**INTRODUCTION**

PRESERVING the integrity of messages exchanged over public channels is one of the classic goals in cryptography and the literature is rich with message authentication code (MAC) algorithms that are designed for the sole purpose of preserving message integrity. Based on their security, MACs can be either unconditionally or computationally secure. Unconditionally secure MACs provide message integrity against forgers with unlimited computational power. On the other hand, computationally secure MACs are only secure when forgers have limited computational power.

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.

CBC-MAC is one of the most known block cipher based MACs, specified in the Federal Information Processing Standards publication 113 and the International Organization for Standardization ISO/IEC 9797-1. CMAC, a modified version of CBC-MAC, is presented in the NIST special publication 800-38B, which was based on the OMAC of. Other block cipher based MACs include, but are not limited to, XOR-MAC and PMAC. The security of different MACs has been exhaustively studied.

The use of one-way cryptographic hash functions for message authentication was introduced by Tsudik. A popular example of the use of iterated cryptographic hash functions in the design of message authentication codes is HMAC, which was proposed by Bellare. HMAC was later adopted as a standard. Another cryptographic hash function based MAC is the MDx-MAC proposed by Preneel and Oorschot. HMAC and two variants of MDx- MAC are specified in the International Organization for Standardization ISO/IEC 9797-2. Bosselaers et al. described how cryptographic hash functions can be carefully coded to take advantage of the structure of the Pentium processor to speed up the authentication process.

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.

Indeed, universal hashing based MACs give better performance when compared to block cipher or cryptographic hashing based MACs. In fact, the fastest MACs in the cryp-tographic literature are based on universal hashing. The main reason behind the performance advantage of universal hashing based MACs is the fact that processing messages block by block using universal hash functions is orders of magnitude faster than processing them block by block using block ciphers or cryptographic hash functions.

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, has undergone large algorithmic changes to increase its speed on short messages.)

Nowadays, however, there is an increasing demand for the deployment of networks consisting of a collection of small devices. In many practical applications, the main purpose of such devices is to communicate short messages. A sensor network, for example, can be deployed to monitor certain events and report some collected data. In many sensor network applications, reported data consist of short confidential measurements. Consider, for instance, a sensor network deployed in a battlefield with the purpose of reporting the existence of moving targets or other temporal activities. In such applications, the confidentiality and integrity of reported events are of critical importance.

In another application, consider the increasingly spreading deployment of radio frequency identification (RFID) systems. In such systems, RFID tags need to identify themselves to authorized RFID readers in an authenticated way that also preserves their privacy. In such scenarios, RFID tags usually encrypt their identity, which is typically a short string (for example, tags unique identifiers are 64-bit long in the EPC Class-1 Generation-2 standard [39]), to protect their privacy. Since the RFID reader must also authenticate the identity of the RFID tag, RFID tags must be equipped with a message authentication mechanism.

Another application that is becoming increasingly important is the deployment of body sensor networks. In such applications, small sensors can be embedded in the patient’s body to report some vital signs. Again, in some applications the confidentiality and integrity of such reported messages can be important.

There have been significant efforts devoted to the design of hardware efficient implementations that suite such small devices. For instance, hardware efficient implementations of block ciphers have been proposed. Implementations of hardware efficient cryptographic hash functions have also been proposed. However, there has been little or no effort in the design of special algorithms that can be used for the design of message authentication codes that can utilize other operations and the special properties of such networks. In this paper, we provide the first such work.