On Security of Universal Hash Function Based Multiple Authentication

Aysajan Abidin

Department of Electrical Engineering, Linköping University, Linköping, Sweden
aysajan@isy.liu.se

Abstract. Universal hash function based multiple authentication was originally proposed by Wegman and Carter in 1981. In this authentication, a series of messages are authenticated by first hashing each message by a fixed (almost) strongly universal$_2$ hash function and then encrypting the hash value with a preshared one-time pad. This authentication is unconditionally secure. In this paper, we show that the unconditional security cannot be guaranteed if the hash function output for the first message is not encrypted, as remarked in [1]. This means that it is not only sufficient, but also necessary, to encrypt the hash of every message to be authenticated in order to have unconditional security. The security loss is demonstrated by a simple existential forgery attack. The impact of the attack is also discussed at the end.

Keywords: ϵ-Almost Strongly Universal hash functions, multiple authentication, unconditionally secure, Quantum Key Distribution.

1 Introduction

Since its first introduction by Wegman and Carter [11] in 1979, Universal hash functions have been extensively studied over the years. They have diverse applications from cryptography to computer science to coding theory. In cryptography, they can be used for, among others, constructing unconditionally secure message authentication codes (MACs). There has been various Universal hash function constructions for authentication by Wegman and Carter, Stinson, and others [3, 5, 6, 8, 9, 12, 14–16, 18, 20–24].

Typical use of Universal hash functions, more accurately ϵ-Almost Strongly Universal$_2$ (ϵ-ASU$_2$) hash functions, in MACs is such that a one-time key (used to identify a hash function in the family) is used to authenticate one message; because two uses of the same key may reveal the key through the message-tag pairs. In this sense, this version of the Wegman-Carter authentication is similar to the one-time pad (OTP). Hence, the key consumption rate of the authentication in this scheme is usually quite high. In this paper we focus on another proposal by Wegman and Carter [24] that uses a fixed ϵ-ASU$_2$ hash function (identified by a fixed key), followed by OTP encryption of the hash function output, so that the hash function can be reused. This scheme is also called counter-based multiple authentication [1] when the OTPs preshared between Alice and Bob can be

T.W. Chim and T.H. Yuen (Eds.): ICICS 2012, LNCS 7618, pp. 303–310, 2012.

identified by counters. The key consumption rate of this scheme asymptotically approaches the tag length.

Contribution. This short paper addresses a simple existential MAC forgery attack when the universal hash function based multiple authentication is used as remarked in [1]. In its original proposal in [24], Wegman and Carter proposed to apply the OTP to the hash of every message that is exchanged. In [1], however, the authors stated that it is not necessary to apply the OTP to the hash of the initial message. As we will see later in Section 3, not using the OTP in the initial round, or in any other round for that matter, will result in the adversary being able to forge the correct tag for his/her chosen message without knowing the authentication key at all. The attack is very simple and straightforward, and also very cheap in terms of computation and storage depending on the properties of the underlying hash function family. But the impact can be deep if such authentication is used in, for example, Quantum Key Distribution (QKD).

2 Background

Definitions. First, some definitions are in order. In what follows, we let $\mathcal{M}$ and $\mathcal{T}$ be finite sets, and $\mathcal{H}$ a class of hash functions from $\mathcal{M} \to \mathcal{T}$.

Definition 1. *A class $\mathcal{H}$ is **Universal$_2$ (U_2)**, if there exists at most $|\mathcal{H}|/|\mathcal{T}|$ hash functions $h \in \mathcal{H}$ such that $h(m_1) = h(m_2)$, for any two distinct $m_1, m_2 \in \mathcal{M}$. If there are at most $\epsilon|\mathcal{H}|$ hash functions instead, the class $\mathcal{H}$ is **ϵ-Almost Universal$_2$ (ϵ-AU_2)**.*

Definition 2. *A class $\mathcal{H}$ is **XOR Universal$_2$ (XU_2)** if there exists at most $|\mathcal{H}|/|\mathcal{T}|$ hash functions $h \in \mathcal{H}$ such that $h(m_1) = h(m_2) \oplus t$, for any two distinct $m_1, m_2 \in \mathcal{M}$ and any $t \in \mathcal{T}$. If there are at most $\epsilon|\mathcal{H}|$ hash functions instead, the class $\mathcal{H}$ is **ϵ-Almost XOR Universal$_2$ (ϵ-AXU_2)**.*

Definition 3. *A class $\mathcal{H}$ is **Strongly Universal$_2$ (SU_2)** if (a) the number of hash functions in $\mathcal{H}$ that takes an arbitrary $m_1 \in \mathcal{M}$ to an arbitrary $t_1 \in \mathcal{T}$ is exactly $|\mathcal{H}|/|\mathcal{T}|$, and (b) the fraction of those functions that also takes an arbitrary $m_2 \neq m_1$ in $\mathcal{M}$ to an arbitrary $t_2 \in \mathcal{T}$ (possibly equal to t_1) is $1/|\mathcal{T}|$. If the fraction in (b) instead is at most ϵ, the class $\mathcal{H}$ is **ϵ-Almost Strongly Universal$_2$ (ϵ-ASU_2)**.*

Here we note that SU_2 is the optimal case, corresponding to $1/|\mathcal{T}|$-ASU_2, since $\epsilon \geq 1/|\mathcal{T}|$ [21]. Also, note that ASU_2 families are AXU_2 and AU_2, and that AXU_2 families are AU_2; however, the reverse is not true.

Definition 4. *A hash function h from $\mathcal{M} \to \mathcal{T}$ is called **XOR-linear** if, for any two m, $m' \in \mathcal{M}$, $h(m \oplus m') = h(m) \oplus h(m')$. Similarly, a family $\mathcal{H}$ is called XOR-linear if any hash function $h \in \mathcal{H}$ is **XOR-linear**.*

Unconditionally Secure MAC. Unconditionally secure authentication theory was first developed by Simmons in [19] and later by Wegman and Carter in [11,24]. Wegman and Carter proposed using the classes of ϵ-ASU$_2$ hash functions for unconditionally secure MAC constructions. The application of ϵ-ASU$_2$ hash functions to construct provably unconditionally secure MACs is straightforward. In these constructions, Alice and Bob share a secret key k to identify a hash function h_k in a family $\mathcal{H}$ of ϵ-ASU$_2$ hash functions from $\mathcal{M} \to \mathcal{T}$. When Alice wants to send a message m to Bob, she computes $t = h_k(m)$ and sends it along with m. Upon receiving m and t, Bob checks the authenticity of m by computing $h_k(m)$ using his share of the key and comparing it with t. If $h_k(m)$ and t are identical, then Bob accepts m as authentic; otherwise, he rejects it. If Eve tries to impersonate Alice and sends m' without knowing the key k, that is, without knowing h_k, the best she can do is to guess the correct tag for m'. The probability of success in this case is $P_1 = 1/|\mathcal{T}|$. If Eve waits and intercepts a message-tag pair (m, t) from Alice and substitutes m with m', then the probability P_2 of guessing the correct tag t' for m' is at most ϵ ($\geq 1/|\mathcal{T}|$). In other words, even seeing a valid message-tag pair does not increase Eve's success probability above ϵ. Therefore, by using a family of ϵ-ASU$_2$ hash functions with suitably chosen ϵ, one can achieve unconditionally secure message authentication. Practical applications require not only ϵ to be small but also the length l of the key k identifying a hash function in ϵ-ASU$_2$ family to be as small as possible.

The most attractive property of unconditionally secure MACs is that the security does not depend on any computational complexity assumptions, as is the case for other MAC schemes like CBC-MAC based on AES or HMAC based on SHA. Also, in terms of speed, Universal hash function based MAC such as UMAC is much faster than its counterparts. Unconditional security, however, comes at a price: the key consumption. This is because the key cannot be reused; repeated use of a key may reveal the whole key through the message-tag pair. For this reason, Wegman and Carter proposed in [24] an efficient and effective way to resolve this by proposing to encrypt the hash function output with an OTP in order to reuse the same key many times. In particular, their proposal is as follows. Alice and Bob share a secret but fixed hash function $h \in \mathcal{H}$ and a series of keys K_i, $i = 1, 2, \cdots$, of length $\log |\mathcal{T}|$ to be used as OTP to encrypt the output of h. Then a series of messages m_i, $i = 1, 2, \cdots$, can be authenticated by using $h(m_i) \oplus K_i$ as the authentication tag. An efficient way to implement this is to use a counter c that is incremented by 1 after each message transmission. In this case, the authentication tag for a message (c, m_c) is computed as

$$t = h(m_c) \oplus K_c, \quad , c = 1, 2, \cdots . \tag{1}$$

This counter-based multiple authentication scheme is provably unconditionally secure. It has also been stated in [1] as a remark that in this scheme the OTP in the initial round can be omitted, since in the authors' own words "it is not necessary". That is, for the first message m_1, $h(m_1)$ can be sent as is. So with this small revision the above scheme becomes as follows: The authentication tag for a message (c, m_c) is now computed as

$$t = \begin{cases} h(m_1), & c = 1, \\ h(m_c) \oplus K_{c-1}, & c = 2, 3, \cdots . \end{cases} \tag{2}$$

We will see in the next section that this new scheme is not secure in general and there may exist a very simple MAC forgery attack in this case.

Related Work. Different variants of the above scheme were proposed after Wegman and Carter's original proposal, such as stateful mode by Shoup [18] and computationally secure version by Brassard [10] and so on. The stateful mode by Shoup [18] is also referred to as Wegman-Carter-Shoup (WCS) authentication. The security bounds for the WCS scheme were improved by [4]. The security of these schemes for various Universal hash function families were studied in Black and Cochran [7] and Handschuh and Preneel [13]. They have demonstrated that for some families of Universal hash functions a single forgery is enough to find another forgery and for many families a few successful forgeries lead to efficient key recovery.

3 The Attack

In this section, we show that the scheme in (2) is not in general secure and present a simple existential forgery attack that exploits the structure of the hash function family used. In particular, if the underlying hash function family is for example XOR-linear, then the attack is straightforward. And there exists (A)SU_2 families of hash functions that are XOR-linear, e.g., the SU_2 family $\mathcal{H}_3$ in [24].

Let us first note that for (1) to be (unconditionally) secure, h (or $\mathcal{H}$) at least needs to be AXU_2 [15]. So, for (2) to be secure, the subset $\mathcal{H}_{m \mapsto t}$ of $\mathcal{H}$ that Eve identifies after seeing the first message-tag pair (m, t) should be AXU_2. We will now see shortly that this requirement does not necessarily be satisfied even when $\mathcal{H}$ is SU_2, the strongest family of all $\mathrm{Universal}_2$ hash function families.

As described in (2), the first message $(1, m_1)$ is sent along with the authentication tag $t_1 = h(m_1)$ from Alice to Bob. Eve intercepts the three-tuple $(1, m_1, t_1)$ and identifies the set $\mathcal{H}_{m_1 \mapsto t_1} := \{f \in \mathcal{H} : f(m_1) = t_1\}$. Note that $|\mathcal{H}_{m_1 \mapsto t_1}| = |\mathcal{H}|/|\mathcal{T}|$ by Definition 3(a). So, at the end of the first round, from Eve's point of view, the (fixed) secret hash function h is taken from $\mathcal{H}_{m_1 \mapsto t_1}$ instead of $\mathcal{H}$. If, for any two distinct $m, m' \in \mathcal{M}$ and any $t \in \mathcal{T}$,

$$|\{f \in \mathcal{H}_{m_1 \mapsto t_1} : f(m) \oplus f(m') = t\}| \leq \epsilon |\mathcal{H}_{m_1 \mapsto t_1}|, \tag{3}$$

then the scheme in described by (2) is secure, since this would mean that $\mathcal{H}_{m_1 \mapsto t_1}$ is ϵ-AXU_2. Here, ϵ is Eve's success probability when attacking the system. The definitions of (A)SU_2 hash functions, however, does not guarantee that (3) holds. In fact, $|\{f \in \mathcal{H}_{m_1 \mapsto t_1} : f(m) \oplus f(m') = t\}|$, for some distinct $m, m' \in \mathcal{M}$ and $t \in \mathcal{T}$, could be as large as $|\mathcal{H}_{m_1 \mapsto t_1}|$. If this is the case, then there is a very simple existential forgery attack that Eve can use to attack the authentication.

In particular, in the second round, Alice sends $(2, m_2, t_2)$, where $t_2 = h(m_2) \oplus K_1$, to Bob. Eve intercepts the three-tuple $(2, m_2, t_2)$ and searches for m_{E} such that $f(m_2) \oplus f(m_{\mathrm{E}}) = t$ is fixed by all $f \in \mathcal{H}_{m_1 \mapsto t_1}$. And then, she sends $(2, m_{\mathrm{E}}, t \oplus t_2)$ to Bob, since

$$h(m_{\mathrm{E}}) \oplus K_1 = h(m_{\mathrm{E}}) \oplus h(m_2) \oplus t_2 = t \oplus t_2. \tag{4}$$

From the above discussion, we naturally arrive at the following theorem about the security of the scheme in (2).

Theorem 1. *Let AUTH be the authentication described in* (2) *where the secret hash function h is chosen from an ASU*$_2$ *family* $\mathcal{H}$*. Then, the success probability of an adversary A attacking AUTH is only upper bounded by the trivial bound 1, that is,*

$$P_{AUTH}^{success}(A) \leq 1. \tag{5}$$

Proof. Suppose that $(1, m_1, t_1)$ with $t_1 = h(m_1)$ is the first message-tag pair and the message number that Alice has sent to Bob. By intercepting the three-tuple, A identifies $\mathcal{H}_{m_1 \mapsto t_1} := \{f \in \mathcal{H} : f(m_1) = t_1\}$. Now, the proof follows directly from the fact, for some distinct $m, m' \in \mathcal{M}$ and $t \in \mathcal{T}$,

$$|\{f \in \mathcal{H}_{m_1 \mapsto t_1} : f(m) \oplus f(m') = t\}| \leq |\mathcal{H}_{m_1 \mapsto t_1}|. \tag{6}$$

It might seem that the computational complexity of the attack is huge at first sight, since identifying the set $\mathcal{H}_{m_1 \mapsto t_1}$ requires an exhaustive search. But, Eve does not need exhaustive search if she knows the structure of the underlying hash function family $\mathcal{H}$. Consider as an example the case when $\mathcal{H}$ is XOR-linear. As mentioned earlier, there are (A)SU$_2$ hash function families that are XOR-linear. In this case, Eve simply observes the first three-tuple $(1, m_1)$ with $t_1 = h(m_1)$ from Alice to Bob, and saves a copy of m_1 and t_1 in her memory. Then in the second round, she intercepts $(2, m_2, t_2)$ with $t_2 = h(m_2) \oplus K_1$, and replaces it with $(2, m_{\mathrm{E}}, t_1 \oplus t_2)$ where $m_{\mathrm{E}} = m_1 \oplus m_2$. Eve now knows that m_{E} will be accepted as authentic, because the hash function h is XOR-linear and then

$$h(m_{\mathrm{E}}) \oplus K_1 = h(m_1 \oplus m_2) \oplus K_1 = h(m_1) \oplus h(m_2) \oplus K_1 = t_1 \oplus t_2. \tag{7}$$

Upon receiving $(2, m_{\mathrm{E}}, t_1 \oplus t_2)$, Bob verifies the authenticity of m_{E} by computing $h(m_{\mathrm{E}}) \oplus K_1$ and comparing it with $t_1 \oplus t_2$. As we have just seen, the correct tag for m_{E} is $t_1 \oplus t_2$. In the subsequent rounds, Eve uses the same strategy to forge the MAC for a new message chosen similarly to m_{E} above. In general, at the i-th round, Eve replaces the three-tuple (i, m_i, t_i) that she intercepted with $(i, m_1 \oplus m_i, t_1 \oplus t_i)$.

Note that this attack is very simple and that Eve does not need to know the actual secret key that is being used. All she needs to do is store the initial message and tag pair from the initial three-tuple. Even if there does not exist $m_{\mathrm{E}} \in \mathcal{M}$ such that $f(m_2) \oplus f(m_{\mathrm{E}}) = t$ is fixed by all $f \in \mathcal{H}_{m_1 \mapsto t_1}$, Eve can choose a message m_{E} for which $f(m_2) \oplus f(m_{\mathrm{E}})$ is fixed by majority of $f \in \mathcal{H}_{m_1 \mapsto t_1}$ and still have a high probability of success. It all depends on the structure and properties

of the underlying hash function family used in the authentication. Therefore, we stress that when the counter-based multiple authentication scheme is used it is very important to encrypt the hash function output of every message that is to be exchanged. And Wegman and Carter were right to propose to encrypt the hash of every message. After all, since both (1) and (2) require asymptotically the same amount of secret key, one does not sacrifice much by masking the hash of every message, should this authentication be used.

4 Impact

We now discuss the impact of the existence of the straightforward attack presented in the previous section in the context of Quantum Key Distribution (QKD). First, let us briefly recall what QKD is and why authentication is needed in QKD.

QKD, first proposed by Bennett and Brassard in 1984 [2], is a provably secure (or universally composably secure) key agreement technique that consist of two parts: quantum transmission over a quantum channel and classical postprocessing over a classical public channel. In QKD, the legitimate users first exchange quantum signals over the quantum channel to generate a raw key. Then, they agree on a shared secret key from the raw key by performing a joint postprocessing by communicating on public channel. QKD is proven to be unconditionally secure, provided that the public channel is immutable; see, for example, [17]. If the public channel is not not authentic, QKD is, like any other key agreement protocol, susceptible to a man-in-the-middle attack. Therefore, authentic public communication channel is a must. Moreover, to guarantee unconditional security of QKD an unconditionally secure authentication is needed.

The standard choice for authentication in QKD is the Wegman-Carter type of authentication, based on ϵ-ASU_2 hashing. To kick-start the authentication, the legitimate parties use preshared secret key. In the first round the users use the pre-shared key, which is long enough to authenticate the messages exchanged in this round. In the following rounds, a part of the key generated in the previous rounds is used for subsequent authentication. Therefore, the key-consumption rate of the authentication directly affects the key output rate of QKD, and so one needs an authentication with less key-consumption rate. Moreover, in QKD no limit is put on Eve's computational power and memory.

When the authentication in (2) is used in QKD, only h is preshared by Alice and Bob. The OTP key in the second round is a portion of the QKD generated key in the first round, and the OTP key in the third round is a portion of the QKD generation in the second round, and so on. So, the OTP keys are not, and need not be, preshared by Alice and Bob. Now in the first round, Eve identifies $\mathcal{H}_{m_1 \mapsto t_1}$ and searches for m_{E} such that $f(m_2) \oplus f(m_{\mathrm{E}}) = t$ is fixed by all or most of $f \in \mathcal{H}_{m_1 \mapsto t_1}$. If the attack is successful, then Eve breaks the QKD in this round and as a consequence learns the QKD generated key, and thus the OTP key K_2 used in the next round. We stress here that the success probability is in general quite high. So in the next round Eve will know $h(m_3) = t_3 \oplus K_2$,

and together with the knowledge of $h(m_1)$ she will be able to find h. Therefore, the consequence of not masking the hash value of the first message can be serious, at least, in QKD.

5 Solution

As we have seen in Section 3, masking the hash function output of every message with an OTP is both necessary and sufficient for unconditional security of the authentication scheme under review. One might, however, wonder whether there are other solutions than to encrypt the hash of the first message in scheme (2). We answer this question in the negative if one aims for unconditional security, unless one uses another unconditionally secure encryption than OTP. Since the attack exploits the fact that the hash value is known for the first message message, masking the hash value of the first message, or any other message for that matter, is necessary.

6 Conclusion

We have reviewed the universal hash function based multiple authentication. We pointed out that masking the hash value of every message is not only sufficient but also necessary to guarantee security. Furthermore, we presented an existential forgery attack. The attack is straightforward and exploits the property of the underlying hash functions. The impact of the attack is also discussed in the context of QKD.

Acknowledgements. The author would like to thank the anonymous reviewers for their valuable comments, and also Jan-Åke Larsson for his useful comments.

References

1. Atici, M., Stinson, D.R.: Universal Hashing and Multiple Authentication. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 16–30. Springer, Heidelberg (1996)
2. Bennett, C.H., Brassard, G.: Quantum cryptography: Public key distribution and coin tossing. In: Proc. IEEE Int. Conf. Comput. Syst. Signal Process, Bangalore, India, pp. 175–179 (1984)
3. Bernstein, D.J.: The Poly1305-AES Message-Authentication Code. In: Gilbert, H., Handschuh, H. (eds.) FSE 2005. LNCS, vol. 3557, pp. 32–49. Springer, Heidelberg (2005)
4. Bernstein, D.J.: Stronger Security Bounds for Wegman-Carter-Shoup Authenticators. In: Cramer, R. (ed.) EUROCRYPT 2005. LNCS, vol. 3494, pp. 164–180. Springer, Heidelberg (2005)
5. Bierbrauer, J., Johansson, T., Kabatianskii, G., Smeets, B.: On Families of Hash Functions via Geometric Codes and Concatenation. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 331–342. Springer, Heidelberg (1994)

6. Black, J.: Message authentication codes. Ph.D. thesis, University of California Davis, USA (2000)
7. Black, J., Cochran, M.: MAC Reforgeability. In: Dunkelman, O. (ed.) FSE 2009. LNCS, vol. 5665, pp. 345–362. Springer, Heidelberg (2009), http://dx.doi.org/10.1007/978-3-642-03317-9_21
8. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: UMAC: Fast and Secure Message Authentication. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 216–233. Springer, Heidelberg (1999)
9. den Boer, B.: A simple and key-economical unconditional authentication scheme. J. Comp. Sec. 2, 65–72 (1993)
10. Brassard, G.: On computationally secure authentication tags requiring short secret shared keys. In: Chaum, D., Rivest, R.L., Sherman, A.T. (eds.) CRYPTO, pp. 79–86. Plenum Press, New York (1982)
11. Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. 18, 143–154 (1979)
12. Halevi, S., Krawczyk, H.: MMH: Software Message Authentication in the Gbit/Second Rates. In: Biham, E. (ed.) FSE 1997. LNCS, vol. 1267, pp. 172–189. Springer, Heidelberg (1997)
13. Handschuh, H., Preneel, B.: Key-Recovery Attacks on Universal Hash Function Based MAC Algorithms. In: Wagner, D. (ed.) CRYPTO 2008. LNCS, vol. 5157, pp. 144–161. Springer, Heidelberg (2008), http://dx.doi.org/10.1007/978-3-540-85174-5_9
14. Krawczyk, H.: LFSR-Based Hashing and Authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
15. Krawczyk, H.: New Hash Functions for Message Authentication. In: Guillou, L.C., Quisquater, J.-J. (eds.) EUROCRYPT 1995. LNCS, vol. 921, pp. 301–310. Springer, Heidelberg (1995)
16. Rogaway, P.: Bucket Hashing and Its Application to Fast Message Authentication. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 29–42. Springer, Heidelberg (1995)
17. Shor, P.W., Preskill, J.: Simple proof of security of the bb84 quantum key distribution protocol. Phys. Rev. Lett. 85, 441–444 (2000)
18. Shoup, V.: On Fast and Provably Secure Message Authentication Based on Universal Hashing. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 313–328. Springer, Heidelberg (1996)
19. Simmons, G.J.: A survey of information authentication. Proceedings of the IEEE 76(5), 603 (1988)
20. Stinson, D.R.: Universal Hashing and Authentication Codes. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 74–85. Springer, Heidelberg (1992)
21. Stinson, D.R.: Combinatorial techniques for universal hashing. J. Comput. Syst. Sci. 48, 337–346 (1994)
22. Stinson, D.R.: On the connections between universal hashing, combinatorial designs and error-correcting codes. Congressus Numerantium 114, 7–27 (1996)
23. Stinson, D.R.: Universal hash families and the leftover hash lemma, and applications to cryptography and computing. J. Combin. Math. Combin. Comput. 42, 3–31 (2002)
24. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. 22, 265–279 (1981)