8 Атака «грубой силой»

8.1 Базовая атака

Пусть криптоаналитик располагает некоторым количеством пар (X_t, Y_t). Как правило, система

$$Y_t = F_K(X_t), \quad t = 1, \ldots, T,$$

имеет единственное решение относительно $K \in \mathcal{K}$ уже при небольшом T.

Криптоаналитик проводит атаку «*грубой силой*», просто выбирая последовательно кандидатов $\hat{K} \in \mathcal{K}$ и проверяя совпадения

$$Y_t \stackrel{?}{=} F_{\hat{K}}(X_t), \quad t = 1, \ldots, T. \qquad (\star)$$

При выполнении всех равенств принимается решение $K = \hat{K}$ и поиск прекращается.

Атака «Грубой силой»

Шифрматериал: $(X_1, Y_1), \ldots, (X_T, Y_T)$.

Выход: $\hat{K}$, решение $(\star)$, или $\perp$.

Шаги:

1. Для $\hat{K} \in \mathcal{K}$:

   (1) если $Y_t = F_{\hat{K}}(X_t)$ для всех $t = 1, 2, \ldots, T$, то возвратить $\hat{K}$.

2. Возвратить $\perp$.

Пусть в качестве $\hat{K}$ выбираются значения $K_1, \ldots, K_{|\mathcal{K}|}$, Пусть τ — число использованных в ходе атаки кандидатов $\hat{K}$. Тогда среднее значение

$$\mathsf{E}\tau = \sum_{t=1}^{|\mathcal{K}|} t\mathsf{P}\left\{K_t = K\right\} = \frac{|\mathcal{K}|(1 + |\mathcal{K}|)}{2|\mathcal{K}|} = \frac{1}{2}(|\mathcal{K}| + 1).$$

Для проведения атаки «грубой силой» используются специализированные микропроцессорные устройства или распределенные сетевые вычисления. Например, в 1999 году с использованием специализированного компьютера Deep Crack была проведена атака «грубой силой» на `DES` ($|\mathcal{K}| = 2^{56}$). Поиск ключа занял 22 часа и 15 минут. В сентябре 2002 г. методом «грубой силы» был найден ключ криптосистемы `RC5` ($|\mathcal{K}| = 2^{64}$). Поиск ключа занял около 4 лет и проводился на 331252 компьютерах сети Internet.

В рамках проекта ECRYPT (`http://ecrypt.eu.org`), выполняемого под эгидой Евросоюза, разработаны и периодически обновляются рекомендации по выбору длин ключей. Рекомендации 2012 года:

длина ключа	защита
32	защита от атак «реального времени» отдельных лиц
64	краткосрочная защита от атак малой организации (бюджет — 10 тыс. \$, FPGA)
72	краткосрочная защита от атак средней организации (бюджет — 300 тыс. \$, FPGA и/или ASIC)
80	краткосрочная защита от атак государственного агенства (бюджет — 300 млн. \$, ASIC)
112	среднесрочная защита (на 20 лет) от атак государственных агенств
128	долгосрочная защита (на 30 лет) от атак государственных агенств
256	защита на все обозримое будущее (даже с учетом создания квантовых компьютеров)

8.2 Простые соотношения

Определение 8.1. *Простым соотношением* для F называется тройка подстановок (g_1, g_2, h), $g_1, g_2 \in S(\{0,1\}^n)$, $h \in S(\mathcal{K})$, $h \neq id$, такая, что равенство

$$g_2 F_{h(K)} g_1(X) = F_K(X)$$

выполняется для всех $X \in \{0,1\}^n$, $K \in \mathcal{K}$. □

Наличие простых соотношений позволяет снизить сложность криптоанализа F методом «грубой силы». Пусть известны пары (X_2, Y_1), (X_2, Y_2) «открытый текст — шифртекст» преобразования F_K. Пусть $X_2 = g_1(X_1)$. Криптоаналитик выбирает ключ-кандидат $\hat{K}$ и выполняет зашифрование $\hat{Y} = F_{\hat{K}}(X_1)$. Если $\hat{K} = K$, то $\hat{Y} = Y_1$, а если $h(\hat{K}) = K$, то

$$Y_2 = F_K g_1(X_1) = F_{h(\hat{K})} g_1(X_1) = g_2^{-1} g_2 F_{h(\hat{K})} g_1(X_1) = g_2^{-1} F_{\hat{K}}(X_1) = g_2^{-1}(\hat{Y})$$

и $\hat{Y} = g_2(Y_2)$. Таким образом, выполнив одно зашифрование, криптоаналитик может проверить сразу два ключа: $\hat{K}$ и $h(\hat{K})$.

Пример 8.1 (DES). Расписание ключей DES обладает следующим свойством: если $\mathrm{KS}(K) = (\kappa_1, \ldots, \kappa_{16})$, то $\mathrm{KS}(\bar{K}) = (\bar{\kappa}_1, \ldots, \bar{\kappa}_{16})$, где черта обозначает инверсию символов двоичных слов (замена 0 на 1 и 1 на 0). При этом:

$$f_\kappa(x) = f_{\bar{\kappa}}(\bar{x}) \Rightarrow \Sigma_\kappa(X) = \bar{\Sigma}_{\bar{\kappa}}(\bar{X}) \Rightarrow F_K(X) = \bar{F}_{\bar{K}}(\bar{X}).$$

Мы получили простое соотношение, которое позволяет проверять одновременно ключи K и $\bar{K}$. □

Пример 8.2 (G). Имеется 255 простых соотношений для криптосистемы G. Действительно, для любого $\alpha \in \{0,1\}^8$ выполняется

$$f_{\kappa \oplus \alpha}(x \oplus \alpha) = f_\kappa(x), \quad \Sigma_{\kappa \oplus \alpha}(X \oplus (\alpha \parallel \alpha)) = \Sigma_\kappa(X) \oplus (\alpha \parallel \alpha)$$

и, следовательно,

$$F_{K \oplus (\alpha \| \alpha \| \alpha \| \alpha)}(X \oplus (\alpha \parallel \alpha)) = F_K(X) \oplus (\alpha \parallel \alpha),$$

Таким образом, при атаке на G криптоаналитик может проверять сразу 256 ключей. □

Упражнение 8.1 (ГОСТ 28147). В криптосистеме Фейстеля ГОСТ 28147-89 используется ключ $K = K_1 \parallel K_2 \parallel \ldots \parallel K_8$, $K_i \in \{0,1\}^{32}$. Расписание ключей определяется следующим образом:

$$\mathrm{KS}(K) = (K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_1, K_2, \ldots, K_8, K_8, \ldots, K_2, K_1).$$

Найти простое соотношение для ГОСТ. □

8.3 Баланс «время — память»

Пусть $(X, Y = F_K(X))$ — пара (открытый текст, шифртекст), по которой требуется найти ключ $K \in \mathcal{K}$. Будем считать, что $N = |\mathcal{K}| = |\{0,1\}^n|$ и уравнение $Y = F_{\hat{K}}(X)$ имеет малое число решений относительно $\hat{K}$ (одно из решений совпадает с K).

Описанная выше атака «грубой силой» проводится

а) за время $O(N)$ на памяти $O(1)$.

Возможна модификация атаки. На предварительном этапе Виктор для всевозможных кандидатов $\hat{K}$ вычисляет $\hat{Y} = F_{\hat{K}}(X)$ и помещает в ячейку памяти по адресу $\hat{Y}$ значение $\hat{K}$ (вообще говоря, ячейки могут содержать несколько значений). На оперативном этапе атаки Виктор получает Y, обращается к ячейке по адресу Y и определяет все $\hat{K}$, которые переводят X в Y. Новая атака проводится

б) за время $O(1)$ на памяти $O(N)$.

М. Хеллман (Hellman) предложил промежуточный между а) и б) вариант (*баланс «время — память»*, time-memory tradeoff) проведения атаки

в) за время $O(N^{2/3})$ на памяти $O(N^{2/3})$.

Суть метода Хеллмана состоит в следующем. Пусть r — некоторая простая функция $\{0,1\}^n \to \mathcal{K}$ (например, перестановка и выбор битов) и пусть

$$h_r \colon \mathcal{K} \to \mathcal{K}, \quad h_r(K) = r(F_K(X)).$$

Тогда задача определения ключа K по заданному Y сводится к решению уравнения $h_r(K) = r(Y)$, т. е. к обращению функции h_r.

На предварительном этапе Виктор определяет траекторию ключей K_0, $K_1 = h_r(K_0), \ldots, K_T = h_r(K_{T-1})$ и сохраняет начало K_0 и конец K_T траектории.

На оперативном этапе Виктор получает Y, вычисляет новую траекторию $Y_0 = r(Y)$, $Y_t = h_r(Y_{t-1})$, $t = 1, \ldots, T$, и на каждом шаге проверяет совпадение $Y_t \stackrel{?}{=} K_T$. Если $K = K_{T-\tau-1}$ (траектория ключей покрыла K), то

$$Y_0 = r(Y) = r(F_{K_{T-\tau-1}}(X)) = h_r(K_{T-\tau-1}) = K_{T-\tau}, Y_1 = h_r(Y_0) = h_r(K_{T-\tau}) = K_{T-\tau+1}, \ldots, Y_\tau = K_T,$$

т. е. искомое совпадение будет найдено.

Детали реализации. Хеллман предложил использовать R таблиц, в каждой из которых выбирать S различных значений K_0 и применять различные функции редукции r при определении h_r.

Время предварительных вычислений: $O(RST)$ (игнорируется). Память: $O(RS)$. Время оперативного этапа: $O(RT)$. Вероятность успеха (вероятность покрытия K траекториями ключей): максимальна, если $R = S = T = \lfloor N^{1/3} \rfloor$.