IETF 95

MICE
Recap

Reference resource, include a hash of that resource

<script src="https://other.origin.example/script.js" integrity="sha384-dOTZf16X8p34q2/kYyEFm0jh8..."> Client checks hash and aborts if it doesn’t match

Hash calculation requires the entire resource

This blocks progressive loads

Or forces nasty handling logic for errors (not always possible)
SOLUTION
MORE HASHING

...and maybe a little hipster crypto

Support both signing and hashing together

  Straight integrity: match hash to expected value

  Signing: sign over hash and check signature

Flexible record sizing allows tuning of chunk sizes

If \( rs \geq \text{Content-Length} \), the result is hash of body \( \| 0x1 \)
PROGRESSIVE INTEGRITY
GENERATION IS RELATIVELY EXPENSIVE
PROGRESSIVE INTEGRITY
FIRST CHUNK IS VALIDATED

VALIDATE FORWARDS
PROGRESSIVE INTEGRITY
RELEASE EACH CHUNK AS IT IS VALIDATED
PROGRESSIVE INTEGRITY
SIGNATURE IS VALID ALL THE WAY

VALIDATE FORWARDS
CONTENT ENCODING
YEAH, I SEEM TO LIKE THOSE

Allows for interstitial interleaving of integrity

Solves questions about when the integrity applies

  Interaction with gzip, brötli, and other C-E resolved

Can compress either before or after authentication
IS A SIMPLER DESIGN BETTER?
OR IS TOO MUCH MERKLE BARELY ENOUGH?

HEADER
FIELD

H
H
H
H

9