International Journal of Information Security and Privacy
Volume 12 • Issue 1 • January-March 2018

Performance Evaluation of SHA-3 Final Round Candidate Algorithms on ARM Cortex–M4 Processor

Rajeev Sobti, Lovely Professional University, Punjab, India
Geetha Ganesan, Lovely Professional University, Punjab, India

ABSTRACT

SHA-3 was an open competition initiated by NIST to design new generation of hash functions. This competition was a necessity to overcome the challenges imposed by multiple attacks on MDx family of hash functions including SHA-0 and SHA-1. For this competition, NIST announced a reference platform which did not cover Embedded and Mobile machines. This paper compares the performance of SHA-3 final round candidate algorithms on ARM Cortex-M4 processor (embedded processor) and presents the results. Cycles per Byte is used as performance metric. Cortex-M4 based Stellaris® LM4F232 Evaluation Board (EK-LM4F232) from Texas Instruments is used for performance evaluation.

KEYWORDS

ARM Cortex-M4, Cryptographic Hash Functions, Cycles per Byte, Performance Evaluation, SHA-3

INTRODUCTION

Cryptographic Hash Functions are crucial in implementing multiple security goals and have led their way into various security applications like: digital signatures, storing passwords, digital time stamping, constructing block ciphers, generating pseudorandom numbers, maintaining secure web connections, and encryption key management etc. Among all hash functions being used, those from SHA (Secure Hash Algorithm) family covering SHA-0 (U.S. Department of Commerce, 1993), SHA-1 (U.S. Department of Commerce, 1995), SHA-2 {SHA-224, SHA-256, SHA-384, SHA-512} (U.S. Department of Commerce, 2002) have been the most commonly used ones. This SHA family of functions was developed by National Security Agency (NSA) and certified as Federal Information Processing Standard (FIPS) by National Institute of Standards and Technology (NIST), US Department of Commerce. All these are based on MD4 and MD5 algorithms, commonly known as MDx family of hash functions. Around year 2004 and later majority of hash functions based on MDx family (MD4, MD5, HAVAL, RIPEMD, SHA-0 and SHA-1) were attacked (Wang, Feng, Lai, & Yu, 2004) (Wang, Lai, Feng, & Chen, 2005) (Wang, Yu, & Yin, Efficient Collision Search Attacks on SHA-0, 2005) (Wang, Yin, & Yu, n. d.) (Biham & Chen, 2004) (Biham et al., 2005). Given that SHA-2 functions are in the same family and share a common heritage and design principles as the earlier broken functions, these attacks shook the long-term confidence of cryptographers in nearly all hash functions. A question that perturbed everybody's mind was what if SHA-2 is compromised or successfully cryptanalyzed or broken and what could be its repercussions? If this proved true, the world would not be left with any option because SHA-2 was the best that we had at that time.

DOI: 10.4018/IJISP.2018010106

To handle this situation, NIST, initiated a design competition (public open competition) in November 2007 for designing next generation of hash functions (U.S. Department of Commerce, n. d.). The objective of the competition was to design a new hash standard named 'SHA-3' to augment current standard (SHA-2). NIST received 64 hash function submissions from over 200 cryptographers around the world. NIST also invited the public to evaluate the submissions and consequently a lot of cryptanalysis and public review were carried out. In December 2010, five algorithms (Blake, Grøstl, JH, Keccak, and Skein) advanced to the final round.

The 'Reference Platform' announced by NIST for SHA-3 competition consisted of general purpose machine (Windows Intel machines). Considerable domain of architectures like the ones prevalent in Smart Cards, Embedded systems, and Mobile platforms were ignored. This paper revolves around these five SHA-3 final round candidate algorithms and evaluation of their performance on architecture other than the one specified in 'Reference Platform' and thus in its way contribute to NIST's public call to evaluate and compare performance of these candidate algorithms. This paper presents the performance comparison of SHA-3 finalists on ARM Cortex-M4 architecture. The choice of the target platform was a two-step decision. In the first step, the decision to go for embedded and mobile platform was directed by the recent surge in usage of these devices. In the second step, for zeroing down on ARM architecture, its market dominance and technical features were the main consideration.

Organization of the Paper

Section 2 gives the brief introduction about SHA-3 final round candidate algorithms. Section 3 presents the hardware and software tools used, and methodology adopted to carry out the evaluation of SHA-3 finalists. Results are presented in Section 4 followed by conclusion and future work in Section 5.

INTRODUCTION TO SHA-3 FINAL ROUND CANDIDATE ALGORITHMS

Keccak

Keccak hash function is SHA-3 winner and is based on sponge construction. Keccak generates arbitrary length output using fixed length permutation (*Keccak-f*) operating on fixed number of bits *'b'*. The basic building block, *Keccak-f*, is characterized by two parameters: bitrate *'r'* and capacity *'c'* and holds the relation $b = r + c$. The permutation, *Keccak – f*, operates state *'a'*, which is three-dimensional array of elements of GF (2). Initially, all the bits of state are initialized to zero. The input message is padded using multi-rate padding and divided into blocks of *'r'* bits each. The sponge construction then proceeds in two phases: 'Absorbing Phase' in which each block of input message is XORed with *'r'* bits of state followed by application of *Keccak-f*. All this is succeeded by 'Squeezing Phase', in which first *'r'* bits of the state are returned as output block, interweaved with application of Keccak-f. The number of blocks is decided by the user depending on the desired hash output size. The details can be referred from (Bertoni, Daemen, Michaël, & Assche, 2011).

Skein

Skein works on three different internal state sizes – 256 bits, 512 bits, and 1024 bits. However, Skein – 512 was the prime proposal and the same is evaluated in this paper. Skein can produce variable length hash output as desired by the user. Skein uses Tweakable block cipher - Threefish as the basic building block and UBI (Unique Block Iteration) chaining mode to process arbitrary input size to generate desired output. Threefish makes use of three mathematical operations: XOR, Addition and Rotation (with a constant) and all operations are done on 64-bit words. The core of Threefish is MIX function that takes two words as input and applies one addition, rotation and XOR operation to update these words. Every round of Threefish–512 makes use of four MIX functions followed by a permutation, named 'Permute' of the eight 64-bit words. A sub-key is inputted every four rounds.