Noninteractive Manual Channel Message Authentication Based On eTCR Hash Functions

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

University of Wollongong, Australia 2500
Email: {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). The most recent NIMAP of Mashatan and Stinson (referred to as MS protocol) uses hybrid collision resistance (HCR) hash function and the HCR hardness is evaluated in the random oracle model. Our protocol in contrast, uses enhanced target collision resistant (eTCR) hash family and is provably secure in the standard model. To provide insights into the advantages of our protocol compared to the MS protocol and also other protocols with similar properties, we revisit a number of security notions for hash and study their relationship in terms of implication or separation, and also constructions that can be used for arbitrary length messages. This study is also of independent interest in the study of hash functions. We show that our new NIMAP can reach the same level of security as the best previously known protocols, whilst is simpler and more efficient from theoretical and practical view point. In particular in case for authentication of a messages such as a 1024 bit public key, our protocol enable one to construct the needed eTCR hash family from any off-the-shelf Merkle-Damgard hash function using randomized hashing mode. The construction only requires the underlying compression function to be *evaluated second preimage resistant(eSPR)*, which is strictly weaker than collision resistance. We note, however, that in the construction using randomize hashing, the reduction from eSPR to eTCR is not tight. We leave it as an open problem to find constructions with tight reduction.

Keywords: 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 no shared secret key between the communicants, nor there is public key infrastructure. However the sender, in addition to an insecure broad-band 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 read a short number on the display part of one device and input it into a second device using its 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 have 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 section refprevious-NIMAPs) which corresponds strongest adversary. We note that in NIMAP the scarce resource is the bandwidth of the manual channel.

Computationally secure NIMAPs using manual authentication.

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 the security of the protocol to collision resistance property of the hash function. To guarantee security against an adversary having time $T = 2^n$, the SAS length must be at least $2n$.

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 strong requirements on the 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 in comparison with BSSW that uses a collision resistant hash function, PV protocol has a weaker security requirement on the hash function (i.e. second preimage resistance), but it needs a secure trapdoor commitment scheme in CRS model which make it a more complex and 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. Usingrandom oracle model it is shown that this is a weaker security property 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 and added assumption of CRS. In section 3 we investigate HCR notion and show there is no clear method for instantiating the hash function used in the protocol and so this leaves efficient construction of NIMAPs forarbitrary length message, in weak manual authentication model, an open problem.

More details on BSSW, PV, and MS protocols is given in Appendix A.

Our Contributions. We propose a new NIMAP in weak manual channel model. Our new NIMAP only 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. Such a family of hash functions can be constructed using randomized hashing mode of a Merkle-Damgard 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) of hash functions, in two directions. Firstly, in terms of implication or separation. That is showing whether one property implies the other one, or there is a clear separation between them. Secondly, we consider constructions that ensure a defined property holds when hashing arbitrary length messages. This latter requirement is important because the length of a message in a manual authentication scenario cannot be restricted. Our comparison also includes *evaluated second preimage resistance (eSPR)* property, a property for compression functions introduced to construct an eTCR hash function family using Merkle-Damgard construction in the randomized hashing mode as proposed by Halevi and Krawczyk [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 interest both because of its direct application to NIMAP but also from the view point of grading properties of hash functions.

Paper Organization. In section 2 we describe communication and security model for manual channel authentication which will be used in evaluation of our proposed protocol. In section 3 we give an overview of security notions for hash function and then provide more details on three security notions, eSPR, eTCR and HCR, directly related to our NIMAP and MS protocol. In section 4 we present our new protocol and analyze its security. We also compare it with previous protocols and show its potential advantages. The paper is concluded in section 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 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 and 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 to 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 requirement 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.

A typical manual channel NIMAP works as follows. 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-coding) into our proposed representation. The transformation is public and so will not affect security of the protocol. Received messages by Bob are denoted as 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 (Successful attack) *An adversary Eve, having resources Q (number of queries made from Alice) and T (time complexity), is successful if with probability at least ϵ, she can make Bob output (Alice , M') while M' has never been an input of protocol on Alice side, i.e. it has*

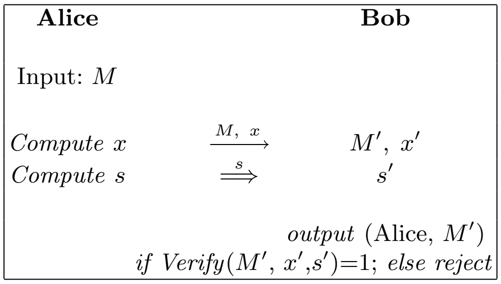

Fig. 1. A typical manual channel NIMAP

never been authenticated by Alice. The protocol is called (T, Q, ϵ)*-secure if there is no* (T, Q, ϵ)*-breaking adversary against it.*

Note that to be considered a successful adversary, Eve should respect the communication and security model restrictions above. For example she can only replay a previously obtained s from Alice but she cannot modify it or inject a new one by herself. 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 any (T, Q, ϵ)-breaking adversary to a $(T', 1, \epsilon')$-breaking adversary, where $\epsilon' = \frac{\epsilon}{Q}$.

3 Hash function and some 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. In some cases the definition is very application specific. For example *Zero-Finder-Resistant* was defined by Brown [4] as difficulty of finding a preimage of zero (i.e. finding a domain element that is hashed to 0) and was shown to be a necessary security assumption on the hash function proving 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 or more 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)* (as rephrased in concrete security framework in [3]).

Briefly and *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 it is computationally hard to find a sibling M', for any given input M, so 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 any given hash value (range element y), so that these constitute a valid (input, output) pair for hash function (i.e. $H(x) = y$).

There are some subtleties regarding formal definitions of security notions for hash functions and studying relations between different notions. A brief summary is provided in Appendix B

3.1 Definitions for eSPR, eTCR and HCR notions

We review three security notions relevant to our discussion (, as well as previously mentioned CR and SPR notions,) in the next section.

The following definition is as in [12], but here the game is parameterized by the length of the provided randomness by indicating l_2 explicitly in the definition.(l_2 and n are security related parameters as shown in [12].) We introduce a state variable *State*, for adversary A, to keep the adversary state 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*:*

```
$\textbf{\textit{Game}}\textit{(HCR}^{[l_2]}\textit{, A)}$
$(M, State) \overset{\$}{\leftarrow} A() \quad //M \in \{0,1\}^{l_1}$
$M' \overset{R}{\leftarrow} \{0,1\}^{l_2}$
$M'' \overset{\$}{\leftarrow} A(M', State) \quad //M'' \in \{0,1\}^{l_1+l_2}$

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

Notice that $HCR^{[l_2]}$ notion for an arbitrary-input-length hash function $H : \{0,1\}^* \rightarrow \{0,1\}^n$ (i.e. if one lets length l_1 be arbitrary) will be a game in which the adversary can output $M \in \{0,1\}^*$ and $M'' \in \{0,1\}^*$, in the above game.

eSPR notion [9] is defined for a compression function using Merkle-Damgard [13, 6] domain extension. Merkle-Damgard method converts a compression function to a hash function for arbitrary length input, while preserving CR property of the compression function. Recently some modifications of this method were shown in [3, 21] that preserve TCR notion while extending hash domain.

For a compression function $H : \{0,1\}^{n+b} \rightarrow \{0,1\}^n$, an L-round Merkle-Damgard construction is a method to construct a compression 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 to 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 input message length is not a multiple of block length b, proper padding can be used. For a fixed initial value C_0 we denote the transformation by $MD_L^{C_0}[H] : \{0,1\}^{Lb} \rightarrow \{0,1\}^n$.

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*

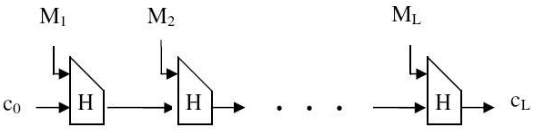

Fig. 2. L-round Merkle-Damgard construction

and given to the adversary (uniform setting) or it is a parameter of the game that adversary may depend on it (non-uniform setting).

```
**Game**(eSPR, A )
$\Delta_1, \ldots, \Delta_L \overset{\$}{\leftarrow} A() \quad //\Delta_i \in \{0,1\}^b, \quad L \geq 2$
$r \overset{R}{\leftarrow} \{0,1\}^b$
$M = \Delta_L \oplus r; \quad C = MD^{C_0}_{L-1}[H](\Delta_1 \oplus r, ..., \Delta_{L-1} \oplus r)$
$M' \overset{\$}{\leftarrow} A(C, M) \quad //M' \in \{0,1\}^{n+b}$ is of the form $(C, M)$

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

eTCR notion [9] is defined for an arbitrary-input-length hash function *family* (unlike HCR or eSPR notions that are defined for a fixed 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:*

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

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

A possible method to construct an eTCR hash function family is using an iterated hash method (e.g. Merkle-Damgard construction) employing a compression function, like in randomized hashing mode of [9]. In such a construction eTCR notion is reduced to eSPR assumption on the compression function (as in Theorem 2), and the length of messages (in blocks) used by the adversary is denoted by L and is considered as another resource parameter of the adversary. So the adversary is denoted as a (T, L, ϵ) adversary.

3.2 Relations among eSPR, eTCR and HCR notions

In this section we study relations between three recent notions, namely, eSPR, eTCR and HCR ntions. First we argue that eSPR notion is not strictly stronger than HCR notion, considering an arbitrary compression function. Then after showing a simple fact that assuming existence of a HCR function implies existence of a (specific) eTCR family , we discuss some practical limitations in designing a HCR secure arbitrary-input-length hash function (in a *proper* manner), by pointing out that Merkle-Damgard method does not suit for this aim. Notice that a proper construction for a HCR hash function is a construction by which one can get HCR-ness without having to have CR property, else it is obvious that any CR function is also HCR. Finally, we review the randomized hashing mode (proposed in [9]), for constructing an arbitrary-input-length eTCR family based on an eSPR compression function, using Merkle-Damgard method.

eSPR versus HCR We want to point out that eSPR assumption is not strictly stronger than HCR assumption for any compression function. The desired fact can be seen combining the following two already known facts. Halevi et al [9] pointed out a separation between eSPR and SPR (denoted r-SPR in [9]). They argued that depending on the structure of a compression function eSPR assumption can be weaker than SPR assumption (i.e., winning eSPR game can be harder than winning SPR game, for some compression functions). We also note that HCR$^{[b]}$ assumption (for any b, and)[1] for any compression function is strictly stronger than SPR assumption. So, it follows that for some compression functions eSPR assumption can be weaker than HCR assumption.
Notice that here we do not need to argue about the cases that HCR (, as

[1] Here we denote HCR game parameter by b to remind that in eSPR game the amount of provided randomness is also b bits.

security assumption of MS protocol) may be weaker than eSPR[2], as our aim is in fact showing the converse of this fact.[3]

Relation between HCR and eTCR We want to show that existence of a (T,ϵ)-HCR $^{[l_2]}$ compression function implies a construction for a (T,ϵ)- eTCR compression function family (namely, a keyed compression function whose key length is l_2 bits).
Assume that we have a (T,ϵ)-HCR$^{[l_2]}$ compression function
$H:\{0,1\}^{l_1+l_2}\rightarrow\{0,1\}^n$. We can easily construct a (specific) compression function family that is (T,ϵ)- eTCR as: $\mathcal{H}=\{H_K\}_{K\in\{0,1\}^{l_2}}$, where $H_K(M)=H(M||k)$ (i.e., $\mathcal{H}:\{0,1\}^{l_2}\times\{0,1\}^{l_1}\rightarrow\{0,1\}^n$). This is easily seen from definitions of HCR and eTCR notions.

HCR for arbitrary length messages We want to show that in Merkle-Damgard hash functions, one cannot achieve more immunity by assuming HCR property rather than CR property. In other words we show that failure of such a hash function to collision finding attacks will also yield to failure in HCR sense.

It is shown in [12] that, in Random Oracle Model, winning HCR game is more difficult than finding any collisions. This could suggest that compared to NIMAPs that rely on CR property, NIMAPs based on HCR assumption would be easier to achieve. In the following we show that for iterated hash functions (like SHA1), a collision finding attack against compression function can also be used to construct an HCR attacker for the hash function and so one cannot use these type of hash functions to be more immune than CR case.

The standard paradigm for constructing an arbitrary-input-length hash function (family), having some defined security property, consists of two steps. First a fixed-input-length hash function(family) (i.e. a compression function) is constructed (having some defined security property) and then a domain extension method is used to convert this compression function (family) to the required arbitrary-input-length hash function (family). A commonly used method of domain extending is Merkle-Damgard method and its extensions [3, 21]. We show that this method cannot be employed for constructing an arbitrary-input-length HCR hash function

[2] If randomized hashing mode used to construct eTCR family, eSPR will be required security assumption on the compression functions security assumption

[3] It is not so hard to show (using counterexamples) that for some compression functions HCR$^{[b]}$ can be weaker than eSPR and complete the argument that there is a separation between HCR and eSPR notions

without having to make CR assumption on the underlying compression function.

Let H be a strengthened Merkle-Damgard hash function (e.g. SHA1). By strengthened Merkle-Damgard we mean Merkle-Damgard with length indicating padding and constant initial value. Now consider the case that the sum of lengths of the message to be sent in NIMAP (i.e. l_1) and the security parameter l_2 (e.g. $l_2 = 70$ as proposed in [12]), becomes more than one block (e.g. more than 512 bits in SHA1). In this case any collision finding adversary A against H can be used to construct an algorithm B that defeats H in $\mathrm{HCR}^{[l_2]}$ sense as follows. Algorithm, B, invokes A to obtain two (equal length) colliding messages M and M' under H. W.l.o.g we assume that their length is a multiple of block length (i.e. $l_1 \in \{0,1\}^{512.m}$ in SHA1 for some integer m), for we can consider complete blocks consisting of messages together with padding as M and M'. In the first move of HCR game, algorithm B commits on M and when receives random challenge $K \in \{0,1\}^{l_2}$, it outputs $M'||K$ as colliding pair with $M||K$.

Reduction from eSPR to eTCR Halevi and Krawczyk [9] constructed a provably secure eTCR hash function family from a compression function that satisfies eSPR assumption by randomizing the input message in a Merkle-Damgard based hash function (like SHA1).

This gives an arbitrary-input-length eTCR hash function family that is provably secure (based on eSPR assumption on the compression function). We note that eSPR is not a strictly stronger assumption than CR about a compression function.

More details can be found in Appendix C. We will use Halevi-Krawczyk's result (Theorem 2 in Appendix C) in following section.

On Hardness of eSPR game.

$(T, L, \epsilon\text{-})$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 a provable matter and) one can have some *intuitions* about this as a result of the fact that at present the best practical cryptanalysis results cannot do much or by some abstract modeling like Random Oracle Model, i.e. modeling a hash function as a random oracle.

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 Appendix C, taking the other easy step, i.e. Random Oracle Model, we provide an intuition about eSPR game difficulty.

4 A new NIMAP based on any eTCR hash function family

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 a huge number representing maximum allowed input length(e.g., $m = 2^{64}$). We want to have a secure NIMAP between a claimant, Alice, and a verifier, Bob, in weak manual channel model. In this model Alice can send messages of length up to m bits (practically up to a length which is determined by required security level) in an authenticated way, over insecure channel using a short authenticated message over manual channel. Our new 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 insecure broadband channel and sends $s = H_x(M)$ over 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.

It is easy to prove the security of this new NIMAP as stated in the following Theorem.

Theorem 1 *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 $T = T_H - \mu Q$, $\epsilon = Q\epsilon_H$ and constant μ represents the maximum time complexity of Alice over all Q queries.*

Proof: First we show that any $(T', 1, \epsilon')$-breaking adversary $\widehat{A}$ against our NIMAP can be used to construct a (T', ϵ')-breaking adversary B against eTCR hash family $\mathcal{H}$. Then the proof is completed as a result of a simple

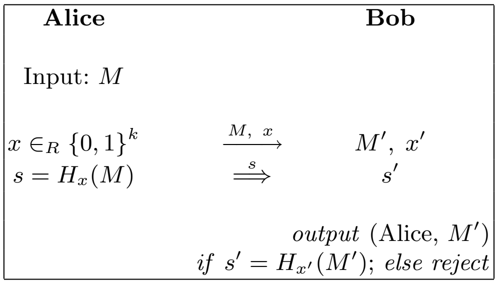

Fig. 3. A new manual channel NIMAP based on eTCR hash family

to prove general reduction from any (T, Q, ϵ)-breaking adversary A to $(T', 1, \epsilon')$-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 our NIMAP, i.e. it can make only one query from Alice to obtain (M, x, s) and then spends time at most T' to mount a successful attack, i.e. to produce (M', x') where $M' \neq M$ and $H_{x'}(M') = s$. There is no condition on x', it can be equal to x or not. Adversary B against H plays eTCR game using $\widehat{A}$ as follows. It runs $\widehat{A}$ and obtains the query M then 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 $s = H_x(M)$ 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. It is obvious that B succeeds at the same time T' and with the same success probability ϵ'.

The second part of proof is already known as a general transformation between any Q-query adversary and 1-query adversary, e.g. [22]. For completeness of our proof, here we include a proof for our simple case(i.e. two-party NIMAP). Let A be a (T, Q, ϵ)-breaking adversary against our 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 else when $i = j$ in which $\widehat{A}$ forwards the query ($j - th$ query of A) to Alice (in real protocol) and uses obtained response from Alice 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 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 security level, from:

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

Security assumptions. It is often plausible to reach a security goal for a protocol in a more efficient way using stronger building blocks, i.e. requiring primitives which should provide stronger security properties. For instance, constructing UOWHF hash families[14] using one-way permutations is much simpler than using one-way functions. However, in practice, the stronger the property a primitive should provide, the harder would it be to construct such a primitive. This motivate researchers to design protocols using weaker assumptions on the underlying primitives. In the following we compare the above mentioned protocols in this manner.

The BSSW protocol uses a fixed (unkeyed) hash function and requires it to be collision resistant (CR). As discussed in Appendix B CR is a very strong assumption which cannot even be formally defined for a single hash function. A possible way out from this formalization problem can be using a CR hash function family (i.e., a keyed hash function), but then one needs to send the key via manual channel as well as hash value that increases the SAS length.

The PV protocol uses a weaker assumption than BSSW protocol, SPR. However, it requires a secure trapdoor commitment scheme in CRS model in addition. So the security level is determined by both the hash function and the trapdoor commitment scheme. Furthermore, the commitment is taken as an input to the hash function; to independently choose these two primitives, one needs the hash domain to be of arbitrary size, i.e., one needs an arbitrary-input-length hash function.

The MS protocol also uses a fixed hash function satisfying HCR property. The HCR$^{[l]}$ is a notion between CR and SPR, depending on the value of l. Unfortunately, the *standard domain extension* Merkle-Damgard construction cannot be used if one wants to construct arbitrary-input-length HCR hash functions from fixed-input-length ones.

Different to BSSW, PV and MS protocol, we use an assumption (eTCR) on hash *family* instead of a fixed hash function. This let us be able deal with the security formally. In fact, using the standard Merkle-Damgard iteration in randomized hashing mode, eTCR hash family can be easily constructed from any eSPR compression function. That means, the security of our protocol can also be reduced to the eSPR notion on a fixed-input-length hash function. It has been shown [9] that eSPR is a strictly weaker than CR assumption. We also showed in section 3 that eSPR is not stronger than SPR or HCR notions. In this sense, our protocol has priority over BSSW, PV and MS protocols.

Manual channel bandwidth. We assume an adversary with the same resources and the level of security (denoted by ϵ) assumed in [12]. Then we compare our protocol to other NIMAPs. 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 claimed, but that requires a demanding assumption of ideally secure SPR hash function. MS can theoretically reach the same level of security using a SAS of 100 bits for $l_2 = 70$ bits.

Compared to BSSW, 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 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, and benefits from a weaker security assumption. One may argue that our SAS length increases logarithmically in length of message. But as will be seen from the following discussion, the claims in PV and MS will not be true for Merkle-Damgard hash functions like MD5, SHA-1, RIPEMD-160, Whirlpool.

PV reaches the short SAS length (i.e, 100) assuming an ideally secure SPR hash function, i.e., a hash function with security level of 2^{-n}, where n is the hash size. This assumption (for an arbitrary-input-length hash function) is very demanding due to Kelsey-Schneier [11] generic attack against iterated hash functions (like MD5, SHA1, RIPEMD-160, Whirlpool) [11]. So a 100-bit SAS will not provide claimed security level of PV for such hash functions.

Also, the SAS of 100 bits in MS should assume that one has a specially designed arbitrary-input-length hash function, for which HCR property is really weaker and easier to achieve than CR property. To the best of our knowledge, there is no such constructions of HCR hash functions in the literature yet. Recall that the iterated Merkle-Damgard method can not be used for this purpose.

Now we show how the SAS length of our protocol is calculated. Using Theorem 1, it is seen that to have our NIMAP be (T, Q, ϵ)-secure (for $T = 2^{70},\ Q = 2^{10},\ \epsilon = 2^{-20}$) we need a hash function family that is $(2^{70}+\mu 2^{10}, 2^{-30})$- eTCR.(Note that μ is a small constant time required to select the random string x and to compute $H_x(M)$ and so the second term of time complexity i.e. $\mu 2^{10}$ is negligible compared to 2^{70}.) Using Theorem 2 in Appendix C, we know that we can construct an eTCR family with $\epsilon = 2^{-30}$ based on the assumption that compression function is eSPR with $\epsilon' = \frac{\epsilon}{L+2}$, where L is the number of blocks in an input message of constructed eTCR functions. Because of non-tightness of the reduction between eTCR and eSPR notions one should compute the required length of SAS for each message length. Assume that we use a compression function with input length 640 bits(i.e., block length $b = 512$ bits) and hash size n bits (e.g. compression function of SHA1 whose output is reduced to n bits, by truncation or applying another proper output function). We assume (as a security assumption) that this compression function provides security level of 2^{-n} in eSPR sense. Using Theorem 2 from Appendix C, for messages of length $512L$ bits, our NIMAP needs a SAS with length $n = 100 + log_2(L+2)$ bits. Notice that if one uses a compression function with hash size n bits, but a larger input length (i.e. block length larger than 512 bits), the number of blocks, i.e., L will be reduced and so a shorter SAS will be needed in our protocol.

5 Conclusion

We proposed a new non-interactive message authentication protocol in manual channel model. The proposed NIMAP uses a family of eTCR secure hash functions. For some of interesting applications, like sending a public key, where message length is small, e.g. 1024 bits, there is already a randomized hashing mode to construct an eTCR hash family using any off-the-shelf Merkle-Damgard hash function like SHA1. In these cases, using our NIMAP, one requires a significantly weaker than collision resistance property from underlying compression function, namely eSPR property, and without need to any change into internal structure of used hash function (e.g. SHA1 code can used as is in randomized hashing mode). If one wants to send very long messages in imaginable special applications like authenticated file transfer through insecure broadband channel (with an emphasize on not using any confidentiality or MAC algorithm), randomized hashing mode is not an optimal way (from SAS length viewpoint in our NIMAP) to construct eTCR hash family. This is due to non-tightness of its security reduction. Although we can still use randomized hashing mode for messages up to about 2^{49} bits using a SAS of only 140 bits in our NIMAP, we recommend constructing an eTCR hash family with a tight security reduction from a weaker than CR notion or from some general complexity assumptions. We leave this as an open question. Any such eTCR hash function family can be used in our NIMAP.

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, Proc. of CRYPTO 97, pp. 470484, Full version of this paper is available from http://www-cse.ucsd.edu/users/mihir/, 1997.
4. D. Brown, Generic groups, collision resistance and ECDSA, Designs, Codes and Cryptography, Vol. 35 (2005) 119152.
5. I.B. Damgard, Collision free hash functions and public key signature schemes, Advances in Cryptology, Proc. Eurocrypt87, LNCS 304, D. Chaum and W.L. Price, Eds., Springer-Verlag, 1988, pp. 203216.
6. I.B. Damgard, A design principle for hash functions, *Advances in Cryptology, Proc. Crypto89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 416427.
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, Crypto 2006, LNCS 4117, pp. 41-59, 2006.
10. D. Hong, B. Preneel and S. Lee, Higher Order Universal One-Way Hash Functions, Proc. ASIACRYPT 2004, LNCS 3329, pp. 201-213, 2004.
11. John Kelsey and Bruce Schneier, Second Preimages on n-Bit Hash Functions for Much Less than 2n Work, EUROCRYPT 2005.
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, *Advances in Cryptology, Proc. Crypto89, LNCS 435*, G. Brassard, Ed., Springer-Verlag, 1990, pp. 428446.
14. M. Naor and M. Yung, "Universal one-way hash functions and their cryptographic applications," *Proc. 21st ACM Symposium on the Theory of Computing*, 1990, pp. 387-394.

15. M. Naor, G. Segev, and A. Smith. Tight bounds for unconditional authentication protocols in the manual channel and shared key models. Cryptology eprint archive, report 2006/175, http://eprint.iacr.org/2006/175.pdf.
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, volume 3860 of Lecture Notes in Computer Science, 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. FSE 2004, 371-388.
19. Ron Rivest, Abelian square-free dithering for iterated hash functions, Presented at ECrypt Hash Function Workshop, June 21, 2005, Cracow.
20. D. R. Stinson, Some Observation on the Theory of Cryptographic Hash Functions, Journal of Design, Codes and Cryptography, 38, 259-277, 2006.
21. V. Shoup, A composite theorem for universal one-way hash functions, Proc. of EUROCRYPT 2000, pp. 445452, 2000.
22. Serge Vaudenay. Secure communications over insecure channels based on short authenticated strings. In Victor Shoup, editor, Advances in Cryptology CRYPTO 05: The 25th Annual International Cryptology Conference, volume 3621 of Lecture Notes in Computer Science, pages 309326, Santa Barbara, California, U.S.A., August 2005. Springer-Verlag.
23. Xiaoyun Wang and Hongbo Yu, How to Break MD5 and Other Hash Functions, EUROCRYPT 2005.
24. Xiaoyun Wang, Yiqun Lisa Yin, and Hongbo Yu, Finding Collisions in the Full SHA-1, CRYPTO 2005.

Appendix

A Previous Works

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.)

It is easy to see that any collision finding adversary (e.g. using only offline computations based on Birthday attack) can be used to break this NIMAP.

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

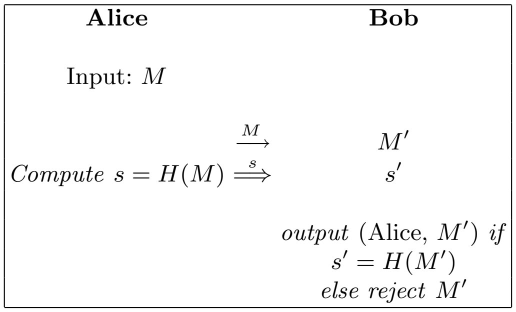
Alice
Bob
Input: M
$\xrightarrow{M}$
M'
$Compute\ s = H(M) \overset{s}{\Longrightarrow}$
s'
output (Alice, M') *if*
$s' = H(M')$
else reject M'

Fig. 4. BSSW protocol

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 the case of any error *open*(.) outputs a special symbol $\perp$. More details can be found in [22, 17].

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.

$$\begin{array}{ccc}
\textbf{Alice} & & \textbf{Bob} \\
 & & \\
\text{Input: } M & & \\
 & & \\
(c,d) \leftarrow \mathit{commit}(K_p, M) & \xrightarrow{c,d} & c', d' \\
s = H(c) & \xRightarrow{s} & s' \\
 & & \\
 & & M' \leftarrow \mathit{open}(K_p, c', d') \\
 & & \mathit{output}\ (\text{Alice, } M')\ \mathit{if} \\
 & & s' = H(c') \text{ and } M' \neq \perp \\
 & & \mathit{else\ reject}\ M'
\end{array}$$

Fig. 5. PV protocol

$$\begin{array}{ccc}
\textbf{Alice} & & \textbf{Bob} \\
 & & \\
\text{Input: } M & & \\
|M| = l_1 & & \\
x \in_R \{0,1\}^{l_2} & \xrightarrow{M,x} & M', x' \\
\mathit{Compute}\ s = H(M||x) & \xRightarrow{s} & s' \\
 & & \\
 & & \mathit{output}\ (\text{Alice, } M')\ \mathit{if} \\
 & & s' = H(M'||x')) \\
 & & \mathit{else\ reject}\ M'
\end{array}$$

Fig. 6. MS protocol

B On security notions for hash functions

A crucial step in formally defining and comparing security notions is to make it clear that what one means by a hash function. There are two proper 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.

Assuming that binary alphabet ($\{0,1\}$) used for representation of strings, A hash function family $\mathcal{H}$ is a family of functions (a two-argument mapping) $\mathcal{H} : \{0,1\}^k \times D \rightarrow \{0,1\}^n$. In this notation, $\{0,1\}^k$ 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 in bits. This setting is also called 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 for brevity just referred to as 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, 20]. 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 $\mathcal{H} : \{0,1\}^k \times \{0,1\}^* \to \{0,1\}^n$ implies existence of a fixed (unkeyed) hash function, $H' : \{0,1\}^* \to \{0,1\}^n$, that is SPR or PR, respectively, as defined by following games. ($x \overset{R}{\leftarrow} X$ and $x \overset{\$}{\leftarrow} X$ represent randomly selecting an element x of the set X according to uniform distribution and some specific distribution, respectively.)

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

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 [20] 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, $A$)
$K \xleftarrow{R} \{0,1\}^k$
$(M, M') \xleftarrow{\$} A(K)$    // $M, M' \in D$

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

A hash function family $\mathcal{H}$ is said $(T, \epsilon) - CR$ if no adversary with time complexity at most T can win the corresponding game with probability at least 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 (instaed 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 [20]) 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 ".

C Relations between eSPR and eTCR

The following theorem reproduced from [9] shows practicality of an eTCR hash function family.

Theorem 2 *[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-Damgard 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 se-*

cure. This family is constructed as $\widetilde{\widetilde{H}_r}(M) = \widetilde{H}(r, M) = MD^{C_0}_{L+1}[h](r, M_1 \oplus r \ldots M_L \oplus r)$*, where* $M = M_1||...||M_L$ *and* C_0 *is a known initial value.*

The second property in addition to eSPR , i.e. being (T', ϵ')- OWH, is a flavor of one-wayness that, assuming a mild structural property for compression function, is implied by eSPR (i.e. $\epsilon' \leq \epsilon$) and so for most practical compression functions that compress a reasonable amount, say 128 bits, it is a redundant assumption and actually the assumption about compression function is still eSPR. We refer the reader to [9] for more discussion on this matter.

On Hardness of eSPR game in Random Oracle Model:
The proof of following proposition on eSPR difficulty is very similar(with some small modification) to that of HCR game as shown by Mashatan and Stinson[12] and is omitted for brevity.(*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])

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 only oracle access to it, i.e. can query* M *and obtain* $H(M)$*. For any adversary* A *making at most* $T = 2^t$ *queries from oracle* H *an upper bound for the success probability in winning eSPR game, provided that* 2^t *is small compared to* 2^n *and* $b \geq t$*, is* $\epsilon \leq 2^{t-n} + 2^{2t-2n-b}$*.*

Notice that the first term in success probability (i.e. 2^{t-n}) is as one expects for SPR notion and the second term can be made negligible (compared to the first term) for proper values of b.