Design and Validation Strategies for Obtaining Assurance in Countermeasures to Power Analysis and Related Attacks*

Paul Kocher

Cryptography Research, Inc.
575 Market Street, 21st Floor
San Francisco, CA 94105, USA.
http://www.cryptography.com
E-mail: paul@cryptography.com.

Abstract. To be secure, tamper resistant cryptographic devices must be protected against DPA and related attacks. Independent testing processes are essential for validating the presence and effectiveness of these countermeasures. Testing methodologies for power analysis vulnerabilities can yield varying degrees of assurance as to the security of the device under test. While insecurity can be demonstrated conclusively, evidence of security is more open-ended. Confidence in a security evaluation depends on many factors including the comprehensiveness of the evaluation, the skill of the evaluator, the nature of the device's design, and the difficulty of exploiting any identified vulnerabilities. This paper reviews testing strategies for power analysis and related attacks, including black box and clear box methods. The paper also examines how appropriate design architectures and evaluation approaches can be combined to yield the strongest evidence of a device's security.

Keywords: differential power analysis, DPA, validation, security evaluation, testing, assurance

1 Background

This paper considers the challenges involved in validating whether cryptographic devices are vulnerable to power analysis attacks. It is assumed that readers are familiar with simple power analysis (SPA) and differential power analysis (DPA), including variants such as high-order DPA. (For an introduction to SPA and DPA, see [1].)

In general, the goal of a security evaluation is to assess the likelihood that a device or system meets some defined security objectives. This process involves analyzing both the design and the process used to produce it. For example, a product that has been carefully designed and tested by experienced experts has

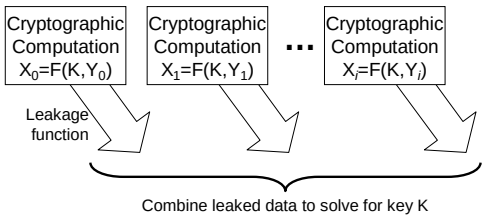
* This paper was prepared for the NIST Physical Security Workshop, September 26-29, 2005.

higher assurance (i.e., a lower probability of failing) than one haphazardly put together by a novice, even if neither has any known defects. While validation efforts may uncover security defects (in which case there is typically zero assurance of security), this paper focuses on the question of how to attain assurance in the security of a device that does not have any known defects.

DPA testing presents special testing challenges. Conventional security testing efforts tend to focus on logical characteristics of digital systems and focus on individual layers in a design. In contrast, power analysis attacks combine several disciplines, including transistor physics, cryptanalysis, digital circuit design, statistics, software development, data acquisition, and analog signal processing. The reason for this is that the security issues underlying DPA cross multiple layers of abstraction. At the lowest level, a transistor's power consumption depends on its switching activity and state. A circuit's power consumption depends on the activity of all its transistors and other elements. Ultimately, the power consumption of a device observed during a cryptographic operation is a combination of myriad details related to its software and hardware implementations.

DPA attacks use statistical techniques to turn this complexity to the advantage of the adversary. In particular, attackers can determine secret keys by detecting minute correlations in power consumption measurements, even if the effects of interest are dominated by unrelated "noise" in the measurements. The goal of a validation effort is to assess whether there could be any compromising signals buried in this noise.

From an evaluator's perspective, variations in a device's power consumption can be thought of as a covert channel that provides some (perhaps very complex and noisy) *leakage function* of the device's state. Variations in actual power measurements can include both entropy (e.g., due to unknown processes or random measurement errors) as well as useful leaked information. The leaked information can include correlations to inputs, outputs, state transitions, intermediates, and other elements of the device state. The attacker's goal is to find a way to use the leaked information to find the secret key.



The evaluator has the task of assessing whether information is being leaked that could compromise keys or other secrets. The feasibility and reliability of this assessment depends on many factors, including the evaluation process and the device design. In general, most devices are too complex to completely model

the applicable leakage functions, but evaluations can still provide valuable information as to how secure devices are and the risk of compromise.

2 Black Box Testing: Simple but limited

For some applications, it is sufficient to have an experienced testing lab perform black-box testing of a product and report whether keys were extracted. The device is deemed to pass testing if the lab is unable to extract the keys.

A basic black box DPA evaluation uses power traces to infer information about a device's cryptographic implementation, including any DPA countermeasures. The analysis typically involves forming and testing hypotheses about how the target device operates. As a result, problems are unlikely to be detected unless the tester has a strong understanding of the range of possible implementation techniques and countermeasures that might be present. With respect to lab capabilities, black box testing typically requires using high-speed deep-memory analog waveform collection equipment along with specialized high-speed software tools for creating selection functions. Tools for averaging, analyzing and visualizing multi-gigabyte datasets are also needed.

Informal black box DPA testing methodologies have several practical advantages. Product vendors do not need to reveal proprietary design information. Evaluation results tend to be unambiguous (either the device was broken or not). Evaluation laboratories are able to focus on attack strategies that they feel are the most likely to yield results. Labs can also update their test processes quickly as new attack strategies become known. Because minimal documentation and overhead are required, testing can also be relatively inexpensive.

Despite these advantages, black box DPA testing has major limitations. Results tend to be inconsistent, as inexperienced labs will miss flaws that more knowledgeable labs would find. Vendors have been known to choose weaker labs which will have lower costs and higher pass rates, making lab staffing and training a difficult problem. Black box DPA testing also requires expertise that has not traditionally been required for security evaluations of cryptographic devices, such as strong applied number theoretic knowledge. Black box DPA testing is also relatively inefficient and can easily miss vulnerabilities. For example, many countermeasures that might pass "cookbook" black-box DPA testing can be broken by adversaries who know the countermeasure design.

Despite these limitations, most products tested for power analysis at Cryptography Research fail during black box testing. Although such black box testing cannot provide conclusive evidence of a product's security, the process can provide a useful and cost-effective way to differentiate products with at least a moderate level of protection from those that are highly vulnerable.

3 Basic Clear Box Testing

In a clear box test, the evaluator's objective is to verify evidence provided by a device's designer as to why a device is resistant to attack. Unlike a black box

test, the evaluator is assumed to have comprehensive design information about the device.

Clear box evaluations make more efficient use of testing resources than black box evaluations. Testing labs can avoid much of the time-consuming trial-and-error guesswork that black box testing often requires to infer how devices operate. Of course, the benefit directly depends on the quality of the documentation; poorly written or incomplete documentation is of little value. Evaluation efficiency is also increased if vendors provide detailed security claims and justifications for those claims. In this way, evaluators can focus on verifying product designers' results, as opposed to having to search for hidden flaws.

Clear box evaluations are generally easier to conduct than comparable black box evaluations. Less knowledge of implementation techniques and countermeasures is required, since evaluators only have to understand the specific techniques used in a given product as opposed to the universe of possible techniques. Similarly, the amount of data that needs to be collected and processed is typically reduced, since clearly irrelevant experiments can generally be avoided.

The strength of the evidence that can be obtained largely depends on a product's design. For devices that reflect ad-hoc or incompletely documented countermeasures, evaluations are generally limited by the lack of a testable scientific basis for the security claims. As a result, evaluation results typically rely on labs' informed opinions as to whether the techniques employed are appropriate. Even in these cases, a good clear box evaluation can help provide confidence that the security will not fail under black box DPA testing by an adversary with limited resources.

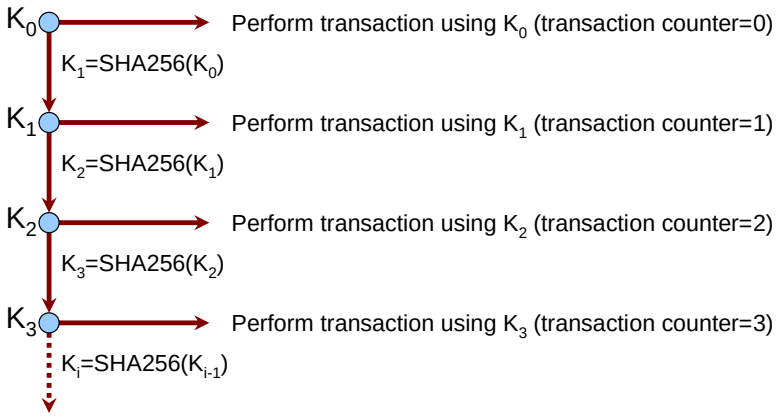
4 High-Assurance Design Elements: Protocols that withstand leakage

To gain higher levels of assurance, devices can use cryptographic constructions that preserve security even if some information about the keys is leaked to adversaries. These approaches can provide fundamentally higher confidence than ad-hoc methods, since they provide a testable justification for the security claims. In particular, protocol designers can provide cryptographic arguments showing that a given design can survive leakage up to a defined threshold leakage rate; manufacturers can then provide engineering evidence that the product's leakage rate is below the threshold. The validation process consists of verifying the properties of the cryptographic protocols and verifying that the implementation leaks less than this amount of information, hopefully with a substantial safety margin.

Unfortunately, most protocols today are not designed to withstand leakage. For example, conventional protocols that compute session keys by encrypting a counter or hashing a key with a nonce can be attacked using DPA because each counter/nonce value potentially reveals new information about the key to the adversary. As a result, even leaks that reveal tiny amounts of information per transaction (such as one thousandth of a bit or less) can compromise secret keys

when attackers combine observations from many transactions. For smart cards and similar small devices, there is no reliable way to ensure that tiny leaks do not exist.

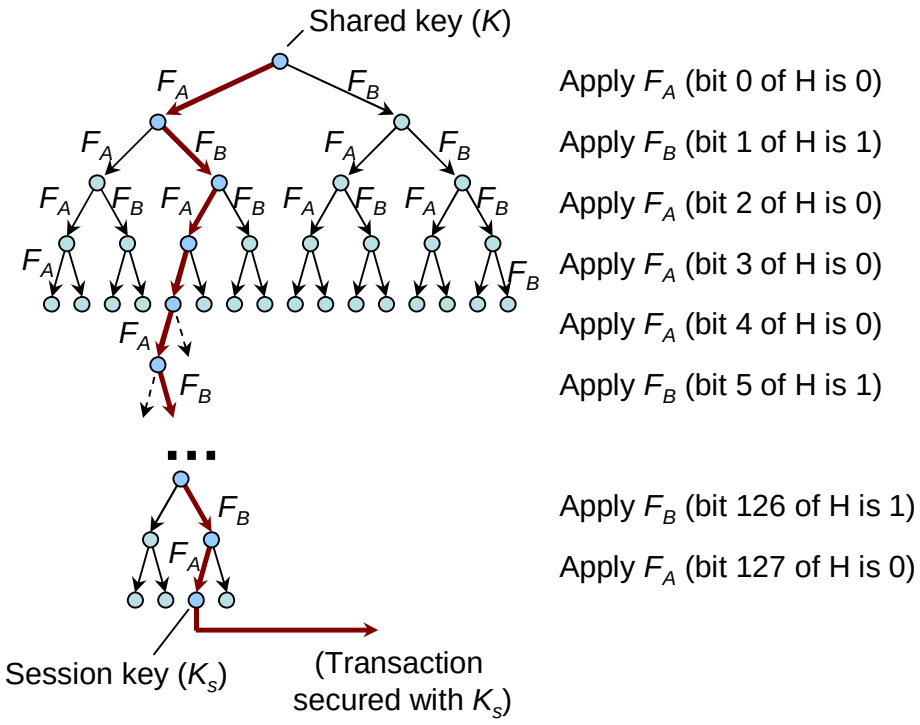
Fortunately, protocols that can survive information leakage can be easy to implement. For example, consider a protocol[2] where a smart card updates a 256-bit key with SHA256 before each transaction with a server. Each SHA256 update destroys the usefulness of previously leaked partial information to the adversary. If L_0 is the upper bound on the amount of information (in bits) leaked per SHA256 key update and L_1 is the maximum amount of information leaked per transaction, then the overall cryptographic security is $256 - 2L_0 - L_1$ bits. (Note that the information content of a leak can be measured in bits even if the leak is probabilistic.) L_0 is counted twice, since adversaries could utilize information leaked from both the SHA256 operation before the transaction and SHA256 operation afterward.



In the protocol above, it does not matter what information is leaked, provided that the quantity of information obtained is less than the required bound. Even if the adversary can choose any L_0 bits of information from the computation $K_2 = \text{SHA256}(K_1)$, the net usable leak will be bounded by $2L_0 + L_1$ provided that the leak from SHA256 does not reveal useful information about K_0 or K_3. The update step redistributes un-leaked entropy in the key making it computationally infeasible for an adversary to combine information leaked prior to the start of the update step with information leaked after the conclusion of the update step.

Protocols that can survive leakage can also be defined for applications where there is not a server (such as secure radios). In an example of such a protocol, two (or more) parties who wish to communicate can generate a shared 128-bit unique value H (e.g., by hashing random contributions from each participant), then use H to select a sequence of hash operations to transform a shared key K (which must be secured against DPA) into a session key K_S. For each bit of H,

the key is transformed in one of two ways (e.g., by applying one of two different hash operations, F_A or F_B).



Each intermediate state is only used in three possible ways: it can be derived as the result of either F_A or F_B, can be transformed with F_A, and can be transformed with F_B. Adversaries can observe each of these operations many times, but these computations will not vary. For example, no other data is mixed in. As a result, there are no variable intermediate values for DPA selections functions to exploit. Because selection functions cannot be applied, DPA-type attacks are no longer applicable.

Let L_0 be the maximum number of bits of secret information leaked from each of F_A and F_B for any given input. (This is a somewhat more restrictive assumption than used for the iterated SHA256 example, since the adversary is assumed to be able to obtain at most L_0 bits about the input no matter how many times the input is transformed.) Let L_1 be the maximum number of bits of secret information leaked from each transaction using K_S. The adversary can thus obtain at most $2L_0$ bits of information about K, $3L_0$ bits of information about any intermediate, and $L_0 + L_1$ bits of information about any K_S. The initial step of deriving H provides each participant with assurance that K_S values will not be used more than once, even if the other participant chooses its

contribution to H maliciously. For additional information about these and other protocols with the ability to withstand leakage, see [2].

Update processes can also be defined for public key algorithms.[3] For example, RSA private key operations can be implemented with key update steps. Update techniques for public key algorithms typically involve encoding the private key as a relationship among a set of randomized parameters. The security of the private key depends on keeping this relationship secret, so it is important that the leakage does not reveal information about the relationship, including if the adversary uses high-order attacks.

5 High-Assurance Validation

To validate a leak-tolerant device's protection against information leakage attacks, the evaluation lab should begin with the designer's security claims. The general process for the evaluation lab then involves the following two steps:

1. Verifying that the designer has made correct claims about the leak tolerance properties of the device's protocols.
2. Verifying that the hardware does not leak in ways that violate the protocol's assumptions, and that there is a suitable safety margin.

The first step consists of evaluating the designer's claims about the device's cryptographic design. For example, if the designer claims that cryptographic strength of 90 bits is maintained with leakage of up to 5 bits per transaction, this should be verified. Evaluators should pay particular attention to the case where adversaries interrupt transactions, e.g. by disconnecting the target device's power during computations.

As part reviewing the protocol, the evaluator should also verify that the designer has completely documented any forms of leakage that must not occur. There will generally be some leakage functions that the cryptography requires will not exist, although these may be so absurd as to pose no practical risk. For example, it is quite reasonable to assume that L_0 bits of information leaked from the computation $K_2 = \text{SHA256}(K_1)$ will not reveal any useful information about the value of K_3 (which has not yet been computed). Assumptions about leakage functions for public key algorithms may require additional attention, as update operations may not be as effective at redistributing entropy.

The second part of the analysis involves characterizing the actual information leaked from the device. The primary goal of this process is to establish an upper bound on the number of bits of information leaked from each transaction. This process is discussed further in the next section. This step should also review any forms of leakage that the design assumes do not exist to verify that these assumptions are correct.

6 Analyzing Leakage Rates

It is impossible to measure the exact amount of useful information leaking from cryptographic device, but useful estimates can be produced. The typical characterization process for devices with countermeasures is as follows:

1. Characterize the leakage from the device with countermeasures disabled.
2. Characterize how countermeasures affect leakage observations.
3. Estimate the overall leakage rate with the countermeasures in effect.

Preferably, the device designer should provide an assessment of the device so that the evaluator can focus on validating security claims.

For Step 1, it is particularly important to disable randomizing countermeasures, since these are otherwise very difficult to distinguish from serious (but as-yet uncharacterized) leaks. Countermeasures that attenuate leaked information (e.g., filters, balancing, etc.) are also helpful to disable, but this is typically not as important.

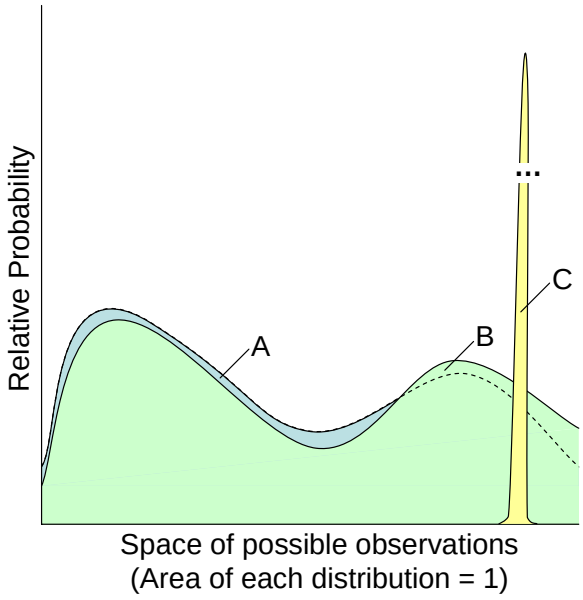
For Step 2, the objective is to characterize how countermeasures affect observations. For countermeasures that use randomness, it is important to verify that the source of the randomness is independent of the secrets and intermediates being protected and cannot be filtered out (e.g., in the temporal or frequency domains). Protocols that rely on limiting the leakage per input, as opposed to per operation, require a more careful evaluation of randomizing countermeasures. The effect of countermeasures that generate noise or random fluctuations should generally be discounted, since adversaries may be able to perform many transactions with each input. Some countermeasures that statistically randomize the representation of data elements within the computation may actually cause each transaction to leak different information, instead of the same information each time, creating a vulnerability to high-order DPA.

For Step 3, the ideal way to characterize the leakage rate is by measuring key/data dependent effects on the distribution of possible measurements. In general, if key-dependent variations are readily apparent, the leakage rate is very high. (For devices that leak badly, the evaluator only needs to demonstrate that the leakage exceeds the cryptographic assumptions; actually quantifying the leakage rate is unnecessary.) In contrast, if groups of measurements using the same key/data cannot be distinguished from groups of operation using different keys, the observed leakage rate is low. The area of difference in the observed distributions determines the information content of the observations. For example, distributions A and B are similar and have relatively low information content, while A and C (or B and C) are largely dissimilar.

It is important to note that the x-axis in the figure corresponds to the space of all *possible* measurements across a transaction, *not* power traces themselves. (For example, a trace consisting of multiple points would be *one* measurement.)

Several caveats need to be considered in the analysis, including:

– The entire measurement space is generally too large to actually enumerate, so approximation techniques are required. These techniques can over- or under-estimate the amount of information leaked.



- Some keys and/or data inputs may give more information than others. Even if the average case is within the leakage rates tolerated by the cryptographic protocols, the worst case may not. Adversaries may be able to trigger worst-case leakage by choosing malicious inputs.
- Uncharacterized leakage (whether in the same or different channels) can reveal more information to adversaries.
- It may be possible to filter out countermeasures whose properties do not exactly match the characteristics of the actual signal.
- Analog characteristics of devices can change significantly between revisions, even if the change does not affect the digital functionality.

In general, these caveats need to be considered by the testing laboratory. These issues also make it important to have a comfortable safety margin between the leakage rates that can be tolerated by a device.

7 The Importance of Good Design

For some products, it is impossible to gain any assurance as to the security against DPA attacks. For example, if the protocols allow adversaries to observe thousands or millions of encryption operations using the same AES key, there is no practical way to verify that leakage rates are low enough to maintain security. Accordingly, these such devices will always have some non-negligible risk. Similarly, some key update processes for some public key algorithms do not provide verifiable protection against high-order DPA attacks, making it unlikely

that an evaluation lab can ensure that the private key will remain secret after large numbers of uses. In other cases, systems are too complex to adequately review given the testing resources available. The designer's security claims or documentation may also be inadequate.

In situations where design limitations of these kinds exist, an evaluation lab cannot conclude with high-assurance that the device under test has effective protection against DPA and related attacks.

8 Conclusion

Power analysis attacks are non-invasive, fast, and leave no physical evidence of tampering. They can be mounted without knowledge of a target's design. As a result, effective testing for DPA and related attacks is essential for devices that must operate securely in hostile environments.

The efficiency of an evaluation, and ultimately the assurance obtained in a device's security, depends on the both the testing process and the design. Black box testing processes can provide a convenient way to reject devices with obvious flaws, but clear box testing is required to obtain much assurance in the testing result. For the highest levels of assurance, devices can use cryptographic protocols designed to withstand leakage of up to a threshold amount of information, then evaluators can assess whether the actual implementation leaks less than this amount of information – with a reasonable safety margin.

9 Acknowledgements

I would like to thank my colleagues Joshua Jaffe and Benjamin Jun for their contributions and assistance.

10 About Cryptography Research

For additional information about Cryptography Research, Inc. (CRI) and its technology and services related to DPA, see http://www.cryptography.com/dpa. CRI licenses a technology portfolio that broadly covers methods for protecting devices against power analysis and related attacks, as well as many specific countermeasure methods. The company also sells the DPA Workstation™ to qualified commercial and government testing laboratories to assist with testing for vulnerabilities to power analysis and related attacks. CRI also provides technical services to assist licensees of its DPA countermeasure technology portfolio.

U.S. patents in CRI's power analysis technology licensing portfolio include: "DES and Other Cryptographic Processes with Leak Minimization for Smartcards and Other Cryptosystems" (US Patent No. 6,278,783), "Secure Modular Exponentiation for Leak Minimization in Smartcards and Other Cryptosystems" (US Patent No. 6,298,442), "Leak Resistant Cryptographic Method and Apparatus" (US Patent No. 6,304,658), "Using Unpredictable Information

to Minimize Leakage from Smartcards and other Cryptosystems" (US Patent No. 6,327,661), "Leak Resistant Cryptographic Method and Apparatus" (US Patent No. 6,381,699), "Balanced Cryptographic Computational Method and Apparatus for Leak Minimization in Smartcards and other Cryptosystems" (US Patent No. 6,510,518), "Leak Resistant Cryptographic Indexed Key Update" (US Patent No. 6,539,092), "Hardware-Level Mitigation and DPA Countermeasures for Cryptographic Devices" (US Patent No. 6,654,884). International patents and other U.S. patents are issued and/or pending.

References

1. P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Advances in Cryptology – Proceedings of Crypto '99*, Lecture Notes in Computer Science, Vol. 1666, Springer-Verlag, 1999, pp. 388-397.
2. P. Kocher, "Leak Resistant Cryptographic Indexed Key Update", *US Patent No. 6,539,092*.
3. P. Kocher and J. Jaffe, "Secure Modular Exponentiation for Leak Minimization in Smartcards and Other Cryptosystems", *US Patent No. 6,298,442*.
4. J. Coron, D. Naccache, and P. Kocher, "Statistics and Information Leakage," *ACM Transactions on Embedded Computing Systems* 3(3):492-508, August 2004.