УДК 681.3.06

к.т.н. с.н.с. Кузнецов А.А., к.т.н. с.н.с. Ковтун В.Ю., Сергиенко Р.В.

АСПЕКТЫ РЕАЛИЗАЦИИ КРИПТОАЛГОРИТМА ADE

1. Введение

Развитие новых информационных технологий обусловило существенное повышение требований, предъявляемых к криптографическим средствам защиты информации. Перспективные криптоалгоритмы должны в сжатые сроки обрабатывать огромные объемы информации и эффективно противостоять современным методам криптоанализа. Проведенный анализ показал, что алгебраические атаки при возможном комбинировании с другими известными подходами можно рассматривать как реальную угрозу надежности симметричных криптоалгоритмов [1 - 5]. В этом смысле, проводимый в Украине открытый конкурс симметричных блоковых криптоалгоритмов нацелен, очевидно, на отбор алгоритмов-кандидатов, использующих отработанные при проведении конкурсов AES и NESSIE подходы к конструированию шифросистем и учитывающих, в тоже время, возможности алгебраических методов криптоанализа, основанных на тонкой организации алгебраической структуры криптоалгоритма.

В работе [6] рассмотрена структура и основные криптопримитивы симметричного алгоритма ADE (Algorithm of Dynamic Encryption), предложенного в качестве кандидата на национальный стандарт симметричного шифрования Украины. В качестве ближайшего прототипа использован AES (FIPS - 197) [7] - национальный стандарт шифрования США.

Основное внимание при разработке ADE уделено возможности динамически управлять процессом линейного рассеивания и нелинейной замены в ходе криптографического преобразования информации. С этой целью в базовую структуру алгоритма AES введены динамически изменяемые блоки (криптопримитивы) линейного рассеивания и нелинейных замен, правила функционирования которых задаются значениями раундовых ключей. Свойства введенных примитивов криптоалгоритма ADE не уступают криптопримитивам алгоритма AES, а за счет их динамической смены на каждом раунде итеративной обработки удается динамично управлять процессом криптографического преобразования информации. Это, с одной стороны, не ухудшает (по сравнению с AES) стойкость ADE к известным методам криптоанализа, с другой стороны, существенно усложняет процесс формирования алгебраических уравнений, аналитически связывающих состояние открытого текста, шифртекста и ключа. Фактически, формируемые системы уравнений имеют вид случайных и плотных систем, что, по мнению авторов, существенно усложняет задачу криптоанализа, в частности, алгебраическими методами. **Целью данной статьи** является исследование аспектов практической реализации симметричного криптографического алгоритма ADE.

2. Структурная схема и основные преобразования алгоритма шифрования ADE

Предложенный в [6] алгоритм ADE построен по классической подстановочно-перестановочной схеме и использует все достоинства алгоритма AES. Основное внимание при разработке ADE уделено возможности динамически управлять процессом линейного рассеивания и нелинейной замены в ходе криптографического преобразования информации. С этой целью в базовую структуру алгоритма AES введены динамически изменяемые блоки (криптопримитивы) линейного рассеивания и нелинейной замены, правила функционирования которых задаются значениями случайного секретного параметра (раундового ключа). Свойства введенных примитивов криптоалгоритма ADE не уступают криптопримитивам ал-

горитма AES, а за счет их динамической смены на каждом раунде итеративной обработки удается динамически управлять процессом криптографического преобразования информации. Это, по мнению авторов, существенно усложняет задачу криптоанализа, в частности, основанных на алгебраических методах [6].

В упрощенном (частном) случае функционирование криптопримитивов ADE сводится к соответствующим блокам алгоритма AES, т.е. при фиксировании некоторых системных параметров алгоритм ADE легко трансформируется в AES. Структурная схема алгоритма ADE в общем виде представлена на рис. 1. На схеме обозначены:

- «SubByte» – блок нелинейного преобразования (подстановка), функционирующий под управлением раундового ключа;
- «ShiftRow» – блок функционального преобразования (на основе циклического сдвига), функционирующий под управлением раундового ключа;
- «MixColumn» – блок линейного преобразования (аналог перестановки), функционирующий под управлением раундового ключа;
- «AddRoundKey» – блок функционального преобразования (сложение по модулю 2 (XOR) случайного секретного параметра (ключа)).

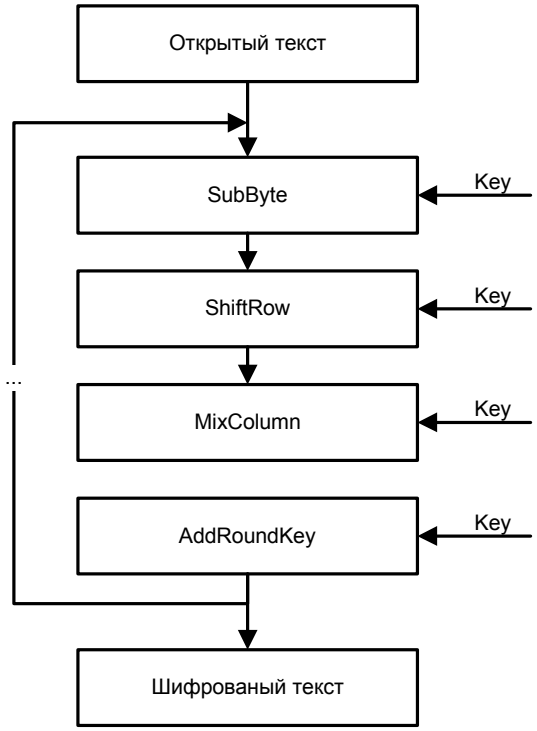

Рис. 1. – Структурная схема алгоритма шифрования ADE

Преобразование ByteSub является нелинейной заменой байтов, которое независимо оперирует с каждым байтом Состояния a и с байтом раундового ключа γ. В качестве S-блока выступает изменяемая матрица подстановок, которая строится получением мультипликативно обратного элемента $(a \cdot \gamma)^{-1}$ над конечным полем $\mathrm{GF}(2^8)$ и путем выполнения аффинного преобразования

$$b = M \cdot (a \cdot \gamma)^{-1} + \beta$$

над примитивным полем $\mathrm{GF}(2)$, M – квадратная невырожденная матрица 8x8, β – вектор из 8-и элементов. Подстановка - стохастичное преобразование, которое представляет собой не-

линейную замену байт, выполняемую с помощью S-блоков независимо для каждого входного байта и реализует отображение

$$\varphi : (A, \Gamma) \to B,$$

где A- множество входных векторов $a = \{a_0, a_1, ..., a_7\}$, B - множество выходных векторов $b = \{b_0, b_1, ..., b_7\}$, Γ - множество случайных векторов $\gamma = \{\gamma_0, \gamma_1, ..., \gamma_7\}$, которые задаются раундовым ключом, $\Gamma \in \mathrm{GF}(2^8)$. Выходной вектор можно представить как

$$y = \varphi(x, \gamma) = M \cdot (x \cdot \gamma)^{-1} + \beta,$$

где умножение $x \cdot \gamma$ выполняется над конечным полем $\mathrm{GF}(2^8)$.

Каждый $b = \{b_0, b_1, ..., b_7\}$ зависит как от $a = \{a_0, a_1, ..., a_7\}$, так и от случайной величины $\gamma = \{\gamma_0, \gamma_1, ..., \gamma_7\}$, которая задается значением раундового ключа. Обратным к ByteSub является замена байтов с применением инвертированной таблицы. Это достигается обращением аффинного отображения, за которым следует взятие мультипликативно обратного в поле $\mathrm{GF}(2^8)$.

Следует отметить, что при выборе восьмиразрядного вектора $\gamma = '0\ 1'$ преобразование ByteSub алгоритма ADE полностью соответствует блоку ByteSub алгоритма AES. Таким образом, в упрощенном (частном) случае (при $\gamma = '0\ 1'$) функционирование криптопримитива ByteSub в ADE сводится к соответствующему блоку алгоритма AES, т.е. на этом этапе зашифровывания алгоритм ADE легко трансформируется в AES.

В преобразовании ShiftRows алгоритма ADE строки Состояния сдвигаются на различное количество позиций, задаваемых значением раундового ключа (байта).

Через вектор

$$\lambda = \{\lambda_0 \quad \lambda_1 \quad \lambda_2 \quad \lambda_3 \quad \lambda_4 \quad \lambda_5 \quad \lambda_6 \quad \lambda_7\}$$

обозначим один байт раундового ключа.

Значения $\{\lambda_0, \lambda_1\}$ задают число сдвигов C0 для 0-й строки Состояния, значения $\{\lambda_2, \lambda_3\}$ задают число сдвигов C1 для 1-й строки Состояния, значения $\{\lambda_4, \lambda_5\}$ задают число сдвигов C2 для 2-й строки Состояния, значения $\{\lambda_6, \lambda_7\}$ задают число сдвигов C3 для 3-й строки Состояния. Отметим, что при соответствующих значениях элемента вектора $\lambda = \{\lambda_0, \lambda_1, ..., \lambda_7\}$ функционирование криптопримитива ShiftRows ADE сводится к соответствующему блоку алгоритма AES, т.е. на этом этапе зашифровывания алгоритм ADE легко трансформируется в AES. Так, например, при

$$\lambda = \{0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 1 \quad 1 \quad 1\}$$

имеем:

$$C0 = 0,\ C1 = 1,\ C2 = 2,\ C3 = 3,$$

т.е. строка 0 не смещается, строка 1 смещается на 1-байт, строка 2 – на 2 байта и строка 3 – на 3 байта, что полностью соответствует функционированию блока ShiftRows алгоритма AES для длины блока $Nb = 4$.

Обратным к ShiftRows есть циклический сдвиг на $Nb - C0$, $Nb - C1$, $Nb - C2$, $Nb - C3$ байт соответственно так, что байт на позиции j в строке i сдвигается на позицию $(j + Nb - Ci) \bmod Nb$.

Для построения динамически изменяемого блока линейного рассеивания алгоритма ADE со значением $B_{k,k}(\theta)$ не уступающим алгоритму AES использован МДР-код – расширенный $(256, 4i, 256 - 4i + 1)$ код Рида-Соломона над $\mathrm{GF}(2^8)$. Слой линейного рассеивания определен над множеством векторов «$4i$ байт в $4i$ байт», $i \in \{1,2,...,32\}$. Другими словами MixColumn алгоритма ADE реализует преобразование i столбцов Состояния:

– при $i=1$ MixColumn алгоритма ADE, как и алгоритма AES, обрабатывает поочередно по одному столбцу состояния;
– при $i=2$ MixColumn алгоритма ADE обрабатывает поочередно по два столбца состояния и т.д.

Для реализации на 32-х разрядных и менее процессорах рекомендуется использование MixColumn с $i=1$. Для 32-х процессоров ($i>1$) рекомендуется использование MixColumn с отображением «$4i$ байт в $4i$ байт».

Рассмотрим расширенный $(256, 4i, 256-4i+1)$ код Рида-Соломона над $\mathrm{GF}(2^8)$. По определению это МДР-код, который обеспечивает максимальное кодовое расстояние при заданных параметрах кода. Порождающая матрица кода задается выражением

$$\mathrm{G}=\begin{pmatrix} '00' & '01' & '02' & \dots & 'FF' \\ '00' & '01' & ('02')^2 & \dots & ('FF')^2 \\ '00' & '01' & ('02')^3 & \dots & ('FF')^3 \\ \dots & \dots & \dots & \dots & \dots \\ '01' & '01' & ('02')^{4i} & \dots & ('FF')^{4i} \end{pmatrix},$$

где возведение в степень производится в поле $\mathrm{GF}(2^8)$, $i \in \{1, 2, \dots, 32\}$.

Используя вектор из одного байта раундового ключа, обозначим его через s (элемент поля $\mathrm{GF}(2^8)$ порядка 255), сформируем невырожденную матрицу

$$\mathrm{M}'=\begin{pmatrix} s & s^2 & s^3 & \dots & s^{4i} \\ s^2 & s^4 & s^6 & \dots & s^{8i} \\ s^3 & s^6 & s^9 & \dots & s^{12i} \\ \dots & \dots & \dots & \dots & \dots \\ s^{4i} & s^{8i} & s^{12i} & \dots & s^{16i} \end{pmatrix},$$

которая является подматрицей определенной выше порождающей матрицы G расширенного $(256, 4i, 256-4i+1)$ кода Рида-Соломона над $\mathrm{GF}(2^8)$. Свойство невырожденности (обратимости) для M' обеспечивается условием максимального порядка элемента s в поле $\mathrm{GF}(2^8)$. В этом случае все столбцы и сроки матрицы M' различны и любая подматрица также обратима.

Рассмотрим вектор $a=\{a_0 \quad a_1 \quad a_2 \quad \dots \quad a_{4i-1}\}$ и вектор $b=\{b_0 \quad b_1 \quad b_2 \quad \dots \quad b_{4i-1}\}$ с элементами из $\mathrm{GF}(2^8)$. Функциональное соответствие

$$b=\theta'(a,s)=a\cdot\mathrm{M}'$$

задает линейное отображение

$$\theta' : \mathrm{GF}(2^8)^k \to \mathrm{GF}(2^8)^k,\ k=4i$$

с числом ветвей линейного рассеивания относительно значения раундового ключа s

$$B_{4,4}(\theta')=4i+1.$$

Набор элементов из $4i$ байт для преобразования в MixColumn формируется посредством объединения столбцов состояния в одномерный массив байт. Для упрощения ниже используется обозначение $\mathrm{MixColumn}_i$.

Инверсия преобразования $\mathrm{MixColumn}_1$ сходна с $\mathrm{MixColumn}_2$. Для рассмотренного способа каждый одномерный массив байт, образованный i-ми столбцами Состояния, преобразуется умножением его на матрицу $(\mathrm{M}')^{-1}$ обратную к матрице M' с операциями над $\mathrm{GF}(2^8)$.

Операция AddRoundKey алгоритма ADE полностью совпадает с аналогичной операцией алгоритма AES. В этой операции к Состоянию применяется преобразование RoundKey по-

средством простого побитного сложения по модулю 2. Раундовый ключ формируется из ключа шифра посредством процедуры формирования ключей. Длина раундового ключа равна длине блока Nb.

Процедура формирования ключей алгоритма ADE определена аналогично операции формирования ключей алгоритма AES. Раундовые ключи формируются из ключа шифра посредством процедуры формирования ключей. Она включает в себя два компонента: Расширение ключа и Выбор раундового ключа. Основной принцип заключается в следующем:

- Общее количество бит раундового ключа равно длине блока, умноженной на количество раундов плюс 1 (т.е. для блока длины 128 бит и 10 раундов необходимо 1408 раундовых ключей).
- Ключ шифра расширяется в Расширенный ключ.
- Раундовые ключи берутся из Расширенного ключа следующим образом: первый раундовый ключ состоит из первых Nb слов, второй – из следующих Nb слов, и так далее.

3. Аспекты реализации

Шифр ADE, как и AES, приспособлен для эффективной реализации на широком спектре процессоров и специализированном аппаратном обеспечении. В данной статье рассмотрены аспекты реализации на 32-битных и 64-битных процессорах, типичных для ПК.

3.1. 32-битный процессор

Различные шаги раундового преобразования алгоритма ADE, как и алгоритма AES, могут быть скомбинированы для обеспечения эффективной реализации на процессорах с длиной слова 32 или выше. В этой секции объяснено, как это можно сделать. Ниже показано, что при реализации четырех выборочных таблиц реализация алгоритма ADE на 32-х битных процессорах не уступает по быстродействию алгоритму AES. На 64-х битных процессорах реализация алгоритма ADE превосходит по быстродействию алгоритм AES.

Выразим один столбец выхода раунда e в величинах байт входа раунда a. В этой секции $a_{i,j}$ обозначает байт в строке i и столбце j, a_j обозначает столбец j Состояния a. Рассмотрим случай $\mathrm{MixColumn}_1$.

Раундовое преобразование с $MixColumn_1$

Реализация прибавления ключей и преобразования $\mathrm{MixColumn}_1$ с матрицей линейного рассеивания

$$\mathrm{M}'=\begin{pmatrix} s & s^2 & s^3 & s^4 \\ s^2 & s^4 & s^6 & s^8 \\ s^3 & s^6 & s^9 & s^{12} \\ s^4 & s^8 & s^{12} & s^{16} \end{pmatrix}$$

аналитически запишется в виде:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix},$$

где

$$\begin{bmatrix} d_{0,j} \\ d_{1,j} \\ d_{2,j} \\ d_{3,j} \end{bmatrix} = [\mathrm{M}] \begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix}.$$

Для преобразований ShiftRows и ByteSub мы имеем:

$$\begin{bmatrix} c_{0,j} \\ c_{1,j} \\ c_{2,j} \\ c_{3,j} \end{bmatrix} = \begin{bmatrix} b_{0,j-C0} \\ b_{1,j-C1} \\ b_{2,j-C2} \\ b_{3,j-C3} \end{bmatrix},$$

где

$$b_{i,j} = \mathrm{S}[a_{i,j}, \gamma],$$

γ – байт раундового ключа, используемый для формирования таблицы замен.

В этом выражении индексы столбцов должны быть взяты по модулю Nb. С помощью замены, приведенные выше выражения, могут быть скомбинированы в:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = [\mathrm{M}] \begin{bmatrix} \mathrm{S}[a_{0,j-C0}, \gamma] \\ \mathrm{S}[a_{1,j-C1}, \gamma] \\ \mathrm{S}[a_{2,j-C2}, \gamma] \\ \mathrm{S}[a_{3,j-C3}, \gamma] \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}.$$

Как показано выше, матрица

$$\mathrm{M}' = \begin{pmatrix} s & s^2 & s^3 & s^4 \\ s^2 & s^4 & s^6 & s^8 \\ s^3 & s^6 & s^9 & s^{12} \\ s^4 & s^8 & s^{12} & s^{16} \end{pmatrix}$$

алгоритма ADE формируется с помощью одного байта раундового ключа *s*, посредством формирования подматрицы порождающей матрицы (256, 4, 253) кода Риа-Соломона над GF(2^8). Таким образом, матричное умножение может быть выражено как линейная комбинация векторов:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \end{bmatrix} = \mathrm{S}[a_{0,j-C0}, \gamma] \begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix} \oplus \mathrm{S}[a_{1,j-C1}, \gamma] \begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \end{bmatrix} \oplus \mathrm{S}[a_{2,j-C2}, \gamma] \begin{bmatrix} s^3 \\ s^6 \\ s^9 \\ s^{12} \end{bmatrix} \oplus \mathrm{S}[a_{3,j-C3}, \gamma] \begin{bmatrix} s^4 \\ s^8 \\ s^{12} \\ s^{16} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix}.$$

Множители $\mathrm{S}[a_{i,j}, \gamma, s]$ четырех векторов получены составлением выборочной таблицы на входах байт $a_{i,j}$ и байта раундового ключа γ в таблице S -блока.

Определим таблицы T_0, T_1, T_2, T_3:

$$\mathrm{T}_0[a, \gamma, s] = \mathrm{S}[a_{0,j-C0}, \gamma] \begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix}, \quad \mathrm{T}_1[a, \gamma, s] = \mathrm{S}[a_{0,j-C0}, \gamma] \begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \end{bmatrix},$$

$$\mathrm{T}_2[a,\gamma,s]=\mathrm{S}[a_{0,j-C0},\gamma]\begin{bmatrix}s^3\\ s^6\\ s^9\\ s^{12}\end{bmatrix},\ \mathrm{T}_3[a,\gamma,s]=\mathrm{S}[a_{0,j-C0},\gamma]\begin{bmatrix}s^4\\ s^8\\ s^{12}\\ s^{16}\end{bmatrix}.$$

Для каждого раунда алгоритма ADE используются свои значения γ, s, $C0$, $C1$, $C2$, $C3$. Следовательно, T_0, T_1, T_2, T_3 – это 4 таблицы (для каждого раунда свои) с 256-ю вводами 4-байтовых слов, занимающие до 4 Кбайт памяти.

Используя эти таблицы, раундовое преобразование может быть выражено как:

$$e_j=\mathrm{T}_0[a_{0,j-C0},\gamma,s]\oplus\mathrm{T}_1[a_{1,j-C1},\gamma,s]\oplus\mathrm{T}_2[a_{2,j-C2},\gamma,s]\oplus\mathrm{T}_3[a_{3,j-C3},\gamma,s]\oplus k_j.$$

Следовательно, реализация выборочных таблиц с 4 КБайтами памяти требует только 4 выборочные таблицы и 4 сложения по модулю 2 на столбец каждого раунда.

Размер кода (важен для минипрограмм) может быть сделан небольшим посредством включения кода для генерации таблиц взамен самих таблиц.

Реализация алгоритма ADE с блоком преобразования $\mathrm{MixColumn}_i$, для $i>1$ на процессорах разрядностью ≤ 32 - не рекомендуется.

3.2. 64-битный процессор

Различные шаги раундового преобразования могут быть скомбинированы для обеспечения быстрой реализации на процессорах с длиной слова 64 или выше. Рассмотрим случай $\mathrm{MixColumn}_1$.

Раундовое преобразование с $MixColumn_1$

Выразим два столбца выхода раунда e в величинах байт входа раунда a. В этой секции, как и в секции 5.2.1, через $a_{i,j}$ обозначим байт в строке i и столбце j, через a_j обозначим j-ый столбец Состояния a. Рассмотрим два последовательно следующих столбца Состояния и скомбинируем преобразования, получим:

$$\begin{bmatrix}e_{0,j}\\ e_{1,j}\\ e_{2,j}\\ e_{3,j}\\ e_{0,j+1}\\ e_{1,j+1}\\ e_{2,j+1}\\ e_{3,j+1}\end{bmatrix}=\begin{bmatrix}[\mathrm{M}]\begin{bmatrix}\mathrm{S}[a_{0,j-C0},\gamma]\\ \mathrm{S}[a_{1,j-C1},\gamma]\\ \mathrm{S}[a_{2j-C2},\gamma]\\ \mathrm{S}[a_{3,j-C3},\gamma]\end{bmatrix}\\ [\mathrm{M}]\begin{bmatrix}\mathrm{S}[a_{0,j+1-C0},\gamma]\\ \mathrm{S}[a_{1,j+1-C1},\gamma]\\ \mathrm{S}[a_{2,j+1-C2},\gamma]\\ \mathrm{S}[a_{3,j+1-C3},\gamma]\end{bmatrix}\end{bmatrix}\oplus\begin{bmatrix}k_{0,j}\\ k_{1,j}\\ k_{2,j}\\ k_{3,j}\\ k_{0,j+1}\\ k_{1,j+1}\\ k_{2,j+1}\\ k_{3,j+1}\end{bmatrix},\ [\mathrm{M}]=\begin{bmatrix}s & s^2 & s^3 & s^4\\ s^2 & s^4 & s^6 & s^8\\ s^3 & s^6 & s^9 & s^{12}\\ s^4 & s^8 & s^{12} & s^{16}\end{bmatrix}.$$

Матричное умножение можно выразить как линейную комбинацию векторов:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \\ e_{0,j+1} \\ e_{1,j+1} \\ e_{2,j+1} \\ e_{3,j+1} \end{bmatrix} = \begin{bmatrix} \mathrm{S}\left[a_{0,j-C0},\gamma\right]\begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix} \oplus \mathrm{S}\left[a_{1,j-C1},\gamma\right]\begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \end{bmatrix} \oplus \mathrm{S}\left[a_{2,j-C2},\gamma\right]\begin{bmatrix} s^3 \\ s^6 \\ s^9 \\ s^{12} \end{bmatrix} \oplus \mathrm{S}\left[a_{3,j-C3},\gamma\right]\begin{bmatrix} s^4 \\ s^8 \\ s^{12} \\ s^{16} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \\ \mathrm{S}\left[a_{0,j+1-C0},\gamma\right]\begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \end{bmatrix} \oplus \mathrm{S}\left[a_{1,j+1-C1},\gamma\right]\begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \end{bmatrix} \oplus \mathrm{S}\left[a_{2,j+1-C2},\gamma\right]\begin{bmatrix} s^3 \\ s^6 \\ s^9 \\ s^{12} \end{bmatrix} \oplus \mathrm{S}\left[a_{3,j+1-C3},\gamma\right]\begin{bmatrix} s^4 \\ s^8 \\ s^{12} \\ s^{16} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j+1} \\ k_{1,j+1} \\ k_{2,j+1} \\ k_{3,j+1} \end{bmatrix} \end{bmatrix}$$

или, во введенных выше обозначениях:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \\ e_{0,j+1} \\ e_{1,j+1} \\ e_{2,j+1} \\ e_{3,j+1} \end{bmatrix} = \begin{bmatrix} \mathrm{T}_0\left[a_{0,j-C0},\gamma,s\right] \oplus \mathrm{T}_1\left[a_{1,j-C1},\gamma,s\right] \oplus \mathrm{T}_2\left[a_{2,j-C2},\gamma,s\right] \oplus \mathrm{T}_2\left[a_{3,j-C3},\gamma,s\right] \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \end{bmatrix} \\ \mathrm{T}_0\left[a_{0,j+1-C0},\gamma,s\right] \oplus \mathrm{T}_1\left[a_{1,j+1-C1},\gamma,s\right] \oplus \mathrm{T}_2\left[a_{2,j+1-C2},\gamma,s\right] \oplus \mathrm{T}_2\left[a_{3,j+1-C3},\gamma,s\right] \oplus \begin{bmatrix} k_{0,j+1} \\ k_{1,j+1} \\ k_{2,j+1} \\ k_{3,j+1} \end{bmatrix} \end{bmatrix}$$

Множители $\mathrm{S}\left[a_{i,j},\gamma,s\right]$ четырех векторов получены составлением выборочной таблицы на входах байт $a_{i,j}$ и байта раундового ключа γ в таблице S -блока.

Определим таблицы T_0, T_1, T_2, T_3:

$$\mathrm{T}_0\left[a,\gamma,s\right] = \begin{bmatrix} \mathrm{T}_0\left[a_{0,j-C0},\gamma,s\right] \\ \mathrm{T}_0\left[a_{0,j+1-C0},\gamma,s\right] \end{bmatrix},\ \mathrm{T}_1\left[a,\gamma,s\right] = \begin{bmatrix} \mathrm{T}_1\left[a_{1,j-C1},\gamma,s\right] \\ \mathrm{T}_1\left[a_{1,j+1-C1},\gamma,s\right] \end{bmatrix},$$

$$\mathrm{T}_2\left[a,\gamma,s\right] = \begin{bmatrix} \mathrm{T}_2\left[a_{2,j-C2},\gamma,s\right] \\ \mathrm{T}_2\left[a_{2,j+1-C2},\gamma,s\right] \end{bmatrix},\ \mathrm{T}_3\left[a,\gamma,s\right] = \begin{bmatrix} \mathrm{T}_3\left[a_{3,j-C3},\gamma,s\right] \\ \mathrm{T}_3\left[a_{3,j+1-C3},\gamma,s\right] \end{bmatrix}.$$

T_0, T_1, T_2, T_3 – это 4 таблицы (для каждого раунда свои) с 65536-ю входами 8-ми байтовых слов, занимающие до 4 МБайт памяти.

Используя эти таблицы, раундовое преобразование может быть выражено как:

$$\begin{bmatrix} e_j \\ e_{j+1} \end{bmatrix} = \mathrm{T}_0'^{*}\left[a_{0,j-C0},a_{0,j+1-C0},\gamma,s\right] \oplus \mathrm{T}_1^{*}\left[a_{1,j-C1},a_{1,j+1-C1},\gamma,s\right] \oplus$$

$$\oplus \mathrm{T}_2^{*}\left[a_{2,j-C2},a_{2,j+1-C2},\gamma,s\right] \oplus \mathrm{T}_2^{*}\left[a_{3,j-C3},a_{3,j+1-C3},\gamma,s\right] \oplus \begin{bmatrix} e_j \\ e_{j+1} \end{bmatrix}.$$

Следовательно, реализация выборочных таблиц с 4 Мбайтами таблиц требует только 4 выборочные таблицы и 4 сложения по модулю 2 на каждые 2 столбца каждого раунда. Очевидно, что предлагаемая реализация на 64-х разрядных процессорах по быстродействию в два раза превосходит реализацию на 32-х разрядных процессорах, в том числе ***превосходит***

в два раза по быстродействию реализацию алгоритма AES на 32-х разрядных процессорах.

Указанное повышение быстродействия сопряжено с существенным повышением объема необходимой памяти для реализации выборочных таблиц (с 4КБайт до 4 МБайт).

Рассмотренный подход по реализации алгоритма ADE с блоком $\mathrm{MixColumn}_1$ на 64-х разрядном процессоре можно расширить на большую разрядность, кратную 32. Фактически, при переходе к разрядности 32 наблюдается увеличение быстродействия алгоритма ADE в $2i$ раз. Платой за это является естественное увеличение (в 2^{10} раз) объема памяти для хранения выборочных таблиц.

Таким образом, быстродействие алгоритма ADE с динамически изменяемыми криптопримитивами **на 64-х разрядной платформе в 2 раза превосходит быстродействие алгоритмов AES/ADE на 32-х разрядных процессорах**. Для эффективной реализации алгоритма ADE требуется 4 Мбайт памяти для каждого из Nr раундов.

Рассмотрим случай использования $\mathrm{MixColumn}_2$.

Раундовое преобразование с MixColumn₂

Выразим два столбца выхода раунда e в величинах байт входа раунда a. Пусть, как и прежде, $a_{i,j}$ обозначает байт в строке i и столбце j, a_j обозначает столбец j Состояния a. Рассмотрим два последовательно следующих столбца Состояния и скомбинируем преобразования, получим:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \\ e_{0,j+1} \\ e_{1,j+1} \\ e_{2,j+1} \\ e_{3,j+1} \end{bmatrix} = [\mathrm{M}] \begin{bmatrix} \mathrm{S}[a_{0,j-C0},\gamma] \\ \mathrm{S}[a_{1,j-C1},\gamma] \\ \mathrm{S}[a_{2j-C2},\gamma] \\ \mathrm{S}[a_{3,j-C3},\gamma] \\ \mathrm{S}[a_{0,j+1-C0},\gamma] \\ \mathrm{S}[a_{1,j+1-C1},\gamma] \\ \mathrm{S}[a_{2,j+1-C2},\gamma] \\ \mathrm{S}[a_{3,j+1-C3},\gamma] \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \\ k_{0,j+1} \\ k_{1,j+1} \\ k_{2,j+1} \\ k_{3,j+1} \end{bmatrix}, \quad \mathrm{M}' = \begin{pmatrix} s & s^2 & s^3 & s^4 & s^5 & s^6 & s^7 & s^8 \\ s^2 & s^4 & s^6 & s^8 & s^{10} & s^{12} & s^{14} & s^{16} \\ s^3 & s^6 & s^9 & s^{12} & s^{15} & s^{18} & s^{21} & s^{24} \\ s^4 & s^8 & s^{12} & s^{16} & s^{20} & s^{24} & s^{28} & s^{32} \\ s^5 & s^{10} & s^{15} & s^{20} & s^{25} & s^{30} & s^{35} & s^{40} \\ s^6 & s^{12} & s^{18} & s^{24} & s^{30} & s^{36} & s^{42} & s^{48} \\ s^7 & s^{14} & s^{21} & s^{28} & s^{35} & s^{42} & s^{49} & s^{56} \\ s^8 & s^{16} & s^{24} & s^{32} & s^{40} & s^{48} & s^{56} & s^{64} \end{pmatrix}.$$

Матричное умножение можно выразить как линейную комбинацию векторов:

$$\begin{bmatrix} e_{0,j} \\ e_{1,j} \\ e_{2,j} \\ e_{3,j} \\ e_{0,j+1} \\ e_{1,j+1} \\ e_{2,j+1} \\ e_{3,j+1} \end{bmatrix} = \mathrm{S}\left[a_{0,j-C0},\gamma\right]\begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \\ s^5 \\ s^6 \\ s^7 \\ s^8 \end{bmatrix} \oplus \mathrm{S}\left[a_{1,j-C1},\gamma\right]\begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \\ s^{10} \\ s^{12} \\ s^{14} \\ s^{16} \end{bmatrix} \oplus \mathrm{S}\left[a_{2,j-C2},\gamma\right]\begin{bmatrix} s^3 \\ s^6 \\ s^9 \\ s^{12} \\ s^{15} \\ s^{18} \\ s^{21} \\ s^{24} \end{bmatrix} \oplus \mathrm{S}\left[a_{3,j-C3},\gamma\right]\begin{bmatrix} s^4 \\ s^8 \\ s^{12} \\ s^{16} \\ s^{20} \\ s^{24} \\ s^{28} \\ s^{32} \end{bmatrix} \oplus$$

$$\oplus \mathrm{S}\left[a_{0,j+1-C0},\gamma\right]\begin{bmatrix} s^5 \\ s^{10} \\ s^{15} \\ s^{20} \\ s^{25} \\ s^{30} \\ s^{35} \\ s^{40} \end{bmatrix} \oplus \mathrm{S}\left[a_{1,j+1-C1},\gamma\right]\begin{bmatrix} s^6 \\ s^{12} \\ s^{18} \\ s^{24} \\ s^{30} \\ s^{36} \\ s^{42} \\ s^{48} \end{bmatrix} \oplus \mathrm{S}\left[a_{2,j+1-C2},\gamma\right]\begin{bmatrix} s^7 \\ s^{14} \\ s^{21} \\ s^{28} \\ s^{35} \\ s^{42} \\ s^{49} \\ s^{56} \end{bmatrix} \oplus \mathrm{S}\left[a_{3,j+1-C3},\gamma\right]\begin{bmatrix} s^8 \\ s^{16} \\ s^{24} \\ s^{32} \\ s^{40} \\ s^{48} \\ s^{56} \\ s^{64} \end{bmatrix} \oplus \begin{bmatrix} k_{0,j} \\ k_{1,j} \\ k_{2,j} \\ k_{3,j} \\ k_{0,j+1} \\ k_{1,j+1} \\ k_{2,j+1} \\ k_{3,j+1} \end{bmatrix}.$$

Определим таблицы T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7:

$$\mathrm{T}_0\left[a,\gamma,s\right] = \mathrm{S}\left[a_{0,j-C0},\gamma\right]\begin{bmatrix} s \\ s^2 \\ s^3 \\ s^4 \\ s^5 \\ s^6 \\ s^7 \\ s^8 \end{bmatrix}, \quad \mathrm{T}_1\left[a,\gamma,s\right] = \mathrm{S}\left[a_{1,j-C1},\gamma\right]\begin{bmatrix} s^2 \\ s^4 \\ s^6 \\ s^8 \\ s^{10} \\ s^{12} \\ s^{14} \\ s^{16} \end{bmatrix},$$

$$\mathrm{T}_2\left[a,\gamma,s\right] = \mathrm{S}\left[a_{2,j-C2},\gamma\right]\begin{bmatrix} s^3 \\ s^6 \\ s^9 \\ s^{12} \\ s^{15} \\ s^{18} \\ s^{21} \\ s^{24} \end{bmatrix}, \quad \mathrm{T}_3\left[a,\gamma,s\right] = \mathrm{S}\left[a_{3,j-C3},\gamma\right]\begin{bmatrix} s^4 \\ s^8 \\ s^{12} \\ s^{16} \\ s^{20} \\ s^{24} \\ s^{28} \\ s^{32} \end{bmatrix},$$

$$\mathrm{T}_4[a,\gamma,s]=\mathrm{S}[a_{0,j+1-C0},\gamma]\begin{bmatrix}s^5\\s^{10}\\s^{15}\\s^{20}\\s^{25}\\s^{30}\\s^{35}\\s^{40}\end{bmatrix},\ \mathrm{T}_5[a,\gamma,s]=\mathrm{S}[a_{1,j+1-C1},\gamma]\begin{bmatrix}s^6\\s^{12}\\s^{18}\\s^{24}\\s^{30}\\s^{36}\\s^{42}\\s^{48}\end{bmatrix},$$

$$\mathrm{T}_6[a,\gamma,s]=\mathrm{S}[a_{2,j+1-C2},\gamma]\begin{bmatrix}s^7\\s^{14}\\s^{21}\\s^{28}\\s^{35}\\s^{42}\\s^{49}\\s^{56}\end{bmatrix},\ \mathrm{T}_7[a,\gamma,s]=\mathrm{S}[a_{3,j+1-C3},\gamma]\begin{bmatrix}s^8\\s^{16}\\s^{24}\\s^{32}\\s^{40}\\s^{48}\\s^{56}\\s^{64}\end{bmatrix}.$$

Для каждого раунда алгоритма ADE используются свои значения γ, s, $C0$, $C1$, $C2$, $C3$. Следовательно, T_0, T_1, T_2, T_3, T_4, T_5, T_6, T_7 – это 4 таблицы (для каждого раунда свои) с 256-ю вводами 8-ти байтовых слов, занимающие до 16 КБайт памяти.

Используя эти таблицы, раундовое преобразование может быть выражено как:

$$e_j=\mathrm{T}_0[a_{0,j-C0},\gamma,s]\oplus\mathrm{T}_1[a_{1,j-C1},\gamma,s]\oplus\mathrm{T}_2[a_{2,j-C2},\gamma,s]\oplus\mathrm{T}_3[a_{3,j-C3},\gamma,s]\oplus$$
$$\oplus\mathrm{T}_4[a_{0,j+1-C0},\gamma,s]\oplus\mathrm{T}_5[a_{1,j+1-C1},\gamma,s]\oplus\mathrm{T}_6[a_{2,j+1-C2},\gamma,s]\oplus\mathrm{T}_7[a_{3,j+1-C3},\gamma,s]\oplus k_j.$$

Следовательно, реализация выборочных таблиц с 16 КБайтами памяти требует только 8 выборочных таблиц и 8 сложений по модулю 2 на каждые 2 столбца каждого раунда.

Реализация алгоритма ADE с блоком преобразования $\mathrm{MixColumn_i}$ для $i>2$ на процессорах разрядностью ≤ 64 не рекомендуется.

Реализация на процессорах с разрядностью $32i$, $i>2$ может быть эффективно реализована с использованием блока $\mathrm{MixColumn_{\le i}}$. Комбинирование раундовых преобразований может быть получено индуктивно по рассмотренной выше схеме.

4. Выводы

Таким образом, проведенные исследования показали, что алгоритм ADE допускает эффективную реализацию на 32-х и 64-х разрядных процессорах. По быстродействию он сопоставим с алгоритмом AES, а для соответствующих параметров превосходит его. Кроме того, алгоритм ADE использует все преимущества алгоритма AES. Это относится и к возможности распараллеливания вычислений. Как показано выше, все 4 компонентные преобразования в раунде действуют параллельным образом на байты, строки или столбцы Состояния. В реализации с выборочными таблицами эти вычисления могут быть выполнены параллельно. ***Перспективным направлением дальнейших исследований*** является оценка криптографических свойств алгоритма, исследование его статистической безопасности.

Список литературы

1. N. Courtois and J. Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. Preprint in Cryptology ePrint Archive: http://eprint.iacr.org/2002/044/.
2. Nicolas Courtois and Josef Pieprzyk: Cryptanalysis of Block Ciphers with Overdefined Systems of Equations, Asiacrypt 2002, LNCS 2501, pp.267-287, Springer. Available at http://eprint.iacr.org/2002/044/.
3. N. Courtois and J. Pieprzyk. Cryptanalysis of Block Ciphers with Overdefined Systems of Equations. In Advances in Cryptology - ASIACRYPT 2002, volume 2501 of Lecture Notes in Computer Science, pages pp. 267 – 287. Springer Verlag Heidelberg, 2002.
4. S. Murphy and M. J. Robshaw. Comments on the Security of the AES and the XSL Technique, 2002.
5. S. Murphy and M. J. Robshaw. Essential Algebraic Structure within the AES. In Advances in Cryptology - CRYPTO 2002, volume 2442 of Lecture Notes in Computer Science, pages pp. 1 – 16. Springer Verlag Heidelberg, 2002.
6. Кузнецов А.А., Сергиенко Р.В., Наумко А.А. Симметричный криптографический алгоритм ADE (Algorithm of Dynamic Encryption) Радиотехника: Всеукр. межвед. науч.-техн. сб. – Харьков: ХТУРЭ. – 2007. – Вып. XXX. – С. XX-XX.
7. National Institute of Standards and Technology, “FIPS-197: Advanced Encryption Standard.” Nov. 2001. Available at http://csrc.nist.gov/publications/fips/ fips197/fips-197.pdf

Поступила в редколлегию

Харьковский Национальный университет радиоэлектроники
Военный институт при Сумском государственном университете