An Ultra-High Throughput AES-Based Authenticated Encryption Scheme for 6G: Design and Implementation

Ravi Anand[1(✉)], Subhadeep Banik[2], Andrea Caforio[3], Kazuhide Fukushima[4], Takanori Isobe[1(✉)], Shisaku Kiyomoto[4], Fukang Liu[1], Yuto Nakano[4], Kosei Sakamoto[1], and Nobuyuki Takeuchi[1]

[1] University of Hyogo, Kobe, Japan
ravianandsps@gmail.com, takanori.isobe@ai.u-hyogo.ac.jp
[2] Universita della Svizzera Italiana, Lugano, Switzerland
[3] École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
[4] KDDI Research, Inc., Saitama, Japan

Abstract. In this paper, we propose Rocca-S, an authenticated encryption scheme with a 256-bit key and a 256-bit tag targeting 6G applications bootstrapped from AES. Rocca-S achieves an encryption/decryption speed of more than 200 Gbps in the latest software environments. In hardware implementation, Rocca-S is the first cryptographic algorithm to achieve speeds more than 2 Tbps without sacrificing other metrics such as occupied silicon area or power/energy consumption making Rocca-S a competitive choice satisfying the requirements of a wide spectrum of environments for 6G applications.

Keywords: Authenticated Encryption · High Throughput · Quantum Security · 6G

1 Introduction

The imminent global standardization of 5G telecommunication networks marks an important turning point for the involved research community whose gaze should hereafter be directed beyond the 5G era. The 6Genesis project kickstarted this endeavour with a white paper [9] in which 6G channels are projected to provide throughput rates upwards of 100 Gbps in software eclipsing their 5G counterparts by more than an order of magnitude. Concerning peak throughput, the paper further illuminates potential avenues that would allow for rates in the Terabit range and states: "*6G research should look at the problem of transmitting up to 1 Tbps per user.*"

Naturally, performance only covers half of the requirements for a prospective 6G standard with the other one being security as discussed by the 3GPP standardization organization which examined the possible impacts of quantum computing in the coming years especially due to Grover's algorithm. The motivation is to design a cipher that provides 256-bit classical security and 128-bit

G. Tsudik et al. (Eds.): ESORICS 2023, LNCS 14344, pp. 229–248, 2024.
https://doi.org/10.1007/978-3-031-50594-2_12

quantum security for key recovery. Apart from the 256-bit key size, proposing an AEAD scheme with a suitable tag length is also important to ensure the security against the state-recovery attack. Specially, if the tag size is smaller than 256 bits, it is possible to use the decryption oracle to mount a fast state-recovery attack with time complexity smaller than 2^{256} [6,22], especially when each message block is absorbed by a weak permutation as in AEGIS-256 [22] and Rocca [19]. By using a keyed permutation for the initialization phase as in AEGIS-256 and the revised version of Rocca [20], while the key-recovery attack can be prevented even if the state-recovery attack succeeds, it may be still a potential threat to the AEAD scheme since attackers can extract more information from the full secret state with this state-recovery attack. In this sense, we believe it is meaningful to design an AEAD scheme with a 256-bit tag.

Motivation. To the best of our knowledge, none of the existing algorithms dedicated to 5G and beyond provide throughput rates of more than 1 Tbps, more than 100 Gbps in software, and support 256-bit tags, including the AEGIS [22], Tiaoxin-346 [17], and Rocca [19,20] as well as 5G ciphers such as ZUC-256 [21] and SNOW-V [3]. This fact motivates a search for new algorithms which meets all three of these requirements for 6G applications.

Contributions and Organization

In this paper, we propose Rocca-S, which is an AES-based authenticated encryption scheme with a 256-bit key and a 256-bit tag, which provides 256- and 128-bit security for key recovery attacks against classical and quantum adversary, respectively. One of the main contribution is to design new round functions supporting 256-bit tags, without sacrificing performance. Rocca-S achieves more than 200 Gbps in latest software environments and more than 1 Tbps in hardware. Rocca-S is the first algorithm that achieves both requirements for 6G, namely a 256-bit tag and throughput rates beyond 1 Terabit. The specification of Rocca-S is given in Sect. 2.

– The design rationale is explained in Sect. 3. The most difficult challenge is to design round functions supporting 256-bit tag without sacrificing performance. To accomplish it, we take advantage of an interesting insight that while increasing the number of aesenc from 4 (Rocca's case) to 6 in the round function, the overhead in software performance can be made negligible by reducing state size from 8 to 7. From a hardware point of view, as these are executed in parallel, there is no overhead regarding throughput, and small state size rather reduces the circuit area. This allows us to add more nonlinear operations into each round and open up new design space to increase the security against forgery attacks while keeping the performance. Our comprehensive large-scale search with MILP-aided tools enables finding a very small class of round functions that guarantee the sufficient level of security against forgery attacks and competitive performance in software and hardware.

– We discuss the security analysis of Rocca-S in Sect. 4. We show that Rocca-S is secure against several classes of attacks by classic and quantum adversaries.
– The software performance is discussed in Sect. 5. We show that Rocca-S outperforms existing algorithms and achieves a throughput of 205 Gbps in AEAD mode on a machine equipped with an Intel(R) Core(TM) i9 12900K.
– Ultimately, we meticulously explore the design space of ASIC implementations by proposing several constructions ranging from round-based circuits to unrolled variants in Sect. 6. In particular, the round-based and unrolled circuits achieve throughput rates that exceed 1 Terabit per second (even crossing the 2 Tbps barrier for some instances) and thus eclipse related AEAD circuits by at least 50% without sacrificing other metrics such as occupied silicon area or power/energy consumption making Rocca-S a competitive choice satisfying the requirements of a wide spectrum of environments.

2 Specification

Throughout this paper, a block denotes a 16-byte value. S is the state of Rocca-S, which is composed of 7 blocks of the form $S = (S[0], S[1], \ldots, S[6])$, where $S[i]$ $(0 \leq i \leq 6)$ are blocks and $S[0]$ is the first block. $\mathsf{AES}(X, Y)$ is defined as:

$$\mathsf{AES}(X, Y) = (\text{MixColumns} \circ \text{ShiftRows} \circ \text{SubBytes}(X)) \oplus Y,$$

where MixColumns, ShiftRows, and SubBytes are the same operations as that of AES. $A(X)$ is $A(X) = \text{MixColumns} \circ \text{ShiftRows} \circ \text{SubBytes}(X)$. $R(S, X_0, X_1)$ is the round function used to update the state S. We utilize the same constants as in Tiaoxin-346 [17], namely $Z_0 =$ `428a2f98d728ae227137449123ef65cd` and $Z_1 =$ `b5c0fbcfec4d3b2fe9b5dba58189dbbc`.

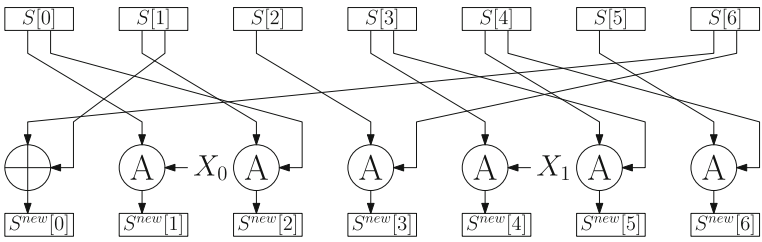

Fig. 1. Round function of Rocca-S (RF-1)

2.1 The Round Update Function

The input of the round function $R(S, X_0, X_1)$ of Rocca-S consists of the state S and two blocks (X_0, X_1). $S^{new} \leftarrow R(S, X_0, X_1)$ is illustrated as Fig. 1 and defined as:

$S^{new}[0] = S[6] \oplus S[1]$, $S^{new}[1] = \mathsf{AES}(S[0], X_0)$,$S^{new}[2] = \mathsf{AES}(S[1], S[0])$,
$S^{new}[3] = \mathsf{AES}(S[2], S[6])$, $S^{new}[4] = \mathsf{AES}(S[3], X_1)$,$S^{new}[5] = \mathsf{AES}(S[4], S[3])$,
$S^{new}[6] = \mathsf{AES}(S[5], S[4])$.

2.2 Specification of Rocca-S

Rocca-S is an authenticated-encryption with associated-data scheme composed of four phases: *initialization, processing the associated data, encryption* and *finalization*. The input consists of a 256-bit key $K_0||K_1 \in \mathbb{F}_2^{128} \times \mathbb{F}_2^{128}$, a 128-bit nonce N, the associated data AD and the message M, where $X||Y$ is the concatenation of X and Y. The output is the corresponding ciphertext C and a 256-bit tag T. $|X|$ is the length of X in bits. Define $\overline{X} = X||0^l$ where 0^l is a zero string of length l bits, and l is the minimal non-negative integer such that $|\overline{X}|$ is a multiple of 256. In addition, write X as $X = X_0||X_1||\ldots||X_{\frac{|X|}{256}-1}$ with $|X_i| = 256$. Further, X_i is written as $X_i = X_i^0||X_i^1$ with $|X_i^0| = |X_i^1| = 128$.

Initialization. First, (N, K_0, K_1) is loaded into the state S in the following way:

$$S[0] = K_1, S[1] = N, S[2] = Z_0, S[3] = K_0, S[4] = Z_1, S[5] = N \oplus K_1, S[6] = 0$$

Here, two 128-bit constants Z_0 and Z_1 are encoded as 16-byte little endian words and loaded into $S[2]$ and $S[3]$ respectively. Then, 16 iterations of the round function $R(S, Z_0, Z_1)$ is applied to the state S. After 16 iterations of the round function, two 128-bit keys are XORed with the state S in the following way;

$$S[0] = S[0] \oplus K_0, S[1] = S[1] \oplus K_0, S[2] = S[2] \oplus K_1, S[3] = S[3] \oplus K_0,$$
$$S[4] = S[4] \oplus K_0, S[5] = S[5] \oplus K_1, S[6] = S[6] \oplus K_1.$$

Processing the Associated Data. If AD is empty, this phase will be skipped. Otherwise, AD is padded to $\overline{AD}$ and the state is updated as $R(S, \overline{AD}_i^0, \overline{AD}_i^1)$ for $i = 0$ to $d - 1$, where $d = \frac{|\overline{AD}|}{256}$.

Encryption. If M is empty, the encryption phase will be skipped. Otherwise, M is first padded to $\overline{M}$ and then $\overline{M}$ will be absorbed with the round function. During this procedure, the ciphertext C is generated. If the last block of M is incomplete and its length is b bits, i.e., $0 < b < 256$, the last block of C will be truncated to the first b bits. Each encryption round unfolds as follows:

$$C_i^0 = \mathsf{AES}(S[3] \oplus S[5], S[0]) \oplus \overline{M}_i^0, C_i^1 = \mathsf{AES}(S[4] \oplus S[6], S[2]) \oplus \overline{M}_i^1, R(S, \overline{M}_i^0, \overline{M}_i^1),$$

where $i = 0$ to $m - 1$, and $m = \frac{|\overline{M}|}{256}$.

Finalization. The state S will again pass through 16 iterations of the round function $\mathsf{R}(S, |AD|, |M|)$ and then the 256-bit tag is computed:

$$T = \bigoplus_{i=0}^{3} S[i] || \bigoplus_{i=4}^{6} S[i].$$

The length of associated data and message is encoded as a 16-byte little endian word and stored into $|AD|$ and $|M|$, respectively. A illustration corresponding to the presented procedures is shown in Fig. 2

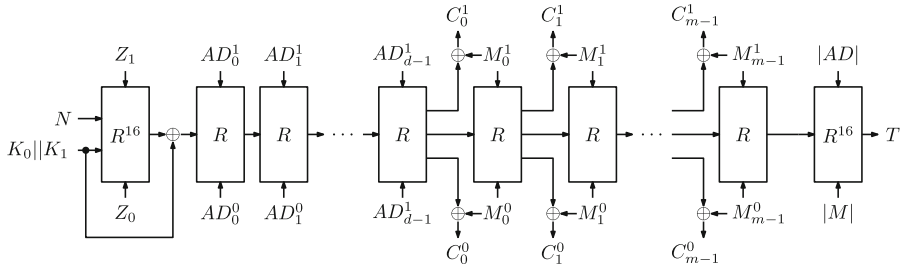

Fig. 2. Overview of Rocca-S

A Raw Encryption Scheme. If the phases of processing the associated data and finalization are removed, a raw encryption scheme is obtained.

A Keystream Generation Scheme. If the phases of processing the associated data and finalization are removed and there is no message injection into round function such that $R(S, 0, 0)$, a keystream generation scheme is obtained.

2.3 Security Claims

Classical Setting. Rocca-S provides 256-bit security against key-recovery and 192-bit security against forgery attacks in the nonce-respecting setting. Rocca-S does not claim security against nonce-misuse setting. We do not claim its security in the related-key and known-key settings. The message length for a fixed key and the number of different messages that are produced for a fixed key are limited to at most 2^{128}. The length of associated data of a fixed key is up to 2^{64}.

Quantum Setting. Rocca-S provides 128-bit key-recovery and forgery security against quantum adversaries with classical online queries. Rocca-S does not claim security against online superposition attacks.

3 Design Rationale

3.1 General Design

The general design of Rocca-S follows the key features of AEGIS family [22], Tiaoxin-346 [17] and Rocca [19,20], i.e., SIMD-friendly round function and efficient permutation-based structure. To further increase the resistance against several attacks on AEGIS and Tiaoxin-346 [11,15] while keeping the performance, we carefully design the nonce and key loading scheme at the initial state and output function. Furthermore, we add key-forward operations in the initialization phase as similar to the suggestion by the recent attacks on Rocca [6],

In our design, we utilize only aesenc as one of the AES-NI instructions, which executes one round of AES with an input state S and a 128-bit round key K:

$$\mathsf{aesenc}(S, K) = (\mathsf{MixColumns} \circ \mathsf{ShiftRows} \circ \mathsf{SubBytes}(S)) \oplus K.$$

Fig. 3 shows the general design of our round function, where A and M denote aesenc and inserted message block, respectively. To maximize the performance in software and minimize the critical path of round functions in hardware, we focus on a class of round functions with the following features.

- Applying only either aesenc or XOR to each block in one round.
- Applying a state-wise permutation before operations of aesenc or XOR.

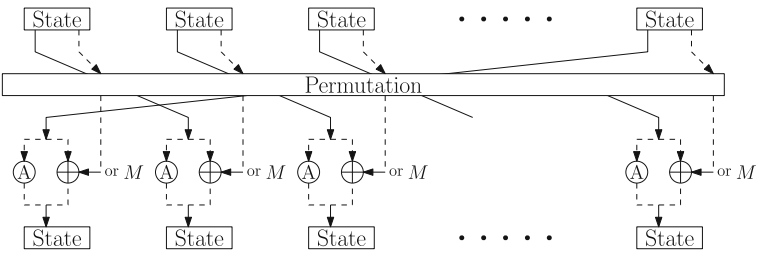

Fig. 3. General Design for the Round Function of Rocca-S.

In hardware, since a state-wise permutation does not cause any delay, the critical path of this round function is the execution of a single aesenc module. This delay is the lower bound for AES-based round functions, meaning that the delay of the round function cannot be reduced any further.

Design Challenge. The existing round functions of AEGIS [22], Tiaoxin-346 [17], Rocca [19,20] and all of Jean and Nikolić's ones [7] can be categorized into this type of class, however, these support only 128-bit tags, i.e., they ensure only 128-bit security against forgery attacks in the classic setting. We should also note here that the bound of forgery attack of Rocca is 2^{-144}, meaning the security

margin is only 16 bits [19,20]. Therefore, our main challenge in this paper is to support a 256-bit tag while maintaining high performance in both software and hardware. To be more specific, the number of active S-boxes for forgery attacks should be much more than that of Rocca [19,20] without sacrificing performance.

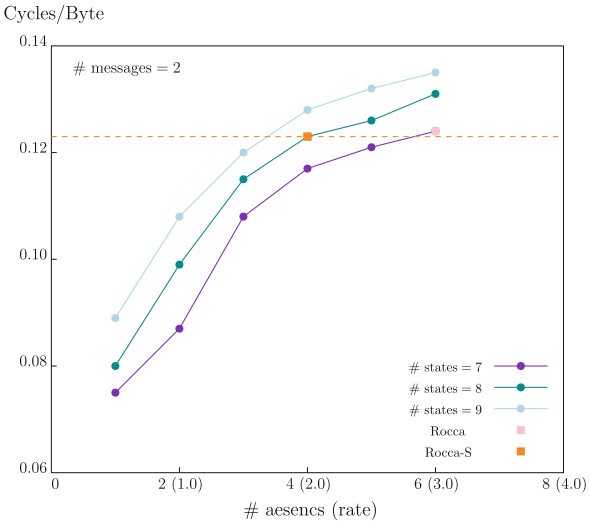

Fig. 4. Relationship between the number of AES-NI and performance.

Our Solution. Our core insights to accomplish this challenge is as follows: (1) smaller state is more efficient and (2) even if increasing #aesenc from 4 to 6, the overhead is marginal in software, as shown in Fig. 4. This allows us to add more nonlinear operations into each round, and then open up new design space of AES-based round functions while keeping the software's performance. Interestingly, the speed of Rocca parameters (#aesenc $= 4$ and #state size $= 8$) is almost same as those of Rocca-S (#aesenc $= 6$ and #state size $= 7$).

From a hardware point of view, as these are executed in parallel, there is no overhead regarding throughput. Beside, smaller *#state* leads to be smaller circuit scale and low energy consumption in hardware.

3.2 Requirements of Round Function for Performance and Security

This section clarifies the requirements for finding optimal parameters of AES-based round functions for our purpose.

Performance. The performance of AES-NI can be measured d by latency and throughput. Latency means the number of clock cycles that are required for the execution of an instruction. Throughput means the number of clock cycles required to wait before the responsible ports can accept the same instruction

again. Dispa These depend on the CPU architectures [18]. This paper focuses on the latest architectures after Intel Ice-lake series CPU where latency and throughput of aesenc are 3 and 0.5, respectively. Let $\#state$, #aesenc, and $\#message$ be the number of states, aesencs, and inserted message blocks in a single round, respectively.

Jean and Nikolić introduced *rate* to estimate the approximate speed of the round function [7], and smaller *rate* leads to a more efficient round function.

Definition 1 (Rate [7]) *Rate is the required number of* aesenc *calls to encrypt a 128-bit message, which is defined as* $rate = \#\mathsf{aesenc} \ / \ \#message$.

They also discussed the number of aesenc in each round to fully take advantage of parallel execution [7], which is expressed by the following equation:

$$\#\mathsf{aesenc} \geq \text{latency/throughput}.$$

If #aesenc is less than (latency)/(throughput), there exist empty cycles in a parallel process of aesenc. To fully take advantage of the parallel processing, #aesenc should be the same or more than (latency)/(throughput), and there are no empty cycles. In the case of our target architectures, it should be $\#\mathsf{aesenc} \geq \text{latency/throughput} = 3/0.5 = 6$. Note that since our output function utilizes two aesenc to be secure against linear attacks [15], as is discussed in Sect. 3.4, #aesenc in the round function should be 4 and more.

Security. We estimate the security against forgery by the lower bound for the number of differentially active S-boxes. Since the maximum differential probability of an S-box is 2^{-6}, we aim at finding round functions in which the lower bound for the number of active S-boxes in the forgery setting is at least 43 so that the differential probability is less than 2^{-256}.

Summary of Requirements. Taking these issues into consideration, we clarify requirements for the AES-based round function for our purposes as follows:

Req 1. *Rate* ($= \#\mathsf{aesenc}/\#message$) is as small as possible.
Req 2. #aesenc is 4 or more in each round function.
Req 3. $\#state$ is as small as possible.
Req 4. The lower bound for the number of active S-boxes is ≥ 43.

3.3 Finding Optimal Parameters of Round Functions

In this section, we search for optimal parameters that satisfy the requirements. Let s, a, and m be $\#state$, #aesenc, and $\#message$, respectively.

Search Method. Once we select *rate* and s according to Req 1 and 3, then we can properly choose pairs of a and m by Req 2. Specifically, we search for all 15 candidates with parameters such that $rate = 2$ to 3 and $s = 6, 7, 8$ as shown in Table 1. For each parameter, we try to search for candidates that satisfy

Req 4 for all patterns of block permutations and the combinations of positions of inserted messages and aesenc/XOR in the target class of Fig. 3 by MILP-aided evaluation. Sakamoto *et al.* estimated the total number of search space as $s! \times \binom{s}{a} \times \binom{s}{m}$ [19]. However, this search space includes equivalent class of round functions. Considering such equivalent classes, we can reduce it by the formula of $\frac{s!}{m!} \times \binom{s}{m} \times \binom{s}{a}$. For example, the candidates of the class of $s = 7$, $a = 4$, and $m = 2$ can be reduced from $2^{21.82}$ in Sakamoto *et al.* [19], to $2^{20.82}$. In our evaluation, if the total number of candidates in the class exceeds 2^{23}, we randomly choose 2^{20} candidates and evaluate these due to the limitations of the computational power.

Table 1. Candidate of round functions for each class.

#state	#aesenc	*#message*	*rate*	Total	# of searched	# of found
6	4	2	2.0	$2^{16.31}$	All	0
6	6	3	2.0	$2^{11.23}$	All	0
6	5	2	2.5	$2^{16.98}$	All	0
6	6	2	3.0	$2^{12.40}$	All	0
7	4	2	2.0	$2^{20.82}$	All	0
7	6	3	2.0	$2^{17.65}$	All	0
7	7	3	2.33	$2^{14.84}$	All	0
7	5	2	2.5	$2^{20.08}$	All	0
7	**6**	**2**	**3.0**	$\mathbf{2^{18.50}}$	**All**	**14**
8	4	2	2.0	$2^{25.24}$	2^{20}	0
8	6	3	2.0	$2^{23.33}$	2^{20}	0
8	7	3	2.33	$2^{21.52}$	All	0
8	5	2	2.5	$2^{24.91}$	2^{20}	0
8	8	3	2.67	$2^{18.52}$	All	0
8	**6**	**2**	**3.0**	$\mathbf{2^{23.91}}$	$\mathbf{2^{20}}$	**784**

Table 2. Lower bound of differentially active S-boxes, maximum rounds of the integral distinguisher, and speeds.

Target	# of active S-boxes					Integral distinguisher	Speed (cycles / Byte)
	6R	7R	8R	9R	10R		
AEGIS-128L	85	86	94	111	120	6R	0.188985
Tiaoxin-346	53	93	99	123	134	15R	0.200404
Rocca	54	62	82	85	93	7R	0.123258
RF-1 (Rocca-S)	**94**	**113**	**122**	**134**	**152**	**5R**	**0.122219**
RF-2	76	88	103	115	131	6R	0.129443
RF-3	96	101	114	129	136	6R	0.118518
RF-4	80	100	108	120	145	5R	0.122185
RF-5	81	86	95	121	141	5R	0.122286
RF-6	97	113	122	139	151	6R	0.129258
RF-7	97	110	128	132	137	6R	0.129523

Table 3. The lower bound of differentially active S-boxes and # rounds for full diffusion .

# of active S-boxes	# rounds for full diffusion	# of candidates
44	6	7
46	**5**	**7**

Results. Table 1 shows the summary of our search. We found 14 candidates in $s = 7$, $a = 6$, and $m = 2$ and 784 candidates in $s = 8$, $a = 6$, and $m = 2$ which satisfy Req 4. Due to Req 3, we choose 14 candidates of the class of $s = 7$, $a = 6$, and $m = 2$. This evaluation requires about 45 d on three computers equipped with AMD Ryzen Threadripper 3990X (64-Core) and 256 GB RAMs.

Selecting Best Round Function for Rocca-S. To determine one round function from 14 candidates of $s = 7$, $a = 6$, and $m = 2$, we evaluate the security and performance of these.

– Table 3 shows the required number of rounds for full diffusion and the lower bound for the number of active S-boxes for forgery setting. We choose seven candidates which attain 46 active S-boxes and achieve the full diffusion after 5 rounds named as RF-1, 2, 3, . . . , 7.

– Table 2 shows the security of the initialization phase of these candidates against differential attacks and integral attacks by a byte-based MILP, assuming that the adversary can control only nonce. In addition, Table 2 compares the speed of the round function of 7 candidates and Rocca, where the speed is measured as the average of the round function executed $2^{23.25}$ times with 64 kB messages on Intel(R) Core(TM) i7-1068NG7 CPU @ 2.30GHz.

Considering results of Table 2, we finally adopt RF-1 as shown in Fig. 1.

3.4 On the Loading Scheme and Output Functions

For the loading scheme of the nonce and key, we mainly want to avoid the case occurring in Tiaoxin-346. Specifically, we expect that after some number of rounds, the whole state words cannot be expressed only in terms of $A(N)$ and (K_0, K_1). If this happens, there will be a useless round and it opens a door for more powerful attacks [11]. By setting $S[5] = N \oplus K_1$, such a case is avoided.

To resist the linear attacks on output function, that has been successfully applied to AEGIS [15], we use the MILP model [2] to search for secure ones. For efficiency and security, we choose the output functions of the following form:

$$C_i^0 = \mathsf{AES}(S[j_0] \oplus S[j_1], S[j_2]) \oplus \overline{M}_i^0,$$
$$C_i^1 = \mathsf{AES}(S[j_3] \oplus S[j_4], S[j_5]) \oplus \overline{M}_i^1,$$

where $j_{u_1} \neq j_{u_2}$ for $u_1 \neq u_2$ and $0 \leq j_0, j_1, j_2, j_3, j_4, j_5 \leq 6$.

Then, with the truncated MILP model [2], for each choice of the tuple $(j_0, j_1, j_2, j_3, j_4, j_5)$, we can compute the lower bound of the number of active S-boxes for a exploitable linear trail that can be used for attacks. For our choice, the lower bound of the number of active S-boxes is 45. Hence, the time complexity of the linear attack will be higher than $2^{45\times 6} = 2^{270}$. Note that there is a big gap between the truncated model and the bit-wise model and the actual linear trail that can be used for attacks may be of much lower bias and the time complexity may be much higher than 2^{270}.

3.5 The Key Feed-Forward Operation

It has been observed by the designers of AEGIS-256 that the internal state can be fully recovered by using the decryption oracle for about 2^t times, where t is the tag size in bits. Specifically, after making 2^t calls to the decryption oracle, the attacker can expect to find another plaintext-ciphertext pair under the same (N, K). Then, a trivial state-recovery attack can be launched since only 1 round of AES is used to update the state at the keystream phase. In Rocca-S, $t = 256$ ensures that the time complexity of this attack is bounded by 2^{256}. Moreover, even if the state is recovered with some other methods, we expect that the key cannot be recovered. In AEGIS-256, this is ensured by using a keyed permutation for the initialization phase. In Rocca-S, we adopt the similar idea, also mentioned in [6]. Specifically, we simply use a key feed-forward operation to prevent the further key-recovery attack because the attackers cannot invert the initialization phase without knowing the key even if the state after this phase is fully known.

4 Security Evaluation

4.1 Differential/Linear Attack

In order to evaluate the security against differential and linear attacks, we compute the lower bound for the number of active S-boxes in the initialization phase by a MILP-aided method [16]. Since the maximal differential/linear probability of the S-box of AES is 2^{-6}, it is sufficient to guarantee the security against differential/linear attacks if there are 43 active S-boxes, as it gives $2^{(-6\times 43)} < 2^{-256}$ as an estimate of the differential/linear probability. Our evaluation shows that there are 68 active S-boxes over 5 rounds in the single-key setting and 53 active S-boxes over 8 rounds in the related-key setting in the initialization phase.

4.2 State-Recovery Attack

At the keystream phase, with the knowledge of plaintexts and ciphertexts, it is possible to recover the internal state with some guess-and-determine (GnD) strategies. To recover the whole internal state, we need to consider at least 4 consecutive rounds at the keystream phase. Specifically, we need to solve the following nonlinear equation system in terms of $S[i]$ $(0 \le i \le 6)$ where α_j $(0 \le j \le 7)$ are known values:

$$
\begin{aligned}
&\alpha_0 = A(S[3] \oplus S[5]) \oplus S[0],\ \alpha_1 = A(S[4] \oplus S[6]) \oplus S[2],\\
&\alpha_2 = A(\underline{A(S[2]) \oplus S[6]} \oplus \underline{A(S[4]) \oplus S[3]}) \oplus \underline{S[1] \oplus S[6]},\\
&\alpha_3 = A(\underline{A(S[3]) \oplus A(S[5]) \oplus S[4]}) \oplus \underline{A(S[1]) \oplus S[0]},\\
&\alpha_4 = A(\underline{A(A(S[1]) \oplus S[0]) \oplus A(S[5]) \oplus S[4]} \oplus \underline{A(A(S[3])) \oplus A(S[2]) \oplus S[6]})\\
&\quad \oplus \underline{A(S[0]) \oplus A(S[5]) \oplus S[4]},\\
&\alpha_5 = A(\underline{A(A(S[2]) \oplus S[6])} \oplus \underline{A(A(S[4]) \oplus S[2]) \oplus A(S[3]))}\\
&\quad \oplus \underline{A(A(S[0])) \oplus S[1] \oplus S[6]},\\
&\alpha_6 = A(\underline{A(A(A(S[0])) \oplus S[1] \oplus S[6]) \oplus A(A(S[4]) \oplus S[3]) \oplus A(S[3])})\\
&\quad \oplus \underline{A(A(A(S[2]) \oplus S[6])) \oplus A(A(S[1]) \oplus S[0]) \oplus A(S[5]) \oplus S[4])}\\
&\quad \oplus \underline{A(s[1] \oplus S[6]) \oplus A(A(S[4]) \oplus S[3]) \oplus A(S[3])},\\
&\alpha_7 = A(\underline{A(A(A(S[1]) \oplus S[0]) \oplus A(S[5]) \oplus S[4])}\\
&\quad \oplus \underline{A(A(A(S[3])) \oplus A(S[1]) \oplus S[0]) \oplus A(A(S[2]) \oplus S[6]))}\\
&\quad \oplus \underline{A(A(S[1] \oplus S[6])) \oplus A(S[0]) \oplus A(S[5]) \oplus S[4]}.
\end{aligned}
$$

It can be found that for (α_2, α_3) 2 rounds of AES, for (α_4, α_5), 3 rounds of AES and for (α_6, α_7), 4 rounds of AES are involved. Indeed, for the state-recovery attack on Rocca discussed in [19], the attacker also needs to consider 4 consecutive rounds and similar 8 equations in 8 variables. However, for all those 8 equations, at most 2 rounds of AES are involved and Rocca still has a strong resistance against this attack. This implies that recovering the state of Rocca-S becomes much more difficult. As 2 rounds of AES can achieve the full diffusion, it soon implies the GnD attack is not a threat and Rocca-S has a strong resistance against this type of state-recovery attack.

4.3 The Linear Bias of the Keystream

It has been discussed in Sect. 3 that the used output functions are chosen in such a way that it can resist the linear attack proposed in [15]. Specifically, by computing the lower bound of the active S-boxes, we expect that the time complexity of the linear attack [15] is higher than 2^{256}.

4.4 Forgery Attack

It has been shown in [17] that the forgery attack is a main threat to the constructions like Tiaoxin-346 and AEGIS as only one-round update is used to absorb each block of associated data and message. Such a concern has been taken into account in our design phase, as reported in Sect. 3. Specifically, in the forgery attack, the aim is to find a differential trail where the attackers can arbitrarily choose differences at the associated data and expect that such a choice of difference can lead to a collision in the internal state after several number of rounds. The resistance against this attack can be efficiently evaluated with an automatic method [16]. As Rocca-S is based on the AES round function, it suffices to prove that the number of active S-boxes in such a trail is larger than 43 as the length of the tag is 256 bits. With the MILP-based method, it is found that the lower bound is 46. However, these estimates do not take into consideration additional constant factors of improvements and optimizations, e.g. using clustering effect, which is why we reduce our security claims. Consequently, Rocca-S can provide 192-bit security against the forgery attack.

4.5 Security Against Quantum Attacks

A quantum adversary has the ability to leverage Grover's algorithm [5] to perform an exhaustive key search given a limited number of plaintext-ciphertext pairs. In the case of Rocca-S, this would require $2^{256/2} = 2^{128}$ iterations, with each iteration involving the evaluation of the quantum implementation of Rocca-S (similar to AES as described in [4]). According to [8], if there exists a classical distinguisher (such as a differential or linear distinguisher) with a probability of 2^{-p}, a quantum adversary can utilize this to mount a distinguishing attack with a time and data complexity of $2^{p/2}$.

However, as demonstrated in the previous sections, the probability p for the distinguishers (differential or linear) of Rocca-S exceeds 256. Therefore, a quantum distinguishing attack would require a time and data complexity of at least 2^{128}. Hence, Rocca-S claims to provide 128-bit quantum security against key recovery and forgery when the adversary is restricted to classical online queries only. It is important to note that Rocca-S *does not* claim security against quantum adversaries with access to online superposition queries.

5 Software Implementation

In this section, we evaluate the performance of Rocca-S and show that modifications only incur small overhead to the performance, despite the increase of

number of AES round functions in one round of state update. For the comparison to existing algorithms, we included Rocca-S as well as AEGIS, SNOW-V, and Tiaoxin to OpenSSL 3.1.0-dev and measured their performances with `speed` command. The implementation of SNOW-V is published in [3], and implementations of Tiaoxin-346 and AEGIS are publicly available.[1]. As shown in Table 4, Rocca-S exhibits the highest performance and achieves a throughput of 205 Gbps, which is the fastest in our comparison even compared to 128-bit key and tag algorithms.

Table 4. Software performance evaluation.

Algorithms	Key length	Tag length	Size of input (bytes)				
			16384	8192	1024	256	64
AEAD (Gbps)							
AEGIS-128	128-bit	128-bit	46.76	45.42	32.42	16.32	5.38
AEGIS-128L			151.53	137.49	60.37	20.68	5.40
Tiaoxin-346 v2			176.94	159.51	68.13	22.90	5.92
AEGIS-256	256-bit		47.96	46.50	33.29	16.72	5.52
AES-256-GCM			60.29	57.67	36.23	15.86	5.06
ChaCha20-Poly1305			22.40	21.71	15.25	6.15	2.15
SNOW-V-GCM			37.87	36.60	25.15	12.15	3.97
Rocca *			199.88	177.41	68.98	22.33	6.01
Rocca-S		256-bit	**205.68**	**183.22**	**74.33**	**24.78**	**6.65**

*: Updated version of Rocca [20], which is secure against the attack [6]

6 Hardware Implementation

The design of Rocca-S lends itself well to hardware implementations as, apart from the state registers and the AES modules of the round and encryption functions, little additional circuitry is required. In this section, we commence by investigating two separate round-based implementations of the Rocca-S specification and compare them to related AES-based AEAD constructions that also feature a key size of 256 bits. Our approach follows a similar structure to what was established in the work by Caforio *et al.* [1] for the SNOW-V stream cipher In particular, the authors investigated several micro-architectural directions to implement the AES round function components.

– *S-Box.* The substitution table can be synthesized in a straightforward fashion by providing the look-up table specification (LUT) to the circuit compiler and letting the tool choose the actual implementation in terms of logic gates. The Decode-Switch-Encode (DSE) architecture mitigates the power overhead of the S-box look-up table by encoding and decoding the inputs and outputs

[1] https://github.com/floodyberry/supercop.

to the look-up table in order to reduce the switching activity of each wire. The combinatorial optimization space of the S-box was explored in a work by Maximov and Ekdahl [13] in which the currently smallest description of the S-box in terms of logic gates was proposed S alongside a low-depth variant F and a trade-off alternative T between the former two.
- *MixColumns.* We can similarly distinguish several ways to implement the linear layer. The currently smallest circuit comprised of 92 two-input XOR gates is due to Maximov [12]. In a separate work, Li *et al.* [10] demonstrated a low-depth implementation consisting of 103 XOR gates. For the sake of conciseness we will limit ourselves to the low-depth circuit of [10].
- *T-Table.* The T-table approach is particularly efficient in software implementations but can also be emulated in hardware similarly to the approach of synthesizing the S-box look-up table mentioned beforehand. Henceforth, the T-table configuration will be denoted by the abbreviation TT.

Simulation Environment. All presented designs were synthesized using the Synopsys Design Compiler (version 2017.09) using two standard cell libraries, i.e., NanGate 15 nm process and the more industry-grade TSMC 28 nm process. The power and energy consumption was then extracted in post-synthesis using the Synopsys Power Compiler via back-annotation.

6.1 Circuits

A round-based implementation of Rocca-S computes one invocation of the round update function R in one clock cycle, hence sixteen cycles are required to execute both the initialization and finalization routines and, in the same vein, the circuit absorbs 256-bit data blocks and outputs 256-bit ciphertext blocks per clock cycle. The approach we follow for the round-based implementation is relatively elementary and can be deduced from the original schematic in Fig. 1. Six AES modules, whose plaintext inputs are directly fed from the state registers, are placed in parallel. Their computed outputs are wired back to the corresponding register inputs thus taking care of the permutation without additional circuitry.

Unrolled Round Function. The round update function of Rocca-S can easily be replicated and chained together in order to compute multiple invocation in a single clock cycle. Although the area increase quickly reaches prohibitive regions, the length of the critical path usually rises at slower pace thus yielding designs that admit the highest throughput.

6.2 Synthesis Results

Our round-based Rocca-S hardware implementations are compared against other AEAD schemes with a key size of 256 bits, namely AEGIS-256, AES-256-GCM and SNOW-V-GCM [3,14,22]. Note that actual published ASIC implementations of said algorithms are hard to come by, hence we chose to devise them for this comparison section. AEGIS-256 is reminiscent of Rocca-S in design and thus can

be adapted accordingly, on the other hand, AES-256-GCM and SNOW-V-GCM require a Galois field multiplication module over 128 bits for which we opted for a straightforward Karatsuba architecture which is then attached to a AES-256 module extracted and extended from the Rocca-S round function and a SNOW-V stream cipher core whose implementation is available in [1].

Circuit Area. The lion's share of gate area in Rocca-S is due to the eight AES round function cores that compose its round function and ciphertext generation function. This induces a significant overhead in comparison to the other schemes. AEGIS-256 requires only six cores whereas AES-256-GCM and SNOW-V-GCM are equipped with only and two core respectively. The Galois field multiplication module, a notoriously difficult function to map to hardware, found in the latter two has an area footprint of roughly 30000 GE across cell libraries and thus constitutes a sizeable percentage of their overall area. Across all implementation choices the area of our round-based Rocca-S circuit remains competitive.

Throughput. The premise of Rocca-S is a high-speed construction that improves on other known schemes in terms of throughput i.e., how many bits per second can be processed. In hardware, this figure is inextricably tied to the length of the critical path which specifies the maximum clock frequency at which a design can be run. In both Rocca-S and AEGIS-256 the critical path is due to the AES modules, thus it is highly variable regarding the choice of round function implementation, whereas in AES-256-GCM and SNOW-V-GCM it is imposed by the field multiplication thus constant across implementation choices. This means that for both AES-256-GCM and SNOW-V-GCM unrolling the round function exerts only marginal effects on the overall throughput. Excluding the initialization and finalization phases, Rocca-S processes 256 bits of data with each clock cycle. Similarly AEGIS-256 and SNOW-V-GCM are able to process one 128-bit data block in one clock cycle whereas AES-256-GCM requires a full AES-256 encryption for each 128-bit plaintext block hence asymptotically for large plaintexts AES-256-GCM only processes roughly 8 bits per clock cycle. Consequently, the ability to accept larger data blocks paired with a competitive critical path allows Rocca-S to reach a throughput well beyond 1 Terabit per second for the NanGate 15 nm cell library regardless of the choice of round function implementation, outperforming the other schemes by at least 50%. Also, a throughput rate beyond 2 Terabits is reached for 2-round unrolled circuits, marking Rocca-S as the first cryptographic algorithm that crosses this barrier as shown in Table 5.

Table 5. Maximum throughput (Tbps) comparison using a TT AES module for round-based and 2-round unrolled circuits.

	Rocca-S	AEGIS-256	AES-256-GCM	SNOW-V-GCM
NanGate 15 nm	**1.653/2.019**	0.970/1.028	0.023/0.024	0.365/0.442
TSMC 28 nm	**0.373/0.409**	0.188/0.190	0.007/0.007	0.088/0.106

Power/Energy Consumption. Our power/energy experiments were conducted on two workloads. A short workload describes the processing of 1024 bits of associated data and 2048 bits of plaintext whereas a long workload consists of 1024 bits of associated data and 1.28 Megabits of plaintext. Again, the round-based Rocca-S circuit stands as competitive choice regarding its power and energy consumption. A list of all obtained power/energy measurements is given in Table 7.

7 Conclusion

In this paper, we proposed the AES-based authenticated encryption scheme Rocca-S with a 256-bit key and a 256-bit tag. Rocca-S achieves a speed of more than 200 Gbps in software. In hardware implementation, Rocca-S is the first cryptographic algorithm to achieve speeds consistently between 1 and 2 Terabits per second without sacrificing other metrics.

Acknowledgments. Takanori Isobe is supported by JST, PRESTO Grant Number JPMJPR2031. This research was in part conducted under a contract of "Research and development on new generation cryptography for secure wireless communication services" among "Research and Development for Expansion of Radio Wave Resources (JPJ000254)", which was supported by the Ministry of Internal Affairs and Communications, Japan. We thank Akinori Hosoyamada, Akiko Inoue, Ryoma Ito, Tetsu Iwata, Kazuhiko Mimematsu, Ferdinand Sibleyras, Yosuke Todo, Patrick Derbez, Pierre-Alain Fouque, André Schrottenloher, Santanu Sarkar, Satyam Kumar, Chandan Dey and anonymous reviewers for their valuable comments.

Appendix

See Fig. 5.

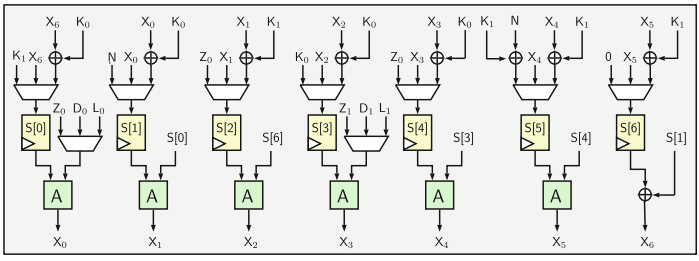

Fig. 5. Rocca-S round function circuit.

Table 6. Circuit area (GE) and Throughput (Critical Path and Max TP) comparison of the investigated AEAD scheme for two cell libraries and several round function implementations. Note that the T-table approach of implementing the round function offers the overall best choice for both Rocca-S and AEGIS-256. This phenomenon was already observed in [1].

(a) Rocca-S

	LUT	DSE	S	F	T	TT
Round-Based						
NanGate 15 nm						
GE	116130	116638	56889	61918	56728	145364
Critical Path (ns)	0.179	0.177	0.232	0.208	0.207	0.154
Max TP (Tbps)	1.431	1.451	1.102	1.229	1.234	1.653
TSMC 28 nm						
GE	100689	110273	58688	63681	57452	134220
Critical Path (ns)	0.95	0.81	1.00	0.83	0.93	0.71
Max TP (Tbps)	0.269	0.316	0.256	0.308	0.275	0.373
2-Round Unrolled						
NanGate 15 nm						
GE	218629	220428	100814	110802	99728	277691
Critical Path (ns)	0.321	0.322	0.412	0.383	0.388	0.253
Max TP (Tbps)	1.590	1.589	1.241	1.333	1.319	2.019
TSMC 28 nm						
GE	188199	208045	104876	114861	102402	255938
Critical Path (ns)	1.78	1.48	1.84	1.53	1.73	1.25
Max TP (Tbps)	0.287	0.348	0.279	0.334	0.294	0.409

(b) AEGIS-256

	LUT	DSE	S	F	T	TT
Round-Based						
NanGate 15 nm						
GE	88521	89116	44266	48009	43854	110591
Critical Path (ns)	0.167	0.165	0.21	0.196	0.198	0.132
Max TP (Tbps)	0.766	0.776	0.610	0.653	0.646	0.970
TSMC 28 nm						
GE	75608	83081	44392	48137	43465	101043
Critical Path (ns)	0.88	0.73	0.92	0.76	0.86	0.68
Max TP (Tbps)	0.145	0.175	0.139	0.168	0.149	0.188
2-Round Unrolled						
NanGate 15 nm						
GE	167333	168650	78939	86426	78125	211599
Critical Path (ns)	0.321	0.318	0.409	0.381	0.384	0.249
Max TP (Tbps)	0.798	0.805	0.626	0.672	0.667	1.028
TSMC 28 nm						
GE	143435	158156	80779	88269	78925	194077
Critical Path (ns)	1.72	1.46	1.79	1.49	1.69	1.35
Max TP (Tbps)	0.149	0.175	0.143	0.172	0.151	0.190

(c) AES-256-GCM

	LUT	DSE	S	F	T	TT
Round-Based						
NanGate 15 nm						
GE	50980	51381	42038	42816	41951	64082
Critical Path (ns)	0.349	0.349	0.349	0.349	0.349	0.349
Max TP (Tbps)	0.023	0.023	0.023	0.023	0.023	0.023
TSMC 28 nm						
GE	50908	52447	44384	45166	44192	62450
Critical Path (ns)	1.23	1.23	1.23	1.23	1.23	1.23
Max TP (Tbps)	0.007	0.007	0.007	0.007	0.007	0.007
2-Round Unrolled						
NanGate 15 nm						
GE	98058	98307	79615	81177	79447	123739
Critical Path (ns)	0.673	0.673	0.674	0.674	0.674	0.674
Max TP (Tbps)	0.024	0.024	0.024	0.024	0.024	0.024
TSMC 28 nm						
GE	98398	101517	85394	86957	85010	121520
Critical Path (ns)	2.14	2.15	2.15	2.13	2.13	2.16
Max TP (Tbps)	0.007	0.007	0.007	0.007	0.007	0.007

(d) SNOW-V-GCM

	LUT	DSE	S	F	T	TT
Round-Based						
NanGate 15 nm						
GE	75729	75912	60964	62205	60827	83074
Critical Path (ns)	0.351	0.351	0.351	0.351	0.351	0.351
Max TP (Tbps)	0.365	0.365	0.365	0.365	0.365	0.365
TSMC 28 nm						
GE	71396	73860	60966	62214	60656	79846
Critical Path (ns)	1.45	1.45	1.45	1.45	1.45	1.45
Max TP (Tbps)	0.088	0.088	0.088	0.088	0.088	0.088
2-Round Unrolled						
NanGate 15 nm						
GE	136139	136622	106715	109212	106445	150935
Critical Path (ns)	0.579	0.577	0.579	0.577	0.579	0.579
Max TP (Tbps)	0.442	0.444	0.442	0.444	0.442	0.442
TSMC 28 nm						
GE	131342	136269	110477	112974	109857	148241
Critical Path (ns)	2.41	2.41	2.41	2.41	2.41	2.41
Max TP (Tbps)	0.106	0.106	0.106	0.106	0.106	0.106

Table 7. Power/energy consumption comparison of the investigated AEAD scheme for two cell libraries and several round function implementations. All figures were obtained by clocking the designs at constant frequency of 10 MHz.

(a) Rocca-S

	LUT	DSE	S	F	T	TT
Round-Based						
Lat. Short (Cycles)	44	44	44	44	44	44
Lat. Long (Cycles)	5036	5036	5036	5036	5036	5036
NanGate 15 nm						
Power (mW)	1.401	0.765	1.255	1.314	1.134	0.881
Energy Short (nJ)	6.165	3.368	5.522	5.780	4.985	3.876
Energy Long (nJ)	705.6	385.5	632.1	661.6	570.5	443.7
TSMC 28 nm						
Power (mW)	0.830	0.389	0.754	0.709	0.690	0.373
Energy Short (nJ)	3.650	1.709	3.316	3.119	3.034	1.642
Energy Long (nJ)	417.8	195.6	379.5	357.0	347.2	187.9
2-Round Unrolled						
Lat. Short (Cycles)	22	22	22	22	22	22
Lat. Long (Cycles)	2518	2518	2518	2518	2518	2518
NanGate 15 nm						
Power (mW)	6.254	2.140	5.631	5.824	5.248	2.202
Energy Short (nJ)	13.76	4.70	12.39	12.81	11.55	4.85
Energy Long (nJ)	1574.1	538.8	1417.8	1466.5	1321.5	554.6
TSMC 28 nm						
Power (mW)	3.963	1.345	3.293	2.985	2.922	1.074
Energy Short (nJ)	8.720	2.959	7.244	6.568	6.429	2.364
Energy Long (nJ)	998.1	338.7	829.1	751.7	735.9	270.5

(b) AEGIS-256

	LUT	DSE	S	F	T	TT
Round-Based						
Lat. Short (Cycles)	48	48	48	48	48	48
Lat. Long (Cycles)	10032	10032	10032	10032	10032	10032
NanGate 15 nm						
Power (mW)	1.106	0.613	1.014	1.081	0.954	0.691
Energy Short (nJ)	5.309	2.945	4.868	5.184	4.579	3.317
Energy Long (nJ)	1109	615.6	1017	1083	956.9	693.3
TSMC 28 nm						
Power (mW)	0.619	0.297	0.564	0.531	0.518	0.288
Energy Short (nJ)	2.975	1.426	2.707	2.549	2.488	1.380
Energy Long (nJ)	621.8	298.1	565.8	532.8	520.1	288.4
2-Round Unrolled						
Lat. Short (Cycles)	24	24	24	24	24	24
Lat. Long (Cycles)	5016	5016	5016	5016	5016	5016
NanGate 15 nm						
Power (mW)	4.147	1.674	3.779	3.887	3.539	1.670
Energy Short (nJ)	9.953	4.018	9.069	9.328	8.494	4.008
Energy Long (nJ)	2080.1	839.7	1895.3	1949.5	1775.3	837.8
TSMC 28 nm						
Power (mW)	2.483	0.918	2.115	1.925	1.901	0.915
Energy Short (nJ)	5.958	2.202	5.077	4.621	4.562	2.196
Energy Long (nJ)	1245.3	460.22	1061.0	965.78	953.49	459.06

(c) AES-256-GCM

	LUT	DSE	S	F	T	TT
Round-Based						
Lat. Short (Cycles)	266	266	266	266	266	266
Lat. Long (Cycles)	160010	160010	160010	160010	160010	160010
NanGate 15 nm						
Power (mW)	0.521	0.417	0.502	0.515	0.490	0.577
Energy Short (nJ)	13.85	11.09	13.36	13.69	13.04	15.33
Energy Long (nJ)	8328	6674	8035	8235	7843	9224
TSMC 28 nm						
Power (mW)	0.326	0.249	0.312	0.303	0.304	0.329
Energy Short (nJ)	8.661	6.645	8.291	8.070	8.076	8.759
Energy Long (nJ)	5209	3997	4987	4854	4857	5269
2-Round Unrolled						
Lat. Short (Cycles)	133	133	133	133	133	133
Lat. Long (Cycles)	80005	80005	80005	80005	80005	80005
NanGate 15 nm						
Power (mW)	1.356	1.145	1.313	1.339	1.287	1.461
Energy Short (nJ)	18.03	15.23	17.46	17.80	17.11	19.43
Energy Long (nJ)	10845	9159.8	10505	10710	10293	11689
TSMC 28 nm						
Power (mW)	0.825	0.671	0.795	0.778	0.777	0.831
Energy Short (nJ)	10.97	8.930	10.58	10.35	10.34	11.05
Energy Long (nJ)	6599.6	5369.9	6362.8	6226.8	6219.6	6645.2

(d) SNOW-V-GCM

	LUT	DSE	S	F	T	TT
Round-Based						
Lat. Short (Cycles)	42	42	42	42	42	42
Lat. Long (Cycles)	10026	10026	10026	10026	10026	10026
NanGate 15 nm						
Power (mW)	0.726	0.602	0.689	0.704	0.676	0.634
Energy Short (nJ)	3.047	2.528	2.895	2.958	2.840	2.662
Energy Long (nJ)	727.4	603.5	691.2	706.0	678.1	635.3
TSMC 28 nm						
Power (mW)	0.408	0.333	0.396	0.389	0.386	0.334
Energy Short (nJ)	1.714	1.399	1.663	1.634	1.619	1.402
Energy Long (nJ)	409.2	333.9	397.1	390.2	386.5	334.8
2-Round Unrolled						
Lat. Short (Cycles)	21	21	21	21	21	21
Lat. Long (Cycles)	5013	5013	5013	5013	5013	5013
NanGate 15 nm						
Power (mW)	1.947	1.520	1.862	1.902	1.820	1.551
Energy Short (nJ)	4.089	3.192	3.910	3.994	3.822	3.257
Energy Long (nJ)	976.13	761.73	933.57	953.62	912.57	777.62
TSMC 28 nm						
Power (mW)	1.090	0.827	1.029	0.996	0.992	0.805
Energy Short (nJ)	2.289	1.736	2.161	2.091	2.082	1.691
Energy Long (nJ)	546.32	414.32	515.89	499.24	497.04	403.65

References

1. Caforio, A., Balli, F., Banik, S.: Melting SNOW-V: improved lightweight architectures. J. Cryptogr. Eng. **12**(1), 53–73 (2022)
2. Eichlseder, M., Nageler, M., Primas, R.: Analyzing the linear keystream biases in AEGIS. IACR Trans. Symmetric Cryptol. **2019**(4), 348–368 (2019)
3. Ekdahl, P., Johansson, T., Maximov, A., Yang, J.: A new SNOW stream cipher called SNOW-V. IACR Trans. Symmetric Cryptol. **2019**(3), 1–42 (2019)
4. Grassl, M., Langenberg, B., Roetteler, M., Steinwandt, R.: Applying Grover's algorithm to AES: quantum resource estimates. In: Takagi, T. (ed.) PQCrypto 2016.

LNCS, vol. 9606, pp. 29–43. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-29360-8_3
5. Grover, L.K.: A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219 (1996)
6. Hosoyamada, A., et al.: Cryptanalysis of Rocca and feasibility of its security claim. IACR Trans. Symmetric Cryptol. **2022**(3), 123–151 (2022)
7. Jean, J., Nikolić, I.: Efficient design strategies based on the AES round function. In: Peyrin, T. (ed.) FSE 2016. LNCS, vol. 9783, pp. 334–353. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-52993-5_17
8. Kaplan, M., Leurent, G., Leverrier, A., Naya-Plasencia, M.: Quantum differential and linear cryptanalysis. IACR Trans. Symmetric Cryptol. **2016**(1), 71–94 (2016)
9. Latva-aho, M., Leppänen, K.: Key drivers and research challenges for 6G ubiquitous wireless intelligence (2019)
10. Li, S., Sun, S., Li, C., Wei, Z., Lei, H.: Constructing low-latency involutory MDS matrices with lightweight circuits. IACR Trans. Symm. Cryptol. **2019**(1), 84–117 (2019)
11. Liu, F., Isobe, T., Meier, W., Sakamoto, K.: Weak keys in reduced aegis and Tiaoxin. Cryptology ePrint Archive, Report 2021/187 (2021). https://eprint.iacr.org/2021/187
12. Maximov, A.: AES MixColumn with 92 XOR gates. Cryptology ePrint Archive, Report 2019/833 (2019). https://eprint.iacr.org/2019/833
13. Maximov, A., Ekdahl, P.: New circuit minimization techniques for smaller and faster AES SBoxes. IACR TCHES **2019**(4), 91–125 (2019). https://tches.iacr.org/index.php/TCHES/article/view/8346
14. David, A.: McGrew and John Viega. The security and performance of the Galois/-counter mode (GCM) of operation. In: Canteaut, A., Viswanathan, K. (eds.) INDOCRYPT 2004. LNCS, vol. 3348, pp. 343–355. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30556-9_27
15. Minaud, B.: Linear biases in AEGIS keystream. In: Joux, A., Youssef, A.M., editors, Selected Areas in Cryptography - SAC 2014–21st International Conference, Montreal, QC, Canada, August 14–15, 2014, Revised Selected Papers, volume 8781 of Lecture Notes in Computer Science, pp. 290–305. Springer (2014)
16. Mouha, N., Wang, Q., Gu, D., Preneel, B.: Differential and linear cryptanalysis using mixed-integer linear programming. In: Wu, C.-K., Yung, M., Lin, D. (eds.) Inscrypt 2011. LNCS, vol. 7537, pp. 57–76. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-34704-7_5
17. Nikolić, I.: Tiaoxin-346: version 2.0. CAESAR Competition (2014)
18. Real-Time and Embedded Sys Lab. uops.info. Official webpage. https://www.uops.info/
19. Sakamoto, K., Liu, F., Nakano, Y., Kiyomoto, S., Isobe, T.: Rocca: an efficient AES-based encryption scheme for beyond 5G. IACR Trans. Symmetric Cryptol. **2021**(2), 1–30 (2021)
20. Sakamoto, K., Liu, F., Nakano, Y., Kiyomoto, S., Isobe, T.: Rocca: an efficient AES-based encryption scheme for beyond 5G (full version). IACR Cryptol. ePrint Arch., 116 (2022)

21. The ZUC design team. The ZUC-256 Stream Cipher. http://www.is.cas.cn/ztzl2016/zouchongzhi/201801/W020180126529970733243.pdf (2018)
22. Wu, Hongjun, Preneel, Bart: AEGIS: a fast authenticated encryption algorithm. In: Lange, Tanja, Lauter, Kristin, Lisoněk, Petr (eds.) SAC 2013. LNCS, vol. 8282, pp. 185–201. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-43414-7_10