ZETA: Towards Tagless Authenticated Encryption

Anindya Shankar Bhandari, Dipanwita Roy Chowdhury
Indian Institute of Technology, Kharagpur

May 24, 2017

Abstract

Tag-based message authentication is a popular cryptographic technique to digitally sign messages. However, for short messages, it often incurs additional costs due to large tags. In this paper, we propose a new scheme that achieves tagless message authentication. The scheme leverages a trade-off between character support and complexity of forgery to provide information security and authenticity.

Keywords

Authenticated Encryption; Minimum Tag Expansion; Authentication; Encryption; Sequence Number; Initialization Vector

1 Introduction

In tag-based digital signature protocols, a message digest is constructed and then digitally signed. This is sent with a message as a verification tag. The verification algorithm, upon receiving the packet, computes the digest of the message and 'signs' it, and compares it with the received tag to look for a match. The entire process is a vital part of Authenticated Encryption [6]. The majority of available schemes for authenticated encryption today are nonce-based [7].
In practice, the current protocols for encryption and authentication are secure and cost-effective. But in the transmission of short messages ($\leq$ 128 bit plaintext), the tags and nonces often incur additional costs. Sending a message that is considerably shorter than or even the same size of the tags and nonces associated with it seems like a gargantuan waste. The size considered secure for message tags is 160 bits and for nonces it is anything that cannot be forged with a high enough probability, therefore 32-bit, 64-bit and 128-bit nonces are often used. We aim to nullify any kind of expansion to the original message.
The AERO scheme was proposed by McGrew and Foley [4, 3] in 2013, an important work in reducing tag length. The scheme appended the sequence number (nonce) with the plaintext, and used a wide psuedo-random permutation to encrypt it. Kazuhiko Minematsu [5] proposed a similar scheme and provides a

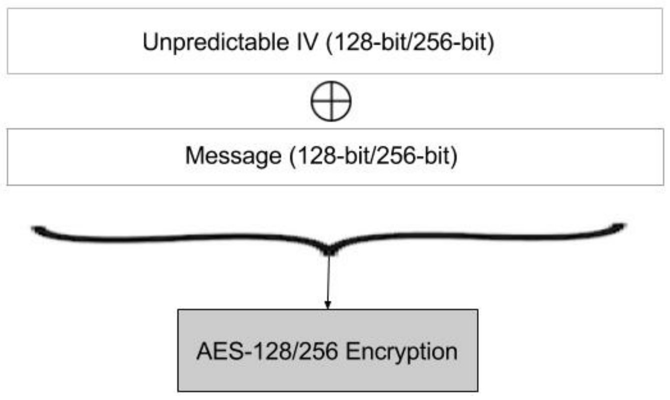

Figure 1: ZETA at a glance

security analysis for it. The scheme involves sending a message M, encrypted to give C, and an encrypted nonce L. Naturally, it is observable that the complexity of a forgery for these schemes is dependent on the length of the nonce, hence it must be sufficiently long.
The scheme we propose in this paper, ZETA - Zero Expansion for Tagless Authentication, does not expand the message by any bits at all. In order to achieve authenticity protection, the scheme utilizes the notion that in message transmission, not all characters that are available in ASCII are actually used in a given scenario. Hence ZETA is meant to be applied in scenarios where a relatively large number of characters remain unused. These scenarios will be explained in this paper.
The rest of this paper is divided as follows: Section 2 will detail the scheme and authentication method. Section 3 analyses its security. Section 4 concludes the paper.

2 The ZETA Scheme

ZETA requires a stateful receiver, it is needed to maintain a psuedorandom IV generator on both ends. It is necessary for the IV to be unpredictable, the reason for which we shall see in the Security Analysis section. Use of a stateful receiver is justified, as many available schemes make use of a stateful receiver as well, such as Bluetooth Low Energy [1]. This IV is XORed with the message before encryption, as seen in Figure 1. Decryption is done by reversing this process.
Authentication: To authenticate, each byte of the decrypted message is parsed to check if the characters are supported in the specific application. The number of supported characters is dependent on the usage scenario. An instruction based system that uses alphabetical codes to denote instructions would require only 26 of the 256 possible characters for valid functioning. This would allow for

an instruction space of $2^{75.2}$. Authentication would require only to iterate over the bytes of the message and check if each byte is supported - 1 ASCII range check per byte (ASCII 0x41 to 0x5A) in this case, so a total of 32 comparisons per message.
An English messaging service would require 95 of the available characters (given in the table below). If the length of the messages may be longer, they can be chained using AES-CBC, using the unpredictable IV, deterministically generated at both ends. It is a known issue that AES-CBC with a predictable IV is not secure against Chosen Plaintext Attacks, and neither would ZETA be secure against Chosen Message Attacks with a predictable IV. To authenticate, one must check that each byte of the message lies in the range ASCII 0x20 to 0x7E. Therefore 1 range check per byte, a total of 64 comparisons to authenticate a 256 bit message - quite fast.

Capital Letters	A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Small Letters	a b c d e f g h i j k l m n o p q r s t u v w x y z
Numbers	0 1 2 3 4 5 6 7 8 9
Punctuation	, . space ? ” ’ “ ‘ ( ) { } [ ] ! - _ / \
Other Characters	@ # $ % <>ˆ & * |˜+ = g̀rave

Based on the complexity of forgery requirement, the block size of AES can be set for the application. This shall be discussed further in the Security Analysis section.
ZETA is meant for lossless channels. In lossy channels, there comes in a requirement to iterate over a set of IVs to obtain the correct plaintext, which might be cumbersome.

3 Security Analysis

In this section we analyse the security of the ZETA scheme. Various known parameters [2] were used.

3.1 Complexity of Forgery

Figure 2 shows us how the complexity of forgery varies with the number of supported characters. It is contrasted with the available plaintext space.
For the aforementioned alphabetical instruction exchange scheme, at 128-bit block length, it achieves a complexity of forgery of $2^{52.8}$. To put this into perspective, let us assume that the receiver can process 50 million ciphertexts per second. This is a generous assumption, most servers would crash long before

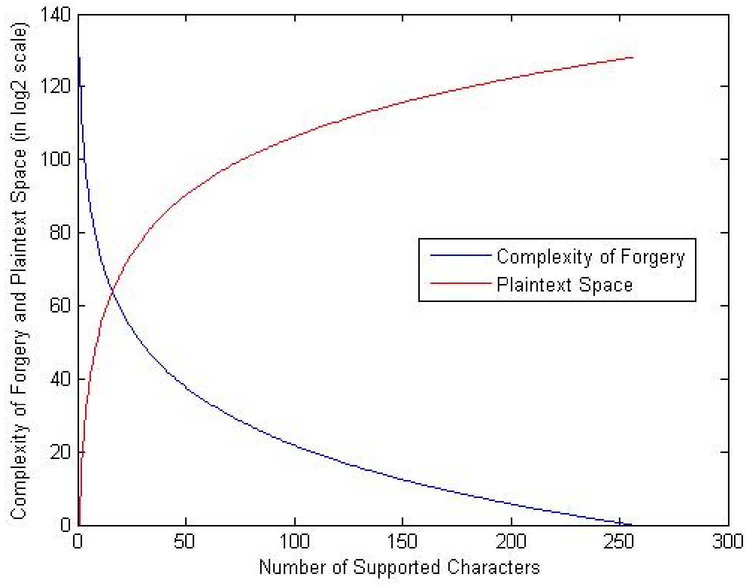

Figure 2: Complexity of Forgery and Plaintext Space variation with character support

such a number is achieved. With this in mind, $2^{52.8}$ takes 5 years to crack a valid forgery.
For the English Messaging service, we shall require the use of 256-bit blocks and AES256 to have an appreciable forgery complexity. 95 supported characters using AES256 yields a complexity of $2^{45.8}$. These 95 supported characters are enough for an English Messaging Service - a real life example scenario where ZETA may be used. 50 million ciphertexts per second for an SMS communication receiver does not make sense - yet with this assumption we would require nonstop bombardment of messages over 14 days for a forgery to succeed.

3.2 Replay Attacks

ZETA avoids replay attacks of the same message as, when a message is repeated, the Initialization Vector has changed, hence the ciphertext is different. The Nonce-Reset problem [8] is avoided, owing to the size of the IV (128-bit or 256-bit, as the case may be). However, as the pre-encryption space is greater than the post-encryption ciphertext space, a certain ciphertext may be replayed yielding a different valid plaintext. For a plaintext size of $\{0,1\}^n$, the pre-encryption space is $\{0,1\}^n \times \{0,1\}^{128}$ (the IV space), and post encryption space is $\{0,1\}^{128}$. Therefore a ciphertext may correspond to 2^n valid messages. Therefore, as the total plaintext space is $\{0,1\}^{128}$, the probability of replay is 2^{n-128}, which is equal to the probability of forgery. Hence the same arguments as above hold for replay attacks.

3.3 Chosen Message Attack (CMA)

We shall now investigate ZETA's resilience against CMA. We are using a modified version of the UF-CMA technique, where we include the possibility of replay attacks, as well as make necessary changes owing to the tagless nature of our scheme. The technique to model the security of our scheme is stated as follows: Say an adversary A sends a set of messages $\{M_i\}$ to a valid ciphertext generating oracle O, which returns the set of valid ciphertexts $\{C_i\}$ - Each encrypted by generating a psuedorandom IV to be XORed with it. O now issues a challenge for the adversary to send a valid ciphertext to it. It does not ask for the forgery of a specific message - any message will do, to make the security definition more robust (Existential Forgeability). Repetitions of queried ciphertexts are allowed - this is necessary to model replay attacks against ZETA. The key in use throughout the query ciphertexts and the challenge ciphertexts remains the same. There is no need for adaptive requests, as the oracle only asks for a forged ciphertext. Lastly, let the valid plaintext size be $\{0,1\}^n$.
To succeed, the attacker needs to modify or generate any ciphertext that shall be valid. However, owing to the avalanche effect of AES, probability of success is unchanged compared to the unknown plaintext situation. The probability of success is 2^{n-128}, and hence we conclude that the advantage is negligible.
For a successful replay attack, the adversary needs to query a total of 2^{128} plaintexts to probabilistically have an IV-collision. From the previous section, we know that a randomly selected ciphertext for a replay attack has a probability of 2^{n-128}.
Let us try an alternate scenario. Say, an adversary knows the IVs in use. Let IV_i denote the i-th IV to be used, and IV_{i+j} be the $(i+j)^{th}$ IV in use. A chooses a message M that need not belong to the valid plaintext space, but which satisfies the following conditions:
(i) $M \oplus IV_i$ is a valid plaintext
(ii) $M \oplus IV_{i+j}$ is a valid plaintext
At the i-th trial, the adversary queries $M \oplus IV_i$. The oracle finds the plaintext to be valid and returns $E_K((M \oplus IV_i) \oplus IV_i) = E_K(M)$. He then queries other ciphertexts and stops at the $(i+j-1)^{th}$ trial, and requests to be issued the challenge. He then sends $E_K(M)$ as his forgery. The oracle decrypts using AES and receives M, and XORs with IV_{i+j} to retrieve $M \oplus IV_{i+j}$, which is a valid plaintext. Hence, the scheme fails with a predictable IV.
Therefore, the most important requirement on the scheme is the presence of an unpredictable IV - otherwise the system is insecure against chosen message attacks.

3.4 IND-CPA security

Let $\Pi = (K, E, D)$ be an authenticated encryption scheme under a certain key K. The original definition uses an ensemble of keys, but we believe the use of a single key gives a more power to the adversary and hence a better estimate of IND-CPA advantage. Then, the IND-CPA advantage of a computationally bounded adversary A for Π is defined as:

$$Adv_{\Pi}^{IND-CPA}(A) \leq |Pr[K : A^{E(.,.)} \implies 1] - Pr[A^{\$(.,.)} \implies 1]|$$

A is an IND-CPA adversary. The task of A is to distinguish the real world, where it is given oracle access to $E_K(.,.)$ under a secret key K, from the random world, where A has access to a random oracle $\$(.,.)$ which returns consistent, random ciphertexts. If no such adversary A can perform significant better than random guessing, then Π protects the privacy of encrypted messages.
In ZETA, say the adversary is querying a single message from the plaintext space. The adversary needs to query the same message a large number of times before he can have a non-negligible advantage over random guessing. After a large number of queries to the oracle, he requests the challenge for that particular message and is asked to distinguish the result of the encryption function and a randomly generated ciphertext. This is mathematically equal to

$$Pr(Correct\ Guess) = \frac{1}{2} \times \frac{2^{128} - \mid r \mid}{2^{128}} + \frac{\mid r \mid}{2^{128}}$$

where r represents the set of requests/queries to the oracle, since this is the probability of a ciphertext for the plaintext to be known (the adversary needs to know only one message to be able to guess). Therefore the advantage $Adv(A)$ is almost equal to $\frac{|r|}{2^{128}}$ (as $\mid r \mid$ is much smaller than 2^{128}), which is negligible.

3.5 Chosen Message Attack with Decryption Assistance

CMA with Decryption Assistance follows the INT-CTXT definition, modified to our use-case. The adversary has access to both encryption and decryption machinery.
In a scheme where an IV is transmitted with the ciphertext (i.e. the receiver is not stateful), an encryption request would provide an IV+ciphertext pair, while a decryption request would need both an IV and a ciphertext as an input. In our case, we are making use of a stateful receiver, hence IVs are not transmitted. Therefore, use of the encryption function changes the IV for any subsequent encryptions. Whereas the use of the Decryption function does no such thing. The goal of the adversary is to make any valid forgery. We assume a Probabilistic Polynomial Time (PPT) adversary.

$$Adv_{\Pi}^{INT-CTXT}(A) \leq Pr[K : A^{E(.,.),D(.,.)} \implies forges]$$

It turns out, this scenario translates to almost the same as the case without Decryption Assistance. Without decryption assistance, the probability of success was 2^{n-128}. So if the adversary is allowed a maximum of $\mid r \mid$ decryption queries to the oracle for a subset of his 2^{128-n} minimum required guesses, then the probability that he will perform a successful forgery by his $r+1^{th}$ guess is equivalent to the complement of the probability that all his guesses have been wrong so far, including his $r+1^{th}$ guess (because, if he finds a correct ciphertext during any of his requests to the oracle, he may use that as his forgery). Therefore,

$$Adv_{\Pi}^{INT-CTXT} = 1 - (\frac{2^{128-n}-1}{2^{128-n}})^{|r+1|}$$

Since $\mid r+1 \mid << 2^{128-n}$, the advantage comes out to be 2^{n-128}. This is because

$$1 - (\frac{2^{128-n}-1}{2^{128-n}})^{|r+1|} \leq \frac{1}{2^{128-n} - \mid r \mid}$$

Therefore, we conclude that Chosen Message Attack with Decryption Assistance does not yield any non-negligible advantage to the adversary.

4 Conclusion

The paper described a scheme in which practically usable complexities of forgery were achieved with *zero-tag expansion*. The method may find wide usage in systems where the cost of communication is very high. Achieving complexities of $2^{52.8}$ and $2^{45.8}$ would usually require 53-bit and 46-bit expansions respectively. We are working on how to increase the complexity of forgery further, while maintaining the use of no tag.
The scheme in its current form may be used in very low end devices. However its applicability does not end there. The scheme may be used in conjunction with a tag as well if a greater forgery complexity is desired - this allows ZETA to give a dramatic reduction in tag-length and increases ZETA's applicability domain to various scenarios.

References

[1] Bluetooth low energy. http://www.bluetooth.com/Pages/Low-Energy.aspx/.

[2] F. abed, C. Forler, and S. Lucks. General classification of the authenticated encryption schemes for the caesar competition. Cryptology ePrint Archive, Report 2014/792, 2014. http://eprint.iacr.org/2014/792.

[3] D. McGrew. Low power wireless scenarios and techniques for saving bandwidth without sacrificing security. NIST Lightweight Cryptography Workshop, 2015.

[4] D. McGrew and J. Foley. Authenticated encryption with replay protection (aero) internet-draft, 2013.

[5] K. Minematsu. Authenticated encryption with small stretch (or, how to accelerate aero). Cryptology ePrint Archive, Report 2015/738, 2015. http://eprint.iacr.org/2015/738.

[6] P. Rogaway. Authenticated-encryption with associated-data. In *Proceedings of the 9th ACM Conference on Computer and Communications Security*, CCS '02, pages 98–107, 2002.

[7] P. Rogaway. *Nonce-Based Symmetric Encryption*, pages 348–358. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[8] E. Zenner. *Nonce Generators and the Nonce Reset Problem*, pages 411–426. Springer Berlin Heidelberg, Berlin, Heidelberg, 2009.