Posudek vedoucího na diplomovou práci

**Daniel Joščák, Hledání kolizí v hašovacích funkcích**

Hlavními výsledky práce je návrh algoritmu na hledání kolizí v hašovací funkci MD5 a analýza složitosti tohoto algoritmu a dalších algoritmů pro hledání kolizí v MD5, které byly v nedávné době publikovány.

Autor napřed ve druhé kapitole popisuje nalezení kolizí v hašovací funkci MD5. Tento výsledek byl oznámen na konferenci CRYPTO 04 v srpnu 2004, první algoritmus pro hledání kolizí byl publikován v dubnu 2005. První algoritmy používaly tzv. metodu modifikace zpráv a nalezení jedné kolize na obyčejném PC trvalo řadu hodin.

Třetí kapitola obsahuje návrh vlastního algoritmu, který je založený na důsledném využití metody modifikace zpráv. Autorův algoritmu hledá jeden první blok kolize v průměru za zhruba 7 minut a je daleko nejrychlejším algoritmem tohoto typu, který byl zveřejněn. Jisté rezervy má autor v hledání druhého bloku kolize, který mu trvá zhruba 2,5 minuty. Celý algoritmus pak hledá jednu kolizi v průměru za 10 minut. V době první implementace algoritmu v listopadu 2005 byl tento algoritmus daleko nejrychlejším algoritmem pro hledání kolizí v MD5 vůbec.

V březnu tohoto roku dva autoři (Stevens a Klíma) nezávisle na sobě přišli s novým nápadem jak algoritmus hledání kolizí v MD5 urychlit pomocí tzv. klonování kandidátů.

Algoritmy pro hledání kolizí v MD5 (stejně jako v jiných iterativních hašovacích funkcích) mají dva hlavní kroky. Napřed je generován kandidát kolize, který splňuje část podmínek, které nalezení kolize zaručují. Dále je potom kandidát kolize ověřován tím, že jsou postupně kontrolovány další podmínky. Pokud nějaká podmínka není splněna, je kandidát odmítnut, vygenerován nový kandidát a ten je opět ověřován. Rychlost algoritmu pak závisí na třech parametrech. Především je to počet podmínek, které jsou kontrolovány během procesu ověřování. Čím menší tento počet je, tím rychlejší algoritmus je. Střední hodnotu počtu těchto podmínek si označíme $E(C)$. Snižování počtu těchto podmínek je podstatou metody modifikace zpráv.

Druhým parametrem je průměrná doba ověřování jednoho kandidáta. Ta úzce souvisí s počtem podmínek, které musí být kontrolovány během ověřování kandidáta. Střední hodnotu doby ověřování si označíme $E(V)$.

Posledním parametrem je doba generování jednoho kandidáta. Právě podstatné snížení střední hodnoty doby nutné pro generování jednoho kandidáta je podstatou metody klonování. Je-li už jeden kandidát vygenerován, pak lze poměrně mnohem rychleji vygenerovat řadu jiných – blízkých kandidátů kolize. Střední hodnotu doby generování jednoho kandidáta si pak označíme E(G).

Střední hodnota doby nutné pro nalezení jednoho bloku kolize se pak rovná

$$E(C)(E(G)+E(V)).$$

Autor v práci pečlivě počítá hodnoty všech veličin $E(C)$, $E(G)$ a $E(V)$ pro všechny tři algoritmy. Doby měří v krocích výpočtu MD5, přičemž jeden krok sestává ze 4 modulárních

sčítání a jednoho výpočtu nelineární funkce tří proměnných v konečné Booleově algebře o 32 generátorech. Pro každý ze tří uvedených algoritmů pak autor spočítá střední dobu nutnou pro nalezení jedné kolize. Jeho teoretické výpočty jsou v téměř přesné shodě s experimentálně získanými daty.

Domnívám se, že po publikaci rychlejších algoritmů je právě metoda výpočtu střední doby nutné pro nalezení jedné kolize na diplomové práci nejcennější. Všechny dosud publikované odhady u jednotlivých algoritmů byly pouze experimentálně naměřené. Metodu výpočtu bude možné použít i při zkoumání algoritmů pro nalezené kolizí v jiných hašovacích funkcích, např. v SHA-0 nebo ve zkrácené verzi SHA-1.

Práce zcela vyhovuje podmínkám pro diplomovou práci a navrhuji ji hodnotit známkou **výborně.**

Doc. RNDr. Jiří Tůma, DrSc.

V Praze dne 22.5.2006