International Journal of Advanced Computer Science and Information Technology (IJACSIT)
Vol. 2, No. 2, April 2013, Page: 1-12, ISSN: 2296-1739

www.elvedit.com

Hash Function of Finalist SHA-3: Analysis Study

Authors

Imad Fakhri Al-shaikhli
Department of Computer Science, International Islamic University of Malaysia
imadf@iium.edu.my
Jalan Gombak, 53100, Malaysia

Mohammad A. Alahmad
Department of Computer Science, International Islamic University of Malaysia
malahmads@yahoo.com
Jalan Gombak, 53100, Malaysia

Khanssaa Munthir
Department of Computer Science, International Islamic University of Malaysia
Khansa95@yahoo.com
Jalan Gombak, 53100, Malaysia

Abstract

The National Institute of Standard and Technology (NIST) has suggested different principles for hash functions to avoid the blunders and to choice the ideal quality of hash function, which to be a measurement for the future of hash function generations. Therefore, the goal of the NIST contenders in SHA-3 between the hash functions is to be chosen as the winner in the end of 2012, and the beginning of 2013. Thus, for this reason the paper addresses the comparative and analysis study of the finalist SHA-3 candidates in: complexity of security, design and structure, as well as performance and cost, to measure the robustness of the algorithms in this area, through the Fundamentals Security Measurement Factors of Hash Function (FSMFHF) of Secure Hash Algorithm (SHA). Therefore, main idea from this comparison and analysis study between the finalist of SHA-3 candidates such as (BLAKE, Grostl, JH, Keccak, and Skein) is to investigate the tight security in the suitable designs of lightweight such as JH and Keccak for the future security of hash function. Moreover, they are investigating the high trade-off in (speed/memory) that implemented in Virtex-7 2000T of FPGAs family hardware. Whereas, excluded the rest of hash functions in this finalist, which are not investigated all the measurements of hash function as mentioned above.

Key Words

Complexity, FSMFHF, Performance, Security.

I. INTRODUCTION

Cryptography is a field of applications that provides privacy, authentication and confidentiality to users. An important sub-field is that of secure communication, which aimed at allowing confidential communication between different parties, such that no unauthorized party has access to content of the messages. However, this field has a long history of successes and failures, as many methods to encode messages emerged along the centuries, always to be broken some time later. The hash functions are one target of secure communication systems by using the messages digests to generate data integrity for detection of unauthorized changes in files. Hash functions are a key cryptographic primitive used in many applications, such as authentication, ensuring data integrity, as well as digital signatures, message-digest algorithm 5 (MD5), MD4, HAVAL, FORK-256, SHA-family, RIPEMD-family, and other forms of authentication. In terms of the security of hash function is a computationally secure form of compression [2]. However, these hash functions have lower behavior in different characteristic of that should be harder to resistance against different attacks. That leads to establish in October of 2007- the institute called National Institute of Standard and Technology (NIST) to identify the various hash standards such as SHA-1, SHA-224, SHA-256, SHA-384 and SHA-512 to measure the Secure Hash Standards (SHS) levels. These levels of security are analysis techniques for developing security communication systems over time [1]. This paper tries to compare and analyze between the finalist of SHA-3 round 3 candidates, in the Fundamentals Security Measurement Factors of Hash Function (FSMFHF) of Secure Hash Algorithm (SHA), of three factors such as: complexity of security hash function, structure and design of hash function, as well as the performance in low cost. Therefore, this paper is structured as follows; Section 2 presented the SHA-3 round 3 finalist candidates; 3 overviews of the main hash function factors measurements for each security factors (FSMFHF); Section 4 compare and analyze SHA-3 round 3 finalist; Section 5 discusses finding and results of the study; finally Section 6 discusses the conclusion and future work of this study.

II. BACKGROUND OF SHA-3

In December 9, 2010, NIST selected five finalists for the final round of the competition; BLAKE, Grostl, JH, Keccak and Skein. BLAKE is proposed by Jean-Philippe Aumasson from FHNW, ETHZ Switzerland universities, and it has four versions; BLAKE224, BLAKE256, BLAKE384 and BLAKE512. Where, Grøstl is an AES-based hash function and one of the five finalists of the NIST SHA-3 competition, designed by Praveen Gauravaram and his team. It combines characteristics of the wide-pipe design and chop-Merkle-Damgard to construct an arbitrary message size of *n* bits digest size in {224, 256, 384, and 512} for the larger size of the hash output. JH is an iterative hash function designed by Hongjun Wu, which process a message blocks of 512 bits and produce the hash algorithms in JH-224, JH-256, JH- 384 and JH-512. In addition, Keccak is a cryptographic hash function designed by Guido Bertoni, joan Daemen, Michael Peeters, and Gilles Van Assche. It supports at least four different output lengths *n* {224, 256, 384, and 512} in a high security levels. Lastly, Skein is a new family of SHA-3 candidates that proposed by Niels Ferguson and his team from Microsoft and Intel Companies, it has three

different internal state sizes: 256, 512, and 1024 bits to produce message digests of any length from 1 to 296 bytes[1]-[5]-[7].

III. THE MAIN HASH FUNCTION FACTORS MEASUREMENTS

This paper represents a comprehensive comparison analysis study of the measurements in Fundamentals Security Measurement Factors of Hash Function (FSMFHF), which addresses different conceptions of security factors of hash function, as the NIST requirements to measure the final chose of SHA-3, such as;

A. The Complexity of Security Hash Function Measurements

The security requirements of hash functions have different usages and depends in their applications for particular protocols, such as signature schemes, MAC, etc. However, the security of hash function becomes more secure and resists serious attacks. Therefore, NIST governance has security requirements to measure the security differential between the finalists SHA-3 candidates in the following [1]:

1. To choose minimum variant that has support safely HMAC mode and randomized hashing.
2. The security requirements have to investigate the pre-image resistance of around n-bits digest values, second pre-image resistance and have around n-*L* bits, where the preimage is the length about 2^L blocks, and collision resistance has around n/2 bits.
3. Each candidates from this group of SHA-3 must be secured against the length extension attack, and;
4. The security of this group is realized in the ideal model of the principal integral building blocks such as block cipher or permutations.

B. The Structure and Design Measurment

This section will review NIST principles in construction and design of finalist SHA-3 candidates, which includes many measurements. These principles are summarized as the following:

1. *HAIFA-design;* is based on the Merkle-Damgard (MD) principle of construction blocks cipher, for padding the message in a specific ways to solve some deficiencies in the original MD construction.
2. *Wide-pipe design;* is an iterated larger size of MD construction than the final hash output, with a final transformation, and chopping at the end of indifferentiable random oracle.
3. *Sponge design;* is a specific type of hash function from chopping-MD construction design.

Threefish of tweakable blocks cipher; is a compression function that take three inputs: a key, a tweak and a block of message, instead of the usual block ciphers that take two input: the key and the block of message.

C. Performance and Cost Implementations of SHA-3 Finallist

The main particular implementations of SHA-3 groups in hardware algorithms on FPGA and ASIC platforms are; high-speed implementations and tight implementations approaches. However, this study provides the high-speed implementations of maximum message and short message in 8 byte of final round of SHA-3 on FPGAs, in comprehensive comparison function, with the application area of these candidates [3]-[7].

IV. Comparison and Analysis of SHA-3 Finalist

This section focuses on different measurements between SHA3 candidates in security and complexity factors, the structure and design, with performance and cost implementations, as shows in the following points:

A. Comparison of Finalist SHA-3 in the Security and Complexity

As shown in Table 1 the analysis and comparison of SHA-3 round 3 finalists from security aspects and it is concluded as the following [3], [4], [5]:

TABLE I: ANALYSIS AND COMPARISON OF SECURITY FINALIST SHA-3

Candidates	Collision Resistance	Round of compression	Complexity		
			Pre-image	2nd pre-image	Pseudo pre-image
Blake-256 Blake-512	Inner-collision	2.5 4	2^{224} 2^{448}	2^{256}	-
Grostl-256 Grostl-512	2^{64} 2^{128}	5 8	2^{256} 2^{512}	$2^{256\text{-}512}$ $2^{512\text{-}1024}$	$2^{244.85}$ 2^{248}
JH-256 JH-512	$2^{96.12}$ $2^{95.63}$	16 22	-	2^{-388} 2^{-900}	-
Keccak-224 Keccak-256 Keccak-512	Near-2^{256} 2^{512}	4-5 5-10 24	2^{112} 2^{1370} 2^{1590}	2^{288} 2^{512} $2^{511.5}$	 2^{1576}
Skein-256 Skein-512 Skein-1024	>2^{-265}	32-36	2^{105}	2^{200}-2^{824}	$2^{511.7}$ 2^{1045}-2^{125}

1. The BLAKE hash function assumed to be ideal security resistant to the generic second pre-image attacks and resistant to length-extension attacks but it is exhibiting for inner-collision, which regarded resistance to Joux's multi-collisions similar to SHA-2.

2. Whereas, using the wide diffusion design strategy in the two Grostl permutations P and Q, to build Grostl in very strong confusion and diffusion. This allows different attacks to utilize this property to attack the Grostl security in different rounds of memory compression function. However, it appears the protection from these attacks by differential in rounds. Thus, there is no threatening on the Grostl security due to rebound attack techniques. However, the third rebound

attack on Grostl is Semi-Free-Start Collision and pre-image as appear in differences of the second permutation that found in Grostl-256, and Grostl-512, to investigate an upper bound of collision resistances and preimge resistance.

3. Compared with JH hash function, which is considered an optimal to improve the upper bounds of the collision, pre-image, and second pre-image via indiferentibility compression function of random oracle, through the suffix-free in padding rule in one permutation. Due to the rebound attack property, security of padding and final truncation that to be as sponge operation to prevent any attack threat on JH. However, Table 1 shows the estimation in collision of compression function and an improvement in the hash function for each elements-sorting complexity that investigated from following equations:

$$Adv_H^{col} = \theta \left(\frac{q^2}{2^n} + \frac{q^3}{2^{L-m}}\right), \quad (1)$$

$$\text{and } Adv_H^{pre}, Adv_H^{sec} = \theta \left(\frac{q}{2^n} + \frac{q^3}{2^{L-m}}\right) \quad (2)$$

4. Where, *Adv* is Adversary to make *q* forward or backward of the random permutation, $attack \in \{pre, sec, col\}$, *H* hash function, *n* output lengths of {224, 256, 384, and 512}, *L* length of block, *m* Maximum of block*, and* θ is theta, for each collision, pre-image and second pre-image resistances, respectively. In contrast, the Keccak hash function has no vulnerabilities in security, due to the sponge construction against the generic attacks. However, this structure has property leads to be more flexibility and simplicity for in-differentiability bound of hash function or (zero-sums) in length output, and tradeoff between bit-rate and security. For example, the bounds of the output in hash function are no more than n = 512, but the bounds in in-differentiability are varying between 512 and 1024. This is because the Constrained Input Constrained Output's (CICO) resistance against different attacks. However, to be an ideal security bound of in-differentiability random oracle and an optimal permutation for collision resistance, preimage resistance, second preimage resistance, etc. in compression function.

5. However, the Skein contestant assumed an ideal security from random oracle in tweakable block cipher from boomerang distinguishers' attacks of the compression function. This propriety has proved the collision resistance in the ideal cipher model, and preserved the based preimage awareness approach for any security bounds in the compression function, thus represented in this equation:

$$Adv_H^{col} = \theta(q^2/ 2^n) \quad (3)$$

B. Comparison of Finalist SHA-3 in Structures and Designs

Table 2 presents some similarities and differences between the five candidates of finalist SHA-3, and it is concluded in the following comparison:

TABLE II: ANALYSIS OF THE STRUCTURE AND DESIGN OF FINALIST SHA-3 CANDIDATES

Hash Algorithm	Message Digits in bits	Block M. Size	Word Size (bits)	Round of Compression	Construction the Design
Blake-224	224	$< 2^{64}$	32	14	Wide-pipe of ChaCha Stream Cipher to HAIFA
Blake-256	256	$< 2^{64}$			
Blake-384	384	$< 2^{128}$	64	16	
Blake-512	512	$< 2^{128}$			
Grøstl-224	512-bits	$2^{8\times64}$	64	9	Wide-pipe of chop-MD and AES block cipher
Grøstl-256					
Grøstl-384	1024-bits	$2^{16\times64}$	128	10	
Grøstl-512					
JH-224	224	256	64	16	Bijective of generalized AES block cipher
JH-256	256	512			
JH-384	384	1024	128	42	
JH-512	512				
Keccak-224	25-50	2^{176}	64-bits words× 32 bits processer	6	sponge& parazoa of hypercube HAIFA design
Keccak-256	200	2^{320}		7	
Keccak-384	800	2^{508}		8	
Keccak-512	1600				
Skein-256	Support any length size	256	Block message size× 128 bits tweak	72	Tweakable block cipher Threefish and UBI
Skein-512		512		72	
Skein-1024		1024		80	

1. Each member of this list has product four-message digest fixed size of {224, 256, 384, and 512} hash function from arbitrary value, except Skein, and Keccak hash functions which have different states. Such as, three internal sizes of {224, 512, and 1024} to product random output length for Skein, and Keccak hash function has product of limited output seven values in {25, 50, 100, 200, 400, 800, and 1600} [9].

2. The designs of this group based on Rijndael block cipher of the basic Merkle-Damgard serial construction, except Keccak, which is a hypercube of sponge construction, which build the three-dimensional array.

3. The compression function for each candidate has different behavior than another, due to the different mode process of these hash function. As illustrated in the following discussion:

i) The compression function of BLAKE hash function is a wide-pipe structure, where BLAKE-256 has 14 rounds and BLAKE-512 has 16 of ChaCha stream cipher in minimize self-

similarity, and this will increase the resistance collision attacks. However, BLAKE has limitation in message length such as 2^{64} and 2^{128} for BLAKE-256 and BLAKE-512 respectively and that considered the same message length of SHA-2. Moreover, BLAKE breaks self-similarity by using a round-reliant permutation the message and the constants. This prevents attacks that utilize the similarity among round functions, thus, BLAKE is non-ideal for compression function and does not achieve the progress that pleasant from this group for long time [4]-[7].

ii) In Grostl hash function, the compression function combines wide-pipe design and chop-Merkle-Damagrd to construct the two different permutations P and Q for short message digest, and for long message digest in four round transformations for matrix of size 8×8 of the 256-bit, and 8×16 of the 512-bit, respectively. Thus, the permutations are regarding an ideal construction for collision resistant, within the exceptional diffusion and confusion properties [6], [5].

iii) Whereas, JHcompression function is constructed from bijective function (a large block cipher with constant key), which is considered a look like-sponge structure from three constants of 42 round function of R_d, in an S-box layer, a liner transformation layer, and a permutation layer, this structure is quite efficient which easier to analyze security of differential attack.

iv) The compression function of Keccak hash function has structured from a seven set of Keccak-*f* permutations to construct the Keccak sponge design [10]. This design has many advantages for a hash function compared to other constructions, because of its characters in absorbing and squeezing for different input length data. Keccak can generates different output lengths; as well as flexibility, to increase the security level by increasing the capacity of bit rate in comparable permutation, in other words, it is constructed of permutation or transformation in simple round of similar block cipher without an iterated of compression function, or key schedule.

v) The Skein compression function uses Unique Block Iteration (UBI) mode, which configured inside every tweakable block cipher separately in different permutations. For example, Skein-256 and Skein-512 have 72 rounds, secretly to build Threefish, whereas, the Skein-1024 has 80 rounds totally. This mode is a chaining of threefish to process an arbitrary input size to a fixed output size. This property has directly addresses many attacks on this hash function [11].

In summary, the behavior for all this list of hash function has different performance through these constructions, but in general, they have similar principles from basic structures for building these hash function like block cipher to distribute the messages words in different method. In contract, Keccak hash function has different behavior in structure for cryptanalysis

operation, which has the sponge construction of absorbing and squeezing, flip-flops, with inverts for messages distribution.

C. Comparison of Finalist SHA-3 in Performance and Cost

The analysis of Table 3 which clarified the differential between the speeds of last round SHA-3 candidate's performance for two messages size with the group applications to achieve the clock frequencies MHz/ps. on Virtex 7 from Xilinx library of FPGAs device for all round 2 SHA-3 designs [6]. Where Speed for Max. M-256 represented the speed cycles value for maximum message; while the Speed for Min. M-64 to represent the speed cycles value for minimize messages.

TABLE III: ANALYSIS AND DIFFERENTIAL IN PERFORMED SPEED OF FINALIST SHA-3 CANDIDATES

Hash Algorithms	Speed for Max. message 2^{256}, 2^{512} MHz/ps	Speed for short message 2^{64} MHz/ps	Hash Applications
BLAKE-224/256 BLAKE-384/512	21.31 21.62-20.98	390.50 625.88	Website links of Perl, PHP, Java script
Grostl-224/256 Grostl-384/512	62.19 106.80	939.88 2565.25	Intel AES-NI Instructions of CPU
JH-224/256 JH-384/512	161.55 161.47	2762.50 2764.38	Message Authentication Code (MAC)
Keccak-256 Keccak-512	45.37 33.12	1110.00 715.78	lightweight
Skein-256 Skein-512	23.09 -	374.38 539.38	multi-processor system

For these comparisons, the study provides an efficient performance of final round SHA-3 to certain the base of the clock frequencies and number of clock cycles consumed in the hardware and software. However, to evaluate these algorithms through the wide differential between the candidates performed, in 256 bytes and 512 bytes as standardizations for each candidate, to recognize between each other in recent area as the following:

1. Seldom, differences between the BLAKE and Skein in speed/memory, as illustrated in table 3 and Figure 1, they show the minimize speed cycles value in limited area of different message sizes in short and long messages in both 256 bytes and 512 bytes. For example, BLAKE candidate has fixed and small set of constants in the memory to implement the short message and long message [8], [9]. Due to, BLAKE hash function has "*parallelism mechanism*" to reduce

the number of computation steps. Whereas, Skein hash function has different behavior in speed than to consume the memory area, which considered an optional "*hash-tree*" approach speed up parallelizable implementations, twice as fast as SHA-512 and three times faster than SHA-256. As a result, for both candidates have an optimal speed value for short messages than the long message, that means in long message no need big number of rounds to distribute the data message. While, the short message need many rounds for distributing the message in different memory space. Therefore, the hash functions spend different rounds to implement the message in hardware. Thus, in BLAKE hash function has rather implemented in speed/space than in speed/memory [11].

2. However, Grostl and Keccak have approximately similar speed values in different area size for both candidates in short and long messages for version 256. Such as, Grostl hash function has efficiently implemented on 128-bit, 64, 32, and 8-bit architectures, for utilizing parallelism round transformations in resource of memory, registers and speed. That leads to increase the speed with utilizing area for shorting message. Otherwise, Keccak hash function has high level of parallelism with weak in diffusion bits for different slices; due to it. It has the inverter properties in translation of the z-direction to map the inner bit structure that leads to slowly in speed[10].

3. Through Figure 1, JH hash function has high level in Cycles/byte for speed/ message trade-off with relatively equals of both JH versions.

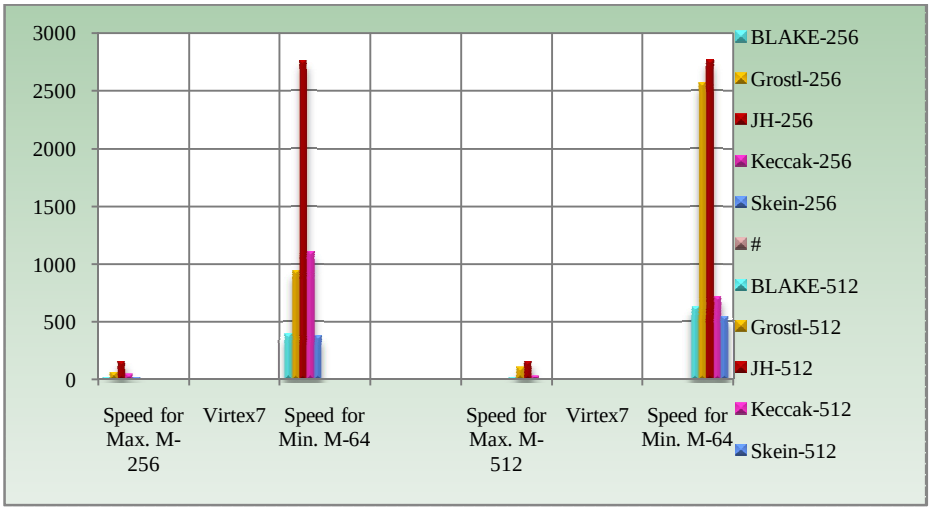

FIGURE I: CHART DIAGRAM FOR ANALYSIS SPEED OF FINALIST SHA-3 IN SPEED CYCLES IN BYTE

However, it is high level in 4 times compared to other candidates in both versions (256 and 512). Whereas, the JH-512 has been increased one time more than Grostl-512 in short message. Because the JH has prototype in the AES design methodology that has simplest approach to design an efficient large block cipher (in hardware and software) from small components.

However, to extend for three or four dimensions to achieve a block cipher in 512-bit or 2048-bit block size. However, in all finalist candidates have implemented on the Virtex-7 2000T from the sequence of FPGAs family to reduce the costs and an optimizing performance in (low power consumption to process and high speed), with an efficient applicable in industry and technology advance. In addition, this group has implemented in basic CPU, language of C and C++ for each BLAKE, Keccak, and Skein with Grostl and JH, respectively.

V. FINDINGS AND RESULTS

The main finding from this analysis and comparison study denotes that the hash function is an optimized conception, which included;

1. Excluded BLAKE and Skein from the future selection of ending 2012, due to the low speed of performance implementations for both hash functions.

2. Keccak has no vulnerabilities in security and efficient structure, as NIST discussed in March of 2012, with average speed of implemented in the same hardware due to the inverter property.

3. Grostl hash function has tight in security, tight in structure cause depended on the permutation, but has contradiction behaviors speed for different messages sizes, such as higher speed in 512 than 256 in the same message size.

4. Whereas, the JH achieves an efficient implementation of the sponge structure in simple design, ideal security, and equals in high speed in both sizes in short size or maximum size, of 256 and 512 bits of message digest, in low cost.

VI. CONCLUSIONS AND FUTURE WORK

In conclusion, the study presented the comprehensive comparative study of the finalist SHA-3 candidates within the research measurements in FSMFHF. However, these measurements have different factors to investigate the NIST requirements to choose the final candidate in the end of this year 2012 and start of 2013. Therefore, from this comparison and analysis are discussed in previous sections. There are enough scope for future researchers based on this paper and there might be numerous ways to develop the described approach. For the possible future works, pertaining in this area such as;

1. Applying the similar way and similar factors have measurements on the rest families of hash function.

2. It is unlikely that it will have a new generation of hash functions like SHA-4 competition before 2030.

3. However, the comparison between hash functions will not be limited only to the fundamentals algorithm families' types or their security characteristic in future work. The comparative studies will be helpful whenever extending for more approaches; for example, hash table, DHT, and tree (BST, Heap, Zero-Knowledge ...etc.).

Thus, to support the knowledge existing in the security algorithms field by knowing the different hash functions' types and their algorithm implementations to increase the researchers' knowledge of the cryptography field; even so, it is crucial for security applications.

REFERENCES

[1] Andreeva E. Mennink B. Preneel B. & Skrobot M. (2012), Security Analysis and Comparison of the SHA-3 Finalists BLAKE, Grostl, JH, Keccak, and Skein.fromKatholiekeUniversiteit Leuven.

[2] Belgium. E. B. Kavun& T. Yalcin (2012), On the Suitability of SHA-3 Finalists for Lightweight Applications. from Horst Görtz Institute, Ruhr University, Chair of Embedded Security, Germany.

[3] Ewan F. Christian F. and Michael G.(2008), *The Twister Hash Function Family*. Publishing Article, Retrieved in October 28, 2008,

[4] Elbirt J. (2009), *Understanding and Applying Cryptography and Data Security*. Book ISBN 978-1-4200-6160-4 (alk. paper).

[5] Imad Fakhri Alshaikhli, Mohammad A. Ahmad, Hanady Mohammad Ahmad (2012). "Protection of the Texts Using Base64 and MD5." JACSTR Conference Vol 2, No 1 (2012)(1): 12.

[6] Imad Fakhri Al Shaikhli, A. M. Z., Rusydi H. Makarim, and Al-Sakib Khan Pathan (2012). "Protection of Integrity and Ownership of PDF Documents Using Invisible Signature." UKSim 14th International Conference on Computer Modelling and Simulation: 533--537.

[7] Homsirikamol E. Rogawski M. & K. Gaj (2010), *Comparing Hardware Performance of Fourteen Round Two SHA-3 Candidates Using FPGAs*. Retrieved December 21, 2010, from George Mason University.

[8] Namin A. H. & Hasan M. A. (2010), *Implementation of the Compression Function for Selected SHA-3 Candidates on FPGA.* Retrieved Feb. 25th, 2010, Conference Publications

[9] Regenscheid A. Perlner R. Chang S. Kelsey J. Nandi M. & Paul S. (2009), *Status Report on the First Round of the SHA-3 Cryptographic Hash Algorithm Competition.* Retrieved September 2009, from National Institute of Standards and Technology, U.S. Department of Commerce. NIST Interagency Report 7764

[10] Rechberger C. (2010), *Second-Preimage Analysis of Reduced SHA-1*, from Katholieke Universiteit Leuven, Department of Electrical Engineering. Publishing paper.

[11] Silva J. E. (2003), *An Overview of Cryptographic Hash Function and Their Uses.* from the SANS Institute Reading Room site. Retrieved January 15, 2003.

[12] Schorr (2010), *Performance Analysis of a Scalable Hardware FPGA Skein Implementation.* Retrieved February 2010, thesis of Master Degree from Kate Gleason College of Engineering Department of Computer Engineering Rochester, New York.