О МОДИФИКАЦИИ ОТЕЧЕСТВЕННОГО НИЗКОРЕСУРСНОГО КРИПТОГРАФИЧЕСКОГО АЛГОРИТМА 2-ГОСТ И ВОПРОСАХ ЕГО РЕАЛИЗАЦИИ НА ПЛИС

А. Дмух, Д. Трифонов, А. Чухно

Москва, 2018 г.

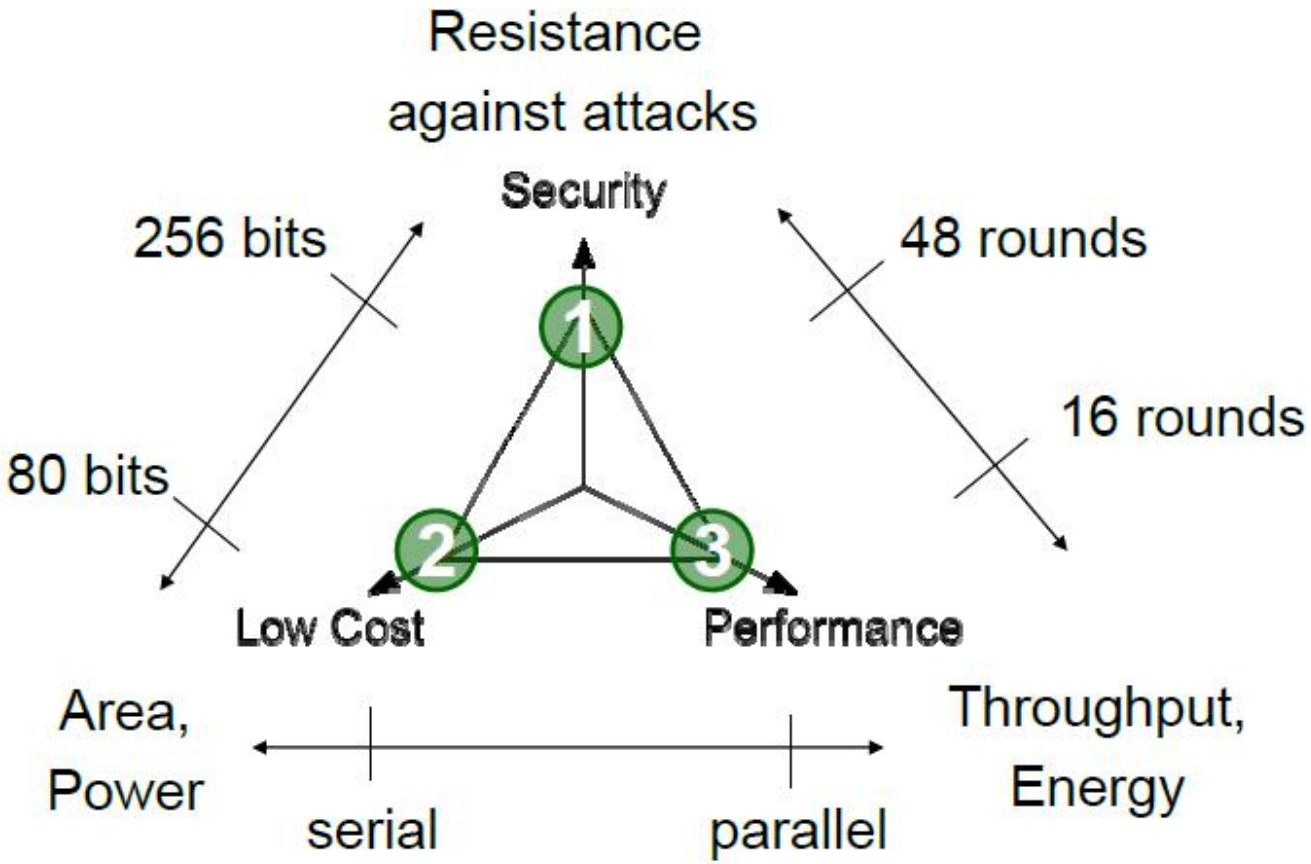
Данная работа выполнена коллективом авторов в инициативном порядке и может не совпадать с чьей-либо официальной позицией.

Axel York Poschmann, PhD

"As light as a feather, and as hard as dragon-scales"

Bilbo Baggins in "The Lord of the Rings: The Fellowship of the Ring".

Axel York Poschmann, PhD



Исследование компании Gartner, Inc.

На 2017 год

насчитывается около 8,4 млрд устройств интернета вещей, подключённых к сети
рост за год: 31%

К 2020 году ожидается 20,4 млрд устройств

Основные регионы

- Китай
- Северная Америка
- Западная Европа

Исследование компании Gartner, Inc.

Основные типы устройств интернета вещей

Для потребителей

- "умные"телевизоры
- ТВ-приставки
- автомобили

Для бизнеса

- "умные"электронные весы
- камеры видеонаблюдения

Низкоресурсная, а не легковесная!

(по словарю Т.Ф. Ефремова)

Низкоресурсная (синонимы)

хорош криптографически, прост для понимания и реализации

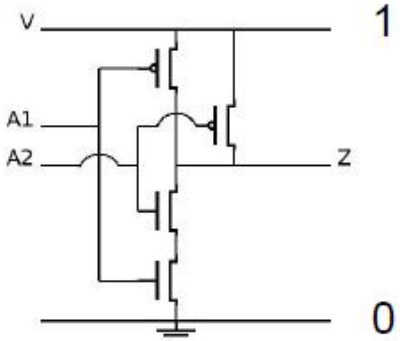
Легковесная, лёгкая (синонимы)

поверхностный, неглубокий, не крутой, слабый

Что такое GE?

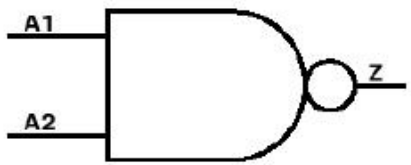
(в отечественной схемотехнической литературе - вентильная модель)

NAND



A1	A2	Z
0	0	1
0	1	1
1	0	1
1	1	0

Standard Cells
UMCL18G212T3



HDNAN2D1
9.677 µm²

1 GE

Athlon XP

13.24 Mio GE

Note for Mathematicians:
NAND + constants = base

Низкоресурсная, а это сколько?

Axel York Poschmann

Lightweight устройство

- компонент защиты информации - до 2 000 GE
- общая реализация - до 10 000 GE

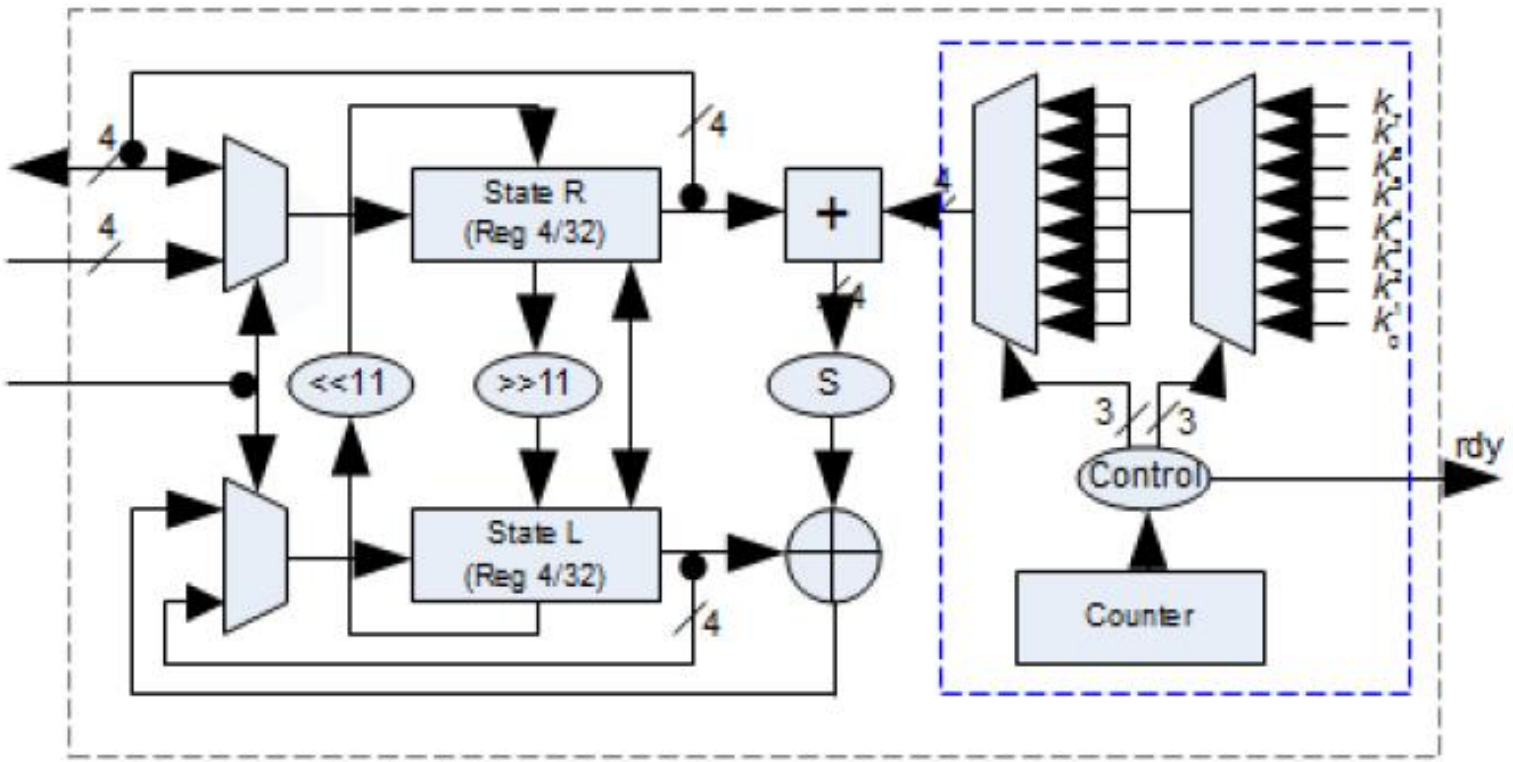
Charalampos Manifavas, George Hatzivasilis, Konstantinos Fysarakis, Konstantinos Rantos

- ultra-lightweight — 1000 logic gates (4 KB ROM, 256 bytes RAM)
- low-cost — 2000 logic gates (4 KB ROM, 8 KB RAM)
- lightweight — 3000 logic gates (32 KB ROM, 8KB RAM)

Способы оценки GE для некоторой криптографической функции

- оценка эффективности реализации на основе аналогичных исследований других авторов
- представление реализуемой функции в базисе NAND и непосредственное вычисление GE
- реализация алгоритмов (примитивов) на конкретной элементной базе с использованием соответствующих средств автоматизированного проектирования

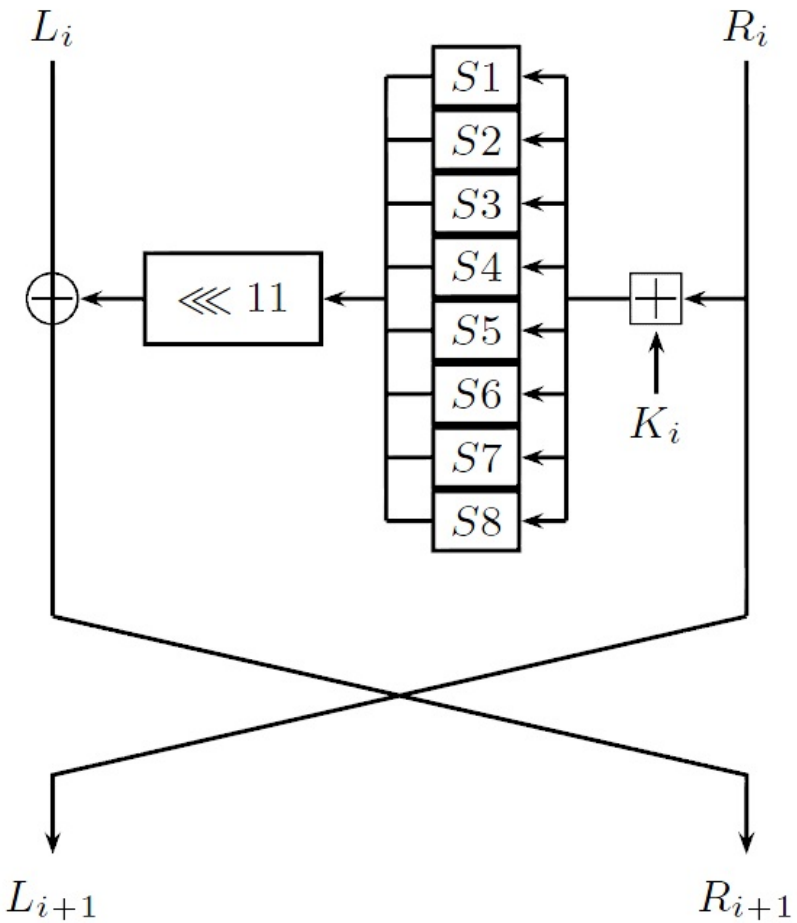
Принципиальная схема реализации алгоритма ГОСТ 28147-89



Правила игры на ПЛИС

1. ключ, как правило - внешний сигнал (нигде не хранится)
2. триггеры буферизации не учитываются
3. ресурсы коммутационных матриц не учитывается
4. тактовая частота работы схемы не учитывается
5. допускаются сколь угодно большие задержки сигнала

Немного о ГОСТ 28447-89 и 2-ГОСТ



Немного о ГОСТ 28447-89 и 2-ГОСТ. Ключевая развёртка

Table : The order of subkeys in GOST and GOST2

Round	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Subkey (GOST)	K^0	K^1	K^2	K^3	K^4	K^5	K^6	K^7	K^0	K^1	K^2	K^3	K^4	K^5	K^6	K^7
Subkey (GOST2)	K^0	K^1	K^2	K^3	K^4	K^5	K^6	K^7	K^3	K^4	K^5	K^6	K^7	K^0	K^1	K^2
Round	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
Subkey (GOST)	K^0	K^1	K^2	K^3	K^4	K^5	K^6	K^7	K^7	K^6	K^5	K^4	K^3	K^2	K^1	K^0
Subkey (GOST2)	K^5	K^6	K^7	K^0	K^1	K^2	K^3	K^4	K^6	K^5	K^4	K^3	K^2	K^1	K^0	K^7

Основные слабости

свойство отражения(reflection property), неподвижная точка (fixed point property)

ГОСТ. Результаты

- Метод Исобе . Трудоёмкость: 2^{225} операций зашифрования, материал: 2^{32} , надёжность: 0.63, объём памяти: 2^{71} бит.
- Первый метод Динура-Дункельмана-Шамира. Трудоёмкость: $1,5 \cdot 2^{192}$ операций зашифрования, материал: 2^{64}, надёжность: 0.63, объём памяти: 2^{71} бит.
- Второй метод Динура-Дункельмана-Шамира. Трудоёмкость: $1,5 \cdot 2^{224}$ операций зашифрования, материал: 2^{32}, надёжность : 0.63, объём памяти: 2^{42} бит.

2-ГОСТ А.А. Дмух, Г.Б. Маршалко, Д.М. Дыгин

предложен в 2013

Основные слабости

свойство отражения(reflection property), неподвижная точка (fixed point property)

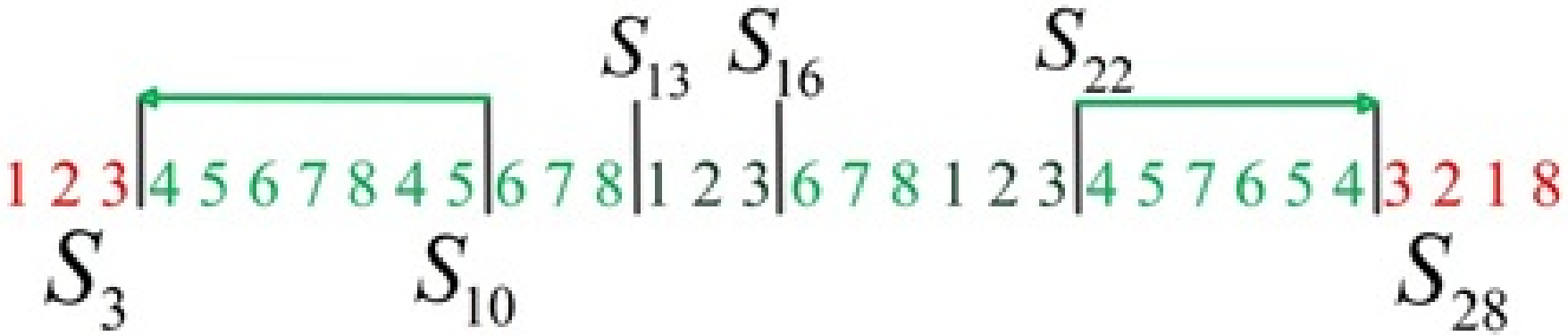
2-ГОСТ. Результаты исследований

Метод Ашура, Бар-Она, Дункельмана. Трудоёмкость: 2^{237} операций зашифрования, необходимый объём памяти: 2^{199} бит.

2-ГОСТ. Наши результаты

Трудоёмкость: $1.5 \cdot 2^{233}$ операций зашифрования, необходимый объём памяти: 2^{198} бит.

Метод Ашура, Бар-Она, Дункельмана. Основная идея



1. Для каждой пары открытый-шифрованный текст, опробуем ключи K_1, K_2, K_3, K_8.
   - $S_{28} = R_{K_3}^{-1}(R_{K_2}^{-1}(R_{K_1}^{-1}(R_{K_8}^{-1}(C_i))))$.
   - $S_3 = R_{K_3}(R_{K_2}(R_{K_1}(P_i)))$.
   - В память Π_1 по адресу $S_3 \| S_{28}$ записываем K_1, K_2, K_3, K_8.
2. Опробуем $S_{10} = S_{16} = S_{22}$, K_4, K_5, K_6, K_7, K_8.
   - $S_{13} = R_{K_6}(R_{K_7}(R_{K_8}(S_{10})))$.
   - Для выбранных $S_{13} \| S_{16}$ опробуем $K_1[0-11], K_3[0-11]$, бит переноса в 12 бит, при вычислении S_{14}. Вычисляем 2^{13} вариантов $K_1[0-11], K_3[0-11], K_2[12-19]$.
   - $S_3 = R_{K_4}^{-1}(R_{K_5}^{-1}(R_{K_6}^{-1}(R_{K_7}^{-1}(R_{K_8}^{-1}(R_{K_4}^{-1}(R_{K_5}^{-1}(S_{10})))))))$.
   - $S_{28} = R_{K_4}(R_{K_5}(R_{K_6}(R_{K_7}(R_{K_5}(R_{K_4}(S_{22}))))))$.
   - По адресу $S_3 \| S_{28}$ из памяти извлекаем $K_1, K_2, K_3, K_8.$, отбраковываем по значениям $K_1[0-11], K_3[0-11], K_2[12-19]$ и K_8.
   - Опробуем ключ $K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8$.

Параметры метода

Память:

$$128 \cdot 2^{128+64} = 2^{199} \text{ бит}$$

Трудоёмкость (шаг 1) $= Q_1$

$$2^{64} \cdot 2^{32 \cdot 4} \cdot \left(\frac{4}{32} + \frac{3}{32} + 1 \right) = 1.2 \cdot 2^{32 \cdot 6}.$$

Трудоёмкость (шаг 2) $= Q_2$

$$2^{32 \cdot 7} \cdot \left(\frac{3}{32} + 2^{12} \cdot \frac{1}{32} \cdot \frac{3}{8} + 2^{12} \cdot \frac{1}{32} \cdot \frac{3}{8} \cdot 2 \cdot 2^{12} + \frac{7}{32} + \frac{6}{32} + Q^* \right)$$

Attention!

В каждой ячейке памяти будет находиться 2^{64} вариантов ключа.

$$\Longrightarrow \quad Q^* \geq 2^{64}$$

Трудоёмкость (полная)

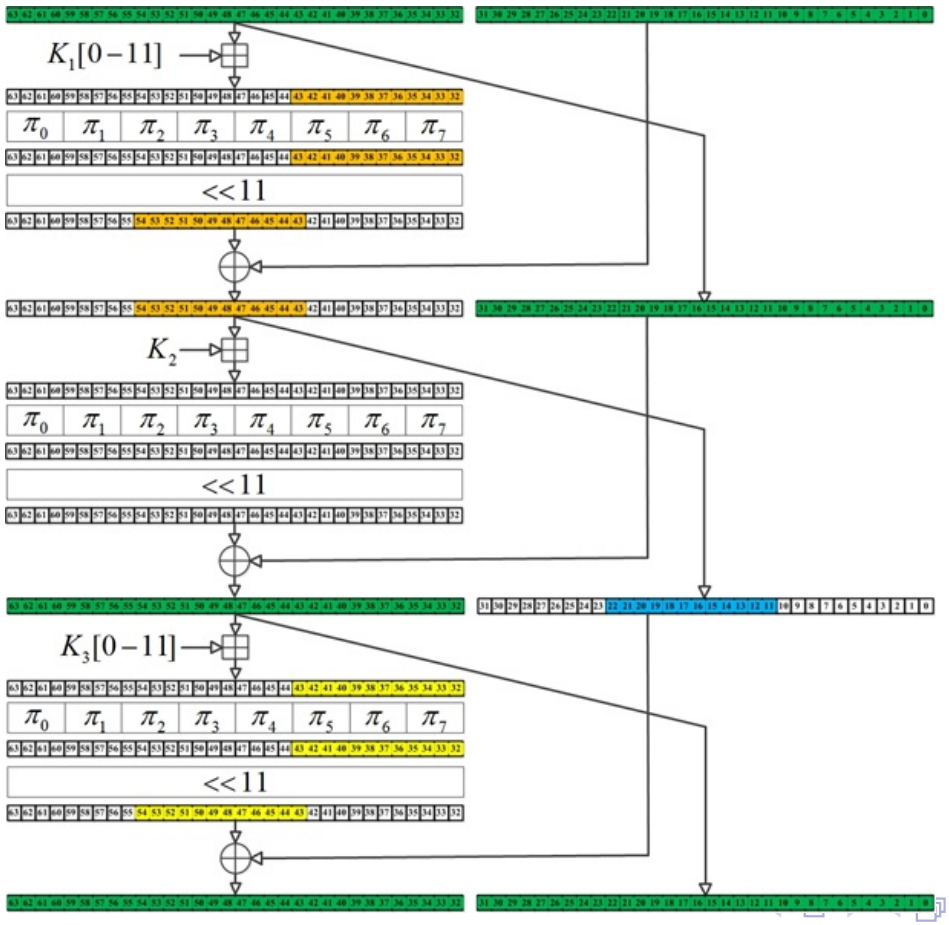
$$Q_1 + Q_2 > 2^{256}$$

Модифицированный метод анализа алгоритма 2-ГОСТ

1. Для каждой пары о.т.-ш.т. опробуем ключи K_1, K_2, K_3, K_8.
   - $S_{28} = R_{K_3}^{-1}(R_{K_2}^{-1}(R_{K_1}^{-1}(R_{K_8}^{-1}(C_i))))$.
   - Вычисляем$S_3 = R_{K_3}(R_{K_2}(R_{K_1}(P_i)))$.
   - В память Π_1 по адресу $S_3 \| S_{28} \| K_1[0-11] \| K_2[12-19] \| K_3[0-11] \| K_8$ записываем $K_1[12-31] \| K_2[0-11, 20-31] \| K_3[12-31]$.
2. Опробуем $S_{10} = S_{16} = S_{22}$, K_4, K_5, K_6, K_7, K_8.
   - $S_{13} = R_{K_6}(R_{K_7}(R_{K_8}(S_{10})))$.
   - $S_3 = R_{K_4}^{-1}(R_{K_5}^{-1}(R_{K_6}^{-1}(R_{K_7}^{-1}(R_{K_8}^{-1}(R_{K_4}^{-1}(R_{K_5}^{-1}(S_{10}))))))))$.
   - $S_{28} = R_{K_4}(R_{K_5}(R_{K_6}(R_{K_7}(R_{K_5}(R_{K_4}(S_{22}))))))$.
   - Для выбранных S_{13}, S_{16}:
     - опробуем $K_1[0-11]$. вычисляем $S_{14}[11-22]$ и записываем $K_1[0-11]$ в память Π_2 по адресу $S_{14}[11-22]$.
     - опробуем $K_3[0-11]$. Вычисляем $S_{15}[43-54]$.
     - Используя память Π_2 получаем $K_1[0-11] \| K_3[0-11]$.
     - Опробуем бит переноса из 12 бита S_{14}, вычисляем 2 варианта $K_1[0-11] \| K_2[12-19] \| K_3[0-11]$.
     - Проверяем $K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8$.

Модифицированный метод анализа алгоритма 2-ГОСТ

Зеленые — известные биты, желтые — опробуемые биты, синие — проверочные биты



Параметры метода

Память:

$$64 \cdot 2^{128+64} + 12 \cdot 2^{12} \approx 2^{198} \text{ bits}$$

Трудоёмкость (шаг 1) = Q_1

$$Q_1 = 2^{64} \cdot 2^{32 \cdot 4} \left(\frac{4}{32} + \frac{3}{32} + 1 \right) = 1.2 \cdot 2^{32 \cdot 6}.$$

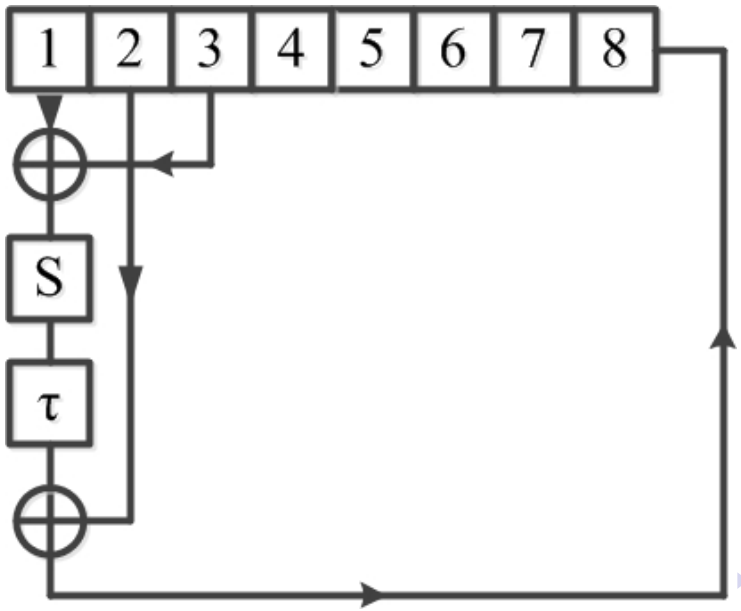
Трудоёмкость (шаг 2) = Q_2

$$Q_2 = 2^{32 \cdot 7} \cdot \left(\frac{16}{32} + 2^{12} \cdot \left(\frac{1}{32} + \frac{1}{32} \right) + 2^{12} \cdot \left(\frac{1}{32} + \frac{1}{32} + 2 \cdot \frac{1}{32} \right) \right) =$$

$$= 1.5 \cdot 2^{233}.$$

Новый способ выбора итерационных ключей алгоритма 2-ГОСТ

для выработки ключевой развёртки используется регистр H, состоящий из восьми ячеек $H_i \in V_{32}$, $i = 1, \ldots, 8$. Выход регистра в i-й такт будет i-ым итерационным ключом. Графически регистр H можно представить в виде



Функционирование регистра H в i-й такт, $i = 1, \ldots, 32$, описывается следующими уравнениями:

$$H_8(i) = \tau(S[H_2(i-1) \oplus H_3(i-1)]) \oplus H_1(i-1),$$

$$H_j(i) = H_{j+1}(i-1),\ j = 1, \ldots, 7,$$

где

- $S(a) = S(a_7, \ldots, a_0) = (\pi(a_7), \ldots, \pi(a_0))$, $a \in V_{32},\ a_i \in V_4,\ \pi : V_4 \to V_4,\ i = 0, 1, \ldots, 7$;
- $\tau \in S_{32}$ подстановка на множестве натуральных чисел $\{1, \ldots, 32\}$ (коммутация).

Подстановка τ задаётся в виде нижней строки подстановки следующим образом

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 \\ 4 & 13 & 22 & 31 & 8 & 17 & 26 & 3 & 12 & 21 & 30 & 7 & 16 & 25 & 2 & 11 & 20 & 29 & 6 & 15 & 24 & 1 & 10 & 19 & 28 & 5 & 14 & 23 & 0 & 9 & 18 & 27 \end{pmatrix}$$

Нелинейное биективное преобразование π выбиралось так, чтобы для реализации указанной подстановки можно было использовать минимально возможное число битовых операций

$$\pi = (0, 8, 6, 13, 5, 15, 7, 12, 4, 14, 2, 3, 9, 1, 11, 10).$$

ГОСТ и 2-ГОСТ. Реализация и криптография.

Ал-м \ Метод	*Isobe*	*DinurDS*	*DunkelmanAB*	Данная работа
ГОСТ28147 – 89	2^{225}	$1,5 \cdot 2^{192}$	2^{256}	2^{256}
2-ГОСТ	2^{256}	2^{256}	2^{237}	$1.6 \cdot 2^{233}$
2-ГОСТ модиф.	2^{256}	2^{256}	2^{256}	2^{256}

Ал-м	Ключ — внеш.сиг.	Фикс. ключ	Произв. ключ
ГОСТ 28147-89	650 GE	1556 GE	2137 GE
2-ГОСТ	750 GE	1570 GE	2158 GE
2-ГОСТ модиф.	750 GE	—	2544 GE

Спасибо за внимание!