ПРОГРАММНО-ТЕХНИЧЕСКИЙ КОМПЛЕКС, ОБЕСПЕЧИВАЮЩИЙ ДИСТАНЦИОННОЕ ЭЛЕКТРОННОЕ ГОЛОСОВАНИЕ ИЗБИРАТЕЛЕЙ (УЧАСТНИКОВ РЕФЕРЕНДУМА) ВНЕ ЗАВИСИМОСТИ ОТ МЕСТА ИХ НАХОЖДЕНИЯ

Описание протокола ДЭГ к выборам, голосование на которых состоится 17, 18 и 19 сентября 2021 г.

2021

Оглавление

1.1 Протокол

П р и м е ч а н и е:
В рамках рассмотрения схем подписи будем считать термины «ключ подписи» и «ключ ЭП» («ключ проверки подписи» и «ключ проверки ЭП» соответственно) равносильными. В рамках рассмотрения схем шифрования и доказательств будем считать термины «ключ расшифрования» и «закрытый ключ» («ключ шифрования» и «открытый ключ» соответственно) равносильными.

1.1.1 Участники протокола

В настоящем разделе приведены участники протокола ДЭГ, исполняющие отдельные роли. Под участниками понимаются компоненты системы или пользователи, выполняющие отдельные операции.

Пользователи:
Участник ДЭГ – гражданин РФ, обладающий активным избирательным правом и включенный в списки участников ДЭГ на основании поданного в электронной форме заявления для участия в ДЭГ.
Организатор (Комиссия ДЭГ) – участник, осуществляющий организацию процесса дистанционного электронного голосования с использованием компонента «Организация и проведение ДЭГ». Генерирует ключевую пару (ключи шифрования и расшифрования бюллетеней), разделяет ключ шифрования бюллетеней (ключ Комиссии).
Наблюдатель – участник, осуществляющий наблюдение за процессом голосования и аудит результатов голосования.

Компоненты системы:
Регистратор (компонент «Список участников ДЭГ» и компонент «Портал ДЭГ»)– участник, осуществляющий идентификацию и аутентификацию пользователей портала ДЭГ с помощью ЕСИА и предоставляющий Участникам ДЭГ право получения бюллетеня путем подписи вслепую.
Избирательный ящик (компонент «Сервис анонимного волеизъявления») – участник, выдающий Участникам ДЭГ бюллетени и принимающий обратно заполненные зашифрованные бюллетени.
Учетчик (компонент «Распределенное хранение данных и подсчет голосов») – участник, представляющий собой хранилище транзакций (бюллетеней Участников ДЭГ и других данных) и осуществляющий подсчет итогов, к которому есть постоянный доступ на запись и/или чтение у участников протокола, а также у избирательных комиссий, организующих выборы (определяющих результаты выборов на территории), которые проводят подготовку исходных данных в ГАС «Выборы» (текст бюллетеня, форма протокола) для передачи Организатору через «воздушный зазор».

1.1.2 Этапы ДЭГ

Ниже представлено описание этапов ДЭГ. Считаем, что к моменту начала ДЭГ завершены следующие процедуры:

- формирование исходных данных голосования (описатель протокола, тексты бюллетеней, количество вариантов, идентификаторы организующих и определяющих результаты выборов на территории избирательных комиссий) и данных для составления списка участников ДЭГ. Соответствующие данные переданы Организатору;
- генерация и регистрация ключей ЭП Организатора и Регистратора, публикующих данные в Учетчик (далее будем считать, что каждое сообщение от них подписывается, подпись проверяется Учетчиком).

1.1.2.1 Инициализация системы

1. Генерация ключей:

- Регистратор генерирует ключевую пару подписи вслепую $(sk_{blind},\ pk_{blind})$ и передает ключ проверки подписи pk_{blind} Организатору.
- Регистратор генерирует ключ K_{com} для схемы commitment.

- Организатор (Комиссия ДЭГ) в присутствии Наблюдателя и представителей средств массовой информации генерируют ключевую пару шифрования ($s_{\text{комиссия}}$, $Q_{\text{комиссия}}$) на выделенном устройстве. Закрытый ключ $s_{\text{комиссия}}$ разделяется по схеме Шамира, доли записываются на защищенные носители, доступ к которым имеют лица, определенные решением комиссии ДЭГ. Открытый ключ $Q_{\text{комиссия}}$ (ключ Комиссии) отображается на экране устройства для всех присутствующих. После этого закрытый ключ уничтожается на устройстве.

- Учетчик генерирует вторую ключевую пару шифрования ($s_{\text{учетчик}}$, $Q_{\text{учетчик}}$) и передает открытый ключ $Q_{\text{учетчик}}$ (ключ комиссии, организующей выборы или определяющей результаты выборов на территории) Организатору.

2. Загрузка данных голосования

1) Организатор для каждого голосования загружает в Учетчик следующие данные:
- идентификатор голосования ($VotingID$);
- время начала приема бюллетеней;
- хэш-значение от текста бюллетеня;
- размерность бюллетеня (общее количество вариантов);
- максимальное количество выбранных вариантов (от 1 до N);
- ключ проверки подписи вслепую.

2) Организатор передает Регистратору список участников ДЭГ. Регистратор в свою очередь формирует commitment-значения для идентификаторов участников ДЭГ (СНИЛС):

$$commitment = HMAC(K_{com}, \text{СНИЛС} \mid VotingID)$$

и публикует их в Учетчике.

3) Организатор передает Избирательному ящику текст бюллетеня и идентификатор голосования $VotingID$.

3. Загрузка ключа шифрования бюллетеней

Организатор публикует в Учетчике ключ шифрования бюллетеней в составе: открытый ключ Комиссии $Q_{\text{комиссия}}$, открытый ключ Учетчика $Q_{\text{учетчик}}$, итоговый ключ шифрования бюллетеней $Q_{\text{итог}}$. Итоговый ключ шифрования бюллетеней формируется следующим образом:

$$Q_{\text{итог}} = H\left(Q_{\text{учетчик}} \mid Q_{\text{комиссия}}\right) \cdot Q_{\text{комиссия}} + H(Q_{\text{комиссия}} \mid Q_{\text{учетчик}}) \cdot Q_{\text{учетчик}},$$

где H – хэш-функция «Стрибог», определенная в ГОСТ Р 34.11-2012 [8].
Комиссия сохраняет у себя открытый ключ Комиссии и итоговый открытый ключ.

4. Получение ключа шифрования

Регистратор получает итоговый открытый ключ $Q_{\text{итог}}$ из Учетчика.

1.1.2.2 Предоставление права получения бюллетеня

5. *Идентификация и аутентификация участника ДЭГ*

Пользователь портала ДЭГ в дни голосования обращается к Регистратору. Регистратор выполняет идентификацию и аутентификацию пользователя с помощью ЕСИА, в результате чего получает от нее подписанный токен идентификации (id_token ЕСИА). Далее проверяет наличие пользователя в списке участников ДЭГ и то, что он еще не голосовал.

6. Получение подписи вслепую и ключа шифрования бюллетеней

Участник ДЭГ генерирует ключевую пару ЭП. Участник ДЭГ и Регистратор выполняют протокол формирования подписи вслепую для ключа проверки ЭП Участника ДЭГ. Регистратор также передает Участнику ДЭГ идентификатор голосования $VotingID$ и ключ шифрования бюллетеней $Q_{\text{итог}}$.

7. Фиксация факта выдачи подписи вслепую

Факт выдачи подписи фиксируется Регистратором в списке участников ДЭГ и сопровождается записью персональных данных и id_token ЕСИА в данный список. Регистратор публикует в Учетчике соответствующее commitment-значение идентификатора Участника ДЭГ и значение маскированной подписи вслепую.

1.1.2.3 Подача заполненного бюллетеня

8. Волеизъявление

Участник ДЭГ анонимно обращается к Избирательному ящику и запрашивает текст бюллетеня по $VotingID$. Применяется механизм передачи ключевой пары ЭП Участника ДЭГ из компонента «Портал ДЭГ» в компонент «Сервис анонимного волеизъявления». Подробное описание процесса передачи ключевой пары ЭП Участника ДЭГ в браузере представлено в разделе 1.1.2.4.

1.1.2.4 Передача ключей в анонимную зону

Поскольку при переходе в анонимную зону происходит переадресация Участника ДЭГ с домена Регистратора на домен Избирательного ящика, ключи в локальном хранилище браузера, которые были сгенерированы при взаимодействии с Регистратором, становятся недоступны, так как политика безопасности браузера не допускает обращение к локальному хранилищу ключей со стороны скрипта с другого домена. Поэтому при переходе используется следующий механизм:

1) Перед переходом на домен Избирательного Ящика, когда Участник ДЭГ еще находится на домене Регистратора, ключевая пара ЭП Участника ДЭГ, подпись Регистратора, полученная вслепую, и идентификатор голосования зашифровываются с помощью схемы аутентифицированного шифрования на случайно сгенерированном ключе длины 512 бит в браузере клиента.

2) Выполняется переход по POST, в теле запроса передается пароль и ключ шифрования бюллетеней. При переходе формируется хэш-ссылка (анкор), в которой после символа # следует шифртекст. При переходе по такой ссылке данные после символа # не передаются на сервер, а остаются только в браузере (то есть при переходе на yyy.zz/ballot-box#1234556789 браузер переходит на yyy.zz/ballot-box, на сервере запрос выглядит как yyy.zz/ballot-box, а 123456789 остается в браузере пользователя).

3) Сервер в ответ возвращает в браузер пароль и ключ шифрования бюллетеней, которым были зашифрованы данные, которые передались в анкоре.

4) Браузер расшифровывает данные.

1. Избирательный ящик направляет Участнику ДЭГ текст бюллетеня.
2. Участник ДЭГ заполняет бюллетень, зашифровывает его с помощью ключа шифрования бюллетеней $Q_{\text{итог}}$ и формирует доказательство корректности содержимого бюллетеня (ZKP). Подробное описание процесса заполнения бюллетеня представлено в разделе 1.1.2.5.

1.1.2.5 Заполнение бюллетеня

Заполненный бюллетень в электронном виде представляется как строка нулей и единиц. Количество символов соответствует количеству вариантов выбора, при этом выбранные варианты представлены единицей 1, остальные варианты выбора представлены нулями 0. Производится шифрование каждого варианта (сообщение $m = 0$ или $m = 1$) с помощью схемы Эль-Гамаля, в результате чего формируются для каждого варианта значения (R, C). Далее формируется доказательство корректности содержимого бюллетеня, состоящее из двух частей. Для каждого шифртекста (R, C) формируется доказательство, что выбор m соответствует либо 0, либо 1. Также формируется доказательство для всего бюллетеня: складываются шифртексты (за счет свойства гомоморфности схемы Эль-Гамаля) для всех вариантов, и для суммарного шифртекста $(\sum R, \sum C)$ формируется доказательство, что сумма $\sum m$ не превышает количества единиц, определенного порядком заполнения бюллетеня.

3. Участник ДЭГ формирует транзакцию, состоящую из зашифрованного бюллетеня, доказательства корректности содержимого бюллетеня, ключа проверки ЭП Участника ДЭГ, значения подписи Регистратора ключа проверки ЭП Участника ДЭГ (полученной вслепую). Все содержимое транзакции подписывается ключом ЭП Участника ДЭГ с помощью алгоритма ГОСТ Р 34.10-2012 [6].

4. Участник ДЭГ направляет транзакцию Избирательному ящику.

5. Избирательный ящик проверяет:
   а) корректность формата бюллетеня (наличие всех необходимых полей, их длина, и т.д.);
   б) корректность подписи Регистратора;
   в) уникальность транзакции с указанным ключом проверки ЭП Участника ДЭГ (по своей внутренней базе данных).

6. Если какая-то из проверок не пройдена, Избирательный ящик возвращает Участнику ДЭГ ошибку. При прохождении всех проверок Избирательный ящик добавляет ключ проверки ЭП Участника ДЭГ в свою внутреннюю базу данных и направляет транзакцию в Учетчик.

9. Публикация бюллетеня

В случае успешных проверок Избирательный ящик направляет транзакцию Участника ДЭГ в Учетчик. Учетчик осуществляет следующие проверки:

- корректность формата бюллетеня (наличие всех необходимых полей, их длина, и т.д.);
- корректность подписи Регистратора;
- корректность ЭП Участника ДЭГ под транзакцией.

В случае успешных проверок транзакция добавляется в Учетчик.

10. Проверка бюллетеня

Для контроля наличия бюллетеня в хранилище Учетчика через некоторое время после подачи бюллетеня Участник ДЭГ с помощью портала наблюдения проверяет, что его транзакция находится в хранилище по своему ключу проверки подписи.

1.1.2.6 Подведение итогов

11. Прекращение выдачи бюллетеней

Организатор посылает Регистратору команду о прекращении выдачи подписей вслепую Участникам ДЭГ. После этого Регистратор прекращает выдачу подписей вслепую.

12. Прекращение приема бюллетеней

Не менее чем через 15 минут Организатор публикует транзакцию о завершении приема бюллетеней в Учетчике. После этого Учетчик не принимает бюллетени.

13. Церемония восстановления закрытого ключа Комиссии

Комиссия проводит церемонию сборки своего закрытого ключа $s_{\text{комиссия}}$.

14. Подсчет итогов.

Учетчик проверяет все доказательства корректности содержимого хранящихся бюллетеней. Для успешно проверенных бюллетеней зашифрованные голоса по каждому варианту складываются, формируя итоговый шифртекст (R, C) по каждому варианту.

15. Расшифрование итогов

1. Для итогового шифртекста (R, C) по каждому варианту Учетчик формирует результат частичного расшифрования как $R' = s_{\text{учетчик}} \cdot R$, где $s_{\text{учетчик}}$ – закрытый ключ Учетчика, и доказательство корректности расшифрования относительно своего открытого ключа $Q_{\text{учетчик}} = s_{\text{учетчик}} \cdot P$ (доказательство того, что R' вычислялся с использованием корректного закрытого ключа $s_{\text{учетчик}}$). Учетчик публикует транзакцию с результатами частичного расшифрования R' и доказательствами ZKP.

2. Организатор загружает закрытый ключ Комиссии $s_{\text{комиссия}}$ в Учетчик. Учетчик проверяет, что загруженный закрытый ключ соответствует ранее опубликованному открытому ключу $Q_{\text{комиссия}}$.

3. Далее Учетчик окончательно расшифровывает шифртексты по каждому варианту с использованием закрытого ключа Комиссии:

$$M = C - H\left(Q_{\text{учетчик}} \mid Q_{\text{комиссия}}\right) \cdot s_{\text{комиссия}} \cdot R - H(Q_{\text{комиссия}} \mid Q_{\text{учетчик}}) \cdot R'.$$

После этого Учетчик перебором по $M = m \cdot P$ восстанавливает m – количество голосов по каждому варианту и публикует итоговые шифртексты (R, C) и расшифрованные результаты m.

1.1.2.7 Аудит

16. Загрузка данных для аудита

После завершения голосования Наблюдатель получает:

1. От Учетчика: все содержимое хранилища;
2. От Регистратора: список участников ДЭГ, которым была выдана подпись вслепую, вместе с id_token от ЕСИА и ключ K_{com};
3. От Оператора ЕСИА: список соответствия СНИЛС Участников ДЭГ и их идентификаторов в ЕСИА.

17. Проверка корректности данных

Наблюдатель проверяет:

- для каждого Участника ДЭГ, которому выдали подпись вслепую (согласно списку, полученному от Регистратора), есть валидный id_token от ЕСИА (проверяет подпись ЕСИА, соответствие идентификаторов ЕСИА, указанных в токене, персональным данным (СНИЛС) из списка);
- для каждого Участника ДЭГ, которому выдали подпись вслепую (согласно списку, полученному от Регистратора), есть отметка в Учетчике в виде commitment-значения и наоборот (проверка происходит, в том числе, с помощью ключа K_{com});
- количество бюллетеней, которые оказались в Учетчике, не превышает количество Участников ДЭГ, которым выдали подпись вслепую;
- корректность подписей Регистратора, выданных вслепую, на всех бюллетенях;
- корректность ЭП Участника ДЭГ на всех бюллетенях;
- загрузился закрытый ключ Комиссии, соответствующий открытому;
- уникальность транзакции с указанным ключом проверки ЭП (нет двух транзакций с одинаковым ключом);
- доказательства корректности содержимого бюллетеней.

Далее Наблюдатель полностью повторяет действия Учетчика по подсчету: суммирует все валидные бюллетени (для которых доказательства из предыдущего пункта верные), проверяет доказательства корректности расшифрования и сравнивает с итогами.

Комиссия проверяет доказательства корректности содержимого бюллетеней, используя сохраненный итоговый ключ, и проверяет равенство количества недействительных бюллетеней в хранилище и количества недействительных бюллетеней, указанных в итогах голосования.

1.2 Базовые криптографические механизмы

Используемые обозначения:

$\|$ – конкатенация двоичных строк;

$str_s(r)$ – представление числа r в виде битовой строки длины s бит (в формате big-endian);

$int(s)$ – число, битовое представление которого в формате big-endian представляет собой строку s;

$msb_l(s)$ – наиболее значащие l битов строки s;

$LCM(a, b)$ – наименьшее общее кратное чисел a и b;

$\mathbb{Z}_m$ – кольцо вычетов по модулю m;

$\mathbb{Z}_m^*$ – мультипликативная группа кольца вычетов по модулю m;

$b \bmod p$ – минимальное неотрицательное число, сравнимое с b по модулю p;

$a \leftarrow_{\$} A$ – элемент a выбирается из множества A случайно равновероятно.

1.2.1 Схема подписи

В качестве схемы подписи используется схема, определенная в ГОСТ Р 34.10-2012 [6]. В качестве базовой эллиптической кривой используется кривая id-tc26-gost-3410-2012-256-paramSetB, параметры которой определены в рекомендациях по стандартизации Р 1323565.1.024-2019 [7].

1.2.2 Протокол формирования и проверки подписи вслепую

Протокол формирования и проверки подписи вслепую задается следующими алгоритмами:

- $KeyGen(k) \to (sk, pk)$: алгоритм генерации ключей, принимающий на вход параметр безопасности k и возвращающий пару ключей (sk, pk), где sk – ключ подписи, pk – ключ проверки подписи;
- $\langle Sign(sk), User(pk, m)\rangle \to (b, \sigma)$: интерактивный протокол, выполняемый между подписывающим, который обладает ключом подписи sk, и клиентом, который обладает сообщением m; подписывающий выдает $b = 1$, если взаимодействие успешно завершилось, и $b = 0$ в противном случае; клиент выдает значение подписи σ в случае успешного завершения протокола и код ошибки в противном случае;
- $Verify(pk, m, \sigma) \to b$: детерминированный алгоритм проверки подписи, принимающий на вход ключ проверки подписи pk, сообщение m и подпись σ и возвращающий единицу, если значение подписи верное, и ноль в противном случае.

В качестве протокола формирования и проверки подписи вслепую используется схема подписи вслепую на основе RSA, предложенная в работе [1] (см. Раздел 1.2.2.1), при этом для хэширования сообщения используется конструкция FDH, определенная в Разделе 1.2.2.2.

1.2.2.1 Схема подписи вслепую на основе RSA

Параметр безопасности k выбирается равным 4096.

Алгоритм генерации ключей:

$KeyGen(k)$

$p, q \leftarrow_{\$} \mathbb{Z}_{2^{k/2}}$:
$\quad p, q - \text{prime}$
$\quad \sqrt{2} \cdot 2^{k/2-1} \leqslant p \leqslant 2^{k/2} - 1$
$\quad \sqrt{2} \cdot 2^{k/2-1} \leqslant q \leqslant 2^{k/2} - 1$
$N \leftarrow pq$
$e \leftarrow 2^{16} + 1$
$d \leftarrow e^{-1} \mod \mathsf{LCM}(p-1, q-1)$
return $(d, (e, N))$

Интерактивный протокол формирования подписи:

$\mathbf{Sign}(d)$		$\mathbf{User}((e, N), m)$
		$h \leftarrow \mathsf{FDH}(m, N)$
		$r \leftarrow_{\$} \mathbb{Z}_N^*$
		$h' \leftarrow r^e h$
	$\xleftarrow{h'}$	
$s \leftarrow (h')^d$		
	$\xrightarrow{s}$	
return 1		$\sigma \leftarrow r^{-1} s$

Алгоритм проверки подписи:

$Verify((e,N),m,\sigma)$

$h \leftarrow \mathsf{FDH}(m,N)$
if $\sigma^e = h$: **return** 1
else : **return** 0

1.2.2.2 Описание конструкции FDH

В качестве базовой хэш-функции H для конструкции FDH используется хэш-функция «Стрибог» с длиной выхода 256 битов, определенная в ГОСТ Р 34.11-2012 [8].
Пусть$flag_1, flag_2$ – фиксированные различные строки длины 1 байт.

Конструкция FDH:

$FDH(m,N)$

$IV \leftarrow 0$
$first\_block \leftarrow \mathsf{int}(\mathsf{msb}_{256}(\mathsf{str}_{4096}(N)))$
while $\mathsf{int}(H(m\|N\|flag_1\|IV)) \geqslant first\_block$:
  $IV++$
$fdh \leftarrow H(m\|N\|flag_1\|IV) \parallel H(m\|N\|flag_2\|IV+1) \parallel \ldots \parallel H(m\|N\|flag_2\|IV+15)$
return fdh

При хэшировании значения IV и N представляются как байтовые строки длиной 1 байт и 512 байт соответственно (в формате big-endian).

1.2.3 Схема шифрования бюллетеней

В качестве схемы шифрования бюллетеней используется схема шифрования Эль-Гамаля на эллиптических кривых, основанная на работе [2]. В качестве базовой эллиптической кривой используется кривая id-tc26-gost-3410-2012-256-paramSetB, параметры которой определены в Р 1323565.1.024-2019 [7].
Схема шифрования с открытым ключом задается следующими алгоритмами:

- $KeyGen(\,) \rightarrow (sk, pk)$: алгоритм генерации ключей, возвращающий пару ключей (sk, pk), где sk - ключ расшифрования, pk - ключ шифрования;
- $Enc(pk, m) \rightarrow c$: алгоритм шифрования, принимающий на вход открытый ключ шифрования pk и сообщение m и возвращающий шифртекст c;
- $Dec(sk, c) \rightarrow m$: алгоритм расшифрования, принимающий на вход ключ расшифрования sk и шифртекст c и возвращающий сообщение m.

Зададим работу этих алгоритмов для схемы шифрования Эль-Гамаля на эллиптических кривых. Будем считать, что q – простой порядок подгруппы группы точек эллиптической кривой, P – генерационная точка порядка q.

$KeyGen(\,)$	$Enc(Q,m)$	$Dec(d,(R,C))$
$d \leftarrow_s \mathbb{Z}_q^*$	$M \leftarrow mP$	$M \leftarrow C - dR$
$Q \leftarrow dP$	$r \leftarrow_s \mathbb{Z}_q^*$	find m: $M = mP$
return (d,Q)	$R \leftarrow rP$	**return** m
	$C \leftarrow M + rQ$	
	return (R,C)	

Нахождение сообщения m по точке M осуществляется перебором возможных значений m (от нуля до общего количества участников ДЭГ, принимающих участие в голосовании).

1.2.4 Доказательство корректности содержимого бюллетеня

Используется схема disjunctive Chaum-Pedersen proof, определенная на базе схем, предложенных в работах [3], [4], и обобщенная на случай множественного выбора (значение открытого текста лежит в диапазоне $[min, max], 0 \leq min \leq max \leq n$, где n – количество вариантов).
В качестве базовой хэш-функции H используется хэш-функция «Стрибог» с длиной выхода $hlen = 256$ битов, определенная в ГОСТ Р 34.11-2012 [8].
Настоящая схема доказательства с нулевым разглашением применяется в паре со схемой шифрования с открытым ключом Эль-Гамаля и использует ту же самую кривую, а также ключи, сгенерированные для шифрования. Кроме того, для генерации доказательства используется случайное значение r, выработанное в процессе работы алгоритма шифрования. Формируемое доказательство позволяет убедить проверяющего, что значение открытого текста, соответствующего определенному шифртексту, принадлежит диапазону возможных значений.
Схема задается следующими алгоритмами:

- $genProof(pk, range, m, r, c) \rightarrow proof$: алгоритм генерации доказательства, принимающий на вход открытый ключ шифрования pk, диапазон возможных значений открытого текста $range$, открытый текст m, случайное значение r и шифртекст c и возвращающий доказательство $proof$;
- $verifyProof(pk, range, c, proof) \rightarrow b$: алгоритм проверки доказательства, принимающий на вход открытый ключ шифрования pk, диапазон возможных значений открытого текста $range$, шифртекст c и доказательство $proof$ и возвращающий “true”, если доказательство верное, и “false” в противном случае.

Зададим работу этих алгоритмов.

$genProof(Q, [min, max], m, r, (R, C))$

$sum \leftarrow 0$
$i \leftarrow min$
while $i \leqslant max$:
  if $m \neq i$:
    $c_i \leftarrow_{\$} \mathbb{Z}_q^*$
    $r_i \leftarrow_{\$} \mathbb{Z}_q^*$
    $A_i \leftarrow r_i P - c_i R$
    $B_i \leftarrow r_i Q - c_i (C - iP)$
    $sum \leftarrow sum + c_i$
  else :
    $index \leftarrow i$
    $tmp \leftarrow_{\$} \mathbb{Z}_q^*$
    $A_i \leftarrow tmp \cdot P$
    $B_i \leftarrow tmp \cdot Q$
  $i \leftarrow i + 1$
$Astr \leftarrow A_{min} \| \ldots \| A_{max}$
$Bstr \leftarrow B_{min} \| \ldots \| B_{max}$
$c \leftarrow H(Q \| R \| C \| Astr \| Bstr)$
$c_{index} \leftarrow c - sum \mod q$
$r_{index} \leftarrow tmp + c_{index} \cdot r \mod q$
$cstr \leftarrow c_{min} \| \ldots \| c_{max}$
$rstr \leftarrow r_{min} \| \ldots \| r_{max}$
return $(Astr, Bstr, cstr, rstr)$

$verifyProof(Q, [min, max], (R, C), (Astr, Bstr, cstr, rstr))$

$sum \leftarrow 0$
$i \leftarrow min$
$A_{min} \| \ldots \| A_{max} \leftarrow Astr$
$B_{min} \| \ldots \| B_{max} \leftarrow Bstr$
$c_{min} \| \ldots \| c_{max} \leftarrow cstr$
$r_{min} \| \ldots \| r_{max} \leftarrow rstr$
while $i \leqslant max$:
  if $r_i P \neq A_i + c_i R$:
    return false
  if $r_i Q \neq B_i + c_i (C - iP)$:
    return false
  $sum \leftarrow sum + c_i$
  $i \leftarrow i + 1$
$c \leftarrow H(Q \| R \| C \| Astr \| Bstr)$
if $c = sum \mod q$:
  return true
return false

1.2.5 Доказательство корректности расшифрования

В качестве схемы доказательства корректности расшифрования используется схема Chaum Pedersen, предложенная в работе [4]. Эта схема позволяет для 4-х точек Y_1, G_1, Y_2, G_2 доказать, что:

$$Y_1 = xG_1, \quad Y_2 = xG_2.$$

Схема задается следующими алгоритмами:

- $genProof(x, Y_1, G_1, Y_2, G_2) \to proof$: алгоритм генерации доказательства, принимающий на вход скаляр x и четыре точки Y_1, G_1, Y_2, G_2, доказательство для которых необходимо сформировать, и возвращающий доказательство $proof$;
- $verifyProof(proof, Y_1, G_1, Y_2, G_2) \to b$: алгоритм проверки доказательства, принимающий на вход доказательство $proof$ и четыре целевые точки и возвращающий "true", если доказательство верное, и "false" в противном случае.

Зададим эти алгоритмы. Пусть H – хэш-функция.

$genProof(x, Y_1, G_1, Y_2, G_2)$	$verifyProof(Y_1, G_1, Y_2, G_2, (w, U_1, U_2))$
$u \leftarrow_{\$} \mathbb{Z}_q^*$	$v \leftarrow H(U_1 \| U_2 \| G_1 \| Y_1 \| G_2 \| Y_2)$
$U_1 \leftarrow uG_1, U_2 \leftarrow uG_2$	**if** $(wG_1 = vY_1 + U_1) \wedge (wG_2 = vY_2 + U_2)$:
$v \leftarrow H(U_1 \| U_2 \| G_1 \| Y_1 \| G_2 \| Y_2)$	**return** true
$w \leftarrow xv + u \mod q$	**return** false
return (w, U_1, U_2)	

Настоящая схема доказательства с нулевым разглашением применяется в паре со схемой шифрования с открытым ключом Эль-Гамаля и использует ту же самую кривую. В качестве базовой хэш-функции H используется хэш-функция «Стрибог» с длиной выхода 256 битов, определенная в ГОСТ Р 34.11-2012 [8].

Формируемое доказательство позволяет убедить проверяющего, что результат R' частичного расшифрования m некоторого известного шифртекста (R, C) был действительно получен с помощью ключа расшифрования d, соответствующего известному открытому ключу шифрования Q. А именно, доказать, что

$$R' = dR, \qquad Q = dP$$

Таким образом, в качестве точек Y_1, G_1, Y_2, G_2 выбираются точки R', R, Q, P соответственно. Скаляр x равен ключу расшифрования d.

1.2.6 Протокол разделения ключа

В качестве протокола разделения ключа используется (t, n)-схема Шамира, определенная в работе [5], позволяющая разделить секрет между n участниками таким образом, что любые t участников, объединившись, могут восстановить секрет.

Пусть есть некоторый секрет $y \in \mathbb{Z}_p^*$ (p – фиксированное простое число) и множество из n участников, каждый из которых имеет свой номер $i \in \{1, \dots, n\}$. Зададим алгоритмы разделения и восстановления секрета. Предполагается, что их выполняет некоторый доверенный диллер.

Алгоритм разделения ключа

1) Для $i = 1, \dots, t-1$ выбрать $\alpha_i \leftarrow_{\$} \mathbb{Z}_p^*$.
2) Построить полином:

$$f(x) = y + \sum_{i=1}^{t-1} \alpha_i x^i$$

3) Вычислить значение $f(x)$ в точках $i \in \{1, \dots, n\}$. Значение $y_i = f(i)$ называется секретной долей i-го участника.
4) Раздать каждому участнику соответствующую секретную долю.

Алгоритм восстановления ключа

1) Выбрать подмножество $T \subseteq \{1, \dots, n\}$ из t участников запросить их секретные доли $y_i, i \in T$.
2) Восстановить полином $f(x)$, используя интерполяцию:

$$f(x) = \sum_{i \in T} l_i(x) y_i \ mod\ p,$$

$$l_i(x) = \prod_{j \in T, i \neq j} \frac{x - j}{i - j} \ mod\ p.$$

3) Вычислить значение $f(0) = y$.

1.2.7 Схема обязательств (commitment)

В качестве схемы обязательств используется схема HMAC_GOSTR3411_2012_256 с длиной ключа 256 бит, определенная в Р 50.1.113–2016 [11].

1.2.8 Схема аутентифицированного шифрования

Алгоритм генерации ключей

Случайным образом генерируются ключи K_{Enc} и K_{Auth} длины 256 битов.

Алгоритм аутентифицированного шифрования

1) Случайным образом генерируется вектор инициализации IV длины 64 бита.
2) Сообщение M шифруется на ключе K_{Enc} с вектором инициализации IV:

$$C = Encrypt(K_{Enc}, IV,\ M).$$

В качестве функции $Encrypt$ используется блочный шифр Магма, определенный в ГОСТ Р 34.12-2015 [9], в режиме гаммирования с обратной связью, определенном в ГОСТ Р 34.13-2015 [10].

3) Для последовательности из IV и C на ключе K_{Auth} считается значение функции HMAC:

$$T = HMAC(K_{Auth}, IV \,||\, C)$$

В качестве схемы HMAC используется схема HMAC_GOSTR3411_2012_256, определенная в Р 50.1.113–2016 [11].

4) Возвращается набор (IV, C, T).

Алгоритм расшифрования

1) Вычисляется значение T' как $HMAC(K_{Auth}, IV \,||\, C)$.
2) Если $T' = T$, происходит расшифрование:

$$M = Decrypt(K_{Enc}, IV,\ C)$$

и возвращается значение M. В противном случае возвращается ошибка.

Список литературы

[1] Bellare M. et al. The One-More-RSA-Inversion Problems and the Security of Chaum's Blind Signature Scheme // Journal of Cryptology. – 2003. – Vol. 16. – No. 3. – P. 185–215.

[2] ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms // IEEE transactions on information theory. – 1985. – Vol. 31. – No. 4. – P. 469–472.

[3] Cramer R., Damgård I., Schoenmakers B. Proofs of partial knowledge and simplified design of witness hiding protocols // Annual International Cryptology Conference. – Springer, Berlin, Heidelberg, 1994. – P. 174–187.

[4] Chaum D., Pedersen T. P. Wallet databases with observers // Annual international cryptology conference. – Springer, Berlin, Heidelberg, 1992. – P. 89–105.

[5] Shamir A. How to share a secret. // Communications of the ACM. – 1979. – Vol. 22. – No. 11. - P. 612–613.

[6] ГОСТ Р 34.10–2012. Информационная технология. Криптографическая защита информации. Процессы формирования и проверки электронной цифровой подписи. М.: Стандартинформ, 2013.

[7] Р 1323565.1.024–2019. Информационная технология. Криптографическая защита информации. Параметры эллиптических кривых для криптографических алгоритмов и протоколов. М.: Стандартинформ, 2019.

[8] ГОСТ Р 34.11–2012. Информационная технология. Криптографическая защита информации. Функция хэширования. М.: Стандартинформ, 2013.

[9] ГОСТ Р 34.12–2015. Информационная технология. Криптографическая защита информации. Блочные шифры. М.: Стандартинформ, 2015.

[10] ГОСТ Р 34.13–2015. Информационная технология. Криптографическая защита информации. Режимы работы блочных шифров. М.: Стандартинформ, 2015.

[11] Р 50.1.113–2016. Информационная технология. Криптографическая защита информации. Криптографические алгоритмы, сопутствующие применению алгоритмов электронной цифровой подписи и функции хэширования. М.: Стандартинформ, 2019.