BGPsec Scalability

or

Protocol Engineering meets Software Engineering

or

What would happen to a route server if BGPsec were deployed end to end today

BGPsec in theory

<Prefix, Path and signature elements, Target> -> hash -> sign -> tx

rx -> hash -> verify -> add new elements -> hash -> sign -> tx

Scalability dimensions

- Number of prefixes.
- Number of prefixes sharing the same path.
- Fanout ratio.
- Caching aspects.

Platform and algorithm specifics

- Memory bandwidth and latency.
- Vectorization.
- Batching.
- Performance model of contemporary compute platforms.

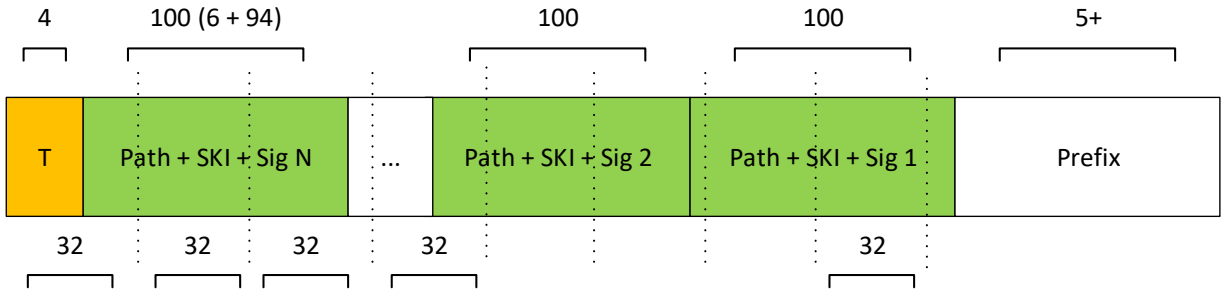
BGPsec in practice

SHA2 for hashing

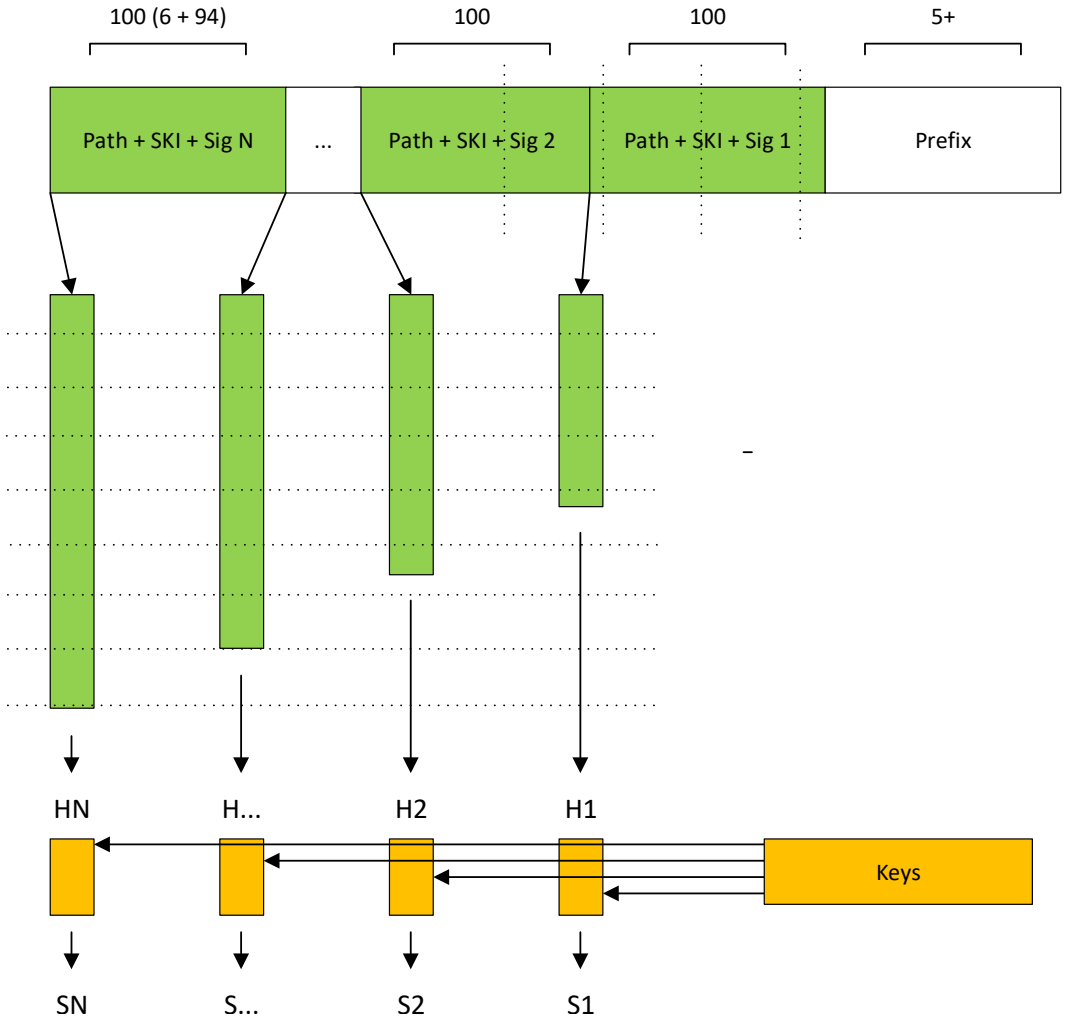
- Computationally inexpensive – but touches memory.
- Operates on 32 byte blocks with 4 byte granularity.
- Vectorizes well.

P-256 for signing/verification

- Computationally expensive – but does not touch memory.
- Verification is significantly more expensive than signing.
- Vectorizes well.



Vectorized SHA2 and P-256



Llinear code block operating on different data in parallel

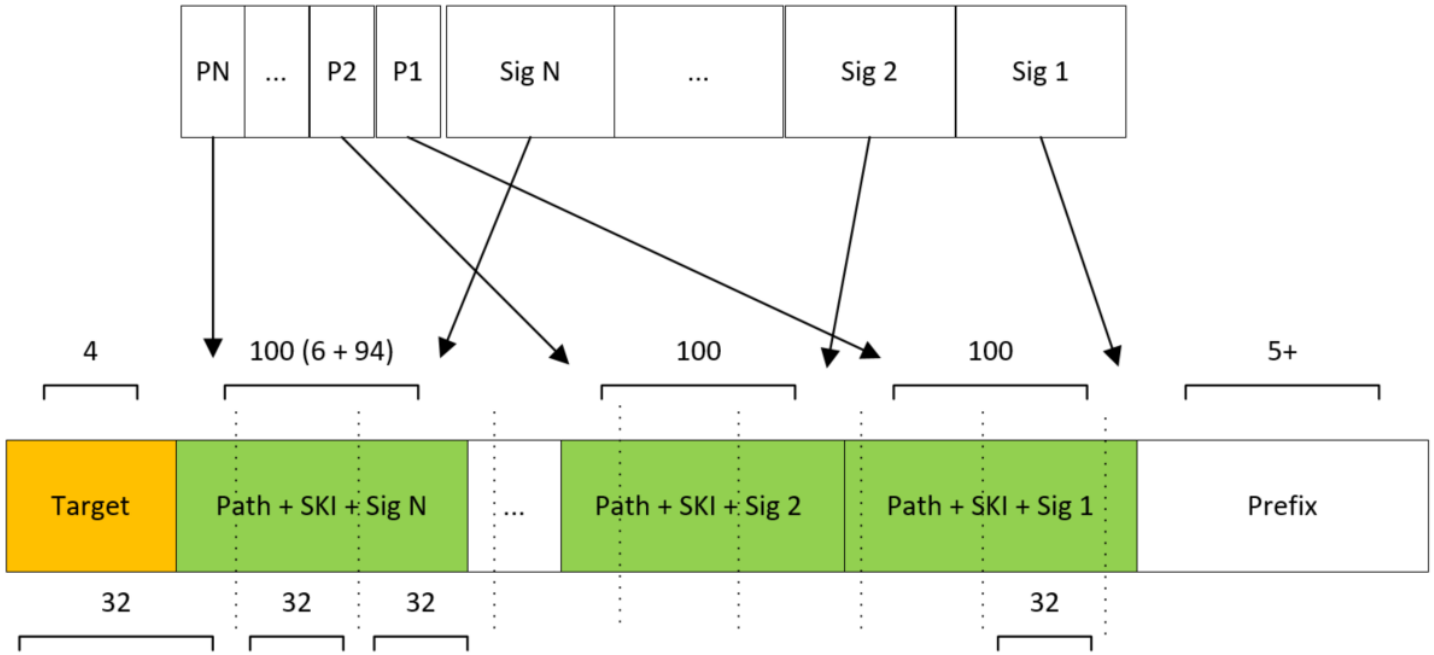
Vector lanes of fixed width

Hash multiple blocks in parallel
Sign multiple hashes in parallel

+20% latency results in +1500% throughput

If data structures allow!

Wire image vs implementation efficiency



Wire format is incompatible with computation format!

Memory access is expensive

SHA2 latency is linearly proportional to block length

Gather operations place significant restrictions on data format

ECDSA signing is computationally expensive but constant, no memory access

ECDSA verification is even more computationally expensive but constant, no memory access

Experiments

- Take realistic state distribution ratios.
- Instrumented implementation for performance tracking.
- Most important – contemporary compute platforms are complex and do not forgive lousy approaches to software engineering. Protocol engineering needs to take software and hardware specifics into account seriously.

```
bgpsec: 12 segments, hash tsc 244934, sign tsc 53678, ratio 4.563024
verify_signature_iov: valid 1, tsc 231645
```

Is BGPsec broken?

- Security wise - no.
- Otherwise – mostly yes.

What can be fixed then?

- BGPsec has some extensibility mechanisms inbuilt. That is good.
- Protocol is versioned.
- Algorithm identifiers could have different meaning in different versions.
- Wire image needs to be rearranged.

Summary

SHA2 hash over SECURE_PATH components and signatures
- Memory access – expensive!
- Vectorizable - if data layout allows.

P-256 signing
- No memory access.
- Vectorizable

P-256 verification
- No memory access, inherently more expensive than signing.
- Vectorizable, batchable (ECDSA*)

- Wire image vs hashable block layout
- Caching is highly desirable

Discussion

Do we care?