Dumbo, Jumbo, and Delirium: Parallel Authenticated Encryption for the Lightweight Circus

(proposal for presentation)

Tim Beyne[1], Yu Long Chen[1], Christoph Dobraunig[2], and Bart Mennink[2]

[1] imec-COSIC, KU Leuven, Belgium
tim.beyne@esat.kuleuven.be, yulong.chen@kuleuven.be
[2] Digital Security Group, Radboud University, Nijmegen, The Netherlands
c.dobraunig@cs.ru.nl, b.mennink@cs.ru.nl

Abstract. With the trend to connect more and more devices to the Internet, authenticated encryption has become a major backbone in securing the communication, not only between these devices and servers, but also the direct communication among these devices. Most authenticated encryption algorithms used in practice are developed to perform well on modern high-end devices, but are not necessarily suited for usage on resource-constrained devices. We present a lightweight authenticated encryption scheme, called Elephant. Elephant retains the advantages of GCM such as parallelism, but is tailored to the needs of resource-constrained devices. The two smallest instances of Elephant, Dumbo and Jumbo, are based on the 160-bit and 176-bit Spongent permutation, respectively, and are particularly suited for hardware; the largest instance of Elephant, Delirium, is based on 200-bit Keccak and is developed towards software use. All three instances are parallelizable, have a small state size while achieving a high level of security, and are constant time by design.

1 Introduction

Authenticated encryption has become an integral part of our modern communication infrastructure. Considering the rise of the Internet of Things, the usage will not only expand, but will also require that authenticated encryption algorithms run on resource-constrained devices. Many modern cryptographic protocols like TLS [18] or the Signal protocol [7,17] rely at their core on authenticated encryption. For instance, TLS 1.3 [18] relies on AES-GCM, or ChaCha20 with Poly1305, whereas in the Signal protocol [7, 17], the task of authenticated encryption can be performed using AES in CBC mode for encryption paired with HMAC-SHA-2 for authentication. While the performance of these constructions may be sufficient on modern high-end systems, they have inadvertently some drawbacks for the usage in lightweight systems.

A first drawback is the use of components such as the AES [8], ChaCha [3], and SHA-2 [9], which were not designed with lightweight applications in mind.

Moreover, ChaCha and SHA-2 make extensive use of modular additions, which is not the best choice for lightweight hardware implementations. A second problem is the need for the implementation of two different primitives (one for encryption and one for authentication) for performing the single task of authenticated encryption, which is a potential waste of resources in lightweight applications. This is still true if the primitives within these constructions are replaced with more lightweight counterparts. Furthermore, the usage of lightweight 64-bit block ciphers for the aforementioned mode implies stringent restrictions on the amount of data that can be safely encrypted [5, 15]. The need for authenticated encryption schemes that perform well on resource-constrained devices has recently been addressed by NIST's call for lightweight authenticated encryption schemes [16]. The call specifies a request for authenticated encryption schemes having at least 112-bit security provided that the online complexity is at most around 2^{50} bytes.

To provide an alternative for lightweight applications, we introduce the authenticated encryption scheme Elephant. The mode of Elephant is a nonce-based encrypt-then-MAC construction, where encryption is performed using counter mode and message authentication using a variant of the Wegman-Carter-Shoup MAC [2, 21, 22]. Both modes use a cryptographic permutation masked using LFSRs, akin to the masked Even-Mansour construction of Granger et al. [11].

The mode is permutation-based and only evaluates this permutation in the forward direction. As such, there is no need to implement multiple primitives or the inverse of the primitive, unlike in OCB-based [14, 19, 20] authenticated encryption schemes. Furthermore, this allows us to rely and build on the extensive literature of permutations used for sponge-based lightweight hashing [1, 6, 12]. That said, Elephant itself is not sponge-based: on the contrary, it departs from the conventional approach of serial permutation-based authenticated encryption. Elephant is parallelizable by design, easy to implement due to the use of LFSRs for masking (no need for finite field multiplication), and finally, it is efficient due to elegant decisions on how the masking should be performed exactly. A security analysis in the ideal permutation model demonstrates that the mode of Elephant is structurally sound.

Due to the parallelizability of Elephant, there is no need for instances with a large permutation: we can go as small as 160-bit permutations while still matching the security goals recommended by the NIST lightweight call [16]. In detail, the Elephant scheme consists of three instances:

1. Dumbo: Elephant-Spongent-π[160]. This instance meets the minimum permutation size as dictated by the security analysis: it achieves 112-bit security provided that the online complexity is at most around 2^{46} blocks. This instance is particularly well-suited for hardware, as Spongent [6] itself is;
2. Jumbo: Elephant-Spongent-π[176]. This is a slightly more conservative instance of Elephant: it is based on the same permutation family, yet achieves 127-bit security under the same conditions on the online complexity. We note, in particular, that Spongent-π[176] is ISO/IEC standardized [6, 13];
3. Delirium: Elephant-Keccak-f[200]. This variant is developed more towards software use, although it still performs reasonably well in hardware. Elephant

instantiated with Keccak-$f[200]$ also achieves 127-bit security, with a higher bound of around 2^{70} blocks on the online complexity. The permutation is the smallest instance of the NIST SHA-3 standard [4,10] that fits our needs.

Dumbo and Jumbo are named after two famous elephants; Delirium is named after a Belgian beer, whose logo is a pink elephant. As each of the permutations is relatively small, all versions of Elephant have a small state size, despite its support for parallelism. The LFSRs used for masking are tailored to the specific instance, one for each, and are developed to operate well with the specific cryptographic permutation. For example, the LFSRs paired with the Spongent instances have been chosen to minimize the number of XOR operations that have to be performed for a state-update, while the Keccak-based instance has been selected to perform well on software platforms.

We note that the three cryptographic permutations in Elephant can also be used for cryptographic hashing – in fact, Spongent [6] and Keccak [4] themselves are sponges – but due to our quest for small permutations, these cryptographic hash functions cannot meet the 112-, or 127-bit security level guaranteed by our authenticated encryption schemes. In contrast, in order to perform sponge-based hashing with at least 112-bit security, a cryptographic permutation of size at least 225 bits must be used.

Acknowledgments. This work was supported in part by the Research Council KU Leuven: GOA TENSE (C16/15/058). Yu Long Chen is supported by a Ph.D. Fellowship from the Research Foundation - Flanders (FWO). Christoph Dobraunig is supported by the Austrian Science Fund (FWF): J 4277-N38. Bart Mennink is supported by a postdoctoral fellowship from the Netherlands Organisation for Scientific Research (NWO) under Veni grant 016.Veni.173.017.

References

1. Aumasson, J., Henzen, L., Meier, W., Naya-Plasencia, M.: Quark: A Lightweight Hash. In: Mangard, S., Standaert, F. (eds.) CHES 2010. LNCS, vol. 6225, pp. 1–15. Springer (2010)
2. Bernstein, D.J.: Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 164–180. Springer (2005)
3. Bernstein, D.J.: Chacha, a variant of salsa20. Online Document: `https://cr.yp.to/chacha/chacha-20080128.pdf` (Jan 2008)
4. Bertoni, G., Daemen, J., Peeters, M., Van Assche, G.: The Keccak reference (January 2011)
5. Bhargavan, K., Leurent, G.: On the Practical (In-)Security of 64-bit Block Ciphers: Collision Attacks on HTTP over TLS and OpenVPN. In: Weippl, E.R., Katzenbeisser, S., Kruegel, C., Myers, A.C., Halevi, S. (eds.) ACM SIGSAC 2016. pp. 456–467. ACM (2016)
6. Bogdanov, A., Knezevic, M., Leander, G., Toz, D., Varici, K., Verbauwhede, I.: Spongent: A Lightweight Hash Function. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 312–325. Springer (2011)

7. Cohn-Gordon, K., Cremers, C.J.F., Dowling, B., Garratt, L., Stebila, D.: A Formal Security Analysis of the Signal Messaging Protocol. In: EuroS&P 2017. pp. 451–466. IEEE (2017)
8. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Information Security and Cryptography, Springer (2002)
9. FIPS 180-4: Secure Hash Standard (March 2012)
10. FIPS 202: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions (August 2015)
11. Granger, R., Jovanovic, P., Mennink, B., Neves, S.: Improved Masking for Tweakable Blockciphers with Applications to Authenticated Encryption. In: Fischlin, M., Coron, J. (eds.) EUROCRYPT 2016, Part I. LNCS, vol. 9665, pp. 263–293. Springer (2016)
12. Guo, J., Peyrin, T., Poschmann, A.: The PHOTON Family of Lightweight Hash Functions. In: Rogaway, P. (ed.) CRYPTO 2011. LNCS, vol. 6841, pp. 222–239. Springer (2011)
13. ISO/IEC 29192-5:2016. Information technology – Security techniques – Lightweight cryptography – Part 5: Hash-functions (2016)
14. Krovetz, T., Rogaway, P.: The Software Performance of Authenticated-Encryption Modes. In: Joux, A. (ed.) FSE 2011. LNCS, vol. 6733, pp. 306–327. Springer (2011)
15. Leurent, G., Sibleyras, F.: The Missing Difference Problem, and Its Applications to Counter Mode Encryption. In: Nielsen, J.B., Rijmen, V. (eds.) EUROCRYPT 2018, Part II. LNCS, vol. 10821, pp. 745–770. Springer (2018)
16. National Institute of Standards and Technology (NIST): Submission requirements and evaluation criteria for the lightweight cryptography standardization process (Aug 2018)
17. Perrin, T., Marlinspike, M.: The double ratchet algorithm (revision 1). Online Document: `https://signal.org/docs/specifications/doubleratchet/doubleratchet.pdf` (Nov 2016)
18. Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.3. RFC 8446 (Aug 2018), `https://rfc-editor.org/rfc/rfc8446.txt`
19. Rogaway, P.: Efficient Instantiations of Tweakable Blockciphers and Refinements to Modes OCB and PMAC. In: Lee, P.J. (ed.) ASIACRYPT 2004. LNCS, vol. 3329, pp. 16–31. Springer (2004)
20. Rogaway, P., Bellare, M., Black, J., Krovetz, T.: OCB: a block-cipher mode of operation for efficient authenticated encryption. In: Reiter, M.K., Samarati, P. (eds.) ACM CCS 2001. pp. 196–205. ACM (2001)
21. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO '96. LNCS, vol. 1109, pp. 313–328. Springer (1996)
22. Wegman, M.N., Carter, L.: New Hash Functions and Their Use in Authentication and Set Equality. J. Comput. Syst. Sci. 22(3), 265–279 (1981)