Научная статья
УДК 004.056(075.8)
DOI:10.31854/1813-324X-2023-9-2-128-142

Методы защиты от угрозы неправильного заполнения избирательного бюллетеня в системе дистанционного электронного голосования

Виктор Алексеевич Яковлев, yakovlev.va@sut.ru
Васан Давуд Салман, salman.vd@sut.ru

Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, Санкт-Петербург, 193232, Российская Федерация

Аннотация: *Рассматривается обобщенная схема дистанционного электронного голосования, основанная на гомоморфном шифровании. Исследуются два метода защиты системы голосования от угрозы со стороны избирателя, заключающиеся в неправильном заполнении бюллетеня избирателем. Оба метода основаны на алгоритмах «доказательства с нулевым разглашением секрета». Получены оценки сложности вычислений при формировании доказательства корректности заполнения бюллетеня избирателем и оценки сложности проверки доказательства контролирующей стороной. Сравнительный анализ сложности реализации обоих методов показал, что метод, основанный на доказательстве на базе равенства логарифмов) имеет меньшую сложность вычислений на стороне избирателя по сравнению с методом, основанном на перемешивании голосов избирателей. В тоже время второй метод (перемешивания голосов) требует в 1,67 раза меньше вычислений в блокчейне, что становится существенным фактором выбора в пользу второго метода при большом количестве избирателей.*

Ключевые слова: *система дистанционного электронного голосования, схема Эль-Гамаля на эллиптической кривой, схема перемешивания, проверка доказательства корректности заполнения бюллетеня, доказательство с нулевым разглашением секрета*

Ссылка для цитирования: Яковлев В.А., Салман В.Д. Методы защиты от угрозы неправильного заполнения избирательного бюллетеня в системе дистанционного электронного голосования // Труды учебных заведений связи. 2023. Т. 9. № 2. С. 128–142. DOI:10.31854/1813-324X-2023-9-2-128-142

Methods of Protection against Threat: Incorrect Ballot Filling by Voter in the Remote Electronic Voting System

Victor Yakovlev, yakovlev.va@sut.ru
Vasan Salman, salman.vd@sut.ru

The Bonch-Bruevich Saint-Petersburg State University of Telecommunications, St. Petersburg, 193232, Russian Federation

Abstract: *A generalized scheme of remote electronic voice based on homomorphic encryption is considered. Two methods of protecting the voting system from the threat from the voter, consisting in incorrect filling of the ballot by*

the voter, are investigated. Both methods are based on the algorithms of "zero-knowledge proof". Evaluations of the complexity of calculations in the formation of proof of the correctness of filling in the ballot by the voter and Evaluations of the complexity of verification of the proof by the controlling party are obtained. A comparative analysis of the complexity of the implementation of both methods has shown that the method based on the proof based on the equality of logarithms has less complexity of calculations on the voter's side compared to the method based on the mixing of votes. At the same time, the second method (the method of mixing votes) requires 1.67 times less calculations in the blockchain, which becomes a significant factor in choosing the second method in favor of a large number of voters.

Keywords: *the elliptic ElGamal scheme, remote electronic voting system, mixing scheme, verification scheme, proof of filling the ballot, zero-knowledge proof system*

For citation: Yakovlev V., Salman V. Methods of Protection against Threat: Incorrect Ballot Filling by Voter in the Remote Electronic Voting System. *Proc. of Telecom. Universities.* 2023;9(2):128–142. (in Russ.) DOI:10.31854/1813-324X-2023-9-2-128-142

Введение

Системы дистанционного электронного голосования (ДЭГ) все шире входят в жизнь современного общества. Распространение получили системы электронного голосования на основе микс-сетей [1–4], на основе слепой подписи [5–7] и на основе гомоморфного шифрования [8–10]. В этих системах решаются две главные задачи: обеспечение тайны и анонимности голосования, в том числе и для избирательной комиссии.

Принцип работы на основе микс-сетей заключается в создании системы из нескольких связанных прокси-серверов, которые называют миксами. Клиент шифрует сообщение один раз с использованием открытых ключей каждого из прокси-серверов в определенном порядке, который знает только он. Расшифровка криптограммы происходит в обратном порядке с помощью секретных ключей микс-серверов, но уже на стороне последних. Так как голоса приходят в избирательную комиссию в «перепутанном» виде, обеспечивается анонимность голосования. Система электронного голосования, основанная на слепой подписи, представляет собой криптографический метод, в котором сообщение m избирателя A подписывается органом подписи B, так что B не получает никакой информации о сообщении m. При этом обеспечивается доверие к переданному сообщению, но сохраняется анонимность избирателя. В системе голосования на основе гомоморфных криптосистем последние зашифровывают свои бюллетени открытым ключом избирательной комиссии. Затем они отправляют свои зашифрованные бюллетени на сервер, который «перемножает» все бюллетени и отправляет получивший результат в избирательную комиссию. Та расшифровывает это произведение бюллетеней и объявляет победителя выборов. Так как расшифрование выполняется сразу всех агрегированных бюллетеней, то обеспечивается анонимность каждого избирателя.

Известны практические системы голосования, в разной степени использующие эти подходы.

Во-первых, ДЭГ в России (см. URL: https://evoting.digitaldem.ru/wp-content/uploads/sites/2/2021/07/ptkdeg_general_description_2021-07-15.pdf). Организатор голосования (Комиссия ДЭГ) и Учетчик (блокчейн) генерируют ключевые пары (ключи шифрования и расшифрования бюллетеней). На блокчейне (БЧ) формируется итоговый открытый ключ шифрования, который передается Регистратору и избирателю. Закрытый ключ разделяется на доли. Избиратель генерирует ключевую пару электронной подписи. Избиратель и Регистратор выполняют протокол формирования подписи вслепую для ключа проверки электронной подписи избирателя. Избиратель заполняет бюллетень из значений 1 – «за» и 0 – «против», шифрует их с помощью ключа шифрования бюллетеней, формирует доказательство корректности содержимого бюллетеня, состоящее в том, что его выбор соответствует либо 0, либо 1. Также формируется доказательство корректности заполнения бюллетеня в целом.

Во-вторых, Provotum (Швейцария) [8]. Каждый сервер генерирует свой собственный открытый ключ (pk); на БЧ формируются общий открытый ключ (pk_{voting}). Далее избиратель заполняет бюллетень из значений, шифрует их с помощью общего ключа и формирует доказательство корректности содержимого бюллетеня, состоящее в том, что его выбор соответствует либо 0, либо 1. Система отличается от других тем, что каждый сервер генерирует свой собственный открытый ключ, а общий открытый ключ формируется в БЧ.

В-третьих, Helios (США) [4, 12]. Сначала сервер генерирует бланк – бюллетень; далее избиратель выбирает своего кандидата из значений (0, 1), и сервер шифрует выбор избирателя, используя открытый ключ; после чего отправляет все зашифрованные бюллетени к микс-серверу, который маскирует и перемешивает их. Микс-сервер также должен доказать, что правильно перемешал бюллетени.

Примечание. Во всех рассмотренных системах использована схема Эль-Гамаля [11] на эллиптической кривой для шифрования и расшифрования бюллетеней.

Для всех систем голосования существует достаточно много угроз, связанных с действиями нарушителя и неправомерными действиями участников протокола голосования [13–15]. В последнее время большое внимание при построении систем электронного голосования уделяется защите от угрозы преднамеренного или непреднамеренного неправильного заполнения бюллетеня голосования избирателем. Эта задача не является тривиальной, так как контроль правильности заполнения бюллетеня должен осуществляться в зашифрованном виде, без раскрытия того, как проголосовал избиратель.

В [16–18] рассматривается протокол электронного голосования с проверкой корректности заполнения бюллетеней. Протокол работает следующим образом: сначала избиратель шифрует свой бюллетень и получает криптограмму B_i. Далее он должен доказать, что в криптограмме зашифрованы значения $(0, 1)$. Для этого формируется доказательство корректности заполнения своего бюллетеня. Криптограмма и доказательство отправляется в избирательную комиссию, которая проверяет доказательства для (B_i): если проверка прошла успешно, то голос избирателя принимается. Далее комиссия расшифровывает и подсчитывает голоса.

В [19] предложена система ДЭГ, использующая гомоморфную схему. В этой работе доказательство корректности заполнения бюллетеня ИК и его проверка разработаны для общего случая, когда вариант выбора избирателя принадлежит заданному диапазону возможных значений. Сложность такого доказательства в значительной степени зависит от количества возможных вариантов голосования на выборах.

В [20] предложена система ДЭГ на основе гомоморфного шифрования, в которой для доказательства корректности заполнения бюллетеня использована схема перемешивания голосов, поданных за кандидатов. Эта схема основывается на работах [2, 3, 21], в которых представлены доказательства корректности выполнения этой процедуры.

Все проверки корректности заполнения бюллетеня выполняются с использованием неинтерактивных схем доказательств с нулевым разглашением секрета.

Целью работы является исследование методов защиты от угрозы неправильного заполнения бюллетеня избирателем в системе ДЭГ, оценка сложности их реализаций и рекомендации по их применению. В п. 1 приведена модель системы ДЭГ, на основе схемы шифрования Эль-Гамаля на эллиптической кривой и угроз, специфических для системы. В п. 2 приведено детальное описание метода проверки корректности заполнения бюллетеня, основанного на доказательстве с нулевым разглашением секрета в задаче дискретного логарифмирования. В п. 3 приведено описание метода проверки корректности заполнения бюллетеня на основе перемешивания голосов избирателя. Описание методов сопровождается числовыми примерами правильного и неправильного заполнения бюллетеня. В п. 4 проведен сравнительный анализ сложности реализации обоих методов и рекомендации по их использованию в системах ДЭГ.

1. Модель системы ДЭГ на основе схемы шифрования Эль-Гамаля на эллиптической кривой

Рассматриваемая в работе система ДЭГ включает в себя: избирателей, сервер, БЧ и ИК (рисунок 1).

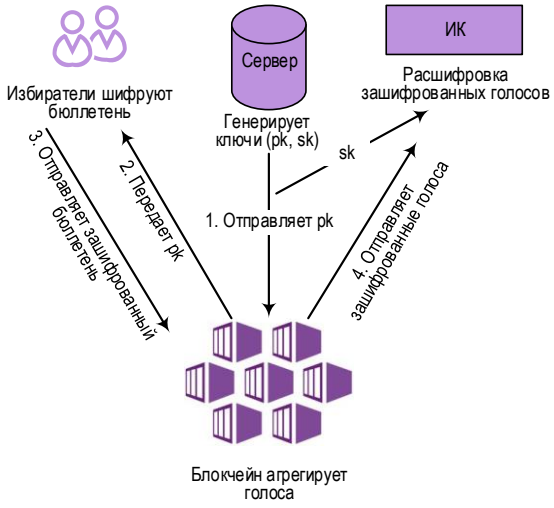

Рис. 1. Схема ДЭГ

Fig. 1. Remote Electronic Voting Scheme

Рассмотрим систему ДЭГ, построенную на основе гомоморфной системы шифрования Эль-Гамаля [11]. Под гомоморфным шифрованием понимается криптографический примитив, представляющий собой функцию шифрования, удовлетворяющую дополнительному требованию гомоморфности относительно каких-либо алгебраических операций над открытыми сообщениями. Свойство гомоморфного шифрования позволяет агрегировать голоса в зашифрованном виде и после расшифровать одну криптограмму, получив сразу результат голосования.

Основными этапами функционирования системы являются:

– инициализация системы;
– аутентификация избирателей;
– голосование и подсчет голосов;
– объявление результатов голосования.

Инициализация системы заключается в выборе системных параметров и генерации ключей. Сервер генерирует открытый и закрытый ключ для криптосистемы, использующей гомоморфное шифрование, и отправляет открытый ключ в БЧ, который передает открытый ключ всем избирателям. Сек-

ретный ключ хранится на сервере или может быть разделен на доли и находиться у хранителей ключа до окончания выборов. После того, как избиратель успешно пройдет этап идентификации и аутентификации, он получает разрешение на участие в голосовании (в работе процесс аутентификации и идентификации избирателя не рассматривается).

Каждый избиратель выбирает кандидата/кандидатов из списка, шифрует свой голос с помощью открытого ключа и отправляет его в БЧ. После завершения голосования в БЧ осуществляется агрегирование голосов, результаты отправляются в избирательную комиссию. Сервер, на котором генерировались открытый и закрытый ключи, передает закрытый ключ избирательной комиссии, а если было разделение ключа, доверенные лица передают свои доли ключа, комиссия, в свою очередь, восстанавливает закрытый ключ. Далее она расшифровывает результаты голосования с помощью закрытого ключа и объявляет итог.

Одна из угроз в данной системе ДЭГ заключается в том, что избиратель может неправильно (умышленно или случайно) заполнить свой бюллетень, и это повлияет на результаты голосования. Чтобы предотвратить эту угрозу, применяются различные методы проверки корректности заполнения бюллетеня. В работе проведен сравнительный анализ двух методов решения этой задачи: основанного на сравнении дискретных логарифмов [16–19] и на проверке корректности перестановки [2, 3, 20]. Оба метода относятся к задачам «доказательства с нулевым разглашением секрета» [22–26].

Рассмотрим далее модель системы ДЭГ на основе схемы гомоморфного шифрования Эль-Гамаля на эллиптической кривой [27–28]. Эта схема и параметры кривой будут далее использоваться для шифрования бюллетеня и выполнения других функций во всей работе.

Генерация ключей

Сервер генерирует эллиптическую кривую вида $y^2 = x^3 + xa + b$ над полем Галуа $GF(p)$ и выбирает базовую точку $P \in E\big(GF(p)\big)$ порядка m.

Сервер случайным образом выбирает закрытый ключ d, $d \in \{1, \dots, m-1\}$. Далее вычисляется открытый ключ: $Q = dP \bmod p$ и генерируется точка $F = rP \bmod$, где r – случайное число, выбираемое в диапазоне $[1, \dots, m-1]$.

Параметры p, E, m, P, F, Q публикуются в БЧ. Секретный ключ d хранится в избирательной комиссии в разделенном на доли виде.

Шифрование бюллетеня

Избиратель V_i, i = 1, 2, …, n, где n – количество избирателей, шифрует сообщение (бюллетень) M_i по схеме Эль-Гамаля с помощью открытого ключа и получает криптограмму:

$$\mathrm{Enc}(M_i) = C_i = (A_i, B_i), \quad (1)$$

где Enc() – функция шифрования; (A_i, B_i) – две части криптограммы C_i,: первая часть $A_i = rP \bmod p$; вторая часть $B_i = (M_i F + rQ) \bmod p$; r – выбирается случайным образом.

Дешифрование бюллетеня

Расшифрование криптограммы осуществляется с помощью закрытого ключа d:

$$\mathrm{Dec}(C_i) = B_i - dA_i \bmod p, \quad (2)$$

где Dec() – функция дешифрования.

Результат расшифровки должен быть равен сообщению M_i.

Криптосистема Эль-Гамаля на эллиптической кривой обладает гомоморфным свойством.

Допустим, есть два шифртекста:

$$C_1 = (A_1, B_1) = (r_1 P, F_1 + r_1 Q) \text{ и} \quad (3)$$

$$C_2 = (A_2, B_2) = (r_2 P, F_2 + r_2 Q). \quad (4)$$

Криптограммы могут быть агрегированы аддитивно:

$$C_3 = C_1 + C_2 = \\ = \big((r_1 + r_2)P,\ (F_1 + F_2) + (r_1 + r_2)Q\big). \quad (5)$$

Тогда при расшифровании C_3 получаем:

$$\mathrm{Dec}(C_3) = F_1 + F_2. \quad (6)$$

Рассмотрим далее способ заполнения бюллетеня.

Заполнение бюллетеня

Бюллетень в электронном виде представляет собой строку символов (1, 0). В зависимости от правил выборов могут быть различные варианты голосования. Например, избиратель может проголосовать за одного кандидата из k кандидатов, или он может проголосовать за двух и более кандидатов (t из N). Но он не может не голосовать. Могут быть и другие правила, установленные избирательной комиссией. Любые отклонения от установленных вариантов голосования, например, использование числа 2 или –1, поданных за какого-то кандидата, будут означать некорректное заполнение бюллетеня. Пример правильного заполнения бюллетеня показан в таблице 1. Избиратель подал голос «за» за первого и четвертого кандидатов, и голос «против» – за остальных кандидатов. Таким образом, бюллетень должен содержать только значения (1, 0). Для того, чтобы подтвердить, что он действительно заполнил свой бюллетень правильно, необходимо использовать методы доказательства корректности заполнения бюллетеня.

ТАБЛИЦА 1. Формирование правильного заполнения бюллетеня

TABLE 1. Formation of the Correct Filling of the Ballot

Кандидаты	*D*1	*D*2	*D*3	*D*4	……	*Dk*
Выбор избирателя	1	0	0	1	….....	0

2. Метод проверки корректности заполнения бюллетеня на основе проверки логарифмов

2.1. Проверка корректности заполнения бюллетеня для каждого шифртекста

Рассмотрим [19] протокол голосования, когда выбирается только один кандидат из k кандидатов. Избиратель может голосовать («за» одного и «против» остальных кандидатов). Проверка доказательств осуществляется на основе неинтерактивного метода с нулевым разглашением секрета (NIZKP, *аббр. от англ.* Non-Interactive Zero-Knowledge Proof) и заключатся в доказательстве сравнения вида:

$$ZP(x|y(x) = z), \quad (7)$$

где x – параметр, не неизвестный проверяющему; z – известная проверяющему величина.

В нашем случае нужно доказать, что для каждой криптограммы C_i выполняется сравнение:

$$ZP(r_i, b_{vi}|C_i = (r_iP, b_{vi}F_i + r_iQ) \vee (C_i = (r_iP, r_iQ)),$$

где b_{vi} – голос i-го избирателя, $b_{vi} \in \{0,1\}$; r_i – случайное число.

Алгоритм голосования, формирование доказательства корректности заполнения бюллетеня и проверки доказательства для вышерассмотренной схемы голосования включает следующие шаги и приведен в таблице 2.

Шаг 1. Загрузка открытого ключа из БЧ.

Шаг 2. Выбор своего кандидата.

Шаг 3. Шифрование бюллетеня по схеме Эль-Гамаля на эллиптической кривой.

Шаг 4. Формирование доказательства того, что он зашифровал свой бюллетень из значений (1, 0).

Последняя колонка (см. таблицы 2 и 3) содержит оценки сложности выполнения соответствующих операций. Символ M обозначает операцию умножения точки эллиптической кривой на целое число. Операции сложения точек не учитывались ввиду их меньшей сложности по сравнению с операцией умножения; H – сложность операции хеширования также не учитывалась.

Далее избиратель отправляет значения (A, B, a_1, b_1, a_2, b_2, u_1, u_2, t_1, t_2) проверяющему (в БЧ), где, согласно алгоритму из таблицы 3, проходит проверка того, что избиратель правильно заполнил свой бюллетень. Здесь же приведены оценки сложности выполнения алгоритма. Если все сравнения выполняются, значит избиратель правильно проголосовал за каждого кандидата при этом проверяющий (БЧ) не знает, как проголосовал избиратель.

ТАБЛИЦА 2. Формирование доказательства корректности заполнения бюллетеня

TABLE 2. Formation of Proof of Correctness of Filling in the Ballot

Избиратель: голосование и формирование доказательства			Оценки сложности (при выборе b_v = 1)
Голосует:	«за» кандидата – $b_{vi} = 1$	«против» кандидата – $b_{vi} = 0$	$O(1)$
Случайным образом выбирает числа $w, r_1, t_1, u_1 \in Z_q$.			$O(1)$
Осуществляет шифрование бюллетеня по каждому кандидату (вычисляет):	$A = (r_1P) \bmod p$; $B = (b_{vi}F + r_1Q) \bmod p$.	$A = (r_1P) \bmod p$; $B = (r_1Q) \bmod p$.	1kM 2kM
Формирует доказательство корректности голосования (вычисляет):	$a_1 = (t_1P - u_1A) \bmod p$; $b_1 = (t_1Q - u_1(B - b_{vi}\mathrm{P}) \bmod p$ $a_2 = wP \bmod p$; $b_2 = wQ \bmod p$.	$a_1 = \mathrm{wP} \bmod p$; $b_1 = wQ \bmod p$; $a_2 = (t_1\mathrm{P} - u_2\mathrm{A}) \bmod p$; $b_2 = (t_1\mathrm{Q} - u_2(\mathrm{B} - b_{vi}\mathrm{P}) \bmod p$	2kM 3kM 1kM 1kM
Вычисляет хэш-функцию $h = H(A, B, a_1, b_1, a_2, b_2) \bmod q$			1H
Вычисляет доказательство:	$h - u_1 \bmod q$; $t_2 = w - r_1u_2 \bmod q$.	$u_1 = h - u_2 \bmod q$; $t_1 = w - r_1u_1 \bmod q$.	$O(k)$ $O(k)$
Всего операций умножения точки эллиптической кривой на число			10kM

ТАБЛИЦА 3. Алгоритм проверки корректности голосования за кандидата

TABLE 3. Algorithm for Verifying the Correctness of Voting for a Candidate

Проверяющий (БЧ)			Оценки сложности
Вычисляет хэш-функцию $h = H(A, B, a_1, b_1, a_2, b_2)$			1H
Проверяет сравнения:	$h \bmod q =^? u_1 + u_2 \bmod q$; $t_1P \bmod p =^? a_1 + u_1A \bmod p$; $t_1Q \bmod p =^? b_1 + u_1(B - b_iP) \bmod p$.	(8) (9) (10)	$O(k)$ 2kM 3kM
Всего операций умножения точки эллиптической кривой на число			5kM

Примечание. В таблицах приняты следующие условные обозначения: H – сложность операции хеширования; M – операция умножения точки эллиптической кривой на целое число; k – количество кандидатов.

Рассмотрим примеры формирования и проверки доказательства корректности заполнения бюллетеня для варианта, когда выбирается один кандидат $D1$ из 4 кандидатов. Пусть на этапе инициализации системы ДЭГ выбраны параметры: $p = 59$, $q = 17$, $(a = 3, b = 9)$. Эллиптическая кривая является несингулярной и имеет следующие точки (выбор кривой и параметров шифрования носят иллюстрационный характер):

{(0, 3), (0, 56), (3, 24), (3, 35), (4, 12), (4, 47), (6, 19), (6, 40), (7, 14), (7, 45), (9, 23), (9, 36), (10, 6), (10, 53), (11, 4), (11, 55), (12, 11), (12, 48), (13, 11), (13, 48), (14, 9), (14, 50), (15, 19), (15, 40), (17, 28), (17, 31), (19, 9), (19, 50), (20, 24), (20, 35), (25, 29), (25, 30), (26, 9), (26, 50), (29, 0), (34, 11), (34, 48), (36, 24), (36, 35), (38, 19), (38, 40), (42, 1), (42, 58), (46, 29), (46, 30), (47, 29), (47, 30), (49, 10), (49, 49), (50, 16), (50, 43), (51, 2), (51, 57), (54, 20), (54, 39), (55, 13), (55, 46), (58, 8), (58, 51), $(\mathcal{O}, \mathcal{O})$}.

Используя схему Эль-Гамаля и выбрав базовую точку $P = (19, 9)$, генерируются ключи: закрытый – $d = 4$ и открытый – $Q = 4(19, 9) \bmod 59 = (6, 19)$. Далее выбирается случайным образом $r = 3$ и вычисляется $F = rP \bmod p = 3(19, 9) \bmod 59 = (54, 39)$. Параметры p, E, m, P, F, Q публикуются в БЧ. (Порядок выполнения операций сложения и умножения точек эллиптической кривой на целое число можно найти в [29]).

Примечание. В дальнейшем будем предполагать, что для вычисления хэш-функции используется некоторый алгоритм, вырабатывающий по заданному аргументу число, которое мы в числовых примерах указываем произвольно.

Рассмотрим два «полярных» случая.

Случай 1. Избиратель правильно заполнил свой бюллетень:

– избиратель V_1 голосует «за» $(b_{v1} = 1)$, выбирает случайным образом $r_1 = 2$;

– шифрует свой бюллетень:

$$(A_1, B_1) = (r_1 P, b_{v1} F + r_1 Q) \bmod p;$$

$\big(2(19, 9) + 1(52, 2) + 2(6, 19)\big) \bmod 59 =$ ((49, 10), (6, 40));

– вычисляет доказательство для криптограммы $(A_1, B_1) = ((49, 10), (6, 40))$; если $b_{v1} = 1$, выполняет вычисления согласно второму столбцу таблицы 2:

а) случайным образом выбирает числа: $t_1 = 2$, w $= 2, u_1 = 5$;

б) вычисляет:

$$a_1 = \big(2(19, 9) - 5(49, 10)\big) \bmod 59 = (34, 48);$$

$$b_1 = (2(6, 19) - 5\big((6, 40) - 1(19, 9)\big) \bmod 59 = (54, 39);$$

$$a_2 = 2(19, 9) \bmod 59 = (49, 10);$$

$$b_2 = 2(6, 19) \bmod 59 = (34, 11).$$

Предположим, что хеширование параметров $(A_1, B_1, a_1, b_1, a_2, b_2)$ дает $h = 3$:

– вычисляет:

$u_2 = 3 - 5 \bmod 17 = 15; t_2 = 2 - 2 \times 15 \bmod 17 = 6;$

– отправляет в БЧ зашифрованный бюллетень:

$$(A_1 = (49, 10), B_1 = (6, 40))$$

и доказательство:

$$\begin{pmatrix} a_1 = (34, 48), b_1 = (54, 39), a_2 = (49, 10), \\ b_2 = (34, 11), u_1 = 5, u_2 = 15, t_1 = 2, t_2 = 6 \end{pmatrix}.$$

Таким же образом шифруются голоса для остальных кандидатов:

$$C_2 = ((6, 19), (19, 9)), C_3 = ((51, 2), (51, 2)),$$

$$C_4 = ((34, 48), (49, 49)),$$

где криптограммы C_2, C_3, C_4 являются зашифрованными значениями точки $\mathcal{O}$ и вычисляются доказательства для этих криптограмм согласно третьему столбцу таблицы 2.

БЧ проверяет, что избиратель правильно заполнил свой бюллетень, выполняя сравнение согласно таблице 3.

Для нашего примера – БЧ:

– вычисляет хэш-функцию $h = 3$, находит:

$$(u_1 + u_2) \bmod q = (5 + 15) \bmod 17 = 3$$

(видим, что сравнение (8) выполняется: $3 = 3$);

– находит $t_1 P \bmod p = (49, 10)$ и $a_1 + u_1 A_1 \bmod p = (49, 10)$;

– проверяет, что сравнение (9) выполняется:

$$t_1 P \bmod p \stackrel{?}{=} a_1 + u_1 A_1 \bmod p; (49, 10) = (49, 10).$$

– вычисляет:

$$t_1 \mathrm{Q} \bmod p = (34, 11)$$

и

$$b_1 + u_1 (B_1 - b_{vi} P) \bmod p = (34, 11);$$

(видим, что сравнение (10) выполняется: $(34, 11) = (34, 11)$.

Таким образом, все сравнения выполнены, следовательно, корректность заполнения бюллетеня для $D1$ доказана.

Аналогично проверяются доказательства корректности голосования за других кандидатов.

Случай 2. Избиратель неправильно заполнил свой бюллетень.

Пусть избиратель поставил число 2 за $D1$. Все шаги алгоритма аналогичны предыдущему примеру:

– при выборе в $b_{v1} = 2$ находит:

$$(A_1, B_1) = (r_1 P, b_{v1} F + r_1 Q) \bmod p =$$

$$= \big((54, 39), (54, 39)\big);$$

– формирует доказательство:

$$a_1 = (6, 40), b_1 = (11, 4), a_2 = (49, 10), b_2 = (34, 11),$$

$$h = 5, u_2 = 3, t_2 = 13.$$

БЧ проверяет доказательство корректности заполнения бюллетеня, проверяя сравнения согласно таблице 3:

– сравнение (8) выполняется: $h \bmod q \stackrel{?}{=} u_1 + u_2 \bmod q; 5 = 5;$

– сравнение (9) выполняется: $(49, 10) = (49, 10);$

– сравнение (10) не выполняется: $(34, 11) \neq \neq (49, 10)$.

Видно, что не все сравнения выполнены, следовательно, корректность заполнения бюллетеня для $D1$ не доказана.

2.2. Проверка корректности заполнения всего бюллетеня

В случае, рассмотренном выше, контролирующий орган может убедиться, что избиратель корректно проголосовал за каждого кандидата («за» или «против»). Но он не может проверить, выполнены ли правила голосования по заданному варианту голосования. То есть, например, избиратель может выбрать трех кандидатов, хотя разрешено выбрать только одного или двух. Эта задача решается проверкой корректности заполнения бюллетеня в целом (см. URL: https://evoting.digitaldem.ru/wp-content/uploads/sites/2/2021/07/ptkdeg_general_description_2021-07-15.pdf). Рассмотрим этот метод.

Пусть $k_{\max}$ – максимальное число голосов «за», при голосовании за k кандидатов. Будем считать, что ключи (открытый, закрытый) сгенерированы, избиратель выполнил следующие действия:

– выбрал кандидатов;

– зашифровал бюллетень с помощью открытого ключа: $C_i = (A_i, B_i) \bmod p$, где $A_i = r_i P \bmod p$; $B_i = F + r_i Q \bmod p$, если $F = b_{vi} P \bmod p$, b_{vi} – выбор избирателем кандидата, $b_{vi} \in \{0,1\}, i = 1, 2, \ldots, k$.

– сформировал доказательство корректности голосования за каждого кандидата, как было описано выше.

Рассмотрим подробно формирование доказательства корректности заполнения бюллетень.

Избиратель вычисляет сумму криптограмм бюллетеня для всех кандидатов:

$$C_\Sigma = (A_\Sigma, B_\Sigma), \tag{11}$$

где $A_\Sigma = \sum_{i=1}^{k} A_i$, $B_\Sigma = \sum_{i=1}^{k} B_i$, $r = \sum r_i, m = \sum m_i$, m – сумма голосов «за», поданных избирателем в пользу всех кандидатов.

Выполняет следующий алгоритм:

1) находит:

$$T = t \cdot Q, \tag{12}$$

где $t \in Z_p$ – случайное число;

2) вычисляет хэш-функцию:

$$h = H(Q, A_\Sigma, B_\Sigma, T, m); \tag{13}$$

3) вычисляет:

$$s = t + r \cdot h; \tag{14}$$

4) посылает в БЧ (T, s, m').

Избиратель с целью обмана может указать суммарное число голосов «за», поданных в пользу всех кандидатов m', отличное от фактического числа голосов m, если $m > k_{\max}$.

БЧ вычисляет: $h = H(Q, \sum A_i, \sum B_i, T, m')$, для чего используются криптограммы $C_i = (A_i, B_i)$ из бюллетеня.

Далее БЧ проверяет сравнение:

$$sQ \stackrel{?}{=} T + h\left(\sum_{i=1}^{k} B_i - m' F_i\right). \tag{15}$$

Если сравнение выполнятся, то $m = m'$. Покажем, что это действительно так:

$$T + h\left(\sum_{i=1}^{k} B_i - m' F_i\right) = tQ + h\left(mF + r_\Sigma Q - m' F\right) =$$

$$= tQ + r_\Sigma \mathrm{h} Q + h(mF - m' F) = sQ + h(mF - m' F) = sQ.$$

Сравнение выполняется.

Видим, что если $m = m'$ и $m' \leq k_{\max}$, то избиратель проголосовал правильно.

Сложность данного алгоритма формирования и проверки доказательства корректности заполнения бюллетеня в целом можно оценить на основе вышеприведенных соотношений так:

– количество умножений точки эллиптической кривой на число на стороне избирателя – 1М;

– количество умножений точки эллиптической кривой на число в БЧ – 3М.

Рассмотрим примеры формирования и проверки доказательства корректности заполнения всего бюллетеня.

Пример 1. Избиратель правильно заполнил бюллетень.

Пусть, согласно регламенту, избиратель V_i может проголосовать «за» за одного или двух из четырех кандидатов $D1, D2, D3, D4$, и он выбрал двух кандидатов и вычислил криптограммы:

$C_1 = ((19, 9), (34, 48))$, $C_2 = ((49, 10), (34, 11))$,

$C_3 = ((54, 39), (54, 20))$ и $C_4 = ((6, 19), (19, 9))$,

$C_\Sigma = ((51, 57), (49, 49))$, $r = \sum r_i = 10$, $m = \sum m_i = 2$.

Далее он создает доказательство корректности голосования, выполняя следующий алгоритм:

– находит $T = t \cdot Q$, где $t \in Z_p$ случайное число, $t = 2$, $T = t \cdot Q = 2(6, 19) \bmod 59 = (34, 11)$;

– вычисляет хэш-функцию:

$$h = H\left(Q, A_\Sigma, B_\Sigma, T, m\right) = 9;$$

– вычисляет $s = t + r \cdot h; s = 2 + 10 * 9 \bmod 17 = 7$;

– посылает в БЧ $(T = (34, 11), s = 7, m' = 2)$.

Далее БЧ вычисляет $h = H(Q, \sum A_i, \sum B_i, T, m') = 9$, и проверяет сравнение $sQ \stackrel{?}{=} T + h(\sum_{i=1}^{k} B_i - m' F_i)$:

$$sQ \bmod p = 7(6, 19) \bmod 59 = (49, 49);$$

$$T + h\left(\sum_{i=1}^{k} B_i - m' F_i\right) = (34,11) +$$

$$+ 9((49, 49) - 2(54, 39) = (49, 49),$$

$(49, 49) = (49, 49)$: сравнение выполняется.

Пример 2. Избиратель неправильно заполнил бюллетень.

Пусть избиратель V_i проголосовал «за» в пользу трех из четырех кандидатов $D1$, $D2$, $D3$, $D4$, хотя, согласно правилу, он может проголосовать за одного или двух из четырех кандидатов ($k_{\max} = 2$).

Этому выбору соответствуют криптограммы:

$$C_1 = ((19,9),(34,48)),\ C_2 = ((49,10),(6,40)),$$
$$C_3 = ((6,19),(6,19)),\ C_4 = ((54,39),(54,20)),$$
$$C_\Sigma = ((51,57),(6,19)),\ r = \sum r_i = 10,\ \ m = \sum m_i = 3.$$

Избиратель выполняет следующий алгоритм:

– находит $T = t \cdot Q$, где $t \in Z_p$ случайной число, $t = 2, T = t \cdot Q = 2(6,19) \bmod 59 = (34,11)$;

– вычисляет хэш-функцию:

$$h = H\left(Q, A_\Sigma, B_\Sigma, T, m'\right) = 4;$$

– вычисляет $s = t + r \cdot h$; $s = 2 + 10 * 14 \bmod 17 = 6$;

– посылает в БЧ ($T = (34,11)$, $s = 6$, $m' = 2$): избиратель, чтобы скрыть, что он проголосовал неправильно, посылает значение $m' = 2$.

Далее БЧ вычисляет $h = H(Q, \sum A_i, \sum B_i, T, m') = 4$ и проверяет сравнение $sQ \stackrel{?}{=} T + h\left(\sum_{i=1}^{k} B_i - m'F_i\right)$:

$$sQ \bmod p = 6(6,19) \bmod 59 = (11,55);$$

$$T + h\left(\sum_{i=1}^{k} B_i - m'F_i\right) = (34,11) +$$
$$+ 4\left((6,19) - 2(54,39)\right) = (11,4),$$

то есть сравнение $(11,55) \neq (11,4)$ не выполняется. Следовательно, обнаружено некорректное заполнение бюллетеня.

3. Метод проверки корректности заполнения бюллетеня на основе перемешивания криптограмм бюллетеня

Идея этого метода [20] заключается в следующем: сначала сервер генерирует бланк – бюллетень, представляющий вектор C из зашифрованных следующим образом криптограмм:

$$C = (C_1, .., C_k).$$

Первая криптограмма вычисляется как:

$$C_1 = (\rho_1 P, F + \rho_1 Q) \bmod p, \qquad (16)$$

где P – базовая точка; $Q = dP \bmod p$ – открытый ключ, и точка $F = M_i P \bmod p$; $P, Q, F \in E_P(GF(P))$.

Остальные криптограммы вычисляют как:

$$C_i = (\rho_i P, \rho_i Q) \bmod p, \qquad (17)$$

где ρ_i выбирается случайным образом, $\rho_1 \in Z_p$.

Сервер публикует C_i и ρ_i на БЧ.

Избиратель для голосования считывает из БЧ бланк-бюллетень и выполняет следующее:

1) убеждается, что информация, полученная с БЧ, корректна; для этого избиратель проверяет, что $\rho_i P = A_i$ и вычисляет $\mathrm{Rev}_r(C_i) = B_i - \rho_i Q$ – в результате должно получиться либо точка F, либо точка $\mathcal{O}$;

2) приступает к голосованию:

– выбирает своего кандидата – Ds;

– выбирает перестановку $\pi(s, i_1, i_2, \ldots, i_{k-1})$;

– перемешивает C в соответствии с выбранной перестановкой и маскирует бюллетень:

а) генерирует случайным образом набор целых чисел $r_i \in Z_p$;

б) вычисляет:

$$C_i' = C_{\pi(i)} + (r_i P, r_i Q) = \left(A_{\pi(i)} + r_i P, B_{\pi(i)} + r_i Q\right) =$$
$$= \left((\rho_i + r_i)P, F_i + (\rho_i + r_i)Q\right) \bmod p,$$

где $i = 1, 2, \ldots, k$, причем $F_i = \mathcal{O}$ для $i = 2, \ldots, k$.

C_i' отправляет в БЧ;

– формирует доказательство корректности перемешивания бюллетеня, для чего:

– получает от БЧ выбранные случайным образом числа s_i и s_i', $s_i, s_i' \in \{0, 1, .., 2^L - 1\}$;

– вычисляет числа $t_i = s_{\pi(i)}, t_i' = s'_{\pi(i)}$,

– генерирует случайным образом набор целых чисел $r'_i \in Z_p$;

– вычисляет $C_i'' = t_i C_i' + (r_i' P + r_i' Q) = (A_i'', B_i'') = (t_i A_i' + r_i' P, t_i B_i' + r_i' Q)$;

– отправляет C', C'', t_i и t_i' в БЧ.

Проверка доказательства заключается в проверке выполнения сравнений [2, 3, 20]:

$$\sum_{i=1}^{k} \mathrm{Dec}(C_i) \times s_i \stackrel{?}{=} \sum_{i=1}^{k} \mathrm{Dec}(C_i') \times t_i, \qquad (18)$$

$$\sum_{i=1}^{k} \mathrm{Dec}(C_i) \times s_i' \stackrel{?}{=} \sum_{i=1}^{k} \mathrm{Dec}(C_i') \times t_i', \qquad (19)$$

$$\sum_{i=1}^{k} \mathrm{Dec}(C_i) \times s_i \times s_i' \stackrel{?}{=} \sum_{i=1}^{k} \mathrm{Dec}(C_i') \times t_i \times t_i'. \qquad (20)$$

Однако непосредственная проверка согласно (18–20) невозможна, так как для этого БЧ должен знать закрытый ключ d. Поэтому проверка доказательств осуществляется на основе NIZKP. Доказательство (18–20) заключается в проверке следующих равенств [3]:

$$ZP(t_i, r_i' | C_i'' = ((t_i C_i' + (r_i' P, r_i' Q)), \qquad (21)$$

где

$$C_i = (A_i, B_i) = (\rho_i P, F_i + \rho_i Q),$$
$$C_i' = (r_i + r_i')P, F_i + (r_i + r_i')Q,$$
$$C_i'' = t_i A_i' + r_i' P, t_i B_i' + r_i' Q),$$
$$ZP(t_i, r_i' r_i | \sum_{i=1}^{k} (t_i\left(C_i \cdot s_i + (r_i P, r_i Q)\right) + (r_i' P, r_i' Q) = \qquad (22)$$
$$= \sum_{i=1}^{k} C_i'',$$

$$ZP(r_i, r_i', t_i, t_i' | \sum_{i=1}^{k} (t_i'(C_i \cdot s_i' + (r_i P, r_i Q)) = \sum_{i=1}^{k} C_i' t_i', \quad (23)$$

$$ZP(r_i, r_i', t_i, t_i' | \sum_{i=1}^{k} (t_i t_i'(C_i \cdot s_i s_i' + (r_i P, r_i Q)) + \\ + t_i'(r_i' P, r_i' Q) = \sum_{i=1}^{k} C_i'' t_i. \quad (24)$$

Проверку сравнений (21–24) будем проводить отдельно для каждой части криптограммы $C_i'' = (A_i'', B_i'')$,

Для проверки (12) необходимо доказать:

$$A_i'' = A_i' t_i + r_i' P, \ B_i'' = B_i' t_i + r_i' Q.$$

Покажем это для A_i''.

Избиратель формирует доказательство следующем образом:

– выбирает случайные числа $z_i, u_i \in Z_p$, вычисляет:

$$L_i = z_i P \bmod p, \ J_i = u_i A_i' \bmod p \quad (25)$$

и находит хеш-функцию $h = H(A_i'', P, L_i, J_i)$;

– вычисляет:

$$\theta_i = z_i + r_i' h_i \bmod q, \alpha_i = u_i + t_i h_i, \\ T_i = \theta_i P + \alpha_i A_i' \bmod p; \quad (26)$$

– пересылает в БЧ (T_i, L_i, J_i).

БЧ вычисляет хеш-функцию $h' = H(A_i'', P, L_i + J_i)$ и проверяет сравнение: $L_i + J_i + h' \cdot A_i'' =^? T_i$. (27)

Покажем, что если перемешивание выполнено правильно и $h = h'$, то сравнение выполняется. Для этого вычислим левую часть:

$$(L_i + J_i + h \cdot A_i'' = z_i P + u_i \cdot A_i' + h(t_i A_i' + r_i P) = \\ = z_i P + h \cdot r_i' P + u_i \cdot A_i' + t_i \cdot h \cdot A_i' = \theta_i P + \alpha_i A_i'.$$

Видно, что левая часть совпала с правой частью $T_i = \theta_i P + \alpha_i A_i'$. Сравнение (26) для A_i'' доказано.

Затем БЧ проверяет ZP (22) для первых частей криптограмм C_i''.

Избиратель генерирует случайное число $w \in Z_p$. Далее вычисляет:

$$T = wP \bmod p; \quad (28)$$

$$r_\Sigma = \sum_{i=1}^{k} r_i t_i + r_i', U = r_\Sigma P \bmod p; \quad (29)$$

хеш-функцию $h = H(P, T, U, A_1'', A_2'', .., A_k'')$;

$$z = w - r_\Sigma \cdot h \bmod q. \quad (30)$$

После чего отправляет в БЧ (T, z).

БЧ вычисляет:

$$U' = \sum_{i=1}^{k} A_i'' - \sum_{i=1}^{k} s_i A_i; \quad (31)$$

хеш-функцию $h' = H(P, T, U, A_1'', A_2'', .., A_k'')$;

$$T' = zP + h'U'. \quad (32)$$

Если $T = T'$, то (22) для первой части криптограмма C_i'' доказано.

Покажем, что это действительно так:

$$U' = \sum_{i=1}^{k} A_i'' - \sum_{i=1}^{k} s_i A_i = \sum_{i=1}^{k} t_i A_i' + r_i' P - \\ - \sum_{i=1}^{k} s_i A_i = \sum_{i=1}^{k} t_i \left(A_{\pi(i)} + r_i P\right) + r_i' P - \sum_{i=1}^{k} s_i A_i = \\ = \sum_{i=1}^{k} t_i A_{\pi(i)} + \sum_{i=1}^{k} t_i r_i P + r_i' P - \sum_{i=1}^{k} s_i A_i = \\ = \sum_{i=1}^{k} t_i A_{\pi(i)} - \sum_{i=1}^{k} s_i A_i + \sum_{i=1}^{k} (t_i r_i + r_i') P = \\ - \sum_{i=1}^{k} s_{\pi(i)} A_i - \sum_{i=1}^{k} s_i A_i + \left(\sum_{i=1}^{k} t_i r_i + r_i'\right) P.$$

Так как для перестановки π():

$$\sum_{i=1}^{k} s_{\pi(i)} A_{\pi(i)} - \sum_{i=1}^{k} s_i A_i = 0,$$

то

$$U' = \left(\sum_{i=1}^{k} t_i r_i + r_i'\right) P = r_\Sigma P. \quad (33)$$

Далее $T' = zP + h'U' = \left(w - r_\Sigma h\right) P + h' r_\Sigma P$. Если $h' = h$, то $T' = T$.

Аналогично проверяются сравнения (23) и (24).

Заметим, что подсчет голосов в такой системе осуществляется на сервере путем покомпонентного агрегирования координат векторов C_i', полученных от всех избирателей, принявших участие в выборах. В этом случае сумма $\sum_{i=1}^{n} C'_1 = \sum_{i=1}^{n} (C_1 v_i)$ – количество голосов (в зашифрованном виде), поданных за первого кандидата ($v_i = (1, 0)$), $\sum_{i=1}^{n} C'_2 = \sum_{i=1}^{n} (C_1 v_i)$ – количество голосов, поданных за второго кандидата и т. д. Расшифрование агрегированных голосов осуществляется избирательной комиссией с использованием секретного ключа d. На основе гомоморфного свойства схемы шифрования Эль-Гамаля получим расшифровку криптограмм, поданных, например, за i-го кандидата: $\mathrm{Dec}(\sum_{i=1}^{n} C'_i) = R_i$.

Логарифмируя это выражение, найдем сумму голосов (R_1), поданных за i-го кандидата. Победителем на выборах будет кандидат, набравший наибольшую сумму голосов – $\max(R_i)$. В таблицах 4, 5 приведены оценки сложности выполнения проверки корректности заполнения бюллетеня на основе перестановок на стороне избирателя и в БЧ.

Рассмотрим пример формирования и проверки доказательства правильности перемешивания бюллетеня. Пусть избиратель V_i может голосовать только за одного из четырех кандидатов $D1$, $D2$, $D3$, $D4$.

ТАБЛИЦА 4. Оценка сложности метода проверки корректности заполнения бюллетеня на основе перестановок

TABLE 4. Evaluation of the Complexity of the Method of Verifying the Correctness of Filling out the Ballot Based on Permutations

Операции, выполняемые избирателем	Оценка сложности для k-кандидатов
1) Проверка C_i, принятых от БЧ, $r_iP = A_i$ и вычисление $\text{Rev}_r(C_i) = B_i - r_iQ$;	2kM
2) Вычисление: $C_i' = ((r_i + r_i')P, F_i + (r_i + r_i')Q)$ и $C_i'' = (t_iA_i' + r_i'P, t_iB_i' + r_i'Q)$;	6kM
3) Вычисление точек эллиптической кривой $L_i = z_iP, J_i = u_iA_i', T_i = \theta_iP + \alpha_iA_i'$ для доказательства (21) для первой части криптограммы A_i (аналогично для второй части B_i).	8kM
4) Вычисление точек эллиптической кривой $U = r_\Sigma P, T = wP$ для доказательства (21–24).	6M
Всего	16kM + 6M

ТАБЛИЦА 5. Оценка сложности процедуры проверки корректности перемешивания бюллетеня

TABLE 5. Evaluation of the Complexity of the Procedure for Checking the Correctness of Mixing the Ballot

Операции, выполняемые БЧ	Оценка сложности
1) Вычисление левой части сравнения (21) $L_i + J_i + h'A_i'' = T_i$ для доказательства (21)	2kM
2) Вычисление $U' = \sum_{i=1}^{k} A_i'' - \sum_{i=1}^{k} s_iA_i$ и левой части сравнения $zP + h'U' = T$.	1kM 2M
Всего	3kM + 2M

Сервер генерирует бланк – бюллетень, содержащий вектор C из зашифрованных криптограмм: $C = (C_1, C_2, C_3, C_4)$ в соответствии с (16, 17), где F – точка на эллиптической кривой такая же, как в п. 3. $C_i = (\rho_iP, \rho_iQ) \bmod p, i = 2, 3, 4$:

$$C_1 = (\rho_1P, F + \rho_1Q) \bmod p = ((49, 10), (6, 40)),$$
$$C_2 = \big((19, 9), (6, 19)\big), C_3 = \big((54, 39), (54, 20)\big),$$
$$C_4 = \big((6, 19), (19, 9)\big).$$

Сервер публикует:

C = ((49, 10), (6, 40)), ((19, 9), (6, 19)), ((54, 39), (54, 20)), ((6, 19), (19, 9)) и $\rho_i = \{2, 1, 3, 4\}$ в БЧ.

Рассмотрим два случая голосования.

Случай 1. Избиратель правильно перемешал голоса в бюллетене.

Избиратель считывает из БЧ бланк-бюллетень для голосования и убеждается, что информация, полученная от БЧ, корректна. Для этого он выполняет проверки: $\rho_1P = A_i$;$\text{Rev}_r(C_i) = B_i - \rho_1Q = F$.

Для нашего примера

$$C_1 = (A_1, B_1) = ((49, 10), (6, 40)).$$

Проверяет сравнение $A_1 =^? \rho_1P \bmod p$; $(49, 10) = (49, 10)$ и $\text{Rev}_r(C_1) = B_1 - \rho_1Q = (54, 39) = F$.

$$C_2 = (A_2, B_2) = ((19, 9), (6, 19)).$$

Аналогично проверяет $A_2 =^? \rho_2P \bmod p$; $(19, 9) = (19, 9)$; и $\text{Rev}_r(C_2) = B_2 - \rho_2Q = \mathcal{O}$.

$$C_3 = (A_3, B_3) = ((54, 39), (54, 20)).$$

Проверяет $A_3 =^? \rho_3P \bmod p$; $(54, 39) = (54, 39)$ и $\text{Rev}_r(C_3) = B_3 - \rho_3Q = \mathcal{O}$.

$$C_4 = (A_4, B_4) = ((6, 19), (19, 9)).$$

Проверяет $A_4 =^? \rho_4P \bmod p$; $(6, 19) = (6, 19)$ и $\text{Rev}_r(C_4) = B_4 - \rho_4Q = \mathcal{O}$.

Все проверки выполнены правильно.

Далее избиратель приступает к голосованию. Во-первых, выбирает своего кандидата – $D4$. Во-вторых, выбирает перестановку: $\pi(1) = 4, \pi(2) = 2, \pi(3) = 3, \pi(4) = 1$. Для этого перемешивает координаты C в соответствии с выбранной перестановкой (таблица 6) и осуществляет маскировку бюллетеня:

– генерирует случайным образом набор целых чисел $r_i = \{4, 2, 1, 3\}$;

– вычисляет $C_i' = C_{\pi(i)} + (r_iP, r_iQ) = (A_{\pi(i)} + r_iP, B_{\pi(i)} + r_iQ) = ((\rho_1 + r_i)P, F_i + (\rho_1 + r_i)Q) \bmod p$, получает:

$\{C_i'\}$ = {((11, 4), (11, 55)), ((54, 39), (54, 20)), ((6, 19), (19, 9)), ((34, 48), (19, 9))}.

В-третьих, формирует доказательство корректности перемешивания бюллетеня. Для этого избиратель получает от БЧ случайным образом выбранные им числа s_i и s_i', где $i = 1, .., 4$. Пусть ($s_1 = 2, s_2 = 3, s_3 = 1, s_4 = 4, s_1' = 1, s_2' = 3, s_3' = 2, s_4' = 4$. Далее избиратель вычисляет числа $t_i = s_{\pi(i)}$, $t_i' = s_{\pi(i)}', i = 1, 2, \dots, k$. Тогда $t_1 = 4, t_2 = 1, t_3 = 3, t_4 = 2$. $t_1' = 4, t_2' = 2, , t_3' = 3, t_4' = 1$. А потом выбирает $r_i' = \{1, 3, 4, 2\}$ и вычисляет:

$$C_i'' = (t_iA_i' + r_i'P, t_iB_i' + r_i'Q),$$

$\{C_i''\}$ = {((6, 40), (19, 50)), ((11, 55), (11, 4)), ((19, 9), (6, 19)), ((19, 9), (51, 57))}

и отправляет C_i', C_i'', t_i и t_i' в БЧ.

ТАБЛИЦА 6. Избиратель правильно перемешал свой бюллетень

TABLE 6. The Voter Shuffled His Ballot Correctly

Кандидаты	$D1$	$D2$	$D3$	$D4$
Криптограммы, составляющие C_i	$C4$	$C2$	$C3$	$\boldsymbol{C1}$

Далее проверим выполнение сравнения (21). Для первой части криптограммы C_1', необходимо доказать, что $A_1'' = A_1't_1 + r_1'P$. В нашем примере мы получили:

$$C_1' = (r_1 + r_1')P, F + (r_1 + r_1')Q = ((11, 4), (11, 5));$$
$$C_1'' = (t_1A_1' + r_1'P, t_1B_1' + r_1'Q) = ((6, 40), (19, 50)).$$

Избиратель формирует доказательство для этой криптограммы следующим образом:

1) выбирает случайные числа $z_1 = 2, u_1 = 1$, вычисляет $L_1 = z_1P \bmod p = (49{,}10), J_1 = u_1A_1' \bmod p = (54{,}20)$ и находит хеш-функцию:

$h_1 = H\big((6, 40), (19, 9), (19, 50)\big) \bmod 17 = 16$;

2) вычисляет:
$\theta_1 = z_1 + r_1' h_1 \bmod q = 2 + 1 * 16 \bmod 17 = 1;$
$\alpha_1 = u_1 + t_1 h_1 \bmod q = 1 + 4 * 16 \bmod 17 = 14;$
$T_1 = \theta_1 P + \alpha_1 A_1' \bmod p = 1(19,9) + 14(11,4) \bmod 59 = (6,40);$

3) пересылает в БЧ ($T_1 = (6,40)$, $L_1 = (49,10)$, $J_1 = (54,20)$).

БЧ вычисляет хеш-функцию $h' = H(A_1'' = (6,40), P = (19,9), L_1 + J_1 = (19,50)) \bmod 17 = 16$ и h_i', а также проверяет сравнение $L_1 + J_1 + h' \cdot A_1'' =? T_1$; $(6,40) = T_1$. Таким образом доказано, что сравнение (21) выполняется для первой части криптограммы C_1.

Аналогично проверяем доказательство (21) для A_2, A_3, A_4.

Проверим выполнение *ZP* (22). Избиратель выполняет следующие действия:

– генерирует случайное число $w = 3$;

– вычисляет $T = wP \bmod p = (54,39)$;

– вычисляет
$r_\Sigma = \sum_{i=1}^{k} r_i t_i + r_i' = 3$, $U = r_\Sigma P \bmod p = (54,39)$;

– вычисляет хеш-функцию $h = H(P = (19,9), T = (54,39), U = (54,39), A_1'' = (6,40), A_2'' = (11,55), A_3'' = (19,9), A_4'' = (19,9)) \bmod 17 = 10$;

– вычисляет $z = w - r_\Sigma \cdot h \bmod q = 6$;

– посылает в БЧ ($T = (54,39), z = 6$).

Далее БЧ вычисляет:

$$U' = \sum_{i=1}^{k} A_i'' - \sum_{i=1}^{k} s_i A_i = \left(\sum_{i=1}^{k} t_i r_i + r_i'\right) P = r_\Sigma P = (54,39);$$

– хеш-функцию $h' = 10$;

– $T' = zP + h'U' = (54,39)$;

– проверяет $T =^? T'$; $(54,39) = (54,39)$, т. е. (22) для первой части криптограмм C_i'' доказано.

Случай 2. Избиратель неправильно перемешал голоса в бюллетене.

Все шаги выполняются, как в предыдущем примере, до момента перемешивания. Избиратель выполняет перестановку π, как показано в таблице 7.

ТАБЛИЦА 7. Избиратель неправильно перемешал свой бюллетень

TABLE 7. The Voter Mixed His Ballot Incorrectly

Кандидаты	*D1*	*D2*	*D3*	*D4*
Криптограммы, составляющие C_1	*C1*	***C1***	*C3*	*C4*

$\{C_i\} = \{((19,9),(6,19)), ((19,9),(6,19)), ((54,39),(54,20)), ((6,19),(19,9))\}$.

Далее осуществляет маскировку бюллетеня, генерирует случайным образом набор целых чисел $r_i = \{4, 2, 1, 3\}$, вычисляет C_i':

$$C_i' = C_{\pi(i)} + (r_i P, r_i Q) = (A_{\pi(i)} + r_i P, B_{\pi(i)} + r_i Q) = ((\rho_i + r_i)P, F_i + (\rho_i + r_i)Q) \bmod p,$$

для $i = 2, .., k$, получает:

$\{C_i'\} = \{((11,4),(54,20)), ((54,39),(11,4)), ((6,19),(19,9)), ((34,48),(19,9))\}$.

Следующий шаг – формирование доказательства корректности перемешивания бюллетеня. Для этого избиратель получает от БЧ случайным образом выбранные числа s_i и s_i', где $i = 1, .., n$: пусть $(s_1 = 2, s_2 = 3, s_3 = 1, s_4 = 4, s_1' = 1, s_2' = 3, s_3' = s_3' = 2, s_4' = 4)$, вычисляет числа t_i, t_i', как в предыдущем примере, выбирает $r_i' = \{1, 2, 3, 4\}$ и вычисляет:

$$C_i'' = (t_i A_i' + r_i' P, t_i B_i' + r_i' Q),$$

$\{C_i''\} = \{((6,40),(19,50)), ((11,55),(11,4)), ((19,9),(6,19)), ((19,9),(51,57))\}$,

после чего отправляет $C_i', C_i'' =$, t_i и t_i' в БЧ.

Далее проверим доказательства (21) для первых частей криптограммы C_1', для этого покажем, что $A_1'' = A_1' t_1 + r_1' P$.

Ранее было получено:

$C_1' = (r_1 + r_1')P$, $F + (r_1 + r_1')Q = ((11,4),(54,20))$ и $C_1'' = (t_1 A_1' + r_1' P, t_1 B_1' + r_1' Q) = ((6,40),(19,50))$.

Избиратель формирует доказательство для каждой криптограммы:

– выбирает случайные числа $z_1 = 2, u_1 = 1$, вычисляет $L_1 = z_1 P \bmod p = (49,10)$, $J_1 = u_1 A_1' \bmod p = (54,20)$ и хеш-функцию $h_1 = H(A_1'' = (6,40), P = (19,9), L_1 + J_1 = (19,50)) \bmod 17 = 16$;

– вычисляет:
$\theta_1 = z_1 + r_1' h_1 \bmod q = 2 + 1 * 16 \bmod 17 = 1;$
$\alpha_1 = u_1 + t_1 h_1 \bmod q = 1 + 4 * 16 \bmod 17 = 14;$
$T_1 = \theta_1 P + \alpha_1 A_1' \bmod p = 1(19,9) + 14(11,4) \bmod 59 = (6,40);$

– пересылает в БЧ ($T_1 = (6,40), P = (19,9), L_1 + J_1 = (19,50)$).

БЧ вычисляет хеш-функцию:

$$h' = H(A_1' = (6,40), P = (19,9), L_1 + J_1 = (19,50)) \bmod 17 = 16$$

и проверяет сравнение $L_1 + J_1 + h' \cdot A_1'' =^? T_1$; $(6,40) = T_1$. Сравнение выполняется, т. е. (21) для первой части криптограммы C_1 доказано.

Аналогично проверяем доказательство (21) для A_2, A_3, A_4.

Проверим доказательство *ZP* (22).

Избиратель генерирует случайное число $w = 3$. После чего вычисляет:

– $T = wP \bmod p = (54,39)$;

– $r_\Sigma = \sum_{i=1}^{k} r_i t_i + r_i' = 3, U = r_\Sigma P \bmod p = (54,39)$;

– хеш-функцию $h = H(P = (19,9), T = (54,39), U = (54,39), A_1'' = (6,40), A_2'' = (11,55), A_3'' = (19,9), A_4'' = (19,9)) \bmod 17 = 10$;

– $z = w - r_\Sigma \cdot h \bmod q = 6$.

Далее посылает в БЧ ($T = (54,39), z = 6$).

БЧ вычисляет:
– $U' = \sum_{i=1}^{k} A_i'' - \sum_{i=1}^{k} s_i A_i = (51, 57)$,
– хеш-функцию $h' = 10$;
– $T' = zP + h'U' = (6, 40)$.
– проверяет $T =^{?} T'$; $(54, 39) \neq (6, 40)$ – не выполнено; следовательно, сравнение (22) для первой части криптограммы C_1'' не доказано.

4. Сравнительный анализ сложности реализации методов доказательства корректности заполнения бюллетеня избирателем

Проведем анализ сложности реализации рассмотренных выше методов проверки корректности заполнения бюллетеня избирателем в системе ДЭГ, основанных на сравнении дискретных логарифмов и на проверке корректности перестановки. Будем полагать, что в обоих случаях для шифрования используются криптосистемы Эль-Гамаля на эллиптической кривой с одинаковыми параметрами (уравнение кривой, длины ключей, длины криптограмм, длины случайных чисел). Результаты сравнения представлены в таблице 8. Сложность вычислений будем оценивать количеством выполнений наиболее сложной операции умножения точки на целое число (буква М). Оценку сложности проведем отдельно для избирателя (доказывающей стороны) и БЧ (проверявшей стороны).

Таблица 8 показывает, что сложность формирования доказательства корректности заполнения бюллетеня для k кандидатов составляет $10k$М + 1М операций умножения для первого метода и $16k$М + 6М – для второго; это примерно на 60 % меньше для первого метода. Наоборот, объем вычислений для проверки доказательства корректности заполнения бюллетеня на одного избирателя, проводимых в БЧ, составляет $5k$М + 3М операций умножения для первого метода и $3k$М + 2М для второго. Т. е. на проверку бюллетеня во втором методе требуется в 1,67 раза меньше вычислений, чем в первом. Этот выигрыш существенно возрастает с увеличением количества избирателей, что показано в таблице 9.

Можно сделать вывод, что при большом количестве избирателей предпочтительным методом проверки является перемешивание зашифрованных голосов у избирателя, так как в этом случае значительно уменьшается нагрузка на БЧ, связанная с проверкой корректности заполнения бюллетеней.

ТАБЛИЦА 8. Сравнение методов доказательства корректности заполнения бюллетеня избирателем

TABLE 8. Comparison of Methods for Confirming the Correctness of Filling out the Voter's Ballot

	Метод на основе	
	сравнения дискретных логарифмов	проверки корректности перестановки
1) Количество операций, выполняемых на стороне избирателями. Шифрование бюллетеня	$3k$М	–
2) Количество операций формирования доказательства избирателем	$7k$М + 5М	$16k$М + 6М
Всего на стороне избирателя	$10k$М + 1М	$16k$М + 6М
Формирование зашифрованных криптограмм		$4k$М (один раз для всех избирателей)
3) Общее количество операций для проверки доказательства в БЧ	$5k$М + 3М	$3k$М + 2М
Всего на стороне БЧ для одного избирателя	$5k$М + 3М	$3k$М + 2М
Всего на стороне БЧ для n избирателей	n ($5k$М + 3М)	n ($3k$М + 2М) + $4k$М

ТАБЛИЦА 9. Оценка сложности вычислений в БЧ для первого и второго методов для разного количества избирателей

TABLE 9. Evaluation the Complexity of Calculations in the BC for the First and Second Methods for Different Numbers of Voters

	n = 1	n = 10	n = 100	n = 1000	n = 10000	n = 100000
k = 3						
Метод 1	28	280	2800	28000	2850000	2800000
Метод 2	23	122	1112	11012	110012	1100012
k = 4						
Метод 1	18	180	1800	18000	180000	1800000
Метод 2	30	156	1416	14016	140016	1400016
k = 5						
Метод 1	23	230	2300	235000	230000	2300000
Метод 2	37	190	1720	17020	170020	1700020
k = 10						
Метод 1	53	530	5300	53000	530000	5300000
Метод 2	72	360	3240	32040	320040	3200040

Заключение

В работе рассмотрена система ДЭГ, построенная на основе гомоморфной криптосистемы Эль-Гамаля на эллиптической кривой. Рассмотрены методы защиты системы ДЭГ от угрозы со стороны избирателя, заключающейся в неправильном заполнении бюллетеня. Нетривиальность решения этой задачи состоит в том, что нужно определить корректность заполнения бюллетеня избирателем, представленного в зашифрованном виде, т. е. без ознакомления с решением, которое сделал избиратель, выбирая кандидатов.

Исследованы два метода проверки корректности заполнения бюллетеня, основанные на применении доказательств с нулевым разглашением секрета. Приведено детальное описание обоих методов, подкрепленное примерами правильного и неправильного заполнения бюллетеня. Оценена сложность реализации методов по количеству операций умножения точки эллиптической кривой на целое число. Сравнительный анализ показал, что первый метод требует меньшего объема вычислений у избирателя. Для второго метода, наоборот, количество операций умножения, проводимых при проверке доказательства, примерно в 1,67 раза меньше, чем для первого.

Можно дать такие рекомендации по применению этих методов. Во-первых, следует принять во внимание, что сложность доказательства корректности заполнения бюллетеня требует во много раз большего количества операций по сравнению с операциями шифрования и расшифрования голоса избирателя, которые принципиально необходимы для обеспечения тайны голосования. Во-вторых, при выборе метода необходимо учитывать масштабность системы ДЭГ. При малом количестве избирателей исследуемые методы примерно равноценны по сложности вычислений. При большом количестве избирателей (более 10000) второй метод предпочтительней.

Также следует учесть, что второй метод имеет преимущество в том, что избиратель сам не шифрует свой бюллетень, а устанавливает зашифрованную метку в бланке бюллетеня на позицию выбираемого кандидата, затем маскирует голоса и перемешивает бюллетень.

Список источников

1. Furukawa J., Mori K., Sako K. An implementation of a Mix-Net Based Network Voting Scheme and Its Use in a Private Organization // In Chaum D., Jakobsson M., Rivest R.L., Ryan P.Y.A., Benaloh J., Kutylowski M., Adida B. ed. Towards Trustworthy Elections. Lecture Notes in Computer Science. Vol. 6000. Berlin, Heidelberg: Springer, 2010. PP. 141–154. DOI:10.1007/978-3-642-12980-3_8

2. Peng K. An efficient shuffling based eVoting scheme // Journal of Systems and Software. 2011. Vol. 84. № 6. PP. 906–922. DOI:10.1016/j.jss.2011.01.001

3. Peng K., Dawson E., Bao F. Modification and optimisation of a shuffling scheme: Stronger security, formal analysis and higher efficiency // International Journal of Information Security. 2011. Vol. 10. PP. 33–47. DOI:10.1007/s10207-010-0117-y

4. Adida B. Helios: Web-based Open-Audit Voting // Proceedings of the 17th USENIX Security Symposium (San Jose, USA, 28 July–1 August 2008). 2008. PP. 335–348.

5. Fujioka A., Okamoto T., Ohta K. A practical secret voting scheme for large scale elections // Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques «Advances in Cryptology – AUSCRYPT '92» (Gold Coast, Australia, 13–16 December 1992). Lecture Notes in Computer Science. Vol. 718. Berlin, Heidelberg: Springer, 1993. PP. 245–251. DOI:10.1007/3-540-57220-1_66

6. Ibrahim S., Kamat M., Salleh M., Aziz S.R.A. Secure E-voting with blind signature // Proceedings of the 4th National Conference of Telecommunication Technology, NCTT 2003, Shah Alam, Malaysia, 14–15 January 2003). IEEE, 2003. PP. 193–197. DOI:10.1109/NCTT.2003.1188334

7. Mateu V., Sebé F., Valls M. Constructing credential-based E-voting systems from offline E-coin protocols // Journal of Network and Computer Applications. 2014. Vol. 42. PP. 39–44. DOI:10.1016/j.jnca.2014.03.009

8. Killer C., Rodrigues B., Scheid E.J., Franco M., Eck M., Zaugg N., et al. Provotum: A Blockchain-based and End-to-end Verifiable Remote Electronic Voting System // Proceedings of the 45th Conference on Local Computer Networks (LCN, Sydney, Australia, 16–19 November 2020). IEEE, 2020. PP. 172–183. DOI:10.1109/LCN48667.2020.9314815

9. Aziz A.A., Qunoo H.N., Samra A.A. Using Homomorphic Cryptographic Solutions on E-voting Systems // International Journal of Computer Network and Information Security. 2018. Vol. 12. Iss. 1. PP. 44–59. DOI:10.5815/ijcnis.2018.01.06

10. Yang X., Yi X., Nepal S., Kelarev A., Han F. A secure Verifiable Ranked Choice Online Voting System Based on Homomorphic Encryption // IEEE Access. 2018. Vol. 6. PP. 20506–20519. DOI:10.1109/ACCESS.2018.2817518

11. ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. Springer – Verlag, 1998. URL: https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/elgamal.pdf (дата обращения 10.04.2023)

12. Alonso L.P., GASCÓ M., del BLANCO D.Y.M., Alonso J.Á.H., Barrat J., Moreton H.A. E-Voting System Evaluation Based on the Council of Europe Recommendations: Helios Voting // IEEE Transactions on Emerging Topics in Computing. 2021. Vol. 9. Iss. 1. PP. 161–173. DOI:10.1109/TETC.2018.2881891

13. Balzarotti D., Banks G., Cova M., Felmetsger V., Kemmerer R., Robertson W., et al. An experience in Testing the Security of Real-World Electronic Voting Systems // IEEE Transactions on Software Engineering. 2010. Vol. 36. Iss. 4. PP. 453–473. DOI:10.1109/TSE.2009.53

14. Estehghari S., Desmedt Y. Exploiting the Client Vulnerabilities in Internet E-voting Systems: Hacking Helios 2.0 as an Example // Proceedings of the Electronic Voting Technology Workshop / Workshop on Trustworthy Elections (Washington,

USA, 9–10 August 2010). 2010.

15. Butterfield K., Zou X. Analysis and Implementation of Internet Based Remote Voting // Proceedings of the 11th International Conference on Mobile Ad Hoc and Sensor Systems (Philadelphia, USA, 28–30 October 2014). IEEE, 2014. DOI:10.1109/MASS.2014.134

16. Cramer R., Franklin M., Schoenmakers B., Yung M. Multi-Authority Secret-Ballot Elections with Linear Work // Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (Saragossa, Spain, 12–16 May 1996). «Advances in Cryptology – EUROCRYPT '96». Lecture Notes in Computer Science. Vol. 1070. Berlin, Heidelberg: Springer, 1996. PP. 72–83. DOI:10.1007/3-540-68339-9_7

17. Seol S., Kim H., Park J.H. An Efficient Open Vote Network for Multiple Candidates // IEEE Access. 2022. Vol. 10. PP. 124291–124304. DOI:10.1109/ACCESS.2022.3224798

18. Hao F., Ryan P.Y.A., Zieliński P. Anonymous voting by two-round public discussion // IET Information Security. 2010. Vol. 4. Iss. 2. PP. 62–67. DOI:10.1049/iet-ifs.2008.0127

19. Cramer R., Gennaro R., Schoenmakers B. A Secure and Optimally Efficient Multi-Authority Election Scheme // Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques (Konstanz, Germany, 11–15 May 1997) «Advances in Cryptology – EUROCRYPT '97». Lecture Notes in Computer Science. Vol. 1233. Berlin, Heidelberg: Springer, 1997. PP. 103–118. DOI:10.1007/3-540-69053-0_9

20. Mateu V., Miret J.M., Sebé F. A hybrid approach to vector-based homomorphic tallying remote voting // International Journal of Information Security. 2016. Vol. 15. Iss. 2. PP. 211–221. DOI:10.1007/s10207-015-0279-8

21. Peng K. A general and efficient countermeasure to relation attacks in mix-based e-voting // International Journal of Information Security. 2011. Vol. 10. Iss. 1. PP. 49–60. DOI:10.1007/s10207-010-0122-1

22. Mohr A. A Survey of Zero-Knowledge Proofs with Applications to Cryptography. 2007. URL: http://austinmohr.com/Work_files/zkp.pdf (дата обращения 10.04.2023)

23. Blum M., Feldman P., Micali S. Non-interactive zero-knowledge and its applications // Proceedings of the 12-th annual ACM symposium on Theory of computing (Chicago, USA, 2–4 May 1988). ACM, 1988. PP. 103–112. DOI:10.1145/62212.62222

24. Huqing W., Zhixin S. Research on Zero-Knowledge Proof Protocol // International Journal of Computer Science Issues. 2013. Vol. 10. Iss. 1. PP. 194–200.

25. Feldman P. A practical scheme for non-interactive verifiable secret sharing // Proceedings of the 28th Annual Symposium on Foundations of Computer Science (Los Angeles, USA, 12–14 October 1987). IEEE, 1987. PP. 427–437. DOI:10.1109/SFCS.1987.4

26. Blum M., De Santis A., Micali S., Persiano G. Noninteractive Zero-Knowledge // SIAM Journal on Computing. 1991. Vol. 20. Iss. 6. DOI: 10.1137/0220068

27. Boruah D., Saikia M. Implementation of ElGamal Elliptic Curve Cryptography over prime field using C // Proceedings of the International Conference on Information Communication and Embedded Systems (ICICES2014, Chennai, India, 27–28 February 2014). IEEE, 2014. DOI:10.1109/ICICES.2014.7033751

28. Kapoor V., Abraham V.S., Singh R. Elliptic curve cryptography // Ubiquity. 2008. Vol. 9. Iss. 20. DOI:10.1145/1378355.1378356

29. Коржик В.И., Яковлев В.А. Основы криптографии. СПб.: ИЦ Интермедия, 2016. 296 с.

References

1. Furukawa J., Mori K., Sako K. An implementation of a Mix-Net Based Network Voting Scheme and Its Use in a Private Organization. *In Chaum D., Jakobsson M., Rivest R.L., Ryan P.Y.A., Benaloh J., Kutylowski M., Adida B. ed. Towards Trustworthy Elections. Lecture Notes in Computer Science. Vol. 6000*. Berlin, Heidelberg: Springer; 2010. p.141–154. DOI:10.1007/978-3-642-12980-3_8

2. Peng K. An efficient shuffling based eVoting scheme. *Journal of Systems and Software*. 2011;84(6):906–922. DOI:10.1016/j.jss.2011.01.001

3. Peng K., Dawson E., Bao F. Modification and optimisation of a shuffling scheme: Stronger security, formal analysis and higher efficiency. *International Journal of Information Security*. 2011;10:33–47. DOI:10.1007/s10207-010-0117-y

4. Adida B. Helios: Web-based Open-Audit Voting. *Proceedings of the 17th USENIX Security Symposium, 28 July–1 August 2008, San Jose, USA*. 2008. p.335–348.

5. Fujioka A., Okamoto T., Ohta K. A practical secret voting scheme for large scale elections. *Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques «Advances in Cryptology – AUSCRYPT '92», 13–16 December 1992, Gold Coast, Australia. Lecture Notes in Computer Science, vol.718*. Berlin, Heidelberg: Springer; 1993. p.245–251. DOI:10.1007/3-540-57220-1_66

6. Ibrahim S., Kamat M., Salleh M., Aziz S.R.A. Secure E-voting with blind signature. *Proceedings of the 4th National Conference of Telecommunication Technology, NCTT 2003, 14–15 January 2003, Shah Alam, Malaysia*. IEEE; 2003. p.193–197. DOI:10.1109/NCTT.2003.1188334

7. Mateu V., Sebé F., Valls M. Constructing credential-based E-voting systems from offline E-coin protocols. *Journal of Network and Computer Applications*. 2014;42:39–44. DOI:10.1016/j.jnca.2014.03.009

8. Killer C., Rodrigues B., Scheid E.J., Franco M., Eck M., Zaugg N., et al. Provotum: A Blockchain-based and End-to-end Verifiable Remote Electronic Voting System. *Proceedings of the 45th Conference on Local Computer Networks, LCN, 16–19 November 2020, Sydney, Australia*. IEEE; 2020. p.172–183. DOI:10.1109/LCN48667.2020.9314815

9. Aziz A.A., Qunoo H.N., Samra A.A. Using Homomorphic Cryptographic Solutions on E-voting Systems. *International Journal of Computer Network and Information Security*. 2018;12(1):44–59. DOI:10.5815/ijcnis.2018.01.06

10. Yang X., Yi X., Nepal S., Kelarev A., Han F. A secure Verifiable Ranked Choice Online Voting System Based on Homomorphic Encryption. *IEEE Access*. 2018;6:20506–20519. DOI:10.1109/ACCESS.2018.2817518

11. ElGamal T. *A public key cryptosystem and a signature scheme based on discrete logarithms*. Springer – Verlag; 1998. URL: https://people.csail.mit.edu/alinush/6.857-spring-2015/papers/elgamal.pdf [Accessed 10th April 2023]
12. Alonso L.P., GASCÓ M., del BLANCO D.Y.M., Alonso J.Á.H., Barrat J., Moreton H.A. E-Voting System Evaluation Based on the Council of Europe Recommendations: Helios Voting. *IEEE Transactions on Emerging Topics in Computing*. 2021;9(1):161–173. DOI:10.1109/TETC.2018.2881891
13. Balzarotti D., Banks G., Cova M., Felmetsger V., Kemmerer R., Robertson W., et al. An experience in Testing the Security of Real-World Electronic Voting Systems. *IEEE Transactions on Software Engineering*. 2010;36(4):453–473. DOI:10.1109/TSE.2009.53
14. Estehghari S., Desmedt Y. Exploiting the Client Vulnerabilities in Internet E-voting Systems: Hacking Helios 2.0 as an Example. *Proceedings of the Electronic Voting Technology Workshop / Workshop on Trustworthy Elections, 9–10 August 2010, Washington, USA*. 2010.
15. Butterfield K., Zou X. Analysis and Implementation of Internet Based Remote Voting. *Proceedings of the 11th International Conference on Mobile Ad Hoc and Sensor Systems, 28–30 October 2014, Philadelphia, USA*. IEEE; 2014. DOI:10.1109/MASS.2014.134
16. Cramer R., Franklin M., Schoenmakers B., Yung M. Multi-Authority Secret-Ballot Elections with Linear Work. *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, 12–16 May 1996, Saragossa, Spain. «Advances in Cryptology – EUROCRYPT '96». Lecture Notes in Computer Science, vol.1070*. Berlin, Heidelberg: Springer; 1996. p.72–83. DOI:10.1007/3-540-68339-9_7
17. Seol S., Kim H., Park J.H. An Efficient Open Vote Network for Multiple Candidates. *IEEE Access*. 2022;10:124291–124304. DOI:10.1109/ACCESS.2022.3224798
18. Hao F., Ryan P.Y.A., Zieliński P. Anonymous voting by two-round public discussion. *IET Information Security*. 2010;4(2):62–67. DOI:10.1049/iet-ifs.2008.0127
19. Cramer R., Gennaro R., Schoenmakers B. A Secure and Optimally Efficient Multi-Authority Election Scheme. *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques, 11–15 May 1997, Konstanz, Germany. «Advances in Cryptology – EUROCRYPT '97». Lecture Notes in Computer Science, vol.1233*. Berlin, Heidelberg: Springer, 1997. p.103–118. DOI:10.1007/3-540-69053-0_9
20. Mateu V., Miret J.M., Sebé F. A hybrid approach to vector-based homomorphic tallying remote voting. *International Journal of Information Security*. 2016;15(20):211–221. DOI:10.1007/s10207-015-0279-8
21. Peng K. A general and efficient countermeasure to relation attacks in mix-based e-voting. *International Journal of Information Security*. 2011;10(1):49–60. DOI:10.1007/s10207-010-0122-1
22. Mohr A. *A Survey of Zero-Knowledge Proofs with Applications to Cryptography*. 2007. URL: http://austinmohr.com/Work_files/zkp.pdf [Accessed 10th April 2023]
23. Blum M., Feldman P., Micali S. Non-interactive zero-knowledge and its applications. *Proceedings of the 12-th annual ACM symposium on Theory of computing, 2–4 May 1988, Chicago, USA*. ACM; 1988. p.103–112. DOI:10.1145/62212.62222
24. Huqing W., Zhixin S. Research on Zero-Knowledge Proof Protocol. *International Journal of Computer Science Issues*. 2013;10(1):194–200.
25. Feldman P. A practical scheme for non-interactive verifiable secret sharing. *Proceedings of the 28th Annual Symposium on Foundations of Computer Science, 12–14 October 1987, Los Angeles, USA*. IEEE; 1987. p.427–437. DOI:10.1109/SFCS.1987.4
26. Blum M., De Santis A., Micali S., Persiano G. Noninteractive Zero-Knowledge. *SIAM Journal on Computing*. 1991;20(6). DOI: 10.1137/0220068
27. Boruah D., Saikia M. Implementation of ElGamal Elliptic Curve Cryptography over prime field using C. *Proceedings of the International Conference on Information Communication and Embedded Systems, ICICES2014, 27–28 February 2014, Chennai, India)*. IEEE; 2014. DOI:10.1109/ICICES.2014.7033751
28. Kapoor V., Abraham V.S., Singh R. Elliptic curve cryptography. *Ubiquity*. 2008;9(20). DOI:10.1145/1378355.378356
29. Korzhik V.I., Yakovlev V.A. *Fundamentals of Cryptography*. Saint Petersburg: IC Intermedia Publ.; 2016. 296 p. (in Russ.)

Статья поступила в редакцию 16.03.2023; одобрена после рецензирования 21.03.2023; принята к публикации 24.03.2023.
The article was submitted 16.03.2023; approved after reviewing 21.03.2023; accepted for publication 24.03.2023.

Информация об авторах:

ЯКОВЛЕВ Виктор Алексеевич — доктор технических наук, профессор, профессор кафедры защищенных систем связи Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0009-0007-2861-9605

САЛМАН Васан Давуд — аспирант кафедры защищенных систем связи Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича
https://orcid.org/0000-0003-4454-7844