Лекция 5

Протоколы привязки к биту (bit commitment)

(Конспект: Ф. Парт)

Одной из задач криптографии является поиск способа передать какую-либо информацию так, чтобы для получателя она оставалась не читаемой до момента получения им ключа, но он был бы уверен в том, что не существует ключа, с помощью которого он прочел бы не то, что подразумевалось отправителем изначально. Иначе говоря, одна из сторон (отправитель) должна уметь посылать бит упакованный в «сундучок», открывающийся только при помощи ключа, придерживаемого отправителем до поры, до времени при себе. Это имеет смысл только в том случае, если отправитель не сможет подменить бит в «сундучке», то есть не существует двух ключей, открывающих «сундучок» по разному.

Протоколы, реализующие этот механизм, называются протоколами привязки к биту.

5.1 Неинтерактивные протоколы

В этом разделе мы рассмотрим простейший *неинтерактивный* вариант протокола привязки к биту.

Определение 5.1. Протоколом привязки к биту называется пара полиномиальных алгоритмов S (вероятностный) и R (детерминированный), таких что:

1. $S : (\sigma, 1^n) \mapsto s_n(\sigma), k_n(s_n)$ кодирует бит σ с заданной степенью надежности n и генерирует ключ для его декодирования.

2. Случайные величины $s_n(0)$ и $s_n(1)$ вычислительно неотличимы. (Это условие гарантирует, что без ключа сундучок не открыть.)

3. R корректно декодирует бит по правильному ключу и отказывается от декодирования если ключ неверен:

$$R(1^n, k_n(s_n), s_n(\sigma)) = \sigma$$

$\forall k\ R(1^n, k, s_n(\sigma)) \in \{\sigma, \perp\}$

Второе условие также гарантирует, что разными ключами сундучок нельзя открыть по-разному.

Теорема 5.1. *Если существуют односторонние перестановки (сохраняющие длину биекции), то существует протокол привязки к биту.*

Доказательство. Пусть g - односторонняя перестановка и h ее трудный бит. Рассмотрим протокол:

1. Алгоритм кодирования $S(\sigma, 1^n)$:
   - сгенерировать случайную строку $r_n \leftarrow U_n$
   - выдать $s_n = (\sigma \oplus h(r_n))g(r_n)$ и $k_n = r_n$
2. Алгоритм декодирования $R(1^n, k, by)$: если $g(k) \neq y$, то выдать $\perp$, иначе выдать $h(k) \oplus b$.

Докажем, что это протокол привязки к биту. Очевидно, R на ключе k_n работает корректно. Так как g - перестановка, существует единственный прообраз подстроки y из n последних битов кодовой строки by, следовательно все прочие ключи R отвергает. Осталось проверить, что случайные величины $s_n(0) = h(r_n)g(r_n)$ и $s_n(1) = (1 \oplus h(r_n))g(r_n)$ вычислительно неотличимы. Пусть они отличимы. Тогда одна из них отличима от U_{n+1}, а значит и каждую из них можно отличить. Следовательно $h(r_n)g(r_n)$ отличима от U_{n+1} - противоречие. □

5.2 Интерактивные протоколы

Неформально, интерактивный протокол $c(A, B)$ - это схема взаимодействия двух алгоритмов A и B, в ходе которого они обмениваются сообщениями. Точнее, A и B поочередно генерируют сообщения в зависимости от истории их взаимодействия, т.е. вычисляют функции: $A(c_1 \dots c_{2k}) = c_{2k+1}$, $B(c_1 \dots c_{2k-1}) = c_{2k}$. Протокол завершается, если один из них сгенерирует пустое сообщение $c_n = \emptyset$. Будем обозначать $view(c(A, B))$ множество сгенерированных сообщений протокола $c(A, B)$.

Дадим определение интерактивного протокола привязки к биту. Пусть теперь привязка осуществляется посредством общения полиномиальных вероятностных алгоритмов $S(1^n, \sigma)$(отправителя) и $R(1^n)$(получателя) в соответствии с некоторым интерактивным протоколом $c(S, R)$. После этого алгоритм S генерирует ключ $k_n = k_n(view(c(S, R))$ и он вместе с $view(c(S, R))$ подается на вход детерминированному алгоритму T, призванному вычислить значение бита σ или установить, что одна из сторон вела себя нечестно и указать какая именно.

Определение 5.2. Интерактивным протоколом привязки к биту называется тройка полиномиальных алгоритмов S, R - вероятностные и T - детерминированный, удовлетворяющих условиям:

1. Пусть S и R честно следовали интерактивному протоколу на стадии привязки. Тогда если S сгенерировал правильный ключ k_n, то алгоритм T правильно вычисляет σ. В противном случае T уличает S в жульничестве (выдает $\perp_S$).

$$T(view(c(S(1^n, \sigma), R(1^n))), k_n) = \sigma$$

$$\forall k\ T(view(c(S(1^n, \sigma), R(1^n))), k) \in \{\sigma, \perp_S\}$$

2. Никакой полиномиальный вероятностный алгоритм R^* на стороне получателя не сможет добиться того, чтобы при честном S алгоритм T неправильно распаковал бит или обвинил S:

$$\forall R^*\ T(view(c(S(1^n, \sigma), R^*(1^n))), k_n) \in \{\sigma, \perp_R\}$$

3. С вероятностью близкой к 1 никакой полиномиальный вероятностный алгоритм S^* на стороне отправителя не сможет организовать диалог с честным R так, чтобы T по разному распаковывал бит при различных ключах или обвинил R. Т.е. $\forall S^*$ с вероятностью близкой к 1 выполнено:

   - $T(view(c(S^*(1^n, \sigma), R(1^n))), k) \neq \perp_R$
   - $\neg\exists k_0,\ k_1$:

$$T(view(c(S^*(1^n, \sigma), R(1^n))), k_0) = 0$$

$$T(view(c(S^*(1^n, \sigma), R(1^n))), k_1) = 1$$

4. Ни для какого полиномиального вероятностного алгоритма R^* протокол общения с S без ключа не несет существенной информации о бите, т.е. случайные величины $view(c(S(1^n, 0), R^*(1^n)))$ и $view(c(S(1^n, 1), R^*(1^n)))$ вычислительно неотличимы.

Теорема 5.2. *Если существует генератор псевдослучайных чисел, то существует интерактивный протокол привязки к биту.*

Доказательство. Пусть $G : \{0,1\}^n \to \{0,1\}^{3n}$ - генератор псевдослучайных чисел. Так как G неотличима от $G \oplus r$ для $\forall r \in \{0,1\}^{3n}$, то алгоритм $S(1^n, \sigma)$ мог бы выдавать в качестве привязки $G(s) \oplus (\sigma \cdot r)$, где $s \leftarrow U_n$, $r \neq 0^{3n}$, а в качестве ключа $k_n = s$ (тут $\sigma \cdot r$ — это умножение бита σ на строку r). Это обеспечило бы вычислительную неотличимость $s_n(0)$ от $s_n(1)$ и дало бы возможность получателю восстановить бит по ключу k_n и привязке y: $\sigma = 0$ если $G(k_n) \oplus y = 0$ и $\sigma = 1$ если $G(k_n) \oplus y = r$. Единственная проблема заключается в том, что если окажется, что $\exists\ s_0, s_1\ G(s_0) = G(s_1) \oplus r$, то S разными ключами $k_0 = s_0, k_1 = s_1$ сможет добиться разного декодирования привязки. Детерминировано вычислять хорошее r довольно сложно. Однако, если его выбирать на стороне R случайно и посылать S, то вероятность того, что r окажется плохим не больше:

$$Pr_{r \leftarrow U_{3n}}[\exists\ s_0, s_1 | G(s_0) = G(s_1) \oplus r] = Pr_{r \leftarrow U_{3n}}[\exists\ s_0, s_1 | G(s_0) \oplus G(s_1) = r] \leq \frac{2^{2n}}{2^{3n}} = \frac{1}{2^n}$$

Таким образом, имеем следующий интерактивный протокол привязки:

1. Алгоритм $R(1^n)$: отослать $r \leftarrow U_{3n}$
2. Алгоритм $S(1^n, \sigma, r)$:
   - Если длина r меньше, чем $3n$, то завершить протокол
   - Сгенерировать $s \leftarrow U_n$ и послать в качестве привязки $G(s) \oplus (\sigma * r)$. В качестве ключа выдать $k_n = s$.

Определение алгоритма T, а также выполнение всех условий определения протокола привязки к биту очевидно. $\square$