

Robust Side Channel Analysis with Blocking and Quantification of Leakage

Alicja Kario
hkario@redhat.com
Red Hat Inc.
Czech Republic

George Pantelakis
gpantela@redhat.com
Red Hat Inc.
Czech Republic

Abstract

While an abundance of *Side Channel Analysis* (SCA) methodologies have arisen over the years with impressive theoretical backgrounds, they fail to take into account some important real-world aspects in the analysis. Most of them fail on three key points. First, most assume *Independent and Identically Distributed* (IID) or normally distributed data points in their framework without verifying that the experimental setup indeed meets that requirement. Second, most methods rely on visual aspects or failed statistical tests to define the success or failure of the tests without quantifying the precision of their measurement with formal boundaries. Third, in most cases, the test harness is not completely oblivious to the secret data, remaining completely free of data-dependent branching, conditional formatting, or conditional triggering, which can leak environmental noise and contaminate the data points. We are proposing a new SCA methodology, which we named *Blocked Leakage Assessment & Quantification* (BLAQ), to bridge this gap between the theoretical mathematical models and the real-world imperfect testing environment.

CCS Concepts

• **General and reference** → *Surveys and overviews; Measurement; Evaluation; Experimentation; Estimation; Design; Validation; Verification*; • **Hardware** → *Power and thermal analysis; Timing analysis and sign-off*; • **Software and its engineering** → *Empirical software validation*; • **Security and privacy** → *Cryptanalysis and other attacks; Digital signatures; Public key encryption; Block and stream ciphers; Hash functions and message authentication codes*.

Keywords

side-channel analysis, carry-over effects, nuisance variables, robust statistics, BLAQ, C-BLAQ, I-BLAQ

1 Introduction

The security of cryptographic implementations lies in the assumption that the mathematical model of security properties, like the hardness of the discrete logarithm problem, or unpredictability of the nonce, maps directly to the physical reality. Unfortunately, this assumption ignores the fundamental nature of computing: the algorithms are executed on physical hardware characterized by thermal dynamics, microarchitectural state, and finite, quantized execution times. SCA has emerged as a bridge between that theoretical cryptographic background and real-world software and hardware engineering, demonstrating that an algorithm's *implementation* can leak secrets even when the underlying *algorithm* is mathematically sound.

The maturity of side-channel research, unfortunately, is not reflected in the robustness of the methods it employs. The field is separated between hardware-centric "trace-based" analysis and software-focused timing evaluation, both of which commonly rely on statistical methods that fail to account for the physical realities of the *System Under Test* (SUT). We observed that popular evaluation frameworks frequently violate the core requirements of the statistical tests they use: most notably the IID assumption. Those oversights result in methodologies that are not only prone to errors in evaluations, but also cause them to be unable to provide rigorous quantification of the level of security.

This work attempts to bridge the gap between applied statistics and security engineering by proposing a framework based on *Randomized Block Design* (RBD) and robust non-parametric statistical inference.

1.1 Contributions

- (1) **Methodology Correction:** We note the systemic misuse of IID-reliant statistical tests in the presence of autocorrelated and non-stationary physical data and propose a rigorous experimental design to restore the statistical validity of the methodologies.
- (2) **Quantification:** We introduce methods to establish empirical precision of the methods through calculation of confidence intervals, shifting the perspective from "detecting presence of leaks" to "certifying the absence of leaks above a specific quantum".
- (3) **Validation:** We validate our approach by auditing widely deployed cryptographic libraries like OpenSSL, NSS, GnuTLS, etc. We demonstrate that our methodology uncovers vulnerabilities in different cryptosystems (RSA, ECDSA) across multiple diverse architectures (x86_64, aarch64, ppc64le, s390x).

Ultimately, this research highlights that advancing side-channel research benefits from treating it as a cross-disciplinary study that demands the same statistical rigor as clinical trials or high-energy physics.

2 Statistical Requirements and Modern Microarchitectures

A common, fundamental requirement of statistical methods and tests is that the collected measurements are IID. However, modern CPUs and computing platforms are full of internal state, which in turn has effects on the measurements. Consequently, collecting measurements from a physical SUT will generate data that is heavily autocorrelated and non-stationary, violating both assumptions of the IID model.

2.1 Autocorrelation and Carry-Over Effects

The independence assumption states that the execution time (or power draw) of the operation N is not influenced by operation $N - 1$. In reality, when a microarchitectural state exists and is persistent, it will create carry-over effects where previous executions directly influence the latency of subsequent operations:

- **RAM and Prefetchers:** Modern hardware employs prefetchers that monitor use of L1 and L2 cache, loading data into those caches when they detect a pattern in memory loads. In such a situation, measurement N will inherit the state created by measurement $N - 1$.
- **DRAM Row Buffers:** Since the memory itself has internal caches, if execution $N - 1$ causes a particular row of memory to be loaded, the execution of N will experience a "row hit", causing a reduced latency in memory access.
- **Branch Predictor:** Modern CPUs utilize multiple layers of branch prediction, some keeping very deep state, like Pattern History Tables. That means that if a previous operation alters the weights of those predictors in measurement $N - 1$, it will directly affect the misprediction rate of execution N .

2.2 Non-Stationarity

The Identical Distribution assumption requires that the measurements from the SUT remain stable throughout the duration of the experiment. Physical environment and hardware power management make it impossible:

- **Dynamic Power Management:** CPUs have built-in time-bound power limits (like Intel's PL1/PL2 limits). A processor may be executing with a high-energy turbo state (PL2) for a specific time window (τ) before the hardware will enforce the low-energy turbo state (PL1). That will create a sudden step in the distribution of times of the executed workload.
- **Thermal Throttling and Drift:** As the SUT executes test vectors, the silicon will heat up, increasing leakage current. That will cause either a drift in power usage or cause the dynamic throttling of the CPU frequency, causing non-stationary drift.
- **System Management Interrupts (SMIs):** The motherboard can trigger high-privileged (ring -2) interrupts for hardware control. These SMIs stop the execution of the OS completely, potentially stealing millions of clock cycles invisibly. That will create unpredictable latency spikes that will skew the distribution of measurements.
- **Asymmetric Topologies:** In CPUs that implement the big.LITTLE or P-core/E-core architectures, thread migrating from one core type to a different one will face fundamentally different execution pipelines.
- **DRAM refresh cycles:** Periodic hardware tasks, like DRAM refresh cycles (which will block memory banks entirely), will also introduce non-stationary noise.

2.3 Summary of Effects

Because of those (and likely other) effects, it is practically impossible to fully "flush" a modern CPU to a pristine, stateless baseline between each measurement. Some of those mechanisms may be disabled, or their effect can be minimized, but then the knowledge

about the size of detected leakage is not directly translatable to how easy it would be to use that leakage for an attack. Using IID-dependent tests (like the Welch's t-test) on such systematically stateful data will lead to unpredictable Type-I errors.

3 Side Channel Analysis Overview

The history of SCA research spans over 30 years, with many groundbreaking methods introduced during that time. Unfortunately, the target audience's confidence in these methods—programmers implementing the cryptographic algorithms—remains low [8]. We posit that this is caused by most of the published research in the area of SCA ignoring some of the statistical principles.

We reviewed the literature, trying to find popular methods that would follow three foundational principles of good experimental design [18]:

- **Requirements of the statistical model are satisfied.** For the statistical methods to be robust, their requirements need to be met. If a statistical test or method requires IID data, then use of it with data that is not-IID will cause it to return unpredictable levels of Type-I and Type-II errors.
- **Quantification of uncertainty.** An experiment that specifies that "after 100 000 measurements, the statistical test was still negative", is not directly applicable to other implementations or environments, as the present noise will be vastly different. By using methods that quantify the uncertainty of the result, the experiment becomes directly comparable between different setups.
- **Controlling for the effects of the test setup.** The gold standard for the experimental design is when those administering the treatment (test harness in case of SCA) are not aware of the precise treatment allocated to the subject (SUT): the study is conducted in a double-blind fashion. When the test harness is aware of the class of the Test Vector tested, then the carry-over effects of it will influence the behavior of SUT, skewing the result.

3.1 Previous Research

STA and DTA [15]: While Kocher's foundational timing attack framework theoretically models side-channel leakage, its practical applicability is quite problematic. Specifically, the model assumes that modular multiplication times are strictly independent of one another and environmental noise, completely omitting formal statistical profiling (such as Shapiro-Wilk testing) or microarchitectural verification. In practice, as demonstrated by Brumley and Boneh [4], highly optimized modern implementations utilizing the Chinese Remainder Theorem, Montgomery multiplication, and Karatsuba's algorithm cause these simplified assumptions of uniform, independent operational delays to break down entirely. Furthermore, rather than establishing rigorous confidence intervals or bounded statistical models to quantify the maximum undetectable leakage under noise, Kocher's methodology relies on speculative edge cases—such as theoretically injected "sharp spikes"—to dismiss blinding countermeasures; an analytical approach that Rakotonirina and Köpf [20] note falls fundamentally short in explaining or synthesizing real-world attacks. Finally, the empirical validation of the attack introduces a severe selection bias by arbitrarily discarding "wildly

Table 1: Methodology Comparison

Method	Model Satisfied	Precision Quantified	Double-blind
STA, DTA [15]	X[4]	X[20]	X[6]
TVLA [10]	X[24]	X[24]	X[1]
SPA, DPA [14]	X[17]	X[22]	X[17]
CPA [3]	X[17]	X[10]	X[17]
Template Attacks [5]	X[23]	✓	X[17]
Stochastic Attacks [21]	X[23]	X[10]	X[17]
SEMA [9]	X[17]	X[10]	X[1]
DEMA [19]	X[1]	X[10]	X[1]
CNN [16]	X[13]	X[10]	X[2]
BLAQ [this work]	✓	✓	✓

aberrant samples" that exceed a 1300 μ s threshold. Because raw, real-world timing distributions inherently resist simple statistical comparison, as Dhem et al. [6] discovered, when standard tools failed to yield efficient results, this manual, data-dependent thresholding artificially deflates the empirical variance, violates the requirement of a data-oblivious test harness, and ultimately compromises the statistical validity of the experimental results.

TVLA [10]: Although Goodwill et al. establish a widely adopted framework for side-channel analysis, there are some statistical and experimental flaws. First, the framework's reliance on Welch's t -test across time samples relies on unverified assumptions of independent and identically distributed points. While the authors justify their ± 4.5 threshold by assuming a normal distribution ($p < 10^{-5}$), active hardware countermeasures generate highly non-Gaussian and multi-modal power distributions. As Wang and Tang [24] observe, applying parametric tests to such non-IID distributions mathematically distorts the t -statistic, rendering the resulting inferences highly prone to both false negatives and positives. Second, Goodwill et al. fall into a classic statistical fallacy by declaring that a device achieves a "definitive PASS" if the statistical tests passed (t -statistic remains within $[-4.5, 4.5]$), confusing it with evidence of the absence of leakage. Without establishing confidence intervals, a passing TVLA score is statistically uninformative, failing to bound the maximum size of an undetectable leak that could easily be exploited under a slightly larger sample size N or a faster system. Finally, the fixed-vs-random testing process introduces a severe instrumentation confound. Because physical communication lines toggle heavily during "random" plaintext transfers but remain static during "fixed" transfers, the external test harness generates data-dependent electromagnetic and power changes. Unless the test harness is strictly data-oblivious and optically isolated, near-field probes inevitably couple with these external I/O line emanations, which, as Agrawal et al. [1] warn, trigger false-positive failures that do not originate from the target's internal cryptographic functions.

SPA, DPA [14]: Kocher, Jaffe, and Jun revolutionized side-channel cryptanalysis with Differential Power Analysis (DPA), but their theoretical framework is constrained by physical limitations. First, the mathematical justification for the flattening of the differential trace relies on the unverified assumption that background measurements and hardware noise behave as independent and identically distributed variables. In actual silicon architectures, however, power

consumption across adjacent clock cycles features heavy temporal autocorrelation and structural dependencies; as Mangard, Oswald, and Popp [17] argue, these noise components are often correlated with the processed data. Fortunately, the chosen independent-samples difference-of-means test mathematically aligns with the experimental design, as the independent, unpaired trace measurements are partitioned into mutually exclusive, independent subsets. Nevertheless, the methodology fails to quantify the precision of its measurements. Rather than establishing strict confidence intervals or exact error boundaries to evaluate security margins, the authors rely on asymptotic trends toward infinity and heuristic visual confirmation of peak, a limitation that Standaert, Malkin, and Yung [22] highlight as a barrier to mathematically quantifying the exact remaining security margins of the SUT. Finally, the physical experimental setup introduces an instrumentation confound. As Mangard, Oswald, and Popp [17] warn, because the power consumption of the communication interface can easily overshadow that of the internal cryptographic function, the absence of strict optical isolation or data-oblivious formatting risks leaking external I/O interference into the measurements.

CPA [3]: Brier, Clavier, and Olivier significantly advanced side-channel analysis by introducing Correlation Power Analysis, but their validation methodology has a few flaws. First, the experimental validation of their leakage model relies on a test program executed exactly 256 times while stepping the target data byte sequentially from 0 to 255. This deterministic execution sequence directly violates the IID assumption, leaving the Pearson correlation coefficient calculations vulnerable to slow drift, such as environmental temperature variations or power fluctuations. As Mangard, Oswald, and Popp [17] argue, such slow changes in ambient conditions lead to a drift in power consumption over time that can introduce artificial variance. Second, the authors define evaluation success through a binary "distinguisher" lens, where security is assessed by whether the correct candidate key maximizes the correlation coefficient, while entirely failing to establish formal confidence intervals or standard error bounds. As Goodwill et al. [10] observe, key-recovery-centric metrics are not suited for validating cryptographic safety because a failed attack does not guarantee security and fails to provide a formal statistical bound on leak under heavy noise. Finally, as with SPA and DPA, the physical experimental setup fails to ensure that the external test harness is data-oblivious.

The same cautions regarding instrumentation raised by Mangard, Oswald, and Popp [17] apply here.

Template Attacks [5]: While Chari, Rao, and Rohatgi introduced Template Attacks as a theoretically scary side-channel threat, the practical execution of their framework needs to be adjusted. First, the template framework’s mathematical principle stands on the assumption that side-channel noise profiles natively approximate a multivariate Gaussian distribution, which the authors adopt as a practical shortcut rather than validating via formal multivariate normality tests on actual physical data. However, empirical side-channel power datasets frequently don’t follow this foundational Gaussian assumption. Despite this unverified assumption, the methodology successfully avoids the fallacy of claiming absolute security based on a failed test; instead, the authors rigorously quantify the mathematical precision of their classification approach by computing formal error probabilities for binary hypotheses using the complementary error function. This quantification provides statistical bounds on classification accuracy. Nevertheless, the physical test harness fails to remain data-oblivious. In their experimental setup, a standard PC directly triggers the digital oscilloscope immediately before sending execution commands to the target smart card. Under this triggering mechanism, data-dependent bit transitions on the physical communication lines generate significant power fluctuations across shared power paths. The same cautions regarding instrumentation raised by Mangard, Oswald, and Popp [17] apply here.

Stochastic Attacks [21]: Schindler, Lemke, and Paar introduced a mathematically elegant stochastic model for differential power analysis, but their methodology is constrained by real-world misconceptions. First, the model directly introduces IID variables as a foundation for its linear regression and maximum likelihood calculations, assuming the noise vector is normally distributed. However, the authors omit tests to verify normality on the physical traces. As Veyrat-Charvillon and Standaert [23] point out, parametric side-channel models like the stochastic attack suffer from structural mis-specification when physical noise exhibits multi-modality, asymmetry, or non-linear coupling, risking false negatives if these assumptions are left unverified. Second, while the paper derives analytical formulas to estimate theoretical success rates under ideal conditions, it fails to quantify the precision of its measurements using confidence intervals or standard error bounds. Instead, it relies strictly on how often the correct subkey achieves the maximum value. As Goodwill et al. [10] notice, key-recovery-centric metrics are fundamentally ill-suited for validating the safety of an implementation because a failed attack does not guarantee security and fails to provide a formal confidence interval or bound on the maximum size of a hidden, undetectable leak under heavy physical noise. Finally, the methodology provides no assurance that the physical test harness is data-oblivious, completely omitting details about the host interface, wiring, or trigger mechanisms used on the SUT during the collection phase. The same cautions regarding instrumentation raised by Mangard, Oswald, and Popp [17] apply here.

SEMA, DEMA [19]: Quisquater and Samyde significantly expanded the landscape of side-channel analysis by porting power analysis to the electromagnetic spectrum; however, their framework

is limited by unverified assumptions and noise-contaminated measurements. First, by porting the difference-of-means architecture as it is to the electromagnetic domain, the authors implicitly assume that high-frequency electromagnetic fields and localized circuit noise natively behave as IID normal variables, omitting checks for verifying it. As Agrawal et al. [1] demonstrate, EM emanations do not mirror additive white Gaussian noise; instead, they are highly dependent on spatial orientation and component geometries, and they feature significant spatial and temporal autocorrelation that can lead to severe misinterpretations of the underlying leakage. Second, the methodology fails to quantify measurement precision or establish rigorous statistical bounds on the effectiveness of its proposed hardware countermeasure. The authors evaluate these defenses through a purely qualitative lens, noting simply that DEMA can be generalized to "defeat" them, rather than calculating formal confidence intervals or error boundaries over finite trace samples. As Goodwill et al. [10] point out, such qualitative approaches relying on visual contrast or peak presence fail to provide a formal confidence interval or a mathematical bound on the maximum size of a hidden, undetectable leak, making a failed attack completely uninformative of a device’s actual security boundaries. Finally, because the physical experimental design operates in an exploratory laboratory setup without a data-oblivious transmission interface or double-blind isolation, it introduces a major instrumentation flaw. Highly sensitive near-field probes act as antennas that inevitably couple with the fields generated by external noise that toggle heavily based on the processed data. As Agrawal et al. [1] warn, because emanations of the external bonding wires and I/O line emanations are orders of magnitude larger than the microscopic fields of internal logic gates, the lack of strict data-oblivious shielding causes the probe to capture this external harness radiation, resulting in a false validation where cross-contamination is misattributed to the target’s internal execution logic.

DEMA, CEMA [9]: Although Gandolfi, Mourtel, and Olivier [9] pioneered the expansion of side-channel cryptanalysis by introducing Differential Electromagnetic Analysis and Correlation Electromagnetic Analysis, their experimental methodology has some real-world constraints. First, by mapping classic differential and correlation techniques directly onto electromagnetic emanations, the methodology assumes that background architectural noise and high-frequency measurement errors come from an IID normal distribution. The authors, however, do not perform any formal statistical test to verify such an assumption under real-world circuit conditions. As Mangard, Oswald, and Popp [17] argue, real-world noise components in hardware cryptosystems frequently exhibit strong temporal dependencies across clock cycles or directly correlate with processed data. Second, rather than constructing formal confidence intervals or statistical error bounds to mathematically define the precision of their measurements, the authors evaluate security margins through the visual presence of high-contrast spikes on an oscilloscope screen. This approach is not suited for safety validation; as Goodwill et al. [10] observe, key-recovery attacks of this nature are inherently optimized for finding keys rather than proving security, and a failed attack cannot bound the maximum size of a hidden, undetectable leak under physical noise. Finally, the authors attempt to validate their setup using broad macro-differential

signals, contrasting additions of 00h and FFh inputs. However, without data-oblivious transmission or strict optical isolation in the test harness, the EM probe couples with the fields generated by the external serial communication and trigger lines. According to Agrawal et al. [1], because these external interferences are orders of magnitude larger than the microscopic fields of internal logic gates, early macro-differential experiments can show false-positive validation where harness contamination is misattributed to the internal cryptographic function.

CNN [16]: Maghrebi, Portigliatti, and Prouff were the first to apply deep learning to side-channel analysis; however, their evaluation framework is constrained by unverified statistical assumptions, unquantified precision metrics, and significant experimental biases. First, the framework’s multi-trace product formula—used to compute global prediction scores—implicitly assumes that side-channel leakages across separate, consecutive executions are perfectly IID. However, collecting training traces sequentially by grouping 1,000 traces per sensitive target state $Z = z$ violates this assumption by introducing systemic temporal environmental drift and hardware carry-over effects. As Kim et al. [13] point out, deep neural networks are highly opportunistic learners that, when trained on sequentially clustered data, easily learn the physical setup’s temporal or thermal drift rather than genuine data-dependent execution characteristics. Second, the authors evaluate their models strictly through empirical key-recovery success markers, plotting average Guessing Entropy curves down to 1 without establishing standard error bands, confidence intervals, or mathematical bounds on the maximum undetectable leakage. As Goodwill et al. [10] argue, such key-recovery metrics are poorly suited for validating the safety of an implementation because a failed attack does not prove security, leaving the true resilience of the hardware unquantified under finite trace sets. Finally, validating the deep learning models on the exact same physical ChipWhisperer platform used for training introduces a severe single-device profiling bias. Without a data-oblivious test harness to prevent serial I/O transitions from bleeding power noise into the traces, the model’s massive parameter capacity easily overfits to static physical anomalies. As Benadjila et al. [2] warn, to prove that a neural network has learned generalizable cryptographic leakage rather than memorizing the unique DC offset, gain, or silicon signatures of a single chip, the model must be trained on one physical device and validated on another.

4 Blocked Leakage Assessment & Quantification

To address the limitations of previous methods, we introduce a novel approach to SCA that prioritizes robustness and takes into account the reality in the behaviors of physical SUTs.

As others have done previously, we randomize the order of executed test vectors to help control for carry-over effects. Additionally, to control for nuisance variables, we perform the analysis using within-group comparisons[7], not independent-samples tests. In statistics literature, that is called a RBD. An equally valid approach is to use Latin Squares [18], but since the effects we are looking for are generally very small, the number of measurements necessary is very large, so such careful design is generally not necessary; that being said, if the cost of procuring the measurements is high (e.g.

the experiment is performed manually by a person triggering an oscilloscope), then Latin Squares approach should be considered.

The other goal of the method is to show the *absence* of leakage. The primary mechanism through which we achieve this is by estimating the uncertainty of the measurements: bootstrapped confidence intervals of pairwise differences of different estimators: mean, median, and trimmed mean. If all the calculated means are close to zero, the confidence intervals span over zero, and the size of the confidence interval is smaller than the smallest quantum of the target system (e.g. 1 CPU clock cycle), then we can say that the precision of the measurement is tighter than the system’s ability to leak information.

Calculation of the confidence intervals also provides information about how noisy the SUT and experimental setup is. A system with low *Signal to Noise Ratio* (SNR) will have bigger confidence intervals than a system with high SNR if both were used to collect the same number of measurements.

The last thing that is universal for the methods in BLAQ family is the use of a simple, blind test harness. The Test Vectors are prepared by a separate application or script, in random order, and then written to a file. A typical test harness will then read a constant number of bytes, without interpreting their contents, starting the measurement right before passing them to the SUT, and finishing the measurement right after receiving the result of the operation. In cryptography, that usually will be a method that performs the high-level operation: an RSA decryption, ECDSA signature, ML-KEM decapsulation, etc. But it can also be used for testing the building blocks of the algorithms: from singular hardware instructions, through individual library functions (like arbitrary precision integer operations), to algorithms that are integrated to bigger cryptosystems (like testing the SHA-2 hash function to be used as part of the bigger signing operation). Then, such collected measurements are assigned to appropriate classes, the measurements are put into blocks, and the appropriate statistical tests are executed.

Based on the result of blocking, we separate the analysis into two distinct subtypes: If all measurement blocks contain all classes of data, then it is a *Complete Block Design* (CBD); if some measurement blocks are missing any class of data, then it is an *Incomplete Block Design* (IBD).

4.1 Complete Block Design

The BLAQ variant operating on CBD, or shortly C-BLAQ, is most often applicable when the SUT normally processes attacker-controlled data: RSA decryption, Diffie-Hellman key agreement, ML-KEM decapsulation, etc. Test vectors for such an experiment can be prepared by selecting a few types of representative inputs (classes), and then for every block to be tested, generating a new test vector of those inputs. For example, for RSA PKCS#1 v1.5 decryption, test vectors that decrypt to messages of different lengths, different errors in padding, and ones that decrypt to plaintexts with many most significant bytes being zero would be different kinds of inputs (or classes of inputs).

By duplicating one of the classes (having two classes that always decrypt to messages of the same size), it is possible to look for test harness effects: a statistically significant difference between them can only be explained by a non-blind test harness. Then the

first of those two classes is the baseline, while the other becomes a control (a canary in the coal mine) against bias introduced by the test harness.

To analyze such data, the two primary tools are the Friedman test and bootstrapping of central tendencies of the pairwise differences: mean, median, and trimmed means.

The Friedman test is used as the deciding factor to declare the presence of leakage: if the p -value is small, that suggests the presence of leakage. In practical implementations we generally used $\alpha = 10^{-5}$ as the cut-off point. In case of a negative result from the Friedman test (large p -value), we turn to the size of the *Confidence Interval* (CI) of the central tendencies. If CI are smaller than the smallest quantum of the system (e.g. 1 CPU clock cycle), then the method can declare *absence of leakage*.

For post-hoc analysis of a positive Friedman test, it is also possible to look at pairwise tests between the different classes, to see which specific class is the outlying one. There, a paired t-test, sign-test, or Wilcoxon signed-rank test can be used to find the classes that likely caused the positive result of the Friedman test.

While we have used the Friedman test, applying repeated measures ANOVA on ranks would also be correct; unfortunately, we have not found an implementation that would be fast enough to handle the data sets that we routinely collect.

4.2 Incomplete Block Design

The BLAQ variant operating on IBD, or shortly I-BLAQ, is most often applicable when the evaluator does not have a good control over the algorithm inputs, so the distribution of the secret variable is a result of chance, not deliberate choice of inputs.

For example, in the ECDSA signing operation, the nonce is generated completely randomly. But, if the evaluator knows the private key used to create the signatures, then they are able to calculate the internally selected nonce. Individual bits of such nonces (provided that the used random number generator is working correctly) will be truly independent of each other. That means that the populations of the Hamming weights of those nonces will follow a binomial distribution. For one of the most commonly used ECDSA curves, the P-256, the median weight will be 128, but more extreme weights (like 64 or 192) will be very rare. By blocking the measurements in groups of around 10, and discarding within-block duplicates at random, we can create unbiased blocks that will form an IBD that controls for autocorrelation and nuisance variables in the experiment. At the same time, because the variable that decides classification (and thus presence of a measurement in the block) is a result of independent sampling (the basic assumption of security of Cryptographically Secure Random Bit Generator) the missing values are not correlated with measurements.

Such data can then be analyzed with the Skillings-Mack test, a general Friedman-type statistic that can be used for arbitrary missing-data structure. Similarly to C-BLAQ, the small p -value indicates the presence of a side-channel. In case the p -value is large, it is necessary to look at the CIs of the computed pairwise differences between the different classes. If the CIs are sufficiently small, then the method can declare *absence of leakage*.

For post-hoc analysis, it is similarly possible to calculate pairwise tests between classes, looking for the outliers in the classes, or a

trend between them (for example: checking if the classes representing big Hamming weights do not take consistently longer time to process than classes with small Hamming weights).

5 Validation of the method

The new methodology that we are proposing helped us uncover several CVEs throughout our research.

Starting with RSA decryption analysis, we have found 47 issues or CVEs associated with 35+ TLS implementations, such as .Net (issue-97866), Apple corecrypto (CVE-2024-23218), BouncyCastle (CVE-2024-30171), cjose (PR-22), cryptlib (CVE-2024-0202), Crypto++ (issue-1247), Erlang/OTP (commit-67aa13), GnuTLS/TLS level (CVE-2023-0361, CVE-2023-5981, CVE-2024-0553), Gocrypto/rsa (CVE-2023-45287), GSKit (CVE-2023-32342), Intel QuickAssist Technology Engine for OpenSSL (CVE-2024-33617, CVE-2024-28885, CVE-2024-31074), iperf3 (CVE-2024-26306), Java (CVE-2024-20952, CVE-2025-21587), jsrsasn (CVE-2024-21484), libgcrypt (CVE-2024-2236), Linux Kernel (CVE-2023-6240), M2Crypto (CVE-2020-25657, CVE-2023-50781), Mbed TLS (CVE-2024-23170), node-rsa (issue-230), Node.js (CVE-2023-46809), NSS/TLS level (CVE-2023-4421, CVE-2023-5388), openssl (CVE-2024-0914), OpenSC (CVE-2023-5992, CVE-2024-29995), OpenSSL/TLS level (CVE-2022-4304), OpenSSL/API level (PR-13817), OpenSSL-ibmca (ChangeLog), @peculiar/webcrypto (issue-66), perl Crypt-OpenSSL-RSA (CVE-2024-2467), pkcs11-provider (CVE-2023-6258), pyca/cryptography (CVE-2020-25659, CVE-2023-50782), PyCryptodome (CVE-2023-52323), python-rsa (CVE-2020-25658), Ruby (CVE-2025-0306), RustCrypto (CVE-2023-49092), Rust OpenSSL (CVE-2024-3296), s2n (advisory), WolfSSL (CVE-2023-6935), and xmlsec (issue-775).

Continuing with the ECDSA signature analysis, we have found 10 issues or CVEs associated with 6+ TLS implementations, such as OpenSSL (CVE-2024-13176, CVE-2025-27587, issue-24252, issue-24253, issue-24254), NSS (CVE-2023-6135), GnuTLS (CVE-2024-28834), ibmca-provider (CVE-2024-22340), python-ecdsa (CVE-2024-23342), libgcrypt (issue-T7519), and Golang, in which no leakage above 1ns was discovered. We also checked an existing CVE-2025-22866 with the I-BLAQ methodology to verify that the severity assignment was correct and that the fix was working, and no leakage bigger than a single CPU cycle was noticed. This is still a work in progress, and we have many more implementations to check.

Finally, in ML-KEM decapsulation, we have tested implementations of ML-KEM in OpenSSL and NSS, and have verified them to be non-leaky to the resolution of 1 ns or better across all implemented parameter sets and across four different hardware architectures (x86_64, aarch64, ppc64le, and s390x). More testing will be taking place as more implementations of ML-KEM arise.

To learn more about the tools used, see Appendix A.

6 Conclusion

This work shows systemic issues in SCA research of cryptographic implementations. The pattern is present both in the research of timing side-channels (most common in software implementations) and in physical side-channels (being of most interest in hardware implementations). While those issues do not invalidate the previous results, they do limit the reproducibility of those methods and their

usefulness for software development and in Continuous Integration systems.

By following statistical best practices (ensuring that requirements of the statistical methods are met, quantifying uncertainty, performing the experiments in a double-blind fashion), we think that the field can make a huge leap forward in the detection of side-channel leakage.

In other words, for the theoretical model to be valid in practice it needs to assume autocorrelation between measurements, for both the underlying random variable and the modeled noise: non-normal distribution, non-stationarity (temporal noise), and heteroskedasticity. In such environments, the design of the experiment is vastly more important than the statistical tests; it is not possible to compensate for a bad experimental setup with a complicated model. The resolution of the measurement should also be taken into account, as timing information usually is available in modern platforms at lower resolution than the maximum clock speed of the CPU (proper experimental setup will handle it through dithering effects), but the statistical tools used need to work with values that are in practice quantized (not continuous), possibly at a resolution higher than the measured signal.

We were also able to show that those best practices are achievable in practice, and allow for both finding new side-channels and verifying that they have been removed completely.

Acknowledgments

This work is supported by the European Commission Horizon Europe program under grant number: 101225691 (<https://doi.org/10.3030/101225691>) (QARC project).

We would like to thank Yuki Yamori, Kate Kočická, and Lumi Yamori for the review of an early draft of the document and help with choosing a name of the algorithm.

Ethics and Privacy Statement

When we detected side-channel leakage in the listed implementations, we have followed responsible disclosure practices: contacted the authors privately or followed the processes documented in the project's respective security policies.

We do not expect that this research will have significant societal impact from publication, as presence and difficulty in fixing side-channel leakage is a well-known problem in cryptography.

All code necessary to reproduce the results discussed in this work are available under Open Source Licenses and presented in Appendix A.

References

- [1] Dakshi Agrawal, Bruce Archambeault, Josyula R. Rao, and Pankaj Rohatgi. 2002. The EM Side-Channel(s). In *Cryptographic Hardware and Embedded Systems — CHES 2002 (Lecture Notes in Computer Science, Vol. 2523)*. Springer-Verlag, Berlin, Heidelberg, 29–45. doi:10.1007/3-540-36400-5_4
- [2] Ryad Benadjila, Emmanuel Prouff, Rémi Strullu, Eleonora Cagli, and Cécile Dumas. 2018. Study of Deep Learning Techniques for Side-Channel Analysis and Introduction to ASCAD Database. Cryptology ePrint Archive, Report 2018/053, 42 pages. <https://eprint.iacr.org/2018/053>
- [3] Eric Brier, Christophe Clavier, and Francis Olivier. 2004. Correlation Power Analysis with a Leakage Model. In *Cryptographic Hardware and Embedded Systems — CHES 2004 (Lecture Notes in Computer Science, Vol. 3156)*. Springer-Verlag, Berlin, Heidelberg, 16–29. doi:10.1007/978-3-540-28632-5_2
- [4] David Brumley and Dan Boneh. 2005. Remote Timing Attacks are Practical. *Computer Networks* 48, 5 (Aug. 2005), 701–716. doi:10.1016/j.comnet.2005.01.010
- [5] Suresh Chari, Josyula R. Rao, and Pankaj Rohatgi. 2002. Template Attacks. In *Cryptographic Hardware and Embedded Systems — CHES 2002 (Lecture Notes in Computer Science, Vol. 2523)*. Springer-Verlag, Berlin, Heidelberg, 13–28. doi:10.1007/3-540-36400-5_3
- [6] Jean-Francois Dhem, Francois Koeune, Philippe-Alexandre Leroux, Patrick Mestre, Jean-Jacques Quisquater, and Jean-Louis Willems. 1998. A Practical Implementation of the Timing Attack. In *Smart Card Research and Applications, Third International Conference, CARDIS '98 (Lecture Notes in Computer Science, Vol. 1820)*. Springer-Verlag, Berlin, Heidelberg, 167–182. doi:10.1007/10721099_15
- [7] A. Adam Ding, Cong Chen, and Thomas Eisenbarth. 2015. Simpler, Faster, and More Robust T-test Based Leakage Detection. Cryptology ePrint Archive, Paper 2015/1215. <https://eprint.iacr.org/2015/1215>
- [8] Marcel Fourné, Daniel De Almeida Braga, Jan Jancar, Mohamed Sabt, Peter Schwabe, Gilles Barthe, Pierre-Alain Fouque, and Yasemin Acar. 2024. “These results must be false”: A usability evaluation of constant-time analysis tools. In *Proceedings of the 33rd USENIX Security Symposium*. USENIX Association, 6705–6722.
- [9] Karine Gandolfi, Christophe Mourtlet, and Francis Olivier. 2001. Electromagnetic Analysis: Concrete Results. In *Cryptographic Hardware and Embedded Systems — CHES 2001 (Lecture Notes in Computer Science, Vol. 2162)*. Springer-Verlag, Berlin, Heidelberg, 251–261. doi:10.1007/3-540-44709-1_21
- [10] Gilbert Goodwill, Benjamin Jun, Josh Jaffe, and Pankaj Rohatgi. 2011. A Testing Methodology for Side-Channel Resistance Self-Assessment. In *Non-Invasive Attack Testing Workshop*. National Institute of Standards and Technology, Gaithersburg, MD, USA, 17 pages. https://csrc.nist.gov/csrc/media/events/non-invasive-attack-testing-workshop/documents/08_goodwill.pdf
- [11] Hubert Kario. 2023. Out of the Box Testing. Cryptology ePrint Archive, Paper 2023/1441. <https://eprint.iacr.org/2023/1441>
- [12] Hubert Kario. 2024. Everlasting ROBOT: The Marvin Attack. In *Computer Security — ESORICS 2023*, Gene Tsudik, Mauro Conti, Kaitai Liang, and Georgios Smaragdakis (Eds.). Springer Nature Switzerland, Cham, 243–262.
- [13] Jaehun Kim, Stjepan Picek, Annelie Heuser, Shivam Bhasin, and Alan Hanjalic. 2019. Make Some Noise: Unleashing the Power of Convolutional Neural Networks for Profiled Side-Channel Analysis. *IACR Transactions on Cryptographic Hardware and Embedded Systems* 2019, 3 (May 2019), 148–179. doi:10.46586/tches.v2019.i3.148-179
- [14] Paul Kocher, Joshua Jaffe, and Benjamin Jun. 1999. Differential Power Analysis. In *Advances in Cryptology — CRYPTO '99 (Lecture Notes in Computer Science, Vol. 1666)*. Springer-Verlag, Berlin, Heidelberg, 388–397. doi:10.1007/3-540-48405-1_25
- [15] Paul C. Kocher. 1996. Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems. In *Advances in Cryptology — CRYPTO '96 (Lecture Notes in Computer Science, Vol. 1109)*. Springer-Verlag, Berlin, Heidelberg, 104–113. doi:10.1007/3-540-68697-5_9
- [16] Houssein Maghrebi, Thibault Portigliatti, and Emmanuel Prouff. 2016. Breaking Cryptographic Implementations Using Deep Learning Techniques. In *Security, Privacy, and Applied Cryptography Engineering — SPACE 2016 (Lecture Notes in Computer Science, Vol. 10076)*. Springer-Verlag, Berlin, Heidelberg, 3–26. doi:10.1007/978-3-319-49445-6_1
- [17] Stefan Mangard, Elisabeth Oswald, and Thomas Popp. 2007. *Power Analysis Attacks: Revealing the Secrets of Smart Cards*. Springer-Verlag, New York, NY, USA. <https://link.springer.com/book/10.1007/978-0-387-38162-6>
- [18] D.C. Montgomery. 2017. *Design and Analysis of Experiments*. John Wiley & Sons, Incorporated.
- [19] Jean-Jacques Quisquater and David Samyde. 2001. ElectroMagnetic Analysis (EMA): Measures and Counter-measures for Smart Cards. In *Smart Card Programming and Security (Lecture Notes in Computer Science, Vol. 2200)*. Springer-Verlag, Berlin, Heidelberg, 200–210. doi:10.1007/3-540-45418-7_17
- [20] Itsaka Rakotonirina and Boris K'opf. 2019. On Aggregation of Information in Timing Attacks. In *4th IEEE European Symposium on Security and Privacy, EuroS&P 2019*. IEEE, Piscataway, NJ, USA, 377–392. doi:10.1109/EuroS&P.2019.00035
- [21] Werner Schindler, Kerstin Lemke, and Christof Paar. 2005. A Stochastic Model for Differential Power Analysis. In *Cryptographic Hardware and Embedded Systems — CHES 2005 (Lecture Notes in Computer Science, Vol. 3728)*. Springer-Verlag, Berlin, Heidelberg, 30–46. doi:10.1007/11545262_3
- [22] François-Xavier Standaert, Tal Malkin, and Moti Yung. 2009. Towards Security Evaluation of Cryptographic Hardware Devices. In *Cryptographic Hardware and Embedded Systems — CHES 2009 (Lecture Notes in Computer Science, Vol. 5747)*. Springer-Verlag, Berlin, Heidelberg, 81–97. doi:10.1007/978-3-642-04138-9_7
- [23] Nicolas Veyrat-Charvillon and François-Xavier Standaert. 2009. Mutual Information Analysis: How, When and Why?. In *Cryptographic Hardware and Embedded Systems — CHES 2009 (Lecture Notes in Computer Science, Vol. 5747)*. Springer-Verlag, Berlin, Heidelberg, 429–443. doi:10.1007/978-3-642-04138-9_30
- [24] Yaru Wang and Ming Tang. 2023. A Survey of Side-Channel Leakage Assessment. *Electronics* 12, 16, Article 3461 (Aug. 2023), 22 pages. doi:10.3390/electronics12163461

A Research Tools

Throughout the research process, we developed a number of tools to help us.

First, we have the harness that we used to gather the data, those include the [Marvin Toolkit](#) for the RSA decryption analysis, [Minerva Toolkit](#) for the ECDSA signature analysis, and [ML-KEM SCT Toolkit](#) for the ML-KEM decapsulation analysis.

After gathering the data, we analyzed it with statistical tooling implemented in the [tlsfuzzer framework](#).

All the tools work on most supported parameters, tested against OpenSSL, NSS, and GnuTLS (except ML-KEM SCT Toolkit yet), for Intel 64 (x86_64), IBM's Enterprise Systems Architecture/390 64-bit (s390x), PowerPC 64-bit (ppc64le), ARM Architecture 64-bit (aarch64).

B Continuation of work

This work is a continuation of previous work by the author [12], [11].