Об одном агоритме развертки ключа из пароля

Алексей Нестеренко

Национальный исследовательский университет «Высшая школа экономики»

Конференция «РусКрипто-2015»
18 марта 2015 г.

Постановка задачи

Функция выработки ключа:

$$F(p, iv) = k_1, k_2, \dots$$

$p \in V_m$ – пароль длины m бит, $0 < m < 1024$,
$iv \in V_r$ – случайный вектор (соль), $0 < r < 64$,
$k_1, k_2, \dots \in \mathbb{V}_n$ – последовательность ключей, $n = 256, 512$.

Другое определение функции выработки ключа:

$$F: \; V_m \times V_r \times \mathbb{Z} \to \mathbb{V}_n, \quad F(p, iv, s) = k_s$$

где $s \in \mathbb{Z}$ – номер ключа.

Требования к победителю конкурса:

1. F - криптографическая функция («случайный» выход, однонаправленность, отсутствие коллизий, статистическая независимость ключей, защита от чтения вперед/назад)
2. отсутствие методов существенного ускорения реализации функции F (по сравнению с «наивным»),
3. затруднение реализации параллельного перебора паролей на многопроцессорных CPU (существенные требования по памяти),
4. сравнимая эффективность реализации алгоритма как на CPU, так и на спецвычислителях (GPU, ПЛИС и т.п.)
5. защита от атак по побочным каналам утечки (включая временные атаки, атаки на энергопотребление и т.п.)

The Password Hashing Competition (февраль, 2015)

Финалисты:

Принципы построения

1. сжимающие отображения (Argon-v2, battcrypt, Lyra2-v3, Parallel-v1, Pufferfish-v1)
2. теория графов (Catena-v3)
3. теория автоматов (Pomelo-v2)
4. теория чисел (возведение в квадрат по модулю составного числа, генератор Блюма) (Makwa)

Наш алгоритм:

теория чисел (разложения иррациональных чисел)

Мотивация

Пусть α действительное число, $b > 1$ целое. Разложение в заданной системе счисления:

$$\alpha = \sum_{n=0}^{\infty} a_n b^{-n}, \quad 0 \leq a_n < b, \quad n = 1, 2, \ldots$$

Theorem (Гаусс)

Если α иррационально, то последовательность $a_1, a_2, \ldots$ – непериодична.

Пусть $\delta_1, \ldots, \delta_k \in \mathbb{Z}_b$ фиксированны,
Тогда число $\alpha \in \mathbb{R}$ *нормально*, если $\lim_{n\to\infty} \frac{N_n(\alpha,\delta_1,\ldots,\delta_k)}{n} = \frac{1}{b^k}$.

Theorem (Борель)

Нормальные числа образуют в $\mathbb{R}$ подмножество меры один.

Потребности реализации

Необходимо предъявить:

1. Выбор системы счисления

$$b = 2^8 \text{ (байты)},\ b = 2^{32} \text{ (слова)},\ b = 2^w,\ w \in \mathbb{N}.$$

2. Множество действительных чисел для разложения,
3. Эффективный алгоритм вычисления последовательности коэффициентов разложения числа.

Иррациональные числа I

Пусть $c_1, c_2, \ldots,$ – периодическая последовательность целых чисел, тогда число

$$\alpha = \sum_{k=0}^{\infty} \frac{c_k}{k!}$$

– иррационально.
(док-во аналогично иррациональности числа e).

Если взять $p = (c_1, \ldots, c_m)$ – пароль, $iv = (c_{m+1}, \ldots, c_{m+r})$ – случайный вектор (соль), то

$$F(p, iv, s) = \underbrace{a_s, a_{s+1}, \ldots}_{\text{ключ - } n \text{ бит}}$$

Иррациональные числа II

Пусть $d, s, m \in \mathbb{N}$, $u_1, \ldots, u_m \in \mathbb{Q}$, $x_1, \ldots, x_m \in N$.

$$\beta = \sum_{k=0}^{\infty} R(k) b^{-k} = \sum_{k=0}^{\infty} \left(\frac{u_1}{(dk + x_1)^s} + \cdots + \frac{u_m}{(dk + x_m)^s} \right) b^{-k},$$

При $s = 1$ число β трансцендентно, например π, => иррационально.
При $s > 1$ иррациональность не доказана.

Если взять $p = (x_1, \ldots, x_m)$ – пароль, $iv = (u_1, \ldots, u_m)$ – случайный вектор (соль), то

$$F(p, iv, s) = \underbrace{a_s, a_{s+1}, \ldots}_{\text{ключ - } n \text{ бит}}$$

Пусть $\alpha_0 = \sum_{k=0}^{\infty} c_k$ и $|c_k| < b^{-k}$ для всех $k > k_0$.
Определим $\delta_{-1} = 0$ и

$$\alpha_k = b(\delta_{k-1} + c_k b^{k-1}), \quad a_k = \lfloor \alpha_k \rfloor, \quad \delta_k = \alpha_k - a_k$$

тогда

$$\lim_{k\to\infty} \sum_{i=0}^{k} a_k b^{-k} = \alpha_0.$$

1. Для чисел β имеем $\alpha_k = b\delta_{k-1} + R(k)$.
2. Вычисления только с целыми числами.
3. Используемые операции: сложение, умножение, остаток от деления в $\mathbb{Z}$.

Что мы можем обеспечить

Преимущества:

1. Криптографические свойства: непериодичность, однонаправленность, статистическая независимость ключей, защита от чтения назад/вперед.
2. Защита о паралльного перебора паролей, высокая сложность реализации на GPU, ПЛИС и т.п.
3. Защита от временных атак (регулярность алгоритма).

Недостатки:

1. отсутствие доказательства нормальности => тестирование ключей после выработки.
2. использование арифметики больших чисел => практическая невозможность реализации алгоритма на устройствах с малой памятью.