Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer

Roel Verdult **Baris Ege**

Radboud University Nijmegen,
The Netherlands

Flavio D. Garcia

School of Computer Science,
University of Birmingham, UK.

Institute for Computing and Information Sciences
Radboud University Nijmegen, The Netherlands
✉*rverdult@cs.ru.nl* 🖱*www.cs.ru.nl/~rverdult*

Disclaimer

- Due to a recent injunction by the High Court of London this talk cannot cover the technical core of the accepted paper

We are responsible for the content of this presentation and any opinions expressed during the presentation are ours and do not necessarily represent the views of the University of Birmingham or Dr Garcia

Interaction

- We stick to these slides and will not answer questions
  (sorry about this; we hope you understand)
- However, we are happy to see discussion about these issues in the community
- Since legal procedures are still ongoing, it is not appropriate to go into details

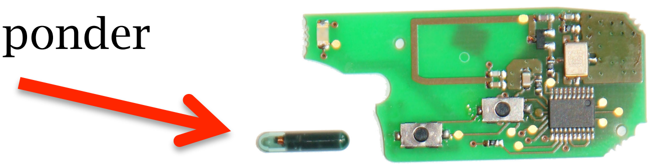
Contents

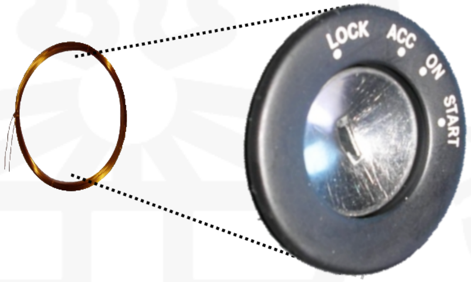
- Introduction
  - Vehicle Immobilizers
- Related Work
  - Related work on vehicle immobilizer systems
  - Weaknesses in Hitag2 cryptography
  - Responsible disclosure
  - Reverse Engineering of security mechanisms
- Mitigation and alternatives
  - Proposed in the academic literature
  - Introduced by the industry
- Conclusion

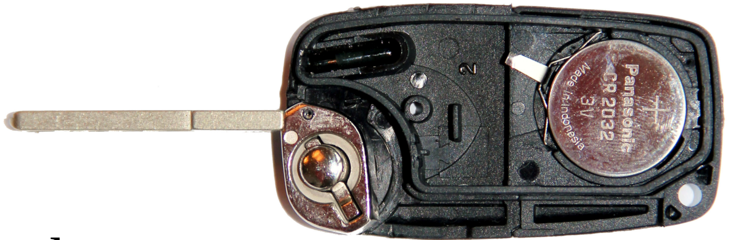
Vehicle Immobilizers

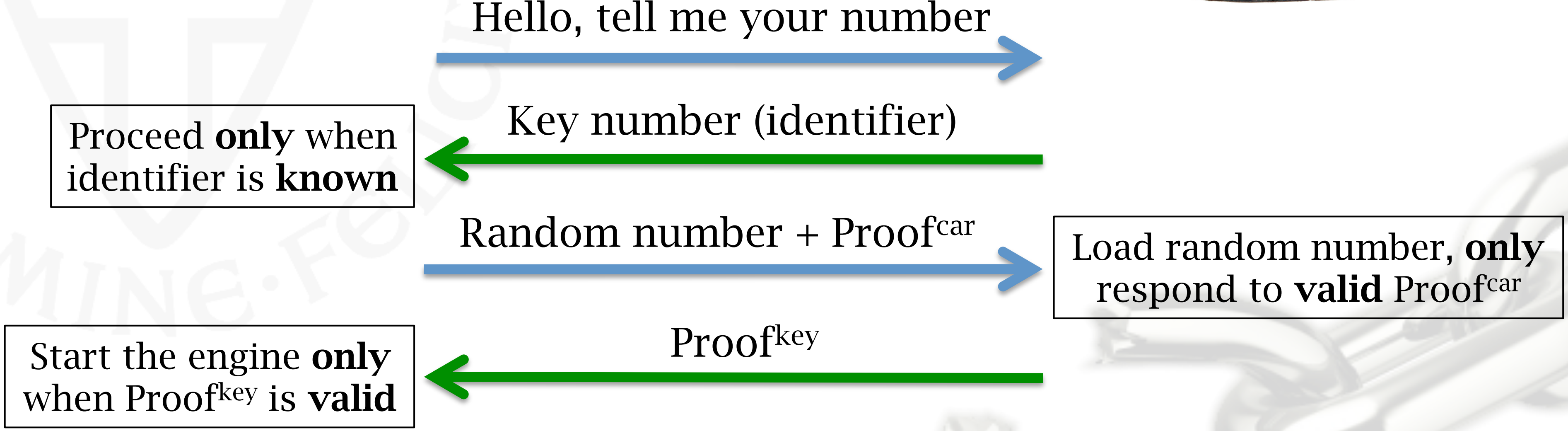
- Electronic anti-theft device
- Introduced in the '90s
- Prevents hot-wiring (like in the movies)
- Mandatory in many countries eg.
  - Europe (EU Directive 95/56/EC)
  - Australia (AS/NZS 4601:1999)
  - Canada (CAN/ULC S338-98)
- Passive RFID transponder (125 KHz)
- **Not** to be confused with remote controls that unlock the car doors (433/868 MHz)

Vehicle Immobilizer Example









Hello, tell me your number

Key number (identifier)

Proceed **only** when identifier is **known**

Random number + $\text{Proof}^{\text{car}}$

Load random number, **only** respond to **valid** $\text{Proof}^{\text{car}}$

$\text{Proof}^{\text{key}}$

Start the engine **only** when $\text{Proof}^{\text{key}}$ is **valid**

Related Work (DST)

- Digital Signature Transponder (DST)
  - Introduced in 1995
  - Key length of 40-bits
  - Complexity 2^{40} encryptions (exhaustive search)
- Publications
  - Stephen C. Bono, Matthew Green, Adam Stubblefield, Ari Juels, Aviel D. Rubin, and Michael Szydlo. **Security analysis of a cryptographically enabled RFID device**. In 14th USENIX Security Symposium (USENIX Security 2005), pages 1–16. USENIX Association.
- **Replaced by DST80 or TRPWS21 (AES 128-Bit)**

Related Work (KeeLoq)

- KeeLoq
  - Introduced in 1996
  - Key length of 64-bits
  - Complexity 2^{44} encryptions (slide / meet-in-the-middle)
- Publications
  - Bogdanov, A**. Linear slide attacks on the KeeLoq block cipher.** In 3rd International Conference on Information Security and Cryptology (IN- SCRYPT 2007), vol. 4990 of Lecture Notes in Computer Science, Springer, pp. 66–80.
  - Courtois, N. T., Bard, G. V., and Wagner, D. **Algebraic and slide attacks on KeeLoq.** In 15th International Workshop on Fast Software Encryption (FSE 2008), vol. 5086 of Lecture Notes in Computer Science, Springer-Verlag, pp. 97–115.
  - Indesteege, S., Keller, N., Dunkelmann, O., Biham, E., and Preneel, B. **A practical attack on KeeLoq.** In 27th International Conference on the Theory and Application of Cryptographic Techniques, Advances in Cryptology (EUROCRYPT 2008), vol. 4965 of Lecture Notes in Computer Science, Springer-Verlag, pp. 1–8.
  - Kasper, M., Kasper, T., Moradi, A., and Paar, C. **Breaking KeeLoq in a flash: on extracting keys at lightning speed.** In 2nd International Conference on Cryptology in Africa, Progress in Cryptology (AFRICACRYPT 2009), vol. 5580 of Lecture Notes in Computer Science, Springer-Verlag, pp. 403–420.
- **No longer deployed in the automotive industry**

Related Work (Hitag2)

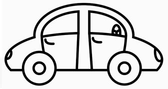
- Hitag2
  - Introduced in 1996
  - Key length of 48-bits
  - Complexity 2^{37} encryptions (cryptanalytic)
- Publications
  - Courtois, N. T., O'Neil, S., and Quisquater, J.-J. **Practical algebraic attacks on the Hitag2 stream cipher**. In 12th Information Security Conference (ISC 2009), vol. 5735 of Lecture Notes in Computer Science, Springer-Verlag, pp. 167–176.
  - Soos, M., Nohl, K., and Castelluccia, C. **Extending SAT solvers to cryptographic problems**. In 12th International Conference on Theory and Applications of Satisfiability Testing (SAT 2009), vol. 5584 of Lecture Notes in Computer Science, Springer-Verlag, pp. 244–257.
  - Sun, S., Hu, L., Xie, Y., and Zeng, X. **Cube cryptanalysis of Hitag2 stream cipher**. In 10th International Conference on Cryptology and Network Security (CANS 2011), vol. 7092 of Lecture Notes in Computer Science, Springer-Verlag, pp. 15–25.
  - Stembera, P., and Novotny, M. **Breaking Hitag2 with reconfigurable hardware**. In 14th Euromicro Conference on Digital System Design (DSD 2011), IEEE Computer Society, pp. 558–563.
  - Immler, V. Breaking hitag 2 revisited. Security, Privacy, and Applied Cryptography Engineering (SPACE 2012) 7644, 126–143.
  - Verdult, R., Garcia, F. D., and Balasch, J**. Gone in 360 seconds: Hijacking with Hitag2**. In 21st USENIX Security Symposium (USENIX Security 2012), USENIX Association, pp. 237–252.
- **Replaced by Hitag3 or Hitag-Pro (AES 128-Bit)**

Identified Hitag2 Weaknesses

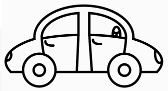
- Weak cryptographic algorithm
  - Cipher design is obsolete (from the early ’90s)
  - Weak and invertible cipher initialization
  - Security is significantly lower than the key size
- Implementation mistakes
  - There is no random number (freshness) introduced by the transponder during authentication
  - Secret key update is not one atomic operation
- Improper usage by car manufacturers
  - Many transponders are configured with the default (or easy to guess) passwords
  - Many cars are configured with weak secret keys which drastically speeds-up key recovery

Practicality of Hitag2 Attacks

1. Communicate with the genuine car-key
   - With maximum wireless distance of **two inches**

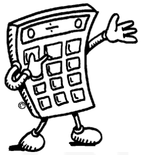


2. Bypass other security measures of the car
   - Force the door locks of the car
   - Disable the alarm (separate protection)
   - Force the ignition lock (hot-wire the car)

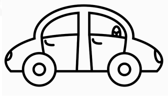


3. Eavesdrop immobilizer messages from the car

4. Communicate **again** with the car-key



5. Perform a complex mathematical computation to recover the secret cryptographic key



6. Emulate the car key and start the car

This is not really a very practical way to steal a car!

Megamos Crypto

- Megamos Crypto
  - Introduced in 1995/1996
  - Key length of 96-bits

- Security issues are comparable to the weaknesses we found in our Hitag2 study

- Practicality of attacks are similar to Hitag2

- **Still deployed in cars**

Responsible disclosure in general

- Notify "problem owner" early on
- Help to understand/solve issues
- Publish after a delay
  - CERT - Carnegie Mellon 45 days
    http://www.cert.org/kb/vul_disclosure.html
  - Google recommends 60 days
    http://googleonlinesecurity.blogspot.nl/2013/05/disclosure-timeline-for-vulnerabilities.html
  - Guidelines from the Dutch government
    http://www.rijksoverheid.nl/bestanden/documenten-en-publicaties/rapporten/2013/01/04/leidraad-om-te-komen-tot-een-praktijk-van-responsible-disclosure/lp-v-j-0000002385.pdf
    - 60 days for software
    - 6 months for hardware

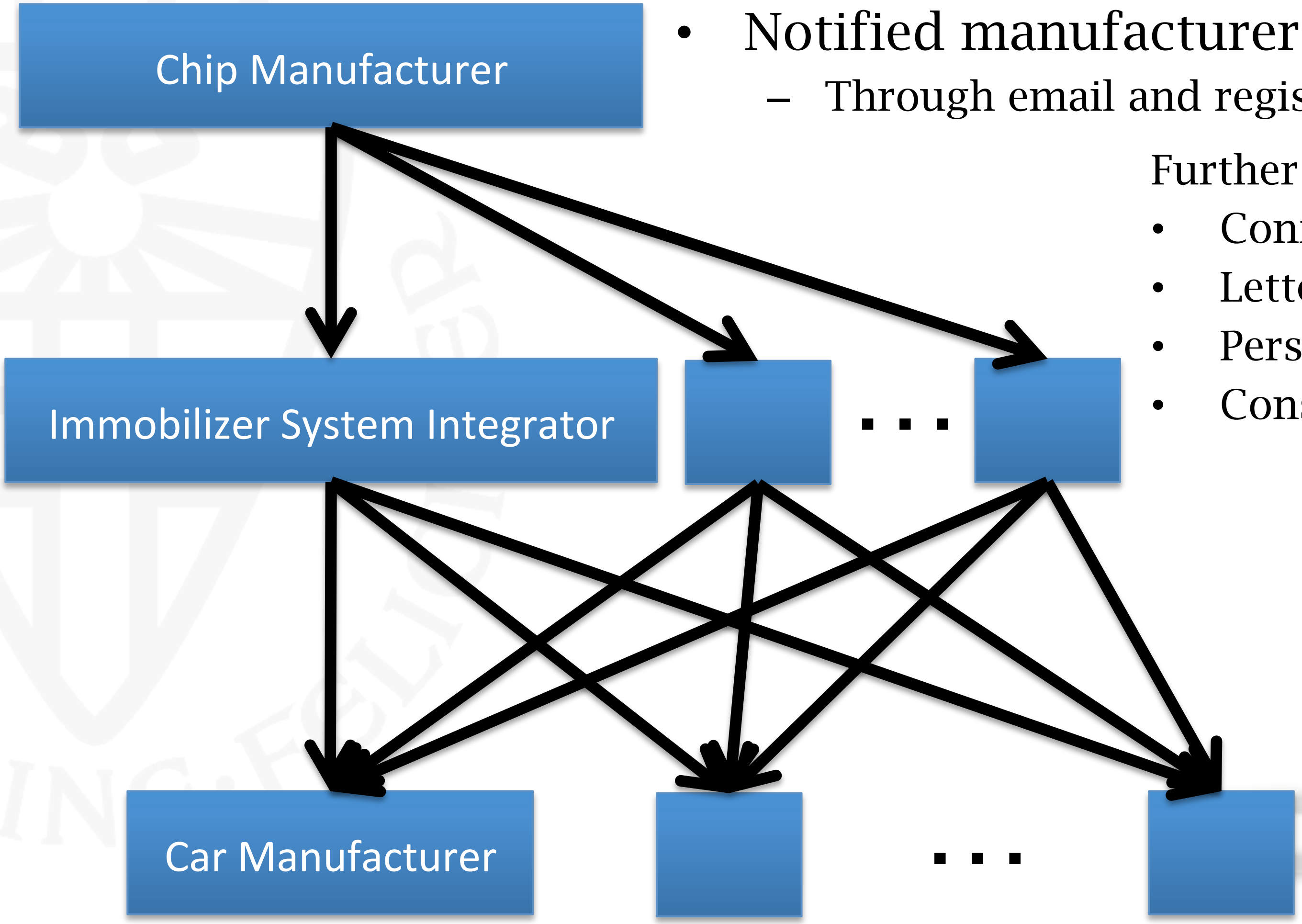
Nijmegen’s History

- Responsible disclosure
- NXP MIFARE Classic chip
  - contested in court by NXP, but they lost
  - published after 6 months
- Nokia NFC cell-phones after 6 months
- Atmel's crypto memory after 6 months
- NXP Hitag2 after 6 months
- **In these cases, the notified parties contacted us within 48 hours**

Notification Chains

- Previous notified parties explicitly demanded that **<u>they</u>** should communicate with their customers
- We should not talk to them directly
  - Such notification is their responsibility
  - They do not share customer information
  - They know which people to contact
- **This is what we have done since then**

Megamos Notification



- Notified manufacturer **twice** in 2012
  - Through email and registered mail

Further collaboration

- Conference call
- Letters and emails
- Personal meeting
- Constructive discussions

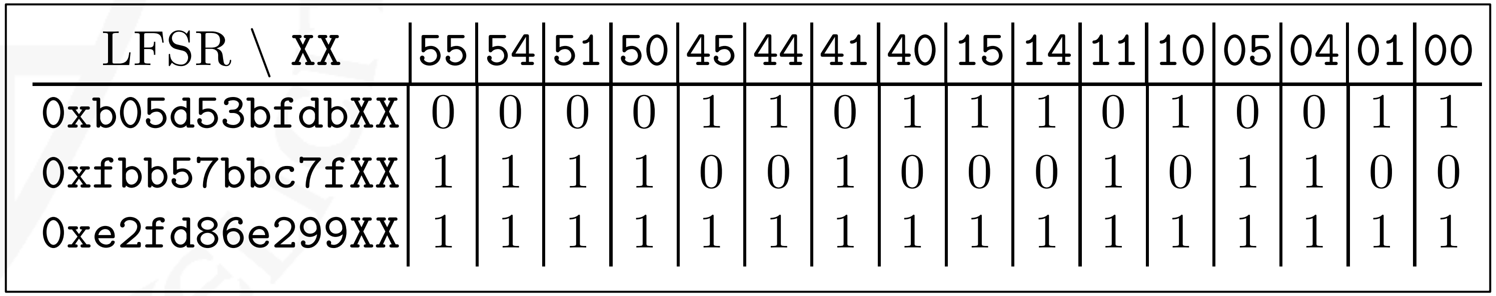
No contacts at this level

Reverse Engineering Methods

- Observing in/output (blackbox)
  - Non-invasive, can be performed remotely
- Decompile firmware (software)
  - Automatic tools are available to assist
- Chip slicing
  - Peeling away layers of a chip
  - Use microscope to shoot photos
- New approach: Side Channel Analysis for Reverse Engineering (SCARE)

Reverse Engineering – Observing in/output

- Publications that used blackbox reverse-engineering
  - Stephen C. Bono, Matthew Green, Adam Stubblefield, Ari Juels, Aviel D. Rubin, and Michael Szydlo. **Security analysis of a cryptographically enabled RFID device**. In 14th USENIX Security Symposium (USENIX Security 2005), pages 1–16. USENIX Association.
  - Flavio D. Garcia, Gerhard de Koning Gans, Ruben Muijrers, Peter van Rossum, Roel Verdult, Ronny Wichers Schreur, and Bart Jacobs. **Dismantling MIFARE Classic**. In *13th European Symposium on Research in Computer Security (ESORICS 2008)*, volume 5283 of *Lecture Notes in Computer Science*, pages 97–114. Springer-Verlag.

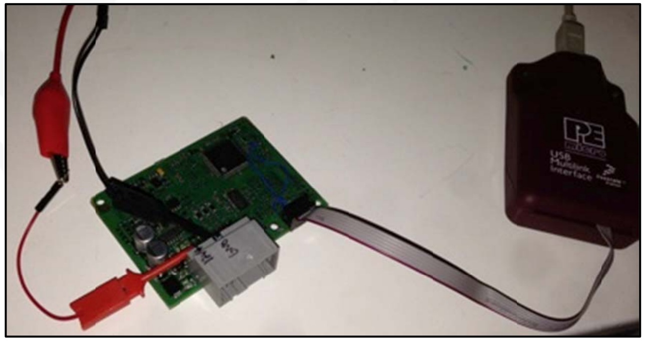


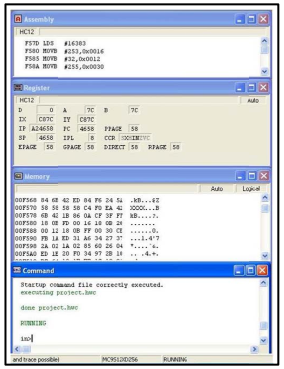
LFSR \ XX	55	54	51	50	45	44	41	40	15	14	11	10	05	04	01	00
0xb05d53bfdbXX	0	0	0	0	1	1	0	1	1	1	0	1	0	0	1	1
0xfbb57bbc7fXX	1	1	1	1	0	0	1	0	0	0	1	0	1	1	0	0
0xe2fd86e299XX	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

  - Stefan Lucks, Andreas Schuler, Erik Tews, Ralf-Philipp Weinmann, and Matthias Wenzel. **Attacks on the DECT authentication mechanisms.** In 9th Cryptographers' Track at the RSA Conference (CT-RSA 2009), volume 5473 of Lecture Notes in Computer Science, pages 48–65. Springer-Verlag, 2009.

Reverse Engineering – Firmware (software)

- Publications about reversing security algorithms
  - Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, Stefan Savage, Karl Koscher, Alexei Czeskis, Franziska Roesner, and Tadayoshi Kohno. **Comprehensive experimental analyses of automotive attack surfaces.** In 20th USENIX Security Symposium (USENIX Security 2011), pages 77–92. USENIX Association, 2011.
  - Benedikt Driessen, Ralf Hund, CarstenWillems, Carsten Paar, and Thorsten Holz. **Don’t trust satellite phones: A security analysis of two satphone standards.** In 33rd IEEE Symposium on Security and Privacy (S&P 2012), pages 128–142. IEEE Computer Society, 2012.
- Miller & Valasek at DEFCON 2013 (funded by DARPA)
  - http://illmatics.com/car_hacking.pdf





```
F57D LDS   #16383
F580 MOVB  #253,0x0016
F585 MOVB  #32,0x0012
F58A MOVB  #255,0x0030
```

```
D      0   A      7C   B       7C
IX  C87C   IY   C87C
IP A24658  PC   4658   PPAGE   58
SP  4658   IPL     8   CCR     SXHINZVC
EPAGE 58   GPAGE  58   DIRECT  58   RPAGE  58
```

```
00F568 84 6B 42 ED 84 F6 24 5A  .kB...$Z
00F570 58 58 58 58 C4 F0 EA 42  XXXX...B
00F578 6B 42 1B 86 0A CF 3F FF  kB....?.
00F580 18 0B FD 00 16 18 0B 20  .......
00F588 00 12 18 0B FF 00 30 CE  ......0.
00F590 FB 1A ED 31 A6 34 27 37  ...1.4'7
00F598 2A 02 1A 02 85 60 26 04  *....`&.
00F5A0 ED 1E 20 F0 34 97 2B 10  .. .4.+.
```

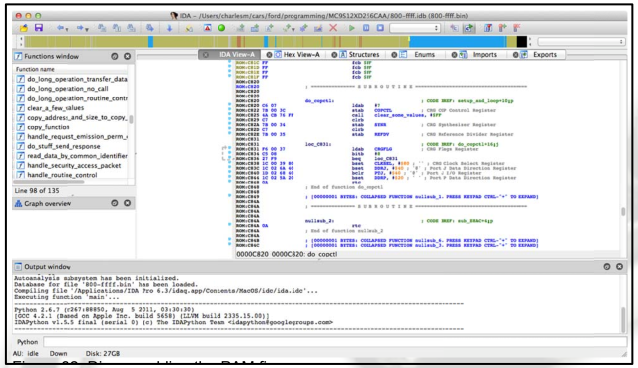
```
Startup command file correctly executed.
executing project.hwc

done project.hwc

RUNNING

in>
```

and trace possible)   MC9S12XD256   RUNNING



Function name
- do_long_operation_transfer_data
- do_long_operation_no_call
- do_long_operation_routine_contr
- clear_a_few_values
- copy_address_and_size_to_copy_
- copy_function
- handle_request_emission_perm_
- do_stuff_send_response
- read_data_by_common_identifier
- handle_security_access_packet
- handle_routine_control

Line 98 of 135

```
ROM:C81C FF                  fcb $FF
ROM:C81D FF                  fcb $FF
ROM:C81E FF                  fcb $FF
ROM:C81F FF                  fcb $FF
ROM:C820
ROM:C820        ; =============== S U B R O U T I N E =======================================
ROM:C820
ROM:C820
ROM:C820        do_copctl:                         ; CODE XREF: setup_and_loop+10↓p
ROM:C820 C6 07               ldab  #7
ROM:C822 7B 00 3C            stab  COPCTL          ; CRG COP Control Register
ROM:C825 4A CB 76 FF         call  clear_some_values, #$FF
ROM:C829 C7                  clrb
ROM:C82A 7B 00 34            stab  SYNR            ; CRG Synthesizer Register
ROM:C82D C7                  clrb
ROM:C82E 7B 00 35            stab  REFDV           ; CRG Reference Divider Register
ROM:C831
ROM:C831        loc_C831:                          ; CODE XREF: do_copctl+16↓j
ROM:C831 F6 00 37            ldab  CRGFLG          ; CRG Flags Register
ROM:C834 C5 08               bitb  #8
ROM:C836 27 F9               beq   loc_C831
ROM:C838 1C 00 39 80         bset  CLKSEL, #$80 ; '' ; CRG Clock Select Register
ROM:C83C 1C 02 6A 40         bset  DDRJ, #$40 ; '@' ; Port J Data Direction Register
ROM:C840 1D 02 68 40         bclr  PTJ, #$40 ; '@' ; Port J I/O Register
ROM:C844 1C 02 5A 20         bset  DDRP, #$20 ; ' ' ; Port P Data Direction Register
ROM:C848 0A                  rtc
ROM:C848        ; End of function do_copctl
ROM:C848
ROM:C849        ; [00000001 BYTES: COLLAPSED FUNCTION nullsub_1. PRESS KEYPAD CTRL-"+" TO EXPAND]
ROM:C84A
ROM:C84A        ; =============== S U B R O U T I N E =======================================
ROM:C84A
ROM:C84A
ROM:C84A        nullsub_2:                         ; CODE XREF: sub_E8AC+4↓p
ROM:C84A 0A                  rtc
ROM:C84A        ; End of function nullsub_2
ROM:C84A
ROM:C84B        ; [00000001 BYTES: COLLAPSED FUNCTION nullsub_6. PRESS KEYPAD CTRL-"+" TO EXPAND]
ROM:C84C        ; [00000001 BYTES: COLLAPSED FUNCTION nullsub_3. PRESS KEYPAD CTRL-"+" TO EXPAND]
```

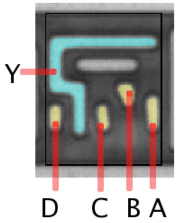
0000C820 0000C820: do copctl

```
Autoanalysis subsystem has been initialized.
Database for file '800-ffff.bin' has been loaded.
Compiling file '/Applications/IDA Pro 6.3/idaq.app/Contents/MacOS/idc/ida.idc'...
Executing function 'main'...
---------------------------------------------------------------------------------------
Python 2.6.7 (r267:88850, Aug  5 2011, 03:30:30)
[GCC 4.2.1 (Based on Apple Inc. build 5658) (LLVM build 2335.15.00)]
IDAPython v1.5.5 final (serial 0) (c) The IDAPython Team <idapython@googlegroups.com>
---------------------------------------------------------------------------------------
```

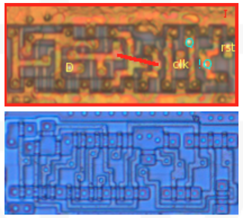
AU: idle   Down   Disk: 27GB

Reverse Engineering – Chip slicing

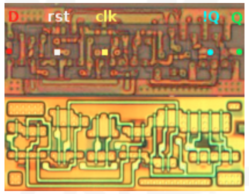
- Publications about chip slicing



  - Karsten Nohl, David Evans, Starbug, and Henryk Plotz. **Reverse engineering a cryptographic RFID tag.** In 17th USENIX Security Symposium (USENIX Security 2008), pages 185–193. USENIX Association, 2008.



  - Karsten Nohl, Erik Tews, and Ralf-Philipp Weinmann. **Cryptanalysis of the DECT standard cipher.** In 17th International Workshop on Fast Software Encryption (FSE 2010), volume 6147 of Lecture Notes in Computer Science, pages 1–18. Springer-Verlag, 2010.



  - Henryk Plotz and Karsten Nohl. **Peeling away layers of an RFID security system.** In 16th International Conference on Financial Cryptography and Data Security (FC 2012), volume 7035 of Lecture Notes in Computer Science, pages 205–219. Springer-Verlag, 2012.

- Pictures are published at http://www.siliconzoo.org

Access to Megamos Crypto

- We used Tango Programmer, whose software is publicly available
  - Contains Megamos Crypto algorithm obfuscated
  - Called “murky” by the judge
  - Available since 2009 and still being distributed
- There seem to be many other sources available which contain the algorithm and are
  - easy accessible
  - **not** protected
  - **not** encrypted
  - **not** obfuscated

Megamos Crypto partly published

- Challenges in Security Engineering event
  - CSE 2012 3-7 Sept. 2012 Bochum, Germany
  - ECRYPT II Summer School
  - European Network of Excellence for Cryptology II
    - Funded by European Commission's Seventh Framework Programme (FP7)
- Jan Krissler (Starbug) presented at CSE 2012
  - Reverse engineering Megamos Crypto via chip slicing
  - Hi-resolution pictures of the hardware logic gates
  - Schematic of a core component of Megamos Crypto
  - Slides are publicly available for download

Mitigation and Alternatives

- Proposed in the academic literature
  - Authentication protocols
  - Key derivation schemes
- Products introduced by the industry
  - Proprietary chips (Hitag3 and DST80)
  - Vehicle immobilizer transponders based on the Advanced Encryption Standard (AES)
    - HITAG Pro, NXP Semiconductors (2007)
    - ATA5580, Atmel Corporation (2010)
    - TRPWS21/TRPBS27, Texas Instruments (2010)

Proposed in the academic literature

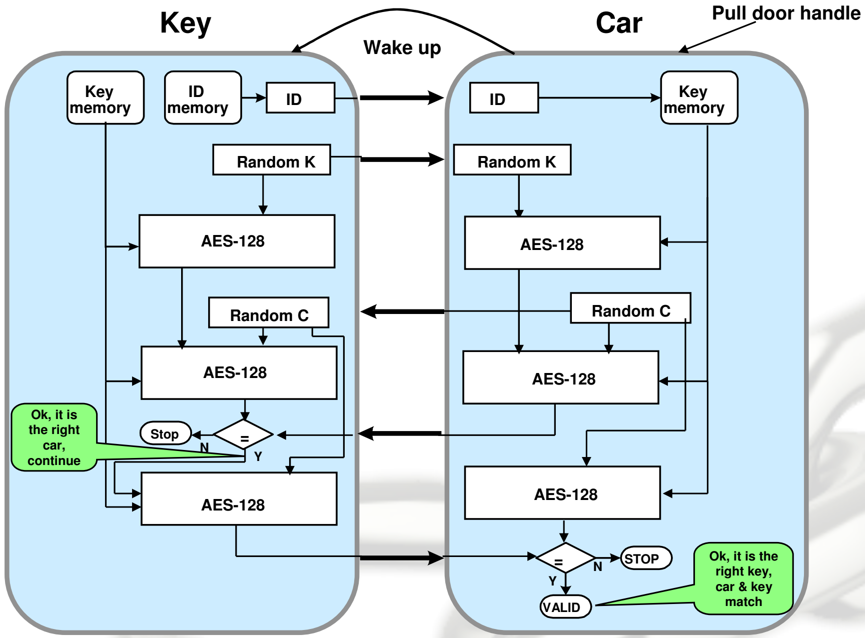
- Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stuble. **An open approach for designing secure electronic immobilizers**. In *Information Security Practice and Experience (ISPEC 2005)*, volume 3439 of *Lecture Notes in Computer Science*, pages 230–242. Springer-Verlag, 2005.
- Kerstin Lemke, Ahmad-Reza Sadeghi, and Christian Stuble. **Anti-theft protection: Electronic immobilizers**. *Embedded Security in Cars*, pages 51–67, 2006.
- Pang-Chieh Wang, Ting-Wei Hou, Jung-Hsuan Wu, and Bo-Chiuan Chen. **A security module for car appliances**. *International Journal of World Academy Of Science, Engineering and Technology*, 26:155– 160, 2007.
- Marko Wolf, Andre Weimerskirch, and Thomas Wollinger. **State of the art: Embedding security in vehicles**. *EURASIP Journal on Embedded Systems*, 2007:074706, 2007.
- Jung-Hsuan Wu, Chien-Chuan Kung, Jhan-Hao Rao, Pang-Chieh Wang, Cheng-Liang Lin, and Ting-Wei Hou. **Design of an in-vehicle anti-theft component**. In *8th International Conference on Intelligent Systems Design and Applications (ISDA 2008)*, volume 1, pages 566–569. IEEE Computer Society, 2008.

Atmel Open Immobilizer Protocol Stack

- Atmel Corporation states in their datasheets:

  “Rather than developing its own proprietary cryptographic functions, Atmel selected and implemented the 128-bit AES-128 global benchmark standard as its data encryption and decryption source. ***This open source standard is freely available to the public for use and scrutiny.*** *Because of this it continues to be favored by industry experts over private and proprietary crypto algorithms.“*

- Key Features
  - No security by obscurity
  - Use of 128-bits AES
  - Car & key send challenge
  - Open protocol design
  - Open source examples
  - Allows public evaluation



Conclusion

- Automotive industry has focus on safety, but not enough on security
- They still use proprietary out-dated cryptographic algorithms
- Maybe they should consider a new update model (like “Patch Tuesday”)

Historical claim

Final paper SHA-512 hash:

```
9d05ba88740499eecea3d8609174b444
43683da139f78b783666954ccc605da8
4601888134bf0c23ba46fb4a88c056bf
bbb629e1ddffcf60fa91880b4d5b4aca
```