МАТЕМАТИЧЕСКИЕ МЕТОДЫ КРИПТОГРАФИИ

DOI 10.17223/20710410/11/5

УДК 519.7

НЕКОТОРЫЕ ПРОТОКОЛЫ ДОВЕРЕННОЙ ЦИФРОВОЙ ПОДПИСИ

Е. А. Толюпа

Ярославский государственный университет им. П. Г. Демидова, г. Ярославль, Россия

E-mail: tolyupa@gmail.com

Предложены усовершенствование протокола доверенной цифровой подписи (ДЦП) авторов J.-Y Lee, J.-H. Cheon и S. Kim, устранившее его уязвимость, позволяющую сторонам создавать доверенный ключ подписания независимо друг от друга, и протокол коллективной ДЦП, который можно использовать при реализации процедуры голосования.

Ключевые слова: *электронная цифровая подпись, доверенная цифровая подпись, анализ безопасности, голосование, коллективная подпись.*

Введение

Традиционные протоколы цифровой подписи (ЦП) позволяют пользователю выполнить подпись под документом самостоятельно, и любой участник электронного информационного обмена может убедиться в верности подписи, однако эти протоколы не позволяют одному участнику, скажем **A**, делегировать свои права другому участнику — **B**, который может подписывать сообщения от имени **A**. Такая необходимость возникает, когда, например, руководитель организации по той или иной причине (по состоянию здоровья или техническим возможностям) не может подписать документ сам. В этой ситуации он может доверить право подписывать документы от своего имени другому лицу, например своему заместителю. Заместитель, получив такое право, может подписывать сообщения от лица руководителя, а проверяющий будет знать, кто и от чьего имени поставил данную подпись. Решить подобную задачу позволяют протоколы доверенной цифровой подписи (ДЦП, или proxy signature).

Основоположниками теории протоколов ДЦП и разработчиками первого такого протокола являются M. Mambo, K. Usuda и E. Okamoto [1]. Ими были сформулированы первые требования к безопасности протоколов ДЦП [1, 2], расширенные позже в [3, 4]. Безопасная ДЦП должна удовлетворять следующим требованиям:

1. Проверяемость — проверяющий может быть убежден, что подпись поставлена с согласия доверителя.

2. Стойкость к фальсификации — только назначенная доверителем сторона может создать верную доверенную подпись от лица доверителя. Другими словами, участник **A** и третья сторона, не выбранная им в качестве доверенного подписчика, не смогут создать верную ДЦП от имени доверенного участника **B**.

3. Строгая идентификация — каждый может идентифицировать соответствующую доверенную сторону из доверенной подписи.

4. Неотрекаемость — если доверенный подписчик создает подпись под документом, то в дальнейшем он не сможет заявить, что подпись выполнена кем-то другим.

5. Противостояние злоупотреблению — доверенная сторона не должна использовать ключ подписания для целей, не разрешенных доверителем в информации о полномочиях. В случае злоупотребления ответственность доверенного подписчика должна определяться явно.

В данной работе изложены базовый протокол ДЦП, предложенный в [1], его модификация LCK из [5] на случай открытого канала связи и атака на нее из [6], демонстрирующая нарушение в ней требования 2 к безопасности. Предложено усовершенствование протокола LCK, устранившее его уязвимость этой атакой и сохранившее его пригодность для открытого канала. Предложен также протокол коллективной ДЦП с защищенным каналом связи, который можно использовать при реализации процедуры голосования. Все протоколы основаны на общей идее распределения ключевой информации, которая позволяет делегировать права подписи доверенной стороне, и построены по следующей общей схеме.

1. Общая схема протокола ДЦП

Основная идея, используемая при создании доверенных подписей, — построить такую схему распределения ключевой информации, чтобы можно было использовать уже существующие алгоритмы цифровой подписи. В распределении ключевой информации лежит следующий принцип.

Пусть имеются два участника информационного обмена **A** и **B** — соответственно доверитель права цифровой подписи и его доверенная сторона (доверенное лицо), получающая право подписывать от имени доверителя. Каждый из участников **A** и **B** имеет свою пару ключей — (x_A, y_A) и (x_B, y_B) соответственно, где первый ключ (x) в паре закрытый, а второй (y) открытый. **A** генерирует доверенность на право подписания от его имени и вместе с информацией о полномочиях (период действия доверенной подписи, типы документов, которые разрешено подписывать доверенной стороне, идентификаторы доверителя и доверенного лица и т. п.) передает (по защищенному или открытому каналу) участнику **B**, который на основании информации, содержащейся в доверенности, идентифицирует **A** и создает пару доверенных ключей (x_p, y_p), предназначенных соответственно для создания и проверки цифровой подписи по некоторому известному алгоритму. Теперь при помощи x_p он может подписывать сообщения от лица **A**, применяя этот алгоритм. Любой проверяющий должен самостоятельно вычислить ключ y_p участника **B**, используя информацию из доверенности участника **A**. Таким образом, убедившись, что ключ участнику **B** действительно доверил участник **A**, можно проверить подпись с помощью процедуры верификации из того же алгоритма цифровой подписи.

Общая схема протокола ДЦП, реализующая эту основную идею, выглядит следующим образом.

Действия доверителя **A**:

1. Генерирует случайное *обязательство* k и вычисляет его *свидетельство* $K = f(k)$, где f — некоторая односторонняя функция.

2. Создает s_A, зависящую от величин из $\{x_A, y_B, k, K\}$ и возможных полномочий m_w, и величины s_A, K и, возможно, m_w, составляющие доверенность, отправляет доверенному лицу **B**.

Действия доверенной стороны **B**:

1. Проверяет (с целью идентификации **A**) выполнение некоторого *идентификационного условия*, зависящего от y_A, y_B и полученных s_A и K.

2. Если оно выполнено, то вычисляет доверенные ключи: x_p — в зависимости от s_A, x_B, y_A, y_B и, возможно, m_w; $y_p = f(x_p)$.

3. С помощью подходящего алгоритма ЦП подписывает некоторый документ M, используя x_p в качестве ключа подписания. Созданную так подпись обозначает $\mathrm{Sign}(M, x_p)$. Формирует доверенную подпись под M как $\sigma = (\mathrm{Sign}(M, x_p), K, y_A, y_B, m_w)$, возможно, без m_w.

Действия **проверяющего**:

1. Вычисляет ключ y_p, применяя некоторый алгоритм к данным y_A, y_B, K и, возможно, m_w, содержащимся в σ.

2. По ключу y_p проверяет подпись $\mathrm{Sign}(M, x_p)$ под документом M при помощи процедуры верификации соответствующего алгоритма ЦП.

Конкретные протоколы, построенные по этой схеме, различаются друг от друга алгоритмами выполнения перечисленных в ней действий участников протокола, которые (алгоритмы), в свою очередь, зависят и от применяемого математического аппарата, и от характера используемого канала передачи данных — открытый или закрытый, и от вида информации о полномочиях, в том числе от ее наличия или отсутствия.

В рассматриваемых далее конкретных протоколах ДЦП используется аппарат теории чисел. В них, кроме уже обозначенного, используются: p и q — большие простые числа, причем $q|p-1$; $g \in \mathbb{Z}_p^*$, порядок g равен q и g является общеизвестным. Считается, что вычисление дискретного логарифма — трудоёмкая задача. Соответственно в роли односторонней функции $f(k)$ выступает $g^k \bmod p$. Первый и второй ключи (x и y) в паре ключей каждого участника и в паре доверенных ключей связаны соотношением $y = g^x \bmod p$. Действие 3 участника **B** и действие 2 **проверяющего** из общей схемы сохраняются во всех протоколах, с тем лишь уточнением, что в последних используется алгоритм ЦП, базирующийся на сложности задачи дискретного логарифмирования. В приводимых ниже описаниях протоколов эти действия опущены.

2. Протокол MUO

Этот протокол предложили M. Mambo, K. Usuda и E. Okamoto (MUO) в [1]. Именно он и является базовым для модификаций, рассматриваемых в данной работе.

Действия **A**:

1. Генерирует случайное число $k \in \mathbb{Z}_q^*$ и вычисляет $K = g^k \bmod p$.

2. Вычисляет $s_A = (x_A + k \cdot K) \bmod q$ и посылает (s_A, K) участнику **B** по защищенному каналу связи.

Действия **B**:

1. Для идентификации **A** проверяет сравнение: $g^{s_A} = y_A \cdot K^K (\bmod p)$.

2. Если оно выполнено, то вычисляет доверенный ключ подписания

$$x_p = (s_A + x_B \cdot y_B) \bmod q.$$

Действия **проверяющего**:

1. Зная (из подписи) свидетельство K, открытые ключи доверителя (y_A) и доверенной стороны (y_B), вычисляет доверенный ключ y_p для проверки подписи по правилу

$$y_p = y_A \cdot K^K \cdot y_B^{y_B} \bmod p.$$

Заметим, что данный протокол не удовлетворяет пятому требованию безопасности, поскольку действие 2 участника **A** не включает в себя передачу информации о полномочиях.

В приводимых ниже модификациях протокола MUO действие 1 участника **A** то же самое, что и в базовом протоколе, в связи с чем в их описаниях оно опущено.

3. Протокол LCK

Протокол MUO требует наличия защищенного канала между доверителем и доверенной стороной. В своей работе [5] J.-Y. Lee, J.-H. Cheon и S. Kim (LCK) модифицировали его, что позволило передавать доверенность по открытому каналу.

Действия **A**:

2. Вычисляет $s_A = (x_A + k \cdot y_B) \bmod q$ и пару (s_A, K) отправляет участнику **B** по общедоступному каналу.

Действия **B**:

1. Проверяет идентификационное условие $g^{s_A} = y_A \cdot K^{y_B} (\bmod p)$.

2. Если оно выполнено, то **B** вычисляет доверенный ключ подписания:

$$x_p = (s_A + x_B \cdot y_A) \bmod q.$$

Действия **проверяющего**:

1. Вычисляет ключ y_p по следующему правилу: $y_p = y_A \cdot K^{y_B} \cdot y_B^{y_A} \bmod p$.

В [6] показано, что такая модификация протокола MUO приводит его к уязвимости, позволяющей:

1) сторонам **A** и **B** доверенный ключ подписания создавать независимо друг от друга, т. е. без участия их в выполнении протокола, и тем самым участнику **A** доверенную подпись за участника **B** ставить без его ведома, а участнику **B** подпись от имени **A** ставить, не имея на то доверенности от последнего;
2) третьей стороне **C** доверенный ключ подписания создать независимо от участников **A** и **B**, т. е. без их участия в выполнении протокола, и тем самым поставить доверенную подпись;
3) третьей стороне **C** доверенность для **B** модифицировать под себя, т. е. изменить её таким образом, как если бы участник **A** доверил право подписи стороне **C**, а не участнику **B**.

Первая атака выглядит следующим образом. Если **A** возьмет случайное число s и вычислит $K = g^s \cdot y_B^{-y_A \cdot y_B^{-1}} \bmod p$ и $x_p = (x_A + s \cdot y_B) \bmod q$, то это x_p будет обладать свойством доверенного ключа подписания, а именно: подпись, поставленную этим ключом, любой проверяющий ее на ключе y_p примет за подпись, поставленную стороной **B** от имени стороны **A**, ибо $y_p = y_A \cdot K^{y_B} \cdot y_B^{y_A} \bmod p = y_A \left(g^s \cdot y_B^{-y_A \cdot y_B^{-1}}\right)^{y_B} y_B^{y_A} \bmod p = y_A \cdot g^{s y_B} \bmod p = g^{x_A} g^{s y_B} \bmod p = g^{x_p} \bmod p$, т. е. (x_p, y_p) является доверенной парой ключей.

Обратно, если участник **B** возьмет случайное s и изготовит $K = g^s \cdot y_A^{-y_B^{-1}} \bmod p$ и $x_p = (s \cdot y_B + x_B \cdot y_A) \bmod q$, то будет $y_p = y_A \cdot K^{y_B} \cdot y_B^{y_A} \bmod p = y_A \cdot y_A^{-1} g^{s y_B} \cdot y_B^{y_A} \bmod p = g^{s y_B} g^{x_B y_A} \bmod p = g^{x_p} \bmod p$, что означает, что и это x_p является доверенным ключом подписания.

Вторая атака выглядит следующим образом. Если **C** возьмет случайное число s и вычислит $K = y_A^{-y_B^{-1}} \cdot y_B^{-y_A \cdot y_B^{-1}} \cdot g^s \bmod p$ и $x_p = s \cdot y_B \bmod q$, то это x_p будет также обладать свойствами доверенного ключа подписания, так как $y_p = y_A \cdot K^{y_B} \cdot y_B^{y_A} \bmod p = y_A \left(y_A^{-y_B^{-1}} \cdot y_B^{-y_A \cdot y_B^{-1}} \cdot g^s\right)^{y_B} y_B^{y_A} \bmod p = y_A \cdot y_A^{-1} \cdot y_B^{-y_A} \cdot g^{s \cdot y_B} \cdot y_B^{y_A} \bmod p = g^{s \cdot y_B} \bmod p = g^{x_p} \bmod p$.

Третья атака заключается в следующем: имея (из открытого канала) доверенность (s_A, K), предназначенную участником **A** для **B**, сторона **C** берёт случайное число $\overline{k} \in \mathbb{Z}_q^*$, вычисляет $\overline{K} = K^{y_B \cdot {y_C}^{-1}} \cdot g^{\overline{k}} \bmod p$ и $\overline{s}_A = s_A + \overline{k} \cdot y_C \bmod q$ и получает пару

$(\overline{s}_A, \overline{K})$, обладающую свойствами доверенности, сгенерированной участником **A** для **C**, так как $g^{\overline{s}_A} \bmod p = g^{s_A+\overline{k}\cdot y_C} \bmod p = g^{x_A+k\cdot y_B+\overline{k}\cdot y_C} \bmod p = y_A \cdot g^{(k\cdot y_B\cdot y_C^{-1}+\overline{k})y_C} \bmod p = y_A \left(K^{y_B\cdot y_C^{-1}} \cdot g^{\overline{k}}\right)^{y_C} \bmod p = y_A \cdot \overline{K}^{y_C} \bmod p$.

4. Авторская модификация протокола MUO

Предлагается следующая модификация протокола MUO с передачей доверенности по открытому каналу и без уязвимости предыдущей модификации.

Действия **A**:

2. Вычисляет $s_A = (x_A + k \cdot y_B \cdot K) \bmod q$ и пару (s_A, K) отправляет участнику **B** по общедоступному каналу связи.

Действия **B**:

1. Проверяет сравнение $g^{s_A} = y_A \cdot K^{y_B\cdot K} (\bmod p)$.

2. Если оно выполнено, то **B** вычисляет доверенный ключ подписания как

$$x_p = (s_A + x_B \cdot y_A) \bmod q.$$

Действия **проверяющего**:

1. Зная (из подписи) свидетельство K, открытые ключи доверителя (y_A) и доверенной стороны (y_B), вычисляет доверенный ключ y_p для проверки подписи по правилу

$$y_p = y_A \cdot K^{y_B\cdot K} \cdot y_B^{y_A} \bmod p.$$

Здесь, как и в протоколе LCK, доверенность s_A зависит от значения y_B. Предположим, что злоумышленник **C** воспользуется переданной по открытому каналу доверенностью и создаст секретный ключ $\overline{x}_p = s_A + x_C \cdot y_A$. В этом случае соответствующий открытый ключ имеет вид $\overline{y}_p = g^{\overline{x}_p} \bmod p = y_A \cdot K^{y_B\cdot K} \cdot y_C^{y_A} \bmod p$. Злоумышленник подписывает документ и заявляет, что сделал это по доверенности от **A**. Согласно протоколу, лицо, проверяющее эту подпись $\sigma = (\mathrm{Sign}(M, x_p), K, y_A, y_C)$, располагая только свидетельством K, открытыми ключами доверителя (y_A) и злоумышленника (y_C), не сможет корректно вычислить $\overline{y}_p$ и проверить подпись σ, не подставив в выражение для $\overline{y}_p$ значения открытого ключа y_B того участника **B**, для которого в действительности предназначалась доверенность, и тем самым уличит мошенника. Таким образом, добавив в доверенность зависимость s_A от y_B, доверитель **A** однозначно определил доверенную сторону, что позволило использовать открытый канал для передачи доверенности.

Покажем, что при невозможности взятия дискретного логарифма атаки из [6], описанные выше, на этот протокол невозможны.

В самом деле, реализация первых двух атак выглядит следующим образом.

В первой атаке участник **A** должен взять такие s и K, чтобы для $\overline{x}_p = (x_A + s \cdot y_B) \bmod q$ выполнилось равенство $g^{\overline{x}_p} \bmod p = y_p$, т. е. сравнение

$$y_A \cdot g^{s y_B} = y_A \cdot K^{y_B\cdot K} \cdot y_B^{y_A} (\bmod p),$$

или, что то же самое,

$$K^K = g^s \cdot y_B^{-y_A\cdot y_B^{-1}} (\bmod p).$$

С другой стороны, участник **B** должен подобрать s и K так, чтобы для $\overline{x}_p = (s \cdot y_B + x_B \cdot y_A) \bmod q$ выполнилось равенство $g^{\overline{x}_p} \bmod p = y_p$, т. е. сравнение

$$K^K = g^s \cdot y_A^{-y_B^{-1}} (\bmod p).$$

Для реализации второй атаки злоумышленник **C** должен подобрать значения s и K так, чтобы для $\overline{x}_p = s \cdot y_B \bmod q$ выполнялось равенство $g^{\overline{x}_p} \bmod p = y_p$, т. е. сравнение

$$K^K = g^s \cdot y_A^{-y_B^{-1}} \cdot y_B^{-y_A \cdot y_B^{-1}} (\mathrm{mod}\, p).$$

Для реализации третьей атаки злоумышленнику **C** необходимо подобрать такие значения $\overline{k}$ и $\overline{K}$, чтобы выполнилось сравнение

$$g^{s_A + \overline{k} \cdot y_C} = y_A \cdot \overline{K}^{y_C \cdot \overline{K}} (\mathrm{mod}\, p),$$

или, что то же самое,

$$\overline{K}^{\overline{K}} = g^{\overline{k}} \cdot K^{y_B \cdot y_C^{-1} \cdot K} (\mathrm{mod}\, p).$$

Следовательно, в каждом случае злоумышленник должен создать такие x и y, которые удовлетворяют сравнению $y^y = ag^x (\mathrm{mod}\, p)$ для заданных a, g и p. При выбранном y нахождение нужного x требует вычисления дискретного логарифма, а при выбранном x нахождение нужного y представляется не менее сложной задачей.

Таким образом, добавив в протокол LCK зависимость s_A от K, удалось убрать его слабость к атакам из [6].

5. Протокол коллективной ДЦП

Построим протокол ДЦП, в котором участвуют доверитель **A** и доверенное множество $\mathfrak{B}$, состоящие из n доверенных сторон $\mathbf{B}_1$, $\mathbf{B}_2$, …, $\mathbf{B}_n$. Доверенная подпись, поставленная любым из участников множества $\mathfrak{B}$, должна удовлетворять следующим условиям.

1) **Проверяющий** может убедиться, что подпись поставил один из участников $\mathfrak{B}$, но не должен узнать, кто именно.
2) Доверитель **A** может однозначно идентифицировать участника множества $\mathfrak{B}$, подписавшего документ.

Пусть для каждого $i = 1, 2, \ldots, n$ имеются: x_i — секретный и y_i — открытый ключи участника $\mathbf{B}_i$, связанные соотношением $y_i = g^{x_i} \bmod p$.

Действия **A**:

1. Генерирует случайные числа $r_1, \ldots, r_n \in \mathbb{Z}_q^*$ и публикует набор чисел $R_1 = g^{r_1} \bmod p$, …, $R_n = g^{r_n} \bmod p$ без указания их соответствия участникам из множества $\mathfrak{B}$. Все значения $R_1, \ldots, R_n$ должны быть различны.

2. Для каждого $i = 1, 2, \ldots, n$ вычисляет $S_i = (x_A + r_i \cdot R_i) \bmod q$ и посылает (S_i, R_i) участнику $\mathbf{B}_i$ по защищенному каналу связи. Значение R_i необходимо передавать участнику $\mathbf{B}_i$, так как последний не знает, какой именно элемент из опубликованных следует применять в вычислениях при идентификации **A**.

Действия множества $\mathfrak{B}$:

Для каждого $i = 1, 2, \ldots, n$ участник $\mathbf{B}_i$ производит следующие действия:

1. Для идентификации участника **A** проверяет сравнение $g^{S_i} = y_A \cdot R_i^{R_i} (\mathrm{mod}\, p)$ и, если оно выполнено, то принимает S_i.

2. Выбирает случайное $k_i \in \mathbb{Z}_q^*$, вычисляет $K_i = g^{k_i} \bmod p$ и $s_i = k_i + x_i \cdot K_i \bmod q$. Отправляет пару (s_i, K_i) как доверенность участникам множества $\mathfrak{B}$ по защищенному каналу связи. На этом шаге $\mathbf{B}_i$ доверил право подписи другим участникам множества $\mathfrak{B}$.

Таким образом, каждый участник $\mathbf{B}_i$ располагает набором

$$(S_i, R_i, s_1, \ldots, s_n, K_1, \ldots, K_n).$$

Действия подписывающего участника $\mathbf{B}_i$:

1. Вычисляет секретный ключ (для доверенного подписания)

$$x_{p_i} = S_i + \sum_{j=1}^{n} s_j (\mathrm{mod}\, q).$$

2. Доверенная подпись под документом M, поставленная с использованием ключа x_{p_i}, имеет вид $\sigma = (\mathrm{Sign}(M, x_{p_i}); y_A; R_i; y_1, \ldots, y_n; K_1, \ldots, K_n)$. Документ с подписью помещается на общедоступный ресурс.

Действия **проверяющего:**

1. Проверяет, что значение R_i из σ содержится в наборе, опубликованном доверителем.

2. Вычисляет открытый ключ (для проверки доверенной подписи)

$$y_{p_i} = y_A \cdot R_i^{R_i} \prod_{j=1}^{n} (K_j \cdot {y_j}^{K_j}) \bmod p. \tag{1}$$

(Непосредственно проверяется, что $y_{p_i} = g^{x_{p_i}} \bmod p$, т. е. y_{p_i} — действительно ключ для проверки подписи $\mathrm{Sign}(M, x_{p_i})$.)

В подписи σ под документом M содержится открытый ключ доверителя (y_A) и параметр R_i, который использовался при вычислении доверенности участника $\mathbf{A}$ участнику $\mathbf{B}_i$ в $\mathfrak{B}$. При вычислении открытого ключа (1) эти параметры позволяют убедиться, что подпись поставлена с согласия доверителя $\mathbf{A}$ одним из участников в $\mathfrak{B}$. Остальные n множителей имеют вид $K_j \cdot y_j^{K_j}$. Исходя из этой информации, проверяющий не может сделать вывод, кто именно из участников множества $\mathfrak{B}$ поставил подпись. Доверитель же $\mathbf{A}$ по элементу R_i может установить, кто из участников в $\mathfrak{B}$ подписал сообщение.

Протокол коллективной ДЦП можно использовать для реализации процедуры голосования. Пусть доверитель хочет, чтобы n доверенных лиц проголосовали от его имени. Для этого он наделяет их соответствующими полномочиями, выполняя указанный протокол. Каждое доверенное лицо вычисляет свой секретный ключ и подписывает сообщение (голос). Проверяющий в момент вычисления открытого ключа контролирует, что значение R_i не повторяется. Это является залогом того, что каждое доверенное лицо проголосовало единожды. Проверяющий не может проверить, как проголосовал каждый из участников. Доверитель может идентифицировать подписавшего по значению R_i, что является борьбой со злоупотреблением со стороны доверенного лица.

Заключение

Таким образом, автору удалось убрать уязвимость протокола LCK и создать более безопасный протокол, в котором для передачи доверенности не требуется защищенный канал связи. Однозначное определение доверителем доверенной стороны можно считать особенностью предложенного протокола, которую целесообразно использовать и при реализации протоколов с секретным каналом. Предложен протокол голосования на базе алгоритма коллективной ДЦП. Автор считает, что разработка прикладных протоколов на базе существующих протоколов ДЦП является интересным направлением, позволяющим найти им применение в существующих задачах.

ЛИТЕРАТУРА

1. *Mambo M., Usuda K., and Okamoto E.* Proxy signatures: Delegation of the power to sign messages // IEICE Trans. Fundamentals. 1996. V. E79-A. No. 9. P. 1338–1353.

2. *Mambo M., Usuda K., and Okamoto E.* Proxy signatures for delegating signing operation // Proc. of 3rd ACM Conference on Computer and Communications Security (CCS'96). ACM Press, 1996. P. 48–57.
3. *Kim S., Park S., and Won D* Proxy signatures, revisited // Information and Communications Security (ICICS'97). 1997. LNCS. V. 1334, P. 223–232.
4. *Lee B., Kim H., and Kim K.* Strong proxy signature and its applications // Proc. of the 2001 Symposium on Cryptography and Information Security (SCIS'01), Oiso, Japan, Jan. 23–26, 2001. V. 2/2. P. 603–608.
5. *Lee J.-Y., Cheon J.-H., and Kim S.* An analysis of proxy signatures: Is a secure channel necessary? // LNCS. 2003. V. 2612. P. 68–79.
6. *Wang G., Bao F., Zhou J., and Deng R. H.* Security Analysis of Some Proxy Signatures // LNCS. 2004. V. 2971. P. 305–319.