Security Failures In Secure Devices

Black Hat DC – February 21, 2008

Christopher Tarnovsky

Flylogic Engineering, LLC.

chris@flylogic.net – www.flylogic.net

Who am I?

- Last 10 years with NDS
  - Anti-piracy effort
  - IC design
  - Software engineer
  - Reverse-engineer expert
  - One patent, one pending

Purpose of this briefing?

- Awareness
- Understanding
- Improve

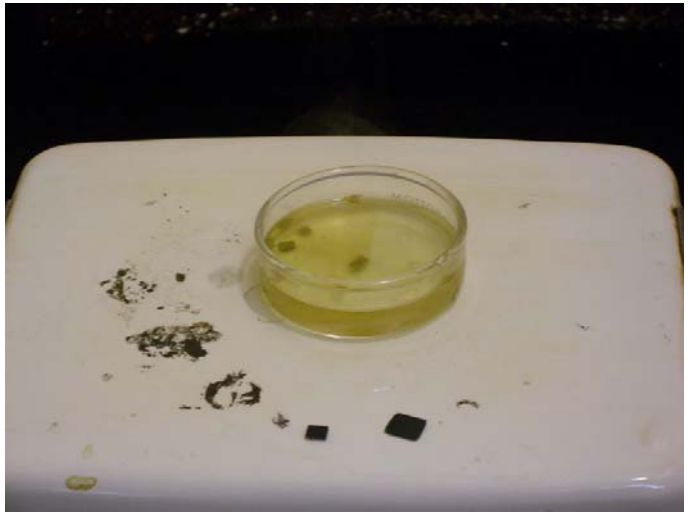
How are failures found?

- Decapsulation of the substrate
- Microscopy
- Invasive probing
- Electrical glitches
- Optical glitches

Decapsulation

- Hot Plate
- Acetone
- Fuming Nitric Acid
- Fuming Sulfuric Acid
- Tweezers
- Dropper

Typical Decap Session



Microscopy

- Use of brightfield optical microscopes
- Zeiss Axiotron (I/II):
  - Good for general imaging to plan attack
- Mitutoyo FS-[50-70]:
  - Good to use for execution of an attack

Invasive Probing

- Physical connection to substrate
- Use low-capacitance buffered driver
- Tri-stated buffer is desired-
  - Allow eavesdropping
  - Overdrive the signal on an event (a trigger)

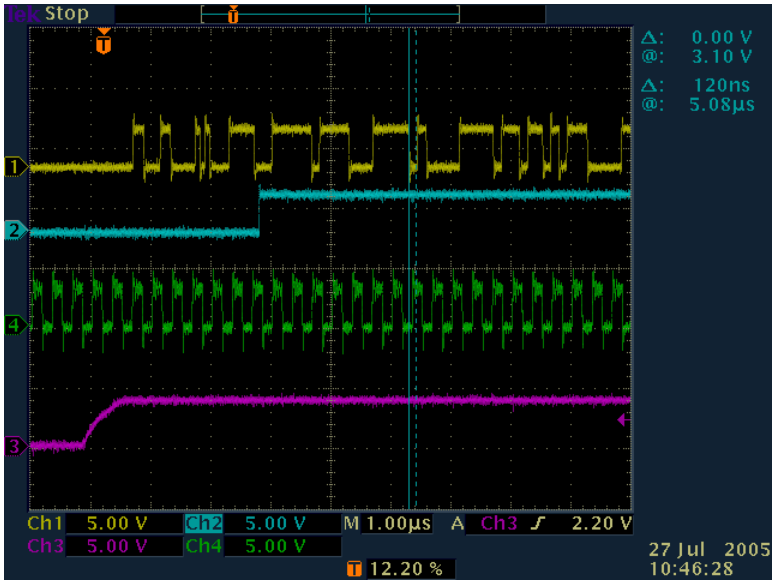
Probing: Typical bus action (listening)

YELLOW: Databus signal

GREEN: Clock

PURPLE: Reset

BLUE: Trigger



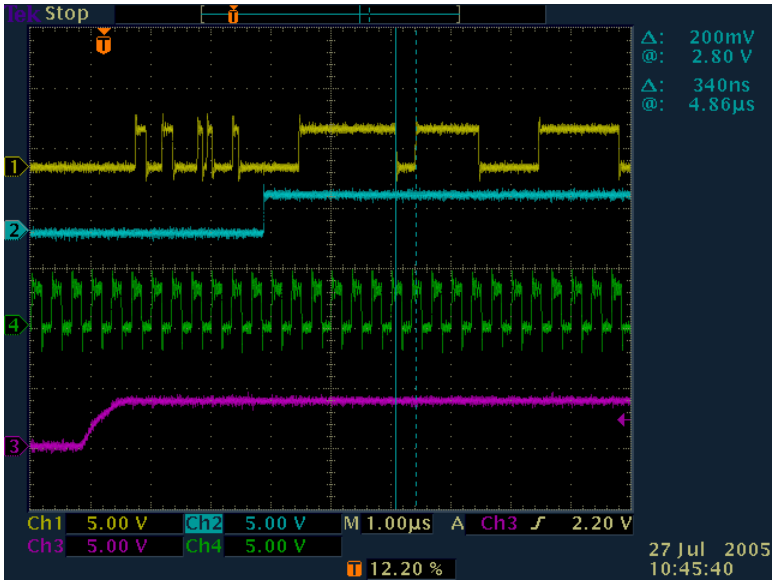
Overdriving last slides databus with a logic ‘0’

YELLOW: Databus signal

GREEN: Clock

PURPLE: Reset

BLUE: Trigger



Electrical Glitches

- Lower input voltage
- Increase clock frequency

Q: Desired result?

A: Lengthen propagation delay!!!

Optical Glitches

- Triggered pulses of light
- Hope for latching of something other than, “good” (e.g. dptr change)

Most devices claim some type of security

- Cryptographic Memories
- Smartcard MCU's
- Off-the-shelf (OTS) MCU's

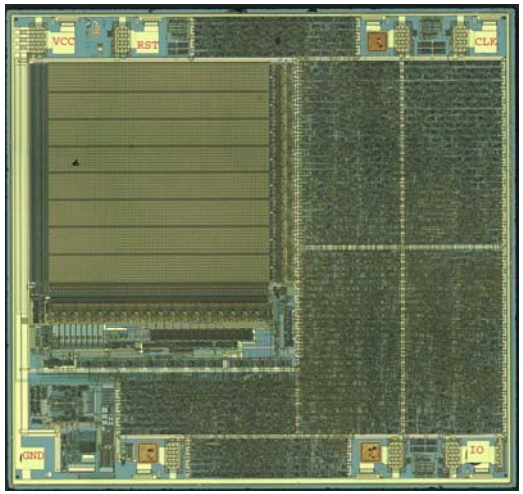
Cryptographic Memories

- Atmel “CryptoMemory”
- Microchip “Keeloq”

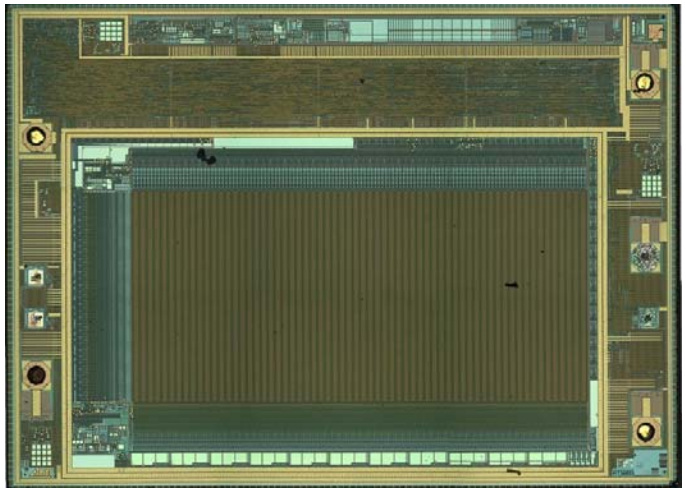
Atmel CryptoMemory

- Two common dies available- 350nm and 500nm
- Fuses determine which family member

Below: 500nm die (e.g. AT88SC0204)



Below: 350nm die (e.g. AT88SC25616C)



Atmel CryptoMemory Claims

- Master (Write7) password is only readable once it has been presented.
- There is a try limit and once it reaches zero, the part is forever locked from changes to its configuration memory.
- OTP Fuses protect the configuration memory.

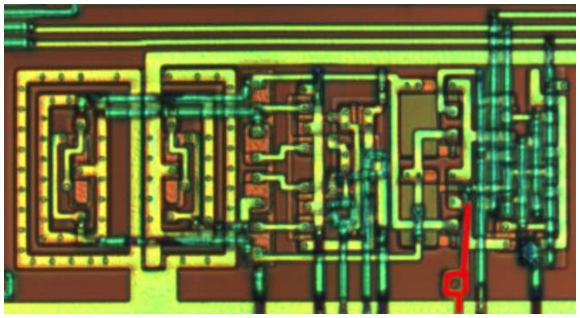
Write7 Password

- Address bus attack allows read back of the Write7 password in the clear.
- Databus attack allows read back of Write7 password after 64 samples have been taken.

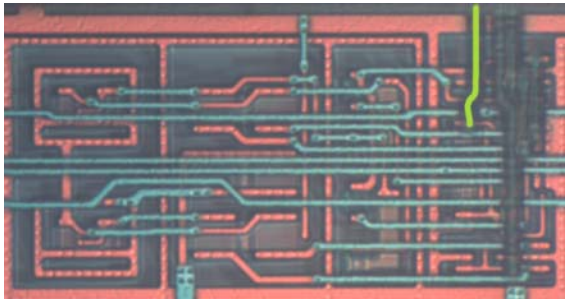
?OTP? Fuse Protection

- Fuses are “resettable” to an unprogrammed state via UV light.
- Watch out for “booby-trap” fuse! If set, part will no longer communicate.

Below: 500nm FUSE – Output in RED



Below: 350nm FUSE – Output in GREEN



More CryptoMemory issues

- Contents contained in “user memory” is stored in the clear (a commonly found problem).
- Exposure of the fuses to UV allows reset allowing changes to config memory if write7 password is known.

User Memory stored in the clear

- Configuration memory “rules” determine if readout of an area requires Crypto.

- A successful attack means:
  - Reset “OTP Perm” fuse to a ‘1’.
  - Learn Write7 password.
  - Apply Write7 password and clear Crypto requirements.
  - Readout memory in the **CLEAR**!!!!

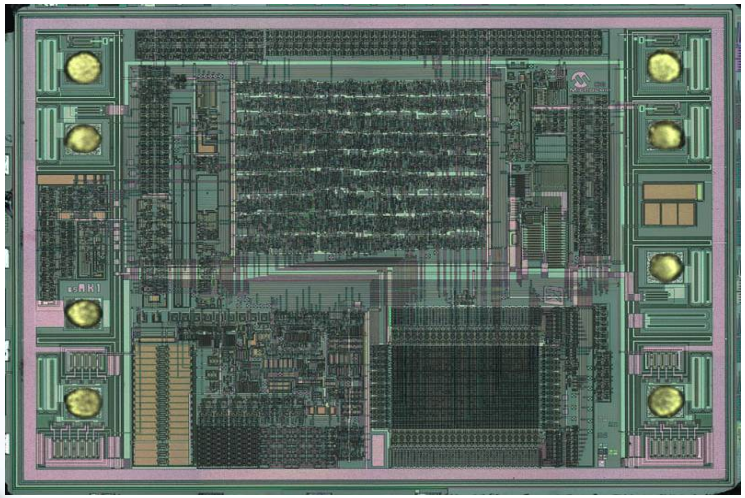
Microchip Keeloq [HCS201..362]

- Used around the globe in products such as:
  - Keyless entry on vehicles
  - Garage door openers (Genie)
  - Identity tokens
  - Burglar alarms

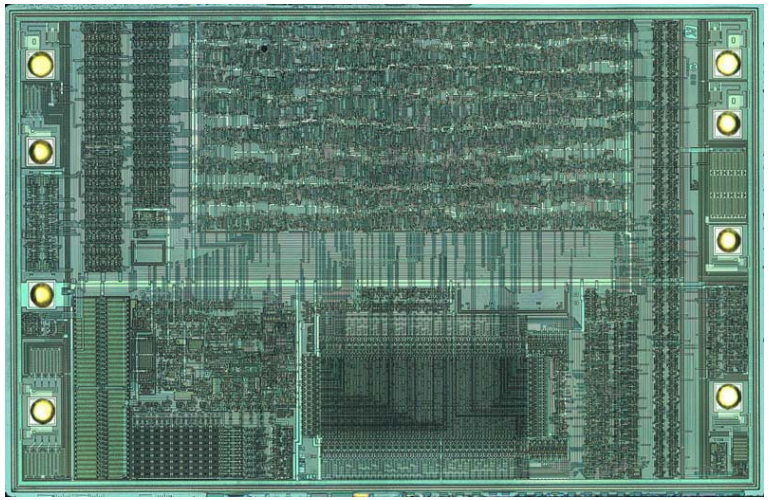
Some are ASICs

- Devices such as HCS201, 300, and 362 are ASICs designed as small state-machines with micro-coded ROM for behavior

Below: HCS201



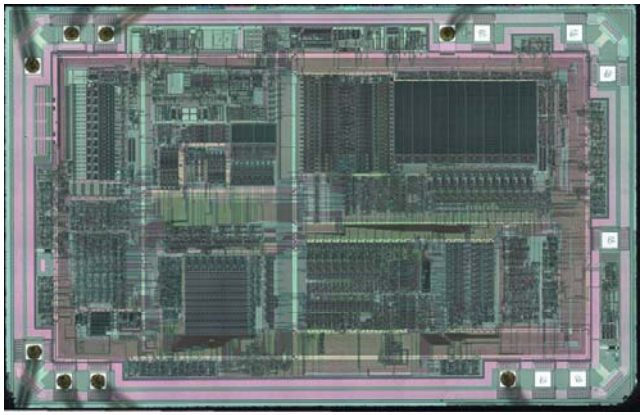
Below: HCS362



And some are not!!!

- Products such as HCS512-515 are actually PIC MCU's with EEPROM!!

Below: Ford keyless entry remote is actually 14-Pin PIC MCU bonded out as an 8 pin SOIC part. EEPROM is self-contained on the substrate.



HCSxxx simple to extract secrets

- Programming documentation claims device will auto-erase previous secrets.
- Only then can you program new secrets.
- Verification of newly programmed secrets can only be done ONCE.

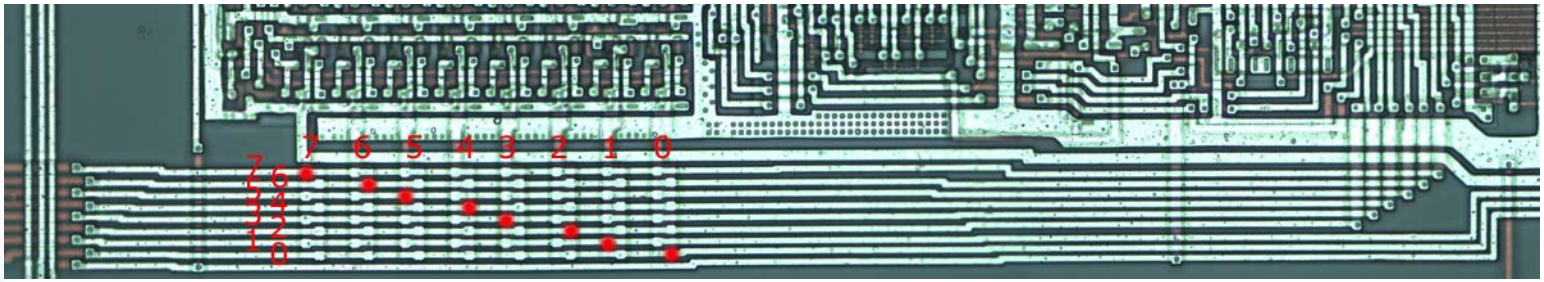
What if bulk-erase didn't occur?

- Microchip forgot something. How about checking if the memory really erased itself!

- The theory behind this is too:
  - Mess up bulk-erase
  - Send in static 00's or FF's (201 or 362?)
  - **Read back original data that was NOT erased!!!!!**

Motorola SC27/28 Smartcard MCU

- Used heavily in GSM (SC28 mostly)
- 6805 Core
- 12.8 KB Masked ROM, 240 Bytes SRAM, 8 KB of EEPROM
- Nothing special inside-
  - Sit on bus anywhere inside and you can see what's going on.
  - Bus ordering was: cpu_latch[7:0] = dbus[7,6,5,4,3,2,1,0];
  - Glitchable: Optically and Electrically



Motorola SC49 Smartcard MCU

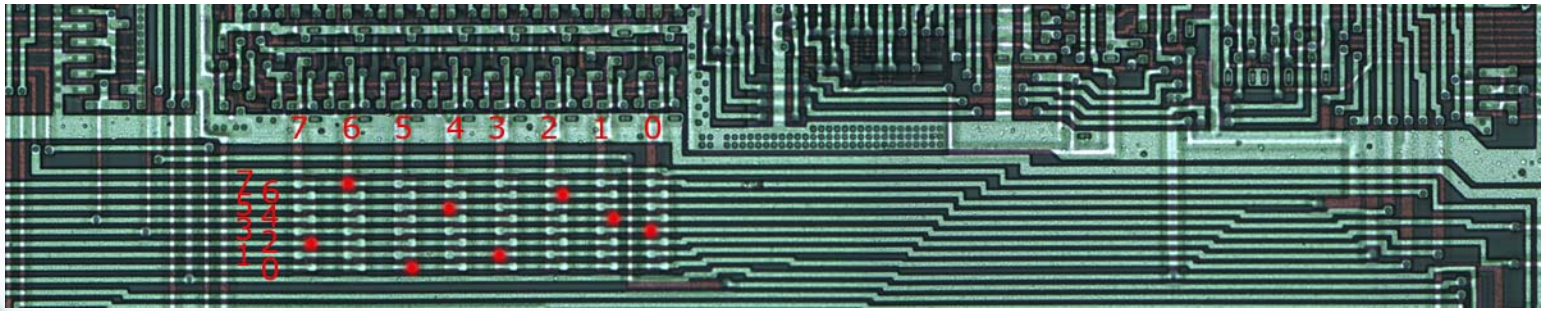
- Tried out in GSM SIM cards sometime in late 90's
- 6805 Core
- Hardware Cryptographic engine
- 11.3KB Masked ROM, 512 Bytes of SRAM, 4 KB of EEPROM
- Scrambled databus to confuse an attacker
  - Operands remain the same
  - Instructions needed be bit swapped
  - An eavesdropper needs to understand the core implementation.

Scrambling the bus? Why?

- Typical areas of probing are
  - Memory bus drivers.
  - Data bus itself where lines are organized in proper CPU bus width.
  - Bus lines are 99.9% of the time in order (0..7 or 7..0) and rarely swapped around!
  - Swapping the outputs of the memory is too easy to spot.

Implementation: Scrambled Bus

- As show in the photo below. Databus runs across the picture and is laid out from top to bottom as D7-D0.
- As shown by the red dots, connections into the instruction latches swap the lines to the properly decoded state for a 6805.
- Bit swap order is: cpu_latch[7:0] = dbus[6,2,4,1,0,7,3,5];
- Databus continues into the ALU to the right like other 6805's.



Infineon SLE66C160S/SLE66C320S

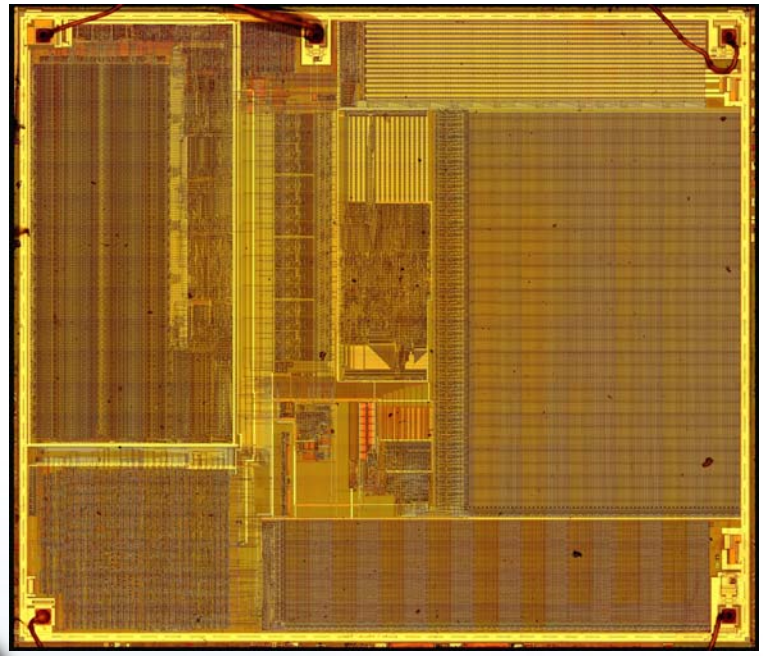
- Found to be used in-
  - GSM SIM cards (32 KB version)
  - Gemplus GEMSAFE (16 KB w/Crypto)

- Infineon quick spec states:
  - Security optimized layout and layout scrambling
  - Irreversible Lock - Out of test mode
  - Non standard dedicated Smart Card CPU–Core

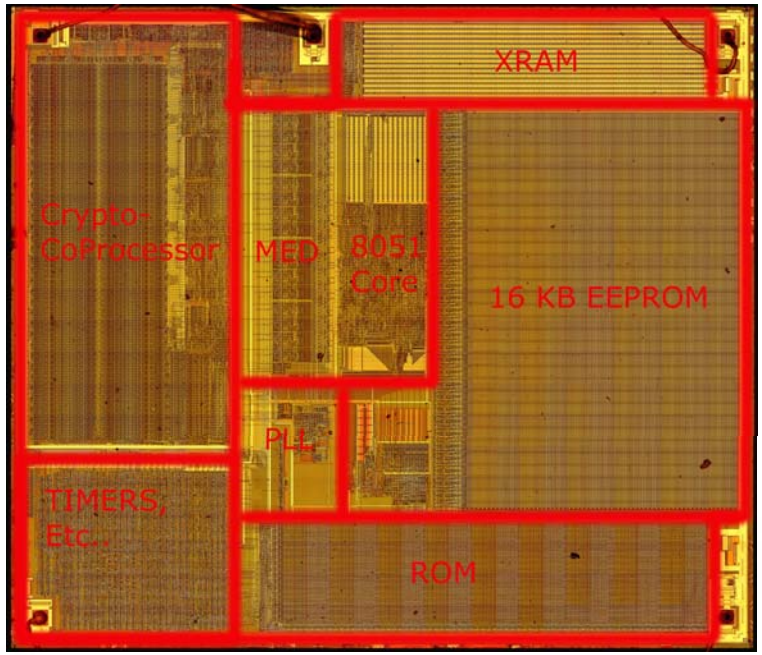
  - Above statements taken from Infineon “Short Product Info., 10.01, SLE 66C160S” (Page 3)

Infineon SLE66 “S” Die Image

Below: Uncommented 100x image

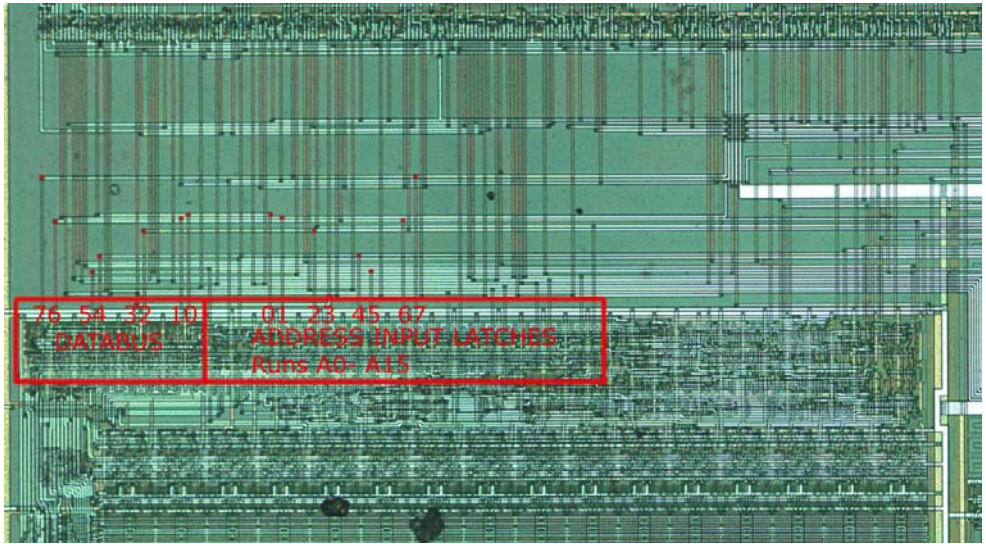


Below: Commented 100x image



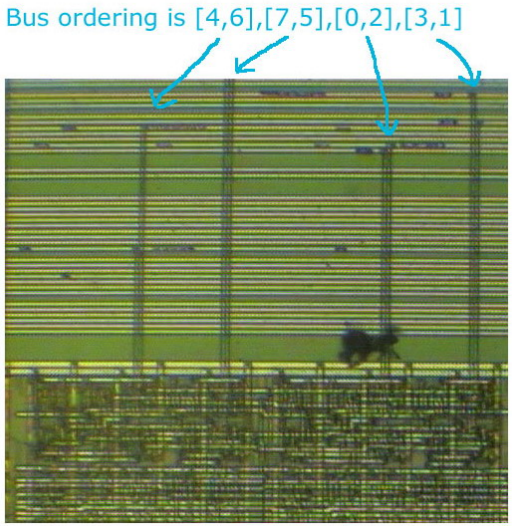
Infineon SLE66 “S” ROM

- ROM Databus output and Address input latches.
- Lower 8 bits of Address is multiplexed (shared) with Databus.
- No scrambling on ROM outputs nor address inputs!!



Infineon SLE66 “S” Main Databus

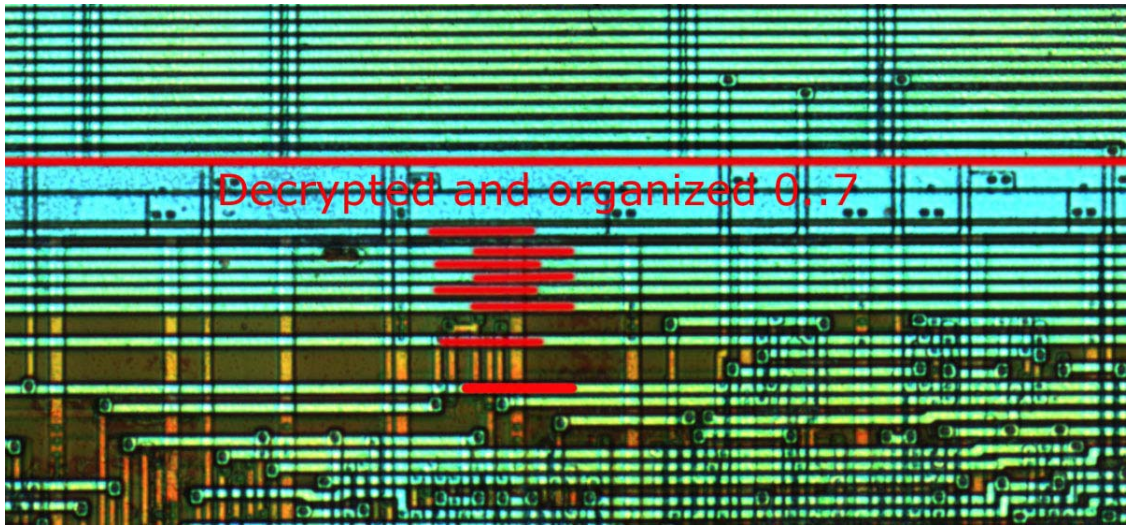
- “Security optimized layout and layout scrambling”
- ? Where ? We got here from the ROM outputs…



Infineon SLE66 “S” Core Databus

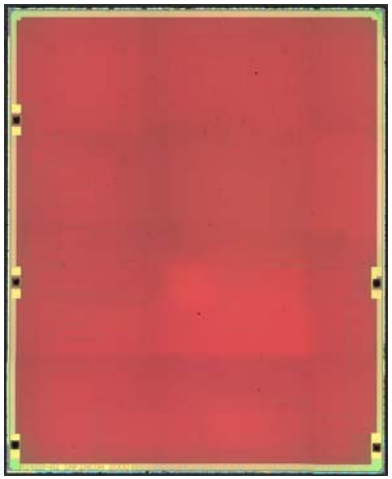
- Below the horizontal solid red line is the CLEAR databus.
- Ordering of the bits is 0,1,2,3,4,5,6,7 and any encryption of the fetch has been decrypted by the MED above out of view.

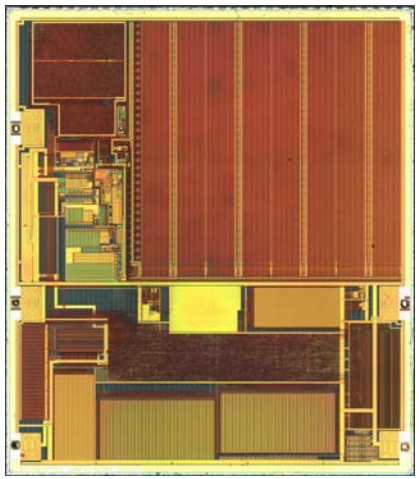
Below: Short red stripes represent clear databus bits 0..7

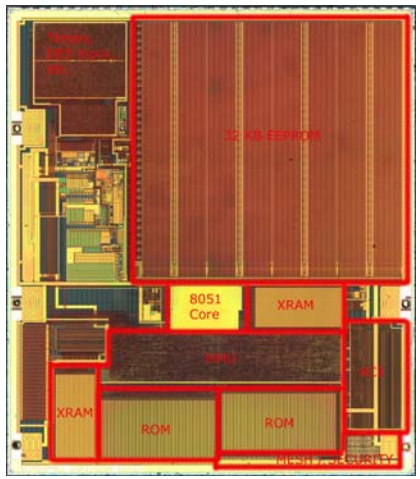


Infineon SLE66CX322P

- Found in GSM SIM cards
- 32 KB EEPROM
- Advanced Crypto Engine (ACE)

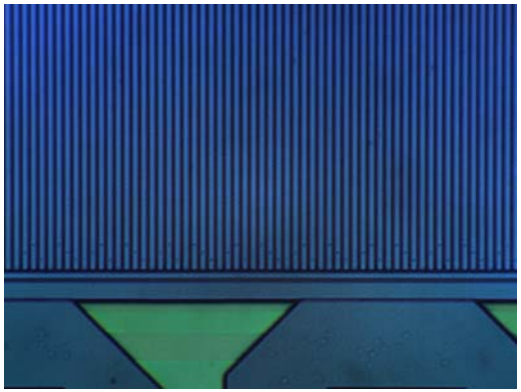


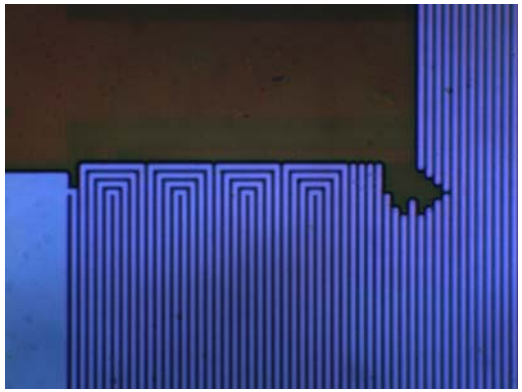


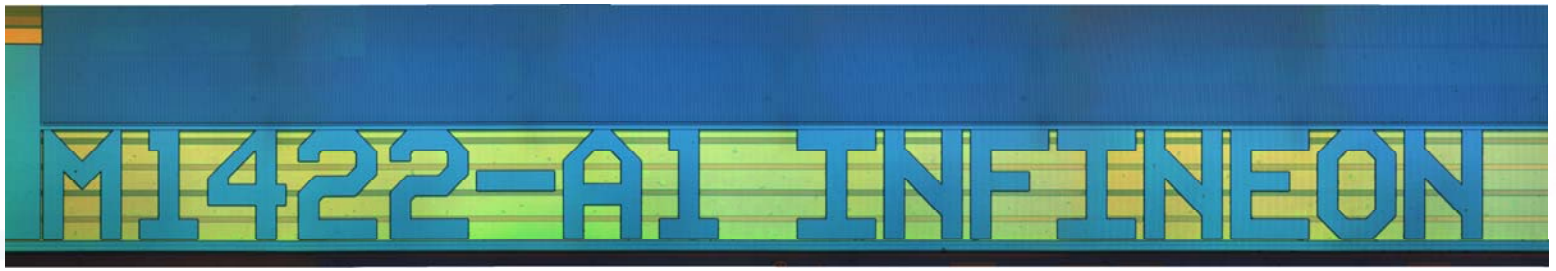


Infineon SLE66 “P” Secure?

- 4 conductor “active” mesh as top metal
- Began in 220nm 3+1 metal process

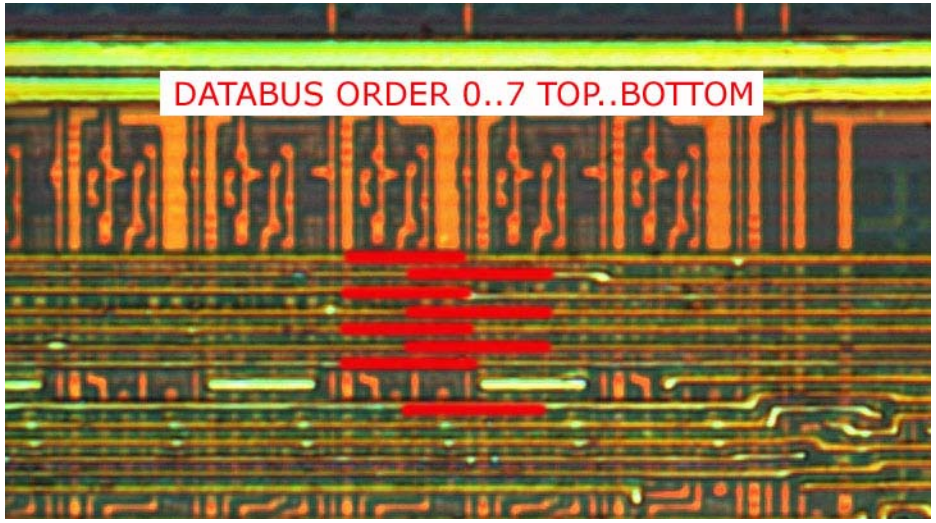






Infineon SLE66 “P” Databus

- Below the horizontal solid red line is the CLEAR databus.
- Ordering of the bits is 0,1,2,3,4,5,6,7.
- Opcode **must** be decrypted at this state in time!



ST Series Smartcards

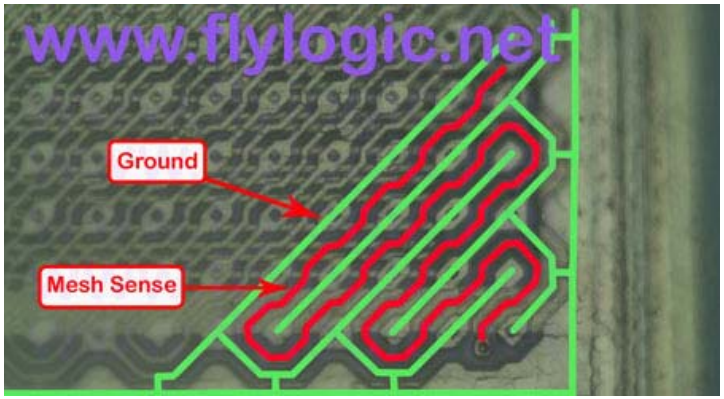
- ST16CF54: Crypto engine, 4 KB EEP
- ST16SF4x: No Crypto, 1-16 KB EEP
- ST19CF68: Crypto engine, 8 KB EEP
- ST19AF08: 20 pin SOIC, 8 KB EEP
- Enhanced 6805 MCU
- Pioneer of the “Mesh” principle

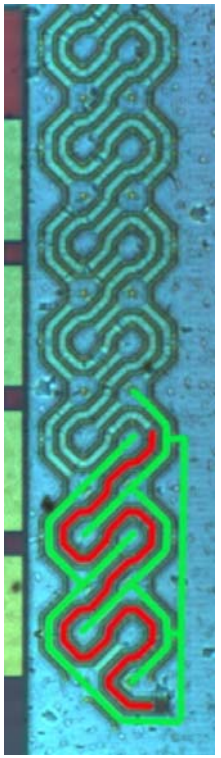
ST Mesh's

- 1st gen: Ground plane with holes (checker-board pattern)
  - » Opening is okay without device knowing
- Generations 2-4 are all “Serpentine” active sense with ground fingers
- 2nd gen: Mesh break results in stopped CPU
  - » Active sense is tied to VDD of the device
- 3rd gen: Mesh break results in BULK erase of EEPROM
  - » Active sense is tied to VDD of the device
- 4th gen: Mesh break results in BULK erase of EEPROM
  - » Active sense is a circuit now coming from opposite side of the device.

ST Mesh Images

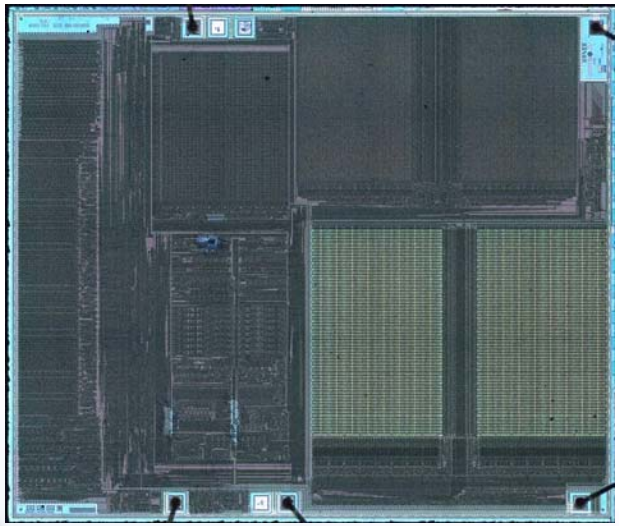
Gen 1 – 4 Meshes

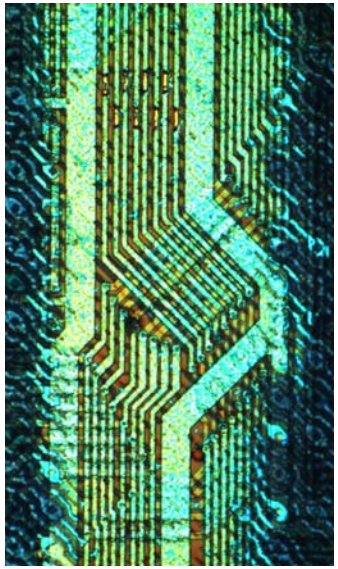


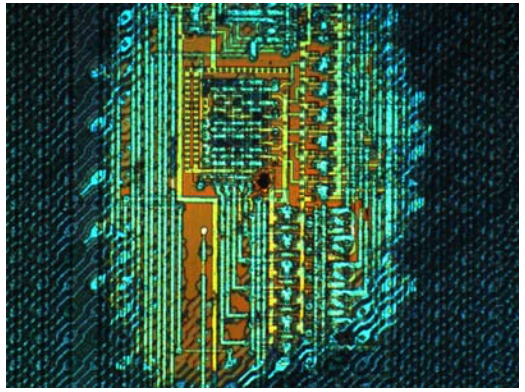


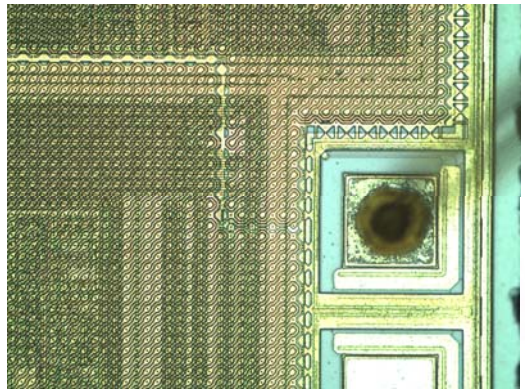
ST16XYZ Series

- Crypto engine available on ST16CF54A/B
- 1/2/4/8/16 KB EEPROM
- Customizable access rules aka firewall
- Filtered clock



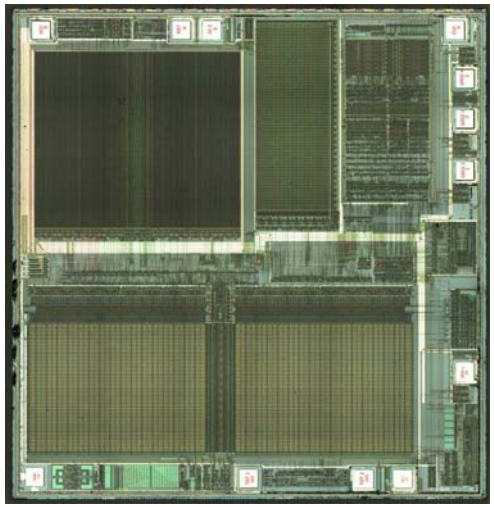


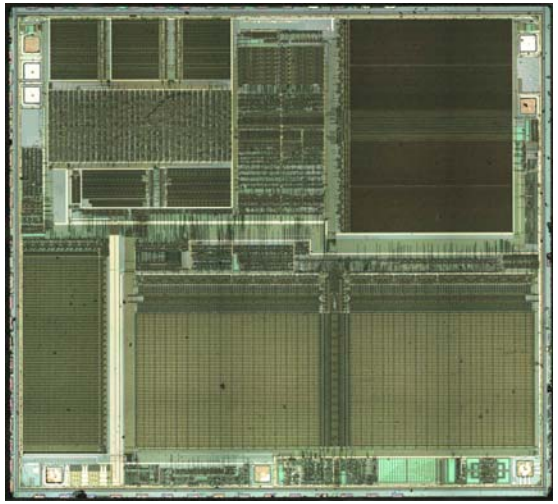




ST19XYZ Die Images

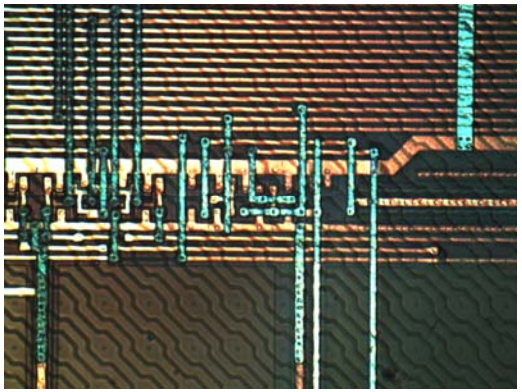
- Began in 600nm 2+1 metal process
- 10-12 MHz internal frequency (VDD dependent)

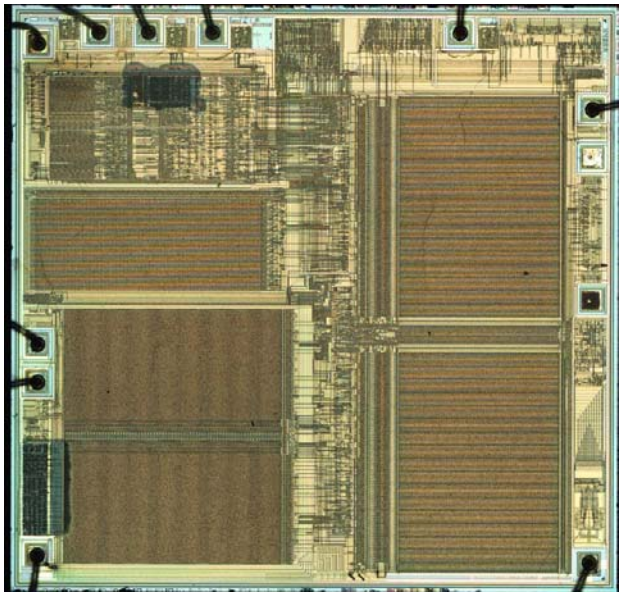


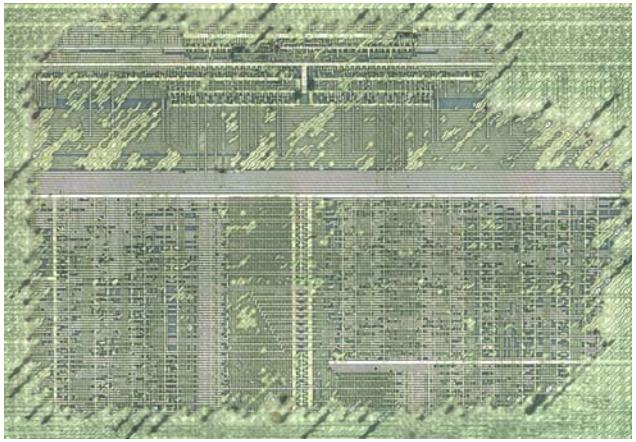


ST19XYZ Series

- Has anything really changed?
- No better than the older ST16 series





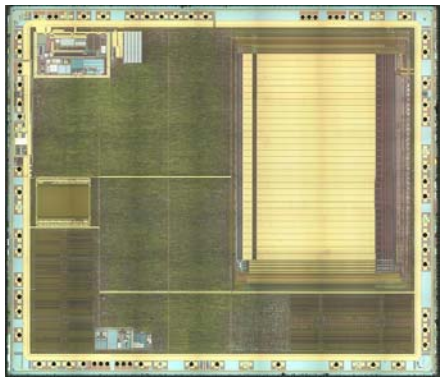


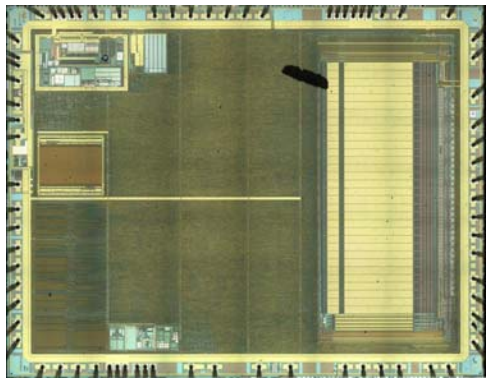
Smartcard vs. Off-The-Shelf (OTS) Devices

- Main differences:
  - Masked-ROM present with no type of common boot-loader
  - Top layer meshes present on many new devices
  - Isolation from outside world interference (UART, PLL, …)
  - Uniqueness per die
- OTS devices are very commonly used as well
  - Some are stronger than others
- Commonly used in USB dongles and other security tokens
  - Atmel AT90xxxUSB
  - Cypress CY7C63xxx
  - Microchip PIC18Fxxx

Atmel AT90xxxUSB

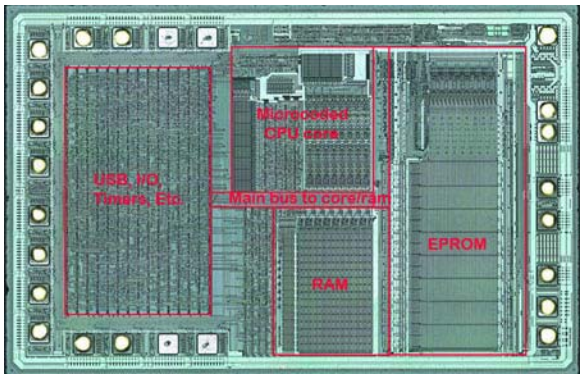
- The most secure of the available OTS choices.
- Fuses are buried in 350nm 3 metal technology
- UV sets the fuses to an undesired state
- AVR executes on a 1:1 clock frequency
- SRAM is cleared on reset in some devices





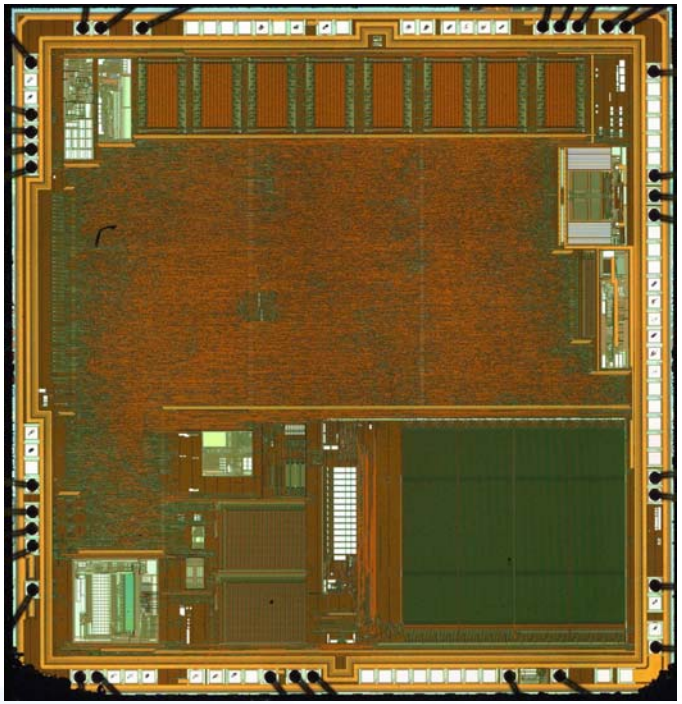
Cypress

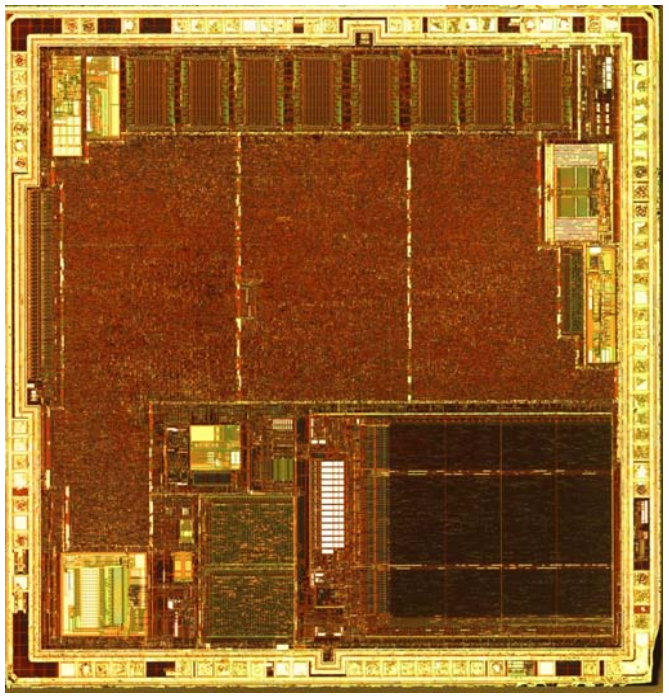
- Probably the most used USB Dongle MCU around!
  - » Has been seen used in keyboards as well
- OTP??? Oh really… What about UV light??
- Single fuse for protection user code.
- The most insecure MCU I have seen used in Dongles



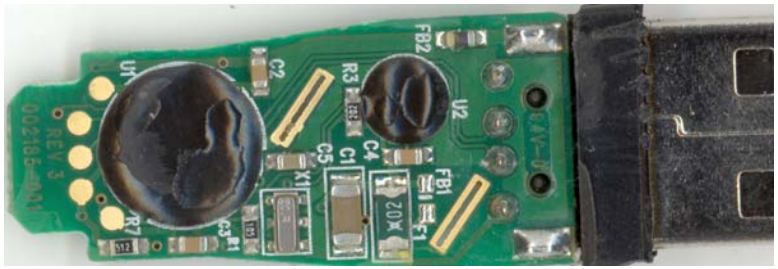
Microchip PIC18Fxxx

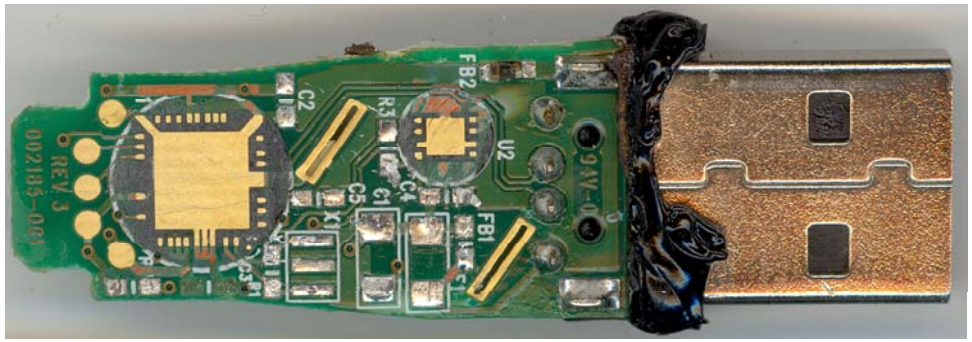
- Found in latest Safe-Net, “Sentinel USB Key”

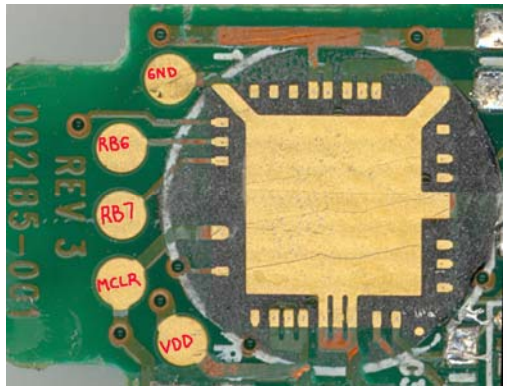




Safenet Sentinel USB Key







Do you trust it?

- http://www.safenet-inc.com/Library/3/sentinelhardwarekeys.pdf
- Page 1 says:
  - **“The Most Secure Hardware Token In The World.**”
  - A unique encryption key is used for every communication session between the application and the hardware token, making brute force attacks virtually impossible. **In addition, the keys include internal authentication, which effectively prevents cloning of the keys**.

No Cloning of the Keys?

- Pictures of the dongle show five (5) test pads on the end of the dongle (VDD, GND, MCLR, RB6, RB7).
- These are used program and serialize the device.
- Once the contents of a device has been extracted, the image of that part and external EEPROM can be cloned into new fresh dongles (or a homebrew prototype PCB).
- Steps required-
  1. Download small boot loader to allow programming of external EEP.
  2. Upon completion, erase flash of PIC and reload with proper image that correlates to EEPROM image loaded in step 1.
- 100% Clone is possible!

“Password” Boot-Strap-Loader (BSL)

- In two words: **STAY AWAY**

- Easy to circumvent
- Technique is becoming very popular
- Force flash reads to static value
- Works an many major manufacturers

- Manufacturer's exaggerate password attack to 2^n
  - This is simply not true.

TSMC

- Many devices today are being fab'd by TSMC.
- Common flash layout is being seen by major manufacturers.
- Understanding flash behavior of one = same as the others.
- Hacker's life is made easier thanks to the common cell library.

Poly-Silicon Fuses

- Before trusting them, learn how they have been implemented.
- Blown fuses are very easy to find (leaves residue)!
- Typically easy to jump the fuse with a single needle.

Blown fuse

Good fuse

In Conclusion

- Most things are not as they seem
- Technology is improving but is not perfect
- Every standard secure IC made to date has been successfully compromised by hackers
- What is made by human can be taken apart by human