Separating two roles of hashing in one-way message authentication

L.H. Nguyen and A.W. Roscoe
Oxford University Computing Laboratory
E-mail addresses: {Long.Nguyen, Bill.Roscoe@comlab.ox.ac.uk}

Abstract

We analyse two new and related families of one-way authentication protocols, where a party wants to authenticate its public information to another. In the first, the objective is to do without shared passwords or a PKI, making use of low-bandwidth empirical/authentic channels where messages cannot be faked or modified. The analysis of these leads to a new security principle, termed *separation of security concerns*, under which protocols should be designed to tackle one-shot attacks and combinatorial search separately. This also leads us develop a new class of protocols for the case such as PKI where a relatively expensive signature mechanism exists. We demonstrate as part of this work that a popular protocol in the area, termed MANA I, neither optimises human effort nor offers as much security as had previously been believed. We offer a number of improved versions for MANA I that provides more security for half the empirical work, using a more general empirical channel.

1 Introduction

We examine some protocols which attempt to transmit a (possibly very long) message from one party to another in such a way that the origin and integrity of the message are authenticated. Initially we set out to do this with just one-way communication and authentication strings without the presence of any initial security infrastructure. This illustrates the power of authenticated empirical channels that are authentic, unspoofable or unfakable, but on the other hands, can be overheard by anyone.

To set this work in context, recall the classic one-way authentication protocol which works where there is a PKI. Here a message M and the name of the sender A are accompanied by the digital signature, or *message authentication code*[1] (MAC) $\{hash(M)\}_{sk(A)}$ and possibly the public-key certificate of the sender A. The receiver knows M really is from A because he can form the cryptographic hash of M and discover if it really was A who signed this value with her secret key.

Although the whole of such a message may be assumed to be sent over a standard Dolev-Yao channel, there is in fact a closer tie-in with the subject matter of this paper than there might appear to be. For public key encryption and decryption are computationally expensive: this means that there is a strong incentive to keep the bandwidth of information transmitted under this form

[1] In this paper we investigate a variety of techniques for providing authentication and integrity evidence for a message. We are inclined, therefore, to use the name “MAC” for a rather wide class of such techniques including ones based on asymmetric cryptography and further concepts that we will discover, rather than just referring, for example, to cases where the participants share a symmetric key.

of cipher to a minimum. We might therefore regard the single message described above as the combination of a (perhaps large) message M over an insecure channel with the smaller one $hash(M)$ over an authenticated one.

Since in many cases the empirical channels we will be using are human mediated, the chief difference from the above analysis is be that our empirical channels are much lower bandwidth yet: the amount of security delivered for a given amount of empirical communication becomes the most important measure of a protocol's effectiveness.

We start by describing a number of non-interactive one-way authentication schemes that use empirical channels in different ways, for example: protocols of Pasini and Vaudenay [11], and Balfanz [1]. These require the transmission of more bits over empirical channels than desirable.

Subsequently, we observe the development of MANA I by Gehrmann, Mitchell and Nyberg [3] that can reduce the number of bits required to be transmitted over the empirical channel significantly. This was the first example in the literature where useful combinatorial search was largely prevented. However, we will see that the scheme neither optimises human effort nor offers as much security as has previously been believed. We offer a number of improved versions for MANA I that provides more security for half the empirical work, using a more general empirical channel.

Having done this work in the context of "authentic" channels, we are able to formulate a general principle from it: the principle of *separation of security concerns*, under which protocols should be designed to resist a one-shot attack and combinatorial search separately. In turn, this principle allows us to devise a new family of protocols that work in the context of a PKI or similar. These schemes will, we believe, often have efficiency advantages over the conventional signature described above thanks to the use of short output *digest* functions that can be computed more efficiently than cryptographic hashes.

1.1 Notation

Capital letters such as A and B are used to identify parties. In common with much of the literature we are citing, the combination of two pieces of data will frequently be written $x \parallel y$ or the ordered pair (x, y). We will assume node A has some public information M that it wants to have authenticated to node B, this might include name/addressing information, its uncertificated public key or Diffie-Hellman token. The notations for several types of channel are given below. These are taken from [8].

- $\longrightarrow_N$, all messages transmitted over the Dolev-Yao network can be overheard, deleted or modified by the intruder.

- $\longrightarrow_{WE}$, this *weak empirical* channel cannot be forged, but it can be blocked, overheard, delayed or replayed [14, 11].

- $\longrightarrow_E$, this is like a weak empirical channel except that it cannot be replayed. It can be delayed, but not sufficiently long so that a message from one session can be used in another [6, 7].[2]

[2] In practice, it is not hard to avoid messages transmitted over the empirical channel being delayed from one to a latter session because it is normally the humans who run the channel. For example, if the humans involved are not away at any time during a protocol run, then empirical messages cannot be delayed from one to another session.

- $\longrightarrow_{SE}$, this is similar to a normal empirical channel, but it also provides stall-free transmission, and cannot be delayed, removed or blocked by the intruder. This is termed a *strong empirical channel* [14, 11].
- $\longrightarrow^{t}_{BE}$ is the same as $\longrightarrow_{SE}$ except that messages cannot take more than time t to arrive. In other words, no empirical message can be accepted more than t time units after it was sent. We will call this a *bounded delay empirical channel.*

Some of the proposed mechanisms such as Authentication without hashing presented in Section 5 and the Improved MANA I make use of a new cryptographic primitive termed a *digest* function. While the desired properties of a digest function are similar to those required of cryptographic hash functions, universal hash functions, and the MAC or check function of [3], it has a less challenging specification than a hash, and is frequently intended to be short output (perhaps 16 to 32 bits). We have previously described digest functions in [6, 7]. The uses we make of them in Sections 5.1 and 5.2 of the present paper are qualitively different, however, since (i) it is no longer necessary that the output of the function is short, and (ii) the only motivation for using them is that they are faster to compute than a hash.

The following is the specification of a short-output, b-bit digest. The specification for a general digest is the same except that 2^{-b} is replaced by some small ϵ and collision w.r.t different keys is also taken into consideration.

As the key k varies uniformly over its range:

1. $digest(k, M)$ is uniformly distributed for any fixed M.
2. For any fixed θ and $M \neq M'$: $\mathbf{Pr}(digest(k, M) = digest(k, M')) \leq 2^{-b}$

The computational complexity of digest functions is crucial for Section 5, as indicated above. The following complexity model is taken from [6, 7, 8]. It is clear that the cost of computing the b-bit output $hash_b(M)$ or $digest(k, M)$ increases linearly with the length of M. It also seems clear that it will increase significantly with b, and a simple model in which each word of a running temporary value of length b is combined with each input word suggests our overall model might be $b \times length(M)$. Since well-known hash algorithms tend to be fixed width, and vary significantly in their individual costs, it is hard to be too definite about this rule, although the nature of the individual algorithms tends to support our hypothesis. We will discuss this issue further in Section 6.

We now summarise an idealised framework for the digest function, proposed in [6, 7]. This has been formally proved to satisfy the above specification exactly. In practice (as opposed to *idealised*) the random numbers required by this scheme would be simulated by a pseudo-random number generator.

Suppose we want to construct a b-bit digest of a $(K-1)$-bit message M. The first thing we do is to pad M with an extra 1-bit at the end, so its length becomes K with $M_K = 1$. For $i = 1, \ldots, b$ and $j = 1, \ldots, K$, suppose $R_{i,j}$ are independent uniform boolean-valued random variables whose values are derived from k.

Using matrix product, we define $digest(k, M) = M \odot R$ where the symbol $\odot$ represents the binary product of the vector M and the matrix R. Instead of deriving a completely random matrix R from the key k, a Toeplitz matrix – where $R_{i,j} = R_{(i+1),(j+1)}$ for all values of i and j, in other words it is constant on any diagonal – can be used to reduce the required number of random bits from $K \times b$ to only $K + b - 1$ without loss of security as suggested in [7, 5].

2 Long authentication string protocols

The analysis of the use of digital-signature MACs in the introduction shows they are closely analogous to the following protocol, devised by Balfanz [1]. In this scheme, A wants to authenticate its information M to B. Here $hash_{160}()$ denotes a 160-bit output cryptographic hash function.

Balfanz non-interactive protocol, [1]			
1.	A	$\longrightarrow_N$	$B : A, M$
2.	A	$\longrightarrow_{WE}$	$B : hash_{160}(A, M)$

The 160-bit hash sent over the weak empirical channel can be delayed and the information M is under the control of the intruder, hence s/he might carry out an off-line attack to find a different M' with the same hash value.[3] That is something which the standard specification of a hash function deems infeasible as it takes about $2^{160/2} = 2^{80}$ hash calculations on average to find such a collision using the birthday paradox.

In order to improve the number of authenticated bits, Pasini and Vaudenay [11] make use of a probabilistic commitment scheme[4] that is at least as secure as a standard cryptographic hash function to commit to the authenticated information. The 80-bit hash of the commitment is then sent over the weak empirical channel. Here the hash function is required to be weakly collision resistant (i.e. the second preimage resistance property: an intruder cannot find a second value v' such that $hash(v) = hash(v')$ for fixed v).

Pasini-Vaudenay non-interactive protocol, [11]			
1.	A	$\longrightarrow_N$	$B : c \parallel d = commit(A, M)$
			B computes $A \parallel M = open(c, d)$
2.	A	$\longrightarrow_{WE}$	$B : hash_{80}(c)$

In [11], Pasini and Vaudenay argue that this provides the same degree of authentication as the Balfanz protocol (namely 2^{80} hash computations) while halving the number of empirical bits thanks to the probabilistic commitment scheme that avoids the possibility of a birthday attack. However, it seems fair to remark that, even 80 bits will seem tedious for most humans to compare carefully in practice.

2.1 Objectives in designing authentication protocols

When designing an authentication protocol, particularly one based on hash functions such as the two above, we typically need (inter alia) to meet the following pair of objectives:

A Combinatorial attacks that involve searching for hash collisions etc are made too difficult to carry out with any reasonable hope of success.

B Whatever guess-work or strategy the attacker can carry out (perhaps involving **A**), his chances of success are sufficiently low.

[3] In the original protocol [1], there is no restriction on the order of sending and receiving Messages 1 and 2.

[4] The commitment scheme used in Pasini-Vaudenay [11] consists of two functions. $c \parallel d = commit(A, M)$ and $A \parallel M = open(c \parallel d)$. A intends to bind a fresh long random nonce R_A and M together without revealing R_A by publishing the commitment c. Eventually sending d (the *decommitment*) reveals R_A, and binds this value firmly to M in the eyes of the receiver. As R_A is a long random nonce the security of the scheme in term of both *binding* and *hiding* is equivalent to a standard cryptographic hash function.

In traditional uses of hashes, these two are inextricably linked, and indeed we would normally characterise the required strength of a hash function as being what is required to overcome both of these simultaneously (and this is the case in both Balfanz and Pasini-Vaudenay).

For example, it is often reasonable (and this is the case with Balfanz) to assume that an attacker can carry out a birthday-style attack, in which the expected number of collisions he can find with N hash calculations in a hash space of size H is $\frac{N \times N}{C \times H}$ for some positive constant C that takes into account both the nature of the search and assumed imperfections in the hash function. It follows that in order to keep the probability of success less than $1/T$, it is necessary (to a close approximation) to make the hash space greater than N^2T/C in size. Notice here that the parameter N comes from **A** above and that T comes from **B**, and the way in which H varies with the two of them is different.

Note that this demonstrates that if we have two uses of a hash function, one of which is vulnerable to birthday attacks and the other only to a plain search (as is achieved in Pasini-Vaudenay), it is not actually true that the two protocols give the same degree of security when a hash function of half length is used for the second, as is perhaps implied by the respective lengths quoted above by Balfanz and Pasini-Vaudenay. The lengths of the hash actually required are $(2\log N + \log T - logC)$ and $(\log N + \log T - logC')$, here C and C' can be the same or different to each other.

3 Short authentication string protocols

Gehrmann, Mitchell, and Nyberg [3] took a different approach to preventing combinatorial search. They use empirical channels to transmit the b-bit output of a check function $m_k()$ together with a b-bit key that has been instrumental in its computation.

MANA I (Gehrmann, Mitchell and Nyberg), [3]		
1a.	A	$\longrightarrow_N B : A, M$
1b.	B	$\longrightarrow_E A :$ 1-bit committed
		A picks a b-bit random number k
2.	A	$\longrightarrow_E B : k, m_k(A \parallel M)$

To eliminate the 1-bit empirical signal in MANA I,[5] Vaudenay proposes to use a strong empirical channel ($\longrightarrow_{SE}$), which achieve stall-free or instant delivery, to send the key and the check-value.[6] Thus $2b$ bits are transmitted in all. In the following description, we will modify the scheme slightly by using a digest function to compute the check-value. The rest of this analysis applies to both versions.

V-MANA I, [14, 11]		
1.	A	$\longrightarrow_N B : A, M$
		A picks a b-bit random number k
2.	A	$\longrightarrow_{SE} B : k, digest(k, A \parallel M)$

The use of the strong empirical channel that provides stall-free transmission leads to a significant

[5] In the original description of MANA I, the pair of parties additionally need to agree on the success of the protocol with the help of human. Since this is irrelevant to our security analysis, we ignore the step in our description of the protocol.

[6] We can replace $\longrightarrow_{SE}$ with a bounded delay empirical one ($\longrightarrow^t_{BE}$) provided B checks that he has received Message 1 before Message 2 could have been sent.

fewer number of authenticated bits transmitted from A to B: these are the first example we have seen of protocols that, given the properties we have assumed of the digest function, come close to preventing the intruder performing any useful combinatorial search. This is because the distribution properties of the digest mean that it is impossible for the intruder to look for a M' that will digest to the same value as M in ignorance of k.

However, the protocol is far from optimal in the human work since any one can modify M blindly in the 1^{st} message transmitted over the insecure normal network, and hope that the digests come out the same in the 2^{nd} message. This will occur with a probability of 2^{-b} irrespective of the value of the key provided that the b-bit digest meets the specification defined in Section 1.1. What this means is that $2b$ empirical bits only guarantee at best a 2^{-b} security level.

Whilst the security proofs presented in [3, 11] are largely correct, what these authors have not discovered is that the bit-length b they choose for the key is too short compared to the digest output and the authenticated information: it is impossible to construct a digest function such that the probability of any one-short attack is no better than 2^{-b}.[7] In fact there is a known theoretical bound on the bit-length of the key [13] that can guarantee the digest meets its specification: $bitlength(k) \geq bitlength(M) - b$.[8]

This result suggests we should aim always to have k noticeably longer than the digest in this style of protocol. Of course to do this without ruining efficiency in human effort, we need to find ways of communicating k over $\longrightarrow_N$ rather than empirically.

4 Improvements to (V-)MANA I

Given two weaknesses discussed in the previous section, we will present improved versions of V-MANA I that optimise the use of the expensive strong empirical channel. These improvements can also apply to MANA I. In other words, human comparison/handling of a b-bit short authentication string (SAS) always corresponds to a probability of 2^{-b} of a successful one-shot attack. Whilst this can only be done at the expense of introducing another (third) message sent over the Dolev-Yao channel we argue that this is not at all a bad trade-off since our highest priority is to minimise the empirical cost.

In contrast to V-MANA I, the key k generated by A in the following protocol can be as long as we want to ensure that the digest function meets the specification in Section 1.1. In addition, we can weaken the assumption that empirical messages' transmission is instantaneous to being of bounded delay as follows.

Improved version of V-MANA I (direct binding) *New*		
1.	$A \longrightarrow_N$	$B : M, hash(k)$
2.	$A \longrightarrow^t_{BE}$	$B : digest(k, M)$
3.	$A \longrightarrow_N$	$B : k$

Note that the message order here and in other improved schemes of V-MANA I is more important than in all preceding protocols: we specify that B will not accept Message 2 within t of receiving Message 1 and that A will not send Message 3 within t of sending Message 2. This is to ensure that

[7] We will give a detailed analysis of the (off-line) computation complexity and its related probability of a successful one-shot attack on this protocol in Appendix A.

[8] We should remark that the bound can be met except for an infinitesimal tolerance for very much smaller lengths than this [9]. However, we suspect that it will be good practice for it to be significantly longer than b.

B was committed to Message 1 when Message 2 was sent, and that Message 3 cannot be received by anyone before B has accepted (if he does) the only Message 2 that A will ever send that relates to it.

Furthermore, we can replace the bounded delay empirical channel and the need to wait by a simple acknowledgement from B to A. The resulting protocol is actually the pairwise version of HCBK protocol [12].

Improved version of MANA I (direct binding) [12, 6, 7]		
$1a.$	$A \longrightarrow_N$	$B : M, hash(k)$
$1b.$	$B \longrightarrow_E$	A : 1-bit commitment
2.	$A \longrightarrow_E$	$B : digest(k, M)$
3.	$A \longrightarrow_N$	$B : k$

We note that this scheme is flexible since the digest and key (Messages 2 and 3) can be released in any order as long as A has received the commitment signal from B in the 1^{st} message. It will often be the case that a bounded delay empirical channel and a one-bit acknowledgement signal are alternatives in this style of protocol design/structure.

Since the SAS in these schemes are functionally dependent on the authentic information M, we term these as the *direct binding* version of Improved (V-)MANA I.

Readers who are interested in the formal security proof as well as variants using indirect binding and Diffie-Hellman can find them in Appendix B.

5 Separation of security concerns

Protocols such as our improved versions of MANA I as well as HCBK [12, 6, 7] only work because it has been possible to separate the two concerns or objectives **A** and **B** as mentioned in Section 2.1. Specifically, these protocols avoid combinatorial attack by pre-committing participants to non-deterministic values such as the keys k, and keep the probability of a one-shot attack working low by choosing a good digest method and a short string of sufficient length.

In these protocols it was *necessary* that we separated these concerns, because it was unreasonable to expect humans to transmit or compare a value as complex as a normal cryptographic hash accurately (or in good temper!). It is interesting to note, however, that it brings a quite unexpected benefit: of the various calculations performed by the participants in the direct binding version of Improved MANA I or HCBK, only the calculation of the short string or digest actually depends on the message M being transmitted. It is reasonable to expect that, because the objective of this calculation is only to meet goal **B** rather than both this and what will almost always be the harder one **A**, it can be done more cheaply as a function of the length of M. A substantial gain is reflected in the complexity model described in Section 1.1.

Since the cost of this calculation is the only one that rises (almost certainly linearly) with the length of M, and all other aspects are constant, we come to the following striking conclusion: *when M is large, protocols based on the computation of a short digest can be more efficient than a traditional message signature scheme or MAC based on a standard cryptographic hash of the whole of M.*

This leads us to propose the following principle:

- **Separation of Security Concerns**: where a single cryptographic primitive is being used to satisfy several different security goals, one should consider if efficiency gains can be made by meeting these goals separately. This particularly applies if the primitive is being applied to a large block of data.

A good illustration of this is the way the objectives of message transmission and authentication are met separately in the useful and popular structure: $A \longrightarrow B : \langle M,\ \mathrm{MAC}(A, M)\rangle$, or $\langle M, sign_A(hash(M))\rangle$. However, for largish M (approximately 10K bytes in our experiments based on SHA-1 and 1024 bit RSA) the time for hashing overtakes the time taken for the signature, and, for much larger messages than this, will dominate.

A particular consequence of the above principle derived from protocols such as our Improved (V-)MANA 1 is the following:

- **Factorisation of cryptographic hashing**: where a cryptographic hash function is being applied to a substantial item of data, analyse whether its security goals can be achieved more cheaply via a combination of a digest function to limit the chances of a one-shot attack, and some constant-time supplementary operations that limit the chances of an attacker to a single try.

5.1 Authentication without hashing

Consider the following protocols as an alternative to the conventional method of authenticating messages with a MAC of the section above.

In the first, A can compute $digest(k, M)$ simultaneously with sending Message 1, but only sends this value to B once the latter has signalled that it is completely committed to the value M by sending a nonce.

Authentication without hashing I (interactive) *New*
1. $A \longrightarrow_N B : M$
2. $B \longrightarrow_N A : N_B$
3. $A \longrightarrow_N B : sign_A(k, digest(k, M), N_B)$

Provided that B has not sent Message 2 until it knows (and is therefore committed to) M, it knows that A has not revealed the hash key k to anyone before that point, as Message 3 depends on N_B. Note that N_B communicated over Dolev-Yao channel is playing the same role of the 1-bit acknowledgement over empirical channel in our Improved MANA I.

In our second protocol, the role of the nonce N_B is replaced by a time stamp ts whose role is to prove that k was not revealed until B was committed to M. A must therefore wait a suitable period between completing Message 1 and sending Message 2.

Authentication without hashing II (non-interactive) *New*
1. $A \longrightarrow_N B : M$
2. $A \longrightarrow_N B : sign_A(k, digest(k, M), ts)$

In this scheme, B cannot accept the protocol run unless receipt of M was complete by time ts, which resembles to the requirement of the bounded delay empirical channel. Notice that this version is suitable for broadcasting a message to many B's simultaneously, but cannot (unlike a traditional digital signature) be used over and over again at different times. This is because the use of the same digest key at different times will allow an intruder to do a combinatorial search for a second M' such that $digest(k, M) = digest(k', M')$, and then deploy this against later recipients of the signature. We therefore will sometimes refer to this family of protocols as *one-time message authentication.*

A further disadvantage of the above protocols is that they do not permit the recipient to begin digesting until after the key k has been received. We believe, however, that they give both parties a significant reduction in processing time over an ordinary cryptographic hash function.

Below we offer one mechanism that overcomes the second difficulty and another one that overcomes both of them, both at extra processing cost.

A can allow B, or B can allow A, the chance to begin digesting immediately by using a confidential mechanism for the agreement of key as can be shown in the two following similar protocols.

Authentication without hashing III (non-interactive) *New*		
$1\alpha.$	$A \longrightarrow_N$	$B : \{k\}_{pk(B)}$
2.	$A \longrightarrow_N$	$B : M$
3.	$A \longrightarrow_N$	$B : sign_A(B, hash(k), digest(k, M))$

Authentication without hashing III (interactive) *New*		
$1\beta.$	$B \longrightarrow_N$	$A : \{k\}_{pk(A)}$
2.	$A \longrightarrow_N$	$B : M$
3.	$A \longrightarrow_N$	$B : sign_A(B, hash(k), digest(k, M))$

Notice that B's name and $hash(k)$ appearing in Message 3 prove to B that Message $1\alpha/\beta$ had k encrypted for B, not any other node. Furthermore, it represents proof to B that the key k is unknown to any one except A and B, we can remove the time stamp as well as timing constraints here. As a result, there is no restriction on the order of sending or receiving these messages: the order above is advantageous because it allows both parties to compute $digest(k, M)$ without delay. Clearly any other way of transmitting secret information from A to B could be used in place of the initial public key encryption.

If there is no need for both parties to digest simultaneously, the three messages can be combined into a single one, and indeed the secret transmission of k can be moved inside the final signed package

Authentication without hashing IV (non-interactive) *New*
$A \longrightarrow_N B \quad : \quad M, sign_A(B, \{k\}_{pk(B)}, digest(k, M))$

5.2 Flexi-MACs

None of the protocols above are relevant to the important practical problem of allowing one user to publish a piece of data together with some form of MAC that any recipient (the expectation being that there will be many of them) can check at any time. In methods I and II the key k is only valid for a short period, whereas in III and IV it is designed only for a single recipient.

We offer the following concept as a partial solution to this problem. It actually requires *more* effort on the part of the sender than the conventional approach, but of course we hope that this will be more than counterbalanced by the large number of recipients who can check it quickly.

All our protocols work by making A choose a key and not allowing the intruder to know the key until B is committed to M and the digest. It does not seem to be possible to achieve this in the circumstances we are now considering, so we turn it around and allow B to choose the key. But of course we are expecting B to be analysing data recorded by A (e.g. on a DVD), not with A herself, so this also sounds impossible. We can, however, simulate it by making A compute a large number of digests of M under different, randomly chosen, keys (set K), which she includes in a single signed block as her "Flexi-MAC" of the message M. B can then select any number of these values that it wishes to at random and check them.

Provided that the verification of each of these individual digests is much easier to compute than a single cryptographic hash, this should still be achievable more quickly than verifying a standard signature. It will also have the advantage that a single signature can be checked to different degrees depending on the perceived security threat and the time/computing power available. The mechanism can be summarised as follows.

- Flexi-MAC(M) $= \{(k, digest(k, M)) | k \in K\}$ concatenated with $sign_A(hash(\{(k, digest(k, M)) | k \in K\}))$
- To verify: select a random set C of the keys represented in K, of a chosen size, and check that the received M digests to the right value for each $k \in C$.

For example, suppose that our "Flexible MAC" consists of 1,000 signed key-and digest combinations, and that we believe that it is inconceivable that an attacker can have found a collision over more than three of these keys simultaneously. Then if the recipient chooses 1,2,3 of the keys at random, it follows that the chances of an attack succeeding are respectively bounded by 0.3%, 0.0006% and $1/(1.6 \times 10^8)$.

The effectiveness of this scheme will depend on how efficient and effective particular digest functions prove to be, and on how much assurance is required by every recipient B separately: we might be quite happy for any given recipient to have a 0.3% of being duped, either because of the application or because different B's share information: if a faked DVD is produced it is likely to be checked many times. We hope that there may be interesting applications in the area of DRM.

Note that by choosing the different keys k used in the Flexi-MAC randomly and after being committed to M, A has gained the same advantage as that of Pasini/Vaudenay relative to Balfanz: birthday attacks are eliminated and so a chosen plain-text will not be an advantage to an attacker in this sense.

In all the protocols we have suggested based on the uses of digest functions not transmitted by humans, there is not the same imperative for them to be *short*. Any reasonable fixed length will suffice. What we still require, of course, is that they are efficient to calculate.

6 Conclusion

In this paper we have analysed the security of two new and related classes of one-way authentication protocols.

We have derived the principle of *separation of security concerns*, that it can be inefficient to use a complex primitive for two difference, and factor-able, purposes. These concerns have an impact on the required length of the SASs manually handled by the human as illustrated in the first family of protocols based on human interaction: Long authentication string: Balfanz and Pasini-Vaudenay; Short authentication string: (V-)MANA I and its family of improved protocols.

The principle also has an impact on computational efficiency gained in our second family of protocol (authentication without hashing) as has been illustrated in Section 5.

The advantage provided by the schemes presented in Section 5.1 is only real if (a) we can substantiate our claim that digest functions can be computed significantly faster than hashes and (b) this advantage is not made irrelevant by issues such as the ratio between communication bandwidth (whether from memory or a peripheral) and processing speed. The second of these issues will vary greatly from application to application.

Both the Toeplitz model presented in Section 1.1 and the results of [9] suggest that the linear model of digest complexity quoted earlier is a little optimistic about how fast one might expect to compute a short one. The former does not satisfy this model since consumption of pseudo-random numbers is independent of the digest width. The latter shows that in order to create a near-perfect digest of any length, an accumulator of length a little larger than the output length must be maintained; this is a more important problem for short as opposed to long outputs. However, both pieces of work suggest that it should be possible to define very good digest functions that do not deviate from our model by very much in the range of output lengths that are likely to concern us. We will report on experiments on computing digest functions, and the comparison with hash functions, in a subsequent paper.

References

[1] D. Balfanz, D. Smetters, P. Stewart, H. Wong. *Talking to strangers: Authentication in Ad Hoc Wireless Networks.* In Symposium on Network and Distributed Systems Security, 2002.

[2] M. Bellare, P. Rogaway. *Entity Authentication and Key Distribution.* CRYPTO 93, LNCS vol. 773, pp. 232-249.

[3] C. Gehrmann, C. Mitchell, K. Nyberg. *Manual Authentication for Wireless Devices.* RSA Cryptobytes, vol. 7, no. 1, pp. 29-37, 2004.

[4] J.-H. Hoepman. *Ephemeral Pairing Problem.* In 8th Int. Conf. Fin. Crypt., LNCS 3110, Springer, pp. 212-226.

[5] H. Krawczyk. *LFSR-based Hashing and Authentication.* CRYPTO 1994, LNCS vol. 839, pp. 129-139.

[6] L.H. Nguyen, A.W. Roscoe. *Efficient group authentication protocol based on human interaction.* Proc of FCS-ARSPA 2006, pp. 9-31.

[7] L.H. Nguyen, A.W. Roscoe. *Authenticating ad hoc networks by comparison of short digests.* Journal of Information and Computation. Vol. 206, Issues 2-4, Feb-Apr 2008, pp. 250-271.

[8] L.H. Nguyen, A.W. Roscoe. Authentication protocols based on low-bandwidth unspoofable channels: a comparative survey. Submitted to Journal of Computer Security.

[9] L.H. Nguyen, A.W. Roscoe. *New theoretical bounds for Universal hash functions.* To appear. See `www.comlab.ox.ac.uk/people/publications/personal/Bill.Roscoe.html`

[10] L.H. Nguyen, A.W. Roscoe. *Separating two roles of hashing in one-way message authentication,* See `www.comlab.ox.ac.uk/people/publications/personal/Bill.Roscoe.html`

[11] S. Pasini, S. Vaudenay. *An Optimal Non-interactive Message Authentication Protocol.* CT-RSA'06, LNCS vol. 3860, pp. 280-294.

[12] A.W. Roscoe. *Human-centred computer security.* Unpublished manuscript, 2005. `www.comlab.ox.ac.uk/people/bill.roscoe/publications/113.pdf`

[13] D.R. Stinson. *Universal Hashing and Authentication Codes.* CRYPTO 1991, LNCS vol. 576, Springer-Verlag, pp. 74-85, 1992.

[14] S. Vaudenay. *Secure Communications over Insecure Channels Based on Short Authenticated Strings.* CRYPTO 2005, LNCS vol. 3621.

A Combinatorial attack on (V-)MANA I

V-MANA I, [14, 11]		
1. A	$\longrightarrow_N$	$B : A, M$
		A picks a b-bit random number k
2. A	$\longrightarrow_{SE}$	$B : k, digest(k, A \parallel M)$

We term b and r the bit-lengths of the digest output and the key k (in this protocol, $b = r = 16$ bits). The intruder first chooses some number c different keys $\{k_1, \cdots, k_c\}$. Based on an off-line brute force search at the cost of $\Theta(2^{bc/2})$ computation steps, related to the birthday paradox, he can expect to find two different M and M' such that for all $k \in \{k_1, \cdots, k_c\}$,[9] we have:

$$digest(k, A \parallel M) = digest(k, A \parallel M')$$

Assuming that the adversary can influence A to send M in the 1^{st} message of the protocol, there is then an attack it can attempt.

- The adversary blocks the message A, M that A sends, checking that it is the particular value that was desired.
- Immediately afterwards (to reduce the chance of A sending the empirical message too soon) it impersonates A to send A, M' to B.

Recall that in the above protocol, the key length r and digest length b are equal. The following calculations where these numbers are kept separate will allow us to draw more general conclusions.

After sending the 1^{st} message, A picks a random key k: with a probability of $\frac{c}{2^r}$, $k \in \{k_1, \cdots, k_c\}$, and the attack is successful. On the other hand, with a probability of $\frac{2^r - c}{2^r} \cdot 2^{-b}$, k is not in this set and so the attack is only successful with a probability of (presumably) 2^{-b}.

Overall, at the cost of $\Theta(2^{cb/2})$ due to the birthday paradox, the chance of a successful one-shot attack is:

$$\mathbf{Pr}_r(c) = c \cdot 2^{-r} + \frac{2^r - c}{2^r} \cdot 2^{-b}$$

When $r = b$ this is significantly larger than the desired probability of 2^{-b}.

The above vulnerability indicates we need to increase the bit-length r of the key to avoid this type of attack. When r increases, 2^r will quickly become significantly bigger than 2^b, this will allow the likelihood of a successful one-shot attack $\mathbf{Pr}_r(c)$ to converge to 2^{-b}. However, this is not feasible in this protocol because the key must be sent with the digest value over the strong empirical channel that is severely limited in bandwidth.

[9] It might be clearer if we define $H_{\{k_1,\cdots,k_c\}}(X) = digest(k_1, X) \parallel \cdots \parallel digest(k_c, X)$, and if *digest* is an ideal digest function, then so is the function H w.r.t its $c \cdot b$ output-bits. As there is no limit on the bit-length of the input X, it normally takes $2^{cb/2}$ computation steps to search for a collision.

B Improved protocols of MANA I and their security analysis

In this Appendix, we will present another two versions of Improved MANA I, which are termed the *indirect binding* and Diffie-Hellman style (or D-H style) protocols.

B.1 Indirect binding and D-H style versions of Improved MANA I

An alternative solution for Improved V-MANA I is to use a commitment scheme to bind $INFO_A$ to a b-bit random nonce R, which is generated by A and released over the bounded empirical channel. This therefore makes use of the *indirect* information binding strategy, as can be seen below.

Improved version of V-MANA I (indirect binding)		
1.	$A \longrightarrow_N$	$B : INFO_A, c$
		$(c, d) = \text{commit}(INFO_A, R)$
2.	$A \longrightarrow^t_{BE}$	$B : R$
3.	$A \longrightarrow_N$	$B : d$
Computational cost: $W(M + W) = 5M + 25$		

The order and time constraints of messages' arrival in this scheme must be the same as in the direct binding version of Improved V-MANA I. However, this protocol is expensive to run because the large $INFO_A$ must be processed by a long output commitment scheme, which is more expensive than a digest function: $W(M+W) = 25+5M$, i.e. an approximate ($W = 5$)-fold increase compared to the direct binding version.

It is interesting to note that this protocol might be regarded as the non-interactive version of the pairwise (indirect binding) protocol of Vaudenay [14].

Similar to the direct binding version of Improved MANA I, we can replace the bounded delay empirical channel with a simple acknowledgement to have the following scheme.

Improved version of MANA I (indirect binding)		
1a.	$A \longrightarrow_N$	$B : INFO_A, c$
		$(c, d) = \text{commit}(INFO_A, R)$
1b.	$B \longrightarrow_E$	A : 1-bit committed signal
2.	$A \longrightarrow_E$	$B : R$
3.	$A \longrightarrow_N$	$B : d$
Computational cost: $W(M + W) = 5M + 25$		

Next we describe another improved scheme, whose main idea is taken root from the pairwise (direct binding) authentication protocol of Hoepman [4].

In the following description, k is a long secret key (160-bit) of A that corresponds to his Diffie-Hellman token g^k he wants to authenticate. In order for the following protocol to be secure, the Diffie-Hellman token g^k must be fresh at each session, unpredictable and kept secret to A when its longhash and b-bit shorthash are revealed in the first two messages.

Improved version of V-MANA I (D-H style)		
1.	$A \longrightarrow_N$	$B : longhash(g^k)$
2.	$A \longrightarrow^t_{BE}$	$B : shorthash(g^k)$
3.	$A \longrightarrow_N$	$B : g^k$
Computational cost: $WM + M = 6M$		

The main difference between this and the direct/indirect binding versions is that there is no $INFO_A$ sent in Message 1 because the Diffie-Hellman token, revealed in Message 3, plays the dual-role of both $INFO_A$ and the long secret key. This results in a cost of order $WM + M = 6M$.

B.2 Security analysis of the Improved (V-)MANA I protocols

We will adapt the Bellare-Rogaway security model where an intruder can control on which node a new protocol instance is launched, and so we are going to define the two kinds of adversaries used our security analysis.

1. **A general adversary** can launch multiple instances of participants (A and B in our protocols). As commonly the case in the literature, the number of times that (s)he can launch an instance of any participant is limited by a finite number, for example $\mathcal{Q}_A$ for A and $\mathcal{Q}_B$ for B. The time complexity of this adversary is bounded by a finite number say T. This is the kind of adversary we want to prove our protocols resist in the security analysis presented here.

2. **A one-shot adversary** is a special case of the general adversary where the number of each participant's instances he can launch is at most once, in other words, $\mathcal{Q}_A = \mathcal{Q}_B = 1$.

We are going to prove that the Improved (V-)MANA I protocols are secure against a one-shot attack in the first step, and then use Theorem 1 stated below to lift the one-shot attack's model to a general attack's model.

The following theorem is the combined result of Lemma 6 of Vaudenay [14] and Theorem 5 of Pasini and Vaudenay [11].

Theorem 1 [11, 14] We consider a general attack such that the number of instances of A (respectively B) is at most $\mathcal{Q}_A$ (respectively $\mathcal{Q}_B$).

If there exists a one-shot attack against the three improved versions of the (V-)MANA I protocol which has success probability p, then a general attack is successful with probability $P \leq p \cdot \mathcal{Q}_A$.

In the following and all subsequent security proofs, we only consider the case when the intruder cannot influence random keys and nonces which are generated by A's instances (possibly launched by the intruder) and which are instrumental in the computation of SASs.[10] Note, we believe that the same assumption has also been made by Vaudenay in his proof of this theorem (i.e. Lemma 6 of [14]).

Proof An instance of A is compatible with an instance of B if B's instance succeeded and received all messages in the right order, where Message 2 is transmitted over the empirical channels from the corresponding A's instance.

The number of possible compatible pairs of instances is upper bounded by $\mathcal{Q}_A\mathcal{Q}_B$, which can be reduced to $\mathcal{Q}_A$ in the Improved (V-)MANA I protocols because

[10] The assumption must be made even though the intruder can launch new instances of any party or device, for otherwise, the intruder could easily fool B into accepting a fake $INFO'_A$ by searching for a digest or short hash collision. Examples are long key k in the direct binding version of Improved (V-)MANA I, and short nonce R and commitment value c in the indirect binding ones.

- In the Improved versions of MANA I, the single SAS (i.e. digest or random nonce) transmitted over empirical channels by definition cannot be mistaken, replayed or delayed from one to another session.
- In the Improved versions of V-MANA I, B can always be offline. As a result, the intruder can simulate all instances of B and picks one who will make the attack succeed.

When an attack is successful, there should exist one compatible pair of instances of A and B which (1) have or compute the same SAS value sent over the empirical channel; and (2) do *not* share the same public data $INFO_A$ that they try to agree on.

Note, the SASs' values of all compatible pairs of instances are uniformly distributed and independent from one another because the SASs are randomised by either random keys (k in direct binding), random nonces (R in indirect binding), or random Diffie-Hellman tokens (g^k in the Diffie-Hellman style version). All of these random elements, which are instrumental in the computation of SASs, are unknown to the intruder at the point when they were generated by A's instances thanks to the above assumption. (This argument remains true even when data $INFO_A$s are controlled by the intruder in the direct binding version, thanks to the use of digest functions).

We know that the probability of a successful attack on each compatible pair of instances is limited to p (i.e. A and B agree on the same digest of different preimage data $INFO_A$s). We therefore have that the general adversary is successful with probability $P \leq p \cdot \mathcal{Q}_A$. ∎

B.2.1 Security analysis of the direct binding improved (V-)MANA I

In the following theorem, the notation (ϵ_c, T_c)-collision-resistant indicates that the success probability of finding a hash collision is upper bounded by ϵ_c in a time T_c. Similarly, (ϵ_i, T_i)-inversion-resistant indicates that the success probability of inverting a hash value is upper bounded by ϵ_i in a T_i.

Theorem 2 Given that *longhash*() is (ϵ_c, T_c)-collision-resistant and (ϵ_i, T_i)-inversion-resistant, a general attack with number of A's (respectively B's) instances bounded by $\mathcal{Q}_A$ (respectively $\mathcal{Q}_B$) is successful against the direct binding versions of Improved (V-)MANA with probability $2^{-b}\mathcal{Q}_A(1 + \epsilon_i + \epsilon_c)$ in a time $Q_A(T_i + T_c)$.

The following proof applies to the direct binding version of Improved V-MANA I, but it can be slightly modified to cope with the direct binding version of Improved MANA I.

Proof We first find the probability of a successful one-shot attack.

A one-shot intruder has no advantage of sending fake $INFO'_A$ and $longhash(k')$ to B (masquerading as A) after the digest is released in Message 2. Therefore, after $INFO_A$ and $longhash(k)$ are sent in Message 1 where k is a private, fresh and long (160-bit) key generated by A in each session and is unknown to any one including the intruder, there are three possibilities that can happen:[11] (1) with probability ϵ_c the intruder can find a hash collision in a time T_c; (2) with probability ϵ_i the intruder can invert the hash value in a time T_i; and (3) with probability $(1 - \epsilon_c - \epsilon_i)$ neither can the intruder find a hash collision nor invert the hash value. Note, there is no need to consider the 2nd-preimage resistance property of a hash function since the intruder does not know key k generated by the honest party A in Message 1.

[11] We assume that given any $INFO_A$ and $longhash(k)$, it is infeasible to gain any advantage in predicting the value of $digest(k, INFO_A)$, i.e. the digest value should be uniformly distributed even in the presence of m and $longhash(k)$.

1. With probability ϵ_c in a time T_c, the adversary can search (off-line) for two distinct keys k' and k'' for which $longhash(k') = longhash(k'')$. The adversary then sends an arbitrarily data $INFO'_A$ ($INFO'_A \neq INFO_A$) and $longhash(k')$ to B (masquerading as A).

Game against the improved V-MANA I (direct binding)– hash collision				
1.	A	$\longrightarrow_N$	$I(B)$	$: INFO_A, longhash(k)$
	$I(A)$	$\longrightarrow_N$	B	$: INFO'_A, longhash(k')$
2.	A	$\longrightarrow_{SE}$	B	$: digest(k, INFO_A)$
3.	A	$\longrightarrow_N$	$I(B)$	$: k$
Winning condition: $digest(k, INFO_A) = digest(k', INFO'_A)$ or $digest(k, INFO_A) = digest(k'', INFO'_A)$				

Prior to sending a key to B in Message 3 the adversary checks to see whether or not $digest(k, INFO_A) = digest(k', INFO'_A)$, and/or $digest(k, INFO_A) = digest(k'', INFO'_A)$. In the first case (which has probability 2^{-b}), the adversary sends k' to B. In the second case (which also has probability 2^{-b}), the adversary sends k'' to B. We conclude that a one-shot attack has probability $2\epsilon_c 2^{-b}$ of success in a time T_c.

2. With probability ϵ_i in a time T_i, the adversary can find a preimage k' such that $longhash(k') = longhash(k)$. The adversary then replaces $INFO_A$ with an arbitrarily data $INFO'_A$ ($INFO'_A \neq INFO_A$) in Message 1.

Game against the improved V-MANA I (direct binding)– hash inversion				
1.	A	$\longrightarrow_N$	$I(B)$	$: INFO_A, longhash(k)$
	$I(A)$	$\longrightarrow_N$	B	$: INFO'_A, longhash(k)$
2.	A	$\longrightarrow_{SE}$	B	$: digest(k, INFO_A)$
3.	A	$\longrightarrow_N$	$I(B)$	$: k$
Winning condition: $digest(k, INFO_A) = digest(k, INFO'_A)$ or $digest(k, INFO_A) = digest(k', INFO'_A)$				

Prior to sending a key to B the adversary checks to see whether or not $digest(k, INFO_A) = digest(k, INFO'_A)$, and/or $digest(k, INFO_A) = digest(k', INFO'_A)$. Similar to the previous case, a one-shot attack has probability $2\epsilon_i 2^{-b}$ of success in a time T_i.

3. On the other hand, with probability $(1 - \epsilon_i - \epsilon_c)$ in a time $(T_i + T_c)$ neither can the adversary search for a hash collision or invert the hash value. Thus the adversary has to select a random pair $(k', INFO'_A)$ where $INFO_A \neq INFO'_A$.

Game against Improved V-MANA I (direct binding) No hash collision and no hash inversion				
1.	A	$\longrightarrow_N$	$I(B)$	$: INFO_A, longhash(k)$
	$I(A)$	$\longrightarrow_N$	B	$: INFO'_A, longhash(k')$
2.	A	$\longrightarrow_E$	B	$: digest(k, INFO_A)$
3.	A	$\longrightarrow_N$	$I(B)$	$: k$
	$I(A)$	$\longrightarrow_N$	B	$: k'$
Winning condition: $INFO_A \neq INFO'_A$ and $digest(k, INFO_A) = digest(k', INFO'_A)$				

Clearly, the probability of success of this case is $(1 - \epsilon_i - \epsilon_c)2^{-b}$ in a time $(T_i + T_c)$ thanks to the digest specification.

We conclude that any one-shot adversary in a time $(T_i + T_c)$ has the following probability of success

$$p \leq 2\epsilon_c 2^{-b} + 2\epsilon_i 2^{-b} + (1 - \epsilon_c - \epsilon_i)2^{-b} = 2^{-b}(1 + \epsilon_c + \epsilon_i)$$

We now can apply Theorem 1 to deduce that any general adversary has probability $2^{-b}\mathcal{Q}_A(1+\epsilon_c+\epsilon_i)$ of success in a time $Q_A(T_i + T_c)$. ∎

B.2.2 Security analysis of the indirect binding improved (V-)MANA I

Theorem 3 Given that a commitment scheme is (ϵ_h, T_h)-hiding and (ϵ_b, T_b)-binding, a general attack with number of A's (respectively B's) instances bounded by $\mathcal{Q}_A$ (respectively $\mathcal{Q}_B$) is successful against the indirect binding versions of Improved (V-)MANA with probability $(\epsilon_h + \epsilon_b)\mathcal{Q}_A$ in a time $Q_A(T_b + T_h)$.

The following proof gives supporting evidence for the security of the indirect binding version of Improved (V-)MANA I.

Proof There are two possibilities that a one-shot attacker can do after receiving $INFO$ and c in Message 1 from A:

- Leaving c unchanged, the intruder sends $INFO'_A$ and c to B (masquerading as A) where $INFO'_A \neq INFO_A$. With probability ϵ_b in a time T_b, the intruder can come up with a d' (which can be either the same as or different from d revealed in Message 3) such that $open(INFO'_A, c, d') = R$ thanks to the binding property of a commitment scheme.
- With probability ϵ_h in a time T_h, the intruder can guess the value of R from $INFO_A$ and c, and then compute (c', d') such that $open(INFO'_A, c', d') = R$ thanks to the hiding property of a commitment scheme.[12]

We can apply Theorem 1 to deduce that any general intruder has a success probability $\mathcal{Q}_A(\epsilon_b + \epsilon_h)$ in a time $Q_A(T_h + T_b)$. ∎

B.2.3 Security analysis of Improved V-MANA I in Diffie-Hellman style

Theorem 4 Given that $longhash()$ is (ϵ_c, T_c)-collision-resistant and (ϵ_i, T_i)-inversion-resistant, a general attack with number of A's (respectively B's) instances bounded by $\mathcal{Q}_A$ (respectively $\mathcal{Q}_B$) is successful against the Improved V-MANA I protocol in Diffie-Hellman (D-H) style with probability $2^{-b}\mathcal{Q}_A(1 + \epsilon_c)$ in a time $Q_A(T_c + T_i)$.

Proof As in the proof of Theorem 2, there are three possibilities which can happen after A releases Message 1:[13]

[12] Since $INFO_A \neq INFO'_A$, it is very unlikely that $c = c'$.

[13] We also assume that given $longhash(g^k)$ it is infeasible for the intruder to gain any advantage in predicting the value of $shorthash(g^k)$.

1. With probability ϵ_c in a time T_c, the adversary can search for two distinct D-H tokens $g^{k'}$ and $g^{k''}$ for which $longhash(g^{k'}) = longhash(g^{k''})$. The adversary then sends $longhash(g^{k'})$ to B (masquerading as A).

Game against the improved V-MANA I (D-H style)– hash collision				
1.	A	$\longrightarrow_N$	$I(B)$	$: longhash(g^k)$
	$I(A)$	$\longrightarrow_N$	B	$: longhash(g^{k'})$
2.	A	$\longrightarrow_{SE}$	B	$: shorthash(g^k)$
3.	A	$\longrightarrow_N$	$I(B)$	$: g^k$
Winning condition: $shorthash(g^k) = shorthash(g^{k'})$ or $shorthash(g^k) = shorthash(g^{k''})$				

   A one-shot attack has probability $2\epsilon_h 2^{-b}$ of success in a time T_c.

2. With probability ϵ_i in a time T_i, the adversary can find a preimage $g^{k'}$ such that $longhash(g^k) = longhash(g^{k'})$. The adversary then replaces g^k with $g^{k'}$ in Message 3 and hopes that they produce the same b-bit hash output. Therefore, the probability of success is $\epsilon_i 2^{-b}$ in a time T_i.

Game against the improved V-MANA I (D-H style)– hash inversion				
1.	A	$\longrightarrow_N$	B	$: longhash(g^k)$
2.	A	$\longrightarrow_{SE}$	B	$: shorthash(g^k)$
3.	A	$\longrightarrow_N$	$I(B)$	$: g^k$
	$I(A)$	$\longrightarrow_N$	B	$: g^{k'}$
Winning condition: $shorthash(g^k) = shorthash(g^{k'})$				

3. On the other hand, with probability $(1 - \epsilon_i - \epsilon_c)$ in a time $T_i + T_c$ neither can the adversary search for a hash collision or invert the hash value. Thus the adversary has to select a random D-H token $g^{k'}$ and send $longhash(g^{k'})$ to B in Message 1 (masquerading as A).

Game against Improved V-MANA I (D-H style) No hash collision and no hash inversion				
1.	A	$\longrightarrow_N$	$I(B)$	$: longhash(g^k)$
	$I(A)$	$\longrightarrow_N$	B	$: longhash(g^{k'})$
2.	A	$\longrightarrow_{SE}$	B	$: shorthash(g^k)$
3.	A	$\longrightarrow_N$	$I(B)$	$: g^k$
	$I(A)$	$\longrightarrow_N$	B	$: g^{k'}$
Winning condition: $shorthash(g^k) = shorthash(g^{k'})$				

   Clearly, the probability of success of this case is $(1 - \epsilon_i - \epsilon_c)2^{-b}$.

We conclude that any one-shot adversary in a time $T_i + T_c$ has the following probability of success

$$p \leq 2\epsilon_c 2^{-b} + \epsilon_i 2^{-b} + (1 - \epsilon_c - \epsilon_i)2^{-b} = 2^{-b}(1 + \epsilon_c)$$

We now can apply Theorem 1 to deduce that any general adversary has a success probability $2^{-b}Q_A(1 + \epsilon_c)$ in a time $Q_A(T_i + T_c)$. ∎