ГОСУДАРСТВЕННЫЙ СТАНДАРТ
РЕСПУБЛИКИ БЕЛАРУСЬ

СТБ 34.101.47-2017

Информационные технологии и безопасность

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Інфармацыйныя тэхналогіі і бяспека

КРЫПТАГРАФІЧНЫЯ АЛГАРЫТМЫ ГЕНЕРАЦЫІ ПСЕЎДАВЫПАДКОВЫХ ЛІКАЎ

Госстандарт
Минск

УДК 004.056.55(083.74)(476) МКС 35.240.40 КП 05

Ключевые слова: криптографический алгоритм, псевдослучайные числа, ключ, синхропосылка, хэширование, одноразовый пароль

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 закрытым акционерным обществом «АВЕСТ» и 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 «31» января 2017 г. № 9

3 ВЗАМЕН СТБ 34.101.47-2012

Издан на русском языке

Содержание

ГОСУДАРСТВЕННЫЙ СТАНДАРТ РЕСПУБЛИКИ БЕЛАРУСЬ

Информационные технологии и безопасность

КРИПТОГРАФИЧЕСКИЕ АЛГОРИТМЫ ГЕНЕРАЦИИ ПСЕВДОСЛУЧАЙНЫХ ЧИСЕЛ

Інфармацыйныя тэхналогіі і бяспека

КРЫПТАГРАФІЧНЫЯ АЛГАРЫТМЫ ГЕНЕРАЦЫІ ПСЕЎДАВЫПАДКОВЫХ ЛІКАЎ

Information technology and security

Cryptographic algorithms of pseudorandom number generation

Дата введения 2017-09-01

1 Область применения

Настоящий стандарт устанавливает криптографические алгоритмы генерации псевдослучайных чисел. Алгоритмы стандарта могут применяться для построения ключей, синхропосылок, одноразовых паролей, других непредсказуемых или уникальных параметров криптографических алгоритмов и протоколов.

Настоящий стандарт применяется при разработке, испытаниях и эксплуатации средств криптографической защиты информации.

2 Нормативные ссылки

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

СТБ 1176.1-99 Информационная технология. Защита информации. Функция хэширования

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

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

СТБ 34.101.65-2014 Информационные технологии и безопасность. Протокол защиты транспортного уровня (TLS)

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

ГОСТ 27463-87 Системы обработки информации. 7-битные кодированные наборы

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

3 Термины и определения

В настоящем стандарте применяются следующие термины с соответствующими определениями:

3.1 имитовставка: Двоичное слово, которое определяется по сообщению с использованием ключа и служит для контроля целостности и подлинности сообщения.

3.2 ключ: Параметр, который управляет криптографическими операциями зашифрования и расшифрования, выработки и проверки электронной цифровой подписи, генерации псевдослучайных чисел и др.

3.3 одноразовый пароль: Пароль, действие которого ограничено сеансом аутентификации или промежутком времени.

3.4 октет: Двоичное слово длины 8.

3.5 пароль: Секрет, который способен запомнить (обработать) человек и который поэтому может принимать сравнительно небольшое число значений.

3.6 псевдослучайные числа: Последовательность элементов, полученная в результате выполнения некоторого алгоритма и используемая в конкретном случае вместо последовательности случайных чисел.

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

3.8 случайные числа: Последовательность элементов, каждый из которых не может быть предсказан (вычислен) только на основе знания предшествующих ему элементов данной последовательности.

3.9 сообщение: Двоичное слово конечной длины.

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

3.11 хэширование: Выработка хэш-значений.

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);
$\{0,1\}^{n*}$	множество всех слов из $\{0,1\}^*$, длина которых кратна n;
$\lvert u\rvert$	длина слова $u \in \{0,1\}^*$;
α^n	слово длины n из одинаковых символов $\alpha \in \{0,1\}$;
$u \parallel v$	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, $10100010 = \texttt{A2}_{16}$);

"01234"	строка графических символов базовой таблицы КОИ-7, представляющая слово $u \in \{0,1\}^{8*}$ так, что каждому символу строки соответствует один октет u (например, "A2" = 0100000100110010);
$x \bmod m$	для целого x и натурального m остаток от деления x на m, т. е. число $r \in \{0,1,\dots,m-1\}$ такое, что m делит $x-r$;
$u \oplus v$	для $u = u_1u_2\dots u_n \in \{0,1\}^n$ и $v = v_1v_2\dots v_n \in \{0,1\}^n$ слово $w = w_1w_2\dots w_n \in \{0,1\}^n$ из символов $w_i = (u_i + v_i) \bmod 2$;
$\bar{u}$	а) для $u = u_1u_2\dots u_8 \in \{0,1\}^8$ число $2^7u_1 + 2^6u_2 + \dots + u_8$ 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 U слово $u \in \{0,1\}^{8n}$ такое, что $\bar{u} = U \bmod 2^{8n}$;
$\lfloor z \rfloor$	для вещественного z максимальное целое, не превосходящее z;
$u \boxplus v$	для $u, v \in \{0,1\}^{8n}$ слово $\langle \bar{u} + \bar{v} \rangle_{8n}$;
$c \leftarrow u$	присвоение переменной c значения u;
hmac$[h]$	определенный в 6.1 алгоритм выработки имитовставки на основе допустимой функции хэширования h.

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 — Шестнадцатеричные символы

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:

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

Некоторые октеты могут кодироваться графическими символами базовой таблицы КОИ-7, определенной в ГОСТ 27463. Правила кодирования заданы в таблице 2. В таблице используется шестнадцатеричное представление слов $u \in \{0,1\}^8$. Если $u = IJ_{16}$, то символ, представляющий u, находится на пересечении строки I и столбца J. Октет 20_{16} кодируется пробелом, октет $7F_{16}$ не кодируется графическим символом.

Таблица 2 — Графические символы базовой таблицы КОИ-7

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2		!	"	#	$	%	&	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	

4.2.2 Слова как числа

Октету $u = u_1 u_2 \dots u_8$ ставится в соответствие байт — число $\bar{u} = 2^7 u_1 + 2^6 u_2 + \dots + u_8$. 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

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

5 Общие положения

5.1 Назначение

Настоящий стандарт определяет криптографические алгоритмы генерации псевдослучайных чисел. В алгоритмах используются ключ и синхропосылка. При соблюдении секретности ключа и уникальности синхропосылки генерируемые числа трудно предугадать или повторить, и поэтому их можно использовать для построения непредсказуемых и уникальных параметров криптографических алгоритмов и протоколов, в том числе других ключей и синхропосылок.

Ключ алгоритма генерации может быть известен одной или нескольким сторонам. В первом случае псевдослучайные числа можно использовать для построения секретных параметров владельца ключа. Во втором случае стороны могут использовать алгоритм для построения общих секретных параметров.

В 6.1 определяется алгоритм выработки имитовставки в режиме HMAC (Hash-based Message Authentication Code), соответствующий [1]. Этот алгоритм используется как вспомогательный при генерации псевдослучайных чисел в 6.3. Кроме этого, алгоритм имеет самостоятельное значение и может применяться непосредственно для выработки имитовставок. Если на вход алгоритма подавать неповторяющиеся синхропосылки, например отметки текущего времени, то выходные имитовставки можно использовать в качестве псевдослучайных чисел.

В 6.2 определяется алгоритм генерации псевдослучайных чисел в режиме счетчика. В этом алгоритме ключ и синхропосылка являются словами фиксированной длины. На шагах алгоритма используются дополнительные входные данные, которые можно выбирать случайным или псевдослучайным образом. Дополнительные входные данные увеличивают неопределенность выходных.

В 6.3 определяется алгоритм генерации псевдослучайных чисел в режиме HMAC. В этом алгоритме ключ и синхропосылка являются словами произвольной длины, дополнительные входные данные не используются. Алгоритм используется в СТБ 34.101.65.

В приложении А определяются алгоритмы генерации одноразовых паролей, а также правила проверки этих паролей. Одноразовые пароли строятся как псевдослучайные числа по имитовставкам HMAC. Алгоритмы и правила приложения соответствуют спецификациям [2] — [4]. Рекомендуется использовать приложение для усиления аутентификации в клиент-серверных системах.

В приложении Б приводится модуль абстрактно-синтаксической нотации версии 1 (АСН.1), определенной в ГОСТ 34.973. Модуль задает идентификаторы алгоритмов и описывает особенности использования в алгоритмах функций хэширования. Рекомендуется использовать модуль при встраивании алгоритмов стандарта в информационные системы, в которых также используется АСН.1.

В приложении В приводятся примеры выполнения алгоритмов стандарта. Примеры можно использовать для проверки корректности реализаций алгоритмов.

5.2 Функция хэширования

В алгоритмах настоящего стандарта используется функция хэширования h, которая ставит в соответствие сообщению $X \in \{0,1\}^*$ его хэш-значение $h(X) \in \{0,1\}^{2l}$.

Функция h должна быть алгоритмически определена в некотором ТНПА. Например в качестве h может использоваться функция, заданная в СТБ 1176.1 или СТБ 34.101.31 (пункт 6.9). Для обеих этих функций $l = 128$.

В алгоритмах режимов HMAC требуется, чтобы h была блочно-итерационной. Это значит, что хэширование состоит в итерационной обработке последовательных блоков X, причем блоки являются двоичными словами одной и той же длины b. Дополнительно требуется, чтобы длина блока b была кратна 8 и не была меньше $2l$. Алгоритм выработки имитовставки в режиме HMAC на основе допустимой функции хэширования h обозначается через $\texttt{hmac}[h]$.

Функции, заданные в СТБ 1176.1 и СТБ 34.101.31, удовлетворяют ограничениям HMAC. Для них $b = 256$.

5.3 Ключ

Ключ, который используется при генерации псевдослучайных чисел, должен вырабатываться без возможности предсказания, распространяться с соблюдением мер конфиденциальности и храниться в секрете.

Один и тот же ключ не должен использоваться в различных алгоритмах настоящего стандарта.

Ключом является двоичное слово фиксированной (для алгоритма из 6.2) или произвольной (для алгоритмов из 6.1, 6.3, приложения А) длины. В режимах HMAC рекомендуется применять ключ, длина которого совпадает с длиной значений используемой функции хэширования h.

Ключ можно генерировать псевдослучайным методом с помощью одного из алгоритмов стандарта. При генерации должен использоваться другой ключ, который также может быть построен псевдослучайным методом еще на одном ключе и т. д. Для построения первоначальных ключей должны применяться генераторы случайных чисел. В приложении Г даются справочные сведения по использованию таких генераторов.

5.4 Синхропосылка

Синхропосылка, которая используется при генерации псевдослучайных чисел, должна быть уникальной. Уникальность означает, что при многократном применении алгоритма генерации с одним и тем же ключом вероятность совпадения используемых синхропосылок пренебрежимо мала.

Синхропосылка не является секретным параметром, может быть общедоступной.

Синхропосылки можно вырабатывать случайным или псевдослучайным методом, строить по отметкам времени, значениям монотонного счетчика, неповторяющимся номерам сообщений и др.

6 Криптографические алгоритмы генерации псевдослучайных чисел

6.1 Выработка имитовставки в режиме HMAC

6.1.1 Функция хэширования

Используется блочно-итерационная функция хэширования h с длиной блока b кратной восьми и значениями длиной $2l \leqslant b$.

6.1.2 Входные и выходные данные

Входными данными алгоритма выработки имитовставки являются ключ $K \in \{0,1\}^*$ и сообщение $X \in \{0,1\}^*$.

Выходными данными является слово $Y \in \{0,1\}^{2l}$ — имитовставка сообщения X на ключе K.

6.1.3 Константы и переменные

Слова $\mathtt{ipad}$, $\mathtt{opad}$. Используются фиксированные слова $\mathtt{ipad}, \mathtt{opad} \in \{0,1\}^b$, составленные из повторенных $b/8$ раз октетов $\mathtt{36}_{16}$ и $\mathtt{5C}_{16}$: $\mathtt{ipad} = \mathtt{36}_{16} \| \mathtt{36}_{16} \| \ldots \| \mathtt{36}_{16}$, $\mathtt{opad} = \mathtt{5C}_{16} \| \mathtt{5C}_{16} \| \ldots \| \mathtt{5C}_{16}$.

Переменная t. Используется переменная $t \in \{0,1\}^b$. Значение t должно быть уничтожено после использования.

6.1.4 Алгоритм

Вычисление имитовставки сообщения X на ключе K состоит в выполнении следующих шагов:

1 Если $|K| \leqslant b$, то $t \leftarrow K \| 0^{b-|K|}$, иначе $t \leftarrow h(K) \| 0^{b-2l}$.

2 $Y \leftarrow h((t \oplus \mathtt{ipad}) \| X)$.

3 $Y \leftarrow h((t \oplus \mathtt{opad}) \| Y)$.

4 Возвратить Y.

6.2 Генерация псевдослучайных чисел в режиме счетчика

6.2.1 Функция хэширования

Используется функция хэширования h со значениями длиной $2l$.

6.2.2 Входные и выходные данные

Входными данными алгоритма генерации псевдослучайных чисел являются натуральное n, ключ $K \in \{0,1\}^{2l}$ и синхропосылка $S \in \{0,1\}^{2l}$. Число n определяет количество генерируемых псевдослучайных чисел.

Используется дополнительное входное слово $X \in \{0,1\}^{2ln}$. Слово X записывается в виде $X = X_1 \| X_2 \| \ldots \| X_n$, где $X_i \in \{0,1\}^{2l}$ — дополнительные данные, которые используются на i-й итерации алгоритма. Слова X_i могут выбираться произвольным образом, в том числе случайным или псевдослучайным методом. По умолчанию $X_i = 0^{2l}$.

Выходными данными алгоритма является слово $Y \in \{0,1\}^{2ln}$ — псевдослучайные числа, полученные на ключе K при использовании синхропосылки S и дополнительных данных X. Слово Y записывается в виде $Y = Y_1 \| Y_2 \| \ldots \| Y_n$, где $Y_i \in \{0,1\}^{2l}$.

6.2.3 Переменные

Используются переменные $s, r \in \{0,1\}^{2l}$. Значение r должно быть уничтожено после использования.

6.2.4 Алгоритм

Генерация псевдослучайных чисел состоит в выполнении следующих шагов:

1 $s \leftarrow S$.
2 $r \leftarrow S \oplus 1^{2l}$.
3 Для $i = 1, 2, \ldots, n$ выполнить:
  1) $Y_i \leftarrow h(K \| s \| X_i \| r)$;
  2) $s \leftarrow s \boxplus \langle 1 \rangle_{2l}$;
  3) $r \leftarrow r \oplus Y_i$.
4 $Y \leftarrow Y_1 \| Y_2 \| \ldots \| Y_n$.
5 Возвратить Y.

6.3 Генерация псевдослучайных чисел в режиме HMAC

6.3.1 Функция хэширования

Используется функция хэширования h, удовлетворяющая ограничениям HMAC (пункт 5.2). Функция h применяется косвенно — как композиционный элемент алгоритма $\texttt{hmac}[h]$.

6.3.2 Входные и выходные данные

Входными данными алгоритма генерации псевдослучайных чисел являются натуральное n, ключ $K \in \{0,1\}^*$ и синхропосылка $S \in \{0,1\}^*$. Число n определяет количество генерируемых псевдослучайных чисел.

Выходными данными алгоритма является слово $Y \in \{0,1\}^{2ln}$ — псевдослучайные числа, полученные на ключе K при использовании синхропосылки S. Слово Y записывается в виде $Y = Y_1 \| Y_2 \| \ldots \| Y_n$, где $Y_i \in \{0,1\}^{2l}$.

6.3.3 Переменные

Используется переменная $r \in \{0,1\}^{2l}$.

6.3.4 Алгоритм

Генерация псевдослучайных чисел состоит в выполнении следующих шагов:

1 $r \leftarrow \texttt{hmac}[h](K, S)$.

2 Для $i = 1, 2, \ldots, n$ выполнить:

   1) $Y_i \leftarrow \texttt{hmac}[h](K, r \parallel S)$;

   2) $r \leftarrow \texttt{hmac}[h](K, r)$.

3 $Y \leftarrow Y_1 \parallel Y_2 \parallel \ldots \parallel Y_n$.

4 Возвратить Y.

Приложение А
(рекомендуемое)
Одноразовые пароли

А.1 Назначение

Одноразовые пароли предназначены для усиления аутентификации в клиент-серверных системах: кроме обычного долговременного (статического) пароля клиент предъявляет серверу дополнительный пароль, срок действия которого ограничен определенным сеансом аутентификации или промежутком времени. Даже если противник узнает пароль текущего сеанса или промежутка, он не сможет использовать его в следующем.

Аутентификация может быть двусторонней: после успешной аутентификации клиента сервер генерирует новый одноразовый пароль и предъявляет его клиенту.

А.2 Режимы

Стороны генерируют одноразовый пароль R, комбинируя общий секретный ключ K с уникальной синхропосылкой. Ключ K должен удовлетворять требованиям 5.3.

В зависимости от способа формирования синхропосылки определены три режима (механизма) генерации паролей: HOTP (HMAC-based One-Time Password), TOTP (Time-based One-Time Password) и OCRA (OATH Challenge-Response Algorithms). Названия режимов соответствуют спецификациям [2] — [4].

В режиме HOTP синхропосылка представляет собой счетчик C, который инкрементируется (увеличивается на 1) всякий раз после выработки пароля клиентом или его успешной проверки сервером. Обновленный счетчик может использоваться для генерации или проверки нового пароля. Используются младшие 64 бита двоичной записи C, и поэтому операции с ним можно вести по модулю 2^{64}.

В режиме TOTP синхропосылка представляет собой округленную отметку текущего времени. Текущее время — это количество полных секунд, прошедших с 0 ч 1 января 1970 года в стандартном времени по Гринвичу (без учета секунды координации). Округление T выполняется с параметрами T_0 и T_s, где T_0— базовая отметка времени, неотрицательное целое число, T_s — шаг времени, натуральное число. Должно выполняться ограничение: $T_0 \leqslant T$. Округление состоит в замене T на $\lfloor (T - T_0)/T_s \rfloor$. Рекомендуется выбирать $T_0 = 0$ и использовать шаг $T_s = 30$, т. е. менять пароль каждые 30 с, или $T_s = 60$, т. е. менять пароль каждую минуту.

Округленная отметка T, а вместе с ней и одноразовый пароль остаются постоянными во временном окне длины T_s. Поэтому сервер не должен принимать пароль клиента дважды в одном окне.

Режим OCRA является универсальным: его синхропосылка может включать и счетчик C, и округленную отметку времени T. Правила HOTP для счетчика и правила TOTP для меток времени переносятся в режим OCRA.

Кроме C и T, синхропосылка OCRA обязательно содержит запрос Q противоположной стороны, возможно дополненный собственным запросом. Запросы должны генерироваться случайным или псевдослучайным образом. Синхропосылка режима OCRA может дополнительно включать хэш-значение P долговременного пароля клиента и идентификатор S сеанса между клиентом и сервером.

Одноразовый пароль может генерироваться специальным аппаратным устройством (токеном) и отображаться на дисплее устройства. При использовании механизма HOTP устройство должно быть снабжено перезаписываемой памятью для хранения счетчика, при использовании механизма TOTP — таймером, при использовании механизма OCRA — устройством ввода запроса.

Ввиду ограниченности возможностей ввода запросы OCRA представляют собой строки в одном из трех алфавитов, полученных сужением множества символов таблицы 2:

$$A = \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}, \mathtt{A}, \mathtt{B}, \ldots, \mathtt{Z}\},$$
$$N = \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}\},$$
$$H = \{\mathtt{0}, \mathtt{1}, \ldots, \mathtt{9}, \mathtt{A}, \mathtt{B}, \ldots, \mathtt{F}\}.$$

Первый алфавит — буквенно-цифровой, второй — цифровой, третий — шестнадцатеричный.

А.3 Пароль

Одноразовый пароль представляет собой число из d десятичных цифр (включая незначащие старшие нули). В режимах HOTP и TOTP длина d должна принимать значения 6, 7 или 8. В режиме OCRA дополнительно разрешены $d \in \{4, 5, 9\}$.

Пароль минимальной длины $d = 4$ рекомендуется применять только в особых случаях — тогда, когда пароль другой длины использовать нельзя.

А.4 Вспомогательные алгоритмы

Используется функция хэширования h, удовлетворяющая ограничениям HMAC (пункт 5.2). Длина хэш-значений не должна быть меньше 160. Функция h применяется косвенно — как композиционный элемент алгоритма $\mathtt{hmac}[h]$.

Имитовставки $\mathtt{hmac}[h]$ обрабатываются вспомогательным алгоритмом `otp-dt`, определенным в А.6.

Для использования в режиме OCRA алгоритм хэширования должен описываться уникальной строкой $\mathrm{Name}(h)$ символов таблицы 2. Строка $\mathrm{Name}(h)$ должна полностью определять действие h. Недопустимы ситуации, когда для описания действия, кроме $\mathrm{Name}(h)$, требуется указывать дополнительные параметры, например начальное хэш-значение.

Функции хэширования СТБ 34.101.31 назначается имя `"HBELT"`.

А.5 Синхронизация и блокировка

В режиме HOTP счетчики клиента и сервера могут разойтись. Это произойдет например тогда, когда клиент ошибочно сгенерировал несколько паролей вместо одного, и ни один из них не отослал серверу. Даже в этом случае сервер может восстановить синхронизацию. Сервер выполняет алгоритм генерации паролей s раз, каждый раз инкрементируя счетчик и сравнивая построенный пароль с присланным клиентом. Вычисления заканчиваются как только совпадение паролей будет обнаружено. При этом окончательный счетчик сервера с большой достоверностью совпадет со счетчиком клиента. Если же ни один из s построенных паролей не подошел, то сервер принимает решение об ошибке аутентификации и возвращается к своему первоначальному счетчику.

Сервер может блокировать клиента после v неудачных попыток аутентификации. Блокировка может быть временной или постоянной.

При проектировании системы аутентификации и выборе параметров d, s и v следует учитывать, что вероятность успешной аутентификации противником, который предъявляет случайный пароль, близка к $sv/10^d$. Среди s и v, сохраняющих удобство использования системы аутентификации, рекомендуется выбирать минимальные.

В режиме TOTP могут разойтись показания таймеров клиента и сервера. Сервер может обработать ошибку аутентификации, увеличивая (уменьшая) округленную отметку времени s_1 (s_{-1}) раз на 1, каждый раз генерируя новый пароль и сравнивая его с присланным клиентом. Таким образом можно подавить разность в округленных показаниях таймеров клиента и сервера, если она лежит в интервале $[-s_{-1}, s_1]$. После совпадения паролей сервер может оценить расхождение таймеров и учитывать его в дальнейших сеансах с данным клиентом.

Правила настройки параметров аутентификации на основе паролей TOTP повторяют правила HOTP с заменой s на $s_{-1} + s_1 + 1$.

А.6 Построение пароля по имитовставке

А.6.1 Входные и выходные данные

Входными данными алгоритма `otp-dt` является количество $d \in \{4, 5, \ldots, 9\}$ цифр в пароле и имитовставка $Y \in \{0,1\}^{2l}$. Имитовставка разбивается на $n = |Y|/8$ октетов:

$$Y = Y_0 \parallel Y_1 \parallel \ldots \parallel Y_{n-1}, \quad Y_i \in \{0,1\}^8.$$

Выходными данными является одноразовый пароль $R \in \{0, 1, \ldots, 10^d - 1\}$.

А.6.2 Переменные

Используются переменные $t \in \{0, 1, \ldots, 15\}$ и $r \in \{0, 1, \ldots, 2^{32} - 1\}$.

А.6.3 Алгоритм

Построение пароля состоит в выполнении следующих шагов:

1 $t \leftarrow \bar{Y}_{n-1} \bmod 16$.
2 $r \leftarrow \bar{Y}_t 2^{24} + \bar{Y}_{t+1} 2^{16} + \bar{Y}_{t+2} 2^8 + \bar{Y}_{t+3}$.
3 $r \leftarrow r \bmod 2^{31}$.
4 $R \leftarrow r \bmod 10^d$.
5 Возвратить R.

А.7 Генерация пароля в режиме HOTP

А.7.1 Входные и выходные данные

Входными данными алгоритма генерации одноразовых паролей в режиме HOTP являются количество $d \in \{6, 7, 8\}$ цифр в пароле, секретный ключ $K \in \{0,1\}^{8*}$ и счетчик C — неотрицательное целое число.

Выходными данными является одноразовый пароль $R \in \{0, 1, \ldots, 10^d - 1\}$.

А.7.2 Переменные

Используются переменные $Y \in \{0,1\}^{2l}$ и $W \in \{0,1\}^{64}$. Значение Y должно быть уничтожено после использования. Переменная W разбивается на 8 октетов:

$$W = W_0 \parallel W_1 \parallel \ldots \parallel W_7, \quad W_i \in \{0,1\}^8.$$

A.7.3 Алгоритм

Генерация пароля в режиме HOTP состоит в выполнении следующих шагов:

1 $W \leftarrow \langle C \rangle_{64}$.

2 $Y \leftarrow \texttt{hmac}[h](K, W_7 \parallel W_6 \parallel \ldots \parallel W_0)$.

3 $R \leftarrow \texttt{otp-dt}(d, Y)$.

4 Возвратить R.

A.8 Генерация пароля в режиме TOTP

A.8.1 Входные и выходные данные

Входными данными алгоритма генерации одноразовых паролей в режиме TOTP являются количество $d \in \{6, 7, 8\}$ цифр в пароле, секретный ключ $K \in \{0, 1\}^{8*}$ и округленная отметка T текущего времени — неотрицательное целое число.

Выходными данными является одноразовый пароль $R \in \{0, 1, \ldots, 10^d - 1\}$.

A.8.2 Алгоритм

Генерация пароля в режиме TOTP повторяет генерацию в режиме HOTP с заменой C на T.

A.9 Генерация пароля в режиме OCRA

A.9.1 Входные данные

Входными данными алгоритма генерации одноразовых паролей в режиме OCRA являются:

1) количество $d \in \{4, 5, \ldots, 9\}$ цифр в пароле;

2) секретный ключ $K \in \{0, 1\}^{8*}$;

3) счетчик C — неотрицательное целое число;

4) запрос (ы) $Q \in \{0, 1\}^{8*}$. Октеты Q должны одновременно принадлежать одному из трех алфавитов: A, N или H. Слово Q содержит запрос противоположной стороны, возможно дополненный собственным запросом. Каждый из запросов должен состоять из целого числа октетов: от 4 до 64. Рекомендуется использовать запрос из 8 символов-цифр алфавита N;

5) округленная отметка T текущего времени — неотрицательное целое число. Округление T должно выполняться с базовой отметкой $T_0 = 0$ и шагом T_s, задаваемым целым числом секунд, минут или часов. Рекомендуется использовать шаг в 1 мин;

6) Хэш-значение P статического пароля клиента. Хэш-значение должно вычисляться с помощью функции h_1, которая совпадает с h или отличается от нее. Функции h_1 должно быть назначено имя $\mathrm{Name}(h_1)$;

7) идентификатор $S \in \{0, 1\}^{8*}$ сеанса между клиентом и сервером. Длина S в октетах не должна превышать 512. Рекомендуется использовать идентификаторы из 64 октетов.

Параметры C, T, P, S являются необязательными, они могут не подаваться на вход алгоритма.

A.9.2 Описатель

Перечень используемых входных параметров и их форматы задаются строковым описателем D, который также подается на вход алгоритма. Описатель имеет следующий вид:

$$D = \texttt{"OCRA-1:HOTP-h-d:[C-]Qfq[-Pp][-Ss][-Ttg]"}.$$

Квадратные скобки окаймляют часть строки, которая может быть исключена (вместе со скобками). Наличие в описателе символа `C` означает включение в состав входных данных счетчика, наличие символа `P` — хэш-значения пароля, `S` — идентификатора сеанса, `T` — округленной отметки времени.

Строчные символы-буквы описателя переопределяются по следующим правилам:

1) символ `h` меняется на строку $\mathrm{Name}(h)$;

2) символ `d` меняется на цифру из множества $\{\mathtt{4}, \mathtt{5}, \ldots, \mathtt{9}\}$, представляющую число d;

3) символ `f` меняется на код алфавита Q: `A`, если используется алфавит A, `N`, если — алфавит N, и `H`, если — H;

4) символ `q` меняется на строку из множества $\{\texttt{"04"}, \texttt{"05"}, \ldots, \texttt{"64"}\}$, представляющую максимальную длину запроса в октетах;

5) символ `p` меняется на строку $\mathrm{Name}(h_1)$;

6) символ `s` меняется на строку из множества $\{\texttt{"001"}, \texttt{"002"}, \ldots, \texttt{"512"}\}$, представляющую длину S в октетах;

7) символ `g` меняется на код единиц шага времени: `S`, если шаг времени измеряется в секундах, `M`, если — в минутах, и `H`, если — в часах;

8) символ `t` меняется на строку, представляющую шаг времени: элемент множества $\{\texttt{"1"}, \texttt{"2"}, \ldots, \texttt{"59"}\}$, если шаг времени измеряется в секундах или минутах, или множества $\{\texttt{"1"}, \texttt{"2"}, \ldots, \texttt{"48"}\}$, если шаг времени измеряется в часах.

А.9.3 Выходные данные

Выходными данными является одноразовый пароль $R \in \{0, 1, \ldots, 10^d - 1\}$.

А.9.4 Переменные

Используются переменные $X \in \{0,1\}^{64}$, $Y \in \{0,1\}^{2l}$ и $W \in \{0,1\}^{64}$. Значение Y должно быть уничтожено после использования. Переменная W разбивается на 8 октетов:

$$W = W_0 \parallel W_1 \parallel \ldots \parallel W_7, \quad W_i \in \{0,1\}^8.$$

А.9.5 Алгоритм

Генерация пароля в режиме OCRA состоит в выполнении следующих шагов:

1 $X \leftarrow D \parallel \mathtt{00}_{16}$.

2 Если счетчик C подан на вход, то

  1) $W \leftarrow \langle C \rangle_{64}$;

  2) $X \leftarrow X \parallel W_7 \parallel W_6 \parallel \ldots \parallel W_0$.

3 $X \leftarrow X \parallel Q \parallel 0^{1024-|Q|}$.

4 Если хэш-значение P подано на вход, то $X \leftarrow X \parallel P$.

5 Если идентификатор S подан на вход, то $X \leftarrow X \parallel S$.

6 Если отметка T подана на вход, то

  1) $W \leftarrow \langle T \rangle_{64}$;

  2) $X \leftarrow X \parallel W_7 \parallel W_6 \parallel \ldots \parallel W_0$.

7 $Y \leftarrow \mathtt{hmac}[h](K, X)$.

8 $R \leftarrow \mathtt{otp\text{-}dt}(d, Y)$.

9 Возвратить R.

A.9.6 Аутентификация в режиме OCRA

Одноразовые пароли OCRA могут использоваться как для односторонней, так и взаимной аутентификации сторон.

Протокол односторонней аутентификации:

1 Сервер формирует запрос Q и отправляет его клиенту.

2 Клиент по запросу Q генерирует одноразовый пароль R и отправляет его серверу.

3 Сервер по запросу Q генерирует одноразовый пароль и сравнивает его с R.

Успешное выполнение всех шагов протокола означает успешную аутентификацию клиента перед сервером. При генерации R кроме запроса Q может дополнительно использоваться любой заранее согласованный набор необязательных параметров C, P, S и T.

Протокол взаимной аутентификации:

1 Клиент формирует запрос Q_C и отправляет его серверу.

2 Сервер формирует запрос Q_S, по $Q = Q_C \parallel Q_S$ генерирует одноразовый пароль R_S и отправляет клиенту Q_S и R_S.

3 Клиент по запросам $Q = Q_C \parallel Q_S$ генерирует одноразовый пароль и сравнивает его с R_S, затем по запросам $Q' = Q_S \parallel Q_C$ генерирует одноразовый пароль R_C и отправляет его серверу.

4 Сервер по запросам $Q' = Q_S \parallel Q_C$ генерирует одноразовый пароль и сравнивает его с R_C.

Успешное выполнение всех шагов протокола означает успешную взаимную аутентификацию сторон. При генерации R_S кроме запросов может дополнительно использоваться любой заранее согласованный набор необязательных параметров C, S и T, а при генерации R_C — любой заранее согласованный набор необязательных параметров C, P, S и T.

Приложение Б
(рекомендуемое)
Модуль АСН.1

Б.1 Идентификаторы

Алгоритмам настоящего стандарта присваиваются следующие идентификаторы:

`hmac-hspec`	алгоритм выработки имитовставки в режиме HMAC (см. 6.1) с функций хэширования, определяемой долговременными параметрами;
`hmac-hbelt`	алгоритм выработки имитовставки в режиме HMAC (см. 6.1) с функцией хэширования, определенной в СТБ 34.101.31;
`brng-ctr-hspec`	алгоритм генерации псевдослучайных чисел в режиме счетчика (см. 6.2) с функцией хэширования, определяемой долговременными параметрами;
`brng-ctr-hbelt`	алгоритм генерации псевдослучайных чисел в режиме счетчика (см. 6.2) с функцией хэширования, определенной в СТБ 34.101.31;
`brng-ctr-stb11761`	алгоритм генерации псевдослучайных чисел в режиме счетчика (см. 6.2) с функцией хэширования, определенной в СТБ 1176.1, и дополнительными уточнениями (см. далее);
`brng-hmac-hspec`	алгоритм генерации псевдослучайных чисел в режиме HMAC (см. 6.3) с функцией хэширования, определяемой долговременными параметрами;
`brng-hmac-hbelt`	алгоритм генерации псевдослучайных чисел в режиме HMAC (см. 6.3) с функцией хэширования, определенной в СТБ 34.101.31;
`hotp-hspec`	алгоритм генерации одноразового пароля в режиме HOTP (см. А.7) с функцией хэширования, определяемой долговременными параметрами;
`hotp-hbelt`	алгоритм генерации одноразового пароля в режиме HOTP (см. А.7) с функцией хэширования, определенной в СТБ 34.101.31;
`totp-hspec`	алгоритм генерации одноразового пароля в режиме TOTP (см. А.7) с функцией хэширования, определяемой долговременными параметрами;
`totp-hbelt`	алгоритм генерации одноразового пароля в режиме TOTP (см. А.7) с функцией хэширования, определенной в СТБ 34.101.31;
`ocra-hspec`	алгоритм генерации одноразового пароля в режиме OCRA (см. А.9) с функцией хэширования, определяемой долговременными параметрами;
`ocra-hbelt`	алгоритм генерации одноразового пароля в режиме OCRA (см. А.9) с функцией хэширования, определенной в СТБ 34.101.31.

Идентификатор алгоритма либо явно определяет используемую функцию хэширования h, либо указывает, что h задается ссылочно, через дополнительные параметры алгоритма.

Для задания h рекомендуется использовать тип `AlgorithmIdentifier` АСН.1, определенный в СТБ 34.101.19 следующим образом:

```
AlgorithmIdentifier ::= SEQUENCE {
  algorithm OBJECT IDENTIFIER,
  parameters ANY DEFINED BY algorithm
}
```

Компонент `algorithm` этого типа должен задавать идентификатор алгоритма хэширования, а компонент `parameters` — параметры данного алгоритма.

При использовании функции хэширования, определенной в СТБ 1176.1:

– компонент `algorithm` должен принимать значение {1 2 112 0 2 0 1176 1 11};

– компонент `parameters` должен иметь тип `OCTET STRING` и принимать значение $\langle H \rangle_{256}$.

Здесь H — долговременный параметр алгоритма хэширования (неотрицательное целое), описанный в п. 5.1 СТБ 1176.1. Долговременный параметр L, также описанный в п. 5.1 СТБ 1176.1, должен равняться 256.

В алгоритме `brng-ctr-stb11761` используется алгоритм хэширования СТБ 1176.1, в котором на шаге 15 вместо проверки $d = n + 2$ используется проверка $d = n + 1$, а параметр H задается так, что

$$\langle H \rangle_{256} = \texttt{4E4E9C9C 9C9C4E4E 9C9C4E4E 4E4E9C9C 9C9C4E4E 4E4E9C9C 4E4E9C9C 9C9C4E4E}_{16}.$$

При таких уточнениях алгоритм `brng-ctr-stb11761` соответствует алгоритму, определенному в [5].

Б.2 Модуль АСН.1

```
Brng-module-v2 {iso(1) member-body(2) by(112) 0 2 0 34 101 47 module(1) ver2(2)}
DEFINITIONS ::=
BEGIN
  brng OBJECT IDENTIFIER ::= {1 2 112 0 2 0 34 101 47}

  hmac-hspec OBJECT IDENTIFIER ::= {brng 11}
  hmac-hbelt OBJECT IDENTIFIER ::= {brng 12}
  brng-ctr-hspec OBJECT IDENTIFIER ::= {brng 21}
  brng-ctr-hbelt OBJECT IDENTIFIER ::= {brng 22}
  brng-ctr-stb11761 OBJECT IDENTIFIER ::= {brng 23}
  brng-hmac-hspec OBJECT IDENTIFIER ::= {brng 31}
  brng-hmac-hbelt OBJECT IDENTIFIER ::= {brng 32}

  hotp-hspec OBJECT IDENTIFIER ::= {brng 111}
  hotp-hbelt OBJECT IDENTIFIER ::= {brng 112}
  totp-hspec OBJECT IDENTIFIER ::= {brng 121}
  totp-hbelt OBJECT IDENTIFIER ::= {brng 122}
  ocra-hspec OBJECT IDENTIFIER ::= {brng 131}
  ocra-hbelt OBJECT IDENTIFIER ::= {brng 132}
END
```

Приложение В
(справочное)
Проверочные примеры

В.1 Выработка имитовставки в режиме HMAC

В таблице В.1 представлен пример выполнения алгоритма `hmac-hbelt`. Здесь и далее названия алгоритмов даются в соответствии с приложением А.

Таблица В.1 — Выработка имитовставки (алгоритм `hmac-hbelt`)

K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03_{16}
X	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF $66D2E88A_{16}$
Y	D4828E63 12B08BB8 3C9FA653 5A463554 9E411FD1 1C0D8289 359A1130 $E930676B_{16}$
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 $03A98BF6_{16}$
X	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF $66D2E88A_{16}$
Y	41FFE864 5AEC0612 E952D2CD F8DD508F 3E4A1D9B 53F6A1DB 293B19FE $76B1879F_{16}$
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6 92BD9B1C E5D14101 5445_{16}
X	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF $66D2E88A_{16}$
Y	7D01B84D 2315C332 277B3653 D7EC6470 7EBA7CDF F7FF7007 7B1DECBD $68F2A144_{16}$

В.2 Генерация псевдослучайных чисел в режиме счетчика

В таблицах В.2, В.3 представлены примеры генерации псевдослучайных чисел с помощью алгоритмов `brng-ctr-hbelt` и `brng-ctr-stb11761`.

Таблица В.2 — Генерация псевдослучайных чисел (алгоритм `brng-ctr-hbelt`)

n	3
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 $03A98BF6_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF $66D2E88A_{16}$
X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B 5CB0C0FF 33C356B8 35C405AE D8E07F99 E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 $D5BB9C4F_{16}$
Y	1F66B5B8 4B733967 4533F032 9C74F218 34281FED 0732429E 0C79235F C273E269 4C0E74B2 CD5811AD 21F23DE7 E0FA742C 3ED6EC48 3C461CE1 5C33A77A A308B7D2 0F51D913 47617C20 BD4AB07A EF4F26A1 AD1362A8 F9A3D42F BE1B8E6F $1C88AAD5_{16}$

Таблица В.3 — Генерация псевдослучайных чисел (алгоритм `brng-ctr-stb11761`)

n	3
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
X	B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D 5BE3D612 17B96181 FE6786AD 716B890B 5CB0C0FF 33C356B8 35C405AE D8E07F99 E12BDC1A E28257EC 703FCCF0 95EE8DF1 C1AB7638 9FE678CA F7C6F860 D5BB9C4F$_{16}$
Y	F7619A01 893ED14E CA0FF583 2797086C FB5B2F90 8CFB4B33 4BC201C9 814D744F 3F616631 B040A16E 0D1EC7A7 CC62000C 377869AD 874E473C 58493143 9FC6D41D 4DAFA372 72A93832 BA8D405F 1DC58F70 B943CAC3 3A926789 C8A3C819 E6F24F98$_{16}$

В.3 Генерация псевдослучайных чисел в режиме HMAC

В таблице В.4 представлен пример генерации псевдослучайных чисел с помощью алгоритма `brng-hmac-hbelt`.

Таблица В.4 — Генерация псевдослучайных чисел (алгоритм `brng-hmac-hbelt`)

n	3
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
S	BE329713 43FC9A48 A02A885F 194B09A1 7ECDA4D0 1544AF8C A58450BF 66D2E88A$_{16}$
Y	AF907A0E 470A3A1B 268ECCCC C0B90F23 9FE94A2D C6E01417 9FC789CB 3C3887E4 695C6B96 B84948F8 D76924E2 2260859D B9B5FE75 7BEDA2E1 7103EE44 655A9FEF 648077CC C5002E05 61C6EF51 2C513B8C 24B4F3A1 57221CFB C1597E96 9778C1E4$_{16}$

В.4 Генерация одноразовых паролей в режиме HOTP

В таблице В.5 представлен пример генерации одноразовых паролей с помощью алгоритма `hotp-hbelt`.

Таблица В.5 — Генерация одноразовых паролей (алгоритм `hotp-hbelt`)

d	8
K	E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6$_{16}$
C	BE329713 43FC9A48$_{16}$
R	21157984
C	BE329713 43FC9A49$_{16}$
R	17877985
C	BE329713 43FC9A4A$_{16}$
R	26078636

В.5 Генерация одноразовых паролей в режиме TOTP

В таблице В.6 представлен пример генерации одноразовых паролей с помощью алгоритма `totp-hbelt`.

Таблица B.6 — Генерация одноразовых паролей (алгоритм `totp-hbelt`)

T_0	0
T_s	60
d	8
K	`E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6`$_{16}$
t	1449165288
T	24152754
R	97660664
t	1449165300
T	24152755
R	94431522
t	1449165419
T	24152756
R	55973851
Примечание — t — отметка текущего времени до округления, T — отметка текущего времени посл округления.	

B.6 Генерация одноразовых паролей в режиме OCRA

В таблице B.7 представлен пример генерации одноразовых паролей с помощью алгоритма `ocra-hbelt`.

Таблица B.7 — Генерация одноразовых паролей (алгоритм `ocra-hbelt`)

D	`"OCRA-1:HOTP-HBELT-8:C-QN08-PHBELT-S064-T1M"`
K	`E9DEE72C 8F0C0FA6 2DDB49F4 6F739647 06075316 ED247A37 39CBA383 03A98BF6`$_{16}$
P	`ABEF9725 D4C5A835 97A367D1 4494CC25 42F20F65 9DDFECC9 61A3EC55 0CBA8C75`$_{16}$
S	`B194BAC8 0A08F53B 366D008E 584A5DE4 8504FA9D 1BB6C7AC 252E72C2 02FDCE0D` `5BE3D612 17B96181 FE6786AD 716B890B 5CB0C0FF 33C356B8 35C405AE D8E07F99`$_{16}$
Q	`"21157984"`
C	`BE329713 43FC9A4B`$_{16}$
T	24152759
R	85199085
Q	`"17877985260786 36"`
C	`BE329713 43FC9A4C`$_{16}$
T	24152769
R	89873725
Q	`"2607863617877985"`
C	`BE329713 43FC9A4D`$_{16}$
T	24152770
R	21318915

Приложение Г
(справочное)
Использование генераторов случайных чисел

Генератор случайных чисел вырабатывает последовательности, каждый следующий элемент которых статистически и вычислительно трудно предсказать по всем предыдущим элементам. Генератор использует один или несколько источников случайности и включает средства обработки данных от источников.

В компьютерных системах распространены следующие источники случайности:

– физические источники, использующие процессы в физических устройствах (например, шум в радиоэлектронных приборах);

– системные источники, использующие состояния, процессы и события операционной системы (системное время, сетевая активность, прерывания);

– источники, основанные на активности операторов (движения мышью, нажатия клавиш).

Предпочтительным является использование физических источников случайности.

Для источника случайности S проводится оценка энтропии (неопределенности, вариабельности) его выходных последовательностей. Для этого строится вероятностная модель источника S и в рамках этой модели определяется величина h такая, что основная вероятностная масса выходных последовательностей длины n сосредоточена на множестве мощности 2^{nh}. Величина h называется удельной энтропией на наблюдение. Например, если S выдает случайные независимые символы алфавита A и вероятность появления символа α равняется p_α, то удельная энтропия рассчитывается по формуле

$$h = -\sum_{\alpha \in A} p_\alpha \log_2 p_\alpha \quad (0 \cdot \log_2 0 = 0). \tag{Г.1}$$

Кроме h существуют и другие характеристики неопределенности. Например, в криптографических приложениях используют минимальную удельную энтропию h_{min}, которая характеризует сложность предсказания самой вероятной выходной последовательности S. Для описанного выше источника минимальная удельная энтропия определяется как

$$h_{min} = \min_{\alpha \in A}(-\log_2 p_\alpha). \tag{Г.2}$$

Оценка величины h (или h_{min}) является сложной задачей, если распределение $\{p_\alpha\}$ известно не полностью, источник S не является стационарным, между выходными символами S имеются зависимости и в других ситуациях. Для оценки h могут применяться статистические методы, основанные на частотах встречаемости в выходных последовательностях m-грамм, а также алгоритмические методы, основанные на коэффициентах обратимого или необратимого сжатия выходных последовательностей.

Если удельная энтропия h оценена, то можно сделать вывод о том, что для надежной генерации секретного ключа длины l требуется использовать не менее l/h наблюдений от источника случайности.

Библиография

[1] Krawchuk H., Bellare M., Canetti R. HMAC: Keyed-Hashing for Message Authentication. Request for Comments: 2104, 1997
(HMAC: ключезависимое хэширование для аутентификации сообщений)

[2] M'Raihi D., Bellare M., Hoornaert F., Naccache D., Ranen O. HOTP: An HMAC-Based One-Time Password Algorithm. Request for Comments: 4226, 2005
(HOTP: Алгоритм генерации одноразовых паролей на основе HMAC)

[3] M'Raihi D., Machani S., Pei M., Rudell J. TOTP: Time-Based One-Time Password Algorithm. Request for Comments: 6238, 2011
(TOTP: Алгоритм генерации одноразовых паролей на основе времени)

[4] M'Raihi D., Rudell J., Bajaj S., Machani S., Naccache D. OCRA: OATH Challenge-Response Algorithm. Request for Comments: 6287, 2011
(OCRA: Алгоритм OATH типа вызов-ответ)

[5] РД РБ 07040.1206-2003. Руководящий документ Республики Беларусь. Автоматизированная система межбанковских расчетов. Процедура выработки псевдослучайных данных с использованием секретного параметра. — Мн.: Национальный Банк Республики Беларусь, 2003

Поправка к официальной редакции

В каком месте	Напечатано	Должно быть
Приложение В, таблица В.7, ячейки во втором столбце со значениями R	77614623 99509664 75687625	85199085 89873725 21318915