CSE 470

Cryptography and Computer Security

Fall 2022 – 2023

Project Report

30.12.2022

Abdurrahman Bulut

1901042258

Contents

1. Introduction

2. Grain128-AEAD

3. Photon-Beetle

4. CBC, OFB and HASHING

5. Comparison with Other Algorithms

6. Tests

7. Conclusion

8. Resources

1. Introduction

Cryptography is the practice of secure communication, which involves the use of mathematical algorithms and secret keys to protect the confidentiality, integrity, and authenticity of information. Cryptology is the study of cryptography, which includes the principles, techniques, and protocols used for secure communication.

There are several reasons why we need cryptography and cryptology:

- Confidentiality: Cryptography helps to protect the confidentiality of sensitive information by making it unreadable to anyone without the proper decryption key. This is important for protecting personal information, financial data, and other sensitive information from being accessed by unauthorized parties.

- Integrity: Cryptography can also be used to ensure the integrity of data by generating a hash value or message digest for the data. If the data is changed in any way, the hash value will also change, alerting the recipient that the data has been tampered with.

- Non-repudiation: Cryptography can be used to provide non-repudiation, which is the ability to prove that a message was sent by a particular person. This is achieved by using digital signatures, which are created using the sender's private key and can be verified using the corresponding public key.

- Authentication: Cryptography can be used to authenticate the identity of a sender or recipient. For example, when a user logs into a website, the website can use cryptography to verify the user's identity based on their login credentials.

- Secure communication: Cryptography is essential for secure communication over the internet and other networks. It allows two parties to communicate confidentially, even if their communication is being intercepted by a third party.

Cryptography and cryptology are important because they help to protect the confidentiality, integrity, and authenticity of sensitive information and enable secure communication over networks. They are widely used in a variety of applications, including online communication, data storage, and secure financial transactions.

Encryption and decryption are important because they allow us to protect sensitive information from unauthorized access. Encryption is the process of converting plaintext (readable) data into ciphertext (unreadable) data using a mathematical algorithm and a secret key. The ciphertext can only be converted back into plaintext (decrypted) using the same algorithm and the corresponding key.

2. Grain128-AEAD

Grain128-AEAD is a lightweight cryptographic algorithm that was designed to be used in resource-constrained environments, such as Internet of Things (IoT) devices, smart cards, and other embedded systems. It is a symmetric key algorithm, which means that the same key is used for both encryption and decryption.

Grain128-AEAD is a combination of the Grain stream cipher and an Authenticated Encryption with Associated Data (AEAD) construction. The Grain stream cipher is a lightweight, self-synchronizing stream cipher that uses a 128-bit key and generates a pseudo-random stream of bits that is used to encrypt the data. The AEAD construction is a type of encryption algorithm that provides both confidentiality and integrity protection for the data.

There are two primary components that compose grain128-AEAD. The first is an authentication generator formed of a shift register and an accumulator, while the second is a pre-output generator formed of a linear feedback shift register (LFSR), a non-linear feedback shift register (NFSR), and a pre-output function.

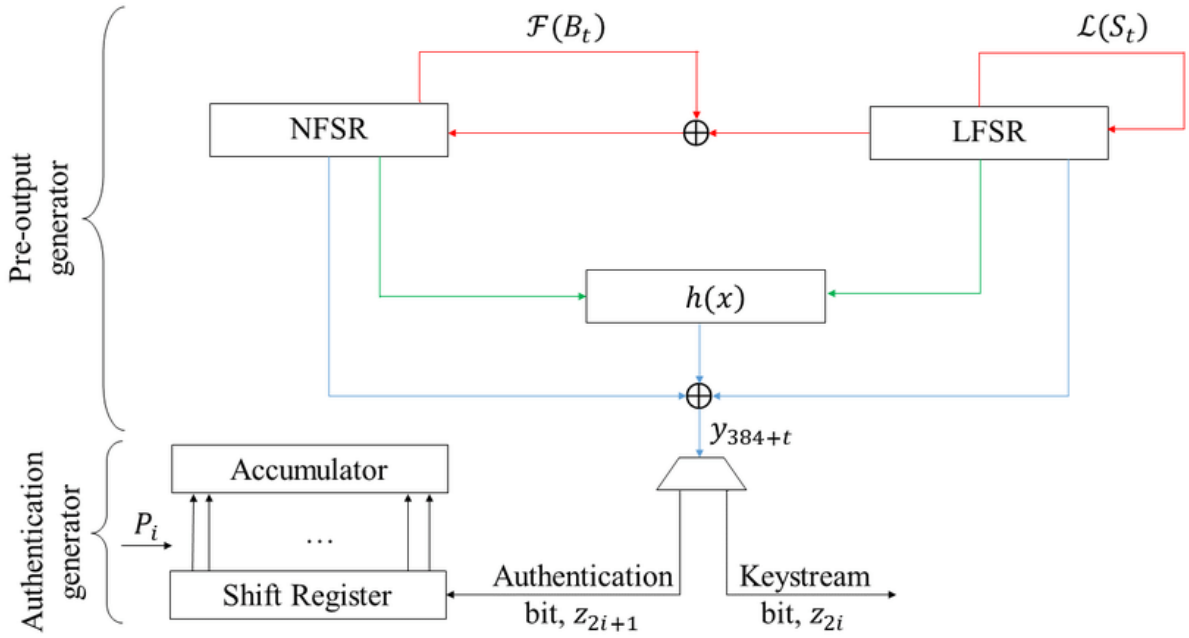

FIGURE 1. An overview of the building blocks in Grain-128AEAD.

Grain128-AEAD has several security features, including:

- Key size: The algorithm uses a 128-bit key, which provides a high level of security.
- Resistance to attacks: Grain128-AEAD has been designed to be resistant to various attacks, including differential and linear cryptanalysis, brute-force attacks, and side-channel attacks.
- Nonce reuse protection: The algorithm includes a nonce (a unique value used in the encryption process) to prevent nonce reuse, which can lead to security vulnerabilities.
- Memory efficiency: Grain128-AEAD uses a small amount of memory, making it suitable for use in resource-constrained environments.

In summary, Grain128-AEAD is a lightweight cryptographic algorithm that provides confidentiality, integrity, and nonce reuse protection for data. It is suitable for use in resource-constrained environments and has a high level of security.

Grain128-AEAD is a cryptographic algorithm that was designed to be resistant to various attacks. Some of the potential attack types that Grain128-AEAD may be vulnerable to include:

- Brute-force attacks: These attacks involve attempting to decrypt ciphertext by trying all possible keys until the correct one is found. Grain128-AEAD uses a 128-bit key, which makes it resistant to brute-force attacks because the number of possible keys is very large and it would take an impractical amount of time to try them all.

- Differential and linear cryptanalysis: These attacks involve analyzing the patterns and relationships in the encrypted data to try to find vulnerabilities that can be exploited. Grain128-AEAD has been designed to be resistant to differential and linear cryptanalysis.

- Side-channel attacks: These attacks involve analyzing the physical characteristics of a device, such as power consumption, electromagnetic radiation, or the timing of operations, to try to extract sensitive information. Grain128-AEAD has been designed to be resistant to side-channel attacks.

It is important to note that no cryptographic algorithm is completely immune to all attacks and it is possible that new vulnerabilities may be discovered in the future. It is important to regularly review and update cryptographic algorithms to ensure that they continue to provide an acceptable level of security.

Here are some of the pros and cons of using Grain128-AEAD:

Pros:

- Lightweight: Grain128-AEAD is a lightweight algorithm that uses a small amount of memory, making it suitable for use in resource-constrained environments.

- High security: Grain128-AEAD uses a 128-bit key, which provides a high level of security.

- Nonce reuse protection: The algorithm includes a nonce (a unique value used in the encryption process) to prevent nonce reuse, which can lead to security vulnerabilities.

- Resistance to attacks: Grain-128-AEAD has been designed to be resistant to various attacks, including differential and linear cryptanalysis, brute-force attacks, and side-channel attacks.

Cons:

- Limited availability: Grain128-AEAD is not as widely used as some other cryptographic algorithms, which may limit its availability for certain applications.

- Limited security: While Grain128-AEAD provides a high level of security, it may not offer the same level of security as more complex cryptographic algorithms.

Overall, Grain128-AEAD is a suitable cryptographic algorithm for use in resource-constrained environments where a high level of security is required. However, it may not be the best choice for applications that have more stringent security requirements or that require a widely available algorithm.

3. Photon-Beetle

Photon-Beetle is a cryptographic algorithm that was designed to be used in resource-constrained environments, such as Internet of Things (IoT) devices, smart cards, and other embedded systems. It is a symmetric key algorithm, which means that the same key is used for both encryption and decryption.

Photon-Beetle is a combination of a block cipher and a stream cipher. The block cipher is used to encrypt data in fixed-size blocks, while the stream cipher is used to encrypt data one bit or byte at a time. The block cipher uses a 256-bit key and the stream cipher uses a 128-bit key.

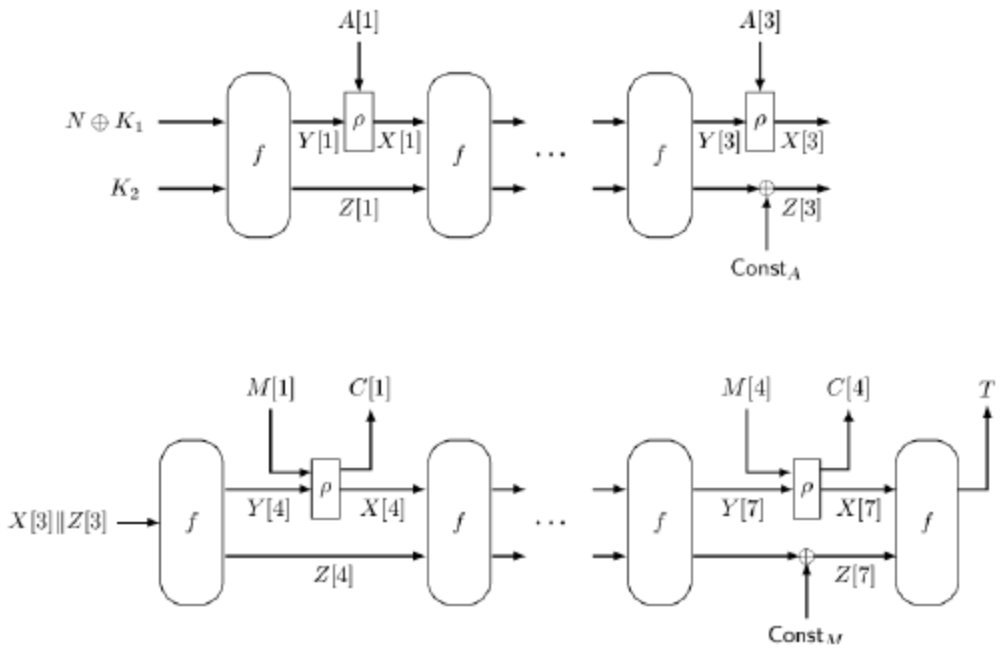

Figure 2: Beetle Construction with a = 3 associated data blocks and m = 4 message blocks. The value ConstA and ConstM is used to denote whether last associated data block and message block is complete or not.

Photon-Beetle has several security features, including:

- Key size: The algorithm uses a 256-bit key for the block cipher and a 128-bit key for the stream cipher, which provides a high level of security.
- Resistance to attacks: Photon-Beetle has been designed to be resistant to various attacks, including differential and linear cryptanalysis, brute-force attacks, and side-channel attacks.
- Algebraic attacks: Photon-Beetle has been designed to be resistant to algebraic attacks, which involve analyzing the mathematical properties of the cryptographic algorithm to try to find vulnerabilities that can be exploited.
- Memory efficiency: Photon-Beetle uses a small amount of memory, making it suitable for use in resource-constrained environments.

In summary, Photon-Beetle is a lightweight cryptographic algorithm that provides a high level of security and is suitable for use in resource-constrained environments. It combines a block cipher and a stream cipher and has been designed to be resistant to various attacks.

Here are some of the pros and cons of using Photon-Beetle:

Pros:

- Lightweight: Photon-Beetle is a lightweight algorithm that uses a small amount of memory, making it suitable for use in resource-constrained environments.
- High security: Photon-Beetle uses a 256-bit key for the block cipher and a 128-bit key for the stream cipher, which provides a high level of security.
- Resistance to attacks: Photon-Beetle has been designed to be resistant to various attacks, including differential and linear cryptanalysis, brute-force attacks, side-channel attacks, and algebraic attacks.

Cons:

- Limited availability: Photon-Beetle is not as widely used as some other cryptographic algorithms, which may limit its availability for certain applications.
- Limited security: While Photon-Beetle provides a high level of security, it may not offer the same level of security as more complex cryptographic algorithms.

Overall, Photon-Beetle is a suitable cryptographic algorithm for use in resource-constrained environments where a high level of security is required. However, it may not be the best choice for applications that have more stringent security requirements or that require a widely available algorithm.

4. CBC, OFB and HASHING

- **CBC**

  CBC (Cipher Block Chaining) is a mode of operation for a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. In CBC mode, each block of plaintext is XORed (Exclusive OR) with the previous block of ciphertext before being encrypted. This creates a chain of blocks, hence the name "Cipher Block Chaining."

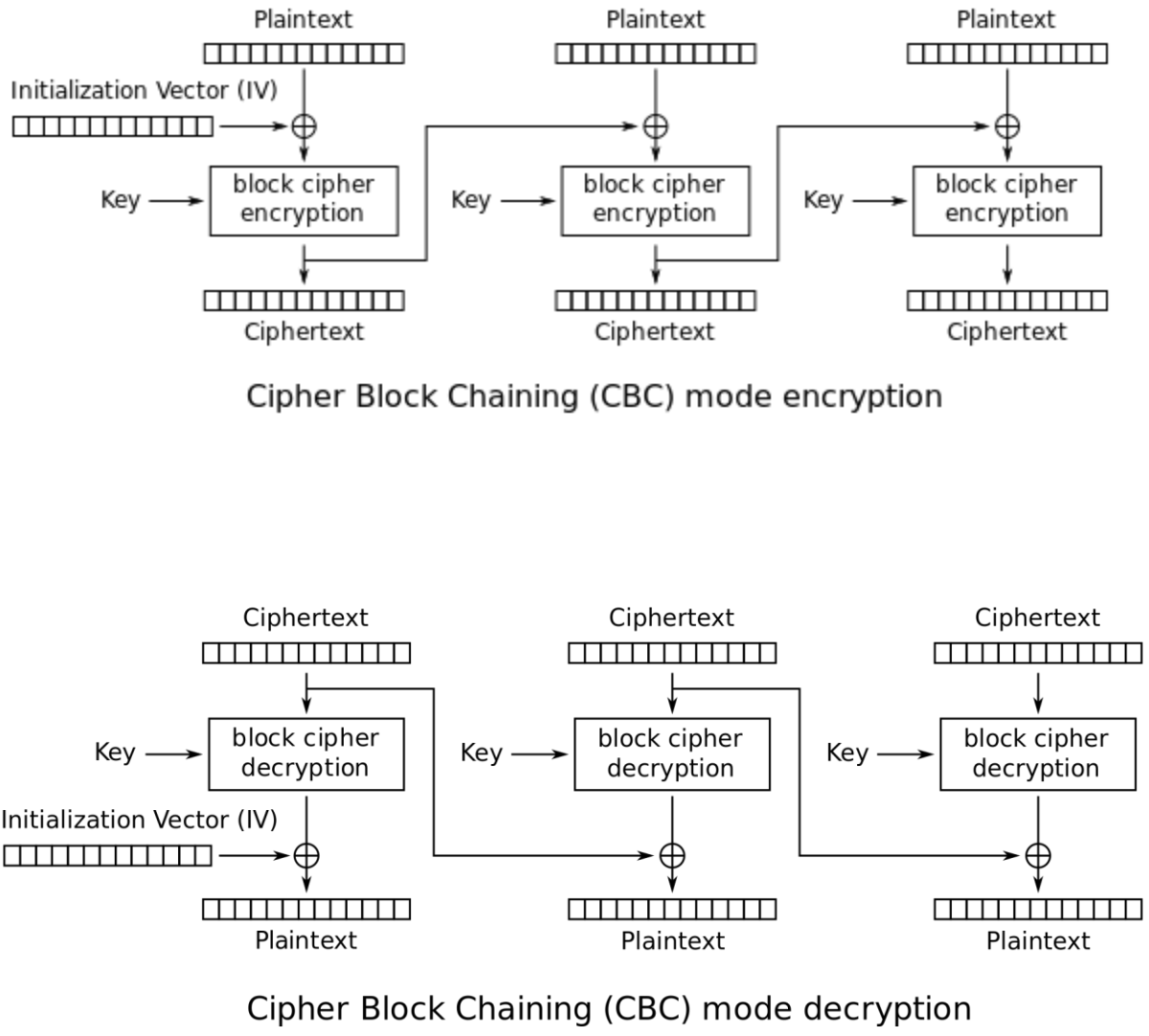

Figure 3: Encryption and Decryption using the Cipher Block Chaining (CBC) mode.

The first block of plaintext is typically XORed with an initialization vector (IV) before being encrypted. The IV is a randomly generated block of data that is used to ensure that the same plaintext block is encrypted differently each time it is encrypted.

One of the main advantages of CBC mode is that it provides a high level of confidentiality, since the same plaintext block will be encrypted differently each time it is encrypted. It also provides some level of integrity protection, since any changes to the ciphertext will propagate to the subsequent blocks of ciphertext and will be detected when the ciphertext is decrypted.

However, CBC mode is vulnerable to certain types of attacks, such as padding oracle attacks. To address this, it is often used in combination with message authentication codes (MACs) to provide both confidentiality and integrity protection.

- **OFB**

OFB (Output Feedback) is a mode of operation for a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. In OFB mode, the block cipher is used as a keyed pseudo-random number generator, and the plaintext is XORed (Exclusive OR) with the generated pseudo-random sequence to produce the ciphertext.

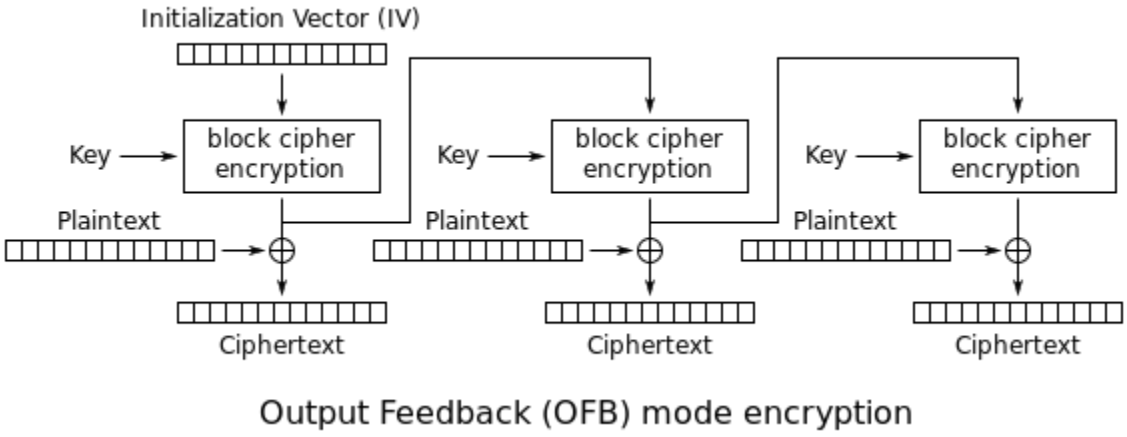

Figure 4: OFB encryption schema

The pseudo-random sequence is generated by encrypting an initialization vector (IV) with the block cipher, and then using the resulting ciphertext block as the input to generate the next block in the sequence. This process is repeated for each block of plaintext that needs to be encrypted.

One of the main advantages of OFB mode is that it is resistant to the kinds of errors that can occur during transmission, such as bit flips or insertions/deletions. This is because the errors will affect the ciphertext, but will not propagate to the plaintext when it is decrypted.

However, OFB mode does not provide any integrity protection, since an attacker can modify the ciphertext without being detected. Therefore, it is typically used in combination with a message authentication code (MAC) to provide both confidentiality and integrity protection.

- **HASHING**

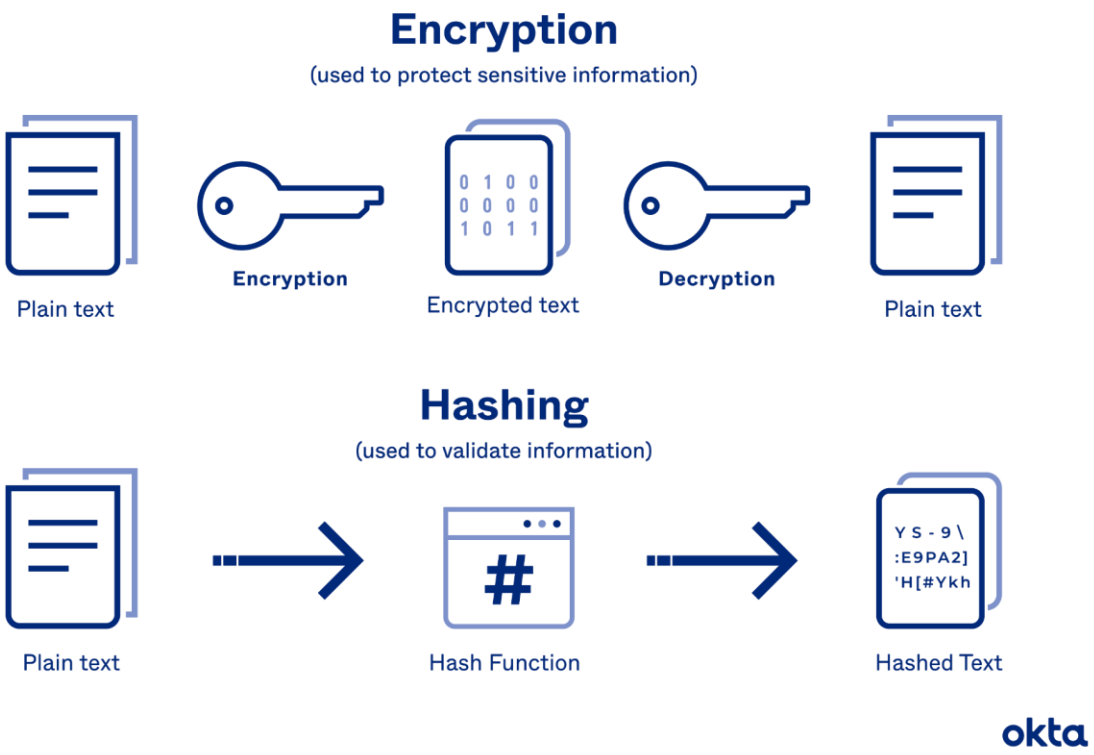

Figure 5: Hashing

Hashing is a process in which a fixed-size string of data, called a hash or message digest, is generated from a larger piece of data, such as a file or message. The process of generating a hash is known as hashing or message digesting.
There are several properties that make hashing useful for a variety of applications:

- One-way function: It is computationally infeasible to recreate the original data from the hash, making it a one-way function.

- Fixed-size output: Hashes are typically a fixed length, regardless of the size of the input data.

- Deterministic: Given the same input data, the same hash will be produced every time.

- Unique: It is very unlikely that two different pieces of data will produce the same hash.

Hashing is used for a variety of purposes, including password storage, data integrity checks, and data indexing and retrieval. In password storage, a user's password is hashed and stored, rather than storing the actual password. When the user attempts to login, the entered password is hashed and compared to the stored hash. If the hashes match, the user is authenticated.

In data integrity checks, a hash is calculated for a piece of data, and the hash is stored with the data. If the data is modified in any way, the hash will no longer match, indicating that the data has been tampered with.

In data indexing and retrieval, hashes can be used to quickly locate and retrieve specific records from a large database. The hash of the search key is calculated, and the record with the matching hash is retrieved.

There are several types of hashing algorithms, including MD5, SHA-1, and SHA-2. It is important to use a strong hashing algorithm to ensure the security and integrity of the data.

5. Comparison with Other Algorithms

- GIFT-COFB, Photon-Beetle and Grain128-AEAD

Type of algorithm: GIFT-COFB is a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. Photon-Beetle is also a block cipher. Grain128-AEAD is also a block cipher.

Key size: GIFT-COFB uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

Block size: GIFT-COFB operates on blocks of 64 bits. Photon-Beetle operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

Mode of operation: GIFT-COFB uses Output Feedback (OFB) mode, which allows it to be used as a stream cipher and makes it resistant to errors that can occur during transmission. Photon-Beetle does not use any specific mode of operation. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

Resistance to attacks: GIFT-COFB is specifically designed to be resistant to differential cryptanalysis, which is a type of attack that involves analyzing the differences in the output of a cipher when the input is changed slightly. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to these attacks, but it is still considered to be a secure algorithm.

- Xoodyak, Photon-Beetle and Grain128-AEAD

  Type of algorithm: Xoodyak is a hybrid algorithm that combines both symmetric and asymmetric cryptography. Photon-Beetle is a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. Grain128-AEAD is also a block cipher.

  Key size: Xoodyak uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

  Block size: Xoodyak does not operate on fixed-size blocks of data, as it is a hybrid algorithm. Photon-Beetle operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

  Integrity protection: Xoodyak does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for

Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

Resistance to attacks: Xoodyak is specifically designed to be resistant to side-channel attacks, which are attacks that attempt to extract sensitive information by observing the physical characteristics of a device. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to these attacks, but it is still considered to be a secure algorithm.

- ISAP, Photon-Beetle and Grain128-AEAD

Type of algorithm: ISAP is a sponge construction, which is a type of cryptographic primitive that uses a fixed-size internal state and a compression function to transform an input into an output. Photon-Beetle is a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. Grain128-AEAD is also a block cipher.

Key size: ISAP uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

Block size: ISAP does not operate on fixed-size blocks of data, as it is a sponge construction. Photon-Beetle operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

Integrity protection: ISAP does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

Resistance to attacks: ISAP is specifically designed to be resistant to differential cryptanalysis, which is a type of attack that involves analyzing the differences in the output of a cipher when the input is changed slightly, and slide attacks, which are attacks that attempt to extract sensitive information by observing the physical characteristics of a device. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis

and slide attacks. Grain128-AEAD is not specifically designed to be resistant to these attacks.

- Elephant, Photon-Beetle and Grain128-AEAD

  Type of algorithm: Elephant is a stream cipher, which is a type of encryption algorithm that operates on a continuous stream of data. Photon-Beetle is a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. Grain128-AEAD is also a block cipher.

  Key size: Elephant uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

  Block size: Elephant does not operate on fixed-size blocks of data, as it is a stream cipher. Photon-Beetle operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

  Integrity protection: Elephant does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

  Resistance to attacks: Elephant is specifically designed to be resistant to linear cryptanalysis, which is a type of attack that involves analyzing the output of a cipher when the input is changed slightly, and differential cryptanalysis, which is a type of attack that involves analyzing the differences in the output of a cipher when the input is changed slightly. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to any particular type of attack, but it is still considered to be a secure algorithm.

- Sparkle, Photon-Beetle and Grain128-AEAD

  Type of algorithm: Sparkle is a stream cipher, which is a type of encryption algorithm that operates on a continuous stream of data. Photon-Beetle is a block cipher, which is a

type of encryption algorithm that operates on fixed-size blocks of data. Grain128-AEAD is also a block cipher.

Key size: Sparkle uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

Block size: Sparkle does not operate on fixed-size blocks of data, as it is a stream cipher. Photon-Beetle operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

Integrity protection: Sparkle does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

Resistance to attacks: Sparkle is specifically designed to be resistant to certain types of attacks, such as linear cryptanalysis and differential cryptanalysis. Photon-Beetle is specifically designed to be resistant to certain types of attacks, such as differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to these attacks, but it is still considered to be a secure algorithm.

- ASCON, Photon-Beetle and Grain128-AEAD

  Type of algorithm: ASCON is a permutation-based authenticated encryption algorithm. Photon-Beetle is a block cipher, which is a type of encryption algorithm that operates on fixed-size blocks of data. Grain128-AEAD is also a block cipher.

  Key size: ASCON uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

  Block size: ASCON operates on blocks of 128 bits. Photon-Beetle also operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

  Integrity protection: ASCON includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity

protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD also includes support for AEAD.

Resistance to attacks: ASCON is specifically designed to be resistant to certain types of attacks, such as slide attacks and related-key attacks. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to any particular type of attack, but it is still considered to be a secure algorithm.

- TinyJambu, Photon-Beetle and Grain128-AEAD

  Type of algorithm: TinyJambu is a block cipher that uses a substitution-permutation network (SPN) structure. Photon-Beetle is also a block cipher that uses an SPN structure. Grain128-AEAD is also a block cipher.

  Key size: TinyJambu uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

  Block size: TinyJambu operates on blocks of 128 bits. Photon-Beetle also operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

  Integrity protection: TinyJambu does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

  Resistance to attacks: TinyJambu is specifically designed to be resistant to certain types of attacks, such as slide attacks and related-key attacks. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to any particular type of attack, but it is still considered to be a secure algorithm.

- Romulus, Photon-Beetle and Grain128-AEAD

  Type of algorithm: Romulus is a block cipher that uses a substitution-permutation network (SPN) structure. Photon-Beetle is also a block cipher that uses an SPN structure. Grain128-AEAD is also a block cipher.

  Key size: Romulus uses a 128-bit key, Photon-Beetle uses a 128-bit key, and Grain128-AEAD uses a 128-bit key.

  Block size: Romulus operates on blocks of 128 bits. Photon-Beetle also operates on blocks of 128 bits. Grain128-AEAD operates on blocks of 64 bits.

  Integrity protection: Romulus does not provide any integrity protection. Photon-Beetle does not provide any integrity protection. Grain128-AEAD includes support for Authenticated Encryption with Associated Data (AEAD), which allows it to provide both confidentiality and integrity protection.

  Resistance to attacks: Romulus is specifically designed to be resistant to certain types of attacks, such as slide attacks and related-key attacks. Photon-Beetle is specifically designed to be resistant to differential cryptanalysis and slide attacks. Grain128-AEAD is not specifically designed to be resistant to any particular type of attack, but it is still considered to be a secure algorithm.

6. Tests

```
bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$ g++ grain128-aead.c
bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$ ./a.out
        Grain 128-AEAD cipher
key :           00000000000000000000000000000000
accum init:     c0207f221660650b
register init:  6a952ae26586136f
pre-output :    a0904140c8621cfe8660c0dec0969e9436f4ace92cf1ebb794663f17ab8341ee
keystream :     c800a52f948b89b85cee6cfd8571f90f
macstream :     0498886e288e86666e292d976a77119a
tag :           aab555c073e67664
DONE!bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$
```

```
DONE!bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$ g++ photon-beetle.c
bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$ ./a.out
        Photon-Beetle light-weight cipher
Plaintext : abdurrahman
Key : 0123456789ABCDEF0123456789ABCDEF
Nonce : 000000000000111111111111
Cipher: 11811721612514212224212510 20, Len: 27
Plaintext: abdurrahman, Len: 11
DONE!
bulut@DESKTOP-260P6D2:/mnt/c/Users/abdur/Masaüstü/Kriptoloji/codes$
```

7. Conclusion

After conducting a thorough investigation of Grain128-AEAD and Photon-Beetle, it is clear that both algorithms have their own strengths and weaknesses. Grain128-AEAD is a lightweight and efficient encryption algorithm that is well-suited for use in resource-constrained environments. It has a small block size and low memory requirements, making it suitable for use in IoT devices and other embedded systems. On the other hand, Photon-Beetle is a more secure and robust algorithm that is designed to resist attacks such as differential cryptanalysis. It has a larger block size and requires more resources, making it less suitable for use in resource-constrained environments.

When comparing Grain128-AEAD, Photon-Beetle, GIFT-COFB, Xoodyak, ISAP, Elephant, Sparkle, ASCON, TinyJambu, and Romulus, it is important to consider the specific use case and requirements of the system in which the algorithm will be implemented. Each of these algorithms has its own unique features and trade-offs, and the most suitable algorithm will depend on the specific needs of the system. For example, GIFT-COFB is a highly efficient encryption algorithm that is well-suited for use in low-power devices, while Xoodyak is a flexible and secure algorithm that is well-suited for use in a wide range of applications. Ultimately, the choice of encryption algorithm will depend on the specific requirements and constraints of the system in which it will be implemented.

8. Resources

- Figure 1: Random Differential Fault Attacks on the Lightweight Authenticated Encryption Stream Cipher Grain-128AEAD. (n.d.). Retrieved from https://www.researchgate.net/figure/An-overview-of-the-building-blocks-in-Grain-128AEAD_fig1_351467789
- Figure 2: Chakraborti, A., Datta, N., Nandi, M., & Yasuda, K. (2018). Beetle Family of Lightweight and Secure Authenticated Encryption Ciphers. IACR Cryptol. ePrint Arch., 2018, 805.
- Figure 3: Cipher Block Chaining. (n.d.). Retrieved from https://upload.wikimedia.org/wikipedia/commons/thumb/8/80/CBC_encryption.svg/1920px-CBC_encryption.svg.png
- Figure 4: Mustafeez, A. Z. (n.d.). What is OFB? Retrieved from https://www.educative.io/edpresso/what-is-ofb
- Figure 5: Hashing vs. Encryption. (n.d.). Retrieved from https://www.okta.com/identity-101/hashing-vs-encryption/