Universidade do Minho

Mestrado em Engenharia Informática

Tecnologia Criptográfica

Auditing a Poly1305 MAC implementation in Jasmin for x86

Miguel Miranda Quaresma
A77049

December 17, 2018

Abstract

Current cryptograghic systems all share one common pitfall: lack of focus in performance. This is motivated by the fact that speed isn't always seen as a primary concern for highly critical systems where the security requirements are extremely high. However, the rise in the use of low powered devices such as IoT, smartphones, smartcards, etc has incentivized the investment in highly performant cryptographic systems that can be used in such devices. The present work aims to audit a component of such a system, namely an implementation of the Poly1305 MAC using Jasmin, a framework for developing high performance and high assurance cryptographic software([1]).

1 Introduction

Poly1305 is one time authenticator developed with performance in mind([2]) that generates a message authentication code for a given input. When combined with a framework for developing high performance and high assurance cryptographic software such as Jasmin, the result is a high performance tool that can be used in devices with low computational power such as IoT, Smartphones, Smart-cards, etc.

2 Poly1305 explained

Poly1305 is a message authentication code(MAC) that guarantees integrity and authenticity of messages. Poly1305 works similarly to a universal hash function, evaluating a given message over a polynomial and using the result as a MAC. Additionaly, Poly1305 evaluates this polynomial over a prime field, from 0 to $2^{130} - 5$, where the name (Poly**1305**) stems from. It uses a 256 bit secret, derived via a Password-Based Key Derivation Function(PBDKF), using the first 128 bits for the key, k and r for calculating the polynomial.

Poly1305 works by breaking the input in 16 bytes blocks, appending each one with a 1 byte(00000001) to prevent forgery. It then proceeds to apply the following algorithm/formula:

```
h = 0
for block in blocks:
    h += block
    h *= r
h+=k mod 2^130-5
```

where:

- `block` is a 17 byte block from the message
- `r` and `k` are 128 bit values derived from the 256 bit key used by Poly1305

3 Conclusion

References

[1] José Bacelar Almeida et al. "Jasmin: High-Assurance and High-Speed Cryptography". In: Oct. 2017, pp. 1807–1823. DOI: 10.1145/3133956.3134078.

[2] Daniel J. Bernstein. "The Poly1305-AES Message-Authentication Code". In: *Fast Software Encryption*. Ed. by Henri Gilbert and Helena Handschuh. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 32–49. ISBN: 978-3-540-31669-5.