Breaking POET Authentication with a Single Query

Jian Guo, Jérémy Jean, Thomas Peyrin, and Lei Wang

Division of Mathematical Sciences, School of Physical and Mathematical Science,
Nanyang Technological University, Singapore

{guojian,JJean,Thomas.Peyrin,Wang.Lei}@ntu.edu.sg

Abstract. In this short article, we describe a very practical and simple attack on the authentication part of POET authenticated encryption mode proposed at FSE 2014. POET is a provably secure scheme that was designed to resist various attacks where the adversary is allowed to repeat the nonce, or even when the message is output before verifying the validity of the tag when querying the decryption oracle. However, we demonstrate that using only a single encryption query and a negligible amount of computations, even without any special misuse from the attacker, it is possible to generate many valid ciphertext/tag pairs for POET. Our work shows that one should not use POET for any application where authentication property is required. Furthermore, we propose a possible patch to overcome this particular issue, yet without backing up this patch with a security proof.

Key words: authenticated encryption, CAESAR, POE, POET, cryptanalysis, authenticity

Authenticated encryption is a very useful cryptographic primitive that might benefit many security engineers and protocol designers, as it provides both privacy and authenticity when sending data. In particular, it avoids the classical threat of a misinterpretation of the privacy-only security provided by a simple encryption mode. The encryption part usually takes as input a message M, some public associated data A, a public nonce value N, a secret key K, and it outputs a ciphertext C and a tag value T. Then a decryption part takes as input a ciphertext C, a tag value T, some public associated data A, a public nonce value N, a secret key K, and outputs either the original message M, or an error character $\perp$ if the authentication process is not valid.

Many authenticated encryption solutions using existing components already exist (like using an encryption scheme for the privacy part and a MAC for the authenticity part), but a single primitive providing both properties at the same time, with a single core function, would potentially permit faster and simpler solutions. This line of research attracted a lot of attention, especially recently due to the incoming CAESAR authenticated competition [1] that will run from 2014 to 2017, and that will propose a portfolio of authenticated encryption solutions approved by the community.

The topic of authenticated encryption is quite complex, as many parameters, security definitions, use-cases have to be considered. Yet, in the past years some very useful properties have been proposed, such as the so-called nonce-misuse resistance. This security property ensures that even when the attacker can ask for the encryption of several messages with the same nonce, the security of the scheme is not completely broken. One can cite for example modes such as SIV [6], COPA [3], McOE [5], ELmE [4] or POET [2] that use a block cipher as basic primitive (and thus can be directly instantiated with AES for example). Such schemes are interesting because reusing a nonce is really an issue that might arise in practical applications (due to the limitations in the possibilities of the upper protocol or hardware, or due to human error when implementing the scheme).

In the same research direction, Fleischmann *et al.* [5] also identified decryption-misuse resistance to be an interesting property. A decryption-misuse authenticated encryption scheme can withstand adversaries that obtain the decryption of the queried ciphertext even thought the validity of the attached queried tag is not verified. Such adversaries are quite strong and model the fact that in practice it might be hard for some applications to wait for the tag to be verified before starting to output the plaintext during decryption (for example because the amount of memory is very small).

At FSE 2014, Abed *et al.* [2] proposed a scheme named POET, based on the POE family of online ciphers which are provably secure against chosen-ciphertext attacks (POE is itself based on

a block cipher). This proposal contains a proof which stipulates that POET is a provably secure authenticated encryption scheme. Moreover, it only requires a single encryption per message block on average. Another advantage of this scheme is that it is pipelinable and thus allows more efficient implementations compared to fully sequential designs such as McOE.

Our contributions. In this article, we show that POET (described in Section 1) is not a secure authenticated encryption scheme, even when not used in nonce-misuse nor decryption-misuse scenario. Our attack uses only a single query to the encryption oracle and a negligible amount of computations. The whole process is described in Section 2 and it allows to generate many valid ciphertext/tag pair and therefore breaks the authenticity property of the POET authenticated encryption mode. Then, we propose in Section 3 a potential simple patch to overcome this issue and hopefully recover the entire authenticity property expected for an authenticated encryption scheme.

1 Description of POET

POET is an authenticated encryption scheme proposed at FSE 2014 [2]. Even though it is based on POE family of online ciphers, we will describe POET directly. We denote $E_K(P)$ the encryption of the plaintext P with the n-bit block cipher E initialized with the k-bit key K (and $D_K(C)$ will denote the decryption process of the ciphertext C with the key K). Furthermore, we denote $F_K(\cdot)$ the ϵ-AXU family F of n-bit hash functions parameterized by K.

POET encryption takes as input a variable-length message M and a variable-length header H, while it outputs a ciphertext C with $|C| = |M|$ and a tag value T. POET decryption takes as input a variable-length ciphertext C, a variable-length header H, a tag value T, and it outputs either a plaintext M with $|M| = |C|$ or an error character $\perp$ in case the authenticity verification failed. Without loss of generality, we will assume in the rest of the article that the length of the messages is always a multiple of n and we denote m the number of message blocks, i.e. $|M| = m \cdot n$. The notation M_i will refer to the i-th n-bit block of message ($M = M_1||\dots||M_m$). We will also assume that the header length as well as the tag length is exactly one n-bit block. Our attack is completely independent of these assumptions, but they simplify its description.

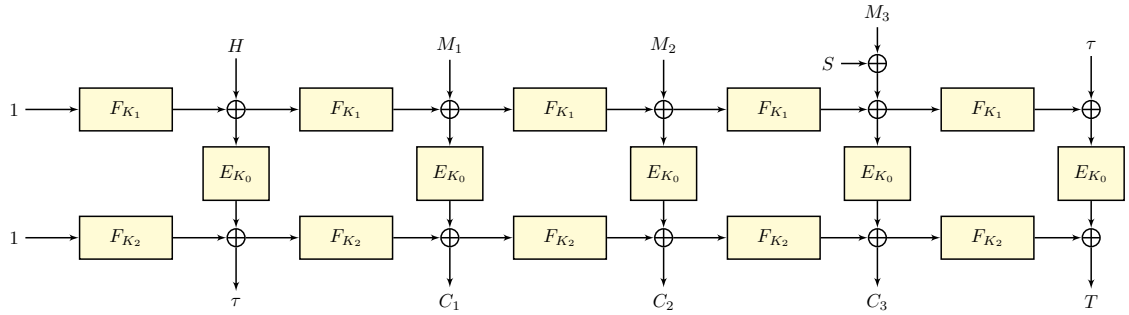

Fig. 1. The POET authenticated encryption mode.

A picture of POET encryption process is given in Figure 1. The keys K_0, K_1 and K_2 are generated from the master key K, but we omit the method since it is not important for our attack (we will consider them to be three independent k-bit keys). The internal state of POET is composed of two n-bit values X and Y (respectively the upper and lower line) and we denote X_i and Y_i the values before handling message block M_i. First, the header block is processed with $X_0 = F_{K_1}(1) \oplus H$ and $Y_0 = F_{K_2}(1) \oplus E_{K_0}(X_0)$, and a pretag value τ is memorized $\tau = Y_0$. Then the message blocks are processed with $X_{i+1} = F_{K_1}(X_i) \oplus M_i$ and $Y_{i+1} = F_{K_2}(Y_i) \oplus E_{K_0}(X_{i+1})$, and the ciphertext C_i is simply $C_i = Y_{i+1}$. Only the last message block M_m is treated differently as

$X_{m+1} = F_{K_1}(X_m) \oplus M_m \oplus E_{K_0}(|M|)$ and $Y_{m+1} = F_{K_2}(Y_m) \oplus E_{K_0}(X_{m+1})$, and the ciphertext C_m is simply $C_m = Y_{m+1}$. Finally, once all the message blocks processed, the tag is computed with $X_{m+2} = F_{K_1}(X_{m+1}) \oplus \tau$ and $T = F_{K_2}(Y_{m+1}) \oplus E_{K_0}(X_{m+2})$.

We omit the POET verification/decryption part here, but we refer to [2] for a complete description of the process.

2 The attack

Our attack is very simple: the idea is to first query a message M composed of several blocks to the encryption oracle and obtain a ciphertext C and a tag T, and then to observe that this T is a valid tag for a new (and yet not queried) ciphertext build by adding any difference to the any block but the last two. This can be seen in Figure 2.

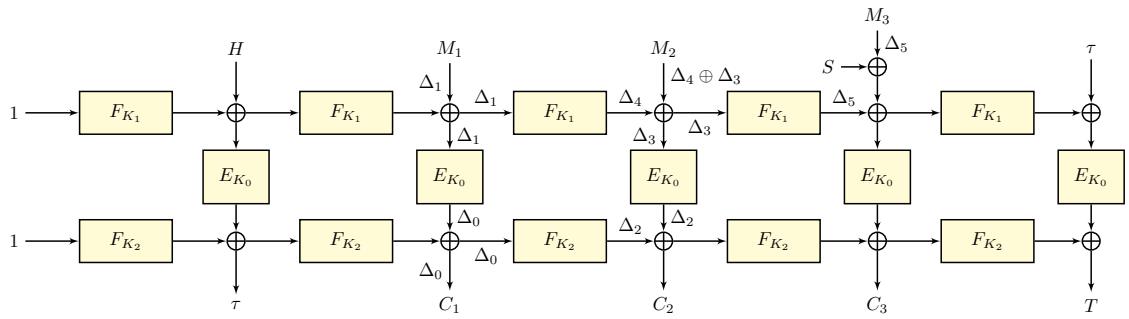

Fig. 2. The single query attack on POET authenticated encryption mode. A difference Δ_0 is inserted on C_1, but no difference is inserted on C_2 and C_3.

In more details: we first pick a random message M composed of three n-bit blocks $M = M_1||M_2||M_3$ and a random header block H. We then query this pair (M, H) to the encryption oracle and obtain a ciphertext $C = C_1||C_2||C_3$ and a tag T. Any pair (C', T) with $C' = C'_1||C'_2||C'_3 = C_1 \oplus \Delta_0||C_2||C_3$ and Δ_0 a random n-bit value is a valid decryption pair, which breaks the authenticity property. We denote X'_i and Y'_i the internal state values when processing ciphertext C' (or corresponding message M'). Moreover, $\delta(X_i)$ will represent the XOR difference between X_i and X'_i (or Y_i and Y'_i), i.e. $\delta(X_i) = X_i \oplus X'_i$.

Now, let us explain why tag T is valid for ciphertext C'. Since $\delta(C_2) = 0$, it means that $\delta(Y_3) = 0$ because $C_2 = Y_3$. Similarly, since $\delta(C_3) = 0$, it means that $\delta(Y_4) = 0$ because $C_3 = Y_4$. Moreover, $C_3 = F_{K_2}(Y_3) \oplus E_{K_0}(X_4)$, so we directly deduce that $\delta(X_4) = 0$. As $\delta(X_4) = \delta(Y_4) = 0$, the entire internal state (X_4, Y_4) does not contain any difference and it is obvious that this will lead to a collision on the tag value. Finally, with a negligible amount of computations, we are able to generate almost any number of valid ciphertext/tag pairs by asking only for single encryption query that must composed of at least 3 blocks.

It is to be noted that we picked a random header value at the beginning of the attack. Therefore, even if a random nonce is inserted in the header, this will not change anything to our technique. Our attack is not based on any misuse from the adversary (like nonce-misuse or decryption-misuse) and works in the classical adversary scenario. Moreover, note also that we can similarly apply the technique on the header part of POET (again only if at least 3 blocks of header are processed) or one both the header and message part (by performing two separate state collisions).

3 Patching POET

Our attack uses the fact that each ciphertext output C_i is equal to the outcoming Y_{i+1} value, due to a structural weakness of the design. In order to avoid this issue, we propose to use a different mixing function than $C_i = Y_{i+1} = F_{K_2}(Y_i) \oplus E_{K_0}(X_{i+1})$, so that C_i and Y_{i+1} are really made distinct, and in a $GF(2)$-non-linear way. The following update function seems to be quite efficient and prevents our attack: $C_i = 2 \cdot F_{K_2}(Y_i) \oplus E_{K_0}(X_{i+1})$ and $Y_{i+1} = F_{K_2}(Y_i) \oplus E_{K_0}(X_{i+1})$. We note that the idea of using a linear mixing was proposed by Datta *et al.* in [4], with the goal of obtaining a parallel online authenticated encryption. This proposal, ELmE, uses a linear mixing function $\rho(x, y) = (x + (\alpha + 1) \cdot y, x + \alpha \cdot w)$, where α is a primitive element of the field $GF(2^n)$.

We emphasize that this patch is a only proposal to avoid the attack presented in this article. While we are confident that it should not harm the basic POET design, a new proper security proof taking this update in account should be provided in order to confidently use this scheme.

Conclusion

In this article we have shown that POET is not a secure authenticated encryption mode, as authenticity notion can be easily broken with only a single encryption query. Therefore, POET should not be used in applications where authentication is a requirement. Patching POET with regards to this issue seems feasible, but we leave as future work a possible security proof.

Acknowledgments

The authors are supported by the Singapore National Research Foundation Fellowship 2012 (NRF-NRFF2012-06).

References

1. CAESAR Competition. http://competitions.cr.yp.to/caesar.html.
2. Farzaneh Abed, Scott Fluhrer, John Foley, Christian Forler, Eik List, Stefan Lucks, David McGrew, and Jakob Wenzel. Pipelineable On-Line Encryption. In *FSE*, 2014, preproceedings version.
3. Elena Andreeva, Andrey Bogdanov, Atul Luykx, Bart Mennink, Elmar Tischhauser, and Kan Yasuda. Parallelizable and Authenticated Online Ciphers. In Kazue Sako and Palash Sarkar, editors, *ASIACRYPT (1)*, volume 8269 of *Lecture Notes in Computer Science*, pages 424–443. Springer, 2013.
4. Nilanjan Datta and Mridul Nandi. Misuse Resistant Parallel Authenticated Encryptions. *IACR Cryptology ePrint Archive*, 2013:767, 2013.
5. Ewan Fleischmann, Christian Forler, and Stefan Lucks. McOE: A Family of Almost Foolproof On-Line Authenticated Encryption Schemes. In *FSE*, pages 196–215, 2012.
6. Phillip Rogaway and Thomas Shrimpton. A Provable-Security Treatment of the Key-Wrap Problem. In *EUROCRYPT*, pages 373–390, 2006.