FALCON-PARE: Full-Range Side-Channel Key Recovery of Falcon via Prior-Enhanced Evidence

Shuyi Chen*, Yuejun Liu*, Jingdian Ming*, Wei Cheng*, Yuhan Zhao*, and Yongbin Zhou*†
*School of Cyber Science and Engineering, Nanjing University of Science and Technology, Nanjing, China
†Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
{chenshuyi, liuyuejun, mingjingdian, wei.cheng, yhzhao96, zhouyongbin}@njust.edu.cn

***Abstract*—Falcon is a compact post-quantum signature scheme selected by the National Institute of Standards and Technology (NIST), and its number-theoretic transform (NTT) computations constitute a critical side-channel attack surface. Existing NTT attacks rely on truncated coefficient ranges and do not cover the full valid range of Falcon secret coefficients, limiting their applicability to complete key recovery. This paper proposes FALCON-PARE, a prior-enhanced evidence framework for full-range key recovery of Falcon. FALCON-PARE combines Falcon's discrete-Gaussian prior with aggregated leakage from range conversion and Montgomery multiplication to construct input-layer soft evidence, which is propagated through the NTT factor graph. FALCON-PARE further uses posterior entropy to select reliable coefficients and public-key relation to complete the remaining unknowns, yielding full recovery of the secret polynomials f and g. We systematically evaluate FALCON-PARE on STM32F407 under `-O0` and `-O3` compiler settings, validating its practicality in terms of profiling-trace requirements, coefficient-wise accuracy, and residual candidate-space reduction. FALCON-PARE achieves single-trace full key recovery with 40 and 120 profiling traces, respectively. At the comparison budgets, FALCON-PARE achieves absolute gains of 54.79% and 73.83% in coefficient-wise accuracy over existing soft-analytical side-channel attacks (SASCA) for NTT leakage, and reduces the residual candidate space from approximately 2^{4352} and 2^{5346} to one, respectively.**

***Index Terms*—Falcon, side-channel analysis, number-theoretic transform, belief propagation, post-quantum cryptography**

I. Introduction

Large-scale quantum computers are expected to undermine the hardness assumptions underlying widely deployed public-key cryptosystems. In particular, Shor's algorithm can solve integer factorization and discrete logarithms in polynomial time, thereby threatening RSA, Diffie–Hellman key exchange, and elliptic-curve cryptography [1]. This quantum threat has motivated the post-quantum cryptography standardization process led by the National Institute of Standards and Technology (NIST). Within this process, Falcon was selected for standardization and is being developed as the basis of the future FIPS 206 FN-DSA standard [2], [3]. Falcon follows the Gentry–Peikert–Vaikuntanathan hash-and-sign paradigm instantiated over NTRU lattices and employs fast Fourier sampling to generate short lattice vectors. Its security relies on the hardness of finding short vectors in structured NTRU lattices [4]. Compared with other lattice-based post-quantum signature schemes, Falcon offers compact signatures, small public keys, and efficient verification, making it particularly attractive for resource-constrained applications with limited communication and storage capacity.

Algorithmic security, however, does not by itself establish the security of concrete implementations [5]. Side-channel attacks exploit information leaked through the physical or microarchitectural behavior of a device, such as timing, power consumption, or electromagnetic emissions [6]–[9], rather than flaws in the underlying cryptographic algorithm or its mathematical assumptions. Moreover, post-quantum schemes introduce implementation components that differ substantially from those in classical public-key cryptography. Operations such as number-theoretic transforms and polynomial multiplication create new secret-dependent leakages. Falcon implementations must therefore be evaluated against side-channel analysis, even when the underlying scheme is secure under its mathematical hardness assumptions.

Existing side-channel studies on Falcon have targeted key modules including fast Fourier transform (FFT)-based multiplication, preimage computation, and integer Gaussian sampling. Karabulut and Aysu [10] exploited floating-point multiplication leakage in FFT-based preimage computations and introduced an extend-and-prune strategy for secret recovery. Guerreau et al. [11] analyzed leakages from Falcon's preimage computation and trapdoor sampling, and adapted hidden-parallelepiped techniques to recover the secret basis from filtered signatures. Zhang et al. [12] improved Gaussian-sampler attacks by using second-order covariance statistics and spectral decomposition to extract secret-dependent directions more efficiently. Lin et al. [13] performed a systematic power analysis of Falcon's integer Gaussian samplers and proposed practical countermeasures against the observed half-Gaussian and sign leakages. Chen et al. [14] proposed SERP-SCA, a strengthened side-channel attack framework for Falcon signatures based on floating-point multiplication leakage. Qiu and Aysu [15] proposed SHIFT SNARE, which exploits single-trace bit-shift leakage in Falcon key-generation sampling to recover secret keys.

The number-theoretic transform (NTT) is also a core arithmetic primitive in Falcon, supporting efficient polynomial

Corresponding authors: Yuejun Liu and Yongbin Zhou. This work is supported in part by the National Natural Science Foundation of China (No.U2336205, No.62502207, No.62302226, No.62302224), Yunnan Provincial Major Science and Technology Special Plan Projects (202502AD080008) and Yunnan Provincial New R&D Institution Cultivation Project (202404BQ040148).

multiplication in computations involving the secret polynomials. Qiu et al. proposed NTT-PEEL, a deterministic NTT-pruning attack that exploits binary modular-reduction leakage in Falcon's NTT and performs forward/backward candidate pruning over a restricted coefficient range [16]. However, it restricts private-key coefficient candidates to a narrow range, whereas the Falcon reference implementation allows a wider valid coefficient range. If a true coefficient falls outside the assumed candidate set, the correct key is removed at the initial stage and cannot be reintroduced by subsequent propagation; as a result, the private key cannot be recovered in this case.

Beyond deterministic NTT pruning, another line of work exploits NTT-related leakage through soft-analytical side-channel attacks (SASCA). SASCA combines noisy side-channel observations with algebraic constraints and performs probabilistic inference over secret variables. Existing SASCA studies on NTT computations have mainly focused on lattice-based encryption schemes such as ring-LWE encryption and Kyber. Primas et al. [17] introduced the first single-trace attack on lattice-based encryption by combining template information from NTT operations with belief propagation (BP) and lattice decoding. Pessl and Primas [18] further improved the practicality of single-trace NTT attacks by targeting constant-time Kyber encryption, using improved BP and lightweight Hamming-weight (HW) templates. Hamburg et al. [19] extended this line of work to chosen-ciphertext k-trace attacks on masked CCA2-secure Kyber, where crafted sparse ciphertexts improve the noise tolerance of NTT-based BP and enable long-term key recovery. These studies demonstrate the effectiveness of probabilistic inference for exploiting NTT-related leakage. However, to the best of our knowledge, existing SASCA-based NTT attacks have not targeted Falcon's NTT computation. Compared with prior SASCA attacks on module-LWE encryption schemes such as Kyber, Falcon presents a different NTT leakage-inference problem. Its secret coefficients have a wider implementation-level range, its modulus is larger ($q = 12289$), and Falcon-512 uses higher-dimensional secret polynomials. These features expand the candidate space and reshape the factor-graph inference problem, thereby posing significant challenges for applying existing SASCA-based NTT attacks to Falcon.

This paper proposes FALCON-PARE, a prior-enhanced evidence framework for full-range key recovery of Falcon. Our contributions are summarized as follows:

- We design FALCON-PARE as a full-range probabilistic inference framework for Falcon secret-key recovery. It retains all valid Falcon-512 coefficient candidates and assigns them soft probabilities rather than truncating the candidate range.
- We construct prior-enhanced soft evidence by combining Falcon's discrete-Gaussian prior with profiled likelihoods from range conversion and Montgomery multiplication. This aggregated evidence strengthens the input layer of the NTT factor graph and improves coefficient-wise posterior inference during BP.
- We introduce an entropy-guided key completion strategy that selects high-confidence posterior coefficients and exploits Falcon's public-key relation to infer the remaining unknowns. This converts reliable partial coefficient information obtained through side-channel inference into complete recovery of the secret polynomials f and g.
- We systematically evaluate FALCON-PARE on Falcon-512 implementations running on an STM32F407 platform under `-O0` and `-O3`, measuring key-recovery accuracy across different compiler optimizations and profiling budgets. These experiments validate the practical applicability of the framework. FALCON-PARE achieves single-trace full key recovery with 40 and 120 profiling traces, respectively. At the comparison budgets, it achieves absolute gains of 54.79% and 73.83% in coefficient-wise recovery accuracy over existing SASCA methods for NTT leakage, and reduces the residual candidate space from approximately 2^{4352} under `-O0` and 2^{5346} under `-O3` to one.

Algorithm 1 Falcon Key Generation

Input: A monic polynomial $\phi \in \mathbb{Z}[x]$, a modulus q
Output: A secret key sk and a public key h

$f, g, F, G \leftarrow \text{NTRUGen}(\phi, q)$
$\mathbf{B} \leftarrow \left[\begin{array}{c|c} g & -f \\ \hline G & -F \end{array}\right]$
$\widehat{\mathbf{B}} \leftarrow \text{FFT}(\mathbf{B})$
$\mathbf{G} \leftarrow \widehat{\mathbf{B}} \times \widehat{\mathbf{B}}^*$
$\mathsf{T} \leftarrow \text{ffLDL}^*(\mathbf{G})$
for each leaf of T **do**
  $\mathsf{leaf.value} \leftarrow \sigma/\sqrt{\mathsf{leaf.value}}$
end for
$\mathsf{sk} \leftarrow (\widehat{\mathbf{B}}, \mathsf{T})$
$h \leftarrow gf^{-1} \bmod q$
return sk, h

Algorithm 2 Falcon Signature Generation

Input: A message $\mathbf{m}$, a secret key sk, a bound $\lfloor \beta^2 \rfloor$
Output: A signature $\mathsf{sig} = (\mathbf{r}, \mathsf{s})$ of $\mathbf{m}$

$\mathbf{r} \leftarrow \{0,1\}^{320}$ uniformly
$c \leftarrow \text{HashToPoint}(\mathbf{r} \parallel \mathbf{m}, q, n)$
$\mathbf{t} \leftarrow \left(-\frac{1}{q}\text{FFT}(c) \odot \text{FFT}(F), \frac{1}{q}\text{FFT}(c) \odot \text{FFT}(f)\right)$
do
  do
    $\mathbf{z} \leftarrow \text{ffSampling}(\mathbf{t}, \mathsf{T})$
    $\mathbf{s} \leftarrow (\mathbf{t} - \mathbf{z})\widehat{\mathbf{B}}$
  while $\|\mathbf{s}\|^2 > \lfloor \beta^2 \rfloor$
$(s_1, s_2) \leftarrow \text{invFFT}(\mathbf{s})$
$\mathsf{s} \leftarrow \text{Compress}(s_2, 8 \cdot \text{sbytelen} - 328)$
while $(\mathsf{s} = \perp)$
return $\mathsf{sig} = (\mathbf{r}, \mathsf{s})$

II. Background

A. Falcon Scheme

Falcon is a hash-and-sign lattice-based signature scheme instantiated over $\mathbb{Z}[x]/(\phi)$, where $\phi = x^n + 1$, $n \in \{512, 1024\}$,

and $q = 12289$ [4]. Its secret key is derived from short polynomials f, g, F, G satisfying the NTRU equation

$$fG - gF = q \pmod{\phi}, \tag{1}$$

where f and g are sampled from a centered discrete Gaussian distribution with standard deviation $\sigma_{\text{key}} = 1.17\sqrt{q/(2n)}$. The public key is $h = gf^{-1} \bmod q$, while the signing key stores an FFT representation of the NTRU basis and an ffLDL tree for fast Fourier sampling. During signing, Falcon hashes the message with a salt to a target point and samples a short preimage; a valid signature (s_1, s_2) satisfies $s_1 + s_2 h \equiv c \pmod{(\phi, q)}$. Key generation and signature generation are summarized in Algorithms 1 and 2, respectively.

The side-channel leakage studied in this paper arises from the modular arithmetic path that handles the secret polynomials f and g. In the Falcon reference implementation, the signed small coefficients of f and g are first converted to their modulo-q representation by `mq_conv_small()`, and the resulting arrays are then transformed to the NTT domain by `mq_NTT()`. These two routines form a critical secret-dependent path that appears in the public-key computation of key generation and in the secret-key computation of signature generation under the dynamic signing mode.

B. Threat Model

We consider a passive profiled side-channel adversary under the standard template-attack and side-channel key-recovery models [20], as in recent attacks against Falcon [10], [16]. That is, the adversary knows the Falcon implementation, public parameters, and public key, and can measure power consumption while the target device executes cryptographic operations. During profiling, the adversary has access to a clone of the target device and controls the secret key and input data, enabling labeled traces to be collected for leakage models of the range-conversion, Montgomery multiplication, and NTT operations. During the attack phase, the adversary obtains a single power trace from the target device for the target key instance. The adversary's goal is to recover the secret polynomials f and g, since recovering f and g is sufficient to reconstruct the remaining signing-key material.

C. Limitations of Deterministic NTT Pruning

A prior deterministic pruning approach, NTT-PEEL, shows that modular-reduction leakage in Falcon's NTT can be combined to reduce secret-coefficient candidates [16]. However, it restricts each secret coefficient to the narrow range $D_{10} = [-10, 10]$. According to the specification, Falcon-512 coefficients of f and g are sampled from a centered discrete Gaussian with $\sigma_{\text{key}} = 1.17\sqrt{q/(2n)} \approx 4.05$, and the reference implementation encodes them as 6-bit signed integers within $D_{31} = [-31, 31]$, rejecting any coefficient outside this range during key generation [4]. To quantify this truncation, we report the coverage of several candidate ranges in Table I. For each range $[-B, B]$, the theoretical coefficient coverage is computed under Falcon's discrete-Gaussian key distribution conditioned on the valid implementation range D_{31}, and the theoretical full-key coverage is obtained by raising this coefficient coverage to the power of 1024, corresponding to all coefficients of f and g. Empirical coverage is measured over 100,000 valid Falcon-512 key pairs generated by the reference implementation. Since the implementation rejects coefficients outside D_{31}, the D_{31} row has 100% coverage.

TABLE I: Coverage of Falcon-512 Secret Coefficients Under Different Truncation Ranges

	Theoretical		Empirical	
Range	Coeff.	Full key	Coeff.	Full key
$[-10, 10]$	99.06000%	0.00631%	99.14791%	0.01564%
$[-15, 15]$	99.98738%	87.87657%	99.98998%	90.24789%
$[-20, 20]$	99.99996%	99.95905%	99.99998%	99.97952%
$[-31, 31]$	100.00000%	100.00000%	100.00000%	100.00000%

Although D_{10} covers most individual coefficients, it does not cover a full Falcon-512 secret key with high probability. In particular, the theoretical per-coefficient coverage of D_{10} is about 99.0600%, whereas the corresponding full-key coverage over the 1024 coefficients of f and g is only 0.00631%. The empirical results follow the same trend: D_{10} covers about 99.1479% of individual coefficients but only 0.01564% of complete keys. Thus, D_{10} represents a high-probability truncation for individual coefficients, not the full valid coefficient range. If the true coefficient $x_i^\star \notin D_{10}$, then $\Pr(X_i = x_i^\star \mid X_i \in D_{10}) = 0$; the correct value is absent from the initial support, and the private key cannot be recovered by pruning over the truncated support alone.

Simply extending the deterministic candidate range restores support coverage but does not by itself ensure tractable recovery. NTT-PEEL reports a residual exponent of $Y_{\text{exp}}(10) \approx 99.8$ after deterministic pruning over the truncated range D_{10} for one 512-coefficient secret polynomial [16]. Expanding the range to D_{31} increases the number of candidate values per coefficient from 21 to 63. For one polynomial, this adds $\log_2\big((63/21)^{512}\big) = 512 \log_2 3 \approx 812$ bits to the initial candidate-space exponent, while the number of observable reduction flags remains unchanged. Thus, the corresponding range-expansion estimate is $Y_{\text{exp}}(31) \approx Y_{\text{exp}}(10) + 512 \log_2 3 \approx 911$. NTT-PEEL further reduces the residual search space from about $2^{99.8}$ to about 2^{24} using lattice-reduction postprocessing [16]. However, this postprocessing depends on the resulting candidate-set structure and cannot be treated as a fixed compression factor transferable to a wider candidate range. Even if the reported 75.8-bit reduction is nevertheless applied to the expanded-range estimate, the remaining search space is still about 2^{836}, well above the 2^{64} serial-depth reference level in NIST's post-quantum cryptography evaluation criteria [21]. This estimate indicates that truncated-range hard pruning does not directly establish tractable recovery over the full valid coefficient range. This motivates our full-range soft-evidence formulation, which keeps all valid coefficient values in the support and weights them using Falcon's prior distribution and observed leakage.

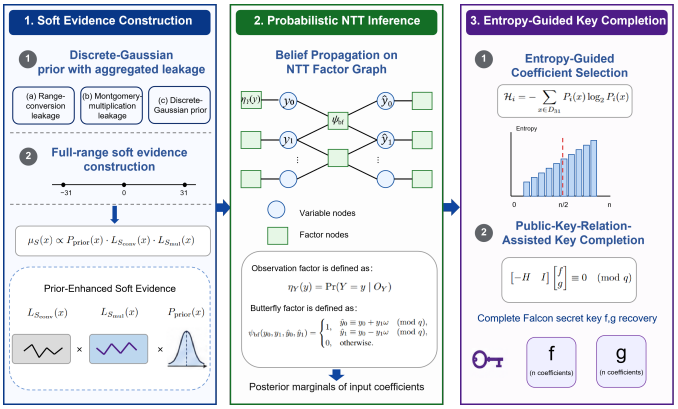


Fig. 1: Overview of the proposed FALCON-PARE framework.

III. The Proposed FALCON-PARE Framework

We present FALCON-PARE, a prior-enhanced evidence framework for full-range Falcon key recovery from aggregated leakage. The acronym PARE captures the main design principles of the framework: **P**rior-enhanced evidence, **A**ggregated leakage, public-key **R**elation-assisted key completion, and **E**ntropy-guided coefficient selection. FALCON-PARE retains the full valid coefficient range, aggregates complementary leakage, propagates the resulting soft evidence through an NTT factor graph, and completes the key from entropy-selected high-confidence coefficients using Falcon's public-key relation.

As shown in Fig. 1, FALCON-PARE consists of three main stages. First, it constructs prior-enhanced soft evidence by combining Falcon's discrete-Gaussian prior with leakage observations from range conversion and Montgomery multiplication. Second, it performs probabilistic inference on an NTT factor graph to propagate the aggregated evidence. Third, it evaluates coefficient reliability using posterior entropy, selects high-confidence coefficients, and uses Falcon's public-key relation to complete the remaining unknown coefficients and recover the secret polynomials.

A. Prior-Enhanced Soft Evidence Construction from Aggregated Leakage

The first stage constructs prior-enhanced soft evidence for Falcon secret coefficients. As discussed in Section II, the secret polynomials f and g are first converted by `mq_conv_small` and then processed by `mq_NTT`; hence, leakage from NTT nodes carries information about their coefficients. However, Falcon's parameter characteristics enlarge the effective candidate space across the NTT factor graph, making it difficult to distinguish the correct secret-key coefficients from NTT-node leakage alone. This limits the direct applicability of existing SASCA-based NTT attacks to Falcon. FALCON-PARE therefore strengthens the input-layer soft evidence by combining Falcon's discrete-Gaussian prior with aggregated leakage from range conversion before the NTT and first-layer Montgomery multiplication in the NTT.

Algorithm 3 `mq_conv_small`

Input: A signed small coefficient x, modulus q
Output: Its modulo-q representation y

1: $y \leftarrow \mathrm{uint32}(x)$ ▷ *leakage point*
2: $y \leftarrow y + (q \,\&\, -(y \gg 31))$ ▷ *leakage point*
3: **return** y

Algorithm 4 `mq_montymul`

Input: Operands $x, y \in \mathbb{Z}_q$, constants $q, \mathsf{Q0I}$
Output: Montgomery product $z \in \mathbb{Z}_q$

1: $z \leftarrow x \cdot y$ ▷ *leakage point*
2: $w \leftarrow ((z \cdot \mathsf{Q0I}) \,\&\, \mathtt{0xFFFF}) \cdot q$ ▷ *leakage point*
3: $z \leftarrow (z + w) \gg 16$
4: $z \leftarrow z - q$ ▷ *leakage point*
5: $z \leftarrow z + (q \,\&\, -(z \gg 31))$
6: **return** z

Discrete-Gaussian prior. For each candidate value $x \in D_{31}$, Falcon samples secret coefficients from a centered discrete Gaussian distribution [4]. This gives the prior

$$P_{\mathrm{prior}}(x) \propto \exp\left(-\frac{x^2}{2\sigma_{\mathrm{key}}^2}\right), \tag{2}$$

where σ_{key} denotes the standard deviation of Falcon's secret-coefficient distribution. This prior does not truncate the coefficient range. Instead, it assigns larger probability mass to more likely central coefficients while preserving low-probability tail coefficients. This is essential for Falcon because tail coefficients may still appear in a complete Falcon-512 secret key. If a true coefficient is removed from the initial support, subsequent propagation cannot reintroduce it.

Aggregated leakage. FALCON-PARE aggregates leakage evidence over selected intermediate values. The first leakage set comes from the range-conversion routine `mq_conv_small`. It includes the cast from a signed coefficient to an unsigned word and the subsequent conditional addition of q according to the sign bit. These operations characterize the transition from the signed small-integer domain to the modulo-q representation. The second leakage set is extracted from the first NTT layer. In this layer, only half of the secret coefficients enter the Montgomery multiplication path, while the remaining coefficients are directly loaded as butterfly inputs. For the multiplied coefficients, FALCON-PARE uses three intermediate values in `mq_montymul`, corresponding to the raw product, the Montgomery correction term, and the post-shift subtraction. Since these intermediate values are deterministic functions of the coefficient candidate and public constants, their HW distributions provide additional candidate-dependent leakage information. The *leakage point* annotations in Algorithms 3 and 4 indicate the intermediate values targeted by our leakage model.

We quantify the candidate separability of the selected intermediates under an ideal HW model [8], [22]. Let X denote an arbitrary secret-coefficient variable of f or g, and let $x \in D_{31}$ be a candidate value of X. For a selected set of leakage points

S, let $\ell_S(x)$ denote the HW vector induced when $X = x$, and let

$$A_S(\ell_S(x)) = \{x' \in D_{31} : \ell_S(x') = \ell_S(x)\}$$

denote the set of candidate values that produce the same HW vector. Here, S_{conv} denotes range-conversion leakage points, S_{mul} denotes first-layer Montgomery-multiplication leakage points, and $S_{\text{conv+mul}} = S_{\text{conv}} \cup S_{\text{mul}}$. We define the unique-discrimination probability as

$$U(S) = \sum_{x \in D_{31}} P_{\text{prior}}(x) \cdot \mathbf{1}\left[|A_S(\ell_S(x))| = 1\right], \tag{3}$$

where $\mathbf{1}[\cdot]$ is the indicator function. The condition $|A_S(\ell_S(x))| = 1$ means that x is the only candidate value that produces the vector $\ell_S(x)$, and thus can be uniquely distinguished under the ideal HW model.

For range-conversion leakage alone, we obtain $U(S_{\text{conv}}) = 0.4103$. When the Montgomery multiplication leakage is included, this value increases to $U(S_{\text{conv+mul}}) = 0.9655$, showing that the multiplication intermediates substantially improve candidate separability. For one 512-coefficient polynomial, half of the coefficients receive only range-conversion evidence, while the other half additionally receive first-layer multiplication evidence. The expected number of uniquely distinguishable coefficients under the ideal model is therefore

$$256U(S_{\text{conv}}) + 256U(S_{\text{conv+mul}}) \approx 352. \tag{4}$$

For the two secret polynomials f and g, which follow the same coefficient distribution, this expectation is approximately 704. This value exceeds the half-key threshold required by the subsequent entropy-guided key completion. The analysis indicates that the selected leakage points provide sufficient structural discrimination to serve as soft evidence sources for FALCON-PARE.

In the actual profiled setting, leakage observations are noisy and are modeled probabilistically. Let O_S denote the observed trace segments associated with leakage set S. The leakage likelihood is defined as

$$L_S(x) = \Pr(O_S \mid X = x). \tag{5}$$

The prior-enhanced soft evidence is then defined as

$$\mu_S(x) \propto P_{\text{prior}}(x) \cdot L_{S_{\text{conv}}}(x) \cdot L_{S_{\text{mul}}}(x), \tag{6}$$

where $L_{S_{\text{mul}}}(x) = 1$ for coefficients without multiplication leakage. FALCON-PARE therefore converts the discrete-Gaussian prior and aggregated leakage likelihoods into full-range input soft evidence, preserving valid tail candidates while reducing coefficient uncertainty.

B. Probabilistic Inference on the NTT Factor Graph

The second stage performs probabilistic inference by representing Falcon's NTT computation as a factor graph. In this graph, variable nodes correspond to NTT input coefficients and intermediate states, while factor nodes encode leakage-likelihood information from side-channel observations and deterministic constraints induced by NTT operations. FALCON-PARE applies BP [23] on this graph to combine the prior-enhanced soft evidence constructed in the first stage.

For any variable node Y associated with a side-channel observation, the corresponding soft evidence is represented by a leakage factor:

$$\eta_Y(y) = \Pr(Y = y \mid O_Y), \tag{7}$$

where y is a candidate value of Y, and O_Y denotes the corresponding observation. For one Falcon NTT butterfly, let Y_0 and Y_1 denote its input variables, and let $\hat{Y}_0$ and $\hat{Y}_1$ denote its output variables. Let y_0, y_1, $\hat{y}_0$, and $\hat{y}_1$ be candidate values assigned to Y_0, Y_1, $\hat{Y}_0$, and $\hat{Y}_1$, respectively. Given the public twiddle factor ω, the butterfly factor [18], [19] is defined as

$$\psi_{\text{bf}}(y_0, y_1, \hat{y}_0, \hat{y}_1) = \begin{cases} 1, & \hat{y}_0 \equiv y_0 + y_1\omega \pmod q, \\ & \hat{y}_1 \equiv y_0 - y_1\omega \pmod q, \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

This factor encodes the deterministic butterfly relation and transfers probabilistic information between the input and output variables. After iterative message passing, FALCON-PARE obtains marginal posterior distributions for the NTT input variables.

C. Entropy-Guided Key Completion via Public-Key Relation

The third stage converts the BP posteriors into complete recovery of the secret polynomials f and g. After BP, FALCON-PARE obtains coefficient-wise marginal posteriors for the NTT input variables, i.e., the modulo-q representatives of the secret coefficients of f and g. Since each valid coefficient in D_{31} has a unique modulo-q representative, the resulting posteriors are equivalently expressed over D_{31}. Let X_i denote the i-th secret coefficient variable, and define $P_i(x) = \Pr(X_i = x \mid \mathcal{O}), x \in D_{31}$, where $\mathcal{O}$ denotes the side-channel observations used in the factor graph. A direct maximum-a-posteriori (MAP) recovery strategy would fix every coefficient independently as $\hat{x}_i = \arg\max_{x \in D_{31}} P_i(x)$. However, complete recovery under this strategy requires all $2n$ coefficient estimates of f and g to be correct simultaneously; any erroneous coefficient prevents exact recovery of the secret key. Rather than requiring all coefficients to be correctly fixed by independent MAP decisions, FALCON-PARE extracts only the most reliable partial information from the BP posteriors. For coefficient i, the posterior entropy is defined as

$$\mathcal{H}_i = -\sum_{x \in D_{31}} P_i(x) \log_2 P_i(x). \tag{9}$$

A smaller $\mathcal{H}_i$ indicates a more concentrated posterior and thus a higher-confidence coefficient estimate. FALCON-PARE ranks the coefficients of each secret polynomial by posterior entropy and selects the lowest-entropy $n/2$ coefficients from each of f and g as the known set. The remaining coefficients are treated as unknowns. Since $h = gf^{-1} \pmod q$, the two secret polynomials satisfy $g - hf \equiv 0 \pmod q$. Let H be the

negacyclic convolution matrix induced by the public key h. The relation can then be represented as

$$\begin{bmatrix} -H & I \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} \equiv 0 \pmod{q}. \tag{10}$$

Define

$$z = \begin{bmatrix} f^T & g^T \end{bmatrix}^T, \qquad M = \begin{bmatrix} -H & I \end{bmatrix}. \tag{11}$$

According to the entropy-guided selection, z is partitioned into a known part z_K and an unknown part z_U; the corresponding columns of M are denoted by M_K and M_U, respectively. Therefore, $M_U z_U + M_K z_K \equiv 0 \pmod{q}$. Before solving, FALCON-PARE checks whether M_U is full rank over $\mathbb{Z}_q$. If M_U is singular, the highest-entropy selected coefficients are replaced by the lowest-entropy unselected coefficients, and the system is reconstructed. Once M_U is full rank, the unknown coefficients are uniquely determined by $z_U \equiv -M_U^{-1} M_K z_K \pmod{q}$. Thus, the entropy-selected coefficients provide reliable partial information, and the public-key relation completes the remaining unknowns. The resulting modular coefficients are converted to their centered representation and output as the recovered secret polynomials.

Overall, FALCON-PARE formulates Falcon secret-key recovery as full-range probabilistic inference followed by public-key relation-assisted completion. It refines all valid coefficient candidates using Falcon's prior distribution and aggregated leakage likelihoods, and propagates the resulting soft evidence through the NTT factor graph. It then uses posterior entropy to select reliable partial information and completes the remaining unknowns via Falcon's public-key relation, yielding the recovered secret polynomials f and g.

IV. Experimental Analysis

A. Experimental Setup

We evaluate FALCON-PARE on the Falcon-512 reference implementation compiled with both `-O0` and `-O3`. The target is an OSR 407 board with an STM32F407IGT6 running at 53 MHz. Power traces are amplified by a PA303 amplifier and sampled at 62 MHz. For each optimization setting, we collect 5,000 traces, of which 4,000 traces form the profiling pool and 1,000 traces are used for testing. All reported recovery results are obtained from independently attacked single testing traces. The measurement setup is shown in Fig. 2. We build HW leakage models [20], [22] for the range-conversion, Montgomery-multiplication, and NTT leakage points described in Section III. To increase the effective number of profiling samples, we aggregate coefficient instances that share the same intermediate type. For range conversion, each profiling trace provides 512 coefficient samples per polynomial, giving $512N_{\text{prof}}$ profiled samples. For NTT butterflies, the upper and lower branches of each layer are modeled separately; each branch provides $256N_{\text{prof}}$ profiled samples. For first-layer Montgomery multiplication, only the active half of the coefficients is used, resulting in $256N_{\text{prof}}$ profiled samples. For each leakage point, 100 points of interest (POIs) are selected only from the profiling traces at each evaluated budget by correlation analysis [22]. Belief propagation is evaluated for at most 20 iterations.

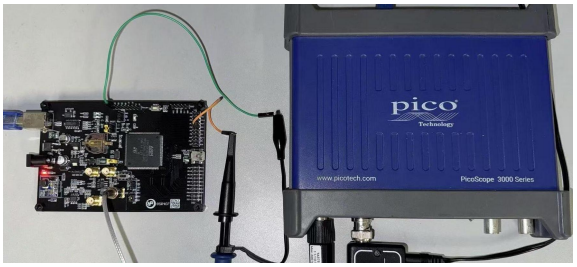


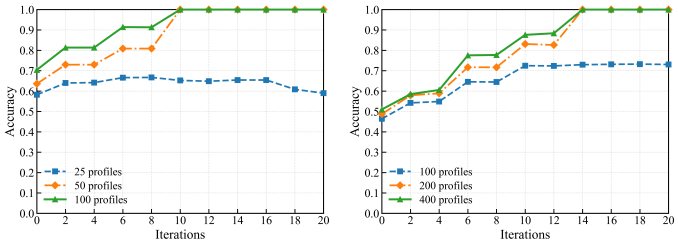
Fig. 2: Experimental platform and measurement setup for FALCON-PARE side-channel evaluation.

B. Recovery Results

The key recovery performance under different compiler settings and profiling-set sizes is summarized in Table II and Fig. 3. We use coefficient-wise recovery accuracy as the metric, defined as the fraction of correctly recovered coefficients among the $2n$ coefficients of f and g. If entropy-guided key completion succeeds, the accuracy is 100.00%; otherwise, each coefficient is recovered by direct MAP decision from the BP posterior. Table II reports the smallest tested profiling budgets that achieve single-trace full key recovery. FALCON-PARE recovers the full key with 40 profiling traces under `-O0` and 120 profiling traces under `-O3`. Fig. 3 shows the recovery accuracy under profiling budgets selected by a doubling strategy. The results show that increasing the profiling budget improves recovery accuracy, and that the `-O3` implementation requires more profiling traces than the `-O0` implementation due to weaker leakage separability. For the subsequent ablation and comparison experiments, we use 50 profiles under `-O0` and 200 profiles under `-O3` as fixed reference profiling budgets for consistent evaluation.

TABLE II: Recovery results of FALCON-PARE.

Setting	Budget	Profiles	BP iter.	Runtime (s)
`-O0`	Min.	40	14	2962.37
	Ref.	50	10	2114.25
`-O3`	Min.	120	18	3966.41
	Ref.	200	14	2961.82



(a) `-O0` (b) `-O3`

Fig. 3: Recovery performance versus profiling size.

C. Ablation Study

We conduct an ablation study to quantify the contribution of the main components of FALCON-PARE. The first con-

TABLE III: Ablation study of FALCON-PARE components.

Configuration	-O0, 50 profiles		-O3, 200 profiles	
	Max. acc.	Acc. gain	Max. acc.	Acc. gain
NTT leakage + Gaussian prior	66.31%	–	50.59%	–
+ Range-conv./Mont. leakage	89.84%	+23.53%	90.63%	+40.04%
+ Key Completion (FALCON-PARE)	**100.00%**	+10.16%	**100.00%**	+9.37%

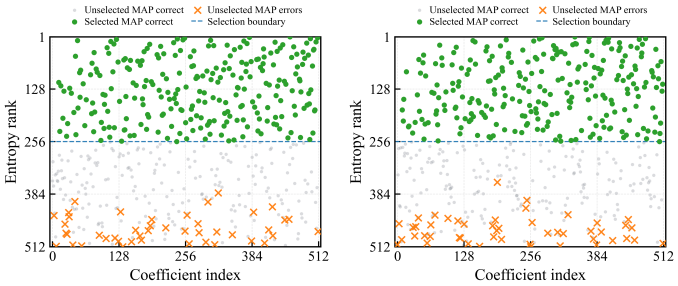
TABLE IV: Reliability of entropy-guided selection.

Setting	Selected-set success	Avg. $\Delta\mathcal{H}$	Avg. Δr
-O0, 50 profiles	10/10	3.121×10^{-5}	118.83
-O3, 200 profiles	10/10	3.100×10^{-3}	184.00

figuration combines the NTT leakage with Falcon's discrete-Gaussian coefficient prior and recovers coefficients by direct MAP decisions. The second configuration further adds the proposed aggregated evidence from range conversion and Montgomery multiplication, while still using direct MAP recovery after BP. The full FALCON-PARE additionally applies entropy-guided key completion. The incremental contribution of each component is reported in Table III. Adding range-conversion and Montgomery-multiplication leakage achieves absolute accuracy gains of 23.53% and 40.04% under -O0 and -O3, respectively. Adding entropy-guided key completion further improves the recovery accuracy by 10.16% and 9.37%. The effect of entropy-guided completion is illustrated in Fig. 4. In the -O3 case with 200 profiling traces at BP iteration 14, direct MAP still leaves errors, whereas the lowest-entropy $n/2$ coefficients selected from each of f and g are all correct. Here, $\Delta\mathcal{H}$ and Δr denote the entropy and rank margins between the selection boundary and the nearest erroneous estimate. Table IV shows that all selected sets are error-free in all 10 independent runs under both settings. Moreover, the number of MAP-correct coefficients exceeds the selected subset, indicating that entropy ranking provides a conservative reliability criterion. Using this reliable subset, FALCON-PARE constructs the linear system induced by the public-key relation, solves the remaining coefficients, and fully recovers both f and g.

D. Comparison with Existing Methods

We first evaluate the recovery feasibility of NTT-PEEL under its restricted coefficient range and contrast it with FALCON-PARE operating over the full coefficient range. NTT-PEEL performs deterministic pruning over the restricted range D_{10}; thus, any true coefficient outside D_{10} is excluded from the initial support and cannot be recovered by subsequent pruning. We generate 100 Falcon-512 key pairs using the reference implementation and count the coefficients outside D_{10} among the 1024 coefficients of f and g. As shown in Table V, all sampled keys are excluded by D_{10}, with 8.90 outside coefficients on average and 16 at maximum. Therefore, NTT-PEEL cannot recover the complete secret key for these sampled instances under its restricted candidate range. In contrast, FALCON-PARE retains the full valid range D_{31} and achieves complete key recovery.



(a) Polynomial f (b) Polynomial g

Fig. 4: Representative entropy-guided coefficient selection.

TABLE V: Comparison with NTT-PEEL.

Method	Range	Excluded keys	Mean outside coeffs.	Full-key recovery
NTT-PEEL [16]	$[-10, 10]$	100/100	8.90	No
FALCON-PARE	$[-31, 31]$	0/100	0.00	Yes

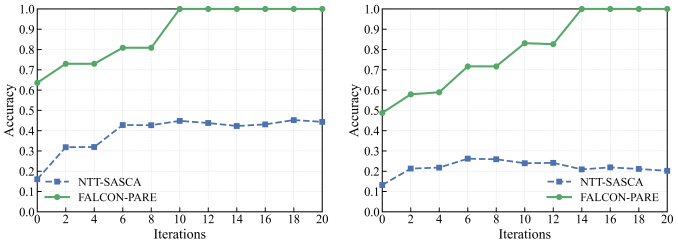
TABLE VI: Comparison with existing NTT-SASCA.

Setting	Method	Prof. traces	Max. acc.	Full-key recovery	B_{cand}
-O0	NTT-SASCA [18], [19]	25	30.08%	No	5161
		50	45.21%	No	4352
		200	86.43%	No	1409
		800	96.00%	No	488
		3200	97.36%	No	338
	FALCON-PARE	25	66.70%	No	2965
		50	**100.00%**	**Yes**	**0**
-O3	NTT-SASCA [18], [19]	100	25.78%	No	5363
		200	26.17%	No	5346
		400	32.13%	No	5060
		800	32.71%	No	5031
		1600	36.82%	No	4819
		3200	39.36%	No	4682
	FALCON-PARE	100	73.24%	No	2485
		200	**100.00%**	**Yes**	**0**

We next compare FALCON-PARE with NTT-SASCA, a baseline from prior SASCA attacks on NTT leakage [18], [19]. The iteration-wise comparison is presented in Fig. 5. Under the reference profiling budgets, NTT-SASCA reaches accuracies of 45.21% under -O0 and 26.17% under -O3. Since FALCON-PARE achieves complete key recovery at the same iterations, the corresponding absolute accuracy gains are 54.79% and 73.83%, respectively. Applying the same key-completion to the baseline, the selected 256 coefficients of f and g contain 159/189 errors under -O0 and 85/65 errors under -O3, respectively; thus, full-key recovery fails.

In addition to the iteration-wise comparison, the maximum coefficient-wise recovery accuracy of NTT-SASCA for each profiling-set size is reported in Table VI. The reported accuracy values are expressed as percentages and rounded to two decimal places. To quantify the gap between coefficient-wise partial recovery and complete key recovery, we estimate the residual candidate space left after direct MAP recovery. Let a denote the coefficient-wise MAP accuracy over the $2n$ coefficients of f and g, and let $m = (1-a)\cdot 2n$ be the number of incorrectly recovered coefficients. Since the erroneous positions and their alternative values are unknown, the bit-size of the residual candidate space is estimated as

$$B_{\text{cand}} = \log_2 \binom{2n}{m} + m \log_2 T,$$



(a) `-O0` (b) `-O3`

Fig. 5: Comparison with the NTT-SASCA baseline under different compiler optimization settings.

where T denotes the number of alternative values for each incorrectly recovered coefficient. For Falcon-512 over the full valid coefficient range D_{31}, we use $T = 62$ as a worst-case full-range estimate, excluding the MAP value already selected by direct recovery. To put these residual spaces in context, we also consider direct enumeration without side-channel information. Enumerating the two secret polynomials over the full valid coefficient range D_{31} gives 63^n candidates for each of f and g. Since full secret-key recovery requires determining both polynomials, the candidate space is $C_{\text{full}} = 63^{2n}$, which corresponds to approximately 2^{6121} candidates for Falcon-512. Under the same profiling budgets, using the best baseline accuracy observed within the tested 20 BP iterations, the baseline leaves residual candidate spaces of approximately 2^{4352} under `-O0` and 2^{5346} under `-O3`. FALCON-PARE achieves complete key recovery under the same budgets and reduces the residual candidate space to one. Even for the best-performing baseline configurations in our tested range, the residual candidate spaces remain approximately 2^{338} under `-O0` and 2^{4682} under `-O3`.

In summary, the experimental results show that FALCON-PARE achieves single-trace full key recovery under both compiler optimization settings while maintaining the full valid coefficient range. The ablation study confirms that aggregated leakage evidence and entropy-guided completion both provide substantial accuracy improvements. Compared with NTT-PEEL, FALCON-PARE avoids the full-key recovery infeasibility caused by restricting the initial candidate domain to D_{10}. Compared with NTT-SASCA, FALCON-PARE achieves substantially higher recovery accuracy under the same profiling budgets and reaches complete key recovery.

V. Conclusion

FALCON-PARE combines full-range prior-enhanced evidence, NTT-factor-graph inference, and public-key-assisted completion for Falcon key recovery. On STM32F407, it achieves single-trace full-key recovery with 40 and 120 profiling traces under `-O0` and `-O3`, respectively, while outperforming the baseline by 54.79 and 73.83 percentage points and reducing the residual candidate space to one.

References

[1] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Foundations of Computer Science (FOCS)*, 1994, pp. 124–134.

[2] NIST, "Post-quantum cryptography standardization," 2026. [Online]. Available: https://csrc.nist.gov/projects/post-quantum-cryptography/post-quantum-cryptography-standardization

[3] NIST, "Post-quantum cryptography: Additional digital signature schemes," 2026. [Online]. Available: https://csrc.nist.gov/projects/pqc-dig-sig

[4] P.-A. Fouque, J. Hoffstein, P. Kirchner, V. Lyubashevsky, T. Pornin, T. Prest, T. Ricosset, G. Seiler, W. Whyte, and Z. Zhang, "Falcon: Fast-Fourier lattice-based compact signatures over NTRU," Falcon specification, version 1.2, 2020. [Online]. Available: https://falcon-sign.info/falcon.pdf

[5] Y. Shao, Y. Liu, Y. Zhou, and M. Shao, "Enhanced key mismatch attacks on lattice-based KEMs: Multi-bit inference and ciphertext generalization," in *Computer Security – ESORICS 2025*, vol. 16054, 2025, pp. 341–358.

[6] P. C. Kocher, "Timing attacks on implementations of Diffie–Hellman, RSA, DSS, and other systems," in *Advances in Cryptology – CRYPTO'96*, 1996, pp. 104–113.

[7] Y. Zhao, W. Cheng, Z. Qiao, Y. Liu, and Y. Zhou, "Rejection matters: Efficient non-profiling side-channel attack on ML-DSA via exploiting public templates," in *Proc. Design, Automation & Test in Europe Conf. (DATE)*, 2026, pp. 1–7.

[8] P. C. Kocher, J. Jaffe, and B. Jun, "Differential power analysis," in *Advances in Cryptology – CRYPTO'99*, 1999, pp. 388–397.

[9] K. Gandolfi, C. Mourtel, and F. Olivier, "Electromagnetic analysis: Concrete results," in *Cryptographic Hardware and Embedded Systems – CHES 2001*, 2001, pp. 251–261.

[10] E. Karabulut and A. Aysu, "FALCON down: Breaking FALCON post-quantum signature scheme through side-channel attacks," in *Proc. 58th ACM/IEEE Design Automation Conf. (DAC)*, 2021, pp. 691–696.

[11] M. Guerreau, A. Martinelli, T. Ricosset, and M. Rossi, "The hidden parallelepiped is back again: Power analysis attacks on Falcon," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2022, no. 3, pp. 141–164, 2022.

[12] S. Zhang, X. Lin, Y. Yu, and W. Wang, "Improved power analysis attacks on Falcon," in *Advances in Cryptology – EUROCRYPT 2023*, Part IV, 2023, pp. 565–595.

[13] X. Lin, S. Zhang, Y. Yu, W. Wang, Q. You, X. Xu, and X. Wang, "Thorough power analysis on Falcon Gaussian samplers and practical countermeasure," in *Public-Key Cryptography – PKC 2025*, 2025, pp. 229–258.

[14] S. Chen, H. Shao, J. Ming, Y. Liu, Y. Gao, and Y. Zhou, "SERP-SCA: A strengthened side-channel attack framework for FALCON signature," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 45, no. 4, pp. 2028–2040, 2026.

[15] J. Qiu and A. Aysu, "SHIFT SNARE: Uncovering secret keys in FALCON via single-trace analysis," *J. Cryptogr. Eng.*, vol. 16, Art. no. 14, 2026, doi: 10.1007/s13389-026-00400-3.

[16] J. Qiu and A. Aysu, "NTT-PEEL: Bit shift side-channel in FALCON's number theoretic transform," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2026, no. 2, pp. 709–735, 2026, doi: 10.46586/tches.v2026.i2.709-735.

[17] R. Primas, P. Pessl, and S. Mangard, "Single-trace side-channel attacks on masked lattice-based encryption," in *Cryptographic Hardware and Embedded Systems – CHES 2017*, 2017, pp. 513–533.

[18] P. Pessl and R. Primas, "More practical single-trace attacks on the number theoretic transform," in *Progress in Cryptology – LATINCRYPT 2019*, 2019, pp. 130–149.

[19] M. Hamburg, J. Hermelink, R. Primas, S. Samardjiska, T. Schamberger, S. Streit, E. Strieder, and C. van Vredendaal, "Chosen ciphertext (k)-trace attacks on masked CCA2 secure Kyber," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 4, pp. 88–113, 2021.

[20] S. Chari, J. R. Rao, and P. Rohatgi, "Template attacks," in *Cryptographic Hardware and Embedded Systems – CHES 2002*, 2002, pp. 13–28.

[21] NIST, "Submission requirements and evaluation criteria for the post-quantum cryptography standardization process," Dec. 2016. [Online]. Available: https://csrc.nist.gov/csrc/media/projects/post-quantum-cryptography/documents/call-for-proposals-final-dec-2016.pdf

[22] E. Brier, C. Clavier, and F. Olivier, "Correlation power analysis with a leakage model," in *Cryptographic Hardware and Embedded Systems – CHES 2004*, 2004, pp. 16–29.

[23] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inf. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.