ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ 34.101.66-2014

Информационные технологии и безопасность

ПРОТОКОЛЫ ФОРМИРОВАНИЯ ОБЩЕГО КЛЮЧА НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Інфармацыйныя тэхналогіі і бяспека

ПРАТАКОЛЫ ФАРМІРАВАННЯ АГУЛЬНАГА КЛЮЧА НА АСНОВЕ ЭЛІПТЫЧНЫХ КРЫВЫХ

Госстандарт
Минск

УДК 004.421.056.5(083.74)(476) МКС 35.240.40 КП 05

Ключевые слова: криптографический протокол, формирование общего ключа, аутентификация, криптографические алгоритмы на основе эллиптических кривых

Предисловие

Цели, основные принципы, положения по государственному регулированию и управлению в области технического нормирования и стандартизации установлены Законом Республики Беларусь «О техническом нормировании и стандартизации».

1 РАЗРАБОТАН учреждением Белорусского государственного университета «Научно-исследовательский институт прикладных проблем математики и информатики»

ВНЕСЕН Оперативно-аналитическим центром при Президенте Республики Беларусь

2 УТВЕРЖДЕН И ВВЕДЕН В ДЕЙСТВИЕ постановлением Госстандарта Республики Беларусь от 22 мая 2014 г. № 23

3 ВВЕДЕН ВПЕРВЫЕ

Издан на русском языке

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

Информационные технологии и безопасность

ПРОТОКОЛЫ ФОРМИРОВАНИЯ ОБЩЕГО КЛЮЧА НА ОСНОВЕ ЭЛЛИПТИЧЕСКИХ КРИВЫХ

Інфармацыйныя тэхналогіі і бяспека

ПРАТАКОЛЫ ФАРМІРАВАННЯ АГУЛЬНАГА КЛЮЧА НА АСНОВЕ ЭЛІПТЫЧНЫХ КРЫВЫХ

Information technology and security

Key establishment protocols based on elliptic curves

Дата введения 2014-09-01

1 Область применения

Настоящий стандарт устанавливает протоколы BMQV, BSTS и BPACE, которые позволяют сторонам-участникам сформировать общий секретный ключ. С помощью общего ключа стороны могут выполнять аутентификацию, шифрование, имитозащиту, другие криптографические операции.

Настоящий стандарт применяется при разработке средств криптографической защиты информации, в том числе средств аутентификации и шифрования.

2 Нормативные ссылки

В настоящем стандарте использованы ссылки на следующие технические нормативные правовые акты в области технического нормирования и стандартизации (далее — ТНПА):

СТБ 34.101.17-2012 Информационные технологии и безопасность. Синтаксис запроса на получение сертификата

СТБ 34.101.19-2012 Информационные технологии. Форматы сертификатов и списков отозванных сертификатов инфраструктуры открытых ключей

СТБ 34.101.31-2011 Информационные технологии. Защита информации. Криптографические алгоритмы шифрования и контроля целостности

СТБ 34.101.45-2013 Информационные технологии и безопасность. Алгоритмы электронной цифровой подписи и транспорта ключа на основе эллиптических кривых

СТБ 34.101.47-2012 Информационные технологии и безопасность. Криптографические алгоритмы генерации псевдослучайных чисел

ГОСТ 34.973-91 (ИСО 8824-87) Информационная технология. Взаимосвязь открытых систем. Спецификация абстрактно-синтаксической нотации версии 1 (АСН.1)

Примечание — При пользовании настоящим стандартом целесообразно проверить действие ТНПА по каталогу, составленному по состоянию на 1 января текущего года, и по соответствующим информационным указателям, опубликованным в текущем году.

Если ссылочные ТНПА заменены (изменены), то при пользовании настоящим стандартом следует руководствоваться замененными (измененными) ТНПА. Если ссылочные ТНПА от-

менены без замены, то положение, в котором дана ссылка на них, применяется в части, не затрагивающей эту ссылку.

3 Термины и определения

В настоящем стандарте применяют следующие термины с соответствующими определениями:

3.1 аутентификация: Проверка подлинности стороны.

3.2 долговременный ключ: Ключ, который используется в нескольких сеансах протокола.

3.3 зашифрование: Преобразование сообщения, направленное на обеспечение его конфиденциальности, которое выполняется с использованием секретного ключа.

3.4 имитовставка: Контрольная характеристика сообщения, которая определяется с использованием секретного ключа и служит для контроля целостности и подлинности сообщения.

3.5 имитозащита: Контроль целостности и подлинности сообщений, который реализуется путем выработки и проверки имитовставок.

3.6 ключ: Параметр, который управляет криптографическими операциями зашифрования и расшифрования, имитозащиты, выработки и проверки ЭЦП, формирования общего ключа и др.

3.7 конфиденциальность: Гарантия того, что сообщения доступны для использования только тем сторонам, которым они предназначены.

3.8 личный ключ: Ключ, который связан с конкретной стороной, не является общедоступным и используется в настоящем стандарте для формирования общего ключа и для выработки электронной цифровой подписи.

3.9 одноразовый ключ: Ключ, который создается, используется и уничтожается в течение одного сеанса протокола.

3.10 октет: Двоичное слово длины 8.

3.11 открытый ключ: Ключ, который строится по личному ключу, связан с конкретной стороной, может быть сделан общедоступным и используется в настоящем стандарте для формирования общего ключа и для проверки электронной цифровой подписи.

3.12 пароль: Секрет, который способен запомнить человек и который поэтому может принимать сравнительно небольшое число значений.

3.13 подлинность: Гарантия того, что сторона действительно та, за кого себя выдает; гарантия того, что сторона действительно является владельцем (создателем, отправителем) определенного сообщения.

3.14 подтверждение ключа: Проверка того, что сформированый стороной общий ключ корректен.

3.15 построение ключа: Создание ключа по общим секретным и дополнительным открытым данным, завершающая стадия формирования общего ключа.

3.16 протокол: Интерактивный криптографический алгоритм, который выполняют несколько сторон-участников, обмениваясь между собой сообщениями, содержащими промежуточные результаты вычислений.

3.17 расшифрование: Преобразование, обратное зашифрованию.

3.18 сеанс протокола: Конкретная реализация (прогон) протокола.

3.19 секретный ключ: Ключ, который связан с конкретными сторонами, не является общедоступным и используется в настоящем стандарте для шифрования, имитозащиты, формирования общего ключа, генерации псевдослучайных чисел.

3.20 сертификат: Структурированные данные, связывающие идентификатор стороны с ее долговременным открытым ключом.

3.21 синхропосылка: Открытые входные данные криптографического алгоритма, которые обеспечивают уникальность результатов криптографического преобразования на фиксированном ключе.

3.22 сообщение: Двоичное слово конечной длины.

3.23 формирование общего ключа: Процедура, в результате которой несколько сторон определяют один и тот же ключ, известный только им.

3.24 хэш-значение: Двоичное слово фиксированной длины, которое определяется по сообщению без использования ключа и служит для контроля целостности сообщения и для представления сообщения в (необратимо) сжатой форме.

3.25 хэширование: Выработка хэш-значений.

3.26 целостность: Гарантия того, что сообщение не изменено при хранении или передаче.

3.27 шифрование: Зашифрование или расшифрование.

3.28 электронная цифровая подпись; ЭЦП: Контрольная характеристика сообщения, которая вырабатывается с использованием личного ключа, проверяется с использованием открытого ключа, служит для контроля целостности и подлинности сообщения и обеспечивает невозможность отказа от авторства.

4 Обозначения

4.1 Список обозначений

$\{0,1\}^n$	множество всех слов длины n в алфавите $\{0,1\}$;
$\{0,1\}^*$	множество всех слов конечной длины в алфавите $\{0,1\}$ (включая пустое слово длины 0);
$\lvert u\rvert$	длина слова $u \in \{0,1\}^*$;
$\{0,1\}^{n*}$	множество всех слов из $\{0,1\}^*$, длина которых кратна n;
α^n	для $\alpha \in \{0,1\}$ слово длины n из одинаковых символов α;
$\langle u\rangle_n$	для $u \in \{0,1\}^*$ слово из первых n символов u, $n \leqslant \lvert u\rvert$;
$u \parallel v$	конкатенация $u_1u_2\dots u_nv_1v_2\dots v_m$ слов $u = u_1u_2\dots u_n$ и $v = v_1v_2\dots v_m$;
$\texttt{01234...}_{16}$	представление $u \in \{0,1\}^{4*}$ шестнадцатеричным словом, при котором последовательным четырем символам u соответствует один шестнадцатеричный символ (например, $10100010 = \texttt{A2}_{16}$);
$x \bmod m$	для целого числа x и натурального числа m остаток от деления x на m, т. е. число $r \in \{0,1,\dots,m-1\}$ такое, что m делит $x - r$;
$x \equiv y \pmod m$	x сравнимо с y по модулю m, т. е. $x \bmod m = y \bmod m$;

$\bar{u}$	а) для $u = u_1 u_2 \dots u_8 \in \{0,1\}^8$ число $2^7 u_1 + 2^6 u_2 + \dots + u_8$ и б) для $u = u_1 \parallel u_2 \parallel \dots \parallel u_n$, $u_i \in \{0,1\}^8$, число $\bar{u}_1 + 2^8 \bar{u}_2 + \dots + 2^{8(n-1)} \bar{u}_n$;
$\langle U \rangle_{8n}$	для целого числа U слово $u \in \{0,1\}^{8n}$ такое, что $\bar{u} = U \bmod 2^{8n}$;
$\mathbb{F}_p$	для простого числа p множество $\{0, 1, \dots, p-1\}$ с операциями сложения и умножения по модулю p, конечное поле из p элементов;
$E^*_{a,b}(\mathbb{F}_p)$	для $a, b \in \mathbb{F}_p$ множество решений (x, y), $x, y \in \mathbb{F}_p$, уравнения $y^2 = x^3 + ax + b$, множество аффинных точек эллиптической кривой;
O	бесконечно удаленная точка;
$E_{a,b}(\mathbb{F}_p)$	множество $E^*_{a,b}(\mathbb{F}_p) \cup \{O\}$ с операцией сложения точек, группа точек эллиптической кривой;
kP	для $P \in E_{a,b}(\mathbb{F}_p)$ сумма k экземпляров P, кратная P точка;
l	уровень стойкости, число из множества $\{128, 192, 256\}$;
$\langle P \rangle_n$	для $P = (x, y) \in E^*_{a,b}(\mathbb{F}_p)$, где $2^{2l-1} < p < 2^{2l}$, слово из первых n символов слова $\langle x \rangle_{2l} \parallel \langle y \rangle_{2l}$, $n \leqslant 4l$;
$c \leftarrow u$	присвоение переменной c значения u;
$c \xleftarrow{R} U$	случайный равновероятный (или псевдослучайный) выбор c из множества U;
A, B	стороны протокола;
$A \to B$	передача сообщения от A к B;
$\left[\text{текст}\right]$	необязательное сообщение (действие) протокола;
$\{\text{текст}\}$	обязательное сообщение (действие) протокола, которое может быть передано (выполнено) предварительно, до сеанса протокола, или неявно;
$\texttt{Cert}(Id, Q)$	сертификат, связывающий идентификатор Id с открытым ключом Q;
`hello`	приветственное сообщение.

4.2 Пояснения к обозначениям

4.2.1 Слова

Двоичные слова представляют собой последовательности символов из алфавита $\{0, 1\}$. Символы нумеруются слева направо от единицы. В настоящем подразделе в качестве примера рассматривается слово

$$w = 10110001100101001011101011001000.$$

В этом слове первый символ — 1, второй — 0, ..., последний — 0.

Слова разбиваются на тетрады из четверок последовательных двоичных символов. Тетрады кодируются шестнадцатеричными символами по следующим правилам (см. таблицу 1):

Таблица 1

тетрада	символ	тетрада	символ	тетрада	символ	тетрада	символ
0000	0_{16}	0001	1_{16}	0010	2_{16}	0011	3_{16}
0100	4_{16}	0101	5_{16}	0110	6_{16}	0111	7_{16}
1000	8_{16}	1001	9_{16}	1010	A_{16}	1011	B_{16}
1100	C_{16}	1101	D_{16}	1110	E_{16}	1111	F_{16}

Пары последовательных тетрад образуют октеты. Последовательные октеты слова w имеют вид:

$$10110001 = B1_{16},\ 10010100 = 94_{16},\ 10111010 = BA_{16},\ 11001000 = C8_{16}.$$

4.2.2 Слова как числа

Октету $u = u_1u_2\dots u_8$ ставится в соответствие байт — число $\bar{u} = 2^7u_1 + 2^6u_2 + \dots + u_8$. Например, октетам w соответствуют байты

$$177 = 2^7 + 2^5 + 2^4 + 1,\ 148 = 2^7 + 2^4 + 2^2,\ 186 = 2^7 + 2^5 + 2^4 + 2^3 + 2^1,\ 200 = 2^7 + 2^6 + 2^3.$$

Число ставится в соответствие не только октетам, но и любому другому двоичному слову, длина которого кратна 8. При этом используется распространенное для многих современных процессоров соглашение «от младших к старшим» (little-endian): считается, что первый байт является младшим, последний — старшим. Например, слову w соответствует число

$$\bar{w} = 177 + 2^8 \cdot 148 + 2^{16} \cdot 186 + 2^{24} \cdot 200 = 3367670961.$$

4.2.3 Конечные поля

Элементы $\mathbb{F}_p$ складываются и умножаются как целые числа с заменой результата на остаток от его деления на p. Множество $\mathbb{F}_p$ с такими операциями является конечным простым полем. Нулевым элементом поля является число 0, а мультипликативной единицей — число 1 (подробнее см. [1]).

Кроме сложения и умножения, в поле $\mathbb{F}_p$ можно выполнять вычитание и деление. Вычитание u состоит в сложении с $p-u$. Деление на $u \in \{1, 2, \dots, p-1\}$ состоит в умножении на число $v \in \{1, 2, \dots, p-1\}$ такое, что $uv \equiv 1 \pmod p$.

Например, в поле $\mathbb{F}_7$ выполняется:

$$4 + 5 = 2, \quad 4 \cdot 5 = 6, \quad 4 - 5 = 4 + (7 - 5) = 6, \quad 4/5 = 4 \cdot 3 = 5.$$

Квадраты ненулевых элементов $\mathbb{F}_p$ называются квадратичными вычетами по модулю p. Например, имеется 3 квадратичных вычета по модулю 7:

$$1 = 1^2, \quad 2 = 3^2, \quad 4 = 2^2.$$

4.2.4 Эллиптические кривые

Пусть $p > 3$ и $4a^3 + 27b^2 \not\equiv 0 \pmod p$. Множество $E^*_{a,b}(\mathbb{F}_p)$ состоит из решений уравнения $y^2 = x^3 + ax + b$ относительно $x, y \in \mathbb{F}_p$. Уравнение такого вида определяет эллиптическую кривую над полем $\mathbb{F}_p$, его решения (x, y) называются аффинными точками

кривой. К аффинным точкам добавляется специальная бесконечно удаленная точка O и образуется множество $E_{a,b}(\mathbb{F}_p)$ (подробнее см. [2]). Например,

$$E_{4,1}(\mathbb{F}_7) = \{O, (0,1), (0,6), (4,2), (4,5)\}.$$

Множество $E_{a,b}(\mathbb{F}_p)$ является аддитивной группой при следующих правилах сложения:

1 $O + P = P + O = P$ для всех $P \in E_{a,b}(\mathbb{F}_p)$.

2 Если $P = (x, y) \in E^*_{a,b}(\mathbb{F}_p)$, то $-P = (x, p - y)$ и $P + (-P) = O$.

3 Если $P_1 = (x_1, y_1) \in E^*_{a,b}(\mathbb{F}_p)$, $P_2 = (x_2, y_2) \in E^*_{a,b}(\mathbb{F}_p)$ и $P_2 \neq -P_1$, то $P_1 + P_2 = (x_3, y_3)$,

где $x_3 = \lambda^2 - x_1 - x_2$, $\quad y_3 = \lambda(x_1 - x_3) - y_1$, $\quad \lambda = \begin{cases} \dfrac{y_2 - y_1}{x_2 - x_1}, & P_1 \neq P_2, \\ \dfrac{3x_1^2 + a}{2y_1}, & P_1 = P_2 \end{cases}$

(вычисления ведутся в $\mathbb{F}_p$).

Сумма k экземпляров точки P называется k-кратной ей точкой и обозначается через kP. Например, для $P = (4,2) \in E_{4,1}(\mathbb{F}_7)$ ее кратные имеют вид:

$$2P = (4,2) + (4,2) = (0,1), \qquad 3P = (0,1) + (4,2) = (0,6),$$
$$4P = 2(0,1) = (4,5), \qquad 5P = (0,1) + (0,6) = O.$$

Считается, что $0P = O$.

4.2.5 Обозначения протоколов

Протоколы настоящего стандарта выполняют две стороны, которые обозначаются символами A и B. Объекты, которыми владеют или управляют стороны (ключи, сообщения, идентификаторы, переменные), снабжаются нижними индексами: A или B.

Квадратными скобками окаймляются необязательные сообщения и действия сторон протокола, связанные с явным подтверждением ключа (см. 5.1).

Фигурными скобками окаймляются сертификаты (см. 5.6) и приветственные сообщения (см. 5.8), которые могут передаваться предварительно или неявно. Действия по передаче и обработке таких сообщений также окаймляются фигурными скобками.

5 Общие положения

5.1 Назначение

Настоящий стандарт определяет криптографические протоколы, которые позволяют двум сторонам сформировать общий, только им известный, ключ. Ключ формируется так, что ни одна из сторон не может определить его значение самостоятельно, без данных от противоположной стороны.

Общий ключ представляет собой двоичное слово длины 256. Этот ключ может использоваться для аутентификации, шифрования, имитозащиты, выработки других общих секретных данных. В криптографических алгоритмах, длины ключей которых меньше 256, может использоваться только часть общего ключа. При формировании этой части в общем ключе следует отбрасывать последние символы и оставлять первые.

Протоколы построены по схемам MQV [3], STS [4] и PACE [5]. Схема определяет общий вид протокола и не содержит исчерпывающих деталей, обеспечивающих совместимость различных его реализаций. В настоящем стандарте зафиксированы все базовые криптографические алгоритмы схем, окончательно определены действия сторон, уточнены форматы передаваемых сообщений. В схемы MQV и STS встроены алгоритмы ЭЦП, схожие с алгоритмами СТБ 34.101.45. В схеме STS выработка одноразовой подписи совмещена с расчетом одноразовых открытых ключей. Окончательные протоколы названы BMQV (7.4), BSTS (7.5) и BPACE (7.6).

Протоколы BMQV и BSTS позволяют сторонам сформировать общий ключ, используя долговременные личные ключи и обмениваясь соответствующими открытыми ключами. Эффективность BMQV выше, но BSTS дополнительно обеспечивает анонимность — злоумышленник, который не вступает во взаимодействие со сторонами, а только перехватывает их сообщения, не получает информации о том, какие стороны участвуют в протоколе.

Протокол BPACE позволяет сторонам сформировать общий ключ, используя общий пароль. Злоумышленнику, который перехватывает все сообщения протокола (вступая или не вступая во взаимодействие со сторонами), вычислительно трудно определить пароль, даже если он короткий или низкоэнтропийный. Выполняя сеанс протокола с легальной стороной, злоумышленник может проверить только один вариант пароля.

Все протоколы настоящего стандарта обеспечивают конфиденциальность ключей, сформированных сторонами. Кроме этого, в протоколах предусмотрена проверка того, что та или другая сторона сформировала корректный ключ, совпадающий с ключом противоположной стороны. Такая проверка называется явным подтверждением ключа. Подтверждать ключ может либо одна, либо обе стороны. Явное подтверждение ключа является обязательной частью BSTS и необязательной частью BMQV и BPACE.

Даже если подтверждение ключа явно не включено в протокол, оно может быть неявно выполнено после его завершения. Если, например, стороны используют общий ключ для шифрования структурированных данных, то нарушение формата данных после расшифрования является признаком того, что ключи сторон отличаются.

Успешное подтверждение ключа в протоколе BPACE означает, что сторона знает секретный пароль. Поэтому протокол может использоваться для аутентификации сторон друг перед другом.

Успешное подтверждение ключа в протоколах BMQV и BSTS означает, что сторона с определенным идентификатором владеет долговременным личным ключом, который соответствует отосланному ею открытому ключу. Данные протоколы также могут использоваться для аутентификации, если дополнительно проверяется соответствие между идентификатором и долговременным открытым ключом стороны. Такое соответствие устанавливается за рамками протоколов и фиксируется в сертификатах (см. 5.6).

В протоколах используются вычисления в группе точек эллиптической кривой над конечным простым полем. Сначала стороны формируют общую секретную точку кривой, а затем по ней и дополнительным открытым данным (в том числе сообщениям протокола) вырабатывают общий ключ. Для этого используется алгоритм построения ключа,

определенный в 6.1. С помощью этого алгоритма формируются также служебные ключи, предназначенные для имитозащиты и шифрования служебных данных протоколов.

В 6.2 определяется вспомогательный алгоритм преобразования двоичного слова в точку эллиптической кривой. Этот алгоритм используется в протоколе BPACE. Алгоритм построен в соответствии с [6].

Все протоколы настоящего стандарта основаны на протоколе Диффи — Хеллмана, который описывается в приложении А. В этом приложении обсуждается стойкость протокола, и определяются условия его безопасного встраивания в высокоуровневые протоколы.

В приложении Б приводятся примеры выполнения протоколов BMQV, BSTS, BPACE. Примеры можно использовать для проверки корректности реализаций протоколов.

В приложении В приводится модуль абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ 34.973. Модуль задает идентификаторы протоколов и других объектов стандарта, описывает структуры данных для хранения ключей и параметров. Рекомендуется использовать модуль при встраивании протоколов в информационные системы, в которых также используется АСН.1.

5.2 Стойкость

Протоколы построены так, что злоумышленнику вычислительно трудно определить общий ключ или выдать себя за другую сторону, не зная долговременный личный ключ этой стороны (BMQV, BSTS) или общий пароль (BPACE).

Стойкость протоколов определяется уровнем $l \in \{128, 192, 256\}$. На уровне l для определения общего ключа по сообщениям протокола злоумышленнику требуется выполнить порядка 2^l операций. Стойкость основывается на сложности дискретного логарифмирования в группе точек эллиптической кривой и на сложности вычислительной задачи Диффи — Хеллмана в этой группе (см. приложение А). Оценки стойкости не изменятся, если на уровне $l \in \{128, 192\}$ длина общего ключа уменьшится с 256 до l.

Протоколы обеспечивают защиту от «чтения назад». Это значит, что определение общего ключа остается такой же трудной задачей, даже если злоумышленнику, дополнительно к сообщениям протокола, становятся известными долговременные личные ключи сторон или их общий пароль, но остаются неизвестными одноразовые личные или секретные ключи.

Уровень l определяет длины параметров, ключей, сообщений и, соответственно, быстродействие протоколов. Следует учитывать, что с ростом l, кроме повышения стойкости, снижается быстродействие.

Явное подтверждение ключа основано на проверке имитовставок, вычисленных на общем служебном ключе. Злоумышленник, который не знает этот ключ, может обойти механизм подтверждения только в одном из 2^{64} сеансов протокола в среднем. В протоколе BSTS, кроме имитовставок, проверяются также зашифрованные одноразовые подписи и сертификаты, поэтому надежность подтверждения ключа в BSTS еще выше.

Явное подтверждение ключа выполняется так, что злоумышленник не может подтвердить ключ, отвечая данными, которыми подтверждает ключ противоположная сторона. Неявное подтверждение ключа, выполняемое за рамками BMQV и BSTS, также должно обладать этим свойством.

Стойкость механизма аутентификации основана на стойкости механизма подтверждения ключа и, дополнительно для BMQV и BSTS, на надежности связывания идентификаторов сторон с их открытыми ключами в сертификатах (см. 5.6). Если связывание задается с помощью ЭЦП доверенной стороны, то надежность связывания определяется стойкостью алгоритмов ЭЦП.

5.3 Параметры эллиптической кривой

Модуль p**.** Используется простое число p, которое удовлетворяет условиям: $2^{2l-1} < p < 2^{2l}$, $p \equiv 3 \pmod 4$. Модуль определяет поле $\mathbb{F}_p$, над которым строится эллиптическая кривая. Можно использовать произвольное допустимое p, в том числе простое специального вида.

Коэффициенты a**,** b**.** Используются числа $a, b \in \mathbb{F}_p$, которые удовлетворяют условиям: $a \neq 0$, b является квадратичным вычетом по модулю p, $4a^3 + 27b^2 \not\equiv 0 \pmod p$. Коэффициенты a, b вместе с модулем p определяют группу точек эллиптической кривой $E_{a,b}(\mathbb{F}_p)$.

Порядок q**.** После построения группы $E_{a,b}(\mathbb{F}_p)$ рассчитывается ее порядок $q = |E_{a,b}(\mathbb{F}_p)|$. Выбирается группа, порядок которой удовлетворяет следующим ограничениям: q — простое, $2^{2l-1} < q < 2^{2l}$, $q \neq p$, q не делит числа вида $p^m - 1$ для $m = 1, 2, \ldots, 50$.

Базовая точка G**.** Используется базовая точка $G \in E^*_{a,b}(\mathbb{F}_p)$ вида $G = (0, y_G)$, где $y_G = b^{(p+1)/4} \bmod p$. Кратные G, $2G$,..., $(q-1)G$ базовой точки пробегают все элементы $E^*_{a,b}(\mathbb{F}_p)$, а $qG = O$.

Параметры должны генерироваться с помощью алгоритма, определенного в СТБ 34.101.45 (пункт 6.1.3), и, соответственно, должны удовлетворять условиям алгоритма проверки параметров, также определенного в СТБ 34.101.45 (пункт 6.1.4).

В СТБ 34.101.45 (приложение Б) приводятся стандартные наборы параметров эллиптической кривой, которые можно использовать напрямую, без повторной генерации или проверки.

5.4 Долговременные ключи

В протоколах BMQV и BSTS стороны используют долговременные личный и открытый ключи. Личным ключом является число $d \in \{1, 2, \ldots, q-1\}$. По личному ключу определяется открытый ключ $Q \in E^*_{a,b}(\mathbb{F}_p)$. Для генерации пары ключей должен использоваться алгоритм, определенный в СТБ 34.101.45 (пункт 6.2.2).

При хранении и распространении должны обеспечиваться конфиденциальность и контроль целостности личного ключа, контроль целостности открытого ключа.

Долговременный личный ключ d должен использоваться только в определенном протоколе: BMQV или BSTS. Ключ d может дополнительно использоваться в алгоритмах ЭЦП, определенных в СТБ 34.101.45, например, для проверки владения ключом при формировании сертификатов, как это описано в 5.6. Использование личного ключа в других алгоритмах запрещено.

В информационных системах ключи представляются двоичными словами. Для обеспечения совместимости рекомендуется представлять личный ключ d словом $\langle d \rangle_{2l}$, а открытый ключ Q — словом $\langle Q \rangle_{4l}$.

5.5 Пароль

В протоколе BPACE стороны используют общий секретный пароль. Паролем является двоичное слово, длина которого кратна 8. Это слово может являться кодированным представлением обычной текстовой строки. Кодировать рекомендуется по правилам UTF-8, заданным в [7].

5.6 Сертификаты

Стороны протоколов BMQV и BSTS характеризуются отличительными идентификаторами Id. Долговременный открытый ключ Q стороны связывается с ее идентификатором и распространяется в виде сертификата $\mathtt{Cert}(Id, Q) \in \{0,1\}^*$. При выполнении протокола сертификат проверяется, Проверка сертификата должна включать проверку корректности связывания и проверку того, что Q действительно является аффинной точкой эллиптической кривой. Для проверки Q может использоваться алгоритм, определенный в СТБ 34.101.45 (6.2.3).

Примечание 1 — В качестве Id может выступать полное имя, сетевой адрес, уникальный номер, контрольная характеристика Q. В последнем случае идентификатор может явно не включаться в сертификат, сам сертификат представлять собой слово $\langle Q \rangle_{4l}$, а его проверка состоять в сравнении идентификатора стороны протокола с контрольной характеристикой Q.

В протоколе BMQV передача и проверка сертификата могут выполняться предварительно, до сеанса протокола, или неявно. Например, сертификат сервера B может предварительно распространяться как часть программы, которая реализует протокол со стороны клиента A. Или сертификат может передаваться по каналу, который обеспечивает целостность и подлинность данных, и тогда проверка этого сертификата выполняется неявно, как следствие передачи по надежному каналу.

Содержание, формат и способ проверки сертификата в настоящем стандарте не оговариваются. Тем не менее, при проектировании системы управления сертификатами следует иметь в виду, что надежность проверки сертификатов определяет надежность аутентификации посредством протоколов (см. 5.2).

Типичным является сертификат открытого ключа, определеный в СТБ 34.101.19. В этом сертификате кроме идентификатора и открытого ключа, определяются параметры криптографических алгоритмов, срок действия, атрибуты, а также ЭЦП доверенной стороны под всеми перечисленными данными. Проверка сертификата СТБ 34.101.19 состоит в проверке подписи доверенной стороны.

Примечание 2 — ЭЦП доверенной стороны может сопровождаться сертификатом ее открытого ключа, нужного для проверки подписи. Новый сертификат подписывается другой доверенной стороной, подпись которой также может сопровождаться сертификатом, и так далее по цепочке, вплоть до достоверно полученного самоподписанного сертификата [подробнее см. СТБ 34.101.19 (раздел 8)]. Сопровождающая ЭЦП доверенной стороны цепочка сертификатов может считаться частью $\mathtt{Cert}(Id, Q)$. При этом проверка $\mathtt{Cert}(Id, Q)$ должна включать проверку сертификатов цепочки.

При формировании $\texttt{Cert}(Id, Q)$ может проверяться, что предполагаемый владелец сертификата знает личный ключ d, которому соответствует Q. Такая проверка предусмотрена, например, в СТБ 34.101.17. Проверка владения личным ключом неявно проводится в каждом из протоколов BMQV или BSTS и, вообще говоря, не обязательна при формировании сертификата для этих протоколов. Если такая проверка все-таки предусмотрена тем или иным регламентом, то она может быть выполнена с помощью алгоритмов ЭЦП. Для этого владельцу сертификата должно быть предложено подписать на ключе d данные сертификата, в том числе Id и Q. Корректность ЭЦП означает, что владелец знает личный ключ.

В описанном способе проверки владения личным ключом должны использоваться алгоритмы ЭЦП, определенные в СТБ 34.101.45. В алгоритмах ЭЦП должны использоваться те же параметры эллиптической кривой, что и в протоколе, в котором будет применяться сертификат.

5.7 Одноразовые ключи

Кроме долговременных ключей и паролей стороны используют одноразовые ключи. Во всех протоколах применяются одноразовый личный ключ $u \in \{1, 2, \ldots, q-1\}$ и одноразовый открытый ключ $V \in E^*_{a,b}(\mathbb{F}_p)$. В протоколе BPACE дополнительно используется одноразовый секретный ключ $R \in \{0,1\}^l$.

Одноразовые личный и секретный ключи должны вырабатываться без возможности предсказания и уничтожаться после использования.

Для создания личных и секретных одноразовых ключей может использоваться физический генератор случайных чисел, удовлетворяющий ТНПА, или алгоритм генерации псевдослучайных чисел, определенный в СТБ 34.101.47 или в другом ТНПА. Входные данные алгоритма должны включать секретный ключ, известный только владельцу генерируемого ключа, и уникальную синхропосылку. Длина ключа алгоритма генерации должна быть не меньше l.

Примечание — Личные ключи — числа из множества $\{1, 2, \ldots, q-1\}$ — генерируются, как правило, в два этапа: сначала строятся случайные или псевдослучайные двоичные слова, которые затем преобразуются в числа. Для генерации личных ключей по такой схеме рекомендуется строить слова $u \in \{0,1\}^{2l}$ до тех пор, пока не будет выполнено условие $\bar{u} \in \{1, 2, \ldots, q-1\}$, и объявлять окончательное число $\bar{u}$ результатом генерации. В среднем потребуется проверить $2^{2l}/(q-1)$ чисел-кандидатов.

5.8 Приветственные сообщения

Протоколы начинаются с обмена приветственными сообщениями. Сначала сторона A отправляет сообщение $\texttt{hello}_A$. В этом сообщении A может указать список протоколов, которые она готова выполнить, настройки протоколов, другие данные. Настройки могут описывать версию протокола, уровень стойкости, параметры эллиптической кривой, подсказку по выбору пароля, отметку времени и др. Сторона B отвечает сообщением $\texttt{hello}_B$. В этом сообщении B может выбрать устраивающий ее протокол и зафиксировать его настройки.

Приветственные сообщения участвуют в построении общего ключа и поэтому действия злоумышленника по изменению приветственных сообщений (например, для изменения настроек протокола) будут обнаружены.

Формат приветственных сообщений в настоящем стандарте не регламентируется. Передача и обработка приветственных сообщений может выполняться предварительно или неявно.

По умолчанию слова $\texttt{hello}_A$ и $\texttt{hello}_B$ пустые, если стороны выполняют заранее оговоренный протокол с фиксированными настройками.

6 Алгоритмы построения ключа и точки эллиптической кривой

6.1 Построение ключа

6.1.1 Входные и выходные данные

Входными данными алгоритма построения ключа являются секретное слово $X \in \{0,1\}^*$, дополнительное слово $S \in \{0,1\}^*$ и номер ключа C — неотрицательное целое число.

Выходными данными является ключ $Y \in \{0,1\}^{256}$.

6.1.2 Вспомогательные алгоритмы

Алгоритм `belt-hash`. Используется алгоритм хэширования `belt-hash`, определенный в СТБ 34.101.31 (пункт 6.9.3). Входными данными алгоритма является слово $X \in \{0,1\}^*$, выходными — его хэш-значение $Y \in \{0,1\}^{256}$.

Алгоритм `belt-keyrep`. Используется алгоритм преобразования ключа `belt-keyrep`, определенный в СТБ 34.101.31 (пункт 7.2.3). Входными данными алгоритма является преобразуемый ключ $X \in \{0,1\}^{256}$, уровень $D \in \{0,1\}^{96}$, заголовок $I \in \{0,1\}^{128}$, длина $m \in \{128, 192, 256\}$. Выходными данными является преобразованный ключ $Y \in \{0,1\}^m$.

6.1.3 Алгоритм построения ключа

Построение ключа состоит в выполнении следующих шагов:

1 $Y \leftarrow \texttt{belt-hash}(X \,\|\, S)$.

2 $Y \leftarrow \texttt{belt-keyrep}(Y, 1^{96}, \langle C \rangle_{128}, 256)$.

3 Возвратить Y.

6.2 Построение точки эллиптической кривой

6.2.1 Входные и выходные данные

Входными данными алгоритма построения точки эллиптической кривой являются параметры p, a и b, которые описывают группу точек. Параметры должны удовлетворять требованиям, заданным в 5.3. По модулю p определяется уровень стойкости l как минимальное натуральное, для которого $p < 2^{2l}$.

Кроме параметров эллиптической кривой, входными данными алгоритма является слово $X \in \{0,1\}^{2l}$.

Выходными данными является точка $W \in E^*_{a,b}(\mathbb{F}_p)$, представляющая X.

6.2.2 Вспомогательные алгоритмы и переменные

Алгоритм `belt-keywrap`. Используется алгоритм `belt-keywrap`, определенный в СТБ 34.101.31 (пункт 6.8.3). Входными данными алгоритма являются защищаемый ключ $X \in \{0,1\}^{8*}$, заголовок $I \in \{0,1\}^{128}$ и ключ защиты $\theta \in \{0,1\}^{256}$. Выходными данными является защищенный ключ $Y \in \{0,1\}^{|X|+128}$.

Переменная H. Используется переменная $H \in \{0,1\}^{2l+128}$. Значение H должно быть уничтожено после использования.

Элементы $\mathbb{F}_p$. Используются переменные $s, t, x_1, x_2, y \in \mathbb{F}_p$. Значения переменных должны быть уничтожены после использования.

6.2.3 Алгоритм построения точки эллиптической кривой

Построение точки W состоит в выполнении следующих шагов:

1 Установить $H \leftarrow \texttt{belt-keywrap}(X, 0^{128}, 0^{256})$.
2 Установить $s \leftarrow \overline{H} \bmod p$.
3 Установить $t \leftarrow -s^2 \bmod p$.
4 Установить $x_1 \leftarrow -b(1 + t + t^2)(a(t + t^2))^{p-2} \bmod p$.
5 Установить $x_2 \leftarrow tx_1 \bmod p$.
6 Установить $y \leftarrow ((x_1)^3 + ax_1 + b) \bmod p$.
7 Установить $s \leftarrow s^3 y \bmod p$.
8 Установить $t \leftarrow y^{p-1-(p+1)/4} \bmod p$.
9 Если $t^2 y \equiv 1 \pmod p$, то $W \leftarrow (x_1, ty \bmod p)$, иначе $W \leftarrow (x_2, st \bmod p)$.
10 Возвратить W.

7 Протоколы

7.1 Входные и выходные данные

Входными данными протоколов BMQV, BSTS, BPACE являются параметры p, a, b, q и G, которые описывают группу точек эллиптической кривой. Параметры должны удовлетворять требованиям, заданным в 5.3. По модулю p определяется уровень стойкости l как минимальное натуральное, для которого $p < 2^{2l}$.

Параметры эллиптической кривой могут согласовываться в приветственных сообщениях $\texttt{hello}_A, \texttt{hello}_B$ (см. 5.8), которые также являются входными данными протоколов.

Кроме параметров эллиптической кривой и приветственных сообщений, входными данными протоколов BMQV и BSTS являются личные ключи d_A, d_B и сертификаты $\texttt{Cert}(Id_A, Q_A)$, $\texttt{Cert}(Id_B, Q_B)$. Ключи и сертификаты должны удовлетворять требованиям, заданным в 5.4, 5.6.

Кроме параметров эллиптической кривой и приветственных сообщений, входными данными протокола BPACE является общий пароль $P \in \{0,1\}^{8*}$.

Выходными данными протоколов является либо общий ключ $K_0 \in \{0,1\}^{256}$, либо признак `ОШИБКА`. Возврат признака `ОШИБКА` означает либо сбой при передаче сообщений протокола, либо нарушение целостности сообщений, либо нарушение их подлинности, либо ошибку аутентификации стороны протокола.

7.2 Вспомогательные алгоритмы

Алгоритм `belt-ecb`. В протоколе BPACE используется алгоритм зашифрования в режиме простой замены `belt-ecb`, определенный в СТБ 34.101.31 (пункт 6.2.3). Входными данными алгоритма являются сообщение $X \in \{0,1\}^*$, длина которого не меньше 128, и ключ $\theta \in \{0,1\}^{256}$. Выходными данными является зашифрованное сообщение $Y \in \{0,1\}^{|X|}$.

Алгоритм `belt-ecb`$^{-1}$. В протоколе BPACE используется алгоритм расшифрования в режиме простой замены `belt-ecb`$^{-1}$, определенный в СТБ 34.101.31 (пункт 6.2.4). Входными данными алгоритма являются зашифрованное сообщение $Y \in \{0,1\}^*$, длина которого не меньше 128, и ключ $\theta \in \{0,1\}^{256}$. Выходными данными является первоначальное сообщение $X \in \{0,1\}^{|Y|}$.

Алгоритм `belt-cfb`. В протоколе BSTS используется алгоритм зашифрования в режиме гаммирования с обратной связью `belt-cfb`, определенный в СТБ 34.101.31 (пункт 6.4.3). Входными данными алгоритма являются сообщение $X \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$. Выходными данными является зашифрованное сообщение $Y \in \{0,1\}^{|X|}$.

Алгоритм `belt-cfb`$^{-1}$. В протоколе BSTS используется алгоритм расшифрования в режиме гаммирования с обратной связью `belt-cfb`$^{-1}$, определенный в СТБ 34.101.31 (пункт 6.4.4). Входными данными алгоритма являются зашифрованное сообщение $Y \in \{0,1\}^*$, ключ $\theta \in \{0,1\}^{256}$ и синхропосылка $S \in \{0,1\}^{128}$. Выходными данными является первоначальное сообщение $X \in \{0,1\}^{|Y|}$.

Алгоритм `belt-mac`. Используется алгоритм выработки имитовставки `belt-mac`, определенный в СТБ 34.101.31 (пункт 6.6.3). Входными данными алгоритма являются сообщение $X \in \{0,1\}^*$ и ключ $\theta \in \{0,1\}^{256}$, выходными — имитовставка $T \in \{0,1\}^{64}$.

Алгоритм `belt-hash`. Используется алгоритм хэширования `belt-hash`, описанный в 6.1.2.

Алгоритм `bake-kdf`. Используется алгоритм построения ключа `bake-kdf`, определенный в 6.1.3.

Алгоритм `bake-swu`. В протоколе BPACE используется алгоритм построения точки эллиптической кривой `bake-swu`, определенный в 6.2.3.

Проверка открытого ключа. Стороны проверяют, что присланные открытые ключи V_A, V_B являются аффинными точками эллиптической кривой. Контроль может быть выполнен с помощью алгоритма проверки открытого ключа, определенного в СТБ 34.101.45 (пункт 6.2.3).

7.3 Переменные

Одноразовые личные ключи. Используются одноразовые личные ключи $u_A, u_B \in \{1, 2, \ldots, q-1\}$. Требования по управлению этими ключами определены в 5.7.

Одноразовые открытые ключи. Используются одноразовые открытые ключи $V_A, V_B \in E^*_{a,b}(\mathbb{F}_p)$.

Одноразовые секретные ключи. В протоколе BPACE используются одноразовые секретные ключи $R_A, R_B \in \{0,1\}^l$. Требования по управлению этими ключами определены в 5.7.

Одноразовые подписи. В протоколах BMQV, BSTS вырабатываются одноразовые подписи $s_A, s_B \in \{0, 1, \ldots, q-1\}$. В BMQV одноразовые подписи должны быть уничтожены после использования.

Общая секретная точка. Формируется общая секретная точка $K \in E_{a,b}(\mathbb{F}_p)$. Значение K должно быть уничтожено после использования.

Имитовставки. Для подтверждения общего ключа K_0 используются имитовставки $T_A, T_B \in \{0,1\}^{64}$.

Служебный ключ K_1. Для формирования имитовставок используется служебный ключ $K_1 \in \{0,1\}^{256}$. Ключ K_1 должен быть уничтожен после использования.

Служебный ключ K_2. В протоколах BSTS и BPACE для шифрования данных используется служебный ключ $K_2 \in \{0,1\}^{256}$. Ключ K_2 должен быть уничтожен после использования.

Зашифрованные данные. В протоколе BSTS с помощью K_2 выполняется зашифрование одноразовых подписей и сертификатов. В результате формируются слова $Y_A, Y_B \in \{0,1\}^*$. В протоколе BPACE зашифровываются одноразовые секретные ключи. В результате формируются слова $Y_A, Y_B \in \{0,1\}^l$.

Переменная t. В протоколах BMQV и BSTS каждая из сторон использует переменную $t \in \{0,1\}^l$.

Переменная W. В протоколе BPACE каждая из сторон использует переменную $W \in E^*_{a,b}(\mathbb{F}_p)$. Значение W должно быть уничтожено после использования.

7.4 Протокол BMQV

7.4.1 Сообщения

В протоколе BMQV стороны пересылают друг другу следующие сообщения:
M0 ($A \to B$): $\{\texttt{hello}_A\}$;
M1 ($B \to A$): $\{\texttt{hello}_B \parallel \}\ \{\texttt{Cert}(Id_B, Q_B) \parallel \}\ \langle V_B \rangle_{4l}$;
M2 ($A \to B$): $\{\texttt{Cert}(Id_A, Q_A) \parallel \}\ \langle V_A \rangle_{4l}\ [\ \parallel T_A]$;
M3 ($B \to A$): $[T_B]$.

7.4.2 Шаги

Формирование общего ключа состоит в выполнении шагов, определенных ниже. При ошибке на любом из шагов, в том числе при отрицательном результате любой проверки, протокол прекращается и возвращается признак `ОШИБКА`.

1 Сторона A:
  1) {отправляет сообщение M0}.

2 Сторона B:
  1) {получает сообщение M0};
  2) $u_B \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
  3) $V_B \leftarrow u_B G$;
  4) отправляет сообщение M1.

3 Сторона A:
  1) получает сообщение M1;
  2) {проверяет $\texttt{Cert}(Id_B, Q_B)$};

3) проверяет, что $V_B \in E^*_{a,b}(\mathbb{F}_p)$;
4) $u_A \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
5) $V_A \leftarrow u_A G$;
6) $t \leftarrow \big\langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \,\|\, \langle V_B \rangle_{2l}) \big\rangle_l$;
7) $s_A \leftarrow (u_A - (2^l + \bar{t}) d_A) \bmod q$;
8) $K \leftarrow s_A (V_B - (2^l + \bar{t}) Q_B)$;
9) если $K = O$, то $K \leftarrow G$;
10) $K_0 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{Cert}(Id_A, Q_A) \,\|\, \texttt{Cert}(Id_B, Q_B) \,\|\, \texttt{hello}_A \,\|\, \texttt{hello}_B, 0)$;
11) $\big[K_1 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{Cert}(Id_A, Q_A) \,\|\, \texttt{Cert}(Id_B, Q_B) \,\|\, \texttt{hello}_A \,\|\, \texttt{hello}_B, 1)\big]$;
12) $\big[T_A \leftarrow \texttt{belt-mac}(0^{128}, K_1)\big]$;
13) отправляет сообщение M2.

4 Сторона B:

1) получает сообщение M2;
2) $\big\{$проверяет $\texttt{Cert}(Id_A, Q_A)\big\}$;
3) проверяет, что $V_A \in E^*_{a,b}(\mathbb{F}_p)$;
4) $t \leftarrow \big\langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \,\|\, \langle V_B \rangle_{2l}) \big\rangle_l$;
5) $s_B \leftarrow (u_B - (2^l + \bar{t}) d_B) \bmod q$;
6) $K \leftarrow s_B (V_A - (2^l + \bar{t}) Q_A)$;
7) если $K = O$, то $K \leftarrow G$;
8) $K_0 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{Cert}(Id_A, Q_A) \,\|\, \texttt{Cert}(Id_B, Q_B) \,\|\, \texttt{hello}_A \,\|\, \texttt{hello}_B, 0)$;
9) $\big[K_1 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{Cert}(Id_A, Q_A) \,\|\, \texttt{Cert}(Id_B, Q_B) \,\|\, \texttt{hello}_A \,\|\, \texttt{hello}_B, 1)\big]$;
10) $\big[$проверяет, что $T_A = \texttt{belt-mac}(0^{128}, K_1)\big]$;
11) $\big[T_B \leftarrow \texttt{belt-mac}(1^{128}, K_1)\big]$;
12) $\big[$отправляет сообщение M3$\big]$.

5 Сторона A:

1) $\big[$получает сообщение M3$\big]$;
2) $\big[$проверяет, что $T_B = \texttt{belt-mac}(1^{128}, K_1)\big]$.

Успешное выполнение всех шагов протокола означает, что стороны выработали общий ключ K_0. Если при выполнении протокола создавалась и обрабатывалась имитовставка T_A (T_B), то успешное завершение дополнительно означает, что сторона A (B) подтвердила свой ключ и, таким образом, прошла аутентификацию перед противоположной стороной.

7.5 Протокол BSTS

7.5.1 Сообщения

В протоколе BSTS стороны пересылают друг другу следующие сообщения:

M0 ($A \to B$): $\big\{\texttt{hello}_A\big\}$;

M1 ($B \to A$): $\big\{\texttt{hello}_B \,\|\, \big\}\ \langle V_B \rangle_{4l}$;

M2 ($A \to B$): $\langle V_A \rangle_{4l} \,\|\, Y_A \,\|\, T_A$;

M3 ($B \to A$): $Y_B \,\|\, T_B$.

7.5.2 Шаги

Формирование общего ключа состоит в выполнении шагов, определенных ниже. При ошибке на любом из шагов, в том числе при отрицательном результате любой проверки, протокол прекращается и возвращается признак `ОШИБКА`.

1 Сторона A:
   1) $\{$отправляет сообщение M0$\}$.

2 Сторона B:
   1) $\{$получает сообщение M0$\}$;
   2) $u_B \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
   3) $V_B \leftarrow u_B G$;
   4) отправляет сообщение M1.

3 Сторона A:
   1) получает сообщение M1;
   2) проверяет, что $V_B \in E^*(\mathbb{F}_p)$;
   3) $u_A \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
   4) $V_A \leftarrow u_A G$;
   5) $K \leftarrow u_A V_B$;
   6) $K_0 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 0)$;
   7) $K_1 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 1)$;
   8) $K_2 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 2)$;
   9) $t \leftarrow \big\langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \parallel \langle V_B \rangle_{2l}) \big\rangle_l$;
   10) $s_A \leftarrow (u_A - (2^l + \bar{t}) d_A) \bmod q$;
   11) $Y_A \leftarrow \texttt{belt-cfb}(\langle s_A \rangle_{2l} \parallel \texttt{Cert}(Id_A, Q_A), K_2, 0^{128})$;
   12) $T_A \leftarrow \texttt{belt-mac}(Y_A \parallel 0^{128}, K_1)$;
   13) отправляет сообщение M2.

4 Сторона B:
   1) получает сообщение M2;
   2) проверяет, что $V_A \in E^*(\mathbb{F}_p)$;
   3) $K \leftarrow u_B V_A$;
   4) $K_0 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 0)$;
   5) $K_1 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 1)$;
   6) $K_2 \leftarrow \texttt{bake-kdf}(\langle K \rangle_{2l}, \texttt{hello}_A \parallel \texttt{hello}_B, 2)$;
   7) проверяет, что $T_A = \texttt{belt-mac}(Y_A \parallel 0^{128}, K_1)$;
   8) $\langle s_A \rangle_{2l} \parallel \texttt{Cert}(Id_A, Q_A) \leftarrow \texttt{belt-cfb}^{-1}(Y_A, K_2, 0^{128})$;
   9) проверяет, что $s_A \in \{0, 1, \ldots, q-1\}$;
   10) проверяет $\texttt{Cert}(Id_A, Q_A)$;
   11) $t \leftarrow \big\langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \parallel \langle V_B \rangle_{2l}) \big\rangle_l$;
   12) проверяет, что $s_A G + (2^l + \bar{t}) Q_A = V_A$;
   13) $s_B \leftarrow (u_B - (2^l + \bar{t}) d_B) \bmod q$;
   14) $Y_B \leftarrow \texttt{belt-cfb}(\langle s_B \rangle_{2l} \parallel \texttt{Cert}(Id_B, Q_B), K_2, 1^{128})$;
   15) $T_B \leftarrow \texttt{belt-mac}(Y_B \parallel 1^{128}, K_1)$;
   16) отправляет сообщение M3.

5 Сторона A:

1) получает сообщение M3;
2) проверяет, что $T_B = \texttt{belt-mac}(Y_B \parallel 1^{128}, K_1)$;
3) $\langle s_B \rangle_{2l} \parallel \texttt{Cert}(Id_B, Q_B) \leftarrow \texttt{belt-cfb}^{-1}(Y_B, K_2, 1^{128})$;
4) проверяет, что $s_B \in \{0, 1, \ldots, q-1\}$;
5) проверяет $\texttt{Cert}(Id_B, Q_B)$;
6) проверяет, что $s_B G + (2^l + \bar{t})Q_B = V_B$.

Успешное выполнение всех шагов протокола означает, что стороны выработали общий ключ K_0 и явно подтвердили его друг другу, в том числе провели взаимную аутентификацию.

7.6 Протокол BPACE

7.6.1 Сообщения

Стороны пересылают друг другу следующие сообщения:
M0 ($A \to B$): $\{\texttt{hello}_A\}$;
M1 ($B \to A$): $\{\texttt{hello}_B \parallel \}\ Y_B$;
M2 ($A \to B$): $Y_A \parallel \langle V_A \rangle_{4l}$;
M3 ($B \to A$): $\langle V_B \rangle_{4l}\ [\parallel T_B]$;
M4 ($A \to B$): $[T_A]$.

7.6.2 Шаги протокола

Формирование общего ключа состоит в выполнении шагов, определенных ниже. При ошибке на любом из шагов, в том числе при отрицательном результате любой проверки, протокол прекращается и возвращается признак ОШИБКА.

1 Сторона A:

1) {отправляет сообщение M0}.

2 Сторона B:

1) {получает сообщение M0};
2) $R_B \xleftarrow{R} \{0,1\}^l$ (в соответствии с требованиями 5.7);
3) $K_2 \leftarrow \texttt{belt-hash}(P)$;
4) $Y_B \leftarrow \texttt{belt-ecb}(R_B, K_2)$;
5) отправляет сообщение M1.

3 Сторона A:

1) получает сообщение M1;
2) проверяет, что $|Y_B| = l$;
3) $K_2 \leftarrow \texttt{belt-hash}(P)$;
4) $R_B \leftarrow \texttt{belt-ecb}^{-1}(Y_B, K_2)$;
5) $R_A \xleftarrow{R} \{0,1\}^l$ (в соответствии с требованиями 5.7);
6) $Y_A \leftarrow \texttt{belt-ecb}(R_A, K_2)$;
7) $W \leftarrow \texttt{bake-swu}(p, a, b, R_A \parallel R_B)$;
8) $u_A \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
9) $V_A \leftarrow u_A W$;
10) отправляет сообщение M2.

4 Сторона B:

1) получает сообщение M2;
2) проверяет, что $V_A \in E^*_{a,b}(\mathbb{F}_p)$;
3) проверяет, что $|Y_A| = l$;
4) $R_A \leftarrow \texttt{belt-ecb}^{-1}(Y_A, K_2)$;
5) $W \leftarrow \texttt{bake-swu}(p, a, b, R_A \parallel R_B)$;
6) $u_B \xleftarrow{R} \{1, 2, \ldots, q-1\}$ (в соответствии с требованиями 5.7);
7) $V_B \leftarrow u_B W$;
8) $K \leftarrow u_B V_A$;
9) $K_0 \leftarrow \texttt{bake-kdf}(\langle K\rangle_{2l}, \langle V_A\rangle_{2l} \parallel \langle V_B\rangle_{2l} \parallel \texttt{hello}_A \parallel \texttt{hello}_B, 0)$;
10) $\big[K_1 \leftarrow \texttt{bake-kdf}(\langle K\rangle_{2l}, \langle V_A\rangle_{2l} \parallel \langle V_B\rangle_{2l} \parallel \texttt{hello}_A \parallel \texttt{hello}_B, 1)\big]$;
11) $\big[T_B \leftarrow \texttt{belt-mac}(1^{128}, K_1)\big]$;
12) отправляет сообщение M3.

5 Сторона A:

1) получает сообщение M3;
2) проверяет, что $V_B \in E^*_{a,b}(\mathbb{F}_p)$;
3) $K \leftarrow u_A V_B$;
4) $K_0 \leftarrow \texttt{bake-kdf}(\langle K\rangle_{2l}, \langle V_A\rangle_{2l} \parallel \langle V_B\rangle_{2l} \parallel \texttt{hello}_A \parallel \texttt{hello}_B, 0)$;
5) $\big[K_1 \leftarrow \texttt{bake-kdf}(\langle K\rangle_{2l}, \langle V_A\rangle_{2l} \parallel \langle V_B\rangle_{2l} \parallel \texttt{hello}_A \parallel \texttt{hello}_B, 1)\big]$;
6) $\big[$проверяет, что $T_B = \texttt{belt-mac}(1^{128}, K_1)\big]$;
7) $\big[T_A \leftarrow \texttt{belt-mac}(0^{128}, K_1)\big]$;
8) $\big[$отправляет сообщение M4$\big]$.

6 Сторона B:

1) $\big[$получает сообщение M4$\big]$;
2) $\big[$проверяет, что $T_A = \texttt{belt-mac}(0^{128}, K_1)\big]$.

Успешное выполнение всех шагов протокола означает, что стороны выработали общий ключ K_0. Если при выполнении протокола создавалась и обрабатывалась имитовставка T_A (T_B), то успешное завершение дополнительно означает, что сторона A (B) подтвердила свой ключ и, таким образом, прошла аутентификацию перед другой стороной.

Приложение А
(справочное)
Протокол Диффи — Хеллмана

Протокол Диффи – Хеллмана, введенный в работе [8], неявно используется в протоколах BMQV, BSTS, BPACE и является базовым для них.

Редакция протокола Диффи — Хеллмана, соответствующая соглашениям настоящего стандарта, имеет следующий вид:

1 Стороны A и B выбирают параметры эллиптической кривой p, a, b, q и G, которые удовлетворяют требованиям, определенным в 5.3.

2 Сторона A генерирует одноразовый личный ключ u_A и определяет соответствующий открытый ключ $V_A = u_A G$. Аналогично, сторона B генерирует одноразовый личный ключ u_B и определяет открытый ключ $V_B = u_B G$. Личные ключи должны генерироваться в соответствии с требованиями, определенными в 5.7.

3 Стороны обмениваются открытыми ключами V_A, V_B. Каждая из сторон проверяет, что полученный ею открытый ключ является элементом $E^*_{a,b}(\mathbb{F}_p)$, и завершает протокол с ошибкой, если проверяемое условие нарушается. Для контроля V_A, V_B может использоваться алгоритм проверки открытых ключей, определенный в СТБ 34.101.45 (пункт 6.2.3).

4 Сторона A определяет точку $K = u_A V_B$, а сторона B — точку $K = u_B V_A$. Точки сторон совпадают: $u_A V_B = u_A u_B G = u_B V_A$.

5 По общей точке K стороны строят общий секретный ключ, используя, например, алгоритм 6.1.3.

Пары ключей (u_A, V_A), (u_B, V_B) могут быть как одноразовыми (их еще называют эфемерными, ephemeral), так и долговременными (статическими, static). Соответственно может быть три варианта протокола Диффи — Хеллмана: с одноразовыми ключами (ephemeral-ephemeral), с долговременными ключами (static-static), с долговременными и одноразовыми ключами (static-ephemeral).

На долговременные и одноразовые ключи протокола Диффи — Хеллмана должны распространяться требования, заданные в 5.4, 5.7.

В сеансах протокола с повторяющимися долговременными ключами общая точка K будет также повторяться. Для защиты от повторов общего ключа стороны должны использовать при его построении дополнительные уникальные (открытые или секретные) данные.

Злоумышленник, который не вступает во взаимодействие со сторонами, а только перехватывает их сообщения, получает в свое распоряжение параметры эллиптической кривой и открытые ключи $u_A G$ и $u_B G$. Злоумышленнику требуется по этим данным определить общую секретную точку $u_A u_B G$. Такая задача называется вычислительной задачей Диффи —Хеллмана и считается трудной, если выбраны надежные параметры эллиптической кривой (см. [2, пункт 4.1.5]).

С другой стороны, протокол Диффи — Хеллмана не защищает от атак злоумышленника «посередине», который выполняет протокол с каждой из сторон по-отдельности, выдавая себя за A стороне B и за B стороне A. Злоумышленник формирует общие ключи с каждой из сторон, а затем полностью контролирует обмен сообщений между ними, перешифровывая данные или пересчитывая имитовставки.

Для защиты от злоумышленника «посередине» протокол Диффи – Хеллмана должен сопровождаться дополнительными механизмами безопасности. Фактически протокол Диффи — Хеллмана должен являться составной частью некоторого высокоуровневого протокола, который эти механизмы поддерживает.

Могут использоваться следующие два механизма, обеспечивающие надежное применение описанной выше редакции протокола Диффи — Хеллмана.

Первый механизм состоит в контроле целостности и подлинности открытых ключей при их передаче между сторонами. Для организации контроля долговременные открытые ключи протокола могут передаваться в виде сертификатов. Могут также использоваться долговременные ключи ЭЦП, с помощью которых стороны вырабатывают и проверяют подписи данных обмена. (Именно такой подход использован в протоколе BSTS). Наконец для организации контроля могут использоваться долговременные секретные ключи имитозащиты, с помощью которых стороны вырабатывают и проверяют имитовставки данных обмена.

Второй механизм состоит в использовании дополнительных секретных данных при построении общего ключа. Имеется в виду, что на вход алгоритма построения ключа кроме общей секретной точки K подаются секретные ключи, предварительно распределенные или полученные в результате выполнения других протоколов.

Приложение Б
(справочное)
Проверочные примеры

Б.1 Параметры эллиптической кривой

Во всех примерах используются параметры эллиптической кривой, заданные в СТБ 34.101.45 (таблица Б.1).

Б.2 Идентификаторы и долговременные ключи

В таблице Б.1 представлены идентификаторы и долговременные ключи сторон протоколов BMQV и BSTS. Сертификаты сторон представляют собой объединение их идентификаторов и открытых ключей: $\mathtt{Cert}(Id, Q) = Id \parallel \langle Q \rangle$.

Таблица Б.1 — Идентификаторы и долговременные ключи

Id_A	416C6963 65_{16}
$\langle d_A \rangle_{256}$	1F66B5B8 4B733967 4533F032 9C74F218 34281FED 0732429E 0C79235F C273E269$_{16}$
$\langle Q_A \rangle_{512}$	BD1A5650 179D79E0 3FCEE49D 4C2BD5DD F54CE46D 0CF11E4F F87BF7A8 90857FD0 7AC6A603 61E8C817 3491686D 461B2826 190C2EDA 5909054A 9AB84D2A B9D99A90$_{16}$
Id_B	426F62$_{16}$
$\langle d_B \rangle_{256}$	4C0E74B2 CD5811AD 21F23DE7 E0FA742C 3ED6EC48 3C461CE1 5C33A77A A308B7D2$_{16}$
$\langle Q_B \rangle_{512}$	CCEEF1A3 13A40664 9D15DA0A 851D486A 695B641B 20611776 252FFDCE 39C71060 7C9EA1F3 3C23D20D FCB8485A 88BE6523 A28ECC32 15B47FA2 89D6C9BE 1CE837C0$_{16}$

Б.3 Приветственные сообщения

Во всех примерах используются пустые приветственные сообщения $\mathtt{hello}_A$, $\mathtt{hello}_B$.

Б.4 Протокол BMQV

В таблице Б.2 представлен сеанс протокола BMQV.

Таблица Б.2 — Сеанс протокола BMQV

$\langle u_B \rangle_{256}$	0F51D913 47617C20 BD4AB07A EF4F26A1 AD1362A8 F9A3D42F BE1B8E6F $1C88AAD5_{16}$
$\langle V_B \rangle_{512}$	9B4EA669 DABDF100 A7D4B6E6 EB76EE52 51912531 F426750A AC8A9DBB 51C54D8D
	6AB7DBF1 5FCBD768 EE68A173 F7B236EF C15A01E2 AA6CD1FE 98B947DA $7B38A2A0_{16}$
$\langle u_A \rangle_{256}$	0A4E8298 BE0839E4 6F19409F 637F4415 572251DD 0D39284F 0F0390D9 $3BBCE9EC_{16}$
$\langle V_A \rangle_{512}$	1D5A382B 962D4ED0 6193258C A6DE535D 8FD7FACB 853171E9 32EF93B5 EE800120
	03DBB7B5 BD070363 80BAFA47 FCA7E6CA 3F179EDD D1AE5086 64790918 $3628EDDC_{16}$
t	BD46F58A DE7C4DF9 826D32AB $A9113428_{16}$
$\langle s_A \rangle_{256}$	AB4EB3A6 D867C861 52E61B64 7F1A32D9 93A7768F 79361F75 0AE7C7A6 $5CD9A233_{16}$
$\langle K \rangle_{512}$	7FF3A0DA CDFECB3C D25F4D3C 334CCCB3 34C71FF7 1E2247DD 0688FA62 DF4C5920
	728CB855 98DA04B4 8D85D32D 0CDCCD92 3D88E844 9BAA5065 B4E4D1CB $EEE31D35_{16}$
K_0	C6F86D0E 468D5EF1 A9955B2E E0CF0581 050C81D1 B4772709 2408E863 $C7EEB48C_{16}$
K_1	E95BA3F6 45C58288 E8A1B37C 10ADD336 DB8BD7F6 75F94963 139769F2 $E260C6A9_{16}$
T_A	413B7E18 $1BAFB337_{16}$
$\langle s_B \rangle_{256}$	B6099633 2B62DDB1 354EC03D A949B528 969E6CA6 D8848C94 013B9CF6 $FF42AEED_{16}$
T_B	B800A203 $3AC7591B_{16}$

Б.5 Протокол BSTS

В таблице Б.3 представлен сеанс протокола BSTS.

Таблица Б.3 — Сеанс протокола BSTS

$\langle u_B \rangle_{256}$	0F51D913 47617C20 BD4AB07A EF4F26A1 AD1362A8 F9A3D42F BE1B8E6F 1C88AAD5$_{16}$
$\langle V_B \rangle_{512}$	9B4EA669 DABDF100 A7D4B6E6 EB76EE52 51912531 F426750A AC8A9DBB 51C54D8D 6AB7DBF1 5FCBD768 EE68A173 F7B236EF C15A01E2 AA6CD1FE 98B947DA 7B38A2A0$_{16}$
$\langle u_A \rangle_{256}$	0A4E8298 BE0839E4 6F19409F 637F4415 572251DD 0D39284F 0F0390D9 3BBCE9EC$_{16}$
$\langle V_A \rangle_{512}$	1D5A382B 962D4ED0 6193258C A6DE535D 8FD7FACB 853171E9 32EF93B5 EE800120 03DBB7B5 BD070363 80BAFA47 FCA7E6CA 3F179EDD D1AE5086 64790918 3628EDDC$_{16}$
$\langle K \rangle_{512}$	C9121850 4B2F10C8 B307B3F8 5A292930 8E48F334 51D2810A AD788DE8 CA4C7347 76932167 30B95FD3 C1439D6C B99A1A0B 2898FC56 3558C8F5 18E235B9 D7441A6E$_{16}$
K_0	78EF2C56 BD6DA211 6BB5BEE8 0CEE5C05 394E7609 183CF7F7 6DF0C2DC FB25C4AD$_{16}$
K_1	F02580E9 5C1E89BD 9E743C02 716E3E31 FA429298 AE0FD1FE 2BBA1B57 02E51B9D$_{16}$
K_2	41283622 4A0C0964 1F3C3B88 8C7804FA 32B94A62 B5CB0066 518409F9 69191776$_{16}$
t	BD46F58A DE7C4DF9 826D32AB A9113428$_{16}$
$\langle s_A \rangle_{256}$	AB4EB3A6 D867C861 52E61B64 7F1A32D9 93A7768F 79361F75 0AE7C7A6 5CD9A233$_{16}$
Y_A	A994115F 297D2FAD 342A0AF5 4FCDA66E 1E6A30FE 966662C4 3C2A73AF A3CADF69 47344287 CB200795 61645867 8B76BA61 924AD05D 80BB81F5 3F8D5C4E 0EF55EBD AFA674D7 ECD74CB0 609DE12B C0463670 64059F01 1607DD18 62407490 1F1C5A40 94C00655 9F$_{16}$
T_A	1306D682 00087987$_{16}$
$\langle s_B \rangle_{256}$	B6099633 2B62DDB1 354EC03D A949B528 969E6CA6 D8848C94 013B9CF6 FF42AEED$_{16}$
Y_B	6D45B2E7 6AF24422 ADC6D5D7 A3CFA37F DCB52F7E 440222F1 AACECB98 BDED357B BD459DF0 A3EE7A3E AFE0199C A5C4C072 7C33909E 4C322216 F6F53E38 3A3727D8 34B5D4F5 C977FC3B 7EBA6DCA 55C0F1A5 69BE3CD3 464B13C3 88D0DAC3 E6A82F9D 2EF3D6$_{16}$
T_B	CA7A5BAC 4EB2910E$_{16}$

Б.6 Протокол BPACE

В таблице Б.4 представлен сеанс протокола BPACE.

Таблица Б.4 — Сеанс протокола BPACE

P	38303836_{16}
R_B	$0F51D913\ 47617C20\ BD4AB07A\ EF4F26A1_{16}$
K_2	$3292E21E\ 6CD50D27\ 2532713B\ A52570A4\ C996319E\ 2436B385\ 7DB0ACB4\ 5660F4EB_{16}$
Y_B	$991E8169\ 0B4C687C\ 86BFD11C\ EBDA2421_{16}$
R_A	$AD1362A8\ F9A3D42F\ BE1B8E6F\ 1C88AAD5_{16}$
Y_A	$CE41B54D\ C13A28BD\ F74CEBD1\ 90881802_{16}$
$\langle W \rangle_{512}$	$014417D3\ 35555731\ 7D2E2AB6\ D0875487\ 8D19E8D9\ 7B71FDC9\ 5DBB2A9B\ 894D16D7$ $7704A0B5\ CAA9CDA1\ 0791E476\ 0671E105\ 0DDEAB70\ 83A74584\ 47866ADB\ 01473810_{16}$
$\langle u_A \rangle_{256}$	$0A4E8298\ BE0839E4\ 6F19409F\ 637F4415\ 572251DD\ 0D39284F\ 0F0390D9\ 3BBCE9EC_{16}$
$\langle V_A \rangle_{512}$	$6B13ACBB\ 086FB876\ 18BCC2EF\ 20A3FA89\ 475654CB\ 367E670A\ 2441730B\ 24B8AB31$ $8209C81C\ 9640C47A\ 77B28E90\ AB9211A1\ DF21DE87\ 8191C314\ 061E347C\ 5125244F_{16}$
$\langle u_B \rangle_{256}$	$F81B29D5\ 71F6452F\ F8B2B97F\ 57E18A58\ BC946FEE\ 45EAB32B\ 06FCAC23\ A33F422B_{16}$
$\langle V_B \rangle_{512}$	$CD3D6487\ DC4EEB23\ 45697818\ 6A069C71\ 375D75C2\ DF198BAD\ 1E61EEA0\ DBBFF737$ $3D1D9ED1\ 7A7AD460\ AA420FB1\ 1952D580\ 78BC1CC9\ F408F2E2\ 58FDE97F\ 22A44C6F_{16}$
$\langle K \rangle_{512}$	$723356E3\ 35ED7062\ 0FFB1842\ 752092C3\ 2603EB66\ 60409205\ 87D80057\ 5BECFC42$ $0C4B4C9B\ 4AEB51D3\ 6FE2EDEB\ 1369CE39\ 676CE544\ 0E29916C\ 97FBA4F3\ ED6A31BD_{16}$
K_0	$DAC4D8F4\ 11F9C523\ D28BBAAB\ 32A5270E\ 4DFA1F0F\ 757EF8E0\ F30AF08F\ BDE1E7F4_{16}$
K_1	$54AC0582\ 84D679CF\ 4C47D3D7\ 2651F3E4\ EF0D61D1\ D0ED5BAF\ 8FF30B89\ 24E599D8_{16}$
T_B	$28FD4859\ D78BA971_{16}$
T_A	$5D93FD9A\ 7CB863AA_{16}$

Приложение В
(рекомендуемое)
Модуль ACH.1

В.1 Идентификаторы

Алгоритмам и протоколам стандарта присваиваются следующие идентификаторы:

`bake-bmqv`	протокол BMQV (7.4);
`bake-bsts`	протокол BSTS (7.5);
`bake-bpace`	протокол BPACE (7.6);
`bake-dh`	алгоритм определения общей точки по личному ключу одной стороны и открытому ключу другой стороны, часть протокола Диффи — Хеллмана (приложение А);
`bake-kdf`	алгоритм построения ключа (6.1.3);
`bake-swu`	алгоритм построения точки эллиптической кривой (6.2.3).

Уровень стойкости протоколов ключа не указывается в их идентификаторах и определяется по размерностям параметров используемой эллиптической кривой.

Долговременному открытому ключу, который используется в протоколах BMQV и BSTS, присваивается идентификатор `bake-pubkey`.

В.2 Описание долговременного открытого ключа

На уровне стойкости l долговременному открытому ключу Q ставится в соответствие двоичное слово $\langle Q \rangle_{4l}$, для описания которого может использоваться тип `PublicKey`, определенный в СТБ 34.101.45 (приложение Д).

В запросе на получение сертификата (СТБ 34.101.17), в сертификатах (СТБ 34.101.19) открытый ключ должен представляться значениями типа `SubjectPublicKeyInfo`. Этот тип также определен в СТБ 34.101.45 (приложение Д).

В.3 Модуль ACH.1

```
Bake-module-v1 {iso(1) member-body(2) by(112) 0 2 0 34 101 66 module(1) ver1(1)}
DEFINITIONS ::=
BEGIN
  bake OBJECT IDENTIFIER ::= {iso(1) member-body(2) by(112) 0 2 0 34 101 66}

  bake-bmqv OBJECT IDENTIFIER ::= {bake 11}
  bake-bsts OBJECT IDENTIFIER ::= {bake 12}
  bake-bpace OBJECT IDENTIFIER ::= {bake 21}
  bake-dh OBJECT IDENTIFIER ::= {bake 31}
  bake-kdf OBJECT IDENTIFIER ::= {bake 101}
  bake-swu OBJECT IDENTIFIER ::= {bake 201}

  bake-keys OBJECT IDENTIFIER ::= {bake keys(2)}
```

```
  bake-pubkey OBJECT IDENTIFIER ::= {bake-keys 1}
END
```

Библиография

[1] Лидл Р., Нидеррайтер Г. Конечные поля
М.: Мир, 1988

[2] Hankerson D., Menezes A., Vanstone S. Guide to Elliptic Curve Cryptography
N. Y.: Springer, 2004
(Введение в криптографию на эллиптических кривых)

[3] Law L., Menezes A., Qu M., Solinas J., Vanstone S. An Efficient Protocol for Authenticated Key Agreement
Designs, Codes and Cryptography, 28(2), 119-134, 2003
(Эффективный протокол формирования общего ключа и аутентификации)

[4] Diffie W., Oorschot P., Wiener M. Authentication and Authenticated Key Exchanges
Designs, Codes and Cryptography, 2(2): 107–125, 1992
(Аутентификация и обмен ключами с аутентификацией)

[5] Bender J., Fischlin M., Kuegler D. Security Analysis of the PACE Key-Agreement Protocol
Cryptology ePrint Archive, Report 2009/624, 2009
(Оценка надежности протокола формирования общего ключа PACE)

[6] Brier E., Coron J., Icart T., Madore D., Randriam H., Tibouch M. Efficient Indifferentiable Hashing into Ordinary Elliptic Curves
Advances in Cryptology — CRYPTO 2010, Lecture Notes in Computer Science, 6223: 237–254, Springer-Verlag, 2010
(Эфективное неразличимое хэширование на обычные эллиптические кривые)

[7] ISO/IEC 10646:2012 Information technology – Universal Coded Character Set (UCS)
International Organization for Standardization, 2012
(Информационные технологии. Универсальный набор кодированных символов (UCS))

[8] Diffie W., Hellman M. New Directions in Cryptography
IEEE Transactions on Information Theory, IT-22, 644–654, 1976
(Новые направления в криптографии)

Поправка к официальной редакции

В каком месте	Напечатано	Должно быть
Подраздел 7.2	**Алгоритм** $\texttt{belt-cfb}^{-1}$. В протоколе BPACE используется алгоритм расшифрования...	**Алгоритм** $\texttt{belt-cfb}^{-1}$. В протоколе BSTS используется алгоритм расшифрования...
Пункт 7.4.2, шаг 4.4	$t \leftarrow \langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \,\|\| \, \langle V_B \rangle_{2l}) \rangle$;	$t \leftarrow \langle \texttt{belt-hash}(\langle V_A \rangle_{2l} \,\|\| \, \langle V_B \rangle_{2l}) \rangle_l$;
Пункт 7.5.2, шаг 4.7	проверяет, что $T_A \leftarrow \texttt{belt-mac}(Y_A \,\|\| \, 0^{128}, K_1)$;	проверяет, что $T_A = \texttt{belt-mac}(Y_A \,\|\| \, 0^{128}, K_1)$;
Приложение А, предпоследний абзац	Могут также использоваться долговременые ключи ЭЦП, с помощью которых стороны вырабатывают и проверяют подписи данных обмена.	Могут также использоваться долговременные ключи ЭЦП, с помощью которых стороны вырабатывают и проверяют подписи данных обмена.