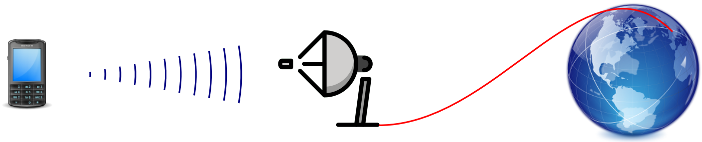
Cryptanalysis of GEA-1 and GEA-2

Christof Beierle, Patrick Derbez, Gregor Leander, Gaëtan Leurent,
Håvard Raddum, Yann Rotella, David Rupprecht, and Lukas Stennes

Journée GDR Sécu

GEA: GPRS Encryption Algorithm



- GPRS is the data protocol of 2G telephony (sometimes called 2.5G)
  - Improved GPRS: EDGE (sometimes called 2.75G)
  - Designed by ETSI SAGE in 1998
- Widely used in the early 2000s
  - The first iPhone didn't support 3G (2008)
  - 3G deployment: 2001–2010-ish
- 2G has been sunset in some countries, but still used in France
  - Fallback when 3G/4G/5G not available
  - Used by some payment terminals

2G security

- Encryption of packets between the phone and the antenna
- Algorithms designed in secret in the 1980s and 1990s, not published

Voice

A5/1 64-bit key, 64-bit state

- Partial leak in 1994, Reverse engineered in 1999
- Best attack: < 1 minute
- In practice: rainbow tables (precomputation of 2^{57})

A5/2 64-bit key, 81-bit state

- Reverse engineered in 1999
- Best attack: 2^{16} (“export version”)
- Deprecated in 2007

A5/3 KASUMI (public) designed in 2002

Data

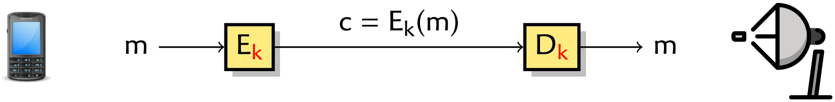
GEA-1 64-bit key, 96-bit state

- Partial leak in 2011 [Nohl & Melette]
- Deprecated in 2013

GEA-2 64-bit, 125-bit state

GEA-3 KASUMI (public) designed in 2002

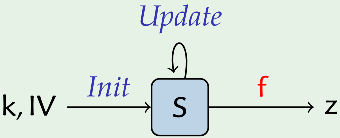
Stream ciphers



- Encrypt a message with a secret key k
- Keystream $z(k) = (z^{(0)}, z^{(1)}, z^{(2)}, \ldots)$
  - $c = E_k(m) = m \oplus z$

Stream cipher

- Internal state $S \in \mathcal{S}$
- State update function $\mathcal{S} \to \mathcal{S}$
- Extraction function $f : \mathcal{S} \to \{0,1\}$
- Initialization $k, IV \to \mathcal{S}$

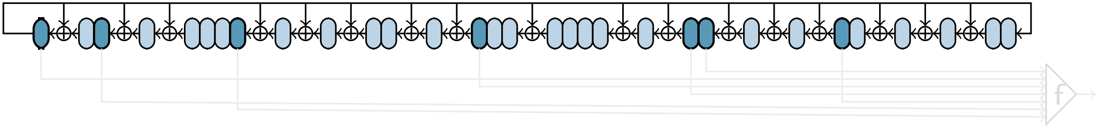


$$S^{(0)} = \mathrm{Init}(k) \qquad S^{(i+1)} = \mathrm{Update}(S^{(i)}) \qquad z^{(i)} = f(S^{(i)})$$

Filter generator

Linear Feedback Shift Register – LFSR (Galois configuration)

- State S: n bits $(s_0, s_1, \dots, s_{n-1})$
- Update depending on taps $\mathcal{A}$: $s_i^{(t+1)} = \begin{cases} s_{i+1}^{(t)} \oplus s_0^{(t)} & \text{if } i \in \mathcal{A} \\ s_{i+1}^{(t)} & \text{else} \end{cases}$
- Polynomial representation: $Q = X^n + \sum_{i \in \mathcal{A}} X^i$
  - If Q is primitive, update corresponds to multiplication by a primitive element
  - Maximal period if $S \neq 0$

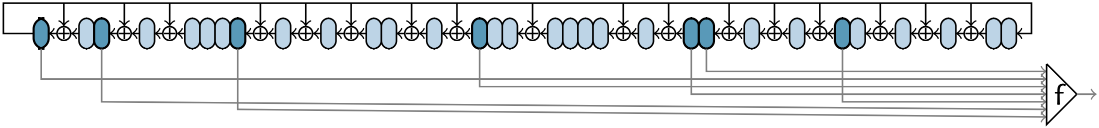


- Filter function to extract keystream from internal state (balanced, non-linear)
- Construction used in A5/1, A5/2, Bluetooth E0

Filter generator

Linear Feedback Shift Register – LFSR (Galois configuration)

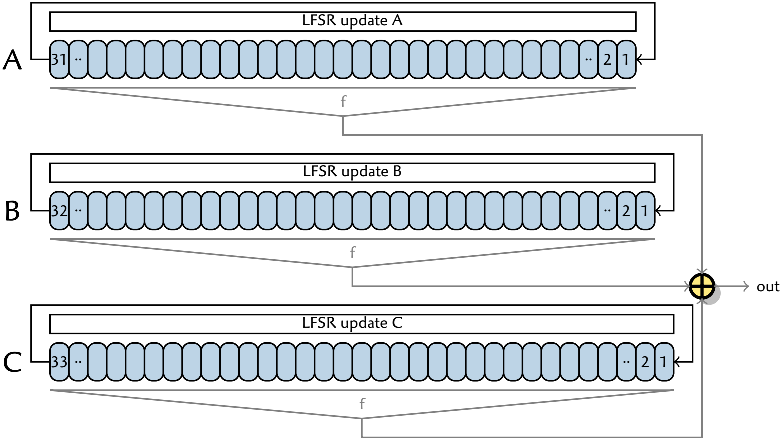
- State S: n bits $(s_0, s_1, \ldots, s_{n-1})$
- Update depending on taps $\mathcal{A}$: $s_i^{(t+1)} = \begin{cases} s_{i+1}^{(t)} \oplus s_0^{(t)} & \text{if } i \in \mathcal{A} \\ s_{i+1}^{(t)} & \text{else} \end{cases}$
- Polynomial representation: $Q = X^n + \sum_{i \in \mathcal{A}} X^i$
  - If Q is primitive, update corresponds to multiplication by a primitive element
  - Maximal period if $S \neq 0$



- Filter function to extract keystream from internal state (balanced, non-linear)
- Construction used in A5/1, A5/2, Bluetooth E0

GEA-1 design

- Received specification from a "source"
- Three filter generators
  - A (31 bits) $\hookrightarrow \texttt{Gen}_A(A)$
  - B (32 bits) $\hookrightarrow \texttt{Gen}_B(B)$
  - C (33 bits) $\hookrightarrow \texttt{Gen}_C(C)$
- Non-linear filtering
  - degree-4 function f



- The keystream is $z = \texttt{Gen}_A(A) \oplus \texttt{Gen}_B(B) \oplus \texttt{Gen}_C(C)$

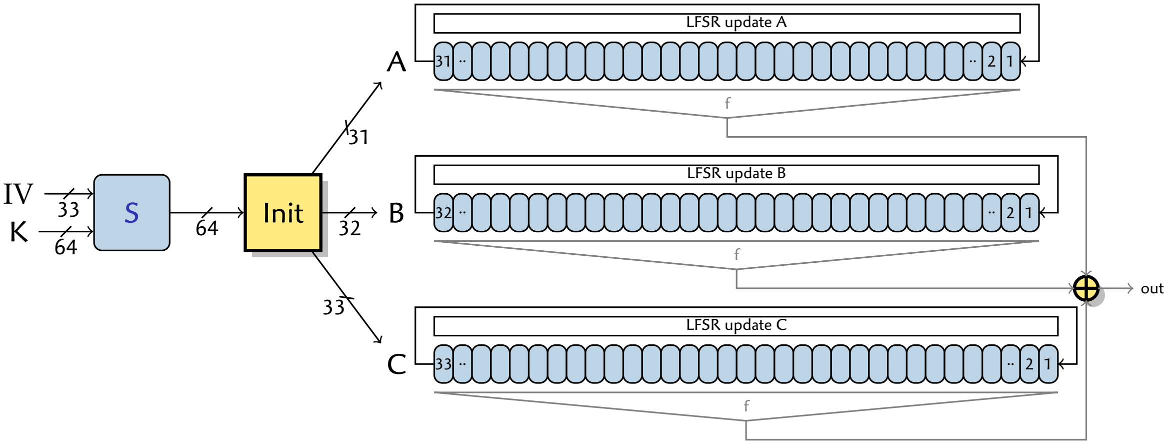
GEA-1 initialization

1. Generate a 64-bit value S from the key and IV
   - Using a NLFSR (non linear)

2. Initialize the three LFSRs from S
   - Set A, B, C to zero
   - Clock them 64 times, xor s_i into the feedback function
     - A uses $s_0, s_1, \dots, s_{64}$
     - B uses $s_{16}, s_{17}, \dots, s_{15}$ (shifted by 16 positions)
     - C uses $s_{32}, s_{33}, \dots, s_{31}$ (shifted by 32 positions)

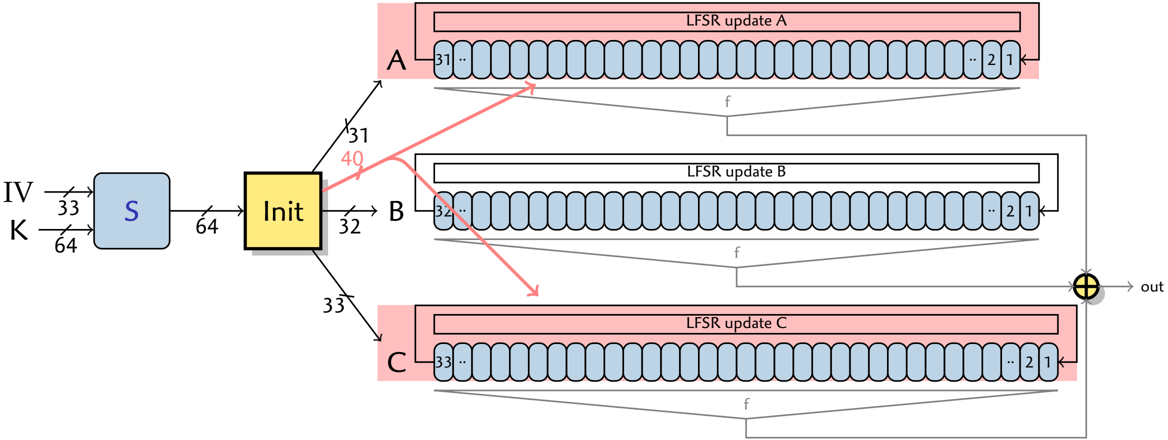
- Initialization of A, B, C from S is linear
  - $S \mapsto A$: 64 bit $\rightarrow$ 31 bits, rank 31
  - $S \mapsto B$: 64 bit $\rightarrow$ 32 bits, rank 32
  - $S \mapsto C$: 64 bit $\rightarrow$ 33 bits, rank 33
  - $S \mapsto (A, B, C)$: 64 bit $\rightarrow$ 96 bits, rank 64
  - $S \mapsto (A, C)$: 64 bit $\rightarrow$ 64 bits, rank 40

GEA-1 initialization



- Initialization of A, B, C from S is linear
  - $S \mapsto A$: 64 bit → 31 bits, rank 31
  - $S \mapsto B$: 64 bit → 32 bits, rank 32
  - $S \mapsto C$: 64 bit → 33 bits, rank 33
  - $S \mapsto (A, B, C)$: 64 bit → 96 bits, rank 64
  - $S \mapsto (A, C)$: 64 bit → 64 bits, rank 40

GEA-1 initialization



- Initialization of A, B, C from S is linear
  - $S \mapsto A$: 64 bit → 31 bits, rank 31
  - $S \mapsto B$: 64 bit → 32 bits, rank 32
  - $S \mapsto C$: 64 bit → 33 bits, rank 33
  - $S \mapsto (A, B, C)$: 64 bit → 96 bits, rank 64
  - $S \mapsto (A, C)$: 64 bit → 64 bits, rank 40

Meet-in-the-Middle attack

- There are 2^{40} possible initial states for (A, C)
- There are 2^{32} possible initial states for B
- The keystream is $z = \mathtt{Gen}_A(A) \oplus \mathtt{Gen}_B(B) \oplus \mathtt{Gen}_C(C)$
  - Split in two independent parts: $\mathtt{Gen}_B(B) = z \oplus \mathtt{Gen}_A(A) \oplus \mathtt{Gen}_C(C)$

Meet-in-the-Middle attack / collision search

0. Capture frame with known plaintext, recover z
1. For all 2^{32} B, compute $\mathtt{Gen}_B(B)$ and store in a hash table
2. For all 2^{40} (A, C), compute $z \oplus \mathtt{Gen}_A(A) \oplus \mathtt{Gen}_C(C)$ and look up in the table

- Recover the key from the initial state (A, B, C)
- Complexity
  - 64 bits of known keystream
  - 2^{40} Time
  - 2^{32} Memory

Backdoor?

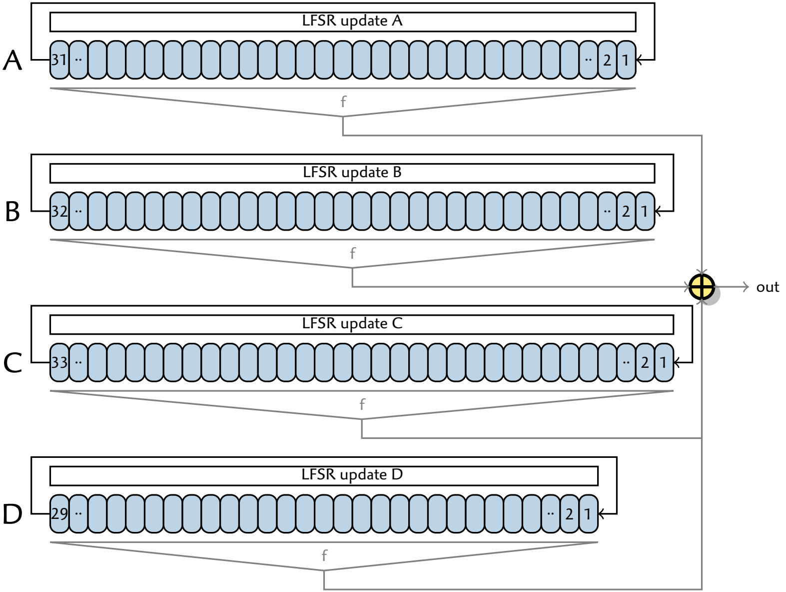
GEA-1 was likely weakened deliberately

- Mapping $S \mapsto A, C$ from 64 bits to 64 bits
  - Having rank 40 is very unlikely
- Experiments with initialization of the same type
  - With 1 million experiments, lowest rank found is 55
  - Follow-up work to build LFSRs and shift with low rank [Beierle, Felke & Leander, 2021]

- In the 1990's, cryptography was subjected to export regulation
  - In France, 40-bit security cryptography can be exported after 1998
- The design document states:
  "the algorithm should be generally exportable taking into account current export restrictions"
  "the strength should be optimized taking into account the above requirement"
- Other examples of "export" ciphersuites: TLS, A5/2 in GSM

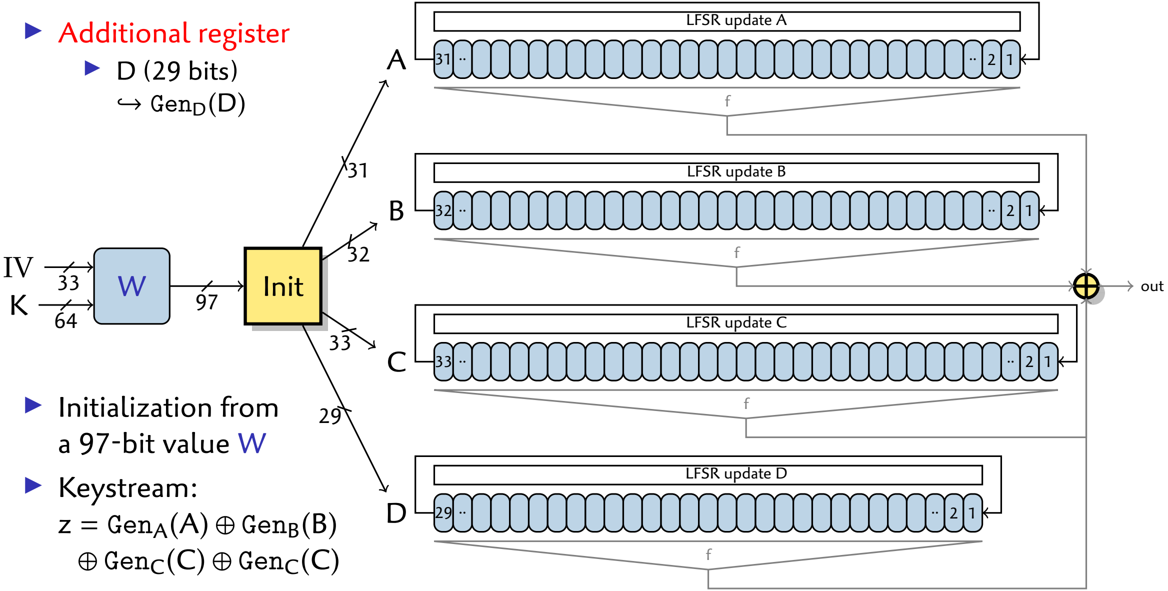
GEA-2 design

- Additional register
  - D (29 bits)
    $\hookrightarrow \mathtt{Gen}_D(D)$
- Initialization from a 97-bit value W
- Keystream:
  $z = \mathtt{Gen}_A(A) \oplus \mathtt{Gen}_B(B) \oplus \mathtt{Gen}_C(C) \oplus \mathtt{Gen}_C(C)$



GEA-2 design

- Additional register
  - D (29 bits)
    $\hookrightarrow \mathrm{Gen}_D(D)$



- Initialization from a 97-bit value W
- Keystream:
  $z = \mathrm{Gen}_A(A) \oplus \mathrm{Gen}_B(B)$
  $\oplus \mathrm{Gen}_C(C) \oplus \mathrm{Gen}_C(C)$

Meet-in-the-Middle attack

- The keystream is $z = \texttt{Gen}_A(A) \oplus \texttt{Gen}_B(B) \oplus \texttt{Gen}_C(C) \oplus \texttt{Gen}_D(D)$
  - Register sizes: 31 (A), 32 (B), 33(C), 29 (D)
- Standard MitM: $\texttt{Gen}_A(A) \oplus \texttt{Gen}_B(B) = z \oplus \texttt{Gen}_C(C) \oplus \texttt{Gen}_D(D)$
  - Complexity $\approx 2^{63}$ ((A, B) is 63 bits, (C, D) is 62 bits)
- No unexpected rank loss

Algebraic attack: linearisation

Writing $z^{(i)} = \mathrm{Gen}_A^{(i)}(A) \oplus \mathrm{Gen}_B^{(i)}(B) \oplus \mathrm{Gen}_C^{(i)}(C) \oplus \mathrm{Gen}_D^{(i)}(D)$ as a polynomial

- $31 + 32 + 33 + 29 =$ 125 variables
- Each keystream bit $z^{(i)}$ gives an equation

Toy example

- Small number of possible monomials
  - LFSR update is linear
  - The filtering function f has algebraic degree 4
  - $\sum_{i=1}^{4} \binom{31}{i} + \binom{32}{i} + \binom{33}{i} + \binom{29}{i} =$ 152682 monomials

- Linearisation attack:
  - Consider each monomial as an independent variable
  - Solve the linear system
  - Complexity $152682^3 \approx 2^{52}$
- Requires about 152682 bits of keystream z
- Problem: GPRS frame is at most 12800 bits

Partial guessing

- We can reduce the number of monomial below 12800 by guessing some state bits
- For instance: guess 15 bits of A, 15 bits of B, 16 bits of C, 13 bits of D
  - Remaining variables: 16 (A) + 17 (B) + 17 (C) + 16 (D)
  - $\sum_{i=1}^{4} \binom{16}{i} + \binom{17}{i} + \binom{17}{i} + \binom{16}{i} = 11468$ monomials (< 12800)
- Solve the remaining system with linear algebra
  - Complexity $\approx 2^{59} \times 12800^3$

Hybrid Meet-in-the-Middle

Strategy

1. Guess parts of A and D
2. Find relations that depend only on B, C: $\phi(B) \oplus \psi(C) = \xi(z)$

- Guess 11 bits of A and 9 bits of D
- Write $w^{(i)} = \mathtt{Gen}_A^{(i)}(A) \oplus \mathtt{Gen}_D^{(i)}(D)$ as a polynomial in the remaining variables (20+20)
- Look for masks m (length 12800) such that $m \cdot w_0 \dots w_{12799}$ is constant
  - $\sum_{i=1}^{4} \binom{20}{i} + \binom{20}{i} = 12390$ non-constant monomials
  - Using linearisation, space of good masks of dimension $12800 - 12390 = 410$
- Build linear function L from 64 independent masks:
  - $z = \mathtt{Gen}_D(D) \oplus \mathtt{Gen}_A(A) \oplus \mathtt{Gen}_B(B) \oplus \mathtt{Gen}_C(C)$
  - $\underbrace{L(z)}_{\text{known}} = \underbrace{L(\mathtt{Gen}_D(D)) \oplus L(\mathtt{Gen}_A(A))}_{\text{constant}} \oplus \underbrace{L(\mathtt{Gen}_B(B))}_{\phi(B)} \oplus \underbrace{L(\mathtt{Gen}_C(C))}_{\psi(C)}$

Linearization: toy example

	1	a_0	a_1	a_2	a_0a_1	a_0a_2	a_1a_2	b_0	b_1	b_0b_1
$w_0 =$	$1\oplus$	$a_0\oplus$						b_0		
$w_1 =$			$a_1\oplus$			$a_0a_2\oplus$			$b_1\oplus$	b_0b_1
$w_2 =$	$1\oplus$	$a_0\oplus$		$a_2\oplus$	$a_0a_1\oplus$					b_0b_1
$w_3 =$	$1\oplus$	$a_0\oplus$	$a_1\oplus$		$a_0a_1\oplus$		$a_1a_2\oplus$	$b_0\oplus$	b_1	
$w_4 =$				$a_2\oplus$		$a_0a_2\oplus$		$b_0\oplus$		b_0b_1
$w_5 =$		$a_0\oplus$		$a_2\oplus$			$a_1a_2\oplus$		$b_1\oplus$	b_0b_1
$w_6 =$			$a_1\oplus$		$a_0a_1\oplus$	$a_0a_2\oplus$		b_0		
$w_7 =$	$1\oplus$	$a_0\oplus$	$a_1\oplus$		$a_0a_1\oplus$		$a_1a_2\oplus$			b_0b_1
$w_8 =$	$1\oplus$	$a_0\oplus$		$a_2\oplus$			$a_1a_2\oplus$		$b_1\oplus$	b_0b_1
$w_9 =$			$a_1\oplus$	$a_2\oplus$		$a_0a_2\oplus$		$b_0\oplus$	$b_1\oplus$	b_0b_1
$w_{10} =$			$a_1\oplus$		$a_0a_1\oplus$	$a_0a_2\oplus$			b_1	
$w_{11} =$		$a_0\oplus$	$a_1\oplus$						$b_1\oplus$	b_0b_1

$$w_0 \oplus w_2 \oplus w_9 \oplus w_{10} = 1$$
$$w_2 \oplus w_5 \oplus w_7 \oplus w_{11} = 0$$
$$w_5 \oplus w_8 = 1$$

Hybrid Meet-in-the-Middle

Precomputation

- For each 2^{20} (a, d) (partial guess of A and D)
  1. Compute linear combinations of w independent of remaining (A, D)
  2. Deduce functions $\phi_{a,d}, \psi_{a,d}, \xi_{a,d}$ such that $\phi_{a,d}(B) = \psi_{a,d}(C) \oplus \xi_{a,d}(z)$

- Complexity: $2^{20} \times 12800^3/64 \approx 2^{54.9}$ 64-bit operations

Meet-in-the-Middle attack / collision search

- For each 2^{20} (a, d) (partial guess of A and D)
  1. For all 2^{32} B, compute $\phi_{a,d}(B)$ and store in a hash table
  2. For all 2^{33} C, compute $\xi_{a,d}(z) \oplus \psi_{a,d}(C)$ and look up in the table
     - If there is match, recover key candidate from a, B, C, d

- Evaluation of $\phi_{a,d}, \psi_{a,d}$ as polynomials with amortized cost 4 [BCCCNSY, CHES'10]
- Complexity: $2^{52} + 2^{53} \approx 2^{53.6}$ memory access; $2^{54} + 2^{55} \approx 2^{55.6}$ 64-bit operations

Improvement: Time-Data Tradeoff

- Classical technique: target one state out of many [Babbage, 1995] [Golic, 1997]
- We target the first 753 states; 753 keystreams of length 12047
  - $(A^{(0)}, B^{(0)}, C^{(0)}, D^{(0)})$ produces keystream $z^{(0)} z^{(1)} z^{(2)} \ldots$
  - $(A^{(1)}, B^{(1)}, C^{(1)}, D^{(1)})$ produces keystream $z^{(1)} z^{(2)} z^{(3)} \ldots$
  - $(A^{(2)}, B^{(2)}, C^{(2)}, D^{(2)})$ produces keystream $z^{(2)} z^{(3)} z^{(4)} \ldots$
- Guess 11 bits of A and 10 bits of D
  - Write $w^{(i)} = \mathtt{Gen}_A^{(i)}(A) \oplus \mathtt{Gen}_D^{(i)}(D)$ as a polynomial in the remaining variables (19+20)
- Look for masks m (length 12047) such that $m \cdot w^{(0)} \ldots w^{(12046)}$ is constant
  - $\sum_{i=1}^{4} \binom{19}{i} + \binom{20}{i} = 11230$ non-constant monomials
  - Using linearisation, space of good masks of dimension $12047 - 11230 = 817$
- Filter masks such that $m \cdot z^{(0)} \ldots z^{(12046)} = m \cdot z^{(1)} \ldots z^{(12047)} = m \cdot z^{(2)} \ldots z^{(12048)} = \cdots$
  - Space of good masks of dimension $817 - 752 = 65$ (752 constraints)
- Build linear function L from 64 independent masks:
  - $z^{(s)} z^{(s+1)} \ldots = \mathtt{Gen}_D(D^{(s)}) \oplus \mathtt{Gen}_A(A^{(s)}) \oplus \mathtt{Gen}_B(B^{(s)}) \oplus \mathtt{Gen}_C(C^{(s)})$
  - $\underbrace{L(z^{(s)} z^{(s+1)} \ldots)}_{\text{independent of } s} = \underbrace{L(\mathtt{Gen}_D(D^{(s)})) \oplus L(\mathtt{Gen}_A(A^{(s)}))}_{\text{constant}} \oplus \underbrace{L(\mathtt{Gen}_B(B^{(s)}))}_{\phi(B^{(s)})} \oplus \underbrace{L(\mathtt{Gen}_C(C^{(s)}))}_{\psi(C^{(s)})}$

Hybrid Meet-in-the-Middle with Time-Data Tradeoff

Meet-in-the-Middle attack / collision search

- For each 2^{21} (a, d) (partial guess of A and D)
  0. Build functions $\phi_{a,d}, \psi_{a,d}, \xi_{a,d}$ such that $\phi_{a,d}(B) \oplus \psi_{a,d}(C) = \xi_{a,d}(z_s z_{s+1} \dots)$
  1. For all 2^{32} B, compute $\phi_{a,d}(B)$ and store in a hash table
  2. For all 2^{33} C, compute $\xi_{a,d}(z) \oplus \psi_{a,d}(C)$ and look up in table
     - If there is match, recover key candidate from a, B, C, d

- On average, only $2^{21}/753 \approx 2^{11.4}$ guesses until it matches one of the 753 targets
- Complexity: $2^{11.4} \times 2^{33.6} \approx 2^{45}$ memory access; $4 \times 2^{45} \approx 2^{47}$ 64-bit operations

Usage and deprecation

- In 2011, large usage of GEA-1 and GEA-2 [Nohl & Melette]
- GEA-1 deprecated in 2013

- In 2021, large usage of GEA-3 (also GEA-0 🤯) [umlaut report]
  - Some operators use GEA-2 as main algorithm
  - One operator seen using GEA-1 sometimes

- GEA-1 still implemented in recent phones!
  - (iPhone 8, Galaxy S9, ...)

- We contacted GSMA and ETSI for responsible disclosure
  - New test-case to verify non-implementation of GEA-1
  - Plans to deprecate GEA-2

Conclusion

- GEA-1 attack completely practical
  - Only 64 bits of known keystream, 2^{40} operations
  - 2.5 hours on a laptop today, practical in the 2000's

- GEA-2 attack borderline practical
  - Full frame known (12800 bits), 2^{45} operations
  - 4 months on a server

- In the early 2000's, internet traffic was mostly in the clear (low TLS use)
- Today, breaking GEA gives some metadata

- Semi-active downgrade attack [Barkan, Biham & Keller, C'2003]
  - Passive: Record frames encrypted with GEA-3
  - Active: force phone to use GEA-1 with same key, recover key

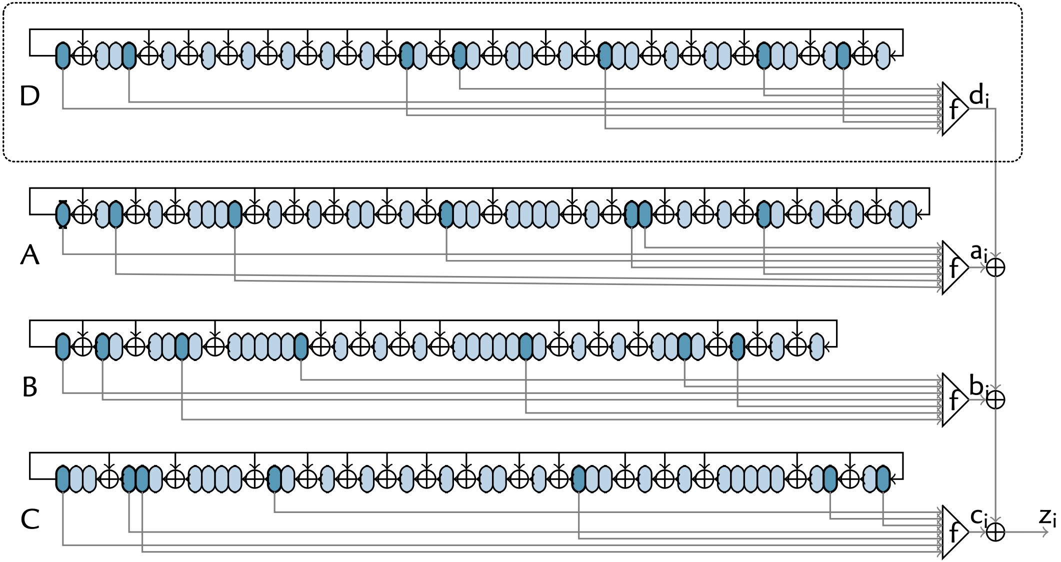
Conclusion

- GEA-1 attack completely practical
  - Only 64 bits of known keystream, 2^{40} operations
  - 2.5 hours on a laptop today, practical in the 2000's
- GEA-2 attack borderline practical
  - Full frame known (12800 bits), 2^{45} operations
  - 4 months on a server
- In the early 2000's, internet traffic was mostly in the clear (low TLS use)
- Today, breaking GEA gives some metadata
- Semi-active downgrade attack [Barkan, Biham & Keller, C'2003]
  - Passive: Record frames encrypted with GEA-3
  - Active: force phone to use GEA-1 with same key, recover key

Conclusion

- GEA-1 attack completely practical
  - Only 64 bits of known keystream, 2^{40} operations
  - 2.5 hours on a laptop today, practical in the 2000's
- GEA-2 attack borderline practical
  - Full frame known (12800 bits), 2^{45} operations
  - 4 months on a server
- Security by obscurity does not work
  - A5/1
  - A5/2
  - GEA-1
  - GEA-2
  - Mifare
  - Keeloq
  - DVDCSS
  - ...
- Backdoors affect the security of everybody
  - GEA-1 used outside "export" countries
  - Downgrade attack as long as weak algorithm are implemented
  - Other example: Logjam, exploiting TLS "export" ciphersuites

GEA-1 and GEA-2



Timeline

1999 GPRS specification

2000 GPRS deployment

2001 First commercial 3G deployment (NTT/Japan)

2002 First 3G deployment in Europe

2002 Specification of A5/3 and GEA-3

2007 First iPhone: GPRS-only

2007 3G deployment in 40 countries

2008 iPhone 3G

2009 Rainbow tables for A5/1 [Nohl & al.]
Plans to speed-up transition to A5/3

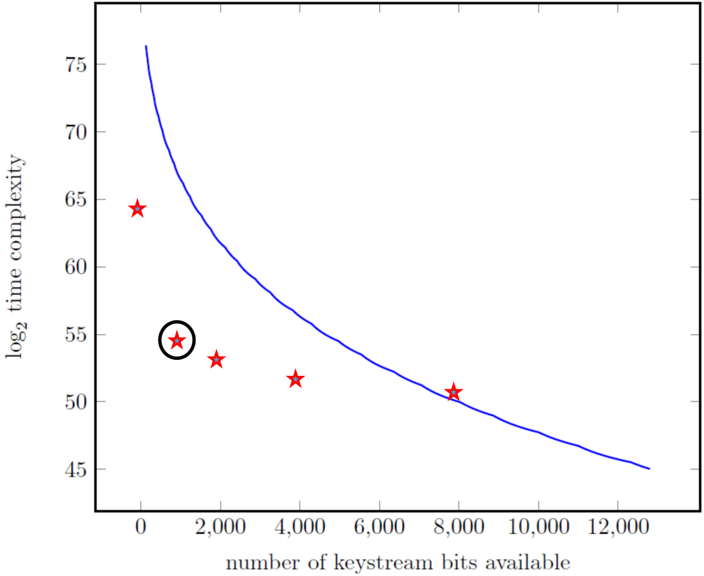
2011 Semi-public analysis of GEA-1 [Nohl & Melette]
GEA-1 and GEA-2 widely used at the time

2013 Deprecation of GEA-1

GEA-1: Reducing memory

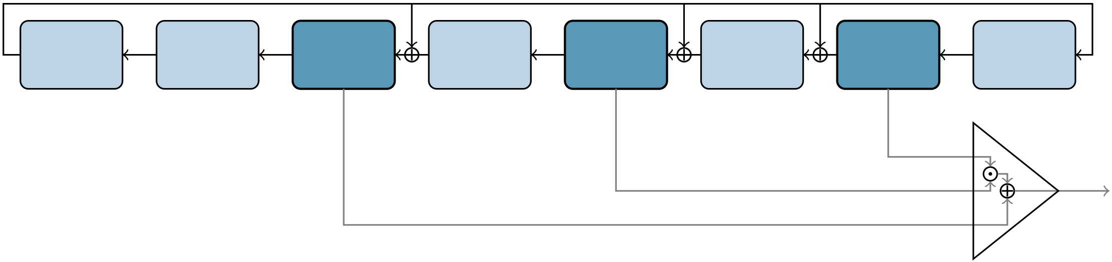
- Memory usage can be reduced significantly [Amzaleg & Dinur, EC'22]
- Reduce memory usage from 2^{32} to 2^{24}
  - (A, C) and (B) are not independent
  - Start by guessing 8 common bits of information
- Further reduce to 2^{19} (4MB) using techniques from 3-XOR cryptanalysis

GEA-2: Time-data tradeoff



- Complexity 2^{45} with full frame (12800 bits)
- Tradeoff with fewer data *(blue line)*
- Better tradeoff with different attack: 4XOR *(stars)* [Amzaleg & Dinur, EC'22]

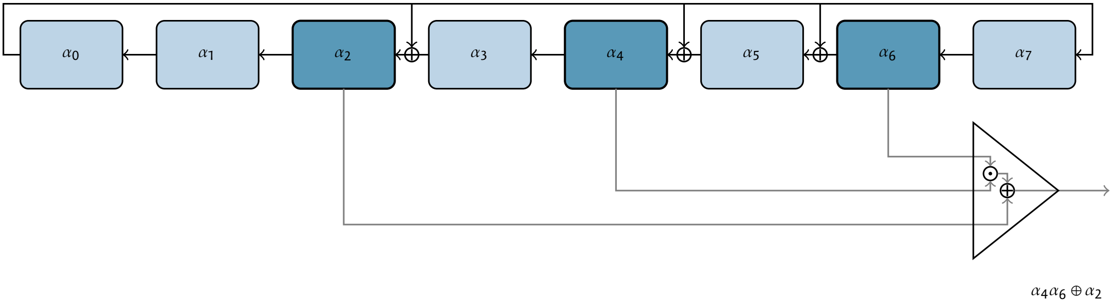
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

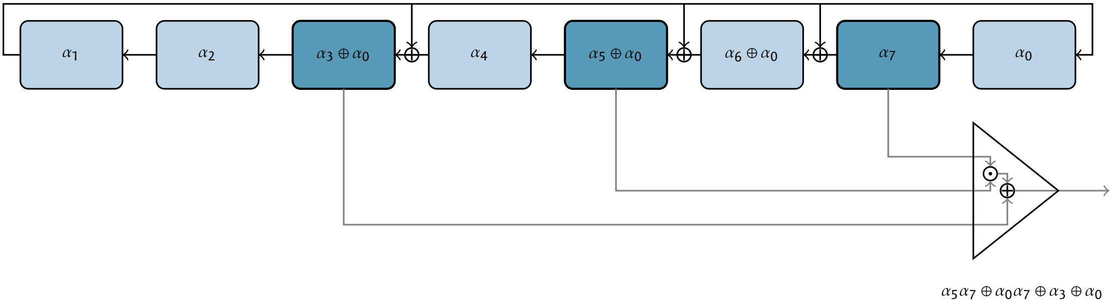
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

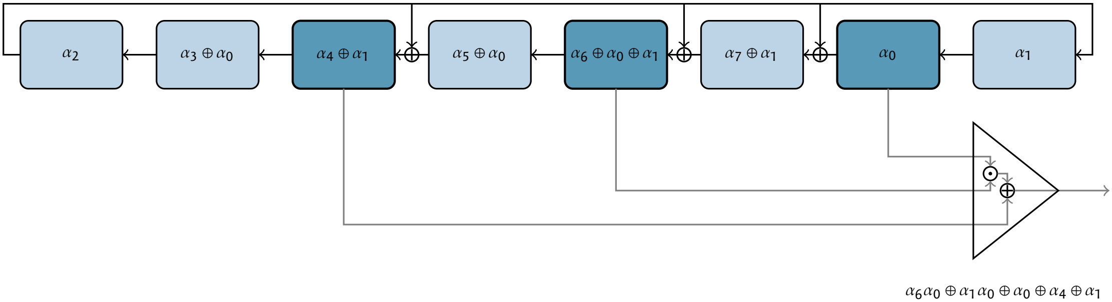
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

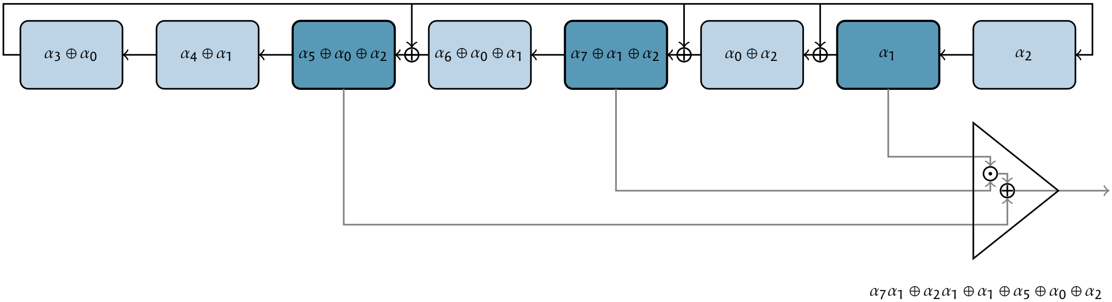
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

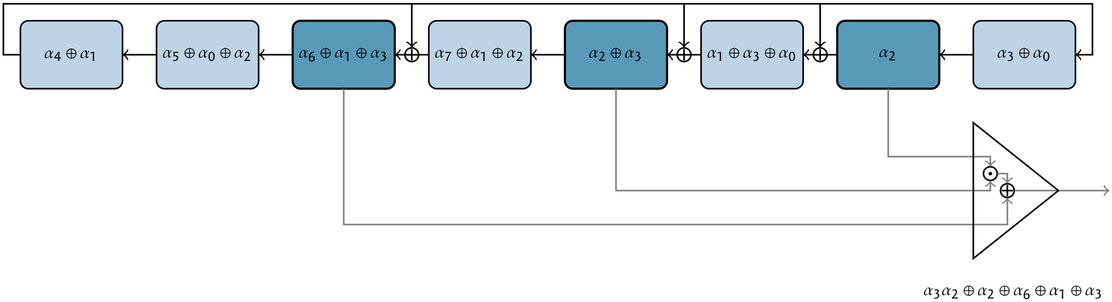
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

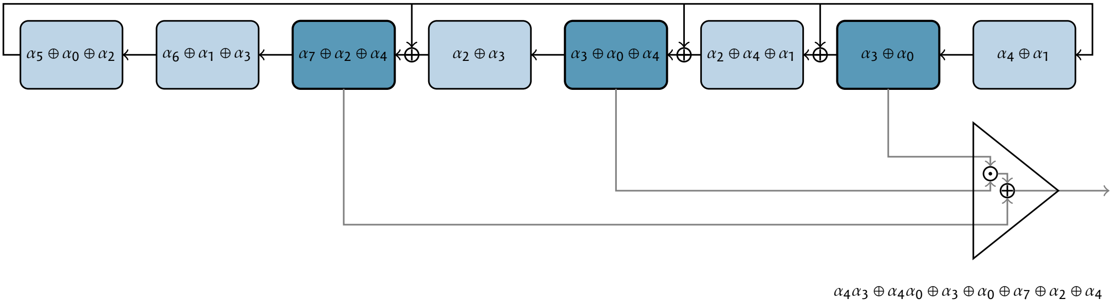
Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back

Toy example



- Filter generator
- Use variables for initial state
- Output can be written as polynomial of the initial state

Go back