Smart Card Platform-Fingerprinting

Abstract

The Smart card has become a vital security component in a wide range of system solutions including banking, telecommunications, Pay-TV, ticketing and citizen identity schemes. The widespread use of smart cards is in part due the fact that they offer tamper-resistant security functionality at sufficiently low cost that they may be issued to large numbers of users. The level of tamper-resistance in modern smart card devices is impressive and has evolved in response to counter a wide range of threats including the leakage of information via side channels. The exploitation of side-channel leakage from smart cards was published in the late nineties and since then has almost invariably been associated with attacks on security. However the side channel leakage signals have some interesting properties and this paper takes a creative look at how they might be used to fingerprint/characterise genuine smart card platforms as a potential means of improving smart card system security by virtue of clone card detection.

Introduction

The smart card is well established as a security token in many commercial system solutions such as mobile communications [1], EMV banking [2] and PAY-TV[3]. The tamper-resistant security properties of the smart card have also lead to its growing exploitation in newer but no less significant applications including ID card schemes[4], electronic ticketing[5], health cards[6] plus general government and entitlement solutions[7]. This expansion in usage will of course attract more attention from criminal organisations, as more and more value will be accessible if only the technological protection can be compromised. It is therefore not surprising that there is a great deal of literature on the subject of smart card attacks and countermeasures[].
Whilst it is comforting to know that countermeasure expertise is usually a few steps ahead of the attackers it is prudent to consider how you would protect your system and service should an attack prove practical. In fact there may be no attack on the card itself as the secret data may be obtained by human engineering or misuse of databases. In the case of an attacker having gained access to some legitimate personal or account information the aim may be to benefit directly by authorising transactions or to create cloned identities with associated rights for sale to third parties.
The prospect of clone cards is not new but this paper takes a creative look at how they might be detected using similar techniques to those used by the hackers, rather than relying on complex system fraud engines and location/behaviour profiling

Attack of the Clones

There have not been too many publicised cases of smart card clones. The satellite TV providers are beset by hackers and have released new security measures as necessary, although the secretive nature of the solutions makes it difficult to report in detail on what has actually taken place. A better publicised example is the compromise of the COMP128-1 algorithm used by some mobile network operators. Before explaining this further it is important to appreciate that COMP128-1 is not the GSM authentication algorithm. It was an example provided by the GSM Association and

MNOs are free to use any public/proprietary algorithm and many have – the smarter ones right from the initial deployment of their networks. It is also important to state that any problems arose fundamentally from the weakness of the algorithm itself and not any failing in the host smart card platform. Whilst the 128 bit secret key (Ki) should have held off brute force attacks, the appearance of leaked design documentation was swiftly followed by researchers at Berkley [A4], extracting a secret key (Ki) in about 130K-160K attempts. The researchers had spotted a flaw that made the algorithm vulnerable to collisions i.e. different random challenges (RAND) giving the same output. The algorithm used multiple rounds of compression based on the Ki and RAND, but referring to Fig 2 one can see that the output bytes of the second round were only dependent on 4 corresponding input bytes. The final gift to the attackers was that at a collision occurring at the second round then propagated to the final result. This is described in more detail within the literature [A4].

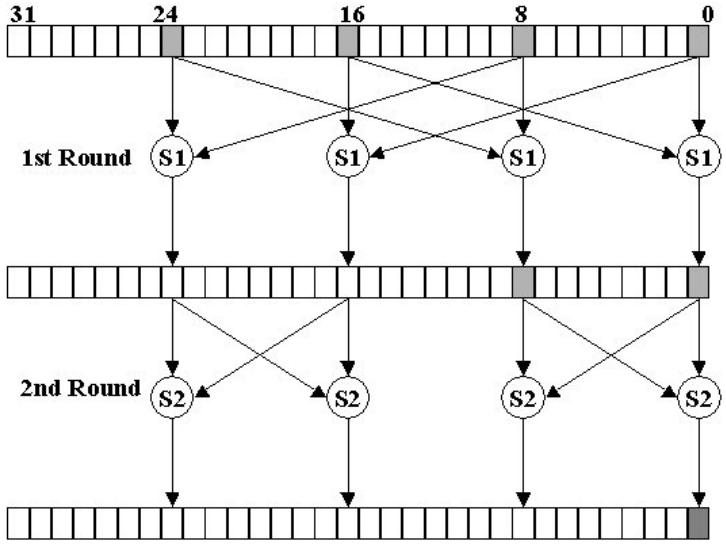


Figure 1 Compression Rounds in Comp128-1

The immediate response was to introduce an authentication retry count into the SIM card so that it would block before the many attempts had been performed. However attack techniques have advanced such that the counter limits now impact the operational life-time of the card. The inescapable conclusion for the layman is that using a poor algorithm is evidence of bad judgement and assuming any weakness will stay a secret is extremely naive.

This kind of thing was known in the early nineties but initially people were not too concerned as extracting the secret key was only a useful step if the hacker could find a supply of suitable smart card platforms to create clone cards. Most SIM cards were then sourced by reputable vendors and so it was not trivial to get hold of suitable programmable smart cards. In the mid-nineties the situation started to change and it was possible to buy a Comp128-1 clone kit (sourced from Asia – cost about $100) that it included not only the attack software but also a couple of blank smart cards.

Today the smart card supply situation has changed enormously due to the increased use of smart cards in a wide variety of system solutions. There are now many programmable cards on the market[], including sophisticated Java Cards[] that support the T=0 protocol used in the majority of mobile devices. Therefore the lack of

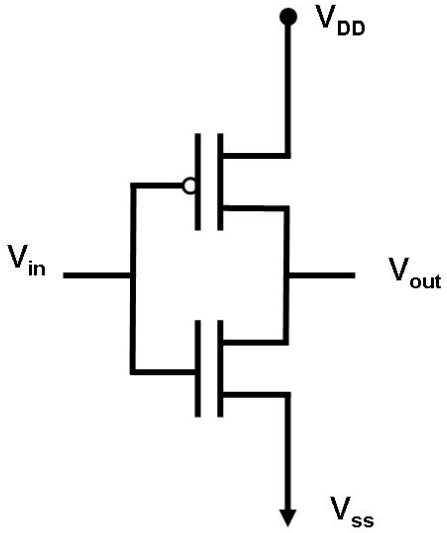
physical clone platforms is no longer a source of comfort or defence. Of course the SIM application is still needed, but with a wide range of available development tools implemented a functional version on say a Java card platform is relatively straightforward.

One thing that remains generally true is that the clone platform is unlikely to be the same as the original card. Therefore if we have some reliable means to differentiate original and clone platforms we could reject cards whose applications behave in a logically identical fashion. For this to be feasible we need to find some detection means equivalent to "fingerprinting" the underlying platforms. This paper will look for smart card fingerprint candidates in the field of information leakage which perversely is normally exploited for smart card attacks.

Information Leakage

There are many real-world examples of processes that unintentionally leak information. There are also cases of intentionally misleading leakage to fool observers. As an example, consider a game of poker, in which everyone plays by the same functional rules and keeps their cards well concealed. If a novice player looks worried or excited when he receives his cards then he leaks information about his hand to the other players. The experienced player may manipulate his leakage i.e. he may block his emotions/expressions or indeed manipulate them to fool others.
Information leakage can be used to gain additional information about players (analogous to smart card platforms) that are carrying out functionally equivalent tasks, however finding a practical metric that is easily manipulated (recall the experienced poker player) requires a little create thought. One definition of creativity is the association of several disparate things to derive a new solution. So we will start by considering submarines!
If anyone visits the submarine museum in Portsmouth England, they have the chance to play with some interactive exhibits, one of which is a sonar demonstrator. This gives you the opportunity to listen to and try and identify a range of signals e.g. is that a battleship or a torpedo coming towards you? Clearly this information is leaking from the vessels but it is not easy to eliminate this as the noise relates to fundamental processes in the operation of the device e.g. engines, propellers etc. It is not easy for the unskilled person to recognise the different signals, however it is reported that skilled sonar operators can not only differentiate between types of vessels but even identify particular vessels and whether they are in need of a service!
There are no doubt techniques that can try to obscure the leaked sonar information but making a battleship sound like a rowing boat seems unlikely. We therefore have one example of a leaked signal that requires expert detection and is also difficult to completely disguise or convincingly duplicate.

Smart cards do not have moving parts to rattle and make sonar signals, but in common with all electronic devices, they do have moving currents. As register values change from a logical 1 to 0, and in CMOS technology for example, pairs of transistors have to change state as shown in Fig x.



As one transistor is turning off whilst the other is turning on there is a momentarily a conducting path between VDD and VSS which results in a surge of current or "spike". A secondary effect of the current surge is to also generate a weak electromagnetic emission. The extent of the surge and emissions depends on a variety of factors, including the threshold switching voltage, temperature, supply voltage and simply the position of the circuit relative to the production wafer []. However both of these phenomena may be detected, captured, displayed and analysed using little more equipment than an oscilloscope, PC plus a resistor and/or antenna. Typical emissions can be seen in fig X.

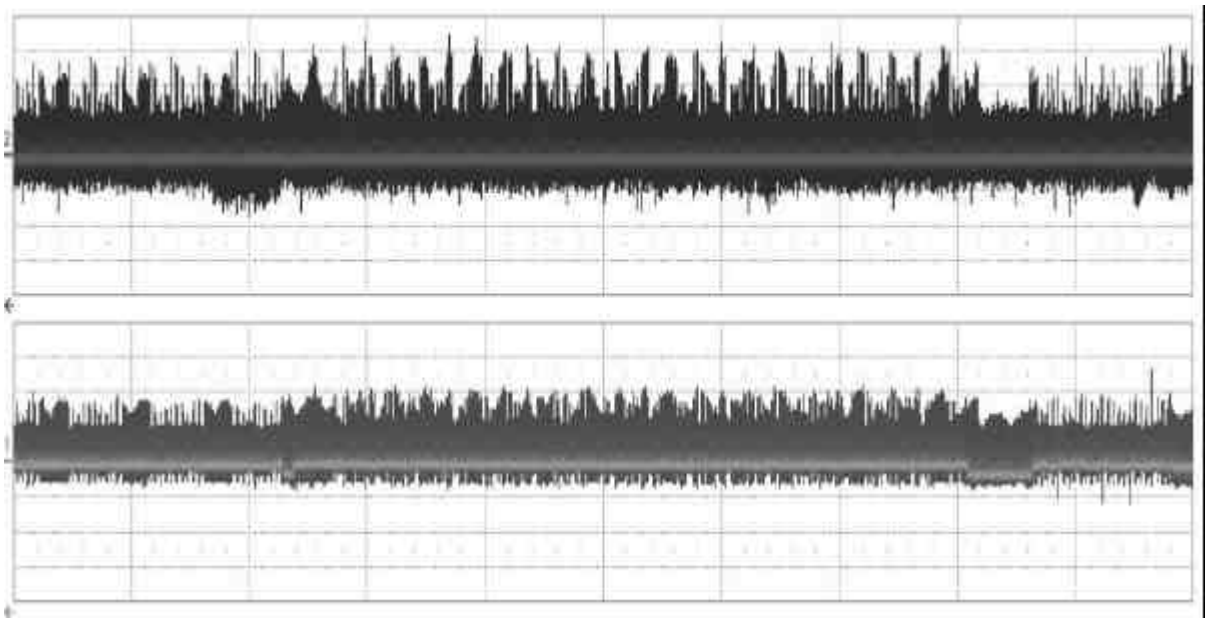


Figure 2 EM signal (upper) v Power Signal (lower) on DES

The complete waveforms result from a combination of card platform and application characteristics. Analysing the waveforms associated with running an algorithm can be a means to extract keys and so is often a tactic used for attacks[]. Signal capture is usually recommended at a rate of 1GS/s which suggests that frequency components of up to 500MHz are relevant. This is clearly way beyond the frequency of the logical I/O which typically runs at between 13-78kbits/s???. The rate is designed to capture

individual spikes associated with logic transitions that are related to the internal clock and configuration of low level hardware i.e. platform dependent properties. Combinations of spikes over the entire waveform are of course in some way dependent on the high level logical function, however it must be recalled that these applications are built upon virtual machines, APIs and operating systems whose implementations and characteristics are usually vendor and chip specific.

From the foregoing discussions it would appear that analysis of power/EM leakage signals shows some promise as card platform fingerprint detection. For this to be practical the factors that control the leakage should not be controllable from the logical application although clearly it makes a contribution to the leakage. The side-channel signal (scs) should be difficult to control precisely as it can be regarded as a function (f) of so many contributing factors including;

a) CPU type
b) Clock speed
c) chip circuit layout
d) Memory types
e) Power smoothing
f) Co-processors
g) Operating System
h) Drivers/APIs
i) Application code

$$scs = f\,(a,b,c,d,e,f,g,h,i) \tag{5}$$

These factors may not be precisely controlled but may be influenced as a function (f_P) of the various parties that co-operated to produce the smart card including the chip manufacturer (M), the card supplier (C) and the issuer (I) so the signal could be expressed as

$$scs = f_P\,(f_M, f_C, f_I) \tag{6}$$

Where;

$$f_M = f_m(a,b,c,d,e,f) \tag{7}$$

$$f_C = f_c\,(g,h) \tag{8}$$

$$f_I = f_i(i) \tag{9}$$

The contributions, f_M and f_C may be influenced during the design phase, however on a normal smart card platform it is only f_I that can be controlled (and to a limited extent) after card issue.

$$scs = f_P\,(f_M, f_C, f_I) \tag{10}$$

If the SCS detector relies on all the contributions then it will not easily be fooled by a duplicated application that can only control f_I.

Clone Card Detection via Fingerprint

Let us recap. A smart card consists of a few essential components. A secure chip incorporating processing and memory elements as well as low level security attack countermeasures plus an operating System and application environment. When used as part of a system solution there will be an application element, some related data storage plus sensitive keys and algorithms. The card communicates with the local reader device via a well-standardised interface channel and protocol. In operation the card will typically receive a challenge message via the local reader, execute a security algorithm using sensitive keys and return a result again via the reader. If the card has the correct functionality, stored information and keys the security function will produce the correct result and satisfy the system of the card legitimacy. In the case of a clone card it must be assumed that the logical functionality has been duplicated so at a local protocol level it would be difficult for an automated system to differentiate the clone card from the original. A potential solution to this problem is to allow the smart card to leak an identifier to the reader via the side-channel, which would be analogous to a "platform-fingerprint". A simplified block diagram is shown below for the case of a card inserted into a reader device such as an ATM

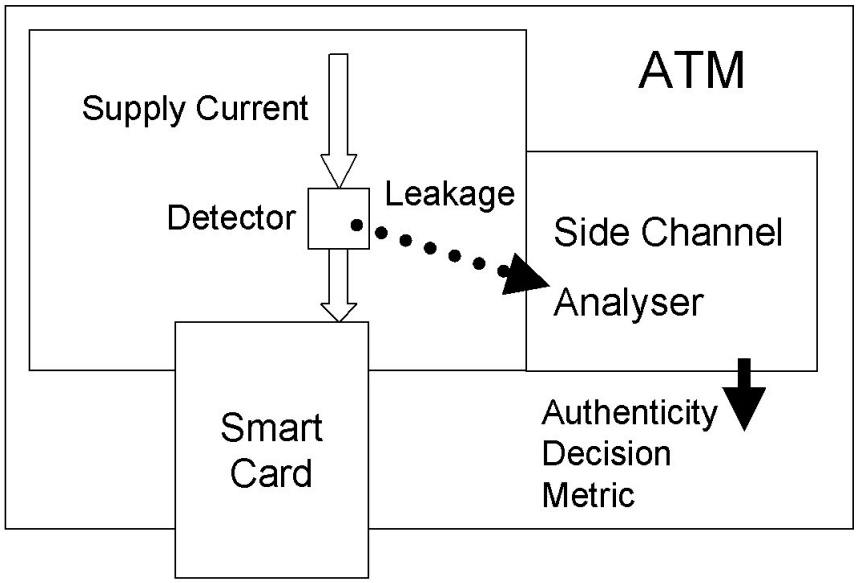


Figure 5 Possible Scenario for Clone detection via Side-Channel Analysis

The fingerprint waveform(s) resulting from natural leakage during start-up and/or normal operation could be sampled and analysed much like a Simple Power Analysis (SPA) attack[] and compared with a known reference. Discovery of the reference value is not necessarily a problem as the principle relies on the difficulty/practicality to fabricate a convincing waveform from a clone card.

Challenges

On first consideration it looks as if the smart card platform fingerprinting might be a means to detect and reject functionally correct clone platforms, however there are a number of significant practical challenges that would need to be overcome.

The challenge for the fingerprint detector is to be sufficiently tolerant of manufacturing variations in similar cards from the same manufacturer, whilst retaining the sensitivity to detect and reject other cards. This also implies that normal variations in environmental conditions that are known to affect side-channel leakage, such as temperature and supply voltage should not affect the comparison. Assuming that this is feasible then one effect is that the detector would be tuned to a particular card/chip from a single manufacturer. During the lifetime of a system there may be many chip/supplier combinations which are perfectly valid and so a strategy is needed to cope with this. There are a number of options including;

1) The reader tests the card against all known valid platform-fingerprints
2) The reader tests a card against its correct platform-fingerprint
3) The network tests a card against its correct platform-fingerprint

Option 1 requires no special knowledge of the particular card and the detector is independent of any functional signalling. Because it is a reader test, the clone can be immediately detected thus preventing any operation.
The disadvantage is that there may be many valid smart card platform combinations and detection may add processing overhead and delay. Each reader would also need to hold not only the latest detection algorithm but also all the valid platform-fingerprints, which adds to storage and management overheads.

Option 2 requires that the reader knows exactly the platform-fingerprint that should match the particular card. The reference would be held in an existing database and downloaded to the appropriate reader device as required. In the case of an ATM you might read the card ID, fetch the appropriate platform template and then invoke a function that can be used for platform-fingerprint generation and detection. In a mobile phone there may be signalling to report a new pairing of phone/SIM at which point the platform-fingerprint template may be extracted from the HLR database and downloaded to the phone. The SIM platform-fingerprint could then be checked on power-up and/or routinely as part of the SIM presence detection. This makes detection easier; however the mobile must permit some communication for the transfer of the platform-fingerprint. Once the template is loaded you can disable the card as soon as you detect a mismatch, providing that the phone is not locked permanently so that the subsequent insertion of a valid card is prevented.

Option 3 requires the least sophisticated detector. Basically the leakage signal/information is captured and reported back to the network. The network with its associated reference templates can then make a decision to block access. This could be handled in at least 2 ways e.g.

a) Block the account – this is simple but also blocks the legitimate user
b) Block the card from working in the reader device - this would just stop the particular reader/account combination but requires more control over the reader capabilities

Another important challenge arises because the smart cards must of course continue to protect themselves against side channel attacks. The countermeasure techniques tend to obscure and disguise side-channel leakage so that sensitive and security related information cannot be recovered by analysis. Clearly these remain important and vital properties but they will affect the overall information leakage from the platform not only in terms of a waveform shape but whether it consistently produced or

manipulated via variable delays. Ideally we would like a platform test function that executes without side-channel countermeasures, however as the countermeasures are usually strongly linked to the low level hardware it may be impossible to achieve this. The impact on detection may be that analysis of a short period of platform activity may be insufficient to make a confident decision about its validity. A common hacker approach to cards with countermeasures is to exploit statistical measures and techniques and this might be the route for the platform-fingerprint detector. The implication would be an assessment would take place over a longer time. For a card inserted into an ATM this may be a problem although in the case of a mobile phone there is a continuous communication path between the card and reader.

Conclusion

This paper has taken a creative look at the problem of clone card detection based on leakage of a "platform-fingerprint" from the card platform itself rather than the logical functionality of the device. The basic idea being to find a characteristic of the legitimate card platform that requires reasonably expert processing but which is also difficult to reproduce on a different card platform. An unusual approach has been considered by attempting positive use of side-channel leakage, which is normally only exploited to attack smart cards. There are however significant practical challenges, not least to manage the set of legitimate card templates. Perhaps the most interesting challenge would be to find an effective detector even in the presence of side-channel countermeasures and this might be more likely when the card and reader are capable of continuous interaction, as in the case of a mobile phone. The practicality of the approach could really only be determined by a lot of experimentation but it would be sweet revenge on the hackers if side-channel leakage could be used against them