Universal hashing and authentication codes

D. R. Stinson
Computer Science and Engineering
University of Nebraska
Lincoln, NE 68588-0115
stinson@bibd.unl.edu

Abstract

In this paper, we study the application of universal hashing to the construction of unconditionally secure authentication codes without secrecy. This idea is most useful when the number of authenticators is exponentially small compared to the number of possible source states (plaintext messages). We formally define some new classes of hash functions and then prove some new bounds and give some general constructions for these classes of hash functions. Then we discuss the implications to authentication codes.

1 Introduction

In this paper, we study the application of universal hashing to the construction of unconditionally secure authentication codes without secrecy. This idea is due to Wegman and Carter [We], who gave a construction which is useful when the number of authenticators is exponentially small compared to the number of possible source states (plaintext messages). We generalize the Wegman and Carter construction by formally defining some new classes of hash functions. We prove some new bounds and give some general constructions for these classes of hash functions. Then we discuss the implications to authentication codes. We are able to decrease the key *length* by a factor of four compared to the Wegman and Carter construction, while maintaining the same security.

The paper is organized as follows. In this introduction we give an informal discussion of the motivation for this paper. Section 2 is a brief review of the necessary background of authentication codes. Section 3 gives relevant definitions from universal hashing. Section 4 presents proofs of new lower bounds on the number of hash functions in certain types of universal classes and gives the basic construction for authentication codes from universal classes of hash functions. Section 5 gives a variety of new constructions for universal classes of hash functions. In Section 6, we bring all the theory together and discuss its implications to the construction of authentication codes. Section 7 makes further comments and discusses open questions.

J. Feigenbaum (Ed.): Advances in Cryptology - CRYPTO '91, LNCS 576, pp. 74-85, 1992.

Here are couple of hypothetical examples to motivate the problems we study. Suppose we have an authentication code with k possible source states and ℓ possible authenticators. That is, we wish to authenticate a $\log k$-bit message with a $\log \ell$-bit authenticator. An opponent who plays either impersonation or substitution can deceive the transmitter with a probability of at least $1/\ell$ (in either case).

One important point is that k and ℓ are independent parameters. k is the number of possible source states; ℓ is a security parameter.' For purposes of discussion, we identify two "reasonable" situations:

case 1 $k = 2^{20}$, $\ell = 2^{20}$

case 2 $k = 2^{2560}$, $\ell = 2^{20}$

Researchers have generally concentrated on the construction of codes which ensure that the opponent's deception probabilities are limited to these lower bounds. The main tool for constructing such codes has been a structure from combinatorial design theory called an orthogonal array (or equivalent structures, such as mutually orthogonal Latin squares, transversal designs or nets). Codes constructed by this method contain the minimum possible number of encoding rules, which is an important consideration since the encoding rule is secret information that must be exchanged over a secure channel before the transmission of a message.

In this paper, we shall use the language of orthogonal arrays. An $OA(n,k,\lambda)$ is a $\lambda n^2 \times k$ array of n symbols, such that in any two columns of the array every one of the possible n^2 pairs of symbols occurs in exactly λ rows. It is shown in [St1] that an $OA(n,k,\lambda)$ gives rise to an authentication code for k source states, with n authenticators and $\lambda n^2 \times k$ encoding rules (each row of the array gives rise in an obvious way to an encoding rule which assigns an authenticator to every possible source state, and the encoding rules are each used with equal probability). It has been known since 1945 (see [Pl]) that if an $OA(n,k,\lambda)$ exists, then

$$\lambda \geq \frac{k(n-1)+1}{n^2}.$$

It follows that $\lambda n^2 \geq n^2$ if $k \leq n+1$ and $\lambda n^2 \geq k(n-1)+1$ if $k \geq n+1$ (λn^2 is the number of encoding rules in the resulting code).

In case 1, we can obtain a code with 2^{40} encoding rules; i.e. 40 bits of key are required. In case 2, the minimum number of encoding rules using this method is $2^{2560}(2^{20}-1)+1$, or about 2580 bits of key.

The observation of Wegman and Carter [We] is that by *not* requiring the deception probabilities to be the theoretical minimum (i.e. $1/\ell$), one can sometimes reduce the number of encoding rules significantly, at least in the case where $k \gg \ell$.

2 Authentication codes

The general theory of of unconditional authentication has been developed by Simmons (see e.g. [Si1] and [Si2]), and has been extensively studied in recent years. In this section, we will give a brief review of some relevant known results concerning authentication without secrecy.

In the usual model for authentication, there are three participants: a *transmitter*, a *receiver*, and an *opponent*. The transmitter wants to communicate some information to the receiver using a public communications channel. The *source state* (or plaintext) is encrypted to obtain the *message* (ciphertext), which is sent through the channel. An *encoding rule* (or key) e defines the message $e(s)$ to be sent to communicate any source state s. Each encoding rule will be a one-to-one function from the source space to the message space. We assume the transmitter has a key source from which he obtains a key. Prior to any messages being sent, this key is communicated to the receiver by means of a secure channel.

We will use the following notation. Let $\mathcal{S}$ be a set of k source states; let $\mathcal{M}$ be a set of v messages; and let $\mathcal{E}$ be a set of encoding rules. Since each encoding rule is a one-to-one function from $\mathcal{S}$ to $\mathcal{M}$, we can represent a code by an $|\mathcal{E}| \times k$ matrix, where the rows are indexed by encoding rules, the columns are indexed by source states, and the entry in row e and column s is $e(s)$. We call this matrix the *encoding matrix*. For any encoding rule $e \in \mathcal{E}$, define $M(e) = \{e(s) : s \in \mathcal{S}\}$, i.e. the set of valid messages under encoding rule e. For an encoding rule e, and a message $m \in M(e)$, define $e^{-1}(m) = s$ if $e(s) = m$.

In this paper, we are studying authentication codes without secrecy. This means that $e(s) = e'(s')$ only if $s = s'$; i.e. the message uniquely determines the source state, irrespective of the encoding rule being used. Hence, we can partition the set of messages $\mathcal{M}$ into k subsets $\mathcal{M}_s$, $s \in \mathcal{S}$, such that $\mathcal{M}_s = \{e(s) : e \in \mathcal{E}\}$.

Suppose the opponent has the ability to introduce messages into the channel and/or to modify existing messages. When the opponent places a (new) message m' into the channel, this is called *impersonation*. When the opponent sees a message m and changes it to a message $m' \neq m$, this is called *substitution*. In either case, his goal is to have m' accepted as authentic by the receiver. That is, if e is the encoding rule being used (which is *not* known to the opponent), then the opponent is hoping that $m' = e(s)$ for some source state s.

We assume that there is some probability distribution on the source states, which is known to all the participants. Given the probability distributions on the source states, the receiver and transmitter will choose a probability distribution for $\mathcal{E}$, called an *encoding strategy*. Once the transmitter/receiver have chosen the encoding strategy, it is possible to determine, for $i = 0, 1$ a probability denoted P_{d_i}, which is the probability that the opponent can deceive the transmitter/receiver by impersonation and substitution, respectively.

It is not difficult to show that $P_{d_0} \geq k/v$ and that $P_{d_0} = k/v$ only if $|\mathcal{M}_s| = v/k$ for every source state s [Si1]. In this paper, we will confine our attention to codes in which $P_{d_0} = k/v$. In this situation, we can define a set $\mathcal{A}$ of $\ell = v/k$ *authenticators* and a mapping $\phi : \mathcal{M} \rightarrow \mathcal{A}$ such that, for every $s \in \mathcal{S}$, $\{\phi(m) : m \in \mathcal{M}_s\} = \mathcal{A}$. We can then obtain an isomorphic code by defining for every encoding rule e an *authentication rule* e^ϕ defined by $e^\phi(s) = \phi(e(s))$ for every source state s. In this new code, every message consists of a source state concatenated with an authenticator from $\mathcal{A}$; i.e. $m = (s, e^\phi(s))$. In terms of $\ell = |\mathcal{A}|$, we have $P_{d_0} \geq 1/\ell$. It then follows that $P_{d_1} \geq 1/\ell$, as well [St2].

Codes with $P_{d_0} = P_{d_1} = 1/\ell$ are in fact equivalent to orthogonal arrays, as follows.

Theorem 2.1 *[St4] Suppose we have an authentication code without secrecy in which* $P_{d_0} = P_{d_1} = k/v = 1/\ell$. *Then* $b \geq k(\ell - 1) + 1$, *and equality occurs if and only if the authentication matrix is an orthogonal array* $OA(\ell, k, \lambda)$ *where* $\lambda = (k(\ell - 1) + 1)/\ell^2$ *and the authentication rules are used with equal probability.*

3 Universal hashing

Universal classes of hash functions were introduced by Carter and Wegman [Ca], and were studied further by Sarwate [Sa], Wegman and Carter [We] and Stinson [St3]. In this paper, we are interested in the application of universal hashing to authentication codes. First, let us review the relevant definitions.

Let A and B be finite sets, and denote $a = |A|$ and $b = |B|$, where $a \geq b$. A function $h : A \rightarrow B$ will be termed a *hash function*. For a hash function h, and for $x, y \in A$, $x \neq y$, define $\delta_h(x, y) = 1$ if $h(x) = h(y)$, and $\delta_h(x, y) = 0$ otherwise. That is, $\delta_h(x, y) = 1$ if and only if the hashed values of x and y collide. For a finite set H of hash functions, define $\delta_H(x, y) = \sum_{h \in H} \delta_h(x, y)$. Hence, $\delta_H(x, y)$ counts the number of hash functions in H under which x and y collide.

The idea of a universal class of hash functions is to define a collection H of hash functions in such a way that a random choice of a function $h \in H$ yields a low probability that any two distinct inputs x and y will collide when their hashed values are computed using the function h. Note that this probability can be computed to be $\delta_H(x, y)/|H|$. By choosing H suitably, it is possible to make this probability small for *all* choices of x and y.

However, for the purpose of practical applications, it is important not only to have $\delta_H(x, y)/|H|$ small for every x and y, but $|H|$ should be small as well.

First, let's consider $\delta_H(x, y)$. We state without proof a bound that was noted in [Sa, p. 42].

Theorem 3.1 *For any class H of hash functions from A to B, there exist distinct elements $x, y \in A$ such that $\delta_H(x, y) \geq |H|(a - b)/(b(a - 1))$, where $a = |A|$ and $b = |B|$.*

Here now are two definitions of classes of hash functions.

1. Let ϵ be a positive real number. H is *ϵ−almost universal$_2$* (or $\epsilon - AU_2$) if $\delta_H(x, y) \leq \epsilon|H|$ for all $x, y \in A$, $x \neq y$.
2. Let ϵ be a positive real number. H is *ϵ−almost strongly-universal$_2$* (or $\epsilon - ASU_2$) if the following two conditions are satisfied:
   (a) for every $x_1 \in A$ and for every $y_1 \in B$, $|\{h \in H : h(x_1) = y_1\}| = |H|/|B|$
   (b) for every $x_1, x_2 \in A$ $(x_1 \neq x_2)$ and for every $y_1, y_2 \in B$,
   $$|\{h \in H : h(x_1) = y_1, h(x_2) = y_2\}| \leq \epsilon|H|/|B|.$$

The first definition is saying that the probability of collision is at most ϵ for any two inputs x and y. The second definition says that any input x_1 is mapped to any hashed value y_1 with probability $1/b$; and given that x_1 is mapped to y_1, the conditional probability that x_2 is mapped to y_2 is at most ϵ, for any $x_2, y_2, x_2 \neq x_1$.

Special cases of these definitions have been previously studied in the literature. For example, $(a - b)/(ab - b) - AU_2$ has been called *optimally universal* [Sa], $(1/b) - AU_2$

has been called *universal* [Ca], and $(1/b) - ASU_2$ has been called *strongly universal* [We]. Note that in an $\epsilon - ASU_2$ class, it must be the case that $\epsilon \geq 1/b$.

$\epsilon - ASU_2$ classes of hash functions can be used for authentication. If we have such a class H of hash functions from A to B, then we can think of the elements of A as source states and the elements of B as authenticators. Each hash function gives rise to an encoding rule, and the encoding rules are used with equal probability. The following result is immediate.

Theorem 3.2 *If there exists an $\epsilon - ASU_2$ class H of hash functions from A to B, then there exists an authentication code for $|A|$ source states, having $|B|$ authenticators and $|H|$ encoding rules, such that $P_{d_0} = 1/|B|$ and $P_{d_1} \leq \epsilon$.*

4 Lower bounds on the size of classes of hash functions

In view of the fact that classes of hash functions give rise to authentication codes, it is of interest to compute lower bounds on the number of hash functions required. We present two lower bounds in this section, which generalize some previously known bounds. The proofs use a nice variance technique, but are omitted from this extended abstract due to space limitations.

Theorem 4.1 *If there exists an $\epsilon - AU_2$ class H of hash functions from A to B, where $a = |A|$ and $b = |B|$, then*

$$|H| \geq \frac{a(b-1)}{a(\epsilon b - 1) + b^2(1-\epsilon)}.$$

By substituting $\epsilon = 1/b$ and $\epsilon = (a-b)/(ab-b)$ into the above bound, respectively, we obtain the following corollary.

Corollary 4.1 *[St3] Suppose H is a class of hash functions from A to B, where $a = |A|$ and $b = |B|$. If H is U_2, then $|H| \geq a/b$. If H is OU_2, then $|H| \geq (a-1)/(b-1)$.*

We next present a lower bound on the number of hash functions in a $\epsilon - ASU_2$ class.

Theorem 4.2 *If there exists an $\epsilon - ASU_2$ class H of hash functions from A to B, where $a = |A|$ and $b = |B|$, then*

$$|H| \geq 1 + \frac{a(b-1)^2}{b\epsilon(a-1) + b - a}.$$

In the case $\epsilon = 1/b$, we get the following corollary.

Corollary 4.2 *[St3] If there exists an SU_2 class H of hash functions from A to B, where $a = |A|$ and $b = |B|$, then $|H| \geq 1 + a(b-1)$. Further, $|H| = 1 + a(b-1)$ if and only if there is an $OA(b, a, \lambda)$, where $\lambda = (a(b-1)+1)/a^2$.*

5 Constructions

In this section, we give some direct and recursive constructions for universal classes of hash functions. First, we recall some direct constructions that are special cases of constructions from [St3] and [Ca].

Theorem 5.1 *Let q be a prime power. Then there exists a U_2 class H of hash functions from A to B, where $|A| = q^2$, $|B| = q$ and $|H| = q$ (hence $\epsilon = 1/q$).*

Proof: Let $A = GF(q) \times GF(q)$, let $B = GF(q)$ and let $H = \{h_x : x \in GF(q)\}$, where $h_x(a, b) = b - ax$. □

Theorem 5.2 *Let q be a prime power. Then there exists an SU_2 class H of hash functions from A to B, where $|A| = q^2$, $|B| = q$ and $|H| = q^3$ (hence $\epsilon = 1/q$).*

Proof: Let $A = GF(q) \times GF(q)$, let $B = GF(q)$ and let $H = \{h_{xyz} : x, y, z \in GF(q)\}$, where $h_{xyz}(a, b) = x + ay + bz$. □

Theorem 5.3 *Let q be a prime power. Then there exists an SU_2 class H of hash functions from A to B, where $|A| = q$, $|B| = q$ and $|H| = q^2$ (hence $\epsilon = 1/q$).*

Proof: Let $A = B = GF(q)$ and let $H = \{h_{xy} : x, y \in GF(q)\}$, where $h_{xy}(a) = x + ay$. □

Now we present some methods of combining classes of hash functions which generalize similar constructions from [We] and [Sa].

Theorem 5.4 (Cartesian Product) *If there exists an $\epsilon - AU_2$ class H of hash functions from A to B, then, for any integer $i \geq 1$, there exists an $\epsilon - AU_2$ class H^i of hash functions from A^i to B^i with $|H| = |H^i|$.*

Proof: For every $h \in H$, define a hash function $h^i : A^i \rightarrow B^i$ by the rule $h^i(a_1, \ldots, a_i) = (h(a_1), \ldots, h(a_i))$. Define $H^i = \{h^i : h \in H\}$. □

Theorem 5.5 (Composition 1) *For $i = 1, 2$, suppose there exists an $\epsilon_i - AU_2$ class H_i of hash functions from A_i to B_i, where $A_2 = B_1$. Then there exists an $\epsilon - AU_2$ class H of hash functions from A_1 to B_2, where $\epsilon = \epsilon_1 + \epsilon_2$ and $|H| = |H_1| \times |H_2|$.*

Proof: For every $h_i \in H_i$, $i = 1, 2$, we define a hash function $h : A_1 \rightarrow B_2$ by the rule $h(a) = h_2(h_1(a))$. Let H be the set of all such hash functions. For any two inputs, the probability of collision is at most $\epsilon_1 + (1 - \epsilon_1)\epsilon_2 < \epsilon_1 + \epsilon_2$. □

Theorem 5.6 (Composition 2) *Suppose H_1 is an $\epsilon_1 - AU_2$ class of hash functions from A_1 to B_1, and suppose H_2 is an $\epsilon_2 - ASU_2$ class of hash functions from B_1 to B_2. Then there exists an $\epsilon - ASU_2$ class H of hash functions from A_1 to B_2, where $\epsilon = \epsilon_1 + \epsilon_2$ and $|H| = |H_1| \times |H_2|$.*

Proof: For every $h_i \in H_i$, $i = 1, 2$, define a hash function $h : A_1 \rightarrow B_2$ by the rule $h(a) = h_2(h_1(a))$. Let H be the set of all such hash functions. Let $x_1, x_2 \in A_1$ $(x_1 \neq x_2)$ and let $y_1, y_2 \in B_2$. How many functions in H map x_1 to y_1 and x_2 to y_2? Suppose first that $y_1 = y_2$. Let p denote that probability that x_1 and x_2 collide under a hash function from H_1. Then the maximum number is at most

$$p|H_1| \times \frac{|H_2|}{b} + (1-p)|H_1| \times \frac{\epsilon_2 |H_2|}{b} \leq (\epsilon_1 + \epsilon_2)|H_1| \times \frac{|H_2|}{b}.$$

If $y_1 \neq y_2$ then the number is less. Since $p \leq \epsilon_1$, it follows that we have an $\epsilon - ASU_2$ class with $\epsilon = \epsilon_1 + \epsilon_2$. □

6 The application of universal hashing to authentication

We now combine the constructions of the previous section to obtain authentication codes.

Theorem 6.1 *Let q be a prime power and let $i \geq 1$ be an integer. Then there exists an $(i/q) - AU_2$ class of q^i hash functions from A to B, where $|A| = q^{2^i}$ and $|B| = q$.*

Proof: Apply Theorems 5.1, 5.4 and 5.5. □

Theorem 6.2 *Let q be a prime power and let $i \geq 1$ be an integer. Then there exists an $((i+1)/q) - ASU_2$ class of q^{i+2} hash functions from A to B, where $|A| = q^{2^i}$ and $|B| = q$.*

Proof: Apply Theorems 6.1, 5.3 and 5.6. □

Theorem 6.3 *Let q be a prime power and let $i \geq 1$ be an integer. Then there exists an $(i/q^2 + 1/q) - ASU_2$ class of q^{2i+3} hash functions from A to B, where $|A| = q^{2^i}$ and $|B| = q$.*

Proof: Apply Theorems 6.1 (replacing q by q^2), 5.2 and 5.6. □

Let's look at the codes we can obtain via Theorem 3.2 for case 2 of the Introduction using the above results. If we apply Theorem 6.2 with $q = 2^{20}$ and $i = 7$, we get an authentication code with 2^{180} encoding rules in which $P_{d_0} = 2^{-20}$ and $P_{d_1} = 2^{-17}$. On the other hand, if we apply Theorem 6.3, we get an authentication code with 2^{340} encoding rules in which $P_{d_0} = 2^{-20}$ and $P_{d_1} < 2^{-19}$. On the other hand, if we require $P_{d_0} = P_{d_1} = 2^{-20}$ then Theorem 2.1 tells us that the number of encoding rules is at least $2^{2560}(2^{20} - 1) + 1 \approx 2^{2580}$. Hence, we obtain an enormous reduction in the size of the key space by increasing P_{d_1} only slightly.

Taking logarithms, we can rephrase the above discussion by saying that authentication of a 2560-bit source with a 20-bit authenticator requires 180, 340 and 2580 bits of key, respectively.

The following bound, obtained from Theorem 6.2, is similar to that given in [We].

Theorem 6.4 *There exists an authentication code for an a'-bit source with a b'-bit authenticator, obtaining deception probabilities $P_{d_0} = 2^{-b'}$ and $P_{d_1} = (\log a' - \log b' + 1)2^{-b'}$, with a key of length $(\log a' - \log b' + 2)b'$.*

Proof: Write $b = 2^{b'} = q$ and $a = 2^{a'} = b^{2^i}$. Then $i = \log a' - \log b'$. Apply Theorem 6.2. □

In Section 3 of [We], a similar construction for a $2/b - ASU_2$ class of hash functions is presented, in which $\log|H|$ is $4(b' + \log\log a')\log a'$. However, it appears to me that the analysis of ϵ too low by a factor of $\log a'$, and that the class is in fact an $\epsilon - ASU_2$ class where $\epsilon = (2\log a')2^{-b'}$. With respect to our example of case 2, the key length would be about 1056. In general, our bound on the key length (Theorem 6.4) is lower by a factor of four.

7 Further comments and open questions

We have been emphasizing the application of hash functions to the construction of authentication codes with $k \gg \ell$. This is because previously known techniques are already quite good when k and ℓ are even polynomially related. For, suppose we have an $\epsilon - ASU_2$ class H of hash functions from A to B where $|A| = a(= k)$ and $|B| = b(= \ell)$. Consider the effect of increasing ϵ from $1/b$ to $2/b$ in the bound of Theorem 4.2. The ratio of the two bounds is about $(a + b)/b$. Hence, the potential for decreasing $|H|$ significantly is much greater when $a \gg b$.

The constructions for $\epsilon - ASU_2$ classes of hash functions given in Theorems 6.1 and 6.2 have $|H|$ considerably larger than the lower bound given in Theorem 4.2. There are relatively few situations where the bound of Theorem 4.2 is known to be met with equality. The only cases known to us are as follows. First, if $\epsilon = 1/b$, then the class is an SU_2 class and is equlvalent to an orthogonal array $OA(b, a, \lambda)$, where $\lambda = (a(b-1)+1)/a^2$. (Corollary 4.2). Some infinite classes of these are known to exist; see [St3].

The only examples with $\epsilon > 1/b$ known to us are given below as Theorem 7.1. The construction uses a *balanced incomplete block design*, or BIBD. A $(v, k, \lambda)-$BIBD is a pair $(X, \mathcal{A})$, where $|X| = v$ is a set of elements called *points* and $\mathcal{A}$ is a family of $k-$subsets of X (called *blocks*) such that every pair of points occurs in exactly λ blocks. It is not difficult to see that every point occurs in precisely $r = \lambda(v-1)/(k-1)$ blocks and that the total number of blocks is $b = \lambda v(v-1)/(k(k-1))$. A $(v, k, \lambda)-$BIBD is *resolvable* if the blocks can be partitioned into r parallel classes, each of which consists of v/k blocks that partition the set of points.

Theorem 7.1 *Suppose there exists a resolvable $(v, k, 1)-BIBD$. Then there exists an $\epsilon - ASU_2$ class H of hash functions from A to B, where $|A| = (v-1)/(k-1)$, $|B| = v/k$, $\epsilon = 1/k$ and $|H| = v$.*

Remark: In terms of $a = |A|$ and $b = |B|$, $|H| = b(a-1)/(a-b)$.

Proof: Let $(X, \mathcal{A})$ be the hypothesized $(v, k, 1)$–BIBD, and let $\mathcal{P}_1, \ldots, \mathcal{P}_r$ be the r parallel classes. Name the blocks in each $\mathcal{P}_i$ as P_{ij}, $1 \leq j \leq v/k$. Let $A = \{1, \ldots, r\}$ and $B = \{1, \ldots, v/k\}$. For each point $x \in X$, we define a hash function h_x by the rule $h_x(i) = j$ if and only if $x \in P_{ij}$. □

The following example illustrates the construction of Theorem 7.1 with $v = 8$, $k = 2$.

Example 7.1 *A $1/2-ASU_2$ class of 8 hash functions from $\{1,2,3,4,5,6,7\}$ to $\{1,2,3,4\}$:*

	1	2	3	4	5	6	7
h_1	1	1	1	1	1	1	1
h_2	1	2	3	2	4	4	3
h_3	3	1	2	3	2	4	4
h_4	4	3	1	2	3	2	4
h_5	4	4	3	1	2	3	2
h_6	2	4	4	3	1	2	3
h_7	3	2	4	4	3	1	2
h_8	2	3	2	4	4	3	1

For applications to authentication codes, the construction of Theorem 7.1 is not of much use, since $a \approx b$ and many other constuctions are known in this case.

A very interesting open problem would be to find examples of $\epsilon - ASU_2$ classes of hash functions H with $\epsilon > 1/b$ and $a \gg b$, such that $|H|$ meets the lower bound of Theorem 4.2 with equality.

Acknowledgements

This research was supported by NSERC grant A9287 and by the Center for Communication and Information Science at the University of Nebraska.

This paper is a preliminary version. A final version has been submitted for publication in the IEEE Transactions on Information Theory.

References

[Ca] J. L Carter and M. N. Wegman. *Universal classes of hash functions*, J. Comput. System Sci. **18** (1979), 143–154.

[Pl] R. L. Plackett and J. P. Burman. *The design of optimum multi-factorial experiments*, Biometrika **33** (1945), 305–325.

[Sa] D. V. Sarwate. *A note on universal classes of hash functions*, Inform. Proc. Letters **10** (1980), 41–45.

[Si1] G. J. Simmons. *Message authentication: a game on hypergraphs*, Congr. Numer. **45** (1984), 161–192.

[Si2] G. J. Simmons. *A survey of information authentication*, Proc. of the IEEE **76** (1988), 603–620.

[St1] D. R. Stinson. *Some constructions and bounds for authentication codes*, J. Cryptology **1** (1988), 37–51.

[St2] D. R. Stinson. *The combinatorics of authentication and secrecy codes*, J. Cryptology **2** (1990), 23–49.

[St3] D. R. Stinson. *Combinatorial techniques for universal hashing*, submitted to J. Comput. System Sci.

[St4] D. R. Stinson. *Combinatorial characterizations of authentication codes*, submitted to Designs, Codes and Cryptography (a preliminary version appears elsewhere in these proceedings).

[We] M. N. Wegman and J. L. Carter. *New hash functions and their use in authentication and set equality*, J. Comput. System Sci. **22** (1981), 265–279.