УДК 004.056

А.В. ПОТИЙ, д-р техн. наук, А.С. КАРПЕНКО

РЕАЛИЗАЦИЯ ПОСТКВАНТОВОГО АЛГОРИТМА ЭЛЕКТРОННО-ЦИФРОВОЙ ПОДПИСИ

Введение

Стойкость криптографических алгоритмов с открытым ключом базируется на вычислительной сложности теоретико-числовых задач факторизации больших чисел, дискретного логарифмирования, преобразований в поле точек на эллиптической кривой. Известными являются алгоритмы RSA, DSA, ECDSA.

Исследования в области квантовых вычислений формируют новые вызовы в данной области криптографии. По утверждениям специалистов NIST и ETSI данные алгоритмы станут небезопасными после того, как появится прикладной квантовый компьютер [1 – 3]. С использованием квантового компьютера и алгоритма Шора известные на сегодняшний день криптоалгоритмы с открытым ключом будут скомпрометированы (табл. 1) [2]. В связи с этим актуальным стал поиск альтернативных криптографических примитивов, которые в перспективе будут устойчивы к атакам, реализуемым на квантовых компьютерах. Такие алгоритмы получили название квантово-безопасные (quantum-safe) (ETSI, EU) или квантово-защищенные (quantum-resistant) (NIST, USA) алгоритмы, а соответствующий раздел криптографии, изучающий проблемы проектирования таких алгоритмов, получил название постквантовой криптографии (post-quantum cryptography). Рабочие группы ETSI и NIST определили перспективные направления, в рамках которых возможно получить приемлемые решения: суперсингулярные эллиптические кривые, мультивариативную криптографию, криптографию на основе помехоустойчивого кодирования, криптографию на основе хеш функций.

Актуальной задачей является поиск алгоритма цифровой подписи как наиболее массового в применении криптопримитива. В Украине принят национальный стандарт функции хеширования Купина ДСТУ 7564:2014. По некоторым оценкам, данная хеш-функция может быть отнесена к постквантовым примитивам. В связи с этим она представляет интерес для построения на ее основе постквантовой ЭЦП класса Hash-Based digital signature (HB-подпись). В данной работе рассматривается вариант построения ЭЦП на основе схемы Lamport-Diffie-Winternitz-Merkle (LDWM) с использованием хеш-функции «Купина». По нашему мнению, это перспективное направление для разработки национальной постквантовой ЭЦП.

Таблица 1

Уровень защищенности используемых алгоритмов

Алгоритм	Длина ключа	Безопасный уровень	
		Классический компьютер	Квантовый компьютер
RSA-1024	1024 b	80 b	0 b
RSA-2048	2048 b	112 b	0 b
ECC-256	256 b	128 b	0 b
ECC-384	384 b	256 b	0 b

Схема одноразовой подписи и схема дерева сигнатур Меркли Lamport-Diffie-Winternitz-Merkle (LDWM) обеспечивают контроль целостности сообщений на принципах несимметричной криптографии. Сообщение подписывается секретным ключом, а проверка осуществляе-

тся открытым ключом. Схема одноразовой подписи может быть использована только для одного сообщения, поэтому есть необходимость использовать общую схему дерева сигнатур Меркли, где одним из компонентов является схема одноразовой подписи.

1. Аспекты реализации схемы ЭЦП

Реализация схемы ЭЦП осуществляется в два этапа: реализация схемы одноразовой подписи и реализация общей схемы дерева сигнатур. В качестве базового элемента ЭЦП в данной схеме будем использовать две хеш-функции SHA-2 и ДСТУ 7564:2014 «Купина». Заметим, что стойкость НВ-подписи зависит от стойкости хеш-функции.

1.1. Первый етап

Для того чтобы непосредственно начать описывать структурные особенности схемы одноразовой подписи, необходимо определить ряд параметров:

- m – длина в байтах каждого элемента LDWM;
- n – длина в байтах на выходе хеш-функции;
- w – параметр Winternitz, который принимает значения {1,2,,4,8};
- p – число байтовых строк, которые составляют LDWM схему;
- ls – число, которое используется в функции контрольной суммы C.

1.1.1. Формирование закрытого ключа.

Закрытый ключ x представляет собой массив размером p, содержащий n-байтные строки, которые генерируются путем использования генератора случайных последовательностей.

```
for i=0, i<p, i++
    set x[i] to a uniformly random value
end  for
    return x
```

1.1.2. Формирование открытого ключа.

В LDWM открытый ключ y генерируется из закрытого ключа x с использованием операции конкатенации и хеш-функции H, которая в ходе выполнения возвращает 256 значение.

```
e = 2^w – 1
   for i=0,i<p,i++
     y[i] = F^e(x[i])
   end for
 return H(y[0] || y[1] || ... || y[p-1])
```

Работа функции F описывается выражением [7]:

```
F^i(x) = / F( F^(i-1)(x) )  for i > 0
         \ x                for i = 0.
```

Данная функция выполняет операцию хеширования i раз и возвращает 20-байтное значение.

1.1.3. Функция Checksum.

Функция Checksum возвращает положительное число sum, которая нужна для предотвращения изменения ЭЦП.

```
sum = 0
   for i=0,i<u,i++
     sum = sum + (2^w – 1) – coef(S, i, w)
   end for
return (sum << ls)
```

1.1.4. Генерация сигнатуры.

Сигнатура вычисляется путем конкатенации значения хеш-функции от сообщения и контрольной суммы с использованием закрытого ключа.

V = (*H*(*message*) || *C*(*H*(*message*)))
  for i=0, i<p, i++
    a = *coef*(*V, i, w*)
    y[i] = *F*^*a*(*x*[*i*])
  end for
return (y[*0*] || *y*[1] || ... || *y*[*p*-1] *)*

1.1.5. Проверка подписи.

Для того чтобы проверить сообщение, получателю необходимо завершить серию *F*, используя значение хеш-функции от сообщения и ее контрольной суммы. Эти действия приводят к получению открытого ключа.

V = (*H*(*message)* || *C*(*H*(*message*)))
  for i=0,i<p,i++
    a = (2^*w* – 1) – *coef*(*V, i, w*)
    z[*i*] = *F*^*a*(*x*[*i*])
  end for
  if public key is equal to H(z[0] || *z*[1] || ... || *z*[*p*-1])
    return 1 (message signature is valid)
  else
    return 0 (*message signature is invalid*)

1.2. Второй етап

Дерево сигнатур Меркли использует два компонента: схему одноразовой подписи и хеш-функцию. Каждая пара открытого/закрытого ключей ассоциирована с *k-way* деревом. Каждый узел дерева содержит *n-byte* значение. Каждый лист дерева содержит пару отрытый/закрытый ключ LDWM. В свою очередь, корень дерева называется «Открытым ключом дерева сигнатур Меркли». Как и в первом этапе реализации схемы, необходимо определить такие параметры:

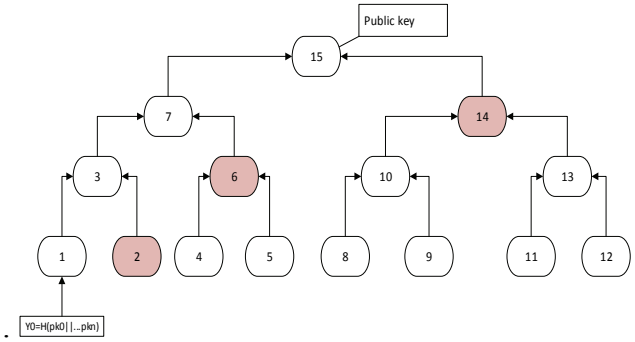
k – число дочерних узлов во внутреннем узле;
h – высота дерева;
n – число, ассоциированное с каждым узлом.

1.2.1. Закрытый ключ.

Закрытый ключ состоит из *k^h* LDWM закрытых ключей и закрытого ключа LDWM следующего, не использованного листа дерева. Генерация производится с использованием псевдослучайного генератора с использованием секретного значения.

1.2.2. Открытый ключ.

Для вычисления открытого ключа необходимо поместить листья дерева значения LDWM открытых ключем, полученных в п. 1.1.2, далее необходимо заполнить вышестоящие узлы путем применения операции конкатенации и хеш-функции к дочерним узлам. В дереве Меркли открытым ключом является корневой узел. На рисунке показан пример генерации открытого ключа: для этого будет использован первый узел (1) со значением Y_0, узел номер 3 будет иметь хеш-значение конкатенации первого и второго узлов. Окрашенные узлы (2,6,14) представляют первый путь аутентификации. Он образован двойниками правых узлов, соединяющих лист и корень дерева, которые используются для проверки открытого ключа. Алгоритм поиска открытого ключа описан в [7].



Генерацию открытого ключа можно представить в виде следующего псевдокода:

for i = 0 *to num_ldwm_keys by steps of k*
level = 0
for j = 0 *to k*-1
push ldwm_public_key(*i*+*j*) *on the data stack*
push level on the integer stack
end for
while the height of the stack is at least k
if the top k elements on the integer stack are equal
pop the top k elements of the data stack
pop the top k elements of the integer stack,
hash the top k elements of the data stack
push the hash result on the data stack
push level on to the integer stack
endif endwhile end for

1.2.3. Генерация подписи.

Подпись состоит из трех элементов:

- подпись LDWM;.
- индекс использованного листа;
- путь аутентифкации.

1.2.4. Проверка подписи.

Для того чтобы проверить подпись, необходимо сначала убедиться в том, что подпись LDWM правильна. В противном случае проверка дальше не осуществляется и считается, что подпись не верна. Дальше необходимо «прошагать» по дереву, использовав выражение

$$p_i = \begin{cases} g(Aut_{i-1} \| p_{i-1}) \text{ if } \lfloor s/(2^{i-1}) \rfloor \equiv 1 \bmod 2; \\ g(p_{i-1} \| Aut_{i-1}) \text{ otherwise.} \end{cases}$$

Если значения корня дерева и вычисленное значение совпали, то принимается решение, что подпись верна, во всех остальных случаях подпись не действительна.

2. Сравнительный анализ

Перед авторами стояла задача реализовать НВ-подпись с использованием национального стандарта хеш-функции и сравнить с реализацией на основе хеш-функции SHA-2.

Украинский национальный стандарт функции хеширования способен работать с длиной хеш-значения 256/384/512. Эта хеш-функция базируется на структуре Rijndael. Стойкость «Купины» представлена в табл. 2.

Таблица 2

Стойкость к криптографическим атакам

Вид атаки	Купина – 256
Коллизия	2^{128}
Прообраз	2^{256}
Второй прообраз	2^{256}
Фиксированной точки	2^{256}
Увеличение длины	2^{256}

Хеш-функция SHA-256 разработана Агентством национальной безопасности США и построена на основе структуры Меркли – Дамгарда. Эта функция имеет при своих 64 раундах достаточную стойкость в 112 бит безопасности.

В табл. 3 представлены результаты сравнения двух реализация НВ-подписи

Таблица 3

Временные соотношения

Алгоритм	Время генерации ключей (мс)	Время подписи (мс)	Время проверки подписи (мс)
SHA-2	50	4	4
«Купина»	47	3	3

В ходе эксперимента была использована схема дерева Меркли с глубиной дерева равной пяти. Данное программное обеспечение было запущено на процессоре Intel(R) Pentium(R) CPU G850 @2.90 GHz.

Заключение

Описаны прикладные аспекты реализации постквантового алгоритма ЭЦП с иллюстрацией алгоритма, описанного на псевдокоде. Показана принципиальная возможность реализации постквантовой цифровой подписи с использованием национального стандарта хеш-функции. В ходе анализа алгоритма с использованием разных хеш-функций выявлено небольшое превосходство алгоритма «Купина-256» над функцией хеширования SHA-256. Это позволяет сделать вывод, что можно направить усилия исследователей в этом направлении и продолжить анализ различных характеристик НВ-подписи на основе национального стандарта хеш-функции. Кроме того, это направление видится как наиболее перспективное для разработки нового национального стандарта цифровой подписи.

Список литературы: 1. *ETSI* GR QSC 001 V.1.1.1 (2016-07). Quntum-Safe Cryptography (QSC); Quantum-safe algorithmic framework. 2. *ETSI* White Paper №8: Quntum safe cryptography and security. – 2015. 3. *NIST* PQC workshop: SAFEcrypto Project, M. O'Niell. – 2015. 4. *NIST* Workshop on Cybersecurity in a Post-Quntum World (2015). PQCrypto project, T Lange. 5. *PQCrypto*. Initial recommendation of Long-term secure post-quantum systems. – 2015. 6. *Горбенко, Ю. І.* Побудування та аналіз систем, протоколів і засобів криптографічного захисту інформації. Ч. 1: Методи побудування та аналізу, стандартизація та застосування криптографічних систем ; за заг. ред. І.Д. Горбенка. – Харків : Форт, 2015. – 960 с. 7. *McGrew D., Curcio M.* Hash-Based Signatures draft-mcgrew-hash-sigs-00[Электронный ресурс] / D. McGrew, M. Curcio. – Режим доступа: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-00 8. *McGrew D., Curcio M.* Hash-Based Signatures draft-mcgrew-hash-sigs-01[Электронный ресурс] / D. McGrew, M. Curcio. – Режим доступа: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-01 9. *McGrew D., Curcio M.* Hash-Based Signatures draft-mcgrew-hash-sigs-03[Электронный ресурс] / D. McGrew, M. Curcio. – Режим доступа: https://tools.ietf.org/html/draft-mcgrew-hash-sigs-03

Национальный аэрокосмический университет имени Н.Е. Жуковского «ХАИ»

Поступила в редколлегию 14.09.2016