АСИММЕТРИЧНЫЙ SPN-ШИФР НА БАЗЕ WHITE-BOX-КРИПТОГРАФИИ И ХАОТИЧЕСКИХ ОТОБРАЖЕНИЙ

к.т.н, Щелкунов Д.А.

КФ МГТУ имени Н.Э. Баумана

White-box-криптография

- ✓Набор техник, позволяющих преобразовать симметричный блочный шифр в асимметричный посредством сокрытия ключа в реализации алгоритма шифрования
- ✓Предназначена для создания быстрых асимметричных шифров, позволяющих и шифровать, и подписывать сообщения
- ✓В случае стойкой и быстрой реализации позволит существенно упростить и сделать более безопасными протоколы обмена зашифрованной информацией (в частности, отпадёт необходимость использовать алгоритм Диффи-Хеллмана)
- ✓Быстрый безопасный обмен между 3-мя и более абонентами с подтверждением авторства сообщений

Предыдущие исследования

- ✓Chow S., Eisen P., Johnson H., Van Oorschot P.C. (2003), **White-Box Cryptography and an AES Implementation**. In: Nyberg K., Heys H. (eds) Selected Areas in Cryptography. SAC 2002. Lecture Notes in Computer Science, vol 2595. Springer, Berlin, Heidelberg
- ✓Olivier Billet and Henri Gilbert. A Traceable Block Cipher. In Advances in Cryptology - ASIACRYPT 2003, volume 2894 of Lecture Notes in Computer Science, pages 331-346. Springer-Verlag, 2003
- ✓Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a White-Box AES Implementation. In Proceedings of the 11th International Workshop on Selected Areas in Cryptography (SAC 2004), volume 3357 of Lecture Notes in Computer Science, pages 227–240. Springer-Verlag, 2004.
- ✓Brecht Wyseur, White-box cryptography, PhD thesis, March 2009
- ✓Dmitry Schelkunov, White-Box Cryptography and SPN ciphers. LRC method, Cryptology ePrint Archive: Report 2010/419
- ✓Brecht Wyseur, White-box cryptography: hiding keys in software, MISC magazine, April 2012
- ✓Joppe W. Bos and Charles Hubain and Wil Michiels and Philippe Teuwen, Differential Computation Analysis: Hiding your White-Box Designs is Not Enough, Cryptology ePrint Archive: Report 2015/753

Атаки на white-box-реализации

Практически во всех white-box-реализациях известны линейная и нелинейная части исходного алгоритма для каждого из раундов. Необходимо «отделить» white-box-преобразования

- ✓ Дифференциальный криптоанализ (в том числе fault injection)
- ✓ Алгебраический криптоанализ
- ✓ **Извлечение нелинейной части** (Olivier Billet, Henri Gilbert, and Charaf Ech-Chatbi. Cryptanalysis of a White-Box AES Implementation)

Метод сокрытия линейной зависимости

$$\begin{cases} x_1 = ((a \cdot b)(\mathrm{mod}\ p_1) \cdot c)(\mathrm{mod}\ p_2) \\ x_2 = (a \cdot (b \cdot c)(\mathrm{mod}\ p_2))(\mathrm{mod}\ p_1) \end{cases} \quad (1)$$

p_1, p_2 – неприводимые полиномы n-й степени, а a, b, c, x_1, x_2 - полиномы степеней, меньших n.

$$x_1 \neq x_2$$

Метод сокрытия линейной зависимости

$$\begin{cases} y_1(x) = (s(x) \cdot a(\mathrm{mod}\ p_1)) \cdot b(\mathrm{mod}\ p_2) \\ y_2(x) = (s(x) \cdot c(\mathrm{mod}\ p_1)) \cdot d(\mathrm{mod}\ p_3) \end{cases} \quad (2)$$

$p_1, p_2, p_3 - $ *неприводимые попарно неравные полиномы над* $GF(\alpha)$

$x, a, b, c, d -$ *произвольные полиномы над* $GF(\alpha)$

$p_1, p_2, p_3, x, a, b, c, d$ *неизвестны*

$y_1(x), y_2(x)$ *заданы таблично*

Метод сокрытия линейной зависимости

$$\begin{cases} y_1(x) = (s(x) \cdot a(\mathrm{mod}\ p_1)) \cdot b(\mathrm{mod}\ p_2) \\ y_2(x) = (s(x) \cdot c(\mathrm{mod}\ p_1)) \cdot d(\mathrm{mod}\ p_3) \end{cases} \quad (2)$$

Задача : найти линейную зависимость между

$s(x) \cdot a(\mathrm{mod}\ p_1)$ *и* $s(x) \cdot c(\mathrm{mod}\ p_1)$

Метод сокрытия линейной зависимости

$$\begin{cases} y_1(x) = (s(x)\cdot a - p_1\cdot q_1)\cdot b (\mathrm{mod}\ p_2) \\ y_2(x) = (s(x)\cdot c - p_1\cdot q_1')\cdot d (\mathrm{mod}\ p_3) \end{cases} \quad (3)$$

$$\begin{cases} y_1(x) = s(x)\cdot a\cdot b - p_1\cdot q_1\cdot b - p_2\cdot q_2 \\ y_2(x) = s(x)\cdot c\cdot d - p_1\cdot q_1'\cdot d - p_3\cdot q_3 \end{cases} \quad (4)$$

$$q_1 = \left\lfloor \frac{s(x)\cdot a}{p_1} \right\rfloor; q_1' = \left\lfloor \frac{s(x)\cdot c}{p_1} \right\rfloor; q_2 = \left\lfloor \frac{s(x)\cdot a\cdot b - p_1\cdot q_1\cdot b}{p_2} \right\rfloor; q_3 = \left\lfloor \frac{s(x)\cdot c\cdot d - p_1\cdot q_1'\cdot d}{p_3} \right\rfloor$$

Задача : по данным $y_1(x)$ и $y_2(x)$ необходимо найти a и c

RLWE?

Метод сокрытия линейной зависимости

Усложним формулу (2)

$$\begin{cases} y_1(x) = (\ldots(s(x) \cdot a(\text{mod } p_1)) \cdot b^{(0)}(\text{mod } p_2^{(0)})\ldots) \cdot b^{(k)}(\text{mod } p_u^{(k)}) \\ y_2(x) = (\ldots(s(x) \cdot c(\text{mod } p_1)) \cdot d^{(0)}(\text{mod } p_3^{(0)})\ldots) \cdot d^{(k)}(\text{mod } p_v^{(k)}) \end{cases} \quad (5)$$

$$p_i^{(\alpha)} \neq p_j^{(\alpha)}$$

$$\text{Сложность} : \min\left(2^{2n(k+1)}, \left(2^n!\right)^2\right)$$

Теория хаоса в криптографии

- ✓Goce Jakimoski and Ljupˇco Kocarev, Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps. IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—I: FUNDAMENTAL THEORY AND APPLICATIONS, VOL. 48, NO. 2, FEBRUARY 2001
- ✓Asim, M., Jeoti, V.: Efficient and simple method for designing chaotic S-boxes. ETRI Journal 30(1), 170–172 (2008)
- ✓Mona Dara and Kooroush Manochehri, A Novel Method for Designing S-Boxes Based on Chaotic Logistic Maps Using Cipher Key. World Applied Sciences Journal 28 (12): 2003-2009, 2013
- ✓Christopher A. Wood, Chaos-Based Symmetric Key Cryptosystems
- ✓Dragan Lambić and Miodrag Živković, COMPARISON OF RANDOM S-BOX GENERATION METHODS. PUBLICATIONS DE L'INSTITUT MATHÉMATIQUE Nouvelle série, tome 93 (107) (2013)

Создание S-box-ов с помощью хаотических отображений

- ✓Хорошие криптографические свойства по результатам целого ряда исследований
- ✓Простые алгоритмы
- ✓Случайность отображений в сочетании с хорошими криптографическими свойствами повышает уровень безопасности

Коды с максимальной дистанцией. MDS-матрица

MDS-матрица (Maximal Distance Separable matrix) – проверочная матрица линейного блокового кода с максимальной дистанцией

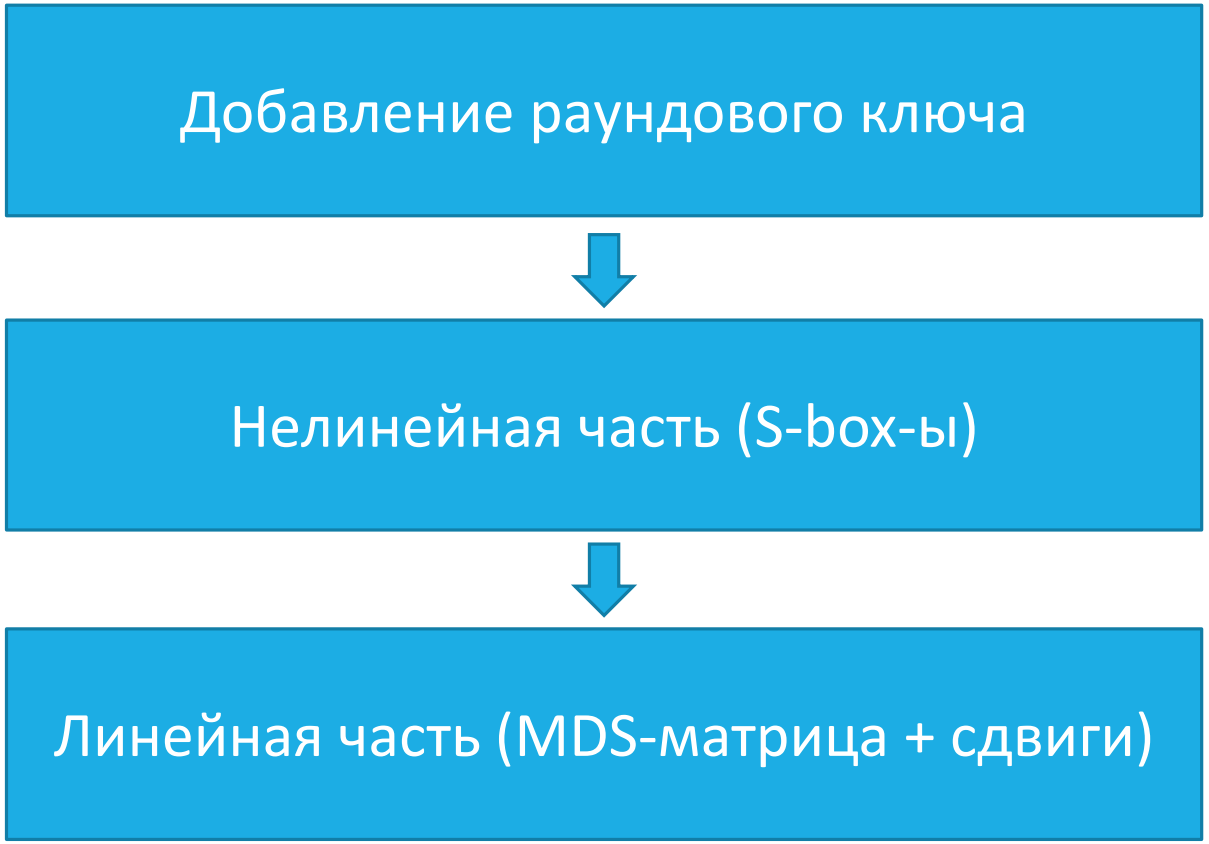
- Обеспечивает максимальное рассеивание за счёт своей структуры
- Используется при создании SPN-шифров в качестве линейных рассеивающих преобразований
- Интересные типы матриц:
  - Матрица Вандермонда
  - Инволютивная матрица (одна и та же MDS-матрица для шифрования и расшифрования)
  - **Матрица Коши**
  - Циркулярная матрица (как в Rijndael)

Матрица Коши

$$a_{ij} = \left(x_i + y_j\right)^{-1}; x_i + y_j \neq 0; 0 \leq i < m; 0 \leq j < n; x_i, y_j, a_{ij} \in GF\left(2^k\right)$$

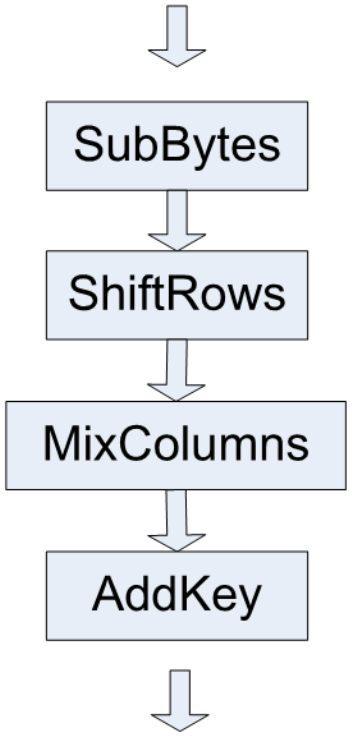
- ✓Является MDS-матрицей
- ✓Лёгкий алгоритм создания независимо от размерности
- ✓Свойство циркулярности не принципиально для white-box-реализации
- ✓Свойство инволютивности вредно для white-box-реализации
- ✓**Следовательно, выбираем матрицу Коши**

Раунд блочного SPN-шифра



Раунд SPN-шифра и T-box-ы (на примере AES-128)

Раунд AES-128



$$\begin{bmatrix} e_{0j} \\ e_{1j} \\ e_{2j} \\ e_{3j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \bullet \begin{bmatrix} S[a_{0j}] \\ S[a_{1j-1}] \\ S[a_{2j-2}] \\ S[a_{3j-3}] \end{bmatrix} \oplus \begin{bmatrix} k_{0j} \\ k_{1j} \\ k_{2j} \\ k_{3j} \end{bmatrix}$$

$$T_0[a] = \begin{bmatrix} S[a] \bullet 02 \\ S[a] \\ S[a] \\ S[a] \bullet 03 \end{bmatrix}; T_1[a] = \begin{bmatrix} S[a] \bullet 03 \\ S[a] \bullet 02 \\ S[a] \\ S[a] \end{bmatrix}$$

$$T_2[a] = \begin{bmatrix} S[a] \\ S[a] \bullet 03 \\ S[a] \bullet 02 \\ S[a] \end{bmatrix}; T_3[a] = \begin{bmatrix} S[a] \\ S[a] \\ S[a] \bullet 03 \\ S[a] \bullet 02 \end{bmatrix}$$

$$\begin{bmatrix} e_{0j} \\ e_{1j} \\ e_{2j} \\ e_{3j} \end{bmatrix} = T_0[a_{0j}] \oplus T_1[a_{1j-1}] \oplus T_2[a_{2j-2}] \oplus T_3[a_{3j-3}] \oplus \begin{bmatrix} k_{0j} \\ k_{1j} \\ k_{2j} \\ k_{3j} \end{bmatrix}$$

Хаотичный асимметричный white-box-шифр

- ✓Таблицы подстановок (8x8 бит) создаются хаотически для каждого из входных байтов каждого раунда
- ✓MDS-матрица (16x16 байт) создаётся хаотически (матрица Коши)
- ✓Для white-box-реализации используется реализация SPN-шифра с помощью T-box-ов
- ✓Линейная зависимость между элементами T-box-ов скрывается с помощью вышеописанного метода сокрытия линейной зависимости
- ✓Набор модифицированных таблиц (T-box-ов) – открытый ключ шифрования. По ним сложно восстановить закрытый ключ – набор таблиц для расшифрования

Раунд шифра

$$Y_j = \begin{bmatrix} y_j^{(0)} \\ y_j^{(1)} \\ \dots \\ y_j^{(15)} \end{bmatrix} = \begin{bmatrix} mix_j^{(0)}(t_j^{(0,0)}(s_j^{(0)}(y_{j-1}^{(0)})) \\ mix_j^{(1)}(t_j^{(1,0)}(s_j^{(0)}(y_{j-1}^{(0)})) \\ \dots \\ mix_j^{(15)}(t_j^{(15,0)}(s_j^{(0)}(y_{j-1}^{(0)})) \end{bmatrix} \oplus \begin{bmatrix} mix_j^{(0)}(t_j^{(0,1)}(s_j^{(1)}(y_{j-1}^{(1)})) \\ mix_j^{(1)}(t_j^{(1,1)}(s_j^{(1)}(y_{j-1}^{(1)})) \\ \dots \\ mix_j^{(15)}(t_j^{(15,1)}(s_j^{(1)}(y_{j-1}^{(1)})) \end{bmatrix} \oplus \dots \oplus \begin{bmatrix} mix_j^{(0)}(t_j^{(0,15)}(s_j^{(15)}(y_{j-1}^{(15)})) \\ mix_j^{(1)}(t_j^{(1,15)}(s_j^{(15)}(y_{j-1}^{(15)})) \\ \dots \\ mix_j^{(15)}(t_j^{(315,15)}(s_j^{(15)}(y_{j-1}^{(15)})) \end{bmatrix} \quad (6)$$

$y_{j\text{-}1}^{(i)}$-байт выходных данных из предыдущего раунда. Если $j = 0$, то $y_{j\text{-}1}^{(i)}$ - байт исходного сообщения. $s_j^{(k)}$ - это таблица подстановок, соответствующая каждому входному байту. $t_j^{(l,k)}$ - умножение на элемент строки соответствующей *MDS*-матрицы в $GF(2^8)$. $mix_j^{(k)}$ - запутывающие преобразования на базе метода сокрытия линейной зависимости

Маскировка линейной зависимости между элементами T-box-а

$$T'_i[a] = \begin{bmatrix} ((...(t_i^{(0)}(a)\cdot b_i^{(0,0)})(\mathrm{mod}\ p_i^{(0,0)})\cdot b_i^{(0,1)}(\mathrm{mod}\ p_i^{(0,1)})...)\cdot b_i^{(0,k_0)}(\mathrm{mod}\ p_i^{(0,k_0)})\oplus val_0 \\ ((...(t_i^{(1)}(a)\cdot b_i^{(1,0)})(\mathrm{mod}\ p_i^{(1,0)})\cdot b_i^{(1,1)}(\mathrm{mod}\ p_i^{(1,1)})...)\cdot b_i^{(1,k_1)}(\mathrm{mod}\ p_i^{(1,k_1)})\oplus val_1 \\ ((...(t_i^{(2)}(a)\cdot b_i^{(2,0)})(\mathrm{mod}\ p_i^{(2,0)})\cdot b_i^{(2,1)}(\mathrm{mod}\ p_i^{(2,1)})...)\cdot b_i^{(2,k_2)}(\mathrm{mod}\ p_i^{(2,k_2)})\oplus val_2 \\ ((...(t_i^{(3)}(a)\cdot b_i^{(3,0)})(\mathrm{mod}\ p_i^{(3,0)})\cdot b_i^{(3,1)}(\mathrm{mod}\ p_i^{(3,1)})...)\cdot b_i^{(3,k_3)}(\mathrm{mod}\ p_i^{(3,k_3)})\oplus val_3 \\ ..... \\ ((...(t_i^{(n)}(a)\cdot b_i^{(n,0)})(\mathrm{mod}\ p_i^{(n,0)})\cdot b_i^{(n,1)}(\mathrm{mod}\ p_i^{(n,1)})...)\cdot b_i^{(n,k_n)}(\mathrm{mod}\ p_i^{(n,k_n)})\oplus val_n \end{bmatrix} \quad (7)$$

$t_i^{(j)}$ - элемент *T-box*-а до применения запутывающих преобразований (*mix*), $b_i^{(j,u)}$ – случайно выбранный полином в $GF(2^h)$, $p_i^{(j,u)}$ – случайно выбранный неприводимый полином *h*-й степени над $GF(2)$

$$p_i^{(0,v)} \neq p_i^{(1,v)} \neq ... \neq p_i^{(n,v)}$$

EVHEN. Хаотичный асимметричный white-box-шифр

- ✓Назван в честь двух величайших математиков: **Эвариста Галуа и Анри Пуанкаре**
- ✓**Обладает скоростью симметричного шифра**
- ✓Позволяет **и шифровать, и подписывать**
- ✓Размер открытого ключа: 640 Кб
- ✓Низкие требования к вычислительным ресурсам: один раунд – 16 операций сложения по модулю 2 16-и байтовых чисел. Нужно всего 3 операции: **выборка из памяти, сложение по модулю 2 и запись в память**

Применение

- ✓ IoT
- ✓ DRM
- ✓ **Везде, где нужна быстрая и не требовательная к вычислительным ресурсам асимметричная криптография**

Спасибо за внимание

Исходный код реализации EVHEN: https://github.com/dmschelkunov/EVHEN

Блог автора: http://dschelkunov.blogspot.com

E-mail автора: d.schelkunov@gmail.com