International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Comparative Study of MAC Algorithms in Pervasive Computing Environment

P. Bakkiya Lakshmi[1], K. Kumar[2]

[1]MCA (final year) Veltech Technical University

[2]Assistant Professor, Veltech Technical University

Abstract: ***Nowadays coming technology, many applications depend upon the existence of small devices that can exchange, share the information and form communication networks. In an expressive manner of such applications, the confidentiality and integrity of the communicated messages are of particular interest area. In this paper work, we are going to propose two novel techniques for authenticating short encrypted messages that are directed to meet the requirements of mobile and pervasive applications. By taking advantage of the fact that the message to be authenticated must also be encrypted, we propose provably secure authentication codes that are more efficient than any message authentication code in the literature. The key idea behind the proposed techniques is to utilize the security that the encryption algorithm can provide to design more efficient authentication mechanisms, as opposed to using standalone authentication primitives. The main uses of one way cryptographic hash function for message authentication. A popular example of iterated cryptographic hash functions in the design of message authentication codes is HMAC, which was proposed by Bell are et al.***

Keywords: communication network, two novel technique for authenticating, cryptography hash functions, secure authentication, MAC Algorithms

1. Introduction

A popular class of unconditionally secure authentication is based on universal hash-function families, pioneered by Carter and Wingman]. Since, the study of unconditionally secure message authentication based on universal hash functions has been attracting research attention, both from the design and analysis standpoints. The basic concept allowing for unconditional security is that the authentication key can only be used to authenticate a limited number of exchanged or share messages. Since the management of one-time keys is considered impractical in many applications, computationally secure MACs have become the method of choice for most real-life applications. In computationally secure MACs, keys can be used to authenticate an arbitrary number of messages. That is, after agreeing on a key, legitimate users can exchange an arbitrary number of authenticated messages with the same key. Depending on the main building block used to construct them, computationally secure MACs can be classified into three main categories: block cipher based, cryptographic hash function based, or universal hash-function family based. The use of universal hash-function families in the Carter-Wegman style is not restricted to the design of unconditionally secure authentication. Computationally secure MACs based on universal hash functions can be constructed with two rounds of computations. In the first round, the message to be authenticated is compressed using a universal hash function. Then, in the second round, the compressed image is processed with a cryptographic function (typically a pseudorandom function1). Popular examples of computationally secure universal hashing based MACs include, but are not limited to, One of the main differences between unconditionally secure MACs based on universal hashing and computationally secure MACs based on universal hashing is the requirement to process the compressed image with a cryptographic primitive in the latter class of MACs. This round of computation is necessary to protect the secret key of the universal hash function. That is, since universal hash functions are not cryptographic functions, the observation of multiple message-image pairs can reveal the value of the hashing key. Since the hashing key is used repeatedly in computationally secure MACs, the exposure of the hashing key will lead to breaking the security of the MAC. Thus, processing the compressed image with a cryptographic primitive is necessary for the security of this class of MACs. This implies that unconditionally secure MACs based on universal hashing are more efficient than computationally secure ones. On the negative side, unconditionally secure universal hashing based MACs are considered impractical in most modern applications, due to the difficulty of managing one-time keys. There are two important observations to make about existing MAC algorithms. First, they are designed independently of any other operations required to be performed on the message to be authenticated. For instance, if the authenticated message must also be encrypted, existing MACs are not designed to utilize the functionality that can be provided by the underlying encryption algorithm. Second, most existing MACs are designed for the general computer communication systems, independently of the properties that messages can possess. For example, one can find that most existing MACs are inefficient when the messages to be authenticated are short. (For instance, UMAC, the fastest reported message authentication code in the cryptographic literature [34], has undergone large algorithmic changes to increase its speed on short messages.

2. Motivation

Let N □ 1 be an upper bound on the length, in bits, of exchanged messages. That is, messages to be authenticated can be no longer than (N □ 1)-bit long. Choose p to be an N-bit long prime integer. (If N is too small to provide the desired security level, p can be chosen large enough to

satisfy the required security level.) Choose an integer ks uniformly at random from the multiplicative group Z_ p; ks is the secret key of the scheme. The prime integer, p, and the secret key, ks, are distributed to legitimate users and will be used for message authentication. Note that the value of p need not be secret, only KS is secret. Let E be any IND-CPA secure encryption algorithm. Let m be a short messages (N 1 bit or shorter) that is to be transmitted to the intended receiver in a confidential manner (by encrypting it with E). Instead of authenticating the message using a traditional MAC algorithm, consider the following procedure. On input a message m, a random nonce r 2 Zp is chosen. (We overload m to denote both the binary string representing the message, and the integer representation of the message as an element of Zp. The same applies to ks and r. The distinction between the two representations will be omitted when it is clear from the context.) Now, r is appended to the message and the resulting m k r, where "k" denotes the concatenation operation, goes to the encryption algorithm as an input. Then, the authentication tag of message m can be calculated as follows.

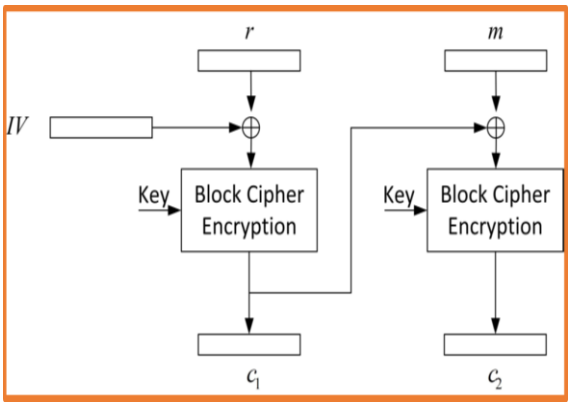

Figure 1: Existing System

The above diagram explains the Cipher Block Chaining (CBC) mode of encryption used for message encryption. The random number, r, is treated as the first block of the plaintext.

3. Problem Definition

There are two notions of enforceability in authentication codes. Namely, a MAC algorithm can be weakly Un forgeable under chosen message attacks (WUF-CMA), or strongly un forgeable under chosen message attacks (SUFCMA). A MAC algorithm is said to be SUF-CMA if, after launching chosen message attacks, it is infeasible to forge a message-tag pair that will be accepted as valid regardless of whether the message is "new" or not, as long as the tag has not been previously attached to the message by an authorized user. If it is only hard to forge valid tags for "new" messages, the MAC algorithm is said to be WUF-CMA.

In this paper, we will modify the original scheme described in Section to make it SUF-CMA, without incurring any extra computational overhead.

4. Literature Survey

Our study to determine the best algorithms in pervasive environment is done as follows the three algorithm sare;(HMAC,CMAC,UMAC).

4.1 HMAC Authentication

Hash-based message authentication code (HMAC) is a mechanism for calculating a message authentication code involving a hash function in combination with a secret key. This can be used to verify the integrity and authenticity of a a message. The use HMAC authentication a digest is computed using a composite of the URI, request timestamp and some other headers (depending on the implementation) using the supplied secret key. The key identifier along with the digest, which is encoded using Base64 is combined and added to the authorization header. Function hmac (key, message)
if (length(key) > blocksize) then key = hash(key) // keys longer than blocksize are shortened
end if
if (length(key) < blocksize) then
key = key ‖ [0x00 * (blocksize - length(key))] // keys shorter than blocksize are zero-padded
(where ‖ is concatenation)
end if
o_key_pad = [0x5c * blocksize] ⊕ key // Where blocksize is that of the underlying hash function
i_key_pad = [0x36 * blocksize] ⊕ key // Where ⊕ is exclusive or (XOR)
return hash(o_key_pad ‖ hash(i_key_pad ‖ message)) // Where ‖ is concatenation end function

4.2 CMAC (Cipher-based Message Authentication Code):

Cipher-based message authentication codes12 (or CMACs) are a tool for calculating message authentication codes using a block cipher coupled with a secret key. You can use an CMAC to verify both the integrity and authenticity of a message.

4.3 Universal hashing

Let's say the hash function is chosen from a class of hash functions H, which maps messages into D, the set of possible message digests. This class is called universal[3,6,7,8] if, for any distinct pair of messages, there are at most |H|/|D| functions that map them to the same member of D.

```
#define uchar unsigned char
void UHash24 (uchar *msg, uchar *secret, int len, uchar *result)
{
uchar r1 = 0, r2 = 0, r3 = 0, s1, s2, s3, byteCnt = 0, bitCnt, byte;
while (len-- > 0) {
if (byteCnt-- == 0) {
s1 = *secret++;
s2 = *secret++;
s3 = *secret++;
byteCnt = 2;
}
```

```
byte = *msg++;
for (bitCnt = 0; bitCnt < 8; bitCnt++) {
if (byte & 1) { /* msg not divisible by x */
r1 ^= s1; /* so add s * 1 */
r2 ^= s2;
r3 ^= s3;
}
byte >>= 1; /* divide message by x */
if (s3 & 0x80) { /* and multiply secret with x, subtracting
the polynomial when necessary to keep its order under 24 */
s3 <<= 1;
if (s2 & 0x80) s3 |= 1;
s2 <<= 1;
if (s1 & 0x80) s2 |= 1;
s1 <<= 1;

s1 ^= 0x1B; /* x^24 + x^4 + x^3 + x + 1 */
}
else {
s3 <<= 1;
if (s2 & 0x80) s3 |= 1;
s2 <<= 1;
if (s1 & 0x80) s2 |= 1;
s1 <<= 1;
}
} /* for each bit in the message */
} /* for each byte in the message */
*result++ ^= r1;
*result++ ^= r2;
*result++ ^= r3;
}
```

5. Architecture Diagram

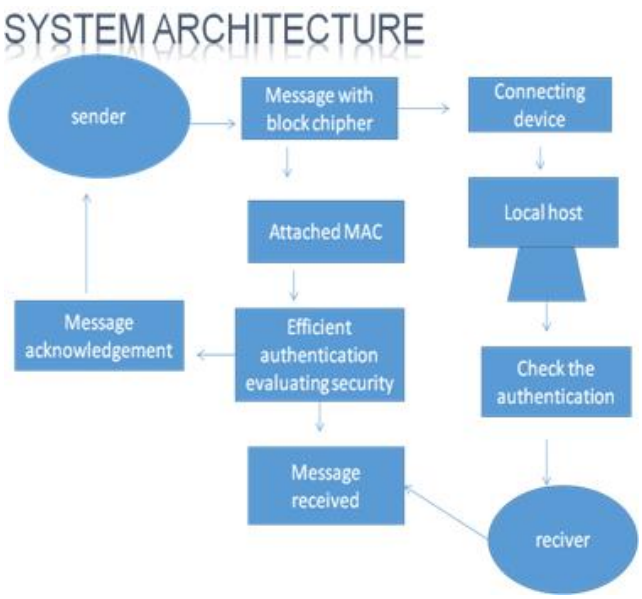

Figure 2: System Architecture

In this section, we describe our first authentication scheme that can be used with any IND-CPA secure encryption algorithm. An important assumption we make is that messages to be authenticated are no longer than a predefined length. This includes applications in which messages are of fixed length
that is known a priori, such as RFID systems in which tags need to authenticate their identifiers, sensor nodes reporting events that belong to certain domain or measurements within a certain range, etc. The novelty of the proposed scheme is to utilize the encryption algorithm to deliver a random string and use it to reach the simplicity and efficiency of one-time pad authentication without the need to manage impractically long keys.

6. Justifications of results

This is mainly because there exist secure MAC algorithms that leak information about the authenticated message (a detailed example of such a MAC.

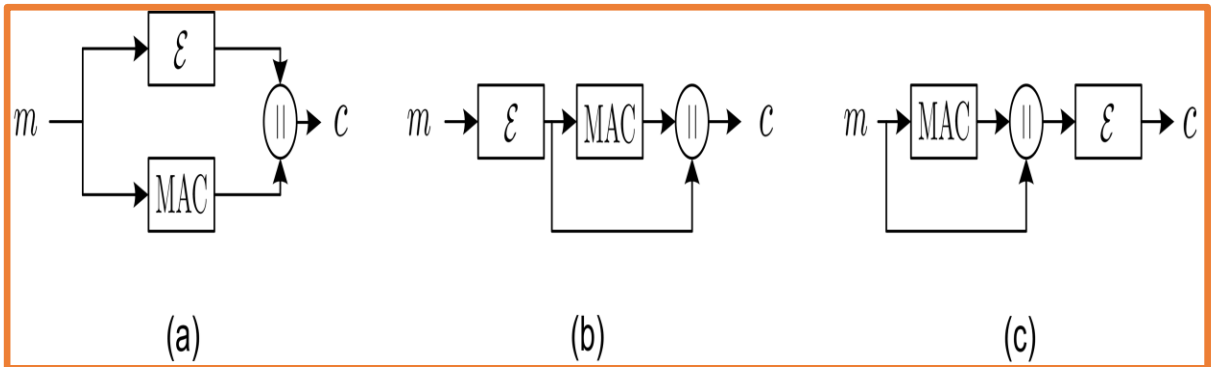

Figure 3: Result justification

However, the proposed authenticated encryption scheme is at least as private as the underlying encryption algorithm. Since the encryption algorithm is IND-CPA secure, the resulting composition provides IND-CPA.

7. Related Works

A popular class of unconditionally secure authentication is based on universal hash-function families, pioneered by Carter and Wegman. Since then, the study of unconditionally secure message authentication based on universal hash functions has been attracting research attention, both from the design and analysis standpoints. The basic concept allowing for unconditional security is that the authentication key can only be used to authenticate a limited number of exchanged messages. Since the management of one-time keys is considered impractical in many applications, computationally secure MACs have become the method of choice for most real-life applications. In computationally secure MACs, keys can be used to authenticate an arbitrary number of messages. That is, after agreeing on a key, legitimate users can exchange an arbitrary number of authenticated messages with the same key. Depending on the main building block used to construct them, computationally secure MACs can be classified into three main categories: block cipher based, cryptographic hash function based, or universal hash-function family based.

8. Conclusion and Future Work

In this work, a new technique for authenticating short encrypted messages is proposed. The fact that the message to 11 be authenticated must also be encrypted is used to deliver a random nonce to the intended receiver via the cipher text. This allowed the design of an authentication code that benefit from the simplicity of unconditionally secure authentication without the need to manage one-time keys. In particular, it has been demonstrated in this paper that authentication tags can be computed with one addition and a one modular multiplication. Given that messages are relatively short, addition and modular multiplication can be performed faster than existing computationally secure MACs[10,11] in the literature of cryptography. When devices are equipped with block ciphers to encrypt messages, a second technique that utilizes the fact that block ciphers can be modeled as strong pseudorandom permutations is proposed to authenticate messages using a single modular addition. The proposed schemes are shown to be orders of magnitude faster, and consume orders of magnitude less energy than traditional MAC algorithms. Therefore, they are more suitable to be used in computationally constrained mobile and pervasive devices.

References

[1] J. Carter and M. Wegman, “Universal classes of hash functions,” in Proceedings of the ninth annual ACM symposium on Theory of computing–STOC’77. ACM, 1977, pp. 106–112.
[2] M. Wegman and J. Carter, “New classes and applications of hash functions,” in 20th Annual Symposium on Foundations of Compute Science–FOCS’79. IEEE, 1979, pp. 175–182.
[3] L. Carter and M. Wegman, “Universal hash functions,” Journal of Computer and System Sciences, vol. 18, no. 2, pp. 143–154, 1979.
[4] M. Wegman and L. Carter, “New hash functions and their use in authentication and set equality,” Journal of Computer and System Sciences, vol. 22, no. 3, pp. 265–279, 1981. J. Bierbrauer, “A2-codes from universal hash classes,” in Advances in Cryptology–EUROCRYPT’95, vol. 921, Lecture Notes in Computer Science. Springer, 1995, pp. 311–318.
[5] M. Atici and D. Stinson, “Universal Hashing and Multiple Authentication,” in Advances in Cryptology–CRYPTO’96, vol. 96, Lecture Notes in Computer Science. Springer, 1996, pp. 16–30.
[6] T. Helleseth and T. Johansson, “Universal hash functions from exponential sums over finite fields and Galois rings,” in Advances in cryptology–CRYPTO’96, vol. 1109, Lecture Notes in Computer Science. Springer, 1996, pp. 31–44.
[7] V. Shoup, “On fast and provably secure message authentication based on universal hashing,” in Advances in Cryptology–CRYPTO’96, vol. 1109, Lecture Notes in Computer Science. Springer, 1996, pp. 313–328.
[8] J. Bierbrauer, “Universal hashing and geometric codes,” Designs, Codes and Cryptography, vol. 11, no. 3, pp. 207–221, 1997.
[9] B. Alomair, A. Clark, and R. Poovendran, “The Power of Primes: Security of Authentication Based on a Universal Hash-Function Family,” Journal of Mathematical Cryptology, vol. 4, no. 2, 2010.
[10] B. Alomair and R. Poovendran, “E-MACs: Towards More Secure and More Efficient Constructions of Secure Channels,” in the 13th International Conference on Information Security and Cryptology – ICISC’10. Springer, 2010.
[11] FIPS 113, “Computer Data Authentication,” Federal Information Processing Standards Publication, 113, 1985.
[12] ISO/IEC 9797-1, “Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher,” 1999.
[13] M. Dworkin, “Recommendation for block cipher modes of operation: The CMAC mode for authentication,” 2005.
[14] T. Iwata and K. Kurosawa, “omac: One-key cbc mac,” in Fast Software Encryption–FSE’03, vol. 2887, Lecture notes in computer science. Springer, 2003, pp. 129–153.