Fault Attack Countermeasures for Error Samplers in Lattice-Based Cryptography

James Howe, Marco Martinoli, Elisabeth Oswald, Cryptography Group, University of Bristol, UK.
Ayesha Khalid Centre for Secure Information Technologies, Queen's University Belfast, Northern Ireland.
Francesco Regazzoni Advanced Learning and Research Institute, Università della Svizzera Italiana, Switzerland.

Motivation

Fault attacks are one of the biggest threats to real-world implementations of cryptographic algorithms. In the attack model, an adversary purposely induces faults and exploits the erroneous behaviour of the circuit to gain some secret-key information. These errors are typically transient in practice, meaning that their effects are reversible. As such, once the fault has propagated through the circuit, the device will continue to operate normally. This approach is advantageous since if the device is not permanently damaged, the attacker can continue to perform many repeated experiments, sufficient to generate and observe the desired effects.

Incorporating **countermeasures** to these fault attacks can incur large amounts of extra resources and/or increase the cryptographic algorithm's runtime. Typical, generic techniques can range from error correcting codes to verify-after-sign, which are essentially **checks** on the outputs. However, these techniques are relatively expensive, so can we do any better?

Assumption

The **Learning With Errors** (**LWE**) problem states that it is hard to retrieve a secret vector $(s_1, \ldots, s_n)$ from the following:

$$\begin{aligned} a_1^1 s_1 + \cdots + a_n^1 s_n + e_1 &= b_1 \\ \vdots \quad \cdots \quad \vdots \quad \vdots \quad &\vdots \\ a_1^m s_1 + \cdots + a_n^m s_n + e_m &= b_m \end{aligned}$$

(or $\mathbf{b} \equiv \mathbf{A}^T\mathbf{s} + \mathbf{e}$), where a_j^i and b_i are publicly known constants, and e_i is drawn from a small error distribution. This error turns the problem from trivial to computationally hard.

There are really strong mathematical reasons why LWE is believed to be hard; the main one relying on a reduction from LWE to certain problems based on **lattices** [Reg05]. These problems (SVP, CVP) are believed to resist against quantum algorithms, thus making LWE a very promising candidate for **post-quantum cryptography**. However, for **real-world** implementations, these schemes needs to be protected from **fault** / **side-channel** attacks.

Contribution

We have designed fault attack countermeasures for error samplers used in lattice-based cryptography. Clearly, the error addition in LWE is key to its computational hardness, and as such, attacking this component has been the foundation of many previous attacks.

The countermeasures are categorised into three classes; **low cost**, **standard**, and **expensive**, named so based on the computational resources required. The countermeasures exploit the expected outputs of the error samplers, whether Gaussian or Binomial, assuming tampering and/or erroneous activity if significant deviation from their expected values occurs. In general, the countermeasures add relatively little to area and almost no impact on the performance of the error sampler.

Test Level	Test Description	Test Formula
Low Cost	Check for repetitions	A counter for if $x_i = c$
Standard	Sample Mean ($\bar{x}$)	$(\Sigma x_i)/n$
	Sample Variance ($\bar{s}$)	$(\Sigma x_i^2 - (\Sigma x_i)^2)/n$
	Standard Error of $\bar{x}$	$\mathrm{SE}_{\bar{x}} = \bar{s}/\sqrt{n}$
	Test Statistic for $\bar{s}$	$T = (n/s)\bar{s}$
	Null Hypothesis	Check if $\lvert\mu\rvert < \bar{x} + t_{\alpha/2}\mathrm{SE}_{\bar{x}}$
	Null Hypothesis	Check if $T < \hat{\chi}^2_{n,\alpha/2}$
Expensive	Chi-Squared Test	$\hat{\chi}^2 = \Sigma\frac{(\mathrm{obs}(k)-\exp(k))^2}{\exp(k)}$
	Test Statistic for $\hat{\chi}^2$	$\chi^2(df = n-1, p\text{-value})$
	Null Hypothesis	Check if $\hat{\chi}^2 < \chi^2$

Hardware Results

Table: Post-place and route results for the proposed countermeasures.

Sampler with Countermeasure	LUT/FF	Slices	DSP/ BRAM	Freq. (MHz)	Clock Cycles	Ops/sec ($\times 10^6$)
Plain CDT Sampler	115/81	33	0/0	297	6	49.5
Low Cost	6/10	3	0/0	-	$+0^\dagger$	-
CDT with Low Cost	123/91	36	0/0	297	6	49.5
Standard	74/58	24	0/0	-	$+1^\dagger$	-
CDT with Standard	182/139	55	0/0	297	6	49.5
Expensive	226/436	126	1/0	-	$+32^\dagger$	-
CDT with Expensive	315/517	149	1/0	297	6	49.5
CDT with Expensive	251/453	129	1/1	193	6	38.6
Plain CDT [HKR+16]‡	112/19	43	0/0	297	5	59.4
Plain CDT [KHR+18]	199/358	81	0/0	100	6	16.67

Hardware Diagrams

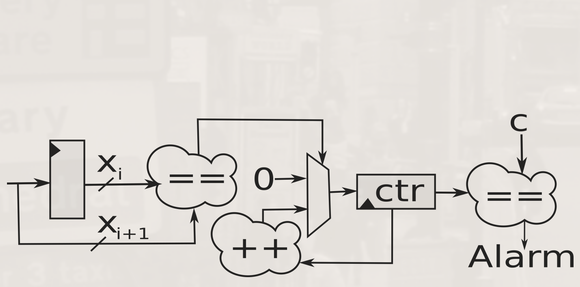

Figure: Low cost countermeasure

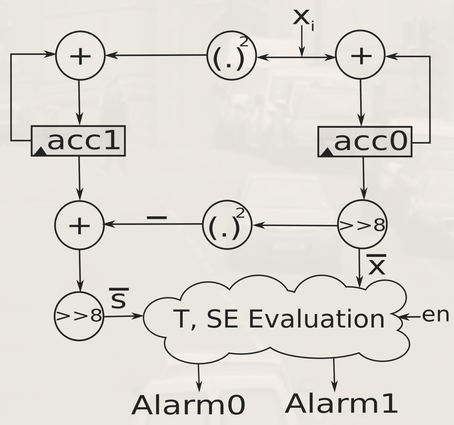

Figure: Standard countermeasure

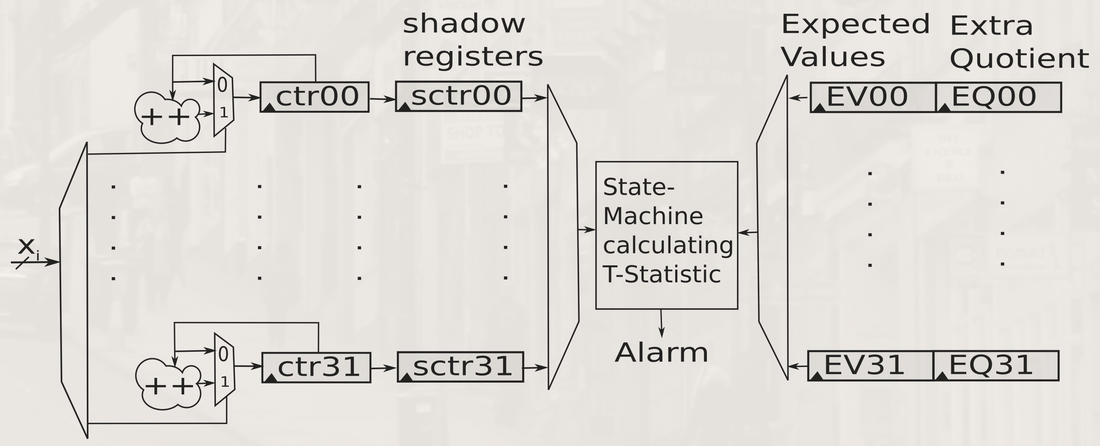

Figure: Expensive countermeasure

This research was supported in part by EPSRC via grant EP/N011635/1 and by the European Union Horizon 2020 SAFEcrypto project (grant no. 644729).