On Smart Cards Security

Ilya O. Levin
D'Crypt Pte Ltd

Asiacrypt 2010

Common Beliefs

Obscure security tokens have security issues

An ISO-compliant smart card is more secure than an obscure security token

Attacking smart cards is hard

Smart Cards

Physicalize cryptographic secrets by having non-exportable objects (keys, PINs, etc.)

~~Schlumberger~~ Cryptoflex

~~Axalto~~

Gemalto

Cryptoflex Secrets

Elementary Files with Read AC set to "Never Allowed"

0000	Cardholder Verification File (CHV1)
0001	Internal keys (DES, 3DES)
0011	External keys (AUT, DES, 3DES)
0012	RSA private keys (1012 – public)
0100	Cardholder Verification File (CHV2)

CHV File Format

Byte(s)	Description	Length
1	File activation byte, LSB	1
2-3	RFU	2
4-11	PIN value	8
12	Number of verification attempts allowed	1
13	Remaining verification attempt counter	1
14-21	Unblocking PIN value	8
22	Number of unblocking attempts allowed (10)	1
23	Remaining unblocking attempt counter	1

Internal Key File Slot Format

Byte(s)	Description	Length
1	RFU (!=0)	1
2	Key length or	1
	00 = last slot	
	01 = empty slot	
3	Algorithm ID	1
	00 = Single-length DES, 56-bit	
	02 = Double-length 3DES, 112-bit	
4-11 or 4-19	Key value	8/16
12 or 20	RFU (!=0)	1

External Key File Slot Format

Byte(s)	Description	Length
1	RFU	1
2	Key length or 00 = last slot 01 = empty slot	1
3	Algorithm ID 00 = Single-length DES, 56-bit 02 = Double-length 3DES, 112-bit	1
4-11 or 4-19	Key value	8/16
12 or 20	Number of verification attempts allowed	1
13 or 21	Remaining verification attempts counter	1
14 or 22	Next key data, starting from a key length	

RSA 1024-bit Private Key File Format

Byte(s)	Description	Length
1	Key block length, MSB = 01	1
2	Key block length, LSB = 43h (323 bytes)	1
3	Key number	1
4-67	Public modulus secret prime factor *P*	64
68-131	Public modulus secret prime factor *Q*	64
132-195	Inverse of the factor *P* ($a = Q^{-1} \bmod P$)	64
196-259	Private subexponent ($c = Ks \bmod (P-1)$)	64
260-323	Private subexponent ($f = Ks \bmod (Q-1)$)	64
…	…	…
EOF	00 00 00	3

Cryptoflex Authentication

CHV1	User PIN
CHV2	Additional PIN (optional)
AUT1	Transport Key (2nd slot of 3f00/0011)

still cannot read EFs with Never Allowed AC

Cryptoflex Authentication

We will skip the details of all related issues here

Assume CHV and AUT1 are known

Ask me offline if interested

Cryptoflex Secrets. The Fun Part

The non-exportable secrets are stored in the elementary files

- We can bypass standard commands and modify content directly with Update Binary/Update Binary Enciphered

Elementary files are transparent, not linear

- We can modify anything in chunks of any size or one byte at a time at any offset

Cryptoflex Secrets. The Fun Part

How to extract the first 112-bit 3DES encryption key from the Internal Key File?

```
1. Establish the card context and satisfy AC
2. Let E = DES_Block_Init(dummy)
3. for i in [4 … 19]
4.     for b in [0 … 255]
5.             Update the i^th byte in EF 0001 with b
6.             if ( DES_Block_Init(dummy) ≡ E ) print b
```

Cryptoflex Secrets. The Fun Part

How to extract the first RSA-1024 private key?

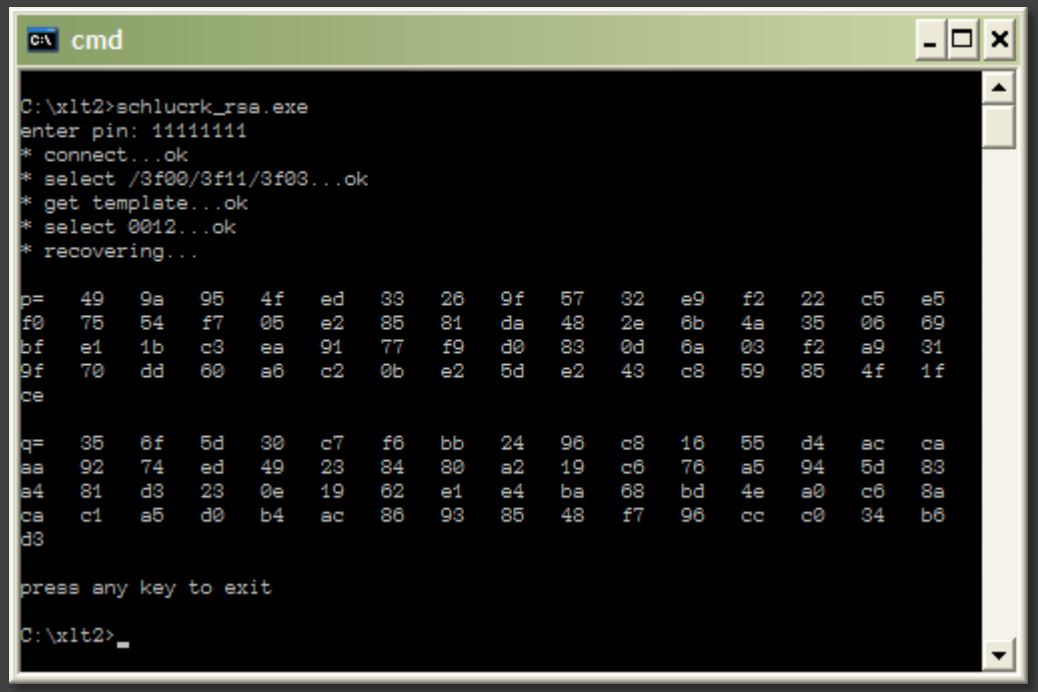
Same as a 3DES key.

Use RSA Signature (Internal Auth) command instead of DES Block Init.

Update bytes 4-67 in EF 0012 to recover the secret factor *P* and bytes 68-131 to recover the secret factor *Q*

Cryptoflex Secrets. The Fun Part

The PoC code recovers a 3DES key in ~5 min and a private RSA-1024 key in ~20 min



```
C:\xlt2>schlucrk_rsa.exe
enter pin: 11111111
* connect...ok
* select /3f00/3f11/3f03...ok
* get template...ok
* select 0012...ok
* recovering...

p=   49   9a   95   4f   ed   33   26   9f   57   32   e9   f2   22   c5   e5
f0   75   54   f7   05   e2   85   81   da   48   2e   6b   4a   35   06   69
bf   e1   1b   c3   ea   91   77   f9   d0   83   0d   6a   03   f2   a9   31
9f   70   dd   60   a6   c2   0b   e2   5d   e2   43   c8   59   85   4f   1f
ce

q=   35   6f   5d   30   c7   f6   bb   24   96   c8   16   55   d4   ac   ca
aa   92   74   ed   49   23   84   80   a2   19   c6   76   a5   94   5d   83
a4   81   d3   23   0e   19   62   e1   e4   ba   68   bd   4e   a0   c6   8a
ca   c1   a5   d0   b4   ac   86   93   85   48   f7   96   cc   c0   34   b6
d3

press any key to exit

C:\xlt2>_
```

Cryptoflex Anamnesis

It is possible to recover cryptographic keys out of “non-exportable” objects

Read access condition restrictions on key files are irrelevant

“*We do not consider this to be a security issue*”

To conclude

Smart cards are not that perfect in real life as we may believe

Poking smart cards is fun, no shiny hardware required

We may need more public research in this area

Thank you