ПОСТ-КВАНТОВАЯ КРИПТОГРАФИЯ

Классические и квантовые вычисления

- Машины Тьюринга, схемы — классические объекты.
- Они локальны и подчиняются классическим законам.
- Но ведь мы живём в квантовом мире! Как это использовать?
- Квантовые вычисления — вычисления, существенно использующие квантовые эффекты.

Квантовые состояния

- Рассмотрим физическую систему, у которой может быть n состояний. Назовём их $|1\rangle, |2\rangle, \ldots, |n\rangle$.

- Квантовое состояние $|\varphi\rangle$ – суперпозиция классических:

$$|\varphi\rangle = \alpha_1 |1\rangle + \alpha_2 |2\rangle + \ldots + \alpha_n |n\rangle.$$

- $\alpha_i \in C$ – амплитуда $|i\rangle$ в $|\varphi\rangle$, $\sum_i |\alpha_i|^2 = 1$

Что можно с ними делать ?

- Математически говоря – состояния $|1\rangle, |2\rangle, \ldots, |n\rangle$ образуют ортонормированный базис гильбертова пространства размерности n.
- Квантовое состояние мы можем либо унитарно изменять, либо измерять.
- Измерение схлопывает его в классическое:
  - измеряя $|\varphi\rangle = \alpha_1 |1\rangle + \alpha_2 |2\rangle + \ldots + \alpha_n |n\rangle$ мы видим α_i с вероятностью $|\alpha_i|^2$

Операции

Можно применить унитарный оператор

$$U\left(\sum_i \alpha_i |i\rangle\right) = \sum_i \beta_i |i\rangle ,$$

т.е. умножить на унитарную матрицу

$$U\alpha = \beta, \quad U^{-1} = U^*.$$

Кубиты

- Кубит (qubit) – это суперпозиция 0 и 1, два базовых состояния:

$$\alpha_1 |1\rangle + \alpha_0 |0\rangle \, , \, |\alpha_0|^2 + |\alpha_1|^2 = 1$$

- Можно рассмотреть два кубита, базис будет

$$|00\rangle = |0\rangle |0\rangle, |01\rangle, |10\rangle, |11\rangle$$

Основные свойства кубита

- Запутывание
- Интерференция
- Параллелизм

Преобразование Адамара

- Пример унитарного преобразования – преобразование Адамара.
- Матрица Адамара:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

- На кубитах:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle = |+\rangle,$$

$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle = |-\rangle.$$

Запутывание

- Бывают запутанные состояния, например:

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

- Математически – тензорное произведение гильбертовых пространств.

- Система из n кубитов описывается набором из 2^n комплексных координат.

Запутывание

- Квантовый трюк номер один: запутывание (entanglement).
- Это как раз свойство нелокальности.
- Рассмотрим состояние

$$\frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

- И измерим первый из кубитов.
- Система спроецируется либо на $|00\rangle$, либо на $|11\rangle$.
- И мы будем знать второй кубит, не измеряя его!

Интерференция

- Запутанные состояния могут под действием унитарных преобразований распутываться.

- Это квантовый трюк номер два: интерференция (interference).
- На примере Адамара:

$$H|+\rangle = \frac{1}{\sqrt{2}}(H|0\rangle + H|1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle + |0\rangle - |1\rangle) = |0\rangle$$

$$H|-\rangle = \frac{1}{\sqrt{2}}(H|0\rangle - H|1\rangle) = \frac{1}{2}(|0\rangle + |1\rangle - |0\rangle + |1\rangle) = |1\rangle$$

Алгоритм Шора

- Теперь давайте рассмотрим алгоритм Шора.
- Дано n = pq, надо вычислить p и q.
- На самом деле алгоритм Шора по числу $x \in Z * n$ находит период f (a) = x a (mod n), т.е. минимальное r, для которого x r ≡ 1 (mod n) начнёт повторяться.

- Почему этого достаточно, чтобы разложить n?

Алгоритм Шора

- Для по крайней мере 1/ 4 всех x'ов r чётный, и $x^{r/2} \neq \mp 1 \ (mod\ n)$
- А тогда $(x^{\frac{r}{2}} + 1)(x^{\frac{r}{2}} - 1)$ = 0 (mod n), и мы всё раскладываем
- Сложность алгоритма $O(log^3 n)$ при использовании $O(log\ \ n)$ кубит.
- Работает в любой коммутативной группе!

Цели пост-квантовой криптографии

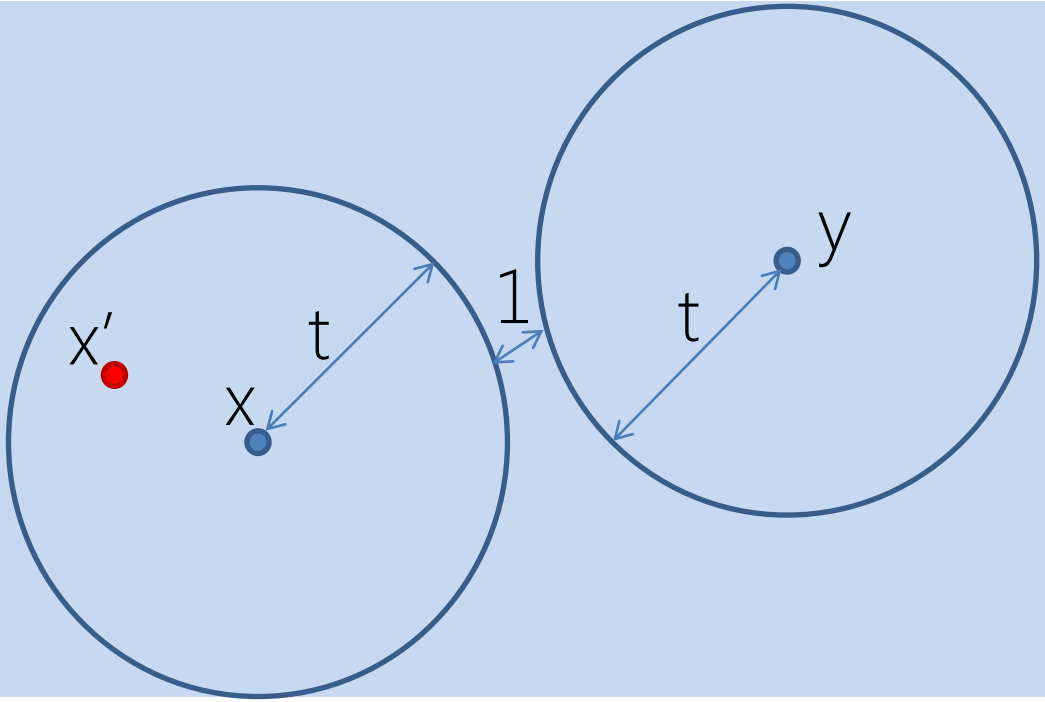
- Построить надёжную криптосистему.
  - Невозможно, пока не докажем, что P≠NP.
- Построить криптосистему, надёжность которой основана на NP-трудной задаче.
  - Не умеем.
- Построить криптосистему, надёжность которой основана на сложной, но вряд ли NP-трудной задаче.
  - Это мы уже делали: криптосистема Рабина, RSA.
- Построить криптосистему, надёжность которой кажется основанной на NP-трудной задаче.

Суть

- Что такое коды, исправляющие ошибки (error–correcting codes)?
- Это коды, которые умеют даже по неправильному кодовому слову достаточно часто выдавать правильное сообщение.
- Задача декодирования: по сигналу понять, какое кодовое слово передавалось.
- Говорят, что код исправляет t ошибок, если он корректно декодирует любой сигнал, искажённый в ≤ t битах.

Исправление ошибок

Пример



Код, исправляющий t ошибок

Код исправляет t ошибок, если его кодовое расстояние не меньше 2t+1

Линейные коды

- Пусть у нас блок размера k переходит в блок размера n при кодировании (n > k, разумеется).
- Предположим, что все биты кодового слова являются линейными функциями от битов сообщения (parity checks).
- Такие коды называются линейными.
- Эквивалентное определение — код линейный, если сумма кодовых слов является кодовым словом.

Линейные коды

Строки порождающей матрицы G образуют базу линейного кода.

Порождающая матрица

Проверочная матрица

Проверочная матрица $\mathbf{H}$ линейного кода $\mathbf{C}$ такова, что для любого кодового слова $\mathbf{x}$ и только для него выполняется $\mathbf{H}x^T = 0^{n-k}$

$$\mathrm{G} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{pmatrix}$$

Порождающая матрица

Проверочная матрица

$$\mathrm{H} = \begin{pmatrix} 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Пример

- Широко известен код Хэмминга (7, 4) (на 4 бита сообщения 7 битов сигнала).
- Главное свойство этого кода — то, что кодовые слова отличаются друг от друга как минимум в трёх битах.
- Другой способ достичь того же — повторять каждый бит три раза.
- Повторять гораздо менее эффективно: сообщение увеличивается в 3 раза, а в коде Хэмминга всего в 7/4 .

Способы задания линейных кодов

- Кодовое слово получается в виде c = sG, где G — матрица, называющаяся генератором кода.
- Если сначала сообщение повторяется, то G = [I | P]
- Например, для (7, 4)–кода Хэмминга

$$G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Синдром

- Для декодирования используются так называемые синдромы.
- Синдром — это разница между реальным сигналом и сигналом, вычисленным на основании полученных битов сообщения.
- Если c = sG, и G = [I_k | P], то синдром z = Hr, где H = [$-P^T$ | I_{n-k}].
- Например, для (7, 4)–кода Хэмминга

$$H = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

Синдромы

- Для всякого валидного кодового слова c $Hc^\top = 0$:

$$Hc^\top = H(sG)^\top = HG^\top s^\top = \begin{pmatrix} -P^\top & I_{n-k} \end{pmatrix} \begin{pmatrix} I_k \\ P \end{pmatrix} s^\top = 0.$$

- Если же полученное слово r неправильное, и в нём есть шум, то результат $Hr^\top$ помогает найти n и декодировать.

Постановка задачи декодирования

- Получаемый вектор r — это сумма кодового слова и шума:

$$r = sG + n^T.$$

- Задача декодирования синдрома — это задача поиска такого вектора шума n, что $Hn = z$.
- Если код исправляет t ошибок, то среди слов, отличающихся от s в $\leq t$ местах, есть только одно решение этого уравнения.
- Проще говоря, для кода, исправляющего t ошибок, в шаре радиуса t вокруг каждого кодового слова других кодовых слов не встречается.

Сложность задачи декодирования

- Для кода Хэмминга решить задачу декодирования несложно. Однако это не всегда так.

- Для абстрактного линейного кода нужно решать задачу MLD (maximum likelihood decoding): по данному вектору с ошибкой найти ближайшее кодовое слово из кода.
- Декодировать произвольный линейный код — NP-трудная задача

Коды Гоппы

Первым (1981) осознал связь между алгебраической геометрией и теорией кодирования.

Валерий Денисович Гоппа

Коды Гоппы

Линейные коды порожденные несингулярными проективными кривыми над конечными полями.

Коды Гоппы имеют важное свойство. Они обладают быстрыми процедурами декодирования по алгоритму Питерсона.

Gary L. Peterson

Пример построения кода

- Зафиксируем число m (обычно 10, 11 или 12), число $n \leq 2^m$ (часто берут $n = 2^m$) и $t \in [2, \frac{2^m-1}{m}]$.
- Выберем последовательность $a_1, \ldots, a_n \in \mathbb{F}_{2^m}$ (если $n = 2^m$, то это просто все элементы в лексикографическом порядке).
- Определим многочлен

$$h = \prod_{i=1}^{n} (x - a_i) \in \mathbb{F}_{2^m}[x].$$

- Зафиксируем неприводимый многочлен g степени t.

Двоичные коды Гоппы

- Тогда множество кодовых слов — это:

$$\Gamma = \Gamma(a_1 \ldots, a_n, g) = \left\{ c \in \mathbb{F}_2^n : \sum_{i=1}^{n} c_i \frac{h}{x - a_i} \equiv 0 \pmod{g} \right\}.$$

- Иначе говоря, множество Γ — это ядро отображения–«синдрома» $\mathbb{F}_2^n \to \mathbb{F}_{2^m}^t$, которое действует как

$$c = (c_0, \ldots, c_{n-1}) \mapsto$$

$$\mapsto b_0 + b_1 x + \ldots + b_{t-1} x^{t-1} \equiv \sum_{i=1}^{n} c_i \frac{h}{x - a_i} \pmod{g}.$$

- Значит, размерность кода — по меньшей мере $n - mt$, и мы получили $(n, \geq n - mt)$-код.

Кодовая криптография

Линейные коды

Алгебраическая теория кодирования

Криптография

Криптосистема Роберта Мак-Элиса, 1978 год

Криптосистема Гаральда Нидеррайтра, 1986 год

Доказано, что данные криптосистемы эквивалентны

Криптосистема МакЭлиса: обзор

- А теперь собственно криптосистема МакЭлиса (McEliece cryptosystem).

Криптосистема МакЭлиса: ключи

- Генерация ключа: фиксируем k, n, t. Каждый участник делает следующее.
  1. Выбрать матрицу кода G размера $k \times n$ для (n, k)-линейного кода, исправляющего t ошибок, для которого известен эффективный алгоритм декодирования (например, код Гоппы).
  2. Выбрать случайную невырожденную матрицу S размера $k \times k$.
  3. Выбрать случайную матрицу перестановки P размера $n \times n$.
  4. Выдать как публичный ключ t и $\hat{G} = SGP$; секретный ключ — (S, G, P).

Криптосистема МакЭлиса: [де]кодирование

- Алгоритм кодирования (даны t, $\hat{G}$ и сообщение m).
  1. Представить сообщение как строку длины k.
  2. Выбрать случайный вектор шума z длины n с $\leq t$ единичками.
  3. Закодировать $c = m\hat{G} + z$.
- Алгоритм декодирования (даны c и ключ (S, G, P)).
  1. Вычислить $\hat{c} = cP^{-1}$.
  2. Декодировать то, что получилось, алгоритмом декодирования кода; получится $\hat{m}$.
  3. Вычислить $m = \hat{m}S^{-1}$.

Криптосистема МакЭлиса: о стойкости

- По сути: мы хотим попросить врага решить NP-трудную задачу.
- Но не можем.
- Поэтому мы берём простой частный случай NP-трудной задачи, а потом «запутываем» его так, чтобы врагу было не догадаться, какой это частный случай.

- Для многих кодов криптосистему МакЭлиса взломали; для кодов Гоппы пока нет.

Атаки на систему

- Атака: найти матрицу G' кода Гоппы, соответствующую данной $\hat{G}$. Этого делать пока никто не умеет.
- Другая атака:
  - выбрать k столбцов из $\hat{G}$, ограничить на них матрицу и векторы; будет $c_k = m_k \hat{G}_k + z_k$.
  - предположить, что $z_k = 0$;
  - решить напрямую систему $c_k = m_k \hat{G}_k$, найти m_k.
- Вероятность того, что $z_k = 0$, очень мала.
- Только в 2008 году смогли успешно атаковать криптосистему МакЭлиса для $n = 1024$, $k = 524$, $t = 50$.

Криптосистема Нидеррайтера

Та же NP-трудная задача декодирования линейных кодов, что и в криптосистеме Мак-Элиса.	Основа криптосистемы
В качестве базовой матрицы используется проверочная матрица группового кода. Открытый ключ представляет собой множество, состоящее из проверочной матрицы общего положения, оснащенное достаточно большим числом t.	Основное отличие от криптосистемы Мак-Элиса
Сообщение передается не информационным блоком кодового слова, а вектором ошибок. Шифротекст же получается в виде синдрома – произведения проверочной матрицы на вектор ошибок.	Шифрование сообщений

Криптосистема Нидеррайтера: ключи

- Генерация ключа: фиксируем k, n, t.
- Каждый участник делает следующее.
  - 1 Выбрать проверочную матрицу кода H размера (n-k) × n для (n, k)-линейного кода, исправляющего t ошибок, для которого известен эффективный алгоритм декодирования (например, код Гоппы).
  - 2 Выбрать случайную невырожденную матрицу S размера (n-k) × (n-k).
  - 3 Выбрать случайную матрицу перестановки P размера n × n.
  - 4 Выдать как публичный ключ t и H’ = SHP; секретный ключ — (S, H, P).

Криптосистема Нидеррайтера: [де]кодирование

- Алгоритм кодирования (даны t, H’, F(x) и сообщение m).
  - 1 Представить сообщение как F(m) строку длины n веса до t.
  - 2 Зашифровать $c = F(m)H'^{T}$.
- Алгоритм декодирования (даны c и ключ (S, H, P)).
  - 1 Вычислить $c' = cS^{-1}$.
  - 2 Декодировать то, что получилось, алгоритмом декодирования кода; получится e’ . Проверить, что вес e’ < t
  - 3 Вычислить $m = F^{-1}(e'P^{-1})$.

Проблемы кодовой криптографии

- Большие размеры ключей
- Отсутствие алгоритмов генерации подписи
  - Единственные известные подходы базируются на системе Нидеррайтера
  - Высокая сложность
- Стойкость основана на допущении о неотличимости

Подписи, основанные на кодах

- Подпись KKS
  - Идея: Построение функции хеширования, отображающей множество сообщений на декодируемый синдром
  - Проблемы: Малое подпространство синдромов – атаки с обучением
- Подпись CSF
  - Идея: Попытка декодирования за расстояние
  - Проблемы:
    - Большая длина
    - Высокая сложность