# Variationally Universal Hashing

Ted Krovetz [a] and Phillip Rogaway [b,c]

[a] *Department of Computer Science, California State University Sacramento CA 95819 USA*

[b] *Department of Computer Science, University of California Davis CA 95616 USA*

[c] *Department of Computer Science, Faculty of Science, Chiang Mai University Chiang Mai 50200 Thailand*

**Abstract**

The strongest well-known measure for the quality of a universal hash-function family $H$ is its being $\varepsilon$-strongly universal, which measures, for randomly chosen $h \in H$, one's inability to guess $h(m')$ even if $h(m)$ is known for some $m \neq m'$. We give example applications in which this measure is too weak, and we introduce a stronger measure for the quality of a hash-function family, $\varepsilon$-*variationally universal*, which measures one's inability to distinguish $h(m')$ from a random value even if $h(m)$ is known for some $m \neq m'$. We explain the utility of this notion and provide an approach for constructing efficiently computable $\varepsilon$-VU hash-function families.

*Key words:* Randomized algorithms, cryptography, hashing, universal hashing.

## 1 Background

A hash-function family $H = \{h: A \to B\}$ is a collection of hash functions, each $h \in H$ having the same domain $A$ and codomain $B$, with $B$ finite. One assumes a hash-function family to be samplable: one can choose a random $h$ from $H$. Carter and Wegman introduced hash-function families, and they and Stinson give various measures of their quality [3,6,7], as we now describe.

Hash-function family $H$ is *strongly universal* (SU) if for all distinct values $m, m'$ from the domain, the pair $(h(m), h(m'))$ is uniformly distributed when $h$ is randomly sampled from $H$. Two relaxations of SU are $\varepsilon$-*almost universal* ($\varepsilon$-AU) and $\varepsilon$-*strongly universal* ($\varepsilon$-SU), where $0 \le \varepsilon \le 1$ is a real number. Hash-function family $H = \{h: A \to B\}$ is $\varepsilon$-AU if the probability that any two distinct values $m, m'$ collide (ie, hash to the same output) when hashed by a randomly selected member of $H$ is at most $\varepsilon$. When $\varepsilon$ is deemed small we say, informally, that $H$ is almost-AU. A hash-function family $H$ is $\varepsilon$-SU if for all distinct $m, m'$ from domain $A$ and all $c, c'$ from codomain $B$,

(1) $\Pr_{h \in H}[h(m) = c] \; = \; \frac{1}{|B|}$, and

(2) $\Pr_{h \in H}[h(m') = c' \mid h(m) = c] \; \leq \; \varepsilon$ .

The first condition says that $h(m)$ is uniformly distributed over $B$ and the second condition says that you cannot guess $h(m')$ with probability better than $\varepsilon$ even if you know $h(m)$. When $\varepsilon$ is deemed small we say, informally, that $H$ is almost-SU.

Almost-AU and almost-SU hash functions have proven to be useful tools. Constructions and software implementations have been given for $\varepsilon$-AU and $\varepsilon$-SU hash-function families with small $\varepsilon$, say $\varepsilon \leq 2^{-30}$, and peak processing rates of less than one CPU cycle per byte of data being hashed [1,2,4]. Known constructions for SU families are much slower to compute.

While the notion of an almost-SU hash-function family might seem strong, we suggest that it is weaker than one may need to be generally useful. As an example, fix a nonempty set $A$ and consider the hash-function family $H^* = \{h_{f,c}\colon A \to \{0,1\}^{128}\}$ where $f\colon A \to \{0,1\}^{64}$ is a function and $c$ is a 64-bit string. Let $h_{f,c}(x) = f(x) \,||\, c$. Choosing a random member of $H^*$ is achieved by uniformly selecting a value $c$ from $\{0,1\}^{64}$ and, for each $x \in A$, assigning to $f(x)$ a uniformly selected value from $\{0,1\}^{64}$. Then $H^*$ is $2^{-64}$-SU, which sounds good, but there are natural applications where it is less appropriate than one might expect. For example, consider a hash table of $2^{10}$ entries where each element $x \in A$ is mapped to a position in the table by using as an index the last 10 bits of $h(x)$. Here we randomly choose $h \in H^*$ before we begin to hash. Since $H^*$ is $2^{-64}$-SU and our table has only $2^{10}$ entries, one might think that $H^*$ should work fine for this application. But clearly it will not, hashing all values to the same table entry. Truncated Wegman-Carter message authentication makes for another natural example. If one uses $H^*$ to make a Wegman-Carter message authentication code [7], xoring hash outputs with a random string, and then, for concision, truncating the result to the final 64-bits, then all security is lost, since all messages produce the same result.

Although $H^*$ is contrived, the examples are not, and they suggest that the definition of $\varepsilon$-SU, which focuses on one's inability to know the *entire* value of $h(m')$ once $h(m)$ is known, may not be a technically desirable way to relax the definition of SU when hash outputs undergo further processing. One should instead capture the idea that *everything* about $h(m')$ looks random, even if one knows $h(m)$. (In particular, the last 10 or 64 bits will look random.) This paper formalizes this notion and gives an efficient construction meeting it, thus creating a more generally useful class of hash functions for applications.

## 2 Almost-VU Hash Functions

First we recall a standard notion for the distance between two probability distributions. If $X$ is a random variable over set $S$ with distribution $D$ and probability mass function $p(x) = \Pr[X = x]$, and $X'$, also over $S$, has distribution $D'$ and mass function $p'(x) = \Pr[X' = x]$, then the *variational distance* between $D$ and $D'$ is

$$dist(D, D') \;=\; \sum_{\substack{y \in S \\ p(y) > p'(y)}} (p(y) - p'(y)) \;=\; \frac{1}{2} \sum_{y \in S} \Big| p(y) - p'(y) \Big| \,.$$

For finite $S$, let Uniform($S$) be the uniform distribution over $S$. When $D'$ is Uniform($S$) then $p(y) - p'(y)$ in this distance measure is $p(y) - 1/|S|$.

Definition of $\varepsilon$-VU. We suggest strengthening the definition of $\varepsilon$-SU by measuring the variational distance, given knowledge of $h(m)$, between the distribution of $h(m')$ and the uniform distribution. We say that hash-function family $H = \{h\colon A \to B\}$ is *$\varepsilon$-variationally universal* ($\varepsilon$-VU) if for all distinct $m, m' \in A$, and all $c \in B$,

(1) $\Pr_{h \in H}[h(m) = c] \;=\; \frac{1}{|B|}$, and

(2) $\frac{1}{2} \sum_{y \in B} \Big| \Pr_{h \in H}[h(m') = y \,|\, h(m) = c] - \frac{1}{|B|} \Big| \;\leq\; \varepsilon$ .

The first condition again says that $h(m)$ is uniformly distributed over $B$ while the second condition says the variational distance between Uniform($B$) and the distribution induced on $h(m')$ when $h(m) = c$ is no more than $\varepsilon$. In other words, we demand that for any distinct $m$ and $m'$, the value $h(m')$ should look uniform even if we know $h(m)$. The quantity $\varepsilon$ measures how far from uniform $h(m')$ might be. If $\varepsilon$ is deemed small we may say, informally, that $H$ is almost-VU.

With regard to the motivating examples, the $\varepsilon$-VU definition ensures good properties when outputs are truncated. It is not hard to show that if $H$ is an $\varepsilon$-VU hash-function family with each function returning $n$-bit strings, then returning the strings truncated to $m$ bits (any $1 \leq m \leq n$ bits may be selected) yields a hash-function family that is still $\varepsilon$-VU. This means it is always safe to truncate bits produced by an almost-VU hash-function family whereas this is not always the case with an almost-SU one.

An equivalent formulation. Another natural way to claim that a hash function appears random over two points is to say that no algorithm can do well at distinguishing between the hash-values of two distinct inputs and a random pair of codomain points. A hash-function family $H = \{h\colon A \to B\}$

would be deemed $\varepsilon$-good under this notion if $h(m)$ is uniform for any $m \in A$, as before, and for all functions $f\colon B^2 \to \{0,1\}$, for all distinct inputs $m, m' \in A$, we have $\Pr_{h\in H}[f(h(m), h(m')) = 1] - \Pr_{x,y\in B}[f(x,y) = 1] \leq \varepsilon$. This notion is *weaker* than our $\varepsilon$-VU definition because the function $f$ has no control of the value $h(m)$ when analyzing the output of $h(m')$. In contrast, the following definition allows the value for $h(m)$ to be arbitrarily chosen and is equivalent to our definition of $\varepsilon$-VU. Hash-function family $H = \{h\colon A \to B\}$ is $\varepsilon$-VU if for all functions $f\colon B \to \{0,1\}$, for all distinct $m, m' \in A$, for all $c \in B$,

(1) $\Pr_{h\in H}[h(m) = c] \;=\; \frac{1}{|B|}$, and

(2) $\Pr_{h\in H}[f(h(m')) = 1 \mid h(m) = c] - \Pr_{b\in B}[f(b) = 1 \mid h(m) = c] \;\leq\; \varepsilon$.

The difference of inequality (2) is maximized when $f$ is the function that returns 1 only on values $y \in B$ for which $Pr_{h\in H}[h(m') = y \mid h(m) = c] > 1/|B|$. When this is the case, computing the difference is identical to computing the variational distance between Uniform($B$) and the distribution induced on $h(m')$ when $h(m) = c$. This indicates that this definition is equivalent to our original formulation of $\varepsilon$-VU.

Almost-SU is weaker than almost-VU. Any almost-VU family of hash functions is almost-SU as well; specifically, if $H$ is an $\varepsilon$-VU hash-function family with codomain $B$ then it is also $(\varepsilon{+}1/|B|)$-SU. The converse is not true. Think back to hash-function family $H^*$ described in Section 1. It is almost-SU, but it is not almost-VU. This hash-function family satisfies part (1) of the $\varepsilon$-VU definition but it only satisfies part (2) for high $\varepsilon$. For each randomly chosen $h \in H^*$ there are only $2^{64}$ strings that can be produced because $h$ always produces the same trailing 64 bits, and for each input all $2^{64}$ possible outputs are equiprobable. So the distance between the distribution for $h(m)$ and Uniform($\{0,1\}^{128}$) is $2^{64}(2^{-64} - 2^{-128}) = 1 - 2^{-64}$.

Are typical almost-SU constructions almost-VU? The degenerate example above notwithstanding, one might wonder if well-known constructions for almost-SU hash functions are already almost-VU. Certainly SU hash-function families are 0-VU, but typical constructions for almost-SU hash functions will not be almost-VU. As an example, consider hashing using polynomial evaluation [1,3,5]. In one form of this paradigm, inputs are broken up into words and the words are interpreted as coefficients in a polynomial over some finite field, say the field with $p$ points. Given a prime $p$, the following hash-function family $H = \{h_{a,b}\colon \mathbb{Z}_p^n \to \mathbb{Z}_p\}$ hashes $n$-vectors and is $(n/p)$-SU. Given $\mathbf{m} = (m_n, m_{n-1}, \ldots, m_2, m_1)$ with all $m_i \in \mathbb{Z}_p$ and keys $a, b \in \mathbb{Z}_p$, the hash of $\mathbf{m}$ is

$$h_{a,b}(\mathbf{m}) = \left(b + \sum_{i=1}^{n} m_i a^i\right) \bmod p\ .$$

Choosing a random element of $H$ is done by choosing a random $a, b \in \mathbb{Z}_p$.

Although this family is $(n/p)$-SU, which is good when $p$ is large and $n$ is not, the hash-function family is not even $(1/3)$-VU. Let $n = 2$ and $p > 3$ be a prime. Let $\mathbf{m} = (0, 0)$, $\mathbf{m}' = (1, 0)$ and $c = 0$. Because $n = 2$ the hash function is evaluated $h_{a,b}(\mathbf{m}) = (m_2 a^2 + m_1 a + b) \bmod p$. Condition (1) of the $\varepsilon$-VU definition requires $h_{a,b}(\mathbf{m})$ be uniformly distributed over $\mathbb{Z}_p$ when $a$ and $b$ are randomly chosen from $\mathbb{Z}_p$, which is satisfied because of the random translation $b$. Condition (2) requires computation of the variational distance between the distribution of $h_{a,b}(\mathbf{m}')$ and Uniform($\mathbb{Z}_p$) when $h_{a,b}(\mathbf{m}) = c$. However, because of the values we have selected for $\mathbf{m}$, $\mathbf{m}'$ and $c$, this computation simplifies to $\frac{1}{2}\sum_{y\in\mathbb{Z}_p}\left|\Pr_{a\in\mathbb{Z}_p}[a^2 = y \bmod p] - \frac{1}{p}\right|$, which is exactly $(p-1)/2p$, a number greater than $1/3$ for any $p > 3$.

## 3 An Almost-VU Construction

While SU hash-families are 0-VU, we have already remarked that no SU constructions are known with efficiency comparable to that of best almost-AU constructions. Composing a high-speed almost-AU hash-function family with an SU hash-function family, however, is a good alternative to using an SU family directly. When hashing large inputs, the composite hash-function family will do the bulk of the work in the fast almost-AU part but will be almost-VU because of the subsequent SU component. We now show that this construction works.

Let $A$, $B$ and $C$ be sets with $B$ and $C$ finite. Let $H = \{h\colon A \to B\}$ and $G = \{g\colon B \to C\}$ be hash-function families. We define the composed family of functions $G \circ H = \{f\colon A \to C\}$ as $\{g \circ h \mid h \in H,\ g \in G\}$. To choose a random element from $G \circ H$ we choose random elements $h \in H$ and $g \in G$ and consider $f = g \circ h$ to be the random element.

**Theorem 1** *Let $A$, $B$ and $C$ be sets with $B$ and $C$ finite. If $H^{\mathrm{au}} = \{h\colon A \to B\}$ is $\varepsilon$-AU and $H^{\mathrm{su}} = \{g\colon B \to C\}$ is SU then $H^{\mathrm{su}} \circ H^{\mathrm{au}}$ is $\varepsilon(1 - 1/|C|)$-VU.*

We note that the theorem's claim is stronger than saying $H^{\mathrm{su}} \circ H^{\mathrm{au}}$ is $\varepsilon$-VU. This is because $\varepsilon = 0$ is perfect for $\varepsilon$-VU whereas $\varepsilon = 1/|C|$ is perfect for $\varepsilon$-SU (and $\varepsilon$-AU), and so scaling between the two is inevitable.

**PROOF.** Let $c \in C$ and let $m, m' \in A$ be distinct. For convenience, let $f$ be shorthand for $g \circ h$ and let all probability measures be over the choice of $h \in H^{\mathrm{au}}$ and $g \in H^{\mathrm{su}}$. Because $H^{\mathrm{su}}$ is strongly universal, $g(h(m))$ is uniformly distributed over $C$ for randomly chosen $g \in H^{\mathrm{su}}$ and any value of $h(m)$.

Let $D$ be the distribution induced on $f(m')$ when $f$ is chosen randomly and $f(m) = c$. Then, we must show $dist(D, \mathrm{Uniform}(C)) \le (\varepsilon - \frac{\varepsilon}{|C|})$. We begin by using the definition of *dist* to rewrite the left-hand side of the desired inequality as

$$\frac{1}{2} \sum_{y \in C} \left| \Pr[f(m') = y \mid f(m) = c] - \frac{1}{|C|} \right| .$$

Rewrite this expression with the $y = c$ term extracted from the summation,

$$\frac{1}{2} \left( \left| \Pr[f(m') = c \mid f(m) = c] - \frac{1}{|C|} \right| + \sum_{\substack{y \in C \\ y \ne c}} \left| \Pr[f(m') = y \mid f(m) = c] - \frac{1}{|C|} \right| \right), \quad (1)$$

and simplify each of the two halves. First, $\Pr[f(m') = c \mid f(m) = c] - 1/|C|$ can be rewritten as

$$\begin{aligned} &\Pr[f(m') = c \mid f(m) = c \wedge h(m') = h(m)] \cdot \Pr[h(m') = h(m)] \quad (2) \\ &+ \Pr[f(m') = c \mid f(m) = c \wedge h(m') \ne h(m)] \cdot \Pr[h(m') \ne h(m)] - \frac{1}{|C|} . \end{aligned}$$

Notice that if $h(m') = h(m)$, then $f(m')$ must be equal to $f(m)$, and that if $h(m')$ and $h(m)$ are distinct, then $\Pr[f(m') = f(m)] = 1/|C|$ because $H^{\mathrm{su}}$ is strongly universal. Letting $p = \Pr[h(m') = h(m)]$, we can simplify Equation 2 to

$$p + \frac{1}{|C|}(1 - p) - \frac{1}{|C|} = p - \frac{p}{|C|} . \quad (3)$$

Next, we look at the term within the summation in Equation 1, $\Pr[f(m') = y \mid f(m) = c] - \frac{1}{|C|}$, with $y \ne c$, which can be rewritten as

$$\begin{aligned} &\Pr[f(m') = y \mid f(m) = c \wedge h(m') = h(m)] \cdot \Pr[h(m') = h(m)] \quad (4) \\ &+ \Pr[f(m') = y \mid f(m) = c \wedge h(m') \ne h(m)] \cdot \Pr[h(m') \ne h(m)] - \frac{1}{|C|} . \end{aligned}$$

This time, because $y \ne c$ and $H^{\mathrm{su}}$ is SU, $\Pr[f(m') = y \mid f(m) = c \wedge h(m') = h(m)] = 0$ and $\Pr[f(m') = y \mid f(m) = c \wedge h(m') \ne h(m)] = 1/|C|$. Again letting $p = \Pr[h(m') = h(m)]$, we can simplify Equation 4 to

$$0 + \frac{1}{|C|}(1 - p) - \frac{1}{|C|} = -\frac{p}{|C|} . \quad (5)$$

Substituting Equations 3 and 5 into Equation 1 results in

$$\frac{1}{2}\left(\left|p-\frac{p}{|C|}\right|+\sum_{\substack{y\in C\\ y\neq c}}\left|-\frac{p}{|C|}\right|\right)=\frac{1}{2}\left(p-\frac{p}{|C|}+(|C|-1)\frac{p}{|C|}\right)=p\left(1-\frac{1}{|C|}\right).$$

Finally, $p \leq \varepsilon$ because $p = \Pr[h(m) = h(m')]$ and $H^{\mathrm{au}}$ is assumed $\varepsilon$-AU. Thus $\mathit{dist}(D, \mathrm{Uniform}(C)) \leq \varepsilon(1 - 1/|C|)$, as desired. □

This construction is a simple way to build a hash-function family that harnesses the speed of fast almost-AU hash families while at the same time providing a stronger guarantee. This same strategy can be applied to accelerate almost-SU hash families too: compose a fast $\varepsilon$-AU hash-function family with an almost-SU hash-function family and you will get a faster almost-SU hash-function family in return. But for the small price of using an SU family rather than an almost-SU family, the guarantee is more generally useful.

## References


[1] D. Bernstein, The Poly1305-AES message-authentication code, in: Proceedings of Fast Software Encryption, FSE 2005, LNCS vol. 3557, Springer-Verlag, 2005, pp. 32–49.

[2] J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, UMAC: Fast and secure message authentication, in: Advances in Cryptology – CRYPTO '99, LNCS vol. 1666, Springer-Verlag, 1999, pp. 216–233.

[3] L. Carter, M. Wegman, Universal classes of hash functions, J. of Computer and System Sciences 18 (1979), pp. 143–154.

[4] S. Halevi, H. Krawczyk, MMH: Software message authentication in the Gbit/second rates, in: Proceedings of Fast Software Encryption, FSE 1997, LNCS vol. 1267, Springer-Verlag, 1997, pp. 172–189

[5] V. Shoup, On fast and provably secure message authentication based on universal hashing, in: Advances in Cryptology – CRYPTO '96, LNCS vol. 1109, Springer-Verlag, 1996, pp. 313–328.

[6] D. Stinson, Universal hashing and authentication codes, Designs, Codes and Cryptography 4 (1994), pp. 369–380.

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