A New Version of Grain-128 with Authentication

Martin Ågren[1]
Martin Hell[1]
Thomas Johansson[1]
Willi Meier[2]

[1] Lund University, Sweden

[2] FHNW, Switzerland

Outline

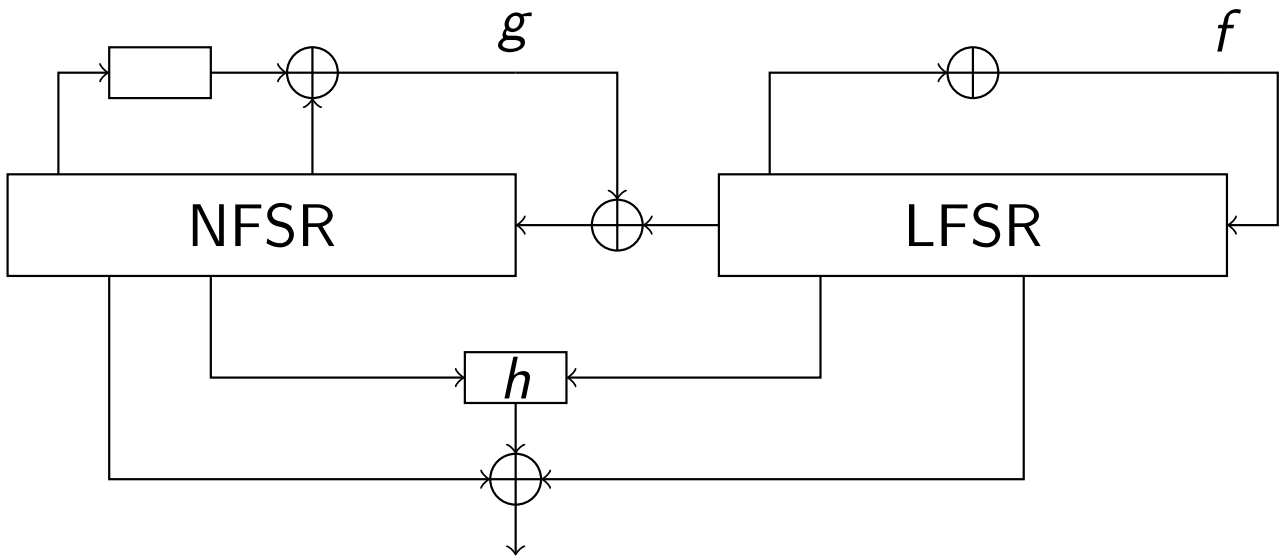
Outline

Motivation and Goals

- Grain-128 is lightweight but some nonlinearities are too lightweight.
- Some applications need built-in authentication
- . . . but leaving it out should be possible.
- Allow for easy updating of existing implementations.
- . . . and trust!

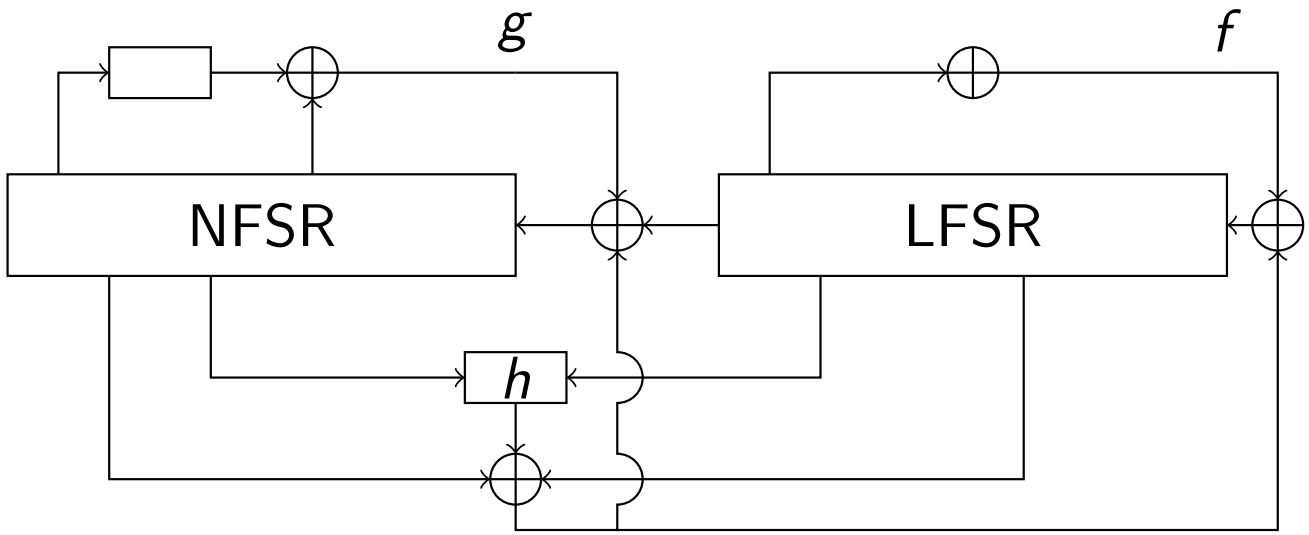
Outline

The Old Grain-128



- 128-bit key, 96-bit IV.
- An LFSR provides a large period.
- An NFSR with degree two updates the state nonlinearly.
- An output function of degree three produces nonlinear output.
- State bits are added linearly to ensure resiliency.
- Initialize in 256 rounds: feed output into the registers.
- Make faster by duplicating Boolean functions.

IV Padding Sliding Property



- The 96-bit IV goes into a 128-bit register and is padded with 111...111. With high probability, a shifted key and a shifted IV will produce the exact same keystream, only with a shift. [Küçük06], [DeCaKüPre08]
- Related-key Chosen-IV. [LeeJeongSungHong08]

Too Little Nonlinearity or Initialization

- Cube, 237/256 [AumDinHenMeiSha09]
- Maxterm, 256/256 [Stankovski10]

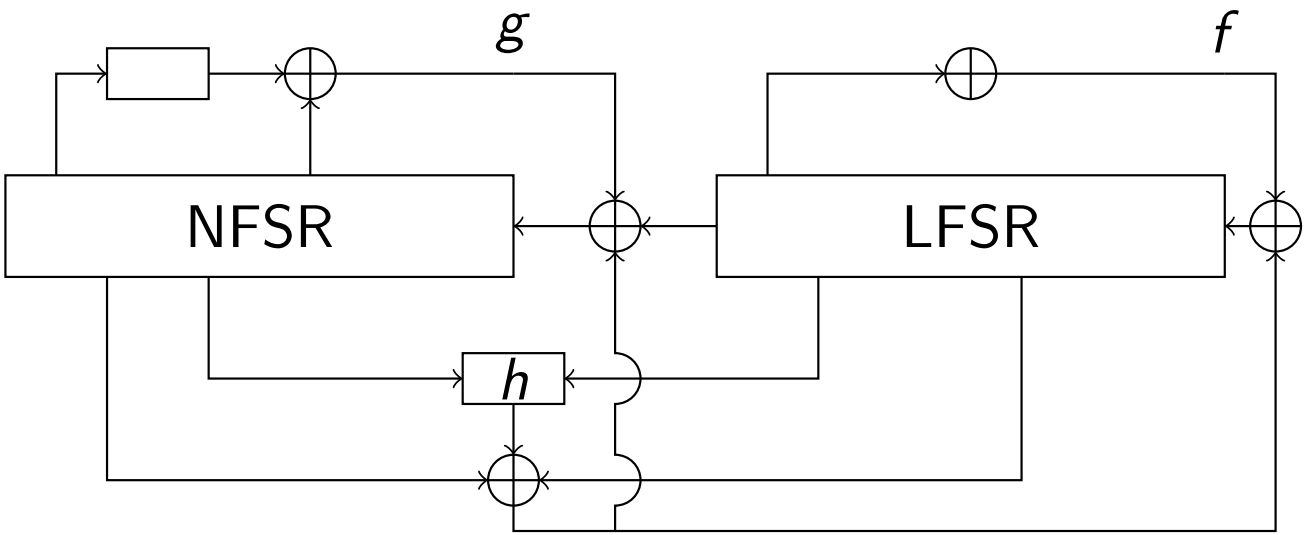
Looking at the first keystream bits, the equations, in unknown key bits, are not complicated enough.

Too Little Nonlinearity and Similar Bits

- Chosen-IV (cube): Assuming ten specific key bits to be zero, the equations simplify "enough". [DinSha11]

Too Little Nonlinearity and Similar Bits

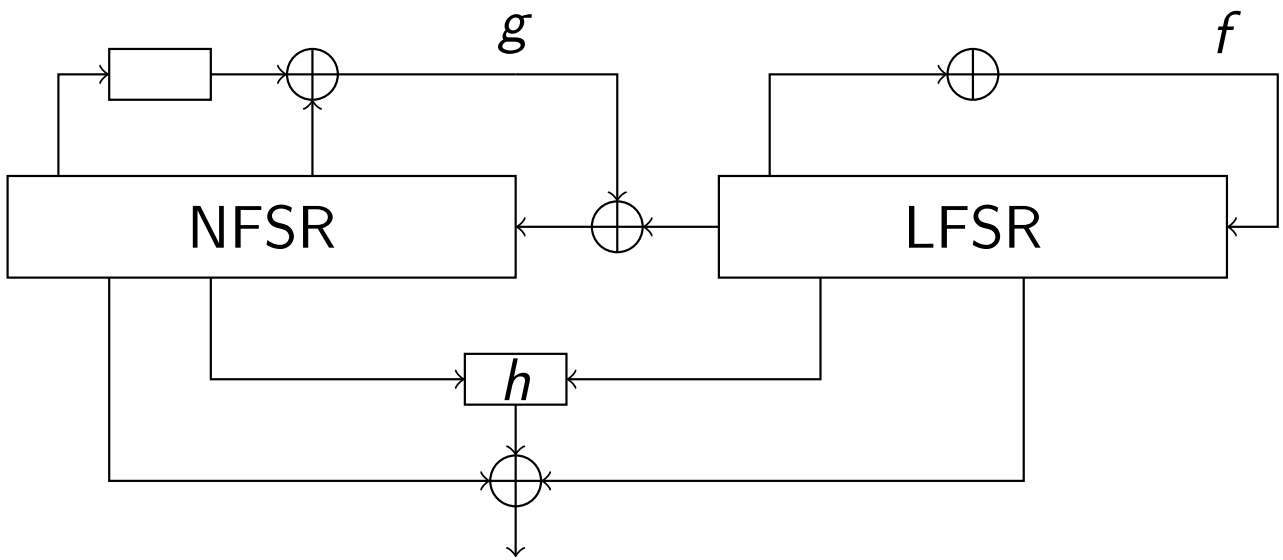
- Chosen-IV (cube): Assuming ten specific key bits to be zero, the equations simplify "enough". [DinSha11]



- Also, b_{i+95} and s_{i+95} are multiplied together. During initialization, they are too similar, meaning the complexity doesn't grow as much as wanted.

Outline

Changes from Grain-128



Grain-128 with changes:

- Pad the IV with 111...11**0**.
- NFSR has nonlinearity **four**.
- Change a tap into the output function: b_{i+95}, $\mathbf{s_{i+94}}$, so that we don't multiply bits that are "similar".

Authentication

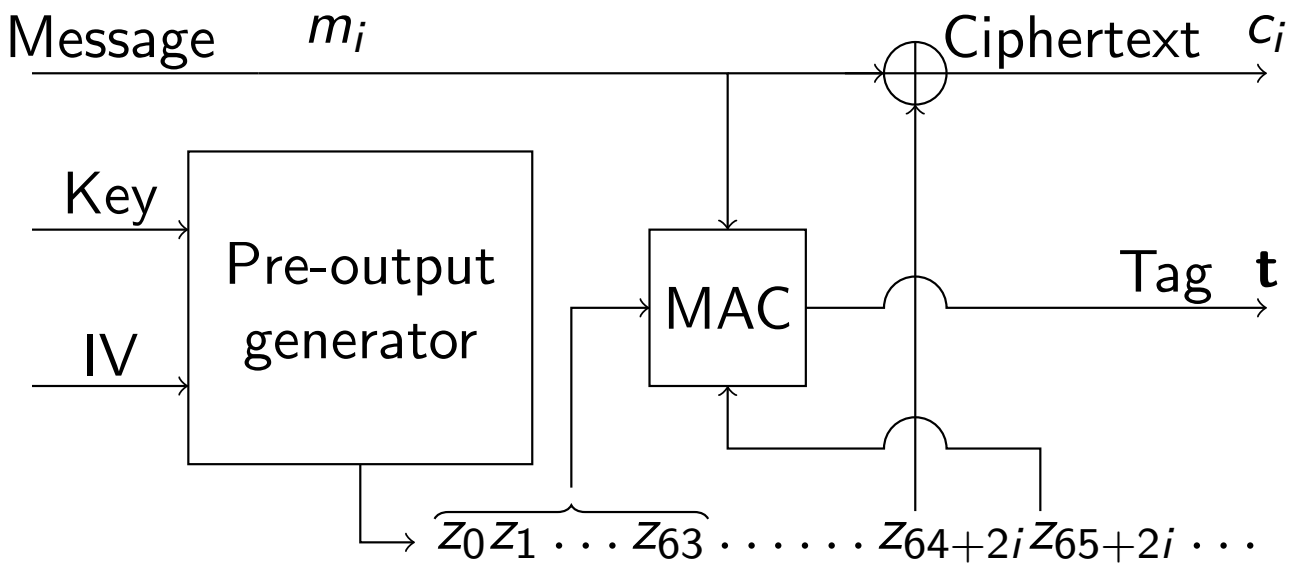
The above algorithm is used to produce *pre-output stream*.
Use different parts of it for different things:

- Encryption
- Authentication

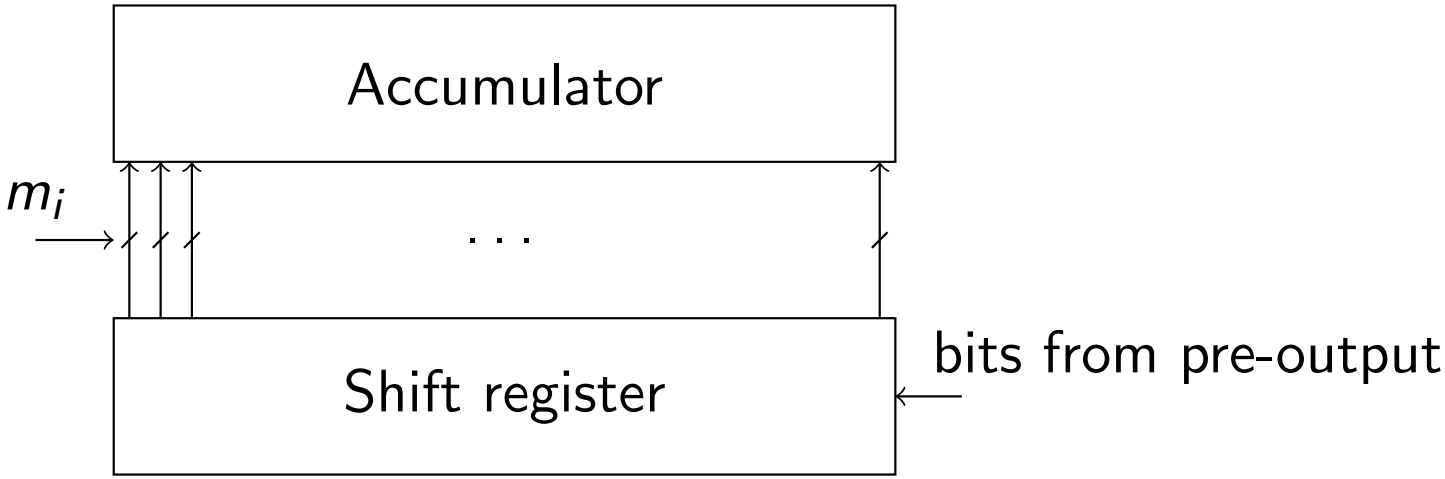
Authentication

The above algorithm is used to produce *pre-output stream*.
Use different parts of it for different things:

- Encryption
- Authentication

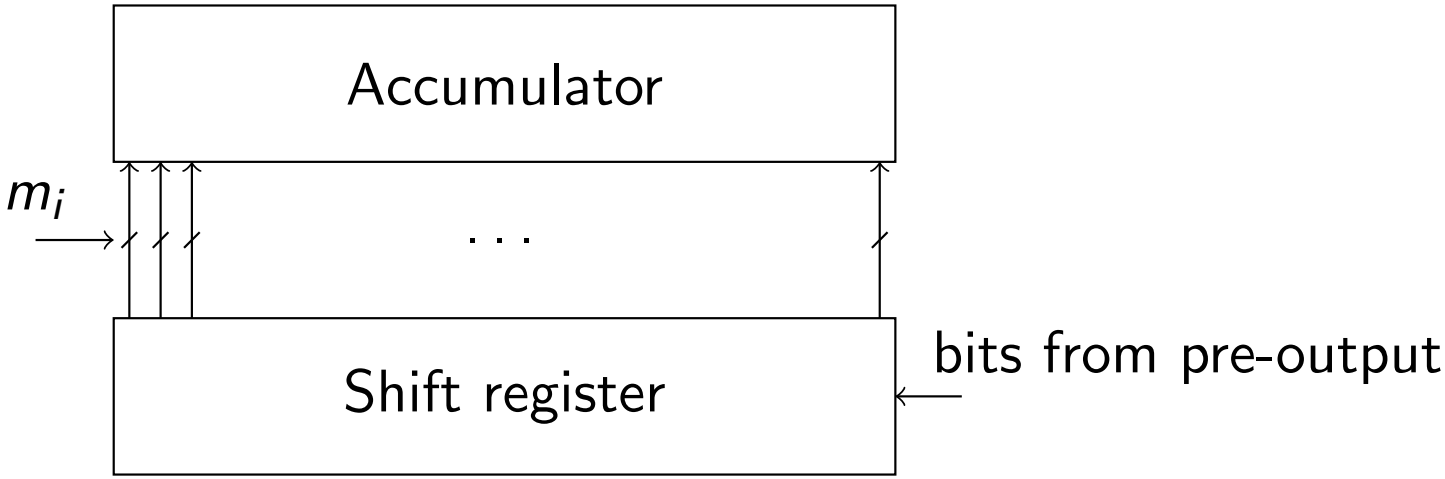


Authentication



- A Wegman-Carter approach.
- Initialize both registers with pre-output bits.
- We multiply the message bit vector by a Toeplitz matrix.

Authentication



- A Wegman-Carter approach.
- Initialize both registers with pre-output bits.
- We multiply the message bit vector by a Toeplitz matrix.
- P_S is the prob. that an attack succeeds.
- With perfectly random input to the shift register, $P_S = 2^{-32}$.
- We have $P_S < 2^{-32} + 2\epsilon$. [Krawczyk95], [ÅHJ11], [Maximov06]

Hardware Characteristics

Several nice aspects:

- We can still increase the speed up to 32x.
- We can leave out the authentication.
- . . . or part of it. w-bit tags for 2^{-w}.

Hardware Characteristics

Several nice aspects:

- We can still increase the speed up to 32x.
- We can leave out the authentication.
- . . . or part of it. w-bit tags for 2^{-w}.

The cheapest one — a version that produces one bit per clock:

- Grain-128: 2133 gate equivalents
- Grain-128a: 2243 gate equivalents; a five per cent increase (as a bonus, we initialize faster.)

Adding authentication, we'd get a total of 2867 gate equivalents.

Outline

Conclusion

Grain-128a

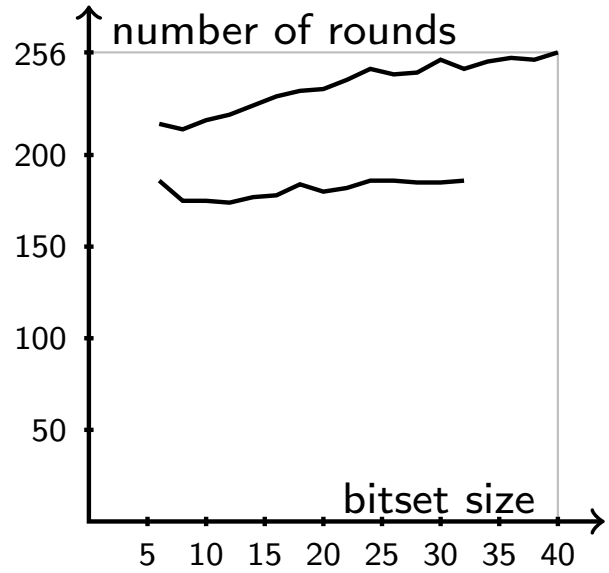
- is at least as secure than Grain-128,
- resists all current cryptanalysis on Grain-128,
- has optional authentication,
- is still hardware-efficient.

Conclusion

Thank you!


SIGIL
AD VTRVMQVE
CAROLINÆ
ORVM

On Cube/Maxterm/AIDA/...



How does a greedy strategy aid in finding good bitsets?
Upper curve: Stankovski's on Grain-128.
Lower curve: Ours on the pre-output of Grain-128a.