О криптографии

Сергей Николенко

Криптография — CS Club, осень 2009

Outline

Криптология

- Наука о шифрах — криптология.
- Она делится на две дисциплины:
  - *криптография*: как засекретить что-то;
  - *криптоанализ*: как рассекретить то, что засекретил криптограф.

Криптография

- Криптография — $\kappa\rho\upsilon\pi\tau o + \gamma\rho\alpha\phi\eta$.
- Главный смысл: передавать сообщения между участниками криптографического протокола так, чтобы другие не смогли их понять.
- Но есть и другие задачи.

Основные задачи криптографии

- *Конфиденциальность*: как сохранить информацию в секрете от всех, кроме имеющих доступ:
  - передача данных по незащищённому каналу;
  - хранение данных на общедоступных носителях.

Основные задачи криптографии

- *Конфиденциальность*: как сохранить информацию в секрете от всех, кроме имеющих доступ:
  - передача данных по незащищённому каналу;
  - хранение данных на общедоступных носителях.
- *Целостность*: как обеспечить передачу данных в целости и сохранности. В частности, как заметить, менял ли кто-то данные по дороге.

Основные задачи криптографии

- *Конфиденциальность*: как сохранить информацию в секрете от всех, кроме имеющих доступ:
  - передача данных по незащищённому каналу;
  - хранение данных на общедоступных носителях.
- *Целостность*: как обеспечить передачу данных в целости и сохранности. В частности, как заметить, менял ли кто-то данные по дороге.
- *Аутентификация*: как доказать, что данные поступают из правильного источника. Две части:
  - entity authentication: как доказать, что я — это я;
  - data origin authentication: как доказать, что моё сообщение — действительно от меня.

Основные задачи криптографии

- *Конфиденциальность*: как сохранить информацию в секрете от всех, кроме имеющих доступ:
  - передача данных по незащищённому каналу;
  - хранение данных на общедоступных носителях.
- *Целостность*: как обеспечить передачу данных в целости и сохранности. В частности, как заметить, менял ли кто-то данные по дороге.
- *Аутентификация*: как доказать, что данные поступают из правильного источника. Две части:
  - entity authentication: как доказать, что я — это я;
  - data origin authentication: как доказать, что моё сообщение — действительно от меня.
- *Non-repudiation* (неотречение): как сделать так, чтобы человек, что-то пообещавший, потом от обещаний не отказывался.

Некриптографические методы

- Если «противник» — бездушный канал с ошибками, то есть некриптографические методы.
- Передача данных — коды, обнаруживающие ошибки и коды, исправляющие ошибки.
- Целостность данных — контрольные суммы. CRC (cyclic redundancy code, циклический избыточный код):
  - сообщению $a_0 a_1 \dots a_{N-1}$ сопоставляем многочлен $P(x) = \sum_{i=0}^{N-1} a_i x^i$;
  - значение CRC — остаток от деления $P(x)$ на $G(x)$, которым определяется CRC.

Основные термины

- Для традиционной задачи: сообщение (plaintext) кодируется (is encrypted) в код/шифр (ciphertext). Затем код декодируется (is decrypted) обратно в сообщение (plaintext):

$$\text{plaintext} \xrightarrow{encryption} \text{ciphertext} \xrightarrow{decryption} \text{plaintext}$$

Сложные задачи

- Декодировать должно быть *сложно*. О том, что значит «сложно», мы ещё будем говорить.
- В современной криптографии, как правило, «сложно» означает, что многие пытались сделать быстрый алгоритм для решения задачи декодирования, но никто пока что не преуспел.
- Более серьёзных безусловных гарантий современная криптография дать не может, потому что про P=NP пока не известно.

Атаки

- Что может делать враг? Против чего мы должны готовиться?

Атаки

- Что может делать враг? Против чего мы должны готовиться?

1. *Ciphertext only*:
   - враг увидел и скопировал некоторое количество шифров, которые он теперь может анализировать;
   - у врага достаточно много таких шифров.

Атаки

- Что может делать враг? Против чего мы должны готовиться?

1. *Ciphertext only*.
2. *Known plaintext*:
   - у врага есть некоторое количество пар $\langle$сообщение, шифр$\rangle$, которые он теперь может анализировать;
   - например, с течением времени содержание старых сообщений становится известным.

Атаки

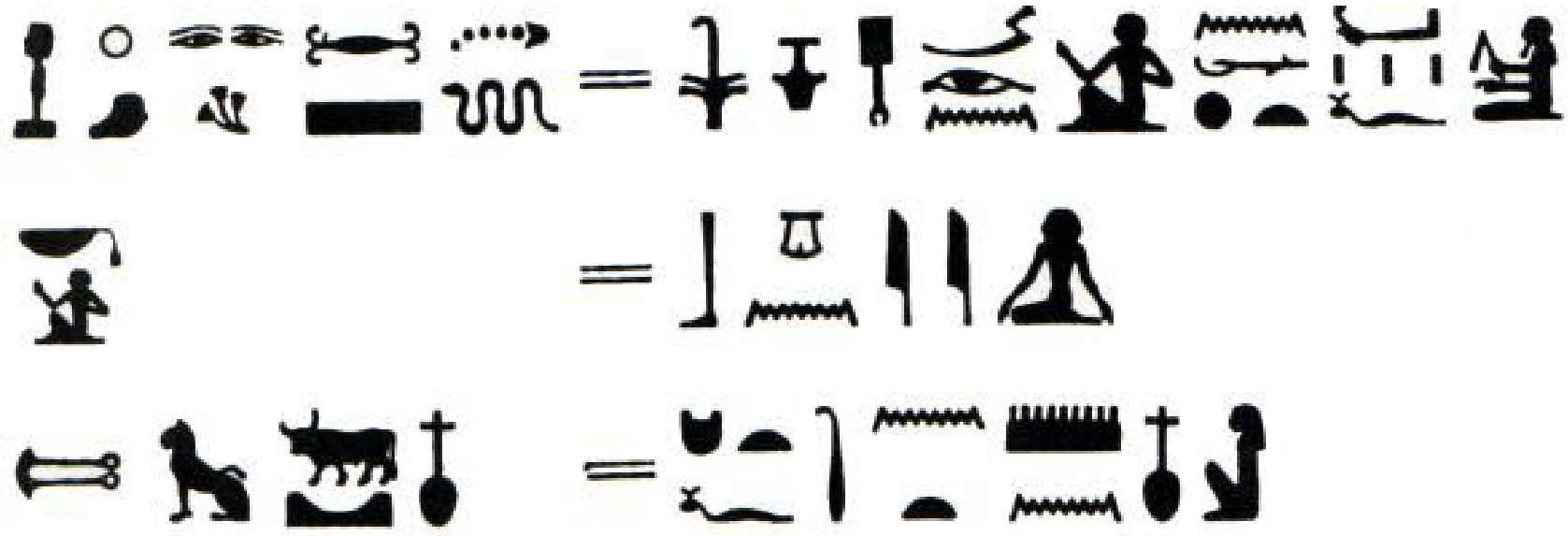
- Что может делать враг? Против чего мы должны готовиться?

1. *Ciphertext only*.
2. *Known plaintext*.
3. *Chosen plaintext*:
   - враг может сам выбрать несколько сообщений и закодировать их при помощи этого алгоритма;
   - например, кодирование — общедоступный сервис.

Outline

Древние цивилизации

- Около 4000 лет назад египтяне заменяли некоторые иероглифы в важных текстах на другие.
- Их было нетрудно расшифровать; видимо, цель была не в сокрытии информации.



Древние цивилизации

- Китай: Ву Джинг Зонг Яо (1044) содержал не только формулу пороха, но и небольшой код для военных целей, но вообще не развито было шифрование.
- Индия: «Камасутра» содержит искусство тайнописи как одно из 64 искусств (йог), рекомендованных для женщин.
- Да и вообще во всех культурах: курды-езиды (против мусульман), жители Тибета, тайцы и т.д.

Греция

- Лисандр: скиталы — кожаная полоска наматывается на цилиндр определённой толщины.

Греция

- Диск Энея (Эней Тактик, 4 в. до н.э.): на диске просверливаются дырки, соответствующие буквам, через них продевается в нужном порядке нить. Расшифровка тривиальна, но сообщение можно мгновенно уничтожить. Он же — книжный шифр: незаметные пометки над буквами книги (первая стеганография).
- Квадрат Полибия: в квадрат выписываются буквы, каждая буква заменяется на ту, что под ней. Шифр — порядок букв в этом квадрате.
- Ещё стеганография: Геродот упоминает, как сообщение о планирующейся атаке персов было записано на основе восковых табличек, которые потом опять покрыли воском.

Шифр Цезаря

- Шифр Цезаря — простой вариант шифра подстановки. Буквы сдвигаются на фиксированное число позиций: A – D, B – E, C – F и т.д.
- Талмуд — *атбаш* (алеф–таф, бет–шин): первая буква заменяется на последнюю, вторая — на предпоследнюю и т.д. В Библии: «лев камай» (сердце моих противников) — «халдеи», «Шешах» — «Вавилон».

Моноалфавитные шифры

- Шифр Цезаря и атбаш — частные случаи *моноалфавитных шифров*.
- В них каждой букве алфавита ставится в соответствие другая буква или символ другого алфавита.
- Т.е. моноалфавитный шифр — перестановка букв алфавита или биекция с другим алфавитом.
- Как взломать моноалфавитный шифр?

Пляшущие человечки

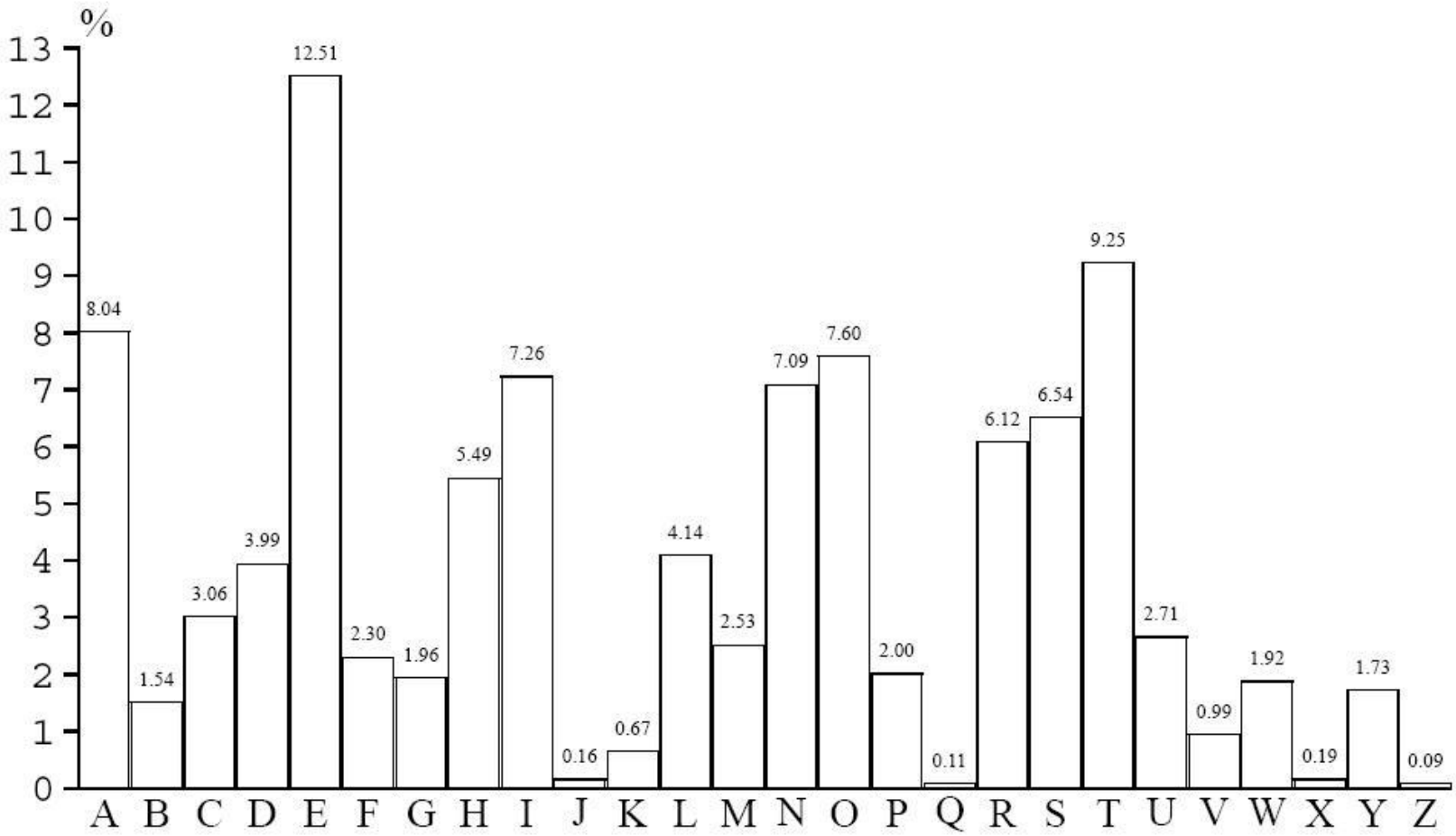
- Пример моноалфавитного шифра:

- Холмс разгадал его, применив *частотный анализ*.

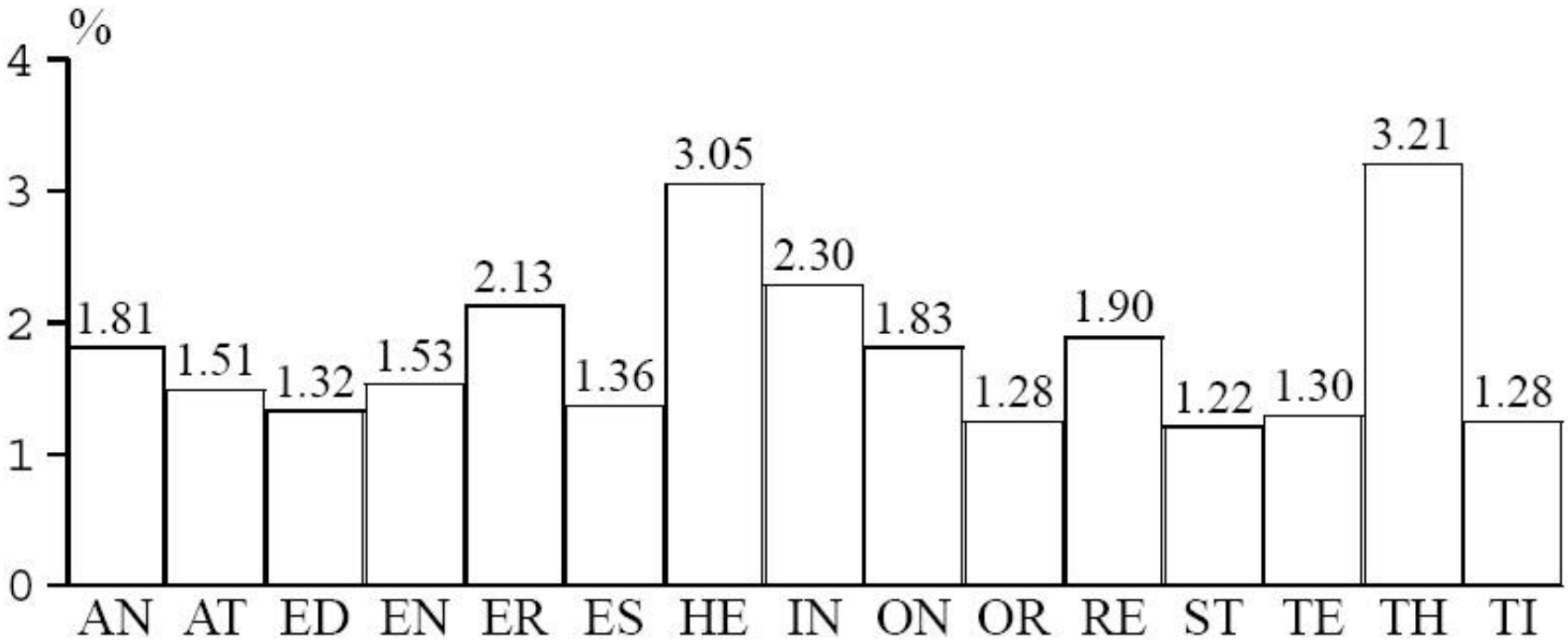
Арабы

- Частотный анализ появился в Аравии.
- 14-томная энциклопедия Ахмада аль-Калькашанди (1412) содержала раздел по криптологии.
- Таблицы сочетаемости букв арабского языка, какая после какой наиболее вероятна, какие буквы самые частые (алиф и лам, конечно). Всё это — на основе Корана.
- Частотным анализом можно взломать моноалфавитные шифры; с биграммами уже сложнее, но тоже можно.

Частотный анализ: моноалфавитный шифр



Частотный анализ: биграммы



Новое время

- До Возрождения криптография считалась «тёмным искусством» и смешивалась с Каббалой.
- С XVI века появилась дипломатия, которой понадобились секретные сообщения.
- Франсуа Виет был в том числе и криптоаналитиком, помогал Генриху IV.
- Английские криптоаналитики на службе Уолсингема расшифровали письма Марии Стюарт и обвинили её в измене.
- И так далее...

Полиалфавитные шифры

- Леон Батиста Альберти — архитектор, художник, композитор, писатель.
- Первый полиалфавитный шифр: внутри код, снаружи сообщение. Время от времени двигаем внутренний диск, изменяя тем самым код.

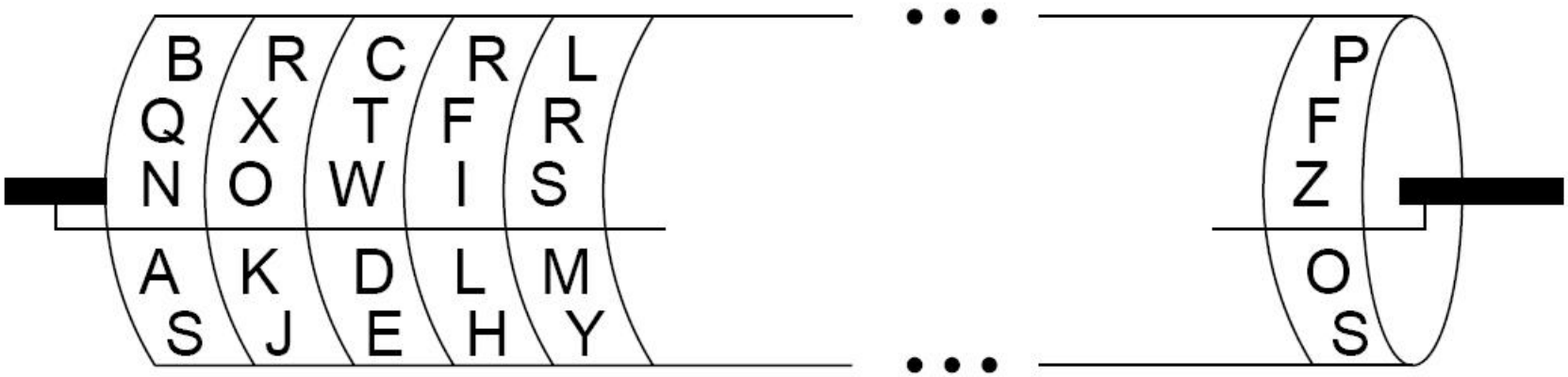
Полиалфавитные шифры

- Иоганн Трисемус, 1508: «Полиграфия», первый труд по криптологии.
- Шифр Трисемуса — квадрат, *tabula recta*, в котором записан алфавит со смещением.
- Первая буква кодируется по первой строке, вторая по второй и т.д.
- Блез де Виженер, 1586 — первым предложил кодировать сообщение им самим: первая расшифрованная буква используется для декодирования второй и т.д.
- Стали появляться секретные отделы криптологов при дворах.

XIX век

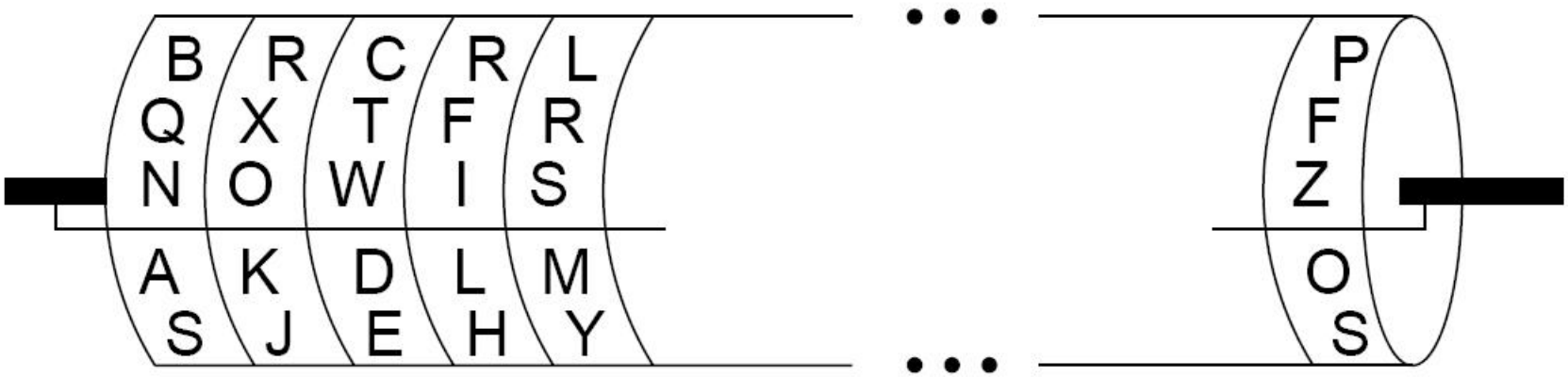
- Многое изменилось с изобретением телеграфа.
- Теперь нужно было кодировать большие объёмы сообщений.
- Прежние шифры были слишком трудоёмки. Перешли на простые коды, секретность достигалась частой сменой кодовых слов. Но были и новые шифры.
- Чарльз Уитстон — Playfair cipher.

Цилиндр Джефферсона



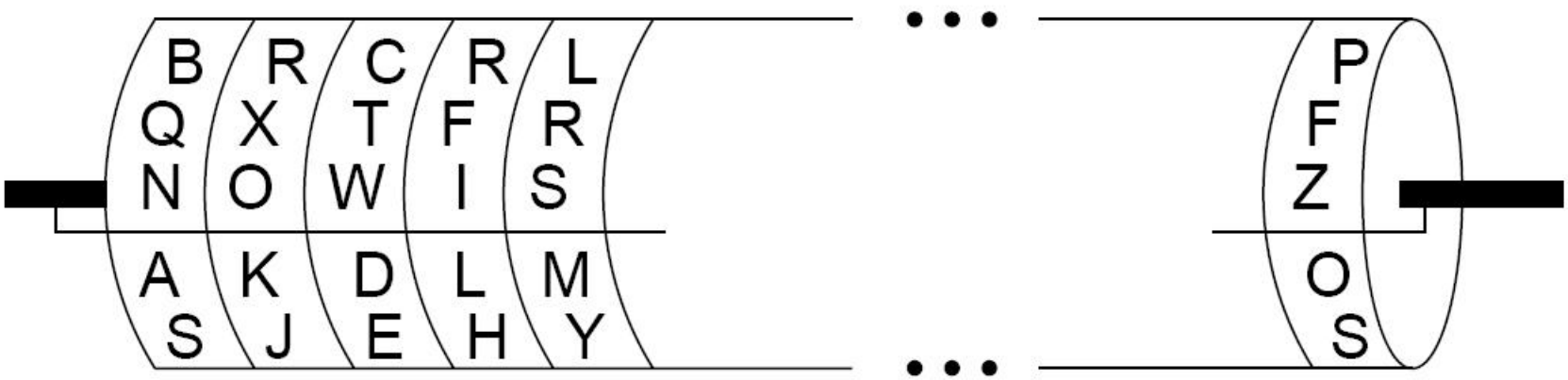
- Томас Джефферсон — «wheel cipher», «цилиндр Джефферсона», очень изящный метод построения полиалфавитного шифра.

Цилиндр Джефферсона



- Код — цилиндр; чтобы зашифровать, нужно повернуть цилиндры так, чтобы получилось сообщение, и записать какую-нибудь другую строчку.
- Чтобы дешифровать, нужно повернуть цилиндры так, чтобы получился код, и поискать строчку, которая имеет смысл как сообщение.

Цилиндр Джефферсона



- Значительно позже появились другие машины, основанные на роторах.
- Самая известная — немецкая «Энигма».

Криптоанализ

- Фридрих Касиски (1863) — метод взлома полиалфавитных шифров.
- Август Керхкоф (1883) — «La Cryptographic militaire»; принципы шифрования:
  - система должна быть невзламываема если не в теории, то на практике;
  - алгоритм шифрования может стать известным противнику, и это не должно привести ко взлому системы;
  - ключ должно быть легко запомнить и легко изменить;
  - криптограммы должно быть возможно передавать по телеграфу;
  - кодирование и декодирование должен быть в состоянии делать один человек.
- Удивительно современные принципы.

Мировые войны

- Во время мировых войн криптоанализ сыграл важнейшую роль.
- Ещё больше, чем телеграф, на криптографию повлияло радио. Теперь можно было перехватывать большие объёмы вражеских сообщений.
- WWI: Британия, Room 40. Из-за блокады перехватывали все немецкие сообщения, и многие шифры успешно декодировали.
- В частности, декодировали и показали американцам планы Германии заключить союз с Мексикой, после чего США вошли в войну.
- WWII: Bletchley Park. Работали многие математики и криптографы (Алан Тьюринг).

Шифр Вернама

- 1910-е годы: Гильберт Вернам:
  - улучшил шифр Виженера;
  - разработал шифр, который *невозможно* взломать. Как это?

Шифр Вернама

- 1910-е годы: Гильберт Вернам:
  - улучшил шифр Виженера;
  - разработал шифр, который *невозможно* взломать. Как это?
- *Одноразовый блокнот*: используем одноразовый секретный ключ k, который просто складываем побитово с сообщением:

$$c = m \oplus k.$$

  Без знаний о ключе и сообщении враг, перехвативший сообщение, получил ровно ноль информации.
- Это, конечно, доказал уже Шеннон в конце 1940-х.

Появление криптографии с публичным ключом

- Whitfield Diffie, Martin Hellman, 1976: «New directions in cryptography». Ralph Merkle. Протокол согласования ключа Диффи-Хеллмана.
- Ron Rivest, Adi Shamir, Leonard Adleman, 1978: RSA, первая криптосистема с открытым ключом.

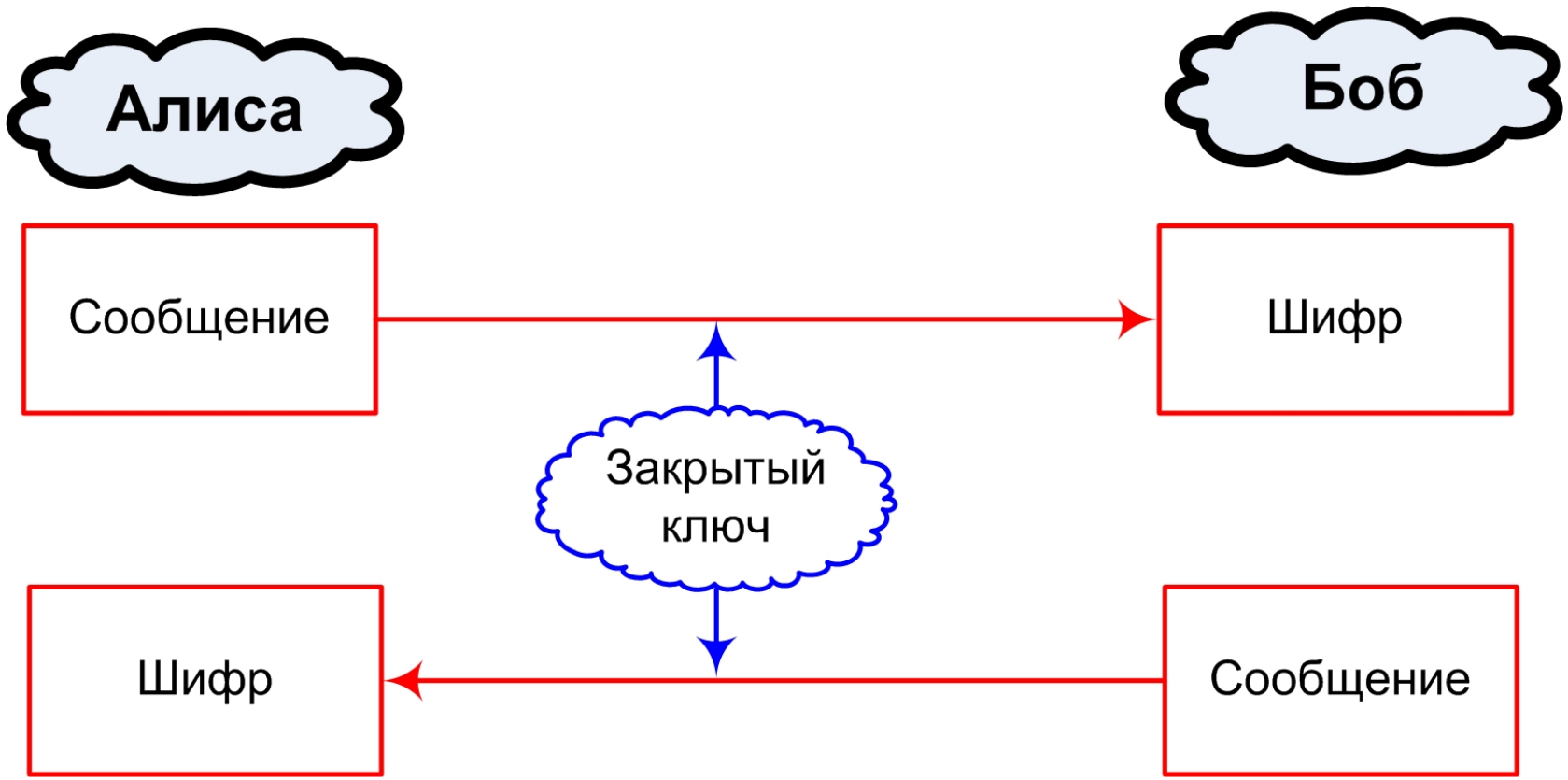
Outline

Криптография в целом

1. Криптография без ключа.
   1. Хеш-функции.
   2. Односторонние перестановки.
   3. Случайные и псевдослучайные последовательности.
2. Криптография с закрытым ключом.
   1. Блочные шифры.
   2. Поточные шифры.
3. Криптография с открытым ключом.
   1. Базовые задачи криптографии с открытым ключом.
      1. Разложение на множители и дискретный логарифм в $\mathbb{Z}_n$.
      2. Дискретный логарифм на эллиптических кривых.
      3. Алгоритмы решения этих задач.
   2. Криптографические примитивы.
      1. Протоколы согласования ключа.
      2. Криптосистемы.
      3. Доказательства с нулевым разглашением.
      4. Разделение секрета.

Криптография с закрытым ключом

- В криптографии с закрытым ключом у двух сообщающихся сторон есть один ключ, который они никому не сообщают.

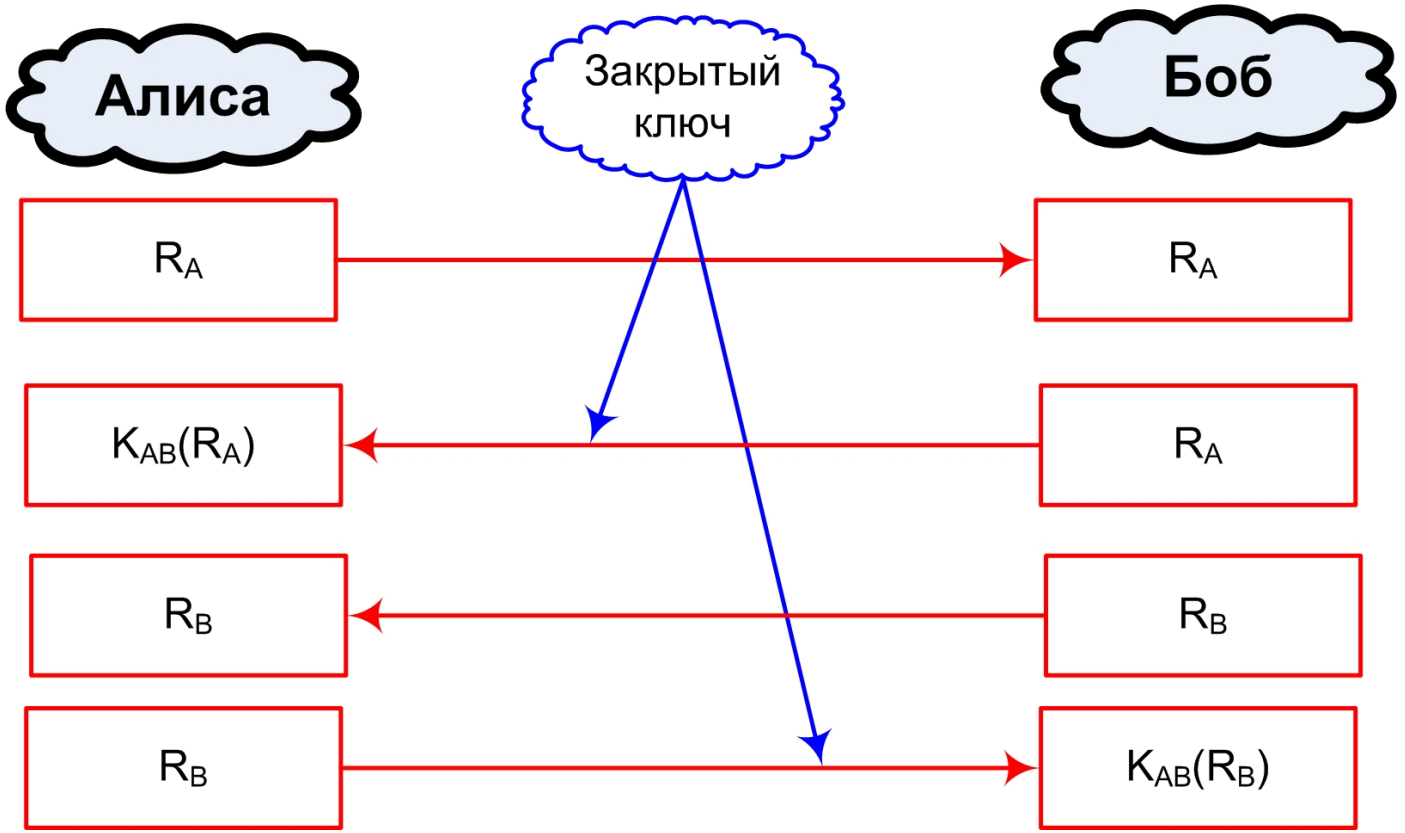


Криптография с закрытым ключом

- Передача по незащищённому каналу: Алиса и Боб кодируют сообщения секретным ключом.
- Хранение: Алиса кодирует своим секретным ключом.
- Аутентификация: как?

Криптография с закрытым ключом

- Алиса просит Боба закодировать секретным ключом, а Боб — Алису. Правда, всё не так просто, но об этом позже.

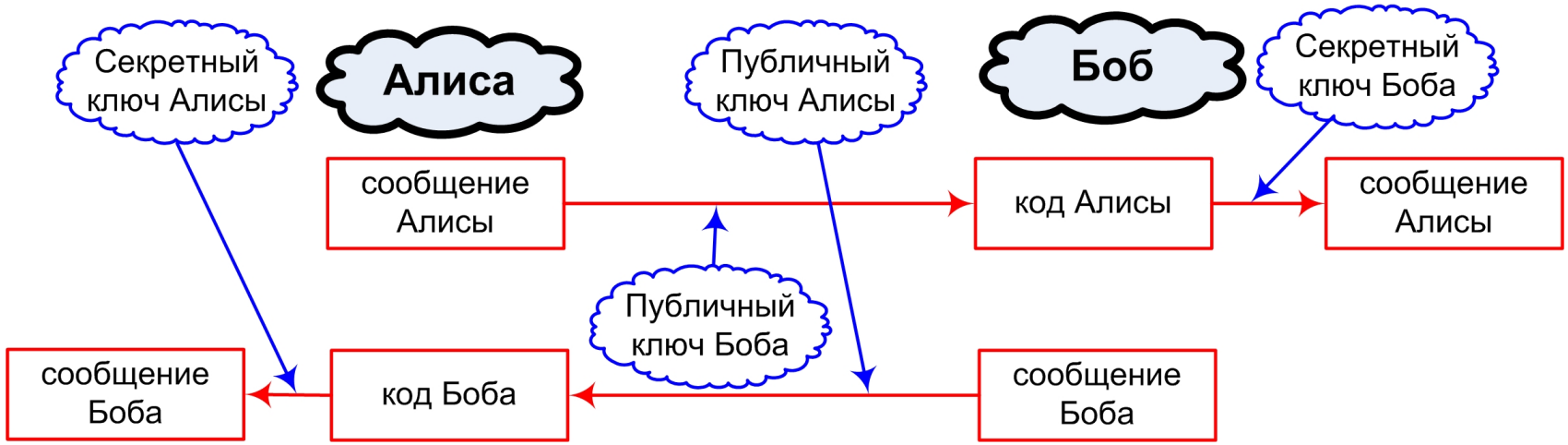


Криптография с закрытым ключом

- Проверка целостности: генерируем криптографический checksum к сообщению, который другие не могут подделать.
- Например, Алиса вычисляет CRC, а затем кодирует его своим секретным ключом.

Криптография с открытым ключом

- В криптографии с открытым ключом у каждого участника два ключа: секретный и публичный.
- Вот передача сообщений:



Криптография с открытым ключом

- Хранение: так же, как с закрытым ключом. Алиса кодирует своим секретным ключом.
- Аутентификация: Алиса передаёт случайное число, Боб его расшифровывает и передаёт обратно.

Криптография с открытым ключом

- Электронная подпись: Алиса может подписать сообщение так, что каждый может проверить подпись, но никто не может подделать подпись. Полезно:
  - Можно проверить, что сообщение осталось прежним (проверка целостности).
  - Алиса не может отказаться от того, что это её сообщение (non-repudiation).

Хеш-функции

- Хеш-функция (hash, message digest) — функция h, которая преобразует сообщение в хеш постоянной длины, причём:
  - $h(m)$ легко вычислить;
  - по $h(m)$ трудно найти m, которое ему соответствует;
  - трудно найти такие m_1 и m_2, что $h(m_1) = h(m_2)$.
- Хеширование паролей.
- Проверка целостности.

О стойкости хеш-функций

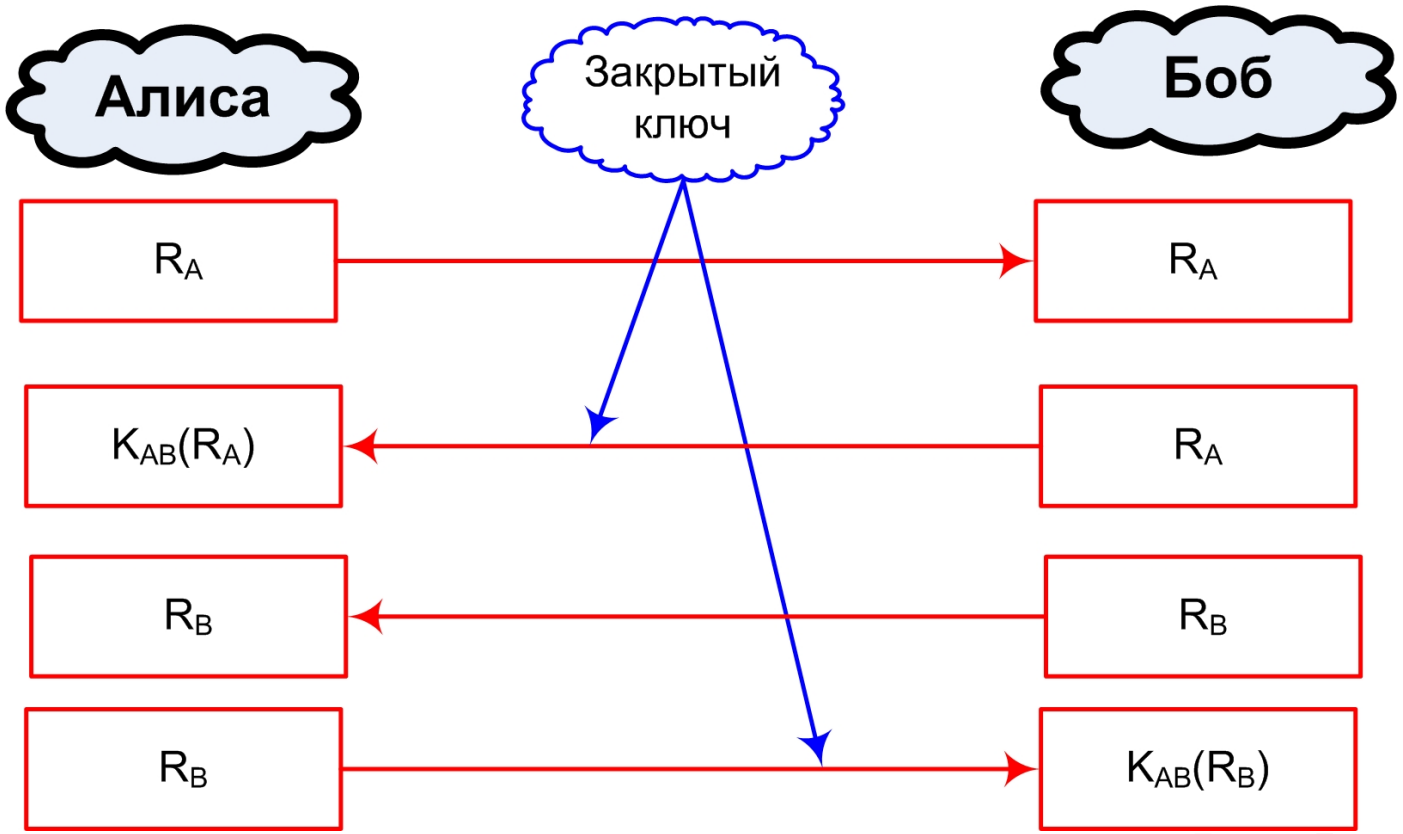
- Рассмотрим хеш-функцию, производящую хеш длины n.
- Сколько нужно операций врагу, чтобы скомпрометировать хеш-функцию?

О стойкости хеш-функций

- Рассмотрим хеш-функцию, производящую хеш длины n.
- Сколько нужно операций врагу, чтобы скомпрометировать хеш-функцию?
- Чтобы грубой силой найти такие m_1 и m_2, что $h(m_1) = h(m_2)$, нужно в среднем $2^{n/2}$ операций.
- Birthday problem: пусть есть n входов и k возможных выходов (в birthday problem — 366); тогда есть $n(n-1)/2$ пар; вероятность совпадения одной пары — $1/k$, значит, нужно около $k/2$ пар, чтобы приблизиться к $1/2$; следовательно, n достаточно выбрать порядка $\sqrt{k}$.
- Поэтому хеш надо делать вдвое длиннее, чем соответствующий ему по стойкости секретный ключ.

От примитива к протоколу

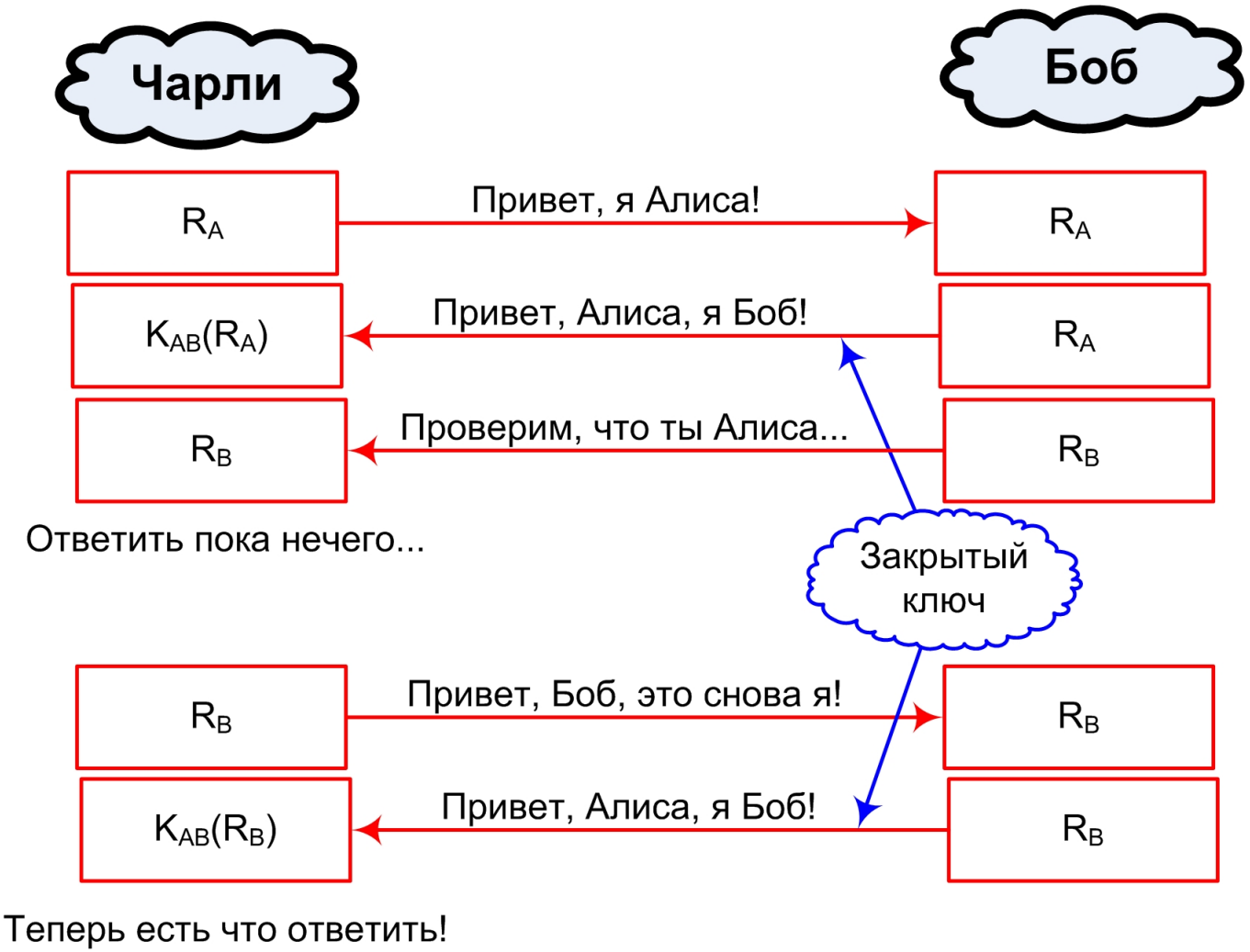
- Кроме того, хорошие криптографические примитивы не всегда дают надёжные протоколы.
- Пример: криптосистема с закрытым ключом.



Беда с этим протоколом

- Это не очень хороший протокол.
- Пусть Боб — это компьютер, и наш противник Чарли умеет инициировать несколько сессий.

Беда с этим протоколом



Дизайн протоколов

- Поэтому ещё одна важная задача криптографии — как от надёжного криптографического примитива перейти к надёжному протоколу.
- Это задача не только математическая; здесь же — разные соображения о том, как ненадёжных людей заставить пользоваться надёжными криптографическими протоколами.

Пароли

- Например, пароли. Предположим, что мы хотим достичь стойкости в 64 бита, т.е. заставить противника перебирать 2^{64} вариантов.
- Случайно сгенерированная строка из [A-Za-z0-9] — 64 варианта на символ (6 битов), итого нужно 11 символов.
- Вы готовы запоминать строку из 11 случайных символов?

Пароли

- Например, пароли. Предположим, что мы хотим достичь стойкости в 64 бита, т.е. заставить противника перебирать 2^{64} вариантов.
- Случайно сгенерированная строка из [A-Za-z0-9] — 64 варианта на символ (6 битов), итого нужно 11 символов.
- Случайно сгенерированная произносимая строка — только [a-z], примерно каждый третий символ — гласная.
- Стойкость получается около 4 битов на символ, надо 16 символов. Тоже многовато.

Пароли

- Например, пароли. Предположим, что мы хотим достичь стойкости в 64 бита, т.е. заставить противника перебирать 2^{64} вариантов.
- Случайно сгенерированная строка из [A-Za-z0-9] — 64 варианта на символ (6 битов), итого нужно 11 символов.
- Случайно сгенерированная произносимая строка — только [a-z], примерно каждый третий символ — гласная.
- Стойкость получается около 4 битов на символ, надо 16 символов. Тоже многовато.
- Если же позволять пользователю самому выбирать пароль, стойкость будет около 2 битов на символ. То есть надо бы 32 символа...

Спасибо за внимание!

- Lecture notes и слайды будут появляться на моей homepage:

  `http://logic.pdmi.ras.ru/~sergey/`

- Присылайте любые замечания, решения упражнений, новые численные примеры и прочее по адресам: `sergey@logic.pdmi.ras.ru`, `snikolenko@gmail.com`
- Заходите в ЖЖ  **smartnik**.