

## Automatic Teller Machine

A cura di:

**Mario Carpentieri**  
**Sergio Avallone**  
**Alfonso De Martino**  
**Marco Palladino**  
**Mauro Paglietta**

Professore:

**De Santis Alfredo**

Sicurezza su Reti

A.A. 2003/2004

1

## Argomenti Trattati

- Ⓢ **Nascita degli ATM**
- Ⓢ **Caratteristiche tecniche degli ATM**
- Ⓢ **Attacchi agli ATM**
- Ⓢ **Decimalization Attack**

Sicurezza su Reti

A.A. 2003/2004

2

## Argomenti Trattati

- Ⓢ **Nascita degli ATM**
- Ⓢ **Caratteristiche tecniche degli ATM**
- Ⓢ **Attacchi agli ATM**
- Ⓢ **Decimalization Attack**

Sicurezza su Reti

A.A. 2003/2004

3

## Nascita degli ATM

Ⓢ **Il primo ATM in Italia più conosciuto con il nome di Bancomat nacque nel 1967 installato nella *Barclays Bank*.**

Ⓢ **L'inventore degli ATM fu Don Wetzel**



*Don Wetzel*

Sicurezza su Reti

A.A. 2003/2004

4

## Evoluzione degli ATM(1)

- Ⓢ **Inizialmente l'ATM fu progettato solo per prelievi o per depositi**
- Ⓢ **Oggi è possibile effettuare :**
  - Ricariche telefoniche**
  - Prenotazioni per Biglietti Aerei**

Sicurezza su Reti

A.A. 2003/2004

5

## Evoluzione degli ATM(2)

- Ⓢ **In futuro sarà possibile:**
  - Ⓢ **Prenotare film al cinema**
  - Ⓢ **Visualizzare l'albergo più vicino**
  - Ⓢ **Analizzare il percorso più conveniente per una destinazione**
  - Ⓢ **Ecc...**

Sicurezza su Reti

A.A. 2003/2004

6

## Argomenti Trattati



- @ Nascita degli ATM
- @ Caratteristiche tecniche degli ATM
- @ Attacchi agli ATM
- @ Decimalization Attack

## Problematiche Tecniche



- @ Riconoscimento banconote
- @ Validazione del PIN

## Gestione della Validazione



@ Quando viene inserita una banconota nel ATM questo assegna un punteggio in base alle sue caratteristiche:

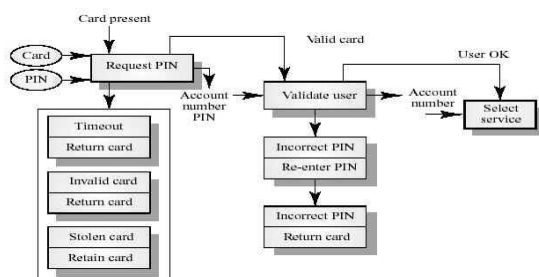
- @ Sfasamenti dei processi di stampa
- @ Variazioni delle caratteristiche cromatiche per l'usura

## Gestione della Validazione



- @ Eccezioni validazione PIN :
  - @ Time out: tempo limitato d'immissione PIN, la carta è restituita
  - @ Carta invalida: la carta non è riconosciuta ed è restituita
  - @ Carta rubata: se la carta riconosciuta è rubata, questa è ritirata

## Gestione della validazione



Schema di validazione di un ATM

## Funzionamento ATM



@ Il funzionamento degli ATM è basato su due concetti fondamentali

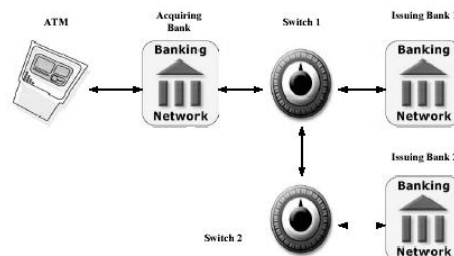
- @ **PIN** (Personal Identification Number) è usato dal possessore di conto per identificarsi alla banca
- @ **PAN** (Personal Account Number) usato dalla banca per identificare il conto usato per una transazione bancaria

## Comunicazione tra banche e ATM

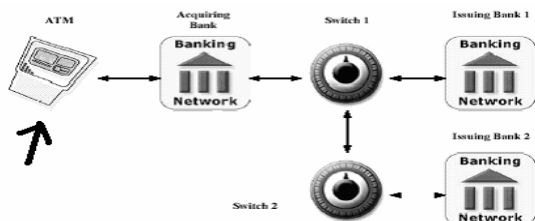


- ☉ L'utente effettua una richiesta ad un ATM di una banca
- ☉ La banca acquirente gestisce tramite uno switch la richiesta del cliente indirizzandola alla banca detentrica del conto
- ☉ La banca detentrica del conto effettuerà le ultime verifiche e, se la carta è valida, effettuerà l'operazione richiesta dal cliente

## Comunicazione tra Banche e ATM



## Problematiche



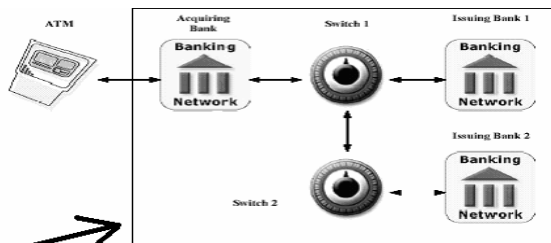
**Il primo problema è nello sportello ATM**

## Soluzione



- ☉ Per risolvere questo problema il cliente deve:
  - ☉ Verificare che al momento della digitazione del PIN non ci sia nessuno a spiarlo
  - ☉ Verificare l'integrità dello sportello ATM
  - ☉ Ricordarsi a memoria il PIN o comunque assicurarsi che non sia vicino alla carta del ATM
  - ☉ Lo sportello ATM deve garantire un timeout
  - ☉ Se viene digitato il PIN errato più volte lo sportello deve ritirare la carta

## Problematiche



**Il secondo problema è proprio nella gestione della rete**

## Soluzione



- ☉ In questo caso per ogni banca e per ogni switch devono:
  - ☉ Essere aggiornati sistematicamente dispositivi di sicurezza
  - ☉ Aumentare le verifiche sull'integrità della rete

## Argomenti Trattati



- Ⓢ **Nascita degli ATM**
- Ⓢ **Caratteristiche tecniche degli ATM**
- Ⓢ **Attacchi agli ATM**
- Ⓢ **Decimalization Attack**

## Tipi di attacchi

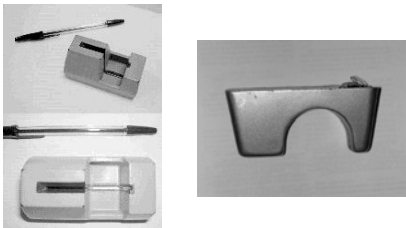


- Ⓢ **Presenteremo ora alcuni degli attacchi più conosciuti riportati agli ATM:**
  - Ⓢ **Lingua di cocodrillo**
  - Ⓢ **Insider attack**
  - Ⓢ **Worm Welchia**

## Lingua di cocodrillo



- Ⓢ **E' richiesta :**
  - Ⓢ **L'applicazione di apparecchi per la cattura di dati, SKIMMER**



## Lingua di cocodrillo



## Insider Attack



- Ⓢ **Presuppone l'esistenza di un membro corrotto all'interno della banca**
- Ⓢ **L'assalitore comincia:**
  - Ⓢ **Esaminando le sequenze di operazioni per un certo periodo**
  - Ⓢ **Registrando i PIN Block che passano attraverso il sistema**

## Insider Attack



- Ⓢ **I PIN in chiaro sono estratti attraverso una sequenza di consultazioni**
- Ⓢ **Ottenute le informazioni necessarie l'assalitore lascia la scena dell'attacco**

## Worm Welchia



- ☉ I bancomat di due banche americane sono stati infettati, nell'agosto del 2003
  - ☉ Da un worm che si propaga attraverso una vulnerabilità nella sicurezza di Windows
- ☉ La scorsa estate un numero non precisato di bancomat su cui girava Windows XP Embedded sono stati spenti
  - ☉ Infettati da Welchia (anche noto come Nachi)

## Worm Welchia



- ☉ In seguito l'azienda ha stipulato un accordo con Sygate Technologies per "fornire una protezione superiore"



## Argomenti Trattati



- ☉ Nascita degli ATM
- ☉ Caratteristiche tecniche degli ATM
- ☉ Attacchi agli ATM
- ☉ Decimalization Attack

## Decimalization Attack



- ☉ Ora ci accingeremo a esporre il metodo più diffuso per attaccare un ATM in base alla tavola di decimalizzazione
- ☉ Faremo vedere come si genera un PIN con l'ausilio della tavola di decimalizzazione
- ☉ Analizzeremo infine gli attacchi che possono essere effettuati su questo schema

## Decimalization Attack



- ☉ Il perno su cui si basa la generazione dei PIN è la tavola di decimalizzazione

0123456789ABCDEF  
0123456789012345

- ☉ Permette di sostituire valori esadecimali con valori decimali, per ottenere PIN digitabili nelle tastiere numeriche standard

## Decimalization Attack



- ☉ Il metodo IBM 3624-Offset fu sviluppato per sostenere la prima generazione degli ATM e fu largamente adoperato
- ☉ Il metodo fu progettato in modo che gli ATM offline sarebbero stati capaci di verificare i PIN dei clienti:
  - ☉ Senza avere bisogno di elaborare e memorizzare database

## Decimalization Attack



### @ Offset :

@ valore che sommato al PIN originale permette al cliente di modificare il proprio conto

@ associato univocamente ad ogni numero di conto

## Decimalization Attack



@ Fu sviluppato uno schema dove i PIN dei clienti potevano essere calcolati a partire dal numero di conto del cliente stesso da una cifratura con una chiave segreta



## Decimalization Attack



@ Il numero di conto fu reso disponibile al cliente su scheda magnetica, quindi l'ATM ebbe bisogno solo di immagazzinare in modo sicuro una sola chiave di crittografia

## Decimalization Attack



### @ Generazione del PIN:

N° conto 4556238577532239

CifraturaDES 3F7C220100CA8AB3

Scarto le cifre non usate e prendo: 3F7C

@ Effettuo il mapping con la tavola di decimalizzazione e lo sommo all'Offset 4244 ottenendo 7816 che sarà il PIN finale

```
0123456789ABCDEF
0123456789012345
```

## Decimalization Attack



### @ Gli ATM usano Hardware Security Module (HSM)

@ processore su cui gira un software che offre servizi relativi alla crittografia e alla sicurezza

### @ L'API dell'HSM contiene operazioni

@ per generare e verificare i PIN

@ decifra le chiavi di zona quando vengono scambiate tra banche

@ supporta un'ampia gamma di funzioni per la gestione della chiave

## Encrypted PIN Verify



```
Encrypted_PIN_Verify(
  A_RETRES , A_ED , // return codes 0,0=yes 4,19=no
  trial_pin_kek_in , pinver_key , // encryption keys for enc inputs
  (UCHAR*)"3624 " "NONE " // PIN block format
  " F" // PIN block pad digit
  (UCHAR*)" " ,
  trial_pin , // encrypted_PIN_block
  I_LONG(2) ,
  (UCHAR*)"IBM-PINO" "PADDIGIT" , // PIN verification method
  I_LONG(4) , // # of PIN digits = 4
  "0123456789012345" // decimalisation table
  "123456789012 " // PAN_data (account number)
  "0000 " // offset data
);
```

### Esempio di funzione di verifica di un PIN

## Decimalization Attack



Ⓢ Gli input cruciali della funzione sono:

- Ⓢ tavola di decimalizzazione
- Ⓢ PAN DATA (che contiene il numero personale dell'account)
- Ⓢ Encrypted PIN Block (EPB)
- Ⓢ Offset

Ⓢ I primi due e il quarto sono forniti in chiaro e sono il punto debole di questo sistema di sicurezza

## Decimalization Attack



Ⓢ L'EPB per effettuare l'attacco si può ottenere :

- Ⓢ inserendo il PIN in un ATM reale e intercettando l'EPB corrispondente
- Ⓢ oppure tramite la funzione *Clear\_PIN\_Encrypt*, che creerà un EPB a partire dal PIN scelto

Ⓢ La funzione ritorna true o false, a seconda che il PIN sia corretto o no

## Decimalization Attack



Ⓢ Gli attacchi riportati alla tavola di decimalizzazione hanno diversi schemi:

- Ⓢ Schema statico
- Ⓢ Schema adattivo
- Ⓢ Schema adattivo con offset del PIN

## Decimalization Attack



Ⓢ Questi attacchi possono essere effettuati solo da un dipendente della banca che ha accesso alla funzione Encrypted PIN Verify

Ⓢ Per effettuare l'attacco con i primi due schemi l'assalitore ha a disposizione:

- Ⓢ tavola di decimalizzazione modificata
- Ⓢ EPB (PIN di prova cifrato)
- Ⓢ PAN DATA

## Decimalization Attack



Ⓢ Nell'ultimo schema l'assalitore non è riuscito ad ottenere nessun PIN di prova cifrato

Ⓢ Quindi i parametri che passerà alla funzione saranno:

- Ⓢ EPB intercettato contenente il PIN corretto
- Ⓢ Offset, usato per modificare il PIN
- Ⓢ tavola di decimalizzazione modificata
- Ⓢ PAN DATA

## Schema Statico



Ⓢ Lo schema iniziale consiste di 2 passi:

- Ⓢ determinare le cifre che compongono il PIN
- Ⓢ determinare le posizioni delle cifre ottenute



## Schema Statico



☞ Poniamo  $D_{orig}$  come tavola di decimalizzazione originale

☞ Per una data cifra  $i$ , consideriamo una tavola di decimalizzazione binaria  $D_i$ , tale che:

$$D_i[x] = 1 \text{ se } D_{orig}[x] = i$$

$$D_i[x] = 0 \text{ altrimenti}$$

## Schema Statico



☞ Ad esempio passiamo alla funzione Encrypted PIN Verify:

☞ l'EPB corrispondete al PIN di prova 0000  
☞ la tavola di decimalizzazione modificata  $D_3$ :

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0

☞ PAN DATA

## Schema Statico



☞ Operazioni di verifica della funzione:

- ☞ cifra il PAN DATA  $\Rightarrow$  38B6 2210 2290 4509
- ☞ prende le prime 4 cifre  $\Rightarrow$  38B6
- ☞ le decimalizza  $\Rightarrow$  1000
- ☞ decifra l'EPB  $\Rightarrow$  0000
- ☞ effettua il matching tra 1000 e 0000
- ☞ ritorna false

☞ Quindi si può dire che il test fallisce quando il PIN originale contiene la cifra  $i$

## Schema Statico



☞ Il secondo passo consiste nel trovare tutti i possibili PIN

- ☞ Cifre Possibilità
- ☞ A AAAA(1)
- ☞ AB ABBB(4),AABB(6),AAAB(4)
- ☞ ABC AABC(12),ABBC(12),ABCC(12)
- ☞ ABCD ABCD(24)

☞ Caso peggiore tre cifre differenti

☞ 36 combinazioni :

AABC	ABCA	ABAC	ACBA	CBAA	BCAA	BACA	CABA	CAAB	BAAC	ACAB	AACB
BBAC	BACB	BABC	BCAB	CABB	ACBB	ABCB	CBAB	CBBA	ABBC	BCBA	BBCA
CCAB	CABC	CACB	CBAC	BACC	ABCC	ACBC	BCAC	BCCA	ACCB	CBCA	CCBA

## Schema Adattivo



☞ Il processo di rottura del PIN con lo schema adattivo può essere rappresentato da un albero di ricerca binario

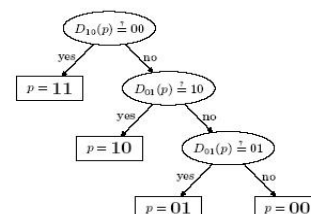
☞ Ogni nodo  $v$  contiene un confronto  $D_v(p_{orig}) = p_v$ , se questo confronto fallisce ci sposteremo sul figlio destro, altrimenti su quello sinistro

## Schema Adattivo

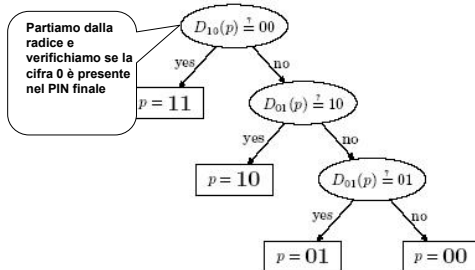


☞ Per semplicità assumiamo che il PIN originale consista di due cifre binarie, e la tavola di decimalizzazione mappi 1 con 0 ed 0 con 1

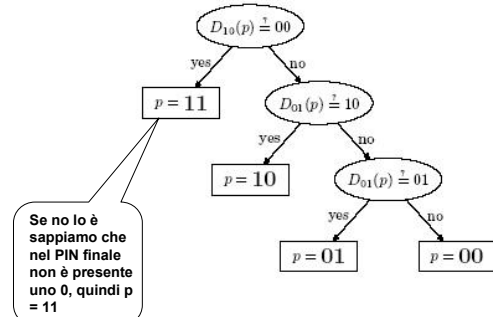
☞ Analizziamo il seguente caso:



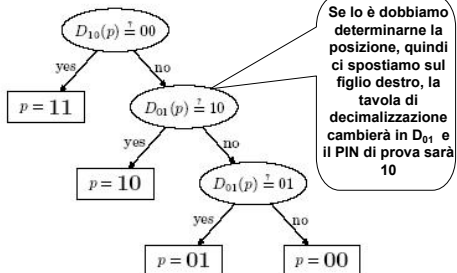
## Schema Adattivo



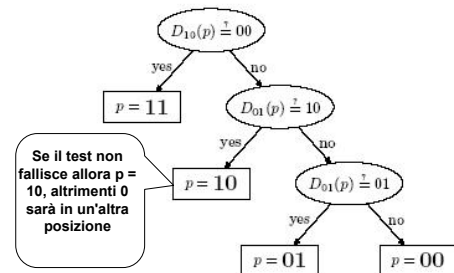
## Schema Adattivo



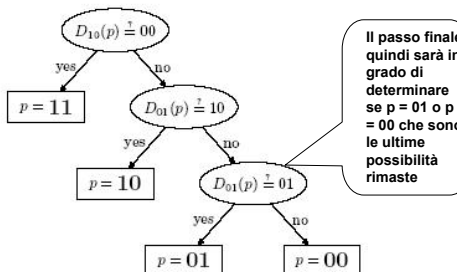
## Schema Adattivo



## Schema Adattivo



## Schema Adattivo



## Schema Adattivo



Il processo di rottura del PIN con lo schema adattivo può essere migliorato associando ad ogni nodo dell'albero una lista di PIN

In particolare:

la lista del nodo radice contiene tutti i PIN  $16^4$

mentre per ogni nodo interno: se è figlio sinistro, la lista associata contiene i PIN del padre che soddisfano il confronto, altrimenti, se è figlio destro, la lista associata contiene i PIN del padre che non soddisfano il confronto

la lista di ogni foglia conterrà un solo elemento, cioè un PIN originale  $p_{orig}$

## Schema Adattivo



Per avere un albero bilanciato scegliamo i valori di  $D_v$  e di  $p_v$  per ogni nodo  $v$  nel seguente modo:

- assumiamo che  $P_v$  sia la lista associata ad ogni nodo  $v$ ;
- scegliamo quella per cui la probabilità di

$$D_v(p) = p_v, \text{ (ove } p \in P_v), \text{ sia circa } \frac{1}{2}$$

- $D_v$  soddisfi la seguente proprietà, per ogni coppia di cifre esadecimali  $x$  ed  $y$ :

$$D_{orig}[x] = D_{orig}[y] \Rightarrow D_v[x] = D_v[y]$$

- con quest'ultima osservazione il numero di PIN si riduce da  $16^4$  a  $10^4$

## Schema Adattivo



Esecuzione dell'algorithmo adattivo per il PIN originale  $p_{orig} = 3491$

Scegliamo ad ogni passo:

- una opportuna tavola di decimalizzazione
- un opportuno PIN di prova cifrato

In questo modo cerchiamo di ottenere un albero di ricerca il più bilanciato possibile

## Schema Adattivo



Il numero iniziale di PIN è  $10^4 = 10000$ ; nel nostro esempio, scegliamo come tavola di decimalizzazione iniziale:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0

PIN di prova:  $p = 0000$

## Schema Adattivo



Confrontiamo il PIN di prova cifrato 0000, con il risultato ottenuto mappando il PIN originale 3491 con la tavola di decimalizzazione scelta, ovvero 0000

Se il confronto è positivo, sappiamo che il PIN originale non contiene 0 e 6

Quindi dalla radice ci spostiamo sul figlio destro, passando da 10000 possibili PIN a 4096

1ª Posizione	2ª Posizione	3ª Posizione	4ª Posizione
1-2-3-4-5-7-8-9	1-2-3-4-5-7-8-9	1-2-3-4-5-7-8-9	1-2-3-4-5-7-8-9

## Schema Adattivo



Per il secondo confronto utilizzeremo la tavola di decimalizzazione:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0

PIN di prova:  $p = 0000$

## Schema Adattivo



Confrontiamo il PIN di prova cifrato 0000, con il risultato ottenuto mappando il PIN originale 3491 con la tavola di decimalizzazione scelta, ovvero 0001

Il risultato negativo del confronto, ci dice che nel PIN originale ci sarà sicuramente la cifra 1

Quindi dal nodo corrente ci spostiamo sul figlio destro, passando da 4096 possibili PIN a 1695

## Schema Adattivo



Per il terzo confronto utilizzeremo la tavola di decimalizzazione:

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
0	1	1	1	1	0	0	0	0	0	1	1	1	1	1	1

PIN di prova:  $p = 1111$

## Schema Adattivo



Confrontiamo il PIN di prova cifrato 1111, con il risultato ottenuto mappando il PIN originale 3491 con la tavola di decimalizzazione scelta, ovvero 1101

Il risultato del confronto negativo, ci dice che dobbiamo eliminare i PIN composti unicamente composti dalle cifre 1,2,3,4,5

Quindi dal nodo corrente ci spostiamo sul figlio destro, passando da 1695 possibili PIN a 1326

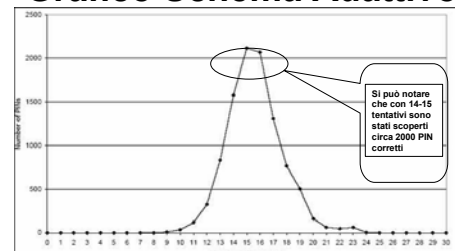
## Esecuzione Schema Adattivo



N° di tentativi	Numero di PIN Possibili	Tavola di Decimalizzazione $D_x$	PIN di prova cifrati	$D_x(P_{orig})$	$P_x = D_x(P_{orig})$
1	10000	1000001000100000	0000	0000	Si
2	4096	0100000000010000	0000	0001	No
3	1695	0111110000011111	1111	1101	No
4	1326	0000000100000000	0000	0000	Si
5	736	0000000010000000	0000	0000	Si
6	302	0010000000001000	0000	0000	Si
7	194	0001000000000100	0000	1000	No
8	84	0000110000000011	0000	0100	No
9	48	0000100000000010	0000	0100	No
10	24	0100000000010000	1000	0001	Si
11	6	0001000000000100	0100	1000	No
12	4	0001000000000100	0010	1000	No
13	2	0000100000000010	0100	0010	No

Esecuzione completa dell'algoritmo adattivo per il Pin 3491

## Grafico Schema Adattivo



Il diagramma ottenuto, facendo iterare l'algoritmo analizzato, ci indica il numero dei tentativi che sono stati effettuati e quanti PIN corretti sono stati trovati

Presentando:

sull'asse delle ordinate il numero dei PIN che possono essere individuati e su quello delle ascisse il numero dei tentativi necessari a determinare i PIN corretti

## Schema adattivo con offset del PIN



Quando l'assalitore non può ottenere nessun PIN di prova cifrato:

- può utilizzare il parametro di Offset
- deve aver intercettato l'EPB che contiene il PIN corretto

Quindi alla funzione dovrà passare:

- EPB
- Offset, usato per modificare il PIN
- tavola di decimalizzazione modificata
- PAN DATA

## Schema adattivo con offset del PIN



Si presuma che un blocco di PIN codificato, contenente il PIN corretto, viene intercettato

Per semplicità consideriamo che, il proprietario del conto non ha cambiato il suo PIN e l'Offset corretto del PIN è 0000

## Schema adattivo con offset del PIN

- Usando il seguente insieme di tavole di decimalizzazione, l'assaltore può determinare quali cifre sono presenti nel PIN corretto:

$$D_i[x] = D_{orig}[x] + 1 \text{ se } D_{orig}[x] = i,$$

$$D_i[x] = D_{orig}[x] \text{ altrimenti}$$

## Schema adattivo con offset del PIN

- Supponiamo che il PIN sia 7816
- Per determinare se la cifra 7 è presente utilizzeremo la tavola di decimalizzazione  $D_7$ :

0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5
0	1	2	3	4	5	6	8	8	9	0	1	2	3	4	5

- Utilizzeremo l'Offset 0000

## Schema Statico

- Operazione di verifica della funzione:
  - cifra il PAN DATA => 78A6 2210 2290 4509
  - prende le prime 4 cifre => 78A6
  - le decimalizza => 8806
  - decifra l'EPB => 7806
  - lo somma con l'offset 0000 => 7806
  - effettua il matching tra 8806 e 7806
  - ritorna false

- Quindi si può dire che il test fallisce quando il PIN originale contiene la cifra  $i$

## Schema adattivo con offset del PIN

- Ora non ci resterà che determinare le posizioni giuste delle cifre trovate
- Supponiamo di voler determinare in un PIN 1583 la corretta posizione di 8
- Alla funzione EPV passeremo:
  - EPB intercettato
  - PAN\_Data
  - tavola di decimalizzazione modificata
  - offset 0001

## Schema adattivo con offset del PIN

- Encrypted\_PIN\_Verify calcola il PIN originale mappato con la tavola modificata:  
PIN originale decimalizzato = 1593
- A questo punto la funzione sommerà l'offset all'EPB intercettato:  
EPB intercettato + offset dell'ipotesi = 1583 + 0001 = 1584
- Nel caso dell'offset 0001 non troveremo match tra il PIN calcolato dalla funzione e la somma tra l'EPB intercettato e l'offset dell'ipotesi

## Schema adattivo con offset del PIN

- Nell'esempio, per la cifra 8 avremo un match positivo per l'offset 0010 (8 si troverà nella posizione corrispondente alla cifra 1 nell'offset)

Offset dell'ipotesi	Tavola di decimalizzazione dell'ipotesi	EPB Intercettato	EPB Intercettato + Offset dell'ipotesi	PIN originale decimalizzato	Risultato
0001	0123 4567 9901 2345	1583	1584	1593	NO
0010	0123 4567 9901 2345	1583	1593	1593	SI
0100	0123 4567 9901 2345	1583	1683	1593	NO
1000	0123 4567 9901 2345	1583	2583	1593	NO

## Schema adattivo con offset del PIN



- Questa procedura è ripetuta finché non sono note le posizioni di tutte le cifre che compongono il PIN originale
- Mediamente sono richieste circa 16 ipotesi per determinare il PIN corretto

## Conclusioni



- Per ovviare a questi attacchi è stato proposto di:
  - cambiare o aggiornare il software degli HSM (Hardware Security Module)
  - perfezionare le tavole di decimalizzazione in modo tale da prevenire questi attacchi

## Conclusioni



- Però queste alternative non sono applicabili a causa del costo delle operazioni e nella difficile implementazione di nuovi algoritmi



## Bibliografia



- Jolyon Clulow: *The Design and analysis of cryptographic application programming interfaces for security devices Version 4.0*
- Mike Bond, Piotr Zielinski: *Decimalisation table attacks for PIN cracking February 2003*
- <http://www.cam.ac.uk> Università di Cambridge

## Bibliografia



- <http://inventors.about.com/inventors/blatm.htm>  
Sito in cui è possibile trovare notizie sull'inventore dell'ATM
- <http://www.thocp.net/hardware/atm.htm>  
Sito in cui è possibile trovare le origini degli ATM
- <http://telemat.det.unifi.it/book/Security/modelli.htm>  
Sito riguardante la sicurezza dei sistemi di pagamento elettronico

FINE