УДК 004.056

Л.В. ДЕРБУНОВИЧ, д-р техн. наук, проф. НТУ «ХПИ»,
Д.Г. КАРАМАН, аспирант НТУ «ХПИ» (г. Харьков)

МЕТОДЫ ФУНКЦИОНАЛЬНОГО ДИАГНОСТИРОВАНИЯ ОШИБОК ШИФРОВАНИЯ В СИММЕТРИЧНЫХ КРИПТОГРАФИЧЕСКИХ СИСТЕМАХ

Проведено аналіз існуючих рішень з виявлення несправностей та помилок у симетричних криптографічних системах. Розглянуто аналітичну модель поширення помилок. Запропоновано узагальнену методику діагностування помилок шифрування, яка базується на спільних операціях, які використовуються у більшості криптографічних алгоритмів.

Known issues analysis of symmetric cryptographic systems fault detection is carried out. Analytic model for error coverage is considered. Generalized procedure for enciphering fault detection based on commonly used operations from most cryptographic algorithms is proposed.

Введение. Общепринятой практикой для защиты передаваемой или хранимой информации стало использование различных криптографических средств. Благодаря открытости и доступности описания, для большинства наиболее часто используемых симметричных алгоритмов шифрования была доказана их стойкость к атакам, направленным на их математический базис. Поэтому в последнее время значительно возрос интерес к атакам, направленным на конкретные реализации этих алгоритмов. Одной из наиболее эффективных и, в следствие этого, часто рассматриваемых в современной научной литературе атак является так называемая атака с внедрением ошибки (*fault injection attack*). Суть ее состоит в том, что если злоумышленнику удастся вызвать сбой в процессе выполнения криптографических преобразований в системе шифрования и воспользоваться результатами ее работы, то это значительно облегчит ему взлом и позволит получить секретный ключ. Впервые практическая эффективность подобной атаки была рассмотрена Д. Боне, Р. А. де Милло и Р. Дж. Липтоном в [1], а позднее предложены практические реализации для таких известных и распространенных симметричных алгоритмов шифрования как *AES* [2, 3] и *RC*5[4]. Кроме того, возникновение ошибки даже в 1 бит на одном из этапов криптографических преобразований, вследствие высокой нелинейности функций шифрования, а так же многократных подстановок и перестановок, предусмотренных алгоритмом, может привести к полной нечитаемости всего блока данных [3, 4]. К еще более катастрофическим последствиям может привести случайное или намеренное возникновение ошибки при расширении ключа.

Большинство решений, представленных в отечественных и зарубежных публикациях, сосредоточено на решении проблемы возникновения ошибок для конкретных алгоритмов шифрования или их отдельных преобразований в различных специфических условиях, тогда как существует необходимость в

разработке общей методологии и обобщенной теории решения описанной проблемы.

Анализ литературы. Активное обсуждение возможностей использования ошибок, возникающих на этапе шифрования, а так же методы их искусственного внедрения (в ведущих отраслевых журналах и на криптографических конференциях) началось с уже упомянутого доклада [1] на конференции *EUROCRYPT* в 1997 году. В докладе рассматривалась возможность использования ошибки в цифровой подписи, сгенерированной с помощью асимметричного алгоритма *RSA*, для упрощенной факторизации модуля и последующего взлома криптографической системы. Кроме того, были рассмотрены возможности использовании аппаратной неисправности для взлома идентификационных протоколов *Fiat-Shamir* и *Schnorr*. Несмотря на то, что авторами рассматривались только асимметричные криптосистемы, тот же подход был позже применен для взлома совсем нового по тем временам симметричного алгоритма *AES* [5, 6, 7]. Наиболее полный и всесторонний обзор способов искусственного внедрения ошибки можно найти в [8]. Наряду с разнообразием методик взлома с использованием ошибки при шифровании были предложены решения по ее обнаружению и устранению ее последствий. В [9] предложено использование непосредственного расшифрования только что зашифрованного блока исходя из того факта, что модуль расшифрования во время работы зашифровывающего модуля простаивает. Авторы рассматривают три различных варианта реализации этого подхода: на уровне всего модуля, на уровне отдельного раунда и на уровне каждой операции алгоритма. Эти варианты различаются между собой сложностью схемы и временем обнаружения ошибки. Такой подход можно применить к любому блочному шифру с циклической структурой, так как он не привязывается к какой-либо конкретной особенности того или иного алгоритма. Однако не всегда модули зашифровывания и расшифровывания реализуются совместно в одной аппаратной единице. В [10] и [11] предложены аналогичные методы по диагностике аппаратных блоков, реализующих криптографические преобразования по стандарту *AES*. Операция подстановки, входящая в состав алгоритма этого стандарта, является нелинейным преобразованием, в результате которого каждый байт исходного состояния заменяется новым байтом в соответствии со специальной таблицей, предусмотренной стандартом. Сбой, произошедший на этом этапе трудно обнаружить. Самый простой и очевидный способ проверки правильности функционирования этого блока: обеспечить его дублирование. Именно это решение с незначительными различиями в реализации описано в этих статьях. Кроме того, в [11] предложена методика использования битов четности (паритета) для диагностики неисправностей(ошибок) в остальных преобразованиях алгоритма. В последнее время предложено много решений для диагностики ошибок на базе специальных кодов. Так, в [12] обнаружение сбоев в аппаратной реализации алгоритма *DES* обеспечивается за счет сгенерированных на основе входных значений дополнительных

битов. В процессе шифрования эти биты передаются вместе с обрабатываемым блоком данных, обновляясь по мере необходимости, и после каждой операции участвуют в проверке блока данных на ошибки. В [3] и [4] предложена схема обнаружения ошибок, основанная на кодах проверки на четность (*parity-based error detection code*) для аппаратных реализаций алгоритмов *AES* и *RC*5 соответственно. В обоих вариантах на определенных этапах зашифровывания для каждого байта блока данных генерируется бит паритета. Схема шифрования, в свою очередь, дополняется средствами прогнозирования значения этих битов после каждой операции над блоком данных. Отклонения, обнаруженные при сравнении сгенерированных и спрогнозированных битов, означают, что в процессе выполнения последней операции возник сбой (ошибка).

Постановка задачи. В статье проводится анализ существующих решений по обнаружению неисправностей и ошибок в симметричных криптографических системах, а также предлагается обобщенный метод диагностирования ошибок шифрования, основанный на общих операциях, применяемых в большинстве криптографических алгоритмов.

Аналитическая модель распространения ошибок. Блок данных, подлежащий обработке, можно обозначить как $D=[d_j]$, $1 \le j \le m$, где d_j – элементарные единицы (байты или битовые слова) этого блока. Сигнатура ошибки представляется вектором $E=[e_j]$, $1 \le j \le m$, для которого e_j удовлетворяет условию $e_j = r_j - p_j$, где r_j – реальное значение бита проверки, а p_j – спрогнозированное значение бита проверки для *j*-той элементарной единицы блока данных. Оператор разности (–) зависит от алгебраической структуры выбранного кода обнаружения ошибок. В отсутствие ошибок $r_j = p_j$, а $E = O$, где O– это нулевой вектор.

В общем виде правило прогнозирования битов проверки выбранного кода выявления ошибок в любом из внутренних преобразований цикла (раунда) имеет следующий вид:

$$p'_j = f_j(p_1, p_2, ..., p_m, D),\ 1 \le j \le m \qquad (1)$$

где $p_1, p_2, ..., p_m$ – набор битов проверки из предыдущих преобразований раунда, а p'_j – биты проверки (БП) текущего преобразования. В идеальном случае БП следует получать только из БП предыдущих преобразований. Иногда это осуществимо, но зачастую они зависят еще и от блока данных D (или его части). Следует так же заметить, что если для каких-либо значений D биты проверки p'_j были получены независимо от p_j, то путь передачи сигнатуры ошибки нарушается.

В соответствии с определением e'_j представляет собой разность между битами проверки, сгенерированными по результатам текущего преобразования и функцией от битов проверки предыдущих преобразований:

$$e'_j = f_j(r_1, r_2, ..., r_m, D) - f_j(p_1, p_2, ..., p_m, D) \quad (2)$$

Это выражение можно записать в следующем виде:

$$e'_j = f_j(p_1 + e_1, p_2 + e_2, ..., p_m + e_m, D) - f_j(p_1, p_2, ..., p_m, D) \quad (3)$$

Уравнение (3) описывает обобщенное правило распространения сигнатуры ошибки. Более глубокий анализ возникновения и поведения ошибок может быть проведен для конкретно выбранного алгоритма шифрования и его реализации.

Способы диагностирования ошибок шифрования. В результате анализа известных схем диагностирования можно выделить два основных подхода к реализации механизмов проверки: дублирование проверяемого оборудования и подача на оба блока одинаковых исходных данных с последующим сравнением полученных результатов и использование различных проверочных кодов, средства реализация которых внедряется в основную схему шифрования.

Первый способ является более очевидным и простым в реализации. Его можно применить в устройствах с любым алгоритмом шифрования и при этом он обеспечит практически полное покрытие неисправностей, как одиночных, так и групповых, вне зависимости от их природы возникновения. Даже в случае возникновения ошибок как в контролируемом, так и в контролирующем блоках, вероятность их полной идентичности слишком мала, чтобы привести к взаимному маскированию. Что касается атак, существующие технологии пока позволяют гарантированно вызвать сбой в оборудовании, но не дают возможности конкретно определить его место. К недостаткам первого способа можно отнести высокую степень аппаратных затрат. В зависимости от степени интеграции и методов исполнения эти затраты могут составлять от 30% до 100% от средств, затрат на реализацию самого алгоритма.

Коды обнаружения ошибок, в свою очередь, могут уступать по эффективности предыдущему способу, но зато их использование может способствовать существенной экономии дополнительных аппаратных средств. Кроме того, механизмы проверки, основанные на таких кодах можно внедрить как на уровне модуля шифрования в целом, так и на уровне его базовых операций. Причем, чем меньше проверяемый блок, тем выше степень покрытия ошибок, как одиночных, так и групповых, но тем больше сложность схемы и объем аппаратных затрат на её реализацию.

Коды обнаружения ошибок можно разделить на две группы: коды, основанные на проверке четности и арифметические коды в остаточных классах по модулю 3, 7 и 15. Выбор конкретного типа используемого кода обычно зависит от характера и сложности проверяемого преобразования. Для организации схемы проверки, основанной на каком-либо из кодов, необходимо

реализовать схему генерации битов проверки для зашифровываемого блока данных, схем прогнозирования изменения этих битов для каждого из используемых при зашифровывании вида операции, а так же схем сравнения этих битов и отработки ситуации с обнаруженной ошибкой. Обычно все эти схемы располагаются между отдельными операциями, преобразованиями или циклами (раундами) алгоритма, что приводит к дополнительным задержкам и негативно сказывается на быстродействии схемы. Исходя из сказанного, авторы решений, основанных на проверочных кодах, стараются найти разумный компромисс между степенью покрытия ошибок, сложностью и быстродействием схемы шифрования с учетом механизмов проверки.

Основные операции симметричных шифров. Перед тем, как выбирать конкретный способ обнаружения ошибок, необходимо проанализировать структуру выбранного алгоритма шифрования и рассмотреть операции, которые используются в его преобразованиях. Все современные алгоритмы шифрования используют определенный набор операций, который включает в себя практически все элементарные логические и арифметические функции (как с переносом, так и по модулю). Список операций, наиболее часто используемых в современных алгоритмах шифрования, приведен в табл. 1.

Таблица 1.

Алгоритм	Исключающее ИЛИ	И	ИЛИ	mod n (n>2)			Расширение	Подстановка (S-Box)	Сдвиг	Цикл. сдвиг	Перемешивание	× mod $GF(x)$
				+	–	×						
Blowfish	32			32				8→32				
Camelia	8, 32, 64	32	32					8→8		32	32, 64	
CAST-256	32			32	32			8→32		32		
DES	32, 48						32→48	6→4			1	
IDEA	16			16		16					16	
MARS	32			32	32	32		8, 9→32		32	8	
RC5	32			32						32		
RC6	32			32		32				32	8	
Rijndael	8							8→8			8	8
Serpent	32							4→4	32	32		
Twofish	32			32				8→8		32	8, 64	8

В перечень алгоритмов были включены все пять финалистов конкурса нового стандарта шифрования *AES*, устаревший, но очень распространенный *DES*, международный стандарт *IDEA*, а также еще несколько наиболее криптостойких современных алгоритмов. Кроме того, таблица отображает, какие операции в каких алгоритмах используются и какая размерность у их операндов в битах.

Сущность методики функционального диагностирования ошибок. Проведя анализ в соответствии с методиками прогнозирования проверочных битов для различных операций, описанными в [3] и [4], были определены наиболее подходящие способы определения ошибок для аппаратных реализаций операций из табл. 1. Результаты приведены в табл. 2.

Для некоторых операций существует несколько вариантов реализации механизмов обнаружения ошибок функционирования. Выбор конкретного варианта зависит от используемого алгоритма, особенностей аппаратной платформы и размера операндов (табл. 1).

Таблица 2.

Операция	Способ определения ошибок
Исключающее ИЛИ	Методы контроля четности
И, ИЛИ	Дублирование
+, –, × mod n ($n > 2$)	Коды в остаточных классах,
Расширение	Дублирование
Подстановка (*S-Box*)	Зависит от реализации, методы контроля четности, дублирование
Сдвиг	Дублирование, коды в остаточных классах
Цикл. сдвиг	Коды в остаточных классах, дублирование
Перемешивание	Дублирование
× mod $GF(x)$	Дублирование, если один из множителей константа – методы контроля четности

Вывод. В статье проведен анализ существующих методов обнаружения неисправностей и ошибок в симметричных криптографических системах и предложен обобщенный подход к реализации процедуры диагностирования ошибок шифрования.

Список литературы: **1**. *Boneh D., DeMillo R.A. and Lipton R.J.* On the Importance of Checking Cryptographic Protocols for Faults // Advances in Cryptology (EUROCRYPT'97), Lecture Notes in Computer Science. – 1997. – vol. 1233. – p. 37-51. **2**. *Blomer J., Seifert J.* Fault Based Cryptanalysis of Advanced Encryption Standard (AES) // Financial Cryptography (FC 2003), Lecture Notes in Computer Science. – 2003. – vol. 2742. – p. 162-181. **3**. *Bertoni G., Breveglieri L., Koren I., Maistri P., Piuri V.* Error Analysis and Detection Procedures for a Hardware Implementation of the Advanced Encryption Standard. // IEEE Trans. on Computers. – 2003. – vol. 52, no. 4. – p. 492-505. **4**. *Bertoni G., Breveglieri L., Koren I., Maistri P., Piuri V.* Concurrent Fault Detection in a Hardware Implementation of the RC5 Encryption Algorithm // Proc. IEEE Int'l Conf. Application-Specific Systems, Architectures, and Processors. – 2003. – p. 410-419. **5**. *Bloemer J., Seifert J.-P.* Fault Based Cryptanalysis of the Advanced Encryption Standard (AES) // Proc. Seventh Int'l Conf. Financial Cryptography. – 2003. – p. 162-181. **6**. *Giraud C.* DFA on AES // Lecture Notes in Computer Science. – Vol. 3373/2005. – p. 27-41. **7**. *Piret G., Quisquater J.-J.* A Differential Fault Attack Technique against SPN Structures, with Application to the AES and Khazad // Proc. Fifth Int'l Workshop Cryptographic Hardware and Embedded Systems (CHES '03). – 2003. – p. 77-88. **9**. *Karri R., Wu K., Mishra P., Yongkook K.* Fault-Based Side-Channel Cryptanalysis Tolerant Rijndael Symmetric Block Cipher Architecture // Proc. IEEE Int. Symp. Defect and Fault Tolerance in VLSI Systems. – 2001. – pp. 427-435. **10**. *G. Di Natale, Flottes M.L., Rouzeyre B.* On-Line Self-Test of AES Hardware Implementations. // DSN 2007 Workshop on Dependable and Secure Nanocomputing In conjunction with the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. – 2007. **11**. *Yu N., Heys H.M.* A Hybrid Approach to Concurrent Error Detection for a Compact ASIC Implementation of the Advanced Encryption Standard // Proceedings of IASTED International Conference on Circuits, Signal, and Systems (CSS 2007). – 2007. **12**. *Butter A.S., Kao C.Y. and Kuruts J.P.* DES Encryption and Decryption Unit with Error Checking, US patent US5432848, 1995.

Поступила в редколлегию 21.11.2008