How Secure is your E-Purse against Side Channel Leakage?

COLIN D. WALTER

Comodo Research Lab
10 Hey Street, Bradford, BD7 1DQ
UNITED KINGDOM

colin.walter@comodo.net http://www.comodo.net

Abstract: - Electronic purses in smartcards are protected by well-designed protocols and strong encryption. However, progress in the design and techniques of attacks using side channel leakage show that implementers need to update and improve tamper resistance on a continuous basis in order to stay ahead of the attacker. This article surveys the state-of-the-art in non-invasive passive attacks and the algorithmic counter-measures which are being developed.

Key-Words: - Side Channel Leakage, Differential Power Analysis, DPA, Electromagnetic Analysis, DEMA, Smart Card, Exponentiation, RSA, Elliptic Curve Cryptography, ECC.

1 Introduction

Smart cards containing electronic purses seem to be the ideal solution to the problems of hard cash: cheap manufacture, transport and counting, exact change, etc., without the need for on-line verification. However, the card must behave as an electronic safe, guarding its contents against attack. The potential for forging unlimited amounts of anonymous digital cash presents a very attractive incentive to an attacker who can work away in secret for months trying to crack the card.

Even before Kocher described attacks based on the data-dependent variation of time or power consumption by embedded cryptosystems [1,2], there was a long history of government security agencies studying unintended compromising "side channel" leakage. In particular, the long-running US Tempest project investigated electromagnetic emanations from VDUs, computers, cables, etc. [3], and requires appropriate shielding for all government computer equipment, and even for some strategic buildings such as the Pentagon.

In Europe, hackers have been reverse engineering pay-TV smart cards for many years [4]. This usually starts with an invasive attack involving de-packaging of the chip [5]. Similar attacks are also now routinely performed on games consoles [6] to reverse engineer the proprietary hardware and software.

In the case of digital cash, open standards such as CEPS [7] and EMV [8] determine most of the necessary hardware detail, cryptographic protocols and functionality. Mathematical security is provided by using secret keys to well-studied, public domain, encryption algorithms.

This article describes some of the latest research for non-invasive discovery of the keys of electronic purses on smartcards and for protecting them in a hostile environment. The bibliography contains many of the key sources to date.

2 On-Line vs Off-Line E-Cash

In a purely on-line system, secret keys can be restricted to the host facilities alone (the banks), so that physical security is only necessary at those locations. Accelerated key revocation is then straightforward in the case of key compromise.

However, a wholly or partially off-line system is much more desirable for micro-payments. It allows much faster, cheaper transactions in much wider circumstances than credit and debit cards allow: it would be far too expensive for every street newspaper or vending machine in the world to contact a bank to authorise every purchase, and very annoying for every passenger on a mass transit system to be held up for 10 seconds or more at the

barrier while the fare is collected from the traveller's bank. In standard terminology, no trusted third party should need to be directly involved.

On the other hand, an off-line system requires secret keys to be hidden inside every purse. These keys are required, in particular, for protecting the balance in the card from unauthorised alteration by the user. A smartcard is ideal for such a purse because it provides the environment to store these keys securely. However, this shifts the security balance much further in the direction of an attacker.

Under the reasonable assumption that the encryption schemes and protocols are fit for purpose through their mathematically proven strength, the attacker just has to break into any of a million smartcards rather than the single secure, heavily guarded, bank vault – and this can be attempted much more easily without being detected.

Although the rewards are less in the short term than from robbing a bank safe, the anonymity of the card means that it could be re-charged and re-used endlessly for off-line transactions, perhaps even cloned for mass use. Transaction logs in cards are expected to be only around 10 records long [9], and so, by avoiding ATMs and other on-line terminals, the audit trail is really too short for illegal activity to be spotted easily when transactions occur. There is, apparently, no provision for off-line point-of-sale terminals to contain a blacklist of suspect cards although there is a limit to the number of consecutive off-line transactions that can take place.

In summary, we can expect forged electronic money to be in circulation in the future in the same way as counterfeit currency is today. How much depends on the success of exciting, current, on-going research.

3 Background

There are two main models for e-purses. One, such as EMV [8], allows for the unconditional transfer between individual purses, and the other, tighter model, such as CEPS [7], has a merchant/card holder model where money circulates only from bank to card to merchant to bank. Generally it includes a full audit trail except where aggregation of individual micro-payments is allowed. The fixed length, closed loop is obligatory except for transfers within the smart card, for transaction reversal, and for unloads from card direct to bank.

However, both models expect that micro-payments will be made over the internet to pay for items such as video clips or pages of copyright material [10]. Explicit requirements are included in the standards to cover this possibility. This, and the vast number of automatic dispensing machines, photocopiers, payphones etc. ensures that low technology, off-line card readers must be permissible. In particular, unlike ATMs, these readers are not an integral part of sealed, tamper-evident boxes which prevent card I/O from being monitored.

Thus we can assume that anyone attacking the card can construct, or has access to, a modified but fully functional reader which is capable for net transactions, but also contains probes for measuring minute variations in power usage and/or electro-magnetic radiation (EMR). We can also assume that the attacker has a digital oscilloscope and PC to capture and store the power and EMR data – the "side channel" leakage traces. This, together with the messages via the main I/O channel to and from the card, provides the raw material for deducing the secret keys non-invasively.

Other forms of attack, such as fault induction [11], tend to be more invasive. They may require de-packaging the chip using fuming nitric acid and acetone, perhaps rendering it useless for future use [5]. Such rough treatment should trigger counter-measures which guarantee the immediate destruction of the secret keys. Some card chips include wire mesh Faraday cages to protect them from EMR leakage and to prevent micro-probing to measure bus data: cutting one of these wires should trip key annihilation.

There is now a considerable body of literature on attacking security systems on smart cards, breaking their protocols, crypto-systems and hardware counter-

measures using a variety of techniques ranging from the theoretical mathematical to the practical electrical [12-18]. One should assume that smart cards coming onto the market address all these problems with cost effective solutions.

The e-purse needs to have a balance which is readable by the user. So it must be in plaintext, but protected against alteration by a signed MAC – a message authentication code obtained by applying a hash function, typically SHA-1, to the text. (A possible alternative might be to sign the balance and transaction logs directly, store only the resulting ciphertext, and use the public key to obtain balances, but this is known to be insecure – a hash function must be employed.)

The RSA public key cryptosystem is the current standard specified for these signatures, and we concentrate on that here. The terminology is that the plaintext P and ciphertext C are related by

$$C = P^e \bmod N \quad \text{and}$$
$$P = C^d \bmod N$$

where e is the public encryption or verification key and d is the private decryption or signature key. Typical numbers here have between 1024 and 2048 bits. Elliptic curve cryptography (ECC) is a likely alternative in the near future because of its shorter key lengths for equivalent security.

The off-line nature of some transactions requires the private signature key d to be stored on the card. Moreover, all encryptions with the private key must be performed on the card since exporting the key cannot be allowed for security reasons. To guarantee to any part of the e-cash system that keys are authentic, they are always provided within a certificate which is signed by the issuing authority. This certificate is itself signed by one of a small number certificate authorities (CAs) which the card can recognise as being legitimate because it stores their public keys. As the signature keys for the certificates are not on the card, the attacker is unable to substitute the balance signature key with his own. Instead, he must discover the balance signature key from the card in order to sign his own forged data.

4 Obtaining the Keys

Every time a transaction occurs, the signature key is re-used by the card to sign a hash of the critical data. Hence, by making a number of small micro-payments over the internet with his modified reader, the attacker can obtain power and/or EMR traces which record key-dependent variations during the process of signing.

Typical power traces are found in [2] for DES and in [19] for RSA. As a rule, EMR produces more detailed traces than power does [20]. The traces show that squares can be distinguished from multiplications. If the standard square-and-multiply exponentiation algorithm is used, there is a square and a multiplication for every 1 bit in the exponent, but only a square for every 0 bit. Hence the bits of the secret exponent can be read directly from a power or EMR trace. So the private key is exposed.

Incidentally, the modular multiplication may involve a conditional extra subtraction of the modulus N. This produces an average timing difference between squares and multiplies which may be utilised if the power differences are unclear [1,21]. Thus, code for exponentiation should ensure that squares and multiplies take the same, unvarying time.

There are several main defences against these attacks on square-and-multiply: secret sharing, blinding, randomisation of inputs, and novel exponentiation algorithms.

The frontline of defence is probably to ensure that an attacker cannot determine the I/O data precisely. Many of the historic mathematical attacks depended on knowing the plaintext or ciphertext. Rivest blinding starts by replacing ciphertext C with $CR^e \bmod N$ for some large random integer R. Then the exponentiation by d produces PR, and post-processing easily yields the plaintext P. This stops an attacker running an identical card with the same input, and choosing bits of d so that the power traces on the two cards match. He has no idea what R is.

A typical secret sharing technique would break the secret d into the sum of two numbers, $d = r+s$, where r is random. Then P would be computed as $C^r \times C^s \bmod$

N. Unfortunately, this is an expensive solution, especially for contactless cards used on a mass transit system where time and power are limited. Moreover, it might not work as a counter-measure: perhaps differential power analysis (DPA) techniques could just align and combine the traces for the two exponentiations to obtain $d = r+s$?

Here it should be mentioned that random noise and variation in the processed text submerge most of the key-dependent power variation. So some averaging over a number of exponentiations is normally necessary to observe the difference between squares and multiplies.

An alternative random change to the exponent is to replace it with $d+rg$ where r is random (typically 32 bits) and $g = \varphi(N)$, for the Euler phi function φ. This has the nice property that $C^{d+rg} \equiv C^d \bmod N$. Such exponent blinding means that a different pattern of squares and multiplies is executed on every signing or decryption. As a result, the averaging over a number of power traces no longer succeeds; it averages away any key dependency.

However, there is still a real danger that squares may be distinguished from multiplies on a *single* exponentiation. For example, data passing along the internal bus makes substantial use of power, and this enables the Hamming weight (the number of bits set to 1) of arguments to be determined [22]: equal weights almost always means equal arguments and therefore a square, otherwise a multiplication. Furthermore, in elliptic curve cryptography, squares and multiplies actually require different numbers of field operations, and so much greater effort must be expended in hiding the differences [18].

The most obvious solution is to use m-ary exponentiation where, for convenience, m is generally a power of 2. Assume that the representation of d in base m is $d = d_n m^n + \ldots + d_1 m + d_0$. The method commences with the pre-computation of the digit powers $C^{(i)} = C^i \bmod N$ for $1 \le i \le m-1$ and the main loop eventually outputs $P = C^d \bmod N$:

$C^{(1)} \leftarrow C$;
For $i \leftarrow 2$ **to** $m-1$ **do**
    $C^{(i)} \leftarrow C^{(i-1)} \times C \bmod N$;
$P \leftarrow C^{(d_n)}$;
For $i \leftarrow n-1$ **downto** 0 **do**
Begin
    $P \leftarrow P^m \bmod N$;
    If $d_i \neq 0$ **then** $P \leftarrow P \times C^{(d_i)} \bmod N$;
End ;

For $m > 2$, there is now an ambiguity over what each multiplication represents: it is not clear which non-zero digit is used.

If squares and multiplies can be distinguished but nothing else, then it is computationally infeasible to attempt a brute force attack which tries every choice for every non-zero digit. Unfortunately, there are sections of power trace from a *single* exponentiation which may be averaged in order to distinguish squares from multiplies. This technique identifies the re-use of operands from the Hamming weights of their component words [23]. Thus, every time $C^{(i)}$ is used, we know the corresponding digit of d is i. Moreover, every operation for which no matching operand $C^{(i)}$ is recognised has to be a square.

The last year or so has seen some solutions to this dilemma, with the development of some new, randomising exponentiation algorithms. Few of these have been thoroughly investigated; some are known to provide little additional security, if any at all. However, one of the most promising is "Mist" [24–26]. It can be viewed as a modification of m-ary exponentiation where m is chosen randomly for each digit of d from a small set, such as {2,3,5}. Also, the direction of processing the digits of d is reversed in order to avoid operands being re-used so widely:

$Q \leftarrow C$;
$P \leftarrow 1$;
While $d > 0$ **do**
Begin
    Choose random m from {2,3,5} ;
    $r \leftarrow d \bmod m$;
    If $r \neq 0$ **then** $P \leftarrow Q^r \times P \bmod N$;
    $Q \leftarrow Q^m \bmod N$;
    $d \leftarrow d \operatorname{div} m$;
End ;

The successive values of r are analogous to the digits d_i in a base m representation. Always picking m=2 turns this back into the right-to-left version of the square-and-multiply algorithm in which processing starts with the least significant bit of d.

MIST is computationally infeasible to break if one assumes squares and multiplies can be distinguished, or that operand re-use can be detected, providing also that exponent blinding is used to prevent data from several exponentiations being combined [26].

5 Counter-Measures

The previous section reviewed the alternation between new, passive, side channel attacks and new software counter-measures in relation to the determination of secret keys to an electronic purse. The main theme is certainly randomisation: of inputs, of keys and of algorithms. There are also hardware measures that can be taken, such as Faraday cages to reduce EMR, large capacitors to even out power consumption, noise generators to decrease the signal-to-noise ratio and clock variation to frustrate trace averaging. However, smart cards have limited scope for these physical counter-measures because the standards must specify a maximum chip size.

Key lengths have interesting properties in relation to these attacks. For symmetric crypto-systems, the number of rounds is usually proportional to the key length, so that the side-channel data per key bit remains constant. This means that increasing key length generally increases the security of the system against DPA.

On the other hand, for arithmetic-based RSA-type asymmetric crypto-systems, all reasonable area-bounded implementations of exponentiation take time proportional to the cube of the key length. This means that more side channel data is available per key bit as key length increases. Consequently, the attacks described above actually become *easier* as key length increases [27]. Normal key management principles limit the lifespan of keys. As well as damage limitation in case of key compromise, this limits the data available for side channel attacks.

Only passive attacks have been discussed here, i.e. those where the smart card has been operated entirely within its normal operating conditions. However, there is a whole battery of active attacks which may or may not involve invasive methods and may or may not trigger key destruction. This is beyond the scope of this article, but briefly varying power, clock or temperatures outside their specified ranges may not be noticed by anti-tamper circuits. Focussed optical, electrical, or ion beams or electro-magnetic pulses can all disturb data or modify instructions. The danger of faults induced by such glitches requires that various checks be made before any output occurs [11], [13].

6 Conclusion

Improvements in protocols, specifications and hardware have all increased the reliability and tamper-resistance of smart card electronic purses. However, the persistence of side channel leakage of secret keys has led to an on-going search for better algorithm implementations as protection against the increasing sophistication of the attacker.

References:

[1] P. Kocher, *Timing Attack on Implementations of Diffie-Hellman, RSA, DSS, and other systems*, Advances in Cryptology – CRYPTO '96, N. Koblitz (editor), LNCS **1109**, Springer-Verlag, 1996, pp 104–113.

[2] P. Kocher, J. Jaffe & B. Jun, *Differential Power Analysis*, Advances in Cryptology – CRYPTO '99, M. Wiener (editor), LNCS **1666**, Springer-Verlag, 1999, pp 388–397.

[3] *The Complete, Unofficial TEMPEST Information Page*, http://www.eskimo.com/~joelm/tempest.html

[4] *Satellite TV Hacking*, http://www.govital.net/~soz/links/satelite_tv_hacking.htm

[5] O. Kommerling & M. G. Kuhn, *Design Principles for Tamper-*

Resistant Smartcard Processors, Proc USENIX Workshop on Smartcard Technology, Chicago, Illinois, 1999.

[6] A. Huang, *Keeping Secrets in Hardware: the Microsoft X-BOX Case Study*, Proc. CHES 2002, Redwood City, Aug 2002, *to appear in* LNCS, Springer-Verlag.

[7] *Common Electronic Purse Specification*, CEPSCO LLC, http://www.cepsco.com

[8] *EMV 4.0 Specifications*, EMVCo LLC, http://www.emvco.com.

[9] *IFD Implementation Guidelines* (tec-gui-763), Mondex, Mastercard, http://www.mondex.com, 1999, Section 3.

[10] Secure Electronic Transaction LLC (SETCo), http://www.setco.org.

[11] D. Boneh, R. A. DeMillo & R. J. Lipton, *On the Importance of Checking Cryptographic Protocols for Faults*, Advances in Cryptology – EUROCRYPT '97, LNCS **1233**, Springer-Verlag, 1997, pp 37–51.

[12] D. Boneh, *Twenty years of attacks on the RSA cryptosystem*, Notices of the American Mathematical Society, vol. **46** (2), Feb 1999, pp 203–213.

[13] Ross Anderson & Markus Kuhn *Tamper Resistance – a Cautionary Note*, Proceedings of the Second USENIX Workshop on Electronic Commerce, Oakland, California, November 18-21, 1996, pp 1–11.

[14] J.-J. Quisquater & D. Samyde, *ElectroMagnetic Analysis (EMA): Measures and Counter-Measures for Smart Cards*, Smart Card Programming and Security (E-smart 2001), LNCS **2140**, Springer-Verlag, 2001, pp 200–210.

[15] J.-J. Quisquater & D. Samyde, *Eddy current for Magnetic Analysis with Active Sensor*, Smart Card Programming and Security (E-smart 2002), LNCS, Springer-Verlag, 2002, *to appear*.

[16] Ç. K. Koç & C. Paar (Eds.), *Cryptographic Hardware and Embedded Systems*, LNCS **1717**, Springer-Verlag, 1999.

[17] Ç. K. Koç & C. Paar (Eds.), *Cryptographic Hardware and Embedded Systems - CHES 2000*, LNCS **1965**, Springer-Verlag, 2000.

[18] Ç. K. Koç, D. Naccache & C. Paar (Eds.), *Cryptographic Hardware and Embedded Systems - CHES 2001*, LNCS **2162**, Springer-Verlag, 2001.

[19] T. S. Messerges, E. A. Dabbish, & R. H. Sloan, *Power Analysis Attacks of Modular Exponentiation in Smartcards*, Cryptographic Hardware and Embedded Systems, LNCS **1717**, Springer-Verlag, 1999, pp 144–157.

[20] D. Agrawal, B. Archambeault, J. R. Rao & P. Rohatgi, *The EM Side-Channel(s)*, Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS, Springer, 2002, *to appear*.

[21] C. D. Walter & S. Thompson, *Distinguishing Exponent Digits by Observing Modular Subtractions*, Topics in Cryptology – CT-RSA 2001, D. Naccache (editor), LNCS **2020**, Springer-Verlag, 2001, pp 192–207.

[22] T. S. Messerges, E. A. Dabbish & R. H. Sloan, *Examining Smart-Card Security under Threat of Power Analysis Attacks*, IEEE Transactions on Computers, vol. **51** (5) May 2002, pp 541–552.

[23] C. D. Walter, *Sliding Windows Succumbs to Big Mac Attack*, Cryptographic Hardware and Embedded Systems - CHES 2001, LNCS **2162**, Springer-Verlag, 2001, pp 286–299.

[24] C. D. Walter, *Improvements in, and relating to, Cryptographic Methods and Apparatus*, UK Patent Application 0126317.7, Comodo Research Lab, 2nd November 2001.

[25] C. D. Walter, *MIST: An Efficient, Randomized Exponentiation Algorithm for Resisting Power Analysis*, Topics in Cryptology – CT-RSA 2002, B. Preneel (editor), LNCS **2271**, Springer-Verlag, 2002, pp 53–66.

[26] C. D. Walter, *Some Security Aspects of the MIST Randomized Exponentiation Algorithm*, Cryptographic Hardware and Embedded Systems (CHES 2002), LNCS, Springer-Verlag, 2002, *to appear*.

[27] C. D. Walter, *Is there Safety in Numbers against Side Channel Leakage?*, RSA Europe Conference, (crypto track), Amsterdam, 15-18th October, 2001.