A Mode of Operation with Partial Encryption and Message Integrity (PEMI)

Philip Hawkes[1] and Gregory G. Rose[1]

QUALCOMM International (Australia)
Level 3, 230 Victoria Road, Gladesville NSW 2111 Australia
{phawkes,ggr}@qualcomm.com

Abstract. At the recent AES Modes of Operation Conference, several modes of operation were proposed for using a block cipher to provide both confidentiality and authentication. These modes require only a little more work than the cost of encryption alone, and come with proofs of security. However, these modes require the entire message to be sent in encrypted form. This can cause problems in situations where some of the message neeeds to be sent in plaintext while still being authenticated. This paper describes a simple variation that allows any choice of message blocks to be sent in plaintext form rather than in encrypted form. This mode, Partial Encryption with Message Integrity (PEMI), is shown to be secure for message integrity and message secrecy.

1 Introduction

In 2000 Jutla [2, 3] proposed new modes of operation for symmetric block ciphers: Integrity Aware Cipher Bock Chaining (IACBC) mode and Integrity-Aware Parallelizable (IAPM) Mode. Provided the underlying block cipher is secure, the modes are proven to provide secure encryption and secure message authentication. Other modes with the same properties have since been proposed: XCBC mode, proposed by Gligor and Donescu [1]; and OCB mode, proposed by Rogaway, Bellare, Black and Krovetz [5]. These modes (IACBC, IAPM, XCBC and OCB) share a common dilemma: the entire message must be sent in encrypted form. In many situations, this is not an issue. However, some protocols require part of the message (such as an IP address) to be sent as plaintext, and it would be desirable for this plaintext portion to be authenticated along with the remainder of the message. This is not possible with the aforementioned modes. This paper presents a mode whereby part of the message can be sent as plaintext, and this plaintext portion to be authenticated along with the remainder of the message. The mode is called *Partial Encryption with Message Integrity* (PEMI) mode.

The paper is arranged as follows. Section 2 contains a description of IAPM and the new mode. A short discussion on the efficiency of PEMI follows in Section 2.2. A proof of message integrity is provided in Section 3. Section 3.1 discusses partial block encryption, where part of a block is sent encrypted and the remaining part of the block is sent unencrypted.

2 Description

Let E_K denote encryption of an l-bit block by a symmetric block cipher using the key K, and let D_K denote decryption by the symmetric block cipher using K. Suppose the message is m blocks in length: $P = P[1], \ldots, P[m]$, and the sender and receiver have agreed on two keys $K0, K1$. The XOR version of IAPM mode is summarized below.[1] This mode requires the generation of a sequence $S[0], \ldots, S[m+1]$ from the initial value IV using the key $K0$. This sequence must have the property that $S[i] \oplus S[j]$ is uniformly distributed for $0 \leq i < j \leq m+1$. The method proposed by Jutla [3] uses only only $t = \lceil \log_2(m+3) \rceil$ encryptions. This method is given in the Appendix. IAPM encryption, decryption and message integrity verification are computed as follows.

IAPM: Encryption with Authentication

Step 1 Choose a random l-bit initial value IV, set $C[0] = IV$.
Step 2 Generate $S[0], \ldots, S[m+1]$ from IV and $K0$.
Step 3 For $1 \leq i \leq m$, compute $C[i] = S[i] \oplus E_{K1}(P[i] \oplus S[i])$.
Step 4 Form $P[i+1] = P[1] \oplus \cdots \oplus P[m]$, and compute,

$$C[m+1] = S[0] \oplus E_{K1}(\mathit{checksum} \oplus S[m+1]).$$

Transmit $C = (C[0], \ldots, C[m+1])$.

IAPM: Decryption with Message Integrity Verification

Step 1 Extract $IV = C[0]$.
Step 2 Generate $S[0], \ldots, S[m+1]$ from IV and $K0$.
Step 3 For $1 \leq i \leq m$ compute $P[i] = S[i] \oplus D_{K1}(C[i] \oplus S[i])$.
Step 4 Compute $P[m+1] = S[m+1] \oplus D_{K1}(C[m+1] \oplus S[0])$.
Step 5 If $P[m+1] = P[0] \oplus \cdots \oplus P[m]$, then the message integrity is verified, and the message is $P = P[1], \ldots, P[m]$.

[Note to referees: We can include a Figure describing IAPM here, but we are currently pressed for time.]

As explained in Section 1, There is demand for a mode in which part of the message is sent in non-encrypted form. The mode proposed below, *Partial Encryption with Message Integrity* (PEMI), is a variation on IAPM. Suppose the message is m blocks in length: $P = P[1], \ldots, P[m]$, and the sender and receiver have agreed on two keys $K0, K1$. The sender and receiver also agree on a set $U \subset \{1, \ldots, m\}$ of block indices for which the plaintext block $P[i]$ will be transmitted rather than the ciphertext $C[i]$ when $i \in U$.

[1] There is a slight discrepancy between the description of IAPM in [2, 3]. This description follows [3].

PEMI: Encryption with Authentication

Step 1 Choose a random l-bit value IV, set $C[0] = IV$.
Step 2 Generate $S[0], \ldots, S[m+1]$ from IV and $K0$ as shown in the Appendix.
Step 3 For $1 \leq i \leq m$, perform one of the following steps.
  – If $i \in U$, then set $C[i] = P[i]$ and compute $Y[i] = S[i] \oplus D_{K1}(P[i] \oplus S[i])$.
  – If $i \notin U$, then compute $C[i] = S[i] \oplus E_{K1}(P[i] \oplus S[i])$ and set $Y[i] = P[i]$.
Step 4 Form $P[m+1] = Y[1] \oplus \cdots \oplus Y[m]$, and compute

$$C[m+1] = S[0] \oplus E_{K1}(P[m+1] \oplus S[m+1]).$$

Transmit $C = (C[0], \ldots, C[m+1])$.

[Note to referees: We can include a Figure describing PEMI here, but we are currently pressed for time.]

Example 1. Suppose that the sender and receiver agree that the first data block will be transmitted in plaintext rather than ciphertext. Then

$$Y[1] = S[1] \oplus D_{K1}(P[1] \oplus S[1]),$$

and $Y[i] = P[i]$ for $2 \leq i \leq m$. The tranmitted message appears as

$$C[0], P[1], C[2], \ldots, C[m], C[m+1],$$

where $C[2], \ldots, C[m]$ contain the encryptions of $P[2], \ldots, P[m]$.

PEMI: Decryption with Message Integrity Verification

Step 1 Extract $IV = C[0]$.
Step 2 Generate $S[0], \ldots, S[m+1]$ from IV and $K0$.
Step 3 For $1 \leq i \leq m$ perform one of the following steps.
  – If $i \in U$, then set $P[i] = C[i]$ and compute $Y[i] = S[i] \oplus D_{K1}(P[i] \oplus S[i])$.
  – If $i \notin U$, then compute $P[i] = S[i] \oplus D_{K1}(C[i] \oplus S[i])$ and set $Y[i] = P[i]$.
Step 4 Compute $P[m+1] = S[m+1] \oplus D_{K1}(C[m+1] \oplus S[0])$.
Step 5 If $P[m+1] = Y[1] \oplus \cdots \oplus Y[m]$, then the message integrity is verified, and the message is $P = (P[1], \ldots, P[m])$.

Note that the message integrity verification for a PEMI ciphertext C is exactly the same as the message integrity verification for an IAPM ciphertext C. While IAPM used the "hidden" values $P[1], \ldots, P[m]$ to compute the checksum, this PEMI mode uses the hidden decrypted values of $P[i]$, since $P[i]$ is transmitted.

2.1 Format

The format of a PEMI ciphertext is a description of which blocks will be sent unencrypted and which blocks will be sent encrypted. The message integrity verification of PEMI is independent of which blocks were sent encrypted and which blocks were sent unencrypted. That is, PEMI does not verify the format of the ciphertext. This opens PEMI to the possibility of an attack in which the claimed format of the ciphertext is changed. We do not consider this to be a serious threat as there are several ways to resist this threat. One simple solution is that the sender and receiver can agree *a priori* on the format. An alternative solution is to define part of the unencrypted plaintext to include formatting data.

2.2 Performance

The same principles used in PEMI can be used to create other modes that provide partial encryption with message integrity. One example is the so-called Hawkes-Rose (HR) variant of IAPM [4]. In the HR variant, the checksum is defined as $P[m+1] = \sum_{j \notin U} P[j]$, with

$$C[m+1] = E_{K1}(P[m+1] \oplus S[m+1]) \oplus S[0] \oplus \sum_{j \in U} [S[i] \oplus E_{K1}(P[i] \oplus S[i])].$$

Here is a comparison of PEMI and the HR variant.

– The HR variant encryption requires only block cipher encryption. PEMI encryption requires both block cipher encryption and block cipher decryption.
– In the HR variant encryption, all block cipher encryptions can be performed in parallel; there is no latency. In PEMI encryption, $P[m+1]$ can only be formed after the block cipher decryptions of the unencrypted blocks. Thusm the computation of $C[m+1]$ cannot take place until after these decryptions. Consequently, not all operations in PEMI encryption can be performed in parallel; there is a latency of one block cipher encryption.
– The HR variant decryption and message integrity verification requires both block cipher encryption and block cipher decryption. PEMI decryption and message integrity verification requires only block cipher decryption.
– In the HR variant decryption and message integrity verification, the block cipher encryptions of the unencrypted blocks must be performed to obtain $A = \sum_{j \in U} [S[i] \oplus E_{K1}(P[i] \oplus S[i])]$, before the receiver can decrypt to form

$$P[m+1] = S[m+1] \oplus D_{K1}(C[m+1] \oplus S[0] \oplus A).$$

  That is, the computation of $P[m+1]$ cannot take place until after these encryptions. Consequently, not all operations in the HR variant decryption and message integrity verification can be performed in parallel; there is a latency of one block cipher encryption. In PEMI decryption and message integrity verification, all block cipher decryptions can be performed in parallel; there is no latency.

All the designs that we considered (for partial encryption with message integrity) appeared to require a mixture of block cipher encryption and block cipher decryption either during encryption or during decryption and message integrity verification. All the designs that we considered also required a latency of at least one block cipher encryption or decryption at some point: either during encryption or during decryption and message integrity verification.

Performance-wise, there is little to differentiate between PEMI and the HR variant. Both schemes allow for significant parallelization, so they are very fast. However, we prefer PEMI because message verification for PEMI is identical to message integrity for IAPM. This means that the security of PEMI for message integrity can be easily related to the security of IAPM for message integrity. As the security proof for IAPM already exists, this makes our work much easier.

3 Proof of Message Integrity

A message integrity attack (MIA) on PEMI or IAPM is a two step process. In the first step, the adversary chooses plaintexts $P^1, \ldots, P^Z$, and gets the sender to reply $C^1, \ldots, C^z$. Note that each plaintext and ciphertext consists of multiple blocks: $C^i = (C^i[0] \ldots, C^i[m^i + 1])$, where m^i is the length of the i-th plaintext. In the second step, the adversary produces a new ciphertext C' (distinct from the other ciphertexts C^k) for which there is a non-negligible probability that it is valid. There is a slight difference between the MIA on PEMI and the MIA on IAPM; the difference is easier to express with some additional notation.

1. **PEMI.** Let $Y[j]$, $1 \leq j \leq m$, be defined as above for PEMI: if $j \in U$, then $Y[j] = P[j]$; otherwise

$$Y[j] = S[j] \oplus D_{K1}(P[j] \oplus S[j]).$$

   For $0 \leq j \leq m$ define $Z[j] = C[j]$, noting that $Z[j] = P[j]$ for $j \in U$. Also define

$$\begin{aligned} Y[m+1] &= Y[1] \oplus \cdots Y[m], \\ Z[m+1] &= S[0] \oplus E_{K1}(Y[m+1] \oplus S[m+1]). \end{aligned}$$

   For the i-th plaintext message P^i, let U^i denote the set of indices for which $P^i[j]$ is sent unencrypted.
2. **IAPM.** For $1 \leq j \leq m$, define $Y[j] = P[j]$ and $Z[j] = C[j] = S[j] \oplus E_{K1}(Z[j] \oplus S[j])$. Finally define

$$\begin{aligned} Y[m+1] &= Y[1] \oplus \cdots Y[m], \\ Z[m+1] &= S[0] \oplus E_{K1}(Y[m+1] \oplus S[m+1]). \end{aligned}$$

The differences between an MIA on PEMI and an MIA on IAPM are in whether the values of $Y^i[j]$ and $Z^i[j]$ are known, unknown, chosen or uniformly distributed.

1. In an MIA on IAPM, the values of $Y^i[j]$, $1 \leq i \leq m+1$, are chosen and thus known. In an MIA on PEMI, the values of $Y^i[j]$, $j \notin U^i$, are chosen and known but the values of $Y^i[j]$, $j \in U^i$, are unknown and expected to be uniformly distributed.
2. In an MIA on IAPM, the values of $Z^i[j]$ are known and expected to be uniformly distributed. In an MIA on PEMI, the values of $Z^i[j]$, $i \notin U^i$, are unknown and expected to be uniformly distributed, but the values of $Z^i[j]$, $i \notin U^i$, are chosen. In an MIA on PEMI, the adversary gets to choose U^i based on $P^1, \ldots, P^{i-1}$, and $C^1, \ldots, C^{i-1}$. This means that the adversary gets to choose which values of $Z^i[j]$ will be plaintexts (and for which the adversary can choose the values) and which values of $Z^i[j]$ will be unknown, uniformly distributed ciphertexts.

We can translate the proof for IAPM [3] into a proof using $Y^i[j]$ and $Z^i[j]$ in the place of $P^i[j]$ and $C^i[j]$. A proof for PEMI results if the proof for IAPM still holds when some values of $Z^i[j]$ are chosen, while the corresponding values of $Y^i[j]$ are unknown and uniformly distributed. The fact that some values of $Y^i[j]$ are unknown and uniformly distributed cannot increase the success probability of the attack. The question is whether the fact that $Z^i[j]$, $j \in U^i$, are chosen increases the success probability.

In Jutla's calculations for bounding the success probability of an MIA on IAPM [3], the "un-predictability" of the ciphertexts has the following effect. To compute a probability where there are, say, t possible ciphertexts resulting the initial value IV, Jutla sums the probabilities over the individual possible ciphertexts and introduces a factor of $1/t$, seeing as the actual value for the ciphertext is not known in advance. As only a bound is required, we obtain a bound β that holds for all t possible fixed ciphertexts. We then see that the bound on the sum of the probabilities is $\sum_{\text{ciphertexts}} \beta/t = \beta$. That is, the bound on the success probability does not depend on how many possible values there are for the ciphertexts.

This means that in the same analysis of PEMI, the same bounds would apply even though the attacker has the opportunity to choose some portion of the ciphertexts. Thus, an attack on the message integrity of PEMI has negligible success probability; that is, PEMI remains secure for message integrity.

The proof of privacy for PEMI remains the same as that for IAPM, except of course that some of the plaintexts are transmitted in clear.

3.1 Partial Block Encryption with Message Integrity

PEMI above relies upon entire blocks being either plaintext or ciphertext, which is an undesirable reflection of the structure of the block cipher. Since the AES works on large blocks of 16 bytes, this might be seen to be unduly restrictive. It is desirable to handle, somehow, the case of partially encrypted blocks. We call a block a "partial block" when some of its contents are to be sent encrypted, and some of it is to be sent clear. Again we assume that the format of the message, in the sense of what part is to be sent unencrypted, is agreed a priori between the parties.

One way to solve this problem is to treat the partial block as a block to be transmitted unencrypted for the purposes of calculating the Message Integrity above. Since this ensures message integrity, we encrypt the desired part with an additive cipher, effectively in counter mode. The IV was used in conjunction with $K0$ to generate the $S[i]$ values, but $K1$ is used to encrypt and decrypt the actual data blocks, so we can reuse the IV in this context.

Consider the i-th block, and assume that we have a bitwise mask $M[i]$ that defines the bits to be encrypted. ($M[i]$ won't be all zeros, or the sender would just send the plaintext; similarly it won't be all ones, or the sender would just send the ciphertext. Beyond that, there is no reason to further constrain which bits are to be encrypted, and which left unencrypted.) The block to be transmitted is calculated so:

$$Partial[i] = P[i] \oplus (M[i] \text{ AND } E_{K1}(IV \oplus i)),$$

where AND denotes the bit-wise AND operation. For each block to be partially encrypted, an extra encryption operation over the basic PEMI is required to generate the counter-mode mask. Since this extra operation can also be done in parallel with the data encryption or decryption operation required by the message integrity calculation or verification (respectively) the overall latency in parallel mode need not increase.

Security of the message integrity follows directly from that of PEMI. The message integrity was already guaranteed when plaintext was sent, so obscuring part of the plaintext cannot weaken the message integrity. Security of the privacy of the data follows directly from the proof of security of basic Counter Mode encryption, which we do not repeat here. We only require that the two encryption operations needed are independent, to prove the security of this enhancement.

In one (privacy) operation, we encrypt $E_{K1}(IV \oplus i)$. In the other (integrity) operation, we encrypt $E_{K1}(P[i] \oplus S[i])$. These values are unrelated, except that the $S[i]$ are related to the IV through encryption with key $K0$ (not $K1$). Therefore, any dependence between these two encryption operations must result from the attacker's ability to relate the input and output of encryption using the key $K0$, which would therefore imply his ability to break the underlying block cipher with an unknown key. Thus the partially encrypted blocks are fully integrity protected and partially encrypted as required.

4 Conclusion

We have proposed a new mode for symmetric block ciphers that provides Partial Encryption with Message Integrity, hence the name PEMI. We have sketched a method in which a proof of message integrity for PEMI can be based on a proof of message integrity for IAPM. The same principles used in PEMI can be used to create other modes that provide partial encryption with message integrity. For example, a PEMI-like mode may also be constructed from OCB [5].

References

1. V. Gligor and P. Donescu. Fast encryption and authentication: XCBC encryption and XECB authenitcation modes. *See* `http://csrc.nist.gov/encryption/modes/proposedmodes/xcbc/xcbc-spec.pdf`, 2001.
2. C. S. Jutla. Encryption modes with almost free message integrity. *See* `http://csrc.nist.gov/encryption/modes/proposedmodes/iacbc/iacbc-spec.pdf`, 2001.
3. C. S. Jutla. Parallelizable encryption mode with almost free message integrity. *See* `http://csrc.nist.gov/encryption/modes/proposedmodes/iapm/iapm-spec.pdf`, 2001.

4. C. S. Jutla. Hawkes-Rose variant of IAPM and mesage integrity. *See* `http://csrc.nist.gov/encryption/modes/proposedmodes/iapm/variantprood.pdf`, 2001.
5. P. Rogaway, M. Bellare, J. Black, and T. Krovetz. OCB: A block-cipher mode of operation for efficient authenticated encryption. *See* `http://csrc.nist.gov/encryption/modes/proposedmodes/ocb/ocb-spec.pdf`, 2001.

5 Appendix

This section describes the method proposed by Jutla [3] for generating the sequence $S[0], \ldots, S[m+1]$ from a key and an initial value IV. The sequence has the property that $S[i] \oplus S[j]$ is uniformly distributed for $0 \leq i < j \leq m+1$. This method requires $t = \lceil(\log_2(m+3)\rceil$ encryptions. This text is taken (almost verbatim) from [3].

1. Set $W_0 = E_{K0}(IV)$, (noting that $IV = C[0]$).
2. Set $S[0] = W_0$.
3. For $j = 1$ to m, perform the following steps:
   (a) Find the index k of the least significant non-zero bit in $(j+1)$.
   (b) If $(j+1) = 2^k$, then compute $W_k = E_{K0}(W_0 + k)$.
   (c) Set $S[j] = S[j+1] \oplus W_k$.