Kryptografia i bezpieczeństwo

Laboratorium - lista nr 2, 18 X

Zadanie 1 (10 pkt) Przechwyciłaś kilkanaście kryptogramów. Wiesz, że każdy z nich powstał jako rezultat szyfrowania wiadomości za pomocą szyfru strumieniowego. Co więcej, do szyfrowania każdej wiadomości wykorzystano ten sam klucz, czyli: $c_i = \mathsf{Enc}(\mathsf{k}, \mathsf{m_i}) = m_i \oplus G(k)$ for $i = 1 \dots l$, gdzie G jest generatorem bitów pseudolosowych, a k jest kluczem tajnym.

Napisz program (i umieść go na swoim koncie na github), który przyjmuje na wejściu l kryptogramów zaszyfrowanych za pomocą szyfru strumieniowego z tym samym kluczem. Na wyjściu program ma zwrócić teksty jawne.

Przeprowadź eksperymenty, aby określić skuteczność programu w zależności od:

- długości kryptogramów,
- liczby kryptogramów dla $l = 1, 2, 3, 4, \dots$ (od jakiej wartości l, program zaczyna działać?)
- typu szyfru strumieniowego (Salsa20, Sosemanuk, RC4, ...)
- wykorzystanego kodowania znaków ASCII/UTF-8/ISO-8859-2?

Aby uzyskać przykładowe dane, wprowadź numer indeksu do formatki na stronie.

Zadanie 2 (5 pkt) Podany kryptogram został zaszyfrowany za pomocą RC4 z kluczem o długości 128-bitów. Udało Ci się zdobyć ostatnie 64 bity klucza. Napisz program, który deszyfruje podany kryptogram poprzez znalezienie odpowiedniego klucza (pozostałych – "pierwszych" 64 bitów).

Dokładniej, kryptogram wygenerowano za pomocą funkcji biblioteki mcrypt:
`mcrypt_module_open('arcfour', '', 'stream', '')`
`mcrypt_generic_init(td, key, iv)`
Natomiast zmienna *key*:

- jest typu string,
- jest wynikiem obcięcia (do pierwszych 16 znaków) ciągu będącego hashem pewnej wartości $key \leftarrow SHA256(sekret).substr(0, 16)$.

Ile kluczy sprawdza Twój program (porównaj worst-case, average case)?

Jaka jest oczekiwana liczba "sekretów", którą należałoby wygenerować, aby uzyskać dla dwóch różnych wartości ten sam klucz?

Ile kluczy musiałby sprawdzić program gdyby klucz powstawał jako wartość prawdziwie losowa?