34 Доказательства с нулевым разглашением

34.1 Протокол Шнорра

В 1990 году немецкий математик и криптограф К. Шнорр предложил протокол, названный впоследствии в его честь. Шнорр использовал тогдашнюю терминологию и называл свой протокол *идентификационным*. В современной криптографии термин «идентификационный протокол» используется редко, протокол Шнорра принято относить к протоколам ZKP. Аббревиатура ZKP означает “Zero-Knowledge Proofs” — *доказательства с нулевым разглашением*.

Протокол Шнорра выполняют две стороны: P (prover) — доказывающий и V (verifier) — проверяющий. Стороны используют циклическую группу $\mathbb{G}$ порядка q с образующим G.

Сторона P публикует $Q \in \mathbb{G}$ и доказывает знание дискретного логарифма $d = \log_G Q$, $d \in \mathbb{Z}_q$. Доказывает, не раскрывая *никакой* информации о d. Отсюда «нулевое разглашение» в аббревиатуре ZKP.

Протокол Шнорра

Предназначен для доказательства знания $d = \log_G Q$

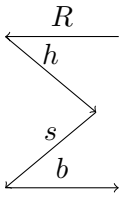
Стороны: P, V.
Шаги:

1. P: $k \xleftarrow{R} \mathbb{Z}_q$, $R \leftarrow kG$.
2. $P \to V$: R.
3. V: $h \xleftarrow{R} \mathbb{Z}_q$.
4. $V \to P$: h.
5. P: $s \leftarrow (k - hd) \bmod q$.
6. $P \to V$: s.
7. V: $b \leftarrow \mathbf{I}\{R = sG + hQ\}$.

Выход. Бит b является результатом выполнения протокола: при $b = 1$ протокол завершен успешно (V признает, что P знает d), при $b = 0$ — нет.
Полнота: $k = (s + hd) \bmod q \Rightarrow R = sG + hQ$. Мы говорим «полнота» вместо стандартного «корректность», чтобы соответствовать сложившейся в области ZKP терминологии (см. § 34.6).

Параметр R принято называть *обязательством* (commitment), параметр h — *запросом* или *вызовом* (challenge), параметр s — *ответом* (response).

Протокол Шнорра относится к классу Σ-протоколов. Буква Σ графически описывает пересылки между сторонами и возврат результата (P справа, V слева):



34.2 Неинтерактивный протокол Шнорра

В протоколе Шнорра P доказывает знание логарифма d *конкретному* проверяющему V, с которым он взаимодействует интерактивно. Можно ли доказать знание d сразу *всем* V, т. е. сделать протокол неинтерактивным? Оказывается, да. И сделать это довольно просто.

Активные действия V в ходе выполнения протокола — это генерация и передача запроса:

$$h \xleftarrow{R} \mathbb{Z}_q.$$

Этот запрос можно генерировать автоматически, без участия V, используя функцию хэширования $\varphi\colon \{0,1\}^* \to \mathbb{Z}_q$:

$$h \leftarrow \varphi(G, Q, R).$$

Как обычно, каждый из объектов G, Q, R кодируется двоичным словом. Конкатенация этих слов дает прообраз φ. Значения φ могут независимо вычислять P и V. Поэтому запрос можно не передавать.

Мы требуем, чтобы функция φ была криптографически стойкой. Это, в частности, означает, что образы φ трудноотличимы от случайных и тогда замена первого способа генерации h на второй также будет трудноотличима.

Замечание 34.1 (ROM). Идеальная функция φ — это функция, выбранная случайно равновероятно из множества допустимых. В криптографии такого рода функции принято называть *случайными оракулами*. Прообразы φ считаются вопросами оракулу, образы — соответствующими ответами. При повторе вопроса оракул повторяет свой предыдущий ответ, а для нового (*"свежего"*, fresh) вопроса выбирает ответ случайно равновероятно из множества допустимых. Анализ криптографических систем и протоколов существенно упрощается при замене используемых в них функций хэширования на случайных оракулов. В случаях замены говорят, что анализ проводится в *модели случайного оракула* (Random Oracle Model, ROM). □

Протокол Шнорра (неинтерактивный)

Шаги:

1. P: $k \overset{R}{\leftarrow} \mathbb{Z}_q$, $R \leftarrow kG$, $h \leftarrow \varphi(G, Q, R)$, $s \leftarrow (k - hd) \bmod q$.
2. $P \to V$: (R, s).
3. V: $h \leftarrow \varphi(G, Q, R)$, $R \overset{?}{=} sG + hQ$.

Преобразование интерактивного протокола в неинтерактивный через автоматизацию действий одной из сторон с помощью функций хэширования называется *преобразованием Фиата — Шамира* или *эвристикой Фиата — Шамира*.

Параметры G и Q функции φ задают контекст неинтерактивного протокола: в какой группе протокол выполняется и относительно какого элемента группы доказывается знание логарифма. Если контекст заранее определен, то эти параметры можно опустить. Или, наоборот, можно разрешить обрабатывать какие угодно контексты, потребовав только, чтобы они описывались словом $X \in \{0,1\}^*$. Теперь

$$h \leftarrow \varphi(X, R).$$

Доказательство знания дискретного логарифма относительно X — это подпись X. Мы преобразовали протокол Шнорра в систему ЭЦП Шнорра со следующими алгоритмами.

Алгоритм `Sign`

Вход: X — сообщение, d — личный ключ.

Выход: (h, s) — подпись.

Шаги:

1. $k \overset{R}{\leftarrow} \mathbb{Z}_q$, $R \leftarrow kG$, $h \leftarrow \varphi(X, R)$.
2. $s \leftarrow (k - hd) \bmod q$.
3. Возвратить (h, s).

Алгоритм `Vfy`

Вход: X, (h, s), Q — открытый ключ.

Выход: 0 или 1 — признак корректности подписи.

Шаги:

1. Если $h \notin \mathbb{Z}_q$ или $s \notin \mathbb{Z}_q$, то возвратить 0.
2. $R \leftarrow sG + hQ$.
3. Возвратить $\mathbf{I}\{h = \varphi(X, R)\}$.

Обратим внимание, что в неинтерактивном протоколе Шнорра аналогом подписи была пара (R, s). В алгоритмах `Sign` и `Vfy` вместо нее используется пара (h, s). Это более экономично, потому что для кодирования $R \in \mathbb{G}$ обычно требуется больше битов, чем для кодирования $h \in \mathbb{Z}_q$.

34.3 Протокол Шнорра на гомоморфизмах

Пусть $\mathbb{H}$ и $\mathbb{G}$ — аддитивные группы, в обеих групповая операция обозначается знаком $+$, и пусть $[\cdot]$ — гомоморфизм $\mathbb{H} \to \mathbb{G}$:

$$[a+b] = [a] + [b] \quad \forall a, b \in \mathbb{H}.$$

Будем говорить, что гомоморфизм $[\cdot]$ *эффективно вычислим* и *труднообратим*, если

1) по a можно эффективно вычислить $[a]$;

2) по $[a]$ вычислительно трудно найти a.

Чтобы упростить изложение, мы не даем строгих определений свойств 1) и 2), таких же строгих, как в определении односторонней функции.

Пример 34.1. Обращение гомоморфизма $\mathbb{Z}_q \to \mathbb{G}, d \mapsto dG$ — это задача DL в группе $\mathbb{G}$. Задача признается трудной, если $\mathbb{G}$ — группа точек эллиптической кривой, выбранная с учетом рассмотренных ранее правил. □

Пример 34.2. Пусть (n, e) — параметры RSA. Обращение гомоморфизма $\mathbb{Z}_n \to \mathbb{Z}_n, x \mapsto x^e \bmod n$ — это задача RSA. Она также признается трудной. □

Протокол Шнорра можно обобщить — использовать для доказательства знания прообразов произвольных эффективно вычислимых и труднообратимых гомоморфизмов.

Протокол Шнорра на гомоморфизмах

Предназначен для доказательства знания d для $Q = [d]$

Шаги:

1. P: $k \overset{R}{\leftarrow} \mathbb{H}$, $R \leftarrow [k]$.
2. $P \to V$: R.
3. V: $h \overset{R}{\leftarrow} \mathbb{Z}_q$, где q — порядок $\mathbb{H}$.
4. $V \to P$: h.
5. P: $s \leftarrow k - hd$ (вычисления в $\mathbb{H}$!).
6. $P \to V$: s.
7. V: $R \overset{?}{=} [s] + hQ$ (вычисления в $\mathbb{G}$!).

Полнота: $k = (s + hd) \Rightarrow [k] = [s + hd] = [s] + h[d] \Rightarrow R = [s] + hQ$.

Для неинтерактивных редакций данного протокола будем использовать следующую запись:

$$\mathrm{NIZKP}\{(Q; d) \colon Q = [d]\}.$$

В этой записи элемент Q называется *экземпляром* (instance), d — *свидетелем* (witness), равенство $Q = [d]$ — *утверждением* (statement).

Пример 34.3 (равенство логарифмов). Пусть G и H — различные образующие группы $\mathbb{G}$. Пусть для $Q_1 = dG$ и $Q_2 = dH$ требуется доказать равенство логарифмов:

$$\log_G Q_1 = \log_H Q_2.$$

Можно использовать протокол Шнорра с гомоморфизмом

$$\mathbb{Z}_q \to \mathbb{G} \times \mathbb{G}, \quad d \mapsto (dG, dH).$$

Пример 34.4 (неравенство логарифмов). Пусть теперь $Q_2 = d'H$, $d' \neq d$, и требуется доказать неравенство логарифмов:

$$\log_G Q_1 \neq \log_H Q_2.$$

Доказывающий может поступить следующим образом:

1) зная $d = \log_G Q_1$, вычислить $Q_2' = dH$ и предъявить его V;

2) дополнительно предъявить доказательство равенства $\log_G Q_1 = \log_H Q_2'$.

Сторона V проверяет доказательство и дополнительно проверяет, что $Q_2' \neq Q_2$. □

34.4 AND- и OR-композиции

Пусть $\mathbb{H}_i$ и $\mathbb{G}_i$ — аддитивные группы, $[\cdot]_i$ — гомоморфизм $\mathbb{H}_i \to \mathbb{G}_i$, $i = 1, 2, \ldots, n$. Будем считать, что порядки $\mathbb{H}_i$ и $\mathbb{G}_i$ делятся на большое простое q. Будем кодировать элементы групп двоичными словами и использовать функцию хэширования $\varphi\colon \{0,1\}^* \to \mathbb{Z}_q$.

Предыдущий протокол позволяет строить доказательства

$$\text{NIZKP}\{(Q_i; d_i)\colon Q_i = [d_i]_i\}, \quad i = 1, 2, \ldots,$$

в которых фигурируют утверждения $Q_i = [d_i]_i$. Покажем, как усложнить протокол и построить доказательства AND- и OR- композиций этих утверждений:

$$\text{NIZKP}\{(Q_1, \ldots, Q_n; d_1, \ldots, d_n)\colon Q_1 = [d_1]_1 \wedge Q_2 = [d_2]_2 \wedge \ldots \wedge Q_n = [d_n]_n\},$$
$$\text{NIZKP}\{(Q_1, \ldots, Q_n; d_1, \ldots, d_n)\colon Q_1 = [d_1]_1 \vee Q_2 = [d_2]_2 \vee \ldots \vee Q_n = [d_n]_n\}.$$

Доказательство AND-композиции — это совокупность доказательств базовых утверждений. Построить все эти доказательства можно следующим образом.

Протокол AND-композиция

Шаги:

1. P:
   (1) для $i = 1, 2, \ldots, n$: $k_i \xleftarrow{R} \mathbb{H}_i$, $R_i \leftarrow [k_i]$;
   (2) $h \leftarrow \varphi(Q_1, Q_2, \ldots, Q_n, R_1, R_2, \ldots, R_n)$;
   (3) для $i = 1, 2, \ldots, n$: $s_i \leftarrow k_i - hd_i$.
2. $P \to V$: $(h, s_1, s_2, \ldots, s_n)$.
3. V:
   (1) для $i = 1, 2, \ldots, n$: $R_i \leftarrow [s_i] + hQ_i$;
   (2) $h \stackrel{?}{=} \varphi(Q_1, Q_2, \ldots, Q_n, R_1, R_2, \ldots, R_n)$.

Доказательство OR-композиции сложнее. Здесь требуется, чтобы P знал прообраз d_i хотя бы для одного образа Q_i. Пусть речь идет о паре с номером j. Для остальных пар P строит псевдодоказательства знания прообраза. Важно, что по итоговому доказательству нельзя понять, номер какого прообраза известен P.

Протокол OR-композиция

Шаги:

1. P:
   (1) для $i = 1, 2, \ldots, n$, $i \neq j$: $h_i \xleftarrow{R} \mathbb{Z}_q$, $s_i \xleftarrow{R} \mathbb{G}_i$, $R_i \leftarrow [s_i] + h_i Q_i$;
   (2) $k_j \xleftarrow{R} \mathbb{H}_j$, $R_j \leftarrow [k_j]_j$;

(3) $h \leftarrow \varphi(Q_1, Q_2, \ldots, Q_n, R_1, R_2, \ldots, R_n)$;

(4) $h_j \leftarrow (h - \sum_{i \neq j} h_i) \bmod q$;

(5) $s_j \leftarrow k_j - h_j d_j$.

2. $P \to V$: $(h_1, h_2, \ldots, h_n, s_1, s_2, \ldots, s_n)$.

3. V:

(1) для $i = 1, 2, \ldots, n$: $R_i \leftarrow [s_i] + h_i Q_i$;

(2) $h \leftarrow \varphi(Q_1, Q_2, \ldots, Q_n, R_1, R_2, \ldots, R_n)$;

(3) $\sum_i h_i \stackrel{?}{\equiv} h \pmod q$.

34.5 Криптосистема ЭльГамаля

В 1985 г. Т. ЭльГамаль предложил две криптографические системы, стойкость которых основывается на сложности задачи DL: систему ЭЦП и криптосистему шифрования с открытым ключом. Систему ЭЦП ЭльГамаля мы уже рассмотрели, перейдем к системе шифрования.

Криптосистема задается 4 алгоритмами: `Setup`, `Gen`, `Enc`, `Dec`. Алгоритм `Setup` строит циклическую группу $\mathbb{G}$ большого простого порядка q и выбирает в ней образующий G. Остальные алгоритмы определяются ниже.

Алгоритм Gen

Вход: $\perp$.

Выход: $d \in \mathbb{Z}_q$ — личный ключ, $Q \in \mathbb{G}$ — открытый ключ.

Шаги:

1. $d \stackrel{R}{\leftarrow} \mathbb{Z}_q$, $Q \leftarrow dG$.
2. Возвратить (d, Q).

Алгоритм Enc

Вход: $M \in \mathbb{G}$ — открытый текст, Q.

Выход: $(C_1, C_2) \in \mathbb{G} \times \mathbb{G}$ — шифртекст.

Шаги:

1. $r \stackrel{R}{\leftarrow} \mathbb{Z}_q$.
2. $(C_1, C_2) \leftarrow (M + rQ, rG)$.
3. Возвратить (C_1, C_2).

Алгоритм Dec

Вход: (C_1, C_2), d.

Выход: M.

Шаги:

1. $M \leftarrow C_1 - dC_2$.
2. Возвратить M.

Скаляр r в алгоритме `Enc` называется *рандомизатором*. Удобно считать, что вызов `Enc` состоит в генерации r с последующим обращением к функции

$$E(M, r) = (M + rQ, rG).$$

Функция E является гомоморфизмом $\mathbb{Z}_q \times \mathbb{G} \to \mathbb{G} \times \mathbb{G}$:

$$E(M, r) + E(M', r') = E(M + M', r + r').$$

Наличие гомоморфизма является, вообще говоря, слабостью криптосистемы ЭльГамаля, которая не позволяет выдержать все гарантии безопасности. Однако во многих случаях не все эти гарантии нужны и, наоборот, гомоморфное свойство оказывается чрезвычайно полезным.

Криптосистема ЭльГамаля часто используется в системах электронного голосования как раз из-за гоморфного свойства. Примерный сценарий использования:

1. Голосование состоит в выборе из двух вариантов: 1 (за) и 0 (против).
2. Избиратель с номером i выбирает вариант $m_i \in \{0, 1\}$, вычисляет $M_i = m_i G$ и зашифровывает его на ключе избирательной комиссии с помощью рандомизатора r_i. В результате получается зашифрованный бюллетень $E(M_i, r_i)$, который избиратель публикует в системе голосования.

3. По окончании голосования зашифрованные бюллетени суммируются:

$$C = \sum_i E(M_i, r_i) = E\left(\sum_i M_i, \sum_i r_i\right).$$

4. Комиссия расшифровывает C и публикует результат

$$M = \sum_i M_i = mG, \quad m = \sum_i m_i.$$

Если число избирателей невелико, то сумма m также невелика и может быть определена «грубой силой».

Обратим внимание, что индивидуальные бюллетени не раскрываются в системе. Если ограничить комиссию однократным расшифрованием суммы бюллетеней (это можно сделать с помощью дополнительных протоколов на основе разделения секрета), то индивидуальные бюллетени не будут известны и комиссии.

Описанный сценарий голосования не является до конца безопасным. Требуются дополнительные механизмы ZKP. Мы их рассмотрим ниже.

1. *Доказательство корректности зашифрования.* Избиратель зашифровывает M и получает $C = (C_1, C_2) = (M + rQ, rG)$. Требуется доказать, что зашифрование выполнено корректно: избиратель знает M и r, пара (M, r) корректно встроена в C.

   Решение. Использовать гомоморфизм $(M, r) \mapsto (M + rQ, rG) = (C_1, C_2)$.

2. *Доказательство корректности выбора.* При зашифровании mG избиратель должен доказать, что $m \in \{0, 1\}$.

   Решение. Использовать OR-композицию доказательств знания прообраза одного из двух гомоморфизмов:

$$r \mapsto (rQ, rG) = (C_1 - 0G, C_2),$$
$$r \mapsto (rQ, rG) = (C_1 - 1G, C_2).$$

3. *Доказательство корректности расшифрования (избиратель).* Избирателю может потребоваться доказать, что $C = (C_1, C_2) = (M + rQ, rG)$ является результатом зашифрования M. При доказательстве избиратель не должен раскрыть r.

   Решение. Использовать гомоморфизм $r \mapsto (rQ, rG) = (C_1 - M, C_2)$.

4. *Доказательство корректности расшифрования (комиссия).* Комиссия расшифровывает $C = (C_1, C_2) = (M + rQ, rG)$. Комиссия должна доказать, что M действительно является результатом расшифрования C.

   Решение. Использовать гомоморфизм $d \mapsto (d(rG), dG) = (dC_2, dG) = (C_1 - M, Q)$.

34.6 Свойства протоколов ZKP

Пусть Σ — протокол доказательства с нулевым разглашением. Пусть P доказывает V утверждение, связанное со знанием секретного свидетеля d. Для этого P отправляет обязательство R, получает от V запрос h и дает на него ответ s. Сторона V проверяет ответ и возвращает признак успеха $b = \Sigma(P, V)$. Тройка (R, h, s) называется *стенограммой* (transcript) протокола.

Каждая из сторон P и V может быть *честной* (honest) или *нечестной* (dishonest). Нечестная сторона неверно выполняет шаги протокола, пытаясь навязать некорректный результат или нарушить нулевое разглашение. Например, P доказывает знание d, на самом деле его не зная. Или V определяет d.

Протокол Σ признается надежным, если он обладает следующими свойствами.

1. *Полнота (completeness)*: если P и V оба честные, то $\mathsf{P}\{\Sigma(P, V) = 1\} = 1$.

2. *Корректность (soundness)*: если P — нечестный (не знает d), V — честный, то $\mathsf{P}\{\Sigma(P,V)=1\} \approx 0$ (пренебрежимо мала).

3. *Нулевое разглашение (zero knowledge)*: если P — честный, то для любого V (честного или нечестного) стенограмма (R,h,s) не содержит информации о d.

Вероятности здесь определяются случайными лентами, которые P и V используют в своей работе.

Для многих протоколов полноту и нулевое разглашение трудно обосновать или проверить. Оказывается удобным работать со следующими редакциями свойств.

4. *Специальная корректность (special soundness)*: существует полиномиальный вероятностный алгоритм $\mathcal{E}$ (knowledge extractor), который принимает на вход две утвердительные стенограммы (R,h,s) и (R,h',s'), $h \neq h'$, и с высокой вероятностью определяет по ним d. Здесь утвердительная стенограмма — это стенограмма, на которой честный V завершает протокол с результатом $\Sigma(P,V)=1$.

   Логика. По утвердительным стенограммам можно эффективно определить d и, следовательно, сторона P, которая участвует в выпуске утвердительных стенограмм, знает d, т. е. является честной.

   Обратим внимание, что в стенограммах повторяется обязательство R, это принципиальный момент. Стенограммы нужной структуры можно получить, если дважды запустить сеанс протокола с P, повторяя случайную ленту P и не повторяя запросы. Идет речь о перезапуске (rewind) P как ПВМТ.

5. *Нулевое разглашение относительно честного V (honest verifier zero knowledge, HVZK)*: существует полиномиальный вероятностный алгоритм $\mathcal{S}$ (simulator), который без использования d строит стенограммы (R,h,s), статистически неотличимые от настоящих (полученных в сеансах двух честных сторон).

   Логика. Коль скоро стенограммы можно построить без секретного свидетеля, они действительно не содержат информации о нем.

Замечание 34.2. Свойство HVZK иногда уточняется: симулятор $\mathcal{S}$ должен построить стенограмму с фиксированным запросом h фиксирован. В таким случаях говорят о специальном нулевом разглашении. □

Пример 34.5 (свойства протокола Шнорра). Мы с самого начала показали, что протокол Шнорра обладает полнотой. Покажем, что он также обладает двумя последними свойствами.

Специальная корректность. Пусть (R,h,s), (R,h',s') — утвердительные стенограммы с $h \neq h'$. Пусть $R = kG$. Тогда

$$s = (k - hd) \bmod q, \quad s' = (k - h'd) \bmod q$$

и

$$d = (h' - h)^{-1}(s - s') \bmod q.$$

Нулевое разглашение относительно честного V. Алгоритм $\mathcal{S}$ генерирует стенограммы следующим образом:

$$s \xleftarrow{R} \mathbb{Z}_q, \quad h \xleftarrow{R} \mathbb{Z}_q, \quad R \leftarrow sG + hQ.$$

Стенограмма является реализацией случайной величины с равномерным распределением на множестве

$$\{(R,h,s) \in \mathbb{G} \times \mathbb{Z}_q \times \mathbb{Z}_q \colon R = sG + hQ\}$$

У настоящих стенограмм такое же распределение вероятностей. □