A REVIEW ON GRAIN-128AEADv2

Ece Beren GENC

Department of Cyber Security, Informatics Institute, Middle East Technical University, Ankara, Turkey
Department of Information Systems, Informatics Institute, Middle East Technical University, Ankara, Turkey
ece.genc@metu.edu.tr

1. Introduction

With the fourth industrial revolution, smart and connected devices start a new era for both our lives and the competition in the market. Internet of Things (IoT) connects devices. Inevitable and rapidly developing technology changes the rules of the game. RFID (radio frequency identification) tags, constrained devices, embedded sensory devices, cyber-physical systems and the idea of driverless future bring the new term: "connectivity". Smart devices communicate with each other wirelessly at large scale. This is a crucial part of today's systems. However, it brings security threats as well. IoT devices improve productivity, efficiency, and scalability. However, in case of possible data leakage, know-hows and even trade secrets can be leaked. In addition to them, these devices provide real-time data collection and analysis. Therefore, real-time responses can be given to the problems, because any security problems can affect analyses results and mislead people. Therefore, cyber security gains importance day by day. According to the "Information-Technology Promotion Agency of Japan" (IPA), "Exteriorization of vulnerability of IoT devices" is in the "The 10 Major Security Threats of 2017" (Toshihiko, 2017; Youssef et al., 2020). Existing cryptography standards do not address the need of these devices. They have limitations in terms of memory, battery, and computational capability. Therefore, cryptography system design requires the trade-off between performance, cost, and security for expected security level (Ragab et al., 2019; Bhattacharya et al., 2020; Youssef et al., 2020). One of the main problems about cryptography algorithms is that they do not meet the needs of the real-time data and are not appropriate for the 8, 16, 32-bit microcontrollers which are seen commonly (Buchanan, Li and Arif, 2017). Lightweight cryptography can be a solution to this problem.

2. Lightweight Cryptography

Lightweight cryptography is a developing area. Designing a cryptography algorithm for such devices should meet different performance and cost metrics. Cost metrics can be size (circuit size, ROM/RAM sizes), energy consumption and memory. On the other hand, performance

metrics can be listed as power consumption, latency, and throughput. IoT devices have low battery capacity. Therefore, implementing existing cryptography algorithms may need extra memory. This can be the limitation for battery-driven devices. Latency is also crucial for these systems. Communication and information exchange should be performed in a short time and delays can cause serious problems. One of the best examples is autonomous cars. They communicate with each other and their environment. Therefore, delay can cause fatal results. Throughput is not the most important performance metric. However, a still moderate level of throughput is expected. Lastly, cryptographic designs should be hardened against the side channel and fault attacks (Civek, 2021). For low-resource applications, software and hardware applications have different metrics. There when designing algorithms for software implementation memory, code size and throughput should be taken into consideration. On the other hand, for hardware implementation area, throughput, latency, and power consumption are important metrics (Youssef et al., 2020). Constrained devices such as RFID tags or handheld devices have limited energy capacity because of the tight battery, lightweight cryptography algorithms should meet the need of energy optimization. It is related to energy consumption which can be described as the electrical energy which is supplied by the battery to operate per unit. Therefore, it is expected to reduce energy usage per encryption. For encryption/decryption of the longer data, stream ciphers require less energy than block ciphers. It means that in terms of energy minimization, stream ciphers should have better performance than the block ciphers on long streams of data. However, in case of encryption of the short stream of data, block ciphers are a better solution. Stream ciphers with simple update function use the advantage of unrolling easily even for higher degrees. Therefore, in one clock cycle, an algorithm can encrypt more bits while a simple update function. Number of initialization rounds is another parameter which influences energy consumption (Banik et al., 2018). Lightweight cryptography algorithms can be classified as lightweight block ciphers, lightweight stream ciphers, lightweight hash functions and elliptic curve cryptography. Each of them has different performance metrics, strengths, and weaknesses (Biswas et al., 2016; Dhanda et al., 2020). Block ciphers focus on key and block size, rounds and key schedules. On the other hand, stream ciphers concentrate on chip area, key length, inner state and key/IV set up cycles (Dhanda et al., 2020). Lightweight cryptography community has been developing. Therefore, day by day algorithms have better performance in terms of inner state minimization, reduction of power and energy consumption, latency, chip area. Therefore, there are still some points that need to be improved (Dhanda et al., 2020).

“International Organization of Standardization” (ISO), “International Electrotechnical Commission” (IEC), “Cryptography Research and Evaluation Committees” (Cryptrec), “European Network of Excellence in Cryptology” (Ecrypt) and “Competition for Authenticated Encryption: Security, Applicability, and Robustness” (CAESAR) are introduced standards or the proposals for the lightweight cryptography. In August 2018 the U.S. National Institute of Standards and Technology (NIST) announced a Lightweight Cryptography Standardization Process and accepted submissions till February 2019. They are evaluating submitted authenticated encryption with associated data (AEAD) and hash algorithms to set a standard (Rezvani and Diehl, 2019). Grain-128 AEADv2 is one the finalists of this competition.

3. History of the Grain-128AEADv2

Grain-128AEADv2 is the member of the Grain family of the stream ciphers. This algorithm is based on Grain-128AEAD and the latest version of Grain family. Grain-128AEADv2 and Grain-128AEAD are very similar but the main difference is modifying initialization by additional tweak. Grain was first proposed in 2005 for the eSTREAM project with 80-bit key. Grain-128 with 128-bit key was proposed in 2006 and it was broken in 2011. In the same year, Grain 128-a with optional authentication was proposed. Grain AEAD was designed with mandatory authentication and support for associated data (Hell et al., 2019). European Network of Excellence in Cryptology (ECRYPT) held a project called eSTREAM to encourage the design of new stream ciphers. 34 stream ciphers were submitted and only 8 of them reached the final round and 7 of them found a place for presentation. Grain-v1 is one of the competitors in hardware profile. Grain-128AEADv2 was designed according to the feedback of the investigations. This algorithm is shaped accordingly Grain-128a which has been investigated since 2011. However, Grain-128a took insights of Grainv1 and Grain-128 which are both detailly analyzed. Grain128AEADv2 has two feedback shift registers which are linear feedback shift register (LFSR) and non-linear feedback shift register (NFSR). Both LFSR and NFSR consist of 128 bits. With Grain-128a, this family of ciphers are incorporated with message authentication code (MAC) (Hridya and Jose, 2019; Hell et al., 2019). In 2015, ISO/IEC 29167-13:2015 standard specified Grain-128a for RFID devices. Taking what is mentioned above into consideration, it is obvious that Grain is a mature algorithm, and it uses all feedback as a design strategy of Grain-128AEADv2. Designers’ main concern was that

keeping changes as small as possible while strengthening it and making it more resistant to attacks (Hell et al., 2019).

4. Algorithm Specification

Grain-128AEADv2 is implemented with a 128-bit linear shift register (LFSR) and 128-bit non-linear shift register (NFSR). Algorithm consists of a pre-output generator and authenticator generator. 128-bit LFSR, 128-bit NFSR and the pre-output function are classified under pre-output generators. On the other hand, a 64-bit shift register which keeps the last 64 odd bits, and a 64-bit accumulator creates the authenticator generator. Encryption and the authentication tag use pseudo-random bits which are generated by a pre-output generator. Pre-output function denoted by h(x) is a nine state Boolean function. Two of the nine bits come from non-linear feedback shift register and others come from linear feedback shift register (Hell et al., 2019).

$B_t = (b^{t0},...,b^t127)$ denotes NFSR state at $t \geq 0$,

$S_t = (s^{t0},...,s^t127)$ denotes LFSR state at $t \geq 0$,

$A_t = (a^{t0},...,a^{t63})$ denotes Accumulator bits at $t \geq 384$,

$R_t = (r^{t0},...,r^{t63})$ denotes Register bits at $t \geq 384$.

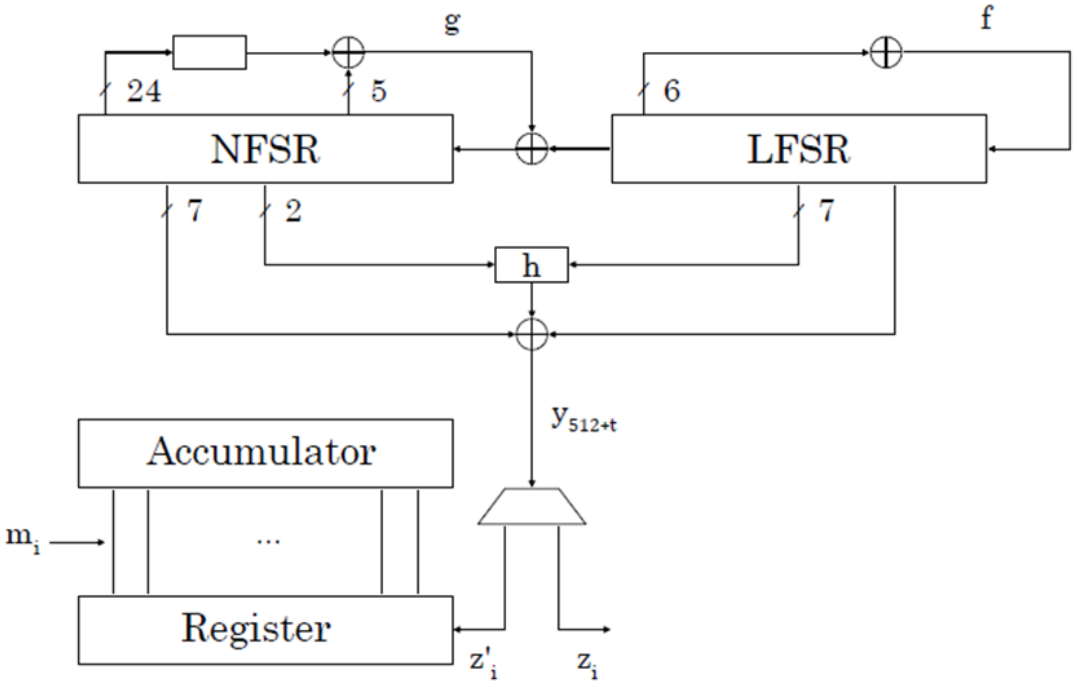

Figure 1: A general view of pre-output generator and authenticator generator in Grain-128AEADv2 (Hell et al., 2019).

Before the keystream is generated, GRAIN−128 is initialized with a 96-bit nonce (IV) and 128-bit key. 128-bit key is loaded to 128 bits of NFSR. 96-bit nonce is loaded into LFSR. Remaining 32 bits of LFSR is filled with 31 ones and a zero. Without generating any keystream, the algorithm is clocked 320 times and the output is fed back to the pre-output function. XORed with the input to both linear feedback shift register and non- linear feedback shift register. And then, the cipher is clocked 64 times. Key is reintroduced and XORed with the input to both linear feedback shift register and non- linear feedback shift register. After the initialization of the pre-output generator, the register and the accumulator including pre-output keystream is loaded to the authenticator generator to initialize it. After the initialization of the register and the accumulator, LFSR and NFSR are updated (Hell et al., 2019).

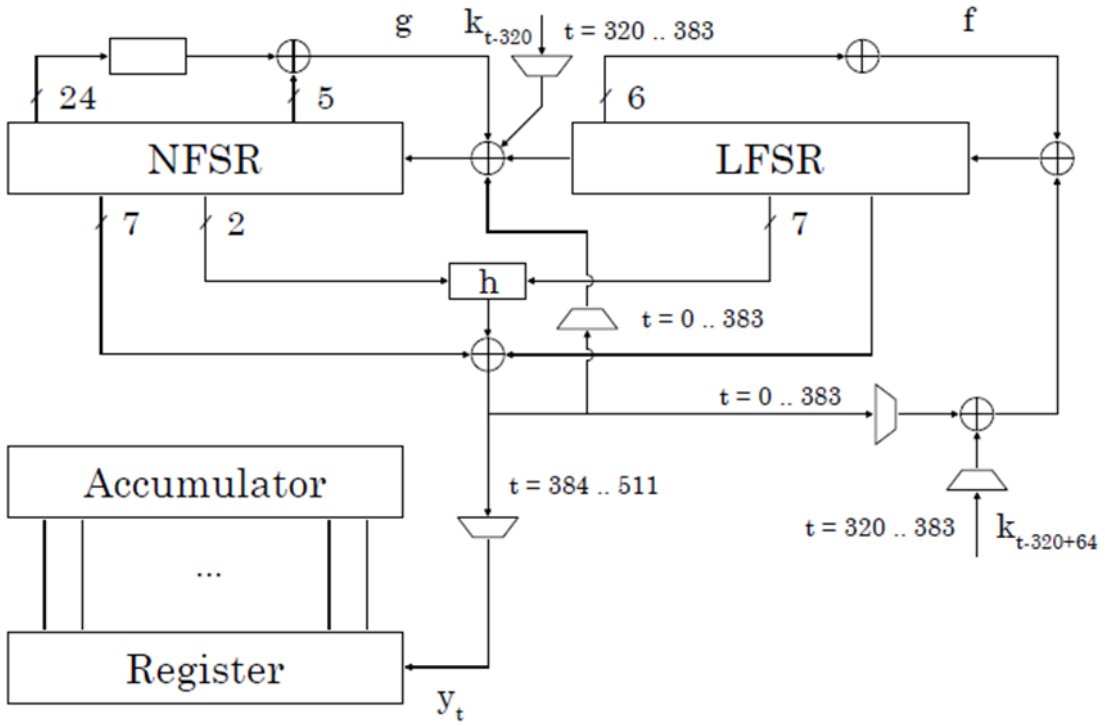

Figure 2: A general view of initialization (Hell et al., 2019).

After the pre-output generator is initialized, it generates keystreams for encryption and authentication which are used to update the register. Generated each even bit is kept as a

keystream bit and each odd bit is kept as an authentication bit. After encryption is applied on message, the accumulator and the shift register are updated (Hell et al., 2019).

Grain-128AEADv2 provides bit-level control. It allows authentication without encryption. Because of the keystream limitation, only 2^{81} pre-output bits can be encrypted with the same nonce and key pair (Hell et al., 2019).

5. Security Analysis and Cryptanalytic Attacks

In 2005, the first Grain family of stream ciphers was introduced and since that date, security of the algorithm has been investigated by researchers and the cryptography community. According to the research papers and results of different attacks, designers of the algorithm have made some changes. However, their main concern is keeping changes as small as possible while making algorithms more robust and secure (Hell et al., 2019).

5.1. Fault Attacks

Fault attack is a variant of side channel attacks. Berzati et al. (2009) focused on LFSR and introduced an attack on Grain-128. Also, Banik et al. (2012) and Sarkar et al. (2015) worked on differential fault attack against Grain family of stream ciphers. Karmakar and Chowdhury (2014) introduced a fault attack which targets NFSR. This approach was different from previous fault attacks because other attacks focused on inducing faults in the LFSR register. New proposed attack was applied to three members of the Grain family which are Grain v1, Grain-128a and Grain-128. Researchers' main concern was to understand whether protecting just LFSR is enough for security or NFSR needs protection. This attack required a small number of faults which is equal to approximately 1,25 times of the state size. For Grainv1 150 faults, for Grain-128 312 faults and for Grain-128a 384 faults were required to be injected. Number of faults, required time and memory are less than the previous fault attacks. Authors targeted both LFSR and NFSR and used SAT solver. They proved that to break the cryptosystem ten or less faults are enough. In practice this attack proposal has limitations, because during the attack it was assumed that the cipher was rekeyed. Also using SAT brought overhead (Karmakar and Chowdhury,2014). Earlier publications on differential fault attack (DFA) against the Grain family of stream ciphers needed hundreds of faults to be injected and they made assumptions on timing and the location of the fault. After this publication, Sarkar et al. (2015) attacked on the Grain family of ciphers. Faults were induced in LFSR, NFSR or both LFSR and NFSR. They assumed that the cipher was rekeyed

multiple times and at random location and time, faults can be injected. By this way, the attacker did not need to control location and time. They generated variables from cipher's algebraic description. Therefore, degrees of the equations did not affect. Researchers broke Grain v1 with 10 faults, Grain-128 with 4 faults and Grain-128a with 10 faults. They believed that results can be improved by using more computational power. Phase-shift fault attacks are a variant of the fault analysis. Hoch and Shamir introduced in 2004, and then in 2018, Hromada and Pethö used this attack on Grainv1. Hridya and Jose (2022) applied a phase-shift fault attack on Grain-128 and they got a key in case of an algorithm with weak key-IV. They reported that this type of attack can be challenging. To sum up, the Grain family of ciphers are still under investigation, and it is an open problem whether the algorithm is resistant to fault attacks or not. However, designers of the Grain-128 AEADv2 emphasize that the algorithm is not resistant to fault attacks and it needs an extra mechanism to increase protection against this attack type. It can be said that fault attacks are still a threat for Grain-128 AEADv2 (Hell et al., 2019).

5.2. Cube Attacks

Cube attacks can be categorized as key derivation attack and it can be applied on both block ciphers and the stream ciphers. However, block ciphers are more resistant than stream ciphers. In 2009, cube attacks as a cryptanalytic technique were proposed by Dinur and Shamir. If the attacker discovers an output function's lower degree polynomial representation, then the secret key can be recovered (Videau, 2011; Dinur and Shamir, 2012). Even though the attacker does not know the polynomial, still a cube attack can be applied. In the original cube attacks, the attacker does not have clean data. Instead, data may contain errors which means that the attacker may have noisy data. However, cube attacks have lower error tolerance. Therefore, robust cube attacks were developed (Dinur and Shamir, 2010; Dinur and Shamir, 2012). Although cube attacks and high order differential attacks have similarities, there are some main differences. Cube attacks try to break cipher algebraically from a given polynomial by solving linear equations. However, high order differential cryptanalysis applies to statistical analysis. Formal differentiation operator is not used by cube attacks. Cube attacks on stream ciphers require knowledge of a bit of the keystream (Dinur and Shamir, 2012). Dinur and Shamir (2011) presented a dynamic cube attack. In the attack, lower degree representation of the output function was exploited in the key and plaintext. It made the algorithm vulnerable to the attack, although the cipher was resistant to all previous

standard cube attacks. If from cube testers, distinguishers were exploited, by using them secret key also could be recovered. It is the difference between standard cube attacks and new variants of the attack. Standard cube attack stands on the discovery of the linear equations, and then the secret key can be found. Cube testers can be used in black box representation of the cryptosystems, and they are generic distinguisher families. They distinguish random function and the cipher by using resultant sums. However, standard cube attacks use the sums of the secret key bits' linear equation. If a big cube is chosen, then it affects the attack complexity. If the degree of the polynomial can be reduced by simplifying the algebraic normal form of the intermediate state bit used to encrypt, then cube testers become more vulnerable and open to threats. Authors presented three attacks on the algorithm. First attack was the full key recovery attack, and they decreased the number of the initialization rounds to 207. They got a 128-bit key. Second attack tried to get a full key with 250 initialization rounds. It performed faster than exhaustive search. Lastly, they attacked the full version of Grain-128 and obtained a recovering subset of two to ten of weak keys. Exhaustive search was slower than the attack. To increase resistance against cube attacks, Grain-128AEADv2 has a function of degree four (Hell et al., 2019).

5.3. Time-Memory Tradeoff and Time-Memory-Data Tradeoff Attacks

Hellman first introduced time-memory trade-off attacks and in the first phase in other word offline phase, large tables were created, and this information was used in the online phase. Stream ciphers are accepted as a one-way function which generates n bit keystream from n bit innerstate. Therefore, recovering the innerstate is critical (Hellman, 1980). This type of attack is crucial for stream cipher in terms of security threats. Therefore, stream ciphers should be resistant to this type of attack. A variant of this attack can decrease the time complexities of the attack both off-line and on-line (Dunkelman, 2008). Time memory trade-off attacks consist of two main phases which are pre-processing and real time phases. In the first phase, the attacker spends much of the time understanding the cryptosystem and making intuition. After this investigation, with the information he gets, the real time phase is started. In the second phase, by using an unknown key a real data is generated and given to the attacker. The main purpose is to find the key as soon as possible. Therefore, the attacker uses the table which is precomputed. This type of attacks mainly has five parameters which are size of search space denoted by N, time required for pre-processing phase denoted by P, time required for real time phase denoted by T, available amount of random-access memory

denoted by M and lastly amount of real time data denoted by D (Biryukov and Shamir, 2000). Babbage (1995), Golic (1997), Biryukov and Shamir (2000), Wagner (2001) and Bjørstad (2013) worked on time-memory tradeoff attacks and they have improved it from Hellman's version. If data is used to improve the attack, then it turns into time-memory-data trade-off attacks. It uses the birthday paradox. Time-memory-data trade-off attacks have successful tries on stream ciphers and cipher was broken. In 1995, Babbage set a rule for resistant algorithm design: "if a secret key length of k bits is required, a state size of at least 2k bits is desirable". De Canniere, Küçük and Preneel (2008) applied a time-memory data attack on Grainv1. Bjørstad (2013) discussed that Grainv1 innerstate generates a keystream with a fixed 21 bits. It means that the algorithm's sampling resistance is 2^{-21}. Innerstate can be recovered after $O(2^{106.5})$ precomputation steps by using $O(2^{53.5})$ bits of the known keystream when the time and memory complexity is $O(2^{71})$. Innerstate design of Grainv1 makes the algorithm target of the attackers for time-memory-data trade-off attacks.

5.4. Chosen IV Attacks

Chosen IV related key attack exploits vulnerability of the symmetric padding. The main purpose is obtaining a shifted keystream by using chosen IV and the related keys. Then they try to recover secret key bits. Lee et al. (2008) attacked Grain and Grainv1 by using this vulnerability. After this feedback, in the algorithm design symmetric padding is replaced with asymmetric padding and existing attacks cannot break the algorithm (De Canniere et al., 2008; Banik et al., 2013). Lee et al.'s attack was based on slide resynchronization attack (Küçük, 2006). Then Banik et al. (2013) propose a key recovery attack on Grain-128a and recover the secret key from secret key bits utilizing nonlinear equations in it. It was more complex than previous attacks. Therefore, it required more time. Chang and Turan (2021) analysed if the state is known by the attacker, can key be recovered. In three different cases, they reintroduced the key to the inner state, and they found that the design of the algorithm was not enough to protect the key from recovery attacks from the inner state. In the first scenario, by using full inner state (including LFSR, NFSR, accumulator and the register) knowledge, the algorithm could be broken after initialization. In the second case, they recovered the state during the encryption, although the attacker did not know the state of the accumulator and the register. In the last scenario, the nonce-misuse setting was attacked. It can be said that the key recovery attack protection strategy of Grain-128AEAD failed. In each clock, the inner state was updated by inserting a single bit of the key. It made the

algorithm vulnerable. After that analysis, Grain-128AEAD was turned into Grain-128AEADv2 with an additional tweak. Purpose of the designers was to increase complexity and prevent recovering the key from the state (Hell et al., 2019).

5.5. Correlation Attacks

Correlation attacks try to find correlation between LFSR and the key stream. (Hell et al., 2019). In 2017, Zhang et al. worked on fast correlation attacks which were applied on LFSR-based ciphers. They applied the attack on small-state ciphers and their strategy was "divide-and-conquer". They reported that the output function should not have pseudo-linear properties and linear approximation of NFSR should be avoided. They attacked Grainv1. However, the attack was unsuccessful because of the bigger length of LFSR. Grain-128a is also resistant to fast correlation attacks, thanks to its bigger state. Todo et al. (2018) attacked Grain-128a and if the data is received from the secret key and nonce, then the algorithm can be broken. Time complexity of the attack is approximately 2^{114}. However, designers of the algorithm highlighted that Grain-128a was in the mode without authentication. Therefore, it can affect the results. In addition to them, any attackers have not been applied this attack on Grain-128AEADv2 (Hell et al., 2019).

6. Hardware Implementation

For the lightweight cryptography algorithms, performance metrics change according to the devices. Constrained devices have limited batteries. They should run without changing the battery. Therefore, low power consumption is important for the design of the algorithm (Hell et al., 2019). In addition, area is correlated with the cost. Therefore, reducing area helps to cost minimization especially for large lot production. Also, sometimes low latency and high-speed transformation of big data is needed. Those points shape the design of the algorithm (Sönnerup et al., 2019). There are some lightweight cryptography algorithms (Sprout, Plantlet and Fruit family algorithms) which were inspired from Grain and their main concern was minimizing hardware footprint. Although their design does not have the same structure with Grain, all of them are based on the same design idea. In the case of Grain, keys are updated in the device and the cipher does not contain key storage. It increases use cases of the algorithm. Grain family of ciphers are hardware-based ciphers, and they are appropriate for the low-end devices. NAND gates, XOR gates and flip flops are the hardware building blocks which can be used for constructing Grain-128AEADv2. (Hell et al., 2019). Grain is

hardware-based stream cipher, and it is suitable for resource limited (memory, number of gates and power consumption) implementations (Dhanda et al., 2020). Tsantikidou and Sklavos (2022) implemented Grainv1, and they achieved small area design with its serial versions. However, Trivium generates more throughput than Grainv1.

Designers aim to create algorithms suitable for both high speed and low-power implementations. For Grain-128AEAD, increasing area causes higher power consumption. However, required energy for encrypting a packet is decreased. Therefore, a trade-off between area, energy, power consumption and throughput are observed (Sönnerup et al., 2019). Sönnerup et al. (2019) made different implementations of Grain-128AEAD by using a 65 nm library to develop a strategy for obtaining either high throughput or low consumption. They reduced power up to 94%, for high-speed implementation they reach up to 33.6 Gb/s throughput. The most power efficient implementation (best value of throughput/power) is observed in parallelization level four. Designers of the algorithm compare implementation results on both Grain-128AEADv2 and Grain-128AEAD. Difference in area can be accepted as negligible, but gate counts are nearly the same. However, initialization time is 33% slower (Hell et al., 2019). Mansouri and Dubrova (2012) proposed an improved hardware implementation. They studied Grain-128a and utilized the combined effect of different techniques. They got an average 56% higher throughput, but the initialization phase increased 21%. As a result, they reach 23%higher throughput in all phases.

7. Conclusion

Constrained devices, applications and IoT technology open the door of a new are. Interoperability, connectivity, real time sensing and decentralization are the new trends of this world. However, with technological improvements, security becomes more and more important. New concepts need tailored made solutions in terms of security. Lightweight cryptography can meet the security needs. Grain-128AEADv2 is one of the lightweight cryptography algorithms. Grain was proposed in 2005, and used the cryptography community's feedback as a design strategy to make algorithms more robust and secure. Latest version of the algorithm is Grain-128AEADv2 and it still has some points to be improved to stand against possible attack.

References

Agren, M., Hell, M., Johansson, T., Meier, W. (2011). Grain-128 a: A New Version Of Grain128 With Optional Authentication. International Journal of Wireless and Mobile Computing 5(1), 48–59.

Babbage, S.H. (1995). Improved "Exhaustive Search" Attacks On Stream Ciphers. In Security and Detection.

Banik, S., Maitra, S., Sarkar, S. (2012). A Differential Fault Attack on the Grain Family under Reasonable Assumptions. Progress in Cryptology - INDOCRYPT 2012 13th International Conference on Cryptology in India, pp. 191-208.

Banik, S., Maitra, S., Sarkar, S. (2012). A Differential Fault Attack on Grain-128a using MACs. Security Privacy and Applied Cryptography Engineering - Second International Conference SPACE 2012, pp. 111-125, November 3-4, 2012.

Banik, S., Maitra, S., Sarkar, S. (2015). Differential Fault Attack against Grain Family with Very Few Faults and Minimal Assumptions. IEEE Trans. Computers, 64(6).

Banik, S., Mikhalev, V., Armknecht, F., Isobe, T., Meier, W., Bogdanov, A., Watanabe, Y., & Regazzoni, F. (2018). Towards Low Energy Stream Ciphers. IACR Transactions on Symmetric Cryptology, 2018(2), 1–19. https://doi.org/10.13154/tosc.v2018.i2.1-19

Banik, S., Maitra, S., Sarkar, S., & Turan, M.S. (2013). A Chosen IV Related Key Attack on Grain-128a. Australasian Conference on Information Security and Privacy.

Beighton, M., Bartlett, H., Simpson, L., Wong, K.KH. (2022). Algebraic Attacks on Grain-Like Keystream Generators. In: Park, J.H., Seo, SH. (eds) Information Security and Cryptology – ICISC 2021. ICISC 2021. Lecture Notes in Computer Science, vol 13218. Springer, Cham. https://doi.org/10.1007/978-3-031-08896-4_12

Biryukov, A. (2005). Some Thoughts on Time-Memory-Data Tradeoffs. IACR Cryptol. ePrint Arch., 2005, 207.

Biryukov, A., Shamir, A. (2000). Cryptanalytic Time/Memory/Data Tradeoffs for Stream Ciphers. In: Okamoto, T. (eds) Advances in Cryptology — ASIACRYPT 2000. ASIACRYPT 2000. Lecture Notes in Computer Science, vol 1976. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44448-3_1

Biswas, K., Muthukkumarasamy, V., Wu, X. W., Singh, K. (2016). Performance evaluation of block ciphers for wireless sensor networks. In R. Choudhary, J. Mandal, N. Auluck, & H. Nagarajaram (Eds.), Advanced Computing and Communication Technologies. Advances in Intelligent Systems and Computing, Vol. 452. Springer, Singapore.

Bjørstad, T. Cryptanalysis of Grain using Time/Memory/Data Tradeoffs, 2013.

Bokhar, M., Alam, S., Hamid, S. (2014). A Detailed Analysis of the Grain Family of Stream Ciphers. International Journal of Computer Network and Information Security, 6. 10.5815/ijcnis.2014.06.05.

Buchanan, W., Li, S., & Asif, R. (2017). Lightweight Cryptography Methods. Journal of Cyber Security Technology, 1(3-4), 187-201.DOI:10.1080/23742917.2017.1384917

Chang, D., Turan, M. S. (2021). Recovering the Key from the Internal State of Grain-128AEAD. IACR Cryptol. ePrint Arch. 2021: 439.

Civek, A. B. (2021). Differential-linear cryptanalysis of Ascon and Drygascon, Master Degree, Middle East Technical University Graduate School of Natural and Applied Sciences, Ankara.

Dalai, D. K., Pal, S., Sarkar, S. (2022). Some Conditional Cube Testers for Grain-128a of Reduced Rounds. IEEE Transactions on Computers, 71(6), 1374-1385. DOI: 10.1109/TC.2021.3085144.

De Cannière, C., Küçük, Ö., Preneel, B. (2008) Analysis of Grain's Initialization Algorithm. Vaudenay, S. (ed.) AFRICACRYPT 2008. LNCS, vol. 5023, pp. 276–289. Springer, Heidelberg.

Deb, S., Bhuyan, B. (2018). Performance evaluation of grain family and espresso ciphers for applications on resource constrained devices. ICT Express, 4(1), 19–23.

Dhanda, S.S., Singh, B., Jindal, P. (2020). Lightweight Cryptography: A Solution to Secure IoT. Wireless Pers Commun, 112, 1947–1980. https://doi.org/10.1007/s11277-020-07134-3

Dinur, I., Shamir, A. (2011). Breaking Grain-128 with Dynamic Cube Attacks. In: Joux, A. (eds) Fast Software Encryption. FSE 2011. Lecture Notes in Computer Science, vol 6733. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-21702-9_10

Dinur, I., Shamir, A. (2012). Applying cube attacks to stream ciphers in realistic scenarios. Cryptogr. Commun. 4, 217–232. https://doi.org/10.1007/s12095-012-0068-4

Dunkelman, O.; Keller, N. Treatment of the initial value in Time-Memory-Data Trade-off attacks on stream ciphers. Inf. Process. Lett. 2008, 107, 133–137.

Golic, J.D. (1997). Cryptanalysis of Alleged A5 Stream Cipher. International Conference on the Theory and Application of Cryptographic Techniques.

Granjal, J., Monteiro, E., Silva, J. S. (2015). Security in the integration of low-power wire- less sensor networks with the internet: A survey. Ad Hoc Networks,24, 264–287.

Hell, M., Johansson, T., & Meier, W. (2007). Grain: A Stream Cipher For Constrained Environments. *Int. J. Wirel. Mob. Comput., 2*, 86-93.

Hell, M., Johansson, T., Maximov, A., Meier, W. (2006). A stream cipher proposal: Grain128. IEEE International Symposium on Information Theory. https://doi.org/10.1109/ISIT.2006.261549

Hell, M., Johansson, T., Maximov, A., & Meier, W. (2008). The Grain Family of Stream Ciphers. The eSTREAM Finalists.

Hell, M., Johansson, T., Maximov, A., Meier, W., & Yoshida, H. (2021). Grain-128AEADv2: Strengthening the Initialization Against Key Reconstruction. IACR Cryptol. ePrint Arch..

Hellman, M. (1980). A Cryptanalytic Time-Memory Trade-Off. *IEEE Transactions on Information Theory*, 26(4). https://doi.org/10.1109/TIT.1980.1056220.

Hoch, J. J., Shamir, A. (2004). Fault Analysis of Stream Ciphers. Cryptographic Hardware and Embedded Systems - CHES 2004: 6th International Workshop Cambridge, pp. 240-253, August 11-13, 2004.

Hromada, V., Petho, T. (2005). Phase-shift Fault Analysis of Grain v1. ECRYPT Stream Cipher Project Report, 71.

Hridya P. R, Jose, J. (2022). Phase-shift Fault Analysis of Grain-128. IACR Cryptol. ePrint Arch. 2022: 387

Kamal, R. (2017). Internet of Things: Architecture and Design Principles, (p. 403), TMH, India, ISBN-13: 978-93-5260-522-4.

Karmakar, S., Chowdhury, D. R. (2014). Fault Analysis of Grain Family of Stream Ciphers. Cryptology ePrint Archive, Paper 2014/261. https://eprint.iacr.org/2014/261

Küçük, Ö. (2006). Slide resynchronization attack on the initialization of Grain 1.0. eSTREAM, ECRYPT Stream Cipher Project, Report 2006/044

Lee, Y., Jeong, K., Sung, J., Hong, S. (2008). Related-Key Chosen IV Attacks on Grain-v1 and Grain-128. ACISP, 5107, pp. 321–335. Springer, Heidelberg.

Li, S. (2017). Security and Vulnerability in the Internet of Things. Securing the Internet of Things, Elsevier, pp. 49-68, 10.1016/B978-0-12-804458-2.00003-2

Maximov, A. (2006). Cryptanalysis of the "Grain" family of stream ciphers. In Proceedings of the 2006 ACM Symposium on Information, Computer and Communications Security (ASIACCS '06). Association for Computing Machinery, New York, NY, USA, 283–288. https://doi.org/10.1145/1128817.1128859

Mansouri, S.S., Dubrova, E. (2013). An Improved Hardware Implementation of the Grain-128a Stream Cipher. In: Kwon, T., Lee, MK., Kwon, D. (eds) Information Security and Cryptology – ICISC 2012. ICISC 2012. Lecture Notes in Computer Science, vol 7839. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-37682-5_20

NISTIR 8114 - Report on Lightweight Cryptography, March 2017.

Quynh, L. N., Tran, T. N., Ngo, C. K., Tran, H. D., Nguyen, V. C., Tran, T. A. (2022). Implementation Of Authenticated Encryption With Associated Data Grain-128aead Algorithm On Stm32f400 Processor Family. The Transport and Communications Science Journal, 73(4). DOI:10.47869/tcsj.73.4.7

Ragab, A., Madani, A., Wahdan, A.A., & Selim, G.E. (2019). Hybrid Cryptosystems for Protecting IoT Smart Devices with Comparative Analysis and Evaluation. Advances in Intelligent Systems and Computing.

Rezvani, B., & Diehl, W. (2019). Hardware Implementations of NIST Lightweight Cryptographic Candidates: A First Look. *IACR Cryptol. ePrint Arch., 2019*, 824.

Singh, S., Sharma, P. K., Moon, S. Y., Park, J. H. (2017). Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions. Journal of Ambient Intelligece & Human Computing. https://doi.org/10.1007/s12652-017-0494-4

Sönnerup, J., Hell, M., Sönnerup, M., Khattar, R. (2019). Efficient Hardware Implementations of Grain-128AEAD. In: Hao, F., Ruj, S., Sen Gupta, S. (eds) Progress in Cryptology – INDOCRYPT 2019. INDOCRYPT 2019. Lecture Notes in Computer Science, 11898. Springer, Cham. https://doi.org/10.1007/978-3-030-35423-7_25

Todo, Y., Isobe, T.,Meier, W., Aoki, K., Zhang, B. (2018). Fast Correlation Attack Revisited: 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19–23, 2018, Proceedings, Part II. 10.1007/978-3-319-96881-0_5.

Toshihiko, O. (2017). Lightweight Cryptography Applicable to Various IoT Devices. NEC Technical Journal,12(1), 67-71.

Tsantikidou, K., Sklavos, N. (2022). Hardware Limitations of Lightweight Cryptographic Designs for IoT in Healthcare. Cryptography, 6(3), 45. https://doi.org/10.3390/cryptography6030045

Turan M.S., Doğanaksoy A., Boztaş S. (2008). On independence and sensitivity of statistical randomness tests. International Conference on Sequences and Their Applications, Springer pp. 18-29.

Verdult, Roel. “Introduction to Cryptanalysis : Attacking Stream Ciphers.” (2015).

Videau, M. (2011). Cube Attack. In: van Tilborg, H.C.A., Jajodia, S. (eds) Encyclopedia of Cryptography and Security. Springer, Boston, MA. https://doi.org/10.1007/978-1-4419-5906-5_342

Youssef, W. E. H., Abdelli, A., Dridi, F., Machhout, M. (2020). Hardware Implementation of Secure Lightweight Cryptographic Designs for IoT Applications. Security and Communication Networks, 2020. https://doi.org/10.1155/2020/8860598

Zhang, B., Gong, X., Meier, W. (2017). Fast Correlation Attacks on Grain-like Small State Stream Ciphers. IACR Transactions on Symmetric Cryptology, 2017(4), 58–81. https://doi.org/10.13154/tosc.v2017.i4.58-8

http://www.owsap.org/index.php/OWASP_Internet_of_Things_Project (Last Access: 2022-12-28).

https://www.iso.org/standard/60682.html (Last Access: 2022-12-28).

https://cdn.standards.iteh.ai/samples/60682/14664246686e487fa20d4e2d73a5b621/ISO-IEC-29167-13-2015.pdf (Last Access: 2022-12-28).

https://csrc.nist.gov/CSRC/media/Presentations/lightweight-cryptography-standardization-process/images-media/Talk-ICMC2021_LWC%20standardization-meltem-Sept2021.pdf (Last Access: 2022-12-28).

https://csrc.nist.gov/CSRC/media/Presentations/Applications-and-Standardization-of-Lightweight-Cr/images-media/Talk-SAC-SummerSchool-meltem-Aug2018.pdf (Last Access: 2022-12-28).

https://csrc.nist.gov/CSRC/media/Projects/Lightweight-Cryptography/documents/final-lwc-submission-requirements-august2018.pdf (Last Access: 2022-12-28).

ECRYPT– European Network of Excellence for Cryptology. eSTREAM: the ECRYPT stream cipher project, 2008. http://www.ecrypt.eu.org/stream/. (Last Access: 2022-12-28).