What are Checksums?

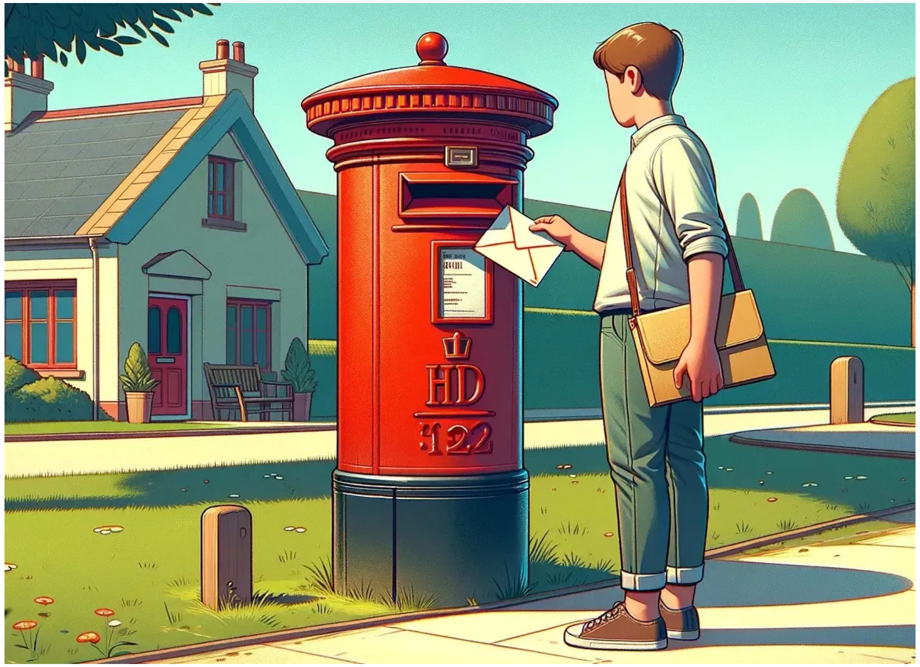
#13 System Design - Checksums

ASHISH PRATAP SINGH

MAY 26, 2024

♡ 91 💬 3 🔁 3 Share

Imagine you're sending an important letter to your friend through the **mail**.



Before sealing the envelope, you take a **photo** of the letter.

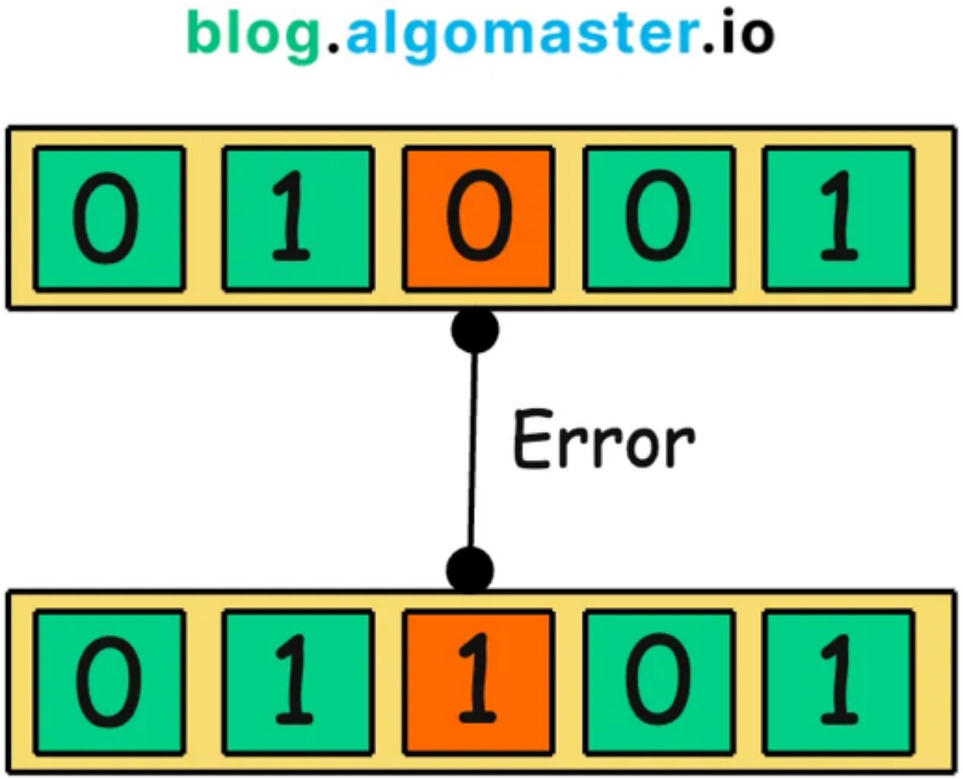
When your friend receives it, they take a photo of the letter and send it back to you.

If the two photos **match**, you know the letter hasn't been tampered with or damaged during transit.

If they don't match, it's a clear sign something went wrong along the way—perhaps the letter was **altered, or part of it was lost or damaged**.

In the digital world, checksums serve a similar purpose as those photos.

Just like taking photos help us answer the question: “**Has the letter been altered or damaged**“, a checksum answers the question: "**Has this data been altered unintentionally or maliciously since it was created, stored, or transmitted?**"



In this article, we'll explore checksums, how they work, different types, and their real world applications.

If you’re enjoying this newsletter and want to get even more value, consider becoming a paid subscriber.

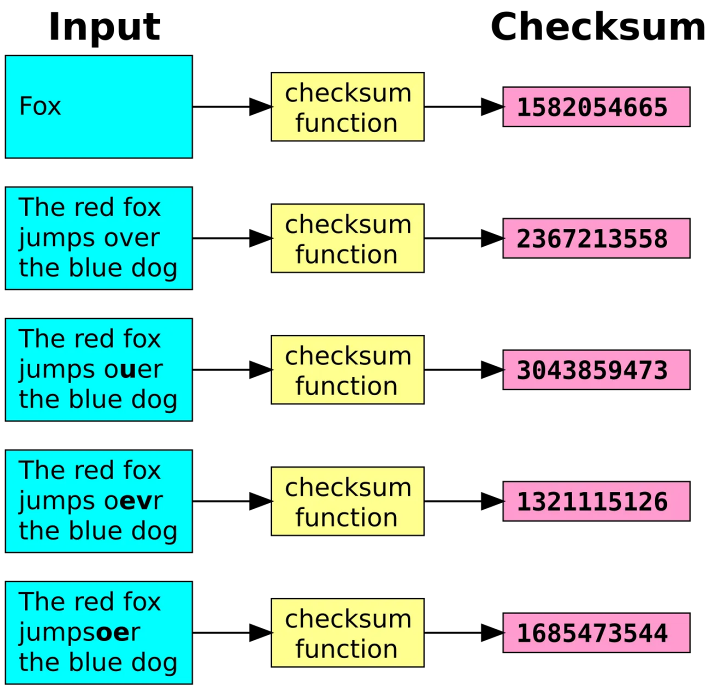
As a paid subscriber, you'll unlock all **premium articles** and gain full access to all premium courses on algomaster.io.

What is a Checksum?

A checksum is a **unique fingerprint** attached to the data before it's transmitted. When the data arrives at the recipient's end, the fingerprint is **recalculated** to ensure it matches the original one.

If the checksum of a piece of data matches the expected value, you can be confident that the data hasn't been modified or damaged.

Checksums are calculated by performing a mathematical operation on the data, such as adding up all the bytes or running it through a **cryptographic hash function**.



Credit: https://en.wikipedia.org/wiki/Checksum

How Does a Checksum Work?

The process of using a checksum for error detection is straightforward:

1. **Calculation:** Before sending or storing data, the original data is processed through a specific algorithm to produce a checksum value.

2. **Transmission/Storage:** The checksum is appended to the data and sent over the network or saved in storage.
3. **Verification:** Upon retrieval or reception, the checksum is recalculated using the same algorithm on the received data. This newly calculated checksum is compared with the original checksum.
4. **Error Detection:** If the two checksum values match, the data is considered intact. If they do not match, it indicates that the data has been altered or corrupted during transmission or storage.

Types of Checksums

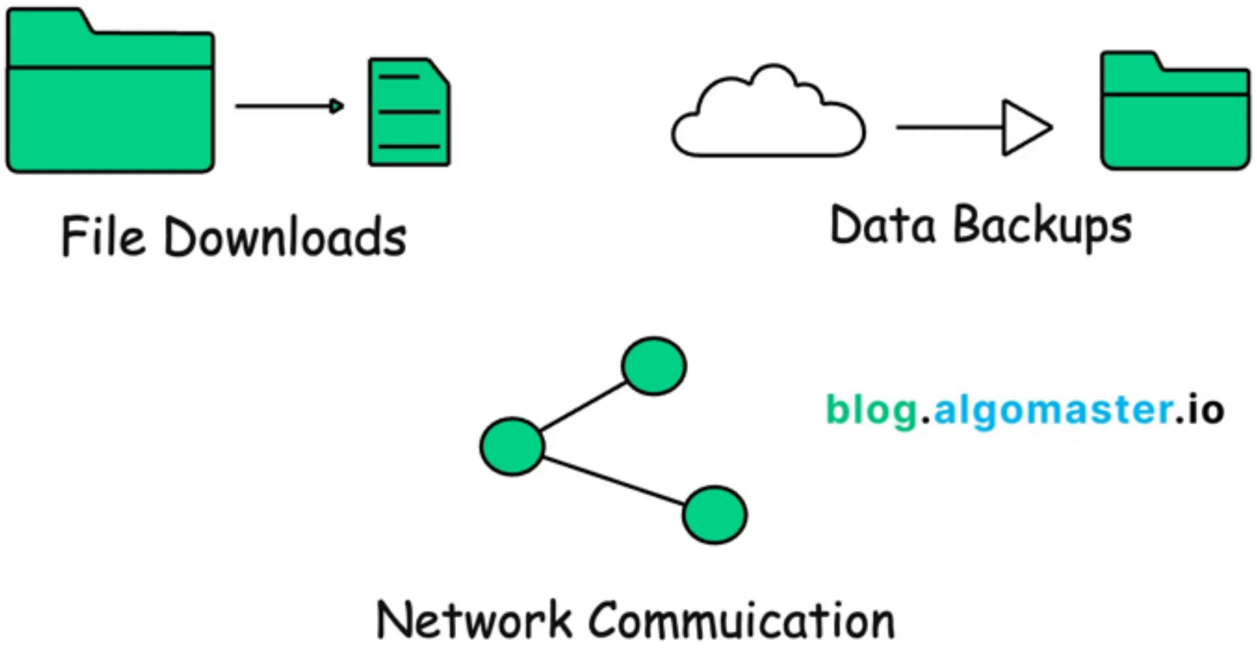
There are several types of checksums, each with its own strengths and weaknesses. Here are a few of the most common:

- **Parity Bit:** A parity bit is a single bit that is added to a group of bits to make the total number of 1s either even (even parity) or odd (odd parity). While it can detect single bit errors, it fails if an even number of bits are flipped.
- **CRC (Cyclic Redundancy Check):** It works by treating the data as a large binary number and dividing it by a predetermined divisor. The remainder of this division becomes the checksum. CRCs are designed to detect common errors caused by

noise in transmission channels.

- **Cryptographic Hash Functions:** These are one-way functions that generate a fixed-size hash value from the data. Popular examples include MD5, SHA-1, and SHA-256.

Real-World Applications of Checksums



- **File downloads**: Checksums verify that downloaded files are complete and uncorrupted.

- **Data backups**: Checksums ensure that backed-up data is accurate and trustworthy.
- **Network communication**: Checksums guarantee that data packets are transmitted correctly, preventing errors and corruption.

To summarize, checksums serve as a vital line of defense, safeguarding against errors and corruption.

From file downloads and data storage to network transmissions and software installations, checksums work tirelessly to detect errors, prevent corruption, and give us confidence in the accuracy of our digital information.

Thank you for reading!

If you found it valuable, hit a like ❤️ and consider subscribing for more such content every week.

If you have any questions or suggestions, leave a comment.

This post is public so feel free to share it.

P.S. If you're enjoying this newsletter and want to get even more value, consider becoming a paid subscriber.

As a paid subscriber, you'll unlock all **premium articles** and gain full access to all premium courses on algomaster.io.

There are group discounts, gift options, and referral bonuses available.

Checkout my Youtube channel for more in-depth content.

Follow me on LinkedIn, X and Medium to stay updated.

Checkout my GitHub repositories for free interview preparation resources.

I hope you have a lovely day!

See you soon,
Ashish

91 Likes • 3 Restacks

← Previous

Next →

Discussion about this post

Comments Restacks

Write a comment...

Ashwani Yadav 27 May 2024

thanks for this article @Ashish. I have a question: what happens if we attach the checksum value in our data and during transmission, the checksum value itself is altered. Then how do we verify if data is corrupted or checksum is corrupted?

LIKE (2) REPLY SHARE

2 replies

2 more comments...

© 2025 Ashish Pratap Singh • Privacy • Terms • Collection notice
Substack is the home for great culture