ВикипедиЯ

Кольцевая подпись

Материал из Википедии — свободной энциклопедии

Кольцевая подпись (англ. *ring signature*) — один из механизмов реализации электронной подписи, при котором известно, что сообщение подписал один из членов списка потенциальных подписантов, но не раскрывающий, кто именно. Подписант самостоятельно формирует список из произвольного числа лиц (включая и себя). Для наложения подписи подписывающему не требуется разрешение, содействие или помощь со стороны включённых в список лиц — используются лишь открытые ключи всех членов списка и собственный закрытый ключ.

Математический алгоритм кольцевой подписи был разработан Рональдом Ривестом, Ади Шамиром и Яэлью Тауман (англ. *Yael Tauman*) и представлен в 2001 году на международной конференции Asiacrypt[1]. Авторы старались в названии подчеркнуть отсутствие центральной или координирующей структуры при формировании такой подписи: «кольцо представляет собой геометрическую фигуру с однородной периферией и без центра».

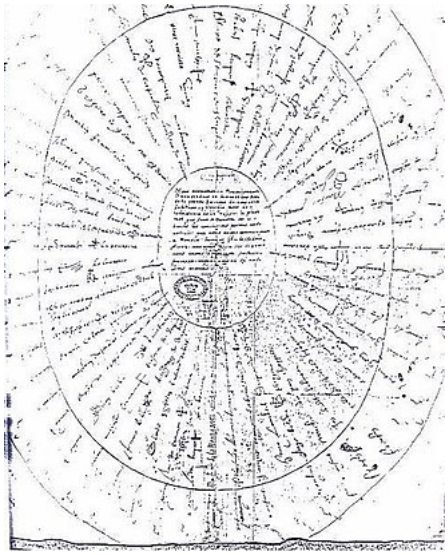
Содержание

История

Идея групповой подписи под прошениями или жалобами, подтверждающей, что все подписанты поддерживают обращение, но не позволяющей выявить её автора или инициатора берёт начало в прошлом. Так, английский термин *Round-robin* известен с XVII столетия и обозначает петицию, которую подписывали по кругу без соблюдения иерархии, чтобы избежать наказания для подписавшегося первым[2] — своеобразный вариант круговой поруки. В 1898 году после осады Сантьяго на Кубе во время Испано-американской войны высокопоставленные офицеры 5-го армейского корпуса подписали по кругу письмо в штаб-квартиру армии в Вашингтоне с требованием вернуть корпус в Соединенные Штаты для лечения и отдыха. Письмо попало в прессу и получило широкую известность, а также вызвало резонанс в правительстве президента Мак-Кинли[3].

Множественная подпись стала электронным аналогом бумажной подписи по кругу. В 1991 году Дэвид Чаум (англ. *David Chaum*) и Евген Ван Хейст (англ. *Eugene Van Heyst*) предложили схему групповой подписи[1], где подписантом является кто-то из членов группы, которую сформировал администратор. Проверяющий может убедиться, что подписант член группы, но не может узнать, кто именно. Администратор может установить подписанта[4].

Кольцевые подписи по существу схожи с групповыми, но, в отличие от последних, нет возможности установления личности подписанта, нет администратора или координатора. Все члены списка, за исключением самого подписанта, могут не знать содержания сообщения[1].



«Прошение о свободе» (1623 год), подписанное по кругу главами 56 семей Валлонии

Общее описание механизма создания и проверки кольцевой подписи

Предполагается, что существует некий перечень, где указана однозначная связь человека с его открытым (публичным) ключом (например, сервер криптографических ключей). Причина появления открытого ключа в этом перечне не имеет значения. Например, человек мог создать ключи RSA только для покупок через Интернет, и может совершенно не знать, что его открытые ключи используются кем-то для создания кольцевой подписи на сообщении, которое он никогда не видел и не хотел подписывать[1]. Общий алгоритм кольцевой подписи допускает одновременное использование ключей, сформированных разными системами подписи с открытым ключом, в том числе имеющих разные размеры ключей и подписи[1].

При создании кольцевой подписи для сообщения **m** подписывающий по своему усмотрению формирует список из открытых ключей (P_1, P_2, ..., P_n), в который включает и свой под номером **i** (его порядковый номер в списке не имеет значения). Всё это вместе с секретным ключом подписывающего S_i как параметры подаётся на вход функции наложения подписи (m, S_i, P_1, ..., P_n), получая на выходе результат **σ**. Хотя каждому открытому ключу из списка соответствует свой уникальный закрытый ключ и используется только один из них (принадлежащий подписывающему), но по получившейся подписи невозможно узнать, какой именно из закрытых ключей использовался при её создании. Даже имея неограниченное число кольцевых подписей, выполненных одним и тем же подписантом, нет возможности его идентифицировать или гарантированно установить, что некоторые подписи были наложены с использованием одного закрытого ключа[1].

Подлинность кольцевой подписи можно проверить, используя **σ**, **m** и только открытые ключи P_1, ..., P_n.[5].

Варианты реализации

В своей статье Ривест, Шамир и Тауман описали кольцевую подпись как способ организовать утечку секретной информации без потери её достоверности. Например, кольцевая подпись «высокопоставленного чиновника Белого дома» не раскроет его личность, но гарантирует, что сообщение подписал кто-то из указанного списка официальных лиц, подтвердив таким образом уровень компетентности. При этом список лиц для кольцевой подписи можно легко составить, взяв публичные ключи из открытых источников[1].

Другое применение, также описанное авторами идеи, предназначено для создания *неоднозначных (спорных) подписей*. В простейшем случае для такого использования кольцевая подпись формируется на основе ключей отправителя и получателя сообщения. Тогда подпись значима для получателя, он уверен, что сообщение

создал отправитель. Но для постороннего человека такая подпись теряет убедительность и однозначность — не будет уверенности, кто именно сформировал и подписал сообщение, ведь это мог быть и сам получатель. Такая подпись, например, не может использоваться в суде для идентификации отправителя[1].

Позднее появились работы, в которых были предложены новые сферы применения кольцевых подписей и альтернативные алгоритмы их формирования[6].

Пороговые кольцевые подписи

В отличие от стандартной пороговой подписи «t-out-of-n», когда **t** из **n** пользователей должны сотрудничать для дешифрования сообщения, этот вариант кольцевой подписи требует, чтобы **t** пользователей сотрудничали в процессе подписания. Для этого **t** участников (i_1, i_2, ..., i_t) должны вычислить сигнатуру **σ** для сообщения **m**, подав **t** закрытых и **n** открытых ключей на вход (m, S_{i_1}, S_{i_2}, ..., S_{i_t}, P_1, ..., P_n)[7].

Связываемые кольцевые подписи

Свойство связности позволяет определить, были ли созданы какие-либо две кольцевые подписи одним и тем же человеком (использовался один и тот же закрытый ключ), но без указания, кто именно. Одним из возможных применений может быть автономная система электронных денег[8].

Прослеживаемая кольцевая подпись

В дополнение к связываемой подписи при повторном использовании может раскрываться открытый ключ подписанта. Такой протокол позволяет реализовать системы тайного электронного голосования, которые позволяют поставить анонимно только одну подпись, но раскрывают участника, проголосовавшего дважды[9].

Криптовалюты

Система CryptoNote допускает кольцевые подписи[10]. Впервые это было использовано в июле 2012 года в криптовалюте Bytecoin[11] [12] (не путать с Bitcoin).

Криптовалюта ShadowCash использует прослеживаемую кольцевую подпись для анонимности отправителя транзакции[13]. Однако первоначальная реализация была ошибочной, что привело к частичной де-анонимизации ShadowCash с первой реализации до февраля 2016 года[14].

Эффективность

Большинство предложенных алгоритмов имеют асимптотический размер результата $O(n)$, то есть размер результирующей подписи прямо пропорционален количеству использованных открытых ключей. Каждый использованный открытый ключ при наложении или проверке кольцевой подписи требует фиксированного объёма вычислений, что значительно лучше аналогов, имевшихся на момент создания протокола[1]. Например, технология CryptoNote реализует кольцевые подписи в платежах p2p, чтобы обеспечить анонимность отправителя[9].

В последнее время появились более эффективные алгоритмы. Существуют схемы с сублинейным размером подписи[15], а также с постоянным размером[16].

Алгоритм

Суть алгоритма кольцевой подписи, предложенной Ривестом, Шамиром и Тауман, состоит в следующем[1] (см. рисунок схемы).

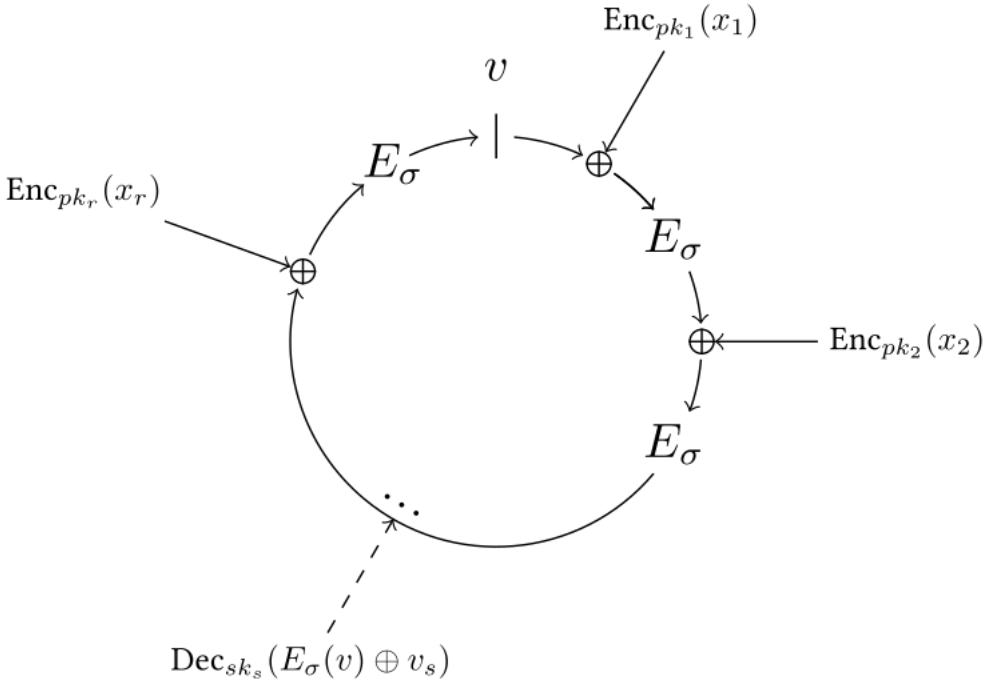


Схема кольцевой подписи

Кольцевая подпись для некоторого сообщения будет сформирована на основе списка из r открытых ключей (обозначены на схеме как pk_r), среди которых ключ подписывающего имеет порядковый номер s. Открытые ключи позволяют зашифровать произвольную информацию (информационный блок x_1, зашифрованный ключом pk_1, обозначен на схеме как $\mathrm{Enc}_{pk_1}(x_1)$). «*Информационные блоки*» здесь и далее не являются частью или результатом обработки подписываемого сообщения и не имеют самостоятельного значения, это сгенерированные случайные данные, становящиеся компонентами подписи.

Имеется *комбинационная функция* от произвольного количества аргументов, по значению которой и значениям всех аргументов, кроме одного, можно однозначно восстановить один недостающий аргумент. Примером такой функции является последовательное суммирование. Если известна итоговая сумма и все слагаемые, кроме одного, то недостающее слагаемое можно вычислить.

В качестве комбинационной функции авторы алгоритма предложили следующую последовательность действий: берётся некоторое стартовое значение (на схеме обозначено v, формируется случайно), над которым и первым аргументом выполняется побитовое исключающее «или» (обозначено на схеме символом ⊕). Затем к результату применяется определённое обратимое преобразование (обозначенное на схеме как E_σ), взаимно однозначно связанное с хеш-суммой подписываемого сообщения. Полученный результат участвует в операции побитового исключающего «или» со вторым аргументом, снова применяется преобразование E_σ, и так далее. В качестве аргументов используются зашифрованные открытыми ключами соответствующие информационные блоки $x_1, \ldots, x_r$.

Выбранное случайное значение v является одновременно и стартовым, и целевым (конечным) значением комбинационной функции: результат всех преобразований должен «пройти по кольцу» и стать равным начальному значению. Информационные блоки $x_1, \ldots, x_r$ для каждого из ключей, кроме блока x_s, соответствующего ключу самого подписывающего, задаются как случайные значения. Подписывающий шифрует информационные блоки соответствующими открытыми ключами. Теперь у подписывающего имеется целевое значение комбинационной функции и все аргументы, кроме одного — того, что соответствует его собственному ключу. Благодаря свойствам комбинационной функции, подписывающий может узнать недостающий аргумент и, использовав собственный закрытый ключ (sk_s), «расшифровать» этот аргумент (Dec_{sk_s}), получив недостающий информационный блок x_s.

Компоненты готовой кольцевой подписи[1]:

- подписываемое сообщение;
- список всех использованных открытых ключей;

- значения всех информационных блоков $x_1, \ldots, x_r$;
- значение комбинационной функции v.

Для проверки подписи нужно[1]:

- при помощи открытых ключей зашифровать информационные блоки;
- при помощи хеш-суммы подписываемого сообщения вычислить значение комбинационной функции, использовав как аргументы v (задающее стартовое значение) и зашифрованные информационные блоки (результаты предыдущего шага);
- убедиться, что полученное значение совпадает с v.

Реализация

Ниже приведена реализация описанного алгоритма на языке Python с использованием ключей RSA.

```
import os, hashlib, random, Crypto.PublicKey.RSA

class ring:
    def __init__(self, k, L=1024):
        self.k = k
        self.l = L
        self.n = len(k)
        self.q = 1 << (L - 1)

    def sign(self, m, z):
        self.permut(m)
        s = [None] * self.n
        u = random.randint(0, self.q)
        c = v = self.E(u)
        for i in (range(z+1, self.n) + range(z)):
            s[i] = random.randint(0, self.q)
            e = self.g(s[i], self.k[i].e, self.k[i].n)
            v = self.E(v^e)
            if (i+1) % self.n == 0:
                c = v
        s[z] = self.g(v^u, self.k[z].d, self.k[z].n)
        return [c] + s

    def verify(self, m, X):
        self.permut(m)
        def _f(i):
            return self.g(X[i+1], self.k[i].e, self.k[i].n)
        y = map(_f, range(len(X)-1))
        def _g(x, i):
            return self.E(x^y[i])
        r = reduce(_g, range(self.n), X[0])
        return r == X[0]

    def permut(self, m):
        self.p = int(hashlib.sha1('%s' % m).hexdigest(),16)

    def E(self, x):
        msg = '%s%s' % (x, self.p)
        return int(hashlib.sha1(msg).hexdigest(), 16)

    def g(self, x, e, n):
        q, r = divmod(x, n)
        if ((q + 1) * n) <= ((1 << self.l) - 1):
            rslt = q * n + pow(r, e, n)
        else:
            rslt = x
        return rslt
```

Подпись и проверка 2 сообщений при кольце из 4 пользователей:

```
size = 4
msg1, msg2 = 'hello', 'world!'

def _rn(_):
  return Crypto.PublicKey.RSA.generate(1024, os.urandom)

key = map(_rn, range(size))
```

```
r = ring(key)
for i in range(size):
    s1 = r.sign(msg1, i)
    s2 = r.sign(msg2, i)
    assert r.verify(msg1, s1) and r.verify(msg2, s2) and not r.verify(msg1, s2)
```

Примечания

1. *Рон Ривест, Ади Шамир, Yael Tauman.* How to leak a secret (https://people.csail.mit.edu/rivest/pubs/RST01.pdf) . — Volume 2248 of Lecture Notes in Computer Science. — Asiacrypt, 2001. — С. 552—565.
2. *И. Ю. Павловская.* Фоносемантический анализ речи. — Изд-во Санкт-Петербургского университета, 2001. — 290 с.
3. *Donald H. Dyal, Brian B. Carpenter, and Mark A. Thomas, eds.* Historical Dictionary of the Spanish American War.. — Westport, Conn.. — Greenwood Press, 1996.
4. *B. Schneier.* Прикладная криптография (http://htrd.su/wiki/_media/zhurnal/2012/03/23/todo_prikladnaja_kriptografija/cryptoshn.pdf) (рус.) // John Wiley & Sons. — 1996. — С. 98.
5. Debnath, Ashmita; Singaravelu, Pradheepkumar & Verma, Shekhar (19 December 2012), "Efficient spatial privacy preserving scheme for sensor network (http://dx.doi.org/10.2478%2Fs13531-012-0048-7)", *Central European Journal of Engineering* T. 3 (1): 1–10, DOI 10.2478/s13531-012-0048-7
6. *Lingling Wang, Guoyin Zhang, Chunguang Ma.* A survey of ring signature // Frontiers of Electrical and Electronic Engineering in China. — 2008. — Vol. 3, no. 1. — P. 10–19. — DOI:10.1007/s11460-008-0012-8 (https://dx.doi.org/10.1007%2Fs11460-008-0012-8).
7. E. Bresson; J. Stern & M. Szydlo (2002), "Threshold ring signatures and applications to ad-hocgroups (http://dx.doi.org/10.1007%2F3-540-45708-9_30)", *Advances in Cryptology: Crypto 2002*: 465–480, DOI 10.1007/3-540-45708-9_30
8. Liu, Joseph K. & Wong, Duncan S. (2005), "Linkable ring signatures: Security models and new schemes (http://dx.doi.org/10.1007%2F11424826_65)", *ICCSA* T. 2: 614–623, DOI 10.1007/11424826_65
9. *Eiichiro Fujisaki, Koutarou Suzuki.* Traceable Ring Signature (https://www.researchgate.net/publication/226082986_Sub-Linear_Size_Traceable_Ring_Signatures_without_Random_Oracles) (англ.) // Public Key Cryptography. — 2007. — P. 181–200.
10. CryptoNote Phylosophy (https://cryptonote.org/inside#untraceable-payments). CryptoNoteTech. Проверено 24 ноября 2017.
11. CryptoNote Currencies (https://cryptonote.org/coins) (англ.) (2015). Проверено 29 ноября 2017.
12. CryptoNote - убийца Bitcoin? (http://bits.media/cryptonote/) (23 июня 2014). Проверено 29 ноября 2017.
13. *Rynomster, Tecnovert.* HShadowCash: Zeroknowledge Anonymous Distributed ECash via Traceable Ring Signatures (https://bravenewcoin.com/assets/Whitepapers/ShadowCash-Zeroknowledge-Anonymous-Distributed-ECash.pdf) . — www.shadow.cash, 2015.
14. *Crypto Fun.* Broken Crypto in Shadowcash (https://shnoe.wordpress.com/2016/02/11/de-anonymizing-shadowcash-and-oz-coin/) (англ.) (недоступная ссылка — *история* (https://web.archive.org/web/*/https://shnoe.wordpress.com/2016/02/11/de-anonymizing-shadowcash-and-oz-coin/)) (13.02.2016). Проверено 29 ноября 2017. Архивировано (https://web.archive.org/web/20160927222205/https://shnoe.wordpress.com/2016/02/11/de-anonymizing-shadowcash-and-oz-coin/) 27 сентября 2016 года.
15. Fujisaki, Eiichiro (2011), "Sub-linear size traceable ring signatures without random oracles", *CTRSA*: 393–415
16. Au, Man Ho; Liu, Joseph K.; Susilo, Willy & Yuen, Tsz Hon (2006), "Constant-Size ID-Based Linkable and Revocable-iff-Linked Ring Signature (http://dx.doi.org/10.1007%2F11941378_26)", *Lecture Notes in Computer Science* T. 4329: 364–378, DOI 10.1007/11941378_26

Литература

- *Г. О. Чикишев.* Одноразовая кольцевая подпись и её применение в электронной наличности (http://mi.mathnet.ru/rus/pdma/y2012/i5/p56). — Приложение. — Журнал «Прикладная дискретная математика» № 5, 2012. — С. 56–58.
- *С. В. Запечников и другие* Кольцевые подписи и их приложения (http://cryptowiki.net/index.php?title=Кольцевые_подписи_и_их_приложения) в Энциклопедии теоретической и прикладной криптографии, МИФИ

Эта страница в последний раз была отредактирована 10 июня 2018 в 09:52.