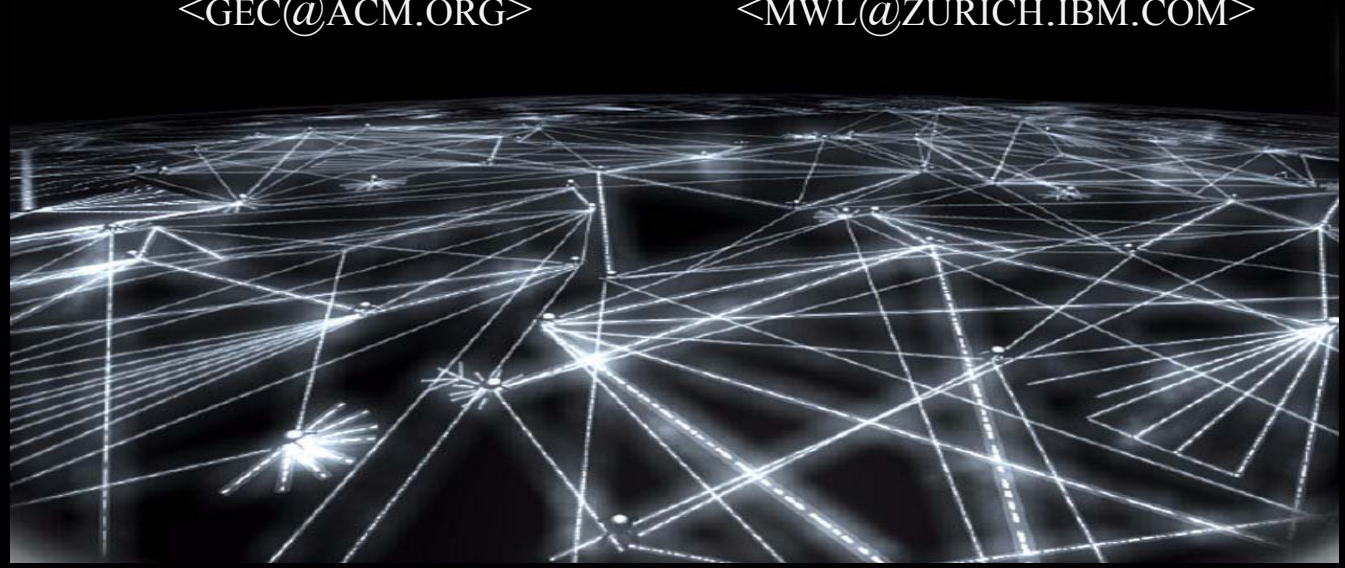
Establishing Trust in Distributed Storage Providers

Germano Caronni + Marcel Waldvogel
<GEC@ACM.ORG> <MWL@ZURICH.IBM.COM>



Sun microsystems

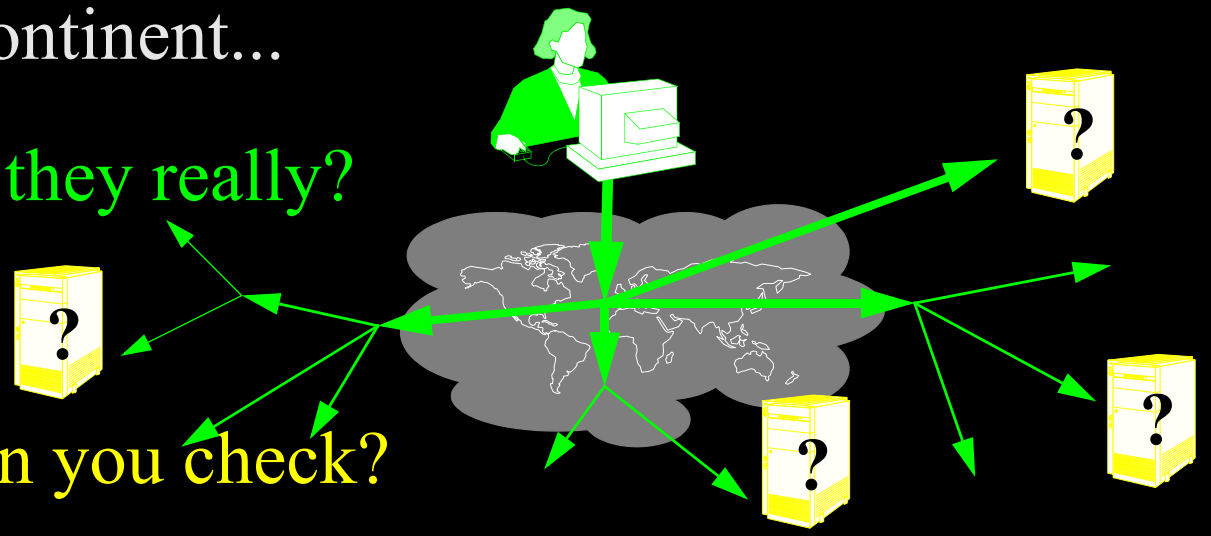
Imagine...

You share data with your peers, and in return you allow them to use your disk...

You buy replicated storage from BigDisk.com and AllBytes.com, and they promise to store it on every continent...

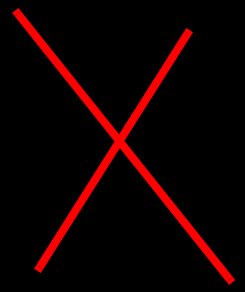
But, do they really?

How can you check?



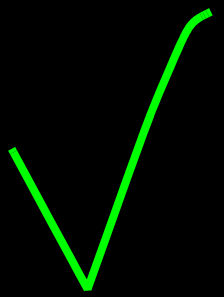
Content

How To Do it Wrongly
- Fetching Blocks
- Requesting Checksums



Colluding Replica Holders

How To Do It Right
- Prevent Forwarding
- DoS Considerations



Summary

How To Do It Wrongly

Concerns:

- Limit trust to bare minimum
- Use technical means to insure trust
- Accept CPU & bandwidth limitations

Two extreme ways of doing verification:

Fetch Content for Verification

Request File (or parts) for verification

Fails because of:
- Dishonest holder can forward request
- High bandwidth consumption
- Can be used to run efficient DoS attacks

But: Proof is in the pudding (you want the data)

Request Hash / Checksum

Failure Potentials:

- Replica holder could precompute hash
- Request could be forwarded / delegated
- Computational cost can be comparatively high
- Again, DoS potential, against replica holder

Summary of Issues

Precomputation

Forwarding of requests to honest holders

Forward to colluding holders

Fetch Data on Demand

Usability for DoS

Colluding Replica Holders

Tenuous: Detection by technical means
- Detect delay of forwarding requests
- Store individualized copies

However, scenario is rather unlikely!
- Large market / replicas at many **different** places
- Small profit, substantial loss in case of revelation
- Trading disk storage against bandwidth?
  Does not pay off!

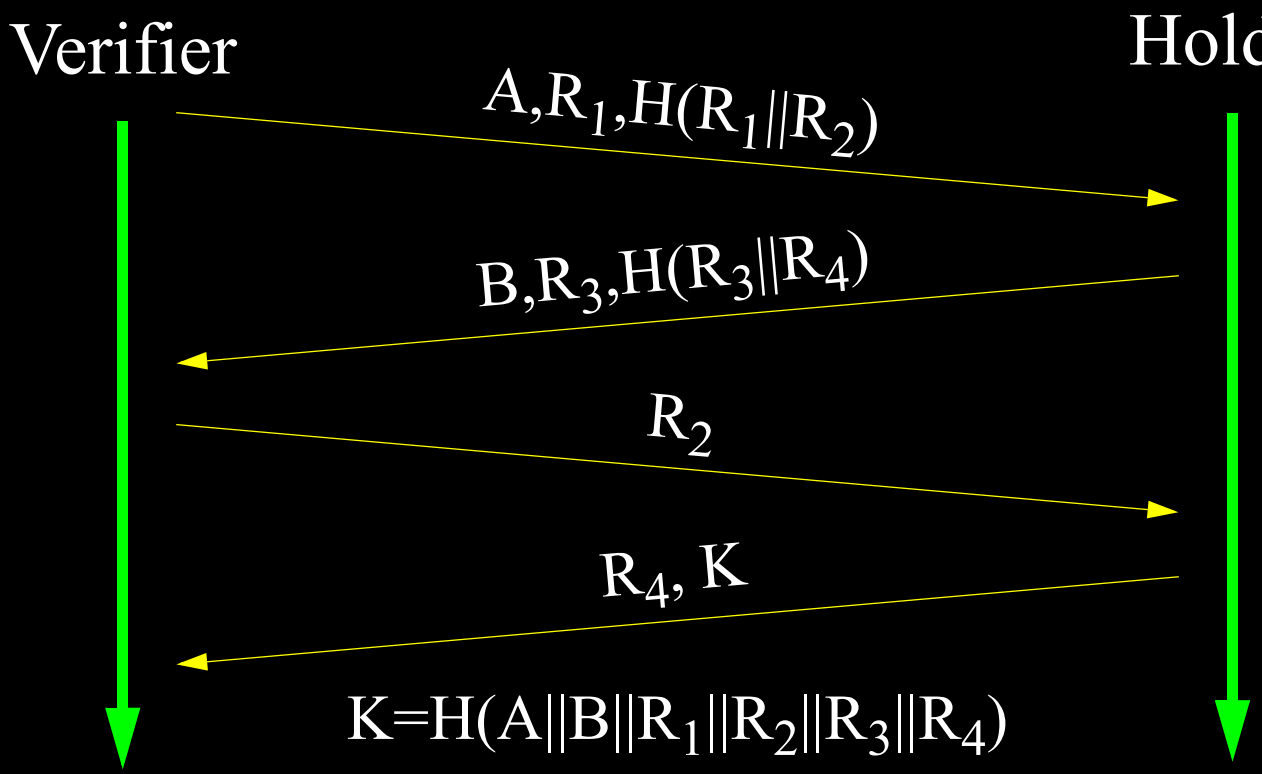
How To Do It Right

No need to authenticate replica holders or verifier, no need for confidentiality. What we need is balancing the proof costs, fairness, and uniqueness.

Starting Point: Bilateral Key Agreement
Use keyed MAC to calculate checksum
- Forwarding to other holders is prevented
- Precomputation / reuse is not possible

Man in the Middle -> Loss of Trust

Bi-Lateral Key Agreement



Simple and effective. Sending K for convenience.

Content Fetching on Demand

Dishonest replica holder could just fetch the file from somewhere else when verification is requested

Expensive in terms of time and bandwidth, time can be measured.

Make the check much less expensive for a honest replica holder. Trade MAC for cheap checksum and data-dependent keying.

Verification Procedure

```
 1   Agree on common key, seed RC4
 2   Set chunk size, maximum step,
     number of steps
 3   position := RC4[8] modulo file size
 4   while step nr. < number of steps do
 5     value := chunk at current file position
 6     insert value into checksum
 7     step size := value + RC4[8]
                    modulo maximum step
 8     position := position + step size
                    modulo file size
 9   end while
10   return checksum
```

DoS Considerations

Verification (and Fetch) could be used for DoS.

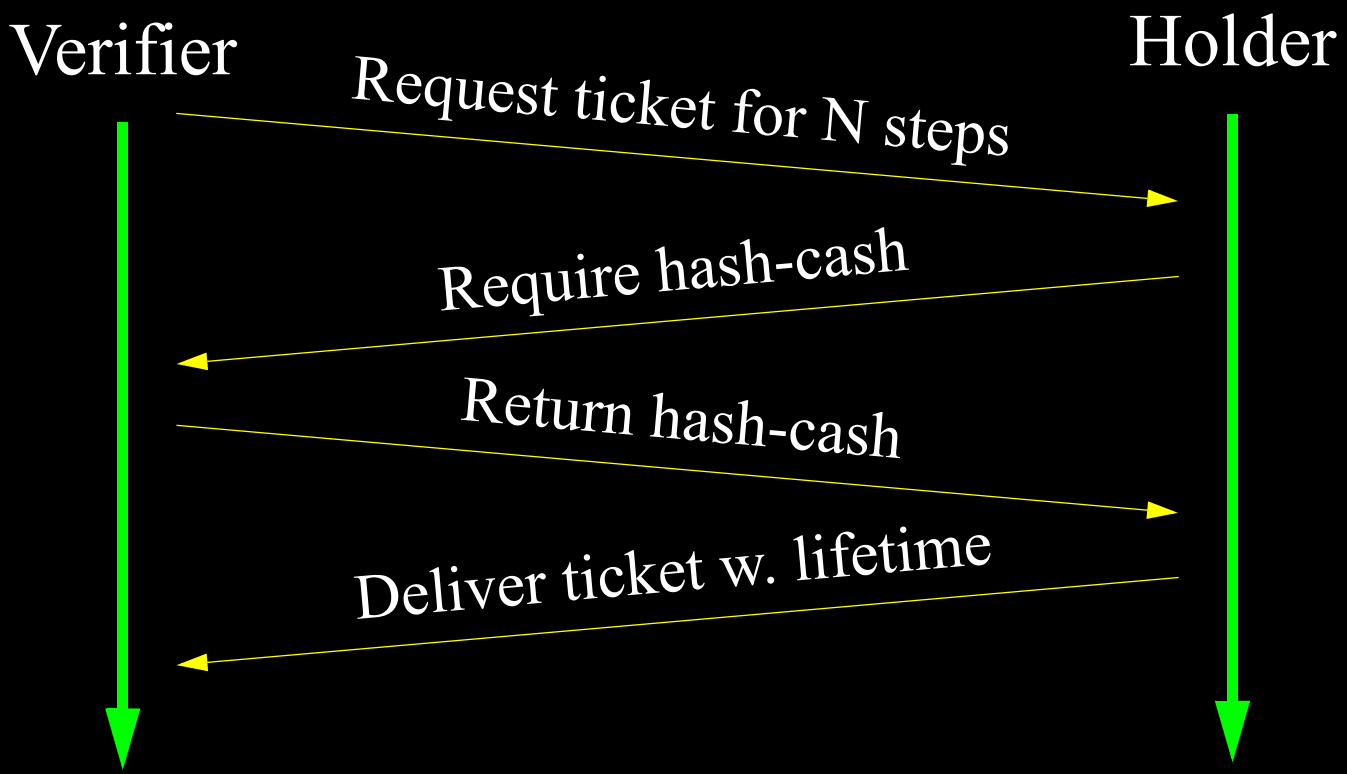
->

Build up trust by chaining requests, and have participants collect an (identity-less) reputation.

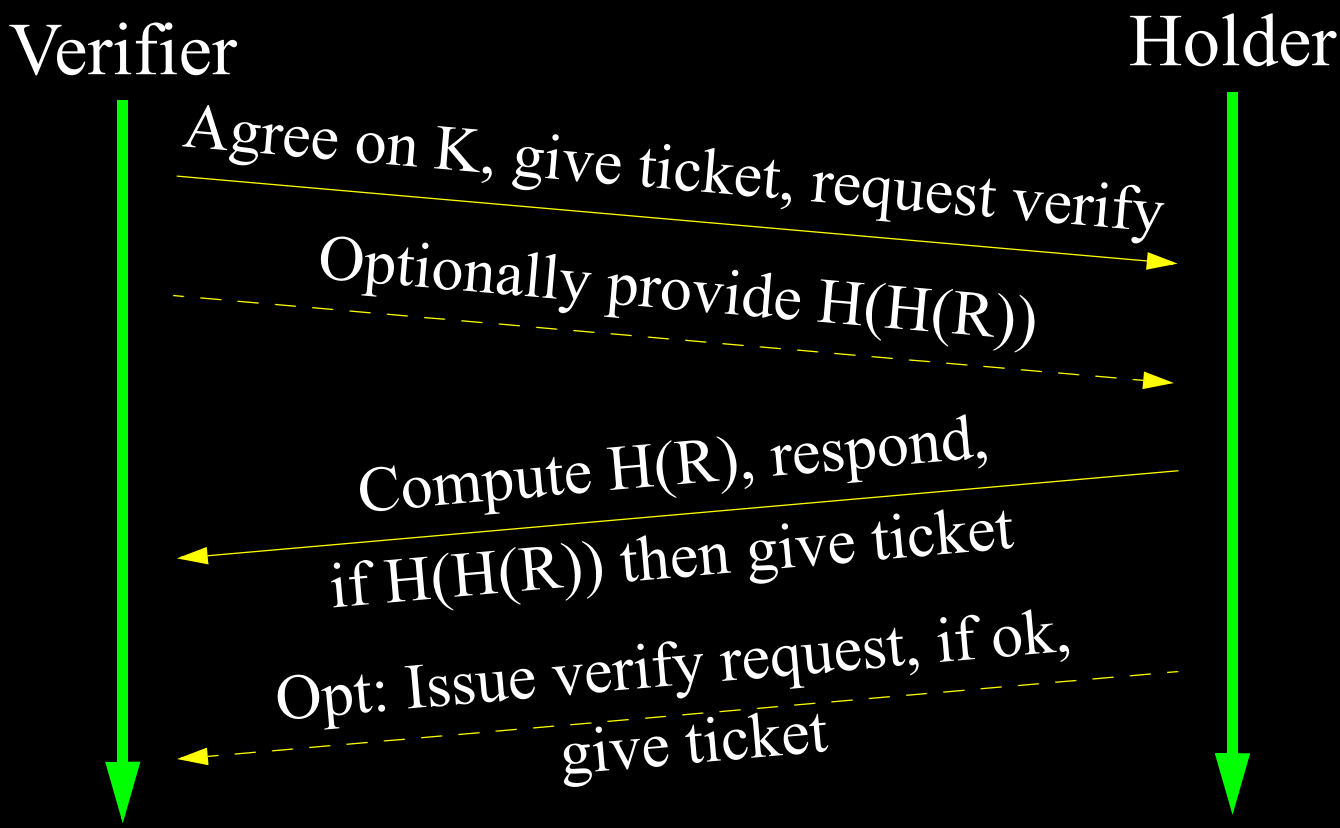
Iterative protocol, increasing costs.

Credits are given and spent – both sides verify, or one side pays (hash) cash.

Iterative Verification Protocol



IVP continued ...



Summary

Non-Trivial Fair Verification
Comparatively cheap in bandwidth / CPU

No delegation, no precomputation

Iterative (symmetric) to build trust

Anonymous: Only content matters.

Questions?