УДК 004.056.5

С.А. Гончар

Криптографічний протокол з часовим розкриттям у мережі рівноправних вузлів

Рассмотрены основные положения криптографии с временным раскрытием. Показано, что наиболее перспективное направление развития таких систем – эволюция методов с использованием третьей стороны – сети равноправных узлов и *bitcoin*-схем. Приведены основные положения представленного протокола: создание числового ключа, отправка сообщения, создание новых блоков и транзакций, а также механизм расшифровки данных.

Ключевые слова: криптография с временным раскрытием, *bitcoin*, временные замки, временные сервера.

Розглянуто основні положення криптографії з часовим розкриттям. Показано, що найбільш перспективним напрямом розвитку таких систем є еволюція методів з використанням третьої сторони – мережі рівноправних вузлів та *bitcoin*-схем. Наведено основні положення представленого протоколу: створення числового ключа, відправка повідомлення, створення нових блоків та транзакцій, а також механізм розшифрування даних.

Ключові слова: криптографія з часовим розкриттям, *bitcoin*, часові замки, часові сервери.

Вступ. Розвиток криптографії дозволив ввести в ужиток такі поняття, як електронні платежі, електронні гроші, електронний підпис та ін. Саме криптографія забезпечує необхідний рівень безпеки учасників таких електронних засобів.

Одним з найцікавіших, але не досить розповсюдженим напрямом у криптографії є так звані «листи у майбутнє». Криптографічні системи такого типу мають в іноземній літературі досить розгалужену термінологію: *time-released crypto* [1], *timed-release encryption* [2], *timed commitments* [3], *time-lapse cryptography* [4], *time-specific encryption* [5]. Загальновживаного українського еквіваленту цих термінів поки що не введено, але, на думку автора, найбільш доцільним буде використання словосполучення «криптографія з часовим розкриттям», що має на увазі сукупність методів, які дозволяють зашифрувати дані в такий спосіб, щоб забезпечити їх розшифрування після закінчення заздалегідь визначеного часу або у заздалегідь визначений час та виключити можливість дострокового доступу до зашифрованих даних.

Постановка задачі

Відповідно до сучасного стану досліджень в межах систем, що розглядаються, виділяють два напрями:

- використання *математичних замків з часовим механізмом* – являють собою обчислювальні задачі, які не можуть бути розв'язані без обчислення на комп'ютері за певний проміжок часу;
- використання третьої сторони для зберігання певної інформації, необхідної в подальшому розшифровуванні повідомлення.

Сутність та методи першого напряму з моменту його запропонування практично не змінились і можливий розвиток *математичних замків з часовим механізмом* пов'язаний у першу чергу з новими конкретними задачами.

Другий напрям має більш розгалужений еволюційний процес, але до широкого розповсюдження шифрування на основі особистих даних (identity based encryption) [6] здійснювалися лише спроби розвинути оригінальну ідею авторів роботи [1]. Зокрема, більшість дослідників присвятила свої роботи питанням мінімізації взаємодії користувача і сервера, гарантуванню поширення ключа і анонімності користувача.

Протокол з часовим розкриттям у мережі рівноправних вузлів

Для криптографії з часовим розкриттям на основі серверів (третьої сторони) за останні десятиліття здійснено багато як практичних, так і теоретичних напрацювань.

На рис. 1 подано схему еволюції основних протоколів.

Прорив у питаннях криптографії з часовим розкриттям відбувся після введення шифрування на основі особистих даних. Основою такого типу шифрування є шифрування відправником повідомлення M у відповідності до конкретного одержувача. Формально таке шифрування можна визначити як кортеж з чотирьох рандомізованих алгоритмів [7]:

• *Встановлення* (1^k). При вході на цей алгоритм параметра безпеки 1^k генеруються відкриті параметри π_{IBE}, що містять у собі хеш-функції та повідомлення M. Додатково генерується основний секрет δ_{IBE}, який залишається конфіденційним завдяки серверу.

• *Отримання* $(\pi_{IBE}, \delta_{IBE}, I)$. Маючи на вході відкриті параметри π_{IBE}, основний секрет δ_{IBE} та характеристики особистості I, видають на вихід закритий ключ sk_I.

• *Шифрування* (π_{IBE}, I, M). Генерує шифротекст c на основі вхідних параметрів π_{IBE}, I, M.

• *Дешифрування* $(\pi_{IBE}, sk_I, \hat{c})$. Видає на вихід оригінальне повідомлення відповідно до $\hat{c}$ або повідомлення про помилку.

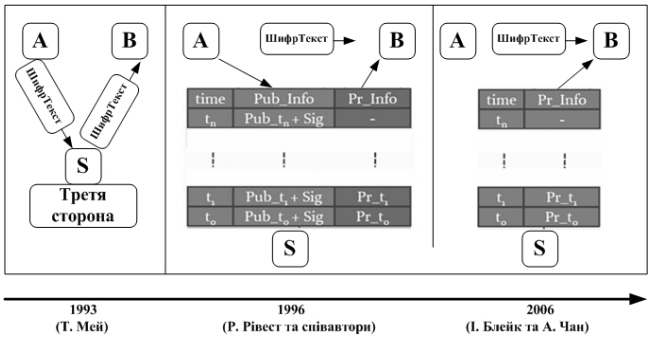

Рис. 1. Схеми шифрування з часовим розкриттям різних авторів

Починаючи з 2003 р., були запропоновані різні протоколи, які базуються на квадратичних залишках і на властивостях білінійного спарювання (відображення) в групах еліптичних кривих.

Існує досить багато напрацювань з використанням шифрування на основі особистих даних, тому розглянемо лише основні з них.

Автори [8] відштовхувалися від схеми, запропонованої І. Блайком та А. Чаном, в якій сервер не взаємодіє з відправником та одержувачем. Це досягається шляхом періодичної генерації специфічних часових «лазівок», які дозволяють виконувати дешифровку тексту, зашифрованого «на майбутнє» або «лазівок» для цифрового підпису. Дана концепція отримала назву *шифрування з часовим розкриттям без взаємодії*. У запропонованій схемі використовуються групи білінійних відображень і припускається, що зашифрований текст завжди містить інформацію про час розкриття, яка подається в кінці повідомлення. Крім того, запропонована схема може бути використана за необхідності розкриття тексту в разі настання певної події.

Дуже докладно питання створення криптосистеми з часовим розкриттям шляхом шифрування на основі особистих даних розглянуто в роботі [5]. Свій метод автори назвали *шифруванням*, *визначеним у часі* (*time-specific encryption*). Згідно з цим методом часовий сервер розсилає так званий *постійний у часі* ключ на початку кожного часового інтервалу. Цей ключ доступний для всіх користувачів та містить у собі характеристику часу t. Відправник може визначити будь-який інтервал часу у процесі шифрування, а одержувач може відновити оригінальне повідомлення тільки якщо має *постійний у часі* ключ, який відповідає часу у заданому відправником інтервалі. При цьому одержувачі додатково забезпечуються закритими ключами та або відкритими ключами, або параметрами на основі особистих даних, а дешифрування потребує використання як закритого ключа, так і відповідного *постійного у часі* ключа. Це забезпечує захист проти нечесного часового серверу та можливість дешифровки тексту лише для обраної групи користувачів.

Розвиток криптографічного інструментарію завжди дає шляхи розвитку існуючих підходів. Очевидно, що можливим і бажаним напрямом такого розвитку може бути створення систем, які не використовують сервери як такі. Зокрема, перспективним може бути використання *bitcoin*-схеми, яка базується на основі мережі рівноправних вузлів (*peer-to-peer*). Кожен вузол в такій мережі є одночасно і сервером, і клієнтом, тобто представляє деяку інформацію і звертається до інших вузлів. Координація так само виконується всіма учасниками одночасно завдяки розгалуженому зберіганню метаінформації усіма учасниками мережі [9].

В *bitcoin*-системі вся інформація про транзакції зберігається в ланцюгу блоків. Кожен блок містить заголовки і список транзакцій. Заголо-

вок складається з декількох властивостей, серед яких є також хеш попереднього блоку. Отже, весь ланцюг блоків зберігає всі транзакції за весь час роботи *bitcoin*. В останніх версіях системи весь ланцюг блоків зберігається повністю кожним клієнтом, що робить цю систему децентралізованою [10].

Пропонується модифікувати *bitcoin*-схему у такий спосіб, щоб мати змогу відправляти користувачам повідомлення, яке буде відкрито в заздалегідь встановлений відправником час, базуючись на основних принципах *bitcoin*-схеми. Це, по-перше, дасть змогу відмовитись від використання серверу.

У такій мережі вся інформація буде зберігатись в ланцюжку блоків. Кожен блок буде містити заголовок і список транзакцій. Заголовок складатиметься з деяких властивостей, серед яких буде також хеш на попередній блок. Отже дані про обмін будь-якою інформацією в запропонованій мережі зберігатимуться в ланцюжку блоків упродовж усього часу роботи.

Зазначимо одну важливу властивість даної схеми: кожен блок перевірятиметься учасниками на коректність. Маємо на увазі, що перед тим як певний блок отримає певний учасник, інші учасники виконають верифікацію цього блоку за певними параметрами, тим самим підтверджуючи цю пересилку, або ж скасують її. Це гарантуватиме, що ніхто з учасників не зможе відправити некоректну інформацію або якісь інші дані.

Розглянемо вміст блоку запропонованої схеми більш детально (рис. 2). Блок складатиметься з наступних параметрів:

- ***hash*** – хеш заголовку блоку;
- ***prev_block*** – хеш попереднього блоку в ланцюжку;
- ***transaction_hash_lis***t – список хешів транзакцій блоку;
- ***transaction_size*** – кількість транзакцій.

Транзакція в свою чергу складатиметься з наступних параметрів:

- ***hash*** – хеш усієї транзакції;
- ***prev_hash*** – хеш попередньої транзакції;
- ***pending_hash*** – хеш транзакції відкриття;
- ***transaction_type*** – тип транзакції. Доступними будуть такі типи транзакцій: *request_key*, *send_key*, *send_message*, *send_generated_part_key*, *send_part_key*;
- ***value*** – дані, які пересилаються;
- ***time*** – час відкриття.

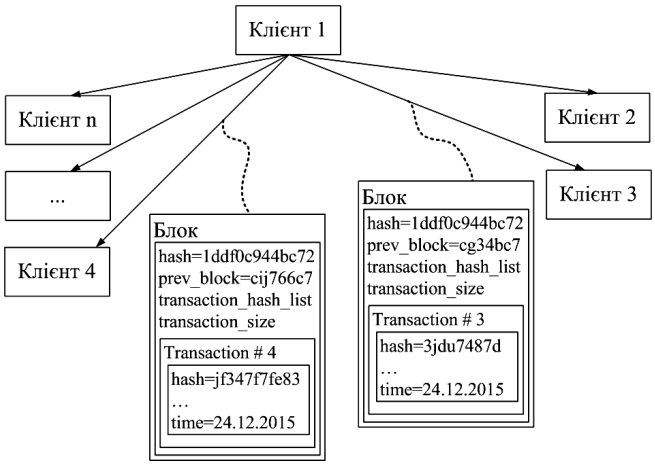

Рис. 2. Схема мережі рівноправних вузлів для реалізації відправки повідомлень з часовим розкриттям

Розглянемо схему відправки повідомлення учасником 1 учаснику 2, коли в певний момент часу в мережі перебувають n учасників. У кожному блоці та транзакції автоматично додаватимуться хеш та попередній хеш, а тому в подальшому це не описується.

1. Створюються сесійні параметри g та m – великі числа, а число m – просте та найбільший спільний дільник (НСД) $(m, n) = 1$.

2. Учасник 1 відправляє запит до учасника 2, щоб той відправив йому в зашифрованому вигляді свій сесійний ключ. Відправка цього запиту виконується через формування нового блоку та нової транзакції для учасника 2. Тип транзакції в цьому випадку – *request_key*. Відправка блоку виконується усім учасникам, а сама транзакція міститься лише в блоці, адресованому учаснику 2 (усі інші учасники отримують порожній блок).

3. Учасник 2 генерує свій сесійний числовий ключ $k2$, виконує обчислення $g^{k2} \bmod m$ та пересилає його учаснику 1. Пересилка відбувається через створення нового блоку та нової транзакції для учасника 1. В транзакцію в цьому випадку розміщується тип транзакції *send_key*, а

також як параметр *value* розміщується його зашифрований ключ.

4. Учасник 1 генерує свій числовий ключ $k1$ і, маючи частину ключа від учасника 2, створює сесійний ключ $g^{k1k2} \bmod m$, яким виконує шифрування повідомлення.

5. Після зашифрування повідомлення учасник 1, використовуючи підхід граничного розподілу секрету між n сторонами, де n сторін – кількість учасників, які знаходяться на даний момент в мережі, окрім учасника 2, якому виконується пересилка повідомлення, виконує розбивання свого зашифрованого ключа $g^{k1} \bmod m$ так, щоб будь-які p і більше сторін мали змогу відновити цей ключ. При цьому будь-які $p - 1$ і менше сторін не матимуть змоги відновити ключ.

6. Учасник 1 створює новий блок, в якому розміщує наступні транзакції:

- для учасника 2 (якому відправляється повідомлення):
  - в полі *transaction_type* розміщується тип *send_message*;
  - в полі *value* розміщується зашифроване повідомлення.
- для учасників, які в даний момент в мережі і брали участь при розподілі ключа:
  - в полі *transaction_type* розміщується *send_generated_part_key*;
  - в полі *time* розміщується час відкриття повідомлення (вказується учасником 1);
  - в полі *value* розміщується розділена частина ключа учасника 1 у відповідності до ідентифікатора кожного учасника, який брав участь у формуванні частин ключа.
- для інших учасників блок залишається порожнім.

Після створення усіх транзакцій виконується розсилка нового блоку учасником 1 усім учасникам в мережі з відповідними транзакціями. Після успішної відправки блоків, учасник 1 видаляє свій числовий ключ, який брав участь у шифруванні повідомлення.

Розглянемо, як буде виконано розшифрування повідомлення в момент часу t учасником 2:

- коли настане час t для розшифрування повідомлення в учасника 2, інші учасники виконають відправку зашифрованої частини ключа учасника 1, отриманої ними в транзакції в момент відправки повідомлення учасником 1. Дана відправка відбуватиметься у такий спосіб: i-й учасник створить новий блок та нову транзакцію для учасника 2, де він розмістить як поле *transaction_type – send_part_key*, а як поле *value* буде вказано відповідну частину ключа. Також в полі *pending_hash* буде вказано хеш відповідної транзакції розкриття, отриманої при пересилці повідомлення. Після формування відповідної транзакції даний блок буде відправлено усім учасникам.
- Учасник 2, отримавши граничну кількість ключів p від інших учасників, виконає зведення цих ключів за використаною граничною схемою і в результаті отримає ключ учасника 1.
- Додавши до отриманого ключа свій ключ, учасник 2 зможе встановити ключ, яким було зашифровано повідомлення учасником 1, та виконати його розшифрування.

Верифікація блоку учасниками в даний момент буде полягати в наступному: у випадку відправки ключа в момент часу t, учасники, які знають цей час t та хеш транзакції, що відкривається, будуть верифікувати цей блок. І якщо виявиться, що i-й учасник відправляє цей блок завчасно, його буде відхилено іншими учасниками мережі. Відповідна перевірка відбуватиметься за будь-якої пересилки блоку і фільтруватиметься за типом транзакції.

Висновки. Подана схема суттєво розширює можливості використання криптографії з часовим розкриттям завдяки виключенню сервера зі схеми, як заздалегідь визначеної третьої сторони. Це дозволяє створити більш захищену мережу обміну повідомленнями з часовим розкриттям порівняно з клієнт–серверною мережею. Власне вищий рівень захищеності досягатиметься через використання інших клієнтів як третіх сторін, а при використанні *bitcoin*-схеми формуватиметься ланцюжок транзакцій, який не дасть змоги зловмисникам імітувати реального клієнта та отримати доступ до певних даних мережі. Пересилка кожного блоку буде верифікуватись, а тому завчасна відправка або ж відправка некоректного блоку буде відкинута.

1. *Rivest R.*, *Shamir A.*, *Wagner D.* Time-lock puzzles and timed-released crypto // Massachusetts Institute of Technology. – Cambridge, MA, USA, 1996. – 9 p.
2. *Crescenzo G.*, *Ostrovsky R*, *Rajagopalan S.* Conditional oblivious transfer and timed-release encryption // Lecture Notes in Comp. Sci. – 1999. – **1592**. – P. 74–89.
3. *Boneh D.*, *Naor M.* Timed commitments // Ibid. – 2000. – **1880**. – P. 236–254.
4. *Rabin M.*, *Thorpe C.* Time-Lapse Cryptography. Technical Report TR-22-06. – Harvard Univ., School of Engin. and Applied Sci., 2006. – 16 p.
5. *Paterson K.G.*, *Quaglia E.A.* Time-specific encryption // Lecture Notes in Comp. Sci. – 2010. – **6280**. – P. 1–16.
6. *Boneh D.*, *Franklin M.* Identity-Based Encryption from the Weil Pairing Matthew Franklin // Ibid. – 2001. – **2139**. – P. 213–229.
7. *Cathalo J.*, *Libert B.*, *Quisquater J.* Efficient and non-interactive timed-release encryption // Lecture Notes in Comp. Sci. – 2005. – **3783**. – P. 291–303.
8. *Provably* secure timed-release public key encryption / J. Cheon, N. Hopper, Y. Kim et al. // ACM Trans. Inf. Syst. Secur. – 2008. – **11**. – 44 p.
9. *Головянко М.В.*, *Плиско Д.А.* Построение распределенной системы онтологий на базе технологии *Peer-To-Peer* (*P2P*): Зб. наук. праць Харків. ун-ту Повітряних Сил. – 2010. – **3(25)**. – С. 131–133.
10. *Гончар С.А.* Використання *bitcoin*-схеми для безпарольної авторизації на сервері // Вісн. Київськ. нац. ун-ту імені Тараса Шевченка. Серія фізико-математичні науки. – 2014. – **2**. – С. 108–111.

E-mail: *sg.gonchar@gmail.com*

С.А. Гончар

Криптографический протокол с временным раскрытием в сети равноправных узлов

Введение. Развитие криптографии позволило ввести в обиход такие понятия, как электронные платежи, электронные деньги, электронная подпись и др. Именно криптография обеспечивает необходимый уровень безопасности участников таких электронных средств.

Одним из самых интересных, но недостаточно распространенным направлением в криптографии являются так называемые «письма в будущее». Криптографические системы такого типа имеют в иностранной литературе достаточно разветвленную терминологию: *time-released crypto* [1], *timed-release encryption* [2], *timed commitments* [3], *time-lapse cryptography* [4], *time-specific encryption* [5]. Общеупотребительного украинского эквивалента этих терминов пока нет, но, по мнению автора, наиболее целесообразно использование словосочетания «криптография с временным раскрытием», что подразумевает совокупность методов, позволяющих зашифровать данные таким образом, чтобы обеспечить их расшифровку после окончания заранее определенного времени или в заранее определенное время и исключить возможность досрочного доступа к зашифрованным данным.

Постановка задачи

Согласно современному состоянию исследований в рамках рассматриваемых систем выделяют два направления:

- использование «математических замков с временным механизмом» – представляют собой вычислительные задачи, которые не могут быть решены без вычислений на компьютере в течение определенного промежутка времени;
- использование третьей стороны для хранения определенной информации, важной в дальнейшем при расшифровке сообщения.

Сущность и методы первого направления с того момента, как его предложили, практически не изменились и возможное развитие «математических замков с временным механизмом» связано, в первую очередь, с новыми конкретными задачами.

Второе направление имеет более разветвленный эволюционный процесс, но до широкого распространения шифрования на основе личных данных (*identity based encryption*) [6] осуществлялись только попытки развить оригинальную идею авторов работы [1]. В частности, большинство исследователей посвятили свои работы вопросам минимизации взаимодействия пользователя и сервера, обеспечению распространенности ключа и анонимности пользователя.

Протокол с временным раскрытием в сети равноправных узлов

Для криптографии с временным раскрытием на основе серверов (третьей стороны) за последние десятилетия осуществлено много как практических, так и теоретических наработок.

На рис. 1 приведена схема эволюции основных протоколов.

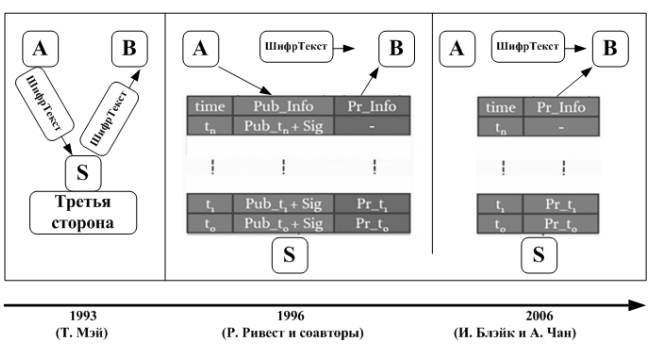

Рис. 1. Схемы шифрования с временным раскрытием различных авторов

Прорыв в вопросах криптографии с временным раскрытием состоялся после введения шифрования на ос-

нове личных данных. Основа такого типа шифрования – шифрование отправителем сообщения M в соответствии с конкретным получателем. Формально такое шифрование можно определить как кортеж из четырех рандомизированных алгоритмов [7]:

• *Установка* (1^k). При входе на этот алгоритм параметра безопасности 1^k, генерируются открытые параметры π_{IBE}, включающие в себя хэш-функции и сообщение M. Дополнительно генерируется основной секрет δ_{IBE}, который остается конфиденциальным благодаря серверу.

• *Получение* $(\pi_{IBE}, \delta_{IBE}, I)$. Имея на входе открытые параметры π_{IBE}, основной секрет δ_{IBE} и характеристики личности I, на выход выдается закрытый ключ sk_I.

• *Шифрование* (π_{IBE}, I, M). Генерируется шифротекст c на основе входных параметров π_{IBE}, I, M.

• *Дешифровка* $(\pi_{IBE}, sk_I, \hat{c})$. Выдается на выход оригинальное сообщение в соответствии с $\hat{c}$ или сообщение об ошибке.

Начиная с 2003 г., были предложены различные протоколы, основанные на квадратичных остатках и на свойствах билинейного спаривания (отображения) на группах эллиптических кривых.

Существует достаточно много наработок с использованием шифрования на основе личных данных, поэтому рассмотрим лишь основные из них.

Авторы [8] исходили из схемы, предложенной И. Блайком и А. Чаном, в которой сервер не взаимодействует с отправителем и получателем. Это достигается путем периодической генерации специфических временных «лазеек», позволяющих выполнять дешифровку текста, зашифрованного «на будущее», или «лазеек» для цифровой подписи. Данная концепция получила название *шифрования с временным раскрытием без взаимодействия*. Предложенная схема использует группы билинейных отображений и предполагает, что зашифрованный текст всегда содержит информацию о вскрытии, которая приводится в конце сообщения. Кроме того, предложенная схема может быть использована при необходимости раскрытия текста при определенном событии.

Очень подробно вопросы создания криптосистемы с временным раскрытием путем шифрования на основе личных данных рассмотрены в [5]. Свой метод авторы назвали *определенным во времени шифрованием* (*time-specific encryption*).

Согласно этому методу временной сервер рассылает так называемый *постоянный во времени* ключ в начале каждого временного интервала. Этот ключ доступен всем пользователям и содержите характеристику времени t. Отправитель может определить любой интервал времени в процессе шифрования, а получатель – восстановить оригинальное сообщение только если имеет *постоянный во времени* ключ, соответствующий времени в заданном отправителем интервале. При этом получатели дополнительно обеспечиваются закрытыми ключами или открытыми ключами, или параметрами на основе личных данных, а дешифрование требует использования как закрытого ключа, так и соответствующего *постоянного во времени* ключа. Это обеспечивает защиту против нечестного временного сервера и возможность дешифровки текста только для избранной группы пользователей.

Развитие криптографического инструментария всегда предлагает пути развития для уже существующих подходов. Очевидно, что возможным и желаемым направлением такого развития может быть создание систем, не использующих серверы как таковые. В частности, перспективным может быть использование *bitcoin*-схемы, которая базируется на сети равноправных узлов (*peer-to-peer*). Каждый узел в такой сети является одновременно и сервером, и клиентом, т.е. представляет некоторую информацию и обращается к другим узлам. Координация также выполняется всеми участниками одновременно, благодаря разветвленной сети хранения метаинформации всеми участниками [9].

В *bitcoin*-системе вся информация о транзакциях сохраняется в цепи блоков. Каждый блок содержит заголовки и список транзакций. Заголовок состоит из нескольких свойств, среди которых есть также хэш предыдущего блока. Таким образом, вся цепочка блоков сохраняет все транзакции за все время работы *bitcoin*. В последних версиях системы вся цепочка блоков сохраняется полностью каждым клиентом, что делает эту систему децентрализованной [10].

Предлагается модифицировать *bitcoin*-схему таким образом, чтобы иметь возможность отправлять пользователям сообщение, которое будет открыто в заранее установленное отправителем время, основываясь на принципах *bitcoin*-схемы. Это в первую очередь позволит отказаться от использования сервера.

В данной сети вся информация будет храниться в цепочке блоков. Каждый блок будет содержать заголовок и список транзакций. Заголовок будет состоять из некоторых свойств, среди которых будет также хэш на предыдущий блок. Так, данные об обмене любой информацией будут сохраняться в цепочке блоков в течение всего времени работы.

Отметим одно важное свойство данной схемы: каждый блок будет проверяться участниками на корректность. Подразумевается, что перед тем как некий блок получит определенный участник, другие участники выполнят верификацию этого блока по определенным параметрам, тем самым, подтвердив эту пересылку, или отменят ее. Это будет гарантировать, что никто из участников не сможет отправить некорректную информацию или досрочно отправить какие-то данные.

Рассмотрим содержание блока предлагаемой схемы более подробно (рис. 2). Блок будет состоять из следующих параметров:

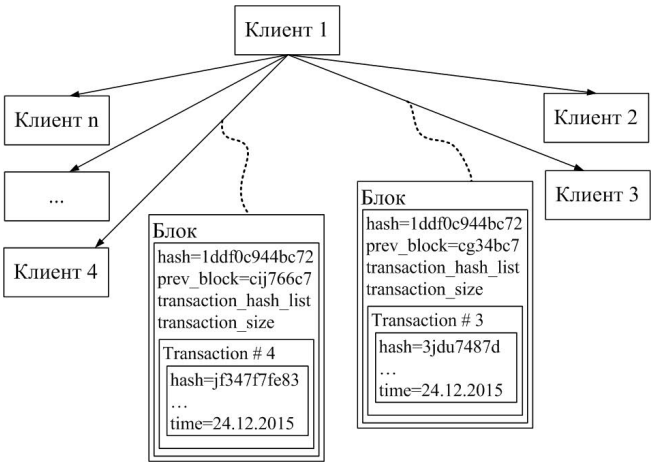

Рис. 2. Схема сети равноправных узлов для реализации отправки сообщений с временным раскрытием

- ***hash*** – хэш заголовка блока;
- ***prev_block*** – хэш предыдущего блока в цепочке;
- ***transaction_hash_list*** – список хэшей транзакций блока;
- ***transaction_size*** – количество транзакций.

Транзакция в свою очередь будет состоять из следующих параметров:

- ***hash*** – хэш всей транзакции;
- ***prev_hash*** – хэш предыдущей транзакции;
- ***pending_hash*** – хэш транзакции открытия;
- ***transaction_type*** – тип транзакции. Будут доступны следующие типы транзакций: ***request_key***, ***send_key***, ***send_message***, ***send_generated_part_key***, ***send_part_key***;
- ***value*** – данные, которые пересылаются;
- ***time*** – время открытия.

Рассмотрим схему отправки сообщения участником 1 участнику 2, когда в определенный момент времени в сети находятся n участников. В каждом блоке и транзакции автоматически будут добавляться хэш и предыдущий хэш, а потому в дальнейшем это не описывается.

1. Создаются сессионные параметры g и m – большие числа, число m – простое и наибольший общий делитель (НОД) $(m, n) = 1$.

2. Участник 1 отправляет запрос участнику 2, чтобы тот отправил ему в зашифрованном виде свой сессионный ключ. Отправка этого запроса выполняется через формирование нового блока и новой транзакции для участника 2. Тип транзакции в этом случае *request_key*. Отправка блока выполняется всем участникам, а собственно транзакция содержится только в блоке, адресованном участнику 2 (остальные участники получают пустой блок).

3. Участник 2 генерирует свой сессионный числовой ключ $k2$, выполняет вычисления $g^{k2} \bmod m$ и пересылает его участнику 1. Пересылка происходит посредством создания нового блока и новой транзакции для участника 1. В транзакцию в этом случае помещается тип транзакции *send_key*, а также в качестве параметра *value* помещается его зашифрованный ключ.

4. Участник 1 генерирует свой числовой ключ $k1$ и, имея часть ключа от участника 2, создает сессионный ключ $g^{k1k2} \bmod m$, которым выполняет шифрование сообщения.

5. После шифрования сообщения участник 1, используя подход предельного распределения секрета между n сторонами, где n сторон – количество участников, находящихся на данный момент в сети, кроме участника 2, которому выполняется пересылка сообщения, осуществляет разбивку своего зашифрованного ключа $g^{k1} \bmod m$ так, чтобы любые p и более сторон имели возможность восстановить этот ключ. При этом любые p-1 и меньше сторон не смогут восстановить ключ.

6. Участник 1 создает новый блок, в который помещает следующие транзакции:

- для участника 2 (которому отправляется сообщение):
  - в поле *transaction_type* помещается тип *send_message*;
  - в поле *value* помещается зашифрованное сообщение.
- Для участников, которые в данный момент в сети и принимали участие при распределении ключа:
  - в поле *transaction_type* помещается *send_generated_part_key*;
  - в поле *time* помещается время открытия сообщения (указывается участником 1);
  - в поле *value* помещается разделенная часть ключа участника 1 в соответствии с идентификатором каждого участника, принимавшего участие в формировании частей ключа.
- Для других участников блок остается пустым.

После создания всех транзакций выполняется рассылка нового блока участником 1 всем участникам в сети с соответствующими транзакциями. После успешной отправки блоков участник 1 удаляет свой числовой ключ, который принимал участие в шифровании сообщения.

Рассмотрим, как будет выполнена расшифровка сообщения в момент времени t участником 2:

- когда наступит время t для расшифровки сообщения у участника 2, остальные участники выполнят отправку зашифрованной части ключа участника 1, которую они получили в транзакции в момент отправки сообщения участником 1. Данная отправка будет происходить так: i-й участник создаст новый блок и новую транзакцию для участника 2, куда он поместит в качестве поля *transaction_type – send_part_key*, а в качестве поля *value* будет указана соответствующая часть ключа. Также в поле *pending_hash* будет указан хэш соответствующей транзакции раскрытия, полученной при пересылке сообщения. После формирования соответствующей транзакции данный блок будет отправлен всем участникам.
- Участник 2, получив предельное количество ключей p от других участников, выполнит объединение

этих ключей по использованной предельной схеме и в результате получит ключ участника 1.

• Добавив к полученному ключу свой ключ, участник 2 сможет установить ключ, которым было зашифровано сообщение участником 1, и выполнить его расшифровку.

Верификация блока участниками в данный момент будет заключаться в следующем: в случае отправки ключа в момент времени t, участники, знающие это время t и хэш открывающейся транзакции, будут верифицировать этот блок. И если окажется, что i-й участник отправляет этот блок досрочно, он будет отклонен другими участниками сети. Соответствующая проверка будет происходить при любой пересылке блока и фильтроваться по типу транзакции.

Заключение. Предложенная схема существенно расширяет возможности использования криптографии с временным раскрытием благодаря исключению сервера как заранее определенной третьей стороны из схемы. Это дает возможность создать более защищенную сеть обмена временными сообщениями в сравнении с клиент–серверной сетью. Собственно высокий уровень защищенности достигаться путем использования других клиентов в качестве третьих сторон, а основываясь на *bitcoin*-схеме получается цепочка транзакций, которая не позволит злоумышленникам имитировать реального клиента и получить доступ к определенным данным сети. Пересылка каждого блока будет верифицирована, а потому заблаговременная отправка или отправка некорректного блока будет отклонена.

UDC 004.056.5

S.A. Gonchar

Time-Released Cryptographic Protocol in Peer-to-Peer Network

Keywords: time-released cryptography, bitcoin, time-lock puzzles, time server.

The main principles of time-released cryptography is observed. It is shown that the evolution of methods using the third part person such as peer-to-peer network and "bitcoin" scheme is a promising way of the time-released crypto systems development. The basic provisions of the proposed protocol are: creating of numerical key, sending a message, creating a new blocks and transactions, and also data decryption mechanism.

The article discusses the principles of new cryptographic protocol that implements the peer-to-peer network and bitcoin schemes to the time-released cryptography concept. The main propositions of time-released cryptography is observed. There are two methods of implementing time-release cryptography: the first is based on the need to perform certain calculations in a specific period of time (time-lock puzzles), and the second requires a third party (or central authority) to carry out the process of issuing some information to open an encrypted text in a certain period or time. The use of the third party, usually a time server, received significant development after the introduction of an identity base encryption since the early 2000s and reached its peak in 2008. However, the new technologies have recently emerged which allow to secure online transactions without relying on a central authority. The steps to create a peer-to-peer network and sending a message do not differ from those for regular messages in the above-mentioned network. The main difference is an specifying of the release time as one of the transaction parameters. At some period of time the key to decrypt messages is sent by any network user that first detects that it is time to decrypt the recipient message. The basis for the safety and accuracy of the time message transmission is that some members of the network sends the block with part of the key before release time, this block will be rejected by other members of the network as a result of the mechanism of its verification. The verification process involves checking of the release time of the message and the hash corresponding transaction. The feature of the created protocol is that due to "bitcoin" formed a chain-chart transactions that does not allow attackers to simulate customers, and to access to certain data network. Also this protocol provides for users a new possibility to improve network protection, because the server is not used, like in other existing time-released protocols.

●