Analysis of The SPEEDY Family of Block Ciphers

Murat Can Oruntak, 2214518
Middle East Technical University

Keywords: Cryptology, Cryptanalysis, Lightweight Cryptography, Speedy, Block Ciphers, Lightweight Block Ciphers

Abstract: In this paper we will analyse SPEEDY-r-192 We will briefly mention about designers' rationale, motivation behind the design, specifications of the algorithm etc. We will also evaluate the papers that have written about performance or cryptanalysis of SPEEDY-r-192

INTRODUCTION

SPEEDY is a family of family of ultra-low latency block ciphers developed by Gregor Leander, Thorben Moos, Amir Moradi and Shahram Rasoolzadeh. Their fundamental motivation was "How to design a secure encryption algorithm which also has fast hardware implementation". For solving this problem designers primarily focused for an ultimate low latency in CMOS hardware. To achieve this goal, they rely on engineering expertise in each design step. Different from other low latency ciphers like PRINCE, PRINCEv2, MANTIS and QARMA they ignored the factors like decryption speed or area and energy efficiency and mainly focused on encryption speed and security. In accordance with these choices their primarily focus is High End CPU's which prioritizes high performance before energy and area restrictions. The SPEEDY family has 3 different version for different security needs which SPEEDY-r-192 with r ∈ {5, 6, 7} r represents the number of rounds and can be used according to trade of between security and performance.

1. Motivation

In recent years, securing the internals of Central Processing Units (CPUs) has become a serious concern. The recent microarchitectural attacks such as Spectre (Kocher, 2020) and Meltdown (Lipp, 2020) show us significant problems in the security architectures of widely used high-end processors. Hardware-based mitigations for such attacks are happening widely, with many of them focusing on stronger encrypted communication both inside of CPU's and between surrounding hardware components and CPU's. In order to achieve this goal some secure caches developed such as CEASER (Qureshi, 2018) and ScatterCache (Werner, 2019). Besides secure caches mechanisms like; memory encryption, dedicated hardware instructions and pointer authentication could also help to solve this problem. High-speed encryption primitives are critical for integrating these features into next-generation processors without significantly impacting performance.

Primary motivation of SPEEDY is becoming a general-purpose fast encryption primitive suitable for all of these applications rather than being restricted or specialized for specific use cases.

2 Related Work and Unique Contribution

Even though designing cryptographic primitives optimized for minimal execution time in hardware is still a relatively new and developing field, there are already some examples, such as PRINCE (Borghoff, 2012), PRINCEv2 (Božilov, 2020), QARMA (Avanzi, 2017) and MANTIS (Beierle, 2016). Even they are great designs in the past neither them or other related works never focused on the Gate and transistor level latency characteristics of the hardware in the design level however SPEEDY prioritized these characteristics. Also, previous works designed to be area and energy efficient and focused on offering both encryption and decryption

in one circuit. On the other hand, SPEEDY aims to maximize encryption speed and security, putting other concerns in the background. Extensive testing across six standard cell libraries shows that SPEEDY-5-192 and SPEEDY-6-192 outperform all other known encryption primitives in hardware latency, while SPEEDY-7-192 is only slightly slower than PRINCE. Given its security levels, this represents a major advancement in the field of ultra-low-latency cryptography.

3. Hardware Specifications and Rationale

Designing a low-latency encryption primitive for hardware involves creating a circuit that establishes a highly non-linear relationship between the plaintext and as many independent key bits as possible, all while operating as quickly as possible.

Since CMOS is the standard for integrated circuit fabrication for a very long-time designers of SPEEDY focused on a circuit representation constructed entirely from CMOS gates. To provide desired features mentioned above with minimum latency they heavily rely on performance of logic gates and logic circuits. In the light of results from "Fan-In-to-Latency-Ratio and Linearity of logic gates in NanGate 45nm Open Cell Library (OCL)" tests they decided use of NAND and OAI gates, which are the best choices for achieving low-latency performance in hardware. Also, with prioritizing minimal fan-out in its logic circuits, SPEEDY successfully reduces delays caused by signal propagation.

Also, there are two critical myths about hardware instructions which are;

1) "Every CMOS standard cell has a fixed delay, and using the same standard cell repeatedly will always contribute the same amount of latency to a signal path"

But this is wrong the delay of a CMOS cell depends on the transition time of its input signals and the capacitive load that the cell needs to drive at its output. Additionally, the delay can vary significantly due to changes in the cell's electrical environment, often by 200-300%.

2) "Adding a gate to a circuit path will always increase the path's latency"

But this is also wrong. Placing a buffer or inverter in the right spot (when logically appropriate) can help charge large capacitive loads more efficiently, which can reduce the overall latency of the path. Therefore, the number of gates (gate depth) does not always directly determine a circuit's latency.

By focusing on these low-level hardware characteristics, SPEEDY can deliver high-speed encryption while maintaining a strong security.

4. Ultra Low-Latency 6-bit S-box

To create an S-box optimized for both high speed in CMOS hardware and strong cryptographic properties, the designers used to the following criteria

- Ultra-low latency: Since NAND and OAI gates were decided as the most effective gates for achieving low latency. The design focuses on using only these gate types while minimizing the number of logic levels required, also the design ensures minimal fan-out (number of outputs driven by a gate) across as many stages as possible to achieve ultra-low latency.

- • Bijective mapping with fully dependent outputs: As required for an SPN cipher, the S-box is designed to implement a bijective mapping, which means each input uniquely maps to an output. Furthermore, to limiting the search to the S-box with fully dependent outputs we ensure that every input bit contributes to the computation of each output bit.

- Small linearity and uniformity: To improve the resistance to the linear and differential attacks designers try to keep the uniformity (u) and linearity (l) at minimum level.

$$u = Uni(S) \coloneqq \max_{\alpha,\beta \in F_2^n, \alpha \neq 0} \{ x \in F_2^n \mid S(x) \oplus S(x \oplus \alpha) = \beta \}$$

$$l = Lin(S) \coloneqq \max_{\alpha,\beta \in F_2^n, \beta \neq 0} \left| \sum_{x \in F_2^n} (-1)^{\langle \alpha, x \rangle \oplus \langle \beta, S(x) \rangle} \right|$$

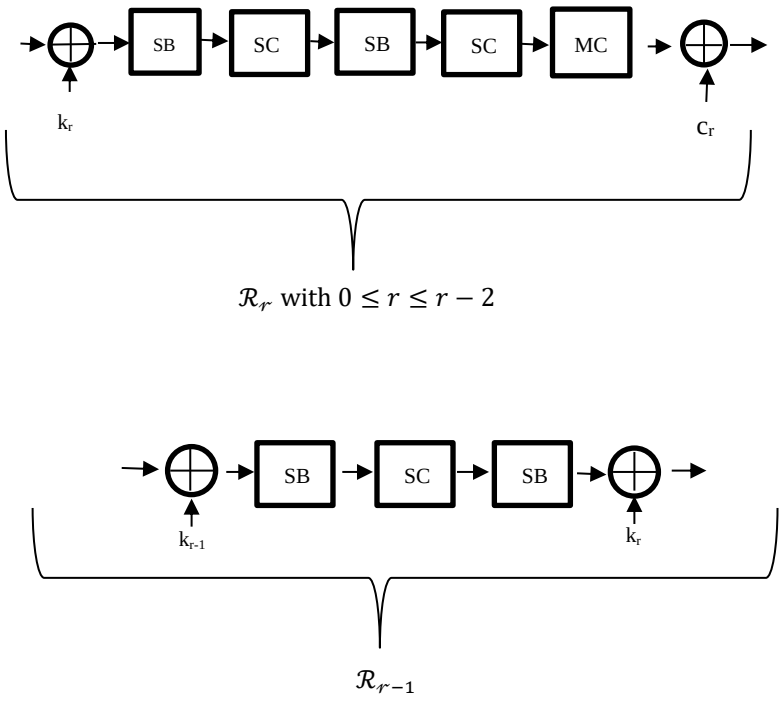
With following these criterias SPEEDY's unique S-Box achieves minimum linearity of 24, minimum uniformity of 8 and full diffusion.

5. Round Function of SPEEDY

The cipher processes a 6ℓ - bit plaintext by initializing the internal state using the same order as the bit indexing. It fills the state sequentially, starting with $X_{[0,0]}$ then $X_{[0,1]}$ and so on. The cipher applies r round functions $\mathbf{R}_r$ (where $0 \leq r < \mathbf{R}$) to the internal state, with the first $r - 1$ rounds of which identical, except for the round keys and constants. Each round function consists of following operations:

1. **SubBox (applied twice)**: Provides non-linearity.
2. **ShiftColumns (applied twice)**: Ensures diffusion by rearranging columns.
3. **MixColumns**: Introduces further diffusion by mixing columns.
4. **AddRoundConstant and AddRoundKey**: Adds constants and keys to the state.

Below you can see the figure representing the whole cipher;



6. Security Analysis

Although SPEEDY can operate with various block and key sizes, the default version is set to 192 bits because it is the least common multiple of 6 (SPEEDY's S-Box width) and 64 (instruction width of high-end CPUs). For SPEEDY-r-192 security claims of 5,6,7 rounds shown in the table below.

Speedy-5-192	$\geq 2^{128}$ time complexity when data complexity is limited to $\leq 2^{64}$
Speedy-6-192	$\geq 2^{128}$ time complexity
Speedy-7-192	$\geq 2^{192}$ time complexity

6 and 7 round versions of SPEEDY are already secure for today and 5 round version is still provides enough security for many practical applications. When we compare the security claim of SPEEDY-5-192 with PRINCE (Borghoff, 2012) or PINCEv2 (Božilov, 2020) it is clearly seen that SPEEDY-5-192 has better security properties.

7. Cryptanalysis and Performance Analysis of SPEEDY

In this section we will evaluate and summarize third-party articles written about SPEEDY's performance analysis and security (cryptanalysis).

7.1 Grover on SPEEDY

Grover's search algorithm is quantum algorithm which significantly speeding up exhaustive key search attacks on symmetric key cryptographic systems (Grover, 1996) In this paper a quantum circuit for the SPEEDY block cipher was presented, with resource estimation based on SPEEDY-7-192 for performing a key search attack to evaluate its NIST security strength. The analysis shows that SPEEDY-7-192 achieves a security strength equivalent to level 1 (same with AES-128) under NIST standards. Even SPEEDY-7-192 provides 192-bit security on classical computers, this level of security is not provided on quantum computers. Moreover, changing the number of rounds in SPEEDY does not enhance its security against quantum attacks. To ensure security against quantum computers, the key size would need to be increased according to Grover's algorithm. (Song, 2021)

7.2 Better steady than speedy: full break of SPEEDY-7-192

In this paper writers attacked all three variants of SPEEDY Family Block Ciphers and fully break the SPEEDY-7-192 version but attacks on other versions like SPEEDY-5-192 and SPEEDY-6-192 does not controvert with the designer's security claims. The attack on SPEEDY-7-192 relies on a 5.5 round distinguisher and can be extended to 7 rounds which contradicts with the designers claim: "the attacker cannot add more than one round to extend a distinguisher". The attack has a data complexity of $2^{187.28}$ and time complexity of $2^{187.84}$ which is clearly contradicts with the 192 bit security claim in the original paper and approximately 16 times faster than exhaustive search (Boura, 2023). Even though these results contradict with the security claims of SPEEDY-7-192 they are not applicable so the practical security of SPEEDY-7-192 remains secure.

7.3 Cryptanalysis of SPEEDY

In this paper authors relied on divide-and-conquer strategy and proposed an attack on 4 round of SPEEDY-5-192 and full round of SPEEDY-7-192 with leveraging key-recovery-friendly distinguishers. Their strategy for search includes two parts which are: 1) Manage the propagation of active patterns derived from input and output differences or masks. In order to achieve that they use a TDTT (truncated differential distribution table) to control the propagation of active patterns derived from the input and output differences/masks. This table highlights the non-random behaviour of the S-box with bit-wise truncated differential propagation, aiding the search algorithm in selecting appropriate input and output differences for the target distinguisher. 2) Build long distinguishers using a divide-and-conquer approach. To construct long distinguishers, they concatenate smaller distinguishers, each covering a few rounds. For example, the 5.5-round distinguisher is built starting with a 4-round distinguisher. To improve the 4-round distinguisher, it is split into two 2-round parts. The input difference for the Mix Column operation in the middle is carefully chosen to minimize the total number of activated S-boxes in its neighboring S-box layers. For the other Mix Columns in the 5.5-round distinguisher, there are no specific restrictions on their input differences. Their linear attack for 7 Round has resulted with $2^{189.41}$ time complexity $2^{185.04}$ memory complexity and $2^{188.50}$ (known plaintext) data complexity. Their differential attack for 7 Round has resulted with $2^{187.39}$ time complexity 2^{156} memory complexity for chosen plaintext setting (but if they use the chosen ciphertext attack mode and choose the active 36-bit structure it can reduce to 2^{36}), and $2^{186.53}$ data complexity. Their differential-linear attack for 4 rounds of SPEEDY-5-192 has resulted with 2^{61} data complexity 2^{105} time complexity and 2^{105} memory complexity. Their differential attack for 4 rounds of SPEEDY-5-192 has resulted with 2^{61} data complexity $2^{119.69}$ time complexity and 2^{183} memory complexity (Wang, 2023).

7.4 Efficient Implementation of SPEEDY Block Cipher on Cortex-M3 and RISC-V Microcontrollers. Mathematics

In this paper writers implemented SPEEDY on a 32-bit microcontroller first time with applying bit-slicing techniques. They do it on both **ARM Cortex-M3** and **RISC-V**. On ARM Cortex-M3, SPEEDY-7-192, SPEEDY-6-192 and SPEEDY-5-192 achieved 85.16 cpb, 75.25 cpb and 65.7 cpb which are better results than 104.1 cpb for AES-128 and 120.4 cpb for GIFT-128 in the same environment. On RISC-V SPEEDY-5-192, SPEEDY-6-192 and SPEEDY-7-192 reached 109.2 cpb, 95.5 cpb and 81.9 cpb even though this time SPEEDY-6-192 which has similar security levels to AES-128 shown slightly worse performance than AES-128 it has still 165x faster than the C implementation of SPEEDY-7-192. These results shows us SPEEDY can run efficiently in software with the right conditions and can be applied to microcontrollers. Also the implementation proposed in this paper works with constant timing that provides advantage against timing attacks. (Kim, 2022)

7.5 Cryptanalysis of reduced round SPEEDY

In this paper authors provided the first outside security assessment for SPEEDY family. They introduce practical distinguishers for 2 rounds (total 32 +32) with 2^{13} and 2^{14} data complexities. And key recovery attacks extending to 3 rounds with; $2^{25.5}$ bit memory, $2^{17.6}$ data and $2^{52.5}$ time requirements for all three versions of SPEEDY Block Ciphers. Even

though these contributions might not seem entirely novel they cover 50% and 60% of SPEEDY-6-192 and SPEEDY-5-192 rounds for the first time at the publication of this paper (Rohit, 2022).

7.6 Observations on the branch number and differential analysis of SPEEDY

In this paper authors used partition branch number (pbnr) to measure the minimum number of active S-Boxes. This metrics helps to evaluate how well SPEEDY spreads input differences across rounds, which is a critical speciality for resisting against differential cryptanalysis attacks. The analysis shows us that actual number of minimum active S-Boxes is significantly smaller than declared in the original article.

$$pbn2 = 13, pbn3 = 23, pbn4 = 35, pbn5 = 45, pbn6 = 57, pbn7 = 67$$

These results will contradict the security margin claimed by the authors. Furthermore when we compared this method with other time consuming search methods accurate results can be obtained for any number of rounds, even when automatic searches become impractical for longer rounds

In the light of these information they conduct a search for optimal differential characteristics of SPEEDY. But instead of assuming every differential transition through S-Box is possible they choose to use Difference Distribution Table (DDT) for the 6-bit S-Box. According to results of *higher-order partition branch number* they found differential probabilities which are;

$2^{-46.2}$, $2^{-76.72}$, $2^{-129.2}$, $2^{-170.0}$, $2^{-216.0}$, and $2^{-266.2}$

for 2-7 rounds SPEEDY.

Lastly with using SPEEDY's simple bit permutation key schedule they extended the differential key trail search method and create a 6 round related-key differential trail with probability of $2^{-179.2}$ with that they can conduct a related key differential attack on full round SPEEDY-7-192 with time complexity of $2^{160.13}$ encryption and data complexity of $2^{186.2}$ chosen plaintext (Zhang, 2024).

8 CONCLUSIONS

To conclude; SPEEDY is a family of ultra-low latency Block Ciphers. It developed for low latency in CMOS hardware. Primary focus of the SPEEDY is security in modern CPU's which includes many applications requires for low latency encryption like memory encryption, pointer authentication or secure caches. SPEEDY is a SPN (SPS) type cipher with a 6-bit width S-Box. It can work with different key and block sizes but since greatest common multiple of 6 and 64 is 192 the default version works with 192-bit key and block sizes. It can iterate over 5-6-7 rounds; all have different performance and security metrics and suitable for different conditions. It's performance metrics shows us SPEEDY outdone it's competitors for any application that prioritizes high performance over area and energy consumption. Lastly even there are some studies that contradicts with SPEEDY's security claim they are not applicable in real life and SPEEDY is secure for known attacks in practice.

REFERENCES

Avanzi, R. (2017). The QARMA block cipher family. Almost MDS matrices over rings with zero divisors, nearly symmetric even-mansour constructions with non-involutory central rounds, and search heuristics for low-latency s-boxes. *IACR Transactions on Symmetric Cryptology*, 4-44.

Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., ... & Sim, S. M. (2016). The SKINNY family of block ciphers and its low-latency variant MANTIS. In Advances in Cryptology–CRYPTO 2016: 36th Annual International Cryptology

Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E. B., Knezevic, M., Knudsen, L. R., ... & Yalçın, T. (2012). PRINCE–a low-latency block cipher for pervasive computing applications. In Advances in Cryptology–ASIACRYPT 2012: 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2-6, 2012. Proceedings 18 (pp. 208-225). Springer Berlin Heidelberg.

Boura, C., David, N., Heim Boissier, R., & Naya-Plasencia, M. (2023, April). Better steady than speedy: full break of SPEEDY-7-192. In Annual International Conference on the Theory and Applications of Cryptographic Techniques (pp. 36-66). Cham: Springer Nature Switzerland.

Božilov, D., Eichlseder, M., Knežević, M., Lambin, B., Leander, G., Moos, T., ... & Wiemer, F. (2021). PRINCEv2: more security for (almost) no overhead. In *Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers 27* (pp. 483-511). Springer International Publishing.

Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219).

Kim, H., Eum, S., Sim, M., & Seo, H. (2022). Efficient Implementation of SPEEDY Block Cipher on Cortex-M3 and RISC-V Microcontrollers. Mathematics, 10(22), 4236.

Kocher, P., Horn, J., Fogh, A., Genkin, D., Gruss, D., Haas, W., ... & Yarom, Y. (2020). Spectre attacks: Exploiting speculative execution. *Communications of the ACM*, *63*(7), 93-101.

Leander, G., Moos, T., Moradi, A., & Rasoolzadeh, S. (2021). The speedy family of block ciphers-engineering an ultra low-latency cipher from gate level for secure processor architectures. Cryptology ePrint Archive.

Lipp, M., Schwarz, M., Gruss, D., Prescher, T., Haas, W., Horn, J., ... & Strackx, R. (2020). Meltdown: Reading kernel memory from user space. *Communications of the ACM*, *63*(6), 46-56.

Qureshi, M. K. (2018, October). CEASER: Mitigating conflict-based cache attacks via encrypted-address and remapping. In *2018 51st Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)* (pp. 775-787). IEEE.

Rohit, R., & Sarkar, S. (2022, July). Cryptanalysis of reduced round SPEEDY. In International Conference

on Cryptology in Africa (pp. 133-149). Cham: Springer Nature Switzerland.

Song, G., Jang, K., Kim, H., Eum, S., Sim, M., Kim, H., ... & Seo, H. (2021). Grover on SPEEDY. Cryptology ePrint Archive.

Wang, J., Niu, C., Liu, Q., Li, M., Preneel, B., & Wang, M. (2023, June). Cryptanalysis of SPEEDY. In Australasian Conference on Information Security and Privacy (pp. 124-156). Cham: Springer Nature Switzerland.

Werner, M., Unterluggauer, T., Giner, L., Schwarz, M., Gruss, D., & Mangard, S. (2019). {ScatterCache}: thwarting cache attacks via cache set randomization. In 28th USENIX Security Symposium (USENIX Security 19) (pp. 675-692).

Zhang, L. (2024). Observations on the branch number and differential analysis of SPEEDY. Designs, Codes and Cryptography, 92(5), 1175-1199.