Differential Power Analysis

Paul Kocher <paul@cryptography.com>
Tue, 09 Jun 1998 20:33:25 -0700

Information is now available online about three related attacks we have developed at Cryptography Research: Simple Power Analysis, Differential Power Analysis, and High-Order Differential Power Analysis.

The basic idea of the attacks is that the power consumption of a device (such as a smartcard) is statistically correlated to the operations it performs. By monitoring the power usage (or electromagnetic radiation, etc.) during cryptographic operations, it is possible to obtain information correlated to the keys. The collected data is then analyzed to actually find the keys. The three attacks use increasingly sophisticated analysis methods.

We have implemented these attacks against a large number of smartcards, and at this point do not believe that any cryptographic smartcards on the market are immune to these analysis techniques.

There is now an initial summary on Differential Power Analysis on our web page at http://www.cryptography.com/dpa, and more information will be put on the website as it becomes available. A condensed text version is also attached below. Paul Kocher

INTRODUCTION TO DIFFERENTIAL POWER ANALYSIS
Paul Kocher, Joshua Jaffe, Ben Jun, Cryptography Research

Introduction: Power Variation

Integrated circuits are built out of individual transistors, which act as voltage-controlled switches. Current flows across the transistor substrate when charge is applied to (or removed from) the gate. This current then delivers charge to the gates of other transistors, interconnect wires, and other circuit loads. The motion of electric charge consumes power and produces electromagnetic radiation, both of which are externally detectable.

Therefore, individual transistors produce externally observable electrical behavior. Because microprocessor logic units exhibit regular transistor switching patterns, it is possible to easily identify macro-characteristics (such as microprocessor activity) by the simple monitoring of power consumption. DPA type attacks perform more sophisticated interpretations of this data.

Simple Power Analysis (SPA)

In SPA attacks, an attacker directly observes a system's power consumption. The amount of power consumed varies depending on the microprocessor instruction performed. Large features such as DES rounds, RSA operations, etc. may be identified, since the operations performed by the microprocessor vary significantly during different parts of these operations. At higher magnification, individual instructions can be differentiated. SPA analysis can, for example, be used to break RSA implementations by revealing differences between multiplication and squaring operations. Similarly, many DES implementations have visible differences within permutations and shifts (e.g., the PC1 permutation or rotates of the C and D registers), and can thus be broken using SPA. While Cryptography Research found many smartcards to be vulnerable to SPA analysis, it is not particularly difficult to build SPA-resistant devices.

The figure [see web site] shows SPA monitoring from a single DES operation performed by a typical smartcard. The upper trace shows the entire encryption operation, including the initial permutation, the 16 DES rounds, and the final permutation. The lower trace is a detailed view of the second and third rounds.

Differential Power Analysis (DPA)

DPA is a much more powerful attack than SPA, and is much more difficult to prevent. While SPA attacks use primarily visual inspection to identify relevant power fluctuations, DPA attacks use statistical analysis and error correction techniques to extract information correlated to secret keys.

Implementation of a DPA attack involves two phases: Data collection and data analysis. Data collection for DPA may be performed as described previously by sampling a device's power consumption during cryptographic

operations as a function of time. For DPA, a number of cryptographic operations using the target key are observed.

The following steps provide an example of a DPA attack process for technical readers. (More detailed information will follow in the near future.) The following explanation presumes a detailed knowledge of the DES algorithm.

1. Make power consumption measurements of the last few rounds of 1000 DES operations. Each sample set consists of 100000 data points. The data collected can be represented as a two- dimensional array S[0...999][0...99999], where the first index is the operation number and the second index is the sample. For this example, the attacker is also assumed to have the encrypted ciphertexts, C[0...999].

2. The attacker next chooses a key-dependent selection function D. In this case, the selection function would have the form D(Ki,C), where Ki is some key information and C is a ciphertext. For the example, the attacker's goal will be to find the 6 bits of the DES key that are provided as the input to the DES S box 4, so Ki is a 6-bit input. The result of D(Ki,C) would be obtained by performing the DES initial permutation (IP) on C to obtain R and L, performing the E expansion on R, extracting the 6-bit input to S4, XORing with Ki, and using the XOR result as the input to the standard DES S4 lookup operation. A target bit (for example, the most significant bit) of the S result is selected. The P permutation is applied to the bit. The result of the D(Ki,C) function is set to 0 if the single-bit P permutation result and the corresponding bit in L are equal, and otherwise D(Ki,C) yields 1.

3. A differential average trace T[0...63][0...99999] is constructed from the data set S using the results of the function D. In particular: [See web site for formula]

4. The attacker knows that there is one correct value for Ki; other values are incorrect. The attack goal is to identify the correct value. In the trace T[i][0...99999] where i=Ki, D(i,C[k]) for any k will equal the value of the target bit in L of the DES operation before the DES F function result was XORed. When the target device performed the DES operations, this bit value was stored in registers, manipulated in logic units, etc. -- yielding detectable power consumption differences. Thus, for the portions of the trace T[i=Ki] where that bit was present and/or manipulated, the sample set T[i] will show power consumption biases. However, for samples T[i != Ki], the value of D(i,C[k]) will not correspond to any operation actually computed by the target device. As a result, the trace T[i] will not be correlated to anything actually performed, and will average to zero. (Actually, T[i != Ki] will show small fluctuations due to noise and error that is not statistically filtered out, and due to biases resulting from statistical properties of the S tables. However, the largest biases will correspond to the correct value of Ki.)

5. The steps above are then repeated for the remaining S boxes to find the 48 key bits for the last round. The attack can then be repeated to find the previous round's subkey (or the remaining 8 bits can be found using a quick search.)

While the effects of a single transistor switching would be normally be impossible to identify from direct observations of a device's power consumption, the statistical operations used in DPA are able to reliably identify extraordinarily small differences in power consumption.

The figure below [see Web site] is a DPA trace from a typical smartcard, showing the power consumption differences from selecting one input bit to a DES encryption function used as a random number generator. (The function of D was chosen to equal the value of plaintext bit 5.) The input initial permutation places this bit as part of the R register, affecting the first-round F function computation and results. Round 2 effects (due to the use of counter mode) are also strong. The trace was produced using 1000 measurements, although the signals would be discernible with far fewer.

High-Order Differential Power Analysis (HO-DPA)

While the DPA techniques described above analyze information across a single event between samples, high-order DPA may be used to correlate information between multiple cryptographic suboperations. Naive attempts to address DPA attacks can introduce or miss vulnerabilities to HO-DPA attacks.

In a high-order DPA attack, signals collected from multiple sources, signals collected using different measuring techniques, and signals with different temporal offsets are combined during application of DPA techniques.

Additionally, more general differential functions (D) may be applied. More advanced signal processing functions may also be applied. The basic HO-DPA processing function is thus a more general form of the of the standard DPA function.

Today HO-DPA are primarily of interest to system implementers and researchers, since no actual systems are known that are vulnerable to HO-DPA that are not also vulnerable to DPA. However, DPA countermeasures must also address HO-DPA attacks to be effective.

Solving the Problems

Cryptography Research has undertaken a substantial development effort to understand hardware security issues and their countermeasures. Cryptography Research has pending patents directed to the technologies and techniques below.

DPA and related attacks span the traditional engineering levels of abstraction. While many previously-known cryptanalytic attacks (such as brute force) can be analyzed by studying cryptographic algorithms, DPA vulnerabilities result from transistor and circuit electrical behaviors which propagate to expose logic gates, microprocessor operation, and software implementations. This ultimately compromises the cryptography.

Techniques for addressing DPA and related attacks can be incorporated at a variety of levels:

Transistor: No feasible alternatives to semiconductors are available today, but alternate computation technologies (such as pure optical computing) may exist in the future. Cryptography Research has developed gate-level logic designs that leak substantially less information.

Circuit, Logic, Microprocessor, and Software: In physically large systems, well-filtered power supplies and physical shielding can make attacks infeasible. For systems with physical or cost constraints, Cryptography Research has developed hardware and software techniques that include ways of reducing the amount of information leaked, introducing noise into measurements, decorrelating internal variables from secret parameters, and temporally decorrelating cryptographic operations. In applications where attackers do not have physical possession of the device performing cryptographic operations, such techniques can be effective. However, because externally-monitorable characteristics remain fundamentally correlated to cryptographic operations, we do not recommend these approaches as a complete solution for applications where attackers might gain physical possession of devices.

Software and Algorithms: The most effective solution is to design and implementing cryptosystems with the assumption that information will leak. Cryptography Research has developed approaches for securing existing cryptographic algorithms (including RSA, DES, DSA, Diffie-Hellman, ElGamal, and Elliptic Curve systems) to make systems remain secure even though the underlying circuits may leak information. In cases where the physical hardware leaks excessively, the leak reduction and masking techniques are also required.

Paul Kocher, President, Cryptography Research, 870 Market St., Suite 1088
San Francisco, CA 94102 415-397-0123 (FAX: -0127) paul@cryptography.com

[This work has enormous potential as one more technique for breaking weakly designed and badly implemented systems, and consequently represents one more forcing function that must be recognized in trying to achieve better systemic security. Unfortunately, it also can break some good good systems. The most important lesson is that computer-communication security is a weak-link problem, and at present, computer-based systems are riddled with weak links. There will always be some weak links, but today there are far too many. PGN]