2011

Fuzzy universal hashing and approximate authentication

Rei Safavi-Naini
University of Calgary, rei@uow.edu.au

Joseph Tonien
University of Wollongong, dong@uow.edu.au

Publication Details

Safavi-Naini, R. & Tonien, D. (2011). Fuzzy universal hashing and approximate authentication. Discrete Mathematics, Algorithims and Applications, 3 (4), 587-607.

Fuzzy universal hashing and approximate authentication

Keywords
fuzzy, universal, authentication, approximate, hashing

Disciplines
Engineering | Science and Technology Studies

Publication Details
Safavi-Naini, R. & Tonien, D. (2011). Fuzzy universal hashing and approximate authentication. Discrete Mathematics, Algorithims and Applications, 3 (4), 587-607.

Fuzzy Universal Hashing and Approximate Authentication

Reihaneh Safavi-Naini
rei@uow.edu.au

Dongvu Tonien
dong@uow.edu.au

School of IT & CS, University of Wollongong, NSW, 2522, Australia

Abstract

Traditional data authentication systems are sensitive to single bit changes and so are unsuitable for message spaces that are naturally 'fuzzy' where 'similar' messages are considered 'the same' or at least indistinguishable. In this paper, we study *unconditional secure* approximate authentication. We generalize *traditional* universal hashing to *fuzzy* universal hashing and use it to construct secure approximate authentication for multiple messages.

1 Introduction

Traditional Message Authentication. The setting for a traditional message authentication is as follows. Two parties, a *sender* A and a *receiver* B, agree on a secret key k. There are two algorithms operated with the key k, an authentication tag generation algorithm TAG_k and a verification algorithm VERIFY_k. If A wants to send a message m to B, then A generates an authentication tag as $t = \text{TAG}_k(m)$ and sends the pair (m, t) to B. Upon receiving (m, t), B executes $\text{VERIFY}_k(m, t)$. The verification algorithm VERIFY_k returns 0 or 1. If it returns 1 then B accepts (m, t) as an authenticated message.

Traditional message authentication detects single bit changes in the message with high probability. That is, if B receives a pair (m', t) where m' is different from the original m (even in a single bit), then the verification algorithm $\text{VERIFY}_k(m', t)$ returns 0 with high probability.

Carter-Wegman's Universal Hashing. Universal hashing has played an important role in constructing traditional authentication system [20, 21, 15, 16, 17, 22, 1, 19, 5, 6, 4]. *Universal* and *strongly universal* hash families was first introduced by Carter and Wegman [10, 23] in a construction of a counter-based authentication scheme for multiple messages. A *strongly universal*$_2$ hash family H is a set of hash functions h mapping $\mathcal{A} \to \mathcal{B}$ such that for any two *distinct* elements a_1, a_2 of $\mathcal{A}$, the values $(h(a_1), h(a_2))$ is uniformly distributed in the set $\mathcal{B}^2$. In Carter-Wegman's counter-based authentication scheme, the sender and the receiver agree on a secret hash function h randomly chosen from a *strongly universal*$_2$ hash family and a tuple of n random elements $(r_1, \ldots, r_n)$ of $\mathcal{B}$. The sender can send up to n messages to the receiver. To send a message $m \in \mathcal{A}$ with a message number $i \in \{1, \ldots, n\}$, the sender generates a tag $t = h(m) \oplus r_i$ and sends (m, i, t) to the receiver. The scheme is proved to be secure against an adversary with infinite computer power. That is, the probability for an adversary to forge a message is no more than $\frac{1}{|B|}$ – the probability of guessing correct tag.

Stinson [20, 21] gave generalized notions of ϵ-*almost universal* and ϵ-*almost strongly universal* hash families. The requirement for an ϵ-*almost strongly universal*$_2$ hash family H is that, for any two

distinct elements a_1, a_2 of $\mathcal{A}$ and for any two elements b_1, b_2 of $\mathcal{B}$, the probability for $h \in H$ that $b_1 = h(a_1)$ and $b_2 = h(a_2)$ is no more than ϵ. If an *ϵ-almost strongly universal$_2$* is used then the probability for an adversary to forge a message in the above Carter-Wegman's counter-based authentication scheme is no more than ϵ. Krawczyk [15] and Rogaway [17] later showed that the same security can be obtained with a weaker condition, called *ϵ-almost xor universal$_2$*, on the hash family (actually Krawczyk used the term *ϵ-otp-secure*). Stinson [22] further generalized the property *ϵ-almost xor universal$_2$* for the case $(\mathcal{B}, +)$ is a commutative group and called it *ϵ-almost Δ universal$_2$*. Black et al [5, 6]'s construction uses both universal hash families and pseudorandom random function families, and reduces the requirement on the hash family to *ϵ-almost universal$_2$*.

Approximate Authentication. Bit sensitivity in traditional message authentication is, however, a severe limitation for message spaces that are naturally 'fuzzy' and where 'similar' messages are considered 'the same' or at least indistinguishable. Such message spaces commonly arise in representing such media objects as images, videos or audio signals, where the distinction between two objects is limited by human sensory systems; or in representing such human biometric data as fingerprintings or retina scans, where two biometric readings from the same person are unlikely to be exactly the same. Authentication of 'fuzzy' message spaces has new challenges. In particular, the authentication system must tolerate variations in the message that are considered 'indistinguishable' but detect changes that are due to malicious tampering.

We consider *unconditional security* and make no assumption regarding the computational resources of the adversaries. This is appropriate when no computational assumption about adversary's power can be made (typically for small mutually distrusting groups of entities that do not know each others computational or technological advantages, e.g. advances in quantum computations, as is the setting between nations), or when the aim is to design systems that last indefinitely.

We formalize the concept of fuzziness in a message space and introduce *fuzzy universal hashing* on fuzzy spaces. Our proposed definitions of fuzzy universal hashings coincide with the traditional definitions of universal hashings when the fuzzy space is the *discrete* fuzzy space, i.e. the space with no fuzziness where every two different messages are considered as distinguishable. We show that a secure approximate authentication scheme for multiple messages can be constructed using fuzzy universal hashing.

Our Contribution. We view our contribution as a bridge connecting between universal hashing and approximate authentication. We first generalize *traditional* universal hashings to *fuzzy* universal hashings and then show that Carter-Wegman's framework can be applied with the proposed fuzzy universal hashing to construct a secure approximate authentication scheme for multiple messages.

FUZZY SPACES. We define a fuzzy space simply as a message space $\mathcal{M}$ with a function $\chi : \mathcal{M} \times \mathcal{M} \to \{0, 1\}$. If $\chi(m, m') = 1$ then the two messages m and m' are considered as *indistinguishable* and if $\chi(m, m') = 0$ then they are *distinguishable*. Unlike other models of fuzziness which based on some kind of distance measure, our model requires the minimum. That is, we only require two properties on the fuzzy function χ:

- every message should be indistinguishable to itself, i.e. $\chi(m, m) = 1$; and
- symmetry, i.e. $\chi(m, m') = \chi(m', m)$.

With a minimum requirement on the fuzzy function, we leave the freedom to individual applications to define their own fuzzy function. In some applications, if there is a distance function d that can measure the difference of two messages then it may define $\chi(m, m') = 1$ if $d(m, m') \leq \sigma$ for some

threshold value σ. In some other applications, if there is an extraction function $\mathsf{Ext} : \mathcal{M} \to \mathcal{E}$ that can be used to extract characteristic features of the message and there is a distance measure d on this set $\mathcal{E}$ of features, then it may define $\chi(m, m') = 1$ if $d(\mathsf{Ext}(m), \mathsf{Ext}(m')) \leq \sigma$ for some threshold value σ.

FUZZY UNIVERSAL HASHING. We define several notions of fuzzy hashings: ϵ-*fuzzy strongly universal*$_1$, ϵ-*fuzzy strongly universal*$_2$, ϵ-*fuzzy strongly differential universal*$_2$ and ϵ-*fuzzy universal*$_2$ which naturally generalize ϵ-*almost strongly universal*$_1$, ϵ-*almost strongly universal*$_2$, ϵ-*almost strongly differential universal*$_2$ and ϵ-*almost universal*$_2$, respectively. We decided to remove all the term *"almost"* because it is implied from the prefix ϵ and we also thought that ϵ-*fuzzy almost strongly universal*$_1$ is rather long. We also follow Bernstein [4] to use the term *"differential"* instead of *"xor"*.

These generalized definitions will coincide with the *traditional* definitions of universal hashing when there is no fuzziness in the message space, i.e. $\chi(m, m') = 0$ for all $m \neq m'$. We prove some bounds on the error term ϵ which again become familiar bounds in the traditional case. We follow Black et al [5, 6] technique to construct ϵ-*fuzzy universal*$_2$ hash families for arbitrary length messages from an ϵ-*fuzzy universal*$_2$ hash family for fixed-length messages.

APPLICATION OF FUZZY UNIVERSAL HASHING TO APPROXIMATE AUTHENTICATION. As *traditional* universal hashing has been used to construct *traditional* secure authentication, our proposed *fuzzy* universal hashing can be used to construct secure *approximate* authentication for multiple messages.

We give two constructions of approximate authentication. The first one is an one-time pad based scheme using an ϵ-*fuzzy strongly differential universal*$_2$ (or an ϵ-*fuzzy strongly universal*$_2$) hash family. The second one is an PRF based scheme using an ϵ-*fuzzy universal*$_2$ hash family and a pseudorandom function family. Both schemes are proven *unconditional* secure.

Our security model is very strong. We let the adversary to access the tagging algorithm, thus, obtaining up to n tags values $t_1, \ldots, t_n$ for any chosen messages $(m_1, counter_1), \ldots, (m_n, counter_n)$ with one condition that no two different messages $m_t \neq m_s$ that have the same message number $counter_t = counter_s$. The adversary wins if it can produce a forgery $(m, counter, t)$ that passes the verification algorithm and either *counter* is a new message number or $counter = counter_j$ and m is distinguishable from m_j.

Related Works. 'Fuzzy information' arises naturally in some situations, such as when a list of answers to subjective questions is used to correctly identify a user, as in the *personal entropy system* of [12], or in the use of biometric data, such as fingerprints or retina scans for authentication [7].

In quantum mechanics the Heisenberg uncertainty principle gives physical limits on the accuracy with which measurements may be made, and information known. Measurements on quantum mechanical systems are therefore also natural fuzzy sources.

Fuzzy extractors [11, 8, 9] provide methods of reconstructing a private string extracted from a fuzzy secret, using a matching public string. *Fuzzy sketches* reconstruction of the fuzzy secret with any public string. In both cases the aim is to ensure the system secrets, the fuzzy secret or private string, not leak. Reusable fuzzy extractors [8] strengthen the security model by allowing adversaries to access extraction and regeneration oracles, and to adaptively choose perturbations of the secret under attack.

Fuzzy extractors and sketches can be applied for user authentication by extracting cryptographic keys from users biometric data and ensuring that the keys (or the biometric data itself) can be reproduced without error. In fuzzy hashing based authentication, however, a randomly chosen hash

function is used to construct a tag for a fuzzy secret, such that tampering with the fuzzy secret can be detected. In the case of biometric data, the tag can be seen as a key dependent hash that can be stored in the server database, instead of the actual fuzzy secret, hence providing a higher level of security.

Authentication of media data requires tolerance to small changes in information which occur during processing such as compression and decompression, or during application of various types of filtering used for enhancement. Traditional data authentication systems are sensitive to single bit changes and so are unsuitable for media authentication. Feature hashing algorithms [13, 14] extract features of media objects that remain invariant under the required set of transformations. Fuzzy hashing based approximate authentication provide a cryptographic model, in the unconditionally secure setting, for keyed feature hashing.

The paper is organized as follows. In section 2 we define fuzzy spaces and fuzzy universal hashing. Relationship between different classes of (fuzzy) hash families and construction of fuzzy hash families are presented in section 2.3 and section 2.4, respectively. In section 3, we discuss application of fuzzy universal hash families to approximate authentication. Section 3.1 presents the one-time pad based approximate authentication and section 3.2 presents the PRF based approximate authentication.

2 Fuzzy Universal Hashing

2.1 Fuzzy Spaces

Let $\mathcal{M}$ be a message space. We will introduce the notion of *fuzzy functions* on the message space $\mathcal{M}$ that decides whether two messages are distinguishable or not. A fuzzy function is a function $\chi : \mathcal{M}\times\mathcal{M} \to \{0,1\}$ such that for any m and m' in $\mathcal{M}$, if $\chi(m,m') = 1$ then m and m' are considered as *"indistinguishable"* and if $\chi(m,m') = 0$ then m and m' are considered as *"distinguishable"*. There are two conditions on the fuzzy function χ:

- $\chi(m,m) = 1$ for all $m \in \mathcal{M}$; and
- $\chi(m,m') = \chi(m',m)$ for all $m, m' \in \mathcal{M}$.

A *fuzzy space* $(\mathcal{M},\chi)$ is a message space $\mathcal{M}$ together with a fuzzy function χ. There are two *special* fuzzy functions. The first one is the *indiscrete fuzzy function* that treats all messages as indistinguishable, i.e. $\chi(m,m') = 1$ for all $m, m' \in \mathcal{M}$. The second one is the *discrete fuzzy function* that treats any two different messages as distinguishable, i.e. $\chi(m,m') = 0$ for all $m \neq m' \in \mathcal{M}$. A message space with the indiscrete fuzzy function forms an *indiscrete fuzzy space*; and a message space with the discrete fuzzy function forms a *discrete fuzzy space*.

In this article, we do not consider indiscrete fuzzy spaces, that is, we implicitly assume that in the considered fuzzy space, there always exist two messages m and m' such that $\chi(m,m') = 0$.

The set of messages that are indistinguishable to m is denoted by

$$\chi_m = \{m' : m' \in \mathcal{M}, \chi(m,m') = 1\}.$$

Denote

$$|\chi| = \max_{m\in\mathcal{M}} |\chi_m|.$$

We always have $m \in \chi_m$ and $|\chi| \geq 1$.

When χ is the discrete fuzzy function then $\chi_m = \{m\}$ for all $m \in \mathcal{M}$ and thus, $|\chi| = 1$. The value $|\chi|$ measures the fuzziness of the space $(\mathcal{M}, \chi)$. In a traditional authentication system, there is no fuzziness in the message space, thus, it is equivalent to the discrete fuzzy space.

When there is a distance function d on the set of messages $\mathcal{M}$ then a fuzzy function can be defined as $\chi(m, m') = 1$ if any only if $d(m, m') \leq \sigma$ for some threshold distance σ.

2.2 Fuzzy Hashing on Fuzzy Spaces

Universal hash families play important role in constructing traditional authentication system. We generalize definitions of several universal hash families for fuzzy spaces. As we will see later, our generalized definitions coincide with the traditional definitions when the fuzzy function is the discrete fuzzy function.

We start with the definition of the traditional ϵ-*strongly universal*$_1$ hash family.

Definition 1 ϵ-Strongly universal$_1$ *(ϵ-SU$_1$)*
Let A, B be finite sets. Let H be a finite hash family mapping A to B. H is ϵ-strongly universal$_1$ *if for every $x \in A$ and for every $y \in B$,*

$$|\{\ h \in H : y = h(x)\ \}| \leq \epsilon |H|.$$

This is our definition of ϵ-*fuzzy strongly universal*$_1$ hash families.

Definition 2 ϵ-Fuzzy strongly universal$_1$ *(ϵ-FSU$_1$)*
Let (A, χ) be a finite fuzzy space, B be a finite set. Let H be a finite hash family mapping A to B. H is ϵ-fuzzy strongly universal$_1$ *if for every $x \in A$ and for every $y \in B$,*

$$|\{\ h \in H : y \in h(\chi_x)\ \}| \leq \epsilon |H|.$$

When χ is the discrete fuzzy function, $\chi_x = \{x\}$, and $y \in h(\chi_x)$ is equivalent to $y = h(x)$, thus the condition in the Definition 2,

$$|\{\ h \in H : y \in h(\chi_x)\ \}| \leq \epsilon |H|$$

coincides with the condition in the Definition 1,

$$|\{\ h \in H : y = h(x)\ \}| \leq \epsilon |H|.$$

Now let us look at the definition of ϵ-*strongly universal*$_2$ hash families.

Definition 3 ϵ-Strongly universal$_2$ *(ϵ-SU$_2$)*
Let A, B be finite sets. Let H be a finite hash family mapping A to B. H is ϵ-strongly universal$_2$ *if for every $x_1, x_2 \in A$ such that $x_1 \neq x_2$ and for every $y_1, y_2 \in B$,*

$$|\{\ h \in H : y_1 = h(x_1), y_2 = h(x_2)\ \}| \leq \epsilon \frac{|H|}{|B|}.$$

We similarly define our ϵ-*fuzzy strongly universal*$_2$ hash families as follows.

Definition 4 ϵ-Fuzzy strongly universal$_2$ *(ϵ-FSU$_2$)*
Let (A, χ) be a finite fuzzy space, B be a finite set. Let H be a finite hash family mapping A to B. H is ϵ-fuzzy strongly universal$_2$ *if for every $x_1, x_2 \in A$ such that $\chi(x_1, x_2) = 0$ and for every $y_1, y_2 \in B$,*

$$|\{\ h \in H : y_1 = h(x_1), y_2 \in h(\chi_{x_2})\ \}| \leq \epsilon \frac{|H|}{|B|}.$$

When χ is the discrete fuzzy function then an ϵ-FSU_2 hash family is ϵ-SU_2.

Definitions of *ϵ-strongly differential universal$_2$* and the generalized *ϵ-fuzzy strongly differential universal$_2$* are as follows.

Definition 5 ϵ-Strongly differential universal$_2$ *(ϵ-SDU$_2$)*
Let A be a finite set, and $(B, +)$ be a finite commutative group. Let H be a finite hash family mapping A to B. H is ϵ-strongly differential universal$_2$ *if for every $x_1, x_2 \in A$ such that $x_1 \neq x_2$ and for every $y \in B$,*

$$|\{\ h \in H : y = h(x_1) - h(x_2)\ \}| \leq \epsilon |H|.$$

Definition 6 ϵ-Fuzzy strongly differential universal$_2$ *(ϵ-FSDU$_2$)*
Let (A, χ) be a finite fuzzy space, $(B, +)$ be a finite commutative group. Let H be a finite hash family mapping A to B. H is ϵ-fuzzy strongly differential universal$_2$ *if for every $x_1, x_2 \in A$ such that $\chi(x_1, x_2) = 0$ and for every $y \in B$,*

$$|\{\ h \in H : y + h(x_1) \in h(\chi_{x_2})\ \}| \leq \epsilon |H|.$$

The following lemma provides a lower bound for the parameter ϵ in ϵ-FSU_1, ϵ-FSU_2 and ϵ-FSDU_2.

Lemma 1 *If H is an ϵ-FSU$_1$ (or ϵ-FSU$_2$, ϵ-FSDU$_2$) hash family mapping A to B then*

$$\epsilon \geq \frac{1}{|B||H|} \max_{x \in A} \sum_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|} \max_{x \in A} \min_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|}.$$

Definitions of *ϵ-universal$_2$* and the generalized *ϵ-fuzzy universal$_2$* are as follows.

Definition 7 ϵ-Universal$_2$ *(ϵ-U$_2$)*
Let A, B be finite sets. Let H be a finite hash family mapping A to B. H is ϵ-universal$_2$ *if for every $x_1, x_2 \in A$ such that $x_1 \neq x_2$,*

$$|\{\ h \in H : h(x_1) = h(x_2)\ \}| \leq \epsilon |H|.$$

Definition 8 ϵ-Fuzzy universal$_2$ *(ϵ-FU$_2$)*
Let (A, χ) be a finite fuzzy space, and B be a finite set. Let H be a finite hash family mapping A to B. H is ϵ-fuzzy universal$_2$ *if for every $x_1, x_2 \in A$ such that $\chi(x_1, x_2) = 0$,*

$$|\{\ h \in H : h(x_1) \in h(\chi_{x_2})\ \}| \leq \epsilon |H|.$$

The following lemma provides a lower bound for the parameter ϵ in ϵ-FU_2.

Lemma 2 *If H is an ϵ-FU_2 hash family mapping A to B then*

$$\epsilon \geq \frac{|A|^2 - |B| \sum_{x \in A} |\chi_x|}{|B|(|A|^2 - \sum_{x \in A} |\chi_x|)} \geq \frac{|A| - |B||\chi|}{|B|(|A| - |\chi|)}.$$

When χ is the discrete fuzzy function then an ϵ-FU_2 hash family is ϵ-U_2. In this case, $|\chi| = 1$, and the lower bound in Lemma 2 becomes the familiar bound [18, 22] for ϵ-U_2,

$$\epsilon \geq \frac{|A| - |B|}{|B|(|A| - 1)}.$$

2.3 Relationship Between Different Classes of Hash Families

In traditional universal hashing [22], ϵ-SU_2 implies ϵ-SU_1, ϵ-SU_2 implies ϵ-SDU_2 and ϵ-SDU_2 implies ϵ-U_2. In the following theorem, we will show that the same implications hold in fuzzy universal hashing.

Theorem 1 *The following implications hold:*

1. ϵ-$FSU_2 \implies \epsilon$-FSU_1;
2. ϵ-$FSU_2 \implies \epsilon$-$FSDU_2$;
3. ϵ-$FSDU_2 \implies \epsilon$-FU_2.

The following theorem provides generic constructions of fuzzy universal hashing from traditional universal hashing.

Theorem 2 *The following implications hold:*

1. ϵ-$SU_1 \implies (\epsilon|\chi|)$-$FSU_1$;
2. ϵ-$SU_2 \implies (\epsilon|\chi|)$-$FSU_2$;
3. ϵ-$SDU_2 \implies (\epsilon|\chi|)$-$FSDU_2$;
4. ϵ-$U_2 \implies (\epsilon|\chi|)$-$FU_2$.

Example. In the *evaluation hash* [19], the set of messages A is the set of all polynomials of degree less than n over $GF(2^\ell)$ that we will denoted by $GF(2^\ell)[t]_n$. Each hash function is indexed by an element of $GF(2^\ell)$. The hash value of a hash function $\alpha \in GF(2^\ell)$ for a message $P \in GF(2^\ell)[t]_n$ is $\alpha P(\alpha)$. It is easy to see that the described hash family is $\frac{n}{2^\ell}$-SDU_2. That is because, for any $y \in GF(2^\ell)$, and for any two different polynomials $P_1, P_2 \in GF(2^\ell)[t]_n$, there are at most n values for $\alpha \in GF(2^\ell)$ such that $y = \alpha P_1(\alpha) - \alpha P_2(\alpha)$. These values α are the roots of the polynomial $Q(t) = y - tP_1(t) + tP_2(t)$ with $1 \leq \deg Q \leq n$.

Suppose now we define a fuzzy function on the message space $GF(2^\ell)[t]_n$ as follows. Two polynomials $P_1, P_2 \in GF(2^\ell)[t]_n$ are considered as "indistinguishable", i.e. $\chi(P_1, P_2) = 1$, if and only if, P_1 and P_2 are different in at most one coefficient and the different value of this coefficient is at

most 1. For example $\chi(t^2+3t+1, t^2+2t+1) = 1$, whereas $\chi(t^2+3t+1, t^2+2t+2) = 0$ and $\chi(t^2+3t+1, t^2+4t+1) = 0$.

With the above fuzzy function, $|\chi| = 2n+1$, hence, by Theorem 2(3), the evaluation hash family is $\frac{n(2n+1)}{2^\ell}$-FSDU_2.

2.4 Constructions of Fuzzy Hashing

From a fuzzy function χ on A, we can naturally define a new fuzzy function χ^ℓ on A^ℓ as follows,

$$\chi^\ell((a_1,\ldots,a_\ell),(b_1,\ldots,b_\ell)) = \chi(a_1,b_1)\times\cdots\times\chi(a_\ell,b_\ell).$$

That is two tuples $(a_1,\ldots,a_\ell),(b_1,\ldots,b_\ell) \in A^\ell$ are indistinguishable if all of their corresponding components are indistinguishable. We have $|\chi^\ell| = |\chi|^\ell$.

Suppose we have a fuzzy hash family $H = \{h : (A,\chi) \to B\}$. Consider the following constructions of fuzzy hash families mapping $(A^\ell,\chi^\ell) \to B^\ell$, $(A,\chi) \to B^\ell$, $(A^\ell,\chi^\ell) \to B$ and $(A^{\ell_1},\chi^{\ell_1}) \to B^{\ell_2}$.

Construction 1: $H : (A^\ell,\chi^\ell) \to B^\ell$
For each h in $H = \{h : (A,\chi) \to B\}$ we define a function mapping $(A^\ell,\chi^\ell) \to B^\ell$ as follows

$$h((a_1,\ldots,a_\ell)) = (h(a_1),\ldots,h(a_\ell)).$$

Construction 2: $H^\ell : (A^\ell,\chi^\ell) \to B^\ell$
For each $(h_1,\ldots,h_\ell) \in H^\ell$ we define a function mapping $(A^\ell,\chi^\ell) \to B^\ell$ as follows

$$(h_1,\ldots,h_\ell)((a_1,\ldots,a_\ell)) = (h_1(a_1),\ldots,h_\ell(a_\ell)).$$

Construction 3: $H^\ell : (A^\ell,\chi^\ell) \to B$
For each $(h_1,\ldots,h_\ell) \in H^\ell$ we define a function mapping $(A^\ell,\chi^\ell) \to B$ as follows

$$(h_1,\ldots,h_\ell)((a_1,\ldots,a_\ell)) = h_1(a_1)+\cdots+h_\ell(a_\ell).$$

Construction 4: $H^\ell : (A,\chi) \to B^\ell$
For each $(h_1,\ldots,h_\ell) \in H^\ell$ we define a function mapping $(A,\chi) \to B^\ell$ as follows

$$(h_1,\ldots,h_\ell)(a) = (h_1(a),\ldots,h_\ell(a)).$$

Construction 5: $H^\ell : (A,\chi) \to B$
For each $(h_1,\ldots,h_\ell) \in H^\ell$ we define a function mapping $(A,\chi) \to B$ as follows

$$(h_1,\ldots,h_\ell)(a) = h_1(a)+\cdots+h_\ell(a).$$

We show that these constructions preserve certain fuzzy properties.

Theorem 3 *If $H : (A,\chi) \to B$ is ϵ-FSU$_1$ (ϵ-FSU$_2$, ϵ-FSDU$_2$, ϵ-FU$_2$) then*

- *in the* Construction 1, *$H : (A^\ell,\chi^\ell) \to B^\ell$ is ϵ-FSU$_1$ ($\epsilon|B|^{\ell-1}$-FSU$_2$, ϵ-FSDU$_2$, ϵ-FU$_2$);*
- *in the* Construction 2, *$H^\ell : (A^\ell,\chi^\ell) \to B^\ell$ is ϵ^ℓ-FSU$_1$ ($\epsilon^\ell|B|^{\ell-1}$-FSU$_2$, ϵ-FSDU$_2$, ϵ-FU$_2$);*
- *in the* Construction 3, *$H^\ell : (A^\ell,\chi^\ell) \to B$ is $\epsilon|\chi|^{\ell-1}$-FSU$_1$ ($\epsilon|\chi|^{\ell-1}$-FSU$_2$, $\epsilon|\chi|^{\ell-1}$-FSDU$_2$);*
- *in the* Construction 4, *$H^\ell : (A,\chi) \to B^\ell$ is ϵ^ℓ-FSU$_1$ (ϵ^ℓ-FSU$_2$, ϵ^ℓ-FSDU$_2$, ϵ^ℓ-FU$_2$);*
- *in the* Construction 5, *$H^\ell : (A,\chi) \to B$ is $\epsilon|\chi|$-FSU$_1$ ($\epsilon|\chi|$-FSU$_2$, $\epsilon|\chi|$-FSDU$_2$).*

2.5 Fuzzy universal hashing for arbitrary-length messages

In this section, we follow Black et al [5, 6] technique and show how to construct an ϵ-*fuzzy universal*$_2$ hash family for arbitrary-length messages. Suppose $A = \{0,1\}^a$ is a fixed length message space with a fuzzy function $\chi : A \times A \to \{0,1\}$. Suppose $B = \{0,1\}^b$ and H is an ϵ-fuzzy universal$_2$ hash family mapping A to B. We first define a new fuzzy function δ on the set $\{0,1\}^*$ of all strings of arbitrary length that extends the old fuzzy function χ. With this new fuzzy function δ on $\{0,1\}^*$, we then use H to define a new ϵ-FU_2 hash family $H^* = \{h^* : \{0,1\}^* \to \{0,1\}^*\}$.

For a string x, let $|x|$ denote the length of x. Let $\mathsf{len}_a(x)$ denote the value ($|x| \mod a$) encoded as $\lceil \log_2 a \rceil$-bit string. If $|x| \leq a$ then let $\mathsf{pad}_a(x)$ denote the string of length a obtained from the string x by zero-padding.

FUZZY FUNCTION δ ON $\{0,1\}^*$ OBTAINED FROM FUZZY FUNCTION χ ON $\{0,1\}^a$.
If $m, m' \in \{0,1\}^*$ are two strings of different lengths then we define $\delta(m,m') = 0$. If $0 < |m| = |m'| < a$ then we define $\delta(m,m') = \chi(\mathsf{pad}_a(m), \mathsf{pad}_a(m'))$. Finally, if $|m| = |m'| \geq a$, write $m = m_1||\ldots||m_{k-1}||m_k$ and $m' = m'_1||\ldots||m'_{k-1}||m'_k$ where $|m_1| = \cdots = |m_{k-1}| = |m'_1| = \cdots = |m'_{k-1}| = a$, $0 < |m_k|, |m'_k| \leq a$, and define

$$\delta(m,m') = \chi(m_1,m'_1) \times \cdots \times \chi(m_{k-1},m'_{k-1}) \times \chi(\mathsf{pad}_a(m_k), \mathsf{pad}_a(m'_k)).$$

The new fuzzy function δ is equal to the old fuzzy function χ when restricted to $\{0,1\}^a$.

ϵ-FU_2 HASH FAMILY $H^* = \{h^* : (\{0,1\}^*, \delta) \to \{0,1\}^*\}$ OBTAINED FROM ϵ-FU_2 HASH FAMILY $H = \{h : (\{0,1\}^a, \chi) \to \{0,1\}^b\}$.
Each function $h \in H$ corresponds to a function $h^* \in H^*$ defined as follows,

$$h^*(m) = h(m_1)||\ldots||h(m_{k-1})||h(\mathsf{pad}_a(m_k))||\mathsf{len}_a(m),$$

where $m = m_1||\ldots||m_{k-1}||m_k$ with $|m_1| = \cdots = |m_{k-1}| = a$ and $0 < |m_k| \leq a$.

Theorem 4 *If $H = \{h : (\{0,1\}^a, \chi) \to \{0,1\}^b\}$ is ϵ-FU_2 then the above described $H^* = \{h^* : (\{0,1\}^*, \delta) \to \{0,1\}^*\}$ is also ϵ-FU_2.*

3 Application to Approximate Authentication

In this section, we discuss the application of fuzzy universal hashing in approximate authentication for multiple messages.

In traditional authentication, universal hashing has been used to construct a secure MAC. Similarly, we will show that our fuzzy hashing can be used to construct secure approximate authentication for multiple messages. Our construction of approximate authentication follows the Carter-Wegman framework [10, 23].

The sender and the receiver share two things: a random chosen hash function h from a fuzzy universal hash family and a secret encryption key. To send a message m, the sender associates it with a message number i, and makes sure that no two different messages associated with the same message number. To generate the authentication tag t for the message m, the sender then hashes the message m and then encrypts the resulting hash using i and the secret encryption key. We will use the one-time pad encryption, that is $t = h(m) + r_i$, where r_i is the ith part of the encryption

key. With this choice, if we want to authenticate up to n messages then the sender and the receiver must share a secret encryption key as a tuple $(r_1, \ldots, r_n)$ of n components.

Another method is to use pseudorandom function families (PRF). In this method, the sender and the receiver share a random chosen function h from a fuzzy universal hash family and a random chosen function f from a pseudorandom function family. To generate the authentication tag t, instead of encrypting the resulting hash $h(m)$, we encode it with the message number i to obtain $\langle h(m), i\rangle$, and then apply the function f and set $t = f(\langle h(m), i\rangle)$.

The tuple (m, i, t) is sent to the receiver who actually receives (m', i, t) for some m' that is indistinguishable to m. The receiver will accept (m', i, t) as authenticated if $t = \text{TAG}_{(secret)}(m, i)$ for some $m \in \chi_{m'}$ where $\text{TAG}_{(secret)}$ denotes the tagging algorithm.

In section 3.1 we discuss one-time pad based approximate authentication scheme and in section 3.2 we discuss PRF-based scheme. We show that in the one-time pad based scheme, if an ϵ-*fuzzy strongly differential universal*$_2$ hash family is used (or an ϵ-*fuzzy strongly universal*$_2$ is used) then the one-time pad based scheme is secure against forgery attack with the probability of $1-\epsilon$. In the PRF-based scheme, if an ϵ-*fuzzy universal*$_2$ hash family is used then the PRF-based scheme is secure against forgery attack with the probability of $1-\epsilon$. This security is proved in *information-theoretical* sense.

3.1 One-time pad based approximate authentication

Let (A, χ) be a finite fuzzy space and $(B, +)$ be a finite commutative group. Let H be a finite hash family mapping A to B. We will use H to construct an approximate authentication scheme FMAC$[H, n]$ where the parameter n denotes the maximum number of messages to authenticate. The message space is A, the authenticator space is B. The scheme FMAC$[H, n]$ = (KEY, TAG, VERIFY) is defined as follows:

```
function KEY
    h ← H
    (r_1, ..., r_n) ← B^n
    return (h, r_1, ..., r_n)
```

```
function TAG_(h,r_1,...,r_n)(m, counter)

    return h(m) + r_counter
```

```
function VERIFY_(h,r_1,...,r_n)(m, counter, t)
    if counter ∈ {1, ..., n} and (t − r_counter) ∈ h(χ_m) then
        return 1 (accept)
    end if
    return 0 (reject)
```

In the FMAC$[H, n]$ scheme, the sender and the receiver share a random chosen function $h \in H$ and a random tuple $(r_1, \ldots, r_n) \in B^n$. To send a message m, the sender associates it with a message number $counter \in \{1, \ldots, n\}$, and makes sure that no two different messages associated with the same message number. The authentication tag t for the message m is generated as $t = h(m) + r_{counter}$. The receiver receives $(m, counter, t)$ and checks if $t - r_{counter} = h(m')$ for some $m' \in \chi_m$ to accept it as an authenticated message. This requires up to $|\chi|$ number of checking.

Forgery Attack Model. Let FMAC$[H, n]$ = (KEY, TAG, VERIFY) be an approximate authentication scheme. Let c be an integer such that $0 \le c \le n$. Let $\mathcal{A}$ be an adversary with access to the

tagging oracle TAG. Consider the following experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c)$. In this experiment, first, we execute the function KEY to obtain a random key $(h, r_1, \ldots, r_n) \in H \times B^n$. Then the adversary $\mathcal{A}$ adaptively queries the tagging oracle TAG with c pairs $(m_1, counter_1)$, $\ldots$, $(m_c, counter_c)$, with the following restriction: no two different messages $m_t \neq m_s$ associated with the same message number $counter_t = counter_s$. The tagging oracle answers each query $(m_i, counter_i)$ with the value $t_i = \mathrm{TAG}_{(h,r_1,\ldots,r_n)}(m_i, counter_i)$. Finally, the adversary $\mathcal{A}$ presents a forgery $(m, counter, t)$. We consider

- $\mathcal{A}$ wins of type 1 if $\mathrm{VERIFY}_{(h,r_1,\ldots,r_n)}(m, counter, t) = 1$, $counter = counter_j$ for some $1 \leq j \leq c$, and $\chi(m, m_j) = 0$; and
- $\mathcal{A}$ wins of type 2 if $\mathrm{VERIFY}_{(h,r_1,\ldots,r_n)}(m, counter, t) = 1$ and $counter \notin \{counter_1, \ldots, counter_c\}$.

Let $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type1)$ and $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type2)$ denote the probability for $\mathcal{A}$ to win of type 1 and type 2, respectively, in the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c)$. We have the following bounds on these probabilities.

Lemma 3 *If H is ϵ-FSDU_2 (or ϵ-FSU_2) then*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type1) \leq \epsilon.$$

Lemma 4 *For any H,*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type2) \leq \frac{1}{|B||H|} \max_{x \in A} \sum_{h \in H} |h(\chi_x)|.$$

It follows from Lemma 3, Lemma 4 and Lemma 1 that, if H is ϵ-FSDU_2 (or ϵ-FSU_2), then the probability, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c)$, for $\mathcal{A}$ to win in the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c)$ is bounded above by ϵ as stated in the following theorem.

Theorem 5 *If H is ϵ-FSDU_2 (or ϵ-FSU_2) then*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c) \leq \epsilon.$$

3.2 PRF based approximate authentication

This section generalizes Black et al's [5, 6] construction of secure PRF-based MAC. Black et al show that from an ϵ-*universal*$_2$ hash family we can construct a secure PRF-based MAC scheme. We show that from an ϵ-*fuzzy universal*$_2$ hash family we can construct a secure PRF-based *approximate* authentication scheme.

Let H be an ϵ-*fuzzy universal*$_2$ hash family mapping $A \to B$. Let F be a pseudorandom function family mapping $\{0,1\}^* \to \{0,1\}^\tau$. Using H and F, we construct an approximate authentication scheme $\mathrm{FMAC}[H, F] = (\mathrm{KEY}, \mathrm{TAG}, \mathrm{VERIFY})$ as follows. Note that $\langle \cdot, \cdot \rangle$ denotes a linear-time computable function that maps a string $x \in \{0,1\}^*$ and a string $y \in \{0,1\}^\nu$ into a string $\langle x, y \rangle$ of length $|x| + \nu + O(1)$, which can be used to recover x and y in linear-time.

```
function KEY
   h ← H
   f ← F
   return (h, f)
```

```
function TAG_(h,f)(m, counter)
   return f(⟨h(m), counter⟩)
```

```
function VERIFY_(h,f)(m, counter, t)
   if counter ∈ {1, ..., n} and t = f(⟨h(m'), counter⟩) for some m' ∈ χ_m then
      return 1 (accept)
   end if
   return 0 (reject)
```

In the FMAC$[H, F]$ scheme, the sender and the receiver share a random chosen function $h \in H$ and a random chosen function $f \in F$. To send a message m, the sender associates it with a message number $counter \in \{0,1\}^\nu$, and makes sure that no two different messages associated with the same message number. To generate the authentication tag t for the message m, the sender hashes the message m, encodes the resulting hash $h(m)$ with the message number $counter$ to obtain $\langle h(m), counter\rangle$, and then applies the function f to get $t = f(\langle h(m), counter\rangle)$. The receiver receives $(m, counter, t)$ and checks if $t = f(\langle h(m'), counter\rangle)$ for some $m' \in \chi_m$ to accept it as an authenticated message. This requires up to $|\chi|$ number of checking.

Forgery Attack Model. Let FMAC$[H, F]$ = (KEY, TAG, VERIFY) be an approximate authentication scheme. Let c be an integer. Let $\mathcal{A}$ be an adversary with access to the tagging oracle TAG. Consider the following experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c)$. In this experiment, first, we execute the function KEY to obtain a random key $(h, f) \in H \times F$. Then the adversary $\mathcal{A}$ adaptively queries the tagging oracle TAG with c pairs $(m_1, counter_1), \ldots, (m_c, counter_c)$, with the following restriction: no two different messages $m_t \neq m_s$ associated with the same message number $counter_t = counter_s$. The tagging oracle answers each query $(m_i, counter_i)$ with the value $t_i = \mathrm{TAG}_{(h,f)}(m_i, counter_i)$. Finally, the adversary $\mathcal{A}$ presents a forgery $(m, counter, t)$. We consider

- $\mathcal{A}$ wins of type 1 if $\mathrm{VERIFY}_{(h,f)}(m, counter, t) = 1$, $counter = counter_j$ for some $1 \leq j \leq c$, and $\chi(m, m_j) = 0$; and
- $\mathcal{A}$ wins of type 2 if $\mathrm{VERIFY}_{(h,f)}(m, counter, t) = 1$ and $counter \notin \{counter_1, \ldots, counter_c\}$.

Let $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c, Type1)$ and $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c, Type2)$ denote the probability for $\mathcal{A}$ to win of type 1 and type 2, respectively, in the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c)$. Let $\mathsf{Rand}(\tau)$ be the set of all functions mapping $\{0,1\}^* \to \{0,1\}^\tau$, so that choosing a random $f \leftarrow \mathsf{Rand}(\tau)$ means associating to each string $x \in \{0,1\}^*$ a random string $f(x) \in \{0,1\}^\tau$. We have the following bounds.

Lemma 5 *If H is ϵ-FU_2 then*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,\mathsf{Rand}(\tau)]}(c, Type1) \leq \max(\epsilon, 2^{-\tau}).$$

Lemma 6

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,\mathsf{Rand}(\tau)]}(c, Type2) \leq 2^{-\tau}.$$

From Lemma 5 and Lemma 6, we have the following bound on the probability, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c)$, for $\mathcal{A}$ to win in the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[H,F]}(c)$.

Theorem 6 *If H is ϵ-FU_2 then*

$$\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,\mathsf{Rand}(\tau)]}(c) \leq \max(\epsilon, 2^{-\tau}).$$

References

[1] M. Atici, D. R. Stinson, Universal Hashing and Multiple Authentication, CRYPTO'96, Lecture Notes in Computer Science 1109, pp. 16–30, 1996.

[2] M. Bellare, R. Canetti, H. Krawczyk, Keying Hash Functions for Message Authentication, CRYPTO'96, Lecture Notes in Computer Science 1109, pp. 1–15, 1996.

[3] M. Bellare, J. Kilian, P. Rogaway, The Security of the Cipher Block Chaining Message Authentication Code, Journal of Computer and System Sciences, Volume 61, Issue 3 (2000), pp. 362-399.

[4] D. J. Bernstein, Stronger Security Bounds for Wegman-Carter-Shoup Authenticators, EUROCRYPT'05, Lecture Notes in Computer Science 3494, pp. 164–180.

[5] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, UMAC: Fast and Secure Message Authentication, CRYPTO'99, Lecture Notes in Computer Science 1666, pp. 216–233, 1999.

[6] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, *UMAC: Fast and Secure Message Authentication,* 1999. URL: `http://www.cs.ucdavis.edu/~rogaway/umac/`.

[7] R. Bolle, J. Connell, S. Pankanti, N. Ratha, A. Senior, *Guide to Biometrics*, Springer-Verlag, 2003.

[8] X. Boyen, Reusable cryptographic fuzzy extractors, CCS'04, 2004.

[9] X. Boyen, Y. Dodis, J. Katz, R. Ostrovsky, A. Smith, Secure Remote Authentication Using Biometric Data, EUROCRYPT'05, Lecture Notes in Computer Science 3494, pp. 147–163.

[10] J. L. Carter, M. N. Wegman, Universal classes of hash functions, Journal of Computer and System Sciences, Volume 18, Issue 2 (1979), pp. 143–154.

[11] Y. Dodis, L. Reyzin, A. Smith, Fuzzy Extractors: How to Generate Strong Keys from Biometrics and Other Noisy Data, EUROCRYPT'04, Lecture Notes in Computer Science 3027, pp. 523–540, 2004.

[12] C. Ellison, C. Hall, R. Milbert, B. Schneier, Protecting secret keys with personal entropy, *Future Generation Computer Systems* **16** (4) (2000), pp. 311–318.

[13] J. Fridrich, M. Goljan, Robust hash functions for digital watermarking, *Proc. IEEE Int. Conf. on Information Technology, Coding and Computing*, pp. 178–183, 2000.

[14] C.-Y. Hsu, C.-S. Lu, Geometric distortion-resilient image hashing system and its application scalability, *Proc. 2004 Workshop on Multimedia and Security*, pp. 81–92, 2004.

[15] H. Krawczyk, LFSR-Based Hashing and Authentication, CRYPTO'94, Lecture Notes in Computer Science 839, pp. 129–139, 1994.

[16] H. Krawczyk, New Hash Functions for Message Authentication, EUROCRYPT'95, Lecture Notes in Computer Science 921, pp. 301–310, 1995.

[17] P. Rogaway, Bucket Hashing and Its Application to Fast Message Authentication, CRYPTO'95, Lecture Notes in Computer Science 963, pp. 29–42, 1995.

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

[19] V. Shoup, On Fast and Provably Secure Message Authentication Based on Universal Hashing, CRYPTO'96, Lecture Notes in Computer Science 1109, pp. 313–328, 1996.

[20] D. R. Stinson, Universal Hashing and Authentication Codes, CRYPTO'91, Lecture Notes in Computer Science 576, pp. 74–85, 1991.

[21] D. R. Stinson, Universal Hashing and Authentication Codes, *Designs, Codes and Cryptography* **4** (1994), pp. 369–380.

[22] D. R. Stinson, On the Connections Between Universal Hashing, Combinatorial Designs and Error-Correcting Codes, *ECCC Report TR95-052.* URL: `http://www.eccc.uni-trier.de/eccc-reports/1995/TR95-052`.

[23] M. N. Wegman, J. L. Carter, New hash functions and their use in authentication and set equality, Journal of Computer and System Sciences, Volume 22, Issue 3 (1981), pp. 265–279.

Appendix

Proof of Lemma 1. Part 1. Assume that H is ϵ-FSU_1. Fix $x \in A$. Consider $|H|$ subsets of B,

$$h(\chi_x), \quad h \in H.$$

Since H is ϵ-FSU_1, each $y \in B$ is in at most $\epsilon|H|$ number of these subsets, therefore,

$$\sum_{h \in H} |h(\chi_x)| \leq \epsilon |H||B|.$$

Thus,

$$\epsilon \geq \frac{1}{|B||H|} \sum_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|} \min_{h \in H} |h(\chi_x)|.$$

This holds for every $x \in A$, so

$$\epsilon \geq \frac{1}{|B||H|} \max_{x \in A} \sum_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|} \max_{x \in A} \min_{h \in H} |h(\chi_x)|.$$

Part 2. Assume that H is ϵ-FSU_2. Fix $x, x' \in A$ such that $\chi(x, x') = 0$. We count the number of tuples $(h, y, y') \in H \times B^2$ such that $y' = h(x')$ and $y \in h(\chi_x)$ by two ways.

In the first way, we choose the function h first. Once h is chosen, y' is specified as $y' = h(x')$ and there are $|h(\chi_x)|$ choices for y. So the number of questioning tuples is

$$\sum_{h \in H} |h(\chi_x)|.$$

In the second way, we choose y and y' first. There are $|B|^2$ choices for (y, y'). Once (y, y') is chosen, since H is ϵ-FSU_2, there are no more than $\epsilon \frac{|H|}{|B|}$ choices for h. So the number of questioning tuples is no more than $|B|^2 \epsilon \frac{|H|}{|B|} = \epsilon |B||H|$.

Therefore,

$$\epsilon |B||H| \geq \sum_{h \in H} |h(\chi_x)|,$$

and

$$\epsilon \geq \frac{1}{|B||H|} \sum_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|} \min_{h \in H} |h(\chi_x)|.$$

This holds for every $x \in A$, so

$$\epsilon \geq \frac{1}{|B||H|} \max_{x \in A} \sum_{h \in H} |h(\chi_x)| \geq \frac{1}{|B|} \max_{x \in A} \min_{h \in H} |h(\chi_x)|.$$

Part 3. Assume that H is ϵ-FSDU_2. Fix $x, x' \in A$ such that $\chi(x, x') = 0$. We count the number of tuples $(h, y) \in H \times B$ such that $y + h(x') \in h(\chi_x)$ by two ways.

In the first way, we choose the function h first. Once h is chosen, there are $|h(\chi_x)|$ choices for y as $y \in h(\chi_x) - h(x')$. So the number of questioning tuples is

$$\sum_{h \in H} |h(\chi_x)|.$$

In the second way, we choose y first. There are $|B|$ choices for y. Once y is chosen, since H is ϵ-FSDU_2, there are no more than $\epsilon|H|$ choices for h. So the number of questioning tuples is no more than $|B|\epsilon|H|$.

Therefore,

$$|B|\epsilon|H| \geq \sum_{h\in H} |h(\chi_x)|,$$

and

$$\epsilon \geq \frac{1}{|B||H|}\sum_{h\in H} |h(\chi_x)| \geq \frac{1}{|B|}\min_{h\in H} |h(\chi_x)|.$$

This holds for every $x \in A$, so

$$\epsilon \geq \frac{1}{|B||H|}\max_{x\in A}\sum_{h\in H} |h(\chi_x)| \geq \frac{1}{|B|}\max_{x\in A}\min_{h\in H} |h(\chi_x)|.$$

Proof of Lemma 2. Assume that H is ϵ-FU_2. Fix $x \in A$. We count the number of tuples $(h, x') \in H \times A$ such that $\chi(x, x') = 0$ and $h(x') \in h(\chi_x)$ by two ways.

In the first way, we choose the function h first. Once h is chosen, x' must be in the set $h^{-1}(h(\chi_x)) \setminus \chi_x$. So the number of questioning tuples is

$$\sum_{h\in H} |h^{-1}(h(\chi_x))| - |H||\chi_x|.$$

In the second way, we choose x' first. There are $|A| - |\chi_x|$ choices for x' since $x' \notin \chi_x$. Once x' is chosen, since H is ϵ-FU_2, there are no more than $\epsilon|H|$ choices for h. So the number of questioning tuples is no more than $\epsilon|H|(|A| - |\chi_x|)$.

Therefore,

$$\epsilon|H|(|A| - |\chi_x|) \geq \sum_{h\in H} |h^{-1}(h(\chi_x))| - |H||\chi_x| \geq \sum_{h\in H} |h^{-1}(h(x))| - |H||\chi_x|,$$

and

$$\epsilon|H|(|A| - |\chi_x|) + |H||\chi_x| \geq \sum_{h\in H} |h^{-1}(h(x))|.$$

This holds for every $x \in A$, so

$$\sum_{x\in A} (\epsilon|H|(|A| - |\chi_x|) + |H||\chi_x|) \geq \sum_{x\in A}\sum_{h\in H} |h^{-1}(h(x))|.$$

By Schwarz's inequality,

$$\sum_{x\in A} |h^{-1}(h(x))| = \sum_{y\in B} |h^{-1}(y)|^2 \geq \frac{1}{|B|}\left(\sum_{y\in B} |h^{-1}(y)|\right)^2 = \frac{|A|^2}{|B|},$$

hence,

$$\sum_{x\in A}\sum_{h\in H} |h^{-1}(h(x))| = \sum_{h\in H}\sum_{x\in A} |h^{-1}(h(x))| \geq \frac{|H||A|^2}{|B|}.$$

On the other hand, we have

$$\sum_{x\in A}(\epsilon|H|(|A|-|\chi_x|)+|H||\chi_x|)=\epsilon|H|\left(|A|^2-\sum_{x\in A}|\chi_x|\right)+|H|\sum_{x\in A}|\chi_x|,$$

so

$$\epsilon|H|\left(|A|^2-\sum_{x\in A}|\chi_x|\right)+|H|\sum_{x\in A}|\chi_x|\geq\frac{|H||A|^2}{|B|}.$$

Since $1\leq|\chi_x|\leq|\chi|<|A|$, we have $|A|\leq\sum_{x\in A}|\chi_x|\leq|\chi||A|<|A|^2$, and

$$\epsilon\geq\frac{|A|^2-|B|\sum_{x\in A}|\chi_x|}{|B|(|A|^2-\sum_{x\in A}|\chi_x|)}\geq\frac{|A|^2-|B||\chi||A|}{|B|(|A|^2-|\chi||A|)}=\frac{|A|-|B||\chi|}{|B|(|A|-|\chi|)}.$$

Proof of Theorem 1.
Part 1. ϵ-FS$U_2 \implies \epsilon$-FSU_1. Assume that H is ϵ-FSU_2, we will show that H is ϵ-FSU_1.

Fix $y_2\in B$ and fix $x_1,x_2\in A$ such that $\chi(x_1,x_2)=0$. For each $y_1\in B$, consider the following subset of H,

$$H_{y_1}=\{\ h\in H: y_1=h(x_1), y_2\in h(\chi_{x_2})\ \}.$$

These subsets H_{y_1} are pairwise disjoint and

$$\bigcup_{y_1\in B}H_{y_1}=\{\ h\in H: y_2\in h(\chi_{x_2})\ \}.$$

Therefore,

$$|\{\ h\in H: y_2\in h(\chi_{x_2})\ \}|=\sum_{y_1\in B}|H_{y_1}|.$$

Since H is ϵ-FSU_2, for every $y_1\in B$, $|H_{y_1}|\leq\epsilon\frac{|H|}{|B|}$. Thus,

$$|\{\ h\in H: y_2\in h(\chi_{x_2})\ \}|\leq\epsilon|H|.$$

This holds for every $x_2\in A$ and every $y_2\in B$, therefore, H is ϵ-FSU_1.

Part 2. ϵ-FS$U_2 \implies \epsilon$-FSDU_2. Assume that H is ϵ-FSU_2, we will show that H is ϵ-FSDU_2.

Fix $y\in B$ and fix $x_1,x_2\in A$ such that $\chi(x_1,x_2)=0$. For each $y_1\in B$, consider the following subset of H,

$$H_{y_1}=\{\ h\in H: y_1=h(x_1), y+y_1\in h(\chi_{x_2})\ \}.$$

These subsets H_{y_1} are pairwise disjoint and

$$\bigcup_{y_1\in B}H_{y_1}=\{\ h\in H: y+h(x_1)\in h(\chi_{x_2})\ \}.$$

Therefore,

$$|\{\ h\in H: y+h(x_1)\in h(\chi_{x_2})\ \}|=\sum_{y_1\in B}|H_{y_1}|.$$

Since H is ϵ-FSU_2, for every $y_1\in B$, $|H_{y_1}|\leq\epsilon\frac{|H|}{|B|}$. Thus,

$$|\{\ h\in H: y+f(x_1)\in h(\chi_{x_2})\ \}|\leq\epsilon|H|.$$

This holds for every $y \in B$ and for every $x_1, x_2 \in A$ such that $\chi(x_1, x_2) = 0$, therefore, H is ϵ-FSDU_2.

Proof of Theorem 4. Let $m, x \in \{0,1\}^*$ such that $\delta(m, x) = 0$. We prove that $Pr_{h^* \in H^*}[h^*(x) \in h^*(\delta_m)] \leq \epsilon$.

Suppose there exist $h^* \in H^*$ and $m' \in \chi_m$ such that $h^*(x) = h^*(m')$. Then, since $|h^*(x)| = |h^*(m')|$, we must have $\lceil |x|/a \rceil = \lceil |m'|/a \rceil$. The last $\lceil \log_2 a \rceil$-bit strings of $h^*(x)$ and $h^*(m')$ equal implies that $|x| = |m'| \pmod a$. Thus, $|x| = |m'| = |m|$. Therefore, if $|x| \neq |m|$ then $Pr_{h^* \in H^*}[h^*(x) \in h^*(\delta_m)] = 0$.

Assume that $|x| = |m|$, write $x = x_1 || \ldots || x_{k-1} || x_k$ and $m = m_1 || \ldots || m_{k-1} || m_k$ where $|x_1| = \cdots = |x_{k-1}| = |m_1| = \cdots = |m_{k-1}| = a$ and $0 < |x_k|, |m_k| \leq a$. Since $\delta(x, m) = 0$, we have $\chi(x_i, m_i) = 0$ for some $i \in [1, k-1]$ or $\chi(\mathsf{pad}_a(x_k), \mathsf{pad}_a(m_k)) = 0$.

For each $h^* \in H^*$, if $h^*(x) \in h^*(\delta_m)$ then for the corresponding $h \in H$, we have $h(x_i) \in h(\chi_{m_i})$ and $h(\mathsf{pad}_a(x_k)) \in h(\chi_{\mathsf{pad}_a(m_k)})$. Thus,

$$Pr_{h^* \in H^*}[h^*(x) \in h^*(\delta_m)] \leq \min(Pr_{h \in H}[h(x_i) \in h(\chi_{m_i})], Pr_{h \in H}[h(\mathsf{pad}_a(x_k)) \in h(\chi_{\mathsf{pad}_a(m_k)})]),$$

and therefore, it follows that $Pr_{h^* \in H^*}[h^*(x) \in h^*(\delta_m)] \leq \epsilon$ since H is ϵ-FU_2.

Proof of Lemma 3. We give the proof for the case H is ϵ-FSDU_2. The case H is ϵ-FSU_2 can be proved similarly.

We can assume that $\mathcal{A}$ queries the tagging algorithm with c pairs $(m_1, counter_1), \ldots, (m_c, counter_c)$, such that c message numbers $counter_1, \ldots, counter_c$ are all distinct. The tagging algorithm answers $\mathcal{A}$ with $t_1, \ldots, t_c$. Finally, $\mathcal{A}$ presents a forgery $(m, counter_j, t)$ and wins of type 1. Consider two sets:

$$K = \{(h, r_1, \ldots, r_n) \in H \times B^n : \textsc{Tag}_{(h,r_1,\ldots,r_n)}(m_i, counter_i) = t_i, \forall i = 1, \ldots, c\}$$

and

$$K' = \{(h, r_1, \ldots, r_n) \in H \times B^n \ : \ \textsc{Tag}_{(h,r_1,\ldots,r_n)}(m_i, counter_i) = t_i, \forall i = 1, \ldots, c, \text{ and } \textsc{Verify}_{(h,r_1,\ldots,r_n)}(m, counter_j, t) = 1 \ \}.$$

Then $\mathbf{Adv}^{\mathrm{FMAC}[H,n]}_{\mathcal{A}}(c, Type1) = \max \frac{|K'|}{|K|}$, where the maximum is taken over all choices of $m_1, \ldots, m_c \in A$, distinct $counter_1, \ldots, counter_c \in \{1, \ldots, n\}$, $t_1, \ldots, t_c \in B$, $j \in \{1, \ldots, c\}$, $m \in A$, $t \in B$ such that $\chi(m, m_j) = 0$.

It is easy to see that $|K| = |H||B|^{n-c}$. Now we count the number of tuples $(h, r_1, \ldots, r_n)$ that belong to the set K'. The condition $\textsc{Verify}_{(h,r_1,\ldots,r_n)}(m, counter_j, t) = 1$ requires $t - r_{counter_j} \in h(\chi_m)$, and the condition $\textsc{Tag}_{(h,r_1,\ldots,r_n)}(m_j, counter_j) = t_j$ requires $r_{counter_j} = t_j - h(m_j)$. So h must be chosen so that $(t - t_j) + h(m_j) \in h(\chi_m)$. Since H is ϵ-FSDU_2 and $\chi(m, m_j) = 0$, there are at most $\epsilon|H|$ choices for h. Once h is chosen, $r_{counter_1}, \ldots, r_{counter_c}$ are specified as $r_{counter_i} = t_i - h(m_i)$. There are $|B|^{n-c}$ choices for $(r_1, \ldots, \hat{r}_{counter_1}, \ldots, \hat{r}_{counter_c}, \ldots, r_n)$. So, $|K'| \leq \epsilon|H||B|^{n-c}$ and $\frac{|K'|}{|K|} \leq \epsilon$. Therefore, $\mathbf{Adv}^{\mathrm{FMAC}[H,n]}_{\mathcal{A}}(c, Type1) \leq \epsilon$.

Proof of Lemma 4. We can assume that $\mathcal{A}$ queries the tagging algorithm with c pairs $(m_1, counter_1)$, $\ldots$, $(m_c, counter_c)$, such that c message numbers $counter_1, \ldots, counter_c$ are all distinct. The

tagging algorithm answers $\mathcal{A}$ with $t_1, \ldots, t_c$. Finally, $\mathcal{A}$ presents a forgery $(m, counter, t)$ with $counter \notin \{counter_1, \ldots, counter_c\}$ and wins of type 2. Consider two sets:

$$K = \{(h, r_1, \ldots, r_n) \in H \times B^n : \textsc{Tag}_{(h,r_1,\ldots,r_n)}(m_i, counter_i) = t_i, \forall i = 1, \ldots, c\}$$

and

$$K' = \{(h, r_1, \ldots, r_n) \in H \times B^n \quad : \quad \textsc{Tag}_{(h,r_1,\ldots,r_n)}(m_i, counter_i) = t_i, \forall i = 1, \ldots, c, \text{ and } \textsc{Verify}_{(h,r_1,\ldots,r_n)}(m, counter, t) = 1 \quad \}.$$

Then $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type1) = \max \frac{|K'|}{|K|}$, where the maximum is taken over all choices of $m_1, \ldots, m_c \in A$, distinct $counter_1, \ldots, counter_c, counter \in \{1, \ldots, n\}$, $t_1, \ldots, t_c \in B$, $m \in A$ and $t \in B$.

We have $|K| = |H||B|^{n-c}$. Now we count the number of tuples $(h, r_1, \ldots, r_n)$ that belong to the set K'. The condition $\textsc{Verify}_{(h,r_1,\ldots,r_n)}(m, counter, t) = 1$ requires $t - r_{counter} \in h(\chi_m)$. So for each choice of h, there are $|h(\chi_m)|$ choices for $r_{counter}$. In total, we have $\sum_{h \in H} |h(\chi_m)|$ choices for the pair $(h, r_{counter})$. Once h is chosen, $r_{counter_1}, \ldots, r_{counter_c}$ are specified as $r_{counter_i} = t_i - h(m_i)$, and there are $|B|^{n-c-1}$ choices for $(r_1, \ldots, \hat{r}_{counter_1}, \ldots, \hat{r}_{counter}, \ldots, \hat{r}_{counter_c}, \ldots, r_n)$. So, $|K'| = |B|^{n-c-1} \sum_{h \in H} |h(\chi_m)|$ and $\frac{|K'|}{|K|} = \frac{1}{|H||B|} \sum_{h \in H} |h(\chi_m)|$. Therefore, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,n]}(c, Type1) \leq \frac{1}{|H||B|} \max_{m \in A} \sum_{h \in H} |h(\chi_m)|$.

Proof of Lemma 5. Suppose in the experiment $\mathbf{Exp}_{\mathcal{A}}^{\mathrm{FMAC}[\mathrm{H},\mathsf{Rand}(\tau)]}(c)$, a random $f \in \mathsf{Rand}(\tau)$ and a random $h \in H$ are chosen. Assume that $\mathcal{A}$ queries the tagging algorithm with c pairs $(m_1, counter_1), \ldots, (m_c, counter_c)$, such that c message numbers $counter_1, \ldots, counter_c$ are all distinct. The tagging algorithm answers $\mathcal{A}$ with $t_1, \ldots, t_c$. Finally, $\mathcal{A}$ presents a forgery $(m, counter_j, t)$ and wins of type 1. In this case, let $\mathsf{Collision}$ be the event that $h(m_j) \in h(\chi_m)$. Since $\chi(m, m_j) = 0$ and H is ϵ-FU_2, $Pr[\mathsf{Collision}] \leq \epsilon$. On the other hand, for any f and h, we have $Pr[\overline{\mathsf{Collision}}] \leq 2^{-\tau}$ because $\mathcal{A}$ must predict $f(\langle h(m'), counter_j \rangle)$ for some $m' \in \chi_m$ having seen only $f(\langle h(m_1), counter_1 \rangle)$, $\ldots$, $f(\langle h(m_c), counter_c \rangle)$ where $\langle h(m'), counter_j \rangle \notin \{\langle h(m_1), counter_1 \rangle, \ldots, \langle h(m_c), counter_c \rangle\}$.

Therefore, $\mathbf{Adv}_{\mathcal{A}}^{\mathrm{FMAC}[H,\mathsf{Rand}(\tau)]}(c, Type1) \leq \max(\epsilon, 2^{-\tau})$.