Non-interactive Manual Channel Message Authentication Based on eTCR Hash Functions

Mohammad Reza Reyhanitabar, Shuhong Wang, and Reihaneh Safavi-Naini

Faculty of Informatics,
University of Wollongong, Australia
{ mrr790, shuhong, rei}@uow.edu.au

Abstract. We present a new non-interactive message authentication protocol in manual channel model (NIMAP, for short) using the weakest assumption on the manual channel (i.e. assuming the strongest adversary). Our protocol uses enhanced target collision resistant (eTCR) hash family and is provably secure in the standard model. We compare our protocol with protocols with similar properties and show that the new NIMAP has the same security level as the best previously known NIMAP whilst it is more practical. In particular, to authenticate a message such as a 1024-bit public key, we require an eTCR hash family that can be constructed from any off-the-shelf Merkle-Damgård hash function using randomized hashing mode. The underlying compression function must be *evaluated second preimage resistant* (eSPR), which is a strictly weaker security property than collision resistance. We also revisit some closely related security notions for hash functions and study their relationships to help understanding our protocol.

Key words: Message authentication, manual channel, eTCR hash family, randomized hashing, hash function security.

1 Introduction

Message authentication protocols provide assurance that a received message is genuine and sent by the claimed sender. Authentication protocols have been studied in asymmetric (assuming PKI) and symmetric (assuming shared secret keys) settings. *Manual channel* (or two-channel) authentication model is a recently proposed model, motivated by security requirements of ad hoc networking applications. In this model a user wants to send an authenticated message to a receiver. There is neither a shared secret key between communicants nor there is a public key infrastructure. However the sender, in addition to an insecure broadband channel (e.g. a wireless channel) that is used to send the message, has access to a second narrow-band channel, referred to as *manual channel*

that is authenticated in the sense that messages over this channel cannot be modified, although they can be delayed, replayed or removed. The channel is low capacity and can only transfer up to a few hundred bits. A manual channel models human assisted channels such as face-to-face communication, telephone conversation between two parties, or communication between two devices facilitated by a human: a person reads a short number on a device display and inputs it into a second device using a keyboard. The *short authentication string* sent over the manual channel is called SAS [22]. A number of interactive and non-interactive protocols have been proposed in this model and their security has been proven in computational and unconditional security frameworks [8, 7, 1, 17, 12, 15]. In this paper we consider computationally secure non-interactive message authentication protocols (NIMAPs) in manual channel model and assume a *weak manual channel* as defined by Vaudenay [22] (see Sect. 2) which corresponds to the strongest adversary. We note that in NIMAP the scarce resource is the bandwidth of the manual channel.

Computationally secure NIMAPs. Several NIMAPs have been proposed in literatures [1, 7, 17, 12]. We briefly review them below and move their details to Appendix A for compact and completeness.

Balfanz, Smetters, Stewart, and Wong [1] (referred to as BSSW protocol) were the first to propose a manual channel NIMAP that was based on collision resistant hash functions. The basic idea is to send the massage m over the insecure channel, and send its hash value, computed using collision resistant hash function, over the manual channel. Vaudenay [22] proposed a formal security model for manual authentication protocols and gave a security reduction from the security of the protocol to collision resistance property of the hash function. He showed that to guarantee security against an adversary having time $T = 2^n$, the SAS length must be at least $2n$ bits.

Gehrmann, Mitchell, and Nyberg [7] proposed a number of protocols, MANA I, II and III, of which only MANA I is a NIMAP. MANA I requires low bandwidth for manual channel. For example to make the probability of a successful attack less than about 2^{-17}, one should use a SAS of length about 40 bits. The protocol requires manual channel to also provide confidentiality and Vaudenay in [22] pointed out that the manual channel must be at least stall-free. We will not include MANA I in our comparisons because of these extra requirements on manual channel.

Pasini-Vaudenay [17] presented a NIMAP (referred to as PV protocol) that requires, a hash function that is second preimage resistant, and a trapdoor commitment scheme in Common Reference String (CRS) model.

Although compared with BSSW that uses collision resistant hash functions, PV protocol has weaker security requirements on hash functions (i.e. second preimage resistance), but it needs a secure trapdoor commitment scheme in CRS model which makes it a more demanding protocol.

Mashatan and Stinson [12] proposed a new property, *Hybrid Collision Resistance(HCR)* for hash functions and proposed a NIMAP (referred to as MS protocol) that is provably secure assuming the hash function is HCR. Mashatan et al use random oracle model to show that HCR is a weaker security property than CR for hash functions and so the protocol is of interest because it achieves the same level of security and efficiency as PV protocol without requiring a complex commitment scheme and the added assumption of CRS. In Section 3 we show that there is no clear method of instantiating the hash function used in this protocol to be used for arbitrary length messages. In particular, we point out that popular Merkle-Damgård construction cannot be used for domain extension of HCR functions. This leaves construction of efficient NIMAPs for arbitrary length messages in weak manual authentication model, an open problem.

Our contributions. We propose a new NIMAP in weak manual channel model that uses a hash function family and is provably secure in standard model. The protocol is based on an *enhanced target collision resistant (eTCR)* hash function family and can be constructed using randomized hashing mode of a Merkle-Damgård hash function (Theorem 4 of [9]).

To evaluate our protocol we consider underlying security assumptions of existing NIMAP protocols that use weak manual channel model. This includes BSSW, PV and MS protocols. In all these cases, and also in the case of our protocol, the security relies on (in BSSW and our protocol reduces to) the required property of the hash function. We give a careful comparison of these properties (collision resistance, second-preimage-resistance, HCR and eTCR) from two view points. Firstly, in terms of implication or separation, i.e showing whether one property implies the other one, or there is a clear separation between them, and secondly, if the property can be guaranteed for arbitrary length messages. This latter requirement removes restriction on the message length sent over the manual channel. Our comparison also includes *evaluated second preimage resistance (eSPR)* property, a property of compression functions introduced to construct eTCR hash function families through Merkle-Damgård construction in the randomized hashing mode [9]. We show that eSPR notion is not strictly stronger than HCR notion, using previously known results [9] that eSPR is not strictly stronger than SPR notion.

The comparison is of of interest because of its direct application to NIMAP and also for grading properties of hash functions.

Paper organization. In Section 2 we describe communication and security model for manual channel authentication. In Section 3 we give an overview of security notions for hash functions and describe the three security notions, eSPR, eTCR and HCR, that are directly related to our NIMAP and MS protocol. In Sect. 4 we present a new protocol and analyze its security. We also compare it with previous protocols and show its potential advantages. The paper is concluded in Sect. 5.

2 Communication and security model

Communication model. We consider the problem of noninteractive authentication between a sender Alice and a verifier Bob: Alice wants to send a message, M, to Bob such that Bob can be assured that the message has come from Alice (entity authentication) and has not been modified by an adversary Eve (message authentication). It is assumed that Alice and Bob have access to two communication channels; a broadband insecure channel (denoted by $\longrightarrow$) and an authenticated narrow-band channel (denoted by $\Longrightarrow$). It is further assumed that the authenticated narrow-band channel is linked to the identity of the sender, i.e. Alice. In other words when Bob receives a message from this channel he is ensured that it is generated by Alice although the message can be a replay of a previous one. The most important restriction on the narrow-band channel is the limitation on the bandwidth: the channel can transmit messages of length at most n which in some applications n can be as small as 32 bits.

As a real world example of this scenario consider user-aided pairing of two wireless devices (e.g. Bluetooth) such as a mobile phone and a laptop. The user can read a message consisting of a number of characters on the screen of mobile phone and type them on laptop keyboard. In this case the user establishes the authenticated channel manually. These kinds of human controlled authenticated channel are also called *manual channels*.

Security model. We assume *weak authenticated channel model* and the *strong adversary* described in Vaudney [22]. The adversary Eve has full control over the broadband channel, i.e. she can read, modify, delay, drop messages, or insert new ones. In the weak manual channel model, it is assumed that Eve can read, delay, replay and drop messages sent over manual channel, but she cannot modify or insert messages into this channel. In other words there is no extra security assumptions, like confidentiality or stall-freeness, on a weak manual channel. A manual channel

with some additional security requirements on it is called a strong manual channel. It is also assumed that the adversary can employ adaptive chosen message attack: she can adaptively choose the input message to be sent by Alice and make Alice to produce messages of the protocol to be sent over the two channels. The number of such queries made by Eve is her *online complexity* and is denoted by Q. A second resource of Eve is her *offline complexity*, denoted by T, denoting the time spent on processing the messages in the attack. We assume that Eve has bounded computational resources.

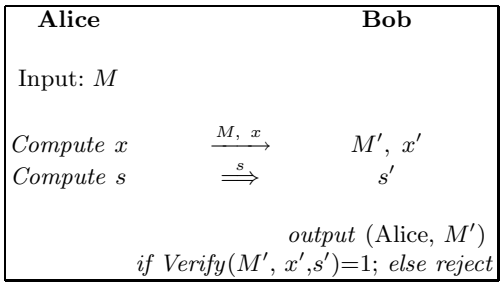

Fig. 1. A typical manual channel NIMAP

A typical manual channel NIMAP works as follows (see Figure 1). On input message M Alice uses (possibly randomized) algorithms to compute a tag x and a short authentication string (SAS) s. The message M together with the tag x are sent over insecure broadband channel and SAS is sent over the authenticated channel. Note that x may be a null string in which case no tag will be sent over the insecure channel. Figure 1 shows communication flows in such a protocol. We note that in PV protocol the message might not be explicitly sent over the insecure channel. However the message in their protocol can be transformed (i.e. re-coded) into our representation. The transformation is public and so will not affect security of the protocol. Received messages by Bob are denoted by M', x' and s' to show possible effects of an adversary. The verification process (accept or reject a received message) by Bob is abstractly denoted by a (publicly known) deterministic binary function *Verify*(.). The function outputs 1 if the acceptance conditions (specified for the protocol) are satisfied by the received message, and 0 otherwise.

Definition 1 (Adversary). *An adversary Eve is said to be a* (T, Q, ϵ)-breaking adversary*, if she has query resource* Q *(number of queries made*

to Alice), time resource T (time complexity) and is successful with probability at least ϵ, in making Bob output (Alice, M') while M' has never been an input of the protocol on Alice side, i.e. it has never been authenticated by Alice.

A protocol is said to be (T, Q, ϵ)-secure *if there exists no* (T, Q, ϵ)-*breaking adversary against it.*

Note that to be considered a successful adversary, Eve should respect the communication and security model described above. For example she can only replay a previously obtained s from Alice but she cannot modify it or inject a new one. More specifically if Eve has made Q queries from Alice and has collected a data set $\{(M_i, x_i, s_i); 1 \leq i \leq Q\}$, then a successful attacker Eve should find an $M' \notin \{M_i; 1 \leq i \leq Q\}$, any x' and an $s' \in \{s_i; 1 \leq i \leq Q\}$ such that *Verify*(M', x', s')=1.

Proving security of a manual channel NIMAP consists of two steps. Firstly one should show that the protocol is $(T', 1, \epsilon')$-secure, i.e. secure against adversaries that can only make one query from Alice (called one-shot adversaries in [22]) and have time complexity T'. This is done by transforming such an adversary against the protocol into an adversary that can defeat security assumptions on the underlying building primitive(s) of protocol. The second step of proof (i.e., showing that protocol is (T, Q, ϵ)-secure) can be done (Lemma 6 in [22]) by transforming a (T, Q, ϵ)-breaking adversary to a $(T', 1, \epsilon')$-breaking adversary, where $\epsilon' = \frac{\epsilon}{Q}$.

3 Hash functions and security notions

Cryptographic hash functions play an important role in design of NIMAPs as well as many other cryptographic protocols like MACs and digital signature schemes. There are numerous informal and formal definitions of security for hash functions. Definitions can be application specific. For example Brown [4] defined *Zero-Finder-Resistance* as the difficulty of finding a preimage for zero (i.e. finding a domain element that is hashed to 0) and showed it to be a necessary security assumption for the hash functions to prove security of DSA algorithm.

The most widely used security notions for hash functions are *Collision resistance(CR)*, *Second-preimage resistance(SPR)* and *Preimage resistance(PR)* and are required in applications such as digital signature, commitment and password protection. Informal definitions of these notions for *a fixed hash function* and formal definitions of CR notion and one of its weaker variants, UOWHF (Universal One Way Hash Function)

for *a family of hash functions*, can be found in [5, 6, 13, 14, 16]. UOWHF notion (originally defined in asymptotic security framework in [14]) is also called *TCR (Target Collision Resistance)* (rephrased in concrete security framework in [3]).

Informally, for a fixed hash function H, CR means that it is computationally hard to find two distinct inputs $M' \neq M$ that collide under hash function, i.e. $H(M) = H(M')$. SPR means that for a given input M, it is computationally hard to find M' such that $M' \neq M$ and $H(M) = H(M')$. PR refers to one-wayness property and means that it is computationally hard to find a preimage (domain element x) for a given hash value (range element y), so that these constitute a valid (input, output) pair for the hash function (i.e. $H(x) = y$).

Regarding CR notion, there is a foundational problem, that is formal definition of CR security notion can only be given for a family of hash functions (also called keyed hash function) and not for a fixed hash function. There are also some other subtleties regarding formal definitions of security notions for hash functions and studying relationships (implications and separations) between different security notions. A brief summary is provided in Appendix B. More details on CR definition dilemma and also a comprehensive formal treatment of security notions (including implications and separations between CR, SPR, PR and TCR notions), can be found in [18, 21, 19].

In comparing two security notions for hash functions, we say that *notion A is stronger than notion B* if A implies B; that is if a hash function H satisfies notion A then it also satisfies notion B. For instance, CR is a stronger security notion than SPR and the implication is shown in [18] and [21] for keyed and unkeyed settings, respectively.

3.1 Definitions for eSPR, eTCR and HCR notions

We review in more details three security notions relevant to the discussion in the next section. First we recall Merkle-Damgård construction that provides a method of extending domain for hash functions.

Merkle-Damgård construction. For a compression function $H : \{0,1\}^{n+b} \rightarrow \{0,1\}^n$, an L-round Merkle-Damgård construction is a method of constructing a hash function $MD_L[H] : \{0,1\}^{n+L.b} \rightarrow \{0,1\}^n$ with an extended domain. For an initial value $C_0 \in \{0,1\}^n$ and a message $M = M_1||M_2||\ldots||M_L$ consisting of L blocks each of size b bits, it outputs an n-bit hash value denoted by C_L as shown in Figure 2:

– The input message M is divided into L blocks $M_1, ..., M_L$, each block M_i of length b bits.
– The chaining variable C is initialized to C_0.
– For i=1 ... L :
$C_i = H(C_{i-1}, M_i)$
– C_L is output as the hash value.

If the input message length is not a multiple of the block length b, proper padding can be used. For a fixed initial value C_0 we denote the transformation by $MD_L^{C_0}[H] : \{0,1\}^{Lb} \rightarrow \{0,1\}^n$.

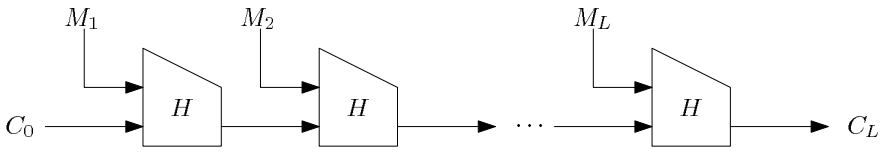

Fig. 2. L-round Merkle-Damgård construction

By *strengthened Merkle-Damgård* we mean Merkle-Damgård with a proper length indicating padding and some fixed initial value. Strengthened Merkle-Damgård's construction converts a compression function to a hash function for arbitrary length input while preserving CR property of the compression function.

In the sequel, we use $\stackrel{\$}{\leftarrow}$ and $\stackrel{R}{\leftarrow}$, to denote randomly selecting (computing) according to a specific distribution (output distribution of a probabilistic algorithm) and uniform distribution, respectively.

For the definition of HCR we follow [12] but parameterize the game explicitly with the length of the randomness (l_2). (As noted in [12], l_2 and n are security related parameters.) We use a state variable *State* to show the state information that the adversary A keeps between its attack phases.

Definition 2 (HCR notion). *A compression function* $H : \{0,1\}^{l_1+l_2} \rightarrow \{0,1\}^n$ *is* $(T, \epsilon) - HCR^{[l_2]}$ *if no adversary A, having time at most T, can win the following game with probability at least* ϵ*:*

***Game**(HCR*$^{[l_2]}$*, A)*
$(M, State) \xleftarrow{\$} A()$ $//M \in \{0,1\}^{l_1}$
$K \xleftarrow{R} \{0,1\}^{l_2}$
$M' \xleftarrow{\$} A(K, State)$ $//M' \in \{0,1\}^{l_1+l_2}$

A wins the game if $M' \neq M||K$ *and* $H(M') = H(M||K)$

Note that $HCR^{[l_2]}$ notion for an arbitrary-input-length hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$ can be defined by a game in which the adversary can output $M \in \{0,1\}^*$ and $M' \in \{0,1\}^*$, in the above game.

eSPR notion is defined for a compression function[9], which is a variant of Second-Preimage Resistance. The notion of eSPR is motived by searching for properties of a compression function that suffice to ensure that the multi-block randomized extension obtained via Merkle-Damgård iteration is TCR or eTCR. This will be beneficial because Merkle-Damgård iteration converts a compression function for fixed-input-length to an arbitrary-input-length hash function (family).

Definition 3 (eSPR notion). *A compression function* $H : \{0,1\}^{n+b} \rightarrow \{0,1\}^n$ *is* (T, L, ϵ)*- eSPR if no adversary, spending time at most* T *and using messages of length* L*(in b-bit blocks), can win the following game with probability at least* ϵ*. It is assumed that the adversary knows the initial value* C_0 *before starting the game, i.e. either* C_0 *is chosen at random and given to the adversary (uniform setting) or it is a parameter of the game that the adversary will receive as an 'advice' (non-uniform setting).*

***Game**(eSPR, A)*
$\Delta_1, \ldots, \Delta_L \xleftarrow{\$} A()$ $//\Delta_i \in \{0,1\}^b, \quad L \geq 2$
$r \xleftarrow{R} \{0,1\}^b$
$M = \Delta_L \oplus r; \quad C = MD_{L-1}^{C_0}[H](\Delta_1 \oplus r, ..., \Delta_{L-1} \oplus r)$
$(C', M') \xleftarrow{\$} A(C, M)$ $//C' \in \{0,1\}^n, \quad M' \in \{0,1\}^b$

A wins the game if $C'||M' \neq C||M$ *and* $H(C'||M') = H(C||M)$

eTCR security notion is defined in [9] for arbitrary-input-length hash function *families*. Note that HCR and eSPR security notions were defined for a single hash function or a fixed compression function.

Definition 4 (eTCR notion). *An arbitrary-input-length hash function* ***family****,* $\mathcal{H} : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^n$*, is* (T, ϵ)*- eTCR*$^{[m]}$*, if no adversary spending time at most* T *can win the following game with probability at*

least ϵ. We use a state variable State to keep adversary state between its attack phases:

$\boldsymbol{Game}(eTCR^{[m]})$
$(M, State) \xleftarrow{\$} A()$ $\quad //M \in \{0,1\}^m$
$K \xleftarrow{R} \{0,1\}^k$
$(K', M') \xleftarrow{\$} A(K, State)$ $\quad //K' \in \{0,1\}^k$ *and* $M' \in \{0,1\}^*$

A wins the game if $(K, M) \neq (K', M')$ *and* $H_K(M) = H_{K'}(M')$

As mentioned previously, a method of constructing an eTCR hash function family is using an iterated hash method (e.g. Merkle-Damgård construction) with a compression function. Halevi et al's iterated construction [9] reduces eTCR notion to eSPR property for the compression function (Theorem 1). In [9], the length(in blocks) of the target message M, is denoted by L ($L = m/b$, where b denotes block length in bits) and is considered as another resource parameter of the adversary. So, alternatively the adversary can be denoted as a (T, L, ϵ) adversary and the notion can be defined as (T, L, ϵ)-eTCR, instead of specifying parameter m as a superscript.

3.2 Relations among eSPR, eTCR and HCR notions

In this section we study relationships between the three notions, eSPR, eTCR and HCR.

eSPR versus HCR. We show that *eSPR notion is not stronger than HCR notion*. That is there exist compression functions that are eSPR but not HCR.

This can be shown by considering the following two relations.

- R1. Halevi et al [9] pointed out a separation between eSPR and SPR and argued that (depending on the structure of the compression function) *there exist compression functions that are eSPR but not SPR.*
- R2. We show *if a compression function is not SPR then it is not HCR either* (i.e., HCR is stronger notion than SPR). This can be seen by noting that an adversary A against SPR property can be used to construct an adversary B against HCR property. To win in HCR game, B forwards $M||K$ to A and outputs A's response (which is a second preimage of $H(M||K)$) as M' in HCR game. Clearly B succeeds whenever A succeeds.

Now if eSPR is stronger than HCR, then combined with R2 we can conclude that eSPR is stronger that SPR. This contradicts R1 and so *eSPR is not a stronger notion than HCR* .

Relation between HCR and eTCR. We show (constructively) that *existence of a* (T, ϵ)*-HCR* $^{[l_2]}$ *compression function implies existence of a* (T, ϵ)*- eTCR compression function family.*
Assume that we have a (T, ϵ)-HCR$^{[l_2]}$ compression function $H : \{0,1\}^{l_1+l_2} \rightarrow \{0,1\}^n$. We construct a compression function family as follows:
$\mathcal{H} = \{H_K\}_{K \in \{0,1\}^{l_2}}$, where $H_K : \{0,1\}^{l_1} \rightarrow \{0,1\}^n$ and $H_K(M) = H(M||K)$. To show that the constructed family $\mathcal{H}$ is (T, ϵ)- eTCR, we note that an adversary A against eTCR property of the family $\mathcal{H}$ can be transformed into an adversary B against HCR property of H with the same advantage. Adversary B plays HCR game against H while accessing A. In the first move, B runs A to choose a message M. After receiving K, B forwards it to A who will generate (K', M') such that $H_K(M) = H_{K'}(M')$. Upon receiving (K', M') form A, adversary B outputs $M'||K'$ in final move of its HCR game. Clearly B wins HCR game against H whenever A wins eTCR game against $\mathcal{H}$.

Using Merkle-Damgård construction for HCR. Let $MD_L[H]$ denote a L-round strengthened Merkle-Damgård construction. We show that a collision finding adversary A against $MD_L[H]$ can be used to construct an algorithm B that defeats $MD_{L+1}[H]$ in HCR$^{[l_2]}$ sense. We assume in HCR game $|K| = l_2 > 0$ (for $l_2 = 0$, HCR is the same as CR). B works as follows:

Algorithm B invokes A to obtain two colliding messages M and M' each of length L blocks. (Note that a successful adversary against strengthened Merkle-Damgård construction results in such a collision). In the first move of HCR game against $MD_{L+1}[H]$, algorithm B commits to M and when receives a random challenge $K \in \{0,1\}^{l_2}$, it outputs $M'||K$ as colliding pair with $M||K$. Clearly B succeeds whenever A succeeds.

In MS protocol, if the sum of the lengths of the message to be sent (i.e. l_1) and the security parameter l_2 (e.g. $l_2 = 70$ as in [12]) becomes more than one block, the hash function should be applied to a message with length more than one block and it should provide HCR property. In above, we showed that without CR assumption on one-round Merkle-Damgård version (i.e. compression function using specified initial value

C_0 as part of input), the hash function cannot provide HCR property as needed in MS in such a case.

Reduction from eSPR to eTCR. The following theorem reproduced from [9] gives an explicit construction for eTCR hash function family.

Theorem 1. *[9] Assume that* $h : \{0,1\}^{n+b} \rightarrow \{0,1\}^n$ *is a* $(T, L+1, \epsilon)$*-eSPR compression function that is also* (T', ϵ')*-OWH. The* $(L+1)$*-round Merkle-Damgård construction based on* h *as compression function and used in randomized hashing mode, defines a family of hash functions* $\widetilde{H_r} : \{0,1\}^b \times \{0,1\}^{Lb} \rightarrow \{0,1\}^n$ *that is* $(T - O(L), L, \epsilon' + (L+1)\epsilon)$*- eTCR secure. This family is constructed as* $\widetilde{H_r}(M) = \widetilde{H}(r, M) = MD^{C_0}_{L+1}[h](r, M_1 \oplus r \ldots M_L \oplus r)$*, where* $M = M_1||...||M_L$ *and* C_0 *is a known initial value.*

As argued in [9], the second property in addition to eSPR , i.e., (T', ϵ')-OWH, is implied by eSPR assuming a mild structural property for the compression function and is redundant. We refer the reader to [9] for more discussion on this matter.

4 A NIMAP based on eTCR hash families

4.1 Protocol description and security reduction

Assume that we have a (T, ϵ)- eTCR hash function family $\mathcal{H} : \{0,1\}^k \times \{0,1\}^{<m} \rightarrow \{0,1\}^n$, where m is the maximum size of input length(e.g., $m = 2^{64}$). We construct a secure NIMAP between a claimant, Alice, and a verifier, Bob, in weak manual channel model. The NIMAP is as follows:

1. On input message M, Alice chooses uniformly at random a key $x \in \{0,1\}^k$ and computes $s = H_x(M)$;
2. Alice sends (M, x) to Bob over the insecure channel and sends $s = H_x(M)$ over the authenticated channel;
3. Bob receives (M', x') via insecure channel and s' via authenticated channel;
4. Bob outputs (Alice, M') if $s' = H_{x'}(M')$ and rejects M' otherwise.

The proposed protocol is illustrated in Figure 3.

The following Theorem guarantees security of the NIMAP.

Theorem 2. *Let* $\mathcal{H} : \{0,1\}^k \times \{0,1\}^{<m} \rightarrow \{0,1\}^n$ *be a* (T_H, ϵ_H)*- eTCR hash function family. The proposed NIMAP as in Figure 3 is a* (T, Q, ϵ)*-secure NIMAP, where* $T = T_H - \mu Q - \sigma$*,* $\epsilon = Q\epsilon_H$*. Constants* μ *and* σ *represent the maximum time complexity of Alice over all* Q *queries and the time required for a single hash computation, respectively.*

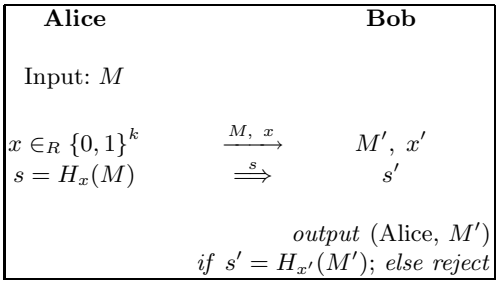

Fig. 3. A new manual channel NIMAP based on eTCR hash family

Proof. First we show that any $(T', 1, \epsilon')$-breaking adversary $\widehat{A}$ against our NIMAP can be used to construct a $(T' + \sigma, \epsilon')$-breaking adversary B against eTCR hash family $\mathcal{H}$. Then we complete the proof by a general reduction from any (T, Q, ϵ)-breaking adversary A to a $(T', 1, \epsilon')$-breaking adversary $\widehat{A}$, where $T' = T + \mu Q$ and $\epsilon' \geq \frac{\epsilon}{Q}$.

To prove the first part, let $\widehat{A}$ be a $(T', 1, \epsilon')$-breaking adversary against the NIMAP. That is, the adversary makes a single query from Alice to obtain (M, x, s) and then spends time at most T' to mount a successful attack, i.e. produces (M', x') where $M' \neq M$ and $H_{x'}(M') = s$. Note that it is possible to have $x' = x$. Adversary B against H plays eTCR game using $\widehat{A}$ as follows. It runs $\widehat{A}$ and obtains the query M and commits to it in the first move of eTCR game. After receiving the hash function key, i.e. $x \in \{0,1\}^k$, B computes $s = H_x(M)$ in time σ, and forwards x and s to $\widehat{A}$. Adversary $\widehat{A}$ within time T' produce (M', x'). Adversary B outputs M' as the second message and x' as the second hash function key in eTCR game. This means that B succeeds in time $T' + \sigma$ and with the same success probability ϵ' as $\widehat{A}$.

The second part of the proof is a general transformation between a Q-query adversary and 1-query adversary [22]. For completeness of the proof, we have included the proof (i.e. two-party NIMAP). Let A be a (T, Q, ϵ)-breaking adversary against the NIMAP. We can construct a $(T', 1, \epsilon')$-breaking adversary $\widehat{A}$ as follows.
Adversary $\widehat{A}$ chooses uniformly at random $j \in \{1, 2, \ldots, Q\}$ and runs A. When A makes its $i-th$ query M^i, adversary $\widehat{A}$ selects at random an $x^i \in_R \{0,1\}^k$, computes $s^i = H_{x^i}(M^i)$ and provide A with x^i and s^i. This is done for every $i-th$ query except when $i = j$ in which case $\widehat{A}$ forwards the query ($j-th$ query of A) to Alice (in real protocol) and uses Alice's response to respond A. When A succeeds, it outputs (M', x', s') where $s' = H_{x'}(M')$, M', is different from all previously queried messages and

s' is a replay of one of the previously obtained authenticated messages. With probability $\frac{1}{Q}$ we have $s' = s^j$ and so $\widehat{A}$ succeeds with probability $\epsilon' \geq \frac{\epsilon}{Q}$. Denote by μ the maximum overall time to run the protocol once, i.e., to compute x and s on an input M, where the maximum is over Q queries made by A. It is easy to see that time complexity of algorithm $\widehat{A}$ is $T' = T + \mu Q$. This completes the proof of the theorem.

□

4.2 Comparison with previous schemes

We compare our proposed NIMAP with the existing NIMAP protocols using weak manual channel, namely BSSW [1], PV [17] and MS [12]. The comparison is made for the same level of security, from following viewpoints:

1. Security assumptions required for the underlying primitives (commitment schemes and/or hash functions)
2. Required bandwidth for the manual channel (i.e., the SAS length).

Security assumptions. We consider security assumptions required by BSSW, PV, MS and our protocol when there is no restriction on the length of the input message.

The BSSW protocol uses a fixed (unkeyed) hash function and requires it to be collision resistant (CR). CR is a strong security assumption for a hash function which cannot be formally defined for a single hash function [2, 18]. To obtain the property for arbitrary length messages Merkle-Damgård construction can be used [19].

The PV protocol uses SPR which is a weaker assumption than CR ([21, 18]). PV protocol also requires a secure trapdoor commitment scheme in CRS model. Furthermore, the commitment string c is taken as an input to the hash function ([17]) and so the hash domain needs to be of arbitrary size (if one uses an arbitrary commitment scheme); i.e., one needs an arbitrary-input-length hash function that provides security in SPR sense.
To compute SAS length, PV assumes that hash function provides ideal security in SPR sense, i.e., a hash function with security level of 2^{-n}, where n is the hash size. This assumption for the case of long messages is not satisfied by iterated Merkle-Damgård hash functions (like MD5, SHA1, RIPEMD-160, Whirlpool) as shown by recent analysis in [11].

MS protocol also uses a fixed hash function satisfying HCR property. The $\text{HCR}^{[l]}$ is a notion between CR and SPR, depending on the value

of l. As shown in subsection 3.2, the commonly used Merkle-Damgård domain extension construction does not guarantee HCR (without CR assumption) and so it is not clear how to construct an arbitrary-input-length HCR hash functions from a fixed-input-length one.

Our NIMAP uses an eTCR hash *family* to hash arbitrary-length messages. Standard Merkle-Damgård iteration in randomized hashing mode can be used to construct such an eTCR hash family from an eSPR compression function (i.e. a fixed-input-length hash function) [9]. Hence *security of our protocol is reduced to eSPR property for a fixed-input-length hash function*. It has been argued [9] that eSPR notion is weaker than CR and also is not stronger than SPR. We also argued in subsection 3.2 that eSPR is not stronger than HCR. The above argument shows that our protocol, when used for arbitrary length messages, requires less demanding security assumption (namely, eSPR-ness of a fixed-input-length hash function) and benefits from provable security framework in constructing eTCR hash family for arbitrary length messages (Theorem 1).

Manual channel bandwidth. Assume an adversary with the same resources and required security level (denoted by ϵ) as in [12]. Namely, we require the NIMAP to be (T, Q, ϵ)-*secure* , where $T \leq 2^{70}$, $Q \leq 2^{10}$ and $\epsilon = 2^{-20}$.

In BSSW the SAS length must be at least 140 bits. In PV protocol a SAS of length 100 is required (, but as mentioned above for arbitrary long messages PV requires that the used hash function provides ideal SPR security for long messages which is not satisfied by Merkle-Damgård constructions due to recent attacks in [11]). MS can theoretically reach the same level of security using a SAS of 100 bits for $l_2 = 70$ bits (, but we are not aware of a practical hash function that provides HCR for arbitrary-length messages without need to a stronger than HCR assumption on the underlying compression function and as we showed Merkle-Damgård constructions cannot be used for this purpose).

Our NIMAP needs a SAS with length $n = 100 + \log_2(L + 2)$ bits, where L denotes the message length in blocks. (See more details and computation of SAS length below.) For a 1024-bit message using SHA1 in randomized hashing mode ($L = 2$), the required SAS length will be 102 bits. Our NIMAP can still use randomized hashing mode for messages up to about 2^{49} bits using a SAS of only 140 bits.

To calculate SAS length for our protocol to have a NIMAP that is (T, Q, ϵ)-secure (for $T = 2^{70}$, $Q = 2^{10}$, $\epsilon = 2^{-20}$), using Theorem 2, we need a hash function family that is 2^{-30} ($=2^{-20}/2^{10}$) secure in eTCR sense. Using Theorem 1, we can construct such an eTCR family assuming

that the compression function is eSPR with $\epsilon = \frac{2^{-30}}{L+2}$ and L being the number of blocks in the input message of the eTCR function .(We assumed that $\epsilon' = \epsilon$ in Theorem 1). The length of SAS (i.e. required n) must be computed for each message length taking into account non-tightness of the reduction between eTCR and eSPR notions. One can use compression function of a standard hash function like SHA1 and truncate its output to n bits. Assuming that the *compression function* provides 2^{-n} security level *in eSPR sense*[1] , i.e. $\epsilon = T2^{-n}$, the SAS length of our NIMAP, i.e. n, for messages of length L blocks, is $n = 100 + log_2(L + 2)$ bits.

5 Conclusion

We proposed a new practical non-interactive message authentication protocol in manual channel model using a family of eTCR secure hash functions. For applications such as sending a public key where message length is small (e.g. 1024 bits), using randomized hashing mode one can construct an eTCR hash family using an off-the-shelf Merkle-Damgård hash function (e.g. SHA1). In this case security of the scheme will be based on eSPR property of the compression function which is strictly weaker than collision resistance property. For longer messages however, randomized hashing may not produce optimal result (shortest SAS) because of the non-tightness of reduction. Using randomized hashing for messages of up to 2^{49} bits results in SAS of around 140 bits. Other constructions of eTCR with tighter reduction can be directly used in the proposed NIMAP and could result in shorter SAS.

References

1. Dirk Balfanz, Diana K. Smetters, Paul Stewart, and H. Chi Wong. Talking to Strangers: Authentication in ad-hoc Wireless Networks. In *Network and Distributed Sytem Security Symposium*, San Diego, California, U.S.A., February 2002.
2. Mihir Bellare and Phillip Rogaway. Introduction to Modern Cryptography, (Page 3 of) Chapter 5: Hash Functions. Available at Bellare's homepage via : http://www-cse.ucsd.edu/users/mihir/cse207/index.html
3. M. Bellare, P. Rogaway. Collision-Resistant Hashing: Towards Making UOWHFs Practical. In *Advances in Cryptology–CRYPTO '97* (1997), Vol. 1294 of *LNCS*, Springer-Verlag, 470-484.
4. D. Brown. Generic Groups, Collision Resistance and ECDSA. Journal of *Designs, Codes and Cryptography*, Vol. 35 (2005), 119–152.

[1] Note that this assumption is not the same as in PV, since here we require such a property from a compression function in eSPR sense (i.e., only for single-block inputs) and not for arbitrary-length messages as in PV in SPR sense

5. I.B. Damgård. Collision Free Hash Functions and Public Key Signature Schemes. In *Advances in Cryptology–EUROCRYPT '87* (1988), Vol. 304 of *LNCS*, Springer-Verlag, 203–216.
6. I.B. Damgård. A Design Principle for Hash Functions. In *Advances in Cryptology–Crypto '89* (1990), Vol. 435 of *LNCS*, Springer-Verlag, 416-427.
7. Christian Gehrmann, Chris J. Mitchell, and Kaisa Nyberg. Manual Authentication for Wireless Devices. *RSA Cryptobytes*, 7(1): 29–37, January 2004.
8. C. Gehrmann and K. Nyberg. Security in Personal Area Networks. *Security for Mobility, IEE*, London, pages 191–230, 2004.
9. Shai Halevi and Hugo Krawczyk. Strengthening Digital Signatures Via Randomized Hashing. In *Advances in Cryptology–CRYPTO '06* (2006), Vol. 4117 of *LNCS*, Springer-Verlag, 41–59.
10. D. Hong, B. Preneel and S. Lee. Higher Order Universal One-Way Hash Functions. In *Advances in Cryptology–ASIACRYPT '04* (2004), Vol. 3329 of *LNCS*, Springer-Verlag, 201–213.
11. John Kelsey and Bruce Schneier. Second Preimages on n-Bit Hash Functions for Much Less than 2^n Work, In *Advances in Cryptology–EUROCRYPT '05* (2005), Vol. 3494 of *LNCS*, Springer-Verlag, 474–490 .
12. Atefeh Mashatan and Douglas R. Stinson. Noninteractive Two-Channel Message Authentication Based on Hybrid-Collision Resistant Hash Functions. *Cryptology ePrint Archive*, Report 2006/302.
13. R. Merkle. One Way Hash Functions and DES. In *Advances in Cryptology–CRYPTO '89* (1990), Vol. 435 of *LNCS*, Springer-Verlag, 428-446.
14. M. Naor and M. Yung. Universal One-Way Hash Functions and Their Cryptographic Applications. In Proc. of *21st ACM Symposium on the Theory of Computing*, 1990, 387–394.
15. M. Naor, G. Segev, and A. Smith. Tight Bounds for Unconditional Authentication Protocols in the Manual Channel and Shared Key Models. In *Advances in Cryptology–CRYPTO '06* (2006), Vol. 4117 of *LNCS*, Springer-Verlag, 214–231.
16. B. Preneel. Analysis and Design of Cryptographic Hash Functions. Doctoral dissertation, K. U. Leuven, 1993.
17. Sylvain Pasini and Serge Vaudenay. An Optimal Non-interactive Message Authentication Protocol. In David Pointcheval, editor, *Topics in Cryptography*, Vol. 3860 of LNCS, pages 280–294, San Jose, California, U.S.A., February 2006. Springer-Verlag.
18. Phillip Rogaway, Thomas Shrimpton. Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. In *Fast Software Encryption*, 11th International Workshop(FSE 2004), Vol. 3017 of *LNCS*, Springer-Verlag 371–388.
19. Phillip Rogaway. Formalizing Human Ignorance: Collision-Resistant Hashing without the Keys. In *Vietcrypt 2006*, volume 4341 of LNCS, pages 221–228. Springer-Verlag, 2006.
20. Ron Rivest. Abelian Square-Free Dithering for Iterated Hash Functions. Presented at *ECRYPT Hash Function Workshop*, June 21, 2005, Cracow.
21. D. R. Stinson. Some Observation on the Theory of Cryptographic Hash Functions. Journal of *Design, Codes and Cryptography*, Vol. 38, 259–277, 2006.
22. Serge Vaudenay. Secure Communications over Insecure Channels Based on Short Authenticated Strings. In *Advances in Cryptology–CRYPTO '05*(2005), Vol. 3621 of *LNCS*, Springer-Verlag, 309-326.

23. Xiaoyun Wang and Hongbo Yu. How to Break MD5 and Other Hash Functions. In *Advances in Cryptology–EUROCRYPT '05*(2005), Vol. 3494 of *LNCS*, Springer-Verlag, 19-35.

Appendix

A Previous work

BSSW protocol [1] is the first NIMAP based on collision resistant hash functions. The protocol is shown in Figure 4. The massage M and its hash are sent over the insecure channel and the manual channel, respectively. It can be shown that a (T, ϵ)- *collision finding* adversary can be transformed to a $(T + \mu, 1, \epsilon)$- *breaking* adversary against this NIMAP, where μ is the overall time complexity of the protocol (i.e., overall time complexity of Alice to respond to one query.) Therefore any collision finding adversary using only offline computations (based on Birthday attack) can be used to break this protocol.

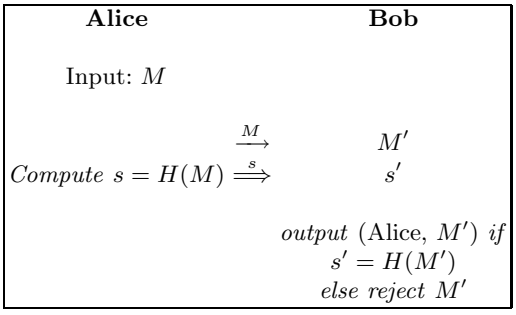

Fig. 4. BSSW protocol

PV protocol [17] is a manual channel NIMAP which uses both a hash function and a trapdoor commitment scheme in CRS model. Its security relies on second preimage resistance of the hash function and security of the trapdoor commitment scheme in the Common Reference String (CRS) model. In CRS model a public random string K_p is assumed to be accessible to all parties in the system. In the definition of a trapdoor commitment scheme to be used in PV protocol, as usual in CRS model, it is assumed that in a *setup*(.) phase a pair of keys (K_p, K_s) is generated and K_p is made publicly available to all parties. The key K_s is secret and can only be used by special algorithms (or oracles) in extensions of the commitment scheme. For example it can be used by *equivocate*(.)

algorithm in equivocable commitment schemes or by *extract*(.) algorithm in extractable commitment schemes. The protocol is shown in Figure 5. This NIMAP uses a weak security property of a hash function (i.e. second preimage resistance) but needs a secure trapdoor commitment scheme in CRS model (which is stronger than the standard model) as well.

The two algorithms, *commit*(.) and *open*(.) are used to generate (commit, decommit) values (represented by (c, d)) and to recover message, respectively. Both these algorithms have access to the CRS, K_p. The *commit*(.) algorithm is probabilistic (randomized) algorithm and *open*(.) is deterministic. In case of any error, *open*(.) outputs a special symbol $\perp$. More details can be found in [22, 17].

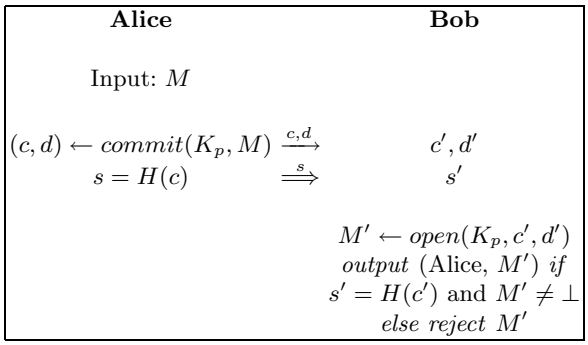

Fig. 5. PV protocol

As noted before the two message flows in PV protocol can be transformed into the form shown in Figure 1, by using *open*(.) function to obtain M and consider the message $(M, x) = (M, (c, d))$ as the message over insecure channel.

MS protocol [12] is a manual channel NIMAP which uses a hash function and requires the hash function to be Hybrid Collision Resistance(HCR) as defined in [12]. The protocol is in weak manual channel model and requires the same bandwidth for the manual channel as PV protocol (to reach to the same level of security). MS protocol is shown in Figure 6.

B On security notions for hash functions

There are some subtleties regarding formal definitions of security notions for hash functions and studying relations between different notions. A

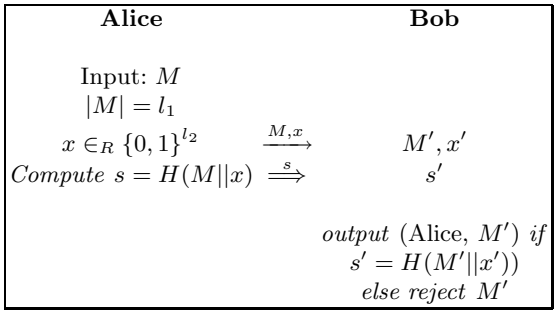

Fig. 6. MS protocol

crucial step in formally defining and comparing security notions is to make it clear that what one means by saying a certain secure hash function. There are usual two ways to view *a hash function*, namely seeing it as *a hash function family* or as *a fixed hash function*. Modeling a hash function as a Random Oracle can be seen as an extreme case in which one assumes a hash function family in which the family consists of all possible function with specified domain and range.

A hash function family $\mathcal{H}$ is a class of functions, each is from domain D to range $\{0,1\}^n$, indentified by an element from $\{0,1\}^k$. In other words a hash family is a two-argument mapping $\mathcal{H} : \{0,1\}^k \times D \rightarrow \{0,1\}^n$. More specifically, $\{0,1\}^k$ represents the set of strings of length k bits, whose elements are used as a key to select a function from the family, D represents the domain of function family, and n is the hash length of the ouput in bits. In this setting $\mathcal{H}$ is also named as *keyed* hash function. A member of this family (i.e. a fixed hash function belonging to this family) is selected by a key $K \in \{0,1\}^k$ and is denoted as $H(K,.)$ or $H_K(.)$. If domain D is the set of all strings of arbitrary length (in practice of length less than a huge number) the family is called an *arbitrary-input-length* hash function family or simply a hash function family. If domain only consists of strings of a fixed length (i.e., $D = \{0,1\}^m$ for a fixed m) the family is called a *fixed-input-length* hash function family or a *compression* function family.

A fixed (unkeyed) hash function H is a function (a one-argument mapping) $H : D \rightarrow \{0,1\}^n$. Similarly we have an (arbitrary-input-length) hash function or a compression function, if domain D is $\{0,1\}^*$ or $\{0,1\}^m$ (for some fixed m), respectively.

Most of efficient practical hash functions (like MD5, SHA1) are designed as a fixed (unkeyed) hash function and it has been a common practice in many of the cryptographic protocols to use a hash function

as a single function (not a family) and security of the protocol on some security assumptions on the hash function, e.g. assuming some properties like CR, SPR or PR from the hash function..

Here we reiterate a problem in giving a formal definition for collision resistance notion. It is well-known that treating a hash function as a family and not a single function, is the only way to give a formal definition of collision resistance notion. Defining CR as a game between an adversary and challenger for a fixed hash function (and saying that it is computationally hard to win this game) is problematic as there is no challenge from the challenger and so an adversary with a priori knowledge of a colliding pair for the function cannot be ruled out. More details on this can be found in[2, 18, 21]. We discuss this matter briefly at the end of this subsection.

For some of security notions for hash function (exempt CR notion), like second preimage resistance notion, there are both sensible formal definitions for a family of hash functions and a fixed hash function. But it is worth noticing that to compare two different security notions (i.e. studying their relative strength or showing separation results) both notions should be defined in the same setting, in order to stay away from fundamental formalization problems arise regarding mathematical meaning of definitions (like in CR notion as above).

Rogaway and Shrimpton [18] gave formal definitions in concrete security framework for basic security notions (CR, SPR, PR) of hash functions and some of their variants. These definitions are given in the setting of keyed hash functions, i.e. considering a family of a hash function rather than one fixed hash function. They also studied all relations (implications and separations) between these notions (in the keyed setting).

To point out some relations between a security notion defined for a family of hash functions and required security assumption(s) on a fixed hash function (to be a member of that family), we consider two security notions , called *aSec* and *aPre* and defined in [18], for a family of hash functions. The notions are defined in terms of games in which a key that is known to the adversary is chosen first and then the challenger chooses a random challenge. (Alternatively the key can be chosen by the adversary using the best strategy.) Here we point out the fact that existence of an *aSec* or *aPre* family of hash functions, say $\mathcal{H} : \{0,1\}^k \times \{0,1\}^* \rightarrow \{0,1\}^n$ implies existence of a fixed (unkeyed) hash function, $H' : \{0,1\}^* \rightarrow \{0,1\}^n$, that is SPR or PR, respectively, as defined by following games. Recall that $x \stackrel{R}{\leftarrow} X$ and $x \stackrel{\$}{\leftarrow} X$ represent randomly selecting an element x of the set X according to uniform dis-

tribution and some specific distribution, respectively.

Game(SPR$^{[m]}$, A)	**Game**(PR$^{[m]}$, A)
$M \xleftarrow{R} \{0,1\}^m$	$M \xleftarrow{R} \{0,1\}^m$; $Y = H'(M)$
$M' \xleftarrow{\$} A(M) \quad //M' \in \{0,1\}^*$	$M' \xleftarrow{\$} A(Y) \quad //M' \in \{0,1\}^*$
A wins the game if:	*A wins the game if*:
$M \neq M' \quad$ and $\quad H'(M) = H'(M')$	$H'(M') = Y$

We say that the hash function H' is $(T,\epsilon)-SPR^{[m]}$ or $(T,\epsilon)-PR^{[m]}$ if no adversary with time complexity at most T can win the corresponding game with probability at least ϵ. If H' is compression function (i.e., $H' : \{0,1\}^m \to \{0,1\}^n$), all inputs will have the same length and one can drop superscript m from notations(i.e., just say $(T,\epsilon) - SPR$ or $(T,\epsilon) - PR$ compression function).

Stinson [21] studied relations between security notions (Zero-Preimage, CR, SPR, and PR) for a fixed hash function via related games. To show an implication between two notions, a black-box reduction is used from any adversary winning one game to an adversary that wins the other game.

Let us end this brief overview by considering the notion of collision resistance. The *formal* definition of CR notion for a hash function *family* was proposed by Damgard [5, 6], in asymptotic security framework. A rephrased variant of this formal definition for a hash function family $\mathcal{H} : \{0,1\}^k \times D \to \{0,1\}^n$, in concrete security framework (as in [18]), is as follows:

Game(CR, A)
$K \xleftarrow{R} \{0,1\}^k$
$(M, M') \xleftarrow{\$} A(K) \quad // \ M, M' \in D$

A wins the game if:
$M \neq M' \quad$ and $\quad H_K(M) = H_K(M')$

A hash function family $\mathcal{H}$ is said (T,ϵ)-CR if no adversary with time complexity at most T can win the CR game above with a probability not smaller than ϵ.

As it is seen from CR game if one wants to consider a fixed hash function, then there would be no input (as a challenge) for adversary and so one cannot say that there is no (T,ϵ) adversary. Consider an adversary

that already saved a colliding pair M, M' in her/his memeory. Such a colliding pair is assured if hash function is compressing and so existence of such a simple adversary is already assured for any fixed (compressing) hash function. This may seem somewhat puzzling because security of many of protocols is based on CR property of a fixed hash function to be used in the protocol. Some options can be imagined for treating this matter. If it is possible modify the protocol to make it use a weaker than CR notion. Or modify it to let application of a hash function family (instead of only a single hash function) and then use a provably secure CR hash function family in it. But what if one wants to study and compare protocols as they are? An (informal) option is pointed out by Brown [4] (see also [21]) assuming CR as a *strong* property that " there is ***no known*** (T, ϵ) adversary" instead of assuming that "there is no (T, ϵ) adversary at all ".

Recently, Rogaway [19] has introduced an interesting way out of this CR formalization dilemma.

C On hardness of eSPR game in random oracle model

(T, L, ϵ)-eSPR property for a given (fixed) compression function is just a security assumption whose validity can be verified by the best cryptanalysis results against the specified compression function, and this is also the case for all other properties defined for a fixed compression function (not for a family of functions). For example considering a compression function like $md5 : \{0,1\}^{128+512} \rightarrow \{0,1\}^{128}$ used in the MD5 hash function, one can assume as a security notion that $md5$ is SPR or PR, but validity of such an assumption is not provable and one can have some feelings about this due to the fact that the best practical cryptanalysis results cannot do much so far.

Regarding the fact that eSPR notion is a very recent one at present we should wait for cryptanalysis results to evaluate popular practical hash functions like MD5 (not so popular now due to recent attacks like [23]) and SHA1. In the following we provide an *intuition* about hardness of eSPR game compared to SPR game under Random Oracle Model. The proof of following proposition is very similar(with some small modification) to that of HCR game as shown by Mashatan and Stinson[12].

Proposition 1 (eSPR difficulty in random oracle model). *Assume that H is a random function from the set of all functions with domain $\{0,1\}^{n+b}$ and range $\{0,1\}^n$ and every adversary has oracle access to it, i.e. can query M and obtain $H(M)$. Let $b \geq t$ and 2^t be much smaller*

than 2^n. Then for any adversary Eve making at most $T = 2^t$ queries to oracle H, an upper bound on the probability of the adversary winning eSPR game is $\epsilon \leq 2^{t-n} + 2^{2t-2n-b}$.

Proof (Hint). Note that because of modeling H as a random oracle we should only consider eSPR adversaries with $L = 2$, for by repeated invocation of random oracle, the output distribution (related to evaluated part in eSPR game) does not change. The rest of proof is very similar to proof of HCR difficulty in [12].

□