Kryptologie pro praxi – volba klíče

Jaký algoritmus máme použít v aplikaci, kterou navrhujeme? Postačí AES se 128bitovým klíčem nebo o čtvrtinu pomalejší AES s 256bitovým klíčem? Jakou hašovací funkci použít k derivaci klíčů? SHA-256 nebo SHA-384 a zkrátit její výstup? O bezpečnosti a vhodných délkách klíčů různých algoritmů se často vedou, tu více tu méně, zasvěcené debaty nejen mezi programátory, bezpečnostními architekty, ale i teoretickými kryptology. Obecná neshoda přitom panuje zejména u budoucích možností faktorizace, podle kterých by se například dala odhadnout vhodná délka modulu RSA, optimální pro všechny strany (výrobce, uživatel, bezpečnostní manažer, státní dozor, atd.). Často je také problém shodnout se na tom, jak vlastně chceme bezpečnost měřit. Představa, že pro kvalitní systém dokážeme do haléře vyčíslit náklady na jeho případné prolomení, je čirou utopií a veškerá čísla získaná tímto způsobem mají nanejvýš určitý „politický“ význam. Ukážeme si zde praktický pohled na to, jaké algoritmy nabízí jakou bezpečnost a jaké kombinace algoritmů má a nemá smysl používat.

Konkrétně se podíváme na kombinace a bezpečnost těch algoritmů, kterým jsme se v tomto seriálu už věnovali – v *tabulce 1* uvádíme příslušné odkazy na články i nejdůležitější normy. Dále uvádíme několik základních zdrojů [1] až [4], které vřele doporučujeme před důležitým rozhodnutím prostudovat. Zde se budeme držet velmi konzervativní normy amerického úřadu NIST [1]. Uvědomme si ovšem, že je to úřad a navíc zodpovědný za vydávání podobných bezpečnostních norem. I když jeho odhady musí být velmi konzervativní, bezpečnost algoritmu RSA je i přesto některými kryptology považována za podhodnocenou. Vedou se o tom nekonečné a často komerčně podbarvené diskuze.

Tabulka 1 Algoritmy, normy a články ve ST

Algoritmus	Standardy	Článek ve ST
DES, TDES	FIPS-PUB 46-3	11/2003
AES	FIPS-PUB 197	11/2003
SHA-1, SHA-256, SHA-384, SHA-512	FIPS-PUB 180-2	02/2004
MAC	NIST SP 800-38B	09/2003
HMAC	FIPS-PUB 198, RFC 2104	02/2004
RSA	PKCS#1, FIPS-PUB 186-2	10/2003, 03/2004
DSA, ECDSA	FIPS-PUB 186-2	04/2004
D-H	PKCS#3, RFC 2631	05/2004

Míra bezpečnosti

Jako míru bezpečnosti zvolíme hodnotu dvojkového logaritmu výpočetní složitosti nejefektivnějšího známého útoku, kterou budeme nazývat bezpečnost v bitech. Vyžaduje-li například prolomení schématu 2^n operací, potom takové schéma poskytuje *n*bitovou bezpečnost. Zdůrazněme, že z pohledu teorie složitosti se jedná o míru velmi orientační. Určení složitosti prolomení nějakého algoritmu totiž rozhodně není triviální, neboť ve většině případů neumíme dokázat, že námi vybraný „referenční“ postup luštění je skutečně ten nejefektivnější. Vezměme si například DES a jeho verze TripleDES, zkráceně TDES. TDES používá tři 56bitové klíče K_1, K_2 a K_3. Pokud jsou nezávislé a náhodné, algoritmus označujeme 3TDES a nabízí nám 3×56 =168bitový klíč a podle metody zkoušení všech klíčů poskytuje nejvýše 168bitovou bezpečnost (ve střední hodnotě pak jen 167bitovou bezpečnost; s ohledem na přehlednost výkladu budeme dále vliv rozdílu mezi maximální a střední složitostí zanedbávat). Protože je však znám útok na 3TDES, který má složitost pouze 2^{112}, NIST ohodnocuje bezpečnost 3TDES jakoby měla jen 112 plnohodnotných bitů klíče. TDES se dvěma klíči (K_1 a K_2 jsou nezávislé, K_3=K_1) označujeme 2TDES. Bude-li mít útočník k dispozici velké množství párů otevřeného a šifrového textu (2^{40}), lze provést útok se složitostí pouze 2^{80}, takže NIST hodnotí bezpečnost 2TDES známkou 80 bitů. U AES zatím nejsou známy podobné útoky, a tak bezpečnost je dána přímo délkou klíče (128, 192 nebo 256 b). U *n*bitové hašovací funkce je bezpečnost s ohledem na hledání kolize díky narozeninovému paradoxu stanovena na *n*/2 bitů. Pokud nám ale na kolizi nezáleží, považujeme bezpečnost za *n*bitovou.

V *tabulce 2* máme ohodnocenu bezpečnost různých algoritmů (jednotlivé termíny: privátní klíč, řád podgrupy, apod. viz předchozí díly seriálu). Nyní se podívejme

Tabulka 2 Srovnání bezpečnosti různých kryptografických algoritmů

Bezpečnost v bitech	Symetrický algoritmus	Hašovací funkce (*)	Hašovací funkce (**)	Asymetrická schémata DSA a D-H		RSA	EC-schémata
				modul (p)	privátní klíč	modul (N)	řád „pracovní“ podgrupy
80	2TDES	SHA-1		1024	160	1024	160
112	3TDES			2048	224	2048	224
128	AES-128	SHA-256		3072	256	3072	256
160			SHA-1				
192	AES-192	SHA-384		7680	384	7680	384
256	AES-256	SHA-512	SHA-256	15360	512	15360	512
384			SHA-384				
512			SHA-512				

*) případ, kdy záleží na tom, aby nedošlo ke kolizi

**) případ, kdy kolize nevadí (například u generování náhodných čísel apod.)

na bezpečnost, kterou nabízí jejich kombinace v nějaké aplikaci. Dejme tomu, že k šifrování dat používáme 3TDES, ke generování jejich klíčů pseudonáhodný generátor na bázi SHA-1 a k jejich přenosu RSA-1024. Jaká je bezpečnost výsledné aplikace? 3TDES nabízí 112 bitů, SHA-1 je použita tak, že nevadí kolize (pro účely generátoru se využije prostor všech 2^{160} hodnot, což je zde podstatné), takže nabízí 160 b a RSA-1024 nabízí bezpečnost 80 b. Suma sumárum, nejslabší místo naší aplikace je RSA, takže celý systém musíme ohodnotit 80bitovou bezpečností. Proč? Protože symetrický klíč k 3TDES, který přenáší RSA, můžeme pomocí luštění RSA (podle názoru NIST) se složitostí 2^{80} získat, i když má 112 b. Zašifrovaná data jím potom odšifrujeme a složitost SHA-1 do hry už nezasáhne – nemusíme pátrat po tom, z čeho byl tento klíč vygenerován. Pro ty, kteří hodnotí složitost RSA-1024 jinak než tabulka, je metodika hodnocení s využitím zavedené míry stejná, jen si do tabulky dosadí svá čísla. Rozbor polemiky o složitosti luštění RSA – viz [5] (k dispozici v [6]). Další poznámka je k hašovacím funkcím. Pokud z výstupu *n*bitové hašovací funkce *h* použijeme pouze *m* bitů, *m*<*n*, používáme ve skutečnosti hašovací funkci *h*' s hašovým kódem *m* bitů, takže původní *n*bitovou bezpečnost musíme adekvátně snížit na hodnotu *m* bitů. U kolizí tomu odpovídá pokles z *n*/2 na *m*/2.

Prognóza bezpečnosti

Při posuzování bezpečnosti nějakého systému musíme brát v úvahu také životnost dat, která mají být kryptografickými technikami chráněna. Uvažme, že jsme v roce 1980 použili tehdy moderní algoritmus DES s 56bitovým klíčem a data, která jsme zašifrovali, jsme přenášeli přes Internet nebo rádiovým spojením. Je tedy mnoho šancí, že tato data mohla být útočníkem zachycena a uložena. V roce 1980 měly stroje lusticí DES nanejvýš tajné služby, v roce 1997 byl DES-Cracker [6] sestrojen veřejně a luštění dříve zachycených dat si můžeme i objednat. O podobné situaci musíme uvažovat i dnes, kdy šifrujeme nějaká data. K tomuto účelu vydal NIST opět příslušnou tabulku viz *tabulka 3*. Musíme si jen stanovit nejzazší dobu, dokdy musí šifrovaná data zůstat ochráněná, podpis být nepadělatelný, autentizace neprolomitelná, atp. – viz první sloupec *tabulky 3*.

NIST také uvádí, že kdyby se kvantová kryptoanalýza přesunula z experimentů do praxe, asymetrické techniky by přestaly být bezpečné. Předpokládá se, že u symetrických by kvantový lusticí stroj jejich bezpečnost posunul z *n* na *n*/2 bitů (viz články v [5]). V současné době se kvantové kryptoanalýzy obávat rozhodně nemusíme, co však bude za 30 let, to nevíme. Nicméně je pravděpodobné, že pokrok na tomto poli by přicházel postupně a během mnoha let spíše, než během několika dní. Pravděpodobně tak bude dost času se na tuto situaci připravit a ochránit alespoň data budoucí, když už ne tajemství minulá.

Poznamenejme, že NSA, zodpovědná za ochranu utajovaných informací v USA (NIST zodpovídá za neutajované), povolila, aby AES-192 a AES-256 mohly být používány pro ochranu dat do stupně utajení TOP SECRET a AES-128 do stupně SECRET. Poprvé v historii tak USA schválily veřejnou šifru pro ochranu přísně tajných dat.

Tabulka 3 Možnosti použití algoritmů pro ochranu dat v jednotlivých letech

Roky (doba funkce)	Symetrické algoritmy pro šifrování a MAC	hašovací funkce (*)	hašovací funkce (**)	asymetrická schémata DSA a D-H		RSA	EC-schémata
				modul (p)	privátní klíč	modul (N)	řád „pracovní" podgrupy
2004–2015 (min. síla 80 bitů)	2TDES, 3TDES, AES(***)	SHA-1, 256, 384, 512	SHA-1, 256, 384, 512	1024	160	1024	160
2016–2035 (min. síla 112 bitů)	3TDES, AES(***) 384, 512	SHA-256, 384, 512	SHA-1, 256, 384, 512	2048	224	2048	224
2036 a dále (min. síla 128 bitů)	AES(***)	SHA-256, 384, 512	SHA-1, 256, 384, 512	3072	256	3072	256

*) případ, kdy záleží na tom, aby nedošlo ke kolizi

**) případ použití hašovací funkce, kdy kolize nevadí

***) všechny tři délky klíče (128, 192, 256)

Závěr

Ukázali jsme si praktický přístup k problematice odhadu bezpečnosti kryptografických schémat, včetně vhodných způsobů jejich kombinace a závislosti požadované bezpečnosti na čase. Zároveň jsme ale několikrát poukázali na to, že se stále jedná o odhady, které mohou postupem času vzít více či méně za své. Ať už k lepšímu nebo k horšímu. Rozhodně proto nedoporučujeme například uvažovat tím způsobem, že schéma vybrané podle uvedeného postupu musí do zvoleného roku skutečně vydržet a to bez jakékoliv další péče. Máme-li už nějakou interpretaci zavést, potom je nejlépe tvrdit, že chceme-li určitou bezpečnost na určitou dobu, potom je volba předepsaného (viz *tabulky 1*, *2* a *3*) důležitá jak samotná strategie válek (odpovídá tématu článku), tak i denní taktika jednotlivých bitev.

Vlastimil Klíma, Tomáš Rosa,
klima@lec.cz, trosa@ebanka.cz

LITERATURA

[1] NIST Draft Special Publication 800-57, http://csrc.nist.gov/CryptoToolkit/kms/guideline-1-Jan03.pdf

[2] Lenstra, Verheul : Selecting Cryptographic Key Size, 2001, http://www.win.tue.nl/~klenstra/key.pdf

[3] A Cost-Based Security Analysis of Symmetric and Asymmetric Key Lengths, RSA Laboratories Bulletin 13, April 2000 (Rev. November 2001), http://www.rsasecurity.com/rsalabs/bulletins/bulletin13.html

[4] RFC 3766: Determining strength for Public Keys, April 2004, ftp://ftp.rfc-editor.org/in-notes/rfc3766.txt

[5] Klíma V.: Dvě čísla za 200 000 dolarů, CHIP 9 a 10/2001

[6] Archivy autorů: http://cryptography.hyperlink.cz a http://crypto.hyperlink.cz