Skipjack a KEA

Jak to dělá tajná služba

23. června 1998 ministerstvo obrany USA překvapivě oznámilo, že National Security Agency (NSA) odtajnila šifrovací algoritmy Skipjack a KEA. Stalo se tak poprvé v padesátileté historii této nejmocnější americké agentury. Se Skipjackem jsme vás seznámili v minulém čísle a dnes se podíváme na zoubek asymetrické šifře Key Exchange Algorithm (KEA), která se používá pro výměnu tajných šifrovacích klíčů na nechráněném komunikačním kanálu.

Že nyní známe algoritmy KEA a Skipjack, je určitě velmi zajímavé. Na základě jejich znalosti však můžeme také nepřímo odhadnout, jaké bezpečnostní principy a hranice jsou přijatelné pro americkou tajnou službu, která je stvořila. Máme tak možnost trochu se jí podívat pod pokličku a dozvědět se, co si o určitých principech konstrukce symetrických a asymetrických šifer myslí největší kryptologické centrum na světě – jak známo, NSA sdružuje nejlepší světové matematiky, fyziky a inženýry pro konstrukci vlastních i luštění cizích šifer.

Symbióza klasických a asymetrických šifer

Než přejdeme k popisu KEA, řekněme si, k čemu je vlastně tento algoritmus dobrý. V moderních systémech ochrany dat je běžné, že je k dispozici jak symetrická šifra, tak asymetrická. Každá z nich plní trochu jinou funkci. **Symetrické šifry** vzhledem ke své rychlosti šifrují vlastní tok dat. Potřebují však k tomu šifrovací klíč, který se většinou ustanovuje na každou relaci (sezení, směr spojení) zvlášť a náhodně. Pak ovšem vzniká otázka, jak tento klíč předat druhé straně (programu, zařízení), resp. jak se na něm tato zařízení nebo programy mají společně domluvit.

A právě k tomu dobře poslouží **asymetrická šifra** (proto také KEA byla zveřejněna současně a v jednom dokumentu se symetrickým algoritmem Skipjack). Asymetrické systémy jsou relativně velmi pomalé, proto šifrování vlastních dat už neprovádějí a přenechávají to mnohem rychlejším šifrám symetrickým. Pro úplnost dodejme, že výměnu klíčů lze provádět i symetrickými algoritmy a různými kryptografickými protokoly, ale zde se soustředíme na tento dnes nejběžnější a nejpopulárnější způsob výměny klíčů. (Pokud byste si chtěli ještě osvěžit některé pojmy z této oblasti, o asymetrické kryptografii a problému diskrétního logaritmu jsme psali v Chipu 2/95 na str. 126.)

Proměnné v algoritmu KEA

Označení	Význam proměnné
p	1024bitový prvočíselný modul definující multiplikativní grupu, v níž se provádějí všechny výpočty; $p = p_{1023}\, p_{1022} \ldots p_0$
q	160bitový prvočíselný dělitel čísla $p-1$; $q = q_{159}\, q_{158} \ldots q_0$
g	1024bitové číslo; $g = g_{1023}\, g_{1022} \ldots g_0$ s vlastností, že je prvkem řádu q multiplikativní grupy modulo p (matematicky: $g^q = 1 \bmod p$)
x	160bitový tajný klíč komunikující strany, takový, že $0<x<q$
Y	1024bitový veřejný klíč komunikující strany; $Y = Y_{1023}\, Y_{1022} \ldots Y_0 = g^x \bmod p$, kde x je tajný klíč (komunikující strany)
pad	80bitová konstanta; $pad_{79}\, pad_{78} \ldots pad_0$= 72F1A87E92824198AB0B (hex.)
r	komunikující stranou náhodně vygenerované 160bitové číslo; $r = r_{159}\, r_{158} \ldots r_0$

Asymetrická šifra KEA

KEA je asymetrický algoritmus určený pro ustanovení (výměnu) 80bitových šifrovacích klíčů ke Skipjacku. KEA navíc sama používá Skipjack k redukci dohodnuté 1024bitové proměnné (w) na 80bitový klíč (***Key***). KEA je vlastně variantou Diffie-Hellmanova algoritmu (D-H), který modifikuje tak, aby Diffie-Hellmanův prvočíselný modul p měl navíc vlastnost, že $p-1$ má 160bitového prvočíselného dělitele q. Po dohodě tajné 1024bitové hodnoty w oběma stranami (označme je klasicky A a B) je na konci tohoto protokolu hodnota w zpracována pomocí Skipjacku na 80bitovou hodnotu – finální tajný klíč ***Key***. Předpokládá se, že obě komunikující strany (zařízení, programy) A a B mají k dispozici společné hodnoty p, q, g, jejichž význam vidíte v tabulce. Postup ustavení klíče ***Key*** je uveden níže; označení A a B u definovaných čísel znamená jejich výpočet nebo volbu uvedenou stranou.

Připomeňme ještě jedno kouzlo, na němž je založena bezpečnost KEA a D-H algoritmu. Vzpomeňte si na školní úlohu, rovnici $g^x = y$ při známých hodnotách g a y. Jejím řešením je $x = log_g\, y$. To je triviální problém logaritmu. Když ale operaci „=“ nahradíme operací „= modulo p“ a řešení x (při znalosti g a y) nehledáme jako reálné, ale jako celé číslo, dostáváme netriviální tzv. ***problém diskrétního logaritmu*** (PDL). Ten je velmi složitý a pro velká prvočísla p současnými prostředky výpočetně nezvládnutelný! Jinými slovy, když diskrétní logaritmus x držíme v tajnosti, nikdo ho není schopen z hodnoty y (rovné $g^x \bmod p$) vypočítat. Právě tohoto principu využívá KEA k ustanovení tajného klíče na nechráněném komunikačním kanálu.

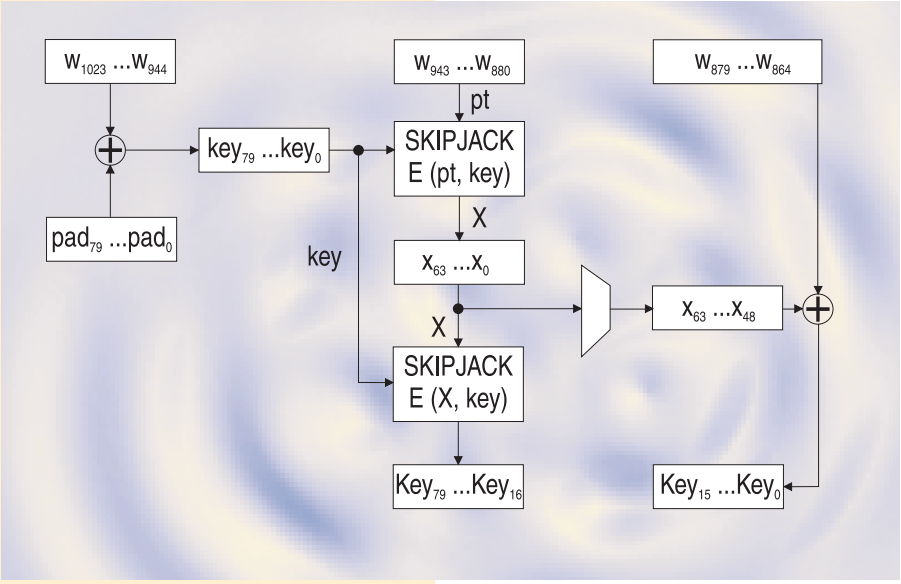
INFOTIPY

Skipjack a KEA

V minulém dílu jsme uvedli všechny informace potřebné k tomu, aby bylo možno Skipjack naprogramovat. To mnozí z vás také udělali – a samozřejmě pak přišli na chybu v tabulce funkce ***F***. Za tu se moc omlouváme, správně má být ***F(BC) = E6***.

U Skipjacku existuje řada možných zrychlení při programové realizaci kódu jak v asembleru, tak v jazyce C. Užitečné informace k oběma algoritmům i programy naleznete na těchto adresách:

ftp://ftp.funet.fi/pub/crypt/cryptography/symmetric/skipjack/
http://www.defenselink.mil/news/Jun1998/b06231998_bt316-98.html
http://csrc.nist.gov/encryption/skipjack-kea.htm
http://jya.com/skipjack-spec.htm



Tvorba tajného klíče *Key* z hodnoty *w*.

Protokol KEA

Následující postup lze s drobnou úpravou (viz dále) použít i v režimu offline, tj. když protistrana není momentálně na příjmu (online). To je běžné například při zasílání elektronické pošty.

A a B si vymění nebo si jinak zjistí (např. přečtou z lokálního disku nebo z důvěryhodného serveru) certifikáty protistrany. Certifikát kromě jiného obsahuje hodnotu Y protistrany a zajišťuje, že tato hodnota protistraně skutečně patří. (Nástroje pro vytváření certifikátů v tomto článku rozebírat nebudeme. Ani standard KEA se tím nezabývá a předpokládá, že to je nějakým způsobem zajištěno.) Na konci tohoto kroku bude proto strana A mít k dispozici hodnotu YB ($= g^{xB} \bmod p$) a strana B bude mít hodnotu YA ($= g^{xA} \bmod p$).

Každá strana si zvolí náhodné číslo r, vypočte z něj R a odešle protistraně. Konkrétně tedy A zvolí rA, vypočte $RA = g^{rA} \bmod p$ a tuto hodnotu odešle straně B. Strana B zvolí rB, vypočte $RB = g^{rB} \bmod p$ a odešle RB straně A.

A i B nyní zkontrolují, že hodnoty R a Y protistrany mají řád q a patří do multiplikativní grupy modulo p. Jinými slovy: A ověří, že $1 < RB, YB < p$, $RB^q = 1 \bmod p$, $YB^q = 1 \bmod p$. Podobně strana B ověří, že $1 < RA, YA < p$, $RA^q = 1 \bmod p$ a $YA^q = 1 \bmod p$.

Nyní obě strany vypočítají hodnotu t (strana A vypočítá tAB, strana B vypočítá tBA). Matematicky se ovšem jedná o stejnou hodnotu $t = tAB = tBA$, kterou díky problému diskrétního logaritmu není schopna vypočítat žádná třetí strana (která by snad chtěla odposlouchávat tuto komunikaci). Strana A vypočítá $tAB = YB^{rA} \bmod p = g^{xBrA} \bmod p$, strana B vypočte $tBA = RA^{xB} \bmod p = g^{rAxB} \bmod p$.

Dále A i B vypočítají hodnotu u (strana A vypočítá uAB, strana B vypočítá uBA). Matematicky se opět jedná o stejnou hodnotu $u = uAB = uBA$, kterou díky PDL není schopna vypočítat žádná třetí strana. Strana A tedy vypočítá $uAB = RB^{xA} \bmod p = g^{rBxA} \bmod p$. Strana B zase vypočte $uBA = YA^{rB} \bmod p = g^{xArB} \bmod p$.

Nyní zbývá, aby obě strany zkontrolovaly, že $w = (t + u) \bmod p$ je nenulové.

V dalším kroku je dohodnutá tajná 1024bitová hodnota w redukována pomocí algoritmu Skipjack na 80bitovou hodnotu Key. Označíme-li nejvýznamnější bity příslušných proměnných nejvyššími indexy, tvorbu Key znázorňuje připojené schéma.

Ochrana elektronické pošty

Při odesílání e-mailu nemáme bohužel k dispozici interaktivní výměnu dat. Konkrétně nám od zamýšleného příjemce elektronické pošty, řekněme B, chybí jím vygenerovaná hodnota RB. Protokol KEA na to pamatuje a pomůžeme si jednoduchým způsobem, který (jak uvidíte, promyslíte-li si řádně problém diskrétního logaritmu) neovlivní bezpečnost přenášených dat: místo hodnoty RB použijeme hodnotu YB, kterou příjemce pochopitelně zná. „Náhodnost“ hodnot t a u bude pak zajištěna jen stranou vysílající, a to prostřednictvím její hodnoty RA. Tu ovšem B dostane také, takže protokol funguje i bez přítomnosti příjemce online.

Závěr

KEA je velmi dobrou modifikací Diffie-Hellmanova algoritmu, jehož bezpečnost je založena na problému diskrétního logaritmu. Dokumenty definující algoritmy Skipjack a KEA obsahují i testovací vektory a na internetu naleznete přímo zdrojové kódy těchto algoritmů. Nepochybuji o tom, že byste mohli některé z nich vylepšit nebo urychlit. Podstatné je, že poprvé v dějinách kryptologie dostáváme symetrickou i asymetrickou šifru přímo od NSA, která je v této oblasti považována za nejpovolanější.

Vlastimil Klíma
(vklima@decros.cz)