extended AES– eAES 1.0

Revision 1.0f, October 11, 2024

John G. Underhill – john.underhill@protonmail.com

eAES extends the traditional AES by introducing larger key sizes (256-bit and 512-bit), more rounds (22 and 30 respectively), and a secure key expansion based on the SHAKE XOF function for enhanced resistance to known and future cryptographic attacks.

1. Introduction......................................................................................................... 2

2. Protocol Description........................................................................................... 3

3. Mathematical Description................................................................................... 5

3. Internal Functions............................................................................................... 8

4. Performance Considerations.............................................................................. 12

5. Application Scenarios........................................................................................ 14

6. Mathematical Description................................................................................. 16

7. Security Analysis............................................................................................... 20

8. Conclusion......................................................................................................... 26

**1. Introduction**

The Extended Advanced Encryption Standard (eAES) is a cryptographic algorithm designed to extend the security features of the widely-used Advanced Encryption Standard (AES). The primary motivation for developing eAES is to provide enhanced resistance against both classical and quantum computing-based cryptographic attacks. The eAES protocol extends AES by increasing the key sizes and the number of rounds, while also utilizing a secure key expansion mechanism that enhances the complexity of the key schedule.

**Key Features of eAES:**

* Extended key sizes of 256 bits and 512 bits.
* Secure key expansion utilizing a custom SHAKE XOF (Extendable Output Function).
* 22 rounds for 256-bit keys and 30 rounds for 512-bit keys.
* Enhanced protection against linear and differential cryptanalysis, as well as algebraic attacks.
* Designed to maintain compatibility with existing AES implementations for systems that support traditional 128-bit and 256-bit AES modes.

The design philosophy behind eAES is to increase security margins while maintaining high performance levels, making it suitable for high-security applications in a post-quantum cryptographic world. The extended key schedule in eAES significantly increases the complexity of attacks that exploit weaknesses in the AES key expansion process.

As quantum computers become a reality, and a massive shift in the effectiveness of current cryptographic systems is underway, eAES offers a solution to preserving the large investements into the AES cipher in software and hardware. It replaces the differentially weak mixing function in the Rijndael key schedule, and replaces it with a cryptographically strong pseudo-random permutation. This ensures maximum linear and differential resistance, and strengthens AES to a post-quantum level.

The addition of a 512-bit key ensures that the same security guarantees that AES-256 provided over AES-128, are now possible against quantum adversaries. The addition of rounds to the cipher was calculated as 2n the best known attack against AES-256 using a related subkey attack that breaks 11 rounds of the cipher. The 256-bit key in eAES uses 22 rounds, or exactly twice the best known attack. This margin of security of at least twice the best known attack, restores the security margins to AES, and prepares it against future attacks on the cipher.

This document provides a complete specification for eAES, including a detailed description of its internal functions, performance analysis, application scenarios, mathematical description, and a comprehensive security analysis comparing it with standard AES implementations.

**2. Protocol Description**

The Extended Advanced Encryption Standard (eAES) operates similarly to the standard AES protocol but introduces several modifications to enhance its security and resistance against a broader range of cryptographic attacks. This section describes the key design elements of the eAES protocol, including the round structure, key expansion mechanism, and encryption process.

**Note:** There are references to the project name for this cipher in the software, in which it was named ‘RHX’, which stands for Rijndael Hash eXtension, coinciding with the CEX library naming scheme, but the formal name of this cipher and protocol is ‘eAES’.

**2.1 Key Sizes and Round Structure**

eAES supports two key sizes: 256-bit and 512-bit. The number of encryption rounds depends on the key size, with 22 rounds for 256-bit keys and 30 rounds for 512-bit keys.

* **256-bit key**: 22 rounds
* **512-bit key**: 30 rounds

Each round in eAES consists of a series of transformations that ensure diffusion and confusion, including substitution, permutation, and mixing operations.

**2.2 Secure Key Expansion Mechanism**

One of the major enhancements in eAES is the introduction of a secure key expansion mechanism. Unlike the standard AES key expansion, eAES uses the custom SHAKE XOF (Extendable Output Function) to expand the input key into the round keys.

The **rhx\_secure\_expand** function is used for key expansion, which involves the following steps:

* The initial key (either 256-bit or 512-bit) is processed by the SHAKE XOF to derive round keys.
* The XOF ensures that the entropy of the initial key is sufficiently distributed across all the round keys, increasing resistance to key schedule attacks.

This approach to key expansion is designed to resist cryptanalytic techniques that exploit weaknesses in the standard AES key schedule, such as related-key and meet-in-the-middle attacks.

**2.3 Encryption Process**

The encryption process in eAES consists of multiple rounds, depending on the key size, and each round is composed of four main operations:

1. **SubBytes**: A non-linear substitution step where each byte in the state matrix is replaced with a corresponding value from a fixed S-box.
2. **ShiftRows**: A transposition step that cyclically shifts the bytes in each row of the state matrix to the left.
3. **MixColumns**: A linear mixing operation that ensures diffusion by multiplying each column of the state matrix by a fixed polynomial in the Galois field (GF(2^8)).
4. **AddRoundKey**: In this step, the current state matrix is XORed with a portion of the expanded key for that round, produced by the secure key expansion mechanism.

For both 256-bit and 512-bit keys, the encryption process begins with an initial AddRoundKey operation, followed by the specified number of rounds (22 for 256-bit and 30 for 512-bit). A final AddRoundKey operation is performed after the last round, as in standard AES.

**2.4 Decryption Process**

The decryption process in eAES mirrors the encryption process, with the inverse operations applied in reverse order:

1. **InvSubBytes**: The inverse of the SubBytes operation, where each byte is substituted by the inverse S-box value.
2. **InvShiftRows**: The rows of the state matrix are shifted to the right, reversing the ShiftRows step.
3. **InvMixColumns**: The columns of the state matrix are mixed using the inverse matrix from the MixColumns operation.
4. **AddRoundKey**: As in encryption, each round key is XORed with the state matrix, but in reverse order, starting with the last round key and moving backward.

**Chapter 3: Mathematical Description**

This chapter describes the cryptographic functions of the eAES (extended AES) protocol, focusing on the modified encryption and key scheduling processes that differ from standard AES.

**3.1 Overview of Extended AES (eAES)**

The eAES protocol extends the Advanced Encryption Standard (AES) by introducing two key lengths (256-bit and 512-bit) and increasing the number of rounds to 22 and 30 respectively. In addition, eAES includes a secure key expansion function to generate round keys using a SHAKE XOF-based key derivation function.

**3.2 Key Expansion in eAES**

The standard AES key schedule is replaced in eAES by a more secure key expansion mechanism. This process is necessary for the 256-bit and 512-bit keys, where the number of rounds exceeds the 14 rounds present in standard AES.

1. The input key (either 256-bit or 512-bit) is expanded using the SHAKE XOF (extendable-output function).
2. The SHAKE XOF outputs a sequence of round keys, ensuring higher entropy than the standard AES key expansion.
3. The key expansion uses a tweakable parameter to introduce additional security, making it resistant to known attacks on the AES key schedule.

Mathematically, if *K* is the input key and SHAKE\_XOF is the XOF function, the round key expansion can be represented as:

*RoundKey\_i = SHAKE\_XOF(K || Tweak, i)*

where *Tweak* is the optional tweak parameter, and *i* is the round index.

**3.3 Encryption Process**

The encryption process of eAES follows the traditional AES structure, but with extended rounds and the modified key expansion. The steps include:

1. **Initial AddRoundKey** – The input data *P* is XORed with the first round key derived from the secure key expansion:

*State = P XOR RoundKey\_0*

1. **SubBytes** – A non-linear substitution step where each byte of the state *S* is replaced with a corresponding value from a predefined S-box:

*S'[i][j] = SBox(S[i][j])*

1. **ShiftRows** – A permutation step where the rows of the state matrix are shifted cyclically. The first row remains unchanged, while the second, third, and fourth rows are shifted by 1, 2, and 3 bytes respectively:

ShiftRows(State)

1. **MixColumns** – A linear mixing step where each column of the state matrix is multiplied by a fixed matrix *M* in the finite field GF(28):

*State' = M × State*

This ensures the diffusion of the plaintext bits across the state matrix.

1. **AddRoundKey** – The state is XORed with the current round key generated from the key schedule:

*State = State XOR RoundKey\_i*

These steps are repeated for 22 rounds in the case of the 256-bit key and for 30 rounds with the 512-bit key.

**3.4 Secure Key Expansion Function**

The secure key expansion in eAES is designed to counter algebraic, differential, and linear attacks that exploit weaknesses in the standard AES key schedule. The expansion process involves:

1. **Input key**: Either 256 or 512 bits.
2. **SHAKE XOF derivation**: The key is fed into the SHAKE XOF to produce round keys.
3. **Tweakable expansion**: The Keccak SHAKE function is customized using a tweak parameter, which allows additional variation in the key expansion process, making it highly resistant to related-key attacks.

This secure expansion increases the resilience of eAES to modern cryptanalytic techniques, particularly those targeting weaknesses in the original AES key schedule.

**4. Cryptographic Details**

In this section, we dive into the cryptographic operations used in eAES. We explain how the secure key schedule operates and break down the encryption and decryption processes. Each cryptographic step builds upon well-known concepts but includes eAES-specific modifications that enhance security.

**4.1 Key Schedule**

The eAES secure key schedule is derived from a custom SHAKE XOF function, ensuring that each round key has a high level of randomness and resistance to known attacks. This mechanism prevents weaknesses present in traditional AES key scheduling by making the round keys highly resistant to differential and algebraic attacks.

The key sizes supported are:

* 256-bit key using 22 rounds
* 512-bit key using 30 rounds

The function works as follows:

1. **Input Key**: The input key, denoted as *K*, is expanded using the custom SHAKE XOF function to derive the round keys. For a 256-bit key, this results in 22 distinct round keys, and for a 512-bit key, 30 round keys are generated.
2. **SHAKE XOF Expansion**: The custom SHAKE XOF function generates output that expands *K* into the required number of round keys. This eliminates linear relations between round keys, mitigating certain forms of cryptographic attacks.

Let *RK₁, RK₂, ..., RKₙ* denote the round keys derived from the key expansion process, where *n* is 22 or 30 depending on the key size. Each round key is applied during the encryption process.

**4.2 Encryption Process**

The encryption process follows a modified AES-like structure but includes key eAES extensions that increase its security margins.

For each round *i*, the following steps are performed:

1. **AddRoundKey**: The current state matrix *S* is XORed with the round key *RKᵢ*. Let *S' = S ⊕ RKᵢ*.
2. **SubBytes**: Each byte in the state matrix *S'* is substituted using a nonlinear S-box transformation, similar to AES, but with additional nonlinearity introduced for added security.
3. **ShiftRows**: The rows of the state matrix are shifted as follows:
   * The first row remains unchanged.
   * The second row is shifted 1 byte to the left.
   * The third row is shifted 2 bytes to the left.
   * The fourth row is shifted 3 bytes to the left.

This step ensures diffusion across the rows of the state matrix.

1. **MixColumns**: Each column of the state matrix is transformed using a linear mixing operation over the Galois Field GF(2⁸), providing further diffusion across the columns. This is identical to the MixColumns operation used in standard AES.
2. **Repeat for 22 or 30 Rounds**: The above steps are repeated for each round, using the corresponding round key *RKᵢ* from the key schedule.

After the final round, the state matrix *S* contains the encrypted ciphertext, denoted *C*.

**4.3 Decryption Process**

Decryption in eAES mirrors the encryption process but reverses the operations. The steps are applied in reverse order, starting from the final round and working backward to recover the plaintext message *M*.

For each round *i* (starting from the last round and moving to the first):

1. **Inverse MixColumns**: The inverse of the MixColumns operation is applied to the state matrix.
2. **Inverse ShiftRows**: The rows of the state matrix are shifted back to their original positions:
   * The first row remains unchanged.
   * The second row is shifted 1 byte to the right.
   * The third row is shifted 2 bytes to the right.
   * The fourth row is shifted 3 bytes to the right.
3. **Inverse SubBytes**: Each byte in the state matrix is substituted using the inverse of the nonlinear S-box transformation.
4. **AddRoundKey**: The current state matrix is XORed with the round key *RKᵢ*. Let *S' = S ⊕ RKᵢ*.
5. **Repeat for 22 or 30 Rounds**: The above steps are repeated for each round in reverse order, using the corresponding round key *RKᵢ* from the key schedule.

After completing all rounds, the final state matrix contains the decrypted plaintext message *M*.

**4.4 Summary of Cryptographic Operations**

eAES utilizes a secure key schedule that ensures strong resistance to cryptanalytic attacks, combined with a modified AES encryption structure. The use of the Keccak SHAKE XOF for key expansion provides additional security, making it resistant to known attacks like differential cryptanalysis and algebraic attacks. With 22 rounds for 256-bit keys and 30 rounds for 512-bit keys, eAES achieves a high level of security, particularly against both classical and quantum cryptographic attacks.

The increased number of rounds, combined with the enhanced key schedule, ensures that eAES is resistant to various advanced attacks, including:

* **Differential Cryptanalysis**: The added nonlinearity in the key schedule and increased number of rounds make differential cryptanalysis extremely difficult to apply effectively.
* **Linear Cryptanalysis**: The diffusion operations and nonlinear transformations of eAES ensure that linear cryptanalysis attacks face significantly greater resistance.
* **Algebraic Attacks**: The secure key schedule ensures that algebraic attacks are less feasible, especially with the introduction of the SHAKE XOF function for key expansion.

In conclusion, the cryptographic framework of eAES extends the security of standard AES by leveraging advanced key expansion techniques, increased rounds, and post-quantum resistance in its key scheduling mechanism.

**5. Security Analysis**

In this section, we will analyze the security of eAES compared to standard AES. We will discuss the resistance of eAES to various types of cryptanalytic attacks, including differential, linear, algebraic, and quantum-related attacks. Additionally, the security benefits of the enhanced key schedule and increased number of rounds will be explored.

**5.1 Resistance to Differential and Linear Cryptanalysis**

Differential and linear cryptanalysis are two of the most common attacks against block ciphers. These attacks exploit patterns in how differences in the input (in the case of differential cryptanalysis) or linear approximations of the cipher (in the case of linear cryptanalysis) propagate through the rounds of the cipher.

**Differential Cryptanalysis:**

The additional rounds of eAES, combined with its nonlinear operations, make it significantly more resistant to differential attacks compared to standard AES. The key schedule of eAES, which is generated using a secure permutation function, introduces additional randomness into the round keys. This reduces the likelihood of differential characteristics being exploited across multiple rounds.

Let *ΔP* denote the difference between two plaintexts and *ΔC* the difference between the corresponding ciphertexts. Differential cryptanalysis attempts to find high-probability paths where *ΔP → ΔC*. In eAES, the increased number of rounds (22 for 256-bit keys, 30 for 512-bit keys) increases diffusion, making such paths exponentially harder to find.

**Linear Cryptanalysis:**

Linear cryptanalysis attempts to find linear relationships between plaintexts, ciphertexts, and the round keys. Due to the secure key schedule and the increased number of rounds in eAES, the cipher provides a high degree of diffusion and nonlinearity, making it difficult to apply linear approximations successfully.

If an attacker tries to find a linear approximation *P ⊕ C* ≈ *K* (where *P* is the plaintext, *C* is the ciphertext, and *K* is the key), the extra rounds and nonlinear transformations ensure that such linear equations do not hold with significant probability.

**5.2 Resistance to Algebraic Attacks**

Algebraic attacks attempt to model the encryption algorithm as a system of polynomial equations over a finite field. For AES, these equations are typically constructed based on the SubBytes, ShiftRows, and MixColumns operations.

The secure key schedule of eAES, which uses a secure permutation to generate round keys, introduces additional complexity into the polynomial system that an attacker would need to solve. The use of SHAKE XOF as a key expansion mechanism ensures that the round keys are highly diffused and difficult to predict, making it much harder to express the encryption algorithm in a tractable algebraic form.

Furthermore, the increased number of rounds in eAES amplifies the complexity of the algebraic system. With 22 rounds for 256-bit keys and 30 rounds for 512-bit keys, the cipher ensures that algebraic attacks require solving systems of equations that are computationally infeasible.

**5.3 Resistance to Key-Recovery Attacks**

Standard AES has been subject to key-recovery attacks, particularly in reduced-round versions of the cipher. The design of eAES, with its secure key schedule and increased number of rounds, addresses these concerns.

The SHAKE XOF function ensures that the round keys are not linearly related, unlike in standard AES where certain weaknesses in the key schedule have been exploited to perform related-key attacks. This improvement in the key schedule, combined with the increased number of rounds, ensures that key-recovery attacks against eAES are far less likely to succeed.

In particular, for a 512-bit key, the key schedule provides a very high level of entropy across all 30 rounds, making related-key and other key-recovery attacks highly impractical.

**5.4 Post-Quantum Security**

One of the main motivations behind the design of eAES is its resistance to attacks by quantum computers. Standard AES is vulnerable to Grover’s algorithm, which can reduce the effective key size of the cipher from *n* bits to *n/2* bits. For AES-128, this implies an effective key size of 64 bits, which is insufficient against a quantum adversary.

eAES mitigates this vulnerability by offering larger key sizes (256-bit and 512-bit), which are more resistant to quantum attacks. In particular:

* For the 256-bit key, the effective key size under Grover’s algorithm would be reduced to 128 bits, which is still considered secure.
* For the 512-bit key, the effective key size would be reduced to 256 bits, which provides an exceptionally high level of security even against quantum adversaries.

By extending the key size and rounds, eAES ensures that it can withstand both classical and quantum adversaries.

**5.5 Comparison to Standard AES**

Standard AES has a 128-bit and 256-bit key size with 10 and 14 rounds respectively, and its security has been well studied for over two decades. However, several attacks have been developed against reduced-round versions of AES, and there are concerns regarding its post-quantum resilience.

In contrast, eAES offers the following improvements:

* **Larger Key Sizes**: eAES supports key sizes of 256 bits and 512 bits, making it far more resistant to brute-force attacks, even in the context of quantum computing.
* **Increased Number of Rounds**: With 22 rounds for the 256-bit key and 30 rounds for the 512-bit key, eAES provides a higher level of security through more diffusion and greater complexity.
* **Enhanced Key Schedule**: The use of SHAKE XOF in the key schedule ensures that round keys are highly random and resistant to related-key and other attacks.
* **Post-Quantum Security**: eAES is specifically designed to resist quantum attacks, which would weaken standard AES’s security guarantees.

**5.6 Security Conclusion**

eAES improves upon standard AES by addressing several known vulnerabilities and introducing measures that provide resilience against both classical and quantum cryptographic attacks. The secure key schedule and increased number of rounds ensure that eAES provides a high level of security across various threat models. Its post-quantum design makes it a future-proof encryption standard, suitable for environments where quantum adversaries may become a reality.

**6. Conclusion**

The eAES (Extended Advanced Encryption Standard) is a significant enhancement over the standard AES cipher. By offering larger key sizes, a more secure key schedule, and additional rounds, eAES addresses several vulnerabilities inherent in the original AES design. These improvements make it highly resistant to various forms of cryptanalysis, including differential, linear, and algebraic attacks.

**6.1 Enhanced Key Schedule**

One of the most critical improvements in eAES is the adoption of a cryptographically-secure permutation as a key schedule; the Keccak SHAKE XOF function, which generates round keys with a greater diffusion of entropy than the standard AES key schedule. This mitigates the risks associated with related-key and other key-recovery attacks, which have been demonstrated in some forms against AES.

**6.2 Increased Resistance to Cryptanalytic Attacks**

The increase in the number of rounds for both key sizes (22 rounds for 256-bit keys and 30 rounds for 512-bit keys) greatly enhances the cipher's security. More rounds mean more diffusion and nonlinearity, making differential and linear attacks exponentially more difficult to execute successfully.

In addition, the enhanced key schedule resists algebraic attacks, making it infeasible to construct solvable systems of equations based on the cipher's operations. The higher number of rounds also increases security against meet-in-the-middle and other multi-round attacks.

**6.3 Post-Quantum Security**

One of the main motivations for designing eAES is its ability to resist quantum attacks. While standard AES is vulnerable to quantum algorithms like Grover's, which reduce the effective key size by half, eAES offers key sizes large enough to remain secure even in the face of such attacks. The 256-bit and 512-bit key sizes ensure that, even under quantum threat, the effective key lengths (128-bit and 256-bit respectively) remain secure.

**6.4 Practical Considerations**

Despite its enhanced security, eAES remains practical for a wide range of applications. While the increased number of rounds may lead to slightly slower encryption and decryption times compared to standard AES, the performance overhead is minimal compared to the added security benefits. For environments where post-quantum security and resistance to modern cryptanalysis are paramount, eAES is a highly suitable choice.

**6.5 Future Directions**

As quantum computing continues to evolve, cryptographic protocols like eAES will play a critical role in securing sensitive data against future adversaries. Future research into the optimization of eAES, as well as continued cryptanalysis efforts, will ensure that it remains a secure and efficient encryption standard for years to come.

**6.6 Final Thoughts**

The eAES protocol represents a forward-thinking approach to cryptographic security. By addressing current vulnerabilities and anticipating future quantum threats, it provides an encryption standard that is both secure and adaptable. Its design strikes a balance between increased security and operational efficiency, making it a robust solution for modern cryptographic needs.