Key commitment and Multi Key Collision Attacks

Uri Ariel Yahalom* Shay Gueron†

Abstract

In this report we examine key commitment in Authenticated encryption and vulnerabilities that arise when AEAD is misuded and robustness is assumed. Recent research shows new class of attacks such as partitioning oracles which arise when AE schemes that are not committing with respect to their keys are misused. As key commitment is not part of AE's design goal, AE schemes in general do not satisfy it.
We show an adaptive chosen ciphertext attack that utilizes key multi-collision against widely used AEAD schemes, including AES-GCM, XSalsa20/Poly1305, and ChaCha20/Poly1305, to build a partitioning oracle.
We also discuss an early implementation of the OPAQUE protocol (a protocol for password-based key exchange) that is vulnerable to partitioning oracle.

*Department of Computer Science, University of Haifa.
†Department of Management, University of Haifa.

1 Background

Authenticated Encryption

Authenticated Encryption (AE)[McGrew, 2008] is a well-studied primitive which avoids the pitfalls of unauthenticated Symmetric-key encryption with relatively small performance overhead AE schemes are used in widely adopted protocols and are the default Symmetric option in modern cryptographic libraries.
The design of encryption historically separated the goals of confidentiality and authenticity, which led to widespread deployment of encryption schemes vulnerable to chosen ciphertext attacks (CCA).
Subsequently, researchers showed how to exploit CCAs to recover plaintext data, most notably via padding and format oracle attacks As a result, cryptographers now advocate the use of authenticated encryption with associated data (AEAD) schemes and CCA-secure public key encryption.
There has since been a shift to adopt fast CCA-secure schemes, notably AES-GCM, and XSalsa20/Poly1305.

AES-GCM

AES-GCM is an AEAD scheme that composes AES in counter mode with a specially designed MAC. It uses an XOR-universal hash function called GHASH. Encryption takes in a nonce N, an AES key K, associated data AD, and plaintext $M_1, \ldots, M_n$ without loss of generality (AES-GCM specification handles various length). and it outputs a ciphertext $C_1, \ldots, C_n$ and an authentication tag T. The ciphertext blocks $C_1, \ldots, C_n$ are generated using counter mode with E, and the tag T is computed by applying GHASH to AD and $C_1, \ldots, C_n$. Decryption also computes the tag, compares it with T, and, if successful, outputs the counter-mode decryption of the ciphertext blocks.

2 Key Commitment

Key commitment guarantees that a ciphertext can only be decrypted under the same key it was encrypted with. If we are to find a ciphertext that decrypts into two valid plaintexts under two different keys do not commit with respect to the key.
Key commitment was initially studied by[Farshim et al., 2017] and while it might seem like an academic pursuit, [Dodis et al., 2018] and [Grubbs et al., 2017] show how to exploit AEADs that do not commit to their key. Nevertheless, AEAD key commitment has not received the required attention and can be overlooked in deployment.
A committing encryption scheme is a scheme for which it is computationally unfeasible to find two keys and a ciphertext that decrypts under both. Security standards for committing AEADs were first formalized by[Farshim et al., 2017].

3 Multi Key Collision Attacks

Multi Key Collision Attacks is an attack where we find a ciphertext - C that decrypts under k different keys $K_1, \ldots, K_k$ to k different valid plaintexts $P_1, \ldots, P_k$.
In practice, to perform a multi key collision attack against AES-GCM we need to solve a system of linear equations. This is possible because of the algebraic properties of the universal hashing in AES-GCM.

4 Partitioning Oracles

Thus far, key commitment has not been considered an essential security goal for most cryptographic applications. This is perhaps because attacks exploiting key collision have arisen in relatively niche applications like message franking for encrypted messages.
[**?**] introduces partitioning oracle attacks, a new type of CCA in password-authenticated key exchange (PAKE). A partitioning oracle arises when an attacker can:

1. efficiently prepare ciphertexts that will decrypt under a large number of keys
2. submit those ciphertexts to an oracle that tells whether decryption succeeds or fails.

This enables gaining information about the password.

Future Work

In the future we plan to build a partitioning oracle on one the rust implementation of OPAQUE[gustin, 2019]

References

References

[Dodis et al., 2018] Dodis, Y., Grubbs, P., Ristenpart, T., and Woodage, J. (2018). *Fast Message Franking: From Invisible Salamanders to Encryptment*, pages 155–186. Lecture Notes in Computer Science. Springer.

[Farshim et al., 2017] Farshim, P., Orlandi, C., and Rosie, R. (2017). Security of symmetric primitives under incorrect usage of keys. *IACR Cryptol. ePrint Arch.*, 2017:288.

[Grubbs et al., 2017] Grubbs, P., Lu, J., and Ristenpart, T. (2017). Message franking via committing authenticated encryption. Cryptology ePrint Archive, Report 2017/664. https://eprint.iacr.org/2017/664.

[gustin, 2019] gustin (2019). opaque. https://github.com/gustin/opaque.

[McGrew, 2008] McGrew, D. (2008). An Interface and Algorithms for Authenticated Encryption. RFC 5116.

Appendix