9. Jean-Pierre Deschamps, Jose Luis Imana, Gustavo D.Sutter Hardware Implementation of Finite-Field Arithmetic. McGraw Hill, March 2009. 7. Шологон О. З. Обчислення структурної складності помножувачів у поліноміальному базисі елементів полів Галуа GF(2^m). Шологон О. З. // Вісник Нац. ун-ту "Львів. політехніка", 2014 8. Черкаський М. В., Мурад Хусейн Халіл. Універсальна SH-модель Комп'ютерні системи та мережі // Вісник Нац. ун-ту"Львівська політехніка". – 2004. – № 523. – С. 150–154. 9. Глухов В. С., Глухова О. В. Результати оцінки структурної складності помножувачів елементів полів Галуа // Вісник Нац. ун-ту "Львів. політехніка", – 2013. – № 773. 10. Албанський І. Б. Дослідження системних характеристик цифрових пристроїв множення реалізованих в різних теоретико-числових базисах [Електронний ресурс] / І. Б. Албанський, О. І. Волинський // Вісник Хмельницького національного університету. – 2012. – № 2. – С. 179–186. 11. Пат. US 5768168A US 08/656,784 Universal Galois field multiplier/ Jin-Hyeok Im; заявл. 30.05.1996, опубл. 16.061998. 12. Пат US 4918638A US 07/107,363 Multiplier in a galois field/ Michihito Matsumoto, Kazuhiro Murase; заявл. 9.10.1987, опубл. 17.04.1990. 13. Virtex-6 FPGA Family: Data heet.- Xilinx, January 2009/ - DS312.10.

УДК 004.052

В. С. Яковина
Національний університет "Львівська політехніка",
кафедра програмного забезпечення

МОДЕЛЮВАННЯ ПАРАМЕТРА ПОТОКУ ВІДМОВ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА ВИЗНАЧЕННЯ ДІАПАЗОНІВ ПОКАЗНИКА ЙОГО СКЛАДНОСТІ

Проведено моделювання поведінки параметра потоку відмов програмного забезпечення у випадку моделі надійності ПЗ з індексом складності, що дало змогу встановити діапазони значень цього індексу та пояснити поведінку функції виявлення помилок залежно від складності програмного продукту.

Ключові слова: програмне забезпечення, якість, надійність, складність, параметр потоку відмов.

SOFTWARE FAILURE INTENSITY MODELLING AND IDENTIFICATION THE MARGINS OF THE COMPLEXITY INDEX

The behaviour of failure intensity function for software reliability model with complexity index depending on the model parameters has been simulated. It allows identifying the margins of the complexity index and explaining the defect revealing function behaviour depending on software complexity.

Key words: software, quality, reliability, complexity, failure intensity.

Вступ

Надійність програмного забезпечення (ПЗ) є одним з найважливіших атрибутів його якості. Сьогодні значні досягнення в науці та створення проривних технологій вимагають все більшого зростання потужностей сучасної обчислювальної техніки. Так, від суто обчислювальних

спеціалізованих засобів програмне забезпечення за останні півстоліття перейшло до усіх галузей життя людини, зокрема пов'язаних з управлінням складними технічними системами, безпекою та життєдіяльністю людини. Разом з тим різким контрастом з бурхливим розвитком сучасних технологій електронної техніки є прогрес в галузі програмного забезпечення, який демонструє значно менші темпи з усіх аспектів, таких як якість, продуктивність, вартість та швидкодія. В результаті виникає протиріччя між відповідальністю і складністю сучасного ПЗ та існуючими методами аналізу та оцінювання його надійності.

Аналіз літературних джерел

Подібно до надійності апаратури, надійність ПЗ на часовому інтервалі характеризується ймовірністю безвідмовної роботи протягом певного періоду часу за певних умов [1, 2]. У результаті виконання програми стан входу перетворюється на стан виходу. Тобто програму можна розглядати як функцію *f*, яка перетворює простір входу на простір виходу, де простір входу – це множина всіх станів входу, а простір виходу – множина всіх станів виходу. Стан входу можна визначити як сукупність вхідних змінних чи типові команди/транзакції над програмою [3].

Для розв’язання задач оцінювання та прогнозування надійності ПЗ сьогодні використовуються моделі надійності різних типів [4, 5]. Поширені моделі надійності, які досліджують закономірності виявлення помилок у програмному проекті протягом однакових інтервалів часу [4]. Моделі, в основу яких покладено підрахунок відмов (динамічні моделі), припускають, що концептуально в програмі наявна скінченна кількість дефектів. Враховуючи, що кількість дефектів є цілим числом, динамічні моделі обчислюють кількість початкових несправностей на етапі відлагодження програми і кількість дефектів, що залишились під час чи вкінці етапу відлагодження. Моделі підрахунку відмов використовують інтенсивність відмов як основну характеристику появи відмови. Залежно від типу моделі передбачено, що інтенсивність відмов кожного дефекту є або сталою функцією часу відлагодження, або випадковою змінною із заданим розподілом [2–7]. В основу найпоширеніших моделей надійності ПЗ цього класу покладено розподіл Пуассона, оскільки використання цього розподілу випадкових величин добре зарекомендувало себе в багатьох областях, де основна особливість полягає в обчисленні кількості незалежних подій протягом інтервалів часу [5, 8]. Як тільки інтенсивність відмов, пов'язана з дефектами певного типу, визначено, інтенсивність відмов програми загалом обчислюється як добуток кількості дефектів, що залишились в програмі, на інтенсивність відмов кожного дефекту [3].

Група моделей на основі неоднорідного пуассонового процесу надає аналітичних засобів для опису поведінки відмов ПЗ під час тестування. Головною проблемою моделей цього типу є визначення вигляду функції математичного сподівання кумулятивної кількості відмов, що спостерігались до деякого моменту часу. До цієї групи належить значна кількість моделей, зокрема експоненційна модель Муси [9], модель неоднорідного пуассонового процесу Гоеля та Окумото [10], модель S-подібного зростання [11], гіперекспонеційного зростання [12], узагальнена модель негомогенного пуассонового процесу Тимошенко–Дідковської [13], узагальнена модель пуассонового процесу з індексом складності [14] тощо.

Постановка задачі

У попередніх дослідженнях автор зі співавторами побудував модель надійності ПЗ з індексом складності [14], на основі поведінки індексу складності формалізовано критерій достатності процесу тестування [15], проаналізовано використання цієї моделі на різних етапах життєвого циклу ПЗ [16] та розроблено метод оцінювання та прогнозування надійності програмного забезпечення на основі цієї моделі [17]. Разом з тим питання класифікації програмних продуктів за значенням індексу складності, так само як і характер залежності параметра потоку відмов ПЗ від параметрів моделі залишаються недослідженими.

Тобто метою цієї роботи є дослідження поведінки параметра потоку відмов ПЗ, що описується моделлю з індексом складності та встановлення діапазонів значень індексу складності для цієї моделі.

Дослідження поведінки інтенсивності відмов моделі з індексом складності

Модель надійності ПЗ з індексом складності [14] належить до "узагальнених пуассонових моделей" разом з узагальненою пуассоновою моделлю [5], узагальненою моделлю неоднорідного пуассонового процесу Гоеля [18] та моделлю Тимошенко–Дідковської [13]. Параметр потоку відмов [19] в моделі [14] описується виразом:

$$\omega(t) = \alpha \cdot \beta^{s+1} \cdot t^s \cdot e^{-\beta \cdot t}. \quad (1)$$

Для дослідження поведінки функції параметра потоку відмов (1) та встановлення її екстремумів знайдемо похідну цієї функції за часом:

$$\frac{d\omega(t)}{dt} = \alpha \cdot \beta^{s+1} \cdot \left(\frac{dt^s}{dt} \cdot e^{-\beta \cdot t} + \frac{de^{-\beta \cdot t}}{dt} \cdot t^s \right) = \alpha \cdot \beta^{s+1} \cdot t^s \cdot e^{-\beta \cdot t} \cdot \left(s \cdot t^{-1} - \beta \right) = \omega(t) \cdot \left(s \cdot t^{-1} - \beta \right). \quad (2)$$

Прирівнявши рівняння (2) до нуля, отримаємо:

$$\alpha \cdot \beta^{s+1} \cdot t^s \cdot e^{-\beta \cdot t} \cdot \left(s \cdot t^{-1} - \beta \right) = 0,$$

$$\text{звідки } t^s \cdot \left(s \cdot t^{-1} - \beta \right) = 0. \quad (3)$$

Розв'язавши рівняння (3), отримаємо значення часу, за якого функція $\omega(t)$ є максимальною:

$$t_{\max} = \frac{s}{\beta}. \quad (4)$$

Як видно з (4), положення максимуму параметра потоку відмов програмного продукту, на відміну від усіх відомих моделей надійності ПЗ, залежить як від якості тестування (параметр β [5]), так і від складності програмного продукту, що тестується. Зсув максимуму функції інтенсивності відмов по часовій шкалі залежно від значення індексу складності ілюструє рис. 1 (криві 2–5). Усі криві на рис. 1 побудовано з використанням однакових значень параметрів α та β (a та b для моделі Гоеля–Окумото), а значення параметра s для кривих 2–5 становило 0,5; 1; 2 та 2,5 відповідно. Зауважимо, що крива, яка відповідала параметру потоку відмов у випадку S-подібної моделі [11], повністю збігалася з кривою 3.

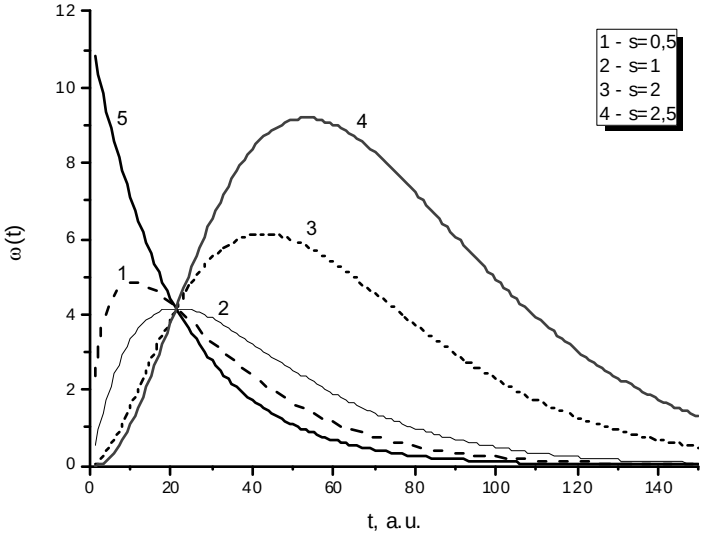

Рис. 1. Залежність поведінки функції параметра потоку відмов з часом для узагальненої моделі з індексом складності (криві 1–4) та моделі Гоеля–Окумото (крива 5)

Проаналізуємо максимальне значення параметра потоку відмов $\omega_{\max}$ та його залежність від параметрів моделі.

$$\omega_{\max}(\alpha, \beta, s) = \omega(t_{\max}) = \alpha \cdot \beta^{s+1} \cdot \left(\frac{s}{\beta} \right)^s \cdot e^{-\beta \cdot \frac{s}{\beta}} = \alpha \cdot \beta \cdot s^s \cdot e^{-s}. \quad (5)$$

Обчисливши часткові похідні функції (5) та прирівнявши їх до нуля, знайдемо екстремальні точки:

$$\frac{\partial \omega_{\max}(\alpha,\beta,s)}{\partial \alpha} = \beta \cdot s^s \cdot e^{-s}, \tag{6}$$

$$\frac{\partial \omega_{\max}(\alpha,\beta,s)}{\partial \beta} = \alpha \cdot s^s \cdot e^{-s}, \tag{7}$$

$$\frac{\partial \omega_{\max}(\alpha,\beta,s)}{\partial s} = \alpha \cdot \beta \cdot s^s \cdot (\ln s + 1) \cdot e^{-s} - \alpha \cdot \beta \cdot e^{-s} \cdot s^s = \alpha \cdot \beta \cdot s^s \cdot e^{-s} \cdot \ln s. \tag{8}$$

Як видно з рівнянь (6) та (7), за параметрами α і β функція (5) є монотонно зростаючою без екстремумів (за винятком нульових значень цих параметрів, що не має фізичного змісту). Разом з тим, з рівняння (8) видно, що поведінка максимального значення параметра потоку відмов залежно від індексу складності s є складнішою і має екстремум в точці $s = 1$. Отже, екстремальна точка є точкою мінімуму функції.

Зауважимо, що при $s = 0$ функція параметра потоку відмов (9) збігається з такою для моделі Гоеля–Окумото (10):

$$\omega(t)\Big|_{s=0} = \alpha \cdot \beta \cdot e^{-\beta \cdot t}, \tag{9}$$

$$\omega(t) = a \cdot b \cdot e^{-b \cdot t}, \tag{10}$$

а при $s = 1$ – S-подібної моделі (вирази (11) та (12) відповідно):

$$\omega(t)\Big|_{s=1} = \alpha \cdot \beta^2 \cdot t \cdot e^{-\beta \cdot t}, \tag{11}$$

$$\omega(t) = \alpha \cdot \beta^2 \cdot t \cdot e^{-\beta \cdot t}. \tag{12}$$

Аналіз отриманих результатів та визначення інтервалів значень індексу складності

Для моделі Гоеля–Окумото параметр a має зміст очікуваної загальної кількості помилок, яку буде виявлено при $t \to \infty$, а параметр b – коефіцієнт виявлення помилок на один дефект (показує швидкість виявлення помилок за одиницю часу на один дефект, присутній у програмі) [5].

У моделі з індексом складності параметр α характеризує загальну кількість помилок у програмному продукті, але на відміну від моделі Гоеля–Окумото не дорівнює їй (очікувана загальна кількість помилок, яка буде виявлена при $t \to \infty$, становить $\alpha \cdot s \cdot \Gamma(s)$, тобто містить модифікатор, який залежить від складності продукту); параметр β так само має розмірність, обернену до розмірності часу і характеризує швидкість виявлення помилок, віднесену до одного дефекту програми; параметр s є показником складності програмного продукту [14] і визначається деякою комплексною метрикою ПЗ. Крім того, з рис. 1 видно, що на тривалість процесу виявлення помилок впливає не тільки параметр β, але й параметр s, що відповідає практиці проведення тестування та виявлення помилок для програмних продуктів різної величини та складності.

Максимальне значення параметра потоку відмов в моделі Гоеля–Окумото становить (за формулою (10)) $a \cdot b$. Порівнявши цей вираз з виразом (5), можна припустити, що максимальне значення параметра потоку відмов (так само, як і загальна кількість помилок) у випадку узагальненої моделі з індексом складності містить базове значення (значення при $s = 0$, що співпадає з моделлю Гоеля–Окумото), помножене на деякий модифікатор (у цьому випадку $s^s \cdot e^{-s}$), який залежить від складності програмного продукту. Припустивши, що значення такого модифікатора не може бути більшим за одиницю, отримаємо граничні значення параметра s для узагальненої моделі надійності ПЗ з індексом складності – значення параметра s не може перевищувати числа Ейлера e. Таке припущення можна обґрунтувати, взявши до уваги, що у випадку базового значення усі помилки виявляються з максимальною інтенсивністю при $t = 0$ і за мінімальний загальний час (див. рис. 1, крива 1), а збільшення тривалості виявлення тієї самої

кількості помилок не може привести до більшого миттєвого значення параметра потоку відмов у довільний момент часу. Відтак, з рівняння (5) отримаємо максимальне значення параметра потоку відмов для трьох основних значень індексу складності (0 – мінімальне значення; 1 – значення, що відповідає мінімуму параметра потоку відмов; e – максимальне значення):

$$\omega_{\max}(\alpha,\beta,s)\Big|_{s=0} = a\cdot\beta,$$

$$\omega_{\max}(\alpha,\beta,s)\Big|_{s=1} = \frac{a\cdot\beta}{e},$$

$$\omega_{\max}(\alpha,\beta,s)\Big|_{s=e} = a\cdot\beta.$$

Залежність нормованого максимального значення параметра потоку відмов від індексу складності програмного продукту наведено на рис. 2.

Діапазони значень індексу складності програмного продукту визначимо з умови рівності площі під кривою максимального значення параметра потоку відмов (рис. 2) в кожному діапазоні. Шляхом ітеративного підбору верхньої межі інтегралу, записаного на основі виразу (5), можна визначити такі діапазони індексу складності програмного продукту (рис. 2):

1. $s \in (0; 0{,}66)$ ПЗ можна вважати нескладним;
2. $s \in [0{,}66; 1{,}6)$ ПЗ можна вважати середньої складності;
3. $s \in [1{,}6; 2{,}28)$ ПЗ можна вважати складним;
4. $s \in [2{,}28; e]$ – можна вважати дуже складним.

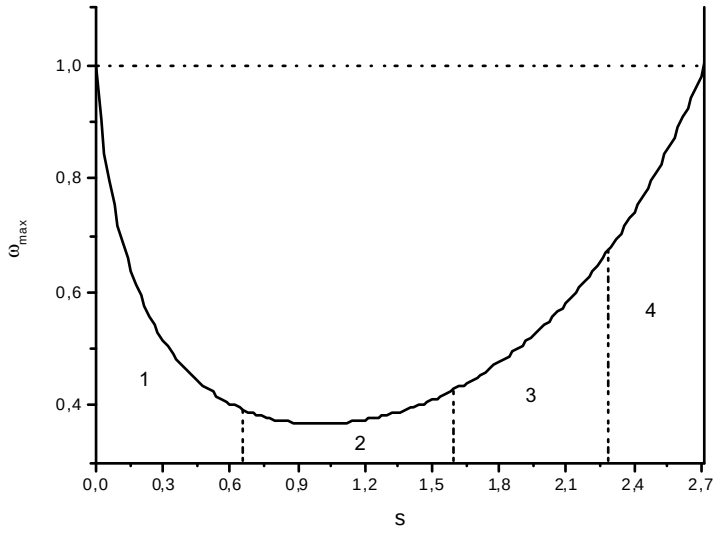

Рис. 2. Залежність нормованого максимального значення параметра потоку відмов від індексу складності s

Поведінку параметра потоку відмов можна пояснити так. Зі збільшенням ступеня складності програмного продукту максимальний параметр потоку відмов зменшується за абсолютним значенням (рис. 2) та зсувається в часі (рис. 1), оскільки тестувальникам потрібно все більше часу на тестування усіх функцій продукту. Для проектів середньої складності значення параметра потоку відмов майже не залежить від складності і проходить через точку мінімуму, при цьому далі зсуваючись в бік більших значень часової шкали. Зі збільшенням індексу складності максимальне значення параметра потоку відмов продовжує зростати і при $s = e$ за абсолютним значенням співпадає з таким для випадку $s = 0$. На думку автора, це може бути пояснено зростанням кількості тестувальників, більшою модульністю, яка повною мірою використовує переваги об'єктно-орієнтованого підходу тощо. При цьому загальна тривалість процесу виявлення помилок

монотонно зростає (див. рис. 1), а максимум параметра потоку відмов настає все пізніше зі зростанням складності продукту, що відповідає якісній картині процесу тестування промислового ПЗ. Однак питання встановлення числових значень діапазонів індексу складності потребує подальшого уточнення на основі даних тестування програмних продуктів різного ступеня складності та порівняння їх метрик коду.

Висновки

Досліджено поведінку функції інтенсивності відмов моделі надійності ПЗ з індексом складності залежно від параметрів моделі, що дало змогу встановити діапазони значень цього індексу та пояснити поведінку функції виявлення помилок залежно від складності програмного продукту. Показано, що модель з індексом складності адекватніше описує тестування програмного продукту, причому залежність характеру виявлення помилок при тестуванні від складності програмного продукту має нелінійний характер і володіє мінімумом при значенні індексу складності $s = 1$. Показано взаємозв'язок виразу для інтенсивності відмов узагальненої моделі надійності з індексом складності з найпоширенішими моделями надійності ПЗ, які є частковими випадками цієї моделі. Показано, що вирази для різних показників узагальненої моделі (загальна кількість помилок, тривалість тестування тощо) відрізняються від базових виразів модифікаторами, які містять індекс складності і нелінійно залежать від цього індексу.

1. Hoang Pham, Michelle Pham "Software Reliability Models for Critical Applications" // EGG—2663 Technical Report (1991). Idaho National Engineering Laboratory, EG&G Idaho Inc. – 98 p. 2. Tariq Hussain Sheakh, S.M.K. Quadri, and VijayPal Singh "A Study of Analytically Improving the Reliability of Software" // International Journal of Research and Reviews in Computer Science, Vol. 3, No. 1, February 2012, P. 1404–1406. 3. Hoang Pham "System software reliability". – Springer-Verlag London Limited, 2006. – 440 p. 4. Половко А.М., Гуров С.В. Основы теории надежности. – СПб.: БХВ-Петербург, 2008. – 704 с. 5. Goel, A.L. Software reliability models: assumptions, limitations, and applicability. // IEEE Transactions on software engineering. – Vol. SE-11 (1985), No 12. – pp. 1411-1423. 6. Cobra Rahmani, Azad Azadmanesh "Exploitation of Quantitative Approaches to Software Reliability". – University of Nebraska at Omaha, 2008. – 32 p. 7. M. Palviainen, A. Evesti, E. Ovaska "The reliability estimation, prediction and measuring of component-based software" // The Journal of Systems and Software, Vol. 84 (2011), pp. 1054–1070. 8. 3. Гнеденко Б. В., Беляев Ю. К., Соловьев А. Д. Математические методы в теории надежности. Основные характеристики надежности и их статистический анализ. – М.: Наука, 1965. – 524 с. 9. Musa J. D. A theory of software reliability and its application // IEEE Transactions on Software Engineering. – SE-1(3). – 1975. – P. 312–327. 10. Goel A. L., Okumoto K. Time-Dependent Error-Detection Rate Model for Software and other Performance Measures // IEEE Transactions on Reliability. – Vol. R-28. – No. 3. – 1979. – P. 206–211. 11. Yamada S., Ohba M., Osaki S. S-shaped reliability growth modeling for software error detection // IEEE Transactions on Reliability. – Vol. R-32. – No.5. – 1983. – P. 475–478. 12. Huang X.Z. The hypergeometric distribution model for predicting the reliability of software // Microelectronics and Reliability, Vol. 24 (1984), No. 1, pp. 11–20. 13. Тимошенко Ю. О., Дідковська М. В. Узагальнена модель негомогенного пуасонівського процесу для оцінювання надійності програмного забезпечення // Проблеми програмування. – № 2–3. – 2004. – С. 480–489. 14. Чабанюк Я. М., Яковина В. С., Федасюк Д. В., Сенів М. М., Хімка У. Т. Побудова і дослідження моделі надійності програмного забезпечення з індексом величини проекту // Інженерія програмного забезпечення. – № 1 (2010). – С. 24–29. 15. Яковина В. С., Сенів М. М., Чабанюк Я. М., Федасюк Д. В., Хімка У. Т. Критерій достатності процесу тестування програмного забезпечення // Вісник Нац. ун-ту "Львівська політехніка" Комп'ютерні науки та інформаційні технології. – № 672 (2010). – С. 346–358. 16. Сенів М. М., Федасюк Д. В., Чабанюк Я. М., Яковина В. С. Аналіз використання моделі надійності програмного забезпечення з динамічним показником складності проекту протягом життєвого циклу // Комп'ютерні технології друкарства. – № 24 (2010). – С. 111–126. 17. Сенів М.,

Яковина В., Чабанюк Я., Федасюк Д. Метод оцінювання та прогнозування надійності програмного забезпечення на основі моделі з динамічним показником величини проекту // Комп'ютинг, Т. 10 (2011), Вип. 2. – С. 97–107. 18. A.L. Goel A guidebook for software reliability assessment // Rep. RADCTR-83-176, Aug. 1983. 19. ГОСТ 27.002-89 Надежность в технике. Основные понятия. Термины и определения.

УДК 004.415

Elena Nyemkova
Lviv Institute of Banking the University of Banking of the NBU

DATA PROTECTION OF BIOMETRIC AUTHENTICATION FOR REMOTE ACCESS TO A BANK ACCOUNT

This article is devoted to the hash function that provides closure of biometrics in information networks for remote access to client account. There is proposed hash function which is described as a table of numbers. Verification of sustainability of cryptographic hash functions is performed. The session key for the hash is determined by the sequence of biometric data. The protocol of mutual authentication of a client and the server of payment system is given. The number of client's safe applications to ATM is estimated.

Key words: biometric template, mutual authentication protocol, transaction, unauthorized debiting, hash function.

ЗАХИСТ ДАНИХ ПРИ БІОМЕТРИЧНІЙ АВТЕНТИФІКАЦІЇ ПІД ЧАС ВІДДАЛЕНОГО ДОСТУПУ ДО БАНКІВСЬКОГО РАХУНКУ

Обґрунтовано вимоги до хеш-функції, яка забезпечує закриття біометричних даних в інформаційних мережах при віддаленому доступі клієнта до рахунку. Запропонована хеш-функція, яка описується у вигляді таблиці чисел. Наведено перевірку криптографічної стійкості хеш-функції. Сеансовий ключ для хеша визначається послідовністю біометричних даних. Запропоновано протокол взаємної автентифікації клієнта і сервера платіжної системи. Наведено оцінку кількості безпечних звернень клієнта до платіжної системи.

Ключові слова: біометричний шаблон, протокол взаємної автентифікації, трансакція, несанкціоноване списання коштів, хеш-функція.

Introduction

Biometric authentication in a computer network is one of the promising areas of information security. Biometric authentication is more secure than passwords and identity documents. It is also the only way to recognize fraud. Currently, biometric systems are not completely reliable in terms of recognition errors, as well as in terms of preservation and transmission of biometric templates online. These difficulties are a barrier to the widespread use of biometric systems in the real world.

Specialists consider two types of attacks in the context of biometric authentication [1–2]. They are forgery attack and data leakage from the database templates. Forgery attack can occur because currently there is no unambiguous method of matching fixed biometric data to their respective owners. Verification