Universidade do Minho

Mestrado em Engenharia Informática

Tecnologia Criptográfica

Auditing a Poly1305 MAC implementation in Jasmin for x86

Miguel Miranda Quaresma
A77049

December 28, 2018

Abstract

Poly1305 is a one time authenticator developed with performance in mind, that generates a message authentication code for a given input and secret key. Jasmin is framework for develpoing high performance and high assurance cryptographic software. The current works aims to examine an implementation of the Poly1305 MAC using the Jasmin framework. The Poly1305 MAC is described in at a high abstraction level, followed by an in-depth analysis of the Jasmin implementation of the algorithm. The work concludes with an auditing/verifcation of the premisses/assumptions that were made for the implementation.

1 Introduction

Message authentication codes play a major role in guaranteeing the authenticity and integrity of data being sent across an untrusted channel. There are many cryptoprimitives that work as MAC's and, for a long time, HMAC(Hash-MACs) were preferred over other MACs due to their performance, with other primitives such as the ones based on universal hashing being discarded. This was further reinforced by the introduction of assembly instructions that performed hash functions directly in hardware, such as Intel's instruction for SHA-256: **INSERT x86 INSTRUCTION HERE**. The development of cryptoprimitives such as Poly1305 MAC using Jasmin, a framework for developing high performance and high assurance cryptographic software [1], allowed this tendency to be reversed by obtaining highly performant implementations with relative ease.

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 byte blocks, appending each one with a 1 byte(00000001) to prevent forgery, via padding attacks. 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.