Misuse-resistant cryptography for JOSE/JWT

Neil Madden
ForgeRock
neil.madden@forgerock.com

February 14, 2018

Abstract

The JSON Object Signing and Encryption (JOSE) standards[15, 17] are widely used in the OAuth 2.0 community, both in standards building on top of the core protocol, such as OpenID Connect, and behind-the-scenes in some implementations as "stateless" access and refresh token formats. In some cases, such tokens may be encrypted to protect system internal data or to preserve privacy expectations. The existing encryption methods[14] for JOSE (AES-CBC with HMAC or AES-GCM) require either a random IV or a unique *nonce*, and fail to maintain security properties if these requirements are violated. In the worst case, AES-GCM completely sacrifices authenticity if a nonce is reused just once. In high-volume multi-server environments this can become a serious possibility. This position paper argues for the adoption of *nonce reuse misuse-resistant authenticated encryption* (MRAE) for JOSE, and in particular for JSON Web Token (JWT)[16] usage. We describe a generic approach to MRAE known as Synthetic IV (or SIV)[22, 13], and discuss its desireable properties for OAuth 2.0 use-cases. SIV is also suitable as a replacement for AES Key-Wrap.

Introduction

The JSON Object Signing and Encryption (JOSE)[15, 17] and JSON Web Token (JWT)[16] standards have seen widespread adoption in the OAuth 2.0 community. OpenID Connect specifies the use of JWTs for representing ID Tokens, but they have also seen use as a "stateless" format for access and refresh tokens, session cookies, and other security tokens. Where such tokens contain information that should be kept private from the receiving party (e.g., client), such as private information about the resource owner, or where the token must pass through untrusted environments (e.g., in an emailed link), it is desireable to encrypt such information rather than just authenticate it. JOSE supports two main encryption methods, based on either AES-CBC with a random IV with a separate HMAC ([14] section 5.2), or AES-GCM with a nonce (section 5.3). In both cases, failure to observe the requirements for unpredictable IVs (for CBC) or unique nonces (for GCM) can result in a significant loss of confidentiality, integrity, or both. For CBC, predictable IVs may result in vulnerability to *chosen plaintext attacks* like the BEAST attack on SSL/TLS[7], that allow an attacker to recover the plaintext of an encrypted message. For GCM, the situation is even more serious as reusing a nonce will both leak the XOR of two plaintexts (exposing them to statistical analysis) and allow an attacker to recover the authentication sub-key, allowing forgeries to be constructed (see [10], Appendix A). In the worst case this can result in a complete loss of both confidentiality and integrity, as shown by the KRACK attacks on nonce-reuse against WPA2[23].

At first glance, it may seem that it is easy to ensure these requirements are satisfied. For CBC we can just generate random IVs using a suitable cryptographically secure random number generator (CSPRNG), while for GCM we can either do the same or use a simple deterministic counter that is guaranteed never to repeat. However, both approaches have proved problematic in practice. For some embedded devices, access to good sources of entropy can be hard to come by, making it difficult to reliably generate unpredictable IVs. For example, a weakness in the SecureRandom implementation on Android phones allowed money to be stolen from Bitcoin wallets on those devices[5].

For GCM, random nonce generation is further complicated by the small maximum size of 96 bits. As recommended by NIST[10] (section 8.3), when using a random IV with AES-GCM it should not be used for more than 2^{32} messages (just over 4 billion). While this sounds like a lot, in an application generating a thousand JWTs per second (not out of the question for a high volume site, especially if DDoS attacks are a threat) this limit would be reached in less than 50 days. At 50,000 per second the limit would be reached in less than a day. There are two possible ways to improve this limit. One is to use a deterministic nonce generation scheme, for instance a counter. However such counters are hard to efficiently synchronize across multiple servers, especially when restarts may occur. A second approach is to use GCM in combination with AES Key Wrap (or some other key-wrapping or key-derivation approach) so that a fresh key is used for each message. This effectively increases the random input for each message from the 96-bit nonce to a 224-bit key+nonce pair (assuming 128-bit keys). While this solves the issue when a reliable source of entropy is available, it does so at a cost of extra processing and expansion of the resulting ciphertext to include the JWE wrapped key.

Misuse Resistant Authenticated Encryption

A solution to these problems has been developed based on the notion of (nonce reuse) *misuse-resistant authenticated encryption* (MRAE)[22][1]. Such encryption schemes were developed to provide the maximum amount of security even when a nonce is accidentally (or deliberately) reused. In such a case, a MRAE scheme sacrifices only a small amount of confidentiality and loses no integrity at all. An attacker may be able to tell if the same message was encrypted using the same key and nonce, but learns no other information. This is a significant improvement over the traditional CBC and GCM schemes currently used in JOSE.

The main drawback of MRAE schemes is that it can be proved that they must necessarily be two-pass *offline* modes. That is, the full plaintext must be processed before any encrypted ciphertext can be generated. This means that such schemes will always be slower than dedicated one-pass online modes like GCM. However, for short messages such as JWTs this overhead is negligible and most existing libraries do already buffer the full token in memory while encrypting it. In local tests, there is negligible performance overhead for using an MRAE mode for encrypting short tokens. In addition, decryption and verification is typically just as fast as other modes ('online' decryption is also possible, but it is dangerous to process decrypted plaintext from any authenticated encryption scheme before the authentication tag has been validated).

Synthetic IV

The original paper defining MRAE[22], also introduced a new block cipher mode of operation known as *Synthetic IV* or *SIV* for short, together with a concrete instantiation of the mode for AES using AES-CMAC for authentication and AES-CTR for encryption. This concrete mode has also been published as RFC 5297[13]. Conceptually, the SIV mode is quite simple. First, the plaintext (and any associated data) is passed through a psuedorandom function (PRF, typically a MAC) to create the synthetic IV. Secondly, the plaintext is encrypted using an IV-based encryption mode using the SIV as the IV. This description glosses over some subtleties that are essential to the security proof, so should only be taken as a sketch.

As presented above, the scheme would be completely deterministic (assuming a deterministic PRF). Deterministic encryption schemes do not meet standard security goals, such as semantic security under a chosen plaintext attack. For example, an attacker can trivially determine if the same message has been encrypted twice using the same key. However, this can be rectified by simply adding a random component to the associated data, effectively acting as a nonce. So long as this random nonce does not repeat then semantic security is preserved. If a nonce is repeated, then the slightly weaker security notion of deterministic authenticated encryption (DAE; [22]) is achieved instead, where an attacker can tell if an entire duplicate message has been encrypted with the same key and nonce but otherwise learns no further information. SIV provably meets the MRAE goal when used in this way.

For use in JOSE, SIV can be employed in two ways:

[1] MRAE and the SIV mode discussed in this section were initially developed in response to NIST's development of the AES KeyWrap algorithms already used in JWE.

1. As a MRAE Content Encryption method making use of a random JWE Initialization Vector, included as one component of the associated data passed to the PRF.
2. Without a random nonce, as an alternative to the AES KeyWrap JWE algorithms. SIV was initially developed to solve the key-wrap problem, and has a number of advantages over the NIST specification.

Original AES-SIV

The original SIV paper, and the RFC based on it[13], develops a concrete encryption scheme using AES. The PRF is formed using AES-CMAC[9], followed by AES in CTR mode for encryption[8]. The scheme is fully described in the RFC and includes a number of performance optimisations, such as the S2V construction for efficiently converting a PRF that takes a single input to one that takes a vector of inputs without requiring an encoding step.

This construction is simple and efficient, and has the advantage of only requiring an AES encryption circuit to perform encryption, decryption, authentication and verification. It is therefore particularly suitable for use on constrained devices, a property it shares with AES-CCM. It is also fast on any machine that has hardware accelation for AES.

A downside of this construction is that it is limited to producing 128-bit SIVs and therefore cannot provide more than 128-bit authentication level (although this should be adequate for most JWT uses).

Other Variations

While the original SIV paper only defined the AES-CMAC/CTR construction described above, the scheme is more general and so alternative combinations can be defined, so long as the PRF and encryption mode match the requirements of the security proofs. For instance, HMAC-SHA2 can be used in place of AES-CMAC, providing authentication security beyond 128-bits. A recently developed library for MRAE known as *Miscreant*[1] implements a variant using AES-PMAC (parallel MAC) to achieve a speedup over CMAC through parallelism (AES-CTR is already fully parallelisable). A proposal for a very fast SIV scheme based on AES-GCM, called AES-GCM-SIV[12], is currently progressing to RFC status in the IRTF CFRG[11].

Non-AES instantiations are also possible. For instance, a combination of the keyed Blake2s hash function[2] with the XChaCha20 'extended-nonce' stream cipher[6] could provide a fast and timing side-channel resistant option, able to encrypt enormous amounts of data with a single key. This combination should satisfy the requirements of the SIV security proofs, but a rigorous analysis has not been performed.

Internet Draft

An Internet Draft has been created proposing to add some SIV-based modes to JOSE[20]. The draft currently proposes to add 4 new JWE Encryption Methods based on SIV with either CMAC or HMAC-SHA2, see Table 1, and 4 new JWE Algorithms based on (deterministic) key-wrapping using SIV without a nonce. In each case the input to the algorithm is a key composed of two parts: a PRF/MAC key and an encryption key (exactly as for the existing CBC-HMAC modes), a random 128-bit JWE Initialization Vector (omitted for the key-wrap algorithms), the Additional Authenticated Data (i.e.,the JWE Encoded Protected Header and optionally the JWE AAD), and the JWE Plaintext. The SIV is calculated by passing the components into the MAC algorithm (using a simple encoding instead of the S2V construction). Finally the plaintext is encrypted using AES in CTR mode using the SIV as the initial counter value to produce the JWE Ciphertext. The SIV is then output as the Authentication Tag.

What about other JOSE algorithms?

Beyond authenticated encryption methods, we might ask what other cryptographic primitives used in JOSE might also be subject to errors due to misuse? It is not feasible to exhaustively analyse all of the options available in JOSE, so we just give a quick sketch here of how nonce or random parameter misuse might affect other JOSE algorithms.

"enc" Value	Description
A128SIV	AES-SIV as in [13] with CMAC and CTR mode and 256-bit key.
A128SIV-HS256	SIV using HMAC-SHA-256-128 and AES-CTR with a 256-bit key.
A192SIV-HS384	SIV using HMAC-SHA-384-192 and AES-CTR with a 384-bit key.
A256SIV-HS512	SIV using HMAC-SHA-512-256 and AES-CTR with a 512-bit key.

Table 1: Proposed JWE Encryption Methods

Regarding JWE algorithms for key management, AES-GCM key-wrapping is obviously subject to the same risks described in this paper for content encryption. The remaining symmetric options – AES Key Wrap, direct encryption, and PBES2 password-based encryption – should be largely secure against misuse. Where PBES2 salts might repeat, this can be hedged somewhat by increasing the iteration count. SIV can be used in place of both GCM and AES-KW for key-wrapping and in this case provides an efficient algorithm with provable security properties and a greater authentication security level. The asymmetric options are RSA with either PKCS#1 v1.5 padding or OAEP, or ECDH-ES, all of which require fresh random values for each message. Recent work on *hedged public key encryption*[3, 4] has investigated how to make public key encryption more robust if such random values are compromised, providing a counterpart to MRAE in the asymmetric setting.

The remaining candidate for examination is digital signatures, which in JOSE means either RSA or ECDSA. As described previously in the impact of SecureRandom failures on Android Bitcoin wallets, ECDSA is also vulnerable if nonces are reused or predictable, potentially allowing recovery of the private key. Thankfully, there are already mitigations available. Firstly, RFC 8037 [19] specifies how to use the Ed25519 and Ed448 elliptic curve digital signature algorithms with JWS. These algorithms do not suffer from the same problem. Secondly, RFC 6979 [21] specifies how to perform ECDSA signing using a deterministic algorithm, similar in spirit to the SIV construction. Neither RSA signing algorithm is susceptible to misuse in this sense, as PKCS#1 signatures are deterministic and PSS signatures are secure even if salt values are repeated (see [18], section 8.1).

Conclusions

In this paper we have discussed how the symmetric encryption methods provided by JOSE may lose security when random IVs are predictable or nonces are reused. In some cases, such security loss can be catastrophic, completely undermining both confidentiality and integrity/authenticity. As a solution to these issues, we recommend the introduction of nonce-reuse misuse-resistant authenticated encryption (MRAE) modes, and the Synthetic IV (SIV) construction in particular. We also recommend that similar approaches are investigated for asymmetric encryption algorithms.

References

[1] Tony Arcieri. Introducing Miscreant: a multi-language misuse resistant encryption library. https://tonyarcieri.com/introducing-miscreant-a-multi-language-misuse-resistant-encryption-library, 2017.

[2] J-P. Aumasson. The BLAKE2 cryptographic hash and message authentication code (MAC). RFC 7693, IETF, November 2015.

[3] M. Bellare, Z. Brakerski, M. Naor, T. Ristenpart, G. Segev, H. Shacham, and S. Yilek. Hedged public-key encryption: How to protect against bad randomness. In *ASIACRYPT 2009*, volume 5912 of *LNCS*, pages 232–249. Springer, December 2009.

[4] M. Bellare and B. Tackmann. Nonce-based cryptography: Retaining security when randomness fails. In *Advances in Cryptology—EUROCRYPT 2016*, volume 9665 of *LNCS*. Springer, 2016.

[5] bitcoin.org. Android security vulnerability. `https://bitcoin.org/en/alert/2013-08-11-android`, 2015.

[6] Frank Denis. XChaCha20. `https://download.libsodium.org/doc/advanced/xchacha20.html`, 2018.

[7] Thai Duong. BEAST. `https://vnhacker.blogspot.co.uk/2011/09/beast.html`, 2011.

[8] Morris Dworkin. Recommendation for block cipher modes of operation: Methods and techniques. Special Publication 800-38A, NIST, December 2001.

[9] Morris Dworkin. Recommendation for block cipher modes of operation: The CMAC mode for authentication. Special Publication 800-38B, NIST, May 2005.

[10] Morris Dworkin. Recommendation for block cipher modes of operation: Galois/counter mode (GCM) and GMAC. Special Publication 800-38D, NIST, November 2007.

[11] S. Gueron, A. Langley, and Y. Lindell. AES-GCM-SIV: Nonce misuse-resistant authenticated encryption. Internet Draft draft-irtf-cfrg-gcmsiv-08, IETF, February 2018. `https://tools.ietf.org/html/draft-irtf-cfrg-gcmsiv-08`.

[12] Shay Gueron, Adam Langley, and Yehuda Lindell. AES-GCM-SIV: Specification and analysis. Cryptology ePrint Archive, Report 2017/168, 2017. `https://eprint.iacr.org/2017/168`.

[13] D. Harkins. Synthetic initialization vector (SIV) authenticated encryption using the advanced encryption standard (AES). RFC 5297, IETF, October 2008.

[14] M. Jones. JSON Web Algorithms (JWA). RFC 7518, IETF, May 2015.

[15] M. Jones, J. Bradley, and N. Sakimura. JSON Web Signature (JWS). RFC 7515, IETF, May 2015.

[16] M. Jones, J. Bradley, and N. Sakimura. JSON Web Token (JWT). RFC 7519, IETF, May 2015.

[17] M. Jones and Hildebrand J. JSON Web Encryption (JWE). RFC 7516, IETF, May 2015.

[18] B. Kaliski, J. Jonsson, and A. Rusch. PKCS #1: RSA cryptography specifications version 2.2. RFC 8017, IETF, November 2016.

[19] I. Liusvaara. CFRG elliptic curve Diffie-Hellman (ECDH) and signatures in JSON object signing and encryption (JOSE). RFC 8037, IETF, January 2017.

[20] Neil Madden. Synthetic IV (SIV) encryption modes for JWE. Internet Draft draft-madden-jose-siv-mode-02, IETF, December 2017. `https://tools.ietf.org/html/draft-madden-jose-siv-mode-02`.

[21] T. Pornin. Deterministic usage of the digital signature algorithm (DSA) and elliptic curve digital signature algorithm (ECDSA). RFC 6979, IETF, August 2013.

[22] Phillip Rogaway and Thomas Shrimpton. Deterministic authenticated-encryption: A provable-security treatment of the key-wrap problem. In *Advances in Cryptology—EUROCRYPT '06*, volume 4004 of *Lecture Notes in Computer Science*, pages 373–390. Springer, 2006.

[23] Mathy Vanhoef and Frank Piessens. Key reinstallation attacks: Forcing nonce reuse in WPA2. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security—CCS '17*, pages 1313–1328, Dallas, TX, USA, October–November 2017.