Использование покомпонентных функций в алгоритме криптографического преобразования ГОСТ Р 34.12-2015

И.И. Василишин[1], С.Ю. Корабельщикова[1], Д.М.-М. Султанов[2], М.С. Пугин[2]

[1]Северный (Арктический) федеральный университет имени М.В. Ломоносова, наб. Северной Двины 17, Архангельск, Россия, 163007
[2]Экспертно-криминалистический центр УМВД РФ по Архангельской области, ул. Воскресенская 3, корп. 2, Архангельск, Россия, 163003

Аннотация. В работе изложен общий подход выбора функций, сохраняющих поле первого аргумента, при симметричном шифровании открытого текста. Даны количественные оценки и общая характеристика вектора значений таких функций. Представлены десять покомпонентных функций алгебры двоичной логики трёх аргументов, замещающие одну функцию поразрядного сложения по модулю два в алгоритме криптографического преобразования ГОСТ Р 34.12-2015. Использование покомпонентных функций расширяет разнообразие промежуточных вариантов раундовых преобразований блочного шифрования, что усложняет алгоритм дешифрования (взлома) шифротекста.

1. Введение

Стандарт ГОСТ Р 34.12-2015 [1] представляет собой симметричный шифр, в котором выполняется преобразование открытого текста блоками фиксированной длины 128 либо 64 разряда и ключом длиной 256 разрядов, при этом алгоритмы зашифрования/расшифрования являются обратными процедурами с использованием многораундовых операций подстановок и преобразований последовательность которых разворачивает поразрядная операция «сложение по модулю два» (СМД) для исходного текста и первого итерационного ключа. Следовательно, предложенное авторами, многовариантное замещение операции СМД приводит к получению большего разнообразия промежуточных вариантов поразрядного сложения, что изменяет результаты итоговых операций, а в целом усложняет алгоритм дешифрования (взлома) шифротекста.

2. Общий анализ существующего алгоритма

Криптографическое преобразование информации, использующееся в ГОСТ Р 34.12-2015, основано на принципах блочного шифрования данных [2, 3] и содержит комбинации операторов, обеспечивающих выполнение свойств симметричного шифрования в стандарте [1]:

- *поразрядного сложения* – формируется операцией СМД над текущим преобразованием (a) и раундовым ключом (k), что соответствует поразрядному преобразованию $X[k]$: $V_{128} \rightarrow V_{128}$, где полученный результат определяет равенство блоков до и после операции СМД и задаётся формулой

$$X[k](a) = k \oplus a, \quad (1)$$

где: $k, a \in V_{128}$;

- *перемешивания информации* – формируется при нелинейном биективном преобразовании над S блоком замены, выполняющем как операцию побайтовой (a_{15}, …, a_0) подстановки 128-разрядного значения $S: V_{128} \rightarrow V_{128}$, где значение a_i определяет индекс массива замены π, а результат $V_{128} \rightarrow V_{128}$ – определяет как равенство блоков до и после замены, так и формирование аналитического усложнения зависимостей между ключом и зашифрованным текстом, и обеспечивается преобразованием

$$S(a) = S(a_{15}||\ldots||a_0) = \pi(a_{15})||\ldots||\pi(a_0), \quad (2)$$

где: $a = a_{15}||\ldots||a_0 \in V_{128}$, $a_i \in V_8$, $i = 0, 1, \ldots, 15$;

- *рассеивания информации* – достигается выполнением как девяти раундового последовательного вложения $F(a)$ для шестнадцатикратного побайтового преобразования в каждом раунде $L(a)$ над 128-разрядным значением блока замены $S(\pi(a))$, так и распространением влияния каждого знака открытого текста на все знаки шифротекста, обеспечивается преобразованиями

$$F[k](a_1, a_0) = LSX[k](a_1) \oplus a_0, a_1, \quad (3)$$

где: $L(a) = R^{16}(a)$; $R(a) = R(a_{15}||\ldots||a_0) = \ell(a_{15}, \ldots, a_0)||\, a_{15}||\ldots||\, a_1$; $k, a_i \in V_{128}$.

Обобщая преобразования, приведённые в уравнениях (1), (2) и (3), сформируем полный алгоритм зашифрования $E_{Ki}(a)$, выполняющий преобразования 128-разрядного исходного блока информации, где используется подстановка прямой нумерации итерационных ключей

$$E_{K1, \ldots, K10}(a) = (k_1 \oplus a)F(a)(k_2 \oplus a)F(a)\ldots(k_9 \oplus a)F(a)(k_{10} \oplus a). \quad (4)$$

Требуется отметить, что полный алгоритм расшифрования информации $D_{Ki}(a)$ использует обратные преобразования $S^{-1}(a)$, $R^{-1}(a)$ и $L^{-1}(a)$, при этом, нумерация подстановки итерационных ключей ведётся в обратной очерёдности

$$D_{K10, \ldots, K1}(a) = (k_{10} \oplus a)F^{-1}(a)(k_9 \oplus a)F^{-1}(a)\ldots(k_2 \oplus a)F^{-1}(a)(k_1 \oplus a). \quad (5)$$

В соответствии с преобразованиями, приведёнными в уравнениях (1) – (5), полная алгоритмическая последовательность зашифрования/расшифрования информации содержит как повторяющуюся операцию СМД над итерационным ключом (k_i) и текущим преобразованием (a), так и пораундовые преобразования смешивания и рассеяния информации для прямых $S(a)$, $R(a)$, $L(a)$ и обратных $S^{-1}(a)$, $R^{-1}(a)$, $L^{-1}(a)$ подстановок. Следовательно, весь периодически повторяющийся процесс зашифрования/расшифрования информации целесообразно разделить, для текущего изложения, на две группы операций: СМД и подстановок.

3. Введение покомпонентных функций

Используемая в алгоритмах криптографического преобразования информации $D(a)$ поразрядная операция СМД принадлежит к функциям булевой алгебры [3], порождается сочетаниями логических значений двух аргументов и является одной из операций, обладающей свойством «*восстановления исходного значения одного из аргументов*» при последовательном применении операции в процессе зашифрования, а затем в процессе расшифрования

$$D(a) = (k \oplus a) \oplus k, \quad (6)$$

где: k – итерационный ключ; a – информация для зашифрования.

Учитывая принадлежность операции СМД к функциям булевой алгебры двух аргументов укажем, что аналогичным свойством обладают и другие функции порождаемые сочетаниями логических значений трёх [4], четырёх и более аргументов, но при этом порождаемые функции не выполняют «классическую» операцию СМД. Следовательно, представим новую функцию, обеспечивающую восстановление исходного её значения, действием эквивалентным «классической» операции СМД и назовём новым понятием – *покомпонентная функция* $M(a)$.

Представим (см. рисунок) местоположение покомпонентных функций $M_j(a)$, в процессе зашифрования 128-разрядного блока информации в виде алгоритма, построенного с учётом преобразования (4), для стандартного и предлагаемого авторами преобразований.

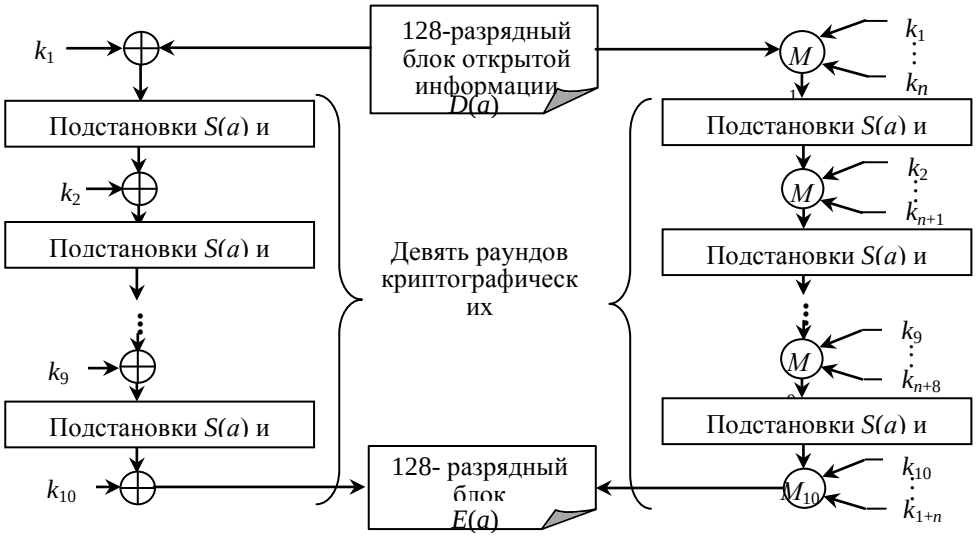

Рисунок 1. Алгоритмы зашифрования: существующий – *слева* и авторский – *справа*.

Как показано на рисунке отличиями в авторском алгоритме зашифрования являются используемые покомпонентные функции $M_j(a)$, замещающие единственную операцию СМД, используемую в преобразовании (4), а также многократное применение итерационных ключей, при этом вырабатывание и количество итерационных ключей остаётся аналогичным стандартному и равно десяти. Требуется отметить, что процесс расшифрования остаётся аналогичным процессу зашифрования, за исключением смены порядковых номеров итерационных ключей на противоположную, аналогично преобразованию (5).

4. Формирование покомпонентных функций

Суть формирования покомпонентных функций заключается в теоретическом определении характеристик функций, обладающих свойством восстановления исходного значения для одного из аргументов при выполнении только логического действия функции над операциями прямого и обратного преобразования, аналогичного использованию операции СМД в уравнении (1). В дальнейшем символ «◈» – определяет логические действия покомпонентной функции.

В теории абстрактной алгебры представлены доказательства существования булевых алгебр для любого количества аргументов, введена индексация булевых функций, а также установлена принадлежность множества индексированных функций булевых алгебр к системам нормальных форм [5]. Используя терминологию [4], представим принципы формирования покомпонентных функций в виде утверждений, справедливых для любого количества аргументов. Для определённости используем покомпонентные функции трёх аргументов.

Утверждение 1. О распределении смысловой нагрузки аргументов потенциально пригодных покомпонентных функций.

Прямое преобразование $M(a)[k]$ – поле аргумента A содержит данные для преобразования, аргументы B и C содержат значения итерационных ключей в прямой нумерации

$$M_j(a)[k] = M_j(A, B, C) = a \diamondsuit k_i \diamondsuit k_{i+1} \equiv X[k](a), \quad (7)$$

где: j – порядковый номер покомпонентной функции; i = 1, 2, …, 10 – порядковый номер итерационного ключа, если $i > 10 \Rightarrow i := i \bmod 10$; k – логические значения разрядов итерационного ключа.

Обратное преобразование $M^{-1}_j(a)[k]$ – поле аргумента A содержит данные для восстановления, аргументы B и C содержат значения итерационных ключей в обратной нумерации

$$M^{-1}_j(a)[k] = M^{-1}_j(A, B, C) = a \diamondsuit k_{i+1} \diamondsuit k_i \equiv X^{-1}[k](a), \quad (8)$$

где: i = 10, 9, …, 1 – порядковый номер итерационного ключа, если $i < 1 \Rightarrow i := i \bmod 10$.

Применим прямое преобразование, указанное в уравнении (7), в процессе зашифрования и обратное уравнение (8) – в процессе расшифрования

$$D(a) = M_j^{-1} M_j(a)[k], \quad (9)$$

следовательно, такое двойное преобразование привело к восстановлению исходных данных, аналогично преобразованию, указанному в уравнении (6).

Утверждение 2. О распределении поля логических нулей и единиц таблицы истинности потенциально пригодных покомпонентных функций.

Функции прямого и обратного преобразования, потенциально пригодные для использования, содержат равное количество нулей и единиц, что подтверждается условием

$$M^{-1}_j M_j(a)[k] = M(M(A, B, C), B, C) = A, \quad (10)$$

откуда получаем:

$$\begin{cases} M(M(0,B,C),B,C) = 0 \\ M(M(1,B,C),B,C) = 1 \end{cases}.$$

Выполняя подстановку для всех сочетаний аргументов, выбираем из потенциально пригодных функций функции, восстанавливающие поле аргумента A. Не будем учитывать здесь функцию, тождественную первому аргументу, и её отрицание, как не зависящие существенно от других аргументов и, следовательно, непригодные для шифрования. В соответствии с приведёнными утверждениями, а также используя теорию абстрактной алгебры, выполним количественные вычисления для функций двух, трёх, и четырёх аргументов по условию уравнения (10), результаты которых приведём в таблице 1.

Таблица 1. Количественные характеристики покомпонентных функций.

№ п/п	Характеристики	Количество аргументов		
		2	3	4
1	Полное количество булевых функций	16	256	65536
2	Потенциально пригодные функции	6	70	17920
3	Все функции, восстанавливающие аргумент *A*	2	14	3584
4	Используемые функции	1	10	2560

Выполняя окончательную выборку покомпонентных функций, количество которых указано в таблице 1, сформируем их совершенную, а затем и минимальную дизъюнктивную формы (МДНФ). Результаты формирования функции МДНФ приведены в таблице 2.

Таблица 2. Покомпонентные функции, используемые для преобразования данных.

№ п/п (*j*)	Покомпонентные функции		Вероятности переходов, %	
	Поле логических нулей и единиц	Функции МДНФ	0→0 1→1	0→1 1→0
1.	00011110	¬ABC ∨A¬B∨A¬C	75	25
2.	00101101	¬AB¬C ∨A¬B∨AC	75	25
3.	01001011	AB∨A¬C∨¬A¬BC	75	25
4.	01101001	AB¬C∨A¬BC∨A¬B¬C∨ABC	50	50
5.	01111000	¬AB∨¬AC∨A¬B¬C	25	75
6.	10000111	AB∨AC∨¬A¬B¬C	75	25
7.	10010110	A¬BC∨AB¬C∨¬A¬B¬C∨¬ABC	50	50
8.	10110100	¬AB∨¬A¬C∨A¬BC	25	75
9.	11010010	¬A¬B∨¬AC∨AB¬C	25	75
10.	11100001	¬A¬B∨¬A¬C∨ABC	25	75

Следует уточнить, что симметричное расположение одинаковых вероятностей переходов в таблице 2, при естественном увеличении весовых коэффициентов поля нулей и единиц, характеризует завершённость и корректность представлений покомпонентных функций. Кроме того, для восстановления аргументов *B* и *C* существуют другие функции, МДНФ которых отсутствуют в таблице 2.

Утверждение 3. О использовании покомпонентных функций в криптографическом преобразовании. Существующий алгоритм криптографического преобразования ГОСТ Р 34.12-2015 использует только одну функцию двух аргументов, а именно функцию СМД (см. левый алгоритм на рисунке). В соответствии с таблицей 2 существует десять разных функций трёх аргументов удовлетворяющих условию восстановления исходного значения поля аргумента A, следовательно, в криптографическом преобразовании информации можно использовать как одну из приведённых в таблице функций, так и любую комбинацию этих функций. При использовании требуемого количества функций трёх аргументов алгоритм зашифрования примет вид, показанный в правой части рисунка.

5. Анализ преобразований покомпонентными функциями

Один из способов практической демонстрации возможностей покомпонентных функций – выполнение преобразований функциями из таблицы 2 и сравнение результатов с результатами приведёнными в ГОСТе Р 34.12-2015 [1].

Для использования покомпонентных функций расширим общий вид свойства поразрядного сложения, как показано в преобразовании (1), для применения функций двоичной логики трёх аргументов [4]

$$X\,[k]\,(a) = M_j\{(a)\,[k_1]\,[k_2]\}, \qquad (11)$$

где: M_j – порядковый номер функций в таблице 2; k_1 и k_2 – итерационные ключи.

Расширение свойства поразрядного сложения в преобразовании (1) до вида в преобразовании (11) вносит другие комбинации операторов, что придаёт всему криптографическому преобразованию новое свойство, а именно – *покомпонентного преобразования*.

Представим результаты покомпонентного преобразования для прямого преобразования $M_j(a)[k]$, соответствующего применению уравнения (7), с учётом преобразования (11), для функций j = 1 и 2 из таблицы 2

$$M_1(a)[k] = a \diamondsuit k_1 \diamondsuit k_2 = 99ba99dc51325510ffefddefbbabddef; \qquad (11, a)$$

$$M_2(a)[k] = a \diamondsuit k_1 \diamondsuit k_2 = 6766232267666700fecc988832221000, \qquad (11, б)$$

где: a = 1122334455667700ffeeddccbbaa9988 [1, стр. 14];

k_1 = 8899aabbccddeeff0011223344556677 [1, стр. 13];

k_2 = fedcba98765432100123456789abcdef [1, стр. 13].

Требуется отметить, что результат поразрядного сложения (1), приведённый в [1, стр. 14], имеет вид

$$X[k](a) = k_1 \oplus a = 99bb99ff99bb99ffffffffffffffffff\,, \qquad (12)$$

закономерно отличающийся от преобразующего уравнения (11), результаты преобразования которого приведены в (11 , *а* и *б*) предложенного авторами способа покомпонентного преобразования.

6. Заключение

Функции алгебры двоичной логики трёх аргументов, расширяющие режимы криптографического преобразования ГОСТ Р 34.12-2015, являются началом ряда аналогичных функций для четырёх, пяти и более аргументов, а их использование, без сомнений, увеличит возможности поразрядного преобразования алгоритма. Дальнейшие исследования будут направлены на разработку нового вида функционального преобразования с последующим его внедрением в программно-аппаратный комплекс, что будет способствовать внедрению следующей версии ГОСТ Р 34.12-2015.

7. Литература

[1] ГОСТ Р 34.12-2015. Информационная технология. Криптографическая защита информации. Блочные шифры. – М.: Стандартинформ, 2015. – 25 с.

[2] Шеннон, К. Работы по теории информации и кибернетике / К. Шеннон. – М.: Изд-во иностранной литературы, 1963. – 830 с.

[3] Фомичев, М.И. Дискретная математика и криптология / М.И. Фомичев. – М.: Диалог-МИФИ, 2003. – 400 с.

[4] Султанов, Д.М.-М. Концепция архитектуры шифровального устройства, реализующего алгоритм криптографического преобразования ГОСТ Р 34.12-2015, на базе ПАК. Расширение режимов криптографического преобразования ГОСТ Р 34.12-2015 на базе специализированного микроконтроллера / Д.М.-М. Султанов, И.И. Василишин, М.С. Пугин // Параллельные Вычислительные технологии Труды междунар. науч. конф. – Челябинск, Издательский центр ЮУрГУ, 2016. – 797 с.

[5] Лидл, Р. Прикладная абстрактная алгебра / Р. Лидл, Г. Пильц. – Екатеринбург: Уральский университет, 1996. – 743 с.

Using component-wise functions in cryptographical transformation algorithm from Russian National Standard GOST R 34.12-2015

I.I. Vasilishin[1], S.Y. Korabalshchikova[1], D.M.-M. Sultanov[2], M.S. Pugin[2]

[1]Northern (Arctic) Federal University named after M.V. Lomonosov; Severnaya Dvina 17, Arkhangelsk, Russia, 163007
[2]Forensic Science Center of the Arkhangelsk Oblast Regional Office of the Ministry of Internal Affairs of Russian Federation, Voskresenskaya St. 3, Arkhangelsk, Russia, 163003

Abstract. The paper presents the general approach to selecting functions, keeping the first argument field, in the process of symmetric encryption of a plaintext. Quantitative estimation and general characteristics of ordinate vector for such functions are given. Ten component-wise functions of binary logic algebra of three arguments, replacing one function of digit-wise addition modulo two in the cryptographic transformation algorithm from Russian National Standard GOST R 34.12-2015, are presented in the paper. Using component-wise functions widens the range of intermediate options of round transformations in block encryption, which complicates the decryption (cracking) algorithm for the cipher.

Keywords: binary logic functions, cryptographic transformation, the encryption algorithm, GOST R 34.12-2015.