

Криптографические примитивы и протоколы

Винарский Евгений
по всем вопросам писать на vinevg2015@gmail.com

Институт Системного программирования

6 ноября 2025 г.

Содержание

- 1 Слабые места защиты информации
- 2 Криптографические примитивы
- 3 Модель противника
- 4 Свойства безопасности
- 5 Уязвимость протокола Нидхема-Шрёдера

Слабые места защиты информации

- Атаки на архитектуру (Криптографическая система не может быть надежнее использованных в ней отдельных алгоритмов шифрования)
- Атаки на конкретные реализации
 - Переполнение буферов
 - Не стёртая до конца секретная информация
- Атаки на сетевое оборудование
- Атаки на пользователей
- Атаки с использованием побочных каналов
- ...

Для того чтобы преодолеть систему защиты, достаточно взломать любой из ее компонентов

Мультиликативная группа

Мультиликативная группа G – непустое множество, на котором определена ассоциативная бинарная операция умножения $(*)$, причём

- Для этой операции имеется нейтральный элемент (1) такой, что $\forall a \in G : a * 1 = 1 * a = a$
- Каждый элемент a множества G имеет обратный a^{-1} , то есть такой $a * a^{-1} = a^{-1} * a = 1$

$|G|$ – порядок группы G (количество элементов в группе G)

| $(G, *)$ | 1 | 2 | 3 | 4 |
|----------|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 4 | 1 | 3 |
| 3 | 3 | 1 | 4 | 2 |
| 4 | 4 | 3 | 2 | 1 |

Пример группы с операцией умножения по модулю 5

Мультиликативная группа

Мультиликативная группа G – непустое множество, на котором определена ассоциативная бинарная операция умножения $(*)$, причём

- Для этой операции имеется нейтральный элемент (1) такой, что $\forall a \in G : a * 1 = 1 * a = a$
- Каждый элемент a множества G имеет обратный a^{-1} , то есть такой $a * a^{-1} = a^{-1} * a = 1$

$|G|$ – порядок группы G (количество элементов в группе G)

| $(G, *)$ | 1 | 2 | 3 | 4 |
|----------|---|---|---|---|
| 1 | 1 | 2 | 3 | 4 |
| 2 | 2 | 4 | 1 | 3 |
| 3 | 3 | 1 | 4 | 2 |
| 4 | 4 | 3 | 2 | 1 |

Пример группы с операцией умножения по модулю 5

Можно ли построить группу с операцией умножения по модулю 4?

МультиPLICATивная группа (2)

Нет, нельзя, не у каждого элемента есть обратный

| $(G, *)$ | 1 | 2 | 3 |
|----------|---|---|---|
| 1 | 1 | 2 | 3 |
| 2 | 2 | 0 | 3 |
| 3 | 3 | 2 | 1 |

- Группа G – циклическая, если существует $g \in G$ такой, что группа G есть множество степеней этого элемента
- Множество натуральных чисел $\{1, \dots, p-1\}$ с операцией умножения по модулю p является группой, если и только если p – простое число

Например, для группы с операцией умножения по модулю 5, $g = 2$ – образующий элемент: $\{2^1 = 2, 2^2 = 4, 2^3 = 3, 2^4 = 1\}$

? $5 \bmod 7$

? образующий элемент в $(\{1, \dots, 6\}, *)$

Криптографические примитивы

- Алгоритмы симметричного шифрования
- Алгоритмы асимметричной криптографии (выработка общих сессионных ключей и т.д.)
- Генератор псевдослучайных чисел
- Алгоритмы хэширования

В этом блоке считаем, что все криптографические примитивы не могут быть скомпрометированы раньше, чем перестанут использоваться

Какие достоинства и недостатки асимметричной криптографии?

Модель противника

- **Модель атаки** Возможности противника по взаимодействию с системой
- **Ресурсы противника** Предположения о вычислительных и информационных ресурсах противника
- **Угроза** Задача противника по нарушению свойств безопасности

Уязвимости криптосистемы возникают, если неправильно выбраны

- ➊ модель атаки
- ➋ угроза
- ➌ предположения о ресурсах

Модель противника (2)

- **Пассивный противник** (противник может читать зашифрованные пересылаемые данные в открытом канале)
- **Dolev-Yao (Активный) Противник** может:
 - читать сообщения в канале
 - модифицировать сообщения в канале
 - удалять сообщения из канала
- Противник, учитывающий временные задержки (может определить, какая именно проверка не прошла, ...)
- ...

Рассуждать о стойкости криптосистемы можно только в терминах модели противника

Протокол (алгоритм) Диффи-Хеллмана

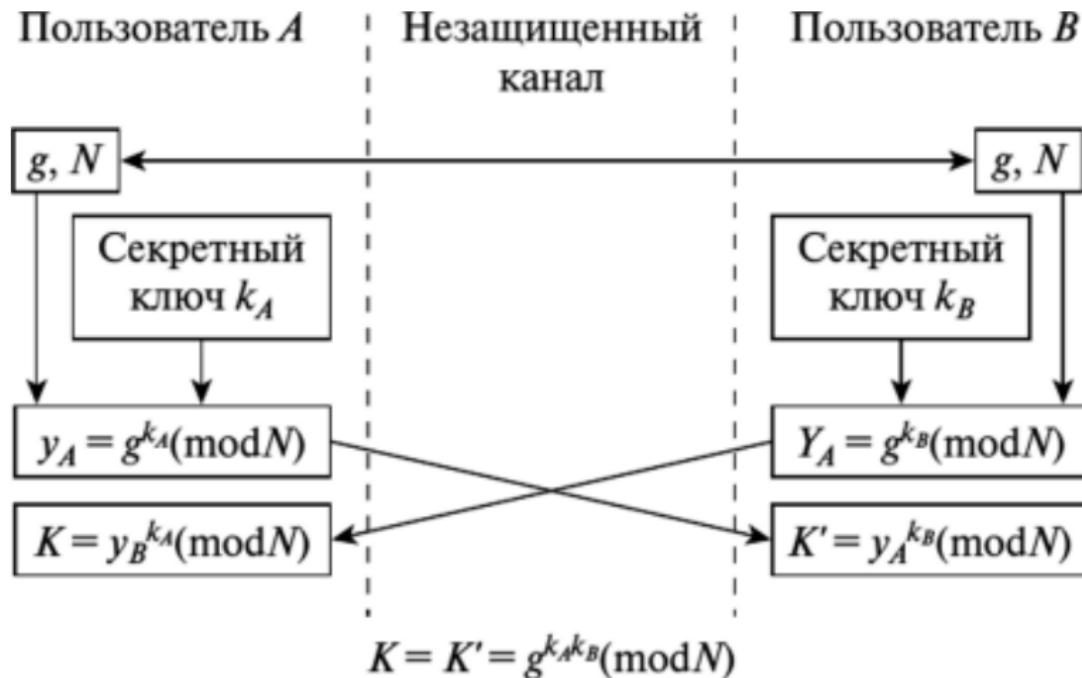


Схема протокола Диффи-Хеллмана

Модель противника

Протокол Диффи-Хелммана при пассивном противнике

- E – пассивный противник, слушающий незащищённый канал
- E известно значение g, N и y_A, y_B

Угроза: Противник E узнал выработанный общий ключ

$$K = g^{k_A k_B} \pmod{N}$$

Модель противника

Протокол Диффи-Хелммана при пассивном противнике

- E – пассивный противник, слушающий незащищённый канал
- E известно значение g, N и y_A, y_B

Угроза: Противник E узнал выработанный общий ключ

$$K = g^{k_A k_B} \pmod{N}$$

Пусть E скомпрометировал $K = g^{k_A k_B} \pmod{N}$, тогда:

- узнал k_A
 - решил задачу дискретного логарифмирования, т.е. вычислил k_A из уравнения $y_A = g^{k_A} \pmod{N}$
- узнал k_B
 - решил задачу дискретного логарифмирования, т.е. вычислил k_B из уравнения $y_B = g^{k_B} \pmod{N}$
- узнал $k_A * k_B$

Протокол Диффи-Хэллмана стойкий по отношению к пассивному противнику

Модель противника

Протокол Диффи-Хелммана при активном противнике

Угроза: Противник E отправил сообщение серверу от лица клиента

Диффи-Хэллман

| Client (pkC , $privC$) | | Server (pkS , $privS$) |
|--|--------------------------------------|------------------------------------|
| - генерация g^x | $\langle pkC, g^x \rangle$ -----> | |
| | $\langle pkS, g^y \rangle$ <----- | - генерация g^y |
| - $client_key = g^{x^y}$ | $\langle NB \rangle pk(B)$ -----> | - <u>server_key</u> = g^{x^y} |
| $key = g^{x^y} -- \text{общий секрет}$ | | |

Модель противника

Протокол Диффи-Хелммана при активном противнике

Угроза: Противник E отправил сообщение серверу от лица клиента

Диффи-Хэллман

| Client (pkC , $privC$) | | Server (pkS , $privS$) |
|-----------------------------------|--------------------------------------|------------------------------------|
| - генерация g^x | $\langle pkC, g^x \rangle$ -----> | |
| | $\langle pkS, g^y \rangle$ <----- | - генерация g^y |
| - $client_key = g^{y^x}$ | $\langle NB \rangle pk(B)$ -----> | - <u>server_key</u> = g^{x^y} |
| $key = g^{(x^y)}$ -- общий секрет | | |

Противник отправляет серверу сообщение $\langle pk_E, g^x \rangle$ от лица клиента



Протокол Диффи-Хэллмана не является стойким по отношению к

Свойства безопасности протоколов выработки общих ключей обмена

- Аутентификация
 - Ложная аутентификация
 - Unknown key share (Неизвестный общий ключ)
- Установление одинаковых ключей
- Секретность ключей обмена
- Уникальность установленных ключей обмена
- Forward secrecy (Прямая секретность)
- Backward secrecy (Обратная секретность)

Протокол Нидхема-Шрёдера

Протокол выработки общего сессионного ключа (N_A, N_B)

| Alice (pkA, privA) | | Bob (pkB, privB) |
|---|--|--|
| <ul style="list-style-type: none">- генерация N_A- шифрует $\langle A, N_A \rangle_{pk(B)}$ | $\langle A, N_A \rangle_{pk(B)}$ -----> | |
| | $\langle N_A, N_B \rangle_{pk(A)}$ <----- | <ul style="list-style-type: none">- дешифровка сообщения $\langle A, N_A \rangle_{pk(B)}$- $\langle N_A, N_B \rangle_{pk(A)}$ |
| <ul style="list-style-type: none">- дешифровка сообщения $\langle N_A, N_B \rangle_{pk(A)}$- $\langle N_B \rangle_{pk(B)}$ | $\langle N_B \rangle_{pk(B)}$ -----> | |
| (N_A, N_B) -- общий секрет | | |

Уязвимость протокола Нидхема-Шрёдера

| Alice | Intruder (pkI, privI) | Bob |
|--|---|--|
| <ul style="list-style-type: none">- генерация N_A- шифрует $\langle A, N_A \rangle_{pk(I)}$ <p>$\langle A, N_A \rangle_{pk(I)}$</p> <p>-----></p> | | |
| | <ul style="list-style-type: none">- дешифрует $\langle A, N_A \rangle_{pk(I)}$- шифрует $\langle A, N_A \rangle_{pk(B)}$ <p>$\langle A, N_A \rangle_{pk(B)}$</p> <p>-----></p> | |
| | | <ul style="list-style-type: none">- дешифровка сообщения $\langle A, N_A \rangle_{pk(B)}$- $\langle N_A, N_B \rangle_{pk(A)}$ <p>$\langle N_A, N_B \rangle_{pk(A)}$</p> <p><-----</p> |

Уязвимость протокола Нидхема-Шрёдера (2)

| | | |
|---|---|--|
| | $\langle N_A, N_B \rangle_{pk(A)}$ ----- <ul style="list-style-type: none">- дешифровка сообщения $\langle N_A, N_B \rangle_{pk(A)}$- $\langle N_B \rangle_{pk(l)}$ $\langle N_B \rangle_{pk(l)}$ -----> | |
| | <ul style="list-style-type: none">- дешифровка $\langle N_B \rangle_{pk(l)}$ $\langle N_B \rangle_{pk(B)}$ -----> | |
| (N_A, N_B) -- общий секрет, который известен <i>l</i> | | |