Short-output universal hash functions & their use in fast and secure data authentication

Long Nguyen and Bill Roscoe

Oxford University
Department of Computer Science

ε-almost universal hash functions (UHF)

Definition: given R is the set of all different keys.

For any pair of different messages $m_1 \neq m_2$, we have

$$\mathrm{Prob}_{\{k \in R\}}[h(k, m_1) = h(k, m_2)] \leq \varepsilon$$

We denote b the bit length of the UHF then $\varepsilon \geq 2^{-b}$

Why short-output UHF?

Operation on word-size values (b = 16-32 bits) is very fast in any computer

Cryptographic applications:

– Message authentication codes: long-output UHF can be securely constructed by concatenating several instances of short-output UHF.

– Manual authentication protocols: humans manually compare a short string (i.e. a short universal hash value) to agree on the same data.

Multiplicative universal hash function

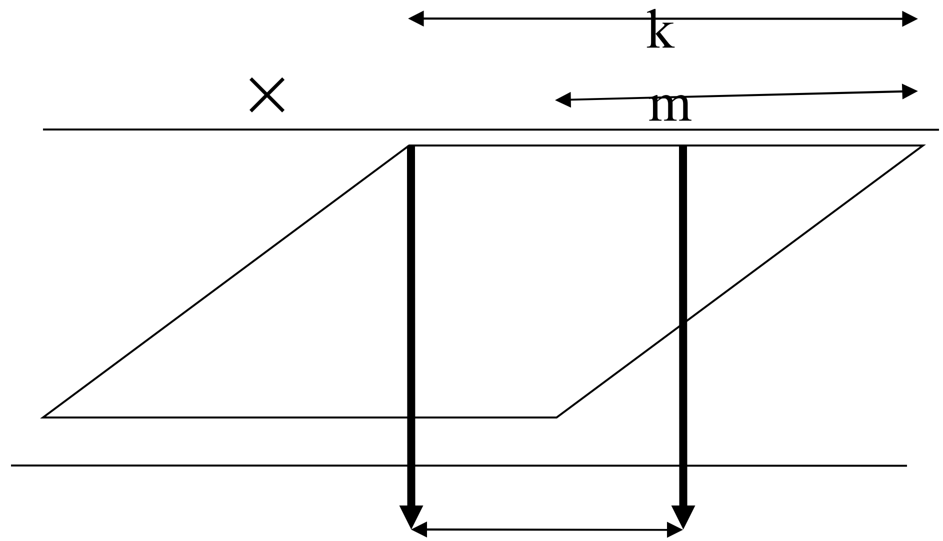
(M. Dietzfelbinger, T. Hagerup, J. Katajainen, M. Penttonen, Journal of Algorithms, 1997, 25:19-51)

Key k must be odd.

$\varepsilon = 2^{1-b}$

(equal-length messages)

Multiplication of a long message is expensive.



$h(k,m) = (k * m \bmod 2^{K}) \text{ div } 2^{K-b}$

Word-multiplication construction: *digest*(*k*,*m*)

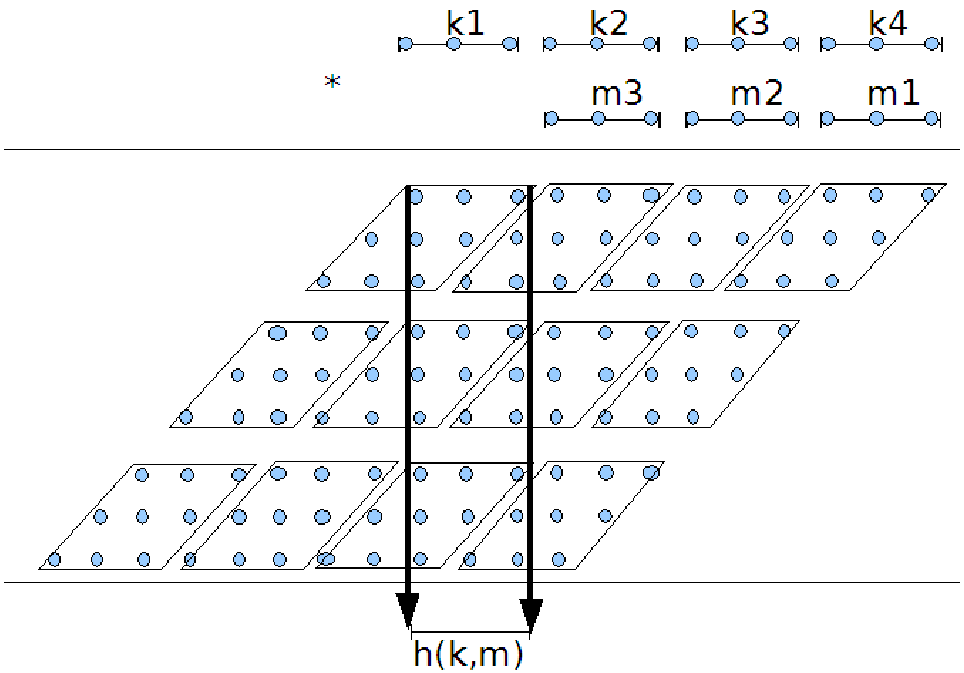
Word-multiplication is fast.

We are interested in the overlap.

$\varepsilon = 2^{1-b}$, where $b \in \{8,16,32\}$
(equal-length messages)

Each message word requires

- $(M+b)/M \approx 1$ key-word
- 2 additions (ADD)
- 2 multiplications (MULT)



k = (k1,k2,k3,k4)
m = (m3,m2,m1)

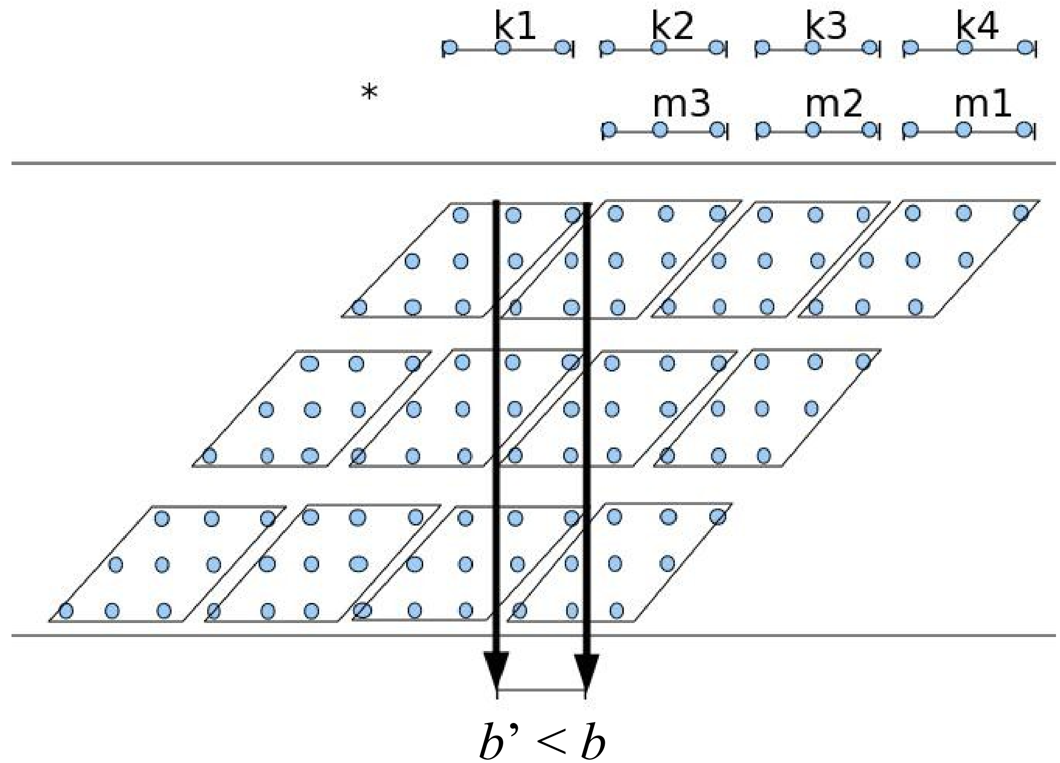
$$\mathrm{digest}(k,m) = \begin{pmatrix} m1 * k1 + (m1*k2 \text{ div } 2^b) + \\ m2 * k2 + (m2*k3 \text{ div } 2^b) + \\ m3 * k3 + (m3*k4 \text{ div } 2^b) \end{pmatrix} \bmod 2^b$$

Shortening digest

Truncation is secure in digest construction:

For any $b' \in \{1,\ldots,b\text{-}1\}$

$\varepsilon = 2 * 2^{-b'}$



k = (k1,k2,k3,k4)
m = (m3,m2,m1)

$$\text{digest}(k,m) = \begin{pmatrix} m1 * k1 + (m1*k2 \text{ div } 2^{b}) + \\ m2 * k2 + (m2*k3 \text{ div } 2^{b}) + \\ m3 * k3 + (m3*k4 \text{ div } 2^{b}) \end{pmatrix} \bmod 2^{b'}$$

MAC: Lengthening digest?

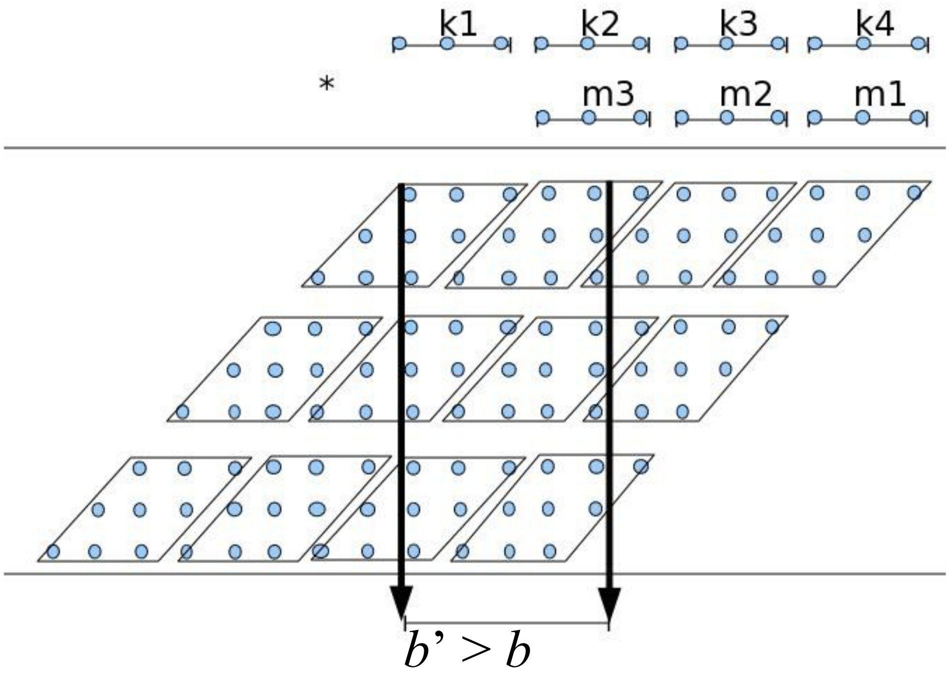
For MAC: we need to increase the output length to $b' > b$.

But the security proof does not work for the following case:

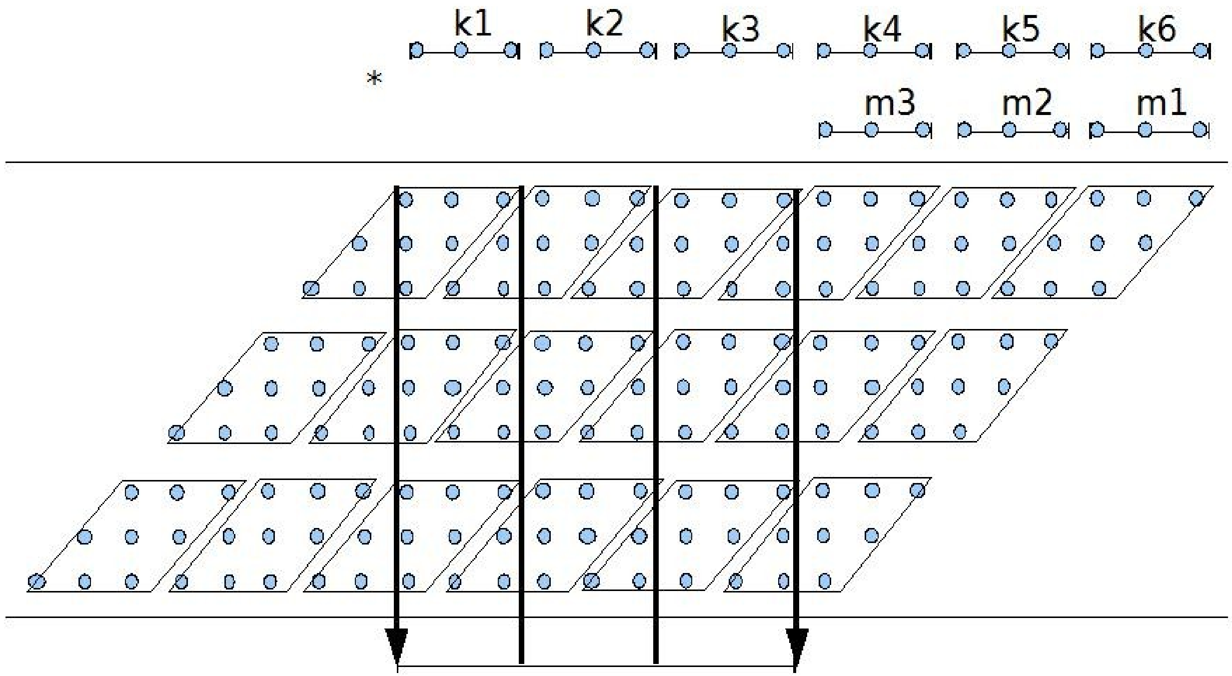
$m_1 = m'_1$

$m_2 = m'_2$

$m_3 \neq m'_3$



Multiple-word digest function



Output bit length is $n * b$ where $b \in \{8,16,32\}$ and $n \in \{1,2,\ldots.\}$

$\varepsilon = (2^{1-b})^n = 2^{n-nb}$

Each message word requires: $(M+nb)/M \approx 1$ key word, $2n$ ADDs & $n+1$ MULTs

Two main competitors: MMH and NH

Our digest function (2010-2011): b-bit output and $\varepsilon = 2 * 2^{-b}$

MMH of Halevi and Krawczyk (1997): b-bit output and $\varepsilon = 6 * 2^{-b}$

NH (within UMAC) of Black et al. (1999): $2b$-bit output and $\varepsilon = 2^{-b}$

- MMH and NH are slightly faster than ours.
- The above security bounds are independent of message length.
- The opposite of *polynomial* based UHF, where collision probability degrades linearly along the length of message being hashed.

MMH

(S. Halevi and H. Krawczyk, FSE 1997)

Fix a prime number $p \in [2^b, 2^b + 2^{b/2}]$:

$$\mathbf{MMH(k,m)} = [(\sum m_i * k_i \bmod 2^{2b}) \bmod p\,] \bmod 2^b$$

For single-word or *b*-bit output: $\varepsilon = 6 * 2^{-b}$

Each message word requires: 1 key-word, 1 ADD, and 1 MULT

For multiple-word or (*n*b*)-bit output: $\varepsilon = 6^n * 2^{-nb}$

Each message word requires: ≈ 1 key-word, *n* ADDs, and *n* MULTs

NH

(J. Black, S. Halevi, H. Krawczyk, T. Krovetz, P. Rogaway, Crypto 1999)

$$\mathbf{NH(k,m)} = \sum (m_{2i-1} + k_{2i-1})\,(m_{2i} + k_{2i}) \bmod 2^{2b}$$

For $2b$-bit output: $\varepsilon = 2^{-b}$

Each message word requires: 1 key-word, 3/2 ADDs, and 1/2 MULT

For multiple-word or $(2n*b)$-bit output: $\varepsilon = 2^{-nb}$

Each message word requires: ≈ 1 key-word, $3n/2$ ADDs, and $n/2$ MULTs

Summary

Scheme	Data length	Key length	MULT per word	ADD per word	ε	Output length
Short-output schemes						
Digest	*M*	*M+b*	*2*	*2*	$2 * 2^{-b}$	*b*
MMH	*M*	*M*	*1*	*1*	$6 * 2^{-b}$	*b*
NH	*M*	*M*	*1/2*	*3/2*	2^{-b}	*2b*

Summary

Scheme	Data length	Key length	MULT per word	ADD per word	ε	Output length
Short-output schemes						
Digest	M	$M+b$	2	2	$2 * 2^{-b}$	b
MMH	M	M	1	1	$6 * 2^{-b}$	b
NH	M	M	$1/2$	$3/2$	2^{-b}	$2b$

Long-output schemes						
Digest	M	$M + nb$	$n+1$	$2n$	$2^n * 2^{-nb}$	nb
MMH	M	$M + (n-1)b$	n	n	$6^n * 2^{-nb}$	nb
NH	M	$M+2(n-1)b$	$n/2$	$3n/2$	2^{-nb}	$2nb$

Message authentication codes

Digest, MMH and NH require key of similar size as data being hashed.

In MAC: each unviersal hash key is reused for a period of time.

Performance

Digest		
Output (bits)	ε	Speed (cpb)
32	$2 * 2^{-32}$	0.53
96	$2^3 * 2^{-96}$	1.54
256	$2^8 * 2^{-256}$	3.44

MMH		
Output (bits)	ε	Speed (cpb)
32	$6 * 2^{-32}$	0.31
96	$6^3 * 2^{-96}$	0.76
256	$6^8 * 2^{-256}$	2.31

NH		
Output (bits)	ε	Speed (cpb)
64	2^{-32}	0.23
192	2^{-96}	0.62
512	2^{-256}	1.90

Our workstation: 1 GHz AMD Athlon 64 X2

	SHA160	SHA256	SHA512
1 GHz AMD Athlon 64 X2 ECRYPT Benchmarking	5.78 [7,14]	12.35 [16,20]	8.54 [10,14]

Manual authentication protocol

1. $A \longrightarrow B$: m_A, $hash(A \| k_A)$

2. $B \longrightarrow A$: m_B, k_B

3. $A \longrightarrow B$: k_A

4. $A \longleftrightarrow B$: $h(k_A \oplus k_B, m_A \| m_B)$

No need of passwords, private keys or PKIs: only human interactions.

Unlike MAC: $h(k,m)$ must have a short output: $b \in \{8,16,32\}$ bits.

But no key $k = k_A \oplus k_B$ is used to hash more than one message, i.e. a long key generation must be done for each protocol run.

To avoid this, we propose: $h(k,m) = digest(k_1, hash(m \| k_2))$

$\varepsilon = 2^{1-b} + \theta$, where θ is the hash collision probability of $hash()$.

Many thanks for your attention.

Manual authentication protocols

- Seek to authenticate (public) data from human trust and human interactions.

- Remove the needs for shared secrets, passwords and PKIs.

- Use cryptographic or universal hash functions.

A protocol of Bafanz et al.

1. $A \longrightarrow B$: m
2. $A \longrightarrow B$: $hash(m)$

- Node A wants to authenticate public data m to B.
- Node A sends m over the high-bandwidth and insecure channel: $\longrightarrow$
- *hash*() is a cryptographic hash function.
- The hash value is *manually* compared by humans over the phone, text messages, or face-to-face conversations: $\longrightarrow$
- However, it is not easy to compare a 160-bit number.

Pair-wise manual authentication protocol

1. $A \longrightarrow B$: m_A, $hash(A \,||\, k_A)$

2. $B \longrightarrow A$: m_B, $hash(B \,||\, k_B)$

3. $A \longrightarrow B$: k_A

4. $B \longrightarrow A$: k_B

5. $A \longrightarrow B$: $h(k_A \oplus k_B\, , m_A \,||\, m_B)$

- Unlike MAC: $h(k,m)$ must have a short output: $b \in \{8,16,32\}$ bits.

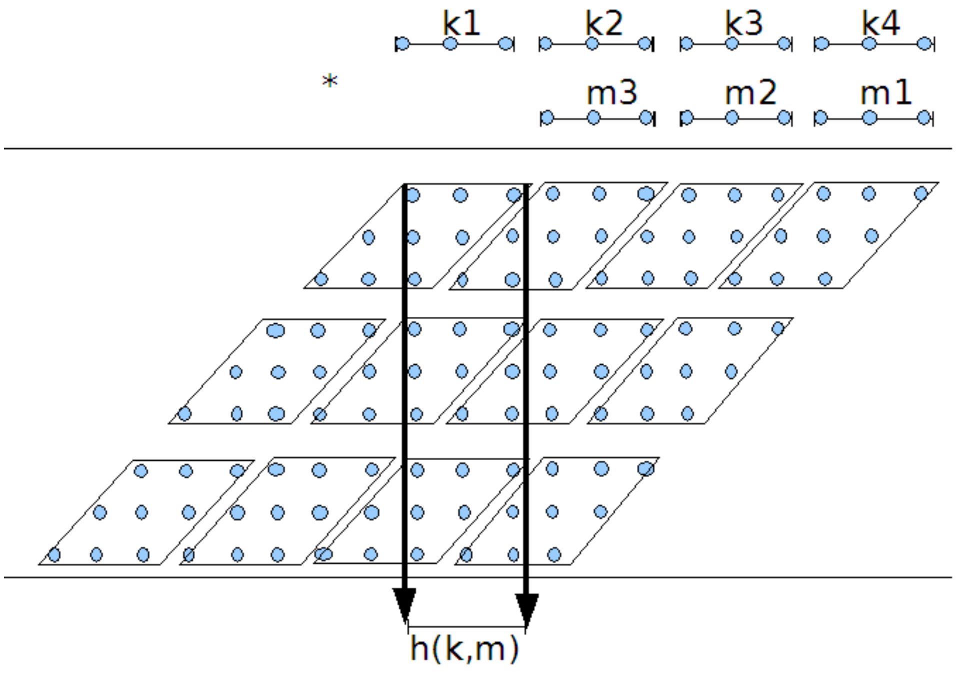
- No key ($k = k_A \oplus k_B$) is used to hash more than one message, and so resistance against substitution attacks is not required.

- What $h(k,m)$ needs to resist is a collision attack.

Tightness of security

Proof says that

If key k is randomly selected from $\{0,1\}^{M+b}$ then $\varepsilon \leq 2^{1-b}$ on equal length messages.



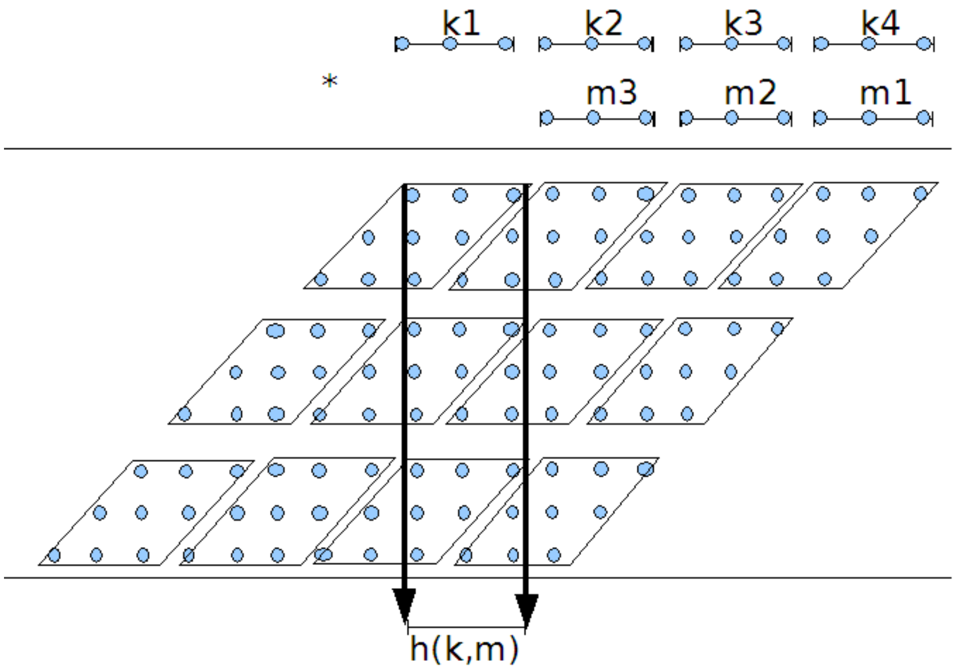
k = (k1,k2,k3,k4)
m = (m3,m2,m1)

$$h(k,m) = \begin{pmatrix} m1 * k1 + (m1*k2 \text{ div } 2^b) + \\ m2 * k2 + (m2*k3 \text{ div } 2^b) + \\ m3 * k3 + (m3*k4 \text{ div } 2^b) \end{pmatrix} \bmod 2^b$$

Tightness of security

Proof says that

If key k is randomly selected from $\{0,1\}^{M+b}$ then $\varepsilon \le 2^{1-b}$ on equal length messages.



Exhaustive tests for small values of $b \in \{6,7,8\}$ shows that:

$$\varepsilon = 1.875 * 2^{-b}$$

k = (k1,k2,k3,k4)

m = (m3,m2,m1)

$$h(k,m) = \begin{pmatrix} m1 * k1 + (m1*k2 \text{ div } 2^b) + \\ m2 * k2 + (m2*k3 \text{ div } 2^b) + \\ m3 * k3 + (m3*k4 \text{ div } 2^b) \end{pmatrix} \bmod 2^b$$