Square

Криптоалгоритм *Square* [1] шифрует 128-битовые (16-байтовые) блоки открытых данных под управлением секретного ключа такого же размера.

Square – итеративный шифр, в котором раундовое преобразование (всего 8 раундов) является композицией четырех преобразований: θ, γ, π и σ. 16-байтовые блоки данных и раундовых подключей, участвующие в криптографическом преобразовании, представляются в виде 4×4-матриц

$$\mathrm{A} = (a_{ij}) = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix},$$

элементами которых являются байты, интерпретируемые в преобразованиях θ и γ как элементы конечного поля $\mathbb{F}_{256} \cong \mathbb{F}_2[x]/f(x)$, где $f(x) = x^8 + x^4 + x^3 + x + 1$.

Линейное преобразование θ определяется как

$$\theta(A) \equiv \{A := C \cdot A\},$$

где

$$C = \begin{pmatrix} 0\mathrm{x}02 & 0\mathrm{x}03 & 0\mathrm{x}01 & 0\mathrm{x}01 \\ 0\mathrm{x}01 & 0\mathrm{x}02 & 0\mathrm{x}03 & 0\mathrm{x}01 \\ 0\mathrm{x}01 & 0\mathrm{x}01 & 0\mathrm{x}02 & 0\mathrm{x}03 \\ 0\mathrm{x}03 & 0\mathrm{x}01 & 0\mathrm{x}01 & 0\mathrm{x}02 \end{pmatrix}.$$

Обратное преобразование имеет вид

$$\theta^{-1}(A) \equiv \{A := C^{-1} \cdot A\},$$

где

$$C^{-1} = \begin{pmatrix} 0\mathrm{x}0e & 0\mathrm{x}0b & 0\mathrm{x}0\mathrm{d} & 0\mathrm{x}09 \\ 0\mathrm{x}09 & 0\mathrm{x}0e & 0\mathrm{x}0b & 0\mathrm{x}0d \\ 0\mathrm{x}0d & 0\mathrm{x}09 & 0\mathrm{x}0e & 0\mathrm{x}0b \\ 0\mathrm{x}0b & 0\mathrm{x}0d & 0\mathrm{x}09 & 0\mathrm{x}0e \end{pmatrix}.$$

Нелинейное преобразование $\gamma(A)$ заключается в замене каждого элемента a_{ij} матрицы A на $S[a_{ij}]$. Используемая при этом подстановка S, заданная на множестве байтов (элементов поля $\mathbb{F}_{256}$), является композицией двух подстановок:

1) $x \longrightarrow x^{254}$, $x \in \mathbb{F}_{256}$ (отметим, что $x^{254} = x^{-1}$, если $x \neq 0$)
2) $x \longrightarrow \varphi(x)$, $\varphi(x) = x \oplus rol_1 x \oplus rol_2 x \oplus rol_3 x \oplus rol_4 x \oplus \$63$,

где $rol_m x$ – циклический сдвиг битов байта x влево на m позиций (φ – аффинное преобразование пространства 8-битовых векторов $\mathbb{F}_2^8$).

Преобразование $\tau(A)$ определяется как транспонирование матрицы A:

$$A = \begin{pmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{pmatrix} \Longrightarrow \tau(A) = \begin{pmatrix} a_{00} & a_{10} & a_{20} & a_{30} \\ a_{01} & a_{11} & a_{21} & a_{31} \\ a_{02} & a_{12} & a_{22} & a_{32} \\ a_{03} & a_{13} & a_{23} & a_{33} \end{pmatrix}.$$

Преобразование $\sigma[k]$ с параметром k (k – раундовый подключ) определяется как

$$\sigma[k](A) \equiv \{A := A \oplus k\},$$

где $\oplus$ – побитовое сложение по модулю 2 соответствующих элементов матриц.

Раундовое преобразование $\rho[k]$ определяется как

$$\rho[k] = \sigma[k] \circ \tau \circ \gamma \circ \theta,$$

где $f \circ g(\mathrm{A}) \equiv f(g(\mathrm{A}))$.

---

[1] Авторы шифра: *Joan Daemen*, *Lars Knudsen* и *Vincent Rijmen* (Бельгия)

Шифрующее преобразование $Square[k_0, k_1,\ldots, k_8]$ под управлением раундовых подключей $k_0, k_1,\ldots, k_8$ определяется как

$\rho[k_8] \circ \rho[k_7] \circ \rho[k_6] \circ \rho[k_5] \circ \rho[k_4] \circ \rho[k_3] \circ \rho[k_2] \circ \rho[k_1] \circ \sigma[k_0] \circ \theta^{-1}$.

Другими словами, имеет место следующий

Алгоритм зашифрования *Square*

Вход: P − 16-байтовый блок открытых данных в виде 4×4-матрицы.

$C := P$;
$\theta^{-1}(C)$;
$\sigma[k_0](C)$;
$\boldsymbol{for}\ i := 1\ \boldsymbol{to}\ 8\ \boldsymbol{do}$ {
$\theta(C)$;
$\gamma(C)$;
$\tau(C)$;
$\sigma[k_i](C)$
}.

Выход: C − 16-байтовый блок шифртекста в виде 4×4-матрицы.

Используя следующие свойства введенных преобразований:

$\tau^{-1} = \tau$,
$\sigma^{-1}[k] = \sigma[k]$,
$\gamma^{-1} \circ \tau = \tau \circ \gamma^{-1}$,
$\sigma[k] \circ \theta^{-1} = \theta^{-1} \circ \sigma[\theta(k)]$,
$\theta \circ \sigma[k'] \circ \rho^{-1}[k''] = \rho'[\theta(k')] \circ \sigma[\theta(k'')] \circ \theta$,

где $\rho'[k] = \sigma[k] \circ \tau \circ \gamma^{-1} \circ \theta^{-1}$, нетрудно установить, что обратное (дешифрующее) преобразование

$Square^{-1}[k_0, k_1,\ldots, k_8] \equiv \theta \circ \sigma[k_0] \circ \rho^{-1}[k_1] \circ \ldots \circ \rho^{-1}[k_8]$

приводится к виду

$Square^{-1}[k_0, k_1,\ldots, k_8] = \rho'[\theta(k_0)] \circ \rho'[\theta(k_1)] \circ \ldots \circ \rho'[\theta(k_7)] \circ \sigma[\theta(k_8)] \circ \theta$.

Другими словами, алгоритм расшифрования получается из алгоритма зашифрования заменой преобразований θ и γ соответственно на θ^{-1} и γ^{-1}с одновременной заменой раундовых подключей зашифрования $k_0, k_1,\ldots, k_8$ на $\theta(k_8), \theta(k_7),\ldots, \theta(k_0)$.

Раундовые подключи зашифрования генерируются на основе секретного ключа K по итерационной схеме:

$k_0 := K$;
$\boldsymbol{for}\ t := 1\ \boldsymbol{to}\ 8\ \boldsymbol{do}\ \ k_t := \psi(k_{t-1})$.

Преобразование ψ определяется следующим образом. Записывая 16-байтовый раундовый подключ k_t в виде 4×4-матрицы

$$k_t = (k_{ij}^t),\ 0 \le i, j \le 3,$$

обозначим строки этой матрицы как $k_0^t, k_1^t, k_2^t, k_3^t$. Определим операцию левого циклического сдвига строки k_i^t как

$rotl(k_i^t) = (k_{i1}^t, k_{i2}^t, k_{i3}^t, k_{i0}^t)$.

Тогда

$k_0^{t+1} := k_0^t \oplus rotl(k_3^t) \oplus \mathrm{C}_t$;
$k_1^{t+1} := k_1^t \oplus k_0^{t+1}$;
$k_2^{t+1} := k_2^t \oplus k_1^{t+1}$;
$k_3^{t+1} := k_3^t \oplus k_2^{t+1}$,

где $C_t = (0\mathrm{x}00, 0\mathrm{x}00, 0\mathrm{x}00, (0\mathrm{x}02)^t)$, $0 \le t \le 8$, – раундовые константы, $(0\mathrm{x}02)^t$ − t-ая степень элемента $0\mathrm{x}02 \in \mathbb{F}_{256}$.