УДК 004.056.55 DOI: 10.30748/soivt.2018.53.23

И.В. Лысенко, М.А. Гвоздинский

Национальный аэрокосмический университет им. Н.Е. Жуковского «ХАИ», Харьков

ПОДХОД К ФОРМИРОВАНИЮ РАСПИСАНИЯ КЛЮЧЕЙ ДЛЯ БЛОЧНОГО СИММЕТРИЧНОГО КРИПТОАЛГОРИТМА ГОСТ 28147-89

Рассмотрены подходы к построению процедур расписания ключей блочных симметричных криптоалгоритмов. В рамках одного из них предлагается подход к формированию расписания ключей для криптоалгоритма ГОСТ 28147-8 в целях повышения его криптостойкости. Предложенный подход является модификацией ранее разработанного подхода к формированию расписания ключей криптоалгоритма ГОСТ 28147-8 на основе идеи зависимости выбора ключа раунда от текущего значения преобразуемого блока данных.

***Ключевые слова**: блочный симметричный криптоалгоритм, ГОСТ 28147-89, расписание ключей.*

Введение

Постановка задачи. Несмотря на значительные достижения современной криптографии проблема защиты данных, в частности, обеспечения их конфиденциальности, не потеряла своей актуальности. Как правило, эта задача решается путём применения преимущественно симметричных (блочных и поточных) криптографических алгоритмов шифрования.

Де юре и де факто международным стандартом блочного симметричного шифрования является криптоалгоритм Rijndael, или, как его чаще называют, AES (в связи с тем, что он оказался победителем конкурса на разработку современного стандарта шифрования Advanced Encryption Standard (AES), проводимого в конце 1990-х гг. Национальным институтом стандартизации и технологий США) [1]. AES, в основе которого лежит нетрадиционная (в смысле отличия от SPN-сети и сети Фейстеля) KALST-сеть, ориентирован на работу с 128-разрядными блоками данных и ключами размерности 128, 192 и 256 битов.

В Украине в качестве обязательного для использования в госучреждениях до недавнего времени применялся блочный симметричный криптоалгортм ГОСТ 28147-89, который был принят, как следует из его обозначения, в 1989 году, и был разработан для шифрования данных, составлявших тайну государственной важности. Данный криптоалгоритм, заслуживший высокую оценку криптолога с мировым именем Брюса Шнайера [2], представляет собой блочный шифр, оперирующий с 64-битными блоками, использующим 256-битный ключ и 32 раунда преобразования; в его основе лежит сеть Фейстеля.

Процедура расширения ключа в алгоритме ГОСТ 28147-89, фактически, отсутствует: в раундах шифрования последовательно используются 32-битные фрагменты K1…K8 исходного 256-битного ключа шифрования в следующем порядке: K1, K2, K3, K4, K5, K6, K7, K8, за исключением последних 8 раундов – в раундах с 25-го по 32-й фрагменты используются в обратном порядке [2]. С точки зрения производительности, ГОСТ 28147-89 существенно уступает современным аналогам, таким, как, например, упомянутый криптоалгоритм AES.

Упомянутые обстоятельства, а также успехи в области криптоанализа алгоритмов блочных симметричных шифров, послужили причинами разработки и введения в действие нового современного стандарта шифрования в Украине, способного стать основой для создания эффективных средств криптографической защиты следующих поколений. Проведенный в Украине открытый конкурс криптографических алгоритмов позволил определить перспективный блочный шифр, на основе которого и был разработан новый национальный криптографический стандарт блочного преобразования ДСТУ 7624:2014 («Калина»), учитывающий мировой опыт построения современных шифров. При разработке стандарта использовался прозрачный и консервативный подход, позволяющий обеспечить доказуемую стойкость к ряду методов криптоанализа. Новый национальный стандарт поддерживает размер блока и длину ключа шифрования 128, 256 и 512 бит (длина ключа равна размеру блока или в два раза превышает его), обеспечивая нормальный, высокий и сверхвысокий уровень стойкости. Разные варианты обеспечивают гибкость выбора параметров для разработчиков систем криптографической защиты, что позволяет получить как наивысший уровень быстродействия, так и наибольший запас стойкости преобразования. Высокоуровневая конструкция использует хорошо исследованную Square-подобную SPN-структуру, применяемую в алгоритмах AES, Whirlpool, «Стрибог», «Кузнечик» и др. Сравнение свойств основных компонентов алгоритма с другими шифрами показывает преимущество именно национального стандарта Украины, обеспечивающего

больший, запас стойкости к основным методам криптоанализа. Производительность шифра сопоставима с AES или превосходит его на 64-битовых платформах. При программной реализации на современном оборудовании быстродействие национального стандарта Украины существенно выше, чем у ГОСТ 28147-89 [3].

Исходя из сказанного выше, можно заметить, что, если не считать для пользователя критичным производительность криптопреобразований, основным недостатком криптоалгоритма ГОСТ 28147-89 является отсутствие как такового расписания ключа. Под этим термином понимаются все части механизма, определяющего вхождение ключевых элементов в шифрующие процедуры. При этом в качестве ключевых элементов могут использоваться непосредственно некоторые части секретного ключа, формируемым по некоторым достаточно сложным процедурам. Данные процедуры называются процедурами усложнения секретного ключа, а полная совокупность вырабатываемых ими ключевых элементов – расширенным ключём. На наш взгляд, можно модифицировать алгоритм формирования расписания ключей для ГОСТ 28147-89, тем самым обеспечив повышение его криптостойкости и использовать его наряду с другими криптоалгоритмами с учётом упомянутого выше обстоятельства. Попытка решения этой задачи уже предпринималась и описана в работе [4].

Цель данной работы: разработка модели формирования расписания ключей для блочного симметричного криптоалгоритма ГОСТ 28147-89.

С этой целью в первую очередь необходимо проанализировать подходы к построению процедур формирования расписания ключей, используемых в блочных симметричных криптоалгоритмах.

Основная часть

Анализ процедур расписания ключей блочных симметричных криптоалгоритмов.

Алгоритм шифрования можно логически разделить на два субалгоритма: собственно, шифрующие преобразования и процедура расширения ключа (рис. 1).

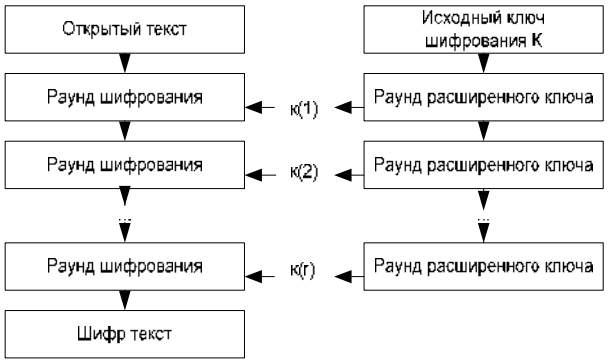

Рис. 1. Назначение процедуры расширения ключа

К процедуре расписания ключей (или, иначе, расширения исходного секретного ключа) предъявляется немало требований, целью которых является повышение криптостойкости и других характеристик алгоритма. Весьма желательно, чтобы процедура расширения ключа могла вычислять ключи «на лету» (on-the-fly), т.е. параллельно с шифрующими преобразованиями: это позволит как распараллеливать вычисления в многопроцессорных системах, так и не тратить память для хранения всего расширенного ключа при шифровании в условиях ограниченных ресурсов.

Помимо этого, существует специальная классификация блочных шифров по типу используемой процедуры генерации расширенного ключа (табл. 1), в соответствии с которой каждому алгоритму может быть присвоен двузначный индекс. При этом первый символ имеет всего два значения (1 или 2), а второй – три (A, B, C) [5].

К первой группе относятся блочные шифры, для которых знание раундового ключа позволяет определить все или некоторые биты основного ключа или остальных раундовых ключей. К этой группе относятся все шифры, в которых используется так называемое расписание ключей, когда каждый раундовый ключ является подмножеством битов основного ключа.

Ко второй группе относятся блочные шифры, для которых главным критерием является зависимость битов раундовых ключей от всех или не всех битов основного ключа. К этой группе относится большинство современных блочных шифров, участвовавших в конкурсе AES (табл. 1).

Достаточно очевидно, что наиболее безопасными согласно классификации являются блочные шифры 2B и 2C. Однако использование стойкой процедуры генерации расширенного ключа приводит к снижению скорости преобразования данных, в некоторых случаях весьма существенному, поскольку требуется дополнительное время на генерацию очередного ключа. Данное обстоятельство особенно критично для систем защиты информации, которые используют режим частой смены ключа, а объем преобразуемой информации небольшой. В этом случае для формирования расширенного ключа предпочтительным является применения простого расписания ключей.

Существует несколько способов построения расписания ключей [6].

Использование предвычислений для формирования расширенного ключа позволяет обеспечить сложную зависимость раундовых ключей от секретного ключа. При этом расширенный ключ представляет собой псевдослучайную последовательность. Недостатком этого подхода является снижение скорости шифрования в приложениях, требующих частой смены ключей.

Таблица 1

Классификация процедур генерации расширенного ключа

Индекс	Характеристика	Примеры
1A	Знание раундового ключа позволяет однозначно восстановить основной ключ и остальные раундовые ключи	SPECTR-128, NDC
1B	Знание раундового ключа позволяет восстановить некоторые биты основного ключа и/или остальных раундовых ключей	DES, ГОСТ, SPECTR-H64
1C	Знание раундового ключа позволяет восстановить некоторые биты основного ключа и/или остальных раундовых ключей после выполнения некоторых сравнительно простых арифметических операций	Rijndael, Crypton, DEAL, I DEA
2A	Не все биты основного ключа используются для формирования раундового ключа, и знание раундового ключа не позволяет восстановить основной или расширенный ключ	DFC, CAST-128
2B	Все биты основного ключа используются для формирования каждого раундового ключа, но знание раундового ключа не позволяет восстановить основной или расширенный ключ	Blowfish, LOK1-97, Serpent, CAST-256, Twofish, RC6, E2, Mars, Frog, HPC
2C	Каждый раундовый ключ формируется независимо от остальных раундовых ключей, и размерность расширенного ключа совпадает с размерностью основного ключа	DES с независимыми ключами, «Калина»

Непосредственное использование секретного ключа заключается в использовании частей (размером 32 или 64 бита) секретного ключа в качестве раундовых ключей. Примером шифров, в которых используется такой подход, является российский стандарт ГОСТ 28147-89. Недостатком такого подхода к формированию раундовых ключей является то, что раундовые ключи являются явно зависимыми, что может быть использовано при криптоанализе. Кроме того, оценка стойкости шифра, выполняемая при его проектировании, существенно усложняется необходимостью учёта данного обстоятельства. Недостатком представляется также наличие большого числа слабых ключей, т. е. таких ключей, для которых процедура зашифрования совпадает с процедурой расшифрования. Достоинством непосредственного использования частей секретного ключа в качестве раундовых ключей является то, что обеспечивается сохранение высокой скорости шифрования в режиме частой смены ключей.

Формирование раундовых подключей в процессе шифрования блока данных. В этом подходе при аппаратной реализации в качестве первого раундового ключа используется часть секретного ключа, а при выполнении первого раунда шифрования осуществляется формирование второго раундового подключа. При выполнении второго раунда шифрования вычисляется третий раундовый ключ и т. д. Такой ход формирования раундовых ключей имеет место как при выполнении зашифрования, так и при выполнении расшифрования. Учитывая связь между очередностью использования раундовых ключей в этих двух режимах, легко увидеть важность обеспечения формирования одинаковых раундовых ключей на *i*-том раунде расшифрования и (R–i+1)-м раунде зашифрования, где R – число раундов криптоалгоритма.

Преобразование подключей в зависимости от преобразуемых данных заключается в том, что части секретного ключа используются непосредственно, но перед их наложением на подблоки данных они преобразуются с помощью операций, зависящих от текущего значения одного из подблоков данных. Такое преобразование (механизм внутреннего усложнения ключа) может быть выполнено одновременно с преобразованием другого подблока данных, поэтому оно не приводит к снижению скорости шифрования, хотя обеспечивает существенное улучшение характеристик раундового преобразования.

Таблица 2

Достоинства и недостатки способов построения расписания ключей

Способ	Достоинства	Недостатки
Использование предвычислений	Обеспечивается сложная зависимость раундовых ключей от секретного ключа	Снижение скорости шифрования Требует дополнительных аппаратных ресурсов
Непосредственное использование секретного ключа	Обеспечивается сохранение высокой скорости шифрования в режиме частой смены ключей	Раундовые ключи являются явно зависимыми. Наличие большого числа слабых ключей
Формирование раундовых подключей в процессе шифрования блока данных	Обеспечивается высокая производительность криптосистемы при частой смене ключей	Требует дополнительных аппаратных ресурсов
Преобразование подключей в зависимости от преобразуемых данных	Позволяет существенно упростить аппаратную реализацию. Обеспечивается высокая производительность криптосистемы при частой смене ключей	Проблема устранений слабых ключей

Модель формирования расписания ключей для криптоалгоритма ГОСТ 28147-89.

На наш взгляд, на основе объединения второго и четвертого из упомянутых подходов может быть предложен подход, идея которого состоит в том, чтобы элементы множества подключей выбирались на каждом раунде не строго установленным и известным образом, а задавались текущим значением преобразуемого подблока.

Для описания предлагаемого подхода примем следующие обозначения:

d – номер раунда (d = 1,…, 32);

K_j – ключ раунда (j = 1,…, 8);

M_{li}, M_{ri} – левый и правый подблоки i-го блока исходных данных M_i;

$N_{li}^{(1)}$, $N_{ri}^{(0)}$ – число единичных (нулевых) битов в подблоках M_{li} и M_{ri} соответственно.

С целью обеспечения неоднозначности (для криптоаналитика) выбора ключа K_j будем использовать преобразования:

$$S_{li} = (N_{li}^{(1)} + d)(\text{mod } 8);$$

$$S_{ri} = (N_{ri}^{(1)} + d)(\text{mod } 8).$$

Здесь параметры S_{li} и S_{ri} задают номер выбираемого раундового ключа K_j на нечётном и чётном раунде шифрования соответственно.

Если S_{li} или S_{ri} = 0, то, например, выбирается значение S_{li} = 3 и S_{ri} = 2.

Можно здесь заметить, что сам выбор значений параметров S_{li} при их равенстве нулю, не обязательно должен быть статическим, но может быть определён по некоторой динамической схеме (процедуре).

Рассмотрим пример.

Пусть $N_{li}^{(1)} = 18$, $N_{ri}^{(1)} = 12$.

Тогда для последовательных раундов имеем:

1: $S_{l1} = (18+1)(\text{mod } 8) = 3$; $S_{r1} = (12+1)(\text{mod } 8) = 5$.
2: $S_{l2} = (18+2)(\text{mod } 8) = 4$; $S_{r2} = (12+2)(\text{mod } 8) = 6$.
3: $S_{l3} = (18+3)(\text{mod } 8) = 5$; $S_{r3} = (12+3)(\text{mod } 8) = 7$.

И таким образом в продолжение 32 раундов.

Результирующая последовательность будет иметь вид: 3, 6, 5, 2, 7, 2, 4, 3, 6, 5, 2, 7, 2, 1, 4, 3, 6, 5, 2, 7, 2, 1, 4, 3, 6, 5, 2, 7, 2, 1, 4. Она задаёт порядок выбора номеров ключей раундов в процессе шифрования данных.

Для реализации этого подхода, очевидно, необходимо обеспечить однозначность использования подключей в процессе шифрования и расшифрования данных. Это можно реализовать шифрованием с помощью исходного ключа K параметров S_{li} и S_{ri} и добавив их к шифротексту. То есть будет передаваться в зашифрованном виде n пар чисел (S_{li}, S_{ri}); для одного блока – 32 пары, если шифруется n блоков, то – 32*n пар чисел. Таким образом в алгоритм вводится избыточность.

Ранее упоминалось о попытке решения данной задачи в работе [4]. Идея заключалась в объединении подходов с использованием предвычислений и непосредственного использования секретного ключа. В одном из вариантов реализации этого подхода для избавления от явной зависимости от секретного ключа, вводится процедура, представляющая собой набор перестановок Π_i, которая независимо от секретного ключа позволит представлять ключ в виде псевдослучайной последовательности K_1. Следует отметить, что перед шифрованием к сообщению M_i добавляется набор перестановок Π_i, который в процессе расшифрования разделяется и используется для дешифрования сообщения M_{i+1}.

Таким образом, этот подход вводит избыточность как и предлагаемый здесь, но кроме этого требует реализации процедуры формирования последовательности перестановок.

Предложенный же подход использует непосредственно секретный ключ и преобразование раундовых ключей в зависимости от преобразуемых данных, что позволяет миновать таких проблем как снижение скорости шифрования и необходимость дополнительных аппаратных ресурсов.

Выводы

Таким образом, можно предположить, что описанный подход должен повысить криптостойкость алгоритма ГОСТ 28147-89 за счёт ввода зависимости раундовых ключей от текущего значения преобразуемых данных и неизвестности для злоумышленника последовательности выбора подключей на раундах шифрования (и увеличеной сложности решения этой задачи).

Недостатком подхода является вводимая избыточность.

Вследствие этого обстоятельства конкретному пользователю нужно определить целесообразность использования данного подхода, принимая во внимание соотношение между требованиями криптостойкости и производительности криптопреобразований.

Список литературы

1. Горбенко І.Д. Прикладна криптологія. Теорія. Практика. Застосування: моногр. / І.Д. Горбенко, Ю.І. Горбенко. – Харків: Форт, 2012. – 870 с.

2. Шнайер Б. Прикладная криптография. Протоколы, алгоритмы, исходные тексты на языке Си / Б. Шнайер. – М.: Триумф, 2003. – 816 с.

3. Горбенко Ю.І. Побудування та аналіз систем, протоколів і засобів криптографічного захисту інформації. Ч.1: Методи побудування та аналізу, стандартизація та застосування криптографічних систем: монографія / Ю.І. Горбенко; за заг. ред. І.Д. Горбенка. – Харків: Форт, 2015. – 959 с.

4. Асташкина Е.Н. Подход к формированию расписания ключей для блочного симметричного криптоалгоритма ГОСТ 28147-89 / Е.Н. Асташкина, И.В. Лысенко // Системи обробки інформації. – Харків: ХУПС. – 2010. – Вип. 6(87). – С. 30-34.

5. Молдовян Н.А. Криптография. От примитивов к синтезу алгоритмов / Н.А. Молдовян, А.А. Молдовян, М.А. Еремеев. – СПб.: БХВ-Петербург, 2004. – 448 с.

6. Молдовян А.А. Криптография: скоростные шифры / А.А. Молдовян, Н.А. Молдовян, Н.Д. Гуц, Б.В. Изотов. – СПб.: БХВ-Петербург, 2002. – 496 с.

References

1. Horbenko, I.D. and Horbenko, U.I. (2012), "*Prikladna kriptologija. Teorija. Praktika. Zastosuvannja: monografija*" [*Applied cryptology. Theory. Practice. Application: monograph*], Fort, Kharkiv, 870 p.

2. Shnajer, B. (2003), "*Prikladnaja kriptografija. Protokoly, algoritmy, ishodnye teksty na jazyke Si*" [*Applied cryptography. Protocols, algorithms, source texts in C language*], Triumf, 816 p.

3. Horbenko, I.D. and Horbenko, U.I. (2015), "*Pobuduvannja ta analiz sistem, protokoliv i zasobiv kriptografichnogo zahistu informacij. Ch.1: Metodi pobuduvannja ta analizu, standartizacija ta zastosuvannja kriptografichnih sistem: monografija*" [*Construction and analysis of systems, protocols and tools of cryptographic information security. Part 1: Methods of construction and analysis, standardization and application of cryptographic systems: monograph*], Fort, Kharkiv, 959 p.

4. Astashkina, E.N. and Lysenko, I.V. (2010), "Podhod k formirovaniju raspisanija kljuchej dlja blochnogo simmetrichnogo kriptoalgoritma GOST 28147-89" [The approach to creating a key schedule for the block symmetric crypto algorithm GOST 28147-89], *Information processing systems*, No. 6 (87). pp. 30-34.

5. Moldovjan, N.A., Moldovjan, A.A. and Eremeev, M.A. (2004), "*Kriptografija. Ot primitivov k sintezu algoritmov*" [*Cryptography. From primitives to the synthesis of algorithms*], BHV-Peterburg, Saint Petersburg, 448 p.

6. Moldovjan, N.A., Moldovjan, A.A., Guc, N.D. and Izotov, B.V. (2002), "*Kriptografija: skorostnye shifry*" [*Cryptography: speed ciphers*], BHV-Peterburg, Saint Petersburg, 496 p.

Надійшла до редколегії 14.02.2018
Схвалена до друку 20.03.2018

Відомості про авторів:

Лисенко Ігор Володимирович
кандидат технічних наук доцент
доцент кафедри Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ»,
Харків, Україна
e-mail: i.lysenko@csn.khai.edu

Гвоздинський Михайло Олександрович
Магістрант Національного аерокосмічного університету ім. М.Є. Жуковського «ХАІ»,
Харків, Україна
https://orcid.org/0000-0001-5979-1896
e-mail: mihailgvozdinskiy@gmail.com

Information about the authors:

Igor Lysenko
Candidate of Technical Sciences Associate Professor
Senior Lecturer of National Aerospace University "Kharkiv Aviation Institute" (KhAI),
Kharkiv, Ukraine
e-mail: i.lysenko@csn.khai.edu

Mykhailo Hvozdynskyi
Master of National Aerospace University "Kharkiv Aviation Institute" (KhAI),
Kharkiv, Ukraine
https://orcid.org/0000-0001-5979-1896
e-mail: mihailgvozdinskiy@gmail.com

ПІДХІД ДО ФОРМУВАННЯ РОЗКЛАДУ КЛЮЧІВ ДЛЯ БЛОЧНОГО СИМЕТРИЧНОГО КРИПТОАЛГОРИТМУ ГОСТ 28147-89

І.В. Лисенко, М.О. Гвоздинський

Розглянуто підходи до побудови процедур розкладу ключів блокових симетричних криптоалгоритмів. В рамках одного з них пропонується підхід до формування розкладу ключів для криптоалгоритму ГОСТ 28147-8 з метою підвищення його криптостійкості. Запропонований підхід є модифікацією раніше розробленого підходу до формування розкладу ключів криптоалгоритму ГОСТ 28147-8 на основі ідеї залежності вибору ключа раунду від поточного значення блоку даних, що перетворюється.

***Ключові слова**: блочний симетричний криптоалгоритм, ГОСТ 28147-89, розклад ключів.*

THE APPROACH TO THE CREATION OF BLOCK SCHEDULING KEYS FOR BLOCK SYMMETRIC CRYPTOGRAPHIC ALGORITHM GOST 28147-89

I. Lysenko, M. Hvozdynskyi

Approaches to the construction of scheduling procedures for keys of block symmetric crypto algorithms are considered. In the framework of one of them, an approach is proposed to the generation of a key schedule for the cryptographic algorithm GOST 28147-8 in order to increase its cryptographic strength. The proposed approach is a modification of the previously developed approach to the generation of the schedule of cryptographic algorithm keys GOST 28147-8 on the basis of the idea of the dependence of the choice of the key of the round on the current value of the data block being converted.

***Keywords:** Symmetric Block cryptographic algorithm, GOST 28147-89, keys schedule.*