On Bad Randomness and Cloning of Contactless Payment and Building Smart Cards

This work was supported by the European Commission under the FP7 project number 242497 Resilient Infrastructure and Building Security (RIBS) and by the Polish Ministry of Science and Higher Education under research project Nr 0 R00 0111 12.

Nicolas T. Courtois[1], Daniel Hulme[1,2] , Kumail Hussain[1]
[1] *University College London,UK*
[2] *NP-Complete Ltd, London, UK*
Email: letter.name@cs.ucl.ac.uk

Jerzy A. Gawinecki[3], Marek Grajek[4]
[3] *Military University of Technology, Warsaw, Poland*
[4] *Independent researcher and writer, Poland*
Email: JGawinecki@wat.edu.pl
Email: marekjg@interia.pl

***Abstract*—In this paper we study the randomness of some random numbers found in real-life smart card products. We have studied a number of symmetric keys, codes and random nonces in the most prominent contactless smart cards used in buildings, small payments and public transportation used by hundreds of millions of people every day. Furthermore we investigate a number of technical questions in order to see to what extent the vulnerabilities we have discovered could be exploited by criminals.**

In particular we look at the case MiFare Classic cards, of which some two hundred million are still in use worldwide. We have examined some 50 real-life cards from different countries to discover that it is not entirely clear if what was previously written about this topic is entirely correct. These facts are highly relevant to the practical feasibility of card cloning in order to enter some buildings, make small purchases or in public transportation in many countries. We also show examples of serious security issues due to poor entropy with another very popular contactless smart card used in many buildings worldwide.

***Keywords*-Random Number Generators (RNG), human factors, cryptography, smart cards, RFID, building access control, contactless payments, HID Prox, HID iClass, MiFare Classic.**

I. Background and History

A good Random Number Generator (RNG) is a crucial element in many security products. Random nonces are required to avoid numerous attacks on protocols such as replay attacks, and ensure freshness of keys and independence of various messages. Randomness can come from a physical source or from cryptographic hardness, and both are frequently combined in real-life products. If there is no physical source and the RNG is fully deterministic we speak about a (cryptographic) Pseudo-Random Number Generator (PRNG).

A. *Some Historical Background*

Throughout history, the same mistakes have been committed again and again. Bad randomness has always been 'the best friend' of code-breakers. Moreover the randomness seems to always be somewhat bad. It seems that when systems evolve or procedures change, or the security awareness improves, the improvement is at most incremental. In fact people are inclined to commit the same mistakes over again, with small incremental improvements.

For decades, random numbers were generated by humans. It is known that before WW2 German cipher clerks were frequently using Enigma message keys of type AAA or XYZ [15]. German security services must have noticed this and they have disappeared. Only to be replaced by consecutive letters on the keyboard like QWE. Once these have been eliminated, keyboard diagonals like QAY were the next trend [15]. Today ALL and EXACTLY THE SAME trends are still present as distinct patterns for example in the probability distributions of real-life passwords [18]. Similar problems will occur with human-generated cryptographic keys serial numbers, nonces and other codes, as will be seen later.

In modern systems an increasing proportion of random numbers is generated by machines not by humans. Strangely enough the situation does NOT seem to improve. Our research shows that very frequently numbers which should be random are NOT random. In smart cards quality random numbers may be an expensive resource however this happens even when they are generated by powerful computers, for example in Microsoft Windows [6].

In fact it seems that engineers are not able to get it right and it became a serious problem in cryptology. Building a secure random number generator requires much more than to produce sequences which at "look" random and pass statistical tests. If the numbers are generated by a cryptographic PRNG, this generator is subject to more or less all classical key recovery attacks on stream ciphers: correlation attacks, algebraic attacks, etc. cf. [4]. Yet new attacks on symmetric cryptography are invented every year.

Another topic which has a substantial history of mistakes is key reuse. This again happened many times during WW2 [15] and happens on a regular basis today. In the same way it is a recurrent bad practice to reuse passwords, yet there is no way to stop users from doing it.

In this paper we will look at the quality of random numbers used as random nonces, cryptographic keys (randomness and re-use) unique serial numbers and other codes in the most widely used contactless smart card systems in building access control, public transportation and small payments. In the next section we overview the security of the most widely used MiFare Classic card. We will then present several real-life cases of how a poor RNG affects their security and card cloning in three different countries. In Section V we study the (poor) entropy of serial numbers of some real-life HID smart cards from 4 different countries.

II. Insufficiently Random Numbers In MiFare Classic

One of the several famous cyber [in]security events in the last years was the collapse of the security of MiFare Classic contactless smart card. This concerns about 70% of all contactless smart cards used worldwide, more than 1 billion cards sold to date, and which are massively used still today, in public transportation (e.g. London Oyster Card), in access control in numerous buildings worldwide and even for small payments. There is already ample literature on this topic. However even today when the topic has reached maturity, some crucial practical details relating to the practical feasibility of various attacks are left unspecified. There are two main reasons for this. In essence the complexity and feasibility of many attacks depends in a strong way on two things:

1) the quality of the random number generator on the card, and
2) how the application which is built around this card generates and manages cryptographic keys (and on randomness of these keys).

In this paper we cover both questions. Only a detailed study of a number of smart cards from different countries and from tens of different application providers can give us some (incomplete) answers to these questions. First we are going to outline very briefly the security of MiFare Classic and explain why and how weak randomness play an essential role in it.

For a decade, the customers of MiFare Classic were left in complete obscurity. Then in 2008 the specification was reverse-engineered [7]. It took about 2 more years to discover how (and how badly) this product can be broken. Early attacks were lightening fast [8], [12] but only in theory. In practice they were very hard to execute. Only in late 2009 we have seen attacks which do NOT require any access to a legitimate reader in the building, which makes most early attacks very hard to execute in practice. These recent "card-only" attacks [17], [10], [11] require minimum access, can be executed at any moment and (in theory) can also be super fast (but only for certain cards, as we are going to see later). In this space we have three very important recent attacks:

1) One super-fast attack by the Nijmegen group [17] requires a very costly pre-computation and hundreds of Gigabytes of storage, to extract keys instantly.
2) If we exclude this attack which is not feasible for ordinary hackers, the so called "Courtois Dark Side attack" [10], [11], [5] is the most popular and the most practical way of extracting keys from these cards. Several implementations of this attack exist, notably MFCUK [5] which works with some of the cheapest contactless card readers available.
3) However this attack is needed only if all the keys in the card are random and unique. If we already know at least one key for a given card, or some default or application-wide keys are used or re-used, all the other keys can be recovered instantly. This is achieved with the so called "Nested Authentication Attack" also by the Nijmegen group [17].

The "Courtois Dark Side attack" [10], [11], [5] is therefore, in many cases but not in all cases, a very plausible first step for an ordinary criminal. This attack depends on three crucial vulnerabilities:

1) In the MiFare Classic card, data to be transmitted are expanded with parity bits, then encrypted with a stream cipher. This is another classic mistake in cryptography, known since the early WW2 Enigma double-indicator system [15]. Exactly the same mistake allows one to break the confidentiality of GSM mobile phone system cf. [2].
   This vulnerability alone is not sufficient. It remains still quite difficult to recover secret keys from the card in any way. This is due to a simple but important security engineering principle. The card never answers anything related to the secret key. This unless the reader is authenticated first [11], [13]. This property makes "card-only attacks" virtually impossible.
2) Interestingly however, there is a bug in this product. Sometimes, the card will actually nevertheless respond to a query. This happens with a low probability of 2^{-8} and can be easily overlooked. We have a sort of "backdoor" which allows the secret key to be extracted, see [11], [17], [10], [5].
3) This is still not good enough. The "Courtois Dark Side attack" [10], [11], [5] relies on yet another very serious vulnerability of the card. The random number generator is typically quite weak in these cards and the attacker can try to reproduce the same random numbers.

The last question of RNG manipulation is crucial for the attack of [10], [11], [5] to be really efficient in practice. Up till now no paper has studied this question in detail.

We need to consider the following:

1) the behavior of different cards used in the real life with detailed statistical analysis
2) How the attacker can influence the RNG and manipulate it in the best possible way (hard). This potentially could lead to further "adaptive" attacks: in which the attacker is able to adapt in the best possible way to the observable characteristics of the random number generator in question.

3) How all this can affect the time complexity of the attack from [10], [11] and its practical implementations [5] which have additional issues such as slow timing or imprecise control.

Clearly there is a lot to be studied. We have examined some 50 real-life cards from different countries. Interestingly for some cards it is not even clear if what was previously written about this topic is at all true. In what follows we are going to see that no smart card we have ever seen confirms exactly what we read in [7], which fact is highly relevant to the practical complexity of the attack in question.

III. The MiFare Classic RNG Weakness

The random number generator in MiFare classic provides random numbers on 32 bits. Interestingly these 32 bits are always redundant (we have never found a counter-example) and depend linearly on only 16 bits. This already is an unnecessary weakness which looks like a voluntary limitation. However this property is impossible to hide. We can assume that the attacker will observe it as soon as he tries a few cards. Moreover we can consider that 16 bits of entropy are sufficient for many applications [1].

Further serious problems with this RNG were made public by Nohl *et al.* [7]. We summarize their claims.

1) First of all they present the connection of a 16-bit LFSR which is used to generate the basic random on 16 bits (later expanded to 32 bits).
2) Secondly they have observed (due to the reverse engineering) that when the card is powered up, the RNG is reset to the initial state. This reset is as much unnecessary and more costly as it is dangerous. A very bad choice which is very hard to defend. Not resetting the RNG when the card is powered off would make it much harder to predict and manipulate, due to the remaining charges inside the card silicon.
3) Moreover, the behavior is claimed to be fully deterministic since the card is powered up, and a strict control of timing allows either to predict the random, or even to produce the desired random at will for the attacker. We say "claimed" because our investigation never fully confirmed the claims of [7].
4) The connection polynomial of the LFSR is claimed to be $x^{16} + x^{14} + x^{13} + x^{11} + 1$. Interestingly it could be different in different cards.
5) Further, it is claimed that the clocking is regular and the LFSR is clocked at 106 kHz and wraps around every 0.6 seconds, after generating all 65,535 possible output values. In real life there could be additional complexity or variable speeds, or additional sources of entropy.
6) The same is also claimed to exists on the reader side. In this paper we only study the card RNG.
7) Overall and in addition, Nohl *et al.* [7] have claimed that one can reproduce exactly the same random each time with accuracy close to 100%.

As we will see later the claims above seem to be close to accurate only for some older cards such as Oyster cards from 2006. To the best of our knowledge [our own embedded firmware implementation of the attack of [10], open implementation with different hardware [5], discussions with the Nijmegen group], nobody has ever achieved what is claimed in [7] on any recent real-life smart card.

It must be noted that it is in general quite **difficult** to see if what is claimed in [7] is correct. In order to verify the claims one needs to be able to either sniff and decode the communications between the card and the reader, or to control the timing with very high accuracy, acquire a lot of data and and perform a lot of statistic treatment and analysis, while accounting for the imperfections of the equipment, antennas and reception/decoding errors which are important with the RFID technology[2]

A. *Is The Theory Correct And To Which Cards It Applies?*

In order to evaluate whether the theory model of [7] is correct, we proceed as follows. We consider the LFSR with the connection polynomial $x^{16} + x^{14} + x^{13} + x^{11} + 1$ given in [7]. This LFSR implies a natural ordering of the $2^{16} - 1$ values which appears on the x axis of many of our graphs, and each random 32-bit nonce can be compressed to a number between $1 \ldots 2^{16} - 1$. Each MiFare Classic card can be modeled as a black box which given a time t in microseconds after powering up the card, produces a certain distribution of values between $1 \ldots 2^{16} - 1$. which are more or less random and more or less clustered around certain values on the x axis, which are achieved by the attacker with imperfect equipment. The goal of the attacker is to produce a distribution with very low entropy where few nonces would repeat many times, and the attacker can try to adapt to the data he observes.

If the theory of [7] is correct, at identical periods of time we will obtain values which are close on the x axis on our graph. However we expect some additional problems due to the imperfections in our attack setup: lack of precision in our timing and more importantly imperfect control of our device. We need to take into account various perturbating factors such as hardware interruptions in the firmware and various delays in the commands sent through USB port. Therefore even if the theory is correct we do not expect just one "sharp" peak on our graph, but possibly several "shadow"

[1] In applied cryptography a "nonce" means "*n*umber used only *once*". All that is required is to be unique and never repeated. Full randomness is NOT needed in many security protocols. Full entropy is typically only crucial for key generation, not for nonces.

[2] These technical reasons may explain why for some 15 years MiFare Classic did not have many serious competitors, and competing firms on the market such as HID have introduced their first RFID card for building which is the first to use "real" cryptographic security only about a decade later.

peaks which could be more diffuse. If the theory is wrong, we might obtain a completely flat (uniform) distribution.

B. A UK University Card

On Figure 1 we present one example of what happens with a building card of a university located in central London when running the MFCUK attack with command line options ”-s40 -S60” for 3 hours. By looking at this sample it is hard to see if the theory of [7] is correct at all.

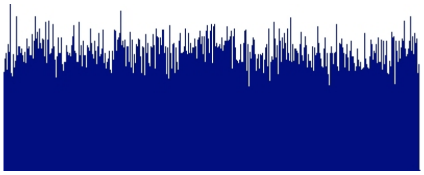

Figure 1. The distribution obtained for a UK university building card

Another question is whether this could be sampled from a uniform distribution. This is not obvious given a relatively small sample size of $n = 36,876$ different points which is the exact number which we have acquired in this experimentation. In order to attempt to answer this question we look at the number of different card nonces obtained. This value is independent of the ordering on our x axis and therefore has nothing to do with the theory of [7]. However it can be compared to what we obtain on average for $n = 36,876$ truly random points. It is possible to see that with $36,876$ random numbers on 16 bits, we should obtain on average about $28,200$ with standard deviation of about 66. We observed 27689 different values which are less, and at about 7.8 standard deviations. We can therefore believe that the distribution is NOT random. Furthermore, from the observed probabilities we have estimated that the entropy of the card random in our attack is about 14.6 bits instead of 16 expected in the case of a uniform distribution. The Min-Entropy observed is about 12.4 bits.

We see that in this attack the distribution is not uniform. However from the point of view of the attacker it is not good, not much better than uniform. These smart cards are hard to break and the attack takes about one day instead of seconds expected if we could reproduce the same card nonce perfectly as assumed in [10], [5]. A big difference between theory and practice.

C. A Malaysian Small Payment Smart Card

We have repeated the same experiment with a Touch N Go smart card in Malaysia. The results are much more worrisome, see Figure 2.

Here we observe very strong peaks, and moreover the peaks are evenly spaced which confirms the theory of [7]. The entropy of the random number in our attack is about 7.0 bits for this card, instead of 16. The Min-Entropy is about 4.4 bits. For simplicity let’s assume that we concentrate our attack on just one value of the card random nonce, the most frequently repeated value. Under this assumption we expect that the attack will require just about $2^{4.4} \cdot 300$ queries. This is estimated knowing that the attack takes 300 queries on average following [10], [11]. This will be about 500 seconds to recover one key at the current speed which is about 13 queries per second.

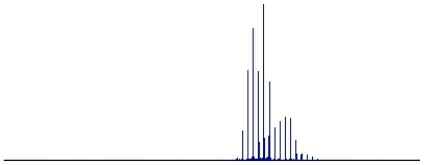

Figure 2. The distribution obtained for a recent Malaysia payment card

The Malaysia cards are **very easy** to break by the attack of [10], [5]. This attack could be executed when sitting near the card owner and without raising any suspicion. This is particularly problematic because the Malaysia Touch N Go card is a MiFare Classic card AND it is used not only in public transportation as it is/was in Warsaw/London and other places. It is also used to pay in shops, fast food restaurants, movie theaters, etc. This cards opens much bigger opportunities for crime and criminal business and is much easier to break than any other card.

The only good thing about it is that cryptographic keys in these cards are different in each card. If a hacker can clone one card, it does NOT affect the security of other cards.

D. London Oyster Card

The attacks described in [10], [17] do NOT work on any recent Oyster card, because all new Oyster cards issued shortly after the publication of these two papers in late 2009 use now a different chip (DesFire). This chip is expected to be cryptographically secure. However many Oyster cards issued before 2010 are still in circulation. We do not have precise data but this could be somewhere between 5 and maybe 30% of all Oyster cards in circulation in the UK. Some of these existing Oyster cards can be broken very easily. On Figure 3 we show what we obtain for an Oyster card which was purchased in 2006, which still works and is used every day in London.

This card is less secure than the Malaysia card. The entropy is about 5.8 and the Min-Entropy is only about 2.8 bits. This means that the attack will require about $2^{2.8} \cdot 300$ queries [10] and should take about 160 seconds per key with the implementation of [5].

This 2006 card is quite old, however it is the worst we have seen among cards in circulation. As far as we can see no attempt was made by Transport for London (TfL)

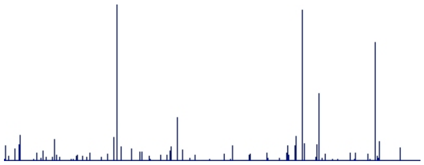

Figure 3. The random nonce distribution obtained for a 2006 Oyster card

authority to incite people who own and use such cards every day to upgrade them to new, more secure cards (DesFire).

Here the opportunities for criminals are not as substantial because in these cards the cryptographic keys are again diversified and only a limited number of such cards are still used. In addition London card systems are online and should be able to detect fraud. Another Oyster card from 2007 we have examined had min-entropy of 13.4 bits which is already relatively good and the attack following [5] with the same setup would take about 3 days per key.

IV. Key Diversification and Key Randomness in Contactless Cards

We have examined many other cards. One of the key problems we have seen is that at many places is that exactly the same cryptographic keys are used in vast numbers of smart cards. This is the case in many buildings in London and elsewhere. An absolute horror story are the cryptographic keys found in MiFare Classic cards used in Poland. Here the keys are NOT random and they look very much like they were generated by a human. Their entropy is excessively low. For example some secret keys literally start with 898989 in hex [11]. In addition the same cryptographic keys are used in many different cards issued by different entities, including transport cards, student cards at several universities and other buildings, including places with highly-ranking military personnel. We don't disclose further details.

V. Other Building Cards

The second most popular smart card type in buildings is HID. HID has two basic sorts of cards: HID's Prox 125kHz card which has no cryptographic security, it just transmits a serial number, and is still very widely used. The HID iClass card is a cryptographic card which can also store data in non-volatile memory. In order to facilitate migration from (old and insecure) Prox cards to more secure(?) cryptographic iClass cards, many door-entry systems will accept both cards, and what is transmitted by the reader is a 26-bit code composed of a facility code and a unique code for each card. Unhappily there is no guarantee that the cryptographic card is more secure. It might actually be less secure. Following [14] the cryptography in these cards is not very strong, important master keys can be extracted from readers, and moreover unlike the old HID Prox and all MiFare Classic cards, these cards do not have "hardware security" in the sense that their serial number can be changed. They can be re-programmed and copied without forcing the attacker to use an expensive hardware card emulator.

We have discovered another issue with HID cards. We have examined some cards from a major corporation with offices in California, London and elsewhere, a real bank situated in an EU country, and from an airport in another EU country. There is little doubt that all these cards can be cloned by hackers cf. [14]. However are there any easier, low-tech attacks? Are serial numbers of the cards actually random? We have compared the Wiegand data transmitted to the back-end systems from one reader for different cards. These data already have critically low entropy of at most 24 bits typically. Many longer variants of Wiegand format exist however it seems that they are less widely used. Unhappily the facility code on 8 bits (typically) is fixed for all cards in a given domain. We are left with 16 bits of entropy at maximum. Any further reduction in the entropy can be fatal.

We have checked only very few cards. However clearly the entropy of these Wiegand data is very poor. Firstly we observed that the facility codes can be the same in different domains of application. For example we have purchased few HID iClass cards which are sold to individual users of laptops and one Prox card from a bank. To our horror both cards had the same facility code. Moreover the serial numbers on 16 bits clearly have very poor entropy; the difference between the number for the bank and a random card which could be owned by any laptop user was 116. This suggest that, due to the birthday paradox, and most probably if we take 10 cards of different bank employees and 10 cards from different laptop users, both facility code and unique card number may become identical purely by accident. One of the laptop users will be able to penetrate inside the bank. This, depending on the smart card readers used by the bank, assuming that only 26-bit of basic Wiegand data are transmitted to the back-end for both cards (a legacy mode).

The cards from the airport we have examined seem slightly more secure. Even though they have a very standard facility code (which is likely to repeat elsewhere), the entropy of the serial numbers seems to be at least 12 bits out of 16. Unfortunately these numbers were consecutive for different cards. This decreases the amount of data which may be available to forensic investigators. The attacker could easily copy a card of one employee, modify the number within a certain interval, obtain another valid card, and penetrate into the building without leaving any traces and without the possibility to connect this incident to any concrete card belonging to a concrete person.

VI. Conclusion

In this paper we looked at the quality of random numbers cryptographic keys, nonces and other codes in real life building security and small payment systems. In particular we looked at MiFare Classic cards hundreds of millions of which are still in massive use worldwide. We have examined some 50 real-life cards from different countries in order to evaluate the practical feasibility of some previously known attacks. In many cases to evaluate the quality of the random numbers is crucial in order to determine to what extent these attacks will really work. We report a number of concrete facts about practical difficulty of card cloning in order to enter some buildings, make small purchases or in public transportation in several countries. For example we have demonstrated that with an open source implementation of our earlier attack [5], [10] it is still possible in 2013 to wirelessly steal cards of fellow passengers for some fraction of smart cards used in London and many more in Malaysia. We also have discovered some minor issues with certain HID cards used at airports and in financial institutions and we present a super simple low-tech attack which is likely to work in a real-life bank building. All these examples show that bad random numbers are likely to facilitate crime.

One interesting recurrent pattern in our research is that every single mistake which could possibly be made, seem to be always made. The security seems to improve extremely slowly, with small incremental steps. There is no doubt that the industry offers now more secure smart card systems and we have shown that more recent smart cards have better random generators. However the customers and end users need to be aware of the security issues in order to upgrade or adopt better solutions. Our recent survey on smart cards used in UK buildings [3] shows that most people do not know what kind of cards they have and what kind of security they have. Many smart card systems are sold under obscure brands and integrated with larger security systems. There is no visibility about what is inside. Businesses who bought these systems do not know if the cloning or other attacks about which they have heard a lot from the press would apply to their building or payment system. Excessive attention goes into hacking while more secure solutions which exist seem to suffer from insufficient visibility.

References

[1] A. Adebanke, *Security of Smart Cards in Building Access Control Systems*, Master thesis, M.Sc. in Information Security, University College London, September 2012.

[2] E. Barkan, E. Biham, and N. Keller, *Instant ciphertext-only cryptanalysis of GSM encrypted communication*, In Journal of Cryptology 21 (2008), no. 3, pp. 392-429.

[3] N. Courtois, D. Hulme and K. Gupta: *Building and Transport Cards: Attacks and Defences*, presentation given at Chip to Cloud security forum, September 19-20 2012, Nice, French Riviera, CDROM and web proceedings, 2012.

[4] N. Courtois and W. Meier, *Algebraic Attacks on Stream Ciphers with Linear Feedback*, Eurocrypt 2003, Warsaw, Poland, LNCS 2656, pp. 345-359, Springer, 2003.

[5] A. Costin, *MFCUK*, an open source C implementation of the Courtois Dark Side attack https://code.google.com/p/mfcuk/

[6] L. Dorrendorf, Z. G. : *Cryptanalysis of the Random Number Generator of the Windows*, Jerusalem, The Hebrew Univ., 2007.

[7] K. Nohl, D. Evans, Starbug, and H. Plötz, *Reverse-Engineering a Cryptographic RFID Tag*, In Usenix Sec. Symp., pp. 185-194, 2008.

[8] N. Courtois, K. Nohl, S. O'Neil: *Algebraic Attacks on MiFare RFID Chips*, Slides available at: http://www.nicolascourtois.com/papers/mifare_rump_ec08.pdf.

[9] N. Courtois: *La Carte à Puce*, 293 slides in English, overview of smart card technology, part of COMPGA12 course taught at University College London in 2007-2013, http://www0.cs.ucl.ac.uk/staff/n.courtois/smartc.pdf

[10] N. Courtois: *The Dark Side of Security by Obscurity and Cloning MiFare Classic Rail and Building Passes Anywhere, Anytime*, In SECRYPT 2009 International Conference on Security and Cryptography: pp. 331-338. INSTICC Press 2009.

[11] N. Courtois, Slides about MiFare Classic, extended version of the talk given at the 2009 workshop on RFID security held at Leuven, Belgium, available at http://www.nicolascourtois.com/papers/mifare_all.pdf.

[12] F. D. Garcia, G. de Koning Gans, R. Muijrers, P. Van Rossum, R. Verdult, R. Schreur, B. Jacobs: *Dismantling MIFARE Classic*, In Esorics 2008, pp. 97-114, 2008

[13] K. Gupta: *Engineering Access Control Systems*, Master thesis, M.Sc. in Information Security, University College London, September 2011.

[14] M. Meriac: *Heart of darkness - exploring the uncharted backwaters of hid iclass(TM) security*, In 24th Chaos Communication Congress, December 2010.

[15] M. Rejewski: *Memories of my work at the Cipher Bureau of the General Staff Second Department 1930-1945*, Adam Mickiewicz University Press, Poznań, Poland, 2011, pp. 28-33.

[16] A. S. Rosli: *Contactless Smart Cards in Malaysia*, Master thesis, M.Sc. in Information Security, University College London, September 2012.

[17] F. D. Garcia, P. Rossum, R. Verdult and R. W. Schreur: Wirelessly Pickpocketing a Mifare Classic Card. In *IEEE Symp. on Security and Privacy, Oakland, May 2009*.

[18] A. Vance, *If Your Password Is 123456, Just Make It HackMe*, The New York Times, 20 Jan 2010. http://www.nytimes.com/2010/01/21/technology/21password.html?_r=0