

Nowe tryby pracy szyfrów blokowych

<http://ipsec.pl/nowe-tryby-pracy-szyfrow-blokowych.html>

<http://csrc.nist.gov/> i <http://csrc.nist.gov/publications/drafts.html#SP800-38D> i dokument SP800-38D i/a_i zawierający zalecenia dla stosowania nowego trybu szyfrów blokowych GCM (Galois/Counter Mode). GCM jest trybem o podwójnej funkcjonalności, szyfrujaco-uwierzelniającym.

Wraz z pojawieniem się bezpiecznych i wydajnych szyfrów blokowych takich jak AES zainteresowanie programistów kryptograficznymi mechanizmami bezpieczeństwa znacznie wzrosło. Jednak w momencie wdrożenia algorytmu programista zawsze natrafia na podstawowy problem - w jakim trybie szyfrowania zastosować dany algorytm? Nadal najpopularniejsze są znane od ponad dwóch dekad tryby takie jak ECB czy CBC.

Jednak zaraz po wyborze trybu pojawia się kolejny problem - jak dodać do niego ochronę integralności lub/i autentyczności? W świetle dzisiejszej wiedzy rezygnacja z ochrony integralności lub błędne jej zaimplementowanie jest kryptograficznym samobójstwem, o czym przekonali się autorzy protokołu SSHv1 (<https://honor.trusecure.com/pipermail/firewall-wizards/1998-June/002845.html>) i CORE SDI, "SSH Insertion Attack", 1998 i/a_i). I czy <http://citeseer.ist.psu.edu/441782.html> szyfrować, a potem uwierzelniać i/a_i, czy na odwrót?

Problem ten rozwiązują nowe tryby szyfrujaco-uwierzelniające, które w jednym kroku (jednej operacji API programistycznego) równocześnie zapewniają ochronę poufności i integralności, przy czym to drugie przy pomocy szyfru blokowego (np. AES), bez konieczności stosowania dodatkowych mechanizmów takich jak HMAC.

Dwa najpopularniejsze algorytmy tego typu to OCB i CCM. Poniżej krótka ich charakterystyka:

<http://www.cs.ucdavis.edu/~rogaway/ocb/> i OCB i/a_i (Offset Codebook Mode), najnowsza wersja 2.0 z 2005 roku i ilość szyfrowań prawie taka sama jak w CBC (czyli jest wydajny) i opatentowany

CCM (Counter Mode with CBC MAC) i tryb licznikowy (CTR) plus CBC MAC (Message Authentication Code) i prawie dwa razy więcej szyfrowań niż OCB i dostępny jako public domain

Z punktu widzenia wydajności najbardziej atrakcyjny jest tryb OCB, opracowany przez Philipa Rogaway'a, znanego z tworzenia kryptograficznych majstersztyków (takich jak szyfr Safer). Niestety programistów odstrasza od niego fakt istnienia patentu na algorytm i konieczność uzyskania licencji na jego stosowanie (choć Rogaway zapewnia, że opłata jest niewielka). Odpowiedzią na OCB jest otwarty CCM, ale jest on prawie dwukrotnie wolniejszy - choć i tak zapewne szybszy od tradycyjnego CBC z HMAC.

Nie miałem czasu analizować trybu GCM, ale fakt przyjęcia go przez CSRC NIST dobrze mu wróży.