A New Standard of Ukraine: The Kupyna Hash Function

Roman Oliynykov[1], Ivan Gorbenko[1], Oleksandr Kazymyrov[4], Victor Ruzhentsev[4], Oleksandr Kuznetsov[3], Yurii Gorbenko[1], Artem Boiko[1], Oleksandr Dyrda[2], Viktor Dolgov[3], Andrii Pushkaryov[2]

[1] JSC Institute of Information Technologies,

[2] State Service of Special Communication and Information Protection of Ukraine,

[3] V.N.Karazin Kharkiv National University,

[4] Kharkiv National University of Radio Electronics

Ukraine

roliynykov@gmail.com, gorbenkoi@iit.kharkov.ua, okazymyrov@gmail.com

Abstract

The Kupyna hash function was approved as the new Ukrainian standard DSTU 7564:2014 in 2015. Main requirements for it were both high security level and good performance of software implementation on general-purpose 64-bit CPUs. The new hash function uses Davies-Meyer compression function based on Even-Mansour cipher construction. Kupyna is built on the transformations of the Kalyna block cipher (Ukrainian standard DSTU 7624:2014 [1, 2]). In this paper we present the adapted English translated specification of the Kupyna hash function as it is described in the national standard of Ukraine.

1 Introducton

Ukraine had been using the Commonwealth of Independent States (CIS) standard GOST 34.311-95 [3] (withdrawn Russian hash standard GOST R 34.11-94 [4]) as the main cryptographic hash function until 2015. Even now this transformation still provides acceptable level of practical security. However, its software implementation is significantly slower and less effective on modern platforms comparing to more recent solutions. In addition, theoretical attacks which are more effective than brute force were discovered [5]. Other CIS countries adopted newer standards instead of GOST R 34.11-94.

The new Ukrainian hash standard DSTU 7564:2014 [6] was developed based on a conservative approach using verified cryptographic constructions [7, 8, 9]. The

Presented at the Norwegian Information Security Conference 2015 (NISK-2015).

new hash function uses the Davies-Meyer compression function based on the Even-Mansour scheme. Internal permutations are built on the transformations of the Kalyna block cipher (the Ukrainian standard DSTU 7624:2014 [1, 2]).

The new standard defines both the hash function and its additional mode for message authentication code generation. In this paper we describe an adapted version of the specification of the Kupyna hash function as it is given in the national standard of Ukraine.

2 Symbols and notations

The following notations are used for the description.

0x	–	a prefix of numbers given in the hexadecimal notation;
$GF(2^8)$	–	the finite field with the irreducible polynomial $f(x) = x^8 + x^4 + x^3 + x^2 + 1$;
$\oplus$	–	logical exclusive OR (XOR) operation for binary vectors;
$\otimes$	–	vector multiplication over the finite field;
$\ll$	–	the left shift of the fixed length sequence (to the most significant symbols); the least significant symbols are filled with 0's; number of symbols to be shifted is defined by the second argument;
$\ggg$	–	the cyclic shift (rotation) right of the fixed length sequence (the least significant symbols are moved to the most significant positions);
$(a_0, a_1, \ldots, a_k)^T$	–	transposition of the row vector $(a_0, a_1, \ldots, a_k)$ into the column $\begin{pmatrix} a_0 \\ a_1 \\ \cdots \\ a_k \end{pmatrix}$;
$a \mod b$	–	a non-negative remainder after dividing $a \in Z$ by $b \in Z^+$
$\omega_j^{(\nu)}, \varsigma_j^{(\nu)}$	–	iteration constants for ν^{th} round of $T_l^{\oplus}$ and T_l^{+} transformations;
H	–	the hash function defined in this description;
$H(IV, M)$	–	a hash code of a message M;
IV	–	an initialization vector;
n	–	the length of the hash code, $n = 8 \cdot s$, $s \in \{1, 2, \ldots, 64\}$;
l	–	the size of the hash function internal state, $l \in \{512, 1024\}$;
M	–	a message for hashing;
m_i	–	i^{th} block of the message M after padding;
N	–	the length (in bits) of the message M without padding;
$R_{l,r}(X)$	–	the function that returns r most significant bits from the input sequence X of l-bit length;
$T_l^{\oplus}$, T_l^{+}	–	the bijective mappings $T_l^{\oplus}, T_l^{+} : V_l \to V_l$, $l \in \{512, 1024\}$ (permutations) that transform an input block of l bit length into output block of the same length;
t	–	the number of rounds (iterations) in $T_l^{\oplus}$ and T_l^{+} transformations;
k	–	the number of blocks in the message M, including padding;
Z^+	–	the set of positive integers;

V_j	–	j-dimensional vector space over $GF(2)$, $j \in Z^+$;
$+$	–	addition operation defined in $Z_{2^{64}}$;
V_j	–	j-dimensional vector space over $GF(2)$, $j \geq 1$;
$\Xi \circ \Lambda$	–	a sequential application of transformations Ξ and Λ (Λ is applied first);
$\prod_{i=1}^{t} \Lambda^{(i)}$	–	a sequential application of the transformations $\Lambda^{(1)}$, $\Lambda^{(2)}$, ..., $\Lambda^{(t)}$ (the transformation $\Lambda^{(1)}$ is applied first);
`Kupyna`-n	–	the notation for the hash function with the hash code of n-bit length.

3 General parameters

The hash function H is an IV-dependent mapping of a message M into the hash code $H(IV, M)$ such that

$$\begin{aligned}
&H^{(IV)} : V_N \to V_n,\ n \in \{8 \cdot s | s = 1, 2, \ldots, 64\},\\
&M \in V_N,\ N \in \{0, 1, \ldots, 2^{96} - 1\},\\
&IV \in V_l,\ l \in \{512, 1024\}.
\end{aligned}$$

Hash function mode of operation for $n \in \{8 \cdot s | s = 1, 2, \ldots, 64\}$ is denoted as `Kupyna`-n. The main recommended modes are `Kupyna-256`, `Kupyna-384` and `Kupyna-512`.

4 Structure of the transformation H

Input message for hashing is always padded (see Section 5) to obtain the length, which is a multiple of the block size. After padding it is divided into the blocks $m_1, m_2, \ldots, m_k$ of l-bit length, which is defined as

$$l = \begin{cases} 512,\ if\ 8 \leq n \leq 256,\\ 1024,\ if\ 256 < n \leq 512. \end{cases}$$

A hash code is obtained accordingly to the following iterative procedure:

$$\begin{aligned}
&h_0 = IV,\\
&h_\nu = T_l^{\oplus}(h_{\nu-1} \oplus m_\nu) \oplus T_l^{+}(m_\nu) \oplus h_{\nu-1}, \qquad \nu = 1, 2, \ldots, k,\\
&H(IV, M) = R_{l,n}(T_l^{\oplus}(h_k) \oplus h_k),
\end{aligned}$$

where

$$IV = \begin{cases} 1 \ll 510,\ \text{if } l = 512,\\ 1 \ll 1023,\ \text{if } l = 1024, \end{cases} \qquad IV \in V_l,$$

$T_l^{\oplus}$, T_l^{+} are bijective mappings of l-bit blocks (see Section 6),

$R_{l,n}(X)$ is a function that returns n most significant bits from the input block X of l-bit length ($n < l$).

A structural scheme of the Kupyna hash function is depicted in Figure 1.

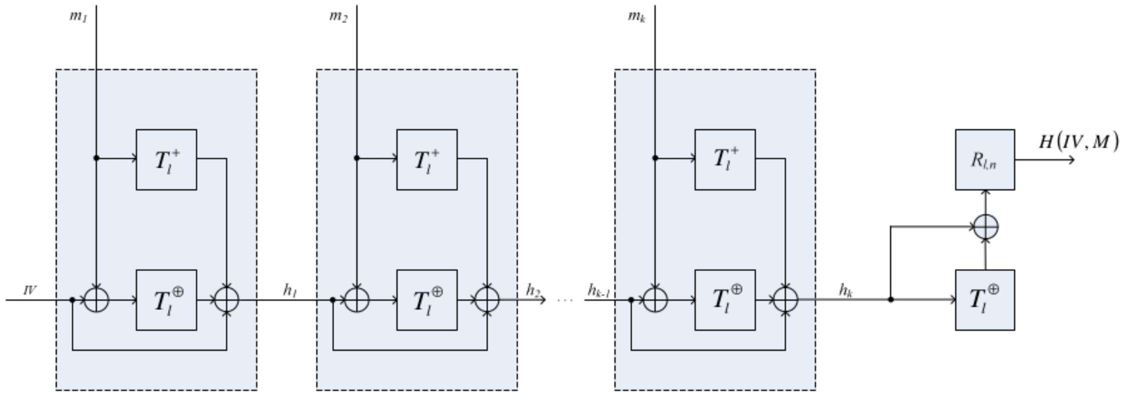

Figure 1: Structural scheme of the Kupyna hash function

Table 1: General parameters for `Kupyna`

Hash code length (n)	Internal state size (l)	Number of rounds (t)	Rows of the state matrix (c)
$8 \le n \le 256$	512	10	8
$256 < n \le 512$	1024	14	16

5 Message padding

The hash function takes a message (a bit string) of N bits length as an input. Each message is padded regardless of its length. Padding follows the message and contains the single '1' bit, then d zero bits, where $d = (-N - 97) \mod l$, and 96 bits containing the message length N (the least significant bits in the message length representation have smaller indexes, i.e. in the little-endian). As a result, the padded bit sequence has the length that is a multiple of the internal state l, $l \in \{512, 1024\}$.

The maximum length of the message is limited to $2^{96} - 1$ bits.

6 $T_l^\oplus$ and T_l^+ transformations

General structure

Transformations $T_l^\oplus$ and T_l^+ are bijective mappings $T_l^\oplus, T_l^+ : V_l \to V_l$, $l \in \{512, 1024\}$. Each mapping is implemented as iterative application of several functions that take input argument $x \in V_l$ as a matrix of $8 \times c$ bytes represented as elements of the $GF(2^8)$ finite field.

The relation between the internal state size (l), number of iterations (t), number of the state matrix columns (c) and the hash code length (n) is given in Table 1.

The internal state matrix is denoted as $G = (g_{i,j})$, $g_{i,j} \in GF(2^8)$ where $i = 0, 1, \ldots, 7$, $j = 0, 1, \ldots, c-1$. It is filled in with input bytes B_1, B_2, ..., $B_{l/8}$ in a column-by-column order. Example for $l = 512$ and $c = 8$ is given in Figure 2. Output bytes are read from the state matrix in the same order.

$T_l^\oplus$ and T_l^+ are defined as

Input byte sequence							
B_1	B_9	B_{17}	B_{25}	B_{33}	B_{41}	B_{49}	B_{57}
B_2	B_{10}	B_{18}	B_{26}	B_{34}	B_{42}	B_{50}	B_{58}
B_3	B_{11}	B_{19}	B_{27}	B_{35}	B_{43}	B_{51}	B_{59}
B_4	B_{12}	B_{20}	B_{28}	B_{36}	B_{44}	B_{52}	B_{60}
B_5	B_{13}	B_{21}	B_{29}	B_{37}	B_{45}	B_{53}	B_{61}
B_6	B_{14}	B_{22}	B_{30}	B_{38}	B_{46}	B_{54}	B_{62}
B_7	B_{15}	B_{23}	B_{31}	B_{39}	B_{47}	B_{55}	B_{63}
B_8	B_{16}	B_{24}	B_{32}	B_{40}	B_{48}	B_{56}	B_{64}

$$\Downarrow$$

Internal state of the hash function							
$g_{0,0}$	$g_{0,1}$	$g_{0,2}$	$g_{0,3}$	$g_{0,4}$	$g_{0,5}$	$g_{0,6}$	$g_{0,7}$
$g_{1,0}$	$g_{1,1}$	$g_{1,2}$	$g_{1,3}$	$g_{1,4}$	$g_{1,5}$	$g_{1,6}$	$g_{1,7}$
$g_{2,0}$	$g_{2,1}$	$g_{2,2}$	$g_{2,3}$	$g_{2,4}$	$g_{2,5}$	$g_{2,6}$	$g_{2,7}$
$g_{3,0}$	$g_{3,1}$	$g_{3,2}$	$g_{3,3}$	$g_{3,4}$	$g_{3,5}$	$g_{3,6}$	$g_{3,7}$
$g_{4,0}$	$g_{4,1}$	$g_{4,2}$	$g_{4,3}$	$g_{4,4}$	$g_{4,5}$	$g_{4,6}$	$g_{4,7}$
$g_{5,0}$	$g_{5,1}$	$g_{5,2}$	$g_{5,3}$	$g_{5,4}$	$g_{5,5}$	$g_{5,6}$	$g_{5,7}$
$g_{6,0}$	$g_{6,1}$	$g_{6,2}$	$g_{6,3}$	$g_{6,4}$	$g_{6,5}$	$g_{6,6}$	$g_{6,7}$
$g_{7,0}$	$g_{7,1}$	$g_{7,2}$	$g_{7,3}$	$g_{7,4}$	$g_{7,5}$	$g_{7,6}$	$g_{7,7}$

Figure 2: Filling the internal state matrix for $T_l^{\oplus}$ and T_l^{+}

$$T_l^{\oplus} = \prod_{\nu=0}^{t-1} \left(\psi \circ \tau^{(l)} \circ \pi' \circ \kappa_\nu^{(l)}\right),$$

$$T_l^{+} = \prod_{\nu=0}^{t-1} \left(\psi \circ \tau^{(l)} \circ \pi' \circ \eta_\nu^{(l)}\right),$$

where

$\kappa_\nu^{(l)}$ – the function of addition of the internal state with the iteration constant modulo 2,

$\eta_\nu^{(l)}$ – the function of addition of the internal state with the iteration constant modulo 2^{64},

π' – the layer of non-linear bijective mapping (S-box layer) that implements byte substitution,

τ_l – permutation of elements $g_{i,j} \in GF(2^8)$ in the internal state (right rotation of the rows),

ψ – the linear transformation (multiplication of the vector and matrix over the finite field).

Input argument $x \in V_l$ and output value $\chi(x) \in V_l$, $\chi \in \{\kappa_\nu^{(l)}, \eta_\nu^{(l)}, \pi', \tau_l, \psi\}$, are represented as matrices of $8 \times c$ size (see Table 1).

Addition with iteration constants

The $\kappa_{\nu}^{(l)}$ function adds (modulo 2) a vector $\omega_j^{(\nu)} = ((j \lll 4) \oplus \nu, 0, 0, 0, 0, 0, 0, 0)^T$, where $\omega_j^{(\nu)} \in V_{64}$, ν is an iteration (round) number, to each column of the internal state matrix $G = (g_{i,j})$.

The $\eta_{\nu}^{(l)}$ function adds (modulo 2^{64}) a vector $\varsigma_j^{(\nu)} = (0xF3, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0, 0xF0\,((c-1-j) \lll 4))^T$, where $\varsigma_j^{(\nu)} \in V_{64}$ and ν is an iteration (round) number, to each column of the internal state matrix $G = (g_{i,j})$, and $0xF3$ is the least significant byte of the $\varsigma_j^{(\nu)}$ vector, $g_{0,j}$ is the least significant byte of the G_j vector.

Layer of non-linear bijective mapping

The π' function substitutes each element $g_{i,j}$ of the internal state matrix $G = (g_{i,j})$ by $\pi_{i \bmod 4}(g_{i,j})$, where $\pi_s : V_8 \mapsto V_8$, $s \in \{0, 1, 2, 3\}$, are substitutions (S-boxes) given in Appendix A. For example, let $g_{i,j}$ be 0x23, then $\pi_0(0x23) = 0x4F$.

Permutation of elements in the internal state

The function τ_l performs cyclic right shift (rotation) for the rows in the state matrix $G = (g_{i,j})$. Rows with numbers $i = 0, 1, \ldots, 6$ of the matrix are rotated by i elements, and the row with the number 7 is rotated by 7 elements for $l = 512$ and by 11 elements for $l = 1024$.

Linear transformation

To perform the function ψ each element $g_{i,j} \in V_8$ of the internal state matrix G is represented as an element of the finite field $GF(2^8)$ formed by the irreducible polynomial $\Upsilon(x) = x^8 + x^4 + x^3 + x^2 + 1$, or $0x11D$ in hexadecimal notation.

Each element of the resulting state matrix $U = (u_{i,j})$ is calculated over $GF(2^8)$ according to the formula

$$u_{i,j} = (\upsilon \ggg i) \otimes G_j,$$

where $\upsilon = (0x01, 0x01, 0x05, 0x01, 0x08, 0x06, 0x07, 0x04)$ is the vector that forms the circulant matrix with the MDS property, and G_j is the j^{th} column of the state matrix G.

The vector υ consists of the hexadecimal constants (bytes) which are elements of the finite field $GF(2^8)$. The right circular shift is made with respect to elements of the set υ.

7 Conclusions

Kupyna is a new Ukrainian standard of cryptographic hash function (DSTU 7564:2014). It uses Davies-Meyer compression function based on Even-Mansour block cipher construction. A message padding and a truncation of the result hash code are obligatory operations in Kupyna. Internal permutations are built on the transformations of the Kalyna block cipher (Ukrainian standard DSTU 7624:2014). Kupyna supports hash code length from 8 bits to 512 bits in 8-bit steps (called `Kupyna`-n). The recommended modes are `Kupyna`-256, `Kupyna`-384 and `Kupyna`-512. The description of the hash function given in this paper is an adapted English version of the Kupyna specification from the original standard.

References

[1] Roman Oliynykov, Ivan Gorbenko, Oleksandr Kazymyrov, Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov, Ruslan Mordvinov, Dmytro Kaidalov. *DSTU 7624:2014. National Standard of Ukraine. Information technologies. Cryptographic Data Security. Symmetric block transformation algorithm.* Ministry of Economical Development and Trade of Ukraine, 2015 (in Ukrainian).

[2] Roman Oliynykov, Ivan Gorbenko, Oleksandr Kazymyrov, Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov, Ruslan Mordvinov, Dmytro Kaidalov. *A new encryption standard of Ukraine: The Kalyna block cipher.* Cryptology ePrint Archive. Report 2015/650, 2015. `http://eprint.iacr.org/2015/650.pdf`

[3] Metrology and Certification of the Commonwealth of Independence States. *GOST 34.311-95. Information technology. Cryptographic Data Security. Hash function.* Metrology and Certification of the Commonwealth of Independence States. Minsk, 1995. (In Russian)

[4] Government Committee of Russia for Standards. *GOST 34.11-94. Information technology. Cryptographic Data Security. Hash function.* Government Committee of Russia for Standards. Moscow, 1994. (In Russian)

[5] Florian Mendel, Norbert Pramstaller, Christian Rechberger, Marcin Kontak, Janusz Szmidt. *Cryptanalysis of the GOST hash function.* Advances in CryptologyâĂŞCRYPTO 2008. âĂŞ Springer Berlin Heidelberg, 2008.

[6] Roman Oliynykov, Ivan Gorbenko, Oleksandr Kazymyrov, Victor Ruzhentsev, Oleksandr Kuznetsov, Yurii Gorbenko, Artem Boiko, Oleksandr Dyrda, Viktor Dolgov, Andrii Pushkaryov. *DSTU 7564:2014. National Standard of Ukraine. Information technologies. Cryptographic Data Security. Hash function.* Ministry of Economical Development and Trade of Ukraine, 2015 (in Ukrainian).

[7] Preneel B. *The first 30 years of cryptographic hash functions and the NIST SHA-3 competition.* Topics in Cryptology-CT-RSA 2010. âĂŞ Springer Berlin Heidelberg, 2010.

[8] Perlner R. et al. *Third-round report of the SHA-3 cryptographic hash algorithm competition.* US Department of Commerce, National Institute of Standards and Technology, 2012.

[9] P.Gauravaram, L.Knudsen, K.Matusiewicz, F.Mendel, C.Rechberger, M.Schlaffer, S.Thomsen. *Groestl – a SHA-3 candidate.* Submission to NIST (2008).

A S-boxes for the Kupyna hash function (hexadecimal notation)

Substitution π_0

A8	43	5F	06	6B	75	6C	59	71	DF	87	95	17	F0	D8	09
6D	F3	1D	CB	C9	4D	2C	AF	79	E0	97	FD	6F	4B	45	39
3E	DD	A3	4F	B4	B6	9A	0E	1F	BF	15	E1	49	D2	93	C6
92	72	9E	61	D1	63	FA	EE	F4	19	D5	AD	58	A4	BB	A1
DC	F2	83	37	42	E4	7A	32	9C	CC	AB	4A	8F	6E	04	27
2E	E7	E2	5A	96	16	23	2B	C2	65	66	0F	BC	A9	47	41
34	48	FC	B7	6A	88	A5	53	86	F9	5B	DB	38	7B	C3	1E
22	33	24	28	36	C7	B2	3B	8E	77	BA	F5	14	9F	08	55
9B	4C	FE	60	5C	DA	18	46	CD	7D	21	B0	3F	1B	89	FF
EB	84	69	3A	9D	D7	D3	70	67	40	B5	DE	5D	30	91	B1
78	11	01	E5	00	68	98	A0	C5	02	A6	74	2D	0B	A2	76
B3	BE	CE	BD	AE	E9	8A	31	1C	EC	F1	99	94	AA	F6	26
2F	EF	E8	8C	35	03	D4	7F	FB	05	C1	5E	90	20	3D	82
F7	EA	0A	0D	7E	F8	50	1A	C4	07	57	B8	3C	62	E3	C8
AC	52	64	10	D0	D9	13	0C	12	29	51	B9	CF	D6	73	8D
81	54	C0	ED	4E	44	A7	2A	85	25	E6	CA	7C	8B	56	80

Substitution π_1

CE	BB	EB	92	EA	CB	13	C1	E9	3A	D6	B2	D2	90	17	F8
42	15	56	B4	65	1C	88	43	C5	5C	36	BA	F5	57	67	8D
31	F6	64	58	9E	F4	22	AA	75	0F	02	B1	DF	6D	73	4D
7C	26	2E	F7	08	5D	44	3E	9F	14	C8	AE	54	10	D8	BC
1A	6B	69	F3	BD	33	AB	FA	D1	9B	68	4E	16	95	91	EE
4C	63	8E	5B	CC	3C	19	A1	81	49	7B	D9	6F	37	60	CA
E7	2B	48	FD	96	45	FC	41	12	0D	79	E5	89	8C	E3	20
30	DC	B7	6C	4A	B5	3F	97	D4	62	2D	06	A4	A5	83	5F
2A	DA	C9	00	7E	A2	55	BF	11	D5	9C	CF	0E	0A	3D	51
7D	93	1B	FE	C4	47	09	86	0B	8F	9D	6A	07	B9	B0	98
18	32	71	4B	EF	3B	70	A0	E4	40	FF	C3	A9	E6	78	F9
8B	46	80	1E	38	E1	B8	A8	E0	0C	23	76	1D	25	24	05
F1	6E	94	28	9A	84	E8	A3	4F	77	D3	85	E2	52	F2	82
50	7A	2F	74	53	B3	61	AF	39	35	DE	CD	1F	99	AC	AD
72	2C	DD	D0	87	BE	5E	A6	EC	04	C6	03	34	FB	DB	59
B6	C2	01	F0	5A	ED	A7	66	21	7F	8A	27	C7	C0	29	D7

Substitution π_2

93	D9	9A	B5	98	22	45	FC	BA	6A	DF	02	9F	DC	51	59
4A	17	2B	C2	94	F4	BB	A3	62	E4	71	D4	CD	70	16	E1
49	3C	C0	D8	5C	9B	AD	85	53	A1	7A	C8	2D	E0	D1	72
A6	2C	C4	E3	76	78	B7	B4	09	3B	0E	41	4C	DE	B2	90
25	A5	D7	03	11	00	C3	2E	92	EF	4E	12	9D	7D	CB	35
10	D5	4F	9E	4D	A9	55	C6	D0	7B	18	97	D3	36	E6	48
56	81	8F	77	CC	9C	B9	E2	AC	B8	2F	15	A4	7C	DA	38
1E	0B	05	D6	14	6E	6C	7E	66	FD	B1	E5	60	AF	5E	33
87	C9	F0	5D	6D	3F	88	8D	C7	F7	1D	E9	EC	ED	80	29
27	CF	99	A8	50	0F	37	24	28	30	95	D2	3E	5B	40	83
B3	69	57	1F	07	1C	8A	BC	20	EB	CE	8E	AB	EE	31	A2
73	F9	CA	3A	1A	FB	0D	C1	FE	FA	F2	6F	BD	96	DD	43
52	B6	08	F3	AE	BE	19	89	32	26	B0	EA	4B	64	84	82
6B	F5	79	BF	01	5F	75	63	1B	23	3D	68	2A	65	E8	91
F6	FF	13	58	F1	47	0A	7F	C5	A7	E7	61	5A	06	46	44
42	04	A0	DB	39	86	54	AA	8C	34	21	8B	F8	0C	74	67

Substitution π_3

68	8D	CA	4D	73	4B	4E	2A	D4	52	26	B3	54	1E	19	1F
22	03	46	3D	2D	4A	53	83	13	8A	B7	D5	25	79	F5	BD
58	2F	0D	02	ED	51	9E	11	F2	3E	55	5E	D1	16	3C	66
70	5D	F3	45	40	CC	E8	94	56	08	CE	1A	3A	D2	E1	DF
B5	38	6E	0E	E5	F4	F9	86	E9	4F	D6	85	23	CF	32	99
31	14	AE	EE	C8	48	D3	30	A1	92	41	B1	18	C4	2C	71
72	44	15	FD	37	BE	5F	AA	9B	88	D8	AB	89	9C	FA	60
EA	BC	62	0C	24	A6	A8	EC	67	20	DB	7C	28	DD	AC	5B
34	7E	10	F1	7B	8F	63	A0	05	9A	43	77	21	BF	27	09
C3	9F	B6	D7	29	C2	EB	C0	A4	8B	8C	1D	FB	FF	C1	B2
97	2E	F8	65	F6	75	07	04	49	33	E4	D9	B9	D0	42	C7
6C	90	00	8E	6F	50	01	C5	DA	47	3F	CD	69	A2	E2	7A
A7	C6	93	0F	0A	06	E6	2B	96	A3	1C	AF	6A	12	84	39
E7	B0	82	F7	FE	9D	87	5C	81	35	DE	B4	A5	FC	80	EF
CB	BB	6B	76	BA	5A	7D	78	0B	95	E3	AD	74	98	3B	36
64	6D	DC	F0	59	A9	4C	17	7F	91	B8	C9	57	1B	E0	61