2.B Algorithm Specifications and Supporting Documentations

2.B.5 Analysis of the NaSHA hash algorithm with respect to known attacks

NaSHA family of cryptographic hash function use Merkle-Damgård domain extender with standard Merkle-Damgård strengthening. It has incorporated also wide-pipe design of Lucks [12, 13] and Coron's [2] suggestions. Compression function of NaSHA is function from $\{0,1\}^{4n}$ to $\{0,1\}^{4n}$ and then only $2n$ bits are kept for the next iterative step. The length of chaining variable is two times wider than the final digest value. With this kind of design we gain resistant to some generic attacks like: Joux multicollision attack [6], length extension attack, Dean fixed point attack [4], Kelsey and Schneier long message 2^{nd} preimage attack [7], Kelsey and Kohno herding attack [8] and 2^{nd} collision attack.

1 Choice of starting bijection and initial values - no trap-doors

We use as starting bijection $f : \mathbb{Z}_2^8 \to \mathbb{Z}_2^8$ for creating extended Feistel networks a well known and well examined function - the improved AES S-box with the APA structure from Cui and Cao [3].

We considered several possibilities as choice of NaSHA S-box: AES S-box [1], improved AES S-box from Liu and all [11] and improved AES S-box with the APA structure from Cui and Cao [3]. All three runners have some proc and cons. The AES S-box is the most famous and the most investigated S-box in cryptology, with good differential and linear resistance and high algebraic degree. But it has simple algebraic structure with only 9 terms. The improved AES S-boxes has also good differential resistance

with differential 4-uniformity and good linear resistance. They have the same algebraic degree as AES S-box, but they have much bigger algebraic complexity of 255 terms for first and 253 terms for second S-box. Their inverse S-box has high algebraic complexity of 255 terms as AES inverse S-box. But both are not enough studied from other authors. Our winner is the third solution because of its algebraic complexity and because it is little bit more studied than the second solution.

In the case of a suspicion a trapdoor being built into the hash, the current S-box can be replaced by any of the other two candidates.

The initial values are randomly generated. If somebody has suspicions of NaSHA initial chaining values, they can be replaced by any others, without changing the security or the performances of NaSHa hash function.

2 Resistance to attacks that change all the additions by XORs

The compression function of NaSHA-(m, k, r) uses additions modulo 2^{32} and 2^{64}, XORs and left rotations, so we have to examine attacks that change all the additions by XORs in NaSHA-(m, k, r). It is important to mention the work of Lipmaa and Moriai [9], which constructed efficient algorithms for computing differential properties of addition modulo 2^n, the work of Lipmaa,Wallen and Dumas [13], which constructed linear-time algorithm for computing the additive differential probability of XOR, and the work of Paul and Preneel [15].

NaSHA-(m, k, r) is resistant to these kind of attacks, because it is using extended Feistel networks [14], which incorporate operations with 8, 16, 32 and 64-bits operations and table lookups, instead of using only combinations of 32 or 64-bits words. Additionally, having in mind that the compression function of NaSHA-(m, k, r) is a function from $\{0,1\}^{4n}$ to $\{0,1\}^{4n}$, at this moment we can not see that it is possible to find concrete values of the arguments of this function such that the additions will behave as XORs.

3 Resistance to linear and differential attacks

Recent collision attacks on some hash functions [16, 17, 18] are in fact differential attacks that involve modular integer subtraction or exclusive-or as

a measure of difference, and some kind of message modification techniques. There are several strategies that one can employ to prevent the success of these attacks. The first one is to attempt to prevent the existence of any "good" differential (a differential path that leads to (near) collisions and holds with probability greater than $2^{-n/2}$), like wide trail strategy for block ciphers. The second strategy would be to reduce the success probability of the attack with restraining the power of the message modification techniques. A third possibility is to consider situations in which single message bits are affecting multiple blocks or maybe entire hash.

The NaSHA-(m, k, r) hash algorithm allows each bit of an input message M to influence almost all bits of the resulting hash value. To verify this let represent $S^{(i)}$ as

$$S^{(i)} = S_1^{(i)}||S_2^{(i)}||S_3^{(i)}||\ldots||S_{2t-2}^{(i)}||S_{2t-1}^{(i)}||S_{2t}^{(i)}.$$

We have that every bit from the bit string $S^{(i)}$ influences all blocks $S_j^{(i+1)}$ with even subindexes ($j = 2, 4, 6, \ldots, 2t$) of the bit string $S^{(i+1)}$. Namely, by Step 6 of NaSHA hash algorithm, we apply the transformations $LinTr_{2^{n+2}}^{2t}$ and $\mathcal{MT}$ on $S^{(i)}$. The linear transformation besides diffusion spread out the influence of bits. The $\mathcal{MT}$ transformation is composition of $\mathcal{A}_l$ and $\rho(\mathcal{RA}_l)$ transformations. Now, if b is a bit from a block $S_j^{(i)}$ of $S^{(i)}$, then all blocks of $\mathcal{A}_l(S^{(i)})$ from the $j+1$-th until $2t$-th are influenced by b. After that, all blocks of $\mathcal{MT}(\mathcal{A}_l(S^{(i)}))$ will be influenced by b. So we have the following theorem.

Theorem 1 *Every bit from the input message M influences all blocks of the hash value NaSHA-$(m, k, r)(M)$.*

Proof. By the above considerations we have that each bit of M influences all blocks with even subindexes of $S^{(N)}$. Since NaSHA-$(m, k, r)(M) = A_4||A_8||\ldots||A_{2t-4}||A_{2t}$, where $A_1||A_2||A_3||\ldots||A_{2t} = (LinTr_{2^{n+2}}^{2t}(S^{(N)}))$, all blocks of NaSHA-$(m, k, r)(M)$ are influenced by each bit of M. □

Much more than Theorem 1 is stating, the internal structure of the quasigroup operation and the addition modulo 2^r allows us to conclude that almost all bits of the hash value are influenced by each bit of the input message.

Also, we have to stress out that our starting bijection has also good resistance to differential attacks with its differential 4-uniformity and its good

resistance to linear attacks with nonlinearity of 112. All these together give as good resistance to any attack that will involve differential cryptanalysis.

The nonlinearity of 112 of the starting bijection f is inherited in the constructed extended Feistel networks in our implementation, so F_{A_1,B_1,C_1} and F_{A_2,B_2,C_2} have also a nonlinearity of 112. We find out that we have gained resistance of NaSHA-(m,k,r) to any attack that will involve liner cryptanalysis.

Because of the inherited nonlinearity of 112 of the extended Feistel networks F_{A_1,B_1,C_1} and F_{A_2,B_2,C_2}, that are used for defining the quasigroup operations in our NaSHA-$(m,2,6)$ implementation, NaSHA-$(m,2,6)$ is resistant to recent Cube attack of Dinur and Shamir [5], that can be applied to wide rang of cryptographic primitives which are provided as a black box (even when nothing is known about its internal structure) as long as at least one output bit can be represented by (an unknown) polynomial of relatively low degree in the secret and public variables.

References

[1] J. Daemen and V. Rijmen, *The Design of Rindael: AES - The Advanced Encryption Standard*, Springer-Verlag, Berlin, 2002

[2] J.-S. Coron, Y. Dodis, C. Malinaud and P. Puniya, *Merkle-Damgård revisited: How to construct a hash function*, CRYPTO 2005, LNCS **3621**

[3] L. Cui and Y. Cao, *A new S-box structure named Affine-Power-Affine*, International Journal of Innovative Computing, Information and Control **3(3)**, (2007), pp. 751–759

[4] R. D. Dean, *Formal Aspects of Mobile Code Security*,Ph.D. dissertation, Princeton University, 1999

[5] I. Dinur and A. Shamir, *Cube Attacks on Tweakable Black Box Polynomials*, *http* : *//eprint.iacr.org*/2008/385

[6] A. Joux, *Multi-collisions in Iterated Hash Functions. Applications to Cascades Constructions*, Advances in Cryptology - CRYPTO 2004, LNCS **3152** (2004), pp 306–316

[7] J. Kelsey and B. Schneier, *Second preimages on n-bit hash functions for much less than 2n work*, Advances in Cryptology - EUROCRYPT 2005, LNCS **3494** (2005), pp 474–490

[8] J. Kelsey and T. Kohno, *Herding Hash Functions and the Nostradamus Attack*, Advances in Cryptology - EUROCRYPT 2006, LNCS **4004** (2006), pp. 183–200

[9] H. Lipmaa and S. Moriai,*Efficient algorithms for computing differential properties of addition*, FSE 2001, LNCS **2355**, (2002), pp. 336–350

[10] H. Lipmaa, J. Wallen and P. Dumas,*On the Additive Differential Probability of Exclusive-Or*, FSE 2004, LNCS **3017**, (2004), pp. 317–331

[11] J. Liu, B. Wei, X. Cheng and X. Wang, *Cryptanalysis of Rijndael S-box and improvement*, Applied Mathematics and Computation **170 (2)**, (2005), pp.958–975

[12] S. Lucks, *Design Principles for Iterated Hash Functions*, Cryptology ePrint Archive, Report 2004/253

[13] S. Lucks, *A Failure-Friendly Design Principle for Hash Functions*, ASIACRYPT 2005, LNCS **3788** (2005), pp. 474–494

[14] S. Markovski and A. Mileva, *Generating huge quasigroups from small non-linear bijections via extended Feistel network*

[15] S. Paul, and B. Preneel, *Near Optimal Algorithms for Solving Differential Equations of Addition with Batch Queries*, In Progress in Cryptology - INDOCRYPT 2005, LNCS **3797**, (2005), S. Maitra, C. E. Veni Madhavan, and R. Venkatesan (eds.), Springer-Verlag, pp. 90–103

[16] X. Wang, H. Yu and Y. L. Yin, *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology - CRYPTO 2005, LNCS **3621**, (2005) pp. 1–16

[17] X. Wang, Y. L. Yin, and H. Yu, *Finding Collisions in the Full SHA-1*, Advances in Cryptology - CRYPTO 2005, LNCS **3621**, (2005) pp. 17–36

[18] X. Wang and H. Yu, *How to Break MD5 and Other Hash Functions*, Advances in Cryptology - EUROCRYPT 2005, LNCS **3494** (2005), pp. 19-35