Is decimalisation table attack really a major threat?

Ji SUN
Computer Science Department, University of Auckland, Auckland, New Zealand
jsun038@ec.auckland.ac.nz

Abstract:
The Cambridge University computer laboratory researchers Mike Bond and Piotr Zielinski described that "In a single 30 minutes lunch-break, an attacker can thus discover approximately 7000 PINs (personal identification numbers) using adaptive decimalisation tables and guesses" [BZ03]. Can you believe it? I don't think this will happen in near future according to my understanding. In this paper I will discuss whether this decimalisation table attack is a real threat to the ATM networks or just a certain hypothetical threat. What are the constraints pressed on this attack that make it unlikely to happen? After analysis of the feasibility the attack, the question "what is the countermeasures for this potential attack?" coming up for discussion. South Africa's branch of Citibank got an order in the high court of London banning the paper "Decimalisation table attacks for PIN cracking" and other ATM (automatic teller machine, cash machine) security related documents on 3rd April 2003 in order to gag public disclosure of crypto vulnerabilities [HC03]. However, this will restrict future explorations into ATM network security.

1. Introduction

"Star survey in ATM news" shows that there is around 82 percent of accountholders in U.S. have an ATM/debit card [SS03]. ATM withdrawal is becoming an ideal way to debit small value (under $50) of cash 24 hours a day throughout the world. ATMs are also widely used by thousands and tens of thousands customers every day to withdraw cash from their account around New Zealand. However, ATMs also become criminals' perfect targets because they can easily steal the clean cash without monitor watching.

ASB Bank, Bank of New Zealand and The National Bank of New Zealand own about 750 machines which are approximately fifty percent of the whole ATM networks inside New Zealand. If accountholders want to make small cash withdrawals (such as NZD$10, $20, $50), they usually prefer to use ATM instead of queuing in front of bank tellers. In some countries (like China) people even can deposit small amount of cash through the ATM, but this is not the case in New Zealand. We only can do the quick deposit inside bank through "fast drop box" instead of ATM. The bank charges me a small amount of transaction fee if I withdraw my cash from other banks' ATM.

"It takes an average of 15 guesses to determine a four digit PIN using decimalisation table attacks algorithm discovered by Bond and Zielinski, instead of the 5000 guesses intended using brute force method [BZ03]. So what is decimalisation table? It is simply a many to one mapping between hexadecimal digits and decimal digits, as shown in figure 1.1

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5

Figure 1.1 decimalisation table

Section 2 of the paper will describe the basic background knowledge of ATM network, IBM PIN generation method, and hardware security module in [BZ03] paper. Bond and Zielinski's three attacks will be described in section 3. We will discuss the feasibility of the decimalisation table attacks in section 4. In Section 5, I represent appreciate and critical comments of this paper, and draw my conclusion in section 6.

2. Background

How can these decimalisation table attacks happen? I will describe the fundamental techniques behind the decimalisation table attacks as following in the first place.

2.1 The architecture of ATM networks

Encryption, translation and verification are three basic operations required for ATM networks. In figure 2.1, two entities or actors share a key to form a key zone; they need to trust each other in order to keep the common key in secret. However, are these two parties they really can trust each other? How they trust each other? Carl Ellison use the words "In God We Trust" in his "The Trust Shell Game" paper [CE98], the word "trust" is a big word in security. PIN need to be reformatted if different entities using different PIN formats (such as IBM 3624 and ISO-1 and so on) in translation operation. Decimalisation table attacks belong to part of PIN verification operation.

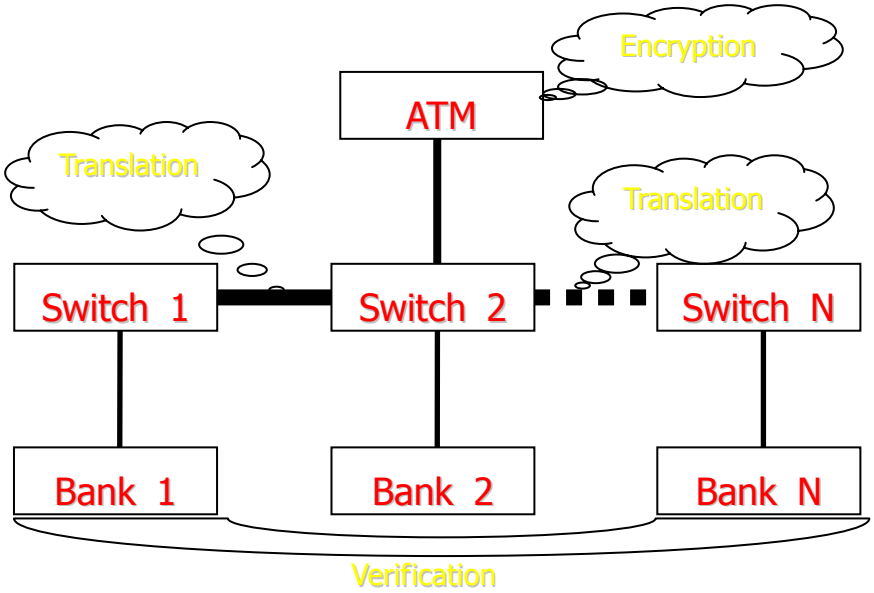

Figure 2.1 Basic operations of Network Architecture

2.2 IBM 3624-Offset PIN Generation Method

ATM is not only a cash machine; it also is an equipment to decode customers' bank cards' PIN. Now I will discuss how ATM works when the accountholders insert their bank card into ATM. Let us explain the example in figure 2.2, assuming my account number is *4556 2385 7753 2239* [RA94], a single cryptographic PIN generation key is securely locked in the ATM black box, the encrypted account number which is *3F7C 2001 00CA 8AB3* will be generated from the original customer' account number and PIN derivation key, this corresponding the step 1 in figure 2.3.

Then, the numeric format encrypted account *3572 2001 0020 8013* is generated from the hexadecimal format encrypted account using decimalisation table mapping, this will match the step 2 in figure 2.3. Now ATM knows the decimalisation PIN (also called intermediate PIN) from the original customer account number, this IPIN will compare with the result of subtracts offset from customer input PIN. If these two IPINs are exactly equal to each other, transaction access granted, otherwise denied by ATM, this is correspond to the third step in figure 2.3. If card holders want to change their bank card PINs, the public offsets which stored in the mainframe database along with card holders' account numbers will be changed correspondingly too in order to keep the fixed IPIN. Card holders only have three chances to input their correct PINs in most ATM rules, otherwise ATM will swallow their bank cards after the third tries.

Questions coming to my mind are: Is that card holders' account numbers and decimalisation PINs what the corrupt bank programmers want? How these frauds can make millions of dollars if they stole those account numbers and PINs? Where did the customers' input PIN store? In ATM, or mainframe databases or anywhere else? We will discuss these concerns in section 3 of this paper.

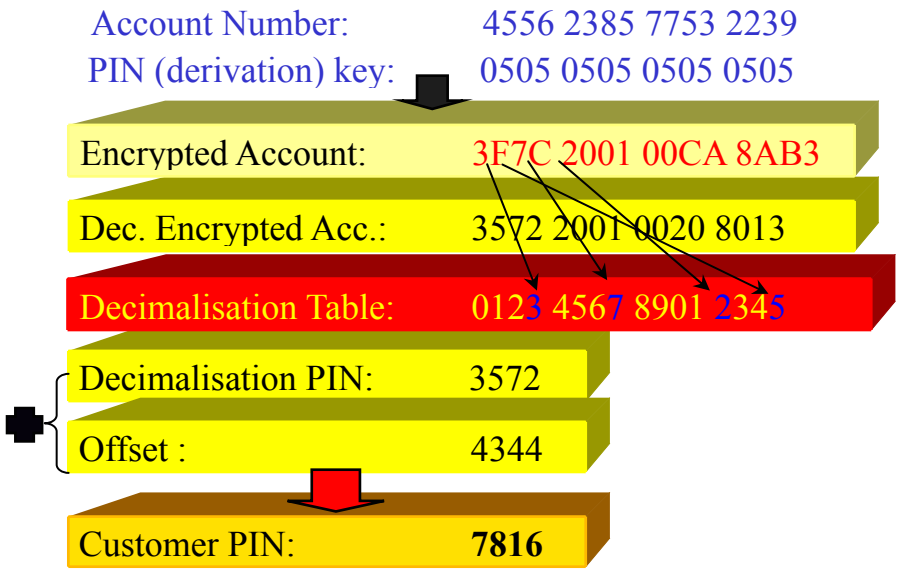

Figure 2.2 IBM 3624-Offset PIN Generation Method

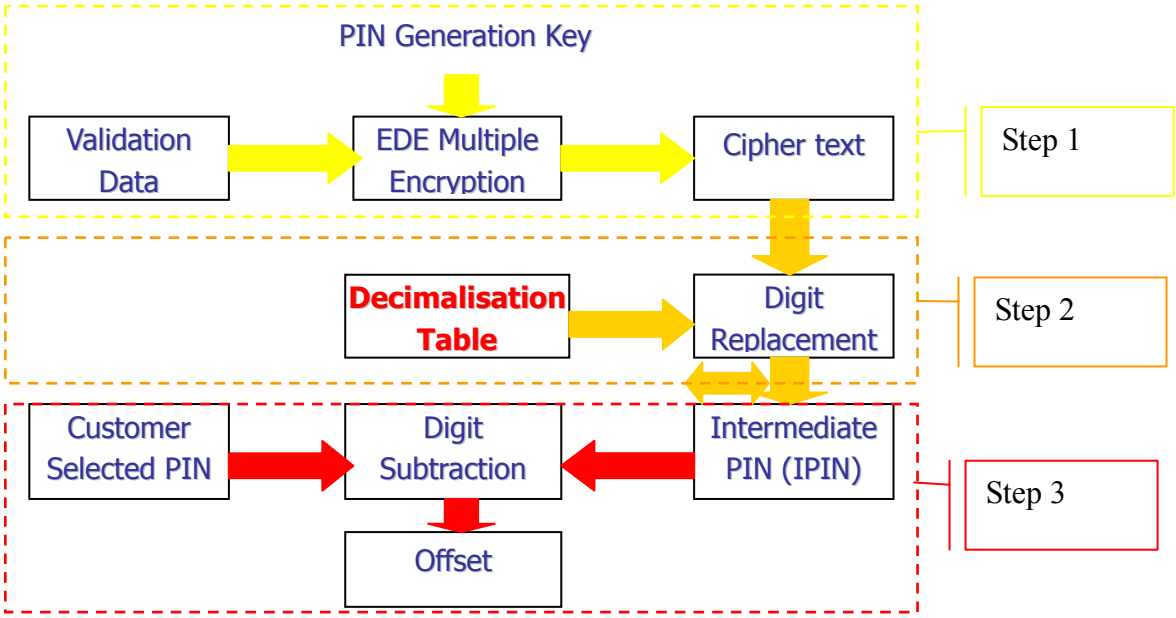

Figure 2.3 PIN verification (offset) copy from [JC02]

2.3 Hardware Security Module (HSM)

HSM is a tamper resistant or responding security module which provides a secure and trusted environment to do sensitive operations, such as gives a YES/NO answer to a programmer's guess [BZ03][JC02].

3. Three schemes

3.1 Initial scheme

The first initial scheme takes average 24 guesses to determine four digits PIN, the disadvantage of this method is almost twice as many guesses in the worse case, 46 guesses are required if the original PIN contains three different digits [BZ03]. There are two phases in this scheme; the first phase takes at most 10 guesses to determine all digits that form the original PIN, for example, only need 5 guesses if the original PIN is 5555. The second phase shows the all possible combination of those digits which were found by the first phase. Table 3.1 explains how to calculate the average 24 guesses in detail, so what should we learn from this table? Let us set our PINs only contain three different digits, it will take longer for frauds to guess, remember, this is the worst case for all three schemes.

Digits	1st phase	Possibilities (only in the 2nd phase)	Total Possibilities
A	10	AAAA(1)	10/2 + 1 = 6
AB	10	ABBB(4),AABB(6),AAAB(4)	10/2 +14 = 19
ABC	10	**AABC(12),ABBC(12),ABCC(12)**	**10/2 + 36 = 41**
ABCD	10	ABCD(24)	10/2 +24 = 29
Average guesses: (6+19+41+29)÷4= 23.75 ≈ 24 guesses			

Table 3.1 Initial scheme [BZ03]

3.2 Adaptive scheme

Two obvious errors appear in Bond and Zielinski paper, the first one is ambiguous number of guesses in an adaptive scheme. In the introduction section of the paper [BZ03], the authors said "an adaptive scheme takes an average of *15 guesses*," then in fourth section talks about the decimalisation table attacks, "adaptive approach reduce the number of necessary *guesses to 22*," finally in fifth section of result, "in the adaptive scheme algorithm, the average has fallen from 24 to *15 guesses* again". So what exactly how many guesses will need for the adaptive scheme still confused me, 15 guesses or 22 guesses?

The second error is about the search tree for the initial scheme. In Bond and Zielinski paper said "at each node, we check whether D_v (p_{orig}) =p_v. Then, we move to *the right child if yes and to the left child otherwise*. [BZ03]". We can tell from figure 3.2, the directions obvious wrong, it should say, we move to the left child if yes and to the right child otherwise. The search tree for initial scheme is unbalance, but for adaptive scheme the tree should be quite balance.

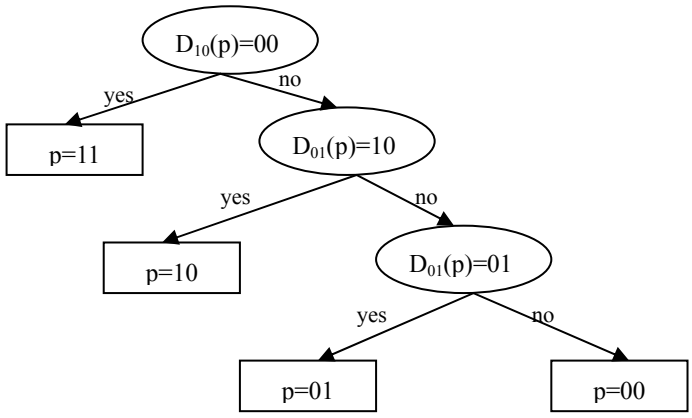

Figure 3.2 the search tree for the initial scheme, copy from fig.6. [BZ03]

3.3 PIN Offset Adaptive scheme

The improved PIN offset adaptive scheme takes approximate 16.5 guesses to determine four digits PIN, as you can see from the following table 3.3. "This scheme can use the offset parameter to compensate for card holder's PIN change, even when the corrupt bank programmer can not steal any encrypted trial PINs and cannot encrypt their own guesses."[BZ03]

Digits	1st phase	2nd phase	Total Possibilities
A	10	All digits are the same (0)	10 + 0 = 10
AB	10	Two different digits (13)	10 +13 = 23
ABC	10	**Three different digits (9)**	**10 + 9 = 19**
ABCD	10	All digits different(6)	10 +6 = 16
Average guesses: (10+23+19+16)÷4= 17 ≈ 16.5 guesses			

Table 3.3 PIN offset adaptive scheme

4. Is the decimalization table attack feasible?

Is this decimalisation table attack discovered by Bond and Zielinski a real threat in the real world systems? Did they really find out the weakest point of the decimalisation table? Will this attack happen in near future or is this attack actually a hypothetic one? How can those corrupt bank programmers steal the PIN generation key from black box in ATM and how they knew the public offset for a given PIN block? Even after they figured out cardholders' account numbers and their corresponding intermediate PINs, how can they turn these into clean cash? Do they need to make duplicate white cards of these cardholders' cards or just translate the whole amount to cash to one account first, and then withdraw the cash from that account? We will discuss all these questions in the following subsections.

4.1 What information is needed to mount the attack?

In order to crash account holders' ATM cards PINs using the decimalisation table attacks, what input parameters (see in table 4.1) and device requirements are the necessary of the attacks by the dishonest bank programmers?

4.1.1 Input Parameters

Parameters need to know by the attackers	Descriptions of the Parameters
Validation Data	Validation data means Card holders' account number; this data can be easily seized by the bank programmers.
Decimalization Table	The bank programmers can modify a single element in this decimalisation table to try to guess IPIN.
Encrypted PIN Block	PIN generation key is located in encrypted PIN block [MB01], if attackers stole the PIN generation key, they can encrypt an original account number to an encrypted PIN by PIN generation key. But how can they get this PIN generation key from black box in ATM?
Offset	The relationship between offset and IPIN is, if attacker adds m to a digit in offset, the corresponding digit in IPIN should subtract m. But the problem is how can attackers steal the right offset for a particular account?
Encrypted PIN Encrypting Key **Encrypted PIN Verification Key**	Essence of the decimalisation table attack. See more details in [RA00][AB01]
Hence, these parameters are essential to the decimalisation table attack by dishonest bank programmers.	

Table 4.1 Input parameters

4.1.2 Device Requirements

Basically the corrupt bank programmers need to have privilege to physically access to the device centre, which includes tamper resistant/responding security module, crypto accelerator, network security module, host security module and hardware security module [JC03]. South Africa researcher Jolyon Clulow published his paper "I know your PIN" [JC02] about this device at the industry specific conference RSA Europe 2002.

4.2 Who can access and modify those secure data?

Who are those attackers? In Bond and Zielinski paper [BZ03], the attackers will be those corrupt bank programmers. Are those dishonest bank programmers belong to insider attack? I think so. "An insider is either a person internal to a given financial institution (e.g. employee, contractor, cleaning staff, etc) or else an individual who has gained access to the financial network, perhaps through some traditional network hacking technique [JC03][RP03]." But what rights those bank programmer have? I think they may have privileges to access the secure data, but no modification rights.

4.3 What is attack strategy?

The strategy of the decimalisation table attack is that the attackers can modify only a single digit to an entry in the decimalisation table and learn what changes may happen on the offset repeatedly.

4.3.1 How to attack?

Here is the example show how attack happened? This example follows the example shown in figure 2.2 which the original customer PIN 7816 is equal to the sum of IPIN 3572 and offset 4344.

Step by step attack	Explanation
Dec. Table (0) = 1123456789012345 Dec. PIN = 3572 Offset = 4344 (will pass)	In the decimalisation table, changes the 0^{th} bit from 0 to 1, nothing will happen because the IPIN does not contain a digit 0.
Dec. Table (1) = 0223456789012345 Dec. PIN = 3572 Offset = 4344 (will pass)	In the decimalisation table, changes the 1^{st} bit from 1 to 2, nothing will happen because the IPIN does not contain a digit 1.
Dec. Table (2) = 0133456789012345 Dec. PIN = 3573 Offset = 4344 (will fail) = 4343 (will pass)	In the decimalisation table, changes the 2^{nd} bit from 2 to 3, if the offset still be 4344, formula $3573 \oplus 4344 \neq 7816$ will fail, attack happened, and the offset changed back to 4343. So the IPIN did contain a digit 2.

Table 4.2 an example of the attack

In table 4.2, we have identified that the fourth digit in the original Decimalisation PIN is a 2 and so the 4th final PIN digit is 2 + 4 = 6, which satisfied the formula (Decimalisation PIN 3572+ Offset 4343 = final PIN 7816).

4.3.2 Is this attack a practical one?

Although it takes only 0.25 second to crack a four digits PIN ATM card using Bond and Zielinski technique (see how to calculate ¼ second in table 4.3), and they describe three potential attacks and fundamental techniques behind decimalisation table in great detail on paper [BZ03], but these authors did not show to audiences exactly how to attack in a real world systems.

For example, they did not indicate how to step by step crack an actually ATM network, they did not share anything like a memo of the attack, pseudo code of algorithm, how to physically access to the device, how to enter the network transporting transaction traffic and how to insert messages to change the decimalisation table, and they never guarantee this attack will be practically happened in near future if HSM manufacturers don't modify their software immediately to fix the flaw. I don't think this attack is a practical one because lack of evidence about the attack feasibility and too many constraints which include the input parameters and device requirements mentioned in previous subsection 4.1, to enforce this attack to happen practically in the real world.

Input parameters:
HSM: 60 trail PINs per second
Attack Guesses: 15 guesses
Number of PINs need to crack: 1 PIN
Formula:
1 PINs * 15 guesses ÷ (HSM) 60 PINs/sec = 0.25 sec
(HSM) 60 PINs/sec * 60 sec/min * 30 min ÷ 15 guesses =7200 PINs
Output parameter:
How long does it take to crack a PIN: 0.25 second
In thirty minutes lunch time can crack: 7200 PINs

Table 4.3 it takes about 0.25 second to crack a PIN using Bond and Zielinski decimalisation table attack algorithm.

4.3.3 Algorithm of decimalisation attack

Clulow shared the decimalisation table attack algorithm (see algorithm 6 Decimalization attack in [JC03]) with public offset parameter input scheme in his dissertation [JC03], but do you think the frauds can implement this algorithm into program and use it to attack the real system? I don't think so, because this is not enough according to lack of understanding fundamental techniques behind the decimalisation table attacks, and precondition of this algorithm is that fraud have to be a bank programmer.

4.4 Can insiders turn this attack to clean cash?

One thing I am concern about is how can insider retrieve the money even he/she got customers' account number and PIN? Don't show anything related in paper [BZ03]. I think this is the aim of the decimalisation table attack, why those attackers choose to attack any ATM networks, because obviously they need some easy and fast money. These attackers are not like some internet hackers who spend hours of works to hack software and share it online to let anybody download the free cracked software. The aim of those hackers is they try to make anybody know information is free.

If we are trying to fully understand Bond and Zielinski's paper, we need to know how actually ATM works. When we open a new ATM card in a bank, we need to set our PINs using input equipment in front of tellers, then the PINs will saved in the mainframe database. When we withdraw cash from ATM, we input our customers' PIN, ATM will check whether my customers' PIN is valid by subtract offset from customer PIN and compare with the result of calculated IPIN from customers' account number by ATM using IBM 3624-offset method. Hence, customers' account number and IPIN should be the secure data which need to pass to the mainframe database to validate customers' account details. So IPIN will store in the mainframe database too.

What I am trying to argue is, although the attackers stole the account number and corresponding IPIN, how can they put the cash into their own pocket. For example, they might transfer all cash into one new account, and then withdraw clean cash from that account, but before they withdraw all money, the bank should notice where these huge amounts of unauthenticated money came from, might stop the transactions. Another way is the attackers might need to steal other kind of passwords, such as some banks need account password to withdraw cash in front of teller, or eight digits online banking password to log on internet banking, or four digits password for telephone banking. For instance, if the attackers know how to map the IPIN with that accountholder's withdrawal password in the mainframe database, they can steal all that accountholder's money instead of only NZD$800 limit for ANZ bank per day using ATM.

4.5 What is wrong with HSM manufacture?

Firstly, it is obviously some functions (such as manipulation of the decimalisation table) are bad and insecure. Moreover, Individual conflicting functions are added to security processor's application programming interfaces (APIs), this make the whole system not secure. Furthermore, lack of an absolute standard which all customers should definitely inherit from [JC03] [AB01], this is the main reason to cause an insecure system we used now. But modify or update the software related to HSM will be very costly.

4. 6 Countermeasure

In my understanding, I don't think the decimalisation table attacks will happen with those restricted constraints I mentioned in the section 4.1. However, let us assume these attacks might occur, how we can predict and solve them in order to get full security. The easiest way is to use those PIN verification methods don't exploit 0123456789012345 as decimalisation tables, but this is usually not the case. The decimalisation table input is still very common used in most PIN verification methods, the best way to secure this input is to protect it cryptographically, only authorised tables can be used by restricted programmers under electronic access control.

4.6.1 Short Term & Long Term Solutions

The short term solution of the decimalisation table attacks is that it isn't possible to change the decimalisation table without permission and use more advanced intrusion detection measures which supplied by different manufacturers[JM01], the longer term solution is to "support for decimalisation is not a robust approach to PIN verification"[BZ03].

4.6.2 Clulow's counter-measures to the attack

First and the best solution to the attacks is to remove weakest algorithms and functions and leave only the strongest, a single standard algorithm for everyone is required. Moreover, the use of message authentication codes to encrypt the input parameters to the functions will be a logical solution to avoid modifying the parameters of decimalisation tables in order to accomplish an attack. Furthermore, a key separation mechanism used by PIN block variance is a useful restriction for an API. Finally, if a function has been shown insecure, electronic access control can disable or remove it. [JC02]

5. Result

The critical comment of this term paper is I did make some my own assumptions based on my understanding about the whole ATM network translation working process, such as ATM may pass customer account and IPIN to the mainframe database and some kind of database keys mapping among IPIN, PIN, credit card PIN, account password, internet banking password and phone banking password.

On the other hand, the appreciate comments are I did try to fully understand Bond and Zielinski paper in detail from the beginning, before I read this paper, I have no idea about ATM network, PIN verification method, IBM PIN calculation methods and so on. I think Clulow's master dissertation is a good help for me to understand the whole process of decimalisation table attacks and other five potential attacks which include ANSI X9.8 attack, extended ANSI X9.8 attack, key separation No.1 attack, key separation No.2 attack and check value attack. [RP03][RP02]

6. Conclusion

Although Bond and Zielinski published “decimalisation table attacks for PIN cracking” and mentioned cracking an ATM PIN only needs 15 guesses using their attack techniques with restricted constraints under some limited environment, I still don’t think this attack is likely to happen. Because those input parameters and device requirements we discussed in section 4.1 are very unlikely to seize or steal from the bank by the corrupt bank programmers.

The best way to improve ATM network security for the decimalisation table attacks is to abandon the decimalisation tables, on the contrary exploit randomly generated PINs stored encrypted in an online database or use the triple data encryption standard based encryption standards for host to host PIN communication in ATM network [DS02].

Further research will to deal with how to use an alternative way to replace the decimalisation tables in PIN verification method, such as random PINs, TDES, or other techniques used in some banks security. [GK02] Another interesting research will be focus on whether these decimalisation table attacks will happen on VISA system. [RA01] Finally, we hope Citibank and diners’ club issue will not restrict future explorations into ATM network security.

References:

[RA94] R. Anderson, "*Why Cryptosystems Fail Communications of the ACM"*, 37(11), pp32-40 (Nov 1994)

[RA00] R. Anderson, "*The Correctness of Crypto Transaction Sets"*, Proc. Cambridge Security Protocols Workshop 2000 LNCS 2133, Springer-Verlag, pp 125-127 (2000)

[RA01] R. Anderson, "Security Issues of PIN Based Payment Systems", Computer laboratory, Pembroke Street, Cambridge CB2 3QG.

[AB01] R. Anderson and M. Bond, "*API-Level Attacks against Embedded Systems*", IEEE Computer Magazine, October 2001, pp. 67-75.

[MB01] M. Bond, *"Attacks on Cryptoprocessor Transaction Sets"*, Proc. Workshop Cryptographic Hardware and Embedded Systems (CHES 2001), LNCS 2162, Springer-Verlag, pp 220-234 (2001)

[BZ03] Bond and P Zielinski, *"Decimalisation Table Attacks for PIN Cracking"*, University of Cambridge Computer Laboratory Technical Report 560, February 2003, 14 pp. Available: http://www.cl.cam.ac.uk/TechReports/UCAM-CL-TR-560.pdf, March 2003.

[JC02] J. Clulow, *"I Know Your PIN"*, RSA Europe, October 2002

[JC03] J. Clulow, *"The Design and Analysis of Cryptographic Application Programming Interfaces for Security Devices"*, M.Sc. Dissertation, University of Natal, Durban, South Africa, January 2003.

[CE98] C. Ellison *"The Trust Shell Game (position paper)"* in Christianson, B. Crispo, W.S. Harbison, M. Roe (Eds.): *Proceedings of 6th International Workshop on Security Protocols*, Cambridge, UK, April 1998. LNCS 1550, p. 36 ff.

[GK02] T. Glaessner and T. Kellermann, and V. McNevin, *"Electronic Security: Risk Mitigation in Financial Transactions – Public Policy Issues"*, World Bank Publication June 2002.

[JM01] J McHugh, "Intrusion and Intrusion Detection," *International Journal of Information Security 1,* 2001, pp. 14-35.

[HC03]*"Court Banned Citibank PIN Cracking Documents"*, 3 April 2003, http://cryptome.sabotage.org/citi-ban.htm

[RP02] *"New Attacks on EFT Networks"*, Redpay Security Bulletin 100-0014-BL, 6 January 2003, 6 pp, available http://www.redpay.com/bulletins/RedpaySecurityBulletinNewAttacksOnEFTNetworks.pdf, March 2003.

[RP03] *"PIN Attacks on EFT Networks"*, Redpay Security Bulletin 100-0019-BL, 28 February 2003, 8 pp, available http://www.redpay.com/bulletins/RedpaySecurityBulletinPINAttacksOnEFTNetworks.pdf, March 2003.

[SS03] "*Star survey shows consumers like idea of debit for small purchases",* ATM news, 21 May 2003, available http://www.atmmarketplace.com/news_story.htm?i=15706 .

[DS02] *"Triple DES (TDES)"* Technical Data Sheet, Part Number: T-CS-EN-0002-100, Document Number: I-IPA01-0081-USR Rev 06, Cadence Design Foundry (UK) Ltd, October 2002.