Шпаргалка по докладу на теорсеминаре по статье «The First and Fourth Public-Key Cryptosystems with Worst-Case/Average-Case Equivalence»

Олейников Иван

17 декабря 2018 г.

1 Предварительные определения

Определение 1 (Криптосистема с публичным ключом). *Криптосистемной с публичным ключом называется тройка вероятностных полиномиальных по времени алгоримов* (G, E, D).

Алиса и Боб могут использовать такую криптосистему для секретной передачи сообщений.

Прежде, чем криптосистемой можно будет шифровать сообщения, нужно сгенерировать пару из публичного и секретного ключей: $(pk, sk) \leftarrow G(1^n)$*, здесь* $n \in \mathbb{N}$ *— это параметр безопасности, от него завясит время работы алгоритмов криптосистемы и трудность взлома криптосистемы. Генерацией ключей занимается получатель сообщений — Алиса, публичный ключ она раздаёт всем, кто в будущем захочет отправлять ей шифрованные сообщения, а секретный оставляет себе и держит в тайне.*

Отправитель Боб, имеющий открытый ключ pk*, может зашифровать сообщение* $b_0 \dots b_l$ *шифрующим алгоритмом:* $x \leftarrow E(pk, b_0 \dots b_l)$*, получив* x *— шифротекст (ещё его могут называть «код») сообщения. После того, как он передаст шифротекст Алисе, имеющей секретный ключ, та сможет восстановить сообщение дешифрующим алгоритмом:* $b_0 \dots b_l \leftarrow D(sk, x)$.

Все алгоритмы криптосистемы вероятностные и мы разрешаем им ошибаться. То есть может оказаться так, что D не сможет восстановить зашифрованное E сообщение. Мы потребуем лишь, чтобы вероятность успешного восстановления сообщения была не меньше какой-то константы от n. Вероятность в этом утверждении берётся по случайным битам всех трёх алгоритмов:

$$\Pr_{G,E,D}[D(sk, E(pk, m)) = m] \geq c > 0, \qquad \text{where } (pk, sk) \leftarrow G(1^n)$$

для любого сообщения m, где c — не зависящая от n константа.

(Если это выполняется, то можно добиться сколь угодно близкой к 1 вероятности успеха. Должно быть очевидно, как это сделать.)

Наших требований хватает для того, чтобы легитимный пользователь, имеющий сектретный ключ, смог корректно восстановить зашифрованные данные. Теперь введём требование, которое, в некотором смысле, гарантирует нам, что не имеющий секретного ключа противник не сможет прочесть зашифрованное сообщение.

Определение 2 (Взлом криптосистемы). *Пусть* A *— полиномиальный по времени вероятностный алгоритм. Рассмотрим такой эксперимент:*

(1) Сгенерируем пару ключей: $(pk, sk) \leftarrow G(1^n)$.

(2) Подадим A публичный ключ и попросим сгенерировать два различных сообщения: $(m_0, m_1) \leftarrow A(pk)$.

(3) Зашифруем случайное из этих сообщений: $i \leftarrow \{0, 1\}$, $x \leftarrow E(pk, m_i)$.

(4) Передадим его A и попросим угадать, какое из сообщений было зашифровано: $i' \leftarrow A(m_0, m_1, x)$.

Если в резльтате такого эксперимента окзалось $i = i'$*, то будет считать, что взлом удался.*

Нам бы хотелось, чтобы вероятность удачного взлома уменьшалась с увеличением n. Для этого введём следующее опредедение.

Определение 3 (Надёжная криптосистема)**.** *Криптосистема называется надёжной, если для любого полиномиального вероятностного алгоритма, вероятность успешного взлома криптосистемы этим алгоритмом становится меньше любого обратного полинома при достаточно больших* n.

(Обратный полином — это функция вида $1/p(n)$*, где* $p(n)$ *— полином.)*

Если взломщик научится взламывать криптосистему с вероятностью, ограниченной снизу обратным полиномом, то он сможет амплифицировать корректность до сколь угодно близкой к единице. (Очевидно, как.)

2 Цель и результат доклада

Цель. Целью доклада было определить криптосистему Ajtai-Dwork (читается «Айтай-Дворк») — задать три её алгоритма G, E, D. После чего доказать два утверждения:

(a) для этой криптосистемы выполняется заданное нами выше требование: вероятность корректного дешифрования ограничена снизу ненулевой константной от n;

(b) если криптосистема ненадёжна, то задача Unique Shortest Vector Problem (uSVP) решается за полиномиальное время в худшем случае.

Многим кажется неправдоподобным, что задача uSVP решаема, поэтому пункт (b) является (условным) доказательством надёжности нашей криптосистемы.

Результат. За доклад мы успели определить саму криптосистему, но до доказательств корректности (a) и надёжности (b) мы не дошли. И даже с заданием криптосистемы были проблемы — у меня были проблемы с алгоритмом дешифрования.

3 Напоминалка о содержании доклада

Напоминим, чем оперировала криптосистема:

Сообщение — Набор из $l + 1$ векторов $b_0 \dots b_l$.

Секретный ключ — Набор из $l + 1$ ортогональных векторов $u_0 \dots u_l$.

Публичный ключ — Три множества векторов из R^{n+l}: V — $l + 1$ штук, D — m' штук, P — $n + l$ штук.

Шифротекст — Вектор $x \in \mathbb{R}^{n+l}$.

Шифрование.

$$x = \sum_{i=0}^{l} v_i b_i + \sum_{i=0}^{m'} d_i \delta_i \mod \mathcal{L}(P),$$

где δ_i — случайные биты, выбираемые для шифрования, а $\mathcal{L}(P)$ — решётка, порождённая векторами P (то есть все их целочисленные линейные комбинации).