JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

MASK-FALCON: A MASKED GAUSSIAN SAMPLER FOR THE FALCON POST-QUANTUM CRYPTOGRAPHY SCHEME

[1]Bhishma P. Shanbhogue, [2]Ashwini V, [3]Chaitanya Lahari D, [4]B Balaaditya, [5]Steephen Raj K

[1]Student, [2]Assistant Professor, [3]Student, [4]Student, [5]Student

[1]Dept. of ECE

[1]BMS College of Engineering, Bengaluru, Karnataka

Abstract: This paper deals with the security of a post-quantum digital signature scheme namely Falcon, which has been standardised by NIST. Falcon relies heavily on discrete Gaussian sampling, which is susceptible to side-channel attacks, which exploit physical leakage from hardware implementations. To mitigate these attacks, MASK-Falcon, a side-channel resistant Gaussian sampler architecture for Falcon cryptography scheme is presented in this paper. The design incorporates first-order Boolean masking to prevent power analysis attacks. The suggested architecture, which exhibits masking behavior, is described in synthesizable RTL and functionally validated through simulation. For leakage-resilient Falcon implementations in post-quantum applications, MASK-Falcon offers a useful and safe hardware building block.

Index Terms - Post-Quantum Cryptography (PQC), Falcon, Gaussian Sampling, Side-Channel Attacks, Boolean masking

---

I. INTRODUCTION

Widely used public-key cryptographic systems like RSA (Rivest-Shamir-Adleman) algorithm and ECC (Elliptic Curve Cryptography) are rendered insecure by Shor's algorithm which makes factorization and discrete logarithm computation on quantum computers efficient. Consequently, the National Institute of Standards and Technology (NIST) have standardised a number of Post-Quantum cryptography schemes, one of which is Falcon, which has real-world applications in secure communication protocols and blockchain.

In Falcon, signing involves two important steps, firstly, the solution to a closest vector problem (CVP) in a lattice, and secondly, the randomised routing, which must follow a discrete Gaussian distribution. This is carried out by a block named SamplerZ which uses CDT (Cumulative Distribution Table) sampling. This enables constant-time operation and precise control over the output distribution.

Gaussian sampler hardware implementations are however, susceptible to side-channel attacks (SCA), such as differential power analysis (DPA) and timing analysis. While intermediate-value-dependent switching activity in digital circuits can reveal secret data through power consumption, variable-latency sampling techniques like rejection sampling or data-dependent table lookups can leak information through execution time. Falcon exacerbates these difficulties by requiring high-precision arithmetic for Fast Fourier Transform (FFT) operations and sampling.

MASK-Falcon, a side-channel-resistant Gaussian sampler architecture is presented in this work, with the sole aim to reduce power-based leakage. The suggested design combines first-order Boolean masking with constant-time cumulative distribution table sampling. The viability of the architecture is shown by RTL simulation, in Xilinx Vivado.

II. PROBLEM ANALYSIS AND SOLUTION

A. PROBLEM ANALYSIS

Falcon is a widely-used post-quantum digital signature scheme because of its efficiency and compact signatures, but its practical hardware implementation poses serious security issues, mainly because of its reliance on discrete Gaussian sampling. The Gaussian sampler uses data that depends on secrets, it is very vulnerable to side-channel attacks like Differential Power Analysis (DPA). DPA is an attack that exploits the statistical dependence between the power consumption of a device and the secret-dependent intermediate values processed during cryptographic computations.

From an architectural perspective, Falcon operates across two incompatible numeric domains. Polynomial arithmetic is performed over the finite ring $\mathbb{Z}_q$ with relatively small bit widths, whereas Gaussian sampling and FFT-based operations require high-precision real-valued arithmetic. The floating-point hardware units significantly increase area and energy consumption and are difficult to protect using masking techniques. Conversely, insufficient numeric precision degrades the statistical correctness of the Gaussian distribution and compromises Falcon's security.

Furthermore, applying masking to Gaussian sampling is non-trivial due to the non-linear comparison operations involved in cumulative distribution evaluation. Naïve masking approaches often require unmasking intermediate values, reintroducing side-channel leakage and undermining their effectiveness.

B. Proposed Solution

To address these challenges, we propose MASK-FALCON, a secure, Gaussian sampler architecture incorporating power side-channel countermeasures. The design is based on three key principles.

First, a constant-time Gaussian-like sampler intended for hardware evaluation of side-channel resistance in lattice-based cryptographic systems is implemented. The sampler generates integer outputs centered around a configurable mean and supports multiple standard deviation values selected at runtime.

Second, the randomness is provided by a Deterministic Random Bit Generator (DRBG) system, this includes a Keccak permutation core, a True Random Number Generator (TRNG) system and a seed accumulator.

Finally, first-order Boolean masking is applied by splitting each sample into two XOR-related shares to mitigate differential power analysis.

While the output distribution approximates a discrete Gaussian and preserves symmetry and variance control, the design does not aim to be a bit-exact implementation of Falcon's SamplerZ. Instead, it serves as a practical and lightweight architecture to study masked sampling and power-leakage behavior in hardware implementations.

III. System Architecture

The proposed MASK-FALCON architecture is centered around a side-channel-resistant Gaussian sampler integrated within a Falcon signature generation flow. The design is modular and fully described in synthesizable RTL. The architecture is shown below.

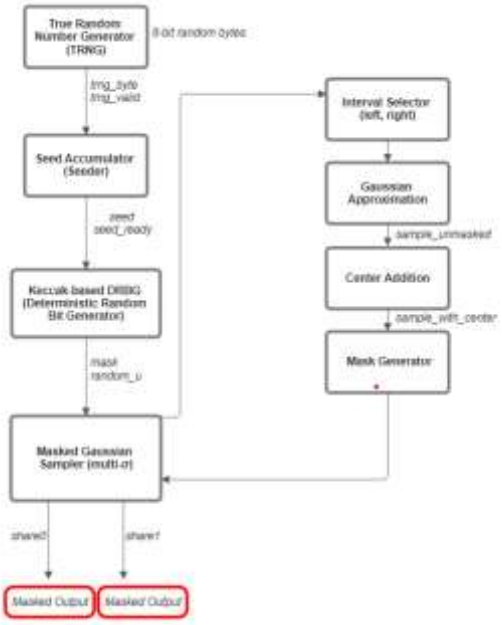

Fig. 1. Suggested architecture

The study of architecture begins with the True Random Number Generator (TRNG). The TRNG is the root source of unpredictability, it generates raw random bytes that are used to initialize the cryptographic state of the system by seeding the DRBG. The seeding process ensures that all subsequent random values, including the masks are unpredictable across executions. The output of TRNG feeds into the seeder.

The seed accumulator collects entropy produced by the TRNG and assembles it into a fixed-length seed suitable for initializing the DRBG. Random bytes from the TRNG are serially accumulated over multiple clock cycles until the required seed width is reached. Once the seed is fully formed, a seed-ready signal is asserted to indicate completion, after which the seed is transferred to the DRBG.

The Deterministic Random Bit Generator (DRBG) expands the high-entropy seed obtained from the seed accumulator into a continuous stream of pseudorandom values required by the sampler. Once initialized, the DRBG produces uniformly distributed random numbers used for generating fresh masking values for each output sample.

The sampler implements SamplerZ-like functionality and it uses a cumulative distribution table, stored as a table in hardware. It then employs binary search to select the proper index for the random number via the interval selector. It then performs center addition and multi-sigma selection from the table.

IV. Methodology And Implementation

The TRNG is realized using two 32-bit Linear Feedback Shift Registers (LFSRs) with different maximal-length polynomials. On each cycle, both LFSRs shift and generate feedback bits, which are XORed to form an 8-bit random byte. The dual-LFSR TRNG implementation is intended for simulation and functional verification only. In production hardware, this must be replaced with a true random source such as a ring oscillator-based TRNG. The polynomials used were:

$$\text{LFSR1: } x^{32} + x^{22} + x^2 + x + 1$$
$$\text{LFSR2: } x^{32} + x^{30} + x^{26} + x^{25} + 1$$

The random bytes from the TRNG are collected from the seed accumulator and a 256-bit seed is formed for the DRBG. The DRBG uses the 256-bit seed from the seed accumulator to generate a stream of random bits for masking the sampler. The DRBG is based on SHAKE256, an extendable-output function (XOF) from the SHA-3 family. It employs a simplified Keccak-based permutation derived from Keccak-f[1600], in which the five step mappings (θ, ρ, π, χ, and ι) are not fully implemented in order to reduce hardware complexity. The Keccak core iterates the simplified permutation 24 times during the 'Absorb' and 'Squeeze' phases. In the 'Absorb' phase, the seed is XOR-ed into the rate portion (1088 bits) of the 1600-bit Keccak state, followed by the permutation. The output blocks are extracted from the rate portion of the state in the 'Squeeze' phase. Each new byte is deterministically computed from the previous state but appears random to an observer. The output of DRBG is an 8-bit byte every clock cycle. These bytes are fed into the Gaussian sampler to generate fresh masks for each sample.

The sampler receives as inputs a center value (the mean around which the sample is drawn), a sigma selection and a pseudo-random value from the DRBG. Using these inputs, it computes 'left' and 'right' values, and combines them to produce a raw, unmasked sample. The sampler implements a multi-sigma approach, allowing it to select from different standard deviations depending on the application (leaf or root nodes in Falcon's tree-based sampling).

The highlight of this architecture is the masking phenomenon. Each sample is split into two shares using a Boolean masking scheme, which involves two important mechanisms, the mask generation and share computation. The former occurs when fresh pseudo-random masks are generated from the DRBG for every sample. Subsequently, the first share (share0) is computed as the XOR of the raw sample with the share1, where the second share (share1) is the mask. When needed, the original sample can be reconstructed by XOR-ing the two shares. This formulation ensures that neither share alone reveals information about the raw sample.

The sampler is designed to support multiple standard deviation values to meet the tree-based sampling requirements. Specifically, four values of sigma (σ = 1.278, 1.40, 1.55 and 1.85) are supported. Each sigma is associated with a pre-computed 21-entry cumulative distribution table stored in on-chip memory. Given an input random value, the sampler performs a logarithmic-time binary search over the table entries to determine the corresponding sample. After base sample selection, a configurable center offset is added to shift the mean of the distribution as required by Falcon's sampling algorithm.

V. Results and Discussion

To verify the correctness of our masked sampler, a comprehensive Verilog testbench was implemented. The testbench stimulates the system through reseed, sample generation and multiple sigma selection scenarios. The simulation console confirms the expected behavior of the system, the individual shares (share0 and share1) appear random, validating the effectiveness of the process. The reconstructed sample (XOR of the shares) correctly produced 0, showing that the masked combination reconstructs the intended sample value exactly.

The simulation results verify the correctness and robustness of the proposed design. Specifically, the cumulative distribution table (CDT) binary search was observed to converge correctly, with the left and right indices coinciding at completion. The sampler produced the expected output for a representative input value (u = 0.5), yielding a sample of zero. The design also demonstrated correct multi-sigma operation across all four supported standard deviation values. In addition, successful DRBG reseeding and continuous mask generation were verified through simulation.

Key metrics from RTL simulation (Vivado 2021.2):

- Sample generation latency: ~320ns (32 clock cycles @ 100MHz)
- DRBG reseed time: ~330ns (requires 32 TRNG bytes)
- Throughput: ~3.1 million samples/second

```
DEBUG: Internal sampler state:
  left =  10, right =  10
  sample_unmasked =      0
  sample_with_center =      0
  current_sigma_sel = 0
Sigma[0] = 1.278 (Leaf):
  Share0 =  30197 (0x75f5)
  Share1 =  30197 (0x75f5)
  Reconstructed =      0 (Expected: 0)
  [PASS] XOR verification:  30197 XOR  30197 =      0
  [PASS] Sample value correct!
```

Fig. 2. Output as shown in TCL console

Users can view all the source files, testbench, and supporting modules in our GitHub repository [1] and try simulating the system themselves to reproduce the results.

This work demonstrates a practical masked Sampler suitable for Falcon signatures. Future extensions of this work include synthesis on FPGA, support for higher-order masking to further enhance side-channel resistance, integration with a complete Falcon sign and verify pipeline, and formal verification of side-channel security properties.

REFERENCES

[1] B. P. Shanbhogue, "Masked Gaussian Sampler," GitHub repository, 2025. [Online]. Available at: https://github.com/BhishBobs/Masked_Gaussian_Sampler

[2] R. K. Zhao, R. Steinfeld, and A. Sakzad, "FACCT: FAst, Compact, and Constant-Time Discrete Gaussian Sampler over Integers," IEEE Transactions on Computers, vol. 69, no. 1, January 2020

[3] T. Wang, C. Zhang, P. Cao, and D. Gu, "Efficient Implementation of Dilithium Signature Scheme on FPGA SoC Platform," IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 30, no. 9, September 2022

[4] E. Karabulut, E. Alkim, and A. Aysu, "Efficient, Flexible, and Constant-Time Gaussian Sampling Hardware for Lattice Cryptography," IEEE Transactions on Computers, vol. 71, no. 8, August 2022

[5] T. Xuan Pham, P. Duong-Ngoc, and H. Lee, "An Efficient Unified Polynomial Arithmetic Unit for CRYSTALS-Dilithium," IEEE Transactions on Circuits and Systems—I: Regular Papers, vol. 70, no. 12, December 2023