Лекция №4

Протокол обмена ключами Диффи-Хэллмана

Елена Киршанова

Курс "Основы криптографии"

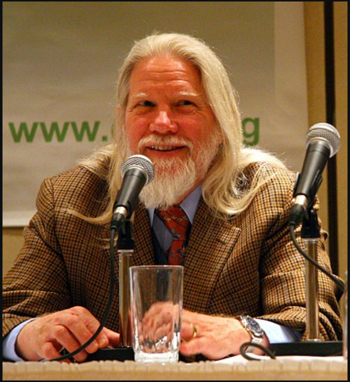
До сих пор...

- Псевдослучайные генераторы
- Потоковые шифры
- Блок-шифры
- Коды Аутентификации сообщений (MAC)
- Хэш-функции
- Шифрование с аутентификацией

Кроме псевдослучайных генераторов, **все** примитивы подразумевают наличие общего секретного ключа.

Как двум сторонам сгенерировать секретный ключ?

Обмен ключами Диффи-Хэллмана (Diffie-Hellman Key Exchange)



Whitfield Diffie ©Wikipedia

Martin Hellman ©Wikipedia

- протокол предложен Whitfield Diffie, Martin Hellman в 1976 *"New Directions in Cryptography"*
- используется в современных протоколах (TLS, Signal)
- в основе трудность задачи дискретного логарифма

Обмен ключами: определение

Сценарий: Алиса и Боб имеют общий канал связи. Их цель: сгенерировать общий секретный ключ

Злоумышленник – пассивный участник, просматривающий канал связи

Протокол обмена ключами – ppt интерактивный алгоритм, состоящий из двух алгоритмов:

1. $\mathtt{GenParam}(1^\lambda) \to \mathtt{param}$
2. $\mathtt{KeyGen}(\mathtt{param}) \to k \in \mathcal{K}$ – интерактивный протокол.

Передаваемая информация по каналу – $\mathtt{trans}$

Обмен ключами: безопасность

$\Pi = (\mathtt{GenParam}, \mathtt{KeyGen})$ – протокол обмена ключами

Челленджер $\mathcal{C}$

$k_0, \mathtt{trans} \leftarrow \Pi(1^\lambda)$

$k_1 \xleftarrow{\$} \mathcal{K}$

$b \xleftarrow{\$} \{0,1\}$

$\xrightarrow{(k_b, \mathtt{trans})}$

$\xleftarrow{\hat{b}}$

Атакующий $\mathcal{A}$

$\mathtt{W}_{\Pi,\mathcal{A}}$ – событие $\hat{b} == b$.

$\mathtt{Adv} = \left|\Pr[\mathtt{W}_{\Pi,\mathcal{A}}] - \frac{1}{2}\right|$ – выигрыш $\mathcal{A}$.

Протокол обмена ключами Π безопасен относительно пассивного атакующего, если для любого ppt $\mathcal{A}$:

$$\mathtt{Adv} = \mathrm{negl}(\lambda).$$

Часть II

Крэш-курс по конечным полям

Модульная арифметика

Возьмем p – большое простое число ($\sim$ 2000 бит)

- $\mathbb{Z}_p = \{0, 1, \ldots, p-1\}$ – конечное поле
- умножение/сложение в $\mathbb{Z}_p$ производится по модулю p, т.е. для $x, y \in \mathbb{Z}_p$

$$x + y \bmod p = \mathrm{rem}(x + y, p)$$
$$x \cdot y \bmod p = \mathrm{rem}(x \cdot y, p)$$

rem – остаток от целочисл. деления

Модульная арифметика

Возьмем p – большое простое число ($\sim$ 2000 бит)

- $\mathbb{Z}_p = \{0, 1, \dots, p-1\}$ – конечное поле
- умножение/сложение в $\mathbb{Z}_p$ производится по модулю p, т.е. для $x, y \in \mathbb{Z}_p$

$$x + y \bmod p = \mathrm{rem}(x + y, p)$$
$$x \cdot y \bmod p = \mathrm{rem}(x \cdot y, p)$$

rem – остаток от целочисл. деления

Пример: $p = 7, \mathbb{Z}_p = \{0, 1, 2, 3, 4, 5, 6\}$

$$5 + 6 \bmod p = \mathrm{rem}(11, 7) = 4$$
$$3 \cdot 3 \bmod p = \mathrm{rem}(9, 7) = 2$$

Модульная арифметика

Возьмем p – большое простое число ($\sim$ 2000 бит)

- $\mathbb{Z}_p = \{0, 1, \dots, p-1\}$ – конечное поле
- умножение/сложение в $\mathbb{Z}_p$ производится по модулю p, т.е. для $x, y \in \mathbb{Z}_p$

$$x + y \bmod p = \mathrm{rem}(x + y, p)$$
$$x \cdot y \bmod p = \mathrm{rem}(x \cdot y, p)$$

rem – остаток от целочисл. деления

Пример: $p = 7, \mathbb{Z}_p = \{0, 1, 2, 3, 4, 5, 6\}$

$$5 + 6 \bmod p = \mathrm{rem}(11, 7) = 4$$
$$3 \cdot 3 \bmod p = \mathrm{rem}(9, 7) = 2$$

- Для ненулевого $x \in \mathbb{Z}_p$, $\exists x^{-1} \in \mathbb{Z}_p$: $x \cdot x^{-1} = 1 \bmod p$
  Пример.: $1^{-1} = 1$, $2^{-1} = 4$, $3^{-1} = 5$, $4^{-1} = 2$, $5^{-1} = 3$, $6^{-1} = 6$ в $\mathbb{Z}_7$
- Множество обратимых элементов $\mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\}$

Структура $\mathbb{Z}_p^*$

- Теорма Ферма: $g^{p-1} = 1 \bmod p \ \ \forall 0 \neq g \in \mathbb{Z}_p$

  Пример.: $2^6 = 64 = 1 \bmod 7$

- $\mathbb{Z}_p^*$ – циклическая группа, т.е.,
  $\exists g \in \mathbb{Z}_p$ s.t. $\mathbb{Z}_p^* = \{1 = g^0, g^1, g^2, \ldots, g^{p-2}\}$

  Пример: $\mathbb{Z}_7^* = \{1, 3, 3^2 = 2, 3^3 = 6, 3^4 = 4, 3^5 = 5\}$

- Не каждый элемент является образующим $\mathbb{Z}_p$, но мы знаем, как его эффективно отыскать

- Порядок $g \in \mathbb{Z}_p$, $\mathrm{ord}(g)$ – наименьшее положительное a т.ч. $g^a = 1$

  Для любого образующего g, $\mathrm{ord}(g) = p - 1$.

Вычисления в $\mathbb{Z}_p$

Эффективность арифметики в $\mathbb{Z}_p$ измеряется в $n = \lceil \log p \rceil$

- Сложение: $\mathcal{O}(n)$ битовых операций
- Умножение: $\mathcal{O}(n^2)$ (или $\mathcal{O}(n^{1.7})$) битовых операций
- Нахождение обратного: $\mathcal{O}(n^2)$ битовых операций

Вычисления в $\mathbb{Z}_p$

Эффективность арифметики в $\mathbb{Z}_p$ измеряется в $n = \lceil \log p \rceil$

- Сложение: $\mathcal{O}(n)$ битовых операций
- Умножение: $\mathcal{O}(n^2)$ (или $\mathcal{O}(n^{1.7})$) битовых операций
- Нахождение обратного: $\mathcal{O}(n^2)$ битовых операций
- Возведение в степень, x^r: $\mathcal{O}(\log r)$ умножений в $\mathbb{Z}_p$ (быстрое возведение в степень)
    - $y \leftarrow g$, $z \leftarrow 1$
    - `for` i `in` $[0, n]$:
    - `if` $r[i] == 1$: $z \leftarrow z \cdot y$
    - $y \leftarrow y^2$
    - `return` z

Вычисления в $\mathbb{Z}_p$

Эффективность арифметики в $\mathbb{Z}_p$ измеряется в $n = \lceil \log p \rceil$

- Сложение: $\mathcal{O}(n)$ битовых операций
- Умножение: $\mathcal{O}(n^2)$ (или $\mathcal{O}(n^{1.7})$) битовых операций
- Нахождение обратного: $\mathcal{O}(n^2)$ битовых операций
- Возведение в степень, x^r: $\mathcal{O}(\log r)$ умножений в $\mathbb{Z}_p$ (быстрое возведение в степень)
  - $y \leftarrow g$, $z \leftarrow 1$
  - `for` i `in` $[0, n]$:
  - `if` $r[i] == 1$: $z \leftarrow z \cdot y$
  - $y \leftarrow y^2$
  - `return` z

  Пример: вычислим g^r для $r = 23 = (10111)_2$. I.e., $g^{23} = g^{16+4+2+1}$.

$$g^1 \to g^{1+2} \to g^{1+2+4} \to g^{1+2+4} \to g^{1+2+4+16}$$

Эти операции эффективны в $\mathbb{Z}_p$

Часть III

Задача Диффи-Хэллмана. Протокол Диффи-Хэллмана

Трудные (сегодня) задачи в $\mathbb{Z}_p$

1. Задача дискретного логарифма (dlog):

   Для g – образующего $\mathbb{Z}_p^*$ и $x \in \mathbb{Z}_p^*$, найти r т.ч. $g^r = x \bmod p$

   Пример: Для $\langle 3 \rangle = \mathbb{Z}_7^*$ и 5 найти $r = 5$ $(3^5 = 5)$.

2. Задача Diffie-Hellman (вычислительная версия) (CDH):

   Для g – образующего $\mathbb{Z}_p^*$, $x = g^r \in \mathbb{Z}_p^*$, $y = g^t \in \mathbb{Z}_p^*$ найти $z = g^{r \cdot t} \bmod p$.

   Пример: Для $\langle 3 \rangle = \mathbb{Z}_7^*$ и $x = 2, y = 6$ найти $z = 3^5 = 5 \bmod p$.

3. Задача принятия решения Diffie-Hellman (DDH):

   Для g – образующего $\mathbb{Z}_p^*$, $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$, отличить тройки

$$(g^a, g^b, g^{ab}) \qquad (g^a, g^b, g^c)$$

Трудные (сегодня) задачи в $\mathbb{Z}_p$

1. Задача дискретного логарифма (dlog):

   Для g – образующего $\mathbb{Z}_p^*$ и $x \in \mathbb{Z}_p^*$, найти r т.ч. $g^r = x \bmod p$

   Пример: Для $\langle 3 \rangle = \mathbb{Z}_7^*$ и 5 найти $r = 5$ $(3^5 = 5)$.

2. Задача Diffie-Hellman (вычислительная версия) (CDH):

   Для g – образующего $\mathbb{Z}_p^*$, $x = g^r \in \mathbb{Z}_p^*$, $y = g^t \in \mathbb{Z}_p^*$ найти $z = g^{r \cdot t} \bmod p$.

   Пример: Для $\langle 3 \rangle = \mathbb{Z}_7^*$ и $x = 2, y = 6$ найти $z = 3^5 = 5 \bmod p$.

3. Задача принятия решения Diffie-Hellman (DDH):

   Для g – образующего $\mathbb{Z}_p^*$, $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$, отличить тройки

$$(g^a, g^b, g^{ab}) \qquad (g^a, g^b, g^c)$$

DDH $\leq$ CDH $\leq$ DLOG

A $\leq$ B = “решение для B дает решение для A”

В общем случае редукции в обратную сторону не известны.

Сложность задач DLOG/CHD/DDH?

Лучший из известных на сегодня алгоритмов для вычисления DLOG в $\mathbb{Z}_p^*$ для $n = \lceil \log p \rceil$: General Number Field Sieve работает за суб-экспоненциальное время

$$e^{n^{1/3}}$$

Для уровня безопасности $\lambda = 128$ бит, необходимо взять

$$n \approx 3072 \text{ бит}$$

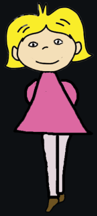
Вместо группы $\mathbb{Z}_p^\star$ на практике используется группа рациональных точек эллиптической кривой.

Преимущество: известны лишь экспоненциальные (от порядка группы) алгоритмы для задачи dlog.

Обмен ключами Диффи-Хэллмана

$\texttt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса



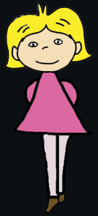
Боб

Обмен ключами Диффи-Хэллмана

$\texttt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса

$a \leftarrow \{2, \ldots, p-2\}$



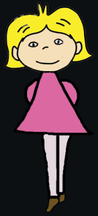
Боб

$b \leftarrow \{2, \ldots, p-2\}$

Обмен ключами Диффи-Хэллмана

$\texttt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса



$a \leftarrow \{2, \ldots, p-2\}$

$k_\mathsf{A} = g^a \bmod p$

$\xrightarrow{k_\mathsf{A}}$

Боб

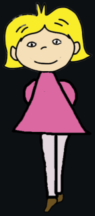
$b \leftarrow \{2, \ldots, p-2\}$

$k_\mathsf{B} = g^b \bmod p$

Обмен ключами Диффи-Хэллмана

$\mathtt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса



$a \leftarrow \{2, \ldots, p-2\}$

$k_{\mathsf{A}} = g^a \bmod p$

$\xrightarrow{k_{\mathsf{A}}}$

$\xleftarrow{k_{\mathsf{B}}}$

Боб

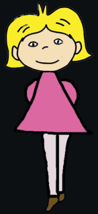
$b \leftarrow \{2, \ldots, p-2\}$

$k_{\mathsf{B}} = g^b \bmod p$

Обмен ключами Диффи-Хэллмана

$\texttt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса



$a \leftarrow \{2, \ldots, p-2\}$

$k_{\mathsf{A}} = g^a \bmod p$

$k_{\mathsf{AB}} = k_{\mathsf{B}}^a$

$\xrightarrow{k_{\mathsf{A}}}$

$\xleftarrow{k_{\mathsf{B}}}$

Боб

$b \leftarrow \{2, \ldots, p-2\}$

$k_{\mathsf{B}} = g^b \bmod p$

$k_{\mathsf{AB}} = k_{\mathsf{A}}^b$

k_{AB} – общий ключ

Корректность: $k_{\mathsf{B}}^a = (g^b)^a = g^{ab} = (g^a)^b = k_{\mathsf{A}}^b$.

Безопасность (неформально): атакующий видит траскрипт g^a, g^b. Для того, чтобы различить g^{ab} от случайного элемента $\mathbb{Z}_p^*$, он должен решить задачу DDH.

Безопасность протокола Диффи-Хэллмана

Теорема. Протокол Диффи-Хэллмана безопасен относительно пассивного атакующего под предположением сложности задачи DDH.

Челленджер $\mathcal{C}$

$$\mathtt{trans} = (g^a, g^b), k_0 = g^{ab}$$

$$k_1 = g^c \to \mathbb{Z}_p^*$$

$$b \xleftarrow{\$} \{0, 1\}$$

$$\xrightarrow{(k_b, \mathtt{trans})}$$

$$\xleftarrow{\hat{b}}$$

Атакующий $\mathcal{A}$

Активная атака "человек по середине" (Man-in-the-middle attack)

Протокол Диффи-Хэллмана в "чистом виде" подвержен **активным** атакам.

Alice

Eve

Bob

$$k_{\mathsf{A}} = g^a \bmod p \qquad \underset{\xleftarrow{\;g^{b'}\;}}{\xrightarrow{\;k_{\mathsf{A}}\;}} \qquad \underset{\xleftarrow{\;k_{\mathsf{B}}\;}}{\xrightarrow{\;g^{a'}\;}} \qquad k_{\mathsf{B}} = g^b \bmod p$$

$$g^{ab'} \qquad\qquad g^{ab'},\ g^{a'b} \qquad\qquad g^{a'b}$$

Фикс: использовать (ассиметрическую) аутентификацию.
См. лекцию о цифровой подписи

Протокол Диффи-Хэллмана на практике

- Атака “человек по середине” предотвращается с помощью цифровой подписи
- В реальных приложениях протокол ДХ вместо группы $\mathbb{Z}_p^*$ использует группу рац, точек эллиптической кривой. Соответствующая задача: EDDH.
- Построить подходящую группу для задач DLOG, CDH, DDH нетривиальна! Не придумывайте свою, используйте стандарты.
- См. `https://safecurves.cr.yp.to/` для выбора хорошей кривой
- ГОСТа для протокола обмена ключами нет, есть RFC `https://www.ietf.org/rfc/rfc5246.txt` и рекомендации.

Часть IV

PAKE. Парольный обмен ключами с аутентификацией

Password Authenticated Key Exchange (PAKE): мотивация

- Классический Диффи-Хэллмана (ДХ) протокол уязвим к активной атаке
- Безопасная реализация ДХ подразумевает аутентифицированный канал
- Аутентификация: либо цифровая подпись, либо парольные решения

Сценарий: две стороны, сервер и клиент, знают пароль pwd

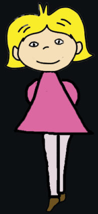
Задача: обезопасить клиента от атакующего, представляющегося сервером.

Упрощенная версия PAKE_0

$\mathcal{H}$– криптографическая хэш-функция. R, S–множество нонсов

$\mathtt{id}_A$ – идентификатор Алисы, $\mathtt{id}_B$ – идентификатор Боба

Алиса (клиент)



pwd

$\xrightarrow{\quad r \quad}$

$\xleftarrow{\quad s \quad}$

Боб (сервер)

$r \xleftarrow{\$} R$

$k = \mathcal{H}(\mathtt{id}_A, \mathtt{id}_B, r, s, pwd)$

$s \xleftarrow{\$} S$

$k = \mathcal{H}(\mathtt{id}_A, \mathtt{id}_B, r, s, pwd)$

k – общий ключ

Проблема: Если pwd легко угадывается, пассивный злоумышленник может легко провести атаку перебором на k

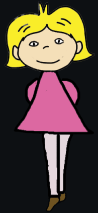
Упрощенная версия PAKE$_1$

$\mathcal{H}$– криптографическая хэш-функция

$\mathtt{id}_A$ – идентификатор Алисы, $\mathtt{id}_B$ – идентификатор Боба

$\mathtt{GenParam} \to (g, p)$, где p – большое простое число и $\langle g \rangle = \mathbb{Z}_p^*$

Алиса (клиент) — Боб (сервер)



pwd

$$a \leftarrow \{2, \ldots, p-2\}$$

$$u = g^a$$

$$\xrightarrow{\ \ u\ \ }$$

$$\xleftarrow{\ \ v\ \ }$$

$$b \leftarrow \{2, \ldots, p-2\}$$

$$v = g^b$$

$$k = \mathcal{H}(\mathtt{id}_A, \mathtt{id}_B, u, v, g^{ab}, pwd)$$

PAKE$_1$ безопасен относительно пассивных злоумышленников при условии сложности задачи CDH.

PAKE$_1$ небезопасен относительно активного атакующего, выдающего себя за сервер

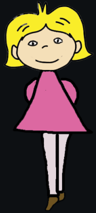
Версия PAKE$_2$

$\mathcal{H}$– криптографическая хэш-функция

$\mathtt{id}_A$ – идентификатор Алисы, $\mathtt{id}_B$ – идентификатор Боба

Зафиксируем большое простое p, и $\langle g \rangle = \mathbb{Z}_p^*$, $\alpha, \beta \in \mathbb{Z}_p^*$

Алиса (клиент)



$pwd \in \mathbb{Z}_p^*$

Боб (сервер)

$a \leftarrow \{2, \ldots, p-2\}$

$u = g^a \cdot \alpha^{pwd}$

$w = (v/\beta^{pwd})^a$

$\xrightarrow{\quad u \quad}$

$\xleftarrow{\quad v \quad}$

$b \leftarrow \{2, \ldots, p-2\}$

$v = g^b \cdot \beta^{pwd}$

$w = (u/\alpha^{pwd})^b$

$$k = \mathcal{H}(\mathtt{id}_A, \mathtt{id}_B, u, v, w, pwd)$$

PAKE$_2$ безопасен относительно активных злоумышленников при условии сложности задачи CDH

PAKE на практике

- Пароли на сервере хранятся в виде $g^{\mathcal{H}(\text{salt}, pwd)}$ (атакующий, получив это значение, не может эффективно вычислить pwd)
- Популярный пример PAKE: Secure Remote Password (SRP) (iCloud Key Vault)
- Другой пример: OPAQUE `https://tools.ietf.org/html/draft-krawczyk-cfrg-opaque-00` — криптографический стойкий и эффективный PAKE