EE515 Paper Presentation

Cryptanalysis of the GPRS Encryption Algorithms GEA-1 and GEA-2

C.Beierle, P. Derbez, G. Leander, G. Leurent, H. Raddum, Y. Rotella, D. Rupprecht, L. Stennes

Eurocrypt 2021

20234181 Seungmin Park

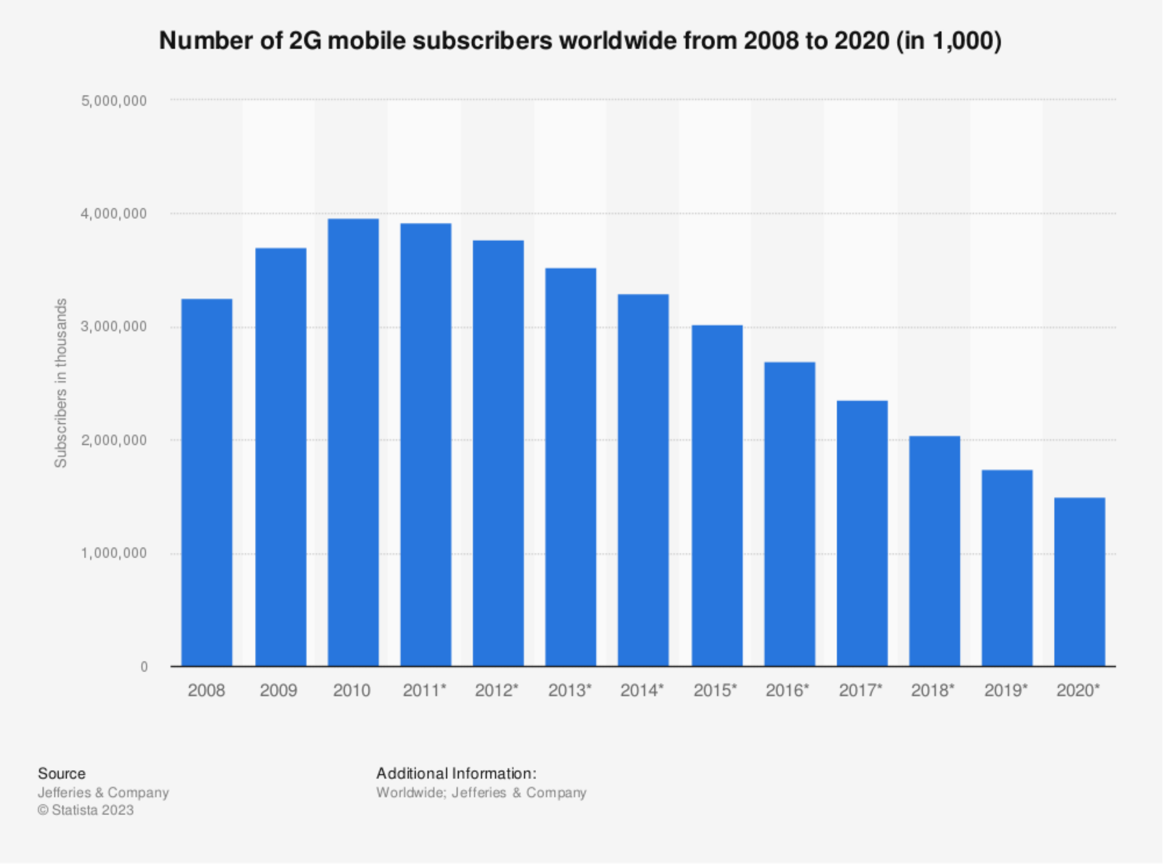
What is GSM and GPRS?

CC by: Rafael Fernandez

GSM (Global System for Mobile Communication, 2G)

GPRS (General Packet Radio Service, 2.5G)

Is 2G Data Connection Still Important?



Past 2G networks [edit]

Country	Network	Shutdown date	Standard
United Kingdom		2033	GSM
Belgium	Orange	2030	GSM
Luxembourg	Orange	2030	GSM
Poland	Orange	2030	GSM
Romania	Orange	2030	GSM
Slovakia	Orange	2030	GSM
Spain	Orange	2030	GSM
Belgium	Telenet	2027	GSM
Belgium	Proximus	2027	GSM
France	Bouygues	2026-12-31	GSM
France	SFR	2026	GSM
France	Orange	2025-12-31	GSM

GEA-1 and GEA-2

Calls (GSM)	Data (GPRS)
A5/1 A5/2	GEA-1 GEA-2

- A proprietary, stream cipher for encrypting GPRS (early 2000s)
- Designed by ETSI Security Algorithms Group of Experts (SAGE) in 1998
- ETSI prohibited the implementation of GEA-1 in 2013 (why?)
- GEA-2 is still mandatory to be implemented
- Still not disclosed or publicly analyzed

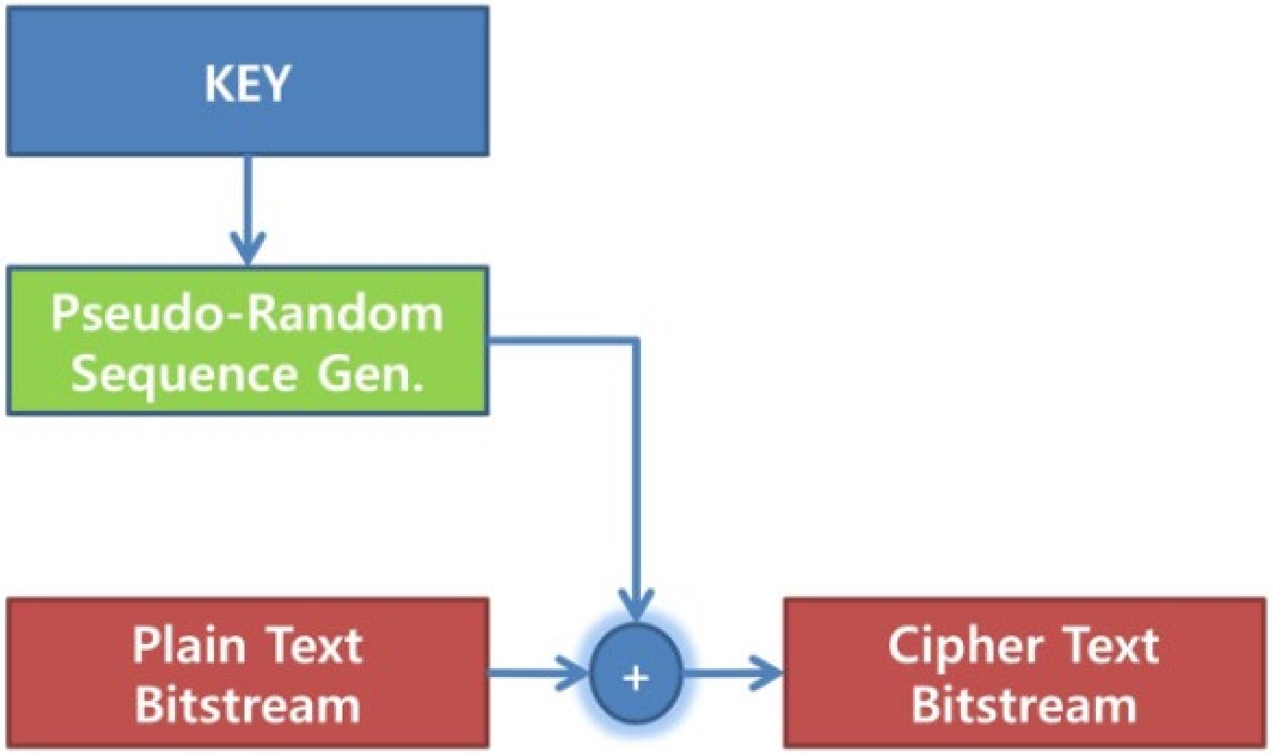
Introduction

- Authors got the source code of GEA-1 and GEA-2 from anonymity
- Both algorithms use 64-bit input key but..
- GEA-1 can be recovered in time 2^{40} GEA-1 evaluations
- GEA-2 still able to break in time $2^{45.1}$ GEA-2 evaluations

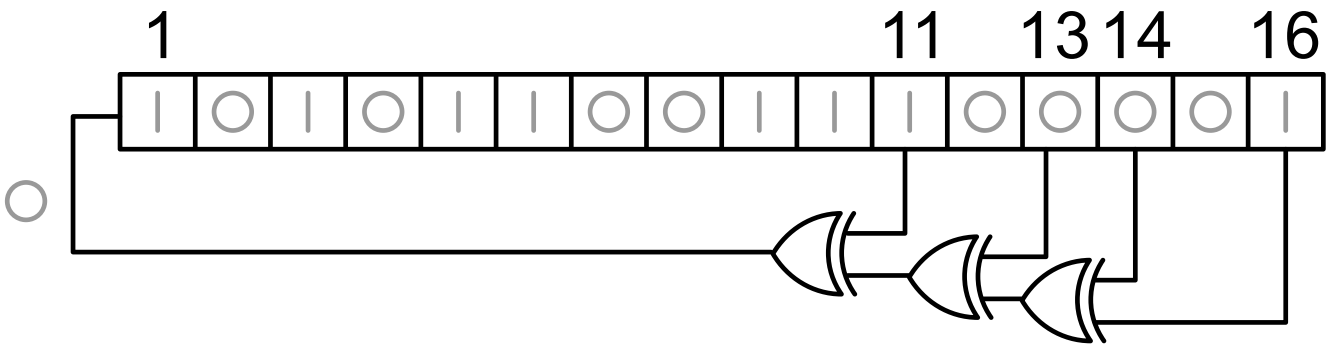
Brute force a 64-bit key needs 2^{64} evaluations!

Background

Stream Cipher

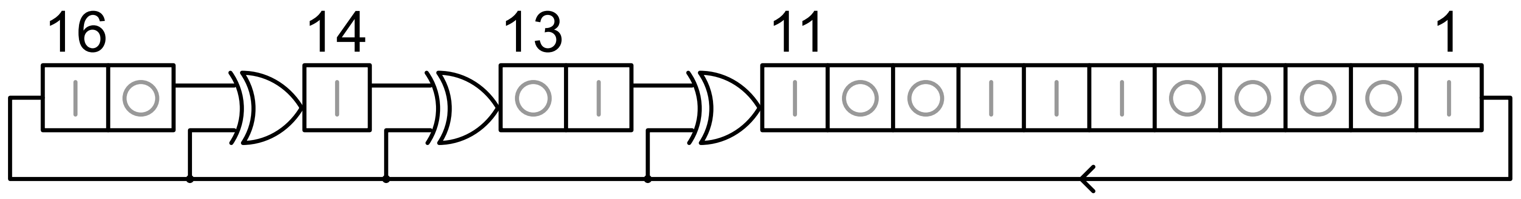


LFSR: Linear-Feedback Shift Register



- Shift register whose next input bit is a linear function of its previous state
- Tap: the bits in LFSR state that influence the input
- Seed: initial state of LFSR
- Maximum period 2^L (L: length of LFSR)

LFSR Galois mode

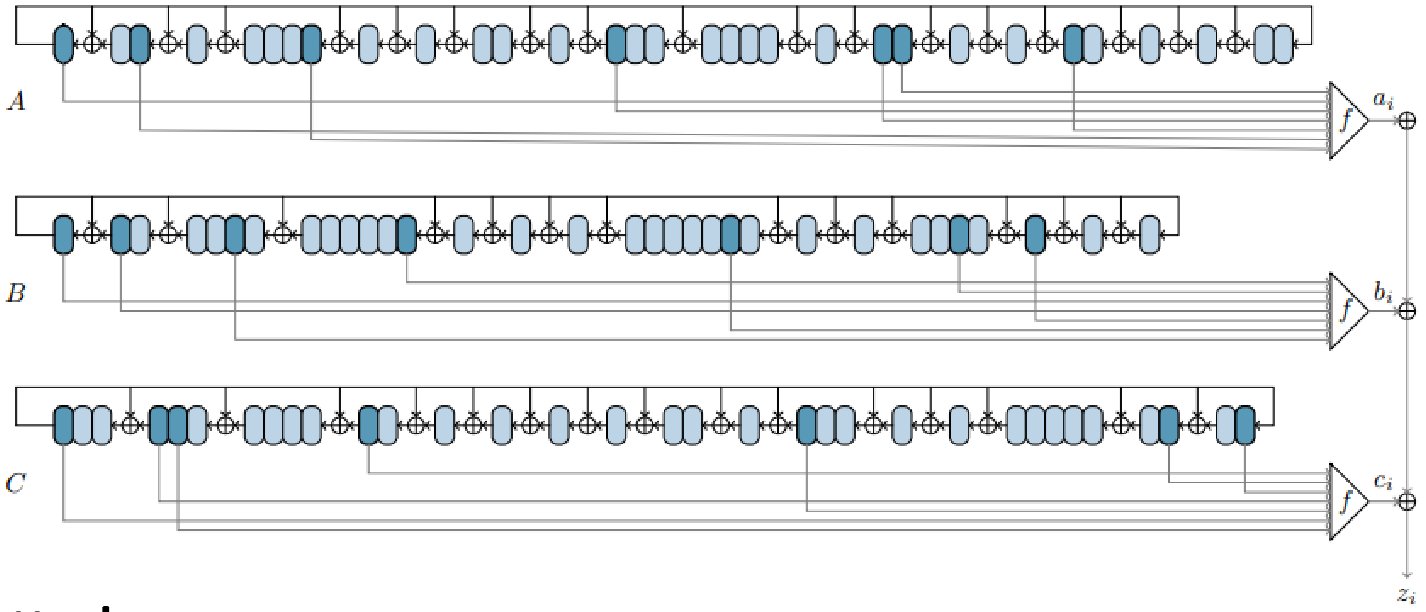


- Output bit 0: just shift to the right
- Output bit 1: bits in the tap positions all flip and then shift to the right
- Well chosen taps makes maximum period LFSR (primitive LFSR)

Cryptanalysis of GEA-1

The Structure of GEA-1 (from source code)

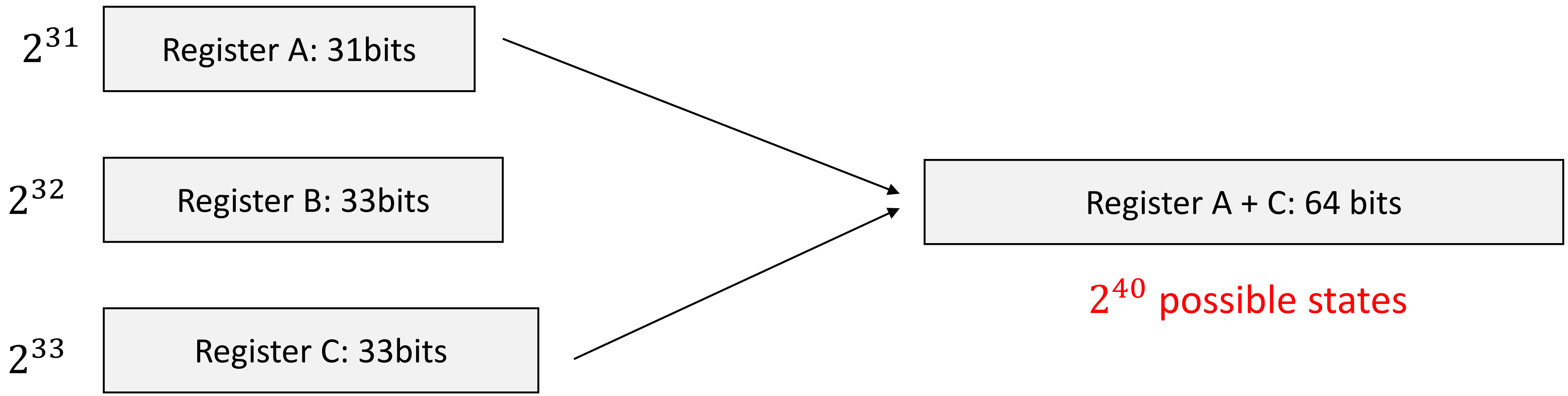
- The 64-bit seed is (linearly) mapped to a 96-bit internal state
- 1600 bytes of keystream $(z_i)_{i\in\{1,\ldots,12800\}}$ are generated by clocking LFSRs



Goal of an attacker

Recover the 64-bit seed (from which we can deduce the 64-bit session key) from some bits of known keystream $(z_i)_{i\in\{1,\ldots,m\}}, m \leq 12800$

The Weakness

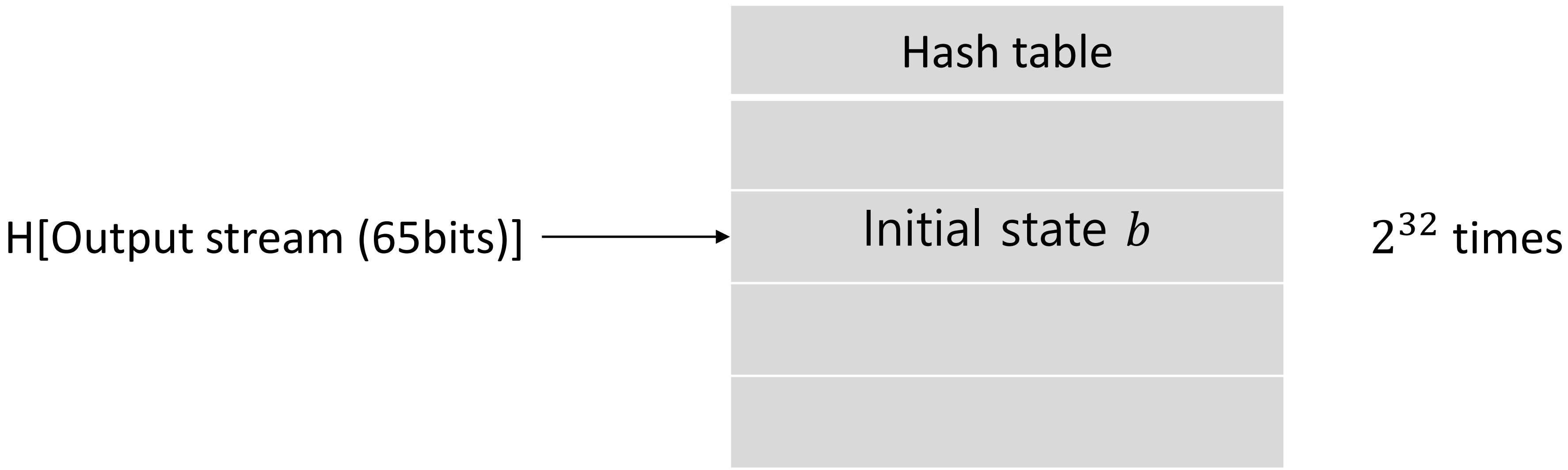


- After the linear initialization process, the joint initial (64-bit) state of registers A and C can only be in a set of 2^{40} possible states

The Attack

Meet-in-the-Middle Attack (Time: 2^{40}, Data: 65 bits of keystream)

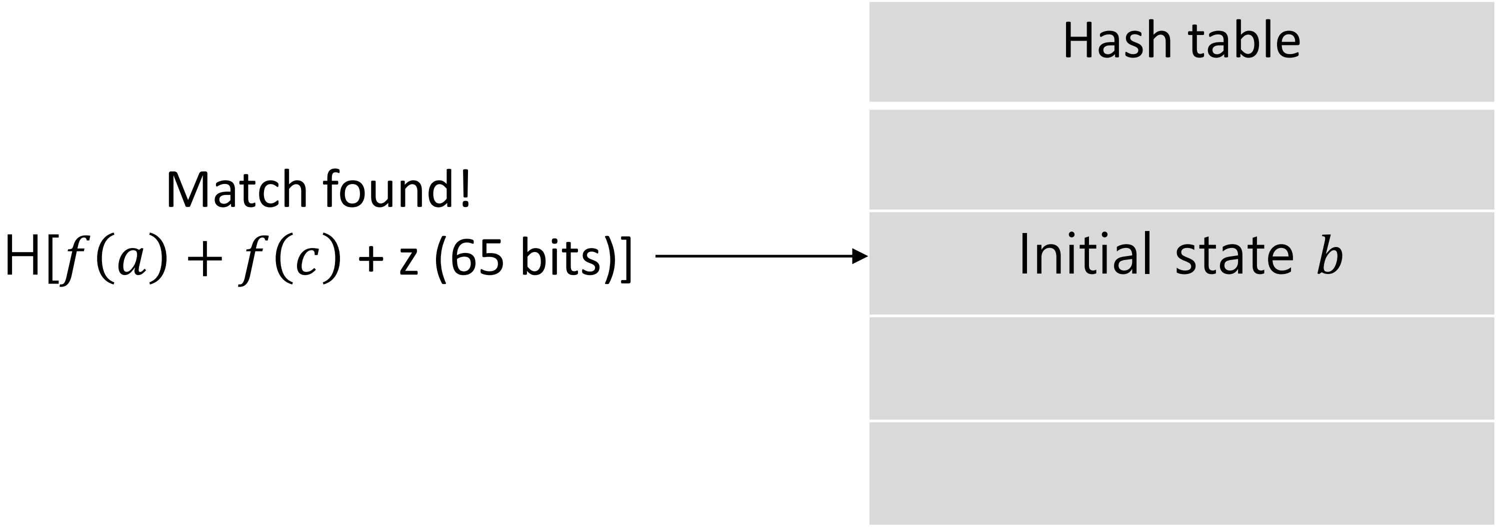
- (Offline step) Store the 65 bits of the output stream $f(b)$ in a hash table for all 2^{32} values of b(initial state of register B), which requires about 44.5 GiB



The Attack

Meet-in-the-Middle Attack (Time: 2^{40}, Data: 65 bits of keystream)

- (Online step) Given the 65 bits of the known keystream z, exhaustively search over the 2^{40} values of (a, c) : joint initial states of register A and C
- Compute $f(a) + f(c)$, and try to find a match for $f(b)$ in the hash table
- Once match is found, we have candidates for the initial register states (a,b,c), and thus a candidate for the 64-bit seed



An Exceptional Property

- Experimentally checked what happens for two random primitive LFSRs (10^6 trials)

Possible states (log_2)	> 58	58	57	56	55	54	53	52
# of spaces	998,027	1,490	366	86	26	5	0	0

- If we assume that these number drop by a factor of 4 in each column, we estimate a probability of 2^{-47} to obtain an image of dimension 40

Fulfilling Export Restrictions as a Design Requirement for GEA-1

To quote from an official document by ETSI from 1998:

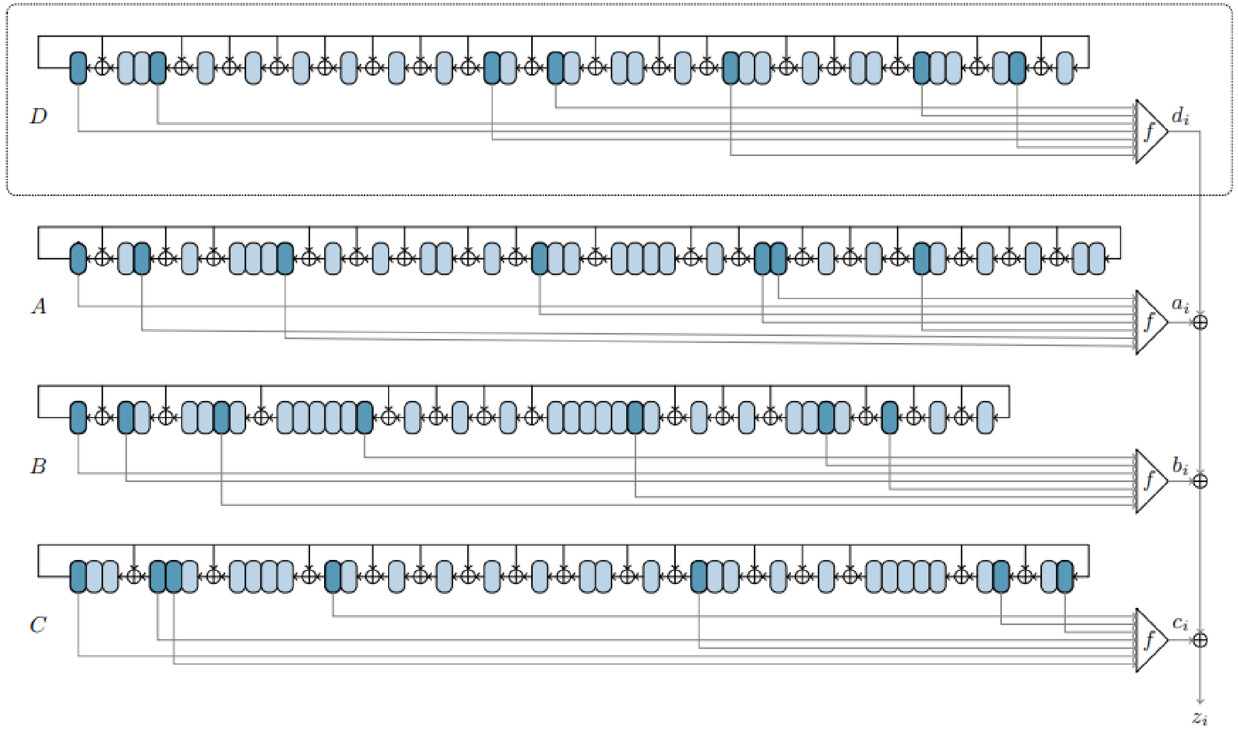
- "the algorithm should be generally exportable taking into account current export restrictions"

Which restrictions exactly?

The official export restrictions are not stated, but there is some indication that it might have been <u>40 bits of security</u>

Cryptanalysis of GEA-2

The Structure of GEA-2 (from source code)



- The idea of targeting the initialization process does not work here
- Idea: Target the keystream generation by a combination of algebraic attacks and list merging

Overview of the Attack on GEA-2

Keystream bit z = f(a) + f(b) + f(c) + f(d)

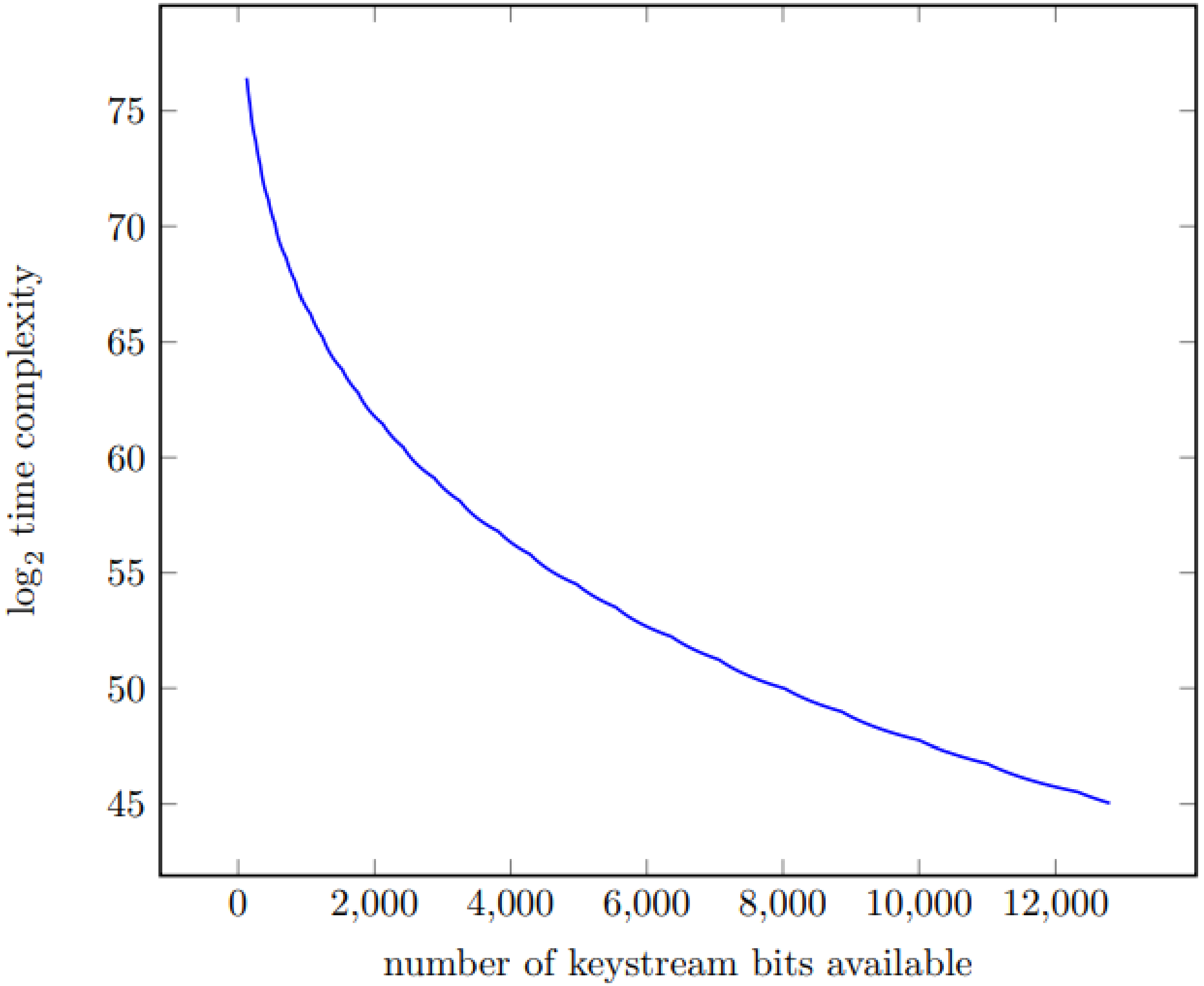
1. Guess n_A bits and n_D bits of the initial state of registers A and D, respectively
2. Construct l many linear equations of keystream bits (only contain guessed bits)
3. Using hash table, find the candidates for registers B and C

Complexity of the Attack on GEA-2

- If we choose $n_A = 11, n_D = 9$, and $l = 64$ we obtain a state-recovery attack with complexity $2^{53.7}$ GEA-2 evaluations and 32 GiB of memory
- (improved version) Roughly $2^{45.1}$ GEA-2 evaluations

Those attacks use all of the available data per frame!

Time Complexity / Data Tradeoff



Responsible Disclosure and Implications

Calls (GSM)	**Data (GPRS)**
A5/1	GEA-1
A5/2	GEA-2

- Most devices (Apple, Samsung, …) support GEA-1 and GEA-2
- GSMA and ETSI Coordinated Vulnerability Disclosure (CVD)
- Now: GEA-1 is disabled in most devices
- Now: Deprecation of GEA-2 in the specification for newer phones

Conclusion

- GEA-1 only offers 40-bit (out of 64) security
- GEA-2 is less weak, but still breakable
- The insecurity of the algorithms has affected out communication until today

Related & Future Work

<- GPRS intercept: Wardriving your country (2011)
(Eavesdropping GPRS traffic & reverse-engineering GEA-1 and GEA-2)

<- ETSI prohibited the implementation of GEA-1 (2013)

<- This paper : EUROCRYPT 2021

<- Refined cryptanalysis of the GPRS ciphers GEA-1 and GEA-2 (2022)

<- New attacks on the GPRS encryption algorithms GEA-1 and GEA-2 (2022)

Q&A: Good questions from students

- Hobin Kim: What are the critical points that allow those publicly available crypto algorithms to be accessed by everyone while maintaining security?
- Kerckhoffs principle: https://en.wikipedia.org/wiki/Kerckhoffs%27s_principle

- Hyeon Heo: Are other cryptographic mechanisms in the real-world totally safe from 'backdoor'? Do there exist concrete steps to prove that a cryptographic algorithm is secure from the 'backdoor'?

Q&A: Best questions from students

- Dongok Kim: Is circuit-level simplicity for these cryptosystems related to their insecurity?

- Kwangmin Kim: I believe that similar backdoor vulnerabilities may exist in other crypto algorithms as shown in this paper. What kind of research is needed to easily find these backdoors?

Thank you!