Differential Power Analysis

Paul Kocher, Joshua Jaffe, and Benjamin Jun

Cryptography Research, Inc.
870 Market Street, Suite 1088
San Francisco, CA 94102, USA.
http://www.cryptography.com
E-mail: {paul,josh,ben}@cryptography.com.

Abstract. Cryptosystem designers frequently assume that secrets will be manipulated in closed, reliable computing environments. Unfortunately, actual computers and microchips leak information about the operations they process. This paper examines specific methods for analyzing power consumption measurements to find secret keys from tamper resistant devices. We also discuss approaches for building cryptosystems that can operate securely in existing hardware that leaks information.

Keywords: differential power analysis, DPA, SPA, cryptanalysis, DES

1 Background

Attacks that involve multiple parts of a security system are difficult to predict and model. If cipher designers, software developers, and hardware engineers do not understand or review each other's work, security assumptions made at each level of a system's design may be incomplete or unrealistic. As a result, security faults often involve unanticipated interactions between components designed by different people.

Many techniques have been designed for testing cryptographic algorithms in isolation. For example, differential cryptanalysis[3] and linear cryptanalysis[8] can exploit extremely small statistical characteristics in a cipher's inputs and outputs. These methods have been well studied because they can be applied by analyzing only one part of a system's architecture — an algorithm's mathematical structure.

A correct implementation of a strong protocol is not necessarily secure. For example, failures can be caused by defective computations[5, 4] and information leaked during secret key operations. Attacks using timing information[7, 11] as well as data collected using invasive measuring techniques[2, 1] have been demonstrated. The U.S. government has invested considerable resources in the classified TEMPEST program to prevent sensitive information from leaking through electromagnetic emanations.

2 Introduction to Power Analysis

Most modern cryptographic devices are implemented using semiconductor logic gates, which are constructed out of transistors. Electrons flow across the sili-

Michael Wiener (Ed.): CRYPTO'99, LNCS 1666, pp. 388-397, 1999.

con substrate when charge is applied to (or removed from) a transistor's gate, consuming power and producing electromagnetic radiation.

To measure a circuit's power consumption, a small (e.g., 50 ohm) resistor is inserted in series with the power or ground input. The voltage difference across the resistor divided by the resistance yields the current. Well-equipped electronics labs have equipment that can digitally sample voltage differences at extraordinarily high rates (over 1GHz) with excellent accuracy (less than 1% error). Devices capable of sampling at 20MHz or faster and transferring the data to a PC can be bought for less than $400.[6]

Simple Power Analysis (SPA) is a technique that involves directly interpreting power consumption measurements collected during cryptographic operations. SPA can yield information about a device's operation as well as key material.

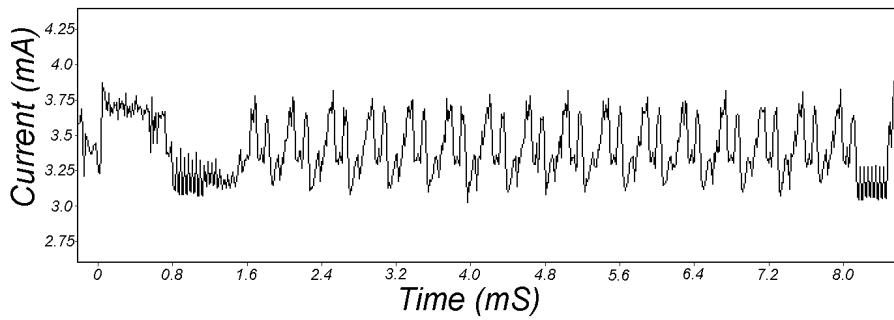

Figure 1: SPA trace showing an entire DES operation.

A *trace* refers to a set of power consumption measurements taken across a cryptographic operation. For example, a 1 millisecond operation sampled at 5 MHz yields a trace containing 5000 points. Figure 1 shows an SPA trace from a typical smart card as it performs a DES operation. Note that the 16 DES rounds are clearly visible.

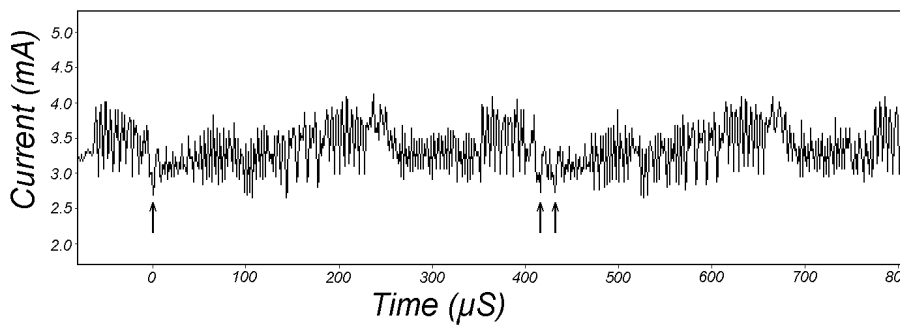

Figure 2: SPA trace showing DES rounds 2 and 3.

Figure 2 is a more detailed view of the same trace showing the second and

third rounds of a DES encryption operation. Many details of the DES operation are now visible. For example, the 28-bit DES key registers C and D are rotated once in round 2 (left arrow) and twice in round 3 (right arrows). In Figure 2, small variations between the rounds just can be perceived. Many of these discernable features are SPA weaknesses caused by conditional jumps based on key bits and computational intermediates.

Figure 3 shows even higher resolution views of the trace showing power consumption through two regions, each of seven clock cycles at 3.5714 MHz. The visible variations between clock cycles result primarily from differences in the power consumption of different microprocessor instructions. The upper trace in Figure 3 shows the execution path through an SPA feature where a jump instruction is performed, and the lower trace shows a case where the jump is not taken. The point of divergence is at clock cycle 6 and is clearly visible.

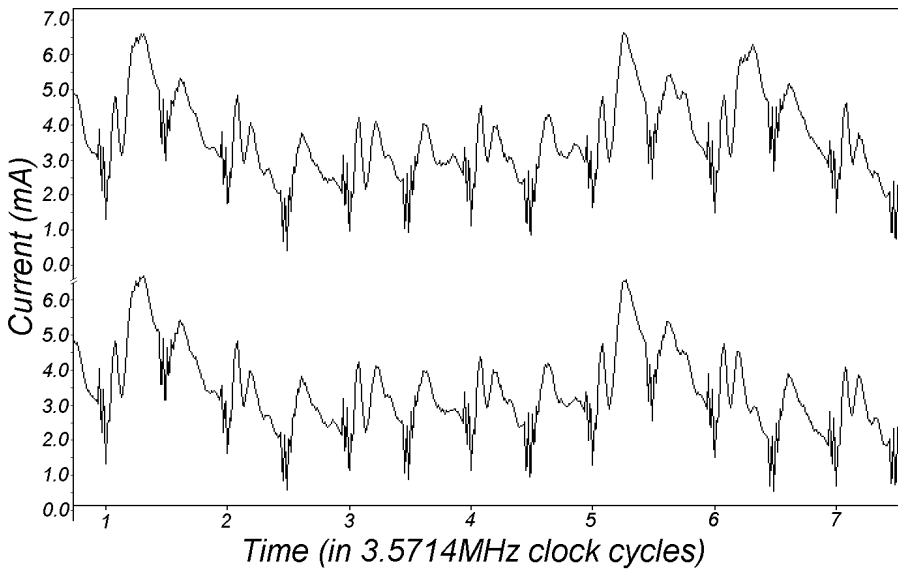

Figure 3: SPA trace showing individual clock cycles.

Because SPA can reveal the sequence of instructions executed, it can be used to break cryptographic implementations in which the execution path depends on the data being processed. For example:

DES key schedule: The DES key schedule computation involves rotating 28-bit key registers. A conditional branch is commonly used to check the bit shifted off the end so that "1" bits can be wrapped around. The resulting power consumption traces for a "1" bit and a "0" bit will contain different SPA features if the execution paths take different branches for each.

DES permutations: DES implementations perform a variety of bit permutations. Conditional branching in software or microcode can cause significant

power consumption differences for "0" and "1" bits.

Comparisons: String or memory comparison operations typically perform a conditional branch when a mismatch is found. This conditional branching causes large SPA (and sometimes timing) characteristics.

Multipliers: Modular multiplication circuits tend to leak a great deal of information about the data they process. The leakage functions depend on the multiplier design, but are often strongly correlated to operand values and Hamming weights.

Exponentiators: A simple modular exponentiation function scans across the exponent, performing a squaring operation in every iteration with an additional multiplication operation for each exponent bit that is equal to "1". The exponent can be compromised if squaring and multiplication operations have different power consumption characteristics, take different amounts of time, or are separated by different code. Modular exponentiation functions that operate on two or more exponent bits at a time may have more complex leakage functions.

3 Preventing SPA

Techniques for preventing simple power analysis are generally fairly simple to implement. Avoiding procedures that use secret intermediates or keys for conditional branching operations will mask many SPA characteristics. In cases such as algorithms that inherently assume branching, this can require creative coding and incur a serious performance penalty.

Also, the microcode in some microprocessors cause large operand-dependent power consumption features. For these systems, even constant execution path code can have serious SPA vulnerabilities.

Most (but not all) hard-wired hardware implementations of symmetric cryptographic algorithms have sufficiently small power consumption variations that SPA does not yield key material.

4 Differential Power Analysis of DES Implementations

In addition to large-scale power variations due to the instruction sequence, there are effects correlated to data values being manipulated. These variations tend to be smaller and are sometimes overshadowed by measurement errors and other noise. In such cases, it is still often possible to break the system using statistical functions tailored to the target algorithm.

Because of its widespread use, the Data Encryption Standard (DES) will be examined in detail. In each of the 16 rounds, the DES encryption algorithm performs eight S box lookup operations. The 8 S boxes each take as input six key bits exclusive-ORed with six bits of the R register and produce four output bits. The 32 S output bits are reordered and exclusive-ORed onto L. The halves L and R are then exchanged. (For a detailed description of the DES algorithm, see [9].)

The DPA selection function $D(C, b, K_s)$ is defined as computing the value of bit $0 \leq b < 32$ of the DES intermediate L at the beginning of the 16th round for ciphertext C, where the 6 key bits entering the S box corresponding to bit b are represented by $0 \leq K_s < 2^6$. Note that if K_s is incorrect, evaluating $D(C, b, K_s)$ will yield the correct value for bit b with probability $P \approx \frac{1}{2}$ for each ciphertext.

To implement the DPA attack, an attacker first observes m encryption operations and captures power traces $\mathbf{T}_{1..m}[1..k]$ containing k samples each. In addition, the attacker records the ciphertexts $C_{1..m}$. No knowledge of the plaintext is required.

DPA analysis uses power consumption measurements to determine whether a key block guess K_s is correct. The attacker computes a k-sample differential trace $\Delta_D[1..k]$ by finding the difference between the average of the traces for which $D(C, b, K_s)$ is one and the average of the traces for which $D(C, b, K_s)$ is zero. Thus $\Delta_D[j]$ is the average over $C_{1..m}$ of the effect due to the value represented by the selection function D on the power consumption measurements at point j. In particular,

$$\begin{aligned}\Delta_D[j] &= \frac{\sum_{i=1}^{m} D(C_i, b, K_s)\mathbf{T}_i[j]}{\sum_{i=1}^{m} D(C_i, b, K_s)} - \frac{\sum_{i=1}^{m} (1 - D(C_i, b, K_s))\,\mathbf{T}_i[j]}{\sum_{i=1}^{m} (1 - D(C_i, b, K_s))} \\ &\approx 2\left(\frac{\sum_{i=1}^{m} D(C_i, b, K_s)\mathbf{T}_i[j]}{\sum_{i=1}^{m} D(C_i, b, K_s)} - \frac{\sum_{i=1}^{m} \mathbf{T}_i[j]}{m}\right).\end{aligned}$$

If K_s is incorrect, the bit computed using D will differ from the actual target bit for about half of the ciphertexts C_i. The selection function $D(C_i, b, K_s)$ is thus effectively *uncorrelated* to what was actually computed by the target device. If a random function is used to divide a set into two subsets, the difference in the averages of the subsets should approach zero as the subset sizes approach infinity. Thus, if K_s is incorrect,

$$\lim_{m \to \infty} \Delta_D[j] \approx 0$$

because trace components uncorrelated to D will diminish with $\frac{1}{\sqrt{m}}$, causing the differential trace to become flat. (The actual trace may not be completely flat, as D with K_s incorrect may have a weak correlation to D with the correct K_s.)

If K_s is correct, however, the computed value for $D(C_i, b, K_s)$ will equal the actual value of target bit b with probability 1. The selection function is thus *correlated* to the value of the bit manipulated in the 16th round. As a result, the $\Delta_D[j]$ approaches the effect of the target bit on the power consumption as $m \to \infty$. Other data values, measurement errors, etc. that are not correlated to D approach zero. Because power consumption is correlated to data bit values, the plot of Δ_D will be flat with spikes in regions where D is correlated to the values being processed.

The correct value of K_s can thus be identified from the spikes in its differential trace. Four values of b correspond to each S box, providing confirmation of key block guesses. Finding all eight K_s yields the entire 48-bit round subkey. The remaining 8 key bits can be found easily using exhaustive search or by analyzing

one additional round. Triple DES keys can be found by analyzing an outer DES operation first, using the resulting key to decrypt the ciphertexts, and attacking the next DES key. DPA can use known plaintext or known ciphertext and can find encryption or decryption keys.

Figure 4 shows four traces prepared using known plaintexts entering a DES encryption function on another smart card. On top is the reference power trace showing the average power consumption during DES operations. Below are three differential traces, where the first was produced using a correct guess for K_s. The lower two traces were produced using incorrect values for K_s. These traces were prepared using 1000 samples ($m = 10^3$). Although the signal is clearly visible in the differential trace, there is a modest amount of noise.

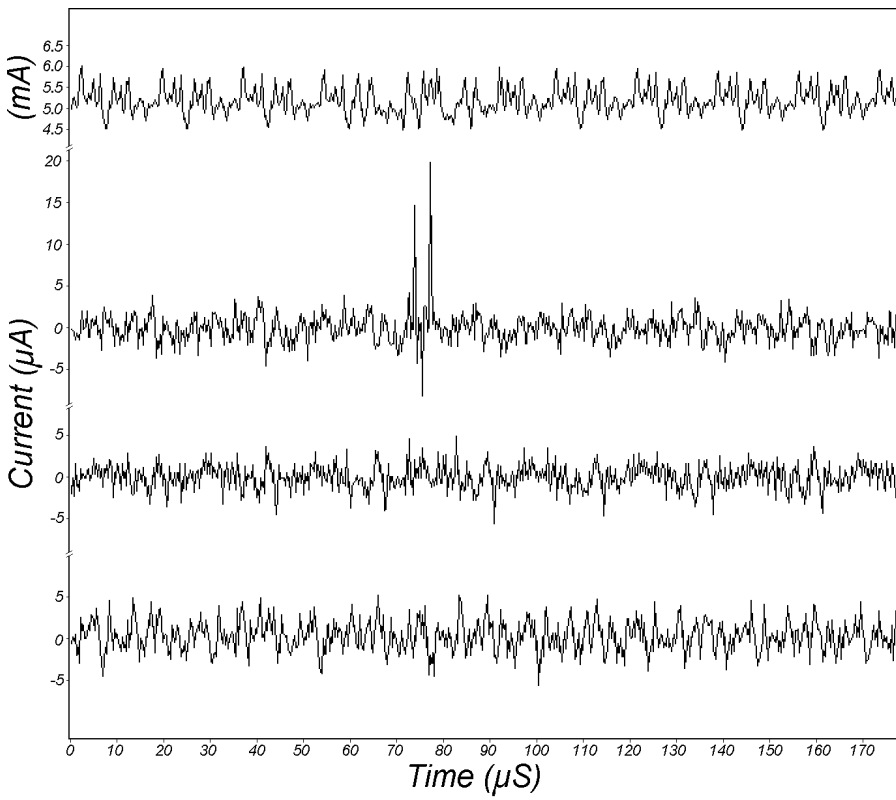

Figure 4: DPA traces, one correct and two incorrect, with power reference.

Figure 5 shows the average effect of a single bit on detailed power consumption measurements. On top is a reference power consumption trace. The center trace shows the standard deviation in the power consumption measurements. Finally, the lower trace shows a differential trace prepared with $m = 10^4$. Note that regions that are not correlated to the bit are more than an order of magnitude closer to zero, indicating that little noise or error remains.

The size of the DPA characteristic is about 40μA, which is several times less than the standard deviation observed at that point. The rise in the standard deviation at clock cycle 6 coinciding with a strong characteristic indicates that the operand value has a significant effect on the instruction power consumption and that there is considerable variation in the operand values being manipulated. Because low-level instructions often manipulate several bits, a selection function can simultaneously select for values of multiple bits. The resulting DPA characteristics tend to have larger peaks, but do not necessarily have better signal-to-noise ratios because fewer samples are included in the averaging.

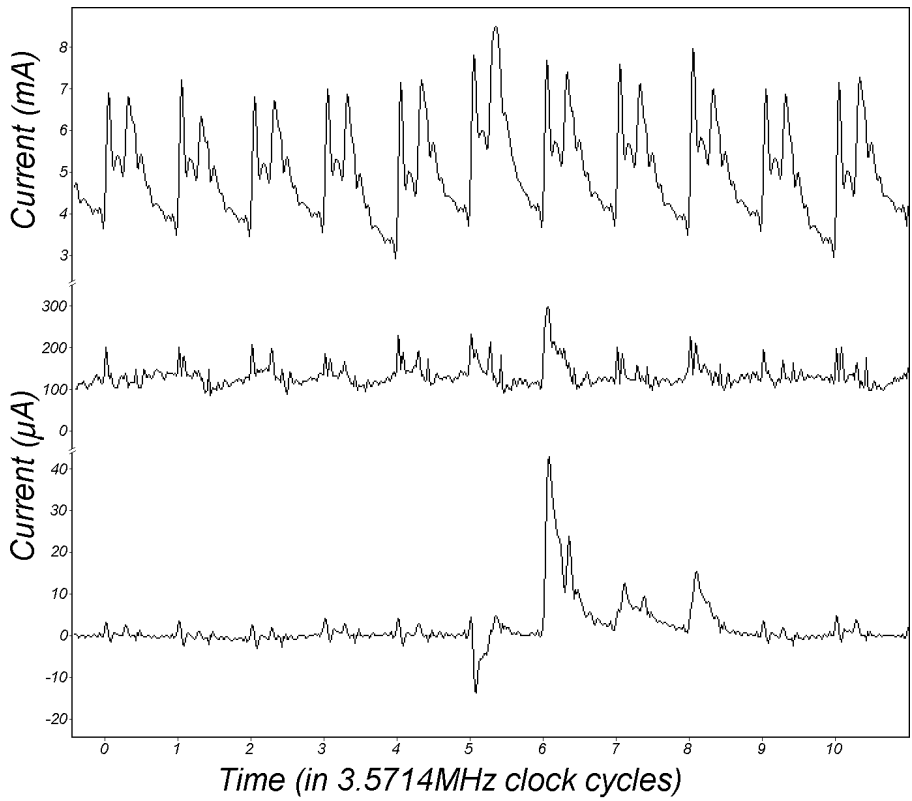

Figure 5: Quantitative DPA measurements

Several sources introduce noise into DPA measurements, including electromagnetic radiation and thermal noise. Quantization errors due to mismatching of device clocks and sample clocks can cause additional errors. Finally, uncorrected temporal misalignment of traces can introduce a large amount of noise into measurements.

Several improvements can be applied to the data collection and DPA analysis processes to reduce the number of samples required or to circumvent countermeasures. For example, it is helpful to correct for the measurement variance,

yielding the significance of the variations instead of their magnitude. One variant of this approach, automated template DPA, can find DES keys using fewer than 15 traces from most smart cards.

More sophisticated selection functions may also be used. Of particular importance are high-order DPA functions that combine multiple samples from within a trace. Selection functions can also assign different weights to different traces or divide traces into more than two categories. Such selection functions can defeat many countermeasures, or attack systems where partial or no information is available about plaintexts or ciphertexts. Data analysis using functions other than ordinary averaging are useful with data sets that have unusual statistical distributions.

5 Differential Power Analysis of Other Algorithms

Public key algorithms can be analyzed using DPA by correlating candidate values for computation intermediates with power consumption measurements. For modular exponentiation operations, it is possible to test exponent bit guesses by testing whether predicted intermediate values are correlated to the actual computation. Chinese Remainder Theorem RSA implementations can also be analyzed, for example by defining selection functions over the CRT reduction or recombination processes.

In general, signals leaking during asymmetric operations tend to be much stronger than those from many symmetric algorithms, for example because of the relatively high computational complexity of multiplication operations. As a result, implementing effective SPA and DPA countermeasures can be challenging.

DPA can be used to break implementations of almost any symmetric or asymmetric algorithm. We have even used the technique to reverse-engineer unknown algorithms and protocols by using DPA data to test hypotheses about a device's computational processes. (It may even be possible to automate this reverse-engineering process.)

6 Preventing DPA

Techniques for preventing DPA and related attacks fall roughly into three categories.

A first approach is to reduce signal sizes, such as by using constant execution path code, choosing operations that leak less information in their power consumption, balancing Hamming Weights and state transitions, and by physically shielding the device. Unfortunately such signal size reduction generally cannot reduce the signal size to zero, as an attacker with an infinite number of samples will still be able to perform DPA on the (heavily-degraded) signal. In practice, aggressive shielding can make attacks infeasible but adds significantly to a device's cost and size.

A second approach involves introducing noise into power consumption measurements. Like signal size reductions, adding noise increases the number of samples required for an attack, possibly to an infeasibly-large number. In addition, execution timing and order can be randomized. Designers and reviewers must approach temporal obfuscation with great caution, however, as many techniques can be used to bypass or compensate for these effects. Several vulnerable products have passed reviews that used naïve data processing methods. For safety, it should be possible to disable temporal obfuscation methods during review and certification testing.

A final approach involves designing cryptosystems with realistic assumptions about the underlying hardware. Nonlinear key update procedures can be employed to ensure that power traces cannot be correlated between transactions. As a simple example, hashing a 160-bit key with SHA[10] should effectively destroy partial information an attacker might have gathered about the key. Similarly, aggressive use of exponent and modulus modification processes in public key schemes can be used to prevent attackers from accumulating data across large numbers of operations. Key use counters can prevent attackers from gathering large numbers of samples.

Using a leak-tolerant design methodology, a cryptosystem designer must define what leakage rates and functions that the cryptography can survive. Leakage functions can be analyzed as oracles providing information about computational processes and data, where the leakage rate is the upper bound on the amount of information provided by the leakage function. Implementers can then use leak reduction and leak masking techniques as needed to meet the specified parameters. Finally, reviewers must verify that the design assumptions are appropriate and correspond to the physical characteristics of the completed device.

7 Related Attacks

Electromagnetic radiation is a particularly serious issue for devices that pass keys or secret intermediates across a bus. Even a simple A.M. radio can detect strong signals from many cryptographic devices. A wide variety of other signal measurement techniques (such as superconducting quantum imaging devices) also show promise. Statistical methods related to SPA and DPA can be used to find signals in noisy data.

8 Conclusions

Power analysis techniques are of great concern because a very large number of vulnerable products are deployed. The attacks are easy to implement, have a very low cost per device, and are non-invasive, making them difficult to detect. Because DPA automatically locates correlated regions in a device's power consumption, the attack can be automated and little or no information about the target implementation is required. Finally, these attacks are not theoretical

or limited to smart cards; in our lab, we have used power analysis techniques to extract keys from almost 50 different products in a variety of physical form factors.

The only reliable solution to DPA involves designing cryptosystems with realistic assumptions about the underlying hardware. DPA highlights the need for people who design algorithms, protocols, software, and hardware to work closely together when producing security products.

References

1. R. Anderson, M. Kuhn, "Low Cost Attacks on Tamper Resistant Devices," *Security Protocol Workshop*, April 1997, `http://www.cl.cam.ac.uk/ftp/users/rja14/tamper2.ps.gz`.
2. R. Anderson and M. Kuhn, "Tamper Resistance – a Cautionary Note", *The Second USENIX Workshop on Electronic Commerce Proceedings*, November 1996, pp. 1-11.
3. E. Biham and A. Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
4. E. Biham and A. Shamir, "Differential Fault Analysis of Secret Key Cryptosystems," *Advances in Cryptology: Proceedings of CRYPTO '97,* Springer-Verlag, August 1997, pp. 513-525.
5. D. Boneh, R. DeMillo, and R. Lipton, "On the Importance of Checking Cryptographic Protocols for Faults," *Advances in Cryptology: Proceedings of EUROCRYPT '97*, Springer-Verlag, May 1997, pp. 37-51.
6. Jameco Electronics, "PC-MultiScope (part #142834)," February 1999 Catalog, p. 103.
7. P. Kocher, "Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems," *Advances in Cryptology: Proceedings of CRYPTO '96,* Springer-Verlag, August 1996, pp. 104-113.
8. M. Matsui, "The First Experimental Cryptanalysis of the Data Encryption Standard," *Advances in Cryptology: Proceedings of CRYPTO '94*, Springer-Verlag, August 1994, pp. 1-11.
9. National Bureau of Standards, "Data Encryption Standard," Federal Information Processing Standards Publication 46, January 1977.
10. National Institute of Standards and Technology, "Secure Hash Standard," Federal Information Processing Standards Publication 180-1, April 1995.
11. J. Dhem, F. Koeune, P. Leroux, P. Mestré, J. Quisquater, and J. Willems, "A practical implementation of the timing attack," *UCL Crypto Group Technical Report Series: CG-1998/1,* 1998.
12. R.L. Rivest, A. Shamir, and L.M. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, **21**, 1978, pp. 120-126.