КРИПТОГРАФИЧЕСКИ СТОЙКИЕ ГЕНЕРАТОРЫ

1. Понятие генератора

$(k, l(k))$-Генератором называется функция $f : (Z_2)^k \to (Z_2)^{l(k)},\ l(k) > k$, где $l(X)$ – некоторый полином, значение которой можно вычислить за полиномиальное относительно k время (рассматривается семейство функций при растущем *параметре безопасности* k). Генератор f называется *псевдослучайным*, если строка $f(s),\ s \in \{0,1\}^k$ при случайном выборе s практически неотличима от строки той же длины (это свойство уточняется ниже), выбираемой случайно из множества $\{0,1\}^{l(k)}$. Тогда значение функции $f(s)$ называют - псевдослучайной битовой строкой, а значение $s \in (z_2)^k$ ее *зерном,* Таким образом, при случайном выборе зерна строка $f(s)$ должна выглядеть как случайный набор бинарных символов. В данной лекции изучаются псевдослучайные генераторы.

2. Полиномиально неразличимые вероятностные распределения

Уточним понятие псевдослучайного $(k, l(k))$-генератора как генератора $f : \{0,1\}^k \to \{0,1\}^{l(k)}$ такого, что при достаточно большим параметре безопасности k при случайном выборе зерна s строку битов $f(s)$ нельзя отличить от случайного набора двоичных знаков за полиномиальное время. Такие генераторы в криптографической литературе называются *криптографически стойкими* псевдослучайными генераторами. Сокращенно их принято называть просто псевдослучайными генераторами.

Сначала поясним, как мы понимаем полиномиальную неразличимость вероятностных распределений.

Пусть p_0 и p_1 два – вероятностных распределения на множестве бинарных строк длины $l(k)$, $\mathbf{A} : (Z_2)^{l(k)} \to \{0,1\}$ – вероятностный алгоритм, Такой алгоритм может быть, например, вероятностной машиной Тьюринга. полиномиальный относительно $l(k)$, ϵ – константа, $\epsilon > 0$. Определим математические ожидания условных вероятностей $p_j(\mathbf{A}(z_1, \dots z_{l(k)}) = 1 | (z_1, \dots z_{l(k)})),\ j \in \{0,1\}$ при распределениях p_0 и p_1 на множестве $(Z_2)^{l(k)}$

$$E_{\mathbf{A}}(p_j(\mathbf{A}(z_1, \dots z_{l(k)}) = 1 | (z_1, \dots z_{l(k)}))) =$$

$$= \sum_{(z_1, \dots z_{l(k)}) \in (z_2)^{l(k)}} p_j(z_1, \dots, z_{l(k)}) \times p(\mathbf{A}(z_1, \dots z_{l(k)}) = 1 | (z_1, \dots z_{l(k)})), j \in \{0,1\}.$$

Следует обратить внимание, что эти средние значения выхода алгоритма A при условии, что на его вход поступают определенным образом распределенные наборы, зависят от параметра безопасности k, поскольку условная вероятность в этой формуле зависит не только от случайных величин, используемых при работе алгоритма, но и от от выбранной строки $(z_1, \dots z_{l(k))})$ и ее длины $l(k)$.

Примечание. Если алгоритм A детерминированный, то условная вероятность

$$p(\mathbf{A}(z_1, \dots z_l(k)) = 1|(z_1, \dots z_{l(k)}))$$

принимает только два значения 0 или 1. Вероятностный алгоритм «догадывается» о значении j и величина $E_{\mathbf{A}}(p_j)$ определяет среднее значение выхода алгоритма при условии, что на его вход поступают наборы в соответствии с распределением вероятностей $p_j,\ j = 0, 1$.

Алгоритм $\mathbf{A}$ называется *ϵ-различителем* для p_0 и p_1, если

$$|E_{\mathbf{A}}(p_0) - E_{\mathbf{A}}(p_1)| \geq \epsilon.$$

Распределения p_0 и p_1 называются ϵ-различимыми, если для них существует ϵ-различитель.

Пусть p_0 – равномерное распределение на множестве $(Z_2)^{l(k)}$, а p_1 - распределение значения $f(s) \in (Z_2)^{l(k)}$ при равновероятном выборе зерна $s \in (Z_2)^k$. (Предположим для простоты, что при этом все выбираемые значения различны). Распределение p_1 очень далеко от равномерного: хотя 2^k двоичных наборов из $(Z_2)^{l(k)}$ выбираются с равными вероятностями $1/2^k$, остальные $2^{l(k)} - 2^k$ наборов не выбираются никогда.

Несмотря на то, что распределения p_0 и p_1 могут сильно различаться, можно тем не менее установить, что они в некоторых случаях при достаточно больших значениях параметра безопасности k ϵ-различимы только при малых ϵ, таких что $\epsilon < \frac{1}{p(k)}$, где $p(X)$ – *любой* полином от переменной X над полем действительных чисел. Тогда говорят, что вероятностные распределения p_0 и p_1 *полиномиально неразличимы.*

Формально это понятие описывается предикатом

$$\forall p(X) \in R[X]\ \exists N\ \forall k \geq N\ |E_{\mathbf{A}}(p_0) - E_{\mathbf{A}}(p_1)| \geq \epsilon \to \epsilon < \frac{1}{p(k)}.$$

Здесь абсолютная разность математических ожиданий зависит от параметра безопасности k и с его ростом становится исчезающе малой. Если же распределения p_0 и p_1 ϵ-различимы при $\epsilon \geq \frac{1}{p(k)}, hboxp(X)$ – *некоторый* полином над полем действительных чисел, то эти распределения называются *полиномиально различимыми.* Формально:

$$\exists p(X) \in R[X] \forall N \ \exists k \geq N \ \ |E_{\mathbf{A}}(p_0) - E_{\mathbf{A}}(p_1)| \geq \epsilon \geq \frac{1}{p(k)}.$$

Приведем пример полиномиально различимых вероятностных распределений.

Пример 1 *Пусть генератор $f(k, l(k))$ производит двоичные наборы четной длины с одинаковым числом нулей и единиц.*

Будем использовать полиномиальный от $l(k)$ (детерминированный) алгоритм $\mathbf{A}$, реализующий функцию

$$\mathbf{A}(z_1, \ldots, z_{l(k)}) = \begin{cases} \mathit{1}, & \text{если в наборе } (z_1, \ldots, z_{l(k)}) \ l(k)/2 \text{ нулей}, \\ \mathit{0}, & \text{в остальных случаях.} \end{cases}$$

В этом случае

$$E_{\mathbf{A}}(p_0) = \frac{\binom{l(k)}{l(k)/2}}{2^{l(k)}}$$

и

$$E_{\mathbf{A}}(p_1) = 1.$$

Так что

$$|E_{\mathbf{A}}(p_0) - E_{\mathbf{A}}(p_1)| = |1 - \frac{\binom{l(k)}{l(k)/2}}{2^{l(k)}}| \geq \frac{1}{2}.$$

Таким образом, такой генератор псевдослучайным не является. Не являются псевдослучайными и генераторы, значениями которых являются начальные отрезки длины $l(k)$ линейных рекуррентных последовательностей или начальные отрезки бинарных последовательностей, образуемых младшими разрядами ЛКП.

Понятие псевдостлучайного (криптографически стойкого) генератора связано с понятием *односторонней функции.*

Односторонние функции f определяются в классе функций $f_k : Z_2^k \rightarrow Z_2^{l(k)}$, где $l(X)$ – некоторый полином.

Функция называется *честной,* если существует полином $q(X)$, такой, что $k \leq q(l(k))$. Это означает, что такая функция не слишком сильно «сжимает» входные значения. Честная функция f_k называется *односторонней,* если

1) Существует полиномиальный алгоритм (алгоритм, исполняющий не более $P(k)$ элементарных операций при вычислении значения функции, $P(X)$ есть некоторый полином) , вычисляющий ее значение $f(x)$ при любом $x \in \{0,1\}^k$.

2) Для любого полиномиального вероятностного алгоритма A и случайно выбранной строки $x \in_R Z_2^k$ и любого полинома $P(x)$ при достаточно больших значениях параметра безопасности k

$$E(p(f(A(f(x))) = f(x)|x)) = \sum_{x \in Z_2^k} 2^{-k} p(f(A(f(x))) = f(x)|x) < 1/P(k).$$

(См.[1].)

Заметим, что данное определение не исключает существования полиномиального алгоритма вычисления прообраза для исчезающе малой доли значений функции.

Ясно, что псевдослучайный генератор должен быть односторонней функцией. Импальяццо, Левин и Луби, а также Хостад в 1989-1990 г.г. доказали, что существование односторонней функции также и достаточно для существования криптографически стойкого генератора.

Теорема 1 *Криптографически стойкие генераторы существуют тогда и только тогда, когда существуют односторонние функции.*

Существование односторонних функций (а значит и псевдослучайных генераторов) не доказано. Практически используются функции, "обращение"которых эквивалентно трудным проблемам теории чисел, например, проблеме факторизации и проблеме квадратичного вычета.

3. Предсказатель следующего бита

Пусть f есть $(k, l(k))$-генератор. Пусть имеется некоторый вероятностный алгоритм $\mathbf{B}_i : (Z_2)^{i-1} \to \{0,1\}$, который, получая первые $i-1$ двоичных символов $(z_1, z_2, \ldots, z_{i-1})$, формируемых генератором f, предсказывает значение i-го символа z_i. Алгоритм определён на множестве $\{0,1\}^{i-1}$ всех двоичных наборов длины $i-1$.

Такой алгоритм $\mathbf{B}_i$ называется *ϵ-предсказателем следующего бита* при заданном $(k, l(k))$-генераторе , если при равновероятном выборе зерна s он определяет значение i-элемента строки $f(s)$ по первым $i-1$ ее элементам с вероятностью не менее $1/2 + \epsilon$, при $\epsilon > 0$.

Теорема 2 *При заданном $(k, l(k))$-генераторе алгоритм $\mathbf{B}_i$ является ϵ-предсказателем следующего бита тогда и только тогда, когда*

$$\sum_{(z_1,\ldots,z_{i-1}) \in (Z_2)^i} p_1(z_1, \ldots, z_{i-1}) \times p(\mathbf{B}(z_1, \ldots, z_{i-1}) = z_i|(z_1, \ldots, z_{i-1})) \geq \frac{1}{2} + \epsilon.$$

Здесь условная вероятность определяется выбором строки $(z_1, \ldots, z_{i-1})$ и случайными величинами при работе вероятностного алгоритма.

Если распределение двоичных наборов длины $i-1$ равномерно, то любой вероятностный предсказывающий алгоритм определит следующий бит с вероятностью, равной $1/2$. Если распределение вероятностей отличается от равномерного, то появляется возможность создания предсказателя с большей вероятностью угадывания,

Предсказатель $\mathbf{B}_i$ следующего бита можно использовать в различающем алгоритме $\mathbf{A}$ в качестве подпрограммы по следующей схеме.

ВХОД: бинарный набор $z_1, \dots z_{l(k)}$. ВЫХОД: $\mathbf{A}(z_1, \dots z_{l(k)}) \in \{0, 1\}$ Вычислить $z = \mathbf{B}_i(z_1, \dots z_{i-1})$ Если $z = z_i$, то $\mathbf{A}(z_1, \dots, z_{l(k)}) = 1$ иначе $\mathbf{A}(z_1, \dots z_{l(k)}) = 0$.

(0.1)

Теорема 3 *Пусть $\mathbf{B}_i$ – ϵ-предсказатель следующего бита для $(k, l(k))$-генератора f. Пусть p_1 – вероятностное распределение на множестве двоичных наборов $(Z_2)^{l(k)}$, порождаемое генератором f, а p_0 – равномерное распределение вероятностей на этом множестве. Тогда различающий алгоритм $\mathbf{A}$ из (0.1) является ϵ-различителем для p_1 и p_0.*

Доказательство. Заметим. что

$$\mathbf{A}(z_1, \dots, z_{l(k)}) = 1 \iff \mathbf{B_i}(z_1, \dots, z_{i-1}) = z_i.$$

Кроме того выход алгоритма $\mathbf{A}$ не зависит от $z_{i+1}, \dots z_{l(k)}$.

Поэтому можно вычислить

$$E_{\mathbf{A}}(p_1) = \sum_{(z_1, \dots z_{l(k)}) \in (Z_2)^{l(k)}} p_1(z_1, \dots z_{l(k)}) \times p(\mathbf{A} = 1 | (z_1, \dots z_{l(k)})) =$$

$$= \sum_{(z_1, \dots z_i) \in (Z_2)^i} p_1(z_1, \dots z_i) \times p(\mathbf{A} = 1 | (z_1, \dots z_i)) =$$

$$= \sum_{(z_1, \dots z_{i-1}) \in (Z_2)^{i-1}} p_1(z_1, \dots z_{i-1}) \times p(\mathbf{B}_i(z_1, \dots z_{i-1}) = z_i | (z_1, \dots z_{i-1})).$$

По теореме 2

$$E_{\mathbf{A}}(p_1) \geq \frac{1}{2} + \epsilon.$$

С другой стороны, каждый предсказатель B_i предсказывает i-ый бит строки из равномерно распределенного множества $\{0, 1\}^{l(k)}$ с вероятностью $\frac{1}{2}$. Не трудно видеть поэтому, что $E_{\mathbf{A}}(p_0) = \frac{1}{2}$. Отсюда $|E_{\mathbf{A}}(p_0) - E_{\mathbf{A}}(p_1)| \geq \epsilon$.

Обратим внимание, что и в данном случае абсолютная разность средних значений зависит от параметра безопасности k.

ϵ-Предсказатель следующего бита называется *полиномиальным*, если $\epsilon \geq \frac{1}{p(k)}$, где $p(x)$ – некоторый полином.

Следствие 1 *Вероятностные распределения p_0 и p_1 полиномиально различимы тогда и только тогда, когда для распределения p_1 существует полиномиальный ϵ-предсказатель следующего бита.*

Главным результатом теории генераторов является то, что предсказатель следующего бита является *универсальным* тестом. То есть такой генератор является криптографически стойким тогда и только тогда, когда не существует полиномиального ϵ-предсказателя следующего бита.

Необходимость определяется предыдущей теоремой, а достаточность утверждается следующей.

Теорема 4 *Если существует ϵ-различитель вероятностного распределения p_1 на множестве двоичных наборов $(Z_2)^{l(k)}$, индуцируемых $(k, l(k))$-генератором f, и равномерного распределения p_0 на том же множестве двоичных наборов , то для некоторого $i, 1 \leq i \leq l(k)$, существует полиномиальный $\epsilon/l(k)$-предсказатель следующего бита $\mathbf{B_i}$ для f.*

Доказательство. Определим вероятностное распределение $q_i, 0 \leq i \leq l(k)$ на $(Z_2)^2$ такое, что первые i бит производятся генератором f, а остальные $l(k) - i$ двоичных знаков выбираются случайно. Тогда $q_0 = p_0$ и $q_{l(k)} = p_1$. Мы имеем

$$|E_{\mathbf{A}}(q_0) - E_{\mathbf{A}}(q_{l(k)})| \geq \epsilon.$$

Учитывая неравенство треугольника, получим

$$|E_{\mathbf{A}}(q_0) - E_{\mathbf{A}}(q_{l(k)})| \leq \sum_{i=1}^{l(k)} |E_{\mathbf{A}}(q_{i-1}) - E_{\mathbf{A}}(q_i)|.$$

Отсюда хотя бы для одного значения $i \leq i \leq l(k)$ выполняется

$$|E_{\mathbf{A}}(q_{i-1}) - E_{\mathbf{A}}(q_i)| \geq \epsilon/l(k).$$

Не нарушая общности, будем считать. что

$$E_{\mathbf{A}}(q_{i-1}) - E_{\mathbf{A}}(q_i) \geq \epsilon/l(k). \qquad (0.2)$$

(Если

$$E_{\mathbf{A}}(q_i) - E_{\mathbf{A}}(q_{i-1}) \geq \epsilon/l(k),$$

то выход предсказателя просто инвертируется и получается предыдущий случай)

Построим предсказатель i-го бита $\mathbf{B}_i$ в виде следующего вероятностного алгоритма.

ВХОД: бинарный набор $(z_1, \dots z_{i-1})$.
ВЫХОД: $\mathbf{B}_i(z_1 \dots z_{i-1}) \in \{0, 1\}$.
Выбрать случайно набор $(z_i, z_{i+1}, \dots z_{l(k)}) \in (Z_2)^{l(k)-i+1}$.
Вычислить $z = \mathbf{A}(z_1, \dots z_{l(k)})$.
Определить $\mathbf{B}_i(z_1, \dots, z_{i-1}) = (z + z_i) \bmod 2$.

(0.3)

Логику алгоритма (0.3) можно пояснить следующим образом. Генератор формирует двоичный набор длины $l(k)$ в соответствии с распределением вероятностей q_{i-1}. Ответ 0 алгоритма $\mathbf{A}$ означает, что алгоритм считает, что этот набор скорее всего был образован в соответствии с распределением вероятностей q_i. (Это следует из условия (0.2)) Распределения q_{i-1} и

q_i отличаются только способом порождения i-го бита: в q_{i-1} он индуцируется случайно, а в q_i – $(k, l(k))$-генератором. Следовательно, если $\mathbf{A}$ отвечает 0, то он предполагает, что z_i должен быть произведен $(k, l(k))$-генератором, а при ответе 1, что он случаен. В первом случае i-ый бит предсказывается как z_i, а во втором – как $1 \oplus z_i$.

Вычислим вероятность правильного предсказания i-го бита. Очевидно, что при ответе "0"алгоритма $\mathbf{A}$, предсказание корректно с вероятностью

$$p_1(z_i|(z_1, \ldots, z_{i-1})),$$

где p_1 есть вероятностное распределение, порождаемое генератором f. При ответе "1"вероятность правильного предсказания есть

$$1 - p_1(z_i|(z_1, \ldots, z_{i-1})).$$

Будем сокращенно обозначать $\mathbf{z} = (z_1, \ldots, z_{l(k)})$. Заметим, что

$$q_{i-1}(\mathbf{z}) \times p_1(z_i|(z_1, \ldots, z_{i-1})) = \frac{q_i(\mathbf{z})}{2}.$$

Действительно,

$$q_{i-1}(z_1, \ldots, z_{l(k)}) \times p_1(z_i|(z_1, \ldots, z_{i-1})) =$$
$$= q_{i-1}(z_1, \ldots, z_{i-1}) \times \frac{1}{2^{l(k)-i+1}} \times p_1(z_i|(z_1, \ldots, z_{i-1})) =$$
$$= q_i(z_1, \ldots, z_{i-1}, z_i) \times \frac{1}{2^{l(k)-i+1}} = \frac{q_i(z_1, \ldots, z_{l(k)})}{2}.$$

Теперь можно вычислить вероятность правильного предсказания.

$$p(z_i = \mathbf{B}_i(z_1, \ldots, z_{i-1})) =$$
$$= \sum_{\mathbf{z} \in (z_2)^{l(k)}} q_{i-1}(\mathbf{z})[p(\mathbf{A} = 0|\mathbf{z}) \times p_1(z_i|(z_1, \ldots, z_{i-1})) +$$
$$+ p(\mathbf{A} = 1|\mathbf{z}) \times (1 - p_i(z_1|(z_1, \ldots, z_{i-1})))] =$$
$$= \sum_{\mathbf{z} \in (Z_2)^{l(k)}} \frac{q_i(\mathbf{z})}{2} \times p(\mathbf{A} = 0|\mathbf{z}) + \sum_{\mathbf{z} \in (Z_2)^{l(k)}} q_{i-1}(\mathbf{z}) \times p(\mathbf{A} = 1|\mathbf{z}) -$$
$$- \sum_{\mathbf{z} \in (Z_2)^{l(k)}} \frac{q_i(\mathbf{z})}{2} \times p(\mathbf{A} = 1|\mathbf{z}) =$$
$$= \sum_{\mathbf{z} \in (Z_2)^{l(k)}} \frac{q_i(\mathbf{z})}{2} \times p(\mathbf{A} = 0|\mathbf{z}) + \sum_{\mathbf{z} \in (Z_2)^{l(k)}} q_{i-1}(\mathbf{z}) \times p(\mathbf{A} = 1|\mathbf{z}) +$$
$$+ \sum_{\mathbf{z} \in (Z_2)^{l(k)}} \frac{q_i(\mathbf{z})}{2} \times p(\mathbf{A} = 1|\mathbf{z}) - \sum_{\mathbf{z} \in (Z_2)^{l(k)}} q_i(\mathbf{z}) \times p(\mathbf{A} = 1|\mathbf{z}) =$$
$$= \sum_{\mathbf{z} \in (Z_2)^{l(k)}} \frac{q_i(\mathbf{z})}{2} \times (p(\mathbf{A} = 0|\mathbf{z}) + p(\mathbf{A} = 1|\mathbf{z}) + E_{\mathbf{A}}(q_{i-1}) - E_{\mathbf{A}}(q_i) =$$

$$= \frac{1}{2} + E_{\mathbf{A}}(q_{i-1}) - E_{\mathbf{A}}(q_i) \geq \frac{1}{2} + \epsilon/l(k).$$

4.Стойкие криптографические системы

Пусть $d_k(m)$ - шифртекст длины $l(n)$, полученный зашифрованием симметричной криптографической системой открытого текста m той же длины на ключе k длины n. Для ее криптоанализа используется полиномиальный вероятностный алгоритм A, получающий на входе шифртекст d и вырабатывающий пару $(i, \sigma),\ i = 1, 2, ..., l(n),\ \sigma \in \{0, 1\}$.

Криптосистема называется *стойкой,* если при равномерном выборе ключа $k \in_R \{0, 1\}^n$ и равномерном выборе открытого текста $m \in_R \{0, 1\}^{l(n)}$ для любого полинома $P(X)$ и всех достаточно больших n

$$p(A(d) - (i, \sigma) \& \sigma = m_i) | m \in_R \{0, 1\}^{l(n)} < \frac{1}{2} + \frac{1}{P(n)}.$$

Эта вероятность также зависит и от случайных величин, выбираемых алгоритмом A в процесса работы.

Ясно, что стойкие криптосистемы существуют тогда и только тогда, когда существуют псевдослучайные генераторы.

Действительно, стойкой является криптосистема формирующая $d = M \oplus f(s)$ при равномерном выборе зерна s в качестве ключа k (здесь $\oplus$-операция поразрядного сложения битовых строк длины $l(k)$). И наоборот, при случайном выборе ключа k и фиксированном открытом тексте m криптограмма d может рассматриваться как значение $f(k)$ генератора.

Следствие 2 *Криптосистема является стойкой тогда и только тогда, когда при случайном выборе ключа и при достаточно большой длине $l(n)$ криптограмма d полиномиально неотличима от случайно выбираемой битовой строки той же длины $l(n)$.*

Примером криптографически стойкой криптосистемы является криптосистема Блюма-Гольдвассер, использующая BBS-генератор.

5. Криптографическая стойкость BBS-генератора

BBS-генератор. Рассмотрим называемый Blum-Blum-Shub - генератор (BBS-генератор) и докажем его криптографическая стойкость в предположении, что проблема квадратичного вычета не имеет полиномиального алгоритма решения. Предположение о существовании полиномиального ϵ-различителя для вероятностного распределения, индуцируемого таким генератором, и для равномерного распределения приводит к противоречию с общепринятым мнением о не существовании полиномиального алгоритма для этой проблемы. Напомним эту проблему: является ли число a, имеющее равный 1 символ Якоби по модулю составного числа $n = p \times q$, (p и q – простые числа) квадратичным вычетом по модулю n? Иными словами, является это число квадратом некоторого числа $a \in Z_n^*$ или же оно есть псевдо-квадрат по модулю n. (В первом случае символы Лежандра чисел p и q равны 1, а во втором они равны -1).

BBS-генератор это $(k, l(k)-$генератор, осуществляющий вычисления по следующему алгоритму.

ВХОД: k, l

ВЫХОД: псевдослучайное двоичное число $(z_1, z_2, \dots z_l)$ длины l.

Сформировать два секретных и разных простых числа p и q

длиной $k/2$ бит, конгруэнтных 3 по модулю 4.

Вычислить $n = p \cdot q$.

Выбрать случайное число $r \in [1, n-1]$

взаимно простое с числом n (НОД$(r, n) = 1$.

Вычислить "зерно"$s = r^2 \bmod n$ и принять $x_0 = s$

Для $i = \overline{1, l}$ выполнять

$\quad x_i = x_{i-1}^2 \bmod n$.

$\quad z_i = x_i \bmod 2$.

Как видим,

$$z_i = (s^{2^i} \bmod n) \bmod 2,\ 1 \le i \le l.$$

Рассмотрим пример конкретного BBS-генератора [1].

Пусть $n = 192649 = 383 \times 503$, $r = 101355$ и $s = 101355^2 \bmod n$=20749. Первые 20 битов, производимые этим BBS-генератором представления в следующей таблице;

i	x_i	z_i	i	x_i	z_i	i	x_i	z_i	i	x_i	z_i
0	20749	1									
1	143135	1	6	80649	1	11	137922	0	16	133015	1
2	177671	1	7	45663	1	12	123175	1	17	106065	1
3	97048	0	8	69442	0	13	8630	0	18	45870	0
4	89992	0	9	186894	0	14	114386	0	19	137171	1
5	174051	1	10	177046	0	15	14863	1	20	48060	0

Используемые в BBS-генераторе составные числа вида $n = pq$, где p и q – различные простые числа, конгруэнтные 3 по модулю 4 называются *числами Блума.*

Напомним, что множество квадратичных вычетов Q_n по модулю n есть множество чисел $a, a \in Z_n^*$ таких, что в Z_n^* имеется число x такое, что $a = x^2 \bmod n$.)

Рассмотрим и докажем следующие свойства чисел Блума 1)$\left(\frac{-1}{p}\right) = \left(\frac{-1}{q}\right) = -1$, следовательно, $\left(\frac{-1}{n}\right) = 1$;

2)Для $y \in Z_n^*$, если $\left(\frac{y}{n}\right) = 1,$то либо $y \in Q_n$, либо $y \in \bar{Q}_n$,

3)каждый вычет$y \in Q_n$ имеет 4 квадратных корня$v, -v, u, -u,$такие, что

а)$\left(\frac{u}{p}\right) = 1$, $\left(\frac{u}{q}\right) = 1$, то есть $v, u \in Q_n$;

б)$\left(\frac{-u}{p}\right) = 1$, $\left(\frac{-u}{q}\right) = 1$;

в)$\left(\frac{v}{p}\right) = -1$, $\left(\frac{v}{q}\right) = 1$;

г)$\left(\frac{-v}{p}\right) = 1$, $\left(\frac{-v}{q}\right) = -1$.

4) функция $f(x) = x^2 \pmod n$ есть перестановка на Q_n;

Доказательство. 1) Используем критерий Эйлера $\left(\frac{u}{p}\right) = u^{\frac{p-1}{2}}$. Получим $\left(\frac{-1}{p}\right) = -1^{\frac{4k+3-1}{2}} = -1^{2k-1} = -1$. Аналогично, $\left(\frac{-1}{q}\right) = -1$. Отсюда $\left(\frac{-1}{n}\right) = 1$.

2)$\left(\frac{y}{n}\right) = 1$ влечет $\left(\frac{y}{p}\right)=\left(\frac{y}{q}\right) = 1$ или $\left(\frac{y}{p}\right)=\left(\frac{y}{q}\right) = -1$. В первом случае $y \in Q_n$, во втором случае $-y \in Q_n$

3)Обозначим $\pm u$ и $\pm v$ 4 квадратных корня из $y \in Q_n$. Только один из них (тот, который имеет символы Лежандра 1 как по модулю p, так и по модулю q) принадлежит Q_n.

4) Следует из 3.

Таким образом, по свойству 3) если n – число Блума, то каждое число $a \in Q_n$ имеет точно 4 квадратных корня по модулю n, точно один из них принадлежит Q_n. Указанный единственный корень, принадлежащий Q_n называется *главным* корнем.

Свойство 4), что преобразование $x_i = x_{i-1}^2$ является перестановкой на множестве Q_n квадратичных вычетов и используется для обоснования криптографической стойкости BBS-генератора.

Предсказатель предыдущего бита. Предсказатель предыдущего бита для $(k, l(k))$-BBS генератора, получая на входе $l(k)$ псевдослучайных двоичных знаков, выработанных генератором при неизвестном предсказателю "зерне"s, пытается угадать (предсказать) значение $z_0 = s \bmod 2$. предсказатель предыдущего бита может быть вероятностным алгоритмом. Предсказатель $\mathbf{B}_0$ предыдущего бита называется *ϵ-предсказателем предыдущего бита*, если вероятность правильного угадывания значения z_0 не менее $1/2+\epsilon, \epsilon > 0$, при вычислении этой вероятности по всем возможным "зернам"s.

Следующая теорема, аналогичная теореме 11.4 приводится без доказательства.

Теорема 5 *Пусть* $\mathbf{A}$ *является ϵ-различителем вероятностных распределений p_1 и p_0, где p_1 – распределение, индуцируемое на $(Z_2)^{l(k)}$ $(k, l(k))$-BBS генератором f, а p_0 - случайное равномерное распределение на этом же множестве. Тогда существует $\epsilon/l(k)$-предсказатель предыдущего бита для f.*

ϵ-предсказатель $\mathbf{B}_0$ предыдущего бита можно использовать для построения вероятностного алгоритма B, который отличает квадратичный вычет по модулю n от псевдоквадрата по модулю n с вероятностью $1/2+\epsilon$, $\epsilon > 0$. Алгоритм B использует B_0 как подпрограмму-оракула и имеет вид:

ВХОД: $x \in Z_n^*$ такое, что $\left(\frac{x}{n}\right) = 1$.
ВЫХОД: ответ "$x \in Q_n$" или "$x \in \tilde{Q}_n$".
Вычислить $s = x^2 \bmod n$ и вычислить $z_0 = s \bmod 2$.
С помощью BBS-генератора вычислить $z_1, \ldots, z_{l(k)-1}$
для данного "зерна"s.
Вычислить $z = \mathbf{B}_0(z_0, \ldots, z_{l(k)-1})$.
Если ($x \bmod 2$)$=z$, то ответ = "$x \in Q_n$" иначе ответ = "$x \in \tilde{Q}_n$."

Теорема 6 *Пусть* $\mathbf{B}_0$ *есть предсказатель предыдущего бита для* $(k, l(k))$*-BBS генератора* f. *Тогда приведённый алгоритм* $\mathbf{B}$ *определяет, является ли* x *квадратичным вычетом, правильно с вероятностью не менее* $1/2 + \epsilon,\ \epsilon > 0$, *при вычислении этой вероятности по всем возможным входам* $x \in Q_n \cup \tilde{Q}_n$.

Доказательство. Поскольку $n = pq$ и $p \equiv q \equiv 3 \bmod 4$, то $\left(\frac{-1}{n}\right) = 1$, так что $-1 \in \tilde{Q}_n$. Следовательно, если

$$\left(\frac{x}{n}\right) = 1,$$

то главный корень числа $s = x^2$ есть x, если $x \in Q_n$ и $-x$, если $x \in \tilde{Q}_n$. Но

$$(-x \bmod n) \bmod 2 \neq (x \bmod n) \bmod 2,$$

Отсюда следует, что алгоритм $\mathbf{B}$ даёт правильный ответ тогда и только тогда, когда $\mathbf{B}_0$ правильно предсказывает z. Отсюда немедленно следует заключение теоремы.

Вероятностные алгоритмы для проблемы квадратичного вычета. Рассмотренная теорема показывает, как можно бы было различать псевдоквадраты и квадратичные вычеты с вероятностью не менее $1/2 + \epsilon,\ \epsilon > 0$, если бы существовал ϵ-предсказатель предыдущего бита.

Это приводит к алгоритму Монте-Карло, дающему правильный ответ с вероятностью не менее $1/2 + \epsilon,\ \epsilon > 0$: для любого $x \in Q_n \cup \tilde{Q}_n$ приведенный ниже алгоритм Монте-Карло $\mathbf{Q}$ даст правильный ответ с вероятностью не менее $1/2 + \epsilon,\ \epsilon > 0$. При этом алгоритм может ошибаться в обе стороны (является не предвзятым, unbiased). Алгоритм $\mathbf{Q}$ использует предыдущий алгоритм $\mathbf{B}$ в качестве подпрограммы.

ВХОД: $x \in Z_n^*$, такое, что $\left(\frac{x}{n}\right) = 1$.
ВЫХОД: ответ "$x \in Q_n$" или "$x \in \tilde{Q}_n$".
Выбрать случайное число $r \in Z_n^*$. Вычислить с вероятностью $1/2$
    $x' = r^2 x \bmod n$
или
    $x' = -r^2 x \bmod n$.
Вычислить $\mathbf{B}(x') \in \{Q, \tilde{Q}\}$.
Если
    $\mathbf{B}(x') = Q$ и $x' = r^2 x \bmod n$
или
    $\mathbf{B}(x') = \tilde{Q}$ и $x' = -r^2 x \bmod n$,
то ответ = "$x \in Q_n$"
иначе ответ = "$x \in \tilde{Q}_n$".

Теорема 7 *Если алгоритм* $\mathbf{B}$ *определяет, является ли* x *квадратичным вычетом, правильно с вероятностью не менее* $1/2 + \epsilon$, $\epsilon > 0$ *то алгоритм Монте-Карло* $\mathbf{Q}$ *решает проблему квадратичного вычета с вероятностью ошибки не более* $1/2 - \epsilon$.

Доказательство. Для каждого заданного входа $x \in Q_n \cup \tilde{Q}_n$ случайно выбирается элемент x', о котором известно, является он квадратичным вычетом или псевдоквадратом. Это позволяет принять решение о статусе элемента x с вероятностью ошибки не более $1/2 - \epsilon$.

Осталось показать, как использовать алгоритм, решающий проблему квадратичного вычета с вероятность ошибки не более $1/2 - \epsilon$ для построения алгоритма, решающего эту проблему со сколь угодно малой вероятностью ошибки δ.

Идея построения такого алгоритма состоит в том, чтобы принимать решения по результатам $2m + 1$ "прогонов"этого базового алгоритма на основе мажоритарного принципа. При этом необходимо установить зависимость величин ϵ, m и δ.

Теорема 8 *Пусть алгоритм Монте Карло* $\mathbf{Q}$, *вероятность ошибки которого не превышает* $1/2 - \epsilon$, $\epsilon > 0$, *применяется* $t = 2m + 1$ *раз при одних и тех же исходных данных* I *и в качестве окончательного результата выбирается наиболее часто встречающийся его ответ. Тогда вероятность ошибки итогового алгоритма не превышает*

$$\frac{(1 - 4\epsilon^2)^m}{2}.$$

Доказательство. Вероятность получения i правильных ответов при t испытаниях не превышает

$$\binom{n}{i} \left(\frac{1}{2} + \epsilon\right)^i \left(\frac{1}{2} - \epsilon\right)^{t-i}.$$

Вероятность того. что наиболее частый ответ окажется неправильным, равна вероятности того, что число правильных ответов в t испытаниях не превысит m. Следовательно, эту вероятность ошибки $p_{\text{ош}}$ можно вычислить следующим образом.

$$p_{\text{ош}} \le \sum_{i=0}^{m} \binom{n}{i} \left(\frac{1}{2}+\epsilon\right)^{i} \left(\frac{1}{2}-\epsilon\right)^{2m+1-i} =$$

$$= \left(\frac{1}{2}+\epsilon\right)^{m} \left(\frac{1}{2}-\epsilon\right)^{m+1} \sum_{i=0}^{m} \binom{n}{i} \left(\frac{1/2-\epsilon}{1/2+\epsilon}\right)^{m-i} \le$$

$$\le \left(\frac{1}{2}+\epsilon\right)^{m} \left(\frac{1}{2}-\epsilon\right)^{m+1} \sum_{i=0}^{m} \binom{n}{i} =$$

$$= \left(\frac{1}{2}+\epsilon\right)^{m} \left(\frac{1}{2}-\epsilon\right)^{m+1} 2^{2m} =$$

$$= \left(\frac{1}{4}-\epsilon^{2}\right)^{m} \left(\frac{1}{2}-\epsilon\right) 2^{2m} =$$

$$= (1-4\epsilon^2)^m \left(\frac{1}{2}-\epsilon\right) \le$$

$$\le \frac{(1-4\epsilon^2)^m}{2},$$

что и требуется.

Допустим, что требуется понизить вероятность ошибки до некоторого значения δ, $0 < \delta < 1/2-\epsilon$. Мы должны выбрать m так, чтобы выполнялось неравенство

$$\frac{(1-4\epsilon^2)^m}{2} \le \delta,$$

Достаточно взять

$$m = \left\lceil \frac{1+\log_2 \delta}{\log_2(1-4\epsilon^2)} \right\rceil.$$

Таким образом, если алгоритм $\mathbf{A_1}$ используется $2m+1$ раз, то голосование ответов приводит к ошибке с вероятностью не более δ. Можно показать. что значение m при этом не превышает $c/(\delta\epsilon^2)$, где c – некоторая константа. Таким образов число "прогонов"алгоритма $\mathbf{A}$ полиномиально относительно $1/\delta$ и $1/\epsilon$.

Пример 2 *Пусть мы имеем алгоритм Монте-Карло, дающий правильный ответ с вероятностью 0,55, то есть* $\epsilon = 0,05$. *Если требуется принимать решения с вероятностью ошибки не более 0,05, то достаточно принять* $m = 230$ *и* $t = 461$.

Заключение. Таким образом, предположение о существовании ϵ-предсказателя предыдущего бита приводит к заключению о возможности построения полиномиального вероятностного алгоритма для проблемы квадратичного вычета, что противоречит современным представлениям о сложности этой проблемы. Противоречие свидетельствует о криптографической стойкости BBS-генератора. Его производительность можно повысить, используя в каждой итерации $m \leq \log_2 \log_2$ младших двоичных знаков текущего значения x_1. Например, при $n \approx 10^{160}$ можно использовать до 9 младших двоичных знаков.

Контрольные вопросы

1. Какой генератор битовых строк называется криптографически стойким?

2. Какие вероятностные распределения называются полиномиально неразличимыми?

3. Как связаны понятия *epsilon*-предсказателя следующего бита и *epsilon*-различителя вероятностных распределений

4. Дайте определение криптографически стойкой криптосистемы, как это понятие связано с понятием псевдослучайного генератора.

5. Как построить BBS-генератор. Как обосновать криптографическую стойкость BBS-генератора?

6. Каким образом алгоритм Монте Карло позволяет понижать вероятность ошибки в определении свойства быть квадратичным вычетом?

Литература

1. Введение в криптографию.
Под ред. В.В.Ященко. – М: МЦНМО-Черо, 1998.

1.Stinson D.R. Cryptography: theory and practice. – CRC Press LLC, Boca Raton, 1995.

2. Menezes A.J., van Oorschoft P., Vanstone S.A. Handbook of Applied Cryptography. – CRC Press, Boca Raton, New York, London, Tokio, 1997.