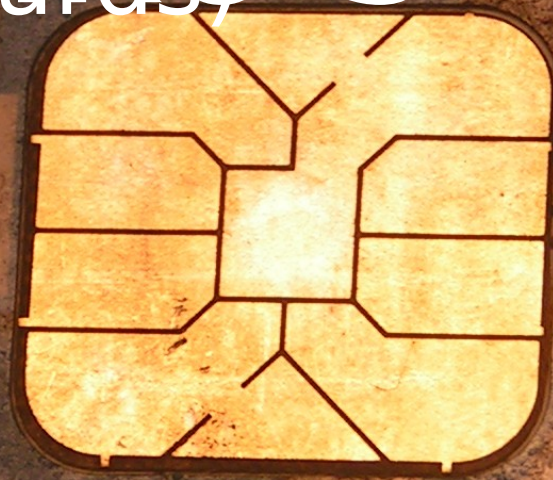


# Hacking FedEx

(How not to implement stored-value smart  
cards)

## Kinko's



Strom Carlson  
Secure Science Corporation  
DEF CON 14  
4 August 2006

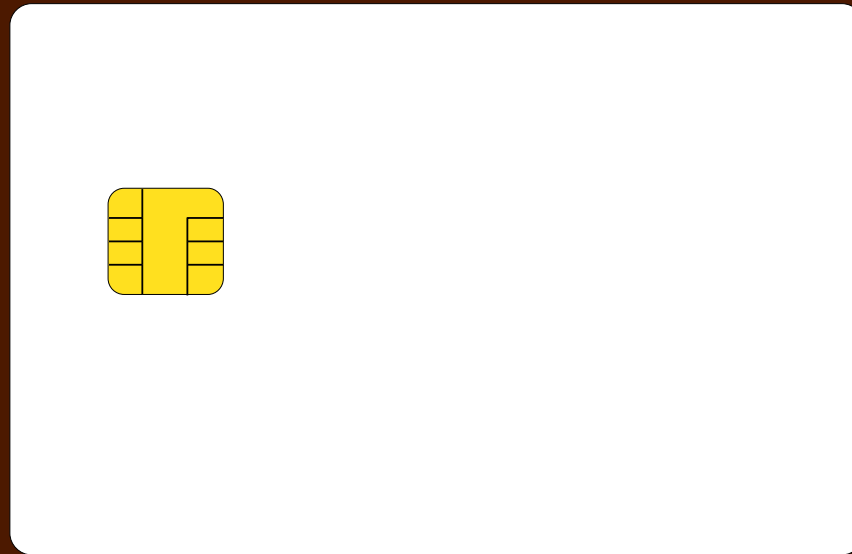
# Part I

# THEORY

# What is a smart card?

Plastic card, typically credit-card sized,  
which carries a microchip

Can use a contact pad, a built-in antenna,  
or both

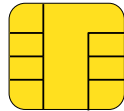


# What is a smart card?

Plastic card, typically credit-card sized,  
which carries a microchip

Can use a contact pad, a built-in antenna,  
or both

I'M SMART!  
LOL



Almost smarter than script kiddies

# Two varieties of cards

## Microprocessor

Typically includes a small microprocessor, some RAM, and some flash ROM.

Often optimized for cryptographic functions  
SIM cards in GSM phones are one example

## Memory

Simple EEPROM, sometimes with a microcontroller

Can sometimes perform basic security functions

# Meet the SLE4442

256-byte EEPROM

First 32 bytes are irrevocably write-protectable

All 256 bytes are readable at any time

Card can only be written to after presenting a three-byte security code to the microcontroller

Card becomes unwritable after three incorrect attempts at writing the security code

<http://www.stromcarlson.com/> | <http://www.securescience.net/>

You can brute force the code if you have F.C

# Meet the SLE4442



# Why use the SLE4442?

One of the cheapest cards on the market

\$0.36 each in quantities of 200,000

<http://www.smartcardworld.com/SLE4442.asp>

Security function prevents casual attacker from altering data

More durable and secure than a magstripe card



# Who uses the SLE4442?

# FedEx Kinko's

(Some random tiny xerox shop you've obviously never heard of)

# ExpressPay

Stored-value card system utilizing the SLE4442

Customers add cash value to cards at a kiosk

Cards are debited as users make copies, use computers, and so on.

Cards can be refilled

Developed by enTrac Technologies of Toronto

Implemented at Kinko's in 2001

# ExpressPay Questions

Is any personally identifiable information stored on the card?

Is a transaction history stored on the card?

Customers can print receipts at the kiosk after using self-service systems

How secure is the data?

Card offers no built-in cryptographic function

Data could theoretically be encrypted before being stored on the card itself

# ExpressPay Questions

Is value even stored on the card?

Cards could just be serialized tokens with all value stored on the back-end

What else is stored on the card?

# Reading the SLE4442

Card conforms only to the following ISO standards:

7816-1: Physical Characteristics

7816-2: Dimensions and Locations of Contacts

7816-3: Electrical characteristics and class indication for integrated circuit(s) cards operating at 5V, 3V and 1.8V

# Reading the SLE4442

Card does not conform to ISO 7816-4

Organization, security and commands for interchange

Contents of command-response pairs exchanged at the interface

Means of retrieval of data elements and data objects in the card

Structures and contents of historical bytes to describe operating characteristics of the card

Structures for applications and data in the card, as seen at the interface when processing commands

(continued...)

# Reading the SLE4442

## ISO 7816-4 Continued:

Access methods to files and data in the card

A security architecture defining access rights to files and data in the card

Means and mechanisms for identifying and addressing applications in the card

Methods for secure messaging

Access methods to the algorithms processed by the card

# Reading the SLE4442

This card cannot be read in a reader which expects only ISO 7816-4 compliant cards.

American Express “Blue” Card readers...no such luck.

Some readers are able to read this card plus a variety of other memory cards



# Reading the SLE4442

## Advanced Card Systems ACR-30U

USB reader

Windows and Linux drivers

Reads the SLE4442

Costs about \$30.00 USD



# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	32-byte header which remains the same across all cards
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 00 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Dollar value stored on the card
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 50 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Date and time the card was first issued
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	YY-MM-DD
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	05-09-21
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	HH:MM:SS.SS
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Serial number 11630032309
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Store number where card was issued  1163
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Individual card number 0032309
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	



# What's on the card?

0x00	A2 13 10 91 46 FF 81 15	0x80	72 00 00 00 00 00 00 00	Another timestamp
0x08	FF FF FF FF FF FF FF FF	0x88	00 00 00 00 00 00 00 39	
0x10	FF FF FF FF FF D2 76 00	0x90	39 31 31 00 31 30 31 00	
0x18	00 04 09 FF FF FF FF FF (1)	0x98	30 30 30 30 30 00 00 00	
0x20	00 00 00 00 00 00 F0 3F (2)	0xA0	00 00 00 00 00 00 00 00	
0x28	00 00 00 00 00 00 00 00	0xA8	00 00 00 00 00 00 00 00	
0x30	20 05 09 21 16 05 45 69 (3) (4)	0xB0	00 00 00 03 00 00 01 00	
0x38	00 00 00 00 00 00 00 00	0xB8	00 00 00 00 00 00 00 00	
0x40	00 00 FF FF FF FF FF FF	0xC0	00 00 00 00 00 00 00 20 (6)	
0x48	FF FF FF FF FF FF FF FF	0xC8	05 09 21 16 05 45 69 00 (7)	
0x50	FF FF FF FF FF FF FF FF	0xD0	00 00 00 FF FF FF FF FF	
0x58	FF FF FF FF FF FF FF FF	0xD8	FF FF FF FF FF FF FF FF	
0x60	31 31 36 33 30 30 33 32	0xE0	FF FF FF FF FF FF FF FF	
0x68	33 30 39 00 00 00 00 00 (5)	0xE8	FF FF FF FF FF FF FF FF	
0x70	00 00 00 00 43 61 73 68	0xF0	FF FF FF FF FF FF FF FF	
0x78	20 43 75 73 74 6F 6D 65	0xF8	FF FF FF FF FF 00 00 00	

# Stored Dollar Value

Initially confusing, as all other values stored on the card are fairly easy-to-understand plaintext

Some values:

7B 14 AE 47 E1 7A A4 BF - \$0.04

00 00 00 00 00 00 00 00 \$0.00

7B 14 AE 47 E1 7A 84 3F \$0.01

7B 14 AE 47 E1 7A A4 3F \$0.04

9A 99 99 99 99 99 A9 3F \$0.05

9A 99 99 99 99 99 C9 3F \$0.20

00 00 00 00 00 00 F0 3F \$1.00

66 66 66 66 66 66 1E 40 \$7.60 <http://www.strocarlson.com> <http://www.securescience.net/>



# Security Code

Card is protected only by a three byte (24 bit) security code

16.7 million possible combinations

All Kinko's cards likely have the same code

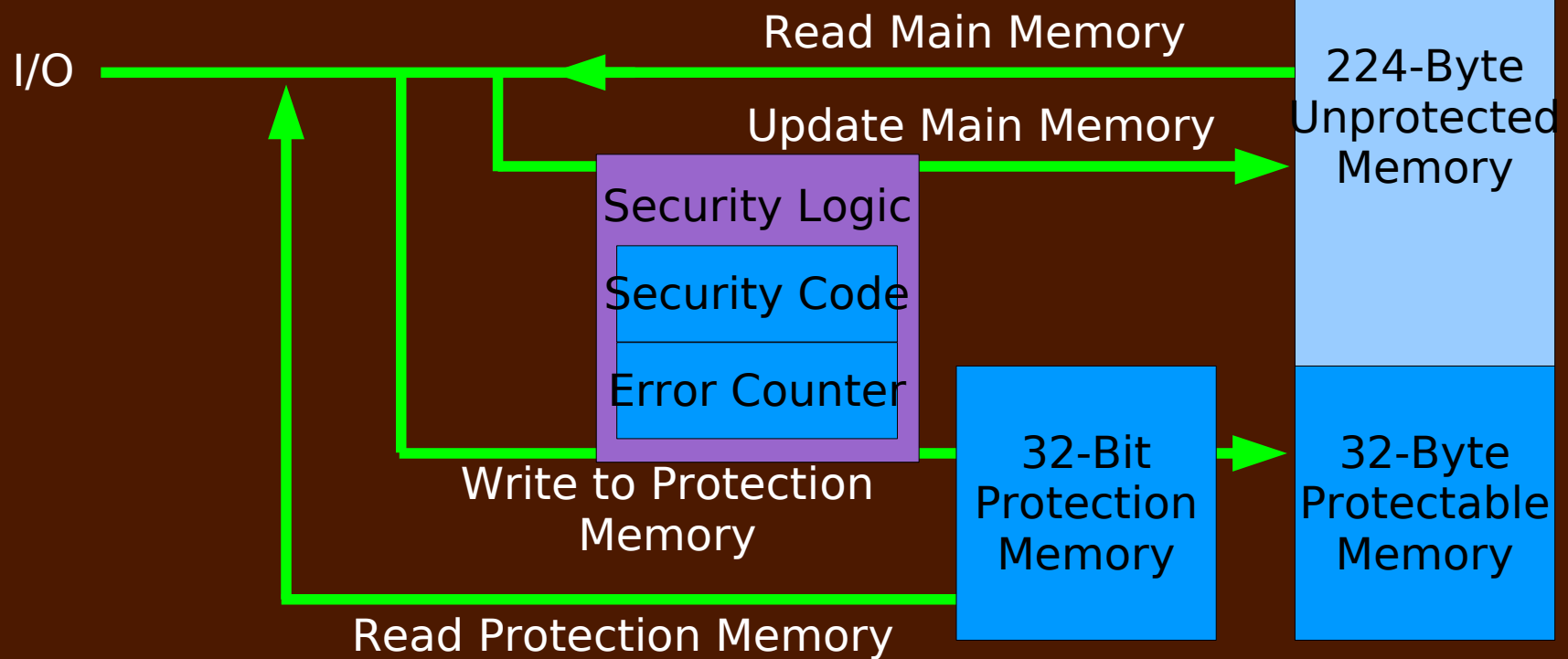
Blank cards from the kiosk do not have the default manufacturer's code

Deriving the code from changeable data may cause card to become inoperable if changeable data becomes corrupted somehow

Everything else about this card is ass easy

# Attack?

Security code can only be read once the correct code has been written to the hardware



# Possible Attacks

Social Engineering

Emulate the SLE4442

Intercept the code during transmission

Read memory directly

# Social Engineering Attack

Contact enTrac technologies and extract the code from some unsuspecting employee

## PROS:

Requires the least technical jiggery-pokery of the available methods; just dial the phone

## CONS:

Code might be so secret that no one but enTrac engineers know about it

Tough to repeat if the code is changed

No technical challenge

# Emulate the SLE4442

Use a smartcard emulator plugged into a laptop

## PROS:

Fairly foolproof; this method is used by developers

## CONS:

Emulator dongles are bulky and don't work well with the motorized transport found in some card readers; the dongle won't go all the way into the slot

SLE4442 emulation software might be difficult to obtain or might not exist at all



# Emulate the SLE4442

Find a microprocessor-based smart card that can behave just like the SLE4442

## PROS:

Easy to clandestinely retrieve the code by sticking the card into any device which attempts to write to it

Elegant attack that renders any SLE4442 system vulnerable

## CONS:

Every microprocessor card I've looked at seems to follow ISO 7816, not SLE4442

# Intercept the security code

Wire the card's contact points to a logic analyzer, capture a transaction, and analyze the data later

## PROS:

Small USB logic analyzers are readily available for under \$300

Wiring can be easily hidden; little chance of card rejection since you're using a real SLE4442

## CONS:

Easy to screw up if you don't have soldering skills

# Read memory directly

Burn the epoxy off the chip and read the security memory directly

## PROS:

None of that tedious mucking around with transactional data

## CONS:

**EXPENSIVE** unless you know someone with the ability and equipment to read directly off the silicon die.

# Part II

# ATTACK!

# Logic Analyzer Attack

Solder wires to a stored-value card

Attach logic analyzer

Go to Kinko's



Check out my totally awesome uber-leet soldering skillz

<http://www.stromcarlson.com/> | <http://www.securescience.net/>



# Commands

- 00001100 – Read Main Memory
- 00011100 – Update Main Memory
- 00101100 – Read Protection Memory
- 00111100 – Write Protection Memory
- 10001100 – Read Security Memory
- 10011100 – Write Security Memory
- 11001100 – Compare Verification Data

# SLE4442 Command Structure

Command Byte

Address Byte

Data Byte



# Security Code Presentation

Read Security Memory

Update Security Memory

3x Compare Verification Data

Numbered byte indicating which byte of security code follows (01, 02, or 03)

One byte of security code

Update Security Memory

Read Security Memory

# Stepping through the data

```

ANSWER TO RESET:      00000100  20  [BEGIN TIMESTAMP]  11111111  FF
01000101  A2          10100000  05  |                11111111  FF
11001000  13          11000000  03  |                11111111  FF
00001000  10          00010100  28  |                11111111  FF
10001001  91          01000100  22  |                11111111  FF
                   01101100  36  |                11111111  FF
0          PROCESSING CYCLE  10100000  05  V                11111111  FF
                   01100110  66  [END TIMESTAMP]  11111111  FF
00001100  READ MAIN MEMORY  00000000  00                11001100  33  [BEGIN SERIAL NUMBER]
10101000  15          00000000  00                11101100  37  |
11111111  FF          00000000  00                00001100  30  |
                   00000000  00                01001100  32  |
0          PROCESSING CYCLE  00000000  00                00001100  30  |
                   00000000  00                00001100  30  |
01001011  D2  [BEGIN HEADER]  00000000  00                01101100  36  |
01101110  76  |                00000000  00                00001100  30  |
00000000  00  |                00000000  00                00001100  30  |
00000000  00  |                00000000  00                00011100  38  V
00100000  04  |                11111111  FF                10001100  31  [END SERIAL NUMBER]
10010000  09  |                11111111  FF                00000000  00
11111111  FF  |                11111111  FF                00000000  00
11111111  FF  |                11111111  FF                00000000  00
11111111  FF  |                11111111  FF                00000000  00
11111111  FF  V                11111111  FF                00000000  00
11111111  FF  [END HEADER]  11111111  FF                00000000  00
00011101  B8  [BEGIN VALUE]  11111111  FF                00000000  00
01111000  1E  |                11111111  FF                00000000  00
10100001  85  |                11111111  FF                00000000  00
11010111  EB  |                11111111  FF                11000010  43
10001010  51  |                11111111  FF                10000110  61
00011101  B8  |                11111111  FF                11001110  73
01111001  9E  V                11111111  FF                00010110  68
11111100  3F  [END VALUE]  11111111  FF                00000100  20
00000000  00                11111111  FF                11000010  43
00000000  00                11111111  FF                10101110  75
00000000  00                11111111  FF                11001110  73
00000000  00                11111111  FF                00101110  74
00000000  00                11111111  FF                11110110  6F
00000000  00                11111111  FF                11110110  6F
00000000  00                11111111  FF                [CONTINUED...]

```

# Stepping through the data

(thousands of very boring bytes later)

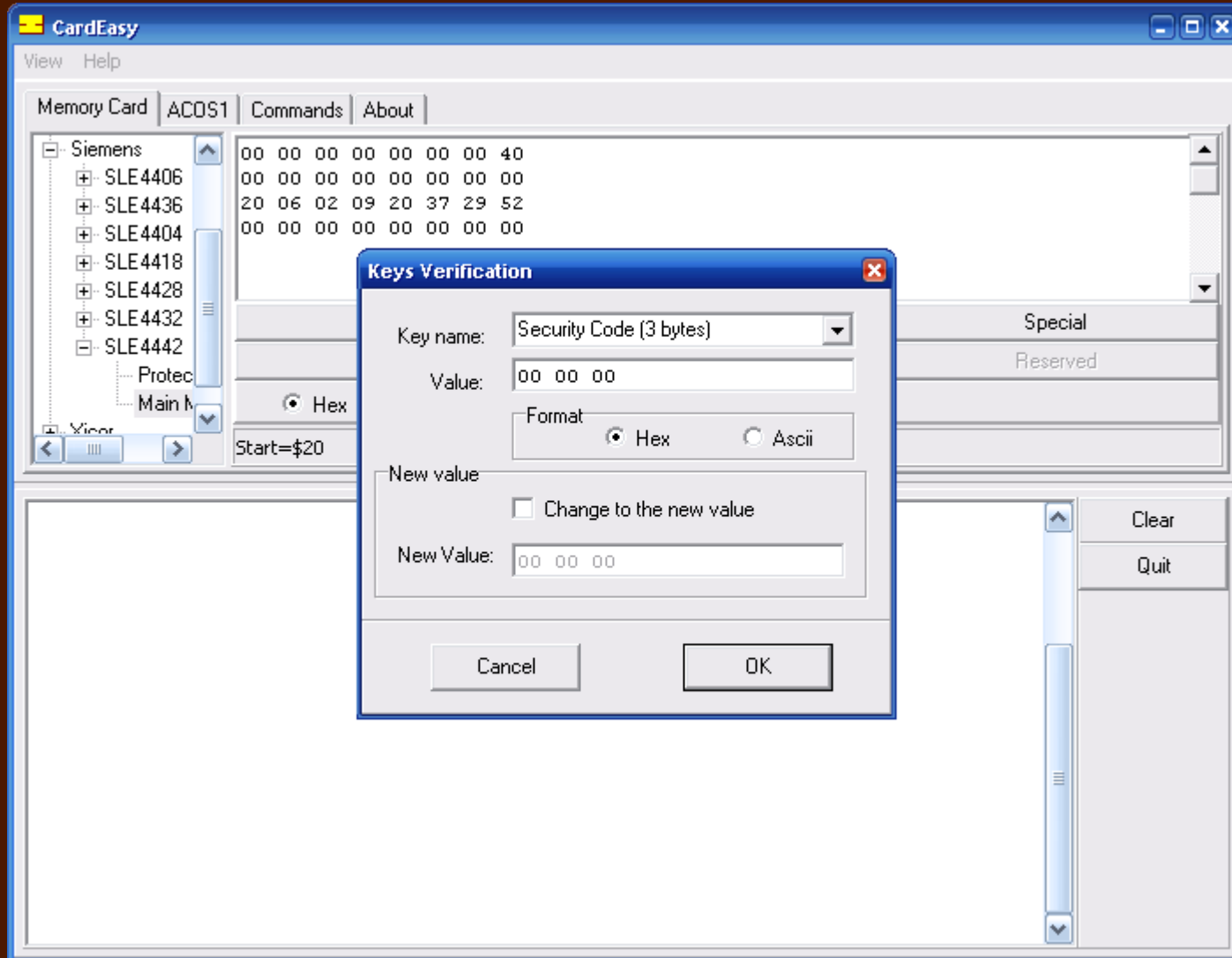
# Stepping through the data

```

11111111 00000000 00000000
11111111 00000000 00000000
11111111 00000000 00000
11111111 00000000 10001100 READ SECURITY MEMORY
11111111 00000 11111111
00000000 11001100 COMPARE VERIFICATION DATA 11111111
00000000 10000000 01 11110000
00000000 [END OF MAIN MEMORY] XXXXXXXX XX
0 1000 11001100 COMPARE VERIFICATION DATA ....
10001100 READ SECURITY MEMORY XXXXXXXX XX
11111111 FF 1000 WHY AM I STILL ANALYZING THIS?
11111111 FF 11001100 COMPARE VERIFICATION DATA
11110000 11000000 03
00000000 XXXXXXXX XX
00000000 1000
00000000 10011100 UPDATE SECURITY MEMORY
00 00000000
10011100 UPDATE SECURITY MEMORY 11100000 ) WHY HELLO THERE !!!!!!!
00000000 10000000 (.) I'M THE INFORMATION SECURITY
01100000 00000000 .|. RESPONSIBILITY CUPCAKE
10000000 00000000 17J AND IT'S MY JOB TO TELL YOU
00000000 00000000 | | THAT YOU ARE GOING TO HAVE TO
00000000 00000000 | | FIGURE OUT THE SECURITY CODE
00000000 00000000 .-| |--.- FOR YOURSELF !!!!!!!!
00000000 00000000 & & ; & ; ; \
00000000 00000000 \ ; & & /
00000000 00000000 F""""---...---""""J
00000000 00000000 | | | | | | | |
00000000 00000000 J | | | | | | F ALSO: DEAD HOOKERS
00000000 00000000 `---.|.|.|.---'
00000000 00000000
00000000 00000000

```

# Manipulating the card



# INTERNETS!

Maximum Balance Reached.

313.37

You cannot put any more money on this card.

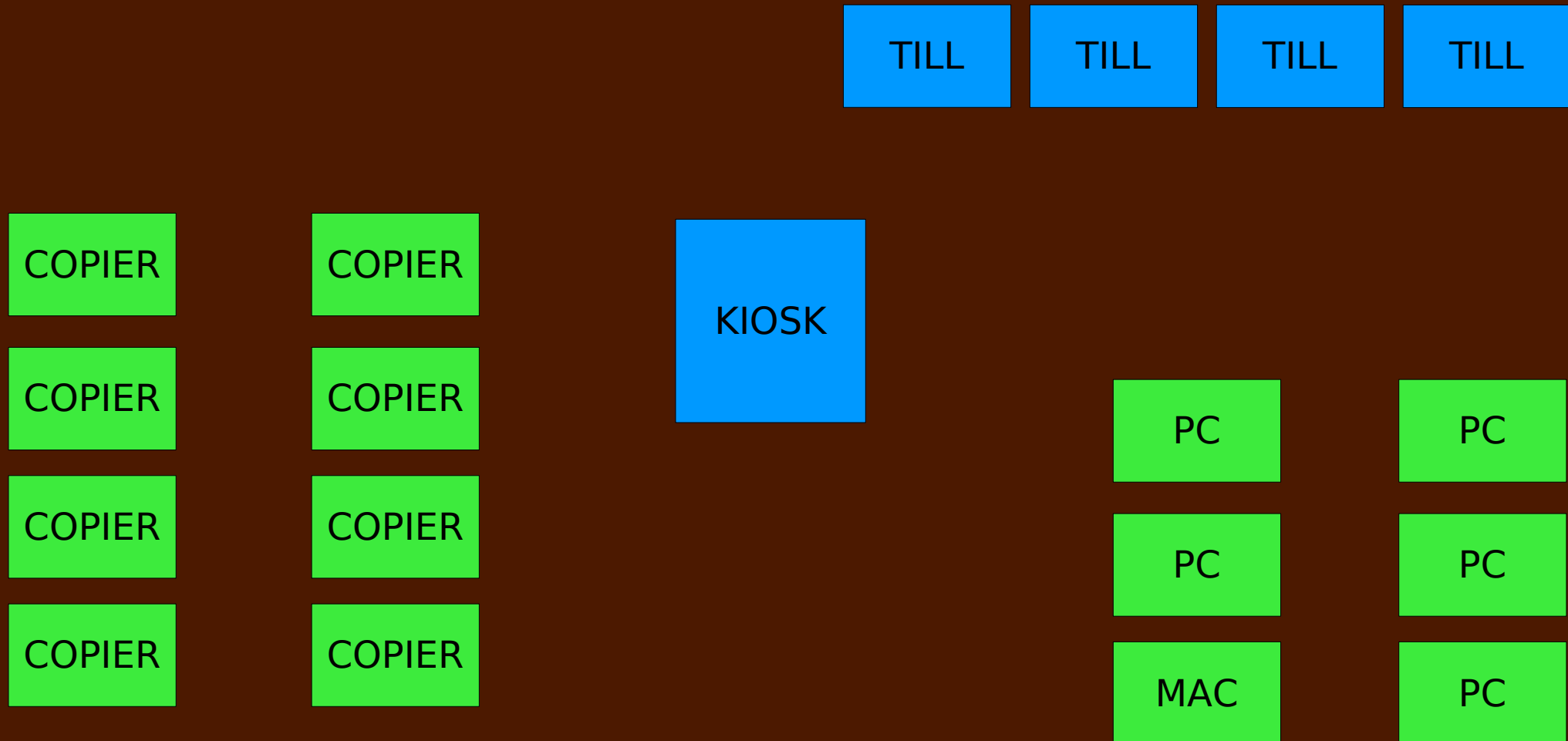
Return My Card

Get a Receipt

# Part III

## PLAYING WITH THE BACK END

# A typical FedEx Kinko's





# Card cloning

Does the system reconcile the balance on the back end with the balance on the card?

Step 1: Buy \$1 card

Step 2: Clone card

Step 3: Make a few xeroxes with original card

Step 4: Print receipt from kiosk with cloned card

# Value Alteration

Does the system do anything if the card balance mysteriously increases?

Step 1: Buy \$1 card

Step 2: Rewrite card value to \$2

Step 3: Make several xeroxes (less than \$1 worth)

Step 4: Print receipt at kiosk

# Weird Serial Numbers

Does the system freak out if the card's serial number is from a nonexistent store?

Does the system verify that the serial number is valid?

Step 1: Buy \$1 card

Step 2: Alter serial number to something unlikely (99687654321 for example – there is no store 9968)

Step 3: Make xerox with altered card

Step 4: Print receipt

# Cloned Cards Part II

Is a card's serial number invalidated if you redeem the card for its stored value?

Step 1: Buy \$1 card and destroy it

Step 2: Buy \$1 card

Step 3: Make xerox with card

Step 4: Clone the card

Step 5: Redeem original card

Step 6: Make xerox with cloned card

Step 7: Print receipt at kiosk with cloned card

# Cloned Cards Part II

OK, maybe it takes some time for the card to be invalidated.

Step 8: Go eat pizza or something

Step 9: Come back and try the cloned card again

# Cloned Cards Part II

Maybe, just maybe, the system might take a whole day to invalidate the card.

Step 10: Go back a day or six later

# Fun Facts

enTrac Technologies has exactly one product: ExpressPay

The company slogan is “Counter Intelligence”

# Part IV

# ENGINEERING A BETTER SYSTEM



# Keep the SLE4442

Change the way values are stored

Change the way the security code works

Change the method for verifying the cards

## PROS

- Does not require hardware changes

- Relatively inexpensive

## CONS

- Still somewhat insecure

# Increasing Security with the SLE4442

## Verify information on the cards

Generate a hash based on the serial number, value, and timestamp, and verify that against a hash stored on the server

Store a second hash on the card based on the same data to verify the card has not been altered

## Don't store values in plaintext on the cards

Make it more difficult to reverse-engineer the contents of the card

# Increasing Security with the SLE4442

Do not store value on the cards themselves

Use the cards only as serialized tokens and pull the value from the network

Store a hash on the card to verify that the value on the network hasn't been altered

Invalidate the cards when they're cashed out

(duh)

# Increasing Security with the SLE4442

Don't use the same security code for every  
single card in circulation

Use a code derived from some randomized  
rotating value stored on the network

Do not base the code on any value stored on  
the card

# Use a different chip

Use a cryptographic secure memory chip

Atmel CryptoMemory chips used by my  
laundromat

Use a chip with a microprocessor

Challenge-response authorization

Encryption of data

Access control

Hidden Goatse

# Charge a deposit for the smartcard

Currently, cards are free for the taking

“More secure cards are too expensive”

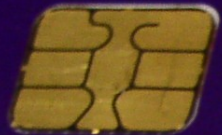
Charge \$1 or \$2 to obtain the card

Refund the deposit when the card is returned to an employee

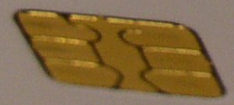
This will help prevent curious tinkerers from obtaining massive numbers of cards for play and analysis



**FedEx** Kinko's<sup>SM</sup>  
Office and Print Center



**fedexkinkos.com**  
1.800.254.6567



6184442

# Resources

<http://www.securescience.net/>

<http://www.stromcarlson.com/>

<http://www.infineon.com/>

<http://www.atmel.com/>

<http://www.smartcardsupply.com/>



# Q&A

