Pod pokličkou kryptologie

Čas od času se lze setkat s názorem, že všechny kryptografické funkce mohou být prolomeny a že celé to snažení okolo nemá moc smysl. To první tvrzení je mnohem blíže pravdě než to druhé. Avšak celé to snažení okolo je více než užitečné. Bez kryptografie by neexistovalo internetové bankovnictví ani mobilní telefony, ani spousta dalších příjemných služeb. Kryptografická autentizace (tj. nemyslí se tím mizerné šifrování hovorů) umožnila u mobilních telefonů účtovat hovory autentizované osobě (vlastníkovi SIM karty), čímž mohl vzniknout trh. Internetové bankovnictví je zase založeno zejména na kryptografii s veřejným klíčem (protokol SSL/TLS).

Ale vraťme se k prvnímu tvrzení. To, že u některých kryptografických funkcí z principu nelze prokázat jejich nerozluštitelnost, nebrání v tom, aby nebylo možné se na internetu podívat na své bankovní konto. Podobně fyzici nemají absolutní jistotu v tom, co je světlo, ale my si i přes tento teoretický nedostatek rozsvítíme stolní svítidlo nebo se přes vlákna vedoucí světlo připojíme k internetu. Přesto je možná zajímavé vědět, na čem kryptologie staví svou důvěru. Odpověď je jednoduchá – na složitosti nelineárních funkcí. Jestliže se stokrát složí lineární funkce, získá se zase pouze lineární funkce. Pokud stokrát složíme nelineární funkci, dostaneme opět nelineární funkci, ale je-li vhodně udělaná, může se blížit funkci zcela náhodné nebo být od ní nerozlišitelná. A proti náhodným funkcím se špatně útočí.

V moderních kryptografických technologiích jde o to, konstruovat co nejsložitější nelineární funkce co nejjednodušeji! To je ale pěkný rozpor, že? V binárním světě je možné každou funkci (a každý její výstupní bit) vyjádřit jako polynomiální funkci vstupních bitů.

Míru složitosti booleovské funkce n proměnných lze stanovit mnoha způsoby. Podívejme se na první z nich. U náhodných booleovských funkcí se často vychází z jejich vyjádření v algebraické normální formě (ANF), což je součet termů typu:

$$\sum_{k_1=0}^{t} \cdots \sum_{k_n=0}^{t} a_{k_1,\cdots,k_n} x_1^{k_1} \cdot \cdots \cdot x_n^{k_n}$$

Například binární funkce tří proměnných $x_1\, x_2 \oplus x_1\, x_3 \oplus x_2\, x_3 \oplus x_1\, x_2\, x_3$ obsahuje čtyři termy, a má tak složitost 4. Připomeňme, že se zde pracuje s bity a že vynechané znaménko násobení znamená operaci AND.

Bude-li se řešit kryptologický problém, jako je třeba hledání šifrovacího klíče k AES nebo hledání vzoru hašovací funkce, je možné ho vždy převést do soustavy rovnic. Rovnice budou mít tvar $f_i(x_1, \ldots, x_n) = c_i$ pro i = 1, 2, …, kde f_i budou booleovské funkce neznámých booleovských proměnných x_1 ,…, x_n a c_i určité konstanty. Kryptologové svou důvěru v daný kryptografický mechanismus staví na tom, že dosud nikdo nenašel efektivní postup pro řešení příslušné soustavy těchto rovnic. Existuje metoda – útok hrubou silou, což je vyzkoušení všech možných hodnot neznámých $x_1, \ldots, x_n$. To vyžaduje provést 2^n operací. Vzhledem k počtu neznámých je to však exponenciální složitost. V reálném světě se lze setkat s úlohami, kde n je minimálně 128, a protože 2^{128} operací není možné za náš život provést, je příslušná úloha během našeho života neřešitelná. Skutečně, matematici tuto úlohu považují za jednu z nejsložitějších úloh (tzv. NP-úplné úlohy). Kdyby se našlo její efektivní řešení, tj. bylo by nutné provést např. jenom n^{10} operací (našla by se metoda řešení v polynomiálním čase), uměli by vyřešit téměř nekonečně mnoho dalších složitých úloh, které jsou na řešení soustavy booleovských rovnic převeditelné.

Tabulka 1 Počet termů v součtu dvou 32bitových celých čísel

bit si	počet termů
s0	2
s1	3
s2	5
s3	9
s4	17
s5	33
s6	65
s7	129
s8	257
s9	513
s10	1025
s11	2049
s12	4097
s13	8193
s14	16385
s15	32769
s16	65537
s17	131073
s18	262145
s19	524289
s20	1048577
s21	2097153
s22	4194305
s23	8388609
s24	16777217
s25	33554433
s26	67108865
s27	134217729
s28	268435457
s29	536870913
s30	1073741825
s31	2147483649
součet	4294967327

Nejnovější kryptografické technologie musely přijít s „trikem“, jak vyřešit onen pěkný rozpor ze začátku. Tím je prostá operace aritmetického sčítání, známá operace ADD, realizovaná u všech rozumných procesorů jedinou instrukcí, trvající jediný takt. Ukážme si, jaké polynomy poskytuje obyčejná operace a + b dvou 32bitových slov a a b.

Označme bity slov indexy 0 až 31. Při sčítání při přechodu doleva postupně vznikají bity přenosu $c_1, c_2, \ldots, c_{31}$. Součet a a b označme $s = a + b$ a jeho bity ($s_{31}, s_{30}, \ldots, s_1, s_0$). Dejme tomu, že by již byly zjištěny bity přenosu. Pak je výsledek $s_0 = a_0 \oplus b_0$, $s_1 = a_1 \oplus b_1 \oplus c_1$, $s_2 = a_2 \oplus b_2 \oplus c_2$, … $s_{31} = a_{31} \oplus b_{31} \oplus c_{31}$. Teď bude nutné bity přenosu dopočítat. Takže postupně:

$c_1 = a_0\, b_0$,

$c_3 = a_2\, b_2 \oplus a_2\, c_2 \oplus b_2\, c_2$

.....

$c_{30} = a_{29}\, b_{29} \oplus a_{29}\, c_{29} \oplus b_{29}\, c_{29}$

$c_{31} = a_{30}\, b_{30} \oplus a_{30}\, c_{30} \oplus b_{30}\, c_{30}$

Dosadí-li se jednotlivé výrazy pro bity carry do vyšších bitů (třeba c_1 do výrazu pro c_2), získají se postupně polynomy vyšších řádů s mnoha sčítanci (termy) různých řádů.

Například ze začátku je

$c_1 = a_0\, b_0$, tj. 1 term řádu 2

$c_2 = a_1\, b_1 \oplus a_1\, a_0\, b_0 \oplus b_1\, a_0\, b_0$, tj. 1 term řádu 2 a 2 termy řádu 3

$c_3 = a_2\, b_2 \oplus a_2\, c_2 \oplus b_2\, c_2$, tj. 1 term řádu 2, 2 termy řádu 3 a 4 termy řádu 4

Počet termů vzrůstá exponenciálně, jak ukazuje následující věta.

Věta: Počet termů i-tého bitu carry (c_i) a i-tého bitu (s_i) součtu dvou 32bitových čísel s = a + b je $2^i - 1$ a $2^i + 1$ a počet termů ve všech 32 bitech součtu je $2^{32} + 31 = 4\,294\,967\,327$.

Ukažme princip důkazu matematickou indukcí. Carry bit c_i vzniká, když se sčítají bity $a_{i-1} + b_{i-1} + c_{i-1}$. Je $c_i = a_{i-1}\, b_{i-1} \oplus c_{i-1}\,(a_{i-1} \oplus b_{i-1})$. Předchozí ANF pro bit carry c_{i-1} nemohla obsahovat proměnné a_{i-1}, b_{i-1}, proto $c_{i-1}\,(a_{i-1} \oplus b_{i-1}$ obsahuje dvojnásobný počet termů, než měla ANF pro c_{i-1}. Když se k tomu připočte term $a_{i-1}\, b_{i-1}$, získá se $P(i) = 1 + 2\,P(i-1) = 1 + 2\,(2^{(i-1)} - 1) = 2^i - 1$. Bit součtu s_i obsahuje navíc lineární členy a(i) ⊕ b(i), tedy $2^i + 1$ termů. Dohromady to dává $2^{32} + 31$. Počet termů v bitech si znázorňuje *tabulka 1*.

Závěr

Ukazuje se, že operace ADD je přímo zázračná kryptologická operace poskytující v minimálním čase veliký počet vysoce nelineárních vztahů. To kryptologové věděli již dávno, avšak nyní byli novými požadavky na rychlost přímo dotlačeni k jejímu využití. Tato operace je základem moderních rychlých kryptografických transformací a splňuje požadavek na růst nelinearity a složitosti vznikajících booleovských rovnic.

Vlastimil Klíma, kryptolog,
vlastimil.klima@knzsro.cz

Literatura

[1] Archiv autora a článků kryptologie pro praxi: http://cryptography.hyperlink.cz.