Survey on the Effectiveness of DAPA-Related Attacks against Shift Register Based AEAD Schemes

NIST Lightweight Cryptography Workshop 2022

Dr Shivam Bhasin[2], Dr Dirmanto Jap[2], Wei Cheng Ng[1], Dr Siang Meng Sim[1,2]

[1] DSO National Laboratories, Singapore
[2] Nanyang Technological University, Singapore

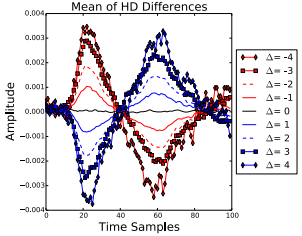
May 11, 2022

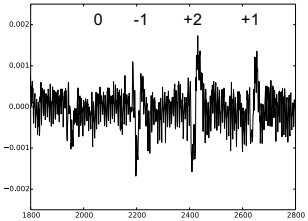
Table of Contents

Introduction

Background

- Side-Channel Analysis (SCA) provides a powerful adversarial model against stream ciphers.
- Leaked side-channel traces can provide the attacker with direct information about the internal shift register states at a given time, which greatly simplifies otherwise high complexity theoretical cryptanalysis.
- Example: Bit value differentials in a register can be simulated with power trace leakages via the Hamming Distance model.





Motivation

- In 2021, Sim et al. [1] combined SCA with differential analysis techniques to formulate a new attack framework known as Differential Analysis-aided Power Attack (DAPA).
- DAPA has thus far been used to achieve key recovery on Trivium and purported SCA-resistant LR-Keymill.
- This work builds upon the results achieved by Sim et al.

Our Contributions

- Amongst the ten NIST LWC finalists, Grain-128AEADv2 and TinyJAMBU use Linear and Non-linear Feedback Shift Registers (LFSRs/NFSRs) as building blocks for their respective primitives.
- We compare the effectiveness of SCA, particularly **DAPA**, against both 1-bit and 32-bit implementations of these shift register based finalists.
- We outline the similarities and differences between the level of DAPA resistance offered by Grain-128AEADv2 and TinyJAMBU, and highlight the features of each cipher that offers (or fails to offer) some degree of inherent SCA resistance.

Our Contributions

Finalist	Key size	Implementation	Bits recovered
Grain-128AEADv2	128	1-bit	128
		32-bit	64
TinyJAMBU	128/192/256	1-bit	128/192/256*
		32-bit	32**

Table: Summary of DAPA on Shift Register based NIST finalists. 128/192/256* refers to recovering the full 128/192/256-bit key with a 6-bit guess. 32** refers to recovering 32 bits of the secret internal register state.

Revisiting DAPA

What is DAPA?

- DAPA is an extension of Differential Power Analysis (DPA) that considers differences in power consumption behaviour beyond the first clock cycle, by tracing the propagation of injected input differences between pairs of cipher instantiations.
- In this work, we recover information about the internal register states by exploiting this difference propagation in both linear and non-linear functions through multiple clocks.

DAPA Methodology

Step 1 (Offline): Determine the differential patterns to track.

Step 2 (Online): Perform the power measurements.

Step 3 (Offline): Recover the internal state by deducing bit relations from the measured power traces.

We inject the differences via the IV to trigger our desired difference pattern, and track these differentials through the ciphers' respective initialization phases.

Differential Patterns and Bit Relations

Case	Possible outcomes	Power diff.
1.1: $[x]y$ vs $[x]\hat{y}$	$x = y$	$+1$
	$x \neq y$	-1
1.2: $[x]y$ vs $[\hat{x}]y$	$x = y$	$+1$
	$x \neq y$	-1
1.3: $[x]y$ vs $[\hat{x}]\hat{y}$	$x = y$	0
	$x \neq y$	

Case	Possible outcomes	Power diff.
2.1: $[z]x \circ y$ vs $[z]x \circ \hat{y}$	$x = 1, x \circ y = z$	$+1$
	$x = 0$	0
	$x = 1, x \circ y \neq z$	-1
2.2: $[x \circ y]z$ vs $[x \circ \hat{y}]z$	$x = 1, x \circ y = z$	$+1$
	$x = 0$	0
	$x = 1, x \circ y \neq z$	-1

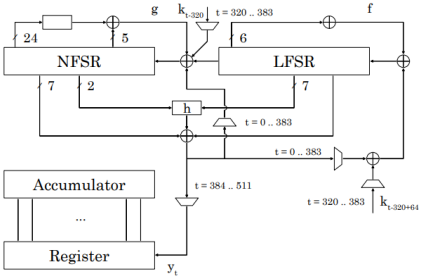
$\circ$ denotes either the AND ($\wedge$) or the NAND ($\overline{\wedge}$) non-linear function.

Observation 1.

In Cases 2.1 and 2.2, there is a change in the parity of the net power trace differences if and only if $x = 1$.

Grain-128 AEAD v2

Specifications

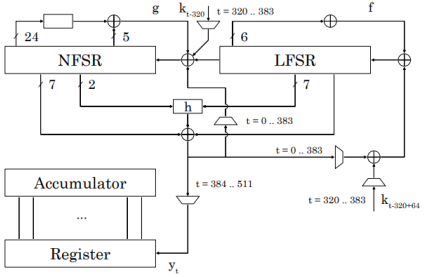


We target the first 320 clock cycles in the **Key and Nonce Initialization phase** of the scheme.

- NFSR ($B_t = [b_0^t, b_1^t, \ldots, b_{127}^t]$) initialized with the 128-bit key.
- LFSR ($S_t = [s_0^t, s_1^t, \ldots, s_{127}^t]$) partially initialized with 96-bit IV, in the following manner: $[IV||0^{31}||1]$.
- First 320 updates:

$$s_{127}^{t+1} = \mathcal{L}(S_t) + y_t,$$
$$b_{127}^{t+1} = s_0^t + \mathcal{F}(B_t) + y_t.$$

Specifications



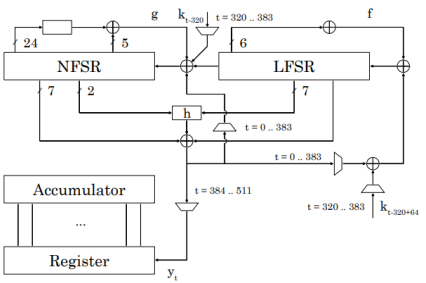
$$
\begin{aligned}
\mathcal{L}(S_t) &= s_0^t + s_7^t + s_{38}^t + s_{70}^t + s_{81}^t + s_{96}^t, \\
\mathcal{F}(B_t) &= b_0^t + b_{26}^t + b_{56}^t + b_{91}^t + b_{96}^t + b_3^t b_{67}^t \\
&\quad + b_{11}^t b_{13}^t + b_{17}^t b_{18}^t + b_{27}^t b_{59}^t + b_{40}^t b_{48}^t \\
&\quad + b_{61}^t b_{65}^t + b_{68}^t b_{84}^t + b_{22}^t b_{24}^t b_{25}^t \\
&\quad + b_{70}^t b_{78}^t b_{82}^t + b_{88}^t b_{92}^t b_{93}^t b_{95}^t, \\
y^t &= h^t + s_{93}^t + b_2^t + b_{15}^t + b_{36}^t \\
&\quad + b_{45}^t + b_{64}^t + b_{73}^t + b_{89}^t, \\
h^t &= b_{12}^t s_8^t + s_{13}^t s_{20}^t + b_{95}^t s_{42}^t \\
&\quad + s_{60}^t s_{79}^t + b_{12}^t b_{95}^t s_{94}^t.
\end{aligned}
$$

Properties

1. Slow diffusion of the non-linear component in the LFSR;
2. Direct application of nonce into the shift register during initialization; and
3. Low algebraic degree of the nonlinear combiner h, and the low weight of variables from the NFSR in h.

$$h^t = \mathbf{b_{12}^t s_8^t} + s_{13}^t s_{20}^t + \mathbf{b_{95}^t s_{42}^t} + s_{60}^t s_{79}^t + \mathbf{b_{12}^t b_{95}^t s_{94}^t}$$

DAPA on 1-bit Implementation of Grain-128AEADv2



- **Phase 1:** Deduce internal states of LFSR registers and y^t values for the first 70 clocks.
- **Phase 2:** Obtain 85 NFSR register values by exploiting quadratic terms in h.
- **Phase 3:** Recover the full internal state and key.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 1: Deduce internal states of LFSR registers

- We recover S_t and y^t for the first 70 clocks inductively using DAPA.
- Base case: We have S_0 from the IV.
- Inductive step:
  - For $0 \leq k < 70$, we consider the pair of ciphers instantiated by the nonce pair p_k, where one cipher instance has a zero IV, while the second cipher instance has an IV that is zero everywhere except at register k.
  - We see that at time $t = k$, this difference induced at register s_k^0 propagates to a difference at register s_0^k.
  - This difference gets propagated via $\mathcal{L}(S_k)$ to register s_{127}^{k+1}.
  - Given that we know $s_{i+1}^k = s_i^{k+1}$ for all $0 \leq i < 127$ by the inductive hypothesis, we can use the differential pattern in Case 1.1 to extract the value of s_{127}^{k+1} and, hence, y^{k+1}, which allows us to uncover the full internal state S_{k+1}.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 2: Exploit quadratic terms in h to recover NFSR register values

Recall:

Observation 1.

In Cases 2.1 and 2.2, there is a change in the parity of the net power trace differences if and only if $x = 1$.

$$h^t = \mathbf{b_{12}^t s_8^t} + s_{13}^t s_{20}^t + \mathbf{b_{95}^t s_{42}^t} + s_{60}^t s_{79}^t + \mathbf{b_{12}^t b_{95}^t s_{94}^t}$$

- If we induce a difference in s_8^t (resp. s_{42}^t) while the remaining terms in h^t remain unchanged, we will be able to derive the values of b_{12}^t (resp. b_{95}^t) via Observation 1, by measuring the parity of the power trace differentials.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 2: Exploit quadratic terms in h to recover NFSR register values

- We are able to perform this technique for $0 \leq k < 34$ to target $b_{12}^k s_8^k$, and $0 \leq k < 51$ to target $b_{95}^k s_{42}^k$, before things begin to get messy.
- This gives us 85 NFSR register values.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 3: Putting it all together

- So far, we have obtained:
  - Internal LFSR register values s_k for $0 \leq k < 198$;
  - Non-linear feedback function values y^k for $0 \leq k < 70$; and
  - Internal NFSR register values b_k for $12 \leq k < 46$ and $95 \leq k < 146$.
- The equations defined by y^k for $0 \leq k < 51$ give us 51 **linear** equations[1] in the 58 unknown variables b_i for $2 \leq i < 11$ and $46 \leq i < 95$.
- By considering the equation for $\mathcal{F}(B_t)$ from $12 \leq t < 27$, we can obtain an additional 15 non-linear equations of degree ≤ 3 in the same 58 variables.
- This gives us 66 equations in 58 variables which we can solve to get the full cipher internal state for clocks $2 \leq t < 18$, allowing us to **recover the full key** with an additional 15-bit check.

[1] We manage to reduce the non-linear terms in h to linear terms by substituting in known values of $\overline{s_t^k}$ and $\overline{b_{95}^k}$ for $0 \leq k < 51$

DAPA on 32-bit Implementation of Grain-128AEADv2

- The 32-bit implementation of Grain-128AEADv2 helps it to gain some resistance against DAPA.
- Have to consider the cumulative effect of all taps within each 32-bit step, instead of isolating the effect of our target difference at the desired clock.
- Any power analysis we perform beyond the first step will have to take the undesired contributions of differences through non-linear functions (particularly via taps at registers s_{94} and b_{95}) into account, which greatly increases the complexity of any potential attack.

DAPA on 32-bit Implementation of Grain-128AEADv2

A Naive Effort: 64-bit Key Recovery

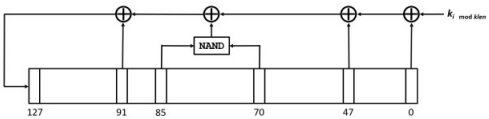
- We aim to make use of only the first 32-bit step in the initialization phase.
- Similar to Phase 2 in the 1-bit implementation attack, we aim to target registers b_{12}^k and b_{95}^k by varying s_8^k and s_{42}^k respectively.
- Example: If we vary s_8^0 to determine b_{12}^0, the difference will be propagated through s_0^8, s_7^1, and possibly s_8^0 (if and only if $b_{12}^0 = 1$ by Observation 1).
  - Each propagated difference will contribute ± 1 to the cumulative power difference across the first 32-bit step, thus their combined contributions will be of even parity.

$$\therefore s_{12}^k = \begin{cases} 0 & \text{if the observed power difference is even} \\ 1 & \text{if the observed power difference is odd} \end{cases}$$

- Can recover NFSR register values b_{12}^k and b_{95}^k for $0 \leq k < 32$.

TinyJAMBU v2

Specifications



We target the NFSR update clock cycles in the second iteration of the **Nonce Setup phase** of the scheme.

Nonce Setup phase:

for i from 0 to 2:

$s_{\{36-38\}} = s_{\{36-38\}} \oplus FrameBits_{\{0-2\}}$

Update the state 640 times using the NFSR

$s_{\{96-127\}} = s_{\{96-127\}} \oplus IV_{\{96-127\}}$

end for

Properties

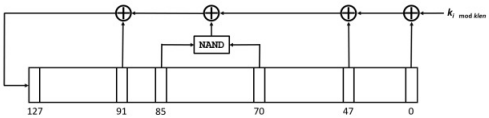
1. Low level of non-linearity; contains only one non-linear (quadratic) function.

2. IV is not loaded into the register directly during initialization, but over multiple iterations 32-bits at a time with intermediate 640 state updates in between.

3. The non-linear and linear contributions to the cipher come from the same register; there does not exist a component that has strictly linear feedback.

Properties 2 and 3 help TinyJAMBU to be **slightly more resistant** against DAPA than Grain-128AEADv2.

DAPA on 1-bit Implementation of TinyJAMBU

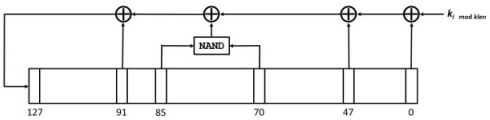


Observation 1.

In Cases 2.1 and 2.2, there is a change in the parity of the net power trace differences if and only if $x = 1$.

Similarly to Phase 2 of our DAPA approach on 1-bit Grain-128, we target the non-linear term here by triggering a difference in s_{85}^t to derive s_{70}^t using Observation 1, and vice versa.

DAPA on 1-bit Implementation of TinyJAMBU

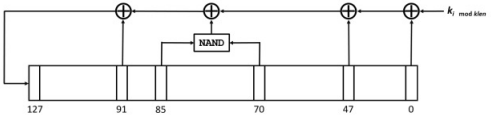


- Here, we have to be mindful to account for the power difference contributions by other potential interfering registers, and account for the parity accordingly.

Clock t	Set of indices i such that s_i^k is active
5	$\{\mathbf{91}\}$
11	$\{\mathbf{85}, 122\}$
26	$\{\mathbf{70}, 107, \underline{113}\}$
42	$\{54, \mathbf{91}, \underline{97}, \underline{112}\}$
48	$\{48, \mathbf{85}, \underline{\mathbf{91}}, \underline{106}, 122\}$
63	$\{33, \mathbf{70}, \underline{76}, \underline{\mathbf{91}}, 107, \underline{113}, 114, \underline{117}\}$

Table: TinyJAMBU Keyed Permutation Active Registers at Clock t for Various Nonce Pairs p_k.

DAPA on 1-bit Implementation of TinyJAMBU



- Consider nonce pairs p_k for $0 \leq k < 32$, and extract information at the clocks listed in the table.
- Derive the internal state of 122 contiguous registers s_{81} to s_{202}, with an additional 60-bit check.
- Guess 6-bits for a full internal state, then use Simple Power Analysis (SPA) to recover the full key during the 640 NFSR clock phase.

Clock t	Set of indices i such that s_i^k is active
5	$\{\mathbf{91}\}$
11	$\{\mathbf{85}, 122\}$
26	$\{\mathbf{70}, 107, \underline{113}\}$
42	$\{54, \mathbf{91}, \underline{97}, \underline{112}\}$
48	$\{48, \mathbf{85}, \mathbf{91}, \underline{106}, 122\}$
63	$\{33, \mathbf{70}, \underline{76}, \mathbf{91}, 107, \underline{113}, 114, \underline{117}\}$

Table: TinyJAMBU Keyed Permutation Active Registers at Clock t for Various Nonce Pairs p_k.

DAPA on 32-bit Implementation of TinyJAMBU

The 32-bit implementation of TinyJAMBU helps it to mitigate the effectiveness of DAPA, perhaps even more severely than Grain-128AEADv2.

Mitigating Factors:

- Have to account for cumulative power differences across all affected registers, particularly those affected by non-linearity.
- Denser tap points that are more likely to be tapped earlier, thus propagating the differences of unintended intermediate registers faster.
- Only 32 bits of IV can be reasonably considered at once, leaving even less room for manoeuvrability.

DAPA on 32-bit Implementation of TinyJAMBU

A Naive Effort: 32-bit Secret State Recovery

- Once again, exploit Observation 1 to target registers at the non-linear tap, s_{70} and s_{85}.
- For nonce pairs p_k where $102 \leq k < 117$, we can clock them through a single 32-bit step to affect only one non-linear tap, s_{85}^{k-85}, and recover the corresponding 15 bits of the internal state s_{70}^{k-85}.
- For nonce pairs p_k where $112 \leq k < 117$, we can clock them through two 32-bit steps to affect only one non-linear tap, s_{70}^{k-70}, at the second 32-bit step, and recover the corresponding 5 bits of the internal state s_{85}^{k-70}.
- We can recover an additional 12 bits of the internal state from nonce pairs p_k where $96 \leq k < 102$, by considering **rectangular differentials**.

Rectangular Differentials

- Rectangular Differentials are an extension of DAPA which considers the net effect of multiple differentials on multiple unknowns over a 2-dimensional differential subspace.
- Consider a set of 4 cipher instantiations with 4 different nonce values: the zero nonce, and the nonces n_a, n_b, and n_{ab}, which differ at registers a, b, and both a and b respectively.
- Let the nonce pair $(0, n_i)$ be denoted p_i. Let $96 \leq a < 102, b = a + 21$, and let $\mathcal{P}_i$ be the power trace difference for nonce pair p_i at clock $t = a - 70 = b - 91$. We then have the following table, from which we can uniquely derive s_{85}^{a-70} and s_{70}^{a-85}.

$(\mathcal{P}_a \mod 2, \mathcal{P}_{ab} - \mathcal{P}_a)$	$\left(s_{85}^{a-70}, s_{70}^{a-85}\right)$
$(0, \mathcal{P}_b)$	$(1, 0)$
$(0, -\mathcal{P}_b)$	$(0, 1)$
$(1, \mathcal{P}_b)$	$(0, 0)$
$(1, -\mathcal{P}_b)$	$(1, 1)$

Table: Rectangular Differential Table for $96 \leq a < 102$, $b = a + 21$.

Conclusion

Conclusion

- 32-bit implementations are more robust!
- Delayed injection of IV helps TinyJAMBU gain some resistance.
- Recommended to apply additional Side-Channel Protection even if a scheme seems inherently resistant to SCA.
- Can possibly extend DAPA and incorporate other traditional cryptanalysis methods into SCA.
  - In this work: Rectangular Differentials.
  - Future work: Consider using Integral Properties? Larger Differential Subspaces?

Thank you!

References I

S. M. Sim, D. Jap, and S. Bhasin, "DAPA: differential analysis aided power attack on (non-) linear feedback shift registers," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2021, no. 1, pp. 169–191, 2021.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 1: Deduce internal states of LFSR registers

This difference gets propagated via $\mathcal{L}(S_k)$ to register s_{127}^{k+1}.

- We need to ensure that we can accurately observe this propagated difference.
- This propagated difference will be isolated and immediately detected by the power measurement differences if there is no interference from any other terms in the feedback functions $\mathcal{L}(S_k)$ and y^k at clock $t = k$.
- **Assumption:** The difference injected in the nonce pair p_k does not propagate to a difference in any other term in the feedback functions $\mathcal{L}(S_k)$ and y^k, besides the term s_0^k, at clock $t = k$.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 1: Deduce internal states of LFSR registers

We can use the differential pattern in Case 1.1 to extract the value of s_{127}^{k+1}

- If a difference is induced in s_{127}^{k}, then comparing s_{127}^{k} and s_{127}^{k+1} would result in Case 1.3, which gives us no information about the value of s_{127}^{k+1}.
- We have to ensure that no difference propagates to s_{127}^{k}.
- **Assumption:** The difference injected in the nonce pair p_k does not propagate to a difference in any term in the feedback functions $\mathcal{L}(S_{k-1})$ and y^{k-1}, at clock $t = k - 1$.

DAPA on 1-bit Implementation of Grain-128AEADv2

Phase 1: Deduce internal states of LFSR registers

Nonce Pair p_k	Set of active registers at clock $t = k$
$0 \leq k < 7$	$S_1 = \{\mathbf{s_0}\}$
$k = 7$	$S_2 = S_1 \cup \{\mathbf{s_{121}}\}$
$8 \leq k < 38$	$S_3 = S_2 \cup \{\underline{b_{120}}, \underline{\mathbf{s_{120}}}\}$
$39 \leq k < 42$	$S_4 = S_3 \cup \{\mathbf{s_{90}}, \mathbf{s_{122}}, \underline{b_{124}}, \underline{\mathbf{s_{124}}}, b_{125}, \mathbf{s_{125}}\}$
$42 \leq k < 70$	$S_5 = S_4 \cup \{\underline{b_{86}}, \underline{\mathbf{s_{86}}}\} \cup S_{118+}$

Nonce Pair p_k	Set of active registers at clock $t = k - 1$
$0 \leq k < 7$	$S_1 = \{\mathbf{s_1}\}$
$k = 7$	$S_2 = S_1 \cup \{\mathbf{s_{122}}\}$
$8 \leq k < 38$	$S_3 = S_2 \cup \{\underline{b_{121}}, \underline{\mathbf{s_{121}}}\}$
$39 \leq k < 42$	$S_4 = S_3 \cup \{\mathbf{s_{91}}, \mathbf{s_{123}}, \underline{b_{125}}, \underline{\mathbf{s_{125}}}, b_{126}, \mathbf{s_{126}}\}$
$42 \leq k < 70$	$S_5 = S_4 \cup \{\underline{b_{87}}, \underline{\mathbf{s_{87}}}\} \cup S_{119+}$