The Hash Function Hamsi

Özgül Küçük
Katholieke Universiteit Leuven, Dept. ESAT/SCD-COSIC, and IBBT
Kasteelpark Arenberg 10, B–3001 Heverlee, Belgium
ozgul.kucuk@esat.kuleuven.be

January 13, 2009

Contents

1 Introduction **4**

2 Specification **5**
2.1 Introduction 5
2.2 The Hash Function Hamsi 6
2.2.1 General Design 6
2.2.2 Initial Values 7
2.2.3 Message Padding 7
2.2.4 Message Expansion 9
2.2.5 Concatenation 10
2.3 The Non-linear Permutation P 11
2.3.1 Addition of Constants and Counter 11
2.3.2 Substitution Layer 12
2.3.3 Diffusion Layer 12
2.3.4 Truncation T 14
2.4 The Non-linear Permutation P_f 15
2.5 Truncations T_{224}, T_{384} 15
2.6 Number of Rounds 16

3 Design Rationale **17**
3.1 Message Expansion 17
3.2 Concatenation 17
3.3 Substitution and Diffusion Layer 18
3.4 Tunable Parameters, Weaker versions 18

4 Implementation **19**
4.0.1 Software 19
4.0.2 Hardware 20

A Appendix **22**
A.1 Tables for Message Expansion of Hamsi-256 and Hamsi-224 . . . 22
A.2 Tables for Message Expansion of Hamsi-512 and Hamsi-384 . . . 24

List of Figures

2.1 General Design of Hamsi 8
2.2 Concatenation in Hamsi-256 and Hamsi-224 11
2.3 Sboxes acting over 4-bits over the columns of an Hamsi state. . . 13
2.4 Application of L in Hamsi-256 and Hamsi-224. 13
2.5 Truncation in Hamsi-256 and Hamsi-224 15

List of Tables

2.1 Hamsi variants and security claims 6
2.2 Notations 6
2.3 Initial Values of Hamsi 9
2.4 Constants used in P 12
2.5 Sbox used in Hamsi 12
2.6 Constants used in P_f 15
2.7 Number of rounds of permutations P and P_f 16

Chapter 1

Introduction

This document contains a proposal for the SHA-3 hash function competition. Specifications of our proposal, Hamsi, is given in Chapter 2. Design rationals are discussed in Chapter 3, followed by hardware and software aspects in Chapter 4. Finally, we include the necessary tables in the Appendix.

Chapter 2

Specification

2.1 Introduction

Hamsi is a family of cryptographic hash functions. There are two instances of Hamsi, Hamsi-256 and Hamsi-512. Table 2.1 summarizes the variants, corresponding parameters and security claims in bits. Hamsi-224 and Hamsi-384 are very similar to Hamsi-256 and Hamsi-512 respectively. They only differ in initial values, and a final truncation. Thus, here we will mainly mention Hamsi-256 and Hamsi-512. Unless explicitly mentioned, operations and data structures for Hamsi-256 and Hamsi-512 apply for their stripped down counterparts, Hamsi-224 and Hamsi-384 respectively.

At the core of Hamsi are the expansion function and round transformations. Round transformation operates on a state matrix of 4 rows. The number of columns is 4 for Hamsi-256, 8 for Hamsi-512. Any entry in the matrix is a word of 32 bits.

Whenever appropriate, these entries are handled big-endian. But, throughout the algorithm, very few places operate on bits, and the only places that require care implementing are the input and output, in which network order among bytes is assumed, and the bits are numbered starting from the MSB of each byte. Other than that, most operations are word operations, and work without the need of endianness conversion within the rounds, even on NUXI machines. One notable exception is the substitution boxes, where the first row of the state matrix is considered the LSB, and similarly the fourth, MSB. This makes a little endian application.

In every round, 4 operations change the matrix. The first is a constant xor into the whole matrix. The second is a simple xor of round number into the LSB bits of s[1]. The third is an Sbox substitution, and the fourth is a diffusion operation on the matrix.

The substitution layer uses a simple Sbox to operate on groups of 4 bits taken from the same bit position in each 4 rows of the state matrix. The result is written back into the same bits.

The diffusion layer operates on 4 words from different positions in the matrix, and the result is written back to those positions.

Table 2.1: Hamsi variants and security claims

Variant	Hash length	Collision resistance	Preimage resistance	2nd-preimage resistance	Message size per iteration
Hamsi-256	256	128	256	256	32
Hamsi-512	512	256	512	512	64
Hamsi-224	224	112	224	224	32
Hamsi-384	384	192	384	384	64

2.2 The Hash Function Hamsi

2.2.1 General Design

In this section we describe the general design, namely the iteration mode of Hamsi. Hamsi is based on the Concatenate-Permute-Truncate design strategy used in several hash functions like Snefru [2] and Grindhal [1]. In addition to this approach, it uses a message expansion and a feedforward of the chaining value in each iteration. The non-linear permutation required for the design uses the linear transformation and one of the Sbox of the block cipher Serpent [3]. General design is shown in Fig. 2.1, but more precisely, Hamsi can be described as the composition of the following mappings:

Message Expansion	$E : \{0,1\}^m \rightarrow \{0,1\}^n$
Concatenation	$C : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^s$
Non-linear Permutations	$P, P_f : \{0,1\}^s \rightarrow \{0,1\}^s$
Truncations	$T : \{0,1\}^s \rightarrow \{0,1\}^n$
	$T_{224} : \{0,1\}^{256} \rightarrow \{0,1\}^{224}$
	$T_{384} : \{0,1\}^{512} \rightarrow \{0,1\}^{384}$

Specifications of the mappings for different variants of Hamsi are given in the following sections. Let $(M_1||M_2||M_3||\ldots||M_l||)$ be properly padded message,

Table 2.2: Notations

F_4	Finite Field with 4 elements
$<<<$	left rotation
$\oplus$	Exclusive or
$<<$	left shift
$[n, m, d]$	Code with length n, dimension m and minimum distance d

then Hamsi variants can be described as follows:

Hamsi-256:

$$h_i = (T \circ P \circ C(E(M_i), h_{i-1})) \oplus h_{i-1}, \quad h_0 = iv_{256}, \quad 0 < i < l \tag{2.1}$$
$$h = (T \circ P_f \circ C(E(M_l), h_{l-1})) \oplus h_{l-1} \tag{2.2}$$

Hamsi-224:

$$h_i = (T \circ P \circ C(E(M_i), h_{i-1})) \oplus h_{i-1}, \quad h_0 = iv_{224}, \quad 0 < i < l \tag{2.3}$$
$$h = (T_{224} \circ P_f \circ C(E(M_l), h_{l-1})) \oplus h_{l-1} \tag{2.4}$$

Hamsi-512:

$$h_i = (T \circ P \circ C(E(M_i), h_{i-1})) \oplus h_{i-1}, \quad h_0 = iv_{512}, \quad 0 < i < l \tag{2.5}$$
$$h = (T \circ P_f \circ C(E(M_l), h_{l-1})) \oplus h_{l-1} \tag{2.6}$$

Hamsi-384:

$$h_i = (T \circ P \circ C(E(M_i), h_{i-1})) \oplus h_{i-1}, \quad h_0 = iv_{384} \quad 0 < i < l \tag{2.7}$$
$$h = (T_{384} \circ P_f \circ C(E(M_l), h_{l-1})) \oplus h_{l-1} \tag{2.8}$$

$m = 32, n = 256, s = 512$ for Hamsi-256 and Hamsi-224
$m = 64, n = 512, s = 1024$ for Hamsi-512 and Hamsi-384.

2.2.2 Initial Values

Inital values are used as the initial chaining value, h_0. Hamsi has 4 initial values; iv_{256}, iv_{224}, iv_{512}, iv_{384} used in Hamsi-256, Hamsi-224, Hamsi-512 and Hamsi-384 respectively. Initial values are obtained from the UTF-8 encoding of the text "Özgül Küçük, Katholieke Universiteit Leuven, Departement Elektrotechniek, Computer Security and Industrial Cryptography, Kasteelpark Arenberg 10, bus 2446, B-3001 Leuven-Heverlee, Belgium."

Initial values are obtained in the following manner. The encoding of the address string is UTF-8. iv_{224} is the first 256 bits, iv_{256} is the second 256 bits, totalling 512. iv_{384} is the second 512 bits and iv_{512} is the third 512 bits. The iv values consist of 32 bit words, each read in a *Big endian* fashion. Thus, the first word of iv_{224} is 0x3c967a67, which is 0x3c96 for "Ö" UTF-8 encoded, 0x7a for "z", and 0x67 for "g", giving us the beginning 4 bytes of the address string "Özg".

2.2.3 Message Padding

Hamsi operates on 32 and 64 bit message blocks in Hamsi-256, Hamsi-224 and Hamsi-512, Hamsi-384, respectively. Message padding is performed as follows; Append '1'-bit to the message and number of '0'-bits filling the last message block. Append the message length as 64-bit unsigned integer as the last message block. Note that Hamsi has maximum message length $2^{64} - 1$.

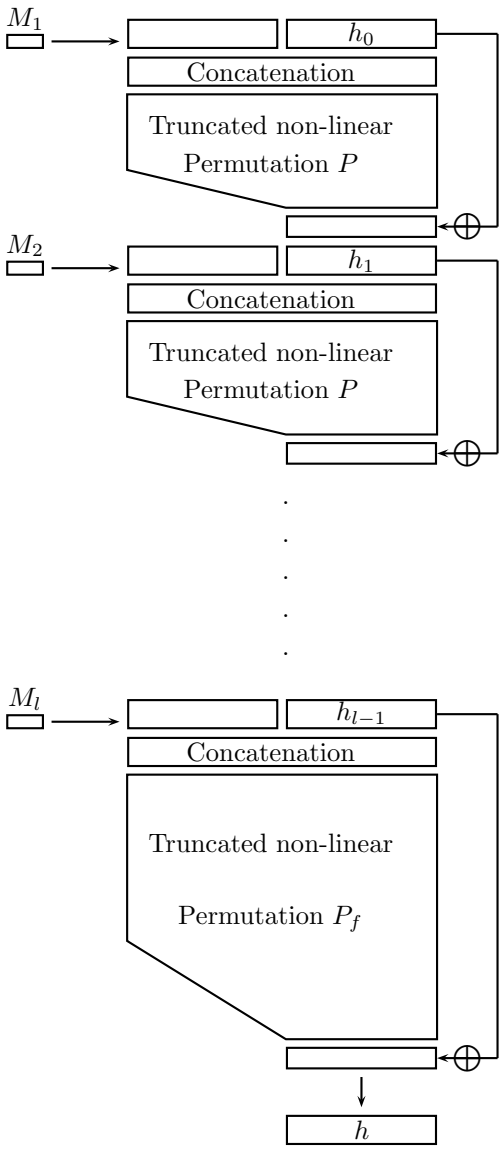

Figure 2.1: General Design of Hamsi

Table 2.3: Initial Values of Hamsi

iv_{224}	0x3c967a67, 0x3cbc6c20, 0xb4c343c3, 0xa73cbc6b 0x2c204b61, 0x74686f6c, 0x69656b65, 0x20556e69
iv_{256}	0x76657273, 0x69746569, 0x74204c65, 0x7576656e 0x2c204b61, 0x74686f6c, 0x69656b65, 0x20556e69
iv_{384}	0x656b7472, 0x6f746563, 0x686e6965, 0x6b2c2043 0x6f6d7075, 0x74657220, 0x53656375, 0x72697479 0x20616e64, 0x20496e64, 0x75737472, 0x69616c20 0x43727970, 0x746f6772, 0x61706879, 0x2c204b61
iv_{512}	0x73746565, 0x6c706172, 0x6b204172, 0x656e6265 0x72672031, 0x302c2062, 0x75732032, 0x3434362c 0x20422d33, 0x30303120, 0x4c657576, 0x656e2d48 0x65766572, 0x6c65652c, 0x2042656c, 0x6769756d

2.2.4 Message Expansion

Hamsi uses linear codes [4] for message expansion. The message expansion of Hamsi-224 and Hamsi-256 expands 32-bit to 256-bit with the code [128,16,70] over F_4. This is defined by $E : \{0,1\}^{32} \rightarrow \{0,1\}^{256}$, as follows, here and below G is the generator matrix of the code:

$$\begin{aligned} E(M_i) &= (M_i \times G),\ M_i \in F_4^{16} \\ &= (m_0, m_1, \ldots, m_7),\ m_i \in F_2^{32} \end{aligned}$$

The linear code [128,16,70] is obtained by truncation of two cordinates from each codeword of the linear code [130,16,72] over F_4; this is done by truncating the last two columns from the generator matrix.

The message expansion of Hamsi-384 and Hamsi-512 expands 64-bit to 512-bit with the code [256,32,131] over F_4. $E : \{0,1\}^{64} \rightarrow \{0,1\}^{512}$, defining the expansion is applied as follows:

$$\begin{aligned} E(M_i) &= (M_i \times G),\ M_i \in F_4^{32} \\ &= (m_0, m_1, \ldots, m_{15}),\ m_i \in F_2^{32} \end{aligned}$$

The generator matrices of the codes used in Hamsi are given in Appendix. The linear code [256,32,131] over $GF(4)$ can be generated by Magma with the

following commands, in the given order:

$$E := ExtendCode(ReedSolomonCode(63, 33)); \tag{2.9}$$
$$D := Dual(E); \tag{2.10}$$
$$S := ShortenCode(D, \{1, 2, \ldots, 21\}); \tag{2.11}$$
$$H := Subcode(S, 11); \tag{2.12}$$
$$M := KMatrixSpace(GF(4), 3, 6); \tag{2.13}$$
$$w := PrimitiveElement(GF(4)); \tag{2.14}$$
$$A := M![1, 0, 0, 1, w, w, 0, 1, 0, w, 1, w, 0, 0, 1, w, w, 1]; \tag{2.15}$$
$$K := Dual(LinearCode(A)); \tag{2.16}$$
$$C := ConcatenatedCode(H, K); \tag{2.17}$$
$$P := PunctureCode(C, 1); \tag{2.18}$$
$$H512 := ShortenCode(P, 1); \tag{2.19}$$

Message expansion in detail. Hamsi expansion can be performed in a method suitable for any arbitrary linear transformation: For every *b*yte of the input, depending on the position of the byte, a table is generated offline, that gives an "output contribution", that corresponds to the output of the input where every other byte position is taken as zeroes. During runtime, for every byte position, the value is looked up from the table corresponding to that position, and all the "output contributions" obtained are xored to get the final output. We perform the expansion in the above mentioned method, but the effect of it is taken to be the same with the method explained below.
We take as input, 16 (or 32 for Hamsi-512) values from F_4. These values, as a vector are multiplied by the generator matrix (which is 16×128 or 32×256). The resulting value is termed M, a vector of 128 (or 256) values from F_4. This is the contribution of the input bytes to the iteration function. These bytes, together with chain values (of same length) are used to initialize the internal state of the iteration function. The M vector is used to obtain the vector m (which is of the same size) by a simple bit permutation. The m vector consists of q words of 32 bit (4 words for Hamsi-256, 8 for Hamsi-512). As can be seen in Fig. 2.2, for each $i < q$, every bit of m_i is teamed with the bit in the same position in m_{q+i} to enter the same Sbox. These couples of bits come from the F_4 values in M. To this effect, M_i and M_{q+i} are used to obtain m_i and m_{q+i}, where all even positioned bits (e.g. for $i < 16$, in bit positions $2 \times i$ of M_i and M_{q+i}) are placed in m_i, and all odd positioned bits are placed in m_{q+i}. If we denote the j^{th} bit of a vector a as $a[j]$, then for $0 \leq j < 16$, $m_i[j] = M_i[2 * j]$, $m_i[16+j] = M_{q+i}[2*j]$, $m_{q+i}[j] = M_i[2*j+1]$, and $m_{q+i}[16+j] = M_{q+i}[2*j+1]$.

2.2.5 Concatenation

The expanded message words $(m_0, m_1, \ldots, m_i)$ are concatenated to the chaining value $(c_0, c_1, \ldots, c_j)$, $(i, j = 7, 15)$ forms an extended state. This is afterwards input to the nonlinear permutation P. Concatenation method determines the

ordering of the bits (and words) input to P.

$$(m_0, m_1, \ldots, m_7, c_0, c_1, \ldots, c_7) \xrightarrow{C}$$

m_0	m_1	c_0	c_1
c_2	c_3	m_2	m_3
m_4	m_5	c_4	c_5
c_6	c_7	m_6	m_7

Figure 2.2: Concatenation in Hamsi-256 and Hamsi-224

In Hamsi-256 and Hamsi-224, $C : \{0,1\}^{256} \times \{0,1\}^{256} \rightarrow \{0,1\}^{512}$ is:

$$C(m_0, m_1, \ldots, m_7, c_0, c_1, \ldots, c_7) =$$
$$(m_0, m_1, c_0, c_1, c_2, c_3, m_2, m_3, m_4, m_5, c_4, c_5, c_6, c_7, m_6, m_7),\ m_i, c_i \in F_2^{32}$$

In Hamsi-512 and Hamsi-384, $C : \{0,1\}^{512} \times \{0,1\}^{512} \rightarrow \{0,1\}^{1024}$ is:

$$C(m_0, m_1, \ldots, m_{14}, m_{15}, c_0, c_1, \ldots, c_{14}, c_{15}) =$$
$$(m_0, m_1, c_0, c_1, m_2, m_3, c_2, c_3, c_4, c_5, m_4, m_5, c_6, c_7, m_6, m_7,$$
$$m_8, m_9, c_8, c_9, m_{10}, m_{11}, c_{10}, c_{11}, c_{12}, c_{13}, m_{12}, m_{13}, c_{14}, c_{15}, m_{14}, m_{15}),\ m_i, c_i \in F_2^{32}$$

2.3 The Non-linear Permutation P

The non-linear permutation consists of 3 layers; input bits are first xored with the constants and a counter, this is followed by the application of 4-bit Sboxes and several applications of the linear transformation L, this is repeated as many as number of rounds. We represent a state of the permutation with $(s_0, s_1, s_2, \ldots, s_j)$, $j = 15, 31$ and $s_i \in F_2^{32}, \quad i = 0, 1, \ldots, j$. This can be visualized with a 4×4 and 4×8 matrix in Hamsi-256 and Hamsi-512, respectively.

2.3.1 Addition of Constants and Counter

The constants $\alpha_i \in F_2^{32}, i = 0, 1, 2, \ldots, 31$ are xored with the input state before the substitution layer together with the counter. We use the round number as the counter c, for the 1st round $c = 0$ and 2nd $c = 1$, etc. We use constants to ensure asymmetry in the same round within the sboxes and the counter in between the rounds. The constans are permutations of the sequence $0, 1, 2, \ldots, 15$ (each 4 bits). In Table 2.4 we give the corresponding representation for the

bitsliced implementation.

In Hamsi-256 and Hamsi-224:

$$(s_0, s_1, \ldots, s_{15}) := (s_0 \oplus \alpha_0, s_1 \oplus \alpha_1 \oplus c, \alpha_2, \ldots, s_{15} \oplus \alpha_{15})$$

In Hamsi-512 and Hamsi-384:

$$(s_0, s_1, \ldots, s_{31}) := (s_0 \oplus \alpha_0, s_1 \oplus \alpha_1 \oplus c, \alpha_2, \ldots, s_{31} \oplus \alpha_{31})$$

Table 2.4: Constants used in P

$\alpha_0 = 0xf0f0ff00$	$\alpha_1 = 0xaaaacccc$	$\alpha_2 = 0xccccf0f0$	$\alpha_3 = 0xaaaaff00$
$\alpha_4 = 0xccccff00$	$\alpha_5 = 0xaaaaf0f0$	$\alpha_6 = 0xaaaaf0f0$	$\alpha_7 = 0xff00cccc$
$\alpha_8 = 0xccccf0f0$	$\alpha_9 = 0xff00aaaa$	$\alpha_{10} = 0xff00cccc$	$\alpha_{11} = 0xf0f0aaaa$
$\alpha_{12} = 0xf0f0cccc$	$\alpha_{13} = 0xff00aaaa$	$\alpha_{14} = 0xccccaaaa$	$\alpha_{15} = 0xf0f0ff00$
$\alpha_{16} = 0xaaaacccc$	$\alpha_{17} = 0xf0f0ff00$	$\alpha_{18} = 0xaaaaff00$	$\alpha_{19} = 0xccccf0f0$
$\alpha_{20} = 0xaaaaf0f0$	$\alpha_{21} = 0xccccff00$	$\alpha_{22} = 0xff00cccc$	$\alpha_{23} = 0xaaaaf0f0$
$\alpha_{24} = 0xff00aaaa$	$\alpha_{25} = 0xccccf0f0$	$\alpha_{26} = 0xf0f0aaaa$	$\alpha_{27} = 0xff00cccc$
$\alpha_{28} = 0xff00aaaa$	$\alpha_{29} = 0xf0f0cccc$	$\alpha_{30} = 0xf0f0ff00$	$\alpha_{31} = 0xccccaaaa$

2.3.2 Substitution Layer

Hamsi uses a 4×4-bit Sbox $S : F_2^4 \rightarrow F_2^4$ [3]. Hamsi is conveniently designed for bitslice implementation. There are 128 (or 256 for Hamsi-512) parallel and identical Sboxes, all can be executed at the same time in computer words of up to 128 bits (or 256 for Hamsi-512). Hence, if you have registers of size 128 bits, you can make use of it to make Hamsi faster. But registers of 32 bits are sufficient for the basic implementation.

2.3.3 Diffusion Layer

The diffusion layer of Hamsi is based on the several applications of the linear transformation $L : \{0,1\}^{128} \rightarrow \{0,1\}^{128}$ [3]. L operates on 32-bit words; inputs and outputs 4, 32-bit words.

Table 2.5: Sbox used in Hamsi

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
s[x]	8	6	7	9	3	C	A	F	D	1	E	4	0	B	5	2

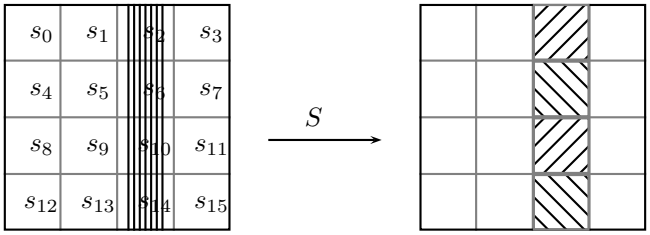

Figure 2.3: Sboxes acting over 4-bits over the columns of an Hamsi state.

Diffusion in Hamsi-256 and Hamsi-224

$$
\begin{aligned}
(s_0, s_5, s_{10}, s_{15}) &:= L(s_0, s_5, s_{10}, s_{15}) \\
(s_1, s_6, s_{11}, s_{12}) &:= L(s_1, s_6, s_{11}, s_{12}) \\
(s_2, s_7, s_8, s_{13}) &:= L(s_2, s_7, s_8, s_{13}) \\
(s_3, s_4, s_9, s_{14}) &:= L(s_3, s_4, s_9, s_{14})
\end{aligned}
$$

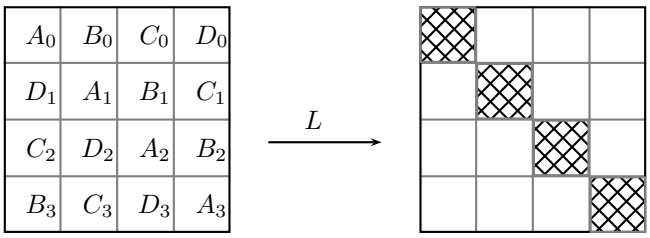

Figure 2.4: Application of L in Hamsi-256 and Hamsi-224.

Diffusion in Hamsi-512 and Hamsi-384

$$
\begin{aligned}
(s_0, s_9, s_{18}, s_{27}) &:= L(s_0, s_9, s_{18}, s_{27}) \\
(s_1, s_{10}, s_{19}, s_{28}) &:= L(s_1, s_{10}, s_{19}, s_{28}) \\
(s_2, s_{11}, s_{20}, s_{29}) &:= L(s_2, s_{11}, s_{20}, s_{29}) \\
(s_3, s_{12}, s_{21}, s_{30}) &:= L(s_3, s_{12}, s_{21}, s_{30}) \\
(s_4, s_{13}, s_{22}, s_{31}) &:= L(s_4, s_{13}, s_{22}, s_{31}) \\
(s_5, s_{14}, s_{23}, s_{24}) &:= L(s_5, s_{14}, s_{23}, s_{24}) \\
(s_6, s_{15}, s_{16}, s_{25}) &:= L(s_6, s_{15}, s_{16}, s_{25}) \\
(s_7, s_8, s_{17}, s_{26}) &:= L(s_7, s_8, s_{17}, s_{26}) \\
(s_0, s_2, s_5, s_{07}) &:= L(s_0, s_2, s_5, s_7) \\
(s_{16}, s_{19}, s_{21}, s_{22}) &:= L(s_{16}, s_{19}, s_{21}, s_{22}) \\
(s_9, s_{11}, s_{12}, s_{14}) &:= L(s_9, s_{11}, s_{12}, s_{14}) \\
(s_{25}, s_{26}, s_{28}, s_{31}) &:= L(s_{25}, s_{26}, s_{28}, s_{31})
\end{aligned}
$$

Note that in Hamsi-512 (and Hamsi-384) L is applied 12 times (3 times more then Hamsi-256). L diffuses over 128-bits and each Hamsi-512 state has 256 bits in each row of the state matrix, L is applied 3 times more in order to achieve diffusion in between the two 128-bits. $L(s_9, s_{11}, s_{12}, s_{14})$ and $L(s_{25}, s_{26}, s_{28}, s_{31})$ need not be applied in the last round because they are already truncated, see 2.3.4.

Description of L

$a, b, c, d \in F_2^{32}$, $L(a, b, c, d)$:

$$
\begin{aligned}
a &:= a <<< 13 \\
c &:= c <<< 3 \\
b &:= b \oplus a \oplus c \\
d &:= d \oplus c \oplus (a << 3) \\
b &:= b <<< 1 \\
d &:= d <<< 7 \\
a &:= a \oplus b \oplus d, \\
c &:= c \oplus d \oplus (b << 7) \\
a &:= a <<< 5 \\
c &:= c <<< 22
\end{aligned}
$$

2.3.4 Truncation T

Truncation $T : \{0,1\}^{512} \rightarrow \{0,1\}^{256}$ in Hamsi-256 and Hamsi-224 is defined as follows:

$$T(s_0, s_1, s_2, \ldots, s_{14}, s_{15}) = (s_0, s_1, s_2, s_3, s_8, s_9, s_{10}, s_{11}) \; each \; s_i \in F_2^{32}$$

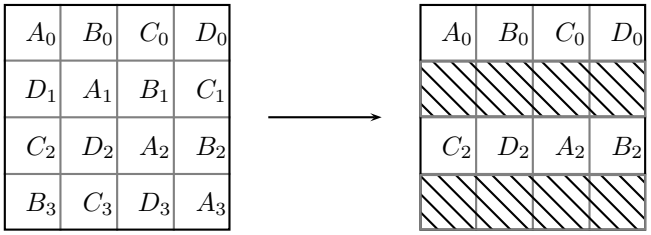

Figure 2.5: Truncation in Hamsi-256 and Hamsi-224

In Hamsi-512 and Hamsi-384 $T : \{0,1\}^{1024} \rightarrow \{0,1\}^{512}$ is as follows:

$$T(s_0, s_1, s_2, \ldots, s_{30}, s_{31}) =$$
$$(s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_{16}, s_{17}, s_{18}, s_{19}, s_{20}, s_{21}, s_{22}, s_{23}) \; each \; s_i \in F_2^{32}$$

Truncation is applied after the last round of the nonlinear permutation. Fig. 2.5 shows the state after the application of linear transformation L and truncation. Similar letters corresponds to the words input to L, like $L(A_0, A_1, A_2, A_3)$,etc.

2.4 The Non-linear Permutation P_f

P and P_f differs only in the number of rounds and constants. P_f is applied to the last message block as final transformation.

Table 2.6: Constants used in P_f

$\alpha_0 = 0xcaf9639c$	$\alpha_1 = 0x0ff0f9c0$	$\alpha_2 = 0x639c0ff0$	$\alpha_3 = 0xcaf9f9c0$
$\alpha_4 = 0x0ff0f9c0$	$\alpha_5 = 0x639ccaf9$	$\alpha_6 = 0xf9c00ff0$	$\alpha_7 = 0x639ccaf9$
$\alpha_8 = 0x639c0ff0$	$\alpha_9 = 0xf9c0caf9$	$\alpha_{10} = 0x0ff0caf9$	$\alpha_{11} = 0xf9c0639c$
$\alpha_{12} = 0xf9c0639c$	$\alpha_{13} = 0xcaf90ff0$	$\alpha_{14} = 0x0ff0639c$	$\alpha_{15} = 0xcaf9f9c0$
$\alpha_{16} = 0x0ff0f9c0$	$\alpha_{17} = 0xcaf9639c$	$\alpha_{18} = 0xcaf9f9c0$	$\alpha_{19} = 0x639c0ff0$
$\alpha_{20} = 0x639ccaf9$	$\alpha_{21} = 0x0ff0f9c0$	$\alpha_{22} = 0x639ccaf9$	$\alpha_{23} = 0xf9c00ff0$
$\alpha_{24} = 0xf9c0caf9$	$\alpha_{25} = 0x639c0ff0$	$\alpha_{26} = 0xf9c0639c$	$\alpha_{27} = 0x0ff0caf9$
$\alpha_{28} = 0xcaf90ff0$	$\alpha_{29} = 0xf9c0639c$	$\alpha_{30} = 0xcaf9f9c0$	$\alpha_{31} = 0x0ff0639c$

2.5 Truncations T_{224}, T_{384}

T_{224} and T_{384} defines the truncation method applied to Hamsi-256 and Hamsi-512 to obtain the digest sizes of 224 and 384 bits.

$T_{224} : \{0,1\}^{256} \rightarrow \{0,1\}^{224}$ is defined as follows:

$$T_{224}(s_0, s_1, \ldots, s_7) = T_{224}(s_0, s_1, s_2, s_3, s_4, s_5, s_6)$$

$T_{384} : \{0,1\}^{512} \rightarrow \{0,1\}^{384}$ is:

$$T_{384}(s_0, s_1, \ldots, s_{15}) = (s_0, s_1, s_3, s_4, s_5, s_6, s_8, s_9, s_{10}, s_{12}, s_{13}, s_{15})$$

2.6 Number of Rounds

Number of rounds recommended for the variants of Hamsi is given below. We would like to stress that number of rounds is the tunable parameter of Hamsi; it can be decreased to obtain weaker versions and increased in order to achieve better security as long as the performance values are not changed drastically.

Table 2.7: Number of rounds of permutations P and P_f

Variant	Hamsi-256	Hamsi-224	Hamsi-512	Hamsi-384
Rounds of P	3	3	6	6
Rounds of P_f	6	6	12	12

Chapter 3

Design Rationale

3.1 Message Expansion

Hamsi operates on relatively small size message blocks, 32 or 64 bits in each iteration. The choice of the small size gives the possibility to use the best known linear codes for the associated parameters, namely [128,16,70] and [256,32,131] over F_4. Each Sbox in the nonlinear permutation inputs 2 bits from the expanded message, hence it makes sense to consider codes over F_4 and F_2 to achieve best possible results in the number of active sboxes.

For Hamsi-224 and Hamsi-256 we considered two codes [256,32,96] over F_2 and [128,16,70] over F_4. To our knowledge we couldn't find a concatenation method to increase the number of active sboxes more or equal then 70, for this reason we believe that the second code gives the best linear choice for the number of active Sboxes. The code [128,16,70] can be constructed in several ways, we choose the one that gives less sparse generator matrix.

Following the same reasoning we choose the best known linear code [256,32,131] over F_4, for message expansion in Hamsi-512 and Hamsi-384.

3.2 Concatenation

Similar to the other components of the design, concatenation method benefits from the bitsliced nature of Hamsi. Concatenation is done in such a way that 2 bits from the expanded message and chaining value are input to the 4×4-bit Sboxes. This can be seen as an initial diffusion.

3.3 Substitution and Diffusion Layer

We use one of Serpent [3] Sbox and the linear transformation for building blocks in the nonlinear permutations, P and P_f. The maximum differential probability of the Sbox is 2^{-2}.

3.4 Tunable Parameters, Weaker versions

Hamsi can be weakened in several ways; the most obvious way is to reduce the number of rounds. We strongly advice the analysis of 1 round for all variants. Lighter and optimal diffusion layers can also be subject of analysis; for example the number of times L applied in Hamsi-512, etc.

Chapter 4

Implementation

4.0.1 Software

Hamsi is designed to be easily iterated through succesive Sbox and diffusion layers. The internal state is placed on a matrix of 4x4 words (or 4x8 for Hamsi-512) of 32 bits. While the diffusion layer works on 4 words, diffusing them, the Sboxes operate on bits that are horizontally aligned, convenient for a bitslice implementation.

Rows of the matrix form the input bits of the Sbox, and logical operations provide a way to execute the Sbox in parallel on all 128 (or 256 for Hamsi-512) Sboxes. This provides a way to improve the speed by placing the rows on larger words if available. If for example the target machine has instructions for 64 bit words, the algorithm can make use of it by executing 64 Sboxes in parallel instead of 32. If furthermore the machine has logical instructions for 128 bit words, Hamsi-256 has 128 parallel Sboxes. For hypothetical machines of 256 bit word sizes, Hamsi-512 still has some more room for speed with its 256 parallel Sboxes. (While Hamsi-256 cannot be further parallelized).

The expansion function is the most complicated part of the whole process, and is implemented via lookup tables. The tables occupy a memory space of 32 KiB, which is suitable even for most embedded devices. For smartcards or similar media, smaller lookup tables may easily be generated simply by taking smaller units of operation. If for example, nibbles are chosen to be the lookup element, the T[4][256][8] table is converted into a T[8][16][8] table, down to a 4 KiB size.

The implementation included, while being the reference implementation, is similar in structure to an optimized implementation (but lacking tweaks for speed). This stems from the very simple design of the algorithm. A further optimization of the software will follow soon.

4.0.2 Hardware

Hamsi is a hardware-friendly algorithm. It's main operations are table lookup, XOR, AND (due to Sboxes), and shifting. Concatenation and Truncation are free in hardware. Hamsi offers implementers a large design space. In order to achieve a high throughput, parallelism inside the Message Expansion, Substitution and Diffusion can be utilized. For example, multiple Lookup Tables can be used to speed up Message Expansion and Substitution, and several L transformations can be performed in parallel. On the other hand, Hamsi can be implemented in a pure sequential manner such that it fits applications where hardware resource is limited.

We are currently working on fast/compact hardware implementation of Hamsi. The implementation results (gate count, throughput, etc) will come soon.

Bibliography

[1] R. Knudsen, L., Rechberger, C., S. Thomsen: Grindahl- a family of hash functions. In Biryukov, A, ed.: Fast Software Encryption, FSE 2007. Volume 4593 of Lecture Notes in Computer Science., Springer-Verlag (2007) 39-57

[2] Merkle R.C: A Fast Software One-Way Hash Function. Journal of Cryptology, 3(1):43-58, 1990.

[3] Serpent: A proposal for the advanced encryption standard. Available from http://www.cl.cam.ac.uk/ rja14/serpent.html.

[4] J.H. van Lint: Introduction to Coding Theory.

[5] Best Known Linear [256,32,d]-Codes in Base 4 Available from http://mint.sbg.ac.at/

Appendix A

Appendix

A.1 Tables for Message Expansion of Hamsi-256 and Hamsi-224

Generator matrix over F_4 for the message expansion is given in the array gen[16][128].

```
gen[16][128] = {
{1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 1, 3, 0, 0, 3, 3, 3, 3, 0, 0, 2, 0, 0,
3, 0, 2, 3, 2, 2, 1, 0, 2, 2, 3, 0, 3, 0, 0, 1, 0, 2, 1, 2, 3, 3, 3, 1, 2, 3, 3, 0, 1, 1, 1,
3, 1, 1, 1, 1, 0, 0, 3, 0, 0, 0, 3, 1, 3, 3, 2, 2, 2, 3, 0, 0, 2, 2, 3, 0, 1, 1, 2, 0, 1, 2,
1, 3, 0, 1, 2, 3, 1, 0, 1, 2, 1, 3, 3, 3, 3, 3, 2, 1, 1, 0, 2, 3, 2, 1, 0, 2, 3, 1, 1, 0, 2,
3, 3, 2, 0, 2, 0},
{0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 3, 3, 0, 2, 1, 1, 1, 3, 0, 1, 2, 0,
2, 3, 1, 0, 2, 3, 1, 1, 1, 3, 0, 3, 2, 3, 0, 3, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, 3, 3, 2, 2,
3, 0, 2, 2, 2, 1, 0, 2, 3, 0, 0, 2, 0, 3, 1, 2, 3, 3, 0, 3, 0, 1, 3, 0, 3, 3, 2, 0, 2, 3, 0,
1, 3, 3, 3, 0, 0, 0, 1, 3, 0, 1, 3, 1, 1, 1, 1, 2, 1, 2, 1, 1, 0, 2, 1, 1, 1, 0, 0, 2, 1, 1,
0, 1, 1, 2, 1, 2},
{0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 0, 3, 3, 2, 0, 3, 3, 1, 3, 1, 1, 2,
2, 2, 2, 3, 1, 3, 0, 1, 0, 0, 1, 0, 1, 2, 3, 3, 3, 0, 2, 0, 2, 2, 3, 2, 1, 2, 2, 1, 0, 0, 1,
0, 2, 3, 1, 1, 2, 1, 2, 2, 3, 0, 2, 1, 2, 1, 0, 3, 2, 1, 0, 3, 1, 0, 1, 0, 0, 0, 3, 0, 1, 2,
3, 3, 3, 0, 2, 2, 3, 0, 2, 2, 3, 3, 1, 3, 3, 3, 0, 1, 2, 2, 0, 3, 1, 1, 1, 0, 3, 3, 3, 2, 0,
3, 2, 0, 1, 2, 3},
{0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 0, 3, 3, 2, 0, 3, 3, 1, 3, 1, 1,
2, 2, 2, 2, 3, 1, 3, 0, 1, 0, 0, 1, 0, 1, 2, 3, 3, 3, 0, 2, 0, 2, 2, 3, 2, 1, 2, 2, 1, 0, 0,
1, 2, 2, 3, 1, 1, 2, 1, 2, 2, 3, 0, 2, 1, 2, 1, 0, 3, 2, 1, 0, 3, 1, 0, 1, 0, 0, 0, 3, 0, 1,
2, 3, 3, 3, 0, 2, 2, 3, 0, 2, 2, 3, 3, 1, 3, 3, 3, 0, 1, 2, 2, 0, 3, 1, 1, 1, 0, 3, 3, 3, 2,
0, 3, 1, 0, 0, 1},
{0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 2, 1, 0, 1, 2, 0, 0, 0, 1, 3, 3, 3, 3, 3, 1,
2, 2, 0, 1, 0, 1, 0, 3, 2, 3, 3, 0, 2, 0, 1, 3, 3, 1, 2, 2, 1, 3, 1, 3, 1, 1, 2, 2, 3, 0, 1,
3, 2, 3, 3, 2, 1, 1, 1, 1, 2, 2, 0, 1, 1, 2, 0, 3, 2, 0, 2, 1, 2, 1, 2, 0, 0, 1, 2, 0, 2, 2,
0, 1, 3, 2, 1, 3, 3, 2, 2, 2, 3, 1, 0, 0, 2, 0, 1, 2, 1, 1, 0, 1, 2, 2, 1, 3, 2, 1, 2, 3, 1,
```

1, 3, 3, 1, 3, 1},
{0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 1, 2, 2, 2, 2, 3, 3, 3, 2, 3, 3,
3, 2, 3, 2, 0, 1, 2, 0, 2, 3, 1, 3, 2, 2, 0, 2, 3, 2, 2, 3, 0, 3, 1, 2, 2, 3, 3, 2, 1, 0, 3,
3, 0, 1, 0, 0, 2, 1, 3, 1, 1, 2, 0, 3, 3, 3, 3, 1, 2, 0, 0, 2, 0, 3, 3, 2, 3, 3, 0, 2, 3, 3,
1, 2, 1, 0, 3, 3, 0, 3, 1, 3, 1, 1, 3, 2, 2, 0, 1, 2, 1, 1, 0, 2, 0, 1, 2, 0, 1, 1, 2, 2, 2,
3, 3, 2, 3, 3, 2},
{0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 2, 2, 0, 2, 1, 0, 0, 0, 0, 3, 3, 2, 2, 3,
1, 3, 3, 1, 3, 1, 2, 2, 1, 3, 1, 1, 1, 2, 2, 3, 2, 2, 1, 3, 1, 2, 1, 2, 3, 0, 1, 3, 1, 2, 3,
1, 0, 3, 2, 3, 0, 2, 3, 3, 1, 1, 0, 3, 1, 1, 2, 2, 0, 0, 0, 0, 3, 1, 1, 3, 1, 0, 2, 0, 1, 2,
0, 3, 2, 2, 1, 1, 0, 0, 0, 0, 0, 3, 3, 1, 0, 0, 1, 2, 1, 1, 0, 2, 3, 3, 1, 3, 2, 2, 2, 2, 3,
0, 1, 2, 2, 1, 1},
{0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 1, 0, 2, 2, 3, 3, 3, 0, 3, 1, 2, 2,
0, 1, 1, 0, 3, 1, 0, 2, 0, 3, 0, 1, 2, 1, 2, 3, 3, 0, 3, 3, 0, 2, 1, 0, 0, 0, 3, 1, 2, 0, 3,
0, 0, 1, 2, 3, 3, 0, 1, 3, 3, 1, 2, 1, 0, 2, 3, 0, 0, 3, 0, 0, 2, 1, 2, 1, 2, 0, 2, 2, 1, 3,
3, 3, 3, 3, 0, 2, 0, 0, 1, 2, 1, 3, 0, 0, 2, 3, 2, 0, 3, 1, 3, 3, 0, 2, 3, 3, 0, 3, 3, 2, 0,
0, 3, 0, 2, 2, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 2, 0, 3, 1, 0, 2, 2, 3, 3, 3, 0, 3, 1, 2,
2, 0, 1, 1, 0, 3, 1, 0, 2, 0, 3, 0, 1, 2, 1, 2, 3, 3, 0, 3, 3, 0, 2, 1, 0, 0, 0, 3, 1, 2, 0,
3, 3, 0, 1, 2, 3, 3, 0, 1, 3, 3, 1, 2, 1, 0, 2, 3, 0, 0, 3, 0, 0, 2, 1, 2, 1, 2, 0, 2, 2, 1,
3, 3, 3, 3, 3, 0, 2, 0, 0, 1, 2, 1, 3, 0, 0, 2, 3, 2, 0, 3, 1, 3, 3, 0, 2, 3, 3, 0, 3, 3, 2,
0, 0, 3, 0, 1, 2},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 3, 1, 2, 3, 1, 2, 0, 0, 1, 3, 3, 1, 3, 1,
0, 2, 1, 3, 0, 1, 0, 1, 1, 3, 2, 3, 2, 1, 2, 2, 2, 2, 2, 0, 1, 1, 1, 2, 1, 0, 2, 2, 0, 0, 2, 1,
2, 3, 0, 3, 2, 2, 3, 1, 0, 1, 3, 1, 2, 0, 3, 1, 3, 2, 2, 0, 3, 1, 1, 0, 1, 1, 2, 3, 0, 1, 3,
2, 1, 3, 0, 2, 1, 3, 2, 3, 1, 2, 0, 3, 1, 2, 2, 3, 0, 1, 0, 2, 3, 2, 0, 0, 3, 1, 0, 3, 3, 2,
0, 2, 3, 3, 3, 3},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 3, 1, 0, 2, 3, 0, 3, 1, 1, 1, 3, 0, 1, 3,
0, 0, 1, 0, 0, 3, 3, 0, 2, 2, 2, 2, 2, 1, 0, 2, 1, 0, 3, 0, 0, 0, 0, 2, 1, 3, 2, 2, 2, 0,
0, 0, 1, 2, 1, 2, 2, 2, 1, 0, 1, 2, 3, 3, 1, 0, 2, 0, 3, 2, 0, 0, 2, 0, 0, 3, 3, 1, 3, 2, 2,
1, 3, 1, 1, 3, 3, 3, 3, 0, 0, 3, 3, 1, 2, 0, 3, 1, 1, 2, 1, 3, 3, 0, 0, 0, 3, 2, 3, 2, 3, 0,
3, 1, 1, 3, 2, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 3, 1, 0, 2, 3, 0, 3, 1, 1, 1, 3, 0, 1,
3, 0, 0, 1, 0, 0, 3, 3, 0, 2, 2, 2, 2, 2, 2, 1, 0, 2, 1, 0, 3, 0, 0, 0, 0, 2, 1, 3, 2, 2, 2,
0, 1, 0, 1, 2, 1, 2, 2, 2, 1, 0, 1, 2, 3, 3, 1, 0, 2, 0, 3, 2, 0, 0, 2, 0, 0, 3, 3, 1, 3, 2,
2, 1, 3, 1, 1, 3, 3, 3, 3, 0, 0, 3, 3, 1, 2, 0, 3, 1, 1, 2, 1, 3, 3, 0, 0, 0, 3, 2, 3, 2, 3,
0, 3, 1, 1, 2, 2},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 3, 3, 1, 0, 2, 3, 0, 3, 1, 1, 1, 3, 0,
1, 3, 0, 0, 1, 0, 0, 3, 3, 0, 2, 2, 2, 2, 2, 2, 1, 0, 2, 1, 0, 3, 0, 0, 0, 0, 0, 2, 1, 3, 2, 2,
2, 3, 1, 0, 1, 2, 1, 2, 2, 2, 1, 0, 1, 2, 3, 3, 1, 0, 2, 0, 3, 2, 0, 0, 2, 0, 0, 3, 3, 1, 3,
2, 2, 1, 3, 1, 1, 3, 3, 3, 3, 0, 0, 3, 3, 1, 2, 0, 3, 1, 1, 2, 1, 3, 3, 0, 0, 0, 3, 2, 3, 2,
3, 0, 0, 1, 1, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 3, 2, 2, 2, 3, 1, 1, 3, 2, 1, 3, 1, 2, 1, 3,
1, 1, 0, 1, 3, 2, 2, 0, 0, 0, 1, 2, 3, 2, 2, 0, 2, 2, 2, 1, 0, 1, 2, 2, 3, 1, 1, 2, 3, 1, 0,
3, 2, 1, 3, 2, 1, 2, 0, 2, 2, 2, 0, 2, 0, 3, 0, 0, 2, 1, 2, 0, 0, 1, 1, 0, 0, 2, 3, 3, 1, 2,
1, 3, 2, 3, 0, 0, 3, 3, 1, 0, 1, 1, 1, 2, 2, 0, 1, 2, 1, 1, 2, 3, 2, 1, 3, 3, 1, 2, 1, 2, 0,
3, 2, 1, 0, 0, 1},

{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 2, 3, 3, 3, 1, 0, 1, 3, 0, 2, 1,
1, 1, 0, 2, 0, 2, 1, 2, 1, 1, 2, 1, 0, 3, 2, 1, 0, 3, 1, 3, 3, 2, 3, 1, 3, 1, 3, 1, 1, 0, 2,
2, 1, 1, 2, 0, 2, 1, 0, 0, 2, 2, 0, 3, 0, 2, 2, 1, 1, 0, 1, 2, 1, 1, 3, 1, 3, 3, 3, 3, 0, 0,
1, 3, 3, 1, 2, 2, 3, 3, 0, 0, 3, 3, 3, 3, 0, 0, 1, 2, 1, 1, 0, 0, 2, 1, 1, 2, 1, 2, 1, 1, 3,
2, 1, 0, 1, 0, 1},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 3, 3, 2, 0, 0, 2, 2, 2, 2, 0, 0, 1, 0, 0, 2,
0, 1, 2, 1, 1, 3, 0, 1, 1, 2, 0, 2, 0, 0, 3, 0, 1, 3, 1, 2, 2, 2, 3, 1, 2, 2, 0, 3, 3, 3, 2,
3, 3, 3, 3, 0, 0, 2, 0, 0, 0, 2, 3, 2, 2, 1, 1, 1, 2, 0, 0, 1, 1, 2, 0, 3, 3, 1, 0, 3, 1, 3,
2, 0, 3, 1, 2, 3, 0, 3, 1, 3, 2, 2, 2, 2, 2, 1, 3, 3, 0, 1, 2, 1, 3, 0, 1, 2, 3, 3, 0, 1, 2,
2, 3, 0, 0, 1, 1}}

A.2 Tables for Message Expansion of Hamsi-512 and Hamsi-384

gen512[32][256] = {
{1,0,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,2,2,0,1,2,3,3,2,0,0,2,1,3,3,2,1,1,0,2,0,2,1,1,3,2,2,1,3,0,3,2,3,2,0,1,1,0,2,2,0,3,0,0,3,2,2,2,0,1,1,2,0,0,2,1,2,1,0,2,0,3,0,
3,2,3,1,3,2,1,2,2,1,2,0,1,0,1,0,3,3,1,0,1,0,0,1,3,3,1,3,3,1,2,2,3,1,3,2,1,2,3,3,2,0,0,2,1,3,0,3,0,1,3,1,3,2,1,2,1,2,2,1,0,0,1,0,0,1,3,
3,1,3,2,2,1,3,3,3,1,2,2,1,2,2,2,2,2,2,3,1,3,2,1,2,1,3,3,1,2,2,3,3,3,3,3,3,0,3,0,2,3,2,3,2,1,1,3,2,1,3,0,3,0,1},
{0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,1,0,2,2,0,0,1,3,3,1,2,0,2,3,0,3,2,0,0,2,1,1,1,0,1,2,0,2,2,1,2,0,1,0,2,1,1,2,3,3,2,3,2,1,3,1,2,1,3,3,2,1,1,3,1,0,3,0,3,0,0,3,
2,2,1,1,1,1,1,1,1,1,3,0,3,0,1,1,3,2,2,1,3,3,1,3,2,1,2,2,3,3,2,0,0,1,1,1,1,1,1,0,2,0,1,2,1,0,1,2,2,0,1,1,1,1,1,1,1,0,0,3,2,2,3,3,1,3,2,1,
2,1,0,1,2,0,2,2,0,1,1,2,0,2,3,1,3,1,2,1,1,1,1,1,1,1,2,3,3,2,0,0,3,1,2,1,2,3,3,0,1,0,1,3,3,0,3,1,0,1,0,1,2,2,0,1},
{0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,3,2,1,3,1,1,3,3,1,2,2,1,0,1,2,0,2,0,2,3,3,0,2,2,1,0,1,0,2,3,2,1,1,3,2,0,1,2,2,0,1,3,3,2,0,0,2,3,3,0,1,1,0,1,1,1,1,1,1,3,3,2,0,
0,2,2,2,0,3,3,0,1,1,3,0,0,3,0,2,0,1,2,1,2,0,1,1,2,0,3,2,2,3,1,1,3,1,2,1,2,3,1,0,0,1,3,3,1,3,0,3,0,1,3,3,1,2,2,1,3,0,0,3,2,2,1,0,1,2,0,
2,2,3,3,2,0,0,0,2,3,3,0,2,2,1,0,1,0,2,0,1,2,2,0,1,1,0,2,0,2,1,0,2,0,1,2,1,3,0,1,0,1,3,1,2,3,2,3,1,3,1,1,3,0,0},
{0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,0,1,0,1,3,1,2,0,0,1,2,2,0,1,1,2,0,1,3,1,0,3,0,0,2,1,2,1,0,2,2,3,1,1,3,3,2,3,0,2,0,3,1,1,3,0,0,0,3,3,0,1,1,2,0,0,2,0,1,1,3,1,1,3,
0,0,0,2,2,0,3,3,1,2,2,1,0,0,0,0,2,1,1,2,2,3,1,3,1,2,3,1,0,0,3,1,3,1,3,2,1,2,0,1,0,3,1,3,0,1,3,1,3,0,2,2,2,2,2,2,2,0,3,0,2,3,2,2,0,1,1,2,
0,1,1,0,2,2,0,1,3,1,0,3,0,0,2,1,2,1,0,3,2,3,0,2,0,3,3,2,0,0,2,0,0,2,1,1,2,2,2,1,0,0,1,1,2,1,3,2,3,2,2,0,3,3,0},
{0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,2,1,0,0,1,0,1,2,2,0,1,2,3,1,3,1,2,2,0,2,3,0,3,2,1,3,3,2,1,1,1,1,1,1,1,1,0,1,2,0,2,2,2,2,0,3,3,0,1,3,0,3,0,1,0,2,0,1,2,1,2,2,0,3,
3,0,3,2,0,2,0,3,3,3,0,1,1,0,3,2,2,3,1,1,2,3,2,1,3,1,0,3,1,1,0,3,1,0,2,0,2,1,0,0,1,3,3,1,1,3,2,2,1,3,3,0,0,3,2,2,0,0,3,2,2,3,3,2,3,1,3,1,
2,0,1,1,0,2,2,2,0,2,3,0,3,2,1,3,3,2,1,1,0,1,2,0,2,3,1,1,3,0,0,3,2,2,3,1,1,2,3,3,2,0,0,2,0,3,0,3,2,0,2,2,0,3,3},
{0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,2,1,0,0,1,2,0,1,1,2,0,1,3,2,2,1,3,0,1,1,0,2,2,2,3,2,1,3,1,1,3,0,3,0,1,3,2,3,0,2,0,2,1,1,2,3,3,0,2,3,3,0,2,0,2,3,3,0,2,0,3,3,0,
1,1,0,1,1,0,2,2,1,3,0,3,0,1,3,0,3,1,0,1,0,3,0,2,3,2,1,0,2,0,2,1,2,3,1,3,1,2,3,3,3,3,3,3,3,1,0,0,3,1,3,3,1,2,2,1,3,1,2,1,2,3,0,1,1,0,2,
2,2,2,2,2,2,2,2,0,0,2,1,1,0,0,2,1,1,2,0,1,0,3,1,3,3,2,3,0,2,0,2,3,1,3,1,2,3,1,1,3,0,0,3,3,1,2,2,1,2,3,2,1,3,1},
{0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,3,3,2,0,0,2,3,1,3,1,2,3,3,1,2,2,1,2,1,0,1,0,2,3,0,1,0,1,3,0,1,3,1,3,0,1,0,1,2,0,2,2,3,0,0,2,3,1,3,1,0,3,0,1,3,1,0,3,0,1,3,0,3,
0,1,2,1,0,1,0,2,0,1,3,1,3,0,1,0,3,3,1,0,0,0,3,2,2,3,3,3,2,0,0,2,2,3,2,1,3,1,1,0,0,1,3,3,0,3,1,1,0,3,2,2,2,2,2,2,3,1,3,2,1,2,2,1,0,1,0,
2,3,0,0,3,2,2,0,2,0,1,2,1,3,2,2,3,1,1,0,0,1,3,3,1,1,0,1,2,0,2,2,3,2,1,3,1,2,2,0,3,3,0,2,2,2,2,2,2,3,0,1,0,1,3},

{0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,1,0,2,2,0,2,3,2,1,3,1,2,2,2,2,2,2,0,2,1,2,1,0,2,2,1,0,0,1,1,3,2,2,1,3,2,0,1,1,2,0,0,2,3,3,0,2,2,0,2,3,0,3,2,0,2,3,0,3,0,1,3,1,
3,0,0,2,1,2,1,0,1,3,2,2,1,3,1,2,3,2,3,1,1,3,3,1,2,2,3,1,1,3,0,0,3,0,1,0,1,3,0,1,0,3,1,3,2,1,2,0,1,0,3,0,0,3,2,2,1,0,2,0,2,1,0,2,1,2,1,
0,0,3,0,2,3,2,0,0,2,1,1,2,3,1,0,0,3,1,2,1,1,2,3,3,2,0,1,1,2,0,3,0,1,0,1,3,0,2,2,0,3,3,3,0,0,3,2,2,2,2,1,0,0,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,0,3,1,3,3,0,1,0,1,3,1,3,3,1,2,2,3,2,3,0,2,0,3,0,0,3,2,2,2,0,3,0,3,2,0,2,3,3,0,2,0,2,3,3,0,2,2,1,3,3,2,1,1,3,2,2,1,3,0,2,0,1,
2,1,2,2,3,1,1,3,3,2,1,1,3,2,1,1,3,0,0,3,2,0,1,1,2,0,0,2,3,3,0,2,1,3,2,2,1,3,3,0,0,3,2,2,3,1,0,0,3,1,1,0,3,3,1,0,3,1,3,2,1,2,3,1,0,0,3,
1,2,0,1,1,2,0,3,0,2,2,3,0,3,3,1,2,2,1,1,2,3,2,3,1,2,1,1,2,3,3,0,3,1,1,0,3,0,3,0,2,3,2,0,1,0,3,1,3,1,0,3,3,1,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,3,3,1,2,2,1,0,0,1,1,2,0,0,1,2,1,0,1,2,0,2,0,3,0,2,3,2,1,1,3,0,0,3,1,3,1,0,3,0,1,3,1,0,3,0,1,1,2,3,3,2,3,3,1,2,2,1,0,0,2,1,
1,2,1,1,1,1,1,1,2,2,3,1,1,3,1,2,2,1,0,0,2,3,1,3,1,2,1,3,1,0,3,0,3,3,1,2,2,1,0,3,0,2,3,2,0,3,1,1,0,3,1,2,3,2,3,1,1,0,2,0,2,1,0,3,1,1,0,
3,2,3,1,3,1,2,3,1,2,1,2,3,2,2,2,2,2,2,1,2,1,3,2,3,2,3,0,0,2,3,2,1,2,0,1,0,0,0,3,2,2,3,0,0,1,3,3,1,1,2,3,2,3,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,1,1,2,3,3,2,3,3,2,0,0,0,1,2,2,0,1,2,0,1,1,2,0,0,0,3,2,2,3,1,2,2,1,0,0,2,0,2,3,0,3,2,0,2,3,0,3,3,3,3,3,3,3,2,2,2,2,2,2,3,2,2,3,
1,1,2,0,0,2,1,1,1,1,1,1,1,1,3,3,0,1,1,0,2,3,2,1,3,1,2,0,2,3,0,3,2,2,2,2,2,2,0,0,3,2,2,3,2,1,2,0,1,0,1,2,1,3,2,3,3,3,2,0,0,2,2,1,2,0,1,
0,2,3,2,1,3,1,3,1,3,2,1,2,3,0,0,3,2,2,2,0,3,0,3,2,0,2,3,3,0,2,3,0,3,1,0,1,1,3,3,1,2,2,2,1,1,2,3,3,1,2,1,3,2,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,1,0,0,3,1,2,1,2,0,1,0,3,3,2,0,0,2,3,2,1,1,3,2,2,1,3,3,2,1,1,0,1,2,0,2,2,1,2,0,1,0,2,3,2,1,3,1,0,3,1,1,0,3,2,1,0,1,0,2,1,1,1,1,
1,1,2,0,3,0,3,2,2,0,2,3,0,3,2,2,2,2,2,2,2,2,3,1,1,3,1,3,2,2,1,3,2,1,3,3,2,1,0,1,1,0,2,2,2,0,3,0,3,2,1,1,0,2,2,0,0,3,1,1,0,3,3,0,1,0,1,
3,1,3,3,1,2,2,0,0,3,2,2,3,2,2,1,0,0,1,1,0,2,0,2,1,0,2,1,2,1,0,3,0,2,2,3,0,1,3,3,1,2,2,2,1,0,1,0,2,1,1,3,0,0,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,3,1,1,0,3,3,0,3,1,0,1,3,1,1,3,0,0,2,2,3,1,1,3,1,1,2,3,3,2,2,0,1,1,2,0,3,0,3,1,0,1,3,0,1,0,1,3,2,1,2,0,1,0,0,2,1,2,1,0,2,0,0,2,
1,1,1,1,3,0,0,3,3,0,2,2,3,0,3,0,0,3,2,2,1,1,1,1,1,1,3,3,1,2,2,1,1,1,2,3,3,2,2,1,0,1,0,2,1,1,3,0,0,3,0,1,1,0,2,2,2,1,2,0,1,0,2,2,1,0,0,
1,1,2,0,0,1,2,1,3,3,1,2,2,2,3,3,2,0,0,3,3,2,0,0,2,2,1,3,3,2,1,3,1,2,1,2,3,1,2,0,0,1,2,0,2,1,2,1,0,1,2,2,1,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,1,2,0,1,0,1,0,3,3,1,0,2,2,0,3,3,0,1,1,1,1,1,1,3,3,3,3,3,3,2,3,1,3,1,2,1,0,3,3,1,0,2,2,1,0,0,1,3,0,3,1,0,1,2,1,3,3,2,1,0,2,0,1,
2,1,1,2,2,1,0,0,3,1,2,1,2,3,0,3,0,2,3,2,2,0,0,2,1,1,2,2,2,2,2,2,3,3,3,3,3,3,0,2,1,2,1,0,1,2,2,1,0,0,2,1,0,1,0,2,3,0,3,1,0,1,2,3,3,2,0,
0,0,1,2,2,0,1,1,2,0,0,1,2,1,1,0,2,2,0,3,1,1,3,0,0,1,1,2,3,3,2,3,1,3,2,1,2,0,1,2,2,0,1,2,1,3,3,2,1,3,3,0,1,1,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,2,3,1,1,3,3,0,0,3,2,2,3,3,1,2,2,1,1,3,2,2,1,3,2,0,0,2,1,1,2,1,0,1,0,2,3,3,0,1,1,0,0,0,1,3,3,1,0,1,0,3,1,3,0,2,2,0,3,3,1,2,3,2,
3,1,2,0,0,2,1,1,1,0,2,0,2,1,2,0,0,2,1,1,3,2,1,1,3,2,3,3,3,3,3,3,0,2,0,1,2,1,1,3,1,0,3,0,0,1,3,1,3,0,2,2,1,0,0,1,0,0,3,2,2,3,2,1,3,3,2,
1,0,1,0,3,1,3,1,2,3,2,3,1,3,2,1,1,3,2,3,2,0,2,0,3,1,2,1,3,2,3,2,3,3,2,0,0,3,2,1,1,3,2,2,3,1,3,1,2,3,0,3,1,0,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,1,1,1,1,1,0,3,0,2,3,2,2,2,2,2,2,2,3,3,1,2,2,1,0,2,0,1,2,1,0,2,1,2,1,0,0,3,3,0,1,1,2,1,1,2,3,3,0,0,1,3,3,1,3,2,0,2,0,3,1,2,1,3,
2,3,0,2,0,1,2,1,3,3,2,0,0,2,0,2,0,1,2,1,2,2,3,1,1,3,1,0,0,1,3,3,0,0,2,1,1,2,2,0,2,3,0,3,1,3,2,2,1,3,2,3,3,2,0,0,1,3,3,1,2,2,1,1,2,3,3,
2,0,0,1,3,3,1,1,2,1,3,2,3,2,2,3,1,1,3,0,3,2,3,2,0,2,0,3,0,3,2,1,1,0,2,2,0,2,2,3,1,1,3,2,3,2,1,3,1,1,0,3,3,1,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,0,0,2,1,1,0,0,3,2,2,3,3,0,0,3,2,2,2,2,2,2,2,2,0,0,2,1,1,2,2,1,3,3,2,1,1,3,0,3,0,1,2,3,0,0,2,3,2,1,1,2,3,3,0,3,2,3,2,0,2,0,3,0,
3,2,0,0,2,1,1,2,3,1,1,3,0,0,0,0,2,1,1,2,1,1,1,1,1,1,0,1,0,3,1,3,3,2,2,3,1,1,3,0,2,2,3,0,3,3,1,2,2,1,1,1,0,2,2,0,1,2,0,0,1,2,3,3,3,3,3,
3,2,1,1,2,3,3,2,0,3,0,3,2,1,1,1,1,1,1,3,2,1,1,3,2,1,1,3,0,0,3,0,1,1,0,2,2,1,1,1,1,1,1,3,0,1,0,1,3,1,2,3,2,3,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,3,2,0,0,2,2,3,2,1,3,1,3,0,2,2,3,0,1,1,1,1,1,1,2,1,0,1,0,2,3,3,0,1,1,0,2,1,3,3,2,1,0,3,1,1,0,3,1,2,3,2,3,1,1,3,1,0,3,0,2,1,1,2,
3,3,0,1,2,2,0,1,3,2,0,2,0,3,2,2,2,2,2,2,2,2,0,3,3,0,2,2,0,3,3,0,1,1,2,3,3,2,0,1,2,2,0,1,2,1,1,2,3,3,3,1,1,3,0,0,2,1,0,1,0,2,1,2,1,3,2,
3,3,2,2,3,1,1,0,0,1,3,3,1,1,1,1,1,1,1,0,2,1,2,1,0,2,3,2,1,3,1,0,3,0,2,3,2,3,2,1,1,3,2,3,3,1,2,2,1,1,3,2,2,1,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,1,1,3,0,0,3,0,1,0,1,3,3,1,2,1,2,3,2,0,0,2,1,1,0,2,1,2,1,0,0,3,3,0,1,1,1,1,2,3,3,2,2,1,2,0,1,0,1,2,1,3,2,3,2,0,2,3,0,3,2,3,0,0,

2,3,3,2,3,0,2,0,0,3,2,3,2,0,3,0,0,3,2,2,0,2,2,0,3,3,0,2,2,0,3,3,3,3,3,3,3,3,2,3,0,2,0,2,3,0,0,2,3,2,2,0,3,3,0,0,2,1,2,1,0,2,0,3,0,3,
2,3,1,0,0,3,1,2,1,1,2,3,3,2,0,0,2,1,1,2,1,3,3,2,1,3,0,1,0,1,3,0,0,3,2,2,3,2,2,3,1,1,3,2,2,2,2,2,2,3,3,1,2,2,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,2,0,3,3,0,2,2,1,0,0,1,3,1,3,2,1,2,0,2,0,1,2,1,2,1,3,3,2,1,1,3,0,3,0,1,3,3,3,3,3,3,3,0,3,1,0,1,2,0,3,0,3,2,3,0,2,2,3,0,0,2,3,3,
0,2,1,0,1,2,0,2,3,2,1,1,3,2,0,3,0,2,3,2,3,2,0,2,0,3,3,2,0,2,0,3,1,0,0,1,3,3,1,0,1,2,0,2,0,2,3,3,0,2,0,2,2,0,3,3,2,1,3,3,2,1,1,1,3,0,0,
3,0,3,1,1,0,3,2,3,0,0,2,3,0,2,0,1,2,1,1,1,2,3,3,2,2,2,1,0,0,1,1,3,3,1,2,2,1,1,1,1,1,1,3,0,0,3,2,2,2,2,2,2,2,2},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,0,3,1,3,2,1,2,0,1,0,1,0,0,1,3,3,1,1,3,0,0,3,2,0,0,2,1,1,2,2,1,0,0,1,2,2,3,1,1,3,1,1,3,0,0,3,0,3,3,0,1,1,0,3,2,3,2,0,1,3,2,2,
1,3,0,0,1,3,3,1,3,2,2,3,1,1,3,0,3,1,0,1,3,2,0,2,0,3,2,3,3,2,0,0,2,1,0,1,0,2,0,2,3,3,0,2,0,3,1,1,0,3,0,0,2,1,1,2,3,3,3,3,3,3,2,1,2,0,1,
0,3,3,1,2,2,1,2,0,2,3,0,3,1,3,0,3,0,1,0,3,0,2,3,2,2,2,0,3,3,0,1,2,0,0,1,2,2,3,1,3,1,2,0,2,0,1,2,1,2,0,3,0,3,2},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,0,1,3,3,1,3,0,3,1,0,1,0,1,0,3,1,3,1,2,2,1,0,0,0,2,0,1,2,1,2,3,3,2,0,0,1,1,1,1,1,1,2,2,1,0,0,1,3,0,3,0,1,3,2,1,1,3,2,3,3,1,2,
2,1,2,1,1,2,3,3,3,1,0,0,3,1,1,0,3,3,1,0,0,3,2,3,2,0,1,1,0,2,2,0,0,2,1,2,1,0,1,3,1,0,3,0,2,1,2,0,1,0,3,2,2,3,1,1,1,0,0,1,3,3,3,0,3,1,0,
1,2,2,2,2,2,2,3,0,2,2,3,0,0,1,3,1,3,0,0,0,3,2,2,3,0,2,2,0,3,3,0,1,2,2,0,1,2,3,2,1,3,1,0,0,2,1,1,2,1,1,3,0,0,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,2,1,1,2,3,3,1,0,3,3,1,0,0,0,1,3,3,1,3,3,0,1,1,0,0,0,2,1,1,2,1,1,0,2,2,0,2,0,0,2,1,1,3,3,0,1,1,0,0,1,3,1,3,0,2,2,3,1,1,3,2,2,2,2,
2,2,2,3,0,0,2,3,0,3,1,1,0,3,1,2,3,2,3,1,3,2,1,1,3,2,0,1,1,0,2,2,2,1,3,3,2,1,2,0,2,3,0,3,3,0,3,1,0,1,3,1,0,0,3,1,0,1,0,3,1,3,1,0,3,3,1,
0,3,0,0,3,2,2,3,1,2,1,2,3,1,3,2,2,1,3,1,3,3,1,2,2,3,2,0,2,0,3,3,2,3,0,2,0,3,0,1,0,1,3,3,2,2,3,1,1,1,2,2,1,0,0}
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,2,0,0,1,2,3,1,3,2,1,2,2,0,3,0,3,2,2,2,3,1,1,3,3,0,3,1,0,1,1,0,2,0,2,1,2,0,3,0,3,2,1,0,1,2,0,2,1,3,0,3,0,1,0,3,2,3,2,0,0,0,3,2,
2,3,2,0,3,0,3,2,0,3,3,0,1,1,1,2,3,2,3,1,2,1,3,3,2,1,2,3,1,3,1,2,1,2,3,2,3,1,0,2,1,2,1,0,2,2,2,2,2,2,3,3,1,2,2,1,2,1,1,2,3,3,3,1,0,0,3,
1,1,2,3,2,3,1,0,3,2,3,2,0,0,3,3,0,1,1,2,2,1,0,0,1,2,3,3,2,0,0,0,0,1,3,3,1,3,2,3,0,2,0,1,0,1,2,0,2,2,2,3,1,1,3}
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,0,1,2,2,0,1,1,0,2,0,2,1,1,1,3,0,0,3,1,1,1,1,1,1,1,0,3,3,1,0,3,3,2,0,0,2,1,1,3,0,0,3,2,0,1,1,2,0,0,1,3,1,3,0,3,2,1,1,3,2,1,3,3,1,
2,2,1,1,3,0,0,3,1,3,0,3,0,1,1,2,1,3,2,3,1,1,2,3,3,2,2,3,2,1,3,1,1,2,1,3,2,3,2,1,3,3,2,1,3,0,0,3,2,2,2,2,2,2,2,2,3,0,0,2,3,0,3,1,1,0,
3,1,2,1,3,2,3,3,2,1,1,3,2,1,3,0,3,0,1,2,3,3,2,0,0,1,1,0,2,2,0,2,1,1,2,3,3,1,0,1,2,0,2,2,0,1,1,2,0,1,1,1,1,1,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,2,3,0,2,0,3,3,2,0,0,2,1,2,2,1,0,0,2,0,0,2,1,1,1,2,3,2,3,1,3,1,1,3,0,0,1,2,2,1,0,0,2,3,1,3,1,2,1,3,2,2,1,3,2,2,3,1,1,3,1,2,0,0,
1,2,1,2,2,1,0,0,0,1,3,1,3,0,2,0,3,0,3,2,3,3,3,3,3,3,3,0,1,0,1,3,2,0,3,0,3,2,1,1,2,3,3,2,0,3,0,2,3,2,3,0,0,3,2,2,0,2,3,3,0,2,2,1,2,0,1,
0,2,0,3,0,3,2,2,2,3,1,1,3,0,1,3,1,3,0,1,1,0,2,2,0,0,1,1,0,2,2,2,3,0,0,2,3,2,0,1,1,2,0,2,3,1,3,1,2,2,0,0,2,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,1,3,2,1,2,3,1,3,2,1,2,1,1,3,0,0,3,1,3,1,0,3,0,2,0,3,0,3,2,0,3,3,0,1,1,2,0,2,3,0,3,0,3,0,3,2,0,3,0,2,2,3,0,0,3,3,0,1,1,3,3,3,3,
3,3,2,3,3,2,0,0,2,0,0,2,1,1,2,1,3,3,2,1,0,3,2,3,2,0,2,0,1,1,2,0,0,0,3,2,2,3,1,0,1,2,0,2,2,0,1,1,2,0,3,1,3,2,1,2,0,1,3,1,3,0,0,1,2,2,0,
1,1,2,0,0,1,2,3,1,0,0,3,1,0,1,0,3,1,3,3,0,2,2,3,0,2,0,0,2,1,1,1,1,2,3,3,2,0,2,0,1,2,1,0,1,0,3,1,3,3,2,2,3,1,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,
0,0,0,1,0,2,0,2,1,1,0,2,0,2,1,1,2,2,1,0,0,2,0,2,3,0,3,1,1,3,0,0,3,1,3,0,3,0,1,3,0,2,2,3,0,3,2,1,1,3,2,3,1,2,1,2,3,1,3,0,3,0,1,1,0,0,1,
3,3,1,1,0,2,2,0,0,2,0,1,2,1,1,1,2,3,3,2,3,2,1,1,3,2,2,3,1,3,1,2,1,3,3,1,2,2,2,0,1,1,2,0,2,3,1,3,1,2,1,0,2,0,2,1,1,3,2,2,1,3,3,2,3,0,2,
0,0,1,2,2,0,1,0,3,1,1,0,3,0,0,1,3,3,1,3,1,2,1,2,3,0,2,0,1,2,1,3,3,3,3,3,3,0,0,2,1,1,2,0,0,1,3,3,1,3,1,0,0,3,1},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,0,0,
0,0,0,3,3,2,0,0,2,3,3,2,0,0,2,3,3,0,1,1,0,3,0,2,2,3,0,1,2,2,1,0,0,0,1,3,1,3,0,3,1,2,1,2,3,2,2,3,1,1,3,3,1,3,2,1,2,0,1,3,1,3,0,0,1,0,3,
1,3,0,1,1,0,2,2,0,0,2,1,1,2,3,3,3,3,3,3,2,2,3,1,1,3,2,3,2,1,3,1,1,2,0,0,1,2,2,3,1,3,1,2,2,3,2,1,3,1,3,3,2,0,0,2,3,3,1,2,2,1,1,0,1,2,0,
2,3,2,3,0,2,0,2,1,2,0,1,0,2,1,1,2,3,3,3,1,3,2,1,2,0,0,2,1,1,2,1,0,0,1,3,3,3,2,2,3,1,1,2,1,1,2,3,3,0,3,1,1,0,3},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,0,0,0,0,0,
0,0,0,3,1,0,0,3,1,1,1,1,1,1,1,3,2,0,2,0,3,3,0,1,0,1,3,0,0,1,3,3,1,0,0,2,1,1,2,0,2,2,0,3,3,0,1,3,1,3,0,1,3,0,1,0,3,
1,3,1,1,0,2,2,0,2,1,2,0,1,0,3,0,3,1,0,1,1,2,3,2,3,1,3,0,1,0,1,3,0,3,1,1,0,3,1,2,0,0,1,2,0,0,2,1,1,2,0,0,1,3,3,1,2,2,1,0,0,1,0,0,1,3,3,
1,2,3,1,3,1,2,3,2,2,3,1,1,2,3,3,2,0,0,2,2,3,1,1,3,2,0,0,2,1,1,3,1,3,2,1,2,0,1,0,3,1,3,2,0,0,2,1,1,3,3,2,0,0,2},

{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,3,1,3,0,0,0,
0,0,0,0,3,1,1,0,3,2,0,0,2,1,1,0,3,2,3,2,0,2,2,1,0,0,1,2,1,1,2,3,3,3,2,2,3,1,1,3,2,0,2,0,3,1,3,2,2,1,3,0,0,2,1,1,2,2,2,1,0,0,1,0,0,1,3,
3,1,0,1,1,0,2,2,3,0,3,1,0,1,1,0,3,3,1,0,1,2,1,3,2,3,2,2,1,0,0,1,2,1,2,0,1,0,0,1,2,2,0,1,3,2,2,3,1,1,2,1,1,2,3,3,2,3,3,2,0,0,2,1,1,2,3,
3,2,3,2,1,3,1,3,1,0,0,3,1,1,1,0,2,2,0,1,1,1,1,1,1,0,2,0,1,2,1,1,0,2,0,2,1,0,0,1,3,3,1,0,2,0,1,2,1,3,1,1,3,0,0},
{0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,0,0,0,
0,0,0,2,1,2,0,1,0,0,2,0,1,2,1,3,2,1,1,3,2,2,3,3,2,0,0,2,3,0,0,2,3,3,1,0,0,3,1,0,3,2,3,2,0,3,3,1,2,2,1,3,2,2,3,1,1,2,3,3,2,0,0,2,1,1,2,
3,3,2,1,0,1,0,2,1,0,3,3,1,0,1,2,3,2,3,1,2,0,3,0,3,2,2,3,3,2,0,0,3,0,3,1,0,1,3,2,3,0,2,0,3,1,0,0,3,1,2,3,0,0,2,3,1,1,0,2,2,0,2,3,0,0,2,
3,3,0,1,0,1,3,0,3,1,1,0,3,0,1,1,0,2,2,2,0,0,2,1,1,0,0,2,1,1,2,3,3,2,0,0,2,2,1,1,2,3,3,0,0,2,1,1,2,2,2,0,3,3,0}}