Kryptologie pro praxi – operační mód

V ST 7 2003 [1] jsme si zavedli pojmy symetrická šifra a vysvětlili rozdíl mezi proudovou a blokovou šifrou. Symetrické šifry mohou sloužit k zajištění různých služeb, nejčastěji jsou to zejména autentizace a utajení. Nejprve se budeme věnovat utajení zpráv pomocí jejich šifrování. Předpokládáme, že odesílatel i příjemce mají k dispozici stejný tajný klíč, kterým šifrují nebo dešifrují zprávy. Uvědomme si, že do tohoto modelu zapadá i situace, kdy si on-line (neboli on-the-fly, tj. transparentně v rámci operace zápis/čtení) šifrujeme data na pevném disku. V případě jejich ukládání (zašifrování) jsme v roli odesílatele a v případě jejich čtení (odšifrování) jsme v roli příjemce. Klíč bývá dnes nejčastěji odvozen hašovacími funkcemi přímo z přístupového hesla nebo fráze. U propracovanějších ochran se klíč získává jako náhodně vygenerovaný řetězec, který je uložený na uživatelově čipové kartě, SIM kartě apod. (takové vysoce kvalitní klíče si málokdo dokáže pamatovat). Při volbě metody šifrování hraje roli obvykle několik faktorů. V první řadě je to rychlost šifrování a odšifrování, poté schopnost vyrovnání se s chybami na komunikačním kanálu (samosynchronizace) a v neposlední řadě i bezpečnostní vlastnosti použité metody za předpokladu použití kvalitních šifer. I u té nejlepší šifry záleží na její aplikaci. Tomu se odborně říká operační mód.

Jak to nedělat

Vezměme si poučení z jednoho šifrovacího prostředku, který používal blokovou šifru k šifrování pevného disku a disket se systémem *FAT*. Odhlédneme od toho, že reálně byla použita *DES*, neboť uvidíme, že v tomto případě na kvalitě použité blokové šifry opravdu nezáleží (klidně to mohl být všeobecně uznávaný *AES*). Produkt na základě vloženého klíče vygeneroval 512 bajtů hesla *H* a to naxoroval na první tabulku *FAT*. Druhou *FAT* ponechal nešifrovanou. Stejné *H* naxoroval i na všechny zbývající 512bajtové sektory disku. Trik s jednou šifrovanou a druhou nešifrovanou tabulkou *FAT* vymyslel návrhář proto, aby mohl kontrolovat správnost vloženého klíče při odšifrování, čímž se zároveň dostáváme k problému autentizace. Jak totiž při odšifrování poznat, že se oprávněný uživatel omylem nespletl v klíči, nebo že klíč zkouší uhodnout neoprávněná osoba? Šifrovací rutina by v tomto případě disk odšifrovala nesprávným heslem, čímž by samozřejmě vznikly naprosté nesmysly, které ovšem počítač jaksi nepozná. Takto rutina nejprve na základě vloženého klíče vygeneruje heslo *H*, zkusmo odšifruje první tabulku *FAT* a porovná ji s druhou *FAT* (nešifrovanou). Pokud srovnání vyjde správně, byl vložen správný klíč a stejným způsobem se pak odšifrují i ostatní sektory disku. Na první pohled logické, ale jedná se o velmi špatný šifrový systém, a to hned v několika směrech. V první řadě je vidět, že vše, co útočník potřebuje, má přímo na disku k dispozici. Xorováním první a druhé tabulky *FAT* získá heslo *H* (*FAT1=FAT2* xor *H*, odkud *H=FAT1* xor *FAT2*) a to jednoduše naxoruje na každý sektor disku, čímž ho odšifruje. První chybou tedy je, že heslo je možné triviálně získat. Druhou chybou je, že heslo je stejné pro každý sektor, tj. pro mnoho otevřených textů. Jak víme z [1], heslo u tohoto módu může být použito pouze pro jeden otevřený text, jinak je možné ho xorováním šifrových textů zcela eliminovat a metodou předpokládaného slova luštit všechny otevřené texty jím zašifrované. To platí i v případě, kdy soubor neobsahuje text, ale programový kód atp.

Operační módy proudových šifer

U proudových šifer obvykle o operačních módech nemluvíme, neboť sama proudová šifra už svůj vlastní mód implicitně definuje – jedná se o proudové zpracování šifrované/dešifrované zprávy znak po znaku. Znakem přitom může být bit, bajt apod. Ostatně řada používaných proudových šifer je dnes tvořena blokovou šifrou pracující v proudovém módu (viz dále módy *OFB*, *CFB* a *CTR*). I když se to nedělá, teoreticky bychom i u čistě proudových šifer (například na bázi lineárních posuvných registrů) mohli definovat módy podobné módům *CFB* a *OFB*, viz dále.

Operační módy blokových šifer

Blokové šifry šifrují najednou celý blok *B* bitů otevřeného textu. Pokud bychom šifrovali blokovou šifrou tak, jak se nám na první pohled nabízí, můžeme udělat řadu chyb. Nabízí se pochopitelně možnost šifrovat tak, že bereme postupně bloky otevřeného textu OT_n, n=1,2, ... a převádíme je na bloky šifrové: $\check{S}T_n=E_K(OT_n)$. Tomuto operačnímu módu říkáme elektronická kódová kniha (*ECB*). Nevýhodou je, že pokud útočník vidí někde stejné bloky šifrového textu, ví, že se pod nimi skrývá tentýž otevřený blok, což často v kontextu poskytuje informaci o jeho hodnotě. V takto šifrované zprávě může útočník navíc bloky šifrového textu vyměňovat, vkládat nebo vyjímat, a tak snadno docilovat pro uživatele nežádoucích a pro útočníka smysluplných (!) změn v otevřeném textu. Aby se eliminovaly slabiny *ECB*, byl zaveden mód zřetězení šifrového textu (*CBC*), který je u blokových šifer nejčastější. Obdobně jako u proudových šifer [1] se zde používá náhodná inicializační hodnota (*IV*), kterou se před zašifrováním modifikuje první blok otevřeného textu. K modifikaci následujících bloků pak slouží vždy předchozí šifrové bloky: $\check{S}T_0=IV$, $\check{S}T_n=E_K(OT_n \text{ xor } \check{S}T_{n-1})$, n=1,2, ... Snadno nahlédneme, že stejné, opakující se, otevřené bloky mají s vysokou pravděpodobností odlišné šifrové obrazy, ať už se vyskytují v rámci jedné či několika různých zpráv. Toto je vlastnost, která je u *CBC* velmi užitečná. Blokové šifry lze použít i jako proudové, tj. použít je tak, že generují proud bloků hesla H_n. Jedná se o mód zpětné vazby z výstupu (*OFB*) a mód zpětné vazby ze šifrového textu (*CFB*). *OFB* charakterizují tyto rovnice: $H_0=IV$, $H_n=E_K(H_{n-1})$, $\check{S}T_n=OT_n$ xor H_n, n=1,2, ... a *CFB* tyto vztahy: $\check{S}T_0=IV$, $H_n=E_K(\check{S}T_{n-1})$, $\check{S}T_n=OT_n$ xor H_n, n=1,2, ... Novým proudovým módem je tzv. čítačový mód (*CTR*). Zde se pomocí *IV* naplní blok čítače, který se zvyšuje o jedničku nebo vhodnou konstantu, přičemž heslo vzniká jeho šifrováním. Zde je jasná perioda takové posloupnosti hesla, která se rovná periodě čítače. Čítačový mód umožňuje vypočítat heslo na libovolné pozici proudu, aniž by příslušný konečný automat musel procházet všechny předchozí stavy. *CTR* charakterizuje rovnice: $CTR_n=(IV+n) \bmod 2^B$, $H_n=E_K(CTR_n)$, $\check{S}T_n=OT_n$ xor H_n, n=1,2, ... Zabezpečovací kód zprávy (*MAC*) je také módem blokové šifry, který řeší otázku autentizace, nebo chcete-li je to obrana proti úmyslným nebo náhodným chybám. K výpočtu *MAC* se použije jiný klíč (*MK*) než k utajení zprávy. MAC se vypočte tak, že zpráva se jakoby zašifruje klíčem MK v módu CBC s nulovým IV, přičemž „průběžný" šifrový text se nikam nevysílá. *MAC* je pak tvořen až posledním blokem $\check{S}T_n$, přičemž je možné ještě jedno přídavné šifrování navíc, tj. $MAC=E_{MK}(\check{S}T_n)$. Z výsledných *B* bitů se obvykle bere jen část o délce potřebné k vytvoření odolného zabezpečovacího kódu. V poslední době vznikají iniciativy k definici nových operačních módů, neboť se ukazuje, že je nutné zajistit jak utajení, tak autentizaci zpráv a to pokud možno jedním módem naráz. Současná kombinace módů *CBC* a *MAC* může být totiž ve výpočetně omezených zařízeních pomalá nebo příliš náročná na systémové prostředky (zejména paměť). Další informace získáte v uvedené literatuře.

Vlastimil Klíma, Tomáš Rosa,
v.klima@volny.cz, t_rosa@volny.cz

LITERATURA

[1] Kryptologie pro praxi (2), Sdělovací technika 2003, č. 7, str. 16.

[2] http://csrc.nist.gov

[3] http://cryptography.hyperlink.cz nebo http://crypto.hyperlink.cz