An Improved Cryptanalysis of Lightweight Stream Cipher Grain-v1

Miodrag J. Mihaljević[1], Nishant Sinha[2], Sugata Gangopadhyay[2],
Subhamoy Maitra[3], Goutam Paul[3] and Kanta Matsuura[4]
[1]Mathematical Institute, Serbian Academy of Sciences and Arts, Belgeade
[2]Indian Institute of Technology, Roorkee
[3]Indian Statistical Institute, Kolkata
[4]The University of Tokyo, Tokyo

- COST CRYPTACUS Workshop -
16-18 November 2017, Nijmegen – Netherlands

Roadmap

- **Part I**:

  Why Grain-v1 is interesting and motivation for the work

- **Part II**:

  Summary of our recent results on Grain-v1 cryptanalysis

- **Part III:**

  Work in progress - An advanced approach for cryptanalysis of Grain-v1

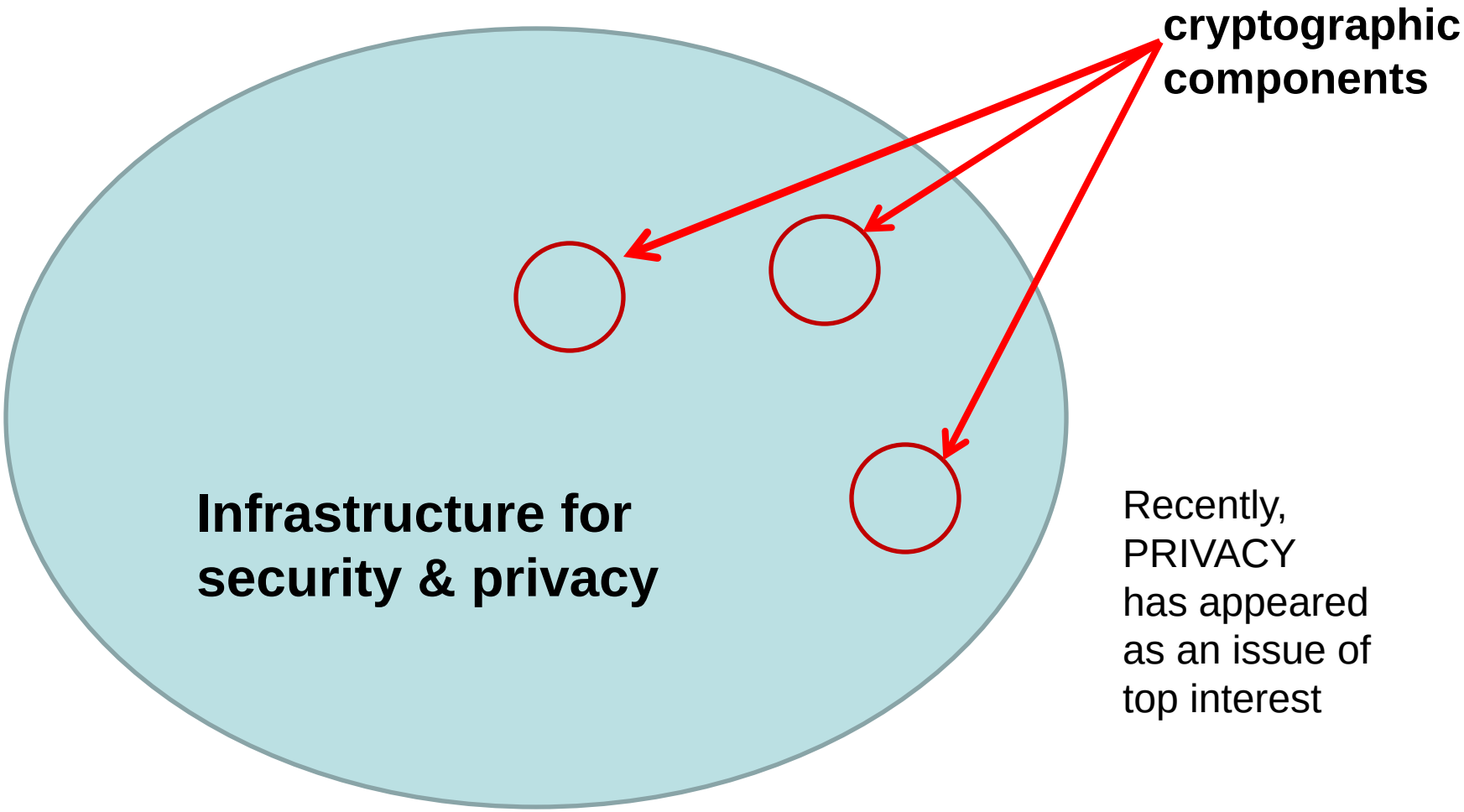
Part I
Why Grain-v1 is interesting and motivation for the work

- Grain family and academic interest

-Lizard: A Grain like lightweight stream cipher reported at FSE 2017

- NIST project on lightweight cryptography (2017)

A bird view (1)



Recently, PRIVACY has appeared as an issue of top interest

A bird view (2): A neverending story

- Development of advanced cryptographic components for information (cyber) security & privacy.
- Development of advanced techniques for security evaluation of cryptographic components for information (cyber) security & privacy.

Motivation for our further work

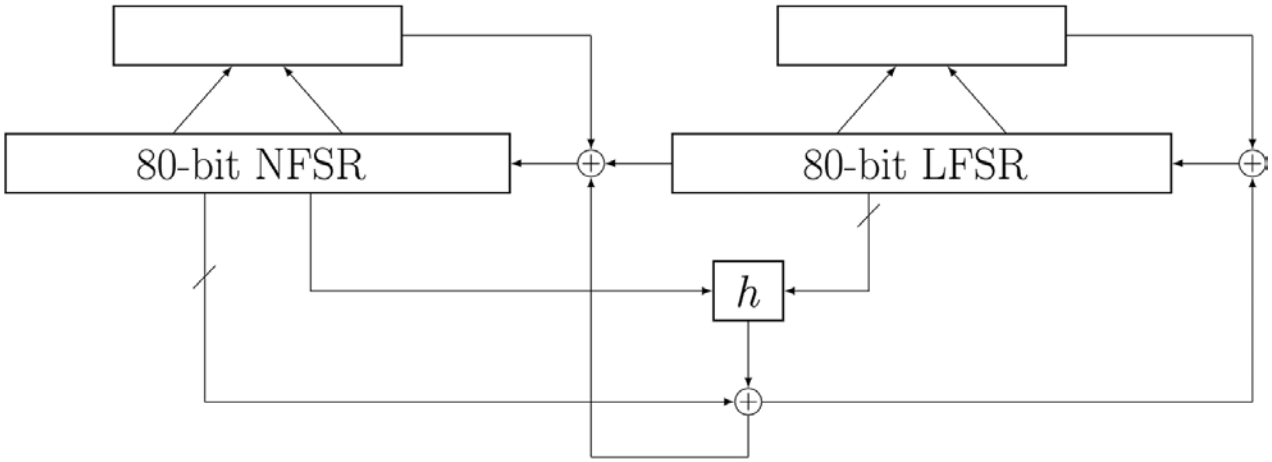
- Grain-v1 is a representative of an interesting and important framework for design of lightweight stream ciphers.
- Security evaluation of Grain-v1 also provides certain guidelines for design of secure lightweight stream ciphers.

Why Grain Family is Interesting

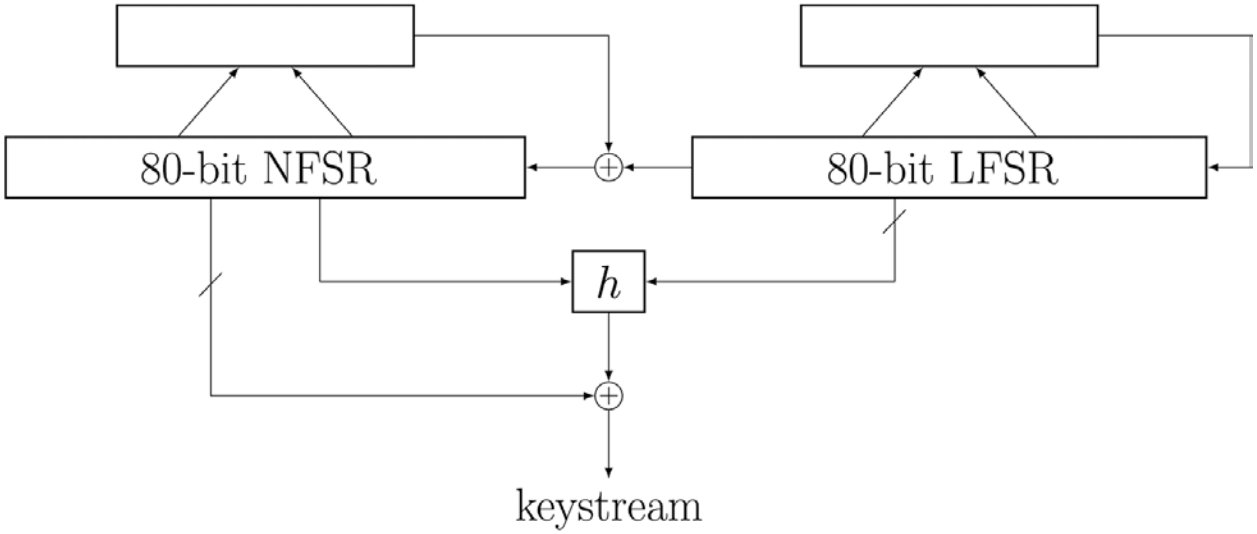
ACADEMIC REFERENCES

Grain-v1: A Member of Grain Family

initialization



keystream generation



Some Recent References

- Z. Ma, T. Tian, W.-F. Qi, "**Improved conditional differential attacks on Grain v1",** *IET Inf. Secur.,* 2017, Vol. 11 Iss. 1, pp. 46-53.
- M. Rahimi, M. Barmshory, M. H. Mansouri, M. R. Aref, "**Dynamic cube attack on Grain-v1**", *IET Inf. Secur*., 2016, Vol. 10, Iss. 4, pp. 165–172.
- S. Banik, "**Conditional differential cryptanalysis of 105 round Grain v1**", *Cryptogr. Commun*. (2016) 8: 113–137.
- Z. Ma, T. Tian, W.-F. Qi, "**Conditional differential attacks on Grain-128a stream cipher",** *IET Inf. Secur.,* 2017, Vol. 11 Iss. 3, pp. 139-145.

Very Recent References: Improvements Originated from Grain Family

- M. Hamann, M. Krause, W. Meier, "**LIZARD – A Lightweight Stream Cipher for Power-constrained Devices**", FSE 2017, to appear in *IACR Transactions on Symmetric Cryptology*.

- E. Dubrova, Martin Hell, "**Espresso: A stream cipher for 5G wireless communication systems**", *Cryptogr. Commun.* (2017) 9: 273–289

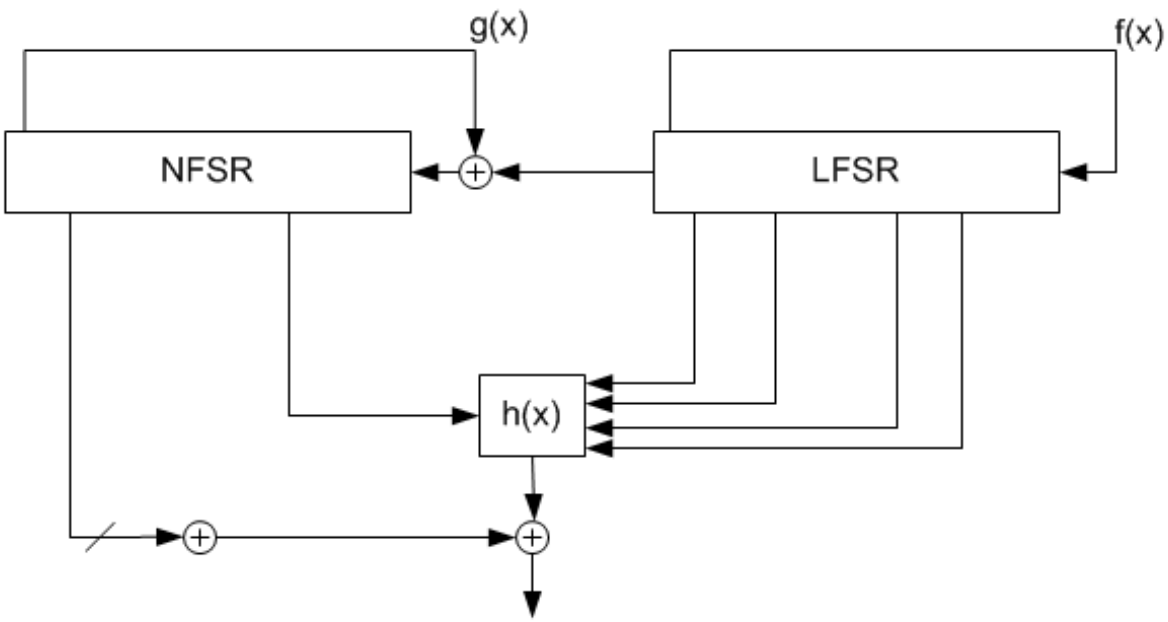
Some of Our References on Grain Family

- M.J. Mihaljevic, S. Gangopadhyay, G. Paul and H. Imai, "**State Recovery of Grain-v1 Employing Normality Order of the Filter Function**", *IET Information Security*, vol. 6, no. 2, pp. 55-64, June 2012.
- M.J. Mihaljevic, S. Gangopadhyay, G. Paul and H. Imai, "**Generic Cryptographic Weakness of k-normal Boolean Functions in Certain Stream Ciphers and Cryptanalysis of Grain-128**", *Periodica Mathematica Hungarica*, vol. 65, no. 2, pp. 205-227, Dec. 2012.
- M.J. Mihaljevic, N. Sinha, S. Gangopadhyay , S. Maitra, G. Paul, K. Matsuura, “**Internal State Recovery of Grain-v1 Stream Cipher Employing Conditional Time-Memory-Data Trade-Off**”, to be submitted.

Why Grain Family is Interesting

AN ORIGIN FOR ADVANCED DESIGNS

Grain-v1 Keystream Generator



LIZARD – A Lightweight Stream Cipher for Power-constrained Devices

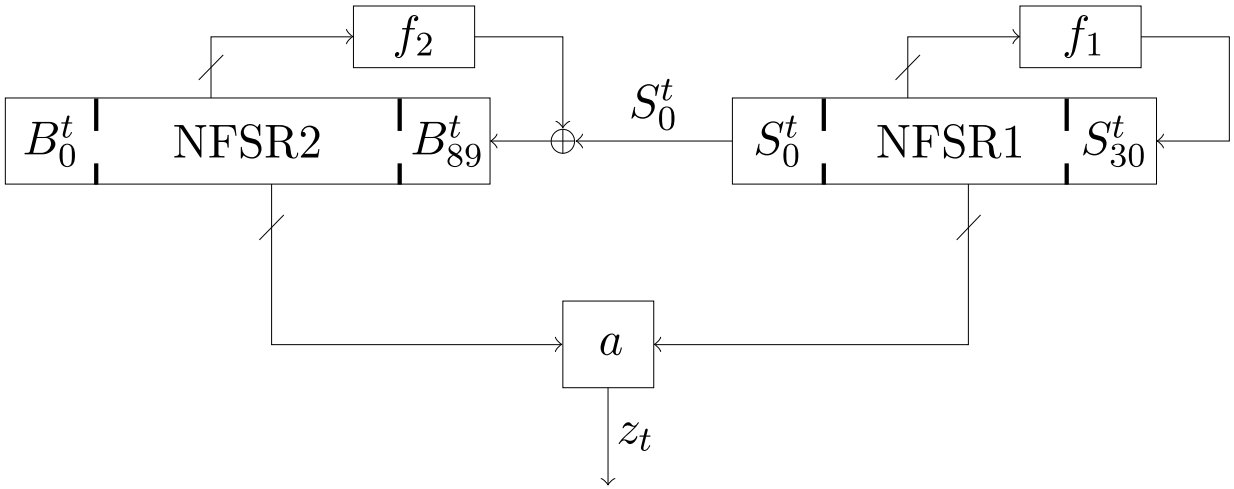
Matthias Hamann[1] Matthias Krause[1] Willi Meier[2]

[1] University of Mannheim, Germany [2] FH Nordwestschweiz, Switzerland

Difference to Grain v1

- **Smaller state size: 121 bit** (compared to 160 bit).
- **Larger key size: 120 bit** (rather than 80 bit), necessary assumption for security proof.
- **Key is introduced not only once, but twice in initialization.**
- Quite different **output function: Inspired by FLIP stream cipher**, uses many (53) inputs.
- Both register feedbacks now nonlinear.
- Efficiently parallelizable up to a factor of 6 (compared to 16).

LIZARD in keystream generation mode



- **NFSR1 (length 31 bit)** has guaranteed period $2^{31} - 1$ (from ACHTERBAHN stream cipher).
- **NFSR2 (length 90 bit)** keeps same cryptographic properties as NFSR in Grain-128a.

Hardware Results

Design	Area [GE]	Power [nW]	Delay [ps]	Load/Ini [clk. cyc.]
LIZARD*	1161	2110	2474	499
Grain v1*	1268	2517	2155	241
Grain v1	1221	3578	2166	161

- Clock speed of **100 kHz**.
- ***** indicates **serialized key/IV loading**.
- **Load/Ini**: Number of clock cycles needed to perform the state initialization.
- After state initialization, all designs produce one keystream bit per clock cycle (i.e., **100 kbit/s**).

Why Grain Family is Interesting

NIST RECOGNITION

NIST Lightweight Cryptograpy Project

NISTIR 8114

Report on Lightweight Cryptography

Kerry A. McKay
Larry Bassham
Meltem Sönmez Turan
Nicky Mouha
Computer Security Division
Information Technology Laboratory

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8114

March 2017

U.S. Department of Commerce
Wilbur L. Ross, Jr., Secretary

National Institute of Standards and Technology
Kent Rochford, Acting NIST Director and Under Secretary of Commerce for Standards and Technology

NISTIR 8114 REPORT ON LIGHTWEIGHT CRYPTOGRAPHY

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8114

- **Stream ciphers are also promising primitives for constrained environments**. The eSTREAM competition, organized by the European Network of Excellence for Cryptology, aimed to identify new stream ciphers that might be suitable for widespread adoption. The finalists of the competition were announced in 2008 and included three stream ciphers for hardware applications with restricted resources:
- ***Grain is widely analyzed and provides implementation flexibility, and also has a version that supports authentication.***
- ***Trivium is a widely analyzed design; however, it only supports 80-bit keys.***
- ***Mickey is less analyzed compared to Grain and Trivium. It provides less implementation flexibility and is susceptible to timing and power analysis, due to irregular clocking.***

- *Authenticated Encryption Algorithms and MACs:* Authenticated encryption algorithms provide performance and resource requirement advantages, because they simultaneously provide confidentiality and integrity protection of messages. NIST approves the CCM [18] and GCM [17] block cipher modes that provide authentication and encryption simultaneously. NIST also approves standalone MACs, CMAC [19], GMAC [17], and HMAC [68], to be used for generating and verifying tags to provide message authentication.

2.5 Lightweight Cryptography Standards

ISO/IEC 29192, *Lightweight Cryptography*, is a six-part standard that specifies lightweight cryptographic algorithms for confidentiality, authentication, identification, non-repudiation, and key exchange. Part 1 includes general information such as security, classification and implementation requirements [39]. Part 2 specifies the block ciphers PRESENT and CLEFIA [40]. An amendment to Part 2 was proposed in 2014 to include the block ciphers SIMON and SPECK [6] with various block and key size combinations. In 2015, the first working drafts of the amendments with SIMON and SPECK were initiated. Part 3 specifies the stream ciphers Enocoro and Trivium [41]. Part 4 specifies three asymmetric techniques, namely (*i*) identification scheme cryptoGPS, (*ii*) authentication and key exchange mechanism ALIKE, and (*iii*) ID-based signature scheme IBS [42]. An amendment to Part 4 included an Elliptic Curve-based authentication scheme called ELLI [43]. Part 5 specifies three hash functions: PHOTON, SPONGENT, and Lesamnta-LW[44]. Part 6 is dedicated to MACs and is currently under development [45].

ISO/IEC 29167, *Automatic Identification and Data Capture Techniques*, provides security services for RFID air interface communications. Part 1 [29] describes the architecture, security features, and requirements for security services for RFID devices. Crypto suites are defined in additional parts. Currently, eight suites that specify the use of AES-128, PRESENT-80, ECC-DH, Grain-128A, AES OFB, ECDSA-ECDH, cryptoGPS, and RAMON security services for air interface communication are published in [30-37]. Additional documents are under development.

Cryptography Research and Evaluation Committees (CRYPTREC) is a project to evaluate and monitor security of cryptographic techniques used in Japanese e-Government systems [13]. CRYPTREC publishes three types of cipher lists: e-Government Recommended Ciphers List, Candidate Recommended Ciphers List and Monitored Ciphers List. The Lightweight Cryptography working group of CRYPTREC, established in 2013, aims to study and support appropriate lightweight cryptography solutions for e-government systems and any applications where lightweight solutions are needed. The working group surveys research on the state of the art in lightweight cryptography and its applications, performs implementation evaluations, and published a report (in Japanese) [14] as a deliverable in 2015. The target algorithms for implementation in the report were AES, Camellia [1], CLEFIA [63], PRESENT [9], LED [25], Piccolo [62], TWINE [65], and PRINCE [10].

3 NIST's Lightweight Cryptography Project

NIST develops standards using several different approaches, as described in [53]. NIST has held competitions to select the AES block cipher and the SHA-3 hash functions. These competitions were significant efforts that took place over many years. For example, the SHA-3 competition was announced in 2007, the winner was announced in 2012, and the standardization process was concluded in 2015. Another approach is to adapt standards of other accredited standards development organizations, as was done with HMAC and RSA standards. NIST researchers also develop standards and guidelines in collaboration with experts in academia, industry and government, if no suitable standard exists.

The landscape for lightweight cryptography is moving so quickly that a standard produced using the competition model is likely to be outdated prior to standardization. Therefore, the most suitable approach for lightweight cryptography, in terms of timeline and project goals, is to develop new recommendations using an open call for proposals to standardize algorithms.

NIST is planning to develop and maintain a portfolio of lightweight algorithms and modes that are approved for limited use. Each algorithm in the portfolio will be tied to one or more *profiles*, which consist of algorithm goals and acceptable ranges for metrics. This is in contrast to other primitives and modes that are approved for general use. Any restrictions on use will be included in the recommendation or standard where the primitives and modes of the portfolio are specified. Algorithm transitions and deprecation guidance will be provided as algorithms in the portfolio are phased out. The lightweight portfolio is not intended to offer alternative algorithms for general use.

3.1 Scope

The scope of NIST's lightweight cryptography project includes all cryptographic primitives and modes that are needed in constrained environments. However, the initial focus of the project is on block ciphers, authenticated encryption schemes, hash functions, message authentication codes, cryptographic permutations, and stream ciphers. When long-term security is needed, these algorithms should either aim for post-quantum security [54], or the application should allow them to be easily replaceable by algorithms with post-quantum security.

While public key cryptography is not included in the initial focus, it is within the scope of this project. However, it should be noted that public key schemes will only be considered for inclusion in the portfolio under two conditions: 1) they are robust against quantum attacks, and 2) they use a combination of general public key cryptographic schemes with lightweight primitives (e.g., a lightweight hash function). Protocol design is also an important part of achieving the desired level of security while meeting requirements of a constrained environment, but protocol standardization is not within the scope of this project.

3.2 Design Considerations

While specific requirements vary by application, there are several generally-desired properties that NIST will be using to evaluate designs.

- ***Security strength***: Any algorithm selected for the portfolio must provide adequate security. In general, the security strength should be at least 112 bits.

- ***Flexibility***: Efficient implementations of an algorithm should be possible across an assortment of platforms. Algorithms should also allow a variety of implementations on a single platform. Tunable algorithms, which use parameters to select properties such as state size and key size, are desirable as they allow implementations with multiple options using fewer resources than multiple algorithms that do not share logic, thereby supporting a wider array of applications.

- ***Low overhead for multiple functions***: Multiple functions (such as encryption and decryption) that share the same core are preferred over functions that have completely different logic. For example, a block cipher where the encryption and decryption operations use similar round functions may be preferable over one that has distinct round functions for encryption and decryption. Different primitives, such as a hash function and block cipher, can also share logic, thus reducing the resources needed to implement multiple algorithms in the same device.

- ***Ciphertext expansion***: The size of the ciphertext has an impact on storage and transmission costs. Algorithms and modes that do not generate a ciphertext that is significantly longer than the plaintext are desirable.

- ***Side channel and fault attacks***: Implementations can leak sensitive information, particularly information about the key or plaintext, in a variety of ways. Side channel attacks use properties of the implementation during execution of the cryptographic operations, such as timing, power consumption, and electromagnetic emissions, to discover this sensitive information. Fault attacks recover this sensitive information by introducing errors in the computation. In the case of pervasive devices, this is particularly notable as attackers may have physical access to the devices, and countermeasures for such attacks may not be present due to constrained resources. Algorithms that are easy to protect against side channel and fault attacks are desirable.

- ***Limits on the number of plaintext-ciphertext pairs***: It may be permissible for algorithm designers to assume an upper bound on the number of plaintext/ciphertext pairs processed, as this limit can be justified for some applications by the constraints of the devices (e.g., limitations on the amount of data that are processed by the same key), or by message formats defined by protocols. However, it must be recognized that an attacker may mount attacks using plaintext that was encrypted under multiple, independent keys (multi-key attacks), which are relevant even when the amount of data encrypted under any single key is limited.

- ***Related-key attacks***: These attacks allow an adversary to discover information about a key by performing operations using multiple unknown keys that have a known relation. This is particularly a threat in protocols where keys are not chosen independently and at random. Resistance to related key attacks may be desirable for some applications.

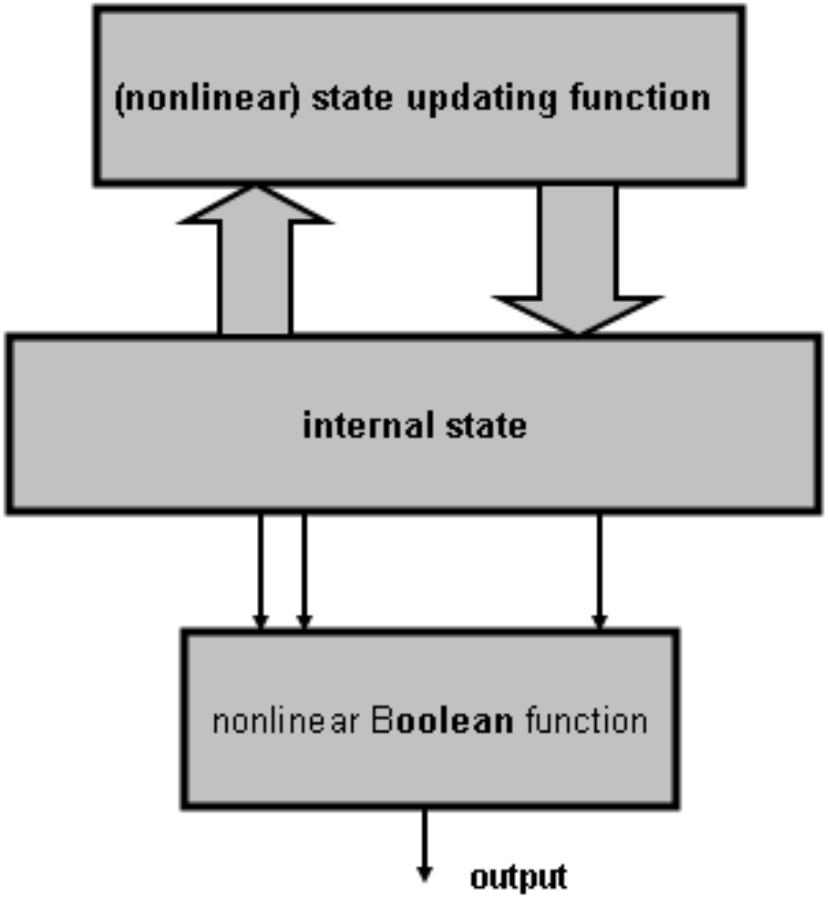
It may not be possible to satisfy all properties, in particular when this increases the resources beyond what is available for a given application. Still, any algorithm selected for the portfolio must provide adequate security. In particular, the security against key-recovery attacks should be at least 112 bits.

Part II

A summary of our recent results on recent

Grain-v1 cryptanalysis

Considered Model of Stream Ciphers



Underlying Ideas for Cryptanalysis

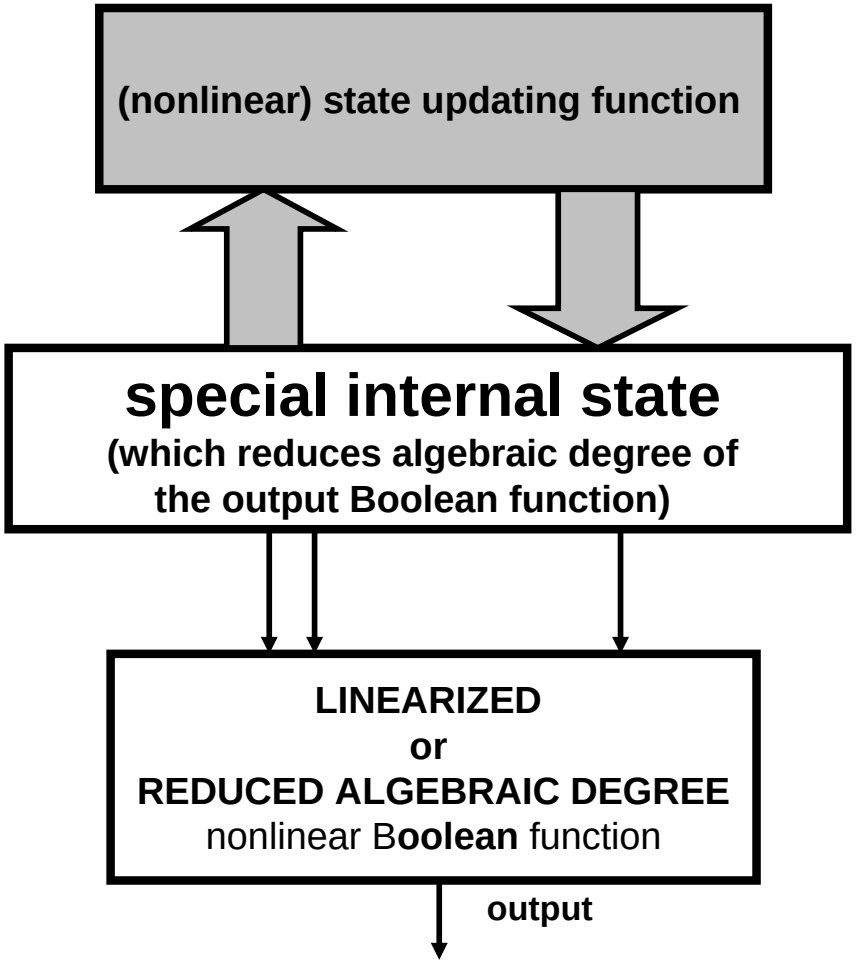
- employment of a number of different linearizations or the algebraic degree reductions of the nonlinear function;

- employment of a guess&determine approach;

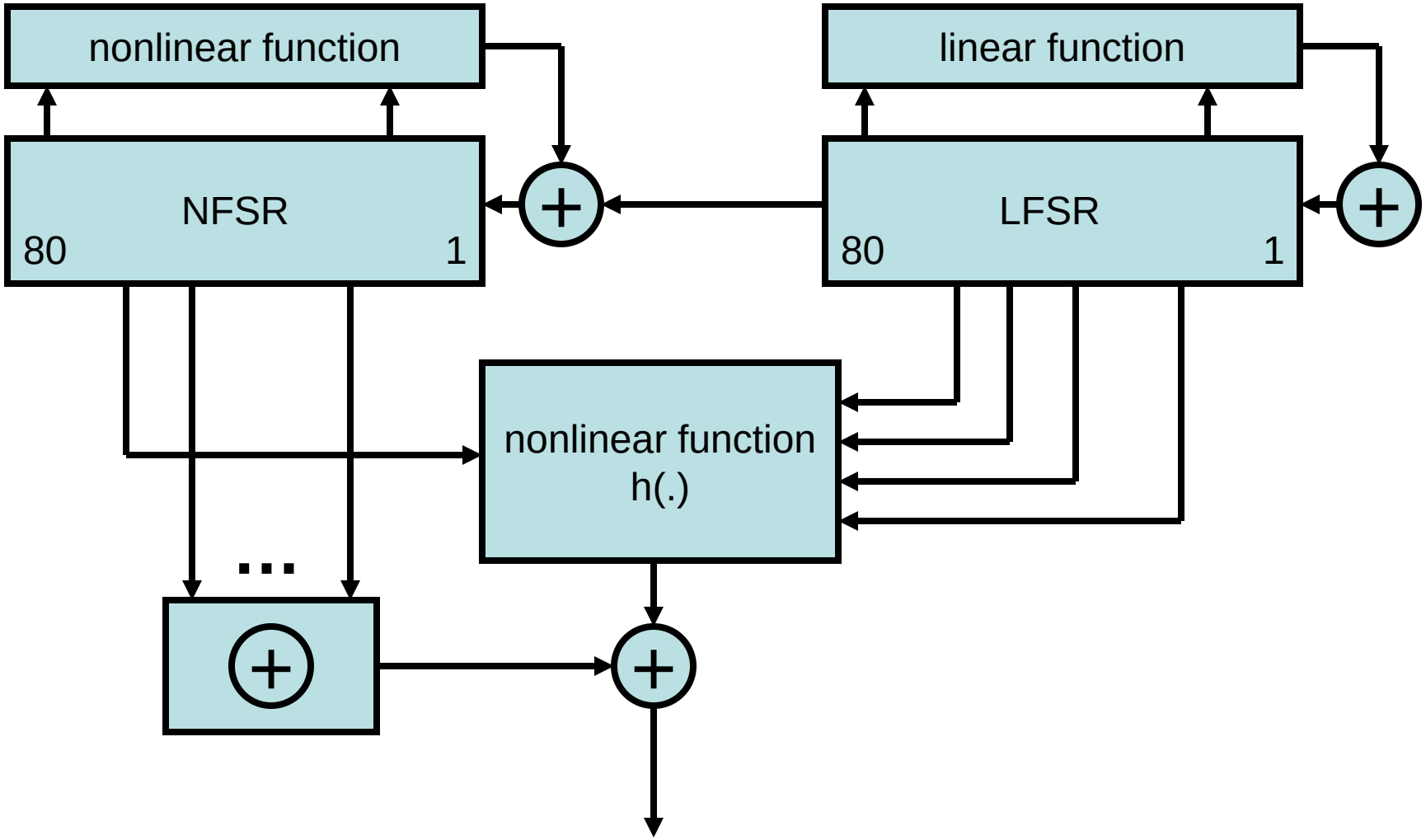
- employment of a dedicated TM-TO paradigm for testing a hypothesis for each of available L-bit segment of the given n-bit keystream;

- reusing of a given sample for each of the employed different linearizations or the algebraic degree reductions.

Linearized Model



Grain-v1 Keystream Generator



Algebraic Description of Grain-v1

The keystream is computed as

$$z_i = (\bigoplus_{k \in \mathcal{A}} b_{i+k}) \oplus h(s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64}, b_{i+63})$$

where $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$, and

$$\begin{aligned} h(x) \;=\; & x_1 \oplus x_4 \oplus x_0x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_0x_1x_2 \\ & \oplus x_0x_2x_3 \oplus x_0x_2x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4, \end{aligned}$$

where the variables x_0, x_1, x_2, x_3 and x_4 correspond to the tap positions s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64} of the LFSR and b_{i+63} of the NFSR, respectively.

The proposed cryptanalysis is based on the following approach:

- employment of a **dedicated restricted guess and determine** approach;
- employment of a **dedicated BSW sampling** which provides efficient recovery of a part of the internal state (under a dedicated restricted guess) based on the given keystream segment;
- employment of a **dedicated time-memory trade-off** approach.

Probabilistic Background (1)

Specification of a dedicated restricted guess and determine approach and a dedicated BSW sampling is based on the following. We consider cryptanalysis over a subset of internal states which fulfil certain characteristic.
Let $\Omega^{(m)}$ be a set of the internal states $\mathbf{u}$ such that

$$\Pr(\mathbf{Z}^{(m)} = \mathbf{z}^{(m)} | \mathbf{u} \in \Omega^{(m)}) = 1,$$

where a state from $\Omega^{(m)}$ generates a sequence with the m-bit prefix $\mathbf{z}^{(m)}$.

Probabilistic Background (2)

$$\Pr(\mathbf{u} \in \Omega^{(m)} | \mathbf{Z}^{(m)} = \mathbf{z}^{(m)}) = \frac{\Pr(\mathbf{u} \in \Omega^{(m)} | \mathbf{Z}^{(m)} = \mathbf{z}^{(m)})}{\Pr(\mathbf{Z}^{(m)} = \mathbf{z}^{(m)})}$$

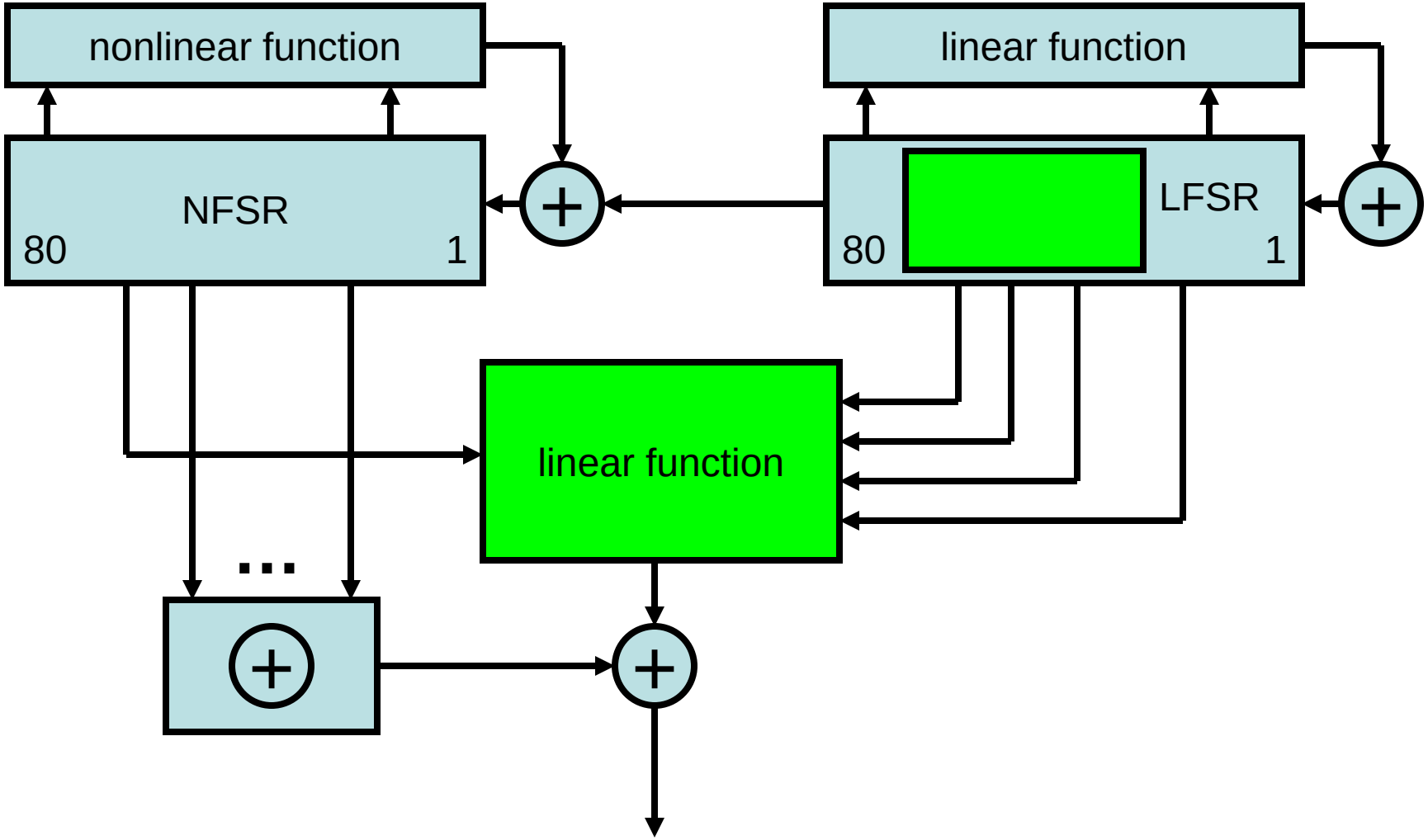
$$= \frac{\Pr(\mathbf{Z}^{(m)} = \mathbf{z}^{(m)}) | \mathbf{u} \in \Omega^{(m)}) \cdot \Pr(\mathbf{u} \in \Omega^{(m)})}{\Pr(\mathbf{Z}^{(m)} = \mathbf{z}^{(m)})}$$

$$= \frac{\Pr(\mathbf{u} \in \Omega^{(m)})}{\Pr(\mathbf{Z}^{(m)} = \mathbf{z}^{(m)})} > \Pr(\mathbf{u} \in \Omega^{(m)}). \qquad (1)$$

Accordingly, giving $\mathbf{z}^{(m)}$, a bias exists towards a subset of the states $\Omega^{(m)}$ and we employ it for design of the improved algorithm and to evaluate its performance.

An important issue is efficient construction of a particular subset of the states $\Omega^{(m)}$: For this purpose we employ k-normality of the employed filter function.

Towards Internal State Recovery: Guess & Linearise



“Enforcing” h(.) to be linear

When $x_2 = 0$ and $x_3 = 0$

$$\begin{aligned} h(x) &= x_1 \oplus x_4 \oplus x_0x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_0x_1x_2 \\ &\quad \oplus x_0x_2x_3 \oplus x_0x_2x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4 \\ &= x_1 \oplus x_4, \end{aligned}$$

where the variables x_0, x_1, x_2, x_3 and x_4 correspond to the tap positions s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64} of the LFSR and b_{i+63} of the NFSR, respectively, and the keystream is computed as

$$z_i = (\bigoplus_{k \in \mathcal{A}} b_{i+k}) \oplus (s_{i+3} \oplus b_{i+63})$$

where $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$.

“Enforcing” h(.) to be constant (a consequence of its k-normality)

When $x_1 = 0$, $x_3 = 0$ and $x_4 = 0$,

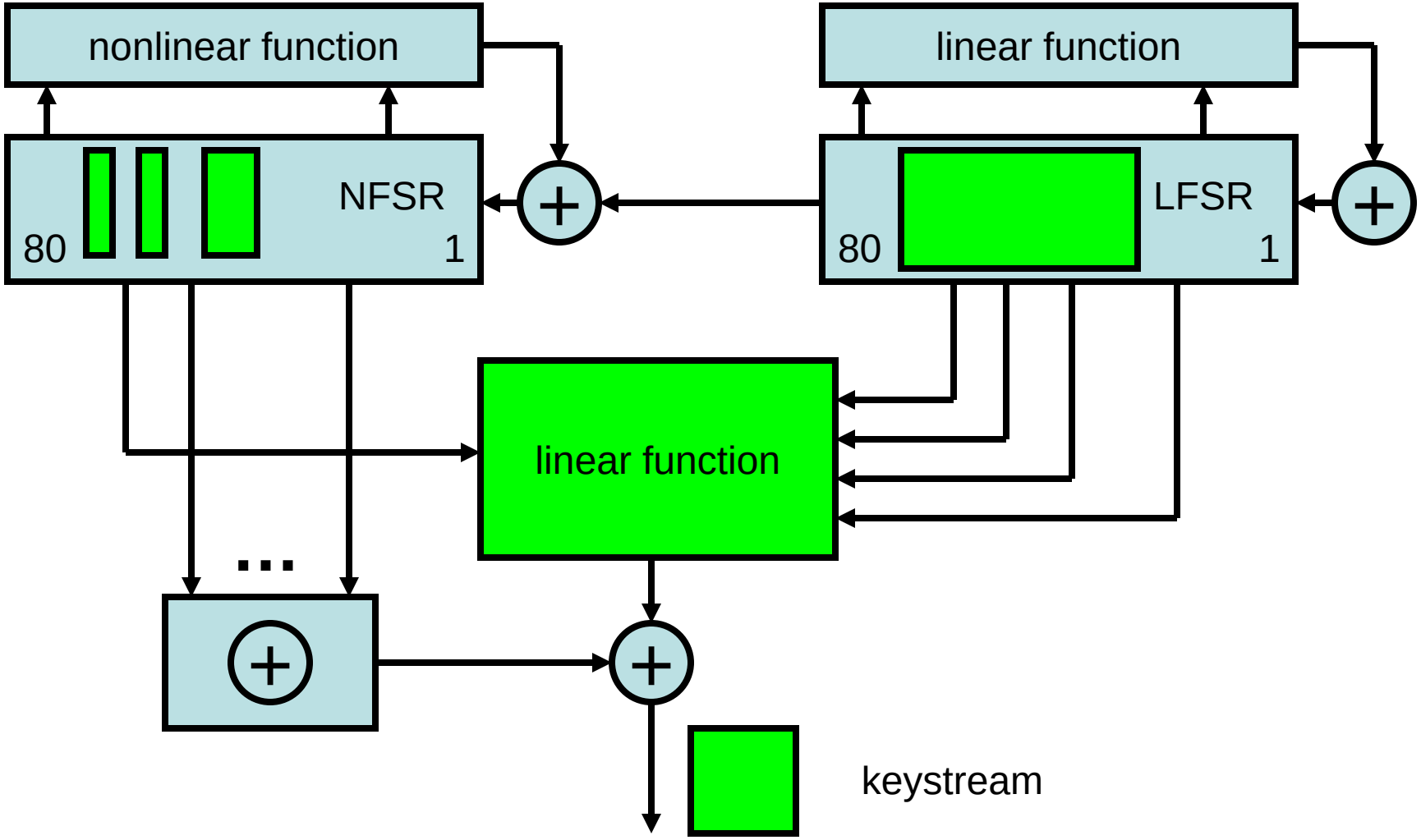
$$\begin{aligned} h(x) &= x_1 \oplus x_4 \oplus x_0x_3 \oplus x_2x_3 \oplus x_3x_4 \oplus x_0x_1x_2 \\ &\quad \oplus x_0x_2x_3 \oplus x_0x_2x_4 \oplus x_1x_2x_4 \oplus x_2x_3x_4 \\ &= 0, \end{aligned}$$

where the variables x_0, x_1, x_2, x_3 and x_4 correspond to the tap positions s_{i+3}, s_{i+25}, s_{i+46}, s_{i+64} of the LFSR and b_{i+63} of the NFSR, respectively, and the keystream is computed as

$$z_i = (\bigoplus_{k \in \mathcal{A}} b_{i+k})$$

where $\mathcal{A} = \{1, 2, 4, 10, 31, 43, 56\}$.

BSW Sampling and a Part of Internal State Recovery under Linearization Guess



Framework for Advanced Cryptanalysis (1)

Assuming that the linearization has been performed, cryptanalysis of the considered model of stream ciphers employs the following framework which is based on the guess&determine and Time-Memory Trade-Off (TM-TO) which employs n-bit output of the considered stream cipher:

- Pre-Processing which should be done only once and provide background for recovering a stream cipher state:

1. identify J different patterns $\mathbf{u}_j^{(\ell_j)}$ on certain ℓ_j, $2^{\ell_j} \leq \frac{n}{J}$, internal state bits, $j = 1, 2, ..., J$, which provides that the employed nonlinear function becomes linear at ℓ_j^* (consecutive) "clock instances";

2. associate to each $\mathbf{u}_j^{(\ell_j)}$ a system of linear equations $\mathcal{L}^{(j)}$ such that given ℓ_j^* stream cipher output bits $\mathbf{z}^{(\ell_j^*)}$, certain ℓ_j^* internal state bits can be recovered solving certain systems of linear equations $\mathcal{L}^{(j)}$, employing a guess&determine approach, $j = 1, 2, ..., L$;

3. construct a number of tables which provide recovering of certain $(L-\ell_j-\ell_j^*)$-dimensional patterns of the guessed bits guessed in the previous steps by inversion of the given output segment into the corresponding internal state employing a dedicated TM-TO paradigm, $j = 1, 2, ..., J$;

Framework for Advanced Cryptanalysis (2)

- Processing which provides recovering of an internal state for given n-bit stream cipher output employing a dedicated TM-TO based hypothesis testing based on all available L-bit keystream segments for each of J linearization approaches specified in the above Pre-Processing.

Advanced Algorithm for Cryptanalysis (1)

The advanced algorithm for cryptanalysis (which is too technical to be discussed here) is built over the following main underlying ideas:

- selection of an appropriate approach for linearization of the nonlinear function taking into account only certain internal states;
- for a given output segment, performing guess and determine paradigm for selection a subset of the candidates for the considered internal state;
- employment of a dedicated Time-Memory-Data Trade-Off based technique for recovering the internal state.

Advanced Algorithm for Cryptanalysis (2)

(a) The cryptanalysis is based on selection of a subset of internal states which imply that the nonlinear function "degrade" to a linear one (or a constant).

(b) The selected subset of states could be considered as obtained by decimation of the sequence of all consecutive internal states.

(c) The cryptanalytic paradigm contains of the following: At each sample point check hypothesis that it corresponds to a state from the selected subset of states employing a dedicated TM-TO based inversion of the stream cipher output to its internal state.

Part III

Novelties in the Advanced Approach

Novelties in the Advanced Approach

- Construction and employment of a **novel dedicated system of equations** for the guess & determine approach
- Development of a dedicated BSW sampling **TMD-TO based on multiple prefix patterns**

Construction and employment of a novel dedicated system of equations for the guess & determine approach

31 internal state bits from 31 consecutive bits of the keystream, after fixing 32 internal state bits and guessing the remaining 97 bits.

Let the vectors $\mathbf{b}^{(i)}$ and $\mathbf{s}^{(i)}$ be the states of the NFSR and LFSR, respectively, at the instant i,

$$\mathbf{s}^{(i)} = [s_i, s_{i+1}, \ldots, s_{i+79}],$$

$$\mathbf{b}^{(i)} = [b_i, b_{i+1}, \ldots, b_{i+79}].$$

Let $\mathbf{u}^{(i)}$ be the internal state of Grain-v1, and accordingly,

$$\begin{aligned}\mathbf{u}^{(i)} &= [\mathbf{s}^{(i)} || \mathbf{b}^{(i)}] \\ &= [s_i, s_{i+1}, \ldots, s_{i+79}, b_i, b_{i+1}, \ldots, b_{i+79}].\end{aligned}$$

The feedback function of LFSR

$$s_{i+80} = s_{i+62} \oplus s_{i+51} \oplus s_{i+38} \oplus s_{i+23} \oplus s_{i+13} \oplus s_i. \quad (20)$$

The feedback function of NFSR

$$\begin{aligned} b_{i+80} = {} & s_i \oplus b_{i+62} \oplus b_{i+60} \oplus b_{i+52} \oplus b_{i+45} \\ & \oplus b_{i+37} \oplus b_{i+33} \oplus b_{i+28} \oplus b_{i+21} \oplus b_{i+14} \\ & \oplus b_{i+9} \oplus b_i \oplus b_{i+63} b_{i+60} \oplus b_{i+37} b_{i+33} \\ & \oplus b_{i+15} b_{i+9} \oplus b_{i+60} b_{i+52} b_{i+45} \\ & \oplus b_{i+33} b_{i+28} b_{i+21} \oplus b_{i+63} b_{i+45} b_{i+28} b_{i+9} \\ & \oplus b_{i+60} b_{i+52} b_{i+37} b_{i+33} \\ & \oplus b_{i+63} b_{i+60} b_{i+21} b_{i+15} \\ & \oplus b_{i+63} b_{i+60} b_{i+52} b_{i+45} b_{i+37} \\ & \oplus b_{i+33} b_{i+28} b_{i+21} b_{i+15} b_{i+9} \\ & \oplus b_{i+52} b_{i+45} b_{i+37} b_{i+33} b_{i+28} b_{i+21}. \end{aligned} \quad (21)$$

4.1 Recovery of 24 bits of the internal state after fixing 6 bits

Now we explain the first case in which the linear relationships involving several internal state bits and keystream bits are used and able to recover 24 bits of internal state. In Table 3, we list a linear restriction of the filter function h which is employed to recover the internal state bits.

Table 3 Restrictions of the filter function h.

Row no.	Constraints	Linear function
1.	$x_2 = 0, x_3 = 1$	$h(x_0, x_1, x_2 = 0, x_3 = 1, x_4) = x_0 + x_1$

As we know that the keystream is computed as follows

$$z_j = \left(\bigoplus_{k \in \mathscr{A}} b_{j+k} \right) \oplus h(s_{j+3}, s_{j+25}, s_{j+46}, s_{j+64}, b_{j+63}) \quad (22)$$

where $\mathscr{A} = \{1, 2, 4, 10, 31, 43, 56\}$.

If $s_{j+46} = 0$, $s_{j+64} = 1$, then by row 1 of Table 3

$$\begin{aligned} z_j = {} & b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \oplus b_{j+10} \oplus b_{j+31} \\ & \oplus b_{j+43} \oplus b_{j+56} \oplus s_{j+3} \oplus s_{j+25}. \end{aligned} \quad (23)$$

Now we prove that considering the equations (22) and (23), it is possible to recover 24 internal state bits from 24 consecutive bits of the keystream, after fixing 6 internal state bits and guessing the remaining 130 bits. This leads reduction of the attack complexities on Grain-v1 which is discussed in the subsequent sections.

In case the 24 consecutive keystream bits (z_0 to z_{23}) are known, we describe stepwise the recovery of 24 internal bits as mentioned above. This stepwise recovery of the state bits are listed in Table 14 which also shows the direction of guessing the bits as well as required equations to recover bits.

From Step 0 to 13, fourteen consecutive bits from b_{10} to b_{23} are recovered by using (22). We derive

$$\begin{aligned} b_{j+10} = {} & z_j \oplus b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \\ & \oplus b_{j+31} \oplus b_{j+43} \oplus b_{j+56} \\ & \oplus h(s_{j+3}, s_{j+25}, s_{j+46}, s_{j+64}, b_{j+63}) \end{aligned} \quad (24)$$

for $j = 0, \ldots, 13$ and by guessing some more internal state bits which are listed in rows 0 to 13 of Table 14.

In Step 14 and 15, two consecutive bits b_{24} and b_{25} are recovered by substituting $j = 14$ and $j = 15$, respectively, in the equation (23). We derive

$$\begin{aligned} b_{j+10} = {} & z_j \oplus b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \\ & \oplus b_{j+31} \oplus b_{j+43} \oplus b_{j+56} \oplus s_{j+3} \oplus s_{j+25}. \end{aligned} \quad (25)$$

under the constraints $s_{j+46} = 0$, $s_{j+64} = 1$, for $j = 14$ and $j = 15$, and by guessing some more internal state bits which are listed in rows 14 and 15 of Table 14 .

In Step 16, b_{26} is recovered by substituting $j = 16$ in the equation (23). and we get

$$\begin{aligned} b_{26} = {} & z_{16} \oplus b_{17} \oplus b_{18} \oplus b_{20} \\ & \oplus b_{47} \oplus b_{69} \oplus b_{72} \oplus s_{19} \oplus s_{41}. \end{aligned} \quad (26)$$

under the constraints $s_{82} = 0$, $s_{80} = 1$ and by guessing some more internal state bits which are listed in row 16 of Table 14. But s_{80} cannot be directly fixed as 1. All bits required to determine s_{80} are already fixed or guessed except s_0 and s_{23}. Thus, we guess s_{23} and s_0 is fixed to $1 + s_{13} + s_{23} + s_{38} + s_{51}$, then s_{80} will be 1 by employing (20) with $i = 0$. The internal state bit b_{26} can be recovered consequently. So far, we have recovered 17 internal state bits.

Table 4 illustrates state bits required to calculate feedback functions of LFSR employing (20) for $i = 0, \ldots, 7$.

Table 4 State bits required to calculate feedback of the LFSR.

i	Obtained feedback bits	Required State bits
0	s_{80}	$s_0, s_{13}, s_{23}, s_{38}, s_{51}, s_{62}$
1	s_{81}	$s_1, s_{14}, s_{24}, s_{39}, s_{52}, s_{63}$
2	s_{82}	$s_2, s_{15}, s_{25}, s_{40}, s_{53}, s_{64}$
3	s_{83}	$s_3, s_{16}, s_{26}, s_{41}, s_{54}, s_{65}$
4	s_{84}	$s_4, s_{17}, s_{27}, s_{42}, s_{55}, s_{66}$
5	s_{85}	$s_5, s_{18}, s_{28}, s_{43}, s_{56}, s_{67}$
6	s_{86}	$s_6, s_{19}, s_{29}, s_{44}, s_{57}, s_{68}$
7	s_{87}	$s_7, s_{20}, s_{30}, s_{45}, s_{58}, s_{69}$

Table 5 illustrates state bits required to calculate feedback functions of NFSR employing (21) for $i = 0, \ldots, 7$. In second column of Table 5 the over-lined state bits are those which are to be recovered before the corresponding feedback bit in the first column is calculated. Therefore the bits have to be recovered in a certain sequence one of which is presented in Table 14.

Table 8 State bits required to calculate feedback of the NFSR.

i	Obtained feedback bits	Required state bits
0	b_{80}	$b_0, b_9, b_{14}, b_{15}, b_{21}, b_{28}, b_{33}, b_{37}, b_{45}, b_{52}, b_{60}, b_{62}, b_{63}, s_0$
1	b_{81}	$b_1, b_{10}, b_{15}, b_{16}, b_{22}, b_{29}, b_{34}, b_{38}, b_{46}, b_{53}, b_{61}, b_{62}, b_{64}, s_1$
2	b_{82}	$b_2, b_{11}, b_{16}, b_{17}, b_{23}, b_{30}, b_{35}, b_{39}, b_{47}, b_{54}, b_{62}, b_{63}, b_{65}, s_2$
3	b_{83}	$b_3, b_{12}, b_{17}, b_{18}, b_{24}, b_{31}, b_{36}, b_{40}, b_{48}, b_{55}, b_{63}, b_{64}, b_{66}, s_3$
4	b_{84}	$b_4, b_{13}, b_{18}, b_{19}, b_{25}, b_{32}, b_{37}, b_{41}, b_{49}, b_{56}, b_{64}, b_{65}, b_{67}, s_4$
5	b_{85}	$b_5, b_{14}, b_{19}, b_{20}, b_{26}, b_{33}, b_{38}, b_{42}, b_{50}, b_{57}, b_{65}, b_{66}, b_{68}, s_5$
6	b_{86}	$b_6, b_{15}, b_{20}, b_{21}, b_{27}, b_{34}, b_{39}, b_{43}, b_{51}, b_{58}, b_{66}, b_{67}, b_{69}, s_6$
7	b_{87}	$b_7, b_{16}, b_{21}, b_{22}, b_{28}, b_{35}, b_{40}, b_{44}, b_{52}, b_{59}, b_{67}, b_{68}, b_{70}, s_7$
8	b_{88}	$b_8, b_{17}, b_{22}, b_{23}, b_{29}, b_{36}, b_{41}, b_{45}, b_{53}, b_{60}, b_{68}, b_{69}, b_{71}, s_8$
9	b_{89}	$b_9, b_{18}, b_{23}, b_{24}, b_{30}, b_{37}, b_{42}, b_{46}, b_{54}, b_{61}, b_{69}, b_{70}, b_{72}, s_9$
10	b_{90}	$b_{10}, b_{19}, b_{24}, b_{25}, b_{31}, b_{38}, b_{43}, b_{47}, b_{55}, b_{62}, b_{70}, b_{71}, b_{73}, s_{10}$
11	b_{91}	$b_{11}, b_{20}, b_{25}, b_{26}, b_{32}, b_{39}, b_{44}, b_{48}, b_{56}, b_{63}, b_{71}, b_{72}, b_{74}, s_{11}$
12	b_{92}	$b_{12}, b_{21}, b_{26}, \overline{b_{27}}, b_{33}, b_{40}, b_{45}, b_{49}, b_{57}, b_{64}, b_{72}, b_{73}, b_{75}, s_{12}$
13	b_{93}	$b_{13}, b_{22}, \overline{b_{27}}, b_{28}, b_{34}, b_{41}, b_{46}, b_{50}, b_{58}, b_{65}, b_{73}, b_{74}, b_{76}, s_{13}$

From Step 21 to 23, the internal state bits s_{53}, s_{54} and s_{55} can be recovered by employing (32). We derive

$$s_{j+25} = z_j \oplus b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \oplus b_{j+10} \oplus b_{j+31} \oplus b_{j+43} \oplus b_{j+56} \oplus b_{j+63} + s_{j+3}s_{j+64} + s_{j+64}b_{j+63}. \quad (44)$$

under the constraints $s_{j+46} = 0$, for $j = 28$, 29 and 30. If we put $j = 28$, 29 and 30 in (44), the values of state bits appear on the right side of equation are already available except b_{j+56}, b_{j+63}, s_{j+64} at $j = 28$, 29 and 30. However, all the bits required to determine them are listed in Table 7 and 8 and known beforehand except s_6, s_{12} and s_{63}. We fix $s_{63} = 0$ and guess rest of them. Thus, the internal state bits s_{53}, s_{54} and s_{55} can be recovered at Step 21, 22, and 23, respectively.

From Step 24 to 27, the internal state bits s_{42}, s_{43}, s_{44} and s_{45} can be recovered by employing (32). We derive

$$s_{j+25} = z_j \oplus b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \oplus b_{j+10} \oplus b_{j+31} \oplus b_{j+43} \oplus b_{j+56} \oplus b_{j+63} + s_{j+3}s_{j+64} + s_{j+64}b_{j+63}. \quad (45)$$

under the constraints $s_{j+46} = 0$, for $j = 17$, 18, 19 and 20, respectively. If we put $j = 17$, 18, 19 and 20 in (45), the values of state bits appear on the right side of equation are already available except s_{20}, b_{j+63} and s_{j+64} at $j = 17$, 18, 19 and 20. We guess s_{20} and all the bits required to determine remaining unknown bits are listed in Table 7 and 8 and known beforehand. Thus, the internal state bits s_{42}, s_{43}, s_{44} and s_{45} can be recovered at Step 24, 25, 26 and 27, respectively.

From Step 28 to 30, three more internal state bits b_{77}, b_{78} and b_{79} can be recovered by employing (32). We derive

$$b_{j+56} = z_j \oplus b_{j+1} \oplus b_{j+2} \oplus b_{j+4} \oplus b_{j+10} \oplus b_{j+31} \oplus b_{j+43} \oplus s_{j+25} \oplus b_{j+63} + s_{j+3}s_{j+64} + s_{j+64}b_{j+63}. \quad (46)$$

under the constraints $s_{j+46} = 0$, for $j = 21$, 22 and 23, respectively. If we put $j = 21$, 22 and 23 in (46), the values of state bits appear on the right side of equation are already available except b_{j+63} and s_{j+64} at $j = 21$, 22 and 23. However, all the bits required to determine them are listed in Table 7 and 8 and known beforehand. Thus, the internal state bits b_{77}, b_{78} and b_{79} can be recovered at Step 28, 29 and 30, respectively. In this way, we have recovered 31 internal state bits of Grain-v1.

Table 9 List of the number of bits fixed, guessed, used and the number of bits recovered thereby in the previous papers and this paper.

References	Bits fixed (ℓ)	Keystream bits used = Bits recovered (ℓ^*)	Bits Guessed ($160 - \ell - \ell^*$)
Bjørstad [4]	0	21	139
Mihaljević et al. [29]	54	18	88
Jiao et al. [20]	51	28	81
First instance	6	24	130
Second instance	32	31	97

For comparison of our results with the existing ones in this direction, we refer to Table 9. In subsequent sections, we discuss a generic framework for cryptanalysis using the results of the current section and then employ TMD-TO technique for recovery of internal state bits of Grain-v1 based on the approaches reported in [3] and [24]. During this explanation, the P, T, M and D are the parameters of TMD-TO and denote required preprocessing time, attack time, disk space, and available data, respectively, for a cryptanalytic attack.

5 Internal State Recovery using a Dedicated Time-Memory-Data Trade-Off Based on Single Prefix Sampling

From the above discussion we note that in case of Grain-v1 cipher, where $l = 160$, it is possible to fix ℓ internal state bits in such a way that given any ℓ^* consecutive bits, denoted by a vector $\mathbf{x}$ say, it is possible to determine ℓ^* more internal state bits so that for each choice of the remaining $(160 - \ell - \ell^*)$ internal state bits, we obtain an internal state of the cipher which produces $\mathbf{x}$ as the first ℓ^* output bits. The success of the conditional TMD-TO used in this section is due to this fundamental observation.

The Time-Memory Trade-Off (TM-TO) approach proposed by Hellman in [19] and TMD-TO due to Biryukov and Shamir [3] are based on the following recursive evaluation of m chains each consisting of I iterations.

$$\begin{aligned} \mathbf{sp}_i = \mathbf{p}_{i,1} &\rightarrow \mathbf{p}_{i,2} = \phi(\mathbf{p}_{i,1}) \\ &\rightarrow \quad \cdots \\ &\rightarrow \mathbf{p}_{i,I+1} = \phi(\mathbf{p}_{i,I}) = \mathbf{ep}_i \end{aligned}$$

Appendix

Table 14 Recovery of 24 bits of the internal state after fixing 6 bits

Step	Corresponding constraints	Key bits	Equations used for recovery	Guessed bits	Feedback bits calculated	Recovered bits
0	-	z_0	$b_{10} = z_0 + b_1 + b_2 + b_4 + b_{31} + b_{43} + b_{56} + h(s_3, s_{25}, s_{46}, s_{64}, b_{83})$	$b_1, b_2, b_4, b_{31}, b_{43}, b_{56}, s_3, s_{25}, s_{46}, s_{64}, b_{83}$	-	b_{10}
1	-	z_1	$b_{11} = z_1 + b_2 + b_3 + b_5 + b_{32} + b_{44} + b_{57} + h(s_4, s_{26}, s_{47}, s_{65}, b_{84})$	$b_2, b_5, b_{32}, b_{44}, b_{57}, s_4, s_{26}, s_{47}, s_{65}, b_{84}$	-	b_{11}
2	-	z_2	$b_{12} = z_2 + b_3 + b_4 + b_6 + b_{33} + b_{45} + b_{58} + h(s_5, s_{27}, s_{48}, s_{66}, b_{85})$	$b_6, b_{33}, b_{45}, b_{58}, s_5, s_{27}, s_{48}, s_{66}, b_{85}$	-	b_{12}
3	-	z_3	$b_{13} = z_3 + b_4 + b_5 + b_7 + b_{34} + b_{46} + b_{59} + h(s_6, s_{28}, s_{49}, s_{67}, b_{86})$	$b_7, b_{34}, b_{46}, b_{59}, s_6, s_{28}, s_{49}, s_{67}, b_{86}$	-	b_{13}
4	-	z_4	$b_{14} = z_4 + b_5 + b_6 + b_8 + b_{35} + b_{47} + b_{60} + h(s_7, s_{29}, s_{50}, s_{68}, b_{87})$	$b_8, b_{35}, b_{47}, b_{60}, s_7, s_{29}, s_{50}, s_{68}, b_{87}$	-	b_{14}
5	-	z_5	$b_{15} = z_5 + b_6 + b_7 + b_9 + b_{36} + b_{48} + b_{61} + h(s_8, s_{30}, s_{51}, s_{69}, b_{88})$	$b_9, b_{36}, b_{48}, b_{61}, s_8, s_{30}, s_{51}, s_{69}, b_{88}$	-	b_{15}
6	-	z_6	$b_{16} = z_6 + b_7 + b_8 + b_{10} + b_{37} + b_{49} + b_{62} + h(s_9, s_{31}, s_{52}, s_{70}, b_{89})$	$b_{37}, b_{49}, b_{62}, s_9, s_{31}, s_{52}, s_{70}, b_{89}$	-	b_{16}
7	-	z_7	$b_{17} = z_7 + b_8 + b_9 + b_{11} + b_{38} + b_{50} + b_{63} + h(s_{10}, s_{32}, s_{53}, s_{71}, b_{70})$	$b_{38}, b_{50}, s_{10}, s_{32}, s_{53}, s_{71}, b_{70}$	-	b_{17}
8	-	z_8	$b_{18} = z_8 + b_9 + b_{10} + b_{12} + b_{39} + b_{51} + b_{64} + h(s_{11}, s_{33}, s_{54}, s_{72}, b_{71})$	$b_{39}, b_{51}, s_{11}, s_{33}, s_{54}, s_{72}, b_{71}$	-	b_{18}
9	-	z_9	$b_{19} = z_9 + b_{10} + b_{11} + b_{13} + b_{40} + b_{52} + b_{65} + h(s_{12}, s_{34}, s_{55}, s_{73}, b_{72})$	$b_{40}, b_{52}, s_{12}, s_{34}, s_{55}, s_{73}, b_{72}$	-	b_{19}
10	-	z_{10}	$b_{20} = z_{10} + b_{11} + b_{12} + b_{14} + b_{41} + b_{53} + b_{66} + h(s_{13}, s_{35}, s_{56}, s_{74}, b_{73})$	$b_{41}, b_{53}, s_{13}, s_{35}, s_{56}, s_{74}, b_{73}$	-	b_{20}
11	-	z_{11}	$b_{21} = z_{11} + b_{12} + b_{13} + b_{15} + b_{42} + b_{54} + b_{67} + h(s_{14}, s_{36}, s_{57}, s_{75}, b_{74})$	$b_{42}, b_{54}, s_{14}, s_{36}, s_{57}, s_{75}, b_{74}$	-	b_{21}
12	-	z_{12}	$b_{22} = z_{12} + b_{13} + b_{14} + b_{16} + b_{43} + b_{55} + b_{68} + h(s_{15}, s_{37}, s_{58}, s_{76}, b_{75})$	$b_{55}, s_{15}, s_{37}, s_{58}, s_{76}, b_{75}$	-	b_{22}
13	-	z_{13}	$b_{23} = z_{13} + b_{14} + b_{15} + b_{17} + b_{44} + b_{56} + b_{69} + h(s_{16}, s_{38}, s_{59}, s_{77}, b_{76})$	$s_{16}, s_{38}, s_{59}, s_{77}, b_{76}$	-	b_{23}
14	$s_{60} = 0, s_{78} = 1$	z_{14}	$b_{24} = z_{14} + b_{15} + b_{16} + b_{18} + b_{45} + b_{57} + b_{70} + s_{17} + s_{39}$	s_{17}, s_{39}	-	b_{24}
15	$s_{61} = 0, s_{79} = 1$	z_{15}	$b_{25} = z_{15} + b_{16} + b_{17} + b_{19} + b_{46} + b_{58} + b_{71} + s_{18} + s_{40}$	s_{18}, s_{40}	-	b_{25}
16	$s_{62} = 0,$ $s_0 = 1 + s_{13} + s_{38} + s_{23} + s_{51}$	z_{16}	$b_{26} = z_{16} + b_{17} + b_{18} + b_{20} + b_{47} + b_{59} + b_{72} + s_{19} + s_{41}$	s_{19}, s_{23}, s_{41}	-	b_{26}
17	-	z_{20}	$b_{30} = z_{20} + b_{21} + b_{22} + b_{24} + b_{51} + b_{63} + b_{76} + h(s_{23}, s_{45}, s_{66}, s_{84}, b_{83})$	s_{42}, s_{45}	s_{84}, b_{83}	b_{30}
18	-	z_{19}	$b_{29} = z_{19} + b_{20} + b_{21} + b_{23} + b_{50} + b_{62} + b_{75} + h(s_{22}, s_{44}, s_{65}, s_{83}, b_{82})$	s_2, s_{22}, s_{44}	s_{83}, b_{82}	b_{29}
19	-	z_{18}	$b_{28} = z_{18} + b_{19} + b_{20} + b_{22} + b_{49} + b_{61} + b_{74} + h(s_{21}, s_{43}, s_{64}, s_{82}, b_{81})$	s_1, s_{21}, s_{43}	s_{82}, b_{81}	b_{28}
20	-	z_{17}	$b_{27} = z_{17} + b_{18} + b_{19} + b_{21} + b_{48} + b_{60} + b_{73} + h(s_{20}, s_{42}, s_{63}, s_{81}, b_{80})$	$s_{20}, s_{24}, s_{63}, b_0$	s_{81}, b_{80}	b_{27}
21	-	z_{21}	$b_{77} = z_{21} + b_{22} + b_{23} + b_{25} + b_{31} + b_{52} + b_{64} + h(s_{24}, s_{46}, s_{67}, s_{85}, b_{84})$	—	s_{85}, b_{84}	b_{77}
22	-	z_{22}	$b_{78} = z_{22} + b_{23} + b_{24} + b_{26} + b_{32} + b_{53} + b_{65} + h(s_{25}, s_{47}, s_{68}, s_{86}, b_{85})$	—	s_{86}, b_{85}	b_{78}
23	-	z_{23}	$b_{79} = z_{23} + b_{24} + b_{25} + b_{27} + b_{33} + b_{54} + b_{66} + h(s_{26}, s_{48}, s_{69}, s_{87}, b_{86})$	—	s_{87}, b_{86}	b_{79}

IET Research Journals, pp. 1–15

Table 15 Recovery of 31 bits of the internal state after fixing 32 bits using linear and quadratic restrictions

Step	Corresponding constraints	Key bits	Equations used for recovery	Guessed bits	Feedback bits calculated	Recovered bits
0	$a_{54} = 0$	z_0	$b_{10} = z_0 + b_1 + b_2 + b_4 + b_{31} + b_{43} + b_{56} + h(a_3, a_{25}, a_{46}, a_{54}, b_{23})$	$b_1, b_2, b_4, b_{31}, b_{43}, b_{56}, a_3, a_{25}, a_{46}, b_{23}$	–	b_{10}
1	$a_{55} = 0$	z_1	$b_{11} = z_1 + b_2 + b_3 + b_5 + b_{32} + b_{44} + b_{57} + h(a_4, a_{26}, a_{47}, a_{55}, b_{24})$	$b_3, b_5, b_{32}, b_{44}, b_{57}, a_4, a_{26}, a_{47}, b_{24}$	–	b_{11}
2	$a_{56} = 0$	z_2	$b_{12} = z_2 + b_3 + b_4 + b_6 + b_{33} + b_{45} + b_{58} + h(a_5, a_{27}, a_{48}, a_{56}, b_{25})$	$b_6, b_{33}, b_{45}, b_{58}, a_5, a_{27}, a_{48}, b_{25}$	–	b_{12}
3	$a_{26} = 0, a_{57} = 0, b_{55} = 0$	z_3	$b_{13} = z_3 + b_4 + b_5 + b_7 + b_{34} + b_{46} + b_{59}$	$b_7, b_{34}, b_{46}, b_{59}$	–	b_{13}
4	$a_{27} = 0, a_{58} = 0, b_{57} = 0$	z_4	$b_{14} = z_4 + b_5 + b_6 + b_8 + b_{35} + b_{47} + b_{60}$	$b_8, b_{35}, b_{47}, b_{60}$	–	b_{14}
5	$a_{59} = 0$	z_5	$b_{15} = z_5 + b_6 + b_7 + b_9 + b_{36} + b_{48} + b_{61} + h(a_8, a_{30}, a_{51}, a_{59}, b_{28})$	$b_9, b_{36}, b_{48}, b_{61}, a_8, a_{30}, a_{51}, b_{28}$	–	b_{15}
6	$a_{31} = 0, a_{70} = 0, b_{69} = 0$	z_6	$b_{16} = z_6 + b_7 + b_8 + b_{10} + b_{37} + b_{49} + b_{62}$	b_{37}, b_{49}, b_{62}	–	b_{16}
7	$a_{32} = 0, a_{71} = 0, b_{70} = 0$	z_7	$b_{17} = z_7 + b_8 + b_9 + b_{11} + b_{38} + b_{50} + b_{63}$	b_{38}, b_{50}	–	b_{17}
8	$a_{33} = 0, a_{72} = 0, b_{71} = 0$	z_8	$b_{18} = z_8 + b_9 + b_{10} + b_{12} + b_{39} + b_{51} + b_{64}$	b_{39}, b_{51}	–	b_{18}
9	$a_{34} = 0, a_{73} = 0, b_{72} = 0$	z_9	$b_{19} = z_9 + b_{10} + b_{11} + b_{13} + b_{40} + b_{52} + b_{65}$	b_{40}, b_{52}	–	b_{19}
10	$a_{74} = 0$	z_{10}	$b_{20} = z_{10} + b_{11} + b_{12} + b_{14} + b_{41} + b_{53} + b_{66} + h(a_{13}, a_{35}, a_{56}, a_{74}, b_{73})$	$b_{41}, b_{53}, a_{13}, a_{35}, a_{56}, b_{73}$	–	b_{20}
11	$a_{75} = 0$	z_{11}	$b_{21} = z_{11} + b_{12} + b_{13} + b_{15} + b_{42} + b_{54} + b_{67} + h(a_{14}, a_{36}, a_{57}, a_{75}, b_{74})$	$b_{42}, b_{54}, a_{14}, a_{36}, a_{57}, b_{74}$	–	b_{21}
12	$a_{76} = 0$	z_{12}	$b_{22} = z_{12} + b_{13} + b_{14} + b_{16} + b_{43} + b_{55} + b_{68} + h(a_{15}, a_{37}, a_{58}, a_{76}, b_{75})$	$b_{55}, a_{15}, a_{37}, a_{58}, b_{75}$	–	b_{22}
13	–	z_{13}	$b_{23} = z_{13} + b_{14} + b_{15} + b_{17} + b_{44} + b_{56} + b_{69} + h(a_{16}, a_{38}, a_{59}, a_{77}, b_{76})$	$a_{16}, a_{38}, a_{59}, a_{77}, b_{76}$	–	b_{23}
14	$a_{60} = 0, a_{78} = 1$	z_{14}	$b_{24} = z_{14} + b_{15} + b_{16} + b_{18} + b_{45} + b_{57} + b_{70} + a_{17} + a_{39}$	a_{17}, a_{39}	–	b_{24}
15	$a_{61} = 0, a_{79} = 1$	z_{15}	$b_{25} = z_{15} + b_{16} + b_{17} + b_{19} + b_{46} + b_{58} + b_{71} + a_{18} + a_{40}$	a_{18}, a_{40}	–	b_{25}
16	$a_{62} = 0,$ $a_0 = 1 + a_{13} + a_{23} + a_{38} + a_{51}$	z_{16}	$a_{41} = z_{16} + b_{17} + b_{18} + b_{20} + b_{26} + b_{47} + b_{59} + b_{72} + a_{19}$	b_{26}, a_{19}, a_{23}	–	a_{41}
17	$\overline{a_{70}} = 0$	z_{34}	$a_{49} = z_{34} + b_{25} + b_{26} + b_{28} + b_{34} + b_{55} + b_{67} + b_{80} + b_{87} + a_{27}a_{88} + a_{88}b_{87}$	b_0, b_{28}, a_7, a_{21}	b_{80}, b_{87}, a_{88}	a_{49}
18	$\overline{a_{69}} = 0$	z_{35}	$b_{27} = z_{35} + b_{28} + b_{30} + b_{36} + b_{57} + b_{69} + b_{82} + a_{51} + b_{89} + a_{29}a_{90} + a_{90}b_{89}$	b_{30}, a_2, a_9, a_{10}	b_{82}, b_{89}, a_{90}	b_{27}
19	$\overline{a_{67}} = 0$	z_{25}	$a_{50} = z_{25} + b_{26} + b_{27} + b_{29} + b_{35} + b_{56} + b_{68} + b_{81} + b_{88} + a_{28}a_{89} + a_{89}b_{88}$	b_{29}, a_1, a_{22}	b_{81}, b_{88}, a_{89}	a_{50}
20	$\overline{a_{73}} = 0$	z_{27}	$a_{52} = z_{27} + b_{28} + b_{29} + b_{31} + b_{37} + b_{58} + b_{70} + b_{83} + b_{90} + a_{30}a_{91} + a_{91}b_{90}$	a_{11}, a_{24}	b_{83}, b_{90}, a_{91}	a_{52}
21	$\overline{a_{74}} = 0, a_{63} = 0$	z_{28}	$a_{53} = z_{28} + b_{29} + b_{30} + b_{32} + b_{38} + b_{59} + b_{71} + b_{84} + b_{91} + a_{31}a_{92} + a_{92}b_{91}$	a_{12}	b_{84}, b_{91}, a_{92}	a_{53}
22	$\overline{a_{75}} = 0$	z_{29}	$a_{54} = z_{29} + b_{30} + b_{31} + b_{33} + b_{39} + b_{60} + b_{72} + b_{85} + b_{92} + a_{32}a_{93} + a_{93}b_{92}$	–	b_{85}, b_{92}, a_{93}	a_{54}
23	$\overline{a_{76}} = 0$	z_{30}	$a_{55} = z_{30} + b_{31} + b_{32} + b_{34} + b_{40} + b_{61} + b_{73} + b_{86} + b_{93} + a_{33}a_{94} + a_{94}b_{93}$	a_5	b_{86}, b_{93}, a_{94}	a_{55}
24	$\overline{a_{63}} = 0$	z_{17}	$a_{42} = z_{17} + b_{18} + b_{19} + b_{21} + b_{27} + b_{48} + b_{60} + b_{73} + b_{80} + a_{20}a_{81} + a_{81}b_{80}$	a_{20}	b_{80}, a_{81}	a_{42}
25	$\overline{a_{64}} = 0$	z_{18}	$a_{43} = z_{18} + b_{19} + b_{20} + b_{22} + b_{28} + b_{49} + b_{61} + b_{74} + b_{81} + a_{21}a_{82} + a_{82}b_{81}$	–	b_{81}, a_{82}	a_{43}
26	$\overline{a_{65}} = 0$	z_{19}	$a_{44} = z_{19} + b_{20} + b_{21} + b_{23} + b_{29} + b_{50} + b_{62} + b_{75} + b_{82} + a_{22}a_{83} + a_{83}b_{82}$	–	b_{82}, a_{83}	a_{44}
27	$\overline{a_{66}} = 0$	z_{20}	$a_{45} = z_{20} + b_{21} + b_{22} + b_{24} + b_{30} + b_{51} + b_{63} + b_{76} + b_{83} + a_{23}a_{84} + a_{84}b_{83}$	–	b_{83}, a_{84}	a_{45}
28	$\overline{a_{67}} = 0$	z_{21}	$b_{77} = z_{21} + b_{22} + b_{23} + b_{25} + b_{31} + b_{52} + b_{64} + a_{46} + b_{84} + a_{24}a_{85} + a_{85}b_{84}$	–	b_{84}, a_{85}	b_{77}
29	$\overline{a_{68}} = 0$	z_{22}	$b_{78} = z_{22} + b_{23} + b_{24} + b_{26} + b_{32} + b_{53} + b_{65} + a_{47} + b_{85} + a_{25}a_{86} + a_{86}b_{85}$	–	b_{85}, a_{86}	b_{78}
30	$\overline{a_{69}} = 0$	z_{23}	$b_{79} = z_{23} + b_{24} + b_{25} + b_{27} + b_{33} + b_{54} + b_{66} + a_{48} + b_{86} + a_{26}a_{87} + a_{87}b_{86}$	–	b_{86}, a_{87}	b_{79}

Thank You Very Much for the Attention,

and

QUESTIONS Please!