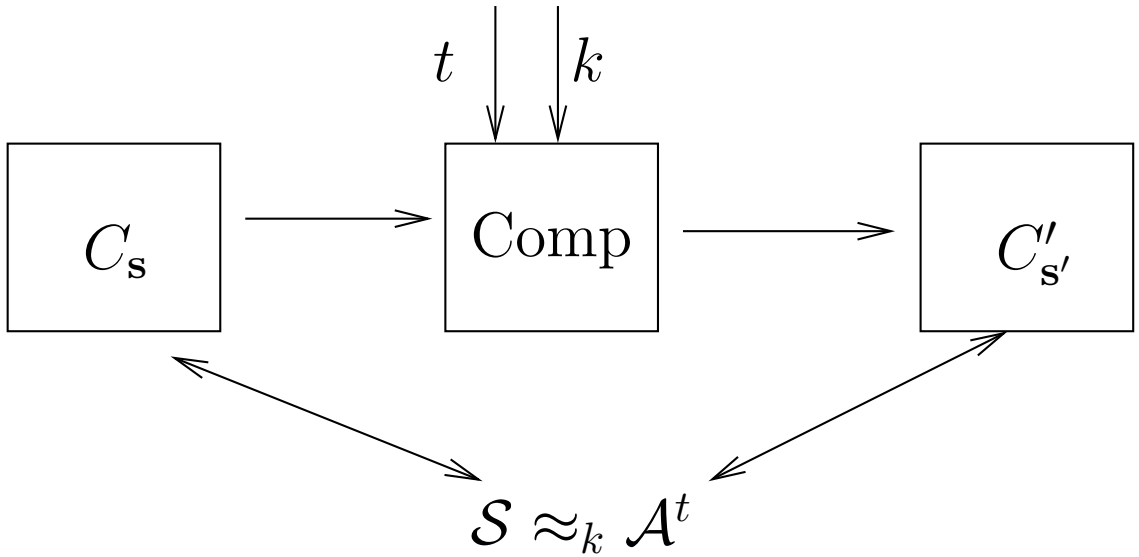
- Inducing faults to the computation [BS97], [BDL97].
- Exposing the device to electromagnetic radiation [GMO01], and others.

Defending against tampering attacks

1. Build circuits using tamper-resilient hardware:
   - Might be quite expensive solution,
   - Might be secure only against known attacks.
2. Employ algorithmic techniques for protecting against tampering attacks, i.e., modify the circuit so that it is resilient:
   - It provides security against unknown attacks.
   - Currently, there is a gap between theoretical modeling and real-world attacks.

This work focuses on algorithmic techniques.

Security against tampering adversaries



1. k: security parameter, t: number of circuit components.
2. Both circuits implement the same functionality.
3. $\mathcal{S}$ is having black-box access to $C_{\mathbf{s}}$,
4. $\mathcal{A}^t$ performs tampered computations on $C'_{\mathbf{s}'}$,
5. The view of the adversary is simulated by $\mathcal{S}$.

Related work & Motivation

- There are 3 constructions which are provably secure against tampering attacks on circuit wires: [IPSW06], [FPV11], [DSK12].
- All of them employ tamper-proof gates (the last two even non-standard gates).
- [SA03]: attacks against circuit transistors.

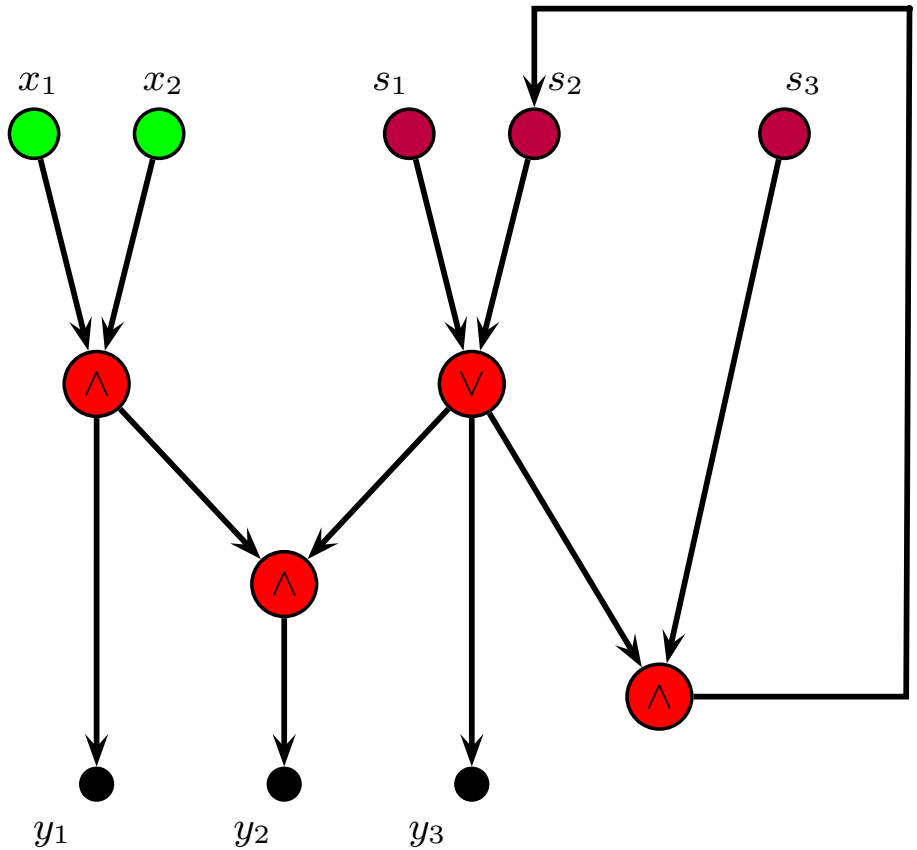
What happens if the adversary tampers with circuit gates?

Our contribution

- A new adversarial model: the attacker against circuit gates.
- An impossibility result on tamper resilience under plausible assumptions w.r.t. both wire and gate attackers.
- Gate adversaries subsume wire adversaries. We prove that gate adversaries are strictly stronger than wire ones.
- We show how to defend against gate adversaries. We state and prove a general theorem about circuit compilers which has as a corollary that the third compiler of [IPSW06] is resilient against gate attacks.

Theoretical Model

Circuit $C_{\mathbf{s}}$: A directed graph $G(V, E)$.
Each $v \in V$ (resp. $e \in E$) represents a circuit gate (resp. wire).
Input gates: x_1, x_2, **output gates**: y_1, y_2, y_3,
private memory gates: s_1, s_2, s_3, and **boolean gates**.

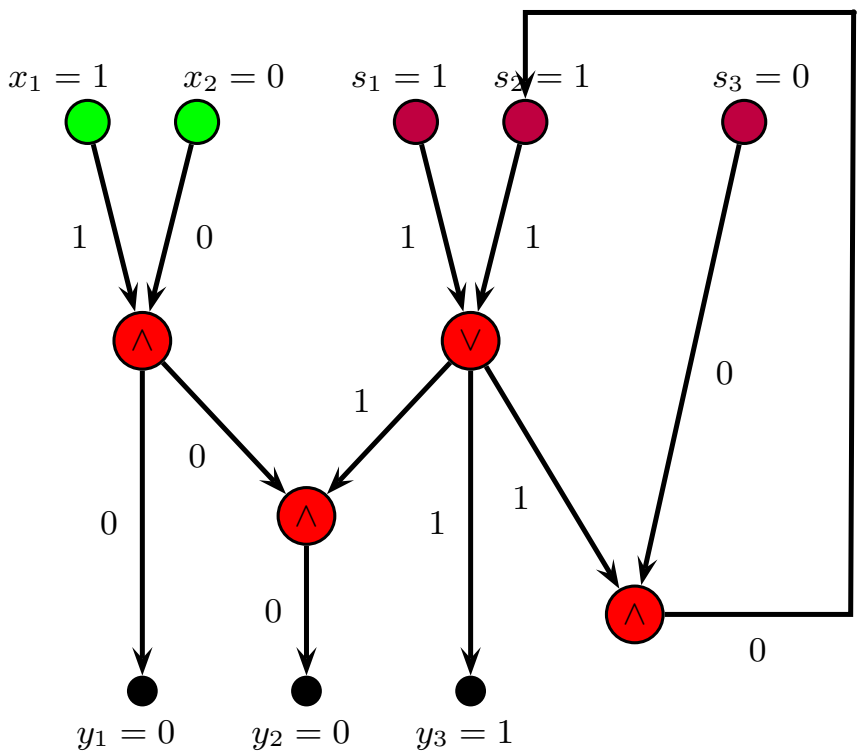


A single round circuit computation is a BFS traversal on G.

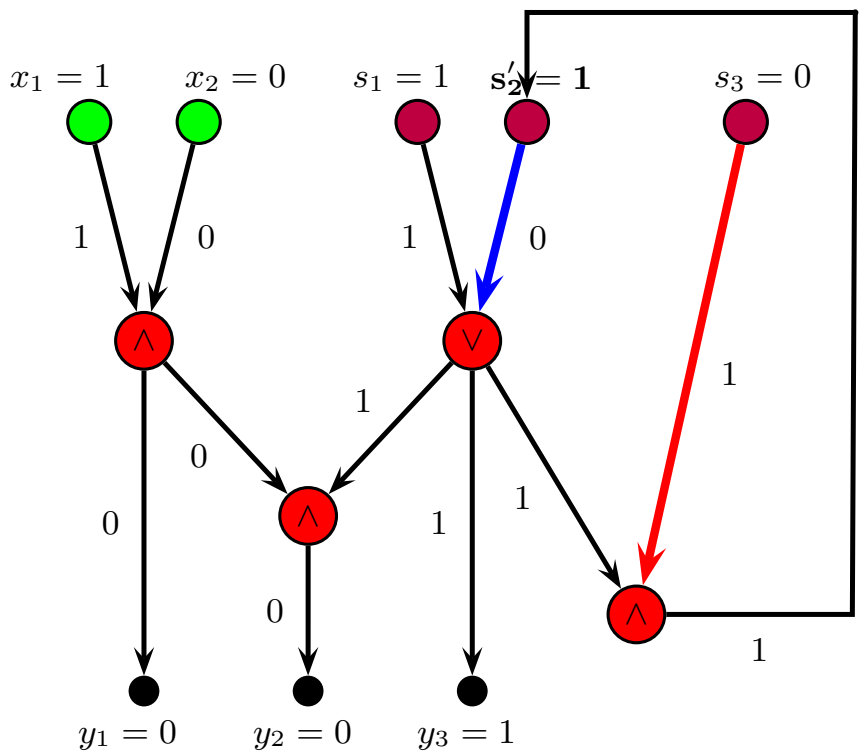
Adversarial models

Previous models: Choose $E' \subseteq E$ and/or a subset of memory gates V', and for each $a \in E' \cup V'$: ***toggle*** **it,** ***reset*** it to 0, ***set*** it to 1. The attacks may be permanent. (Example: reset to 0, toggle)

Original computation



Tampered computation



Gate attacker

Choose a subset of circuit gates $V' \subseteq V$*, and for each* $g \in V'$*, substitute* g *with some* g'*, where* $\mathsf{arity}(g) = \mathsf{arity}(g')$*. For binary fan-in there are* 16 *functions from* $\{0,1\}^2 \to \{0,1\}$.

Impossibility

Theorem (informally)

Security is unachievable if we allow an adversary to tamper with $(k-1)d$ *circuit wires or* d *gates, where* d *denotes the circuit depth and* k *is the circuit's fan-in.*

Any compiler that receives $C_{\mathbf{s}}$, t, k, and produces circuit $C'_{\mathbf{s}'}$ of depth no greater than t, is insecure regardless of its size.

Impossibility (proof sketch)

1. **Non-triviality (assumption)**: For every circuit $C_{\mathbf{s}}$ and every PPT adversary $\mathcal{A}$ there exists non-negligible $f(m)$, $m = |\mathbf{s}|$, s.t.

$$\Pr[\mathcal{A}^{C_{\mathbf{s}}(\cdot)}(\cdot) = \mathbf{s}] < 1 - f(m).$$

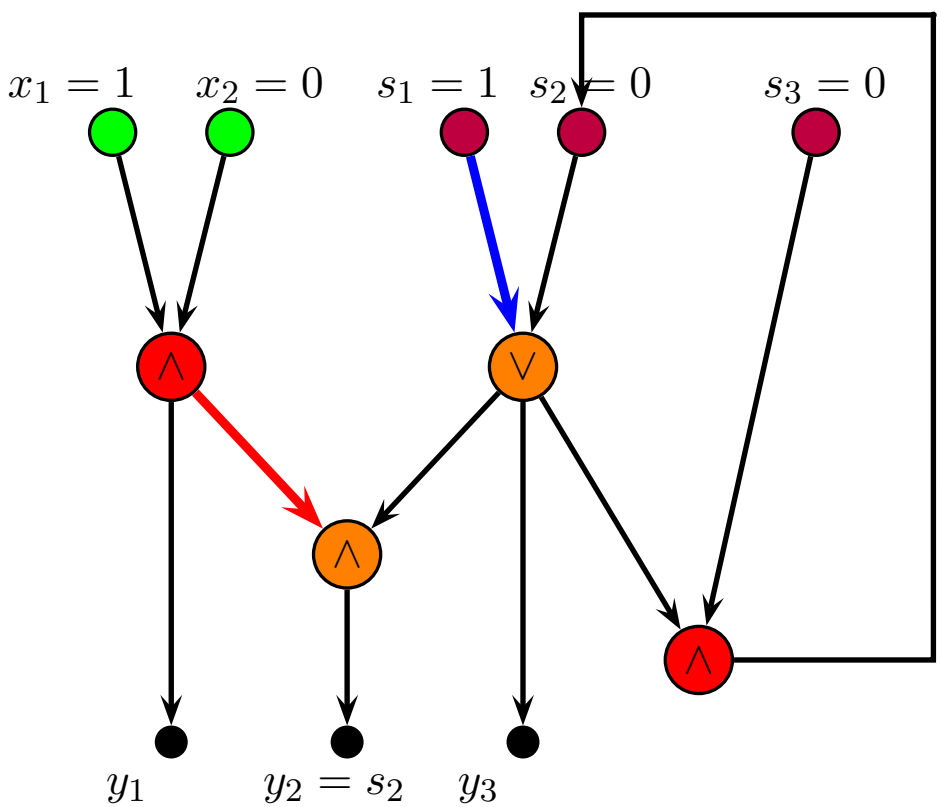
2. **Weakly unpredictable bit**: We prove that for every non-trivial circuit there exists an index i, $1 \leq i \leq m$, s.t. for every $\mathcal{A}$ there exists a non-negligible function $\delta(m)$ such that

$$\Pr[\mathcal{A}^{C_{\mathbf{s}}(\cdot)}(\cdot) = s_i] < 1 - \delta(m).$$

3. We define a tampering adversary with tampering ability up to the depth of the circiut who learns the **weakly unpredictable** bit with probability equal to 1.
4. We prove that this adversary is unsimulatable.

Impossibility (proof sketch)

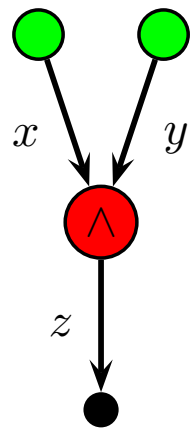
Let s_2 be the *weakly unpredictable bit*.
Wire adversary: reset to 0, set to 1.
Gate adversary: $f(x, y) = y$.



Wire adv.: $(k-1)d$ wires
Gate adv.: d gates

Relation between gate and wire adversaries

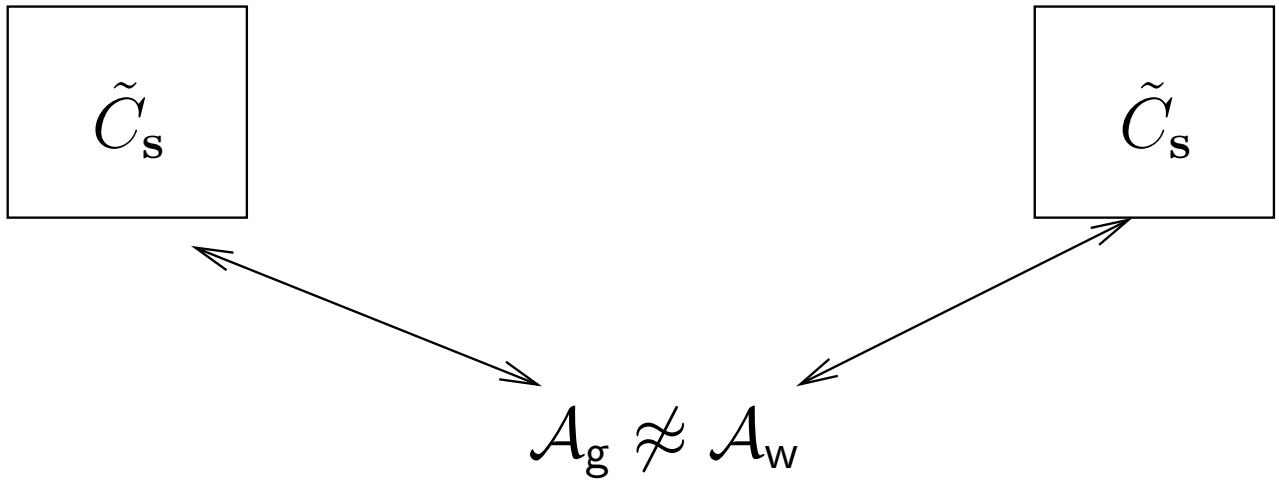
- We consider boolean circuits with binary fan-in.
- There are 16 functions from $\{0,1\}^2$ to $\{0,1\}$.
- **Any tampering attack on wires is simulatable by the gate attacker, e.g.,**:



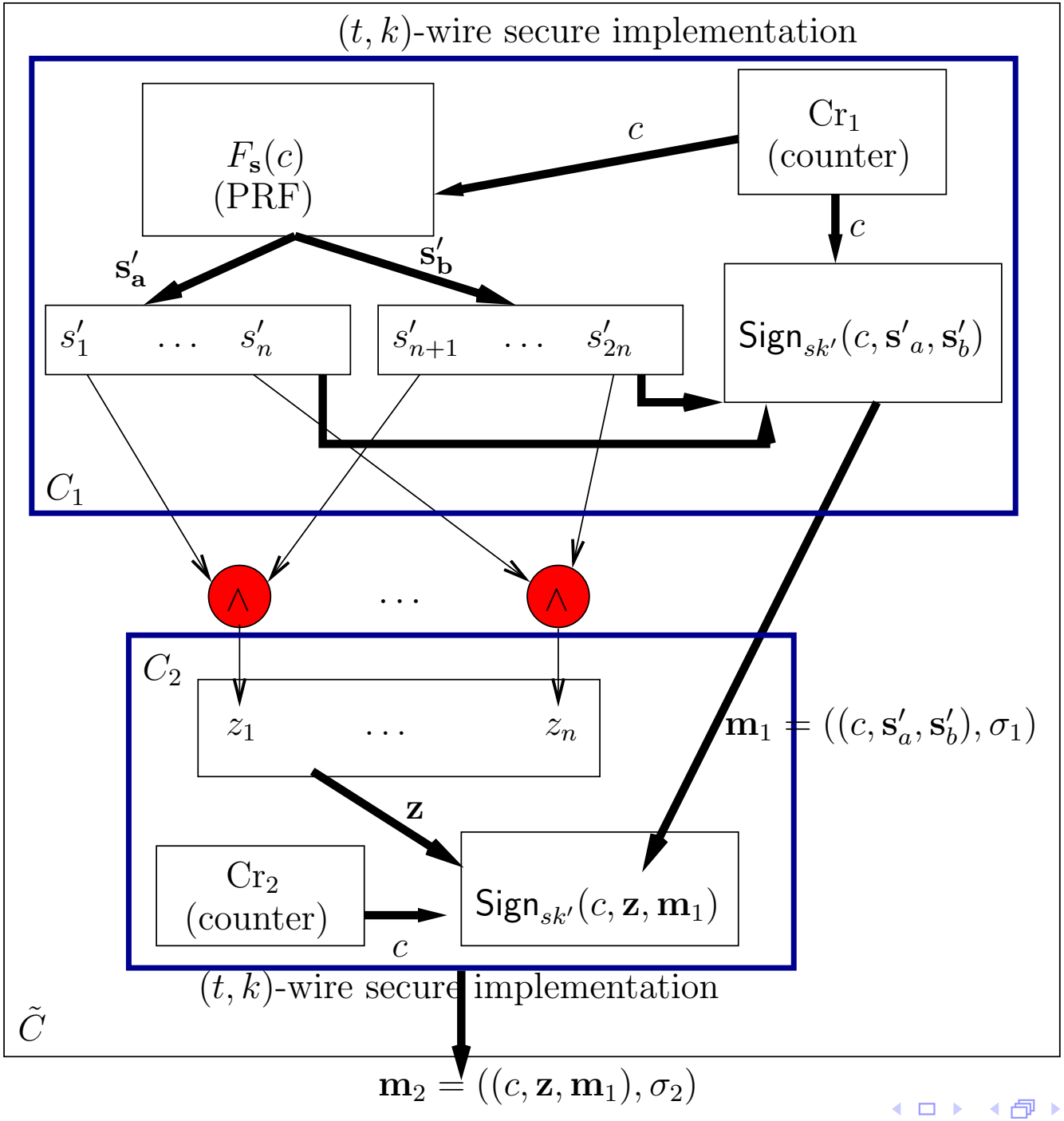
wire attack	$f(x,y)$
(T, z)	$\neg(x \wedge y)$
$(\mathsf{T}, (x,y,z))$	$x \vee y$
(S, x)	y
(T, x)	$\neg x \wedge y$
(R, x)	0

Gate adversaries are strictly stronger

- **Main observation**: the wire adversary cannot produce the XOR and NXOR tampering effects.
- For all $t, k \in \mathbb{N}$, polynomial in n ,we construct a circuit $\tilde{C}$ whose size depends on n, t and k, s.t.



- $\mathcal{A}_{\mathsf{g}}$ tampers with n circuit gates.
- $\mathcal{A}_{\mathsf{w}}$ tampers with up to t circuit wires, where t can be arbitrarily larger than n.


(t,k)-wire secure implementation
$F_{\mathbf{s}}(c)$
(PRF)
c
Cr_1
(counter)
c
$\mathbf{s'_a}$
$\mathbf{s'_b}$
$s'_1 \quad \ldots \quad s'_n$
$s'_{n+1} \quad \ldots \quad s'_{2n}$
$\mathsf{Sign}_{sk'}(c, \mathbf{s}'_a, \mathbf{s}'_b)$
C_1
$\wedge$
$\ldots$
$\wedge$
C_2
$z_1 \quad \ldots \quad z_n$
$\mathbf{m}_1 = ((c, \mathbf{s}'_a, \mathbf{s}'_b), \sigma_1)$
$\mathbf{z}$
Cr_2
(counter)
c
$\mathsf{Sign}_{sk'}(c, \mathbf{z}, \mathbf{m}_1)$
(t,k)-wire secure implementation
$\tilde{C}$
$\mathbf{m}_2 = ((c, \mathbf{z}, \mathbf{m}_1), \sigma_2)$

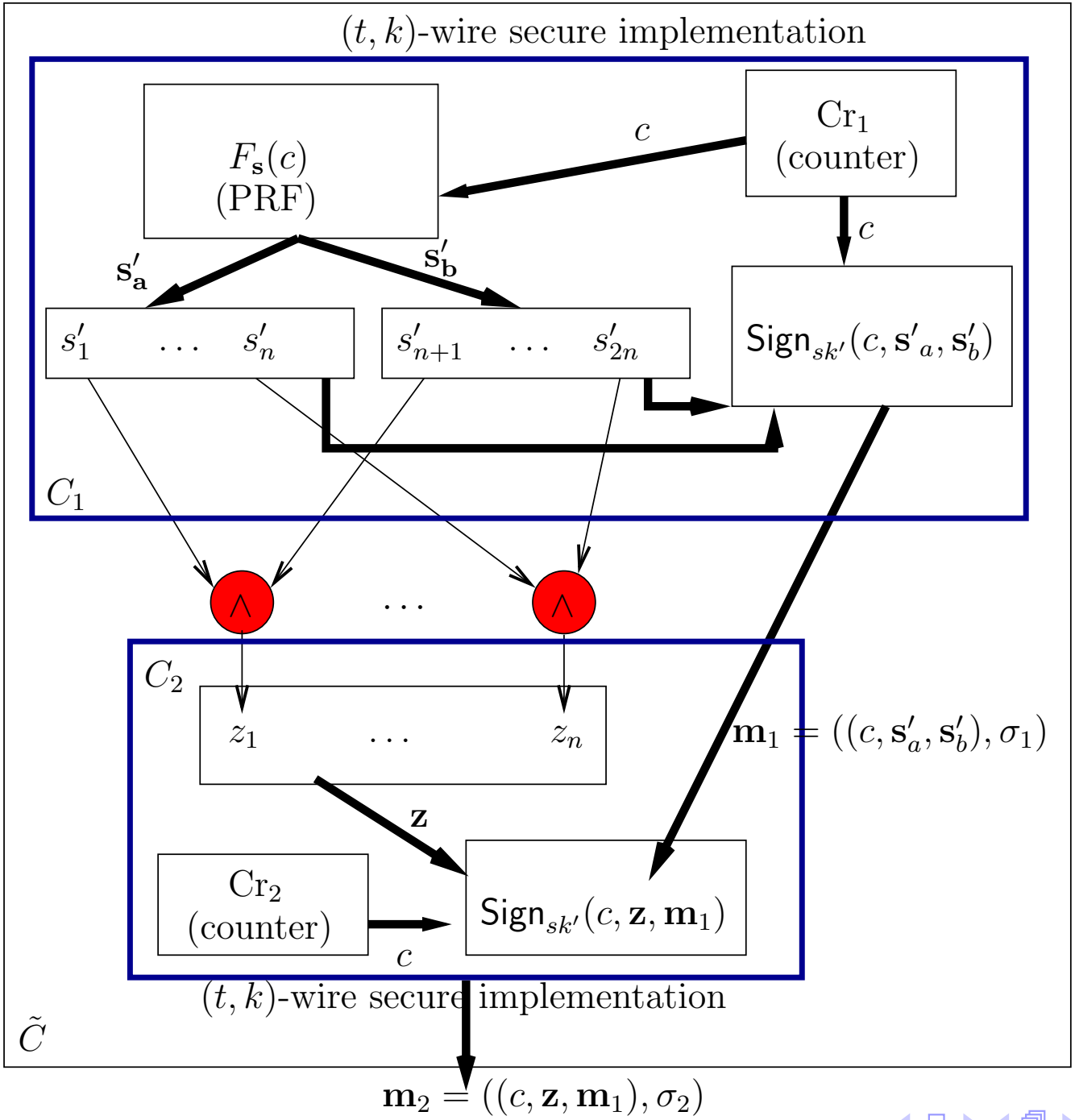
Gate adversaries are strictly stronger (proof idea)

The strategy of $\mathcal{A}_{\mathsf{g}}$:

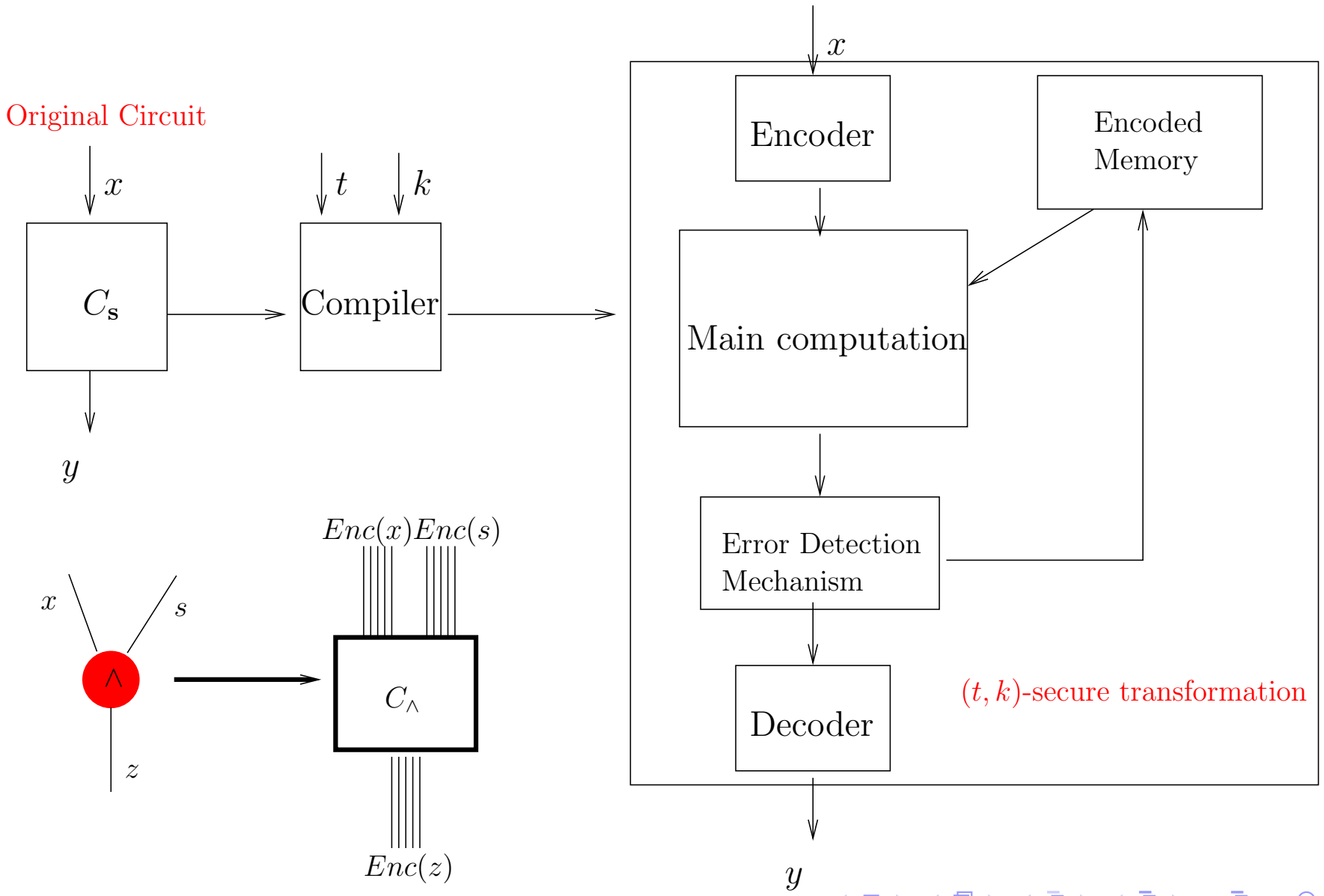
- In one round, $\mathcal{A}_{\mathsf{g}}$ transforms the AND gates into XOR gates and then returns the output of the circuit, i.e., returns $((c, \mathbf{z}, \mathbf{m}_1), \sigma_2)$, where $\mathbf{m}_1 = ((c, \mathbf{s}'_a, \mathbf{s}'_b), \sigma_1)$ and $\mathbf{z} = \mathbf{s}'_a \oplus \mathbf{s}'_b$, while in the normal execution $\mathbf{z} = \mathbf{s}'_a \wedge \mathbf{s}'_b$.
- $\mathcal{A}_{\mathsf{w}}$ needs to produce the same tampering effect while having access to $\tilde{C}$ for polynomially many rounds.

Attack vectors for $\mathcal{A}_{\mathsf{w}}$:

- Do nothing hoping that $\mathbf{s}'_a \wedge \mathbf{s}'_b = \mathbf{s}'_a \oplus \mathbf{s}'_b$. This happens with negligible probability in n.
- Attack the AND gates directly and try to produce the XOR.
- Attack C_1 or C_2 so as to retrieve the secret keys.
- Forge a valid message-signature pair having the desired structure.
- Substitute $\mathbf{m}_1$ with $\mathbf{m}'_1$ taken from a previous computation. Then, the counter values would be different.


(t,k)-wire secure implementation
$F_{\mathbf{s}}(c)$
(PRF)
c
Cr_1
(counter)
c
$\mathbf{s'_a}$
$\mathbf{s'_b}$
$s'_1 \quad \ldots \quad s'_n$
$s'_{n+1} \quad \ldots \quad s'_{2n}$
$\mathsf{Sign}_{sk'}(c, \mathbf{s}'_a, \mathbf{s}'_b)$
C_1
$\ldots$
C_2
$z_1 \quad \ldots \quad z_n$
$\mathbf{m}_1 = ((c, \mathbf{s}'_a, \mathbf{s}'_b), \sigma_1)$
$\mathbf{z}$
Cr_2
(counter)
c
$\mathsf{Sign}_{sk'}(c, \mathbf{z}, \mathbf{m}_1)$
(t,k)-wire secure implementation
$\tilde{C}$
$\mathbf{m}_2 = ((c, \mathbf{z}, \mathbf{m}_1), \sigma_2)$
Gate attacker

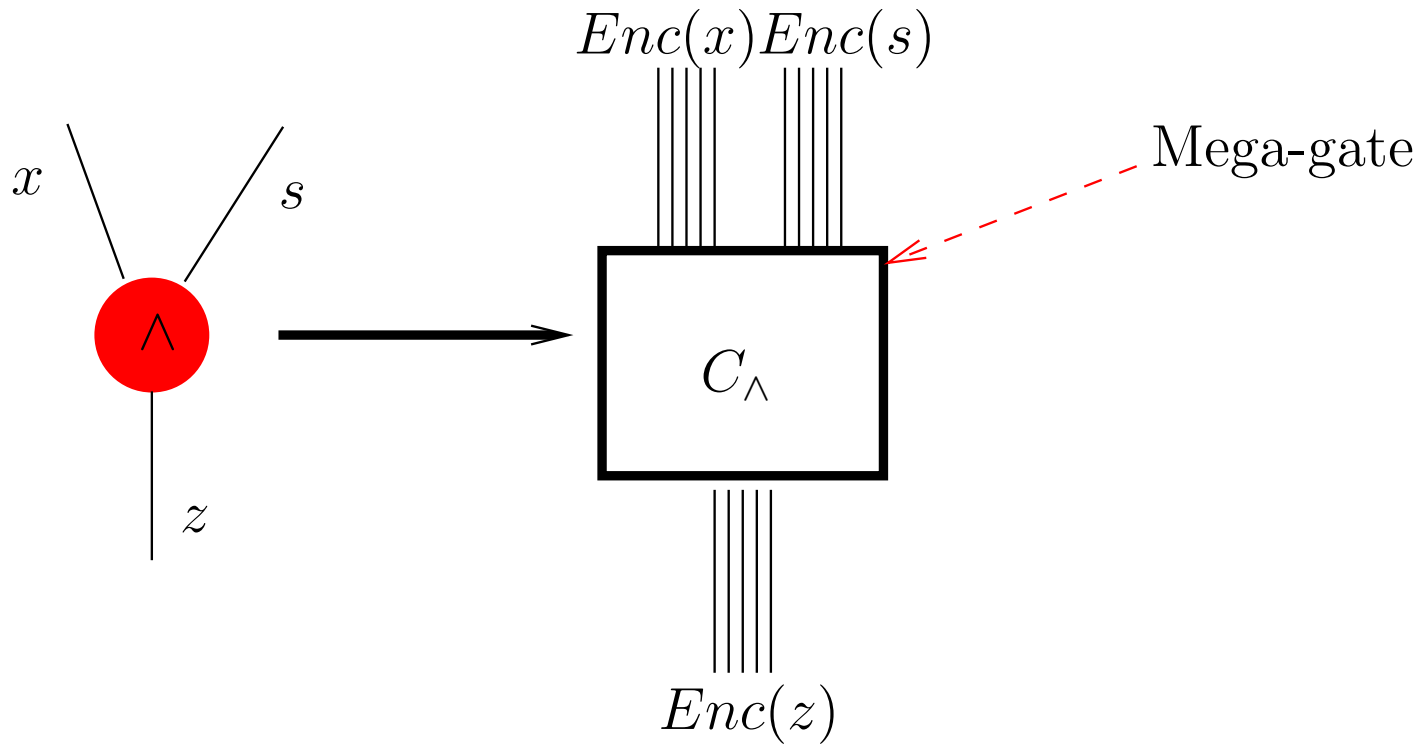
A general compiler strategy



The encoding of [IPSW06]

A randomized additive k-secret sharing:

- x: input bit, s: private memory bit.
- Additive secret sharing $x = r_1 \oplus \ldots \oplus r_k$.
- Then replicate each r_i $2kt$ times (do the same for s).
- $Enc(x) = (r_1^{2kt}, \ldots, r_k^{2kt})$ of length $2k^2t$.
- k: security parameter, t: max. number of attacks.



Security of [IPSW06] against wire attackers

Security relies on:

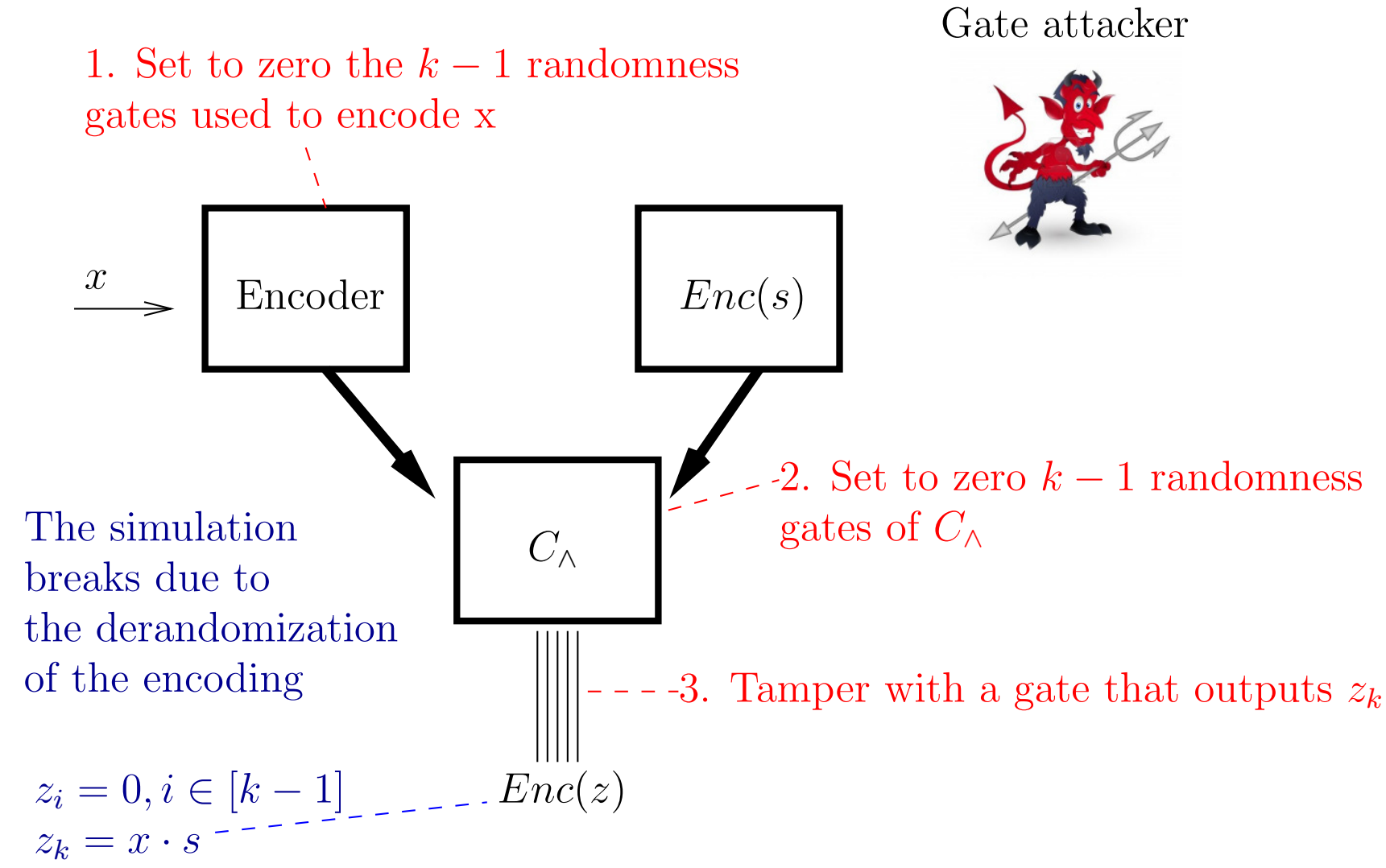
1. The randomization of the encoding.
2. The refreshing of the randomization after each mega-gate operation.

In the case of wire tampering the randomization produced by randomness gates is sufficient.

We show this is not the case for gate attackers.

The gate attack against randomness gates

If each r_i^{2kt} is the output of a randomness gate with fan-out $2kt$ (as in the middle-stage compiler of [IPSW06]):



Circuit compilers and defending against tampering attackers

We introduce a set of characteristics w.r.t. a class of tampering attackers and we prove:
Theorem. Any circuit compiler that satisfies this set of characteristics against a class of tampering attackers produces circuits that are tamper resilient against this class of attackers.

Finally, we show that substituting randomness gates with PRNGs, the [IPSW06] compiler satisfies the set of characteristics w.r.t. gate attackers.

Corollary. There is a circuit compiler that transforms any circuit to a circuit that is tamper-resilient against gate-attackers.

Tamper resilient circuits: The Adversary at the Gates

Yiannis Tselekounis
Joint work with Aggelos Kiayias
University of Athens

ePrint: http://eprint.iacr.org/2013/797

Thank you!