Extended version of the paper sent to EUROCRYPT 2007.

A New Concept of Hash Functions SNMAC Using a Special Block Cipher and NMAC/HMAC Constructions

Vlastimil KLÍMA[*]
October 2006[**]

Abstract. In this paper, we present new security proofs of well-known hash constructions NMAC/HMAC proposed by Bellare et al. in 1996. We show that block ciphers should be used in hash functions in another way than it has been so far. We introduce a new cryptographic primitive called *special block cipher* (SBC) which is resistant to attacks specific for block ciphers used in hash functions. We propose to use SBC in the NMAC/HMAC constructions, what gives rise to the new concept of hash functions called *Special NMAC* (SNMAC*)*. From our new NMAC/HMAC security proofs it follows that SNMAC hash functions are computationally resistant to preimage and collision attacks. Moreover, at CRYPTO 2005 Coron et al. proved that SNMAC is indifferentiable from a random oracle in the limit. SNMAC construction is general and it enables various proposals using different instances of the special block ciphers. We propose a special block cipher DN (Double Net) and define hash function HDN (Hash Double Net) as the SNMAC construction based on DN.

Keywords: Hash function, NMAC, HMAC, collision resistance, preimage resistance, conception.

Contents

[*] independent consultant, v.klima (at) volny.cz, http://cryptography.hyperlink.cz, In the paper, we present a part of the project ST20052005017 of Czech National Security Authority

[**] the second version of eprint paper will contain the Appendices 2 and 3 (waiting for approval of the publication)

1. Introduction

It is well known that the most of common hash functions are vulnerable to message extension attack [Tsu92], i.e. having given h(*M*) and suitable N, it is possible to determine *h*(*M* || *N*). Even though this property differentiates these functions from random oracles, it was tolerated for a long time. In 2004 and 2005, further generic problems of hash functions were discovered. There were Joux's multicollision attack [Jou04] and Kelsey-Schneier's multicollision attack and second preimage attack [KS05]. Note that all modern hash functions are vulnerable to these three generic attacks ([Tsu92], [Jou04], [KS05]), resulting in their noticeable difference from random oracles behavior. Originally, the class of hash functions SHA-2 [SHA-2] was assumed as a possible replacement for functions MD5 and SHA-1. However, hash function family SHA-2 is also vulnerable to all generic attacks; moreover, their design criteria have never been published. Not surprisingly, the attacks on SHA-2 are coming forward ([HPR04], [SKH04], [YB05], [YBP05], [MPRR06a], [MPRR06b]).

Furthermore, practical cryptanalysis of hash functions made a big progress in recent years. Serious weaknesses were discovered in many hash functions, especially in MD5, SHA-0 and SHA-1 ([MD5], [SHA-0], [SHA-1]). Generic attacks on the currently strongest hash functions ([Tsu92], [Jou04], [KS05]) and practical attacks on functions from classes MD and SHA showed a necessity of a new hash function concept proposal ([Sch04]).

At CRYPTO 1996, Bellare et al. [BCK96] proposed the constructions NMAC/HMAC. At CRYPTO 2005 Coron et al. [CDMP05] examined NMAC construction with two oracles and a HMAC construction with an ideal block cipher in the Davies-Meyer form. They proved that these constructions become indifferentiable from random oracles as their block length increases. In this paper, we prove, for the first time, quantitative estimates of resistance of these constructions against preimage and collision attacks. It follows from Theorems 1 to 4, that the attacker has to do roughly 2^n operations for finding a preimage or $2^{n/2}$ operations for finding a collision of NMAC/HMAC, which is the same estimate as for a random oracle. Thus NMAC/HMAC constructions proposed in 1996 by Bellare et al. [BCK96], become practical as well as theoretical based candidates for the new generation of hash functions.

Recently, Bellare [Bel06] showed that it is even possible to weaken traditional requirements on a compression function in the HMAC construction. For instance, it suffices that the compression function is a pseudorandom one for HMAC to be pseudorandom.

Since NMAC/HMAC are computationally resistant to preimage and collision attacks, generic attacks (discovered by Joux and Kelsey-Schneier [Jou04], [KS05]) are not to be feared of. The remaining generic attack is the message extension attack. Gauravaram et al. [GHA06] presented this attack only for a very artificial form of the NMAC interior functions.

The second part of the paper deals with a practical construction of NMAC/HMAC functions and a design of the SNMAC hash function SNMAC. Contemporary attacks on hash functions in the MD and SHA families, including SHA-2, are caused by weak nonlinearities in the block cipher employed and its key expansion ([KML02], [BDK03], [HKK03], [KKH04], [SKH04], [KKL04], [BDK05], [HKL05], [KBP05], [MPRR06a], [MPRR06b], [YWYP06], [BDK07]). To avoid modern attacks, hash functions should remove these weak functions from their design and replace them by a contemporary block cipher *technology* [Bih05]. By the word technology, we mean several well-established and proven principles and building blocks of block ciphers. If we use this technology, we obtain a hash function as on Fig. 1.

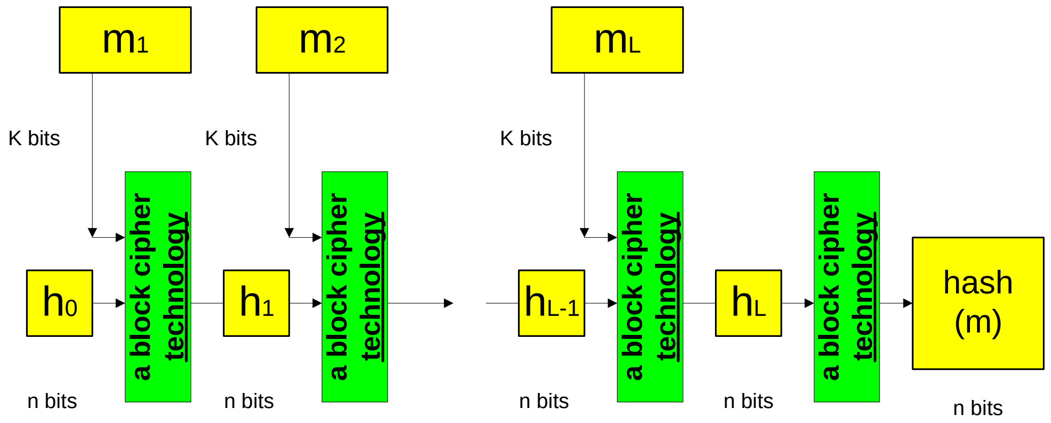
m1
m2
mL
K bits
K bits
K bits
h0
h1
hL-1
hL
a block cipher technology
a block cipher technology
a block cipher technology
a block cipher technology
hash (m)
n bits
n bits
n bits
n bits
n bits

Fig. 1: Hash function based on block cipher technology

We show that block ciphers should be used in hash functions in another way than we have seen so far. We call them special block ciphers (SBC) and we formulate their properties. This new cryptographic primitive surpasses the classical conception of a block cipher. The basic property of SBC is that an attacker can fully control its key. The block ciphers have not been designed with this requirement, yet. Therefore, contemporary block ciphers are not too suitable for being used in hash functions. We have to subordinate the design of these block ciphers to the aforesaid new demand that the attacker has full control over the plaintext and the key. SBC is not only a theoretical conception, a practical example of it is given in Appendix 2 and [Kli06b].

NMAC/HMAC security proofs are based on the facts that f and g are independent random oracles (in NMAC) and E is an ideal block cipher (in HMAC). If we dispose of a special block cipher, we can use it directly as a random oracle in the NMAC construction (Fig. 3), which is more general than HMAC construction (Fig. 4). We call this construction SNMAC (Special NMAC) according to the usage of the special block cipher in the NMAC model. Note that HMAC model uses a classical block cipher. If we replaced it by a special block cipher in a meaningful way, we would obtain SNMAC also. So SNMAC construction is a kind of trade-off between HMAC and NMAC. We will get it from below by "strengthening" HMAC (using a special block cipher instead of classical one) or from above by "weakening" NMAC (using a special block cipher instead of ideal random oracles).

We propose the SNMAC conception as a candidate for the new generation of hash functions. It is computationally resistant against preimage and collision attacks, in the limit it is indifferentiable from a random oracle and its construction enables variable designs using various SBC. SNMAC uses the special block cipher in the compression function in a special way, as prescribed the expression $h_i = \mathrm{SBC}_{h_{i-1} \,||\, m_i}(\mathrm{Const}_0)$. It exploits the fact that for decades the block ciphers were designed in such a way that it was computationally infeasible to determine the encryption key from the knowledge of any amount of plaintexts and ciphertexts. Consequently, the construction SNMAC is protected against preimage attack inherently, as the preimage of a compression function corresponds to the key of SBC. Furthermore, the construction uses the property that SBC with a fixed plaintext and a variable key behaves as a random oracle.

The paper is organized as follows: In Section 2, concepts of NMAC and HMAC are defined. In Section 3, the main theorems about security of NMAC/HMAC are presented. In Section 4, new concepts of block ciphers and hash functions, SBC and SNMAC, are introduced. Examples of instances of SBC and SNMAC are presented in Section 5. We conclude in Section 6. Proofs of main theorems are presented in Appendix 1. Appendices 2 and 3 contain the definition of a special block cipher DN (Double Net) and a hash function HDN (Hash Double Net) as a SNMAC construction based on DN.

2. Definitions of NMAC and HMAC

Basic construction. In practice, we meet the necessity to hash messages in parts - for instance, getting a message as a sequence from the communication channel and not having enough memory to save the whole stream. Let us imagine the hash function as a finite automaton. After processing a part of the message, we obtain some internal state of the automaton, which is called a context in the case of hash function. This context and the next part of the message is the input into the next step of the automaton. The starting state of the automaton we call initializing value. Thus, we obtain the basic model based on using the compression function *f* (Fig. 1). From the natural requirement that the compression function is defined for the constant input width, we obtain the necessity of message padding and its splitting into blocks with the same length. As a result, we obtain the classical Merkle-Damgard's model of an iterative hash function, which is the principle of all modern hash functions [Mer89] [Dam89].

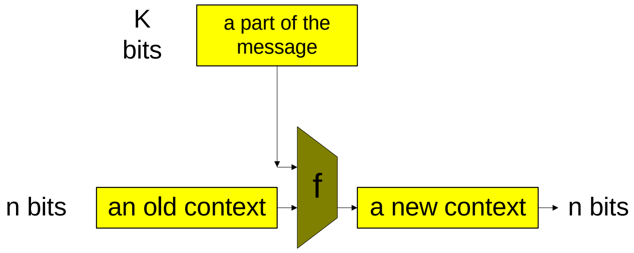

Fig. 2: An iterative hash function

Unfortunately, it is exactly this model that has those three generic weaknesses; independently on the content of the compression function *f*. Especially, it enables us to find multicollisions and multi-preimages in an easier way than in the case of a random oracle ([Jou04], [KS05]). However, we cannot leave the natural construction based on the iterative principle ([Mer89], [Dam89], [BCK96]). Therefore, we have to put up with the fact that the hash function of the new generation will not be theoretically resistant to multicollision and multi-preimage attacks. Appropriate defense should be here their computational complexity. We have to design these functions in such a way that the attacks would demand too many operations. In the construction on Fig. 3 and 4, the final conversion functions are used. This is a precaution against the third generic attack through the message extension. It will not avoid the attack theoretically [GHA06], but it will make it practically negligible. It will not be easy to use $h(M)$ for computing $h(M \| N)$ as the functions *f* and *g* are different (the case of NMAC). The computation of $h(M)$ ends by operation *g*, whereas in computing $h(M \| N)$ there is operation *f* used on its place. When we use two random oracles *f* and *g*, we obtain the construction NMAC (Fig. 3) according to [BCK96], [CDMP05]. If we build these oracles using a block cipher for instance in the Davies-Meyer form [MMO85], we will obtain construction HMAC (Fig. 4) according to [BCK96], [CDMP05]. Note that formally this definition of HMAC is a little different from the standardized one [RFC2104].

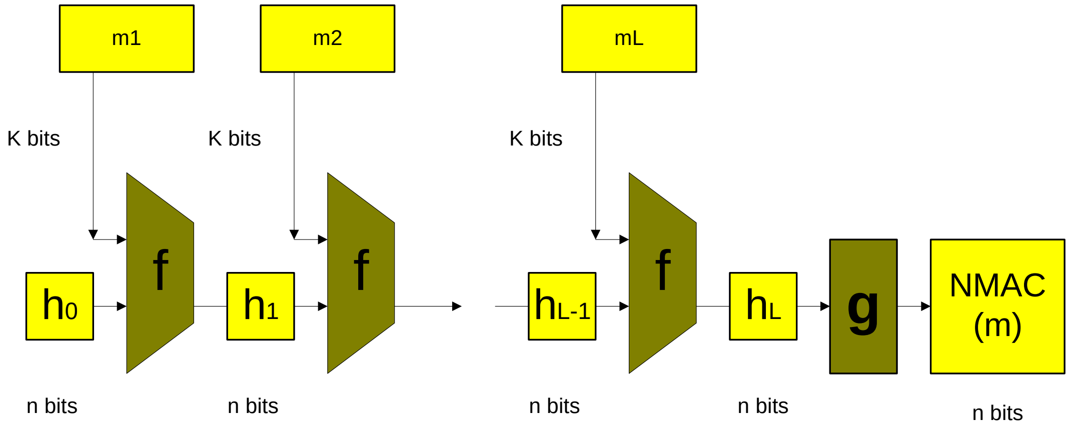

Fig.3: Definition of hash function NMAC (cf. [BCK96], [CDMP05])

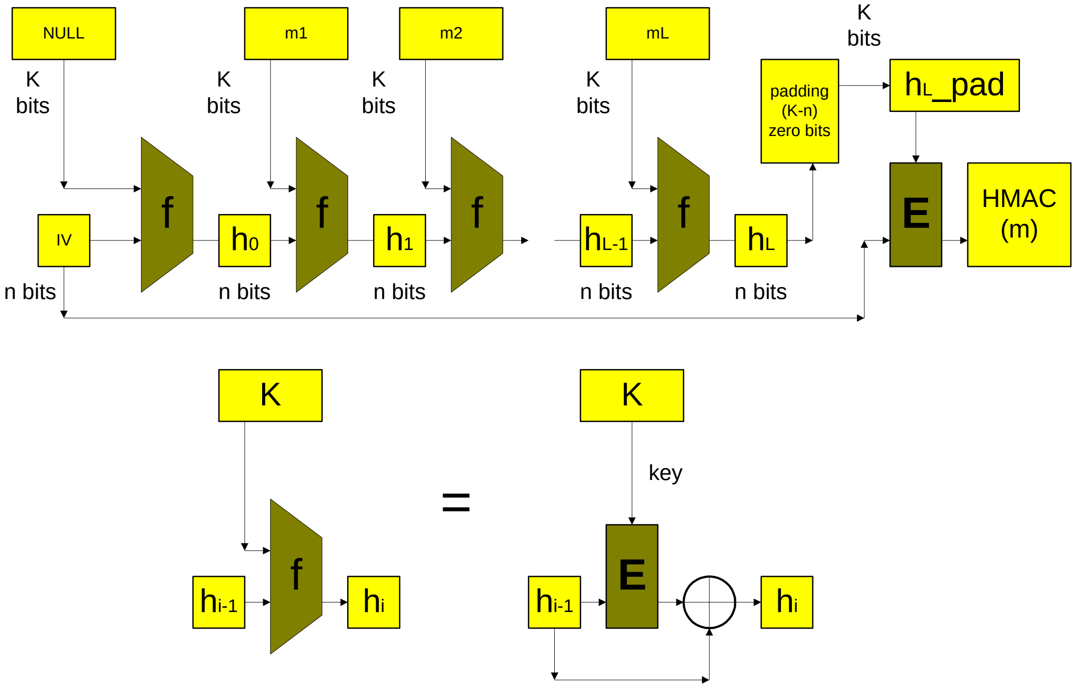

Fig. 4: Definition of hash function HMAC (cf. [BCK96], [CDMP05])

In both HMAC and NMAC models, we suppose that the message is padded (by bit 1, zero bits, length of the original message) to blocks of the same length of *K* bits, similarly as in the case of SHA-2.

3. Security of Hash Functions HMAC and NMAC

In this section, we present theorems on HMAC and NMAC resistance against preimage and collision attacks. Theorems 1 to 4 contain quantitative estimates of probabilities of finding a collision or a preimage, as a function of the number of operations at the attacker's disposal.
The estimates are very tight, since the lower and upper boundaries are of the same order. Proofs of Theorems 1 to 4 are presented in Appendix 1.

It follows from Theorems 1 to 4, that the attacker has to do roughly $2^{n/2}$ operations for finding a collision and roughly 2^n operations for finding a preimage. Therefore, the hash functions HMAC and NMAC behave as random oracles in these situations.

In the following, we will use a usual definition of the black-box model of a block cipher according to [BRS02].

3.1. Theorems on HMAC security

Let us denote BC(K, n) the set of all block ciphers E, which have K bits key and n bits block. Let E be a randomly chosen block cipher from the set BC(K, n), i.e. $E \xleftarrow{\$}$ BC(K, n), where the symbol $\xleftarrow{\$} M$ denotes a random choice of an object from the set M.

Black-box model ([BRS02], p. 322). Our model is the one dating to Shannon [S49] and used in [W84], [KR96], [EM91]. Let us fix a key length K and a block length n of a block cipher E. An adversary A is given an access to oracles E and E^{-1}, where E is a random block cipher E: $\{0, 1\}^K \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ and E^{-1} is its inverse. That is, each key $k \in \{0, 1\}^K$ denotes a randomly selected permutation $E_k = E(k, *)$ on $\{0, 1\}^n$, and the adversary is given oracles E and E^{-1}. The latter one, on input (k, y), returns the element x such that $y = E_k(x)$.
A_E_E$^{-1}(\sigma)$ denotes the algorithm chosen according to a parameter σ. Even though we will assume HMAC with $K \geq n$, some proofs are still valid for $K < n$, as well. For the final operation in HMAC, it is necessary to pad n bits long variable h_L to K bits long value. We denote this operation as h_{L}_*pad*, which means padding h_L by K - n zero bits. We denote HMAC^E (or shortly HMAC) the hash function HMAC based on a block cipher E according to Fig. 4.

Conventions ([BRS02], p. 328). For the remainder of this paper, we assume the following significant conventions. First, an adversary does not ask an oracle query for which the response is already known; namely, if A asks a query $E_k(x)$ and this returns y, then A does not ask a subsequent query of $E_k(x)$ or $E^{-1}{}_k$(y); and if A asks $E^{-1}{}_k$(y) and this returns x, then A does not ask a subsequent query of $E^{-1}{}_k(y)$ or $E_k(x)$. Second, when a (collision-finding) adversary A for HMAC outputs M and M', adversary A has already computed HMAC(M) and HMAC(M'), in the sense that A has made the necessary E or E^{-1} queries in all iterations during evaluation HMAC(M) and HMAC(M'). Similarly, when an (inverting adversary) A for HMAC outputs a message M, we assume that A has already computed HMAC(M), in the sense that A has made the necessary E or E^{-1} queries in all iterations during evaluation HMAC(M).

3.2. HMAC preimage resistance

Theorem 1. HMAC preimage resistance.

Let Pr_E_σ = Pr[E $\xleftarrow{\$}$ BC(K, n); σ $\xleftarrow{\$}$ $\{0, 1\}^n$; $M \longleftarrow$ A_E_E^{-1}(σ): HMAC^E(M) = σ] be the probability of the event that for a randomly chosen hash value σ and a randomly chosen block cipher E the adversary A_E_E^{-1} (using the algorithm A_E_E$^{-1}(\sigma)$) will obtain the value of the message M, where HMAC^E(M) = σ, i.e. he or she will find a preimage for σ. Let us denote Adv_inv_HMAC[n](q) = Max {Pr_E_σ} where the maximum is taken over all adversaries A_E_E$^{-1}(\sigma)$ that ask at most q oracle queries (i.e., E queries plus E^{-1} queries). Choose $n \in N$ and $K \geq n$. Then for any $1 \leq q < 2^n$

$$0.3 * q / 2^n \leq \text{Adv\_inv\_HMAC[n](q)} \leq 1.0 * q / 2^n.$$

3.3. HMAC collision resistance

Theorem 2. HMAC collision resistance.

Denote the advantage of *A* in finding collisions in HMAC as the real number Adv_coll_HMAC[n](A) = Pr[$E \xleftarrow{\$}$ BC(K, n); (M, M') $\xleftarrow{\$}$ A: $M' \neq M$ & HMAC$^{\mathrm{E}}$(M) = HMAC$^{\mathrm{E}}$(M')]. For any $1 \leq q$ we define Adv_coll_HMAC[n](q) = Max{Adv_coll_HMAC[n](A)} where the maximum is taken over all adversaries A_E_E^{-1} that ask at most q oracle queries (i.e., E queries plus E^{-1} queries). Choose $n \in N$ and $K \geq n \geq 3$. Then for any $1 < q \leq 2^{n/2}$

$$0.158 * q(q-2)/2^n \leq \text{Adv\_coll\_HMAC[n](q)} \leq 1.5 * q(q-1)/2^n.$$

3.4. Theorems on NMAC security

Let us denote RO(p, q) the set of all random oracles with p bits long input and q bits long output. We denote NMAC^{f,g} or shortly NMAC such a construction of NMAC defined above that uses random oracles $f \in$ RO($K + n$) and g$\in$ RO(n, n). Let A_f_g denotes an adversary (any algorithm), which has an access to oracles f and g. A_f_g(σ) denotes the algorithm chosen according to a parameter σ.

Conventions (cf. [BRS02], p. 328). In the case of NMAC, we assume the following significant conventions, similar to HMAC. When a (collision-finding) adversary *A* for NMAC outputs *M* and *M´*, the adversary *A* has already computed NMAC(*M*) and NMAC(*M´*) in such sense that *A* has made the necessary *f* and *g* queries in all iterations during the evaluation of NMAC(*M*) and NMAC(*M´*). Similarly, when an (inverting adversary) *A* for NMAC outputs a message *M,* we assume that *A* has already computed NMAC(*M*) in such a sense that *A* has made the necessary *f* and *g* queries in all iterations during the evaluation of NMAC(*M*).

3.5. NMAC preimage resistance

Theorem 3. NMAC preimage resistance.

Let Pr_f_g_σ = Pr[f $\xleftarrow{\$}$ RO(K + n, n); g $\xleftarrow{\$}$ RO(n, n); σ $\xleftarrow{\$}$ {0,1}^{n}; $M \longleftarrow$ A_f_g(σ): NMAC(M) = σ] be the probability of the event that for a randomly chosen hash value σ and a randomly chosen oracles f and g the adversary A_f_g (using the algorithm A_f_g(σ))obtains the value of the message M, such that NMAC(M) = σ, i.e. he or she finds a preimage for σ. Let us denote Adv_inv_NMAC[n](q) = Max {Pr_f_g_σ} where the maximum is taken over all adversaries A_f_g_σ that ask at most q oracle queries (i.e., f queries plus g queries). Choose $n \in N$. Then for any $1 \leq q < 2^n$

$$0.3 * q/2^n \leq \text{Adv\_inv\_NMAC[n] (q)} \leq 1.0 * q / 2^n.$$

3.6. NMAC collision resistance

Theorem 4. NMAC collision resistance.

Denote the advantage of *A* in finding collisions in NMAC as the real number Adv_coll_NMAC[n](A) = Pr[f $\xleftarrow{\$}$ RO(K + n, n); g $\xleftarrow{\$}$ RO(n, n); (M, M') $\xleftarrow{\$}$ A: $M' \neq M$ & NMAC(M) = NMAC(M')]. For any $1 \leq q$ define Adv_coll_NMAC[n](q) = Max {Adv_coll_NMAC[n](A)} where the maximum is taken over all adversaries A_f_g that ask at most q oracle queries (i.e., f queries plus g queries). Choose $n \in N$. Then for any $1 < q \leq 2^{n/2}$

$$0.158 * q(q-2)/2^n \leq \text{Adv\_coll\_NMAC[n](q)} \leq 0.5 * q(q-1)/2^n.$$

4. A New Concept of SBC and SNMAC

In this section, we introduce a concept of the special block cipher. Basing on the concept, we will define the hash function SNMAC. Recall the reasons that caused the problems of contemporary hash functions from MD and SHA families:

- block ciphers, used in compression functions, are processing the key and plaintext in fundamentally different ways (inhomogeneously),
- block ciphers, used in compression functions, enable controlling changes of one input (a plaintext or a key) by changes of the other one,
- component functions enable propagation of differences from inputs to differences in outputs,
- component functions are weakly nonlinear, there are highly probable linear relations between their inputs and outputs.

Biham [Bih05] proposed to start using block cipher technology in hash functions. We mean such building blocks that are strongly nonlinear and resistant to differential and linear cryptanalysis. So, let us assume that in the compression function f, $h_{\mathrm{i}} = \mathrm{f}(h_{i\text{-}1}, m_i)$, we will use a block cipher. We can even use it several times, if necessary.

In contemporary attacks on hash functions, the changes in $\mathrm{h}_{\mathrm{i\text{-}1}}$ and m_{i} are made simultaneously in such a way that the appropriate differences in h_{i} occur. Since the function f is built from a block cipher and the attacker is able to manipulate with all variables $(h_{i\text{-}1}, m_i)$ of f, he or she is able to manipulate with all variables, which enter the block cipher. Thus, in the case of hash functions, there is an extra situation that the attacker has a chance to manipulate both key and plaintext of the block cipher being used. This ability is independent on the way in which the block cipher is incorporated in the hash function.

There have been many ways studied on how to use block ciphers in the constructions of hash functions. Nevertheless, any classical block cipher has never been designed under the assumption that the attacker would have the chance to manipulate with its key howsoever. Conversely, the key is usually processed by weaker functions than the data entry in majority of modern ciphers. For instance, in the case of TripleDES it is a linear function, in the case of AES it is a weak nonlinear function.

Homogeneity. To guarantee the impossibility of using weaknesses either in the data entry or in the key entry processing, we demand all variable bits of used block cipher to be processed with the equal quality and in a similar way. We call this property the homogeneity. We also demand homogeneity for the output bits of the block cipher. An example of homogeneously processed input and output bits could be for instance a random substitution box (a permutation), despite of the fact that the output bit functions could be very different. Classical block ciphers almost never fulfill the requirement of homogeneity. Almost all of the modern block ciphers process the key by weaker functions than the functions, which process the data entry. On the other hand, the set of key bits and the set of data bits are both processed separately homogeneously almost every time. Thus we can achieve the requirement of homogeneity by setting the key or the data of a classical block cipher to a constant, whereas the remaining entry will be processed homogeneously.

Special block cipher (SBC) and a special NMAC (SNMAC). Let us achieve the property of homogeneity of a block cipher (E), used in a compressing function (f), by leading all input bits $X = h_{i\text{-}1} \,||\, m_i$ of a compressing function (i.e. data block m_i and a context $h_{i\text{-}1}$) to the plaintext and setting the key as a constant: $\mathrm{f}(X) = E_{\mathrm{Const0}}(X)$. The compression function f should be one-way to prevent preimage attack, however, this requirement is not satisfied by our construction. On the other hand, for decades, the block ciphers have been developed in such a way that it is computationally infeasible to derive the key from the knowledge of plaintext-ciphertext pairs. If we use this fact, we naturally obtain the construction f(X) = E_X(Const$_0$), i.e. all variable bits lead to the key entry of the block cipher and the block cipher is used only with a constant plaintext. Therefore, further on, we will assume only the construction f(X) = E_X(Const$_0$). In this case, we call E a special block cipher. This name is appropriate, because E is used only with two different constant plaintexts (Const$_0$ for the oracle f and Const$_1$ for the oracle g). On the basis of SBC and NMAC we can now define the hash function SNMAC, as is illustrated in Fig. 5.

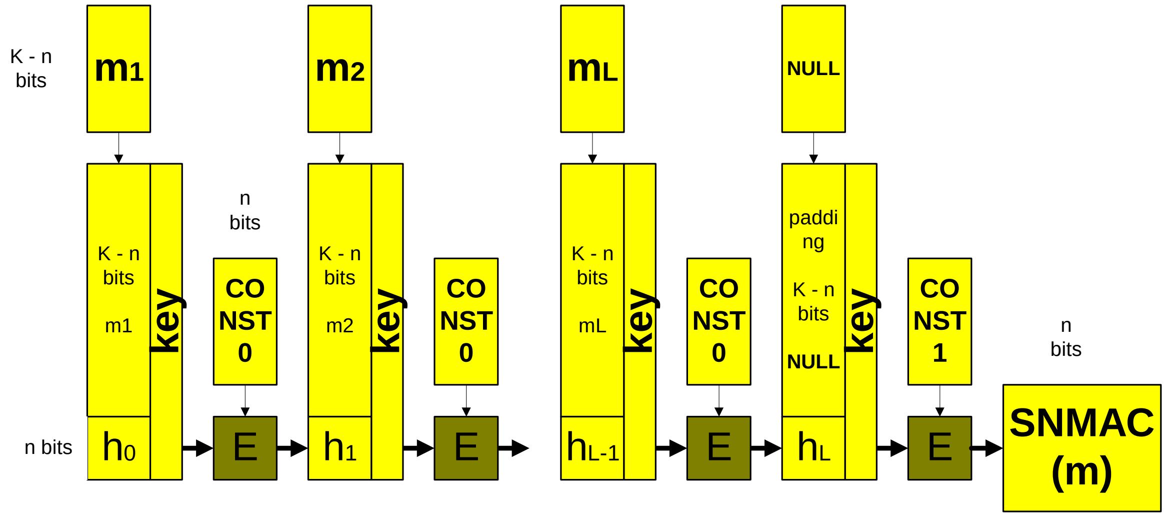

Fig. 5: Definition of SNMAC, based on SBC and NMAC

The concept of the special block cipher is new, so that its definition could be further refined. For the security proofs of SNMAC we will need the property that E: $\{0, 1\}^K$ x Const$_0$ → $\{0, 1\}^n$: (k, Const$_0$) → y = E_k(Const$_0$) = f(k) and E: $\{0, 1\}^K$ x Const$_1$ → $\{0, 1\}^n$: (k, Const$_1$) → y = E_k(Const$_1$) = g(k) are random oracles with respect to the variable key. For f and g to be high-quality functions for any choice of constants Const$_0$ a Const$_1$, we will demand that E: $\{0, 1\}^K$ x $\{0, 1\}^n$ → $\{0, 1\}^n$: (k, x) → y = E_k(x) is high quality as a whole mapping with a variable plaintext and a variable key.

All differential and linear attacks, which are successful in hash functions, are in the case of SBC transformed into differential and linear attacks using the key. Therefore, contrary to contemporary block ciphers, in the case of SBC we will especially demand the resistance against differential and linear attacks, leading from the key entry. We can extend this demand also to the data entry (as it was variable) and to a combination of key and data entry. So, we demand that there are no differential nor linear relations between variables (k, x) and y = E_k(x) with usable probability. In other words, the requirements on SBC are the same as on contemporary block cipher plus the necessity of stronger processing of the key. The key in SBC should be processed with the same cryptographic quality as the plaintext in classical block ciphers. So, what can we say about SBC?

A special block cipher E:

- processes the key on the same quality level as the data entry,
- processes all key bits on the same quality level (homogeneously),
- in opposite to classical block ciphers it will be natural to use it with the key length usually many times greater then the block length, e.g. $K = 4096$, resp. 8192 and $n = 256$, resp. 512,
- is designed using block cipher technology,
- is not primarily designed for data encryption,
- it is used in a hash function with a constant plaintext, all variable bits enter E through the key entry,
- assuming the SBC has also a variable plaintext, it should be cryptographically strong classical block cipher,
- the attacker is able to manipulate with the key in any way.

The definition of SBC is not complete yet, further research is necessary.

Definition. Hash function SNMAC. The hash function SNMAC is an iterative hash function of the NMAC type ([BCK96], [CDMP05]), which uses a *special block cipher E* with n bits long block and K bits long key. It has a compression function f and a final conversion g, where

f: $\{0, 1\}^K \rightarrow \{0, 1\}^n : X \rightarrow E_X(\text{Const}_0)$,

g: $\{0, 1\}^n \rightarrow \{0, 1\}^n : X \rightarrow E_{X \,||\, \text{NULL}}(\text{Const}_1)$,

$K \geq n$, Const_0 and Const_1 are different constants and NULL denotes the string of K - n zero bits. Hashing of a message m has three steps.

Step 1. Padding

We pad the message m by one bit 1, then by the smallest number (allowing an empty string) of bits 0 and by 128bit long number (which represents the length of the original message m in bits) in such a way, that the length of the padded message is the smallest (L) multiple of the number K - n, where L is an appropriate natural number. We divide this padded message into L blocks of K - n bits, $m = m_1 \,||\, ... \,||\, m_{L-1} \,||\, m_L$.

We define (Fig. 4) h_0 as a constant (initializing value).

Step 2. Iterations

$h_i = f(h_{i-1} \,||\, m_i)$, $i = 1, ..., L$,

Step 3. Final conversion

SNMAC( ) = $g(h_L)$.

The attacker's goal. The encryption key was the main goal in the case of the classical block ciphers. In the case of the special block cipher, the attacker has even the ability to manipulate with the key. So there is a question what is its goal now. Because hash function SNMAC is based on SBC, the attacker's goal will be a preimage or a collision of SBC. More generally, the goal will be the ability to control the relationship between the input and the output of the special block cipher in any way. It could lead to finding some properties that differentiate the hash function from random oracle. We will have to fix the classical block cipher. For the classical block ciphers, any manipulation with the key is quite an unnatural requirement. They are not prepared for it and don't have the defensive precautions against it. The key expansion is usually weak, incomparable with the data processing. Therefore, the key processing should be strengthened to the level of processing of the plaintext in contemporary block ciphers.

Why is it not desirable to use a high-quality classical block cipher in the hash function construction? From the work of Coron et al. [CDMP05] and Theorems in Section 3, it follows that the NMAC and HMAC constructions are computationally secure against collision and preimage. Would it not be therefore sufficient to use a high-quality classical block cipher in the HMAC construction? The answer is negative. The disadvantage of current block ciphers is that they process the key and data alternatively (by the different way), inhomogeneously and with different cryptographic strength. This inhomogeneity was used to attack the block ciphers and the hash functions (see for instance [BDK03], [BDK05], [HKL05], [KBP05], [KHL04], [KKH04], [KLS04], [KML02], [SKH04], [Kli06a], [YWYP06] and currently [BDK07]). From the present attacks on hash functions, it follows that the values of data and the contexts have the same cryptographic value and, therefore, they should be processed homogeneously (with the same quality). No one classical block cipher has this property. Also, none of the classical block ciphers was built with the assumption that the attacker has full control over the key. Thus the SBC design will be different from the classical block ciphers, even if it can use their proven building blocks.

5. A Concrete Instance of SBC and SNMAC

The SNMAC construction based on SBC is general and it enables to use various instances of SBC. As an instance of a SBC, we proposed the algorithm DN (Double Net). Using this algorithm in the SNMAC construction, we obtained the hash function called HDN (Hash Double Net). The descriptions of DN and HDN are presented in Appendices 2 and 3. Source codes, test samples, etc. will be available on [Kli06b]. DN has the key length 8192 bits and the block length 512 bits. HDN has 512-bits code and the unoptimized speed of hashing is 3 – 4 times slower than for SHA-512. The lower speed of HDN with respect to SHA-512 is comprehensible after a comparison of both functions. SHA-512 uses weaker internal nonlinear functions, whereas HDN integrates block cipher technology and a large security margin.

6. Conclusion

Generic problems of hash functions showed a need for a new hash function concept proposal. New security proofs enable to use the constructions NMAC/HMAC, which were proposed by Bellare et al. in 1996 [BCK96]. In this paper, we, for the first time, prove quantitative estimations of resistance of these constructions against preimage and collision attacks. It also follows from here that they are computationally resistant to multicollisions and multi-preimages, as well. Coron et al. [CDMP05] at CRYPTO 2005 showed that NMAC/HMAC are random oracles in the limit. Together with the quantitative proofs proposed, this gives very good guarantees to security of these constructions.

The second part of the paper deals with the practical construction of functions NMAC/HMAC and the proposals of hash function SNMAC on the basis of a block cipher. We show that block ciphers should be used in hash functions in another way than it has been so far. We call them special block ciphers (SBC) and we formulate their properties. This new cryptographic primitive surpasses the classical conception of block ciphers. The basic property of SBC is that an attacker has full control over its key. With this requirement the block ciphers have not been designed yet. Therefore, contemporary block ciphers are not too suitable for being used in hash functions.

We propose a new class of hash functions called SNMAC as NMAC construction using a special block cipher. This concept is a candidate for the new generation hash functions. It is computationally resistant to preimage and collision attacks, it is a random oracle in the limit and it enables various proposals using different instances of the special block cipher.

As an example we also propose a special block cipher DN (Double Net) and define a hash function HDN (Hash Double Net) as the SNMAC construction based on DN.

Acknowledgements. I am grateful to Tomáš Rosa for many helpful comments and inspiring suggestions on the previous versions of the paper.

7. References

[BCK96] M. Bellare, R. Canetti and H. Krawczyk. Keying hash functions for message authentication. Advances in Cryptology – CRYPTO ’96, Lecture Notes in Computer Science Vol. 1109, pp. 1-15, Springer-Verlag, 1996.

[Bel06] M. Bellare. New Proofs for NMAC and HMAC: Security without Collision-Resistance. To be published, Advances in Cryptology – CRYPTO ’06, Lecture Notes in Computer Science Vol. 4117, Springer-Verlag, 2006, Cryptology ePrint Archive, Report 2006/043.

[BCJ05] E. Biham, R. Chen, A. Joux, P. Carribault, Ch. Lemuet and W. Jalby. Collisions of SHA-0 and Reduced SHA-1. Advances in Cryptology –EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 36–57, Springer-Verlag, 2005.

[BDK03] E. Biham, O. Dunkelman, and N. Keller. Rectangle Attacks on 49-Round SHACAL-1, FSE 2003, Lecture Notes in Computer Science Vol. 2887, pp. 22-35, Springer-Verlag, 2003.

[BDK05] E. Biham, O. Dunkelman, and N. Keller. Related-Key Boomerang and Rectangle Attacks, Advances in Cryptology – EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 507–525, Springer-Verlag, 2005.

[BDK07] E. Biham, O. Dunkelman, and N. Keller. A Simple Related-Key Attack on the Full SHACAL-1, to be published, CT-RSA 2007, RSA Conference 2007, Cryptographers' Track, February 5-9, 2007, Moscone Center, San Francisco, USA.

[Bih05] E. Biham: Recent advances in hash functions and the way to go, Conference on Hash Functions (Ecrypt Network of Excellence in Cryptology), June 23-24, 2005, Przegorzaly (Krakow), Poland, http://www.ecrypt.eu.org/stvl/hfw/Biham.ps.

[BRS02] J. Black, P. Rogaway, T. Shrimpton. Black-Box Analysis of the Block-Cipher-Based Hash-Function Constructions from PGV. Advances in Cryptology – CRYPTO 2002, Lecture Notes in Computer Science Vol. 2442, pp. 320-335, Springer-Verlag, 2002. Extended version: Cryptology ePrint Archive, Report 2002/066, http://eprint.iacr.org/2002/066.

[CDMP05] J. S. Coron, Y. Dodis, C. Malinaud and P. Puniya. Merkle-Damgard Revisited: how to construct a hash-function. Advances in Cryptology – CRYPTO 2005, Lecture Notes in Computer Science Vol. 3621, pp. 430 - 448, Springer-Verlag, 2005.

[Dam89] I. Damgard. A Design Principle for Hash Functions. Advances in Cryptology - CRYPTO 1989, Lecture Notes in Computer Science Vol. 435, pp. 416–427, Springer-Verlag, 1990.

[EM91] S. Even and Y. Mansour. A construction of a cipher from a single pseudorandom permutation. In Advances in Cryptology – ASIACRYPT ’91, Lecture Notes in Computer Science Vol. 739, pp. 210–224. Springer-Verlag, 1992.

[GHA06] P. Gauravaram, S. Hirose and S. Annadurai. An Update on the analysis and design of NMAC and HMAC functions. To be published in International Journal of Network Security

[HPR04] P. Hawkes, M. Paddon, and G. G. Rose. On Corrective Patterns for the SHA-2 Family. Cryptology ePrint Archive, Report 2004/207, 2004.

[HKK03] S. Hong, J. Kim, G. Kim, J. Sung, C. Lee and S. Lee. Impossible Differential Attack on 30-Round SHACAL-2, INDOCRYT 2003, Lecture Notes in Computer Science Vol. 2904, pp. 97-106, Springer-Verlag, 2003.

[HKL05] S. Hong, J. Kim, S. Lee and B. Preneel. Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192, FSE 2005, Lecture Notes in Computer Science Vol. 3557, pp. 368–383, Springer-Verlag, 2005.

[Jou04] A. Joux. Multicollisions in Iterated Hash Functions. Advances in Cryptology - CRYPTO 2004, Lecture Notes in Computer Science Vol. 3152, pp. 306–316, Springer-Verlag, 2004.

[KML02] J. Kim, D. Moon, W. Lee, S. Hong, S. Lee, and S. Jung. Amplified Boomerang Attack against Reduced-Round SHACAL, Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science Vol. 2501, pp. 243 - 253, Springer-Verlag, 2002.

[KBP05] J. Kim, A. Biryukov, B. Preneel, and S. Lee. On the Security of Encryption Modes of MD4, MD5 and HAVAL, ICICS 2005, Lecture Notes in Computer Science Vol. 3783, pp. 147-158, Springer-Verlag, 2005.

[KK05] J. Kelsey and T. Kohno.Herding Hash Functions and the Nostradamus Attack, Cryptographic Hash Workshop, held in NIST, Gaithersburg, Maryland, 2005, IACR Cryptology ePrint Archive, Report 2005/281, 2005.

[KKH04] J. Kim, G. Kim, S. Hong, S. Lee and D. Hong.The Related-Key Rectangle Attack-Application to SHACAL-1, ACISP 2004, Lecture Notes in Computer Science Vol. 3108, pp. 123-136, Springer-Verlag, 2004.

[KKL04] J. Kim, G. Kim, S. Lee, J. Lim and J. Song. Related-Key Attacks on Reduced Rounds of SHACAL-2, INDOCRYPT 2004, Lecture Notes in Computer Science Vol. 3348, pp. 36 - 44, Springer-Verlag, 2004.

[Kli06a] V. Klima. Tunnels in Hash Functions: MD5 Collisions Within a Minute, Cryptology ePrint Archive, Report 2006/105, 18 March, 2006.

[Kli06b] V. Klima. SNMAC homepage http://cryptography.hyperlink.cz/SNMAC/SNMAC.html.

[KLS04] J. Kim, G. Kim, S. Lee, J. Lim and J. Song. Related-Key Attacks on Reduced Rounds of SHACAL-2, INDOCRYPT 2004, Lecture Notes in Computer Science Vol. 3348, pp. 36 - 44, Springer-Verlag, 2004.

[KML02] J. Kim, D. Moon, W. Lee, S. Hong, S. Lee, and S. Jung. Amplified Boomerang Attack against Reduced-Round SHACAL, Advances in Cryptology - ASIACRYPT 2002, Lecture Notes in Computer Science Vol. 2501, pp. 243 - 253, Springer-Verlag, 2002.

[KR96] J. Kilian and P. Rogaway. How to protect DES against exhaustive key search. Journal of Cryptology, 14(1):17–35, 2001. Earlier version in CRYPTO '96.

[KS05] J. Kelsey and B. Schneier. Second Preimages on n-Bit Hash Functions for Much Less than 2^n. Advances in Cryptology - EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 474–490, Springer-Verlag, 2005.

[MD5] R. Rivest. The MD5 message-digest algorithm, Internet RFC 1321, April 1992.

[Mer89] R. C. Merkle. One Way Hash Functions and DES. Advances in Cryptology - CRYPTO 1989, Lecture Notes in Computer Science Vol. 435, pp. 428–446, Springer-Verlag, 1990.

[MMO85] S. M. Matyas, C. H. Meyer and J. Oseas. Generating strong one-way functions with cryptographic algorithm. IBM Techn. Disclosure Bull., Vol. 27, No. 10A, 1985, pp. 5658 - 5659.

[MPRR06a] F. Mendel, N.Pramstaller, C.Rechberger, and V.Rijmen. Analysis of Step-Reduced SHA-256, to be published, FSE 2006

[MPRR06b] F.Mendel, N.Pramstaller, C.Rechberger, and V.Rijmen. The Impact of Carries on the Complexity of Collision Attacks on SHA-1, to be published, FSE 2006

[S49] C. Shannon. Communication theory of secrecy systems. Bell Systems Technical Journal, 28(4):656–715, 1949.

[Sch04] B. Schneier. Cryptanalysis of MD5 and SHA. Crypto-Gram Newsletter, September 2004, http://www.schneier.com/crypto-gram-0409.html#3

[SHA-0] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180, May 1993.

[SHA-1] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180-1, April 1995.

[SHA-2] National Institute of Standards and Technology. Secure hash standard. Federal Information Processing Standard, FIPS PUB 180-2, August 2000.

[SKH04] Y. Shin, J. Kim, G. Kim, S. Hong and S. Lee. Differential-Linear Type Attacks on Reduced Rounds of SHACAL-2, ACISP 2004, Lecture Notes in Computer Science Vol. 3108, pp. 110–122. , Springer-Verlag, 2004.

[Tsu92] G. Tsudik. Message authentication with one-way hash functions. ACM Computer Communications Review, 22(5):29-38, 1992.

[W84] R. Winternitz. A secure one-way hash function built from DES. In Proceedings of the IEEE Symposium on Information Security and Privacy, pp. 88–90. IEEE Press, 1984.

[WY05] X. Wang and H. Yu. How to Break MD5 and Other Hash Functions. Advances in Cryptology - EUROCRYPT 2005, Lecture Notes in Computer Science Vol. 3494, pp. 19–35, Springer-Verlag, 2005.

[WYY05a] X. Wang, H. Yu and Y. L. Yin. Efficient Collision Search Attacks on SHA-0. Advances in Cryptology - CRYPTO ’05, Lecture Notes in Computer Science Vol. 3621, pp. 1–16, Springer-Verlag, 2005.

[WYY05b] X. Wang, Y. L. Yin and H. Yu. Finding collisions in the full SHA-1. Advances in Cryptology - CRYPTO ’05, Lecture Notes in Computer Science Vol. 3621, pp. 17–36, Springer-Verlag, 2005.

[YB05] H. Yoshida and A. Biryukov. Analysis of a SHA-256 Variant, SAC 2005, Lecture Notes in Computer Science Vol. 3897, pp. 245 – 260, Springer-Verlag, 2005.

[YBP05] H. Yoshida, A. Biryukov, and B. Preneel. Some applications of the Biham-Chen attack to SHA-like hash functions, CRYPTOGRAPHIC HASH WORKSHOP, NIST, Gaithersburg, Maryland, USA, October 31 - November 1, 2005, http://csrc.nist.gov/pki/HashWorkshop/2005/Oct31_Presentations/Yoshida_cameraNistHash.pdf

[YWYP06] H.Yu, X.Wang, A.Yun and S. Park. Cryptanalysis of the Full HAVAL with 4 and 5 Passes. To be published, FSE 2006.

8. Appendix 1: Proofs of Theorems

In our proofs, we will use the following lemma.

Lemma 1.

(1) for every $x \in$ <0, 1> it holds that $1 - e^{-x} \geq (1 - e^{-1})\, x$,

(2) for every $x \in$ <0, 1> it holds that $e^{-x} \geq 1 - x$,

(3) for every $x \in (0, 1)$ and $q \in N$ it holds that $1 - qx \leq (1 - x)^q$,

(4) for every $x \in (0, 1)$ and $q \in N$ it holds that $(1 - x/q)^q \leq e^{-x}$.

Proof. It follows from properties of power and exponential functions.

8.1. Proof of Theorem 1

We will show that it holds $0.3 * q / 2^n \leq \text{Adv\_inv\_HMAC[n](q)}$.

Let E be a block cipher chosen randomly from the set BC(K, n) and σ is a value chosen randomly from the set $R = \{0, 1\}^n$. It is sufficient to prove that Pr_E_σ ≥ 0.3 * $q/2^n$ for a particular adversary A defined at our will. Let us define our adversary. The adversary A looks for K bits long block m, for which $h_1 = E_m(h_0)$ xor h_0 and $E_{h1\_pad}(\mathrm{IV}) = \sigma$. While doing this, she asks the oracle E at most q times at all. She does not ask the oracle E^{-1}. For every $1 \leq i \leq q/2$ she chooses the key arbitrarily and asks the oracle E for the value of $E_{ki}(h_0)$. She computes $y_\mathrm{i} = E_{k\mathrm{i}}(h_0)$ xor h_0 and obtains the value $H_i = \mathrm{HMAC}^\mathrm{E}(k_i) = E_{\mathrm{yi\_pad}}(\mathrm{IV})$ from the oracle. If $H_i = \sigma$ for some i, the adversary found preimage of hash value σ and returns $M = \mathrm{k_i}$ as the message. If $E_{\mathrm{yi\_pad}}(\mathrm{IV})$ is not equal to σ for any $1 \leq i \leq q/2$, she returns a negative answer. From the definition of the random block cipher, it follows that if we fix the plaintext x (here $x = h_0$ or $x = \mathrm{IV}$), then the mapping $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E_k(x)$ is a random oracle with K bits long input and n bits long output. Therefore, $\{y_i\}_{1 \leq i \leq q}$ contains $q/2$ independent random values from the set $\{0, 1\}^n$ and $\{y_i\_pad\}_{1 \leq i \leq q/2}$ is the set of $q/2$ independent values from the set $\{0, 1\}^K$. Under the same assumption (now $x = \mathrm{IV}$), it follows that $\{H_i\}_{1 \leq i \leq q/2}$ is the set of $q/2$ randomly chosen values from the set $\{0, 1\}^n$. The probability P that there is the value σ in the set $\{H_i\}_{1 \leq i \leq q/2}$ is equal to

$$P = 1-\left(1-\frac{1}{2^n}\right)^{q/2} \geq^{(4)} 1-e^{-q/2^{n+1}} \geq^{(1)} (1-e^{-1})*q/2^{n+1} \geq 0.3*q/2^n,$$

using Lemma 1 (4), (1), and the fact that $q/2^{n+1} < 1$, qed.

Now we will show that $\mathrm{Adv\_inv\_HMAC[n](q)} \leq 1.0*q/2^n$.

Let E be a randomly chosen block cipher from the set $\mathrm{BC}(K, n)$ and σ is a randomly chosen value from the set $R = \{0, 1\}^n$. Let A denote an adversary. It is sufficient to prove that Pr_E_σ $\leq q/2^n$. *E*ven if the adversary has the best strategy, he can use only queries to oracles E and E^{-1}. During the activity of the adversary A, the oracle E is creating a list of records of queries and answers (k_i, x_i, y_i) and the oracle E^{-1} is creating the list of (k_j, x_j, y_j), where the input of E is (k_i, x_i) and the output is $y_i = E_{ki}(x_i)$; the input of E^{-1} is (k_j, y_j) and output is $x_j = E^{-1}_{kj}(y_j)$. In these lists, we distinguish the cases $x_i = \mathrm{IV}$, $x_i \neq \mathrm{IV}$, $y_j = \sigma$, and $y_j \neq \sigma$. Altogether we have
q_1 queries to E of the type (k_i, IV, y_i),
q_3 queries to E of the type $(k_i, \neq\mathrm{IV}, y_i)$,
q_2 queries to E^{-1} of the type (k_j, x_j, σ),
q_4 queries to E^{-1} of the type $(k_j, x_j, \neq\sigma)$,
where $q_1 + q_2 + q_3 + q_4 \leq q$.

If the algorithm A is successful, then there is at least one record, where $y_i = \sigma$ in the list (q_1) or there is at least one record, where $x_j = \mathrm{IV}$ in the list (q_2). It is unnecessary to examine queries in the lists (q_3) and (q_4). From the definition of the random block cipher it follows that if we fix the plaintext x (here $x = \mathrm{IV}$), then the mapping $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E_k(x)$ is a random oracle with K bits long input and n bits long output. Therefore the set $\{y_i\}_{1 \leq i \leq q1}$ from the list (q_1) is the set of q_1 random values from the set $\{0, 1\}^n$. From the definition of random block cipher it follows that if we fix the ciphertext y (here $y = \sigma$ for any σ) then the mapping $\{0,1\}^K \rightarrow \{0,1\}^n : k \rightarrow E^{-1}_k(\sigma)$ is a random oracle with K bits long input and n bits long output. Therefore, the set $\{x_j\}_{1 \leq j \leq q2}$ from the list (q_2) is the set of q_2 randomly chosen values from the set $\{0, 1\}^n$. The probability P that there is the value σ in the set $\{y_i\}_{1 \leq i \leq q1}$ or that there is a value IV in the set $\{x_j\}_{1 \leq j \leq q2}$ is equal to

$$P = 1-\left(1-\frac{1}{2^n}\right)^{q_1} * \left(1-\frac{1}{2^n}\right)^{q_2} \leq 1-\left(1-\frac{1}{2^n}\right)^{q} \leq^{(3)} q/2^n.$$

We used here Lemma 1 (3). This finishes the proof of Theorem 1.

8.2. Proof of Theorem 2

We will show that it holds $0.158 * (q-2)/2^n \le \text{Adv\_coll\_HMAC[n](q)}$

It is sufficient to prove that $\text{Adv\_coll\_HMAC[n](A)} \ge 0.158 * q(q-2)/2^n$ for a particular adversary *A* defined at our will. Let us define her. The adversary *A* looks for colliding messages $x_i \neq x_j$, which contain only one *K* bit long block including the padding:
$y_i = E_{xi}(h_0)$ and $E_{(yi \oplus h0)\_\mathrm{pad}}(\mathrm{IV}) = \sigma$ and
$y_j = E_{xj}(h_0)$ and $E_{(\mathrm{yj} \oplus h0)\_\mathrm{pad}}(\mathrm{IV}) = \sigma$ for some σ.
Note that the collision can occur

(1) after the encryption of h_0 ($y_i = y_j$) or
(2) in the second step after the encryption of IV by different keys $(\mathrm{y_i} \oplus \mathrm{h_0})$_pad and $(\mathrm{y_j} \oplus h_0)$*_pad*.

Procedure of the adversary corresponds to these two possibilities. The adversary asks the oracle *E* at most *q* times at all. She makes q_1 queries for the encryption of the value h_0 and q_2 queries for the encryption of the value IV, $q_1 = q_2 = q/2$. For *i*, $1 \le i \le q_1$, the adversary *A* chooses *K* bits long keys k_i arbitrarily and will obtain the values $y_\mathrm{i} = E_{ki}(h_0)$ from the oracle, what creates the list (k_i, h_0, y_i). If there are two equal values y_i and y_j in the third position, she obtains the collision for messages k_i and k_j. If not, the list contains q_1 different random values in the first item and the adversary continues by creating the second list. For every $1 \le i \le q_2$ she takes values y_i from the first list and from the oracle *E* she obtains values $Y_i = E_{(yi \oplus h0)\_\mathrm{pad}}(\mathrm{IV})$. She creates the list $((y_i \oplus h_0)\_\mathrm{pad}, \mathrm{IV}, Y_i)_{1 \le i \le q2}$. If there are two equal values Y_i and Y_j in the third position, she will obtain the collision for messages k_i and k_j. If not, the adversary returns a negative response (collision not found). Let us denote *p* the probability that the adversary *A* will find a collision by this procedure, and *P* that she does not. From the definition of the random block cipher, it follows that if we fix the plaintext *x* (here $x = h_0$ and $x = \mathrm{IV}$), then the mapping $\{0,1\}^K \to \{0,1\}^n : \mathrm{k} \to E_k(x)$ is a random oracle with *K* bits long input and *n* bits long output. Therefore, $\{y_i\}_{1 \le i \le q1}$ is the set of q_1 random values from the set $\{0, 1\}^n$ and $\{(y_i \oplus h_0)\_\mathrm{pad}\}\}_{1 \le i \le q1}$ is the set of q_1 values from the set $\{0, 1\}^K$. Under the same assumption $\{Y_i\}_{1 \le i \le q2}$ is the set of q_2 random values from the set $\{0, 1\}^n$. The probability that the adversary does not find a collision either in q_1 steps (1) or in q_2 steps (2) is

$$P = \prod_{i=1}^{q1-1}\left(1-\frac{i}{2^n}\right) * \prod_{i=1}^{q2-1}\left(1-\frac{i}{2^n}\right) \le^{(2)} \prod_{i=1}^{q1-1}\left(e^{-i/2^n}\right) * \prod_{i=1}^{q2-1}\left(e^{-i/2^n}\right) =$$

$$= e^{-q1(q1-1)/2^{n+1} - q2(q2-1)/2^{n+1}} = e^{(-q(q-2)/4)/2^n}.$$

We used here the fact (2) from Lemma 1 and that $q_1 = q_2 = q/2$. The probability that there will be a collision, is $p = 1 - P \ge 1 - e^{(-q(q-2)/4)/2^n} \ge^{(1)} (1-e^{-1}) * (q(q-2)/4)/2^n \ge 0.158 * q(q-2)/2^n.$ We used here the fact (1) from Lemma 1, i.e. $1 - \mathrm{e}^{-x} \ge (1-\mathrm{e}^{-1})x$ for all $0 \le x \le 1$. In the role of *x*, there is the expression $x = (q(q\text{-}2)/4)/2^n$. According to our assumptions, we have $q \le 2^{n/2}$, so that $x \le 1/4$, i.e. $x \le 1$ and the treatment was correct. The proof is finished.

Now we will show that $\text{Adv\_coll\_HMAC[n](q)} \le 1.5 * q(q-1)/2^n.$

We have to prove $\text{Adv\_coll\_HMAC[n](A)} \le 1.5 * q(q-1)/2^n$ for any adversary *A*. Even if the adversary has the best strategy, she can use only queries to oracles *E* and E^{-1}. We can model the activity of the adversary A by her queries. We create the table T of records (x, h, y, $h \oplus y$), where the adversary chooses the key x and one of the values y or h. She will obtain $y = E_x(h)$ from the oracle E or $h = E^{-1}_x$(y) from the oracle E^{-1}. The fourth item ($h \oplus y$) is for an informative purpose only. The table T can have at most q records. Because the attacker will not ask the oracles if she knows the answer, we can assume that there are no two records with the same values (x, h) or (x, y). The attack does not succeed if q records fulfill the table. The attacker is searching for colliding messages $M \neq M'$, which have several K bits long blocks including the padding. The messages can differ in a number of blocks. If there is a collision, there are two cases:

- **Internal collision.** In this case, the collision occurred before the final operation.
- **Final collision.** In this case, the collision occurred after the final operation and it has occurred nowhere before the final operation.

Internal collision

Let us elaborate the internal collision first. In this case, there has to be created a record (x_i, h_{i-1}, y_i, $h_{i\text{-}1} \oplus y_i$) in the table T during processing the first message and a record (x_j, $h_{j\text{-}1}$, y_j, $h_{j\text{-}1} \oplus y_j$), during processing the second message, where $i \neq j$ are indexes of oracle's records (they are not indexes of blocks of messages), (x_i, $h_{i\text{-}1}$) ≠ (x_j, $h_{j\text{-}1}$) and $h_{i\text{-}1} \oplus y_i = h_{j\text{-}1} \oplus y_j$. For a fixed i = 2, ..., q, let us denote $\mathrm{C_i}$ the event that there is an index j, $1 \leq j < i$ such that j-th and i-th records in the table have the same values in the fourth position. Denote Pr[$\mathrm{C_i}$] probability of this event. Let us denote t the number of records j in the table T, $1 \leq j < i$, for which $x_j = x_i$ in the first position and $h_{j\text{-}1} \neq h_{i\text{-}1}$ in the second position. It holds $0 \leq t \leq\ i$ - 1. In these records, we have the same key (x_i) and t different values h in the second position. So we already have recorded t different values $E_{xi}(h)$ in the third position for a given key x_i and for t different values h. Therefore the answer $y_\mathrm{i} = E_{xi}(h_\mathrm{i})$ on i-th query (h_i) the oracle chooses (randomly) from the set of 2^n - t values. Because h_i is a constant, the expression $y_i \oplus h_i$ is also taken randomly from the set of 2^n - t values. The probability that it equals to one value from the set of (at most) (i - 1) values in the fourth position of the table T, is less or equal to (i - 1)/ (2^n - t). So it holds $\Pr[\mathrm{C_i}] \leq (i-1)/(2^n - t) \leq (i-1)/(2^n - (i-1))$. We used the fact that from the definition of the random block cipher it follows that for any fixed $k \in \{0,1\}^K$ the mapping $\{0,1\}^n \rightarrow \{0,1\}^n : x \rightarrow E_k(x)$ is a random permutation on $\{0,1\}^n$. Because $i \leq q \leq 2^{n/2} \leq 2^{n-1}$ +1, it holds $\Pr[\mathrm{C_i}] \leq (i-1)/(2^n - (i-1)) \leq (i-1)/2^{n-1}$. Let P_{int} denotes the probability that there is an internal collision in the table T. We have

$$P_{\text{int}} = \Pr[C_2] + \Pr[C_3] + ... + \Pr[C_q] \leq \sum_{i=2}^{q} \frac{i-1}{2^{n-1}} = q(q-1)/2^n.$$

Final collision

Now, we will assume that there is a final collision and no internal collision. Let us denote P_{fin} the probability of such an event. For the simplicity, we assume that in the table T there are only records with the value IV in the second position (we eventually decrease q). Since we assume that $K \geq n$, the key for the final operation is n bits long value h_N, which is eventually padded by zero bits to K bits long key. From the final collision, it follows that in the table T there are at least two records with different keys $x_i \neq x_j$ in the first position, with the same value IV in the second position and with the same value in the third position E_{xi}(IV) = E_{xj}(IV). For i = 2, ..., q, let us denote $\mathrm{C_i}$ the event that there is j, $1 \leq j < i$ such that j-th record and i-th record in the table have the same values in the third position, i.e. E_{xi}(IV) = E_{xj}(IV). Let Pr[$\mathrm{C_i}$] denotes the probability of that event. Since the mapping $\{0, 1\}^K \rightarrow \{0, 1\}^n : k \rightarrow E_k$(IV) is the random oracle, the value E_{xi}(IV) is chosen randomly from the set of all 2^n values, so that

$\Pr[\mathrm{C_i}] \le (i-1)/2^n$. If there is a final collision, then there an event $\mathrm{C_i}$, for some $2 \le i \le q$, must have occurred. From here, we have

$$P_{fin} \le \Pr[C_2] + \Pr[C_3] + ... + \Pr[C_q] \le \sum_{i=2}^{q} \frac{i-1}{2^n} \le q(q-1)/2^{n+1}.$$

Total estimate

If the adversary finds a collision, it has to be either internal or final. From here, it follows $\mathrm{Adv\_coll\_HMAC[n](A)} \le P_{\mathrm{int}} + P_{fin} \le q(q-1)/2^n + q(q-1)/2^{n+1} = 1.5 * q(q-1)/2^n, qed.$

8.3. Proof of Theorem 3

We will show that it holds $0.3 * q/2^n \le \mathrm{Adv\_inv\_NMAC[n]\,(q)}.$

It suffices to proof that Pr_f_g_σ ≥ 0.3 * $q/2^n$ for a particular adversary A defined at our will. Let us define our adversary. The adversary A looks for one K bits long block m, for which h_1 = f(m, h_0) and g(h_1) = σ. While doing this she asks q_1 times the oracle f and q_2 times the oracle g, where $q_1 = q_2 = q/2$. For every $1 \le i \le q_1$, she chooses k_i arbitrarily and obtains the value $y_i = f(k_i, h_0)$ from the oracle f and the value $H_i = g(y_i)$ from the oracle g. If $H_i = \sigma$ for some i, the adversary found a preimage of the hash value σ and returns as the message $M = k_i$. If H_i is not equal to σ for any $1 \le i \le q_1$, she returns a negative answer. Since f is a random oracle, the set $\{y_i\}_{1 \le i \le q1}$ contains q_1 random values from the set $\{0, 1\}^n$. Since g is a random oracle, the set $\{H_i\}_{1 \le i \le q2}$ is the set of q_2 randomly chosen values from the set $\{0, 1\}^n$. The probability P that there is the value σ in the set $\{H_i\}_{1 \le i \le q2}$ is equal to

$$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q_2} \ge^{(4)} 1 - e^{-q_2/2^n} \ge^{(1)} (1 - e^{-1}) * q_2/2^n \ge 0.6 * q_2/2^n = 0.3 * q/2^n,$$

using Lemma 1 (4), (1) and the fact that $q/2^n$<1, qed.

Now we will show that $\mathrm{Adv\_inv\_NMAC[n]\,(q)} \le q/2^n.$

Let A be an adversary (an algorithm) and σ be a hash value. It is sufficient to prove that Pr_f_g_σ ≤ $q/2^n$. During the activity of the adversary A, the oracle f is creating a list of records (x_i, h_i, z_i), where $z_i = f(x_i, h_i)$, and the oracle g is creating a list of q_2 records (h_j, y_j), where $y_j = g(h_j)$ and $q_1 + q_2 \le q$. If the algorithm A succeeds, then there is at least one record in the second list, where y_j = σ. From the definition of the random oracle g, it follows that the set $\{y_j\}_{1 \le j \le q2}$ is a set of q_2 randomly chosen values from the set $\{0, 1\}^n$. The probability P that there is the value σ in the set $\{y_j\}_{1 \le j \le q2}$ is equal to

$P = 1 - \left(1 - \frac{1}{2^n}\right)^{q_2} \le^{(3)} q_2/2^n \le q/2^n$ using Lemma 1 (3). The proof is finished.

8.4. Proof of Theorem 4

We will show that it holds

$$\mathrm{Adv\_coll\_NMAC[n](q)} \ge 0.158 * q(q-2)/2^n.$$

It is sufficient to prove that $\mathrm{Adv\_coll\_NMAC[n](A)} \ge 0.158 * q(q-2)/2^n$ for a particular adversary A defined at our will. Let us define her. The adversary A looks for colliding messages $x_i \ne x_j$, which contain only one K bit long block including the padding:

We have
$y_i = f(x_i, h_0)$ and $g(y_i) = \sigma$ and
$y_j = f(x_j, h_0)$ and $g(y_j) = \sigma$ for some σ.
Note that the collision can occur

- in the first step ($y_i = y_j$) or
- in the second step ($y_i \neq y_j$), after final operation g

Procedure of the adversary corresponds to these two possibilities. She makes q_1 queries to oracle f and q_2 queries to oracle g, where $q_1 = q_2 = q/2$. For every $1 \le i \le q_1$, the adversary A chooses K bits long inputs x_i arbitrarily and obtains the list of records (x_i, h_0, y_i), where $y_i = f(x_i, h_0)$. If there are two equal values y_i and y_j in the third position, she has the collision for messages x_i and x_j. If not, the adversary continues by creating the second list. For every $1 \le i \le q_2$, she takes values $\mathrm{y_i}$ from the first list and from the oracle g she obtains values $Y_i = g(y_i)$. She creates the list $(y_i, Y_i)_{1 \le i \le q2}$. If there are two equal n bits long values Y_i and Y_j in the third position, she has the collision for messages x_i and x_j. If not, the adversary returns a negative response (collision not found). Let us denote p the probability that the adversary A will find a collision by this procedure, and P that she doesn't. From the definition of the random oracle f, it follows that $\{y_i\}_{1 \le i \le q1}$ is the set of q_1 random values from the set $\{0, 1\}^n$. From the definition of the random oracle g, it follows that $\{Y_i\}_{1 \le i \le q2}$ is the set of q_2 randomly chosen values from the set $\{0, 1\}^n$. The probability that the adversary will not find a collision in $q_1 + q_2$ oracle calls is

$$P = \prod_{i=1}^{q_1-1}\left(1-\frac{i}{2^n}\right) * \prod_{i=1}^{q_2-1}\left(1-\frac{i}{2^n}\right) \le^{(2)} \prod_{i=1}^{q_1-1}\left(e^{-i/2^n}\right) * \prod_{i=1}^{q_2-1}\left(e^{-i/2^n}\right) =$$

$$= e^{-q_1(q_1-1)/2^{n+1} - q_2(q_2-1)/2^{n+1}} = e^{(-q(q-2)/4)/2^n}.$$

We used here the fact (2) from Lemma 1 and that $q_1 = q_2 = q/2$. The probability that there will be a collision, is

$$p = 1 - P \ge 1 - e^{(-q(q-2)/4)/2^n} \ge^{(1)} (1-e^{-1}) * (q(q-2)/4)/2^n \ge 0.158 * q(q-2)/2^n.$$

We used here the fact (1) from Lemma 1, i.e. $1 - e^{-z} \ge (1-e^{-1})z$ for all $0 \le z \le 1$. In the role of z, there is the expression $z = (q(q-2)/4)/\ 2^n$. According to presumptions we have $q \le 2^{n/2}$, so that $z \le 1$ and the treatment was correct. The proof is finished.

Now, we will show that $\mathrm{Adv\_coll\_NMAC[n](q)} \le 0.5 * q(q-1)/2^n$.

It suffices to prove that $\mathrm{Adv\_coll\_NMAC[n](A)} \le 0.5 * q(q-1)/2^n$ for any adversary A and $1 < q \le 2^n + 1$ (for greater q the right side of the target inequality is greater then one). We can model the activity of the adversary A by his queries to oracles f and g. The oracle f creates a table $\mathrm{T_f}$ of records (x, h, y), where (x, h) is the input, taken by the adversary, and $y = f(x, h)$ is the response. The oracle g creates a table $\mathrm{T_g}$ of records (X, Y), where X is the input, taken by the adversary and $Y = g(X)$ is the response. Let us denote q_1 number of queries to oracle f and q_2 number of queries to oracle g. We have $q \le q_1 + q_2$. The attacker A is searching for colliding messages $M \neq M'$, which have several K bits long blocks including the padding. The messages can differ in a number of blocks. If there is a collision, there are two cases:

- **Internal collision.** In this case, the collision occurred before the final operation g.
- **Final collision.** In this case, the collision occurred after the final operation and it has occurred nowhere before the final operation g.

Internal collision

Let us elaborate the internal collision first. In this case, there has to be created a record (x_i, h_{i-1}, $h_i = f(x_i, h_{i-1})$) in the table T_f during processing of the first message and a record (x_j, h_{j-1}, $h_j = f(x_j, h_{j-1})$) during processing of the second message, where $i \neq j$ are indexes of oracle's records (they are not indexes of blocks of messages), $(x_i, h_{i-1}) \neq (x_j, h_{j-1})$ and $h_i = h_j$, where x_i is some K bits long block of the first message and x_j is some K bits long block of the second message. For fixed $i = 2, ..., q$, let us denote C_i the event that there is an index j, $1 \leq j < i$, such that j-th and i-th records in the table T_f have the same values in the third position, whereas $(x_i, h_{i-1}) \neq (x_j, h_{j-1})$. Denote $\Pr[C_i]$ probability of this event. Since f is a random oracle, it chooses the answer y_i to i-th query randomly from the set of 2^n values. The probability that it equals to one value from the set of values in the third position in the table T_f is less or equal to $(i - 1)/2^n$, i.e. $\Pr[C_i] \leq (i-1)/2^n$. Let P_{int} denote the probability that there is an internal collision. We have

$$P_{\text{int}} = \Pr[C_2] + \Pr[C_3] + ... + \Pr[C_{q_1}] \leq \sum_{i=2}^{q_1} \frac{i-1}{2^n} \leq q_1(q_1 - 1)/2^{n+1}.$$

Final collision

Now, we will assume that there is a final collision and no internal collision. The oracle g creates the table T_g with records $(X_i, g(X_i))$, $1 \leq i \leq q_2$. Denote P_{fin} the probability of the final collision. For fixed $i = 2, ..., q_2$, let us denote C_i the event that there is j, $1 \leq j < i$, such that j-th record and i-th record in the table T_g have the same values in the second position, i.e. $g(X_i) = g(X_j)$. Let $\Pr[C_i]$ denote the probability of such an event. Since g is a random oracle, the value $g(X_i)$ is chosen randomly from the set of all 2^n values, so that $\Pr[C_i] \leq (i-1)/2^n$. If there is a final collision, then there was an event C_i for some $2 \leq i \leq q_2$. Therefore, we have

$$P_{fin} \leq \Pr[C_2] + \Pr[C_3] + ... + \Pr[C_{q_2}] \leq \sum_{i=2}^{q_2} \frac{i-1}{2^n} \leq q_2(q_2 - 1)/2^{n+1}.$$

Total estimate

If the adversary finds a collision, it has to be either internal or final. From here, it follows

$$\text{Adv\_coll\_NMAC[n](A)} \leq P_{\text{int}} + P_{fin} \leq q_1(q_1 - 1)/2^{n+1} + q_2(q_2 - 1)/2^{n+1} \leq$$

$$\leq [(q_1 + q_2)^2 - (q_1 + q_2)]/2^{n+1} = (q^2 - q)/2^{n+1},$$

qed.

9. Appendix 2: Definition of the Special Block Cipher DN (Double Net)

Will be added soon, after the approval of its publications.

10. Appendix 3: Definition of the Hash Function HDN (Hash Double Net)

Will be added soon, after the approval of its publications.