Информационная безопасность

Контрольные суммы

Солдатов А. Е

2 мая 2025

Российский университет дружбы народов, Москва, Россия

НКАбд-04-23

Докладчик

- Солдатов Алексей Евгеньевич
- Студент
- НКАбд-04-23
- Российский университет дружбы народов
- 1132236009@pfur.ru

Цели и задачи

Целью данной презентации является разобраться в понятии контрольных сумм и узнать для чего они нужны.

Что такое контрольная сумма?

Контрольные суммы представляют собой метод проверки целостности данных, который используется для обнаружения ошибок, возникающих при передаче или хранении информации. Это простая, но мощная техника, обеспечивающая высокую надежность передачи данных.

Общий алгоритм действий

1. Берут данные, для которых нужно составить контрольную сумму.
2. По специальному алгоритму эти данные превращаются в одну строку из символов.
3. Эту строку текста прикладывают к исходному файлу и говорят — ребята, вот контрольная сумма (то есть строка). Если вы не уверены, что всё скачали правильно, проверьте.
4. Те, кто скачал исходный файл, запускают программу проверки контрольных сумм и говорят ей — вот файл, а вот его контрольная сумма, проверь, пожалуйста, всё ли тут правильно.
5. Программа сама составляет контрольную сумму по тому же алгоритму и сравнивает с вашей.
6. Если контрольные суммы совпадают — всё отлично, данные в порядке, можно пользоваться. Если нет — программа выведет сообщение, что суммы отличаются. Это значит, что во время скачивания возникла ошибка или кто-то специально подменил исходные данные, чтобы навредить вам.

Checksum (контрольная сумма)

Описание: Простая контрольная сумма представляет собой сумму всех байтов данных, часто ограниченную определенной длиной (например, 8 или 16 бит). Этот метод широко применяется благодаря своей простоте и скорости.

Метод расчета:

```
Инициализация суммы нулем.
Для каждого байта данных добавляется его значение к общей сумме.
Если сумма превышает максимальное значение, она обрезается до необходимо
Финальное значение суммы является контрольной суммой.
```

LRC (Longitudinal Redundancy Check)

Описание: LRC представляет собой побитную или побайтную контрольную сумму, которая рассчитывается по каждому столбцу блока данных. Этот метод используется для улучшения обнаружения ошибок в многобайтовых данных.

Метод расчета(LRC-8):

```
Инициализация регистра нулями, длина которого равна 8 битам.
Для каждого байта данных выполняется суммирование по байтно.
Отнять получившееся значение от числа FF(Hex).
Прибавить к получившемуся значению 1.
```

CRC-16-CCITT

Описание: CRC-16-CCITT — это 16-битная контрольная сумма, широко применяемая в телекоммуникациях и мобильных системах.

Полином: x^{16} + x^{12} + x^5 + 1

Метод расчета:

```
Инициализация регистра значением 0xFFFF.
Выполнение побитового XOR для каждого байта данных с содержимым регистра
Если старший бит регистра равен 1, выполнение побитового XOR с полиномом
Сдвиг регистра влево на один бит.
Повторение шагов 2-4 для всех байтов данных.
Финальное значение регистра является контрольной суммой.
```

CRC-32

Описание: CRC-32 — это 32-битная контрольная сумма, которая используется в различных цифровых системах для обеспечения надежности данных.

Полином: x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1

Метод расчета:

```
Инициализация регистра значением 0xFFFFFFFF.
Выполнение побитового XOR для каждого байта данных с содержимым регистра
Если старший бит регистра равен 1, выполнение побитового XOR с полиномом
Сдвиг регистра вправо на один бит.
Повторение шагов 2-4 для всех байтов данных.
Инверсия финального значения регистра для получения контрольной суммы.
```

CRC-64-ISO

Описание: CRC-64-ISO — это 64-битная контрольная сумма, использующаяся для высоконадежной проверки целостности данных. CRC (Cyclic Redundancy Check) алгоритмы являются одними из самых надежных и часто применяемых.

Полином: x^{64} + x^4 + x^3 + x + 1

Метод расчета:

```
Инициализация регистра значением 0xFFFFFFFFFFFFFFFF.
Для каждого байта выполняется побитовое XOR с текущим содержимым регистра
Если старший бит регистра равен 1, выполняется побитовое XOR с полиномом
Регистры сдвигаются вправо на один бит.
Повторение шагов 2-4 для всех байтов данных.
Значение регистра инвертируется для получения контрольной суммы.
```

Итоги

1. Зачем нужны контрольные суммы?
2. Какой тип контрольной суммы выбрать?
3. Могут ли использоваться несколько типов контрольных сумм одновременно?
4. В каких областях применяются контрольные суммы?

Выводы

Разобрались в понятии контрольных сумм и узнали для чего они нужны.