УДК 004.056 (043.2)

С.Г. Семенов[1], Б.М. Резанов[1], В.В. Босько[2]

[1] *Національний технічний університет «Харківський політехнічний інститут», Харків*
[2] *Кіровоградський національний технічний університет, Кіровоград*

ПРОЦЕДУРИ ДВОХФАКТОРНОЇ АУТЕНТИФІКАЦІЇ ДЛЯ ЗАБЕЗПЕЧЕННЯ ЗАХИСТУ ЕЛЕКТРОННОГО ЦИФРОВОГО ПІДПИСУ

У статті проведено аналіз механізмів захисту електронного цифрового підпису. Визначено, що однім зі шляхів забезпечення захисту електронного цифрового підпису є використання процедур двохфакторної аутентифікації. Розроблено процедури двохфакторної аутентифікації для забезпечення захисту електронного цифрового підпису та проілюстровано за допомогою мови моделювання UML. Розроблено та проілюстровано загальну архітектуру програмної системи. Визначено доцільність подальшої реалізації наведених процедур.

Ключеві слова: *електронний цифровий підпис, двохфакторна аутентифікація, UML.*

Постановка задачі

Для створення ефективного апаратно-програмного забезпечення інформаційної безпеки (ІБ) будь-якої системи необхідно поєднання існуючих механізмів захисту різних типів, що забезпечують недопущення певних, характерних для цієї системи видів порушень ІБ, оперативне виявлення чинників порушень ІБ а також реагування на порушення ІБ, включаючи ліквідацію наслідків . Одним з механізмів, що вирішують широкий спектр завдань превентивного інформаційного захисту є електронний цифровий підпис (ЕЦП), що представляє математичну схему, призначену для відображення дійсності електронних повідомлень, документів, транзакцій, а також аутентифікації джерела повідомлень [1-6]. Слід зауважити, що всі властивості, що характеризують якість ЕЦП (справжності, безвідмовності і т.і.) залежать від закритого ключа, який не повинен бути відкликаний до його використання. [1, 2].

Проведені дослідження і аналіз літератури [4, 5] показали, що всі криптосистеми, що функціонують на принципах використання відкритого / закритого ключа, повністю залежать від захищеності закритих ключів і даних, необхідних для їх формування. Відомо, що закритий ключ ЕЦП може зберігатися на комп'ютері користувача і бути захищений локальним паролем. Однак такий спосіб має ряд недоліків. Зокрема користувач повністю прив'язаний до комп'ютера при формуванні підпису, і в той же час безпека закритого ключа повністю залежить від безпеки самого комп'ютера.

Аналіз сучасних розробок в області аутентифікації [1-6] показав, що більш надійною альтернативою зберігання закритого ключа є смарт-карта. При цьому така смарт-карта повинна бути оснащена захистом від несанкціонованого доступу (НСД).

Проведені дослідження показали, що одним з ефективних механізмів захисту смарт-карт від несанкціонованого доступу є процедура двохфакторної аутентифікації.

Аналіз літератури [1-6] показав, що в даний час існує ряд сучасних розробок і практичних реалізацій протоколів, програмних продуктів і цифрових пристроїв, що виконують функції двохфакторної аутентифікації. Однак вони не позбавлені недоліків. Наприклад, в статті [1] автором проведено детальний аналіз недоліків реалізації процедур двохфакторної аутентифікації відомого інтернет-ресурсу - «Вконтакте», і наведено перелік уразливостей, що призводять до відключення другого фактору. Також одним з основних недоліків практично усіх подібних систем є відсутність контролю цілісності даних (відсутність механізмів контролю за підробкою), та складність мультиагентного адміністрування.

З іншого боку будь-яке ускладнення процедур двохфакторної аутентифікації призводить до значного збільшення вартості продуктів. Тому актуальною стає задача розробки і реалізації процедур двохфакторної аутентифікації на основі існуючих телекомунікаційних рішень. Це може дозволити знизити собівартість продукту і забезпечити якісний рівень захисту ЕЦП.

Основна частина

Для більш детального опису процедур двохфакторної аутентифікації скористаємось структурно-функціональною моделью [6] в основу якої положені основні принципи уніфікованої мови моделювання UML. UML – це сімейство графічних нотацій, в основі якого міститься єдина метамодель [6]. UML являє собою відкритий стандарт, який знаходиться під керуванням групи Object Management Group – відкритого консорціуму компаній. UML дозволяє описати моделі та сценарії взаємодії у системі з різних точок зору для подальшого втілення, наочно відобразити компоненти ситеми, учасників процесу та їх взаємодію.

© С.Г. Семенов, Б.М. Резанов, В.В. Босько

В системі присутні такі актори.

1. Ресурс – щось, чим володіє клієнт.

2. Клієнт – людина, яка реєструється в системі для того, щоб налаштувати багатофакторну аутентифікацію для захисту власного ресурсу від несанкціонованого доступу за допомогою проектованої системи. Клієнт є власником облікового запису і саме він вносить кошти для сплати послуг системи.

3. Адміністратор – людина, назначена клієнтом для управління системою аутентифікації на одному або кількох ресурсах.

4. Менеджер системи – абстрактна сутність, яка може керувати системою аутентифікації. Реальною реалізацією такої сутності є клієнт та адміністратор системи.

5. Користувач – людина, яка використовує ресурс клієнта, має аутентифікатор та використовує його для отримання доступу до запитуваного ресурсу.

В системі присутні наступні прецеденти.

1. Реєстрація клієнта в системі. Цей прецедент ініціюється клієнтом. Він забезпечує можливість отримати доступ до всієї функціональності системи.

2. Управління ресурсами клієнта. Ініціюється менеджером системи (адміністратором чи клієнтом). Дозволяє створити або видалити ресурс, дозволити або заборонити використання системи ресурсом клієнта, редагувати інформацію про ресурс.

3. Управління користувачами. Ініціюється менеджером системи. Дозволяє реєструвати та видаляти користувачів в системі, редагувати інформацію про них, блокувати або розблокувати доступ до свого ресурсу, назначати або видаляти токен користувача.

4. Запит доступу до ресурсу клієнта. Ініціюється користувачем. Для отримання доступу до ресурсів клієнта користувачеві необхідно пройти процедуру аутентифікації на сайті клієнта. Для цього йому необхідно заповнити форму аутентифікації на сайті клієнта.

5. Управління адміністраторами. Ініціюється клієнтом. Дозволяє додати або видалити, блокувати або розблокувати адміністратора в системі.

6. Запит на аутентифікацію користувача. Ініціюється ресурсом клієнта у той час, коли користувач намагається пройти процедуру аутентифікації. Дозволяє перевірити чи правильні дані ввів користувач для аутентифікації.

7. Перегляд статистики. Ініціюється менеджером системи. Дозволяє переглянути різноманітну інформацію про роботу і використання системи.

8. Поповнення рахунку. Ініціюється клієнтом. Дозволяє внести кошти для сплати послуг, що надає система клієнту.

Діаграма прецедентів системи приведена на рис. 1.

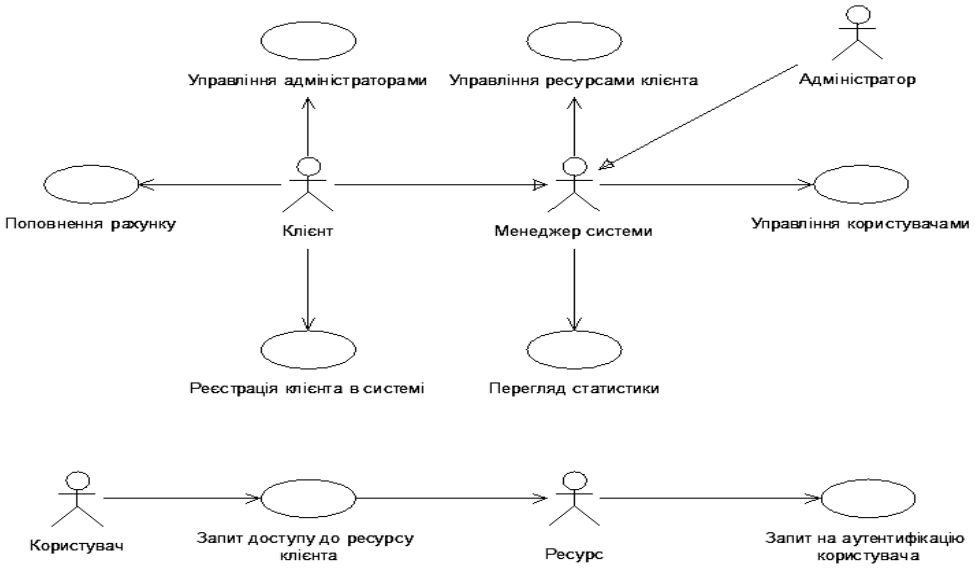

Рис. 1. Діаграма прецедентів програмної системи

Потоки подій та діаграми послідовностей програмної системи. Основна функція системи полягає в тому, що вона має дати відповідь, чи є користувач тим, за кого себе видає. Виходячи з цього, прецедент «Запит на аутентифікацію користувача» є досить важливим елементом моделі. Щоб краще його зрозуміти розглянемо потік подій цього прецеденту.

Передумови. Клієнт має бути зареєстрований в системі та мати доданий ресурс, який буде виконувати запит.

Головний потік. Прецедент починає виконуватись, коли ресурс клієнта робить запит на аутентифікацію користувача, що звернувся до нього. Перед цим ресурс клієнта отримав від користувача аутентифікаційні данні: логін, пароль, одноразовий пароль. Отримавши данні, ресурс клієнта формує запит та передає його до проектованої системи. Окрім аутентифікаційних даних користувача ресурс додає до запиту власні ідентифікаційні записи для того, щоб система могла успішно його ідентифікувати.

Система отримує інформацію і перевіряє вірність ідентифікатора ресурсу (E-1), після чого перевіряє правильність логіна та пароля користувача. Перевірка логіна та пароля містить в собі перевірку чи має даний користувач доступ до даного ресурсу (E-2). Після цього система генерує одноразовий пароль для даного користувача (S-1) та перевіряє, чи відповідає згенерований пароль тому, що був переданий ресурсом від користувача (E-3). В кінці система перевіряє обмеження, накладені на цього користувача (E-4), формує відповідь (S-2) і відправляє її ресурсу, що виконав запит. На основі цієї відповіді ресурс вирішує що робити з користувачем далі.

Підпотоки. Підпоток S-1 відповідає за генерацію одноразового пароля. Система завантажує необхідні для генерації дані з БД, визначає, по якому алгоритму має відбуватись генерація і, діючи згідно цього алгоритму, визначає поточний одноразовий пароль. Підпоток S-2 формує відповідь на запит про аутентифікацію користувача. Формуючи відповідь, система відповідає на основне питання «Чи може користувач мати доступ до данного ресурсу?». Перевірка даних іде поетапно. Кожен наступний етап виконується тільки в тому разі, коли всі попередні етапи успішно завершені. Відповідно, якщо якийсь з етапів завершується невдало, то не має сенсу виконувати наступні перевірки, система відразу переходить до формування відповіді про неможливість доступу користувача до данного ресурсу.

Якщо всі етапи успішно пройдені, то це свідчить про те, що отримані відомості є вірними, відповідно формується відповідь про те, що користувач може отримувати доступ до ресурсу.

Альтернативні потоки. E-1: отримані невірні ідентифікаційні дані ресурсу. Система виконує підпоток S-2, де буде сформована відповідь про заборону доступу з причини неможливої ідентифікації ресурсу. E-2: невірний логін або пароль. Якщо користувач з таким логіном не існує, або цей користувач не має доступу до запитуваного ресурсу, або пароль не є вірним, то система виконує підпоток S-2, де буде сформована відповідь про заборону доступу з указанням причини. E-3: невірний одноразовий пароль. Система виконує підпоток S-2, де буде сформована відповідь із забороною доступу по причині невірного одноразового пароля. E-4: користувач не пройшов накладених на нього обмежень. Система дозволяє блокувати користувачів по заданим параметрам: географічній зоні, з якої входить користувач та часу, в який вібувається запит. Ці параметри визначаються менеджером для кожного ресурсу. Якщо користувач не проходить накладених обмежень, то система виконує підпоток S-2, де в якості причини відмови буде вказано, яке саме обмеження не пройшов користувач.

Якщо на якомусь етапі стає зрозуміло, що подальша перевірка не має змісту, то система відразу переходить до формування відповіді про неможливість доступу даного користувача до даного ресурсу. У відповіді також буде вказана причина, з якої користувач не пройшов перевірку. Це допоможе споживачам системи більш зручно спостерігати за її роботою та вчасно реагувати на можливі проблеми.

Для того, щоб краще зрозуміти та наочно відобразити як проходять процеси в програмній системі використовують діграми послідовностей. Діаграма послідовності основного потоку подій прецедента «Запит на аутентифікацію користувача» представлена на рис. 2.

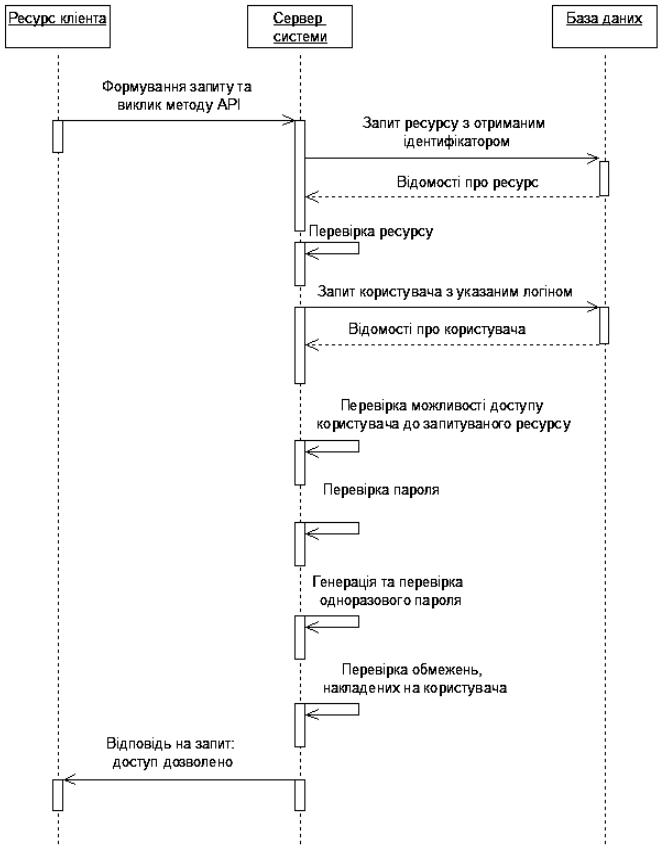

Рис. 2. Діаграма послідовності основного потоку подій для прецедента «Запит на аутентифікацію користувача»

Діаграма діяльності та діаграма станів програмної системи. Для опису інформаційних процесів, що відбуваються в системі, скористаємось діаграмою діяльності. Даний тип діаграм дозволяє відобразити послідовність передачі повідомлень між об'єктами. Цей тип діаграми не акцентує увагу на конкретній взаємодії, головний акцент приділяється послідовності прийому та передачі повідомлень. Розглянемо, що відбувається у прецеденті «Управління адміністраторами». Як відомо, адміністраторами може керувати тільки клієнт, тому кожен раз, коли аутентифікований суб'єкт намагається отримати доступ до сторінки керування адміністраторами,

необхідно виконати процес авторизації – перевірити, чи має аутентифікований суб'єкт право на перегляд цієї частини системи. Детальний процес зображено на рис. 3.

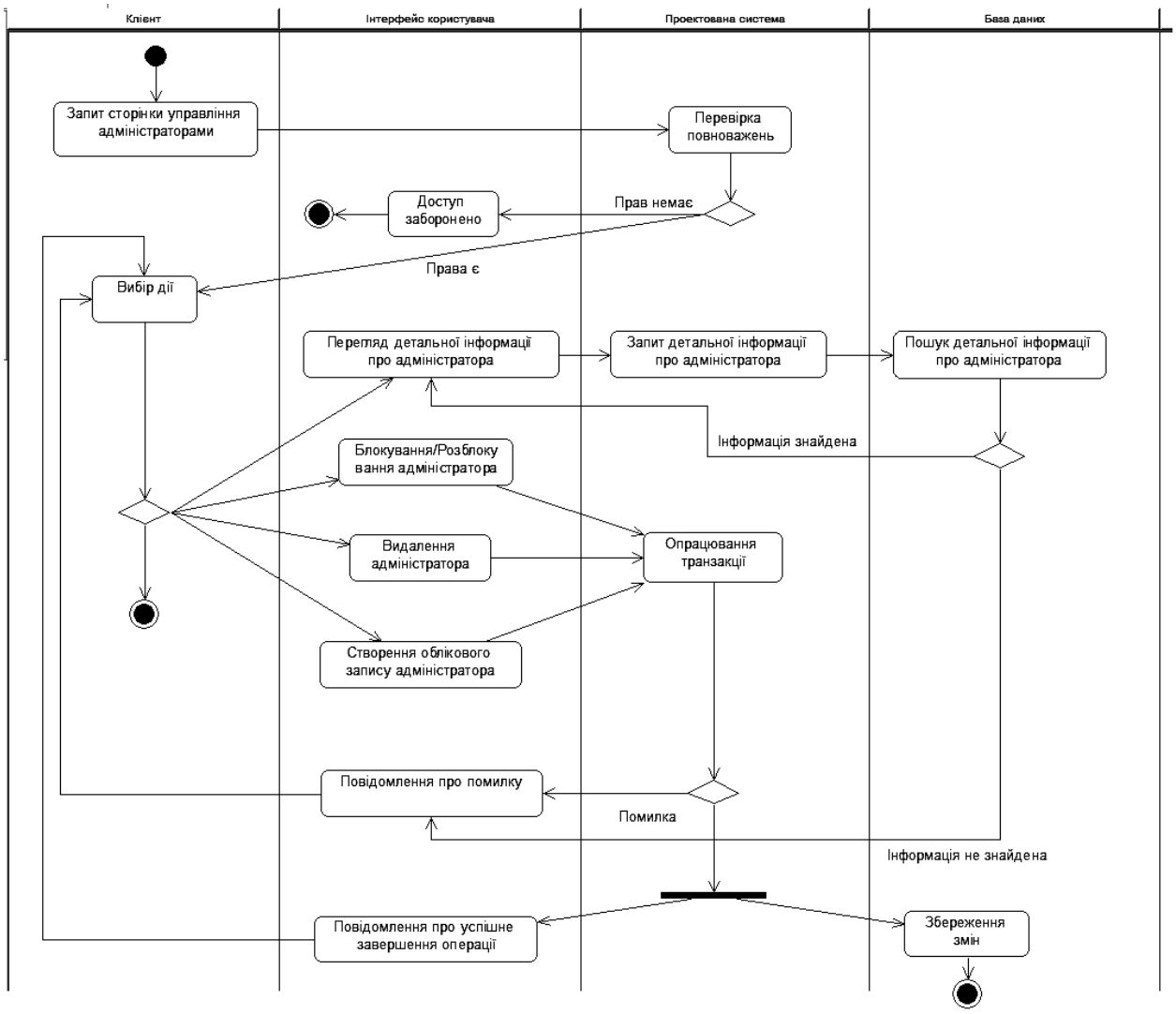

Рис. 3. Діаграма діяльності прецедента «Управління адміністраторами»

Обліковий запис адміністратора в системі може знаходитись у різних станах. При додаванні нового облікового запису адміністратора клієнтом створюється непідтверджений обліковий запис. Після того, як людина вказана у якості адміністратора, завершить реєстрацію створюється повноцінний обліковий запис. За бажанням клієнта адміністратор може бути заблокований або розблокований. Перемикання стану адміністратора відбувається шляхом перемикання стану чекбоксу на сторінці перегляду інформації про адміністраторів. Нормальний стан для адміністратора «Активний», це значить, що він має доступ до управління системою аутенифікації на визначеному клієнтом ресурсі. Якщо адміністратор знаходиться у стані «Заблокований», то він не має доступу до системи і не може виконувати свої функції. Як бачимо, робота адміністратора в системі завершується, коли клієнт видаляє обліковий запис адміністратора. При цьому не важливо в якому стані знаходився обліковий запис: в активному чи заблокованому.

Загальна схема роботи програмної системи. Виходячи з поставлених вимог та отриманих моделей поведінки зобразимо загальну архітектурну схему програмної системи на рис. 4.

Для ефективної роботи користувачів системи призначений компонент «Advanced GUI» – вдосконалений графічний інтерфейс користувача. Саме через нього користувач отримує доступ до всіх функцій системи. Основна вимого до графічного інтерфейсу – він повинен бути зручний у користуванні. Від цього напряму залежить сприйняття системи користувачем.

Зовнішні системи, які позначені «External system», взаємодіють з проектованою системою через набір методів API. Обмін між системами відбувається через JavaScript Object Notation (далі – JSON) або Simple Object Access Protocol (далі – SOAP). Для полегшення інтеграції між системами призначена спеціальна бібліотека, яка на рисунку позначена літерами «LB».

Вона завантажується клієнтом та встановлюється на власному ресурсі.

Такий підхід дозволяє зменшити витрати часу на інтеграцію сервісу. Також у системі присутні модулі для відправлення повідомлень (SMS Transport Module), модуль ліцензування (License Module), та модуль нарахування оплати за послуги системи (Billing Module).

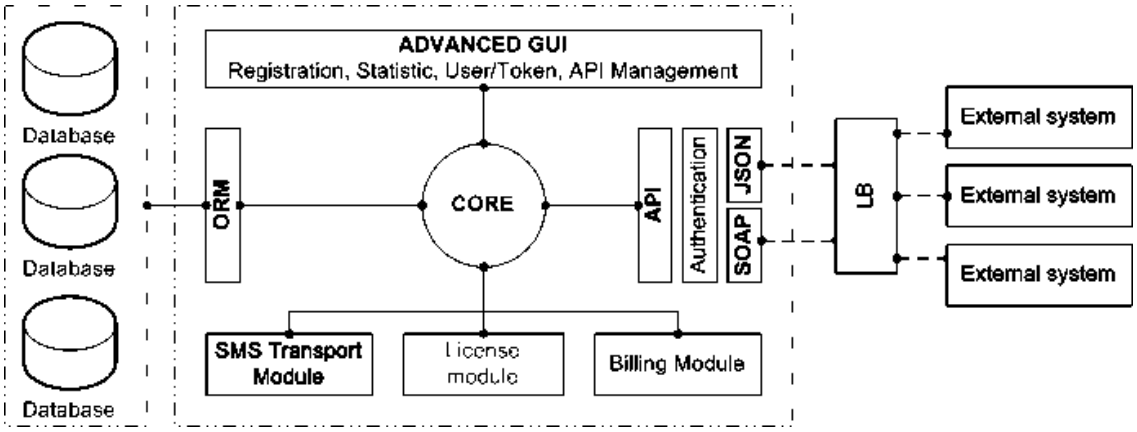

Рис. 4. Загальна архітектура програмної системи

Висновки

Таким чином, у статті запропоновано процедури двохфакторної аутентифікації для забезпечення захисту ЕЦП.

Ці процедури використовують стандартні підходи щодо обміну даними, але в той же час унеможливлюють загрозу відключення другого фактору аутентифікації та забезпечують захист механізму ЕЦП.

Відміною рисою запропонованих процедур є контрольованість цілісності даних виконуваних транзакцій (забезпечення захисту від змін або підробки даних).

Крім того, на відміну від аналогічних систем, розроблена система кроссплатформенна, та дозволяє мати одному адміністратору кілька ресурсів, з різними користувачами, настройками фільтрації (мультиагентність адміністрування).

У подальшому розроблені процедури дають можливість програмно-апаратної реалізації пристроїв двохфакторної аутентифікації.

Список літератури

1. Двойная аутентификация [Електронний ресурс]. – Режим доступу: https://habrahabr.ru/post/315862/.

2. Закон Украины Об электронной цифровой подписи [Електронний ресурс]. – Режим доступу: http://www.buhgalteria.com.ua/Hit.html?id=496

3. Идентификация и аутентификация с помощью ЭЦП [Електронний ресурс]. – Режим доступу: http://megalektsii.ru/s44162t1.html

4. Конявская С. Плюсы и минусы двухфакторной аутентификации [Електронний ресурс] / С. Конявская. – Режим доступу http://www.itsec.ru/articles2/Oborandteh/plyusy_i_minusy_dvuhfakt_autentifikacii.

5. Семенов С.Г. Протоколи захисту інформації у комп'ютерних системах та мережах / С.Г. Семенов, О.О. Кузнецов. – Х.: ХНУРЕ, 2009. – 184 с.

6. Хассан Гома UML-проектирование систем реального времени параллельных и распределенных приложений / Гома Хассан. – М.: ДМК Пресс 2011. – 704 с.

Надійшла до редколегії 18.05.2016

Рецензент: д-р техн. наук, проф. О.О. Можаєв, Національний технічний університет «ХПІ», Харків.

ПРОЦЕДУРЫ ДВУХФАКТОРНОЙ АУТЕНТИФИКАЦИИ ДЛЯ ОБЕСПЕЧЕНИЯ ЗАЩИТЫ ЭЛЕКТРОННОЙ ЦИФРОВОЙ ПОДПИСИ

С.Г. Семенов, Б.Н. Резанов, В.В. Боско

В статье проведен анализ механизмов защиты электронной цифровой подписи. Определено, что одним из путей обеспечения защиты электронной цифровой подписи является использование процедур двухфакторной аутентификации. Разработаны процедуры двухфакторной аутентификации для обеспечения защиты электронной цифровой подписи и проиллюстрировано с помощью языка моделирования UML. Разработан и проиллюстрировано общую архитектуру программной системы. Определена целесообразность дальнейшей реализации указанных процедур.

Ключевые слова: *электронная цифровая подпись, двухфакторная аутентификация, UML.*

PROCEDURES TWO-FACTOR AUTHENTICATION TO PROTECT DIGITAL SIGNATURE

S.G. Semenov, B.M. Rezanova V.V. Bos'ko

The article analyzes the mechanisms to protect digital signature. Determined that one of the ways to protect digital signature treatments is the use of two-factor authentication. The procedure of two-factor authentication to protect digital signature and illustrated using modeling language UML. Designed and illustrated the overall architecture of software systems. Determined expediency further implementation of these procedures.

Keywords: *digital signature, two-factor authentication, UML.*