Designing authenticated encryption modes of operation

Wojciech Oszywa, Rafał Gliwa
Military Communication Institute, 05-130 Zegrze, Poland

Abstract: A block cipher is a function which maps n-bit plaintext block to n-bit ciphertext block using k-bit key K. For messages exceeding n bits, it is necessary to use a block cipher in an appropriate mode of operation. A mode of operation of a block cipher is an algorithm, which specifies how one has to apply an n-bit block cipher to achieve the security goal. In this paper we focus on authenticated encryption modes, which simultaneously provide both privacy and authenticity of the message. We introduce two kinds of authenticated encryption modes: single-pass modes and two-pass combined modes. We take into consideration a few most important properties, i.e. error propagation, synchronization, parallelizability, keying material requirements and pre-processing capability. We analyze what are advantages and disadvantages of using Cipher Block Chaining and Counter encryption mode of operation. How to achieve the authentication: CBC-MACs versus universal hash-based MACs. We consider, in what order these modes should be applied to a message in order to achieve the authenticated encryption. Finally we mention the meaning of a provable security in proving the security of authenticated encryption modes.

Keywords: block cipher, mode of operation, authentication, encryption, authenticated encryption.

A block cipher is a flexible building block – it can be used for encryption and authenticated encryption, to construct MAC algorithms and hash functions [1]. A block cipher is a function which maps n-bit plaintext block to n-bit ciphertext block using k-bit key K. It is obvious that applications need to protect the confidentiality of strings of arbitrary length. Therefore, for messages exceeding n bits, it is necessary to use a block cipher in an appropriate mode of operation. A mode of operation of a block cipher is an algorithm which specifies how one has to apply an n-bit block cipher to achieve the security goal. Depending on the security goal, the National Institute of Standards and Technology (NIST) distinguishes four kinds of block cipher modes of operation. These are: encryption modes, authentication modes, authenticated encryption modes and other modes. Let's focus on the authenticated encryption issue. In general, the term authenticated encryption scheme (AE scheme) is used to refer to a shared-key based transform, whose goal is to provide both privacy and authenticity of the encapsulated data [2]. By privacy we mean keeping information secret from all but those, who are authorized to see
it. By authenticity we mean corroborating the source of information (also known as data origin authentication), including data integrity, i.e. ensuring that information has not been altered by unauthorized or unknown means. In the authenticated encryption scheme the encryption process applied by the sender takes the key and a plaintext to return a ciphertext, while the decryption process applied by the receiver takes the same key and a ciphertext to return either a plaintext or a special symbol indicating that it considers the ciphertext invalid or unauthentic. In many application settings we wish not only to encrypt and authenticate the message, but we wish also to include auxiliary data, which should be authenticated but left unencrypted. An example might be a network packet where the payload should be encrypted and authenticated, but the header should be only authenticated (not encrypted). The reason is that routers must be able to read the headers of packets in order to know how to properly route them. This need caused some designers of AE schemes allow “associated data” to be included as input to their schemes. Such schemes have been named the authenticated encryption with associated data (AEAD) schemes [3].

![Diagram: Encryption process of AEAD scheme](image)

There are many possible ways to design authenticated encryption schemes: generic compositions, single-pass modes, two-pass combined modes and authenticated encryption primitives. A traditional way of achieving both authenticity and privacy, called a generic composition [1][4], was to simply find two algorithms yielding each of these properties and then use their combination on a message. Generic compositions are relatively slow, because they are two-pass constructions, i.e. every block of message is processed twice, once by the encryption algorithm and then by the authentication algorithm. Moreover, they require the use of two independent keys, one for encryption and the other for authentication. An alternative to generic compositions are dedicated block cipher modes. When NIST approved
in September 2000 the Advanced Encryption Standard (AES), the new block cipher modes attracted a lot of attention. There was a need to update long-standing modes of operation and an opportunity to consider the development of new ones. From that time a number of new modes have appeared, including constructions which achieve simultaneously both privacy and authenticity, in one or two steps. Single-pass modes [5][6][7] provide authenticated encryption just after single processing of a message. As all of single-pass modes have been protected by patents, which causes serious restrictions in their usage, two-pass combined modes have appeared. They process data in two steps, like generic compositions, but unlike generic compositions they use not two, but only one key for both encryption and authentication. The first to be developed was the CCM [8], then EAX [9] tried to solve some of the CCM problems, then CWC [10] has been developed to improve EAX and finally GCM [11] has been designed. An aspect of great significance for the security of AE schemes is the correct handling with a special value, called nonce (‘number used once’), because the misuse of nonce would generally lead to the complete collapse of systems based on these AE modes. The problem of nonce misuse has been settled by the introduction of deterministic authenticated encryption (DAE) [12][13]. To present a complete view on AE schemes, one more approach to achieving authenticated encryption must be mentioned – that is certain stream ciphers, which, in general, use a plaintext for the generation of a keystream [14][15]. It is the least popular approach and its development is not in progress.

The National Institute of Standards and Technology (NIST) is in the process of recommending modes in a series of special publications. Currently, there are eight approved block cipher modes, including two authenticated encryption modes: CCM (Counter with CBC-MAC), specified in the NIST Special Publications 800-38C [8] and GCM (Galois/ Counter mode), recommended in the NIST Special Publications 800-38D [11]. The NIST is still open for different submissions. The submission of mode of operation should include the following six items:

- Cover Sheet,
- Mode Specification,
- Summary of Properties,
- Test Vectors,
- Performance Estimates,
- Intellectual Property Statements.

From the designer’s point of view, we are especially interested in the Summary of Properties item. It specifies main properties, which have to be taken into account by the designers of block cipher mode of operation. These are:

- Security Function (encryption, authentication, authenticated encryption, hashing, pseudorandom bit generation, etc.),
- Error Propagation (e.g., none, m bits, n blocks, infinite),
- Synchronization,
- Parallelizability (e.g., sequential, interleaved, fully parallelizable),
- Keying Material Requirements (e.g., 1 key, 2 keys),
- Counter/IV/Nonce Requirements,
- Memory Requirements,
- Pre-processing Capability,
- Message Length Requirements (e.g., arbitrary length, padding necessary),
- Ciphertext Expansion (e.g., none, \( m \) bits, \( n \) blocks),
- Other Characteristics.

Designing AE mode one should pay particular attention to following matters.

Figure 2. Authenticated encryption schemes
Security Function

An authenticated encryption mode of operation must provide both privacy and authenticity of the message. The designer should consider also the possibility to include auxiliary data, which should be authenticated but left unencrypted. This means that the encapsulation algorithm, on the input of a pair of auxiliary data and message (A, M), and some nonce N, encapsulates (A, M) in a way that protects the privacy of M and the integrity of both A and M.

Error Propagation (e.g. none, m bits, n blocks, infinite)

Decryption of ciphertext containing bit errors may result in various effects on the recovered plaintext, including the propagation of errors to subsequent plaintext blocks. Different error characteristics are acceptable in various applications. Error propagation is the property that an error in the i-th ciphertext block is inherited by the i-th and all subsequent plaintext blocks. Modes with error propagation are well-suited for situations where errors in transmissions are either unlikely to happen or taken care of by noncryptographic means like error-correcting codes, and situations, where an erroneous data transmission is dealt with by retransmission. A special kind of error propagation is called error recovery. It is the property that an error in the i-th ciphertext is inherited by only a few plaintext blocks after which the mode resynchronizes. Modes with error recovery are suited for situations where a retransmission after an erroneous data transmission is not possible or regarded too expensive.

Synchronization

Synchronization of the AE mode is based on the nonce. The nonce must either be sent with the ciphertext or the receiver must know how to derive the nonce on its own.

Parallelizability

Parallelizability is an significant advantage for hardware implementations. In order to make the AE mode parallel, one should consider using the counter mode CTR for encryption and the Carter-Wegman scheme for authentication. Parallelizability requirement excludes MAC schemes based on hash function, because they work step by step.

Keying Material Requirements

Authenticated encryption mode should operate single-key and the mode's key is the same as the underlying block cipher key. In practice, AE modes do internally use two keys: the main block cipher key and a hash key, which is derived using the block cipher key. Implementers can decide whether to store the derived hash key in memory or whether to re-derive it as needed.
Counter/IV/Nonce Requirements

To promote interoperability and simplicity of design it is recommended to fix the length of a counter/IV/nonce, for e.g. 128 bits. The AE scheme is considered secure as long as one does not query the encryption algorithm twice with the same nonce. The mode specification does not include nonce management, it lies in the users' hands.

Memory Requirements

The memory requirements are basically those of the underlying block cipher. To minimize memory requirements it is worth to provide privacy with block cipher in the counter mode CTR. Then, it is enough to implement a forward function of a block cipher algorithm – an inverse cipher function is not used in CTR mode. Besides, using only one key gives additional savings.

Pre-processing Capability

Pre-processing capability significantly improves the efficiency of authenticated encryption. In the case of the CTR mode for encryption, the underlying CTR mode keystream can be pre-computed. By preprocessing associated data, one can reduce computation time if the associated data remains static or changes only infrequently.

Message Length Requirements

If the length of the last plaintext block is smaller than the block size, padding is or is not necessary, depending on used mechanism. To pad is to append a number of bits to that block, so that it becomes a multiple of or equal to the relevant block size. In the case the sender and the receiver agree on the used padding, the padding bits do not need to be transmitted. The most commonly used padding method is to append a single ‘1’ bit to the data string and then to pad the resulting string by as few ‘0’ bits, possibly none, as are necessary to complete the final block.

Ciphertext Expansion (e.g. none, m bits, n blocks)

The ciphertext expansion is the minimum number of bits possible, while still providing a t-bit tag T. That is, on the input of a pair (A, M), a nonce N, and a key K, mode outputs a ciphertext C with length |C| = |M| + t.

Other Characteristics

A number of options and interoperability: it is recommended to use a minimum number of options i.e. only the choice of the underlying block cipher and the tag length. Having fewer options makes interoperability easier.

On-line: it is worth considering the possibility of processing data as it arrives, rather than waiting for the entire message to be buffered before beginning the en-
cryption processes. This may be advantageous when encrypting streaming data sources. (Note, however, that the decryptor should still buffer the entire message and check the tag before revealing the plaintext and associated data.) To achieve on-line mode, the length of a message cannot be a necessary parameter to begin its processing.

What mostly affects the above mentioned parameters is a choice of the underlying encryption mode and the authentication scheme. Two of the most popular encryption modes are: Cipher Block Chaining mode (CBC) [16] and Counter mode (CTR) [16]. The CBC mode is a confidentiality mode whose encryption process features the combining ('chaining') of the plaintext blocks with the previous ciphertext blocks. The CBC mode requires an initialization vector IV to combine with the first plaintext block. The Counter (CTR) mode is a confidentiality mode that features the application of the forward cipher function to a set of input blocks, called counters, to produce a sequence of output blocks that are exclusive-ORed with the plaintext to produce the ciphertext, and vice versa. The sequence of counters must have the property that each block in the sequence is different from every other block. This condition is not restricted to a single message: across all of the messages that are encrypted under a given key, all of the counters must be distinct.

![Figure 3. The CBC mode](image-url)
These are the effects of applying one of the modes: CBC or CTR.

**Error propagation:**
- CBC: a bit error(s) at any position of ciphertext block affect(s) the decipherment of this block to the random value block; the recovered next block plaintext has bit errors precisely at the same position as the ciphertext block.
- CTR: the bit error(s) in the decrypted ciphertext block occur(s) in the same bit position(s) as in the ciphertext block; the other bit positions are not affected.

**Synchronization:**
- CBC: a bit error(s) in ciphertext block affect(s) two decrypted blocks, but the third and consecutive blocks are decrypted correctly.
- CTR: a bit error(s) in ciphertext block affect(s) only one decrypted block, the second and consecutive blocks are decrypted correctly.

Neither CBC, nor CTR mode can self-synchronize after the loss of ciphertext bits in a block.
Parallelizability:

- CBC: in the CBC encryption, the input block to each forward cipher operation (except the first one) depends on the result of the previous forward cipher operation, so the forward cipher operations cannot be performed in parallel. In the CBC decryption, however, the input blocks for the inverse cipher function, i.e. the ciphertext blocks, are immediately available, so that multiple inverse cipher operations can be performed in parallel.
- CTR: in both CTR encryption and the CTR decryption, the forward cipher functions can be performed in parallel; similarly, the plaintext block that corresponds to any particular ciphertext block can be recovered independently from the other plaintext blocks, if the corresponding counter block can be determined.

Preprocessing:

- CBC: because of chaining mechanism, no preprocessing is possible.
- CTR: possible preprocessing of counter values. The forward cipher functions can be applied to the counters prior to the availability of the plaintext or ciphertext data.

Considering an authentication scheme we have a choice of: MAC constructions based on block ciphers, MAC constructions based on hash functions and a universal hash-based MACs [17]. MAC based on block ciphers are mostly based on the cipher block chaining mode. General problems with CBC-MAC are:

- lack of parallel processing: the inherited CBC chaining dependency leads to a general disability of the CBC-MAC to process messages in parallel;
- no possibility for preprocessing: because the message blocks are the input for the block cipher;
- insecure for arbitrary long messages i.e. if the input is not of a fixed length.

MACs based on hash functions form a MAC from hash functions by including a secret key as part of the hash input. Unfortunately HMAC type constructions make impossible parallel computations, because they process the message iteratively.

A universal hashing was introduced by Carter and Wegman and it can be used for message authentication, for example by hashing a message with a function drawn from a universal hash function family and encrypting the output of the hash function with a block cipher. The encrypted hash output serves as MAC result. The universal hashing approach is more efficient than the previous approaches. The speed of such MACs is comparable and better than other MAC algorithms. The universal hashing paradigm has reduced the efficiency problem of fast authentication to fast universal hashing. The recent findings of internal collisions on some hash functions, such as SHA seem to endanger as well the security of the hash functions and subsequently the hash-based
MACs utilizing them. Thus, the universal hashing approaches seem to remain at present most perspective variants for the construction of secure and efficient MACs.

It is interesting also to analyze the order of processing the message by the AE scheme. It affects the verification of message by the receiver. Fast verification is achieved in the case of first verifying and then decrypting the message. This order of processing in decryption allows for an immediate check for malicious interventions and therefore saves the additional cost of unnecessary decryption. The construction “encrypt then MAC” is hence most efficient in this respect. It allows the receiver of a bogus message to discard it before decryption. But the disadvantage is that the ciphertext, not the plaintext, is authenticated.

A significant advantage of the AE mode is the use of only a single key to generate the MAC and encrypt is. It requires less memory storage space and no additional key management, in comparison to the usage of one key by generic compositions. In practice, AE modes do internally use two keys: the main block cipher key and a hash key, which is derived using the block cipher key.

A crucial aspect of the AE mode is that its security depends on the correct usage of a nonce ("a number used once", within some established context). If there was no nonce there would only be one ciphertext for a given plaintext and a key, and this means that the scheme would necessarily leak information. A nonce is an input to the encryption process and the same nonce is required for the decryption operation. The nonce does not have to be random or secret or unpredictable. A counter makes a perfectly good nonce, as does a random value. The nonce, however, needs to be handled with great care, because the misuse of a nonce (i.e. repeating the same value) would generally lead to the complete collapse of systems based on these AE modes. This is due to the fact that the security designs have no concern in what happens when the nonce assumption becomes no longer true. It is the user's obligation to ensure that nonces do not repeat within a session. The problem of a nonce misuse has been settled by the introduction of deterministic authenticated encryption (DAE). The DAE mode provides a deterministic, stateless algorithm that produces a ciphertext from the pair of a header and a message [12][13]. A deterministic, nonce-less form of authenticated encryption has been used to protect the transportation of cryptographic keys, generally referred to as "Key Wrapping".

With new objects it is often hard to know how much trust to put in their security. The problem with a protocol design is that a poorly designed protocol can be insecure even though the underlying atomic primitive is good. An example is ECB (Electronic Code-Book) mode encryption with a block cipher. It is not a good encryption scheme because partial information about the plaintext leaks. Yet this is no fault of the underlying atomic primitive (e.g. AES). Rather, the atomic primitive was misused. In the past, to be sure of security of a new algorithm, it was necessary to give enough long time for good cryptanalysts to examine it. This approach was changing along with the introduction of the provable security theory [18]. If the object
is a “primitive,” such as a block cipher, no proof of security is possible, so instead we hope for security once we have shown that no known attacks (e.g. differential cryptanalysis) seem to work. However, for algorithms which are built on top of these primitives, called “modes”, we can prove some things about their security; namely that they are as secure as the primitives which underlie them.

**Conclusion**

Combining encryption and authentication to get practical security is a difficult task. It is very easy to accidentally combine secure encryption scheme with secure MAC and still get insecure authenticated encryption scheme. This problem does not occur in the case of dedicated authenticated encryption modes, because they clearly specify how to achieve both privacy and authenticity and there is no longer the risk of someone accidentally combing a privacy component with an authenticity component in an insecure way.

**REFERENCES**


