Групповая Подпись на Основе Алгоритма Шнорра для Децентрализованных Систем

Александр Курбатов
Кафедра Безопасности Информационных Технологий
Харьковский национальный университет радиоэлектроники
Харьков, Украина
oleksandr.kurbatov@nure.ua

Геннадий Халимов
Кафедра Безопасности Информационных Технологий
Харьковский национальный университет радиоэлектроники
Харьков, Украина
gennadykhalimov@gmail.com

Group Signature Based on the Schnorr Algorithm for Decentralized Systems

Oleksandr Kurbatov
Information Security Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
oleksandr.kurbatov@nure.ua

Gennady Khalimov
Information Security Department
Kharkiv National University of Radio Electronics
Kharkiv, Ukraine
gennadykhalimov@gmail.com

***Аннотация*—в данной статье описываются основные шаги алгоритма подписи Шнорра, его улучшения и те преимущества, которые можно получить при использовании этой схемы подписи. Поскольку цифровые подписи в некотором роде определяют связь публичного и секретного ключа, выбор алгоритма цифровой подписи крайне важен для обеспечения конфиденциальности и эффективности.**

***Abstract*—This article describes the main steps of the Schnorr signature algorithm, its improvements and the benefits that can be achieved by using this signature scheme. Since digital signatures somehow determine the relationship between public and private keys, the choice of the digital signature algorithm is extremely important for ensuring privacy and efficiency.**

Ключевые слова—схема; алгоритм; цифровая подпись; Шнорр; мультиподпись.

Keywords—algorithm; digital signature; Schnorr; scheme; multisignature.

I. Введение

Децентрализованные системы активно используют алгоритм подписи ECDSA на кривой secp256k1 для проверки целостности и авторства данных. Алгоритм ECDSA стандартизирован, но имеет некоторые недостатки[1]:

- теоретически не доказана безопасность ECDSA;
- при использовании подписи ECDSA третья сторона без доступа к закрытому ключу, имеет возможность изменить существующую подпись для конкретного открытого ключа или создать подпись, которая действительна для того же открытого ключа и сообщения.

На текущий момент было предложено несколько различных вариантов замены ECDSA. Одним из альтернативных алгоритмов стал алгоритм цифровой подписи Шнорра.

II. Описание и особенности реализации алгоритма Шнорра

На данный момент предложено два варианта реализации подписи Шнорра. Первая реализация подписи представляет собой комбинацию (e, s) и верифицируется при помощи формулы [1]:

$$e = H(sG - eP \mid\mid m).$$

Такая реализация позволяет избавиться от сложности кодирования точки в значение подписи.

Второй вариант подписи определяется значениями (R,s) и верифицируется уравнением:

$$sG = R + H(R \mid\mid m)P.$$

Подобная схема позволяет реализовать простую схему пакетной подписи. При этом могут возникнуть некоторые проблемы, связанные с кодированием точки в значение подписи, однако существует несколько способов решить их. В данной работе подробнее будет рассмотрен второй вариант, так как он представляет больший интерес и полезность.

Параметрами алгоритма являются значения:

- значение модуля *p*, порядка кривой *n* и значение базовой точки *G* (генератора группы);
- параметры эллиптической кривой;
- *x(P)* и *y(P)* являются координатами точки *P* на эллиптической кривой.

Алгоритм подписи выглядит следующим образом:

1. Вычисляется $k = Hash(d \,||\, m) \bmod n$, где *d* - значение личного ключа.
2. Вычисляется значение $R = kG$.
3. Если $jacobi(y(R)) \neq 1$, то $k = n - k$.
4. Вычисляется $e = Hash(x(R)||dG||m) \bmod n$.

Значение подписи равно $sig = x(R)||(k + +ed \bmod n)$.

Алгоритм верификации подписи состоит из следующих шагов:

1. Проверяется соответствие открытого ключа заданной эллиптической кривой.
2. Вычисляется r как левые 32 байта подписи.
3. Значение s определяется правой частью значения подписи (32 байта).
4. Вычисляется $e = Hash(r||P||m) \bmod n$.
5. Если точка R не является точной на бесконечности, $jacobi(y(R)) \neq 1$ и $x(R) \neq r$.

III. Групповая подпись Шнорра

Помимо простой подписи, алгоритм подписи Шнорра позволяет реализовать групповую подпись [2]. При помощи интерактивной схемы (мультиподпись, пороговая подпись) участники могут создавать общую подпись и проверять ее с помощью заранее сгенерированного общего ключа. Это позволяет реализовать схему подписи n-of-n, которая с точки зрения верификатора ничем не будет отличаться от обычной одиночной подписи. Кроме того, при комбинации алгоритма подписи Шнорра и обязательств Педерсена возможно получить интерактивную схему пороговой подписи, которая гарантирует, что подписи могут быть созданы не только произвольными, но и заранее определенными наборами подписантов [3]. Таким образом, могут быть реализованы схемы типа m-of-n.

Групповая подпись, выработанная при помощи алгоритма Шнорра, обладает следующими характеристиками:

- подпись имеет один и тот же размер независимо от количества подписантов;
- избегается структурная утечка информации о ключах участников.

Алгоритм мультиподписи состоит из следующих шагов [2]:

1. Вычисляются $R_1 = r_1 G, R_2 = r_2 G, \ldots, R_i = r_i G$.
2. Вычисляется значение $R = R_1 + R_2 + \ldots + R_i$.
3. $s = s_1 + s_2 + \ldots + s_i, s_i = r_i + Hash(P, R_i, m) \cdot r_i$.
4. Полученные значения (R, s)- значение подписи.

Верификация мультиподписи происходит следующим образом:

1. Проверяется соответствие общего открытого ключа заданной эллиптической кривой.
2. Вычисляется r как левые 32 байта подписи.
3. Значение s определяется правой частью значения подписи (32 байта).
4. Вычисляется $e = Hash(r|| P || m) \bmod n$.
5. Если точка R не является точной на бесконечности, $jacobi(y(R)) \neq 1$ и $x(R) \neq r$.

Можно заметить, что алгоритм верификации мультиподписи ничем не отличается от алгоритма верификации одиночной. Поэтому даже сторона-верификатор не будет знать о факте проведения мультиподписи, не говоря уже о третьих сторонах.

Однако существует возможность мухлежа одной из подписывающих сторон, в случае проведения которого для верификации подписи понадобится ключ только одной из сторон. Рассмотрим ситуацию, когда два участника *A* и *B* формируют мультиподпись. При нормальном взаимодействии происходит следующее [2]:

1. *A* и *B* генерируют свои личные ключи *a* и *b* соответственно.
2. После этого происходит вычисление открытых ключей сторон, в результате имеются два публичных ключа $pA = aG, pB = bG$.
3. Участники формируют общий открытый ключ, который равен $pAB = pA + pB$. С помощью этого ключа в дальнейшем и будет происходить верификация.
4. После этого стороны обмениваются ключами и вырабатывается значение подписи. В результате мультиподпись равна $sigAB = sig(pAB)$. Свойства алгоритма Шнорра позволяют провести агрегацию значений ключей с обеспечением их линейности.
5. В результате подпись $sigAB$ возможно будет проверить только ключом pAB, при этом участники не знают о секретных ключах друг друга.

В случае мошенничества одной из сторон, происходит следующее:

1. *A* и *B* генерируют свои личные ключи *a* и *b* соответственно.
2. После этого происходит вычисление открытых ключей сторон, в результате имеются два публичных ключа $pA = aG, pB = bG$.
3. Сторона B вычисляет еще один открытый ключ, который равен $false\_pB = pB - pA$.
4. Стороны обмениваются открытыми ключами. Только в этом случае сторона *A* получает ложный ключ стороны *B*. В результате значение общего ключа равно $pAB = pA + pB - pA$, а значение мультиподписи равно $sigAB = sig(pA + pB - pA) = sig(pB)$. В этом случае получается что подпись была выработана только с использованием ключа *B*, и подтвердить свою подпись может только сторона *B*.

Данная схема атаки называется Rouge Key Attack Существует две схемы, позволяющие избежать такого рода атаки. К ним относится схемы Беллара-Ньювена и MuSig [2].

IV. Схема Беллара-Ньювена и MuSig

Такая схема не позволяет провести Rouge Key Attack, однако для верификации подписи требуются ключи всех сторон и становится возможно определить факт использования мультиподписи. Алгоритм подписи выглядит следующим образом [2]:

1. Вычисляется значение $L = H(X_1 + X_2 + \ldots + X_i)$.
2. Вычисляется $R = r_1 \cdot G + r_2 \cdot G + \ldots + r_3 \cdot G$.
3. $s = s_1 + s_2 + \cdots + s_i,\ s_i = r_1 + H(L, X_1, R, m) \cdot x_i$
4. **Полученные значения** (R, s)- значение подписи.

Верификация выполняется следующим образом:

$$s \cdot G = R + H(L, X_1, R, m) \cdot X_1 + \ldots + H(L, X_i, R, m) \cdot X_i.$$

Следующим решением, позволяющим избежать вероятности атаки Rouge Key и при этом использовать для проверки всего лишь один (общий) публичный ключ является MuSig [2]. Так при подписывании используются ключи всех подписантов, при проверке цифровой подписи используется агрегированный открытый ключ. В этом случае при использовании одного ключа верификация мультиподписи ничем не отличается от верификации одиночной. Схема выглядит следующим образом:

1. Вычисляется значение $L = H(X_1 + X_2 + \ldots + X_i)$.
2. $X = H(L, X_1) \cdot X_1 + H(L, X_2) \cdot X_2 + \ldots + H(L, X_2) \cdot X_2$.
3. Вычисляется $R = r_1 \cdot G + r_2 \cdot G + \ldots + r_i \cdot G$.
4. $s = s_1 + s_2 + \ldots + s_i,\ s_i = r_i + H(X, R, m) \cdot H(L, X) \cdot x_i$.
5. **Полученные значения** (R, s)- значение подписи.

Верификация мультиподписи, выработанной с помощью MuSig ничем не отличается от верификации одиночной подписи и при этом избегает угрозы проведения Rouge Key Attack.

На графике ниже (рис. 1) представлена зависимость времени верификации от количества подписываемых сообщений при помощи ECDSA.

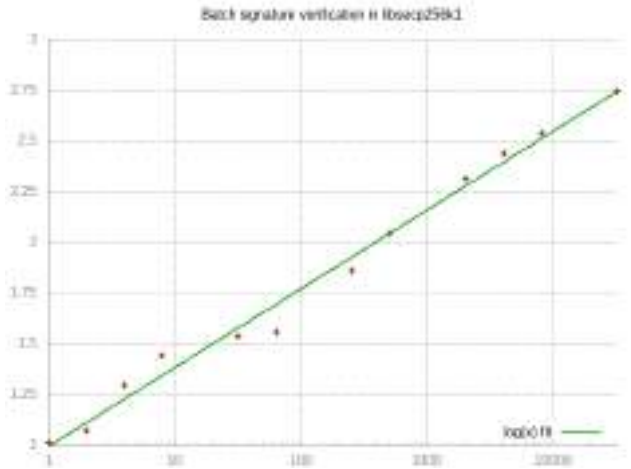

Рис. 1. График зависимости времени верификации от размера подписей в пакете

Можно заметить, что время верификации одиночных подписей по отдельности примерно совпадает с пакетным временем. Соответственно эффективность реализации пакетной подписи - минимальна.

На рис. 2 показано сравнение времени пакетной подписи при помощи ECDSA и с помощью алгоритма Шнорра в зависимости от размера подписи.

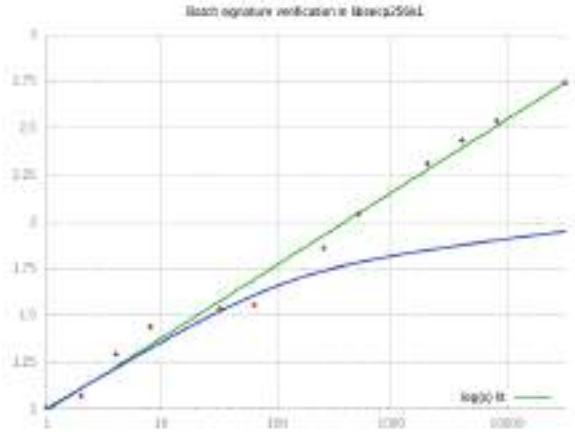

Рис. 2. График сравнения ECDSA и подписи Шнорра.

Из графика следует, что время пакетной подписи для алгоритма Шнорра (нижняя кривая) меньше, чем время, которое затрачивается при использовании алгоритма ECDSA.

V. Выводы

Так как в подписи Шнорра, можно использовать те же кривые, что и для ECDSA, то закрытые и открытые ключи идентичны для обеих схем. Это свойство позволяет избежать возникновениям гипотез относительно безопасности используемых кривых.

Подпись алгоритма Шнорра является подписью с теоретически доказанной безопасностью. Такого доказательства для ECDSA не существует. Подпись Шнорра позволяет существенно снизить время верификации для группы подписей.

Практическая реализация подписи Шнорра лежит в плоскости разработки технической спецификации. В июне 2018 года был обнародован проект, в котором излагается техническая реализация алгоритма [1]. На данном этапе проводится интенсивный анализ и изучение основных механизмов подписи Шнорра.

Литература References

[1] “BIP-Schnorr” [Online]. Available: https://github.com/sipa/bips/blob/bip-schnorr/bip-schnorr.mediawiki

[2] “Key aggregation for Schnorr signature” [Online]. Available: https://blockstream.com/2018/01/23/musig-key-aggregation-schnorr-signatures.html

[3] “Why Schnorr signatures will help solve 2 of Bitcoin’s biggest problems today” [Online]. Available: https://medium.com/@SDWouters/why-schnorr-signatures-will-help-solve-2-of-bitcoins-biggest-problems-today-9b7718e7861c