New Universal Hash Functions

Aysajan Abidin and Jan-Åke Larsson

Department of Electrical Engineering,
Linköping University, SE-581 83 Linköping, Sweden
aysajan@isy.liu.se, jan-ake.larsson@liu.se

Abstract. Universal hash functions are important building blocks for unconditionally secure message authentication codes. In this paper, we present a new construction of a class of ϵ-Almost Strongly Universal$_2$ hash functions with much smaller description (or key) length than the Wegman-Carter construction. Unlike some other constructions, our new construction has a very short key length and a security parameter ϵ that is independent of the message length, which makes it suitable for authentication in practical applications such as Quantum Cryptography.

Keywords: Universal hash functions, ϵ-Almost Strongly Universal hash functions, authentication, Quantum Cryptography.

1 Introduction

Universal hash functions were first introduced by Wegman and Carter [7] in 1979, and since then they have been extensively studied. They are used in diverse cryptographic tasks such as unconditionally secure authentication, error-correction and randomness extraction (or privacy amplification, within Quantum Cryptography). Over the years, various Universal hash function families are constructed by Wegman and Carter, Stinson, and others [3, 4, 6, 13, 14, 17, 20–24]. The important properties are the description length (key consumption), the security parameter, and the computational efficiency, more on this below.

This paper addresses a new construction of Universal hash functions. In particular, we present a new construction of ϵ-Almost Strongly Universal$_2$ (ϵ-ASU$_2$) hash functions, that not only have small description length but also a security parameter ϵ that is independent of the message length. The construction combines the LFSR-based hashing proposed by Krawczyk in [13] with the composition theorem by Stinson in [20] for constructing Universal hash functions. Given its properties, the new construction is also computationally efficient.

1.1 Universal Hash Function Families

First, let us recall the definitions of Universal and ϵ-ASU$_2$ hash functions and the composition theorem for Universal hash functions.

Definition 1 (Universal$_2$ hash functions). *Let $\mathcal{M}$ and $\mathcal{T}$ be finite sets. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is* ***Universal$_2$ (U$_2$)*** *if there exist at*

F. Armknecht and S. Lucks (Eds.): WEWoRC 2011, LNCS 7242, pp. 99–108, 2012.

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 (XOR Universal$_2$ hash functions). *Let $\mathcal{M}$ and $\mathcal{T}$ be as before. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ 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 (Strongly Universal$_2$ hash functions). *Let $\mathcal{M}$ and $\mathcal{T}$ be as before. A class $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$ is **Strongly Universal$_2$ (SU$_2$)** if the following two conditions are satisfied:*

(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 $|\mathcal{H}|/|\mathcal{T}|$.*

(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$_2$)**.*

Note that $\epsilon \geq 1/|\mathcal{T}|$ [21] so that SU$_2$ hash functions are the optimal case, corresponding to $1/|\mathcal{T}|$-ASU$_2$ hash functions.

There are several ways to construct classes of ϵ-ASU$_2$ hash functions, and in this paper we will use the following theorem from [20].

Theorem 1 (Composition). *Let $\mathcal{F}$ be a set of ϵ_1-AU$_2$ hash functions from $\mathcal{M} \to \mathcal{Z}$, and let $\mathcal{G}$ be a set of ϵ_2-ASU$_2$ hash functions from $\mathcal{Z} \to \mathcal{T}$. Then, $\mathcal{H} = \mathcal{G} \circ \mathcal{F}$ is an ϵ-ASU$_2$ hash function family from $\mathcal{M} \to \mathcal{T}$ with $\epsilon = \epsilon_1 + \epsilon_2$.*

We will also use ideas from [13,14], in which an ϵ-AXU$_2$ family is composed with a one-time pad, resulting in an ϵ-ASU$_2$ family (note that the above theorem does not apply). The resulting family has a security parameter ϵ that depends on the message length. In this paper, we will use a different approach that enables use of the theorem, and keeps $|\mathcal{H}|$ small while giving a security parameter ϵ that only depends on the tag length, not the message length.

1.2 Information-Theoretically Secure Authentication

The class of ϵ-ASU$_2$ hash functions can straightforwardly be applied for information-theoretically secure message authentication. In this scenario, two legitimate users (Alice and Bob) share a secret key k long enough to identify a hash function h_k in a family of ϵ-ASU$_2$ hash functions. 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.

Now, if an adversary tries to impersonate Alice and sends m' without knowing the key k, or h_k, the best he/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 the adversary 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' increases somewhat but is bounded by ϵ ($\geq 1/|\mathcal{T}|$). In other words, even seeing a valid message-tag pair does not increase the adversary's success probability above ϵ. Therefore, by using a family of ϵ-ASU$_2$ hash functions with suitably chosen ϵ, one can achieve information-theoretically secure message authentication.

In addition to requiring ϵ to be small, practical applications require also the length l of the key k identifying a hash function in the family of ϵ-ASU$_2$ hash functions to be as small as possible. This latter requirement is especially important in Quantum Cryptography (QC).

1.3 Application to Authentication in Quantum Cryptography

Quantum Cryptography (QC), also known as Quantum Key Distribution (QKD), is a key agreement technique based on the laws of quantum mechanics. The users first exchange quantum signals over a so-called quantum channel to generate a raw key by measuring the quantum signals. Then, they extract a common secret key from the raw key by performing a joint post-processing by communicating on an immutable public channel. The first QKD protocol known as BB84 was proposed by Bennett and Brassard in 1984 [2].

QKD is proven to be information-theoretically secure, provided that the public channel is immutable; see, for example, [19]. In the case that the public channel is not immutable (not authentic), QKD can easily be broken by a man-in-the-middle attack. Therefore, ensuring the authenticity of the public channel is a must. More specifically, the adversary must not be able to insert or modify the classical messages exchanged over the public channel between the legitimate users during the post-processing phase of the QKD protocol. Also, to guarantee information-theoretic security of QKD the authentication used must be information-theoretically secure.

This is achieved via ϵ-ASU$_2$ hashing, and thus needs shared secret key. In the first round the users must use pre-shared secret key, which should be long enough to authenticate the classical messages in the round. In the following rounds, key generated in the previous rounds must be used. Hence, the key-consumption rate of the authentication protocol used in QKD directly influences the key output rate. Because of this, one needs an authentication with small key-consumption rate. Furthermore, in QKD very long messages need to be authenticated, so it is desirable to have a scheme where it is simple to do this, without changing parameters of the communication protocol. Thus, there is a need for a hash function family that is small but still has a security parameter ϵ that only depends on the tag length, not the message length.

1.4 Lower Bounds

There are lower bounds on the description length (or key length) for ϵ-ASU$_2$ hash functions derived by Stinson [20], Kabatiankii *et al.* [12], Gemmel and Naor [9], and Nguyen and Roscoe [16]. In [16], the authors provided new combinatorial bounds that are tighter than the other bounds for the key length. They also identified a value for ϵ that represents a threshold in the behaviour of the various bounds and classified different lower bounds in relation to the threshold value of ϵ. Here, we only recall the lower bound by Stinson in [20]. Interested readers may refer to the above references for details of the other bounds.

Theorem 2 (Lower bound for ϵ-ASU$_2$ hash function families [20]). *If there exists an ϵ-ASU$_2$ family $\mathcal{H}$ of hash functions from $\mathcal{M}$ to $\mathcal{T}$, then*

$$|\mathcal{H}| \geq \frac{|\mathcal{M}|(|\mathcal{T}|-1)^2}{|\mathcal{T}|\epsilon(|\mathcal{M}|-1)+|\mathcal{T}|-|\mathcal{M}|}+1. \tag{1}$$

The proof can be found in [20]. In the SU$_2$ case, this simplifies to

$$|\mathcal{H}| \geq |\mathcal{M}|(|\mathcal{T}|-1)+1. \tag{2}$$

Otherwise, if $|\mathcal{M}| \gg |\mathcal{T}|$ the bound simplifies to (in terms of key length)

$$\log|\mathcal{H}| \geq 2\log(|\mathcal{T}|-1)-\log(\epsilon|\mathcal{T}|-1)+1. \tag{3}$$

Here and below "log" denotes the base 2 logarithm. If in addition $\epsilon = c/|\mathcal{T}|$ for some constant c and $|\mathcal{T}|$ is large, the right-hand side is close to $2\log|\mathcal{T}|$. If one allows ϵ to increase when $|\mathcal{M}|$ increases, the bounds decrease which makes it easier to approach $2\log|\mathcal{T}|$, as we shall see below.

1.5 Comparison of Some Existing Constructions

Now, let us briefly compare the key length and security parameter ϵ of a few constructions of ϵ-ASU$_2$ hash function families. In Table 1, the value of ϵ and the key length for five different constructions are listed for comparison. One can find a more detailed overview of various constructions of ϵ-ASU$_2$ hash functions by different authors in Refs. [1, 17].

As can be seen from the table, the constructions by Wegman-Carter and Bierbrauer *et al.* have $\epsilon = 2/|\mathcal{T}|$ while the others have values of ϵ that depend on the message length either logarithmically (Stinson) or linearly (den Boer and Krawczyk). In terms of key length, den Boer's construction is the best followed by Krawczyk, having key lengths $2\log|\mathcal{T}|$ and $3\log|\mathcal{T}|+1$, respectively, and both are determined only by the tag length. The next good scheme in terms of key length is the construction by Bierbrauer *et al.,* for which the key length $\approx 3\log|\mathcal{T}|+2\log\log|\mathcal{M}|$. The key length for the constructions by Stinson and Wegman-Carter are logarithmic in the message length, but the construction by Stinson consumes approximately a quarter of the key that is needed for the

Table 1. The key length and ϵ for different constructions. The key length for Bierbrauer *et al* is approximate because of the need to invert se^s in the construction. This involves the Lambert W function (see, e.g., [8]), whose asymptotics for large s gives the expression below.

Construction	ϵ	Key length
Wegman-Carter [24]	$2/\lvert\mathcal{T}\rvert$	$4(\log\lvert\mathcal{T}\rvert + \log\log\log\lvert\mathcal{M}\rvert)\log\log\lvert\mathcal{M}\rvert$
Stinson [20]	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 1)/\lvert\mathcal{T}\rvert$	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 2)\log\lvert\mathcal{T}\rvert$
den Boer [6]	$(\log\lvert\mathcal{M}\rvert/\log\lvert\mathcal{T}\rvert)/\lvert\mathcal{T}\rvert$	$2\log\lvert\mathcal{T}\rvert$
Bierbrauer *et al.* [3]	$2/\lvert\mathcal{T}\rvert$	$\approx 3\log\lvert\mathcal{T}\rvert + 2\log\log\lvert\mathcal{M}\rvert$
Krawczyk [13]	$(1 + 2\log\lvert\mathcal{M}\rvert)/\lvert\mathcal{T}\rvert$	$3\log\lvert\mathcal{T}\rvert + 1$

Wegman-Carter. As mentioned earlier, we aim for a construction with small key length and ϵ independent of message size.

There are also other constructions such as Bucket hashing by Rogaway [18], MMH (Multilinear Modular Hashing) by Halevi and Krawczyk [10], and UMAC by Black *et al.* [5]. All three are very fast but have some properties that make them undesirable from the point of view of this paper. Bucket hashing has a very long key and output and is only ϵ-AU$_2$ and so the hash output has to be further mapped by an (A)SU$_2$ hash function to make it ϵ-ASU$_2$. Another paper [11] proposes a bucket hashing scheme with small key, but this does not have fixed ϵ and still has a comparatively long output. MMH [10] and UMAC [5] are not economical in terms of key length; the key lengths are very large in comparison to the above schemes.

1.6 Our Contribution

In this paper, we use LFSR-based hashing [13] and compose with an SU$_2$ hash function family. This enables the composition theorem, and gives a new ϵ-ASU$_2$ hash function family. One particular choice of parameters in the construction gives $\epsilon = 2/|\mathcal{T}|$ just as in Wegman-Carter's initial construction, while retaining a small description length. This construction is suitable for use in authentication, especially in Quantum Cryptography because of its low key-consumption property and the small fixed ϵ. Also, the new construction is also computationally efficient because the LFSR can efficiently be implemented in both software and hardware; the subsequent SU$_2$ hash function family operates on a much shorter intermediate bitstring, and is therefore also comparatively efficient.

2 The New Construction

In this section, we present our new construction of an ϵ-ASU$_2$ hash function family. To do this we need the first step in the construction by Krawczyk, the LFSR-based hashing [13].

2.1 LFSR-Based Hashing

In [13], Krawczyk presented an elegant way of constructing ϵ-AU$_2$ hash functions. The basic idea is to use an LFSR with a short key, a secret initial string and a secret feedback polynomial, to generate a longer key that selects a hash function in an ϵ-AU$_2$ hash function family. This can be viewed as selecting a certain subset of the linear maps from binary vectors m in $\mathcal{M}$ to binary vectors t in $\mathcal{T}$.

The full set of linear maps from $\mathcal{M}$ to $\mathcal{T}$ was found to be an SU$_2$ hash function family already by Wegman and Carter in [7], there denoted $\mathcal{H}_3$. In matrix language, $\mathcal{H}_3$ consists of $\log|\mathcal{T}| \times \log|\mathcal{M}|$ binary matrices, so that the description length of the hash functions in $\mathcal{H}_3$ is $(\log|\mathcal{M}|)(\log|\mathcal{T}|)$, which makes it impractical. However, if the matrices are restricted to be Toeplitz matrices (constant on diagonals), then the corresponding set of hash functions is still Universal$_2$, see [15]. The description length of the hash functions is now reduced to $\log|\mathcal{M}| + \log|\mathcal{T}| - 1$, since a Toeplitz matrix can be uniquely identified by the first column and the first row of the matrix.

With a further restriction on the Toeplitz matrix, it is possible to obtain an ϵ_1-AXU$_2$ hash function family with a much smaller description length. In particular, if the consecutive columns of the Toeplitz matrix are restricted to be the consecutive states of an LFSR of length $\log|\mathcal{T}|$, then the hash functions with these matrices form an ϵ_1-AXU$_2$ hash function family with $\epsilon_1 = (2\log|\mathcal{M}|)/|\mathcal{T}|$. The description length of the hash functions in this family is $2\log|\mathcal{T}| + 1$, which is the sum of the length of the initial state and the feedback polynomial; see [13] for details.

Krawczyk's construction continues with a composition with a one-time pad. In the next section, we will take a different route and not use the XOR property of the family, but only the ϵ-AU$_2$ property. In Krawczyk's construction, as mentioned in the introduction, the composition of an ϵ_1-AXU$_2$ family with a one-time pad (of length $|\mathcal{T}|$) is an ϵ-ASU$_2$ family with $\epsilon = \epsilon_1 + 1/|\mathcal{T}|$ [13]. The one-time pad has length $\log|\mathcal{T}|$. Therefore, the construction by Krawczyk has $\epsilon = (1 + 2\log|\mathcal{M}|)/|\mathcal{T}|$ and the key length $3\log|\mathcal{T}| + 1$, which is the sum of the length of the key for LFSR-based hash function and of the one-time pad. We note that the feedback polynomials used in the LFSR-based hashing are irreducible, so that the actual key length is slightly less than $3\log|\mathcal{T}| + 1$, see [13,14] for details on usage and key length.

2.2 LFSR-Based Hashing Followed by an SU$_2$ Hash Function Family

Our goal is to construct an ϵ-ASU$_2$ hash function family from $\mathcal{M}$ to $\mathcal{T}$ with $\epsilon = 2/|\mathcal{T}|$ and with a small key length. To this end, we use LFSR-based hashing and the composition theorem. Recall that the composition theorem states that if $h = g \circ f$ is the composition of an ϵ_1-AU$_2$ hash function f from $\mathcal{M} \to \mathcal{Z}$ with an SU$_2$ hash function g from $\mathcal{Z} \to \mathcal{T}$, then h is ϵ-ASU$_2$ with $\epsilon = \epsilon_1 + 1/|\mathcal{T}|$ from $\mathcal{M} \to \mathcal{T}$. Also, if f is an LFSR-based hash function from $\mathcal{M} \to \mathcal{Z}$, then f is an ϵ_1-AU$_2$ with $\epsilon_1 = (2\log|\mathcal{M}|)/|\mathcal{Z}|$. Therefore, to make

$$\frac{2\log|\mathcal{M}|}{|\mathcal{Z}|}+\frac{1}{|\mathcal{T}|}=\frac{2}{|\mathcal{T}|}, \tag{4}$$

we need to have

$$\frac{2\log|\mathcal{M}|}{|\mathcal{Z}|}=\frac{1}{|\mathcal{T}|}, \tag{5}$$

which gives us

$$|\mathcal{Z}|=2|\mathcal{T}|\log|\mathcal{M}|. \tag{6}$$

This gives the following construction: let $\mathcal{F}$ be a set of LFSR-based hash functions from $\mathcal{M}\to\mathcal{Z}$, where $\mathcal{Z}$ is an intermediate set of bit strings of length $\log|\mathcal{T}|+\log\log|\mathcal{M}|+1$. From eqn. (5), we see that $\mathcal{F}$ is an ϵ-AU$_2$ hash function family with $\epsilon=(2\log|\mathcal{M}|)/|\mathcal{Z}|=1/|\mathcal{T}|$. Let $\mathcal{G}$ be a set of SU$_2$ hash functions from $\mathcal{Z}\to\mathcal{T}$, and $\mathcal{H}=\mathcal{G}\circ\mathcal{F}$. Then, by the composition theorem, it follows that $\mathcal{H}$ is a family of ϵ-ASU$_2$ hash functions from $\mathcal{M}\to\mathcal{T}$ with

$$\epsilon=2/|\mathcal{T}|. \tag{7}$$

As before, the family $\mathcal{F}$ of LFSR-based hash functions from $\mathcal{M}\to\mathcal{Z}$ has description length $l_{\mathcal{F}}=2\log|\mathcal{Z}|+1$. And since $\mathcal{Z}$ is a set of strings of length $\log|\mathcal{T}|+\log\log|\mathcal{M}|+1$ we obtain $l_{\mathcal{F}}=2\log|\mathcal{T}|+2\log\log|\mathcal{M}|+3$.

For the SU$_2$ family of hash functions $\mathcal{G}$, the shortest possible description length is slightly smaller than $\log|\mathcal{Z}|+\log|\mathcal{T}|$ because of the bound (2). The construction in Lemma 10 of Bierbrauer *et al.* [3] almost reaches this with a key length of exactly $\log|\mathcal{Z}|+\log|\mathcal{T}|$, which gives a description length of the SU$_2$ hash functions in $\mathcal{G}$ of $l_{\mathcal{G}}=2\log|\mathcal{T}|+\log\log|\mathcal{M}|+1$. Explicitly, let π be a linear map from $\mathcal{Z}\to\mathcal{T}$. Then, the family $\mathcal{G}$ of hash functions $g:\mathcal{Z}\to\mathcal{T}$ defined as $g_{z,t}(r)=\pi(zr)+t$, where $z,r\in\mathcal{Z}$ and $t\in\mathcal{T}$, is SU$_2$. This family works well as $\mathcal{G}$, but note that any SU$_2$ family with key length equal to the message (intermediate bit string) length plus the tag length would give the same total key length

$$l_{\mathcal{H}}=l_{\mathcal{F}}+l_{\mathcal{G}}=4\log|\mathcal{T}|+3\log\log|\mathcal{M}|+4. \tag{8}$$

3 Comparison with Existing Constructions

Let us now compare the above construction with existing constructions in terms of the key length, security parameter, and performance. Table 2 lists the relevant data in terms of the key length and the security parameter ϵ.

As can be seen from the table, the new construction like the constructions by Wegman-Carter [24] and Bierbrauer *et al.* [3] has a fixed $\epsilon=2/|\mathcal{T}|$, while the others have ϵ dependent logarithmically or linearly on the message length $\log|\mathcal{M}|$. In terms of the key length, our construction clearly consumes much less key than the constructions by Wegman-Carter [24] and Stinson [20], but not as little as the constructions by den Boer, Krawczyk, and Bierbrauer. The construction by den Boer has the lowest key length $2\log|\mathcal{T}|$ at the cost of an increase in ϵ.

Table 2. The key length and ϵ for the new and the existing constructions. Approximations as before.

Construction	ϵ	Key length
Wegman-Carter [24]	$2/\lvert\mathcal{T}\rvert$	$4(\log\lvert\mathcal{T}\rvert + \log\log\log\lvert\mathcal{M}\rvert)\log\log\lvert\mathcal{M}\rvert$
Stinson [20]	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 1)/\lvert\mathcal{T}\rvert$	$(\log\log\lvert\mathcal{M}\rvert - \log\log\lvert\mathcal{T}\rvert + 2)\log\lvert\mathcal{T}\rvert$
den Boer [6]	$(\log\lvert\mathcal{M}\rvert/\log\lvert\mathcal{T}\rvert)/\lvert\mathcal{T}\rvert$	$2\log\lvert\mathcal{T}\rvert$
Bierbrauer *et al.* [3]	$2/\lvert\mathcal{T}\rvert$	$\approx 3\log\lvert\mathcal{T}\rvert + 2\log\log\lvert\mathcal{M}\rvert$
Krawczyk [13]	$(1 + 2\log\lvert\mathcal{M}\rvert)/\lvert\mathcal{T}\rvert$	$3\log\lvert\mathcal{T}\rvert + 1$
This construction	$2/\lvert\mathcal{T}\rvert$	$4\log\lvert\mathcal{T}\rvert + 3\log\log\lvert\mathcal{M}\rvert + 4$

Another way to compare the schemes is to fix the security parameter ϵ, and from that and the message length $\log|\mathcal{M}|$ determine tag length $\log|\mathcal{T}|$ and key length. This is done in Table 3, but only for the four last alternatives of Table 2. As we can see from the table, the tag length does not depend on $\log|\mathcal{M}|$ for Bierbrauer *et al.* and the present scheme, while it increases when the message size increases for den Boer [6] and Krawczyk [13]. In terms of key length dependence on $\log|\mathcal{M}|$, the constructions by den Boer [6] and Bierbrauer *et al.* are somewhat better than Krawczyk [13] and the current constructions.

Table 3. The key length and tag length, given ϵ and $|\mathcal{M}|$. Here, also the entries for den Boer are approximate; an approximation of the inverse to $|\mathcal{T}|\log|\mathcal{T}|$ again involves the asymptotics of the Lambert W function.

Construction	$\log\lvert\mathcal{T}\rvert$	Key length
den Boer [6]	$\approx -\log\epsilon + \log\log\lvert\mathcal{M}\rvert$	$\approx -2\log\epsilon + 2\log\log\lvert\mathcal{M}\rvert$
Bierbrauer *et al.* [3]	$-\log\epsilon + 1$	$\approx -3\log\epsilon + 2\log\log\lvert\mathcal{M}\rvert$
Krawczyk [13]	$-\log\epsilon + \log(1 + 2\log\lvert\mathcal{M}\rvert)$	$-3\log\epsilon + 3\log(1 + 2\log\lvert\mathcal{M}\rvert) + 1$
This construction	$-\log\epsilon + 1$	$-4\log\epsilon + 3\log\log\lvert\mathcal{M}\rvert + 8$

Finally, simplicity of use and setup and performance should be briefly addressed. It is simpler to aim for a given security if there is only one parameter to adjust, and this would be a benefit of the present construction and the one by Bierbrauer *et al.* [3]. If the security parameter ϵ is fixed, so is the tag length in these two. The other two need to change tag length when the message length changes.

In terms of performance, the present construction and the one by Krawczyk [13] seem to have an advantage, since both decrease the size of the long message by using an LFSR which can efficiently be implemented in hardware and software. The other two use modular arithmetic in larger fields, which is somewhat less efficient. After shortening the message, Krawczyk's construction uses an OTP, again using efficient binary arithmetic, while the construction in this paper maps the intermediate short string into a tag by an SU_2 hash function. The difference between the two operations is not so large, since the length of the intermediate string is not so long in our construction. In all, the hash function family proposed in this paper compares well to the others in both cases, in that it is the only family that has both a simple relation between security parameter and construction parameters, *and* is efficient.

4 Conclusion

We have presented a simple new construction of an efficient ϵ-ASU_2 hash function family with small description length, for which the security parameter is independent of message length. The construction uses the idea of LFSR-based hashing together with Stinson's composition theorem for Universal hash function families. The resulting family has a key consumption that is logarithmic in the message length and linear in the tag length or logarithmic in the security parameter, with small (constant) coefficients. It is efficient, given that it requires a short key. These properties make our construction very suitable for information-theoretically secure authentication purposes in practical applications, especially in Quantum Cryptography.

Acknowledgments. We would like to thank the anonymous reviewers for their valuable comments on an earlier version of the paper.

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. 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)
4. Black, J.: Message authentication codes. Ph.D. thesis, University of California Davis, USA (2000)
5. 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)
6. den Boer, B.: A simple and key-economical unconditional authentication scheme. J. Comp. Sec. 2, 65–72 (1993)
7. Carter, L., Wegman, M.N.: Universal classes of hash functions. J. Comput. Syst. Sci. 18, 143–154 (1979)
8. Corless, R.M., Gonnet, G.H., Hare, D.E.G., Jeffrey, D.J., Knuth, D.E.: On the Lambert W function. Adv. Comput. Math. 5, 329–359 (1996)

9. Gemmell, P., Naor, M.: Codes for Interactive Authentication. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 355–367. Springer, Heidelberg (1994)
10. 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)
11. Johansson, T.: Bucket Hashing with a Small Key Size. In: Fumy, W. (ed.) EUROCRYPT 1997. LNCS, vol. 1233, pp. 149–162. Springer, Heidelberg (1997)
12. Kabatianskii, G., Smeets, B.J.M., Johansson, T.: On the cardinality of systematic authentication codes via error-correcting codes. IEEE Trans. Inf. Theory 42, 566–578 (1996)
13. Krawczyk, H.: LFSR-Based Hashing and Authentication. In: Desmedt, Y.G. (ed.) CRYPTO 1994. LNCS, vol. 839, pp. 129–139. Springer, Heidelberg (1994)
14. 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)
15. Mansour, Y., Nisan, N., Tiwari, P.: The computational complexity of universal hashing. In: Ortiz, H. (ed.) Proc. STOC 1990, pp. 235–243. ACM, New York (1990)
16. Nguyen, L.H., Roscoe, A.W.: A new bound for t-wise almost universal hash functions. IACR Cryptology ePrint Archive, Report 2009/153 (2009), http://eprint.iacr.org/2009/153
17. Preneel, B.: Analysis and design of cryptographic hash functions. Ph.D. thesis, Katholieke Universiteit Leuven, Belgium (1993)
18. 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)
19. Shor, P.W., Preskill, J.: Simple proof of security of the bb84 quantum key distribution protocol. Phys. Rev. Lett. 85, 441–444 (2000)
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)