Counter Expertise Review on the TNO Security Analysis of the Dutch OV-Chipkaart

OV-Chipkaart Security Issues – Tutorial for Non-Expert Readers

The current debate concerning the OV-Chipkaart security was triggered by some developments that are quite technical in nature and relate to specialist areas of expertise including cryptography, smart cards and general information or system security. In order to help improve understanding of the problems, this paper attempts to explain the core issues in simple terms. The main objective is to help decision makers form balanced and informed viewpoints on the matter. Whilst the focus will be on the Mifare Classic algorithm used for the OV-Chipkaart, it is important to realise that this forms only a part (albeit a critical one) of the overall system solution and the major concern is how any problem weakens the overall system security and then who would seek to exploit it and why.

Deciding on a System Security Solution

Before considering the OV-Chipkaart, let us first imagine a situation where cars are needed to transport two employees. Employee1 is very high-profile, critical to the business and perhaps at risk of a personal attack. Employee2 works mainly behind the scenes, but handles sensitive documents that thieves may attempt to steal. For Employee1 we might buy an expensive and impressive looking bullet-proof limousine, whereas for Employee2 it may be sufficient to buy a car that is simply safe and reliable to use and can be securely locked. Purchasing another limousine for Employee2 could be an unnecessary expense.

> *The simple example is meant to illustrate that the choice of a security solution and the associated cost depends on what we are trying to protect (the assets) and the anticipated types of attacks and attackers. There is also no such thing as 100% or fool-proof security, but a critical balance of risks against reasonable countermeasures – sometimes called a risk model.*

Keeping with cars as an example, we would expect a new car to give a good few years of trouble-free use. Thereafter it is still usable, but it may start to be a little more reliant on the "repair shop" and may need enhancing perhaps to protect more valuable items. If you are image conscious you might replace the car before problems starts to show, but if not, you could hang on until the car is no longer usable. This is meant to show;

> *A security solution has a life cycle. At the beginning, a properly designed security solution should be trouble-free, but as the technology begins to age there may be attacks against it. These attacks can often be held in check for some time by supplementary measures, but ultimately there will be a need for replacement (migration) with an updated solution. The decision to migrate will be influenced by the cost compared to losses from the old solution, and also by reputation/image reasons.*

The OV-Chipkaart as an example of a smart card enabled transport system, is protecting low-to-medium financial value and primarily against fraud attempts/attacks by reasonably equipped and resourceful criminals. One would therefore expect the risk model to identify the need for a medium (or higher) security smart card, based on technology that is still in a useful phase of its lifecycle.

A Brief Overview of the Cryptographic Security of the OV-Chipkaart System

The smart card used in the OV-Chipkaart system, as most smart card solutions that involve transactions with some financial value, includes security measures to protect both the service providers as well as the customers. General solutions may include the design of special security protocols, implementation of programs to perform some security task, as well as anti-tampering measures. In the OV-Chipkaart system, information is exchanged between cards and readers, and the goal of these security measures is to protect these messages and ensure that the system behaves as intended.

The OV-Chipkaart system uses the Mifare Classic card, and the security measures included in the card have two main objectives:

1. The card needs to prove to the reader that it is a genuine card. Likewise, the reader needs to prove to the card that it is a genuine reader. This process is called *mutual authentication,* and the goal is that both the reader and the card are convinced they are to start information exchange with a genuine peer.
2. It is desirable that all information exchanged between the card and the reader is protected against unauthorised eavesdropping and modification (potentially malicious).

These are classical problems in communication protocols, and how to address them is one of the main topics of the field of *cryptography*. One way to achieve this is to use *cryptographic algorithms* (or *ciphers*). These are mathematical algorithms that can be implemented either in software or hardware, and perform complex operations with the data. For example, an *encryption algorithm* can be used to *encrypt* (scramble) data such that the original message is unrecognisable. In general, such algorithms (and variants) can be used in security protocols to prove that all parties involved in the protocol are genuine, as well as to protect exchanged information. In the Mifare Classic card, this algorithm is known as CRYPTO1.

To achieve this goal however, it is required that the genuine, authorised parties (i.e. the card and the reader) share some secret information that is only known by them, and no one else[1]. In the case of the OV-Chipkaart system, these are:

1. The details of the cryptographic algorithm itself, including its structure and implementation;
2. A secret string of binary digits, or bits (i.e. 0s and 1s), that is used by the algorithm to perform the cryptographic operations in the data exchanged between the reader and the card. This string is called a *key,* and should be unique for each card (actually, each card has a number of keys). A genuine reader knows the keys for all cards (so it can communicate with any card); each card only knows its own keys.

We note that if an adversary who was trying to compromise the system knew both the algorithm and a particular card key, he could potentially construct a cloned card, write information in a genuine card, or simply eavesdrop the data exchange between a card and a reader. There are a number of scenarios in which an adversary could use his knowledge of the algorithm and the key to attack the system, and usually there would be further security measures in place within the system to (at least) limit the impact of such attacks. However it is reasonable to state that, to a certain extent, the security of the current OV-Chipkaart system relies on both the *algorithm* and *key* remaining secret.

1 Note in a transport system there are multiple readers that would know the card's secret.

Here we note however, the first security weakness in the OV-Chipkaart card: the security of the system seems to rely (at least partially) on the secrecy of the cryptographic algorithm itself. A well-known principle in cryptography states that a cryptosystem should not rely on the secrecy of the algorithm for its security. This is known as *Kerckhoffs' principle*, named after the 19th century Dutch cryptographer Auguste Kerckhoffs. This reliance is known amongst the security community as "security by obscurity" and is widely derided as a flawed design principle for security systems. There a number of reasons for this. First, it is widely accepted that sooner or later the algorithm will be *leaked* (e.g. by "trusted" employees, sub-contractors or lost documents) or simply recovered, by reverse-engineering of the card. The latter method is exactly what happened with the algorithm used in the OV-Chipkaart smart card.

Second, due to its closed/proprietary design, it is very likely that CRYPTO1 has received relatively limited assessment of its security. It is again a widely accepted best practice in cryptographic systems to only use algorithms that have been openly evaluated by the expert cryptographic community. Such a review could potentially uncover and repair some structural weaknesses that may have not been obvious to the cipher's designers (and closed circle of internal evaluators) and thus an open peer review gives stronger assurance that the algorithm is robust and secure.

If we ignore for the moment whether CRYPTO1 has any structural weaknesses then does disclosure of the algorithm necessarily mean that the OV-Chipkaart cards are not secure? Well the answer now depends on another piece of information shared between the card and the reader: a secret key. Knowledge of one of the shared secrets (the OV-Chipkaart algorithm) does not necessarily imply knowledge of the other shared secret (the key). In fact, *Kerckhoffs' principle* states that a cryptosystem should remain secure if *all* the details of the system, except the key, are of public knowledge, and thus secrecy of the key should be *sufficient* for the OV-Chipkaart card to remain secure. However, here we find a second weakness of the OV-Chipkaart.

The secret key is used by the algorithm as input to *encrypt* and *decrypt* data exchanged between the card and the reader. The secret key is a string of bits of fixed length, so if you know the algorithm and have observed some exchange of information between a card and a reader that was protected using a specific key, you could simply try all strings of same length (i.e. all possible keys) until you find the correct one. This is a form of attack that is applicable to all ciphers and it is known as *exhaustive key search* (or *brute force*) attack. The ultimate goal of a cryptographic algorithm designer is that this form of attack is the most efficient attack against the cipher. The designer then has to ensure that the number of possible keys is so large, that it would be impractical for even a well equipped adversary to recover a secret key via exhaustive search.

Keys are essentially strings of bits of certain fixed length (say n), thus there are as many keys as bit strings of length n. So if n=1, we have only 2 possible keys: 0 and 1. If n=2, we have 4 possible keys, namely 00, 01, 10 and 11. If n=3, we have 8 possible keys: 000, 001, 010, 011, 100, 101, 110 and 111. And in general, for any n, we have 2^n distinct n-bit strings, and thus 2^n possible keys.

The OV-Chipkaart algorithm uses keys of 48 bits in length. So there are 2^{48} possible keys, which is approximately 3×10^{14} (i.e. the digit 3 followed by fourteen 0s). This looks like a very large number, and a brute force attack would have to try (on average) almost all the 48-bit strings to find the correct key. This may appear impossible, but in practice it is relatively easily achievable with purpose-built computer devices using current technology. By way of illustration, a well-known cryptographic algorithm called the Data Encryption Standard (DES), which was until recently a US

government standard and has been widely used by both government and industry alike, is now considered obsolete because a few years ago an exhaustive search attack that could reveal its 56-bit key was practically demonstrated. We note that there are 256 times more DES keys than there are CRYPTO1 keys (as $2^{56} = 256 \times 2^{48}$) and although the specific details of the attack on DES may not be directly applicable against CRYPTO1, it should be reasonable to assume that an exhaustive key search against the CRYPTO1 algorithm is quite practical.

Moreover there are further forms of optimisation that could be used to speed up this brute force attack. Overall, it is widely agreed that the length of the key used by the OV-Chipkaart algorithm is just too short for today's standards. Most modern ciphers use keys with length between 80 bits and 256 bits (the successor of DES, called AES, uses keys of length 128, 192 and 256 bits). This is perhaps the most obvious weakness in the algorithm used in the OV-Chipkaart system, and with such short key length, the security measures provided by the card alone are at best of limited effectiveness.

Unfortunately, the short key length is not the only problem for CRYPTO1 as it seems to suffer from another weakness. Research results, following the disclosure of the OV-Chipkaart cryptographic algorithm seem to suggest that CRYPTO1 does indeed present structural weaknesses; that can be exploited to compromise the security of the card. This suggests attack optimisations that are far more efficient that exhaustive key search.

Random Numbers

A further weakness demonstrated in the recent public presentation is not directly related to the CRYPTO1 algorithm, but rather on how it is used in the protocol between the card and the reader. In addition to the secret key (and the message), the algorithm takes two other inputs: the card ID number, and a number R, which is freshly generated in every new transaction and is used to ensure that the algorithm behaves in different way every time it encrypts/process data in the security protocol. This is particularly important to protect against a form of attack called a *replay attack,* where an attacker, possibly the holder of a genuine card, repeats a previous exchange of data between the card and a reader, potentially modifying the data held in the card (e.g. travel credit) in a fraudulent way.

Thus you would like this number R to be *random*, i.e. essentially unpredictable and hard-to-guess. However the researchers who reverse engineered the Mifare Classic card's cryptographic algorithm, claimed that there are relatively few variations for this number R and with careful timing, one can control the generation of the "random" number that is used each time, virtually eliminating it as a safeguard against attacks.

The Reality of Attacks

We mentioned earlier that there is no such thing as 100% security and in principle any target can be attacked given sufficient time, expertise, resources and motivation. Attacks become of concern when they are shown to be feasible with available equipment and this concern escalates as the attack time, necessary expertise and equipment costs fall. The brute force attack is crude, but effective with known equipment and so the question for the experts is not whether an attack is possible, but whether there are cheaper and faster alternatives. Even so, just because an attack is practical does not necessarily mean that it is worth attempting (ignoring proof-of-concept demonstrations). The real test is when the value to be had from a criminal attack on the system, outweighs the cost and effort of carrying out the attack and of course the risk of being caught. Therefore in a transport system, the ticket cost is an important factor. Whilst some season tickets can be quite expensive, it is better to consider fraud on a day travel basis, as a counterfeit season ticket product will not sell for thousands of Euros if it might be disabled within a few days. We should remember that transport systems are not completely defenceless even if they are valuable attack targets and even if their card algorithms can no longer be relied upon. For many years, transport system operators have been dealing with “resourceful” people who would seek to defraud them and there are many back-office detective and corrective measures in place that can be used to limit such abuse, even when using the most basic paper and magnetic stripe tickets. Of course these last lines of defence were not designed to resist a sustained technical assault and so the best policy of course is to use appropriate smart card technology that resists all anticipated attacks.

APPENDIX A

The explanation in the main body of this tutorial should provide sufficient clarification for most readers, however some may have noted that CRYPTO1 is described as a "stream cipher" and be curious about what this means. A simplified example of such a cipher is shown in Figure 1.

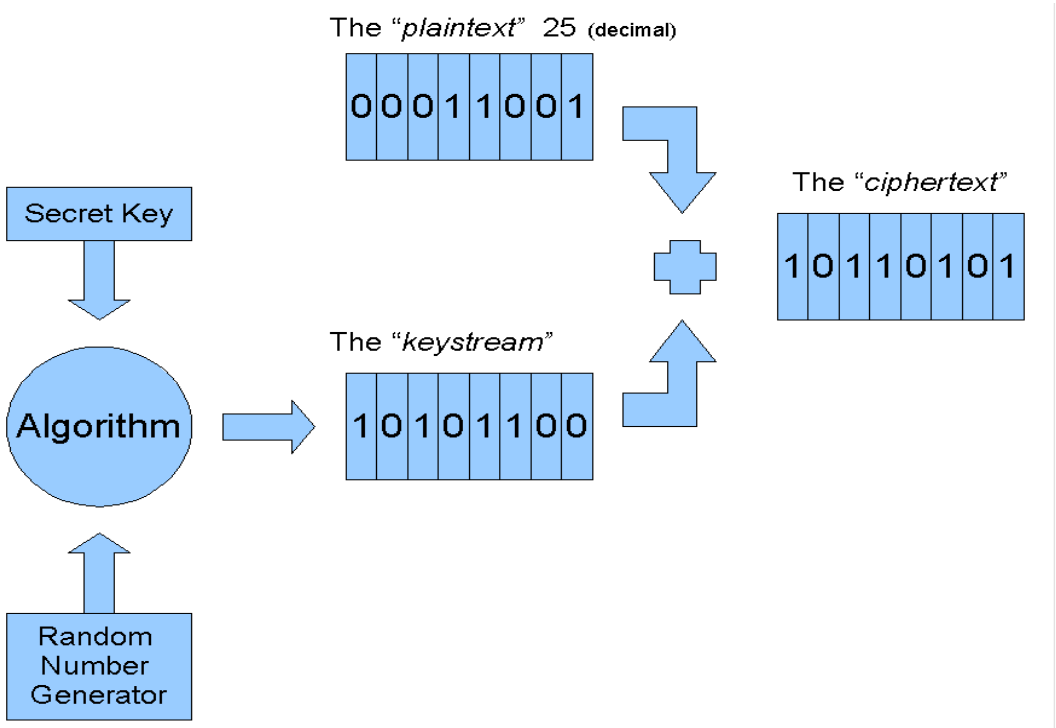

Figure 1 Simplified Stream Cipher Example

In Figure 1, what we wish to send can be considered as a stream of information that is ultimately represented by a stream of bits that can take the value 0 or 1. This is sometimes called the *plaintext* and in the example we have shown how the number 25 could appear as a binary string representation. So that an attacker cannot see our sequence of 0s and 1s (and discover we are sending the number 25), we add the stream (in a very simple bit-wise[2] fashion) to something we call the *keystream*. The *keystream* has been generated by our algorithm using a number of inputs including the secret key. The resulting output stream that is transmitted to the reader (called the *ciphertext*) is no longer recognisable as our *plaintext* and so an attacker should not be able to determine what information was sent. The reader device also has the algorithm and key and so can generate the same *keystream* that was used to encrypt the message. If the *keystream* is simply added again to the ciphertext, the *plaintext* is revealed.

2 Adding (XOR) is done a bit at a time i.e. if we add 0+0 or 1+1 the result is 0, if we add 0+1 or 1+0 the result is 1