Authenticated Encryption

Elena Pagnin

FDAT085

Abstract. Encryption (Enc) is the cryptographic primitive to achieve data confidentiality. Intuitively, encryption schemes enable Alice and Bob to communicate while Eve cannot understand the content of their conversation. Despite its "hiding power", encryption does not directly guarantee message integrity, that is, no one can prevent Eve from modifying Alice's encrypted data in such a way that Bob decrypts a message different from the original one. The cryptographic primitive to guarantee that the data in transit from Alice to Bob has not been modified by Eve, is called Message Authentication Codes (MAC). In this short paper, we discuss the combination of Enc and MAC: Authenticated Encryption (AE) and present a motivating example for this new cryptographic tool. Moreover, we provide a simplified overview of the algorithms that define AE schemes and compare the three possible ways to combine Enc and MAC. We conclude presenting an overview of the evolution of AE schemes and describing the current state of the CAESAR competition, which aims at defining a candidate for the first standard AE scheme (essentially the authenticated encryption equivalent of AES or SHA-3).

1 Introduction

Consider the following scenario. Alice and Bob are police officers in charge of catching Eve, the head of a gang settled in Göteborg. In order to catch Eve, Alice sets mobile checkpoints around the city. *How can Alice communicate to Bob to move his check point to a different location in a secure way?* In particular, we want to prevent Eve (or Eve's gang) from (1) reading the content of the message and (2) changing the content of the message; while, at the same time, we want Bob to be able to check (3) if the message he received actually came from Alice. Authenticated Encryption (AE) represents the perfect solution in this scenario as it guarantees (1) confidentiality, (2) integrity, and (3) authenticity of the encapsulated data.

1.1 Notation

Throughout this short paper, Enc denotes the encryption algorithm of a symmetric encryption scheme and MAC denotes a message authentication code. The concatenation of two strings s_1, s_2 is denoted as $s_1||s_2$. We recall below some highlevel definitions which might come handy. The Enc algorithm takes as input a secret key (denoted as K_E) and a plaintext message (denoted as m) and outputs a ciphertext (c). Intuitively, an encryption scheme is secure if an adversary, holding a ciphertext c cannot (in polynomial time) determine what is the plaintext message it decrypts to. A MAC takes as input a secret key (denoted as K_M) and a message (denoted as m) and outputs a tag t, which is usually shorter than m. Intuitively, a message authentication code is secure if an adversary, holding pairs $(m_i, t_i = \mathsf{MAC}(K_M, m_i))$ cannot (in polynomial time) produce a tampered tag t' such that $t' = \mathsf{MAC}(K_M, m')$ for a new message $m' \notin \{m_i\}$.

2 Authenticated Encryption

Authenticated Encryption is a form of symmetric-key encryption which simultaneously provides data confidentiality, integrity and authenticity. The need for AE emerged at the beginning of our century from the observation that a number of practical attacks introduced into production protocols (such as SSL/TLS) was caused by the lack of authentication, and data integrity check. The syntax of an AE scheme can be simplified with the following three algorithms:

$\mathsf{KeyGen}(1^\lambda) \rightarrow \mathsf{key}$. The key generation algorithm takes as input the security parameter λ (in unary notation) and outputs the secret key of the scheme key. Usually this key is composed of two parts $\mathsf{key} = (K_E, K_M)$.

$\mathsf{AE.Enc}(\mathsf{key}, m, \mathsf{aux}) \rightarrow (c, t)$. The encryption algorithm takes as input the secret key, a message m, and some associated data aux. The output is the authenticated-encrypted ciphertext, which can be divide in two parts c and t.

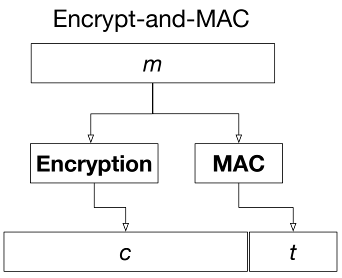
$\mathsf{AE.Dec}(\mathsf{key}, (c, t)) \rightarrow m \cup \perp$. The decryption algorithm takes as input the secret key and an authenticated-encrypted ciphertext (c, t). It outputs a plaintext m or an error symbol $\perp$. Intuitively, AE.Dec returns $\perp$ if the ciphertext is not integer (*e.g.,* it has been modified, or is not coming from the expected source), otherwise it returns the plaintext corresponding to the authenticated ciphertext.

3 Classical Composition Methods

The most natural way to create AE schemes is to build them combining existing primitives. This approach is formally called *generic composition* [4]. Intuitively, this corresponds to slot-in two (or more) building blocks into a new construction. For instance, we can pick a secure encryption scheme and a secure message authentication code (under two independent keys) and combine them to achieve (simultaneously) data integrity, authenticity and secrecy (the goals of authenticated encryption). There are exactly three ways to do it: Encrypt-and-MAC, Encrypt-then-MAC, MAC-then-Encrypt. Below we formalise these methods and give a brief discussion about their (in)security[1].

3.1 Encrypt-and-MAC

The Encrypt-and-MAC method is the most natural attempt to provide authenticated encryption. A variant of this approach is used in the transport layer of SSH [16]. In this paradigm, one first encrypts the plaintext message m, to obtain a ciphertext c. One then appends to the ciphertext c a MAC of the plaintext called tag t. Formally we have: $\mathsf{AE}_{E\&M}\big((K_E, K_M), m\big) = \mathsf{Enc}(K_E, m)||\mathsf{MAC}(K_M, m)$. The figure on the right gives a figurative representation of this method. The authenticated decryption procedure is performed by first decrypting c to get the plaintext m, and then verifying if $t == \mathsf{MAC}(K_M, m)$. It is well-known that this approach is the least secure among the three we present [4]. To give a quick reason why it is so, consider the fact that often the tag t output by a MAC contains information about the message it has been computed on (since MAC do not aim at

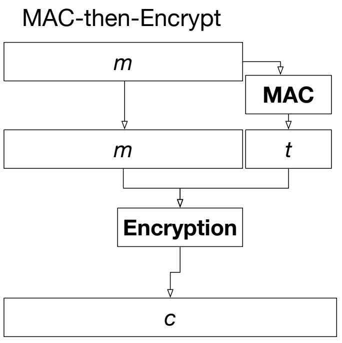


protecting the message's secrecy). Therefore, sending t weakens the security of the encryption scheme (as it is leaking some information about the plaintext m).

[1] In the remainder of this section, we consider MAC to be a strongly unforgeable message authentication code, and Enc to be a IND-CPA secure encryption scheme.

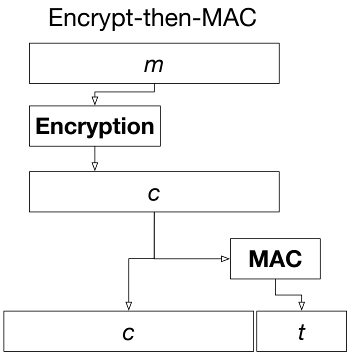
3.2 MAC-then-Encrypt

The MAC-then-Encrypt paradigm aims at fixing the *leakage of information* that affects the previous method by first MAC-ing the message, and then encrypting both the message and the tag produced by the MAC. As shown in [4], this approach behaves only slightly better than the previous one.
The SSL/TLS protocol employs an adaptation of the MAC-then-Encrypt method to achieve authenticated encryption of messages [9]. Formally we have: $\mathsf{AE}_{MtE}\big((K_E, K_M), m\big) = \mathsf{Enc}\big(K_E, m||\mathsf{MAC}(K_M, m)\big)$, Authenticated decryption is performed by first decrypting the ciphertext c and then checking the integrity of the plaintext by comparing $t == \mathsf{MAC}(K_M, m)$. The main issue with this solution is that it does not guarantee the integrity of the received ciphertext. For instance, an adversary could substitute c with $c' \neq c$ such that c' decrypts to a message m' of the adversary's choice and an opportune tag t' for which $t' == \mathsf{MAC}(K_M, m')$ holds.



3.3 Encrypt-then-MAC

The Encrypt-then-MAC approach is most secure one among the three presented methods [4]. The *IPSEC* protocol is currently using a variant of this authentication encryption scheme [11]. This paradigm works as follows: one first encrypts the plaintext message m, to obtain a ciphertext c. Then, the ciphertxt c is given as input to the MAC to obtain the tag t. The crucial difference from previous methods is that here the MAC is not applied to the plaintext message m, but to the ciphertexts. Formally we have:
$\mathsf{AE}_{EtM}\big((K_E, K_M), m\big) = \mathsf{Enc}(K_E, m)||\mathsf{MAC}(K_M, c)$, where $c = \mathsf{Enc}(K_E, m)$. Authenticated decryption is performed by first checking the integrity of the ciphertext, *i.e.,* check if $t == \mathsf{MAC}(K_M, c)$ and then proceeding with the standard decryption procedure. We see that in this approach the MAC provides ciphertext integrity (which implies plaintext integrity). This property is essentially what makes the Encrypt-then-MAC paradigm the *best* one (security-wise). Beside its neat structure, this approach still requires two independent keys for encryption and authentication, and it is not robust to implementation errors [8].



4 State of the Art of Authenticated Encryption

The composition methods presented in Section 2 are a natural way to build AE from existing primitives, however, this is not the only approach to do so. Over the years, more efficient AE schemes have been constructed using different techniques [10, 14], including keyless permutations [6] and stream ciphers [2]. These new approaches have changed our understanding of AE from a *blending* of two primitives into a cryptographic building block on its own.

4.1 A widely used AE scheme: GCM

As of 2017, "Galois Counter Mode" (GCM) [12, 13] is the best known technique to transform an encryption scheme into an AE one[2]. GCM is often applied to AES (Advance Encryption Standard), and the AES-GCM scheme is employed in different settings, *e.g.,* protocols for Ethernet security [3], OpenVPN [7], and many other secure web-communication services. The increasing number of application scenarios for authenticated encryption, as well as the discovery of weak keys for GCM [15], have risen the need for defining a reference AE scheme to be used as standard. For this reason the international cryptologic research community has launched a world-wide competition to select a portfolio of algorithms for AE. Although CAESAR is not a standardisation project, standard committees will take inspiration from the winning candidate.

4.2 The CAESAR Competition

The CAESAR[3] competition started in 2013 and was co-founded by NIST[4] and Dan Bernstein. The goal of CAESAR is to invite the cryptographic community to design new secure AE schemes. The winner of CAESAR is expected to improve on AES-GCM by either providing a higher level of security (with similar performance) or by being faster (with a similar level of security) [5]. The CAESAR committee is expected to publish a collection of DIAC (Directions in Authenticated Ciphers) in summer 2017. An interesting by-product of the competition is the development of cryptanalysis techniques to attack the candidate schemes.

Fifty-seven independent AE schemes were proposed after the first CAESAR call for submission [5, 1]. After more than one year intensive review and cryptanalysis nine of these first-round candidates where broken and withdrawn from the competition. We are now at the final round of CAESAR and only fifteen candidates are left. All the finalist schemes are based on one of the following five underlying constructions: block cipher, stream cipher, compression function (hash), keyless permutation (sponge), dedicated. For a comparison of the finalist designs we refer the interested reader to [1].

5 Conclusions

The crucial difference between plain encryption and authenticated encryption is that the latter one provides authenticity *in addition* to data confidentiality. Usually AE schemes have a more complicated structure than just an encryption or a MAC scheme. In order to stimulate the design of new authenticated encryption schemes a competition has been launched in 2013. The aim is to identify an AE scheme that improves on existing proposals and is secure, versatile and hopefully easier to understand. We retain that the winner of this competition will inspire the first standard for AE. Moreover, we hope that by 2020, secure implementations of authenticated encryption will be available for a number of protocols, web-application and communication services, and finally mitigate the damages generated by well-known flaws in current implementations.

References

1. F. Abed, C. Forler, and S. Lucks. General classification of the authenticated encryption schemes for the caesar competition. *Computer Science Review*, 22:13–26, 2016.

[2] Explaining the details of this transformation is out of the scope of this short paper. To the curious readers, we summarise GCM as a two-pass mode based on a block cipher (in "counter mode") that employs a polynomial hash function on Galois fields.

[3] Competition for Authenticated Encryption: Security, Applicability, and Robustness.

[4] US National Institute of Standards and Technology.

2. M. Ågren, M. Hell, T. Johansson, and W. Meier. Grain-128a: a new version of grain-128 with optional authentication. *International Journal of Wireless and Mobile Computing*, 5(1):48–59, 2011.
3. H. Altunbasak, S. Krasser, H. Owen, J. Grimminger, H.-P. Huth, and J. Sokol. Securing layer 2 in local area networks. *Networking-ICN 2005*, pages 699–706, 2005.
4. M. Bellare and C. Namprempre. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 531–545. Springer, 2000.
5. D. J. Bernstein. Caesar call for submissions. January 27, 2014.
6. G. Bertoni, J. Daemen, M. Peeters, and G. Van Assche. Permutation-based encryption, authentication and authenticated encryption. *Directions in Authenticated Ciphers*, 2012.
7. K. Bhargavan and G. Leurent. On the practical (in-) security of 64-bit block ciphers: Collision attacks on http over tls and openvpn. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 456–467. ACM, 2016.
8. N. Ferguson and B. Schneier. A cryptographic evaluation of ipsec. *Counterpane Internet Security, Inc*, 3031:14, 2000.
9. A. Freier, P. Karlton, and P. Kocher. The secure sockets layer (ssl) protocol version 3.0. 2011.
10. C. S. Jutla. Encryption modes with almost free message integrity. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 529–544. Springer, 2001.
11. S. Kent. Ip encapsulating security payload (esp). 2005.
12. D. McGrew and J. Viega. The galois/counter mode of operation (gcm). *NIST Modes Operation Symmetric Key Block Ciphers*, 2005.
13. D. A. McGrew and J. Viega. The security and performance of the galois/counter mode (gcm) of operation. In *International Conference on Cryptology in India*, pages 343–355. Springer, 2004.
14. P. Rogaway, M. Bellare, and J. Black. Ocb: A block-cipher mode of operation for efficient authenticated encryption. *ACM Transactions on Information and System Security (TISSEC)*, 6(3):365–403, 2003.
15. M.-J. O. Saarinen. Cycling attacks on gcm, ghash and other polynomial macs and hashes. In *Fast Software Encryption*, pages 216–225. Springer, 2012.
16. T. Ylonen and C. Lonvick. The secure shell (ssh) transport layer protocol. 2006.