

Specyfika wykorzystania funkcji XOR w kryptografii

<http://ipsec.pl/leksykon-kryptograficzny/specyfika-wykorzystania-funkcji-xor-w-kryptografii.html>

Jedną z bitowych operacji logicznych stanowiących podstawę kryptografii. Daje w wyniku jedynkę dla dwóch różnych bitów, zero dla dwóch identycznych. Inne nazwy: exclusive OR , różnica symetryczna, dodawanie modulo 2. Symbolem jest plus w okręgu (\oplus).

Jeśli zapiszemy operację XOR jako $A \oplus B = C$ to tablica wyników będzie wyglądać następująco:

A\B	0	1
0	0	1
1	1	0

XOR można łatwo zrealizować przy pomocy operacji logicznych OR, AND i NAND:

$$\{XOR = (A \text{ NAND } B) \text{ AND } (A \text{ OR } B)\}$$

Istotną cechą funkcji XOR jest jej przemienność, w kontekście kryptografii zwana symetrycznością.

$$A \oplus B = C \quad C \oplus B = A \quad C \oplus A = B$$

Cecha ta powoduje, że XOR jest chętnie wykorzystywany do szyfrowania - wystarczy dodać za jego pomocą bit klucza (K) do bitu tekstu jawnego (P), by otrzymać bit kryptogramu (C). Odszyfrowanie polega na dodaniu klucza do kryptogramu - wynikiem jest tekst jawny. Można to zapisać następująco:

$$\text{Szyfrowanie: } P \oplus K = C \quad \text{Deszyfrowanie: } C \oplus K = P$$

Operacje XOR wykorzystują bardzo często <http://echelon.pl/leksykon/ssstrum.php> i szyfry strumieniowe. Wiąże się z tym jednak pewne niebezpieczeństwo, niekiedy nie zauważane przez autorów oprogramowania szyfrującego. Weźmy dwa teksty jawne P i Q , zaszyfrowane tym samym tajnym kluczem K :

$$P \oplus K = C_1 \quad Q \oplus K = C_2$$

Uzyskaliśmy dwa kryptogramy C_1 i C_2 . Ich przechwycenie przez atakującego nie powinno dać mu żadnych informacji o tekście jawnym bez znajomości klucza K . Jednak w tym przypadku tak nie jest! Spójrzmy co stanie się po dodaniu tych dwóch tekstów do siebie za pomocą XOR:

$$C_1 \oplus C_2 = (P \oplus K) \oplus (K \oplus Q) = P \oplus Q$$

W wyniku uzyskuje się więc dwa teksty jawne P i Q , "sklejone" za pomocą XOR. Tajny czynnik w postaci klucza K został wyeliminowany, a rozdzielenie tekstów P i Q jest relatywnie proste jeśli zostaną spełnione określone warunki. Przykładowy zrzut przedstawia tekst w języku polskim (P) xorowany z ciągiem liter "e" (Q). Widać wyraźnie regularności, a znając całość lub fragmenty Q atakujący może łatwo odzyskać tekst jawny P (atak z wybranym tekstem jawnym).

```
00000000 45 45 45 32 d6 04 01 1f 8f 45 12 45 35 0a 09 16 —EEE2Ö....E.E5...—
00000010 06 00 45 08 0a da 00 45 15 17 1f 00 0f d4 83 45 —..E..Ú.E.....Ô.E— 00000020 15 0a 09
0c 06 0f 04 45 15 0a 09 0c 11 1c 06 1f —.....E.....— 00000030 0b 04 45 48 6f 45 45 45 15 17
1f 00 16 11 17 1f —.EHoEEE.....— 00000040 00 02 04 0f d4 45 15 0a 16 d6 0a 12 0c 00 45 35
—....ÔE...Ö....E5— 00000050 0c 36 49 45 35 2a 45 0c 45 29 35 37 49 45 06 1f —.6IE5*E.E)57IE..—
00000060 d6 0a 0b 0e 0a 12 0c 00 6f 45 45 45 79 —Ö.....oEEEy— i/pre;
```

Podsumowując: nie wolno stosować tego samego klucza do szyfrowania sztywnym strumieniowym więcej niż raz. Jeśli musimy zastosować ten sam klucz wiele razy, dodawajmy do niego za każdym razem inny modyfikator <http://ipsec.pl/leksykon-kryptograficzny/rola-salt-w-szyfrowaniu-hasel.html>.

Przykładem błędnego zastosowania szyfrowania opartego o XOR są dwie stare wersje produktów Microsoft: <http://ipsec.pl/kryptografia/bledy-w-protokolach-kryptograficznych.html> i PPTPv1 oraz <http://ipsec.pl/kryptografia/2010/tajemnice-szyfrowania-dokumentow-microsoft-office.html> i Office 2002 (XP) i starszej.