Dismantling Megamos Crypto: Wirelessly Lockpicking a Vehicle Immobilizer

Roel Verdult
Radboud University
Nijmegen, NL

Flavio D. Garcia
University of
Birmingham, UK

Baris Ege
Radboud University
Nijmegen, NL

Why this special paper presentation?

- This paper was first accepted at Usenix Security'13
- VW sought an injunction from the High Court of London to prevent publication
- The High Court of London granted an interim injunction and therefore we had to withdraw the article
- We have now reached an amicable settlement without any admission of liability
- We will talk about the technical content of the paper but not about the details of the case

Vehicle Immobilizers

- Passive RFID Tags (125 KHz)
- Prevent hot-wiring
- Mandatory
  - Europe (EU Directive 95/56/EC)
  - Australia (AS/NZS 4601:1999)
  - Canada (CAN/ULC S338- 98)
- Prevent insurance fraud
- Should not be confused with remote controls that unlock the car doors (433 MHz)

Vehicle Immobilizers

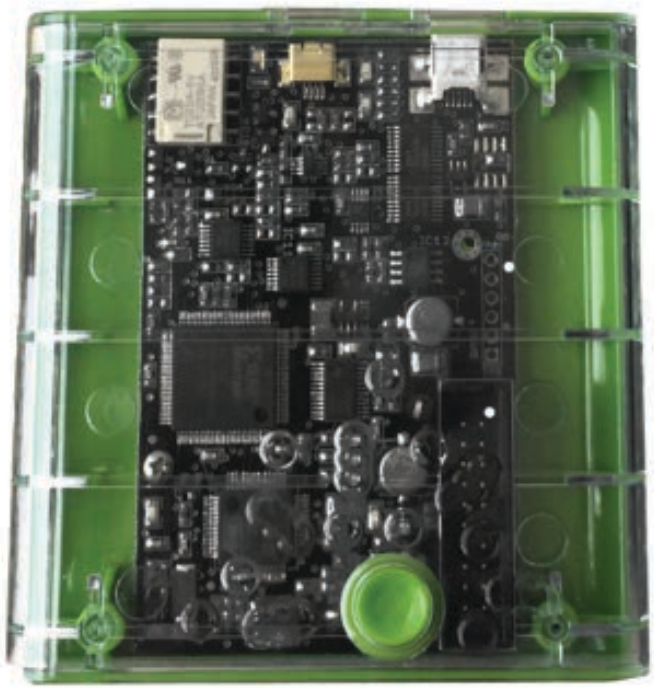
Three main immobilizer chips used (2012-13)

- TI's DST (**40**-bit key)
  - Bono et al. "Security Analysis of a Cryptographically-Enabled RFID Device" [Usenix Security'05]
- NXP's Hitag2 (**48**-bit key)
  - Analysed in our paper "Gone in 360 Seconds: Hijacking with Hitag2" [Usenix Security'12]
- EM's Megamos Crypto (**96**-bit key)
  - **This talk**

Megamos Crypto Usage (2013)

Make	Models
Alfa Romeo	147, 156, GT
Audi	A1, A2, A3, **A4 (2000)**, A6, **A8 (1998)**, Allroad, Cabrio, Coupe, Q7, S2, S3, S4, S6, S8, **TT (2000)**
Buick	Regal
Cadillac	CTS-V, SRX
Chevrolet	Aveo, Kalos, Matiz, Nubira, Spark, Evanda, Tacuma
Citroen	**Jumper (2008)**, Relay
Daewoo	Kalos, Lanos, Leganza, Matiz, Nubira, Tacuma
DAF	CF, LF, XF
Ferrari	California, 612 Schaglietti
Fiat	Albea, Doblo, Idea, Mille, Multipla, Palio, **Punto (2002)**, Seicento, Siena, **Stilo (2001)**, **Ducato (2004)**
Holden	Barina, Frontera
Honda	Accord, Civic, CR-V, FR-V, HR-V, Insight, **Jazz (2002, 2006)**, Legend, Logo, S2000, Shuttle, Stream
Isuzu	Rodeo
Iveco	Eurocargo, Daily
Kia	Carnival, Clarus, Pride, Shuma, Sportage
Lancia	Lybra, Musa, Thesis, Y
Maserati	Quattroporte
Opel	Frontera
Pontiac	G3
Porsche	911, 968, Boxster
Seat	Altea, Cordoba, **Ibiza (2014)**, Leon, Toledo
Skoda	**Fabia (2011)**, Felicia, Octavia, Roomster, Super, Yeti
Ssangyong	Korando, Musso, Rexton
Tagaz	Road Partner
Volkswagen	Amarok, Beetle, Bora, Caddy, Crafter, Cross Golf, Dasher, Eos, Fox, Gol, **Golf (2006, 2008)**, Individual, Jetta, Multivan, New Beetle, Parati, Polo, Quantum, Rabbit, Saveiro, Santana, **Scirocco (2011)**, Touran, **Tiguan (2010)**, Voyage, **Passat (1998, 2005)**, Transporter
Volvo	C30, **S40 (2005)**, S60, S80, **V50 (2005)**, V70, XC70, XC90, XC94

Hardware Setup



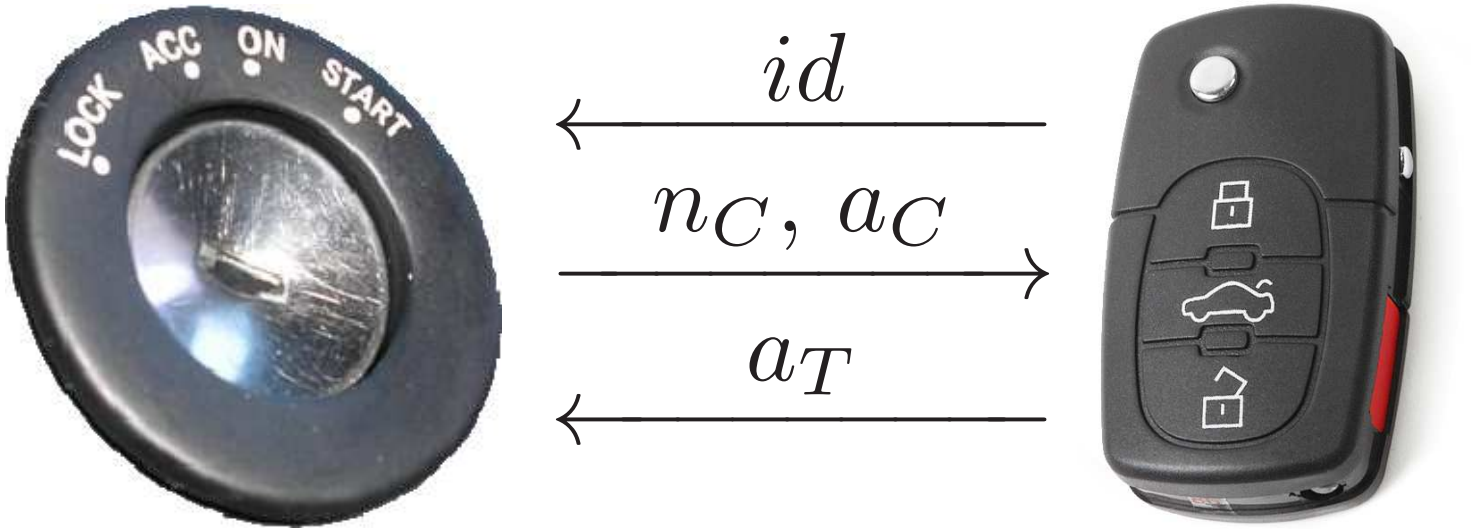
- Proxmark III
  - 125 kHz, 13.56 MHz
  - ADC, uC and FPGA
  - Open design/source

Tag Memory layout (from datasheet)

Block	Content	Denoted by
0	user memory	um_0 um_{15}
1	user memory, lock bits	um_{16} $um_{29} l_0 l_1$
2	device identi cation	id_0 id_{15}
3	device identi cation	id_{16} id_{31}
4	crypto key	k_0 k_{15}
5	crypto key	k_{16} k_{31}
6	crypto key	k_{32} k_{47}
7	crypto key	k_{48} k_{63}
8	crypto key	k_{64} k_{79}
9	crypto key	k_{80} k_{95}
10	pin code	pin_0 pin_{15}
11	pin code	pin_{16} pin_{31}
12	user memory	um_{30} um_{45}
13	user memory	um_{46} um_{61}
14	user memory	um_{62} um_{77}
15	user memory	um_{78} um_{93}

read-only
write-only
read-write

Megamos Authentication Protocol



id = 32-bit Tag identifier

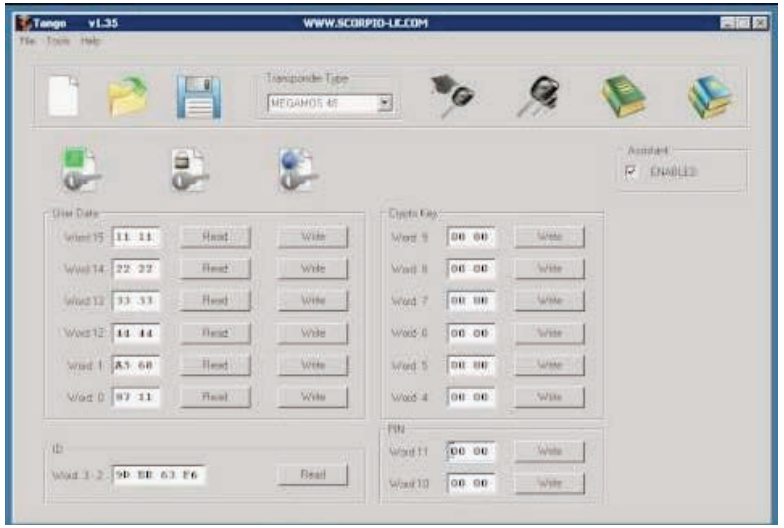
n_C = 56-bit Car nonce

a_C = 28-bit Car authenticator (keystream)

a_T = 20-bit Tag authenticator (keystream)

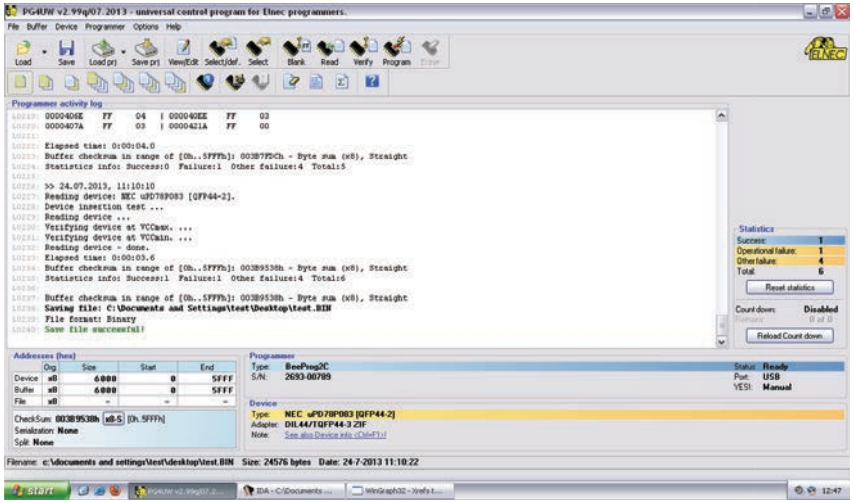
Reverse engineering Megamos Crypto

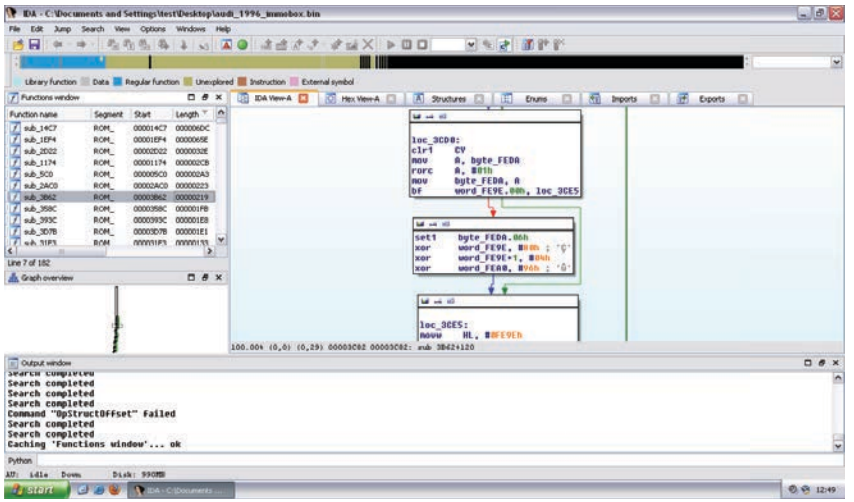
- We discovered that the Tango Programmer (car diagnostic tool) uses the Megamos Crypto algorithm since 2009 (for testing purposes only)
- We reverse-engineered the algorithm from the freely available Tango software package bypassing its obfuscation.



… but you can also read it directly from the car's ECU

NEC uPD78P083 has simply **no protection**





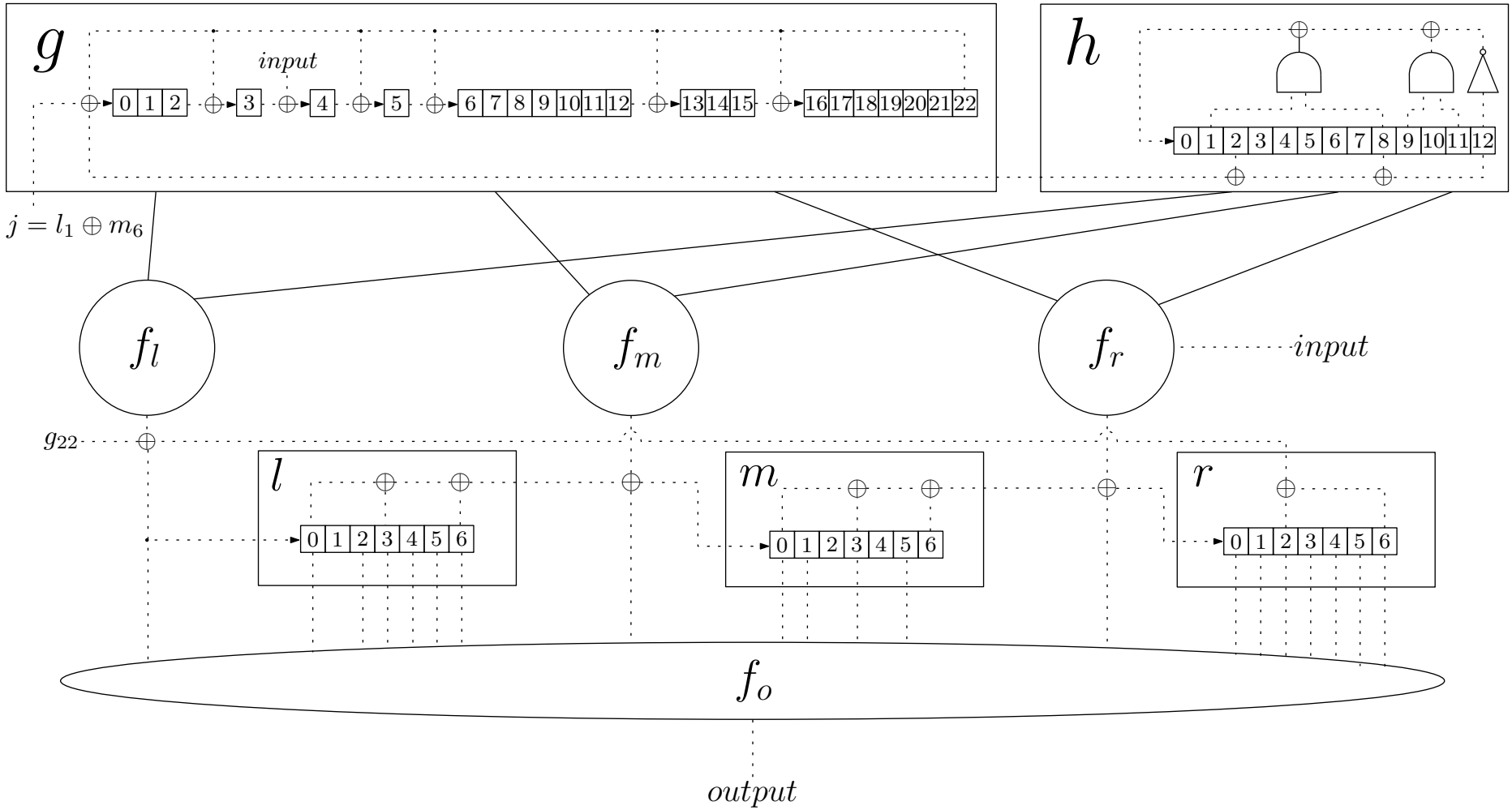
Cryptanalysis - Pre-requisites

- Requires access to the **car and the car key**
- Adversary needs to turn the ignition on twice and eavesdrop two traces

Origin	Message
Car	3
Transponder	A9 08 4D EC
Car	5
Transponder	80 00 95 13
Car	F
Transponder	AA AA AA AA AA AA AA AA
Car	6 | 3F FE 1F B6 CC 51 3F | 0^7 | F3 55 F1 A
Transponder	60 9D 6

Complexity analysis of the cipher

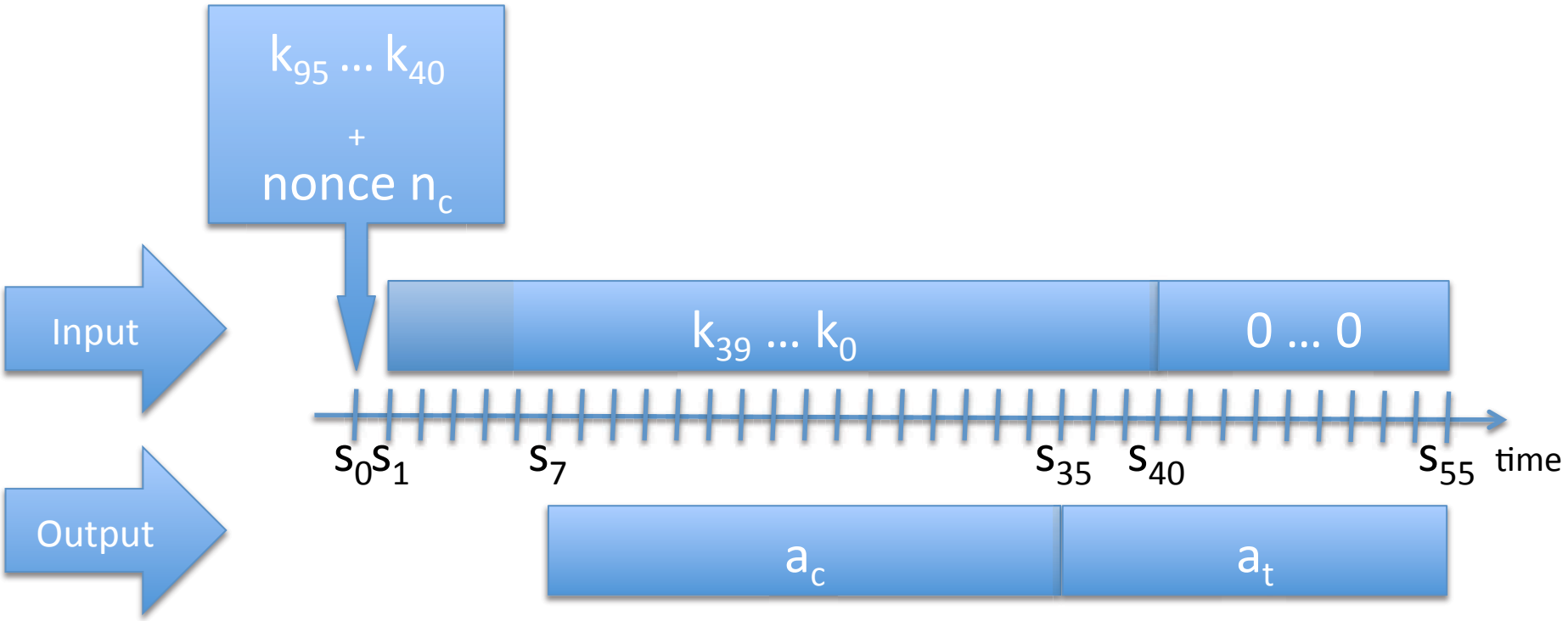
The Megamos Crypto Cipher



Secret key size = 96 bits

Internal state size = 23 + 13 + 3x7 = 57 bits

Megamos Crypto Initialization and workings

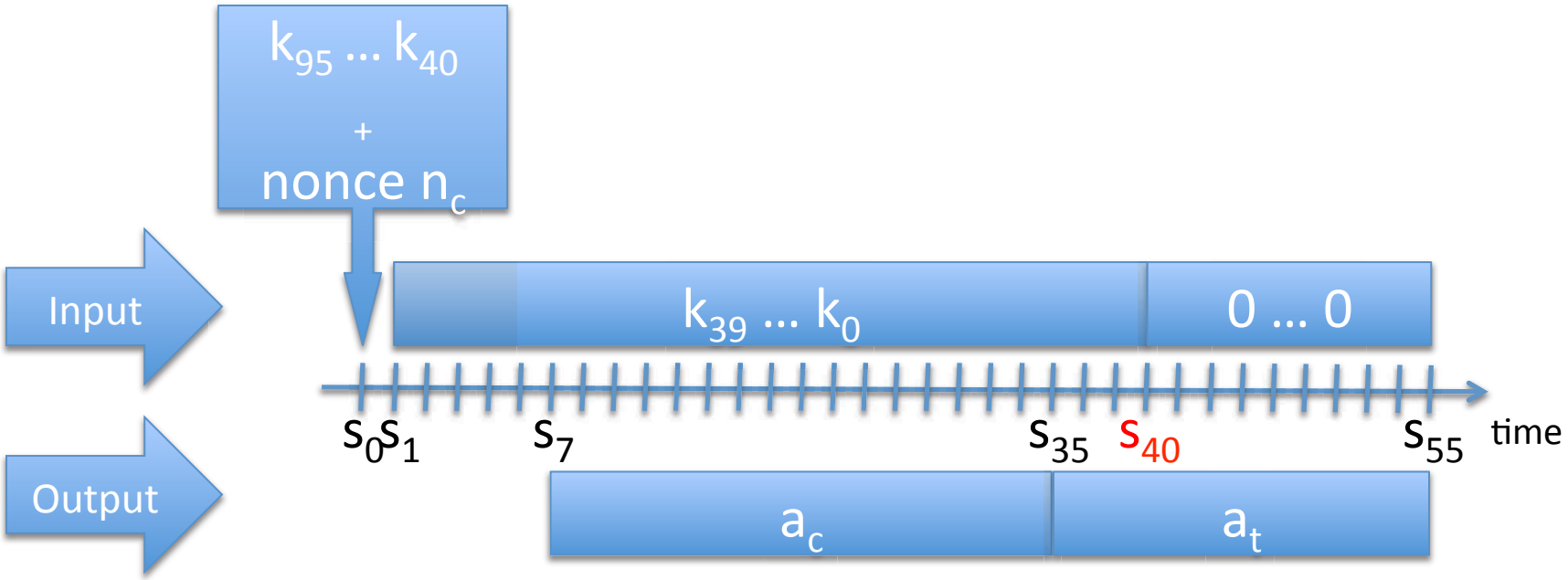


n_C = 56-bit Car nonce

a_C = 28-bit Car authenticator (keystream)

a_T = 20-bit Tag authenticator (keystream)

Cryptanalysis of Megamos Crypto



- Take the first authentication trace
- Trying all 2^{56} states $s_{40,}$ running 15 steps discarding on the output leaves 2^{41} candidate states
- After running the cipher backwards to s_7 we still have 2^{41} candidates
- Running backwards to s_0 guessing 7 bit leaves 2^{48} candidate keys.
- Check against a second authentication trace singles out the key.

Cryptanalysis of Megamos Crypto

- Total attack complexity reduced from 2^{96} to less than 2^{56} encryptions
- Takes less than two days on a Copacobana‘05
- This complexity can be further reduced by precomputation:
  - E.g., using a 12 Terabyte table reduces the complexity to 2^{49} table lookups
  - This has some practical limitations

Partial Key-update Attack

Observations:

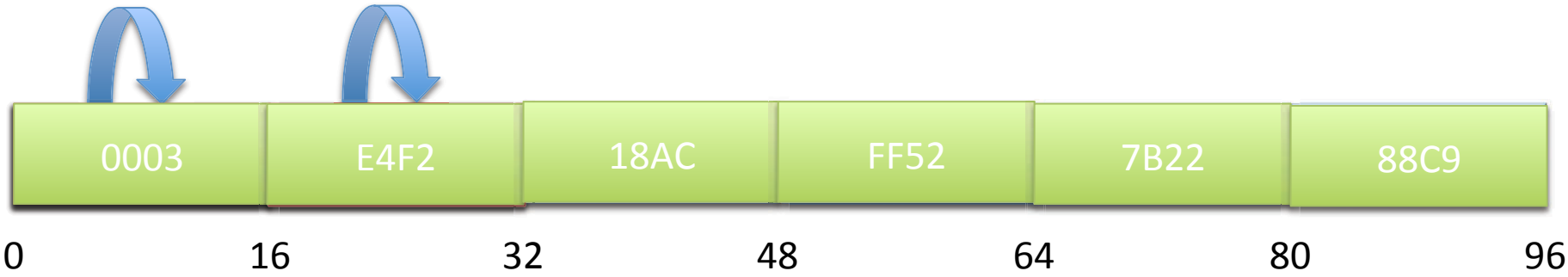
During our research, the majority of deployed tags we found were:

- Unlocked $l_0 = 0$ (writable)
- Could be unlocked with a default PIN code

Block	Content	Denoted by
0	user memory	um_0 um_{15}
1	user memory, lock bits	um_{16} $um_{29} l_0 l_1$
2	device identi cation	id_0 id_{15}
3	device identi cation	id_{16} id_{31}
4	crypto key	k_0 k_{15}
5	crypto key	k_{16} k_{31}
6	crypto key	k_{32} k_{47}
7	crypto key	k_{48} k_{63}
8	crypto key	k_{64} k_{79}
9	crypto key	k_{80} k_{95}
10	pin code	pin_0 pin_{15}
11	pin code	pin_{16} pin_{31}
12	user memory	um_{30} um_{45}
13	user memory	um_{46} um_{61}
14	user memory	um_{62} um_{77}
15	user memory	um_{78} um_{93}

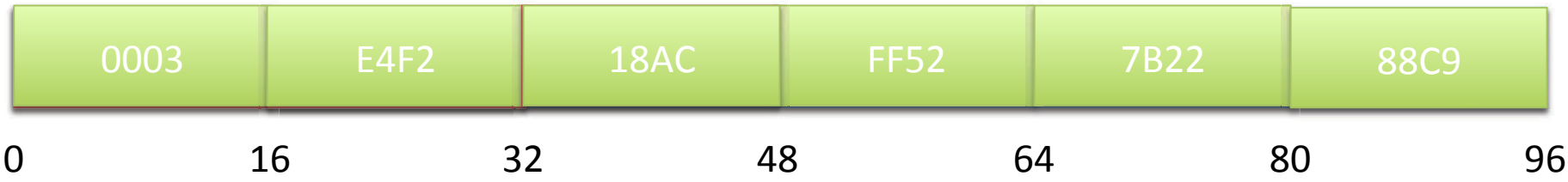
- The 96-bit secret key is written to the tag in blocks of 16 bits instead of being an atomic operation.

Partial Key-update Attack (simple)



- Get one authentication attempt from the car
- Guess 16 bits, write on one block then authenticate to the tag.
- If it succeeds you learn 16 key bits.
- This requires 6 x 2^{16} writes and authenticate
- Takes 25' per block ≈ 2.5 hours in total, using a Proxmark

Partial Key-update Attack (optimized)



- Same principle but only write zeros once in the first block
- Then increment the nonce and authenticate until the tag accepts
  - **Remember key is added to the nonce** during initialisation
- Repeat for another two blocks then combine with the cryptanalytic attack searching for the remaining bits
- This attack requires 6 writes and 3×2^{16} authentications with the tag and negligible computational complexity
- The whole attack takes <30 minutes using a Proxmark III

Immobilizer Demo

Weak key attack

Some interesting keys we found

- If the key starts with 32 zero bits then you can use a time-memory trade-off as in [Oechslin'03]
- Build (once) a 1.5 Terabyte rainbow table (less than one week to build)
- Computational complexity of 2^{37} encryptions
- Few minutes computation on a laptop

Weak key attack

Some even more interesting keys we found

Car	Secret key
A 1	00000000**d**8 **b**3**9**67c**5**a**3**c**3**b**2**9
A 2	00000000**d**9 **b**7**9**d7a**5**b**3**c**3**b**2**8
B 1	0000000**0**0 **0**0**0**1**0**4**0**5**0**5**0**9**0**5

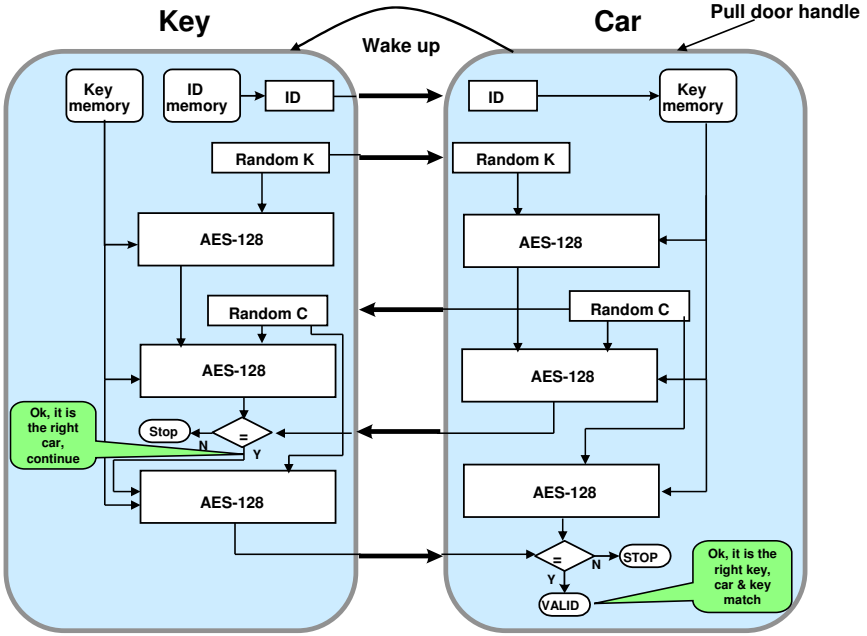
- These keys appear to have at most 32 bits of entropy
- An exhaustive search on such key takes only seconds

Mitigation and Alternatives

- Car owners can set lock-bit l_0 to one, set a random PIN. This prevents our partial key update attack.
- Set full entropy keys (locksmiths, dealers)
- Vehicle immobilizer tags based on the Advanced Encryption Standard (AES)
    - HITAG Pro, NXP Semiconductors (2007)
    - ATA5580, Atmel Corporation (2010)
    - TRPWS21/TRPBS27, Texas Instruments (2010)

Atmel Open Immobiliser Protocol Stack

- Atmel Corporation states in the 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



Responsible disclosure

- We carefully followed the official guidelines from the Dutch Government [1]
- We notified the chip manufacturer in November 2012, nine months ahead of scheduled publication at Usenix'13.
- We invested many days to inform them properly
  - conference call
  - several letters and emails
  - personal meeting
- We understand that measures have been taken to prevent our weak-key and partial key-update attacks in newer vehicles

[1] https://www.ncsc.nl/english/current-topics/responsible-disclosure-guideline.html

Thanks for staying around!

Acknowledgements

We would like to thank the following colleagues and friends for their firm support (in alphabetical order)

Ross Anderson
Robert Carolina
Tom Chothia
Riccardo Focardi
Dorine Gebbink
Casey Henderson
Bart Jacobs
Sam King
Bas Kortmann
Kenny Paterson
Carolyn Pike
Jon Rowe
Mark Ryan
Graham Steel