# Лекция 6. Протоколы аутентификации

Косолапов Ю.В.

ЮФУ

8 октября 2020 г.

# Содержание

# Определение аутентификации

## Определение

**Аутентификация** (authentication) – установление (то есть проверка и подтверждение) подлинности различных аспектов информационного взаимодействия: содержания и источника передаваемых сообщений, сеанса связи, времени взаимодействия и т. д.

- Является важной составной частью проблемы обеспечения достоверности получаемой информации.
- Особенно остро эта проблема стоит в случае не доверяющих друг другу сторон, когда источником угроз может служить не только третья сторона (противник), но и сторона, с которой осуществляется информационное взаимодействие (нарушитель).

# Виды аутентификации

# Виды аутентификации

- **Аутентификация абонента (пользователя)** – доказательство абонентом соответствия своему имени как участника протокола.

# Виды аутентификации

- **Аутентификация абонента (пользователя)** – доказательство абонентом соответствия своему имени как участника протокола.
  - *односторонняя*
  - *взаимная*
  - *интерактивная*

# Виды аутентификации

- **Аутентификация абонента (пользователя)** – доказательство абонентом соответствия своему имени как участника протокола.
  - *односторонняя*
  - *взаимная*
  - *интерактивная*
- **Аутентификация сообщения** – проверка того, что сообщение было получено неповрежденным, неизмененным (с момента отправления), то есть проверка целостности.

# Виды аутентификации

- **Аутентификация абонента (пользователя)** – доказательство абонентом соответствия своему имени как участника протокола.
  - *односторонняя*
  - *взаимная*
  - *интерактивная*
- **Аутентификация сообщения** – проверка того, что сообщение было получено неповрежденным, неизмененным (с момента отправления), то есть проверка целостности.
- **Аутентификация источника данных** – проверка и подтверждение того, что набор данных (сообщение, документ) был создан именно заявленным источником. Не надо путать с аутентификацией отправителя, так как он мог передать документ, созданный и подписанный другим лицом.

# Определение идентификации

## Определение

**Идентификация (identification)** – процедура установления присвоенного данной стороне уникального системного имени — идентификатора, которое позволяет отличать ее от других сторон.

# Определение идентификации

## Определение

**Идентификация (identification)** – процедура установления присвоенного данной стороне уникального системного имени — идентификатора, которое позволяет отличать ее от других сторон.

- Обычно процедура идентификации заключается в предъявлении этого имени и предшествует процедуре аутентификации, то есть подтверждению правильности идентификации.

# Определение идентификации

## Определение

**Идентификация (identification)** – процедура установления присвоенного данной стороне уникального системного имени — идентификатора, которое позволяет отличать ее от других сторон.

- Обычно процедура идентификации заключается в предъявлении этого имени и предшествует процедуре аутентификации, то есть подтверждению правильности идентификации.
- Этот термин часто для краткости используют для обозначения общей процедуры идентификации/аутентификации сторон.

## Example

**Идентификация пользователя** – присвоение пользователям идентификаторов и проверка вхождения предъявляемых идентификаторов в список присвоенных идентификаторов. Обязательно должна дополняться аутентификацией — проверкой принадлежности пользователю предъявленного им идентификатора.

# Про обозначения

# Про обозначения

- Ту сторону (того участника протокола), которая доказывает, что предъявленный идентификатор принадлежит именно ей, будем обозначать $P$ (prover) и называть **доказывающей** стороной.

# Про обозначения

- Ту сторону (того участника протокола), которая доказывает, что предъявленный идентификатор принадлежит именно ей, будем обозначать $P$ (prover) и называть **доказывающей** стороной.
- Ту сторону (того участника протокола), которая проверяет, что предъявленный идентификатор принадлежит доказывающей стороне, будем обозначать $V$ (verifier) и называть **проверяющей** стороной.

# Про обозначения

- Ту сторону (того участника протокола), которая доказывает, что предъявленный идентификатор принадлежит именно ей, будем обозначать $P$ (prover) и называть **доказывающей** стороной.
- Ту сторону (того участника протокола), которая проверяет, что предъявленный идентификатор принадлежит доказывающей стороне, будем обозначать $V$ (verifier) и называть **проверяющей** стороной.
- В протоколе аутентификации выделяют две фазы:

# Про обозначения

- Ту сторону (того участника протокола), которая доказывает, что предъявленный идентификатор принадлежит именно ей, будем обозначать $P$ (prover) и называть **доказывающей** стороной.
- Ту сторону (того участника протокола), которая проверяет, что предъявленный идентификатор принадлежит доказывающей стороне, будем обозначать $V$ (verifier) и называть **проверяющей** стороной.
- В протоколе аутентификации выделяют две фазы:
  - **Фаза регистрация пользователя** в информационной системе (ИС) проверяющей стороны: запись в базу $\mathrm{DB}$ ИС идентификатора и набора данных, с помощью которого будет производиться аутентификация.

# Про обозначения

- Ту сторону (того участника протокола), которая доказывает, что предъявленный идентификатор принадлежит именно ей, будем обозначать $P$ (prover) и называть **доказывающей** стороной.
- Ту сторону (того участника протокола), которая проверяет, что предъявленный идентификатор принадлежит доказывающей стороне, будем обозначать $V$ (verifier) и называть **проверяющей** стороной.
- В протоколе аутентификации выделяют две фазы:
  - **Фаза регистрация пользователя** в информационной системе (ИС) проверяющей стороны: запись в базу $\mathrm{DB}$ ИС идентификатора и набора данных, с помощью которого будет производиться аутентификация.
  - **Фаза аутентификации**: с помощью предоставленных пользователем данных и хранящихся в базе ИС данных выполняется проверка.

# Схема аутентификации

**Фаза регистрации**

$P \rightarrow V$ : DATA

$V$ : Запись DATA в DB

**Фаза аутентификации**

$P \rightarrow V$ : DATA

$V$ : CALCULATION

$V \rightarrow P$ : DATA

$P$ : CALCULATION

.....

$V \rightarrow P$ : RESULT

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

- **Фактор знания (What you know?)**. Что-то, что мы знаем: пароль, PIN-код, кличку домашнего таракана и т.п.

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

- **Фактор знания (What you know?)**. Что-то, что мы знаем: пароль, PIN-код, кличку домашнего таракана и т.п.
- **Фактор владения (What you have?)**. Что-то, что мы имеем: токен, смарт-карта, SIM-карта, ключ от комнаты.

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

- **Фактор знания (What you know?)**. Что-то, что мы знаем: пароль, PIN-код, кличку домашнего таракана и т.п.
- **Фактор владения (What you have?)**. Что-то, что мы имеем: токен, смарт-карта, SIM-карта, ключ от комнаты.
- **Фактор свойства (What you are?)**. Что-то, что является частью нас: голос, пальцы, глаза, лицо, клавиатурный почерк (?).

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

- **Фактор знания (What you know?)**. Что-то, что мы знаем: пароль, PIN-код, кличку домашнего таракана и т.п.
- **Фактор владения (What you have?)**. Что-то, что мы имеем: токен, смарт-карта, SIM-карта, ключ от комнаты.
- **Фактор свойства (What you are?)**. Что-то, что является частью нас: голос, пальцы, глаза, лицо, клавиатурный почерк (?).

Если при аутентификации используется один фактор, то это однофакторная аутентификация; если два фактора, то это двухфакторная аутентификация, а если три — трехфакторная.

# Многофакторная аутентификация пользователя

Выделяют три фактора аутентификации пользователя:

- **Фактор знания (What you know?)**. Что-то, что мы знаем: пароль, PIN-код, кличку домашнего таракана и т.п.
- **Фактор владения (What you have?)**. Что-то, что мы имеем: токен, смарт-карта, SIM-карта, ключ от комнаты.
- **Фактор свойства (What you are?)**. Что-то, что является частью нас: голос, пальцы, глаза, лицо, клавиатурный почерк (?).

Если при аутентификации используется один фактор, то это однофакторная аутентификация; если два фактора, то это двухфакторная аутентификация, а если три — трехфакторная.

**Вопрос**

При аутентификации пользователя требуется ввести пароль и кличку домашнего питомца. Сколько факторов используется?

# Слабая аутентификация (на основе фиксированного пароля)

Вариант 1: $\mathrm{DB}=\{\textsc{login:password}\}$

- $P \to V$ : L/p (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

---

# Слабая аутентификация (на основе фиксированного пароля)

Вариант 1: $\mathrm{DB}=\{\textsc{login:password}\}$

- $P \to V$ : $\textsc{l}/\textsc{p}$ (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

---

Вариант 2 (на основе хэш-функции $h$): $\mathrm{DB} = \{\textsc{login} : h(\textsc{password})\}$

- $P \to V$ : $\textsc{l}/\textsc{p}$ (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

---

# Слабая аутентификация (на основе фиксированного пароля)

Вариант 1: $\mathrm{DB}=\{\textsc{login:password}\}$

- $P \to V$ : L/p (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

---

Вариант 2 (на основе хэш-функции $h$): $\mathrm{DB} = \{\textsc{login} : h(\textsc{password})\}$

- $P \to V$ : L/P (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

---

Вариант 3 («подсоленные» пароли): $\mathrm{DB} = \{\textsc{login} : h(\textsc{password} \parallel \textsc{salt})\}$

- $P \to V$: L/P (логин/пароль)
- $V$ : сравнение пароля по базе $\mathrm{DB}$
- $V \to P$ : result.

# Для самостоятельного изучения (и на практических занятиях)

Атаки:

# Для самостоятельного изучения (и на практических занятиях)

Атаки:

- перехват данных в канале передачи данных и последующее использование: поэтому это «слабая» аутентификация (в рамках модели угроз Долева-Яо). Защита — шифрование канала (но требуется предварительно обменяться ключами).

# Для самостоятельного изучения (и на практических занятиях)

Атаки:

- перехват данных в канале передачи данных и последующее использование: поэтому это «слабая» аутентификация (в рамках модели угроз Долева-Яо). Защита — шифрование канала (но требуется предварительно обменяться ключами).
- просмотр вводимого пароля (shoulder surfing attack)

# Для самостоятельного изучения (и на практических занятиях)

Атаки:

- перехват данных в канале передачи данных и последующее использование: поэтому это «слабая» аутентификация (в рамках модели угроз Долева-Яо). Защита — шифрование канала (но требуется предварительно обменяться ключами).
- просмотр вводимого пароля (shoulder surfing attack)
- перебор по словарю (помогают защититься подсоленные пароли)

# Для самостоятельного изучения (и на практических занятиях)

Атаки:

- перехват данных в канале передачи данных и последующее использование: поэтому это «слабая» аутентификация (в рамках модели угроз Долева-Яо). Защита — шифрование канала (но требуется предварительно обменяться ключами).
- просмотр вводимого пароля (shoulder surfing attack)
- перебор по словарю (помогают защититься подсоленные пароли)
- поиск коллизий для хэш-функций (для защиты нужно выбирать криптографически стойкую хэш-функцию)

**Задание**

Разобраться с протоколом аутентификации Лэмпорта на основе одноразовых паролей [a].

[a] Можно найти в книге Алферова, Зубова, Кузьмина и Черемышкина „Основы криптографии“, раздел „Одноразовые пароли“

# Общая схема

# Общая схема

- Доказывающая сторона отправляет проверяющей свой идентификатор, сообщая о намерении пройти процедуру аутентификации.

# Общая схема

- Доказывающая сторона отправляет проверяющей свой идентификатор, сообщая о намерении пройти процедуру аутентификации.
- Проверяющая сторона отправляет «хитрый» запрос (**nonce**) доказывающей стороне.

# Общая схема

- Доказывающая сторона отправляет проверяющей свой идентификатор, сообщая о намерении пройти процедуру аутентификации.
- Проверяющая сторона отправляет «хитрый» запрос (**nonce**) доказывающей стороне.
- Доказывающая сторона должна «правильно» ответить на запрос.

# Общая схема

- Доказывающая сторона отправляет проверяющей свой идентификатор, сообщая о намерении пройти процедуру аутентификации.
- Проверяющая сторона отправляет «хитрый» запрос (**nonce**) доказывающей стороне.
- Доказывающая сторона должна «правильно» ответить на запрос.

**Важно 1**

**Nonce** – это данные, которые используются один раз: счетчик пакетов, временная метка, случайное число из большого диапазона ($\in \{0,1\}^n, n \geqslant 128$).

# Общая схема

- Доказывающая сторона отправляет проверяющей свой идентификатор, сообщая о намерении пройти процедуру аутентификации.
- Проверяющая сторона отправляет «хитрый» запрос (**nonce**) доказывающей стороне.
- Доказывающая сторона должна «правильно» ответить на запрос.

## Важно 1

**Nonce** – это данные, которые используются один раз: счетчик пакетов, временная метка, случайное число из большого диапазона ($\in \{0,1\}^n, n \geqslant 128$).

## Важно 2

Если требуется только уникальность значений, то используют счетчик пакетов или временную метку; если требуется непредсказуемость значения, то используют случайные числа.

# На основе симметричного шифра (версия 0)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 0):

# На основе симметричного шифра (версия 0)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 0):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : \mathrm{NONCE}.$$

# На основе симметричного шифра (версия 0)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 0):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : \mathrm{NONCE}.$$

**Недостаток**

Атакующий может использовать участника $P$ для расшифрования любого текста, зашифрованного на ключе $k_{P,V}$ (decryption oracle).
**Возможное решение**: использовать NONCE специальной структуры: если при расшифровании структура нарушена, то не отвечать (не выполнять третий шаг).

# На основе симметричного шифра (версия 1)

**Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\text{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 1):**

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : E_{k_{P,V}}(\text{NONCE}).$$

# На основе симметричного шифра (версия 1)

**Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\text{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 1):**

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : E_{k_{P,V}}(\text{NONCE}).$$

## Замечание

**В этом протоколе на третьем шаге можно использовать необратимую функцию вместо симметричного шифрования. Например, использовать криптографическую хэш-функцию $h$:**

$$P \to V : h(k_{P,V} \parallel \text{NONCE}).$$

# На основе симметричного шифра (версия 1)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\text{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 1):

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : E_{k_{P,V}}(\text{NONCE}).$$

## Замечание

В этом протоколе на третьем шаге можно использовать необратимую функцию вместо симметричного шифрования. Например, использовать криптографическую хэш-функцию $h$:

$$P \to V : h(k_{P,V} \parallel \text{NONCE}).$$

## Недостаток

В рамках модели Долева-Яо, атакующий знает открытый текст NONCE и соответствующий шифртекст $E_{k_{P,V}}(\text{NONCE})$ (или свертку $h(k_{P,V} \parallel \text{NONCE})$). Накапливая такие пары (для разных сеансов между $P$ и $V$), атакующий может попытаться провести атаку по известному открытому тексту (known plain text attack) и получить информацию о ключе $k_{P,V}$.

$$\{(\text{NONCE}_1, E_{k_{P,V}}(\text{NONCE}_1)), \ldots, (\text{NONCE}_K, E_{k_{P,V}}(\text{NONCE}_K))\} \to k_{P,V}.$$

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

Протокол на основе временной метки (версия 3):

$$P \to V : \mathrm{ID}(P), E_{k_{P,V}}(\mathrm{TIMESTAMP})$$

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

Протокол на основе временной метки (версия 3):

$$P \to V : \mathrm{ID}(P), E_{k_{P,V}}(\mathrm{TIMESTAMP})$$

**Достоинства версии 3**:

- только один раунд вместо трех

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

Протокол на основе временной метки (версия 3):

$$P \to V : \mathrm{ID}(P), E_{k_{P,V}}(\mathrm{TIMESTAMP})$$

**Достоинства версии 3**:

- только один раунд вместо трех
- $V$ не обязан запоминать запросы (NONCE) для каждого пользователя — менее подвержен атакам типа отказа в обслуживании (DoS – denial-of-service)

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

Протокол на основе временной метки (версия 3):

$$P \to V : \mathrm{ID}(P), E_{k_{P,V}}(\mathrm{TIMESTAMP})$$

**Достоинства версии 3**:

- только один раунд вместо трех
- $V$ не обязан запоминать запросы ($\mathrm{NONCE}$) для каждого пользователя — менее подвержен атакам типа отказа в обслуживании (DoS – denial-of-service)

**Недостатки версии 3:**

- $V$ должен принимать только недавние ответы (в рамках некоторого заранее известного промежутка); все равно остается время у атакующего для совершения атаки повтором (replay attack)

# На основе симметричного шифра (версии 2 и 3)

Пусть $k_{P,V}$ — секретный ключ, известный $P$ и $V$, $\mathrm{ID}(P)$ — идентификатор доказывающей стороны. Протокол (версия 2):

$$P \to V : \mathrm{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

Протокол на основе временной метки (версия 3):

$$P \to V : \mathrm{ID}(P), E_{k_{P,V}}(\mathrm{TIMESTAMP})$$

**Достоинства версии 3**:

- только один раунд вместо трех
- $V$ не обязан запоминать запросы (NONCE) для каждого пользователя — менее подвержен атакам типа отказа в обслуживании (DoS – denial-of-service)

**Недостатки версии 3:**

- $V$ должен принимать только недавние ответы (в рамках некоторого заранее известного промежутка); все равно остается время у атакующего для совершения атаки повтором (replay attack)
- необходима синхронизация часов у $P$ и $V$

# Пара важных замечаний (1)

## Важное замечание 1

- Аутентификация позволяет убедиться проверяющей стороне **только** в том, что доказывающая сторона способна правильно отвечать на запросы.

# Пара важных замечаний (1)

## Важное замечание 1

- Аутентификация позволяет убедиться проверяющей стороне **только** в том, что доказывающая сторона способна правильно отвечать на запросы.
- При этом нет гарантии в том, что общение не идет через атакующего. Атакующий может пропускать сообщения «через себя»:

$$P \to M \to V : \mathrm{ID}(P)$$
$$V \to M \to P : E_{k_{P,V}}(\mathrm{NONCE})$$
$$P \to M \to V : E_{k_{P,V}}(\mathrm{NONCE} + 1).$$

# Пара важных замечаний (2)

## Важное замечание 2

- После успешной аутентификации, как правило, вырабатывается **сессионный ключ** $k_s$, на котором шифруются последующие сообщения, при этом в сообщения подмешивается NONCE, который использовался при аутентификации.

# Пара важных замечаний (2)

## Важное замечание 2

- После успешной аутентификации, как правило, вырабатывается **сессионный ключ** $k_s$, на котором шифруются последующие сообщения, при этом в сообщения подмешивается NONCE, который использовался при аутентификации.
- Тем самым происходит **связывание** передаваемых сообщений и процедуры аутентификации. В этом случае говорят об **аутентифицированной сессии**:

# Пара важных замечаний (2)

## Важное замечание 2

- После успешной аутентификации, как правило, вырабатывается **сессионный ключ** $k_s$, на котором шифруются последующие сообщения, при этом в сообщения подмешивается NONCE, который использовался при аутентификации.
- Тем самым происходит **связывание** передаваемых сообщений и процедуры аутентификации. В этом случае говорят об **аутентифицированной сессии**:

Аутентификация (плохо):

$$P \to V : \text{ID}(P)$$
$$V \to P : E_{k_{P,V}}(\text{NONCE})$$
$$P \to V : E_{k_{P,V}}(\text{NONCE} + 1)$$

Обмен данными :

$$P \to V : \text{Данные1}$$
$$V \to P : \text{Данные2}$$

# Пара важных замечаний (2)

## Важное замечание 2

- После успешной аутентификации, как правило, вырабатывается **сессионный ключ** $k_s$, на котором шифруются последующие сообщения, при этом в сообщения подмешивается NONCE, который использовался при аутентификации.
- Тем самым происходит **связывание** передаваемых сообщений и процедуры аутентификации. В этом случае говорят об **аутентифицированной сессии**:

| Аутентификация (плохо): | Аутентификация (хорошо): |
|---|---|
| $P \to V : \text{ID}(P)$ | $P \to V : \text{ID}(P)$ |
| $V \to P : E_{k_{P,V}}(\text{NONCE})$ | $V \to P : E_{k_{P,V}}(\text{NONCE})$ |
| $P \to V : E_{k_{P,V}}(\text{NONCE}+1)$ | $P \to V : E_{k_{P,V}}(\text{NONCE}+1, k_s)$ |
| Обмен данными : | Обмен данными : |
| $P \to V : \text{Данные1}$ | $P \to V : E_{k_s}(\text{Данные1}, \text{NONCE})$ |
| $V \to P : \text{Данные2}$ | $V \to P : E_{k_s}(\text{Данные2}, \text{NONCE})$ |

# На основе асимметричного шифра

Пусть $pk_P$ и $sk_P$ — публичный ключ и секретный ключ доказывающей стороны $P$.
Протокол 1 (на основе шифрования):

# На основе асимметричного шифра

Пусть $pk_P$ и $sk_P$ — публичный ключ и секретный ключ доказывающей стороны $P$.
Протокол 1 (на основе шифрования):

$$P \to V : \textsc{id}(P)$$
$$V \to P : E_{pk_P}(\textsc{nonce})$$
$$P \to V : \textsc{nonce}.$$

# На основе асимметричного шифра

Пусть $pk_P$ и $sk_P$ — публичный ключ и секретный ключ доказывающей стороны $P$.
Протокол 1 (на основе шифрования):

$$P \to V : \text{ID}(P)$$
$$V \to P : E_{pk_P}(\text{NONCE})$$
$$P \to V : \text{NONCE}.$$

Протокол 2 (на основе ЭЦП):

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : SIGN_{sk_P}(\text{NONCE}).$$

# На основе асимметричного шифра

Пусть $pk_P$ и $sk_P$ — публичный ключ и секретный ключ доказывающей стороны $P$.
Протокол 1 (на основе шифрования):

$$P \to V : \text{ID}(P)$$
$$V \to P : E_{pk_P}(\text{NONCE})$$
$$P \to V : \text{NONCE}.$$

Протокол 2 (на основе ЭЦП):

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : SIGN_{sk_P}(\text{NONCE}).$$

## Вопрос

В чем недостаток протокола 2?

# На основе асимметричного шифра

Пусть $pk_P$ и $sk_P$ — публичный ключ и секретный ключ доказывающей стороны $P$.
Протокол 1 (на основе шифрования):

$$P \to V : \text{ID}(P)$$
$$V \to P : E_{pk_P}(\text{NONCE})$$
$$P \to V : \text{NONCE}.$$

Протокол 2 (на основе ЭЦП):

$$P \to V : \text{ID}(P)$$
$$V \to P : \text{NONCE}$$
$$P \to V : SIGN_{sk_P}(\text{NONCE}).$$

## Вопрос

В чем недостаток протокола 2?

## Вывод

Для аутентификации и шифрования (подписи) данных должны использоваться разные пары асимметричных ключей.

# Цель

## Цель протоколов двусторонней аутентификации

Имеются два участника: A и B. Каждый из них намерен аутентифицировать другого (перед началом сеанса). Таким образом, каждый будет выступать и в роли доказывающей стороны, и в роли проверяющей стороны.

# На основе симметричной криптографии

Протокол 1 (на основе симметричного шифра, общий секретный ключ $k_{A,B}$)

$$A \to B : \mathrm{ID}(A), \mathrm{NONCE}_A$$
$$B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$$
$$A \to B : E_{k_{A,B}}(\mathrm{NONCE}_B)$$

**Проблема!**

Этот протокол подвержен атаке отражением (reflection attack).

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)
- $I_B \to A : \mathrm{ID}(B), \mathrm{NONCE}_A$

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)
- $I_B \to A : \mathrm{ID}(B), \mathrm{NONCE}_A$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}'_A$

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)
- $I_B \to A : \mathrm{ID}(B), \mathrm{NONCE}_A$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}'_A$
- $I_B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)
- $I_B \to A : \mathrm{ID}(B), \mathrm{NONCE}_A$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}'_A$
- $I_B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_B)$

# Атака отражением для протокола 1

**Цель атаки**: атакующий $I$ подменяет легального участника $B$ и должен убедить $A$ в том, что $A$ общается с $B$; $I_B$ – это атакующий $I$ в роли $B$.

- $A \to I_B : \mathrm{ID}(A), \mathrm{NONCE}_A$ ($A$ думает, что общается с $B$)
- $I_B \to A : \mathrm{ID}(B), \mathrm{NONCE}_A$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}'_A$
- $I_B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$
- $A \to I_B : E_{k_{A,B}}(\mathrm{NONCE}_B)$

## Замечание

Атака проходит в том случае, если каждый может инициировать процедуру аутентификации.

# Частичная защита от атаки

В момент аутентификации передавать секретный сессионный ключ $k_s$, на котором далее шифровать передаваемые данные.

# Частичная защита от атаки

В момент аутентификации передавать секретный сессионный ключ $k_s$, на котором далее шифровать передаваемые данные.

$$A \to B : \mathrm{ID}(A), \mathrm{NONCE}_A$$
$$B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$$
$$A \to B : E_{k_{A,B}}(\mathrm{NONCE}_B \parallel k_s)$$
$$A \to B : E_{k_s}(\mathrm{DATA})$$

# Частичная защита от атаки

В момент аутентификации передавать секретный сессионный ключ $k_s$, на котором далее шифровать передаваемые данные.

$$A \to B : \mathrm{ID}(A), \mathrm{NONCE}_A$$
$$B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A), \mathrm{NONCE}_B$$
$$A \to B : E_{k_{A,B}}(\mathrm{NONCE}_B \parallel k_s)$$
$$A \to B : E_{k_s}(\mathrm{DATA})$$

## Замечание

При атаке отражением сторона $A$ подумает, что общается с $B$, но при этом атакующий не сможет прочитать сообщения, зашифрованные на ключе $k_s$.

## Вывод

Протокол взаимной атуентификации не должен быть симметричным. Участники не должны быть взаимозаменяемыми!

# Некоторые способы исключения симметричности

# Некоторые способы исключения симметричности

- Использовать NONCE специальной структуры, например, для $A$ – четные, а для $B$ – нечетные.

# Некоторые способы исключения симметричности

- Использовать $\mathrm{NONCE}$ специальной структуры, например, для $A$ – четные, а для $B$ – нечетные.
- Включать в ответ на $\mathrm{NONCE}$ идентификатор отправителя, зашифрованный вместе с $\mathrm{NONCE}$.

$$A \to B : \mathrm{ID}(A), \mathrm{NONCE}_A$$
$$B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A \parallel \mathrm{ID}(B)), \mathrm{NONCE}_B$$
$$A \to B : E_{k_{A,B}}(\mathrm{NONCE}_B \parallel \mathrm{ID}(A))$$

# Некоторые способы исключения симметричности

- Использовать $\mathrm{NONCE}$ специальной структуры, например, для $A$ – четные, а для $B$ – нечетные.
- Включать в ответ на $\mathrm{NONCE}$ идентификатор отправителя, зашифрованный вместе с $\mathrm{NONCE}$.

$$A \to B : \mathrm{ID}(A), \mathrm{NONCE}_A$$
$$B \to A : E_{k_{A,B}}(\mathrm{NONCE}_A \parallel \mathrm{ID}(B)), \mathrm{NONCE}_B$$
$$A \to B : E_{k_{A,B}}(\mathrm{NONCE}_B \parallel \mathrm{ID}(A))$$

**Общий принцип двусторонних протоколов**

Инициатор протокола **сначала** должен доказать свою подлинность (соответствие предъявляемому идентификатору), а уже после этого инициатор может проверять идентичность другого участника.

# Домашнее задание

Какие недостатки у следующего протокола (на основе временных меток)?

$$A \to B : \text{ID}(A), E_{k_{A,B}}(\text{ID}(A) \parallel \text{TIMESTAMP})$$
$$B \to A : E_{k_{A,B}}(\text{ID}(B) \parallel (\text{TIMESTAMP} + 1))$$

# Протокол Нидхема-Шрёдера

Кроме аутентификации генерируется ключ для последующего шифрования данных.

# Протокол Нидхема-Шрёдера

Кроме аутентификации генерируется ключ для последующего шифрования данных.
Ключи участников:

- $A$: $(pk_A, sk_A)$
- $B$: $(pk_B, sk_B)$

# Протокол Нидхема-Шрёдера

Кроме аутентификации генерируется ключ для последующего шифрования данных.

Ключи участников:

- $A$: $(pk_A, sk_A)$
- $B$: $(pk_B, sk_B)$

Протокол:

$$A \to B : \text{ID}(A), E_{pk_B}(\text{ID}(A) \parallel \text{NONCE}_A)$$
$$B \to A : E_{pk_A}(\text{NONCE}_A \parallel \text{NONCE}_B)$$
$$A \to B : E_{pk_B}(\text{NONCE}_B \parallel k_s)$$
$$A \to B : E_{k_s}(\text{DATA} \parallel \text{ID}(A) \parallel \text{COUNTER})$$
$$\ldots$$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $I \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $I \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $A \to I : E_{pk_I}(\mathrm{NONCE}_B \parallel k_s)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $I \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $A \to I : E_{pk_I}(\mathrm{NONCE}_B \parallel k_s)$
- $I_A \to B : E_{pk_B}(\mathrm{NONCE}_B \parallel k_s)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $I \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $A \to I : E_{pk_I}(\mathrm{NONCE}_B \parallel k_s)$
- $I_A \to B : E_{pk_B}(\mathrm{NONCE}_B \parallel k_s)$
- $I_A \to B : E_{k_s}(\ldots)$

# Атака на протокол Нидхема-Шрёдера

**Цель атаки**: заставить участника $B$ думать, что он общается с $A$, а не с атакующим $I$.

- $A \to I : \mathrm{ID}(A), E_{pk_I}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $I_A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$
- $B \to I_A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $I \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{NONCE}_B)$
- $A \to I : E_{pk_I}(\mathrm{NONCE}_B \parallel k_s)$
- $I_A \to B : E_{pk_B}(\mathrm{NONCE}_B \parallel k_s)$
- $I_A \to B : E_{k_s}(\ldots)$

**Важно!**

Участник $A$ знает, что общается с $I$. Это нормально в рамках модели Долева-Яо.

# Протокол Нидхема-Шрёдера (исправленная версия)

# Протокол Нидхема-Шрёдера (исправленная версия)

Ключи участников:

- $A$: $(pk_A, sk_A)$
- $B$: $(pk_B, sk_B)$

# Протокол Нидхема-Шрёдера (исправленная версия)

Ключи участников:

- $A$: $(pk_A, sk_A)$
- $B$: $(pk_B, sk_B)$

Протокол:

$$A \to B : \mathrm{ID}(A), E_{pk_B}(\mathrm{ID}(A) \parallel \mathrm{NONCE}_A)$$
$$B \to A : E_{pk_A}(\mathrm{NONCE}_A \parallel \mathrm{ID}(B) \parallel \mathrm{NONCE}_B)$$
$$A \to B : E_{pk_B}(\mathrm{NONCE}_B \parallel k_s)$$
$$A \to B : E_{k_s}(\mathrm{DATA} \parallel \mathrm{ID}(A) \parallel \mathrm{COUNTER})$$
$$\ldots$$

# Заключение

Спасибо за внимание!