УДК 004.424.47:336.717.1

О. В. Клювак

КРИПТОГРАФІЧНА СТІЙКІСТЬ КОМБІНАЦІЙНОГО ХЕШУВАННЯ АВТЕНТИФІКАЦІЙНИХ ДАНИХ В ІНТЕРНЕТ-ПЛАТІЖНИХ СИСТЕМАХ

Описано механізм комбінаційного хешування в Інтернет-платіжних системах. Здійснено перевірку на крипостійкість комбінаційного хешування на основі визначення ймовірності виникнення колізії.

Ключові слова: *Інтернет-платіжна система, Інтернет-трансакція, банківська платіжна картка (БПК), автентифікаційні дані, автентифікація, комбінаційне хешування, ОВК (он-лайн власний код), крипостійкість, колізія.*

Постановка проблеми. У схемі здійснення трансакцій в Інтерет-платіжних системах щодо безпеки найбільш критичним кроком є передання клієнтом-покупцем реквізитів БПК, які дають можливість автентифікувати іншою стороною (продавцем) держателя платіжного засобу. Оскільки реквізити картки є практично відкритою інформацією, яка нанесена на самому пластику і може стати доступною кожному за недотримання правил безпеки, рекомендується банківським установам генерувати спеціальний код для здійснення Інтернет-трансакцій (онлайн-власний код), який є прикріплений до карткового рахунку та за якого зникає необхідність вводити реквізити БПК для проведення автентифікації та перевірки стану рахунку на стороні продавця. Зрозуміло, що і цей захід безпеки не є достаннім. Для убезпечення від перехоплення даних, які передаються через Інтернет-платіжну систему, застосовують механізм хешування. Проте навіть такі криптостійкі хеш-функції, як MD-5 та SHA-1, на нинішній день уже не є настільки надійними. Тому алгоритми хешування потребують постійного вдосконалення, а при розробці необхідно здійснювати перевірку на крипостійкість.

Аналіз останніх досліджень і публікацій. Серед українських авторів, які у свої працях висвітлювали проблеми в галузі забезпечення автентичності банківських даних, криптографічних засобів захисту в платіжних-системах банків і хешування даних для забезпечення автентичності в комп'ютерних системах і мережах, можна виділити таких, як: С. П. Євсєєв, А. А. Кузнецов, Т. Ю. Самбурська та інші. Серед іноземних авторів, які займаються дослідженнями в галузі криптографічного захисту даних, які передаються через комп'ютерні системи і мережі, зокрема хешування даних для забезпечення їхньої автентичності, можна виділити такі: Peter Kankowski, William Stallings,

Man Young Rhee, Sugata Sanyal, Ayu Tiwari and Sudip Sanyal, V. Pasupathinathan, J. Pieprzyk, H. Wang and J.Y. Cho, Sungwoo Kang, Haeryong Park, Donghyeon Cheon, Kilsoo Chun, Jaeil Lee, Альфред В. Ахо, Джон Хопкрофт, Джеффрі Д. Ульман, Ю. А. Семенов та інші.

Мета статті – описати комбінаційну схему хешування даних, котрі передаються під час здійснення Інтернет-трансакції, і перевірити схему на криптостійкість за допомогою визначення ймовірності виникнення колізії.

Обґрунтування отриманих наукових результатів. Комбінаційний тип хешування, який пропонуємо застосовувати до даних, які передає держатель картки на сервер Інтернет-платіжної системи при здійсненні автентифікації, полягає в тому, що генератор хеш-коду на вході отримує два елементи даних:

- сам код, який має бути захешовано і передано іншій стороні;
- код схеми хешування.

Крім коду, платник отримує ще одну зі схем комбінаційного хешування. Цю схему також знає приймаюча сторона, отож, її не передають при пересиланні захешованого коду при здійсненні трансакції. Хешування за такою схемою передбачає використання одночасно дев'яти хеш-функцій (RSHash, JSHash, PJWHash, ELFHash, BKDRHash, SDBMHash, DJBHash, DEKHash, APHash). При цьому кожна з них отримує на вході однакове поле, у цьому варіанті – код ОВК, представлений у текстовому форматі. Результати всіх функцій конкатенуються в єдине, 288-бітне поле (кожна функція генерує чотирибайтове число, отож, при конкатенації результатів дев'яти функцій утворюється поле довжиною 36 байтів). Те, в якій послідовності викликається згадані функції, визначається кодом схеми хешування (КСХ). КСХ являє собою текстовий рядок довжиною 9 байтів, кожен символ якого ідентифікує одну певну хеш-функцію.

Наприклад, якщо КСХ має вигляд RJPEBSDKA, то результуючий захешований код буде такий:

RSHash(ОВК) & JSHash(ОВК) & PJWHash(ОВК) & ELFHash(ОВК) & BKDRHash(ОВК) & SDBMHash(ОВК) & DJBHash(ОВК) & DEKHash(ОВК) & APHash(ОВК), де & – операція конкатенації.

Зрозуміло, що при здійсненні автентифікації обидві сторони трансакції повинні бути здатні розрахувати хеш-код комбінаційним методом. Одна сторона генерує його і відправляє його іншій, а та, у свою чергу, отримавши його, мусить згенерувати заново і звірити з отриманим. Тільки повна тотожність обох рядків є умовою для успішного проходження автентифікації. Оскільки генеруватися хеш-код має однаково, обидві сторони мають у своєму розпорядженні однаковий програмний модуль для обчислення. Для зручності використання в рамках різних програмних систем цей модуль скомпоновано у форматі динамічної бібліотеки (hashCascade.dll). Програмна система за раху-

нок виклику відповідного методу бібліотеки може отримати потрібний код, використавши на вході ОВК і КСХ.

Суть комбінаційного хешування полягає в тому, що хеш-функції викликаються не у сталій послідовності, а в тій послідовності, яка визначається схемою комбінаційного хешування. Тому виклики даних функцій організовані не як статичні виклики, а як виклики змінної на зразок «функція». При виконанні хешування перебирається увесь ланцюжок схеми хешування, і кожному елементові цього рядка знаходять відповідну хеш-функцію. Після цього найзручніше взяти адресу цієї хеш-функції та викликати її за цією адресою. Така схема дозволяє легко організувати перебір усіх функцій в одному циклі. Результати викликів усіх хеш-функцій конкатенується в єдиний 36-байтовий рядок, який повертається з бібліотеки як результат.

Для визначення криптостійкості запропонованої схеми хешування слід розрахувати ймовірність виникнення колізії та порівняти із 32-, 64- і 160-бітними хеш-функціями [1–3].

Загалом, оцінити ймовірність виникнення колізії в хеш-функціях найпростіше так. Нехай результуюче значення хеш-функції може мати максимальне значення N. Оскільки, число є цілим і додатним, це означає, що може бути N можливих значень результату цієї хеш-функції. Нехай усі можливі результати хеш-функції являють собою масив (від 0 до максимального значення з кроком 1). Після того, як довільним чином вибрати з цього масиву будь-яке значення, у цьому масиві залишається ще N – 1 значень, відмінних від першого. Звідси, імовірність генерації двох однакових чисел, які не дорівнюють один одному, становить $(N - 1) / N$.

Після вибору цих двох чисел із масиву в ньому залишається N – 2 унікальних значень. Тому ймовірність вибрати ще одне унікальне значення, яке не дорівнює попередньому, становить $(N - 2) / N$. Оскільки вибір першого і другого чисел є незалежними один від одного, то загальна ймовірність того, що ми виберемо із N значень три унікальні числа, дорівнює добутку цих двох імовірностей, а це становить

$$\frac{N-1}{N} \times \frac{N-2}{N}. \qquad (1)$$

Таким чином, можна зробити вибірку будь-яку кількість разів. Припустимо, що нашу вибірку зроблено k разів. Після кожного із цих разів імовірність того, що вибрано унікальне число, звісно, змінюється, а саме зменшується на $1 / N$. Це пояснюється тим, що кількість уже вибраних чисел збільшилося на 1, тому ймовірність вибрати це саме число зростає. А оскільки ці ймовірності ми множимо між собою, то загальна ймовірність, що з усіх k вибраних чисел жодне не буде повторюватися, дорівнює цьому добутку:

$$\frac{N-1}{N}\times\frac{N-2}{N}\times\ldots\times\frac{N-(k-2)}{N}\times\frac{N-(k-1)}{N}. \quad (2)$$

За великих значень N і k обчислення цієї ймовірності є досить тривалим, тому на практиці це ж число можна обчислити наближено так:

$$e^{\frac{-k(k-1)}{2N}}. \quad (3)$$

Оскільки ймовірність того, що вибірка k чисел з N можливих міститиме значення, які повторюються, обчислюються за формулою (3), а ймовірність того, що не міститиме, є доповнювальною до 1. Тобто ймовірність, що таких дублів не буде, виражається формулою:

$$P=1-e^{\frac{-k(k-1)}{2N}}. \quad (4)$$

Формулу (4) можна ще більше спростити. Відомо, що вираз $1 - e^{-x}$ за достатньо малих x (а для хеш-функції це так і є) апроксимується за такою формулою:

$$P=1-e^{-x}\approx X. \quad (5)$$

Тому формулу (4) можна спростити так:

$$P=1-e^{\frac{-k(k-1)}{2N}}=\frac{k(k-1)}{2N}. \quad (6)$$

Формула (6) є досить зручною, оскільки дозволяє не тільки дуже прискорити це обчислення, а й навіть дозволяє збільшити точність обчислення за рахунок того, що при виконанні обчислень за формулами (2) і (4) у реальних умовах відбувається втрата точності при здійсненні операцій над числами з плаваючою точкою в математичних співпроцесорах. Оскільки в нашому варіанті k набуває досить великих значень, то формулу можна спростити без вагомої втрати точності:

$$P=\frac{k^2}{2N}. \quad (7)$$

Тепер поближче до хеш-функцій. Стандартна хеш-функція генерує на виході число довжиною 4 байти, тобто 32 біти. Тому результат хеш-функції може бути в межах від 0 до 2^{32}, а це значить, що кількість можливих варіантів, тобто $N = 2^{32}$. При цьому відомому N, використовуючи будь-яку з попередніх формул, можна побудувати графік залежності ймовірності виникнення колізії для 32-бітного числа залежно від величини вибірки (k) (*рис.*) [4].

Принагідно можна зауважити, що для будь-якого іншого значення N характер графіка буде такий самий.

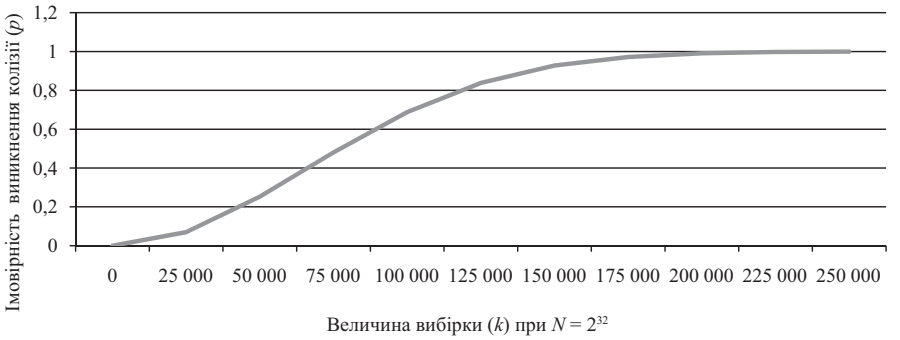

Рис. **Графік залежності ймовірності виникнення колізії (*p*) від величини вибірки (*k*) за $N = 2^{32}$**

Примітка. Складено за даними [5]

Важливою характеристикою якості хеш-функції є величина вибірки, за якої ймовірність виникнення колізії дорівнює 50%. Розрахуємо ці величини для 9 хеш-функцій: RSHash, JSHash, PJWHash, ELFHash, BKDRHash, SDBMHash, DJBHash, EKHash, APHash. Для цього розв'яжемо рівняння:

$$05 = \frac{k^2}{2^{32}} \Rightarrow k = 65\,536.$$

Це число означає, що при використанні хеш-функції 65 536 разів є ймовірність 50%, що серед них буде бодай одна колізія. Звісно, це число зростатиме при збільшенні *N*, а зменшуватися – при зменшенні ймовірності колізії.

Табл. наводить деякі значення *k* за різних значень *N* і *p* [4; 5].

Таблиця

Імовірності виникненя колізій для чисел різної довжини

32 біти*	**64 біти***	**160 біти***	**288 бітів****	**Імовірність виникнення колізії****
77 163	$5{,}06 \times 10^{9}$	$1{,}42 \times 10^{24}$	$2{,}2301 \times 10^{43}$	0,5
30 084	$1{,}97 \times 10^{9}$	$5{,}55 \times 10^{23}$	$9{,}9732 \times 10^{42}$	0,1
9 292	609×10^{6}	$1{,}71 \times 10^{23}$	$3{,}1538 \times 10^{42}$	0,01
2 932	192×10^{6}	$5{,}41 \times 10^{22}$	$9{,}9732 \times 10^{41}$	0,001
927	$60{,}7 \times 10^{6}$	$1{,}71 \times 10^{22}$	$3{,}1538 \times 10^{41}$	0,0001
294	$19{,}2 \times 10^{6}$	$5{,}41 \times 10^{21}$	$9{,}9732 \times 10^{40}$	0,00001
93	$6{,}07 \times 10^{6}$	$1{,}71 \times 10^{21}$	$3{,}1538 \times 10^{40}$	0,000001
30	$1{,}92 \times 10^{6}$	$5{,}41 \times 10^{20}$	$9{,}9732 \times 10^{39}$	10^{-7}
10	607 401	$1{,}71 \times 10^{20}$	$3{,}1538 \times 10^{39}$	10^{-8}
-	192 077	$5{,}41 \times 10^{19}$	$9{,}9732 \times 10^{38}$	10^{-9}
-	60 740	$1{,}71 \times 10^{19}$	$3{,}1538 \times 10^{38}$	10^{-10}

Закінчення табл.

32 біти*	64 біти*	160 біти*	288 бітів**	Імовірність виникнення колізії**
-	19 208	$5{,}41 \times 10^{18}$	$9{,}9732 \times 10^{37}$	10^{-11}
-	6 074	$1{,}71 \times 10^{18}$	$3{,}1538 \times 10^{37}$	10^{-12}
-	1 921	$5{,}41 \times 10^{17}$	$9{,}9732 \times 10^{36}$	10^{-13}
-	608	$1{,}71 \times 10^{17}$	$3{,}1538 \times 10^{36}$	10^{-14}
-	193	$5{,}41 \times 10^{16}$	$9{,}9732 \times 10^{35}$	10^{-15}
-	61	$1{,}71 \times 10^{16}$	$3{,}1538 \times 10^{35}$	10^{-16}
-	20	$5{,}41 \times 10^{15}$	$9{,}9732 \times 10^{34}$	10^{-17}
-	7	$1{,}71 \times 10^{15}$	$3{,}1538 \times 10^{34}$	10^{-18}
-	-	-	$9{,}9732 \times 10^{33}$	10^{-19}
-	-	-	$3{,}1538 \times 10^{33}$	10^{-20}
-	-	-	$3{,}1538 \times 10^{28}$	10^{-30}
-	-	-	$3{,}1538 \times 10^{23}$	10^{-40}
-	-	-	$3{,}1538 \times 10^{18}$	10^{-50}
-	-	-	$3{,}1538 \times 10^{13}$	10^{-60}
-	-	-	$3{,}1538 \times 10^{8}$	10^{-70}
-	-	-	3,154	10^{-80}
-	-	-	10	10^{-85}
-	-	-	3	10^{-86}

Примітки:
* джерело[5];
** розраховано на основі [5].

Таблиця показує, яку величину вибірки потрібно зробити, щоб досягнути визначеної в останньому стовпчику ймовірності колізії, для різних довжин результату хеш-функцій. Для порівняння були взяті хеш-функції, які генерують 32-бітне число, 64-бітне і 160-бітне, а також наш варіант – зборку з 9-ти стандартних хеш-функцій. Кожна із використаних функцій генерує 32-бітне число, яке є зручним для 32-бітної архітектури процесора, але має більшу схильність до колізій. Як видно з *таблиці*, довжина числа степенево впливає на зменшення ймовірності виникнення колізії. Особливо відрізняються серед цих результатів результати звичайної хеш-функції та комбінованої (288-бітної). Для того, щоб отримати ймовірність виникнення колізії, 50% вибірка для 288-бітної хеш-функції повинна бути більшою у $2{,}8901 \times 10^{38}$ разів. Очевидно, що це надзвичайно велике число [5].

З іншого боку, імовірність виникнення колізії для 32-бітного числа вичерпується на 9-му порядку (1 зі 100 млн). Тоді як для 288-бітного вона може доходити до 86-го порядку. Це означає, що при генеруванні трьох 288-бітних хеш-функцій імовірність колізії в мільйон разів менша, ніж імовірність вибрати одну і ту саму елементарну частинку двічі в будь-які точці Всесвіту.

Висновки. Ефективний механізм автентифікації в Інтернет-платіжних системах передбачає уникнення передання реквізитів платіжного засобу шляхом введення у схему трансакції «он-лайн власного коду» держателя картки, який є прив'язкою до карткового рахунку. При хешуванні даних, які надсилає держатель картки під час здійснення Інтернет-трансакції, і банку, і клієнту відомі і код, і хеш-функції, якими здійснюється формування хеш-коду, а тому банк може перевірити надісланий захешований код шляхом зіставлення з тим, який був отриманий шляхом проведення таких самих дій із тим самим кодом на боці банку. Доцільність застосування хешування даних в Інтернет-платіжних системах пояснюється тим, що ця операція незворотна, а це, у свою чергу, убезпечує від розшифрування вихідного коду зловмисником, якщо б навіть йому вдалося перехопити вже захешовані дані. При хешуванні даних завжди існує ймовірність виникнення колізій, тобто отримання однакового результату за різних вхідних даних. Дану ймовірність можна зменшити ще на кілька порядків при застосуванні схеми комбінаційного хешування. А саме: імовірність виникнення колізії для 32-бітного числа (наприклад, для MD-5 і SHA-1) вичерпується на 9-му порядку (1 зі 100 млн), тоді як для 288-бітного (такою є запропонована комбінаційна схема хешування) вона може доходити до 86-го порядку.

Список використаних джерел

1. Исследование протоколов и механизмов защиты информации в компьютерных системах и сетях / А. А. Кузнецов, С. П. Евсеев, Б. П. Томашевский, Ю. И. Жмурко // Збірник наукових праць Харківського університету Повітряних сил України ім. І. Кожедуба. – 2007. – № 2(14). – С. 102–111.
2. Credit Card Encryption and Password Hashing Utility Component [Electronic resource]. – Access mode: http://www.caritas.org.au/Content/NavigationMenu/Caritas_Documents/PDFs/asiUtil_CreditCardEncryption.pdf.
3. Семенов Ю. А. Аутентификация в Интернет [Электронный ресурс]. – Режим доступа: http://docs.luksian.com/networks/techs/intro/?f=./6/authent.shtml.
4. Ахо Альфред, В., Хопкрофт, Джон, Ульман, Джеффри, Д. Структуры данных и алгоритмы: Уч. пособие: Пер. с англ.– М.: Издательский дом «Вильямс», 2007. – 400 с.
5. Peter Kankowski Hash functions: An empirical comparison [Electronic resource]. – Access mode: http://www.strchr.com/hash_functions.

Клювак О. В.

Криптографическая стойкость комбинационного хеширования аутентификационных данных в Интернет-платежных системах

Описан механизм комбинационного хеширования в Интернет-платежных системах. Осуществлена проверка на криптостойкость комбинационного хеширования на основе определения вероятности возникновения коллизии.

Ключевые слова: *Интернет-платежная система, Интернет-трансакция, банковская платежная карточка, аутентификационные данные, аутентификация, хэш-код, комбинационное хэширования, ОСК (онлайн собственный код), криптостойкость, коллизия.*

Klyuvak O. V.

The cryptographic strength of combinational hashing the authentication data in Internet-payment system

It is described the combinational hashing mechanism in Internet-payment system. It is tested the reliability of combinational hashing by means of the probability of hash collision.

Key words: *Internet payment-system, online transaction, banking payment card, authentication data, authentication, combinational hashing, OOC (online own code), cryptographic strength, collision.*

Клювак Оксана Володимирівна – фахівець 2-ї категорії наукового відділу Львівського інституту банківської справи Університету банківської справи Національного банку України (м. Київ).

УДК 004.056

С. Т. Іванишин

МОДЕЛЮВАННЯ АВТОМАТИЗАЦІЇ БЕЗПЕКИ ВНУТРІШНЬОГО АУДИТУ В БАНКУ

Розглянуто питання інформаційної безпеки при проведенні віддаленого внутрішнього аудиту в банку. Обґрунтовано необхідність автоматизації безпеки при проведенні внутрішнього аудиту віддалено через агресивне середовище – Інтернет. Для віддаленого проведення внутрішнього аудиту визначено можливі канали витоку інформації, режими захисту, рівні контролю. Побудовано автоматизовану модель захисту інформації банку. Надано практичні рекомендації щодо впровадження наявних систем захисту від витоку.

Ключові слова: *внутрішній аудит у банку, захист каналу, захист периметра, контент-контроль, автоматизація безпеки.*

Постановка проблеми. Сучасна система менеджменту в банку згідно з вимогами міжнародного стандарту ISO 9001 включає процес управління якістю. Упровадження системи управління якістю забезпечує банкові значний ефект шляхом економії коштів завдяки збільшенню обсягів продажу банківського продукту та комплексності обслуговування, а також зменшенню кількості клієнтських претензій, підвищенню якості обслуговування клієнтів, припливу нових клієнтів [1].