Математические и компьютерные основы защиты информации

Лекция 5

Антон Николаевич Гайдук | УНИВЕР

vk.com/gaidukedu

09 марта 2023 г.

Содержание дисциплины

Раздел I Введение

- Тема 1. Введение. История. Основные понятия.

Раздел II Симметричная криптография

- Тема 2 Классические шифры.
- Тема 3 Поточные алгоритмы шифрования.
- Тема 4 Блочные алгоритмы шифрования.

- Тема 5 Функции хэширования.
- Тема 6 Математические методы криптоанализа.

Раздел III Асимметричная криптография

- Тема 7 Протокол Диффи-Хэллмана.
- Тема 8 Криптосистемы с открытым ключом.
- Тема 9 Электронная цифровая подпись.

Раздел II Симметричная криптография

Тема 5 Функции хэширования.

- Понятие функции хэширования
- Примеры функции хэширования
- Атака дней рождения на хэш функцию
- Блочно-итерационные функции хэширования
- Конструкция Меркля-Дамгарда
- Применение хэш-функций

$$\mathcal{A}^* \longrightarrow \mathcal{A}^n$$

История: хэш-функции

- 1953, Hans Peter Luhn, IBM: идея
- 1956, Arnold Dumey, US Army, NSA: $N \pmod p$.
- 1968, Robert Morris, Bell Labs, NSA: Communications of the ACM — термин "hashing".

VIRUSTOTAL

Analyse suspicious files, domains, IPs and URLs to detect malware and other breaches, automatically share them with the security community.

FILE | URL | SEARCH

Computing hash 83%

Хэш-функция

Хэш-функцией или функцией хэширование называется отображение

$$h : \{0,1\}^* \to \{0,1\}^n, n \in \mathbb{N}.$$

Требования

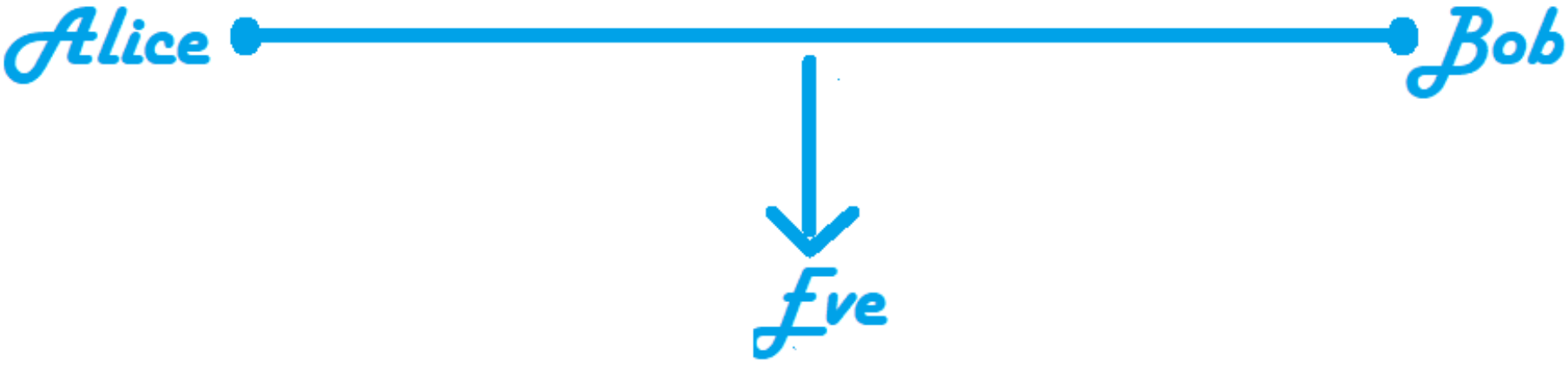
- Быстрота вычислений (полиномиальная от длины входного слова сложность),
- Минимум коллизий.

Примеры

- $N \pmod p$,
- Контрольные суммы: CRC (cyclic redundancy code) CRC8, CRC32, CRC64
- алгоритм хэширования Пирсона для строк

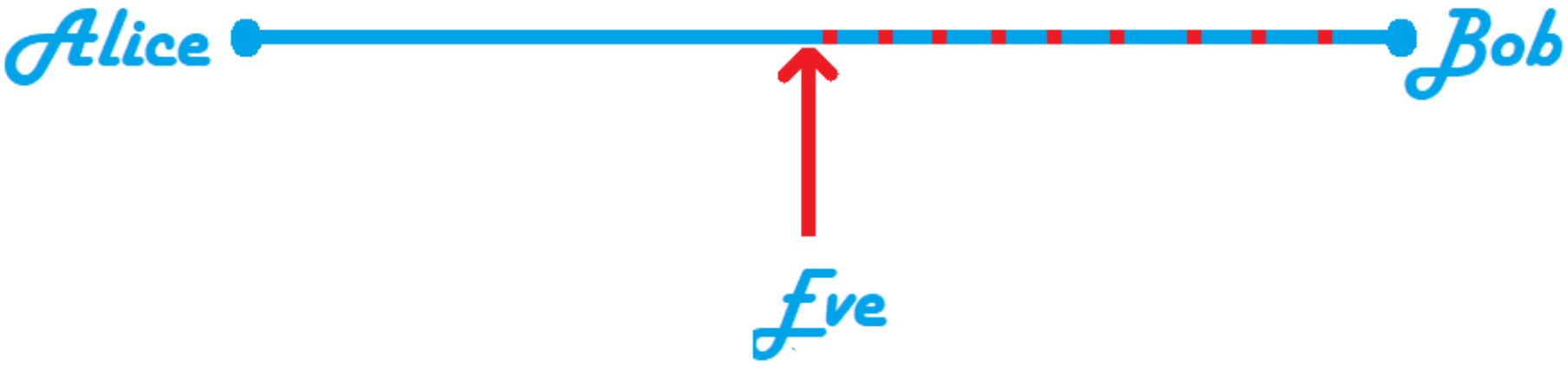
Криптографические хэш-функции

Угрозы в открытом канале связи



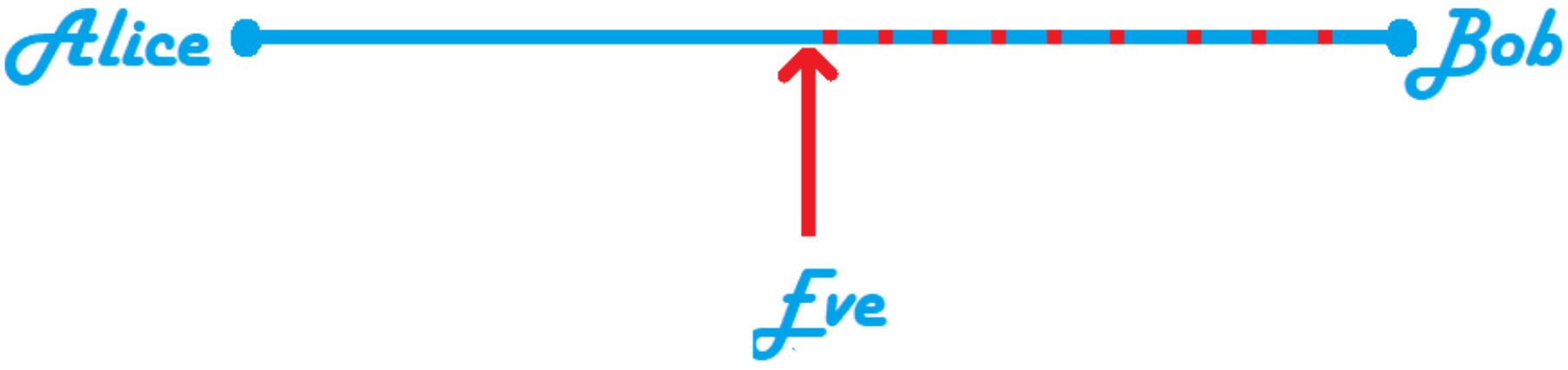
- Ева перехватывает сообщения (нарушение конфиденциальности)

Угрозы в открытом канале связи



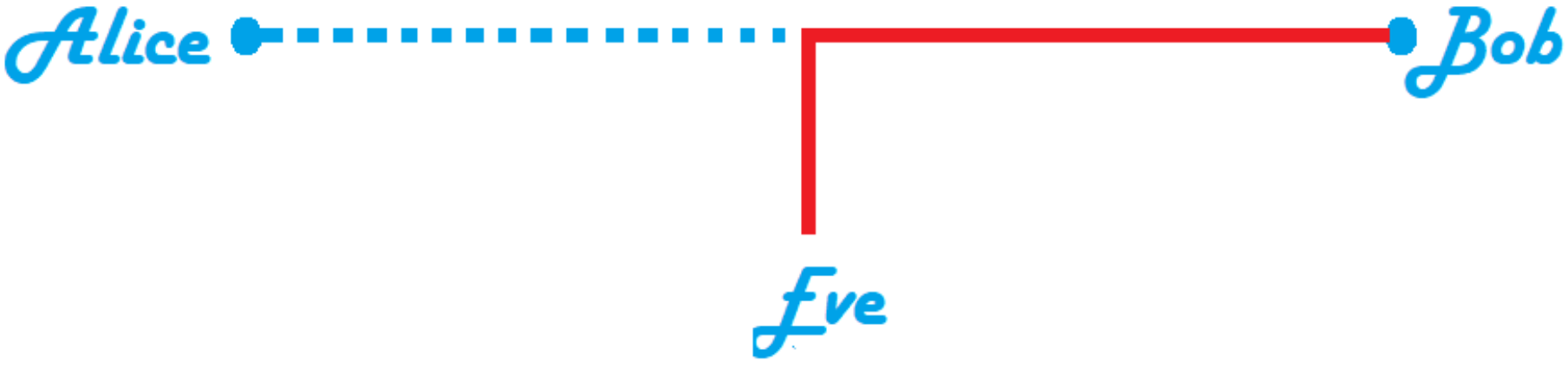
- Ева перехватывает сообщения (нарушение конфиденциальности)
- Ева модифицирует сообщения (нарушение целостности)

Угрозы в открытом канале связи



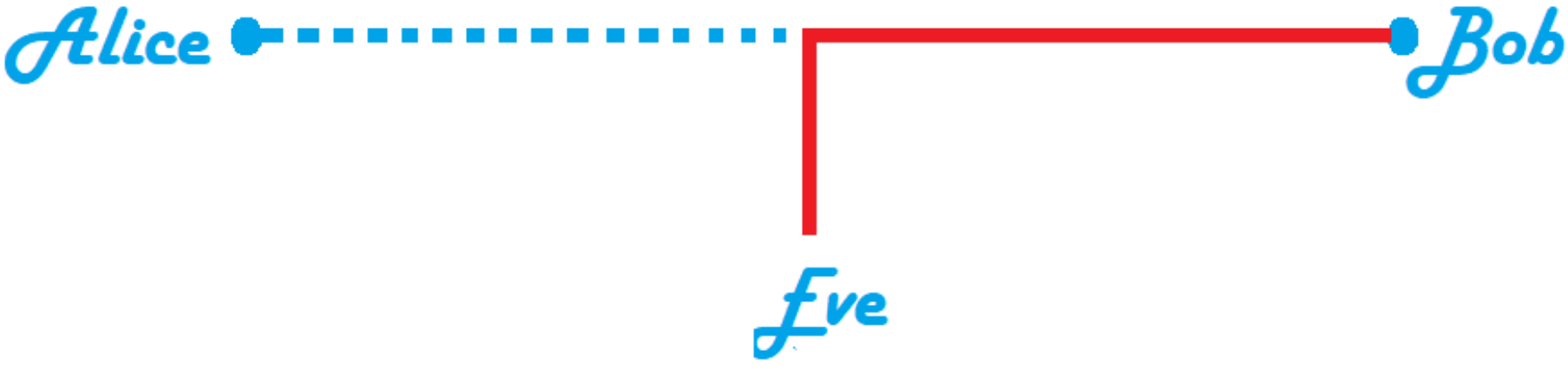
Контроль целостности информации — обнаружение модификации информации

Угрозы в открытом канале связи



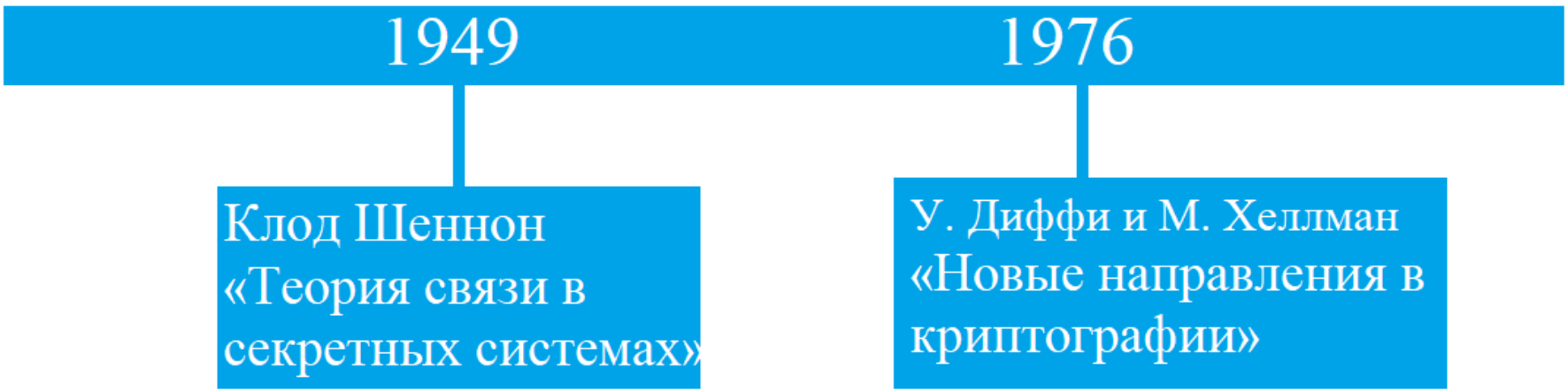
- Ева перехватывает сообщения (нарушение конфиденциальности)
- Ева модифицирует сообщения (нарушение целостности)
- Ева фальсифицирует сообщения (подделка авторства)

Угрозы в открытом канале связи



Аутентификация – подтверждение подлинности сторон (идентификация) и самой информации в процессе информационного взаимодействия. Информация, передаваемая по каналу связи, должна быть аутентифицирована по источнику, времени создания, содержанию данных, времени пересылки и т.д.

История


1949
1976
Клод Шеннон «Теория связи в секретных системах»
У. Диффи и М. Хеллман «Новые направления в криптографии»

Diffie and Hellman, 1976. New directions in cryptography

- Необходимость односторонней функции для схемы ЭЦП
- More precisely, a function f is a one-way function if, for any argument x in the domain of f, it is easy to compute the corresponding value $f(x)$, yet, for almost all y in the range of f, it is computationally infeasible to solve the equation $y = f(x)$ for any suitable argument x.
- "Let g be a one-way mapping from binary N-space to binary n-space...". "Take the N bit message m and operate on it with g to obtain the n bit vector m'."
- "It must be hard even given m to find a different inverse image of m'".

Пример one-way function

Блочный или поточный алгоритм шифрования.

История: криптографические хэш функции

- 1978, Rabin: 64 битная конструкция на основе блочного алгоритма DES
- 1979, Yuval: построение коллизии (за время $2^{n/2}$ на основе парадокса дней рождений)
- 1979, Merkle:

Стойкостью к нахождению прообраза хэш-функции называется вычислительная сложность алгоритма, который по заданному хэш-значению находит соответствующее ему входное сообщение.

Стойкостью к нахождению второго прообраза хэш-функции называется вычислительная сложность алгоритма нахождения любого другого прообраза, который давал бы такое же хэш-значение, как и заданный.

Стойкостью к нахождению коллизии хэш-функции называется вычислительная сложность алгоритма, который находит два входных сообщения имеющих одинаковые хэш-значения.

Хэш функции: задачи криптоанализа

H1 По заданному $Y = h(X)$ определить X.

H2 Для заданного X найти $X' \neq X$ такой, что $h(X) = h(X')$.

H3 Найти различные X и X' такие, что $h(X) = h(X')$.

Задача	Название	Мотивация	Стойкая хэш-функция
H1	Обращение	найти пароль X по данным аутентификации $h(X\|\|S)$	односторонняя
H2	Определение 2-го прообраза	Подмена файла X' на X	свободная от коллизий
H3	Построение коллизии	Подбираем два различных документа — X (подлинный) и X' (поддельный). Отдаем X на подпись (ЭЦП), потом подпись X присоединяем к X'	строго свободная от коллизий

Хэш функции: задачи криптоанализа

Всякий алгоритм, который находит 2-й прообраз, является также алгоритмом построения коллизии. Поэтому если h строго свободна от коллизий, то h свободна от коллизий.

Модель случайного оракула

В криптографии случайным оракулом называется идеализированная хеш-функция, которая на каждый новый запрос выдает случайный ответ, равномерно распределённый по области значений, с условием: если один и тот же запрос поступит дважды, то ответ должен быть одинаковым.

... a good hash function behaves as a random oracle ...

Если существует атака на хэш-функцию, сложность которой ниже, чем сложность атаки для случайного оракула, то хэш-функция считается уязвимой к данной атаке.

Парадокс «дней рождения»

Пусть $\mathcal{A} = \{a_1, a_2, \ldots, a_N\}$, $\quad a_i \neq a_j$ для $i \neq j$.
Из множества $\mathcal{A}$ случайным образом, независимо выбираются m элементов с возвратом.

Например, $N = 7, m = 4$: $\mathcal{A} = \{a_1, a_2, \ldots, a_7\} \to a_2, a_5, a_7, a_5$.

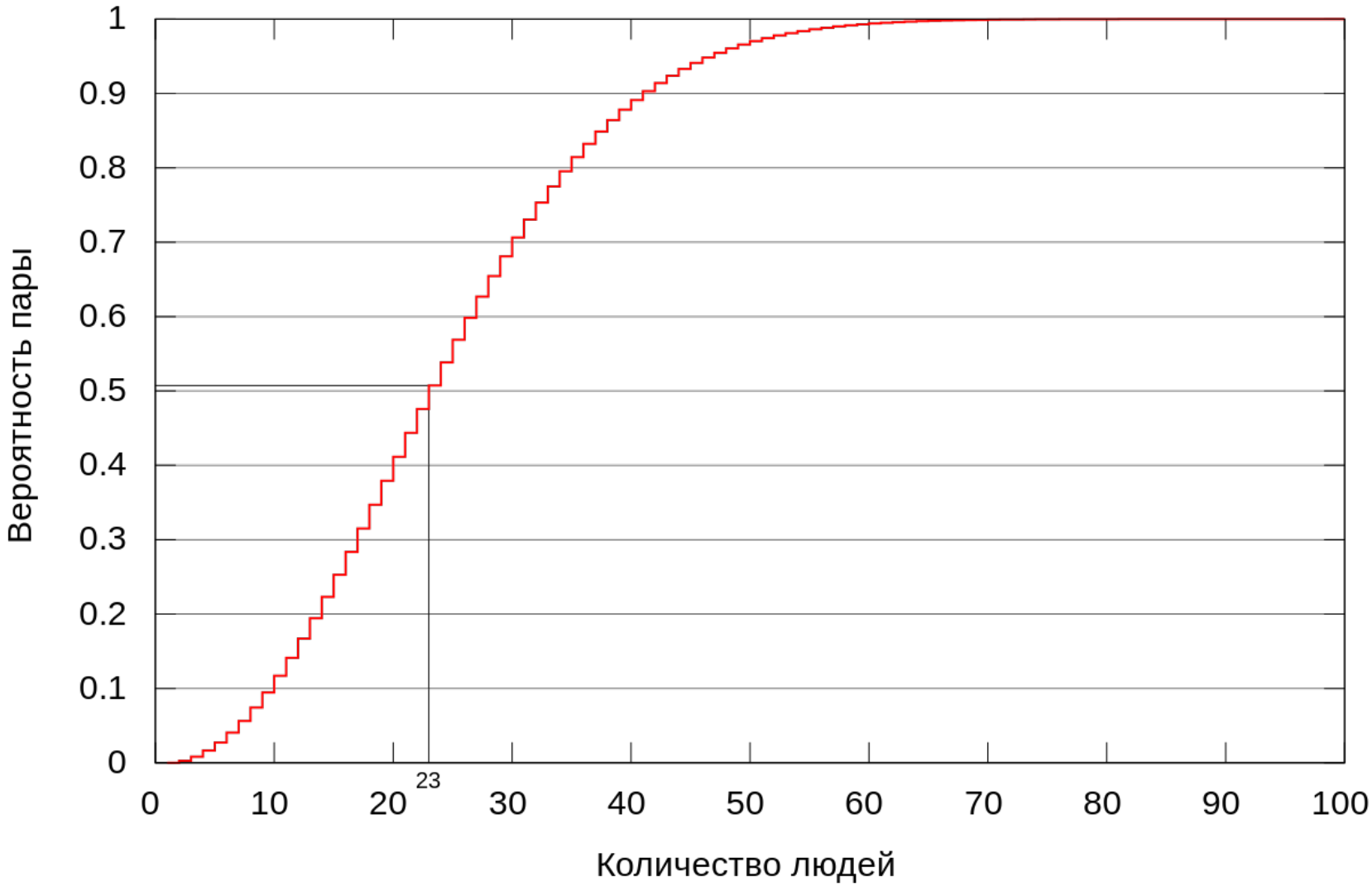
Тогда, вероятность события состоящего в том, что среди выбранных m элементов найдется хотя бы одна коллизия, т.е. найдутся $1 \leqslant i < j \leqslant m$ такие, что $a_i = a_j$ равна:

$$p = 1 - \prod_{i=0}^{m-1} \frac{N-i}{N} \geqslant 1 - \prod_{i=0}^{m-1} e^{-\frac{i}{N}} \geqslant 1 - e^{-\frac{m(m-1)}{2N}}, \qquad (1 - x \leqslant e^{-x}).$$

При $m(m-1) = 2N$:

$$p \geqslant 1 - e^{-1} \approx 0,63.$$

Парадокс «дней рождения»



Атака «дней рождения» на хэш-функцию

Пусть n — длина хэш-значения.
Базовый алгоритм:

1. Выбрать конечное множество $\mathcal{X} \subset \{0,1\}^*$ мощностью $|\mathcal{X}| \gg 2^n$.
2. Зарезервировать массив $\mathtt{H}$ из 2^n ячеек памяти. В ячейках размещаются элементы из $\mathcal{X}$, ячейки индексируются словами из $\{0,1\}^n$: $\mathtt{H}[y]$ — ячейка по индексу y. Первоначально все ячейки заполняются символом $\perp$ (пусто).
3. $X \xleftarrow{R} \mathcal{X}$.
4. $y \leftarrow h(X)$.
5. Если $\mathtt{H}[y] \neq \perp$ и $X \neq \mathtt{H}[y]$, то коллизия найдена: вернуть $(X, \mathtt{H}[y])$.
6. $\mathtt{H}[y] \leftarrow X$, перейти к шагу 3.

Среднее время построения коллизии (обращений к функции h):

$$\sim 2^{n/2}.$$

Примеры хэш-функций

Функция хэширования	Год	Длина хэш-значения в битах	Размер внутреннего состояния
MD2	1989	128	128
MD5	1991	128	128
RIPEMD-160	1992	160	160
SHA-1	1993	160	160
SHA-2	2001	224, 256, 384, 512	256/512
SHA-3	2015	224, 256, 384, 512	1600

Построение хэш-функций

Построение хэш-функций

- на основе блочных алгоритмов шифрования,
- на основе методов теории чисел,
- на основе теории конечных автоматов.

Хэш-функции на основе блочных шифров

- естественный подход,
- доказательство стойкости хэш-функции основано на стойкости блочного шифра.

но, появляются слабости связанные с ключевым расписанием блочного алгоритма шифрования, которые не влияют на стойкость шифрования, но влияют на стойкость хэш-функции.

Хэш-функции на основе блочных шифров

- Данные обрабатываются блоками фиксированной длины w,
- Идея: итерационный процесс,
- Используется функция сжатия σ (шаговая функция хэширования),
- Обоснование стойкости хэш-функции сводится к обоснованию стойкости функции сжатия σ.

Базовый алгоритм: $\sigma : \{0,1\}^{w+n} \to \{0,1\}^n$
вход $X \in \{0,1\}^* \to X\|0^m$, $m = w - |X| \bmod w$ (паддинг).

$$X\|0^m \to X_1\| \ldots \|X_T, X_i \in \{0,1\}^w.$$

$X_{T+1} = |X|$ — усиление Меркля-Дамгарда

1. $y \leftarrow y_0$ — начальное значение.
2. Для $i = \overline{1, T+1}$: $y \leftarrow \sigma(X_i\|y)$.
3. вернуть y.

Теорема

Пусть в схеме Меркля-Дамгарда шаговая функция хэширования σ строго свободна от коллизий. Тогда построенная на ее основе хэш-функция также строго свободна от коллизий.

Функция сжатия на основе блочных шифров

- Пусть $E = \{E_k : k \in \mathcal{K}\}$ — блочный шифр,
- Пусть длина блока хэшируемых данных, длина хэш-значения, длина блока и длина ключа блочного шифра E совпадают.
- Тогда функцию сжатия σ можно строить по схеме:

$$\sigma(X\|y) = E_{\alpha_1 X \oplus \alpha_2 y}(\alpha_3 X \oplus \alpha_4 y) \oplus \alpha_5 X \oplus \alpha_6 y,$$

где $X, y \in \{0,1\}^n$, $\alpha_i \in \{0,1\}, i = \overline{1,6}$ — фиксированные константы.

Не всякий выбор констант дает криптографически стойкую функцию σ

Например, $\sigma(X\|y) = E_y(X \oplus y) \oplus y$ — не является односторонней. Действительно, для заданного $h \in \{0,1\}^n$ выбираем произвольное y и находим $X = E_y^{-1}(h \oplus y) \oplus y$. Тогда

$$\sigma(X\|y) = E_y(X \oplus y) \oplus y = E_y(E_y^{-1}(h \oplus y) \oplus y \oplus y) \oplus y = h \oplus y \oplus y = h.$$

Функция сжатия на основе блочных шифров

№	$\sigma(X\|\|y)$	Название
1	$E_y(X) \oplus X$	Матиаса - Мейера - Озеаса
2	$E_y(X \oplus y) \oplus X \oplus y$	
3	$E_y(X) \oplus X \oplus y$	Миягучи - Приниля
4	$E_y(X \oplus y) \oplus X$	
5	$E_X(y) \oplus y$	Дэвиса - Мейера
6	$E_X(X \oplus y) \oplus X \oplus y$	
7	$E_X(y) \oplus X \oplus y$	
8	$E_X(X \oplus y) \oplus y$	
9	$E_{X \oplus y}(X) \oplus X$	
10	$E_{X \oplus y}(y) \oplus y$	LOKI
11	$E_{X \oplus y}(X) \oplus y$	
12	$E_{X \oplus y}(y) \oplus X$	

Применение хэш-функций

Применение хэш-функций

Контрольные суммы

Вычисляется хэш-значение файла $X \in \{0,1\}^*$. Последующее совпадение сохраненного хэш-значения с $h(X)$ служит подтверждением того, что файл X не был изменен.

```
Get-FileHash e:\download\MiKOZI03.pdf

Algorithm : SHA256
Hash      : 8D9067E97E101B41B316471A391690446198F9564FA1A6B86332DB8A57B1974A
Path      : e:\download\MiKOZI03.pdf
```

Применение хэш-функций

Построение ключей

По паролю $X \in \{0,1\}^*$ строится секретный ключ $k = h(X)$.

Применение хэш-функций

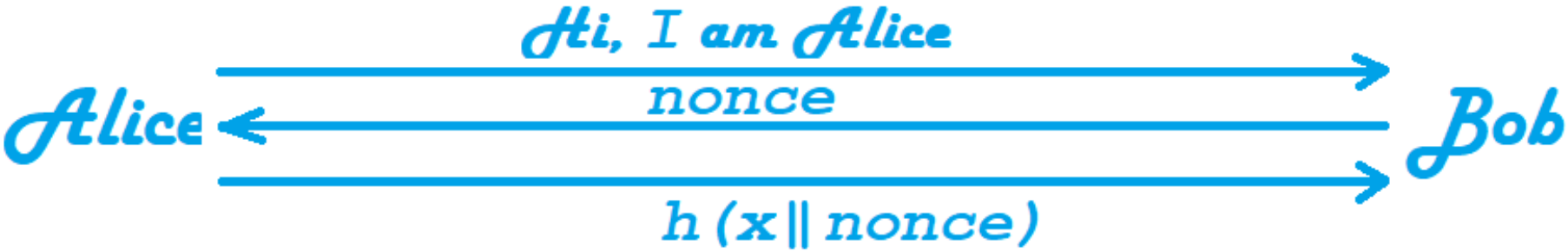
Генерация псевдослучайных чисел

$$y_i = h(k \| s_i),$$

где k — секретный ключ, s_i — неповторяющиеся числа.

Применение хэш-функций

Аутентификация



Имитозащита (HMAC)

Имитосистема: $H = \{h_k : k \in \mathcal{K}\}$,
$h_k : \mathcal{K} \times \{0,1\}^* \to \{0,1\}^n$ — ключезависимая хэш-функция:

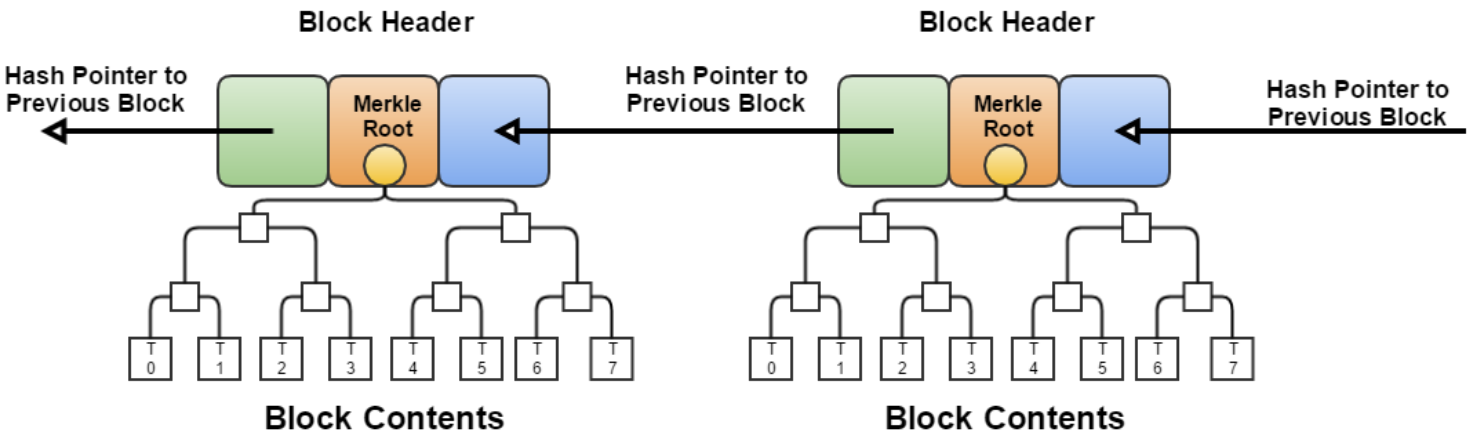
$$h_k(X) = h((k \oplus \alpha) \| h((k \oplus \beta) \| X)) \qquad k \in \{0,1\}^n,$$

$\alpha =$ `0x5c 0x5c 0x5c ... 0x5c`,
$\beta =$ `0x36 0x36 0x36 ... 0x36`.

`HMAC-MD5, HMAC-SHA1, HMAC-RIPEMD160, HMAC-SHA256`

Применение хэш-функций

Технология блокчейн



Основные свойства структуры данных блокчейн:

- данные группируются в блоки;
- линейное свойство: блокчейн имеет линейную структуру, ближайшим аналогом которой является связанный список блоков;
- согласованность: содержимое и структура блокчейна согласована между всеми участниками сети;
- неизменность: установлен порядок следования блоков друг за другом, новые данные добавляются только в конец цепочки.