Side Channel Attack Countermeasures in Cryptographic Systems

James Fell, james.fell@alumni.york.ac.uk

5th November 2015

Side channel attacks against cryptographic systems involve identifying ways in which their physical implementations leak useful information. A cryptographic algorithm may be secure on paper but when implemented on physical hardware some of the secret data, such as key bits, may potentially be recovered by an attacker by measuring various physical properties whilst encryption or decryption is being performed [9].

Common side channels include power usage, timing, sound and temperature. Whenever there is a relationship between secret data and some measurable physical property of the system there is the potential for a side channel attack. Countering these attacks is therefore achieved by either ensuring that the information extracted from the side channel has no relation to anything secret, or by making sure no information is leaked in the first place and hence removing the side channel. This is a general principle regardless of the specific side channel under consideration.

The rest of this essay reviews the most successful countermeasures that can be used to make different classes of side channel attacks as difficult as possible. An understanding of basic principles of cryptography is assumed.

Timing Attacks

If a cryptographic system carries out an operation using secret data as input and the time taken to complete this operation is dependent on the value of this input, then a timing side channel is present. By timing how long operations take, it is possible for an attacker to infer information about the input.

When planning to counter timing attacks one approach is to ensure that all conditional branches in the implementation take the same length of time to execute, and hence the total execution time is always constant independent of inputs. This is not without its difficulties in practice. Some instructions (for example multiplication) take a different length of time to execute depending on their input. There are also issues with compilers automatically doing their own optimisations to code which can undo any attempts made by the programmer to balance out the timing [1].

A better approach is therefore to avoid taking time consuming conditional branches based on data that is intended to be kept secret. For example, it is possible when performing calculations as part of a public key cryptography algorithm to always carry out both multiplications and exponentiations and then simply conditionally return whichever of the two results was really needed and ignore the other [9]. In this case there is still a conditional branch but its effect on timing is relatively small. The aim is to remove any correlation between secret data such as the encryption key and the length of time taken to execute the function.

Another approach to defending against timing attacks is to add noise by adding instructions with no purpose other than introducing random delays. The more noise that is added, the more samples the

attacker needs to collect in order to be successful. According to [1] the number of samples required increases approximately in line with the timing noise squared. While not making an attack impossible this can potentially make it impractical.

An interesting approach to carrying out a timing attack is based on the effect of the CPU cache. A form of this is detailed in [3] with regard to attacking implementations of the AES encryption algorithm. When table lookups (for example an S-box) are used in an encryption algorithm implementation this has often incorrectly been assumed to be safe from timing attacks. The assumption is that looking pre-calculated values up in a table should always take the same amount of time. For a good example of this mistake see [12] from NIST. In reality, if the same value is looked up twice, the second time is quicker due to the CPU cache storing recently accessed memory. According to [3] it is possible to implement AES without S-box lookup tables, instead "using constant-time bit operations: xor, constant-distance shift, etc" and have an implementation that does not have this vulnerability, but it is much slower than the traditional way of implementing AES.

Power Analysis

Similar to timing based side channels, it is often the case that operations can consume different amounts of power depending on their input. Observing power consumption whilst cryptographic operations are performed can therefore potentially reveal information about the secret key being used.

When planning countermeasures to power analysis attacks it is necessary to consider resistance to both Simple Power Analysis (SPA) and Differential Power Analysis (DPA), the former being much easier to defeat than the latter [9]. SPA involves recording power usage a single time and examining it, whereas DPA involves collecting a large number of samples and then trying to use statistical analysis to remove noise.

In the case of SPA, avoiding taking conditional branches based on data that is intended to be kept secret is a good defence, as detailed already regarding timing attacks. This has much the same effect on power consumption as it does on timing. It minimises (but does not completely remove) differences in power consumption based on input. It is also possible to select instructions known to leak less information through their power consumption. An attacker carrying out a DPA attack with enough samples will likely be able to overcome this countermeasure though.

A better countermeasure is the technique of power consumption balancing which involves maintaining total power consumption at some constant value independent of inputs. This is achieved by using dummy registers and performing dummy operations on them [2, 9]. When implemented correctly this can defeat DPA as well as SPA.

As with countering timing attacks, introducing noise can be a useful way to make power analysis attacks harder [2]. Once again adding noise only increases the number of samples that an attacker requires in order to carry out the attack, rather than actually making it impossible. If this number can be increased to a sufficiently high value though this can still make DPA infeasible due to an attacker simply not being able to collect enough samples. This countermeasure is detailed in [10] involving the addition of enough random calculations to the algorithm that the noise level when

monitoring power usage is increased enough to make any DPA spikes too small to detect using a practical number of samples.

Many devices have physical countermeasures to make it difficult to tamper with them, including gaining access to the power or ground connections for measuring power consumption. For example there are tamper-resistant circuits which cease to function and also erase sensitive data automatically when physically attacked [11]. In the case of smart cards though, both the power and the clock signal are typically provided by the smart card reader, not the smart card itself. For this reason, it is usually straight forward for an attacker to read timing and power data when a cryptosystem is implemented on a standard smart card.

Electromagnetic Analysis

Electromagnetic radiation from a cryptosystem can also be used as an effective side channel. Coils placed near to the target device can measure the electromagenetic field generated by the flow of current as well as that caused by the coupling of components that are placed close together. This has led to the attacks known as Simple Electromagnetic Analysis (SEMA) and Differential Electromagnetic Analysis (DEMA) [4]. Electromagnetic attacks are especially attractive in circumstances where effective countermeasures against power analysis have been deployed making that attack impossible [13]. For example, tamper-resistant circuits can be attacked this way.

Countermeasures to EMA attacks are discussed in [14] and fall into two broad categories. The first is to reduce electromagnetic emissions and the second is to inject noise into the emissions. Specific countermeasures mentioned include constructing a Faraday cage around the device and using asynchronous processors due to their reduced electromagnetic signature.

A form of EMA used as a non-cryptographic attack is of course 'van Eck phreaking', an attack which has been public knowledge since the mid 1980s. As explained in Wim van Eck's original paper [7] it is possible to reconstruct the picture that is displayed on a computer monitor by observing electromagnetic radiation from a distance.

Acoustic Analysis

The side channel attack of acoustic cryptanalysis involves recording and analysing sounds emitted by a target device.

In [5] Shamir et al describe an attack where a full GnuPG RSA key is recovered from a laptop using a microphone 4m away from it. Countermeasures described in the paper include acoustic shielding, blinding and ciphertext normalization. According to [5] acoustic shielding is often not practical and usually cannot be implemented perfectly. Blinding and ciphertext normalization on the other hand are both effective defences.

Thermal Analysis

Temperature can also be used as a side channel, such as in a thermal imaging attack. This involves measuring infrared emissions from a processor during the execution of a cryptographic algorithm in order to infer some or all of the key bits [15]. As the temperature of the processor is closely related to its power consumption, countermeasures aimed at thwarting power analysis attacks should also

reduce the vulnerability caused by a thermal side channel. Power consumption balancing and introducing noise are therefore recommended.

Outside of cryptography, a creative example of a thermal side channel attack is presented in [8]. If a lot of traffic is generated to a server over a network, its CPU temperature rises and this causes clock skew as its clock frequency slightly increases. This can be measured remotely by examining packet timestamps. It was shown that this can be used to identify the server hosting a Tor hidden service in very specific circumstances, if a server is accessible both through Tor and directly over the Internet.

Fault Injection

Fault injection attacks can be carried out by varying the voltage or the clock speed provided to the cryptosystem. The output from the system when operating under faulty conditions may leak information that is not available under normal operation [6]. Many samples can be collected this way and then used to perform Differential Fault Analysis. This attack differs from all the others described so far in that it is an active attack rather than passive. In terms of countermeasures, it is possible to implement alarms that detect things such as abnormal voltages or clock speeds and shut the system down.

Blinding

A popular technique used to defend against all of the side channels that we have discussed, specifically when implementing public key cryptography algorithms, is that of blinding [9]. The technique of blinding involves changing the input to the modular exponentiation part of the public key algorithm in a way that allows the output to then be corrected afterwards. A detailed mathematical explanation of this is given in [1] but essentially an extra multiplication is done to the input before the exponentiation and then afterwards a second extra multiplication is done to the output to correct it. The extra multiplications involve first a randomly generated number and then a second number derived from the first. The final result of the operation is the same as if the two extra multiplications had never happened, but any side channel attack will only reveal information about the mutated input and not the real secret data.

Conclusion

This essay looked at several classes of side channel attacks that can be used to compromise a seemingly secure cryptographic system. Countermeasures were suggested in each case.

It should be noted that multi-channel attacks are possible by combining two or more of these side channels and hence ending up with a more efficient attack than is possible when relying on a single side channel. It should also be noted that just as attackers can benefit from combining multiple side channels, defenders can benefit from combining multiple countermeasures.

No single countermeasure is perfect. The best course of action would therefore be to carry out a cost-benefit analysis of each possible countermeasure in relation to the specific application under consideration, and then attempt to implement each one that provides a sufficient increase in security relative to its cost. As always there is a trade-off between security and other factors such as cost, therefore the goal should be to make each attack 'difficult enough' based on the threat model.

References

[1] - Paul C. Kocher. “Timing Attacks on Implementations of Diffie-Hellman, RSA, DSS, and Other Systems”. Advances in Cryptology — CRYPTO ’96. Volume 1109 of the series Lecture Notes in Computer Science pp 104-113. July 2001.

[2] – Paul C. Kocher et al. “Introduction to differential power analysis”. Journal of Cryptographic Engineering. Vol 1, Issue 1 pp5–27. April 2011.

[3] - Daniel J. Bernstein, “Cache-timing attacks on AES”. Department of Mathematics, Statistics, and Computer Science, The University of Illinois at Chicago. 2005.

[4] - Francois Koeune and Francois-Xavier Standaert, “A Tutorial on Physical Security and Side-Channel Attacks”. Foundations of Security Analysis and Design III. Volume 3655 of the series Lecture Notes in Computer Science pp 78-108.

[5] - Daniel Genkin, Adi Shamir and Eran Tromer. “RSA Key Extraction via Low-Bandwidth Acoustic Cryptanalysis”. Advances in Cryptology – CRYPTO 2014. Volume 8616 of the series Lecture Notes in Computer Science pp 444-461. 18th December 2013.

[6] - Matthias Jacob, Dan Boneh and Edward Felten. “Attacking an obfuscated cipher by injecting faults”. Digital Rights Management. Volume 2696 of the series Lecture Notes in Computer Science pp 16-31.

[7] – Wim van Eck. “Electromagnetic Radiation from Video Display Units: An Eavesdropping Risk?”. Computers and Security. Volume 4, Issue 4, pp 269 – 286. December 1985.

[8] – Steven J. Murdoch. “Hot or Not: Revealing Hidden Services by their Clock Skew”. Proceedings of the 13th ACM conference on Computer and communications security, pp 27-36. October 2006.

[9] – Hagai Bar-El. “Introduction to side channel attacks”. Discretix Technologies Ltd, Israel.

[10] - Thomas S. Messerges, Ezzy A. Dabbish, and Robert H. Sloan, “Investigations of Power Analysis Attacks on Smartcards”. Proceedings of the USENIX Workshop on Smartcard Technology. May 1999.

[11] – Ross Anderson and Markus Kuhn. “Tamper Resistance: A Cautionary Note”. Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2. November 1996.

[12] - James Nechvatal et al, “Report on the development of the Advanced Encryption Standard (AES)”, Journal of Research of the National Institute of Standards and Technology 106 (2001).

[13] – Fred de Beer et al. “Practical Electro-Magnetic Analysis”. Non-Invasive Attack Testing Workshop NIAT-2011.

[14] – J-J.Quisquater and D. Samyde. “Electromagnetic analysis (EMA): Measures and countermeasures for smart cards”. In E-smart, pages 200-210, 2001

[15] - Joy Persial G, Prabhu M, Shanmugalakshmi R. “Side Channel Attack Survey” in International Journal of Recent Trends in Electrical & Electronics Engineering. Sept 2011.