DESIGNING OF SPECIAL PURPOSE MACS FOR BUILDING MORE EFFICIENT SECURE CHANNELS

[1]Nisha S G, [2]Manjunatha Kumar B H
Department of CSE,Sri Siddhartha Institute of Technology,Tumkur, India
Email: [1]nishasg20@gmail.com, [2]bhm.cse@gmail.com

Abstract—Cryptography is the art and science of keeping messages Secure. In this paper, we propose the deployment of a new cryptographic primitive for the construction of secure channels using the Generic composition method of Cryptography. We introduce the design of E-MACs, Message Authentication Codes for Encrypted messages. By proposing the first instance of E-MACs, we show how the structure of the Encrypt-and-Authenticate Composition system can be utilized to increase the efficiency and security of the authentication process. The paper shows how a universal hash function based E-MAC can be computed with fewer operations than what standard universal hash functions based MACs require. Moreover, we will also show how E-MACs can further utilize the special structure of the E&A system to improve the security of the authentication process.

***Keywords*—Confidentiality; authenticity; message authentication code (MAC); authenticated encryption; universal hash families**

I. INTRODUCTION

There are two main approaches for the construction of secure channels in cryptography: a dedicated approach and a generic approach. In the dedicated approach, a cryptographic primitive is designed to achieve authenticated encryption as a standalone system. In the generic approach, an authentication primitive is combined with an encryption primitive to provide message integrity and confidentiality.

Generic compositions can be constructed in three different ways:

- MtE: MAC-then-encrypt. Use two keys. First authenticate the plaintext by computing the MAC value as T = MAC(K1, M). Then encrypt the message plus tag: E (K2, (M || T). This approach is taken by the SSL/TLS protocols
- EtM: Encrypt-then-MAC. Use two keys. First encrypt the message to yield the ciphertext C = E (K2, M). Then authenticate the ciphertext with T = MAC (K1, C) to yield the pair (C, T). This approach is used in the IPSec protocol
- E&M: Encrypt-and-MAC. Use two keys. Encrypt the message to yield the ciphertext C = E (K2, M).Authenticate the plaintext with T = MAC (K1, M) to yield the pair (C, T). These operations can be performed in either order. This approach is used by the SSH protocol

.

Over dedicated primitives, generic compositions possess several design and analysis advantages due to their modularity and the fact that encryption and authentication schemes can be designed, analyzed, and replaced independently from each other in this system. Further, and most important, generic compositions can lead to faster implementations of authenticated encryption when fast encryption algorithms, such as stream ciphers, are combined with fast MACs, such as universal hash functions based MACs.

However, generic compositions are more involved than just combining an encryption algorithm and a MAC algorithm. In this system the security of different generic compositions of authenticated encryption systems is analyzed.

Using a secure encryption algorithm (secure in the sense that it provides privacy against chosen plaintext attacks) and a secure MAC (secure in the sense that it provides unforgeability against chosen-message attacks), it was shown that only the EtA will guarantee the construction of secure channels. Therefore, special attention must be paid to the design of secure channels if the E&A or the AtE compositions are used. Although significant efforts have been devoted to the design of dedicated authenticated encryption primitives and the analysis of the generic compositions, little effort has been made to the design of new primitives in order to improve the efficiency and security of generic compositions. In this system, we introduce the design of special purpose MACs to be used in the construction of E&A and AtE compositions. The main motive behind this work was the intuition that MACs used in the generic construction of authenticated encryption systems, unlike standard MACs, can utilize the fact that messages to be authenticated must also be encrypted. That is, since both the encryption and authentication algorithms are applied to the same message, there might be some redundancy in the computations of the two primitives. If this turned out to be the case, removing such redundancy can improve the efficiency of the overall operation. The E&A and AtE compositions, however, impose an extra requirement on the MAC algorithm. As opposed to the EtA compositions, the tag in the E&A and AtE compositions is a function of the plaintext message (not the ciphertext as in EtA). Therefore, the tag must be at least as confidential as the ciphertext since, otherwise, the secrecy of the plaintext can be compromised by an adversary observing its corresponding tag.

In this system, we propose the deployment of a new cryptographic primitive for the construction of secure channels using the E&A and AtE compositions. We introduce the design of E-MACs: Authentication Codes for Encrypted Messages. By proposing the first instance of E-MACs, we show how the structure of the E&A and AtE systems can be utilized to increase the efficiency and security of the authentication process. In particular, we show how a universal hash function based E-MAC can be computed with fewer operations than what standard universal hash functions based MACs require. That is, we will demonstrate that universal hash functions based E-MACs can be implemented without the need to apply any cryptographic operation to the compressed image. Moreover, we will also demonstrate that E-MACs can further utilize the special structures of the E&A and AtE systems to improve the security of the authentication process. That is, we will show how universal hash functions based E-MACs can be secured against the key-recovery attack, to which standard universal hash functions based MACs are known to be vulnerable. Finally, we will show that the extra confidentiality requirement on E-MACs can be achieved rather easily, again, by taking advantage of the E&A and AtE structures.

II. RELATED WORK

In Efficient Authentication for Mobile and Pervasive Computing With today’s technology, many applications rely on the existence of small devices that can exchange information and form communication networks. In a significant portion of such applications, the confidentiality and integrity of the communicated messages are of particular interest. In this work, we propose a novel technique 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 technique is to append a short random string to the plaintext message before encryption to facilitate a more efficient authentication.

The drawback here is that there is no file splitting or Packet Filtering scheme in this system and less security due to lack message authentication on packets

In The Keyed-Hash Message Authentication Code (HMAC).This standard describes a keyed-hash message authentication code (HMAC), a mechanism for message authentication using cryptographic hash functions. HMAC can be used with any iterative Approved cryptographic hash function, in combination with a shared secret key. The disadvantages are the type HMAC is Platform dependent. There is no intermediary accessory to authenticate the Message.

Combining message encryption and authentication. The first part of the paper explains the need for combining message encryption and authentication. We begin with the example to emphasize the fact that privacy‡ does not imply authenticity. Then we prove, one needs both privacy and authenticity, even if one's aim is just getting privacy. In the second part we present an overview of different methods for providing authenticated encryption (AE) i.e. generic compositions, single-pass modes and two-pass combined modes. We analyze what are the advantages and disadvantages of different AE constructions. In the third part of the paper we focus on nonce§ based authenticated encryption modes. Our motivation is the wish to know the methodology of designing authenticated encryption mode of operation. Disadvantages are there is no file splitting or Packet Filtering scheme in this system. Less Security due to single key mode operations.

III. SECURITY MODEL

Authenticity

Functions that provide a way to verify the integrity of information (for example, against unauthorized changes over a communications network) and which use a shared secret key are called MAC (message authentication codes). The notion of a MAC and its security definition is well understood [4]. Here we outline the main ingredients of this definition as used later in the paper. A MAC scheme is described as a family of (deterministic) functions over a given domain and range. The key shared by the parties that use the MAC scheme determines a specific function from this family. This specific Function is used to compute an authentication tag on each transmitted message and the tag is appended to the message. A recipient of the information that knows the MAC key can re-compute the tag on the received message and compare to the received tag. Security of a MAC scheme is defined through the inability of an attacker to produce a forgery, namely, to generate a message, not transmitted between the legitimate parties, with its valid authentication tag.

SHA 512

SHA-2: A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. SHA-2 is a set of cryptographic hash functions designed by the NSA (U.S. National Security Agency).[3] SHA stands for Secure Hash Algorithm. Cryptographic hash functions are mathematical operations run on digital data; by comparing the computed "hash" (the execution of the algorithm) to a known and expected hash value, a person can determine the data's integrity. For example, computing the hash of a downloaded file and comparing the result to a previously published hash result can show whether the download has been modified or tampered with. A key aspect of cryptographic hash functions is their collision resistance: nobody should be able to find two different input values that result in the same hash output. SHA-2 includes significant changes from its predecessor, SHA-1. The SHA-2 family consists of six hash functions with digests (hash values) that are 224, 256, 384 or 512 bits: SHA-224, SHA-256, SHA-384, SHA-512, SHA-512/224, and SHA-512/256.

Privacy AES

An encryption scheme is a triple of (probabilistic) algorithms (KEYGEN; ENC;DEC) where KEYGEN defines the process (and resultant probability distribution

tion) by which keys are generated, while ENC and DEC are the encryption and decryption operations with the usual inverse properties. To simplify notation We use ENC to denote the encryption operation itself but also as representing the whole scheme. The main notion behind the common Definitions of security of encryption are semantic security, or its (usually) equivalent formulation via plaintext indistinguishability. In this formulation an attacker against a scheme ENC is given a target ciphertext y and two candidate plaintexts x1; x2 such that y = ENC(xi). The encryption scheme has the indistinguishability property if the attacker cannot guess the right value of i with probability significantly better than 1=2. The security of the scheme is quantified via the time invested by the attacker and the probability beyond ½ to guess correctly.

The Advanced Encryption Standard (AES) specifies a FIPS-approved cryptographic algorithm that can be used to protect electronic data. The AES algorithm is a symmetric block cipher that can encrypt (encipher) and decrypt (decipher) information. Encryption converts data to an unintelligible form called ciphertext; decrypting the ciphertext converts the data back into its original form, called plaintext.

IV. THE PROPOSED E-MAC

- Service Provider

In this module, the Service Provider browses the file, init MAC and splits file into packets and sends their data files to the Receiver via EMAC Router by providing the IP address.

- E MAC Router
  The EMAC Router we introduce the design of a new cryptographic primitive to be used in the construction of secure channels. Instead of using general purpose MACs, we used the deployment of special purpose MACs, named E-MACs. The EMAC Router will receive the packets sent by the source verifies the packet and forwards to the destination and verifies the sent packets. The EMAC Router is Responsible for view the credentials like view attackers and MAC with their tags Packets, Sender MAC, Receiver MAC. The EMAC Router will perform the revoke the contents injected by the malicious user.
- Remote Receiver (End User)
  In this module, the End user can receive the data file and verify the mac details by comparing the sent mac and received mac from the Router. If the packet is injected with fake content then the packet can be revoked by the router.
- Attacker

In this module, the Attacker injects the fake key to the particular packet in the EMAC Router. Once a message that causes a collision is found, partial information about the hashing keys can be exposed. Using this key information an attacker can forge valid tags for fake messages. The EMAC Router can revoke the malicious injected data in packets

V. OVERVIEW OF PROPOSED E-MAC

Software Requirement Specification

This Chapter describes about the requirements. It specifies the hardware and software requirements that are required in order to run the application properly. The Software Requirement Specification (SRS) is explained in detail, which includes overview of this dissertation as well as the functional and non-functional requirement of this dissertation. SRS for E-MACs: Towards More Secure and More Efficient Constructions of Secure Channels

Functional	Control the Congestion in Router; Mac Key generation, File splitting, Providing the Confidentiality by generating the MAC and Packet verification in Router ,Protects the Files in Network, The data will collect by using data aggregate router. Find the malicious content in the packet.
Non- Functional	The Sender and Receiver never know the Key Generation.
External interface	LAN , Routers, WAN
Performance	Finding File Attackers Information, Viewing different group and their respective keys, Viewing white key details and File Details, Viewing the sent and received MAC in the EMAC Router, Viewing the attackers in EMAC Router, End User Can receive the file, End user can revoke the malicious content.
Attributes	File Management, Confidentiality, authenticity, message authentication code (MAC), authenticated encryption, universal hash families.

Table: 3.1 Summaries of SRS

Functional Requirements

Functional Requirement defines a function of a software system and how the system must behave when presented with specific inputs or conditions. These may include calculations, data manipulation and processing and other specific functionality. In this system following are the functional requirements:-

- The Service provider has to browses the file and initializes MAC.
- The Service provider splits file into packets/ block and send their data files to the Receiver via EMAC Router.

- The EMAC Router will receive the packets sent by the source and verifies the packet.
- EMAC Router forwards to the destination and verifies the sent packets in the destination.
- End user can receive the data file and verify the mac details by comparing the sent mac and received mac from the Router.
- The Attributes are File Management, Confidentiality, authenticity, message authentication code (MAC), authenticated encryption, universal hash families.

Non – Functional Requirements

Non – Functional requirements, as the name suggests, are those requirements that are not directly concerned with the specific functions delivered by the system. They may relate to emergent system properties such as reliability response time and store occupancy. Alternatively, they may define constraints on the system such as the capability of the Input Output devices and the data representations used in system interfaces. Many non-functional requirements relate to the system as whole rather than to individual system features. This means they are often critical than the individual functional requirements. The following non-functional requirements are worthy of attention.

A. Encrypt-and-Authenticate Composition

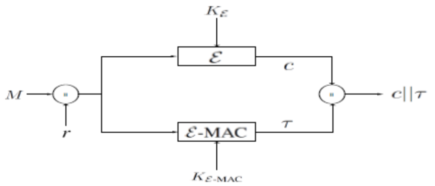

Fig. 2. A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an E&A composition.

A. Authenticate-then-Encrypt Composition

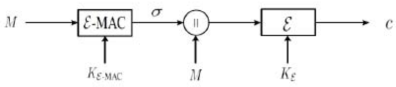

Fig. 3. A block diagram illustrating the use of $\mathcal{E}$-MAC to construct an AtE composition.

A block diagram depicting the use of the proposed E-MAC to construct an Authenticate-then-Encrypt composition is shown in Figure 3.As in the E&A, assume legitimate users agreed on using an encryption algorithm, E, that provides indistinguishability under chosen plaintext attacks (IND-CPA). Based on a security parameter N, legitimate users choose p to be the largest N-bit long prime integer.

VI. PERFORMANCE OF E-MACS

The proposed E-MACs, being generic, can be used alongside stream ciphers, one of the major performance advantages of generic compositions over dedicated ones (since stream ciphers are known to be much faster than their block cipher counterparts.

One class of MACs that is of a particular interest, due its fast implementation, is the class of MACs based on universal hash-function families. In universal hash function families based MACs, the message to be authenticated is first compressed using a universal hash function in the Carter-Legman style and, then, the compressed image is processed with a cryptographic function. Indeed, processing messages using universal hash functions is faster than processing them block by block using block ciphers. Combined with the fact that processing short strings is faster than processing longer ones, it becomes evident why universal hash functions based MACs are the fastest for message authentication. (The speed champions of MACs in the literature of cryptography are UMAC and hash127; both of which are based on universal hash functions).

Recall that universal hash function based MACs consist of two sequential operations: a universal hashing followed by a cryptographic operation. Observe further that universal hashing is much faster than cryptographic primitives. For instance, while universal hash functions can run in about 0.34 cycles/byte , the cryptographic hash functions SHA-256 and SHA-512 run in about 23.73 cycles/byte and 40.18 cycles/byte, respectively. That is, universal hashing computations are typically orders of magnitude faster than cryptographic computations. Therefore, it is evident how eliminating the need to post-process the compressed image with a cryptographic

Function will have an impact on the computational efficiency of the overall construction.

Recently, however, Handschuh and Preneel discovered a vulnerability in universal hash functions based MACs. They demonstrated that once a collision in the universal hash function is achieved, subsequent forgeries can succeed with higher probabilities. Their attack is not directed to a specific universal hash family and can be applied to all such MACs. The recommendation of the work in is not to reuse the universal hash function keys, thus going back to the earliest use of universal hash families for unconditionally secure authentication, or proceeding with the less efficient, yet more secure, block cipher based MACs.

VII. SECURITY ANALYSIS

Weak unforgeability is the standard notion it should be computationally infeasible for the adversary to find a message-tag pair in which the message is “new,” even after a chosen-message attack. Strong unforgeability requires that it be computationally infeasible for the adversary to find a new message-tag pair even after a chosen message attack. (The message does not have to be new as long as the output tag was not previously attached to this message by the legitimate parties.)

We note that any pseudorandom function is a strongly unforgeable MAC, and most practical MACs seem to be strongly unforgeable. Therefore, analyzing the composition methods under this notion is a realistic and useful approach.

Secure: The composite encryption scheme in question is proven to meet the security requirement in question, assuming only that the component encryption scheme meets IND-CPA and the message authentication scheme is unforgeable under chosen-message attack.

— Insecure: There exists some IND-CPA secure symmetric encryption and some message authentication scheme unforgeable under chosen-message attack such that the composite scheme based on them does not meet the security requirement in question.

As we can see from Figure 3, the encrypt-then-MAC method of is secure from all points of view, making it a good choice for a standard. The use of a generic composition method secure in the sense above is advantageous from the point of view both of performance and of security architecture. The performance benefit arises from the presence of fast MACs such as HMAC and UMAC. The architectural benefits arise from the stringent notion of security being used. To be secure, the composition must be secure for all possible secure instantiations of its constituent primitives. An application can thus choose a symmetric encryption scheme and a message authentication scheme independently (these are usually already supported by existing security analyses) and then appeal to some fixed and standard composition technique to combine them. No tailored security analysis of the composed scheme is required.

VIII. FUTURE ENHANCEMENT

The proposed universal hash family used for the implementation of the proposed E-MAC is not the only possible solution. Different assumptions about the underlying encryption algorithm may lead to different constructions of E-MACs. That is, whether the encryption is a stream cipher, cipher block chaining (CBC) mode block cipher, electronic code book (ECB) mode block cipher, etc., can have an impact on the design and performance of the composition. We only show here how the semantic security of the underlying encryption algorithm can be utilized to improve the efficiency and security of message authentication. Further improvements in E-MACs performance using specific modes of operations is left for a continuing research in this direction.

IX. CONCLUSION

In this work, we studied the generic composition of authenticated encryption systems. We introduced EMACs, a new symmetric-key cryptographic primitive that can be used in the construction of E&A and AtE compositions. By taking advantage of the E&A and AtE structures, the use of E-MACs is shown to improve the efficiency and security of the authentication operation. More precisely, since the message to be authenticated is encrypted, universal hash functions based E-MACs can designed without the need to apply cryptographic operations on the compressed image, since this can be replaced by operations performed by the encryption algorithm. Further, by appending a random string at the end of the plaintext message, E-MAC can be secured against key-recovery attacks.

REFERENCES

[1] B. Alomair and R. Poovendran, “E-MACs: Towards More Secure and More Efficient Constructions of Secure Channels,” in ICISC’10. Springer, 2010.

[2] V. Gligor and P. Donescu, “Fast Encryption and Authentication: XCBC Encryption and XECB Authentication Modes,” in FSE’01 workshop. Springer, 2002.

[3] C. Jutla, “Encryption modes with almost free message integrity,” Journal of Cryptology, vol. 21, no. 4, pp. 547–578, 2008.

[4] P. Rogaway, M. Bellare, J. Black, and T. Krovetz, “OCB: A blockcipher mode of

operation for efficient authenticated encryption," in ACM CCS'01. 2001.
[5] N. Ferguson, D. Whiting, B. Schneier, J. Kelsey, and T. Kohno, "Helix: Fast encryption and authentication in a single cryptographic primitive," in FSE'03 workshop. Springer, 2003.
[6] M. Bellare, P. Rogaway, and D. Wagner, "The EAX mode of operation," in FSE'04 workshop. Springer, 2004.
[7] B. Alomair, "Authenticated Encryption: How Reordering can Impact Performance," in ACNS'12. Springer, 2012.
[8] T. Ylonen and C. Lonvick, "The Secure Shell (SSH) Transport Layer Protocol," RFC 4253, Tech. Rep., 2006.
[9] A. Freier, P. Karlton, and P. Kocher, "The SSL Protocol Version 3.0," Internet Engineering Task Force (IETF), 2011.