

DNSSEC live signing at scale

Filippo Valsorda

Why CloudFlare needs live signing

- Lots (lots!) of small, light traffic zones
- Heavily distributed network (34 data centers)
- **Dynamically generated records**
- Zone walking protection

Why CloudFlare needs live signing

- Lots (lots!) of small, light traffic zones

It would be a waste to sign all the records for all the zones over and over again when each datacenter only gets asked a fraction.

Why CloudFlare needs live signing

- Lots (lots!) of small, light traffic zones
- Heavily distributed network (34 data centers)

We would have to continuously distribute huge amounts of RRSIG data to the edges, or sign everything everywhere.

Why CloudFlare needs live signing

- Lots (lots!) of small, light traffic zones
- Heavily distributed network (34 data centers)
- Dynamically generated records

CloudFlare generates dynamic records all the time, which can't be predicted and signed offline!

(think attacks rerouting, geolocation based answers)

Why CloudFlare needs live signing

- Lots (lots!) of small, light traffic zones
- Heavily distributed network (34 data centers)
- Dynamically generated records
- Zone walking protection

NSEC3 does not provide any actual confidentiality.

NSEC5 is not here yet.

Issues with live signing

- Speed!
- Negative answers
- Key management

Constraints

Keep size small, and don't require full zonefiles

Our solutions!

CloudFlare's DNS(SEC) overview

- RRDNS is our in-house DNS server written in Go
- Resilient against attacks and abuse
- No zonefiles, records are pulled from a global distributed database
- Full featured (dynamic answers, CNAME flattening, ...)
- DNSSEC is just a "filter" applied to the answer

Solving speed (and size): ECDSA P256

- ECDSA P256 signatures are > 3x faster than RSA1024

Measured on OpenSSL 1.0.2 on our servers

- We (Vlad Krasnov) ported OpenSSL ASM to Go

21X speedup for the sign: <https://go-review.googlesource.com/#/c/8968/>

- Bonus: small signatures, small keys, modern crypto!
- Supported by most validators, working on registrars

Solving speed (and size): ECDSA P256

Standard Go crypto:

BenchmarkSingleSignECDSA	832295 ns/op
BenchmarkSingleSignRSA	6003261 ns/op

Go with Vlad's changes:

BenchmarkSingleSignECDSA	60806 ns/op
BenchmarkSingleSignRSA	3124274 ns/op

<https://blog.cloudflare.com/go-crypto-bridging-the-performance-gap/>

Solving speed (and size): ECDSA P256

```
ietf.org. 1800 IN DNSKEY 256 3 5 AwEAAAdDECajHaTjfSoNTY58WcBah1BxPKVIHBz4IfLjfqMvium4lgKtK ZLe97DgJ5/NQrNEGGQmr6fKv
Uj67cfrZUojZ2cGRizVhgk0qZ9scaTVX NuXLM5Tw7VW0VIceYAUuH2mPTiEV6Mh JVUsw6dvmNs J4XwCgNgreAmX hoMEiWEjBB+wjYZQ5GtZHBFKVXACSWTiCtddHcue0eSVPi5
WH94Vlubh HfiytNPZLr0bhUCHT6k0tNE6phLoHnXWU+6vpsYpZ6GhMw/R9BFxw5Pd P5IwBgoWk2/XFVRSKG9Lr 61b2z1R126xeUww46RVy3hanV3vN07LM5H niqaYc1Bbhk=
ietf.org. 1800 IN DNSKEY 257 3 5 AwEAAaav... 8LGP0wQBFVLOEM9 BRfqxz9p/sZ+8AByyqFHLdZc Ho0GF7CgB50KYMvG0gysuYQ1
oPlwbq7Ws5WywbutbXyG24lMwy4jijlJ UsaFrS5EvUu4ydmuRc/TGnEXnN1XQkU+wa1T4cLTrmcWjoY80qud6lD a Jdj1cKr2nX1NrmMRowIu3DIVtGbQJmzpukpDVZaYMMAm8M5
vz4U2vRCV ETLgDoQ7rhsiD127J8gVExj08B0113jCajbFRcmUtFTjH4z7jXP2ZzD cXsgpe4LYFuenFQAcRBR1 E6oaykHR7r1Pqqmw58nIELJUFOmcb/BdRLg byTeurFlnxs=
ietf.org. 1800 IN RRSIG DNSKEY 5 2 1 20160422162528 42 ietf.org. 43650 45586 ietf.org. dp001u/mE0ZmcergtT4RA5DdV8E
i3nTYvsuTFKqEou4Smku5Up01giVp sOpdDRwvei5g2HC8VK/ nKHDh... 0/7yDr2TK529YHee0MTVeHqk6YeyyiFvCL1XMLt3jj4/G3pjo
z7mS8M NLgysKQMEZqJHfZhARZeSNIuK/QpRJhBX9UQYrv6IJ/215wqL6C6aeb FYe+bhn3G2s9apnuQFiq0xo3ybyQJm06UEPjuEnn8uLXnXT1RdthZbnY g5yZReSwb4jVYQKC
yX4Pnm09TtrpduZQqz120v+8nMITf4HJnSj7EvPN AxmCXg==
```

RSA:

1181 BYTES

```
filippo.io. 3600 IN DNSKEY 257 3 13 DGpDkudNu/XQT1Km
QkXFtKcfZPxHGV07qSTIcDXS33/WtT8UUG7LyxAg KznsRSFEhiQVR53E69/E57IFm8b6Zw==
filippo.io. 3600 IN DNSKEY 256 3 13 koPbw9wmYZ7ggcjn
Q6ayHyhHaDNMYELKTqT+qRGZpWScrr/1Bcrm10Z 1PuQHB3Azhii+sb0PYFkH1ruxLhe5g==
filippo.io. 3600 IN DNSKEY 13 2 3600 20150523
162528 20150422162528 42 filippo.io. KgjopS+z5rsK+grfGMuA2a1/vQ9S5tBX0Jq
ZbeTOYB0hfHG7S16hqR1 xfoibSJA1BiX5r9Ujo5YVU/NE1H0TQ==
```

ECDSA:

305 BYTES

Solving negatives: “Black Lies”

- To answer a NXDOMAIN normally we need:
 - Database lookups for previous and next name
 - 2 or 3 signatures (NSEC/NSEC3) - slow and big!
 - Previous and next name disclosure

Solving negatives: "Black Lies"

```
mipappstg.comcast.com. 3600 IN NSEC mmgr.comcast.com. CNAME RRSIG NSEC
mipappstg.comcast.com. 3600 IN RRSIG NSEC 5 3 3600 20150508165102 2015050
1134602 39162 comcast.com. 0jKZ/h3bkK/AXs0kkg2Cbd13+aabCnCnp0sW9QHSrX8xcD04+SdxYx+E
F6PtFUUYh0KA8u9dcir7nkqI2Et326oAPuV8gbY6cLB8sFTceK6Fz0V0 /cIXrZyggy/VPf82FuBcoZsQnAb
erV0sI6RRbwjatPW65Wlo1bqKBrr9 Z7Q=
comcast.com. 3600 IN NSEC 208.20.10.201.comcast.com. A NS SOA
MX TXT RRSIG NSEC DNSKEY
comcast.com. 3600 IN RRSIG NSEC 5 2 3600 20150508165102 2015050
1134602 39162 comcast.com. TdPdnLkg5Zf12/rgskPWG194L+WigPn4AUD59p0qaX/T1fDmXU0g7WXH
38RORuUGmBmu7HSqzCekxJf1S//4ohw07NP3gSTz5dtW6co0Hw1E5n0 XaW+5nQC7pSBBjxa99DrUtPtpk6
2WACXuug/6A61FcIov0ppknsU1/12 fsQ=

;; Query time: 344 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu May 07 14:05:56 BST 2015
;; MSG SIZE rcvd: 736
```

Solving negatives: “Black Lies”

- RFC 4470 introduces “white lies” for online signing:
 - Generate a NSEC on the name’s immediate predecessor, covering up to the successor (RFC4471)
 - Same with the wildcard
 - Solves: zone walking, database lookups
 - Still, 2 signatures to say one thing :(

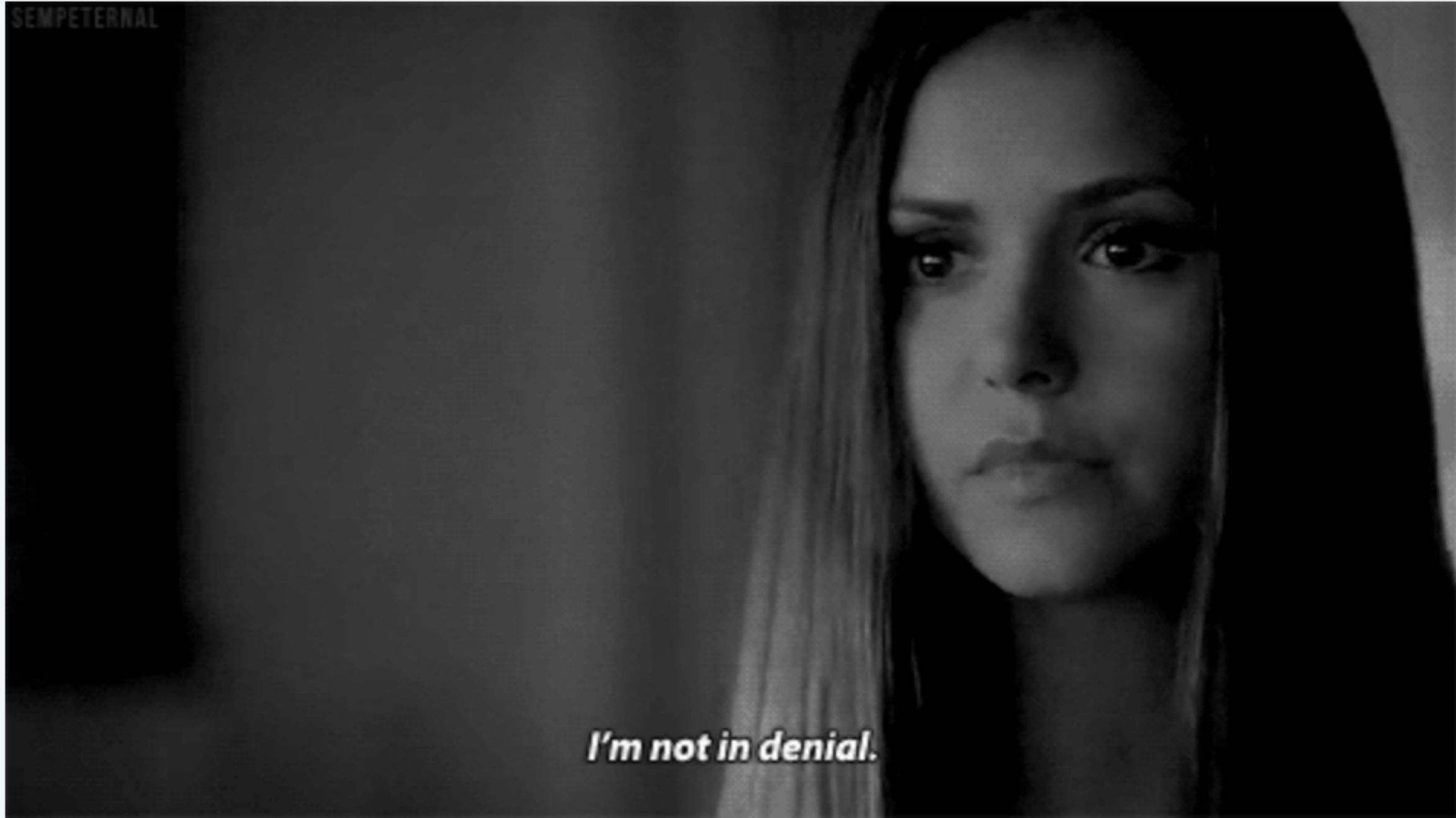
Solving negatives: "Black Lies"

- Our solution: true lies. Just sign a NOERROR.

JAN
31

dnsreactions

SEMPETERNAL



I'm not in denial.

When +dnssec turns NXDOMAIN into NOERROR

Solving negatives: "Black Lies"

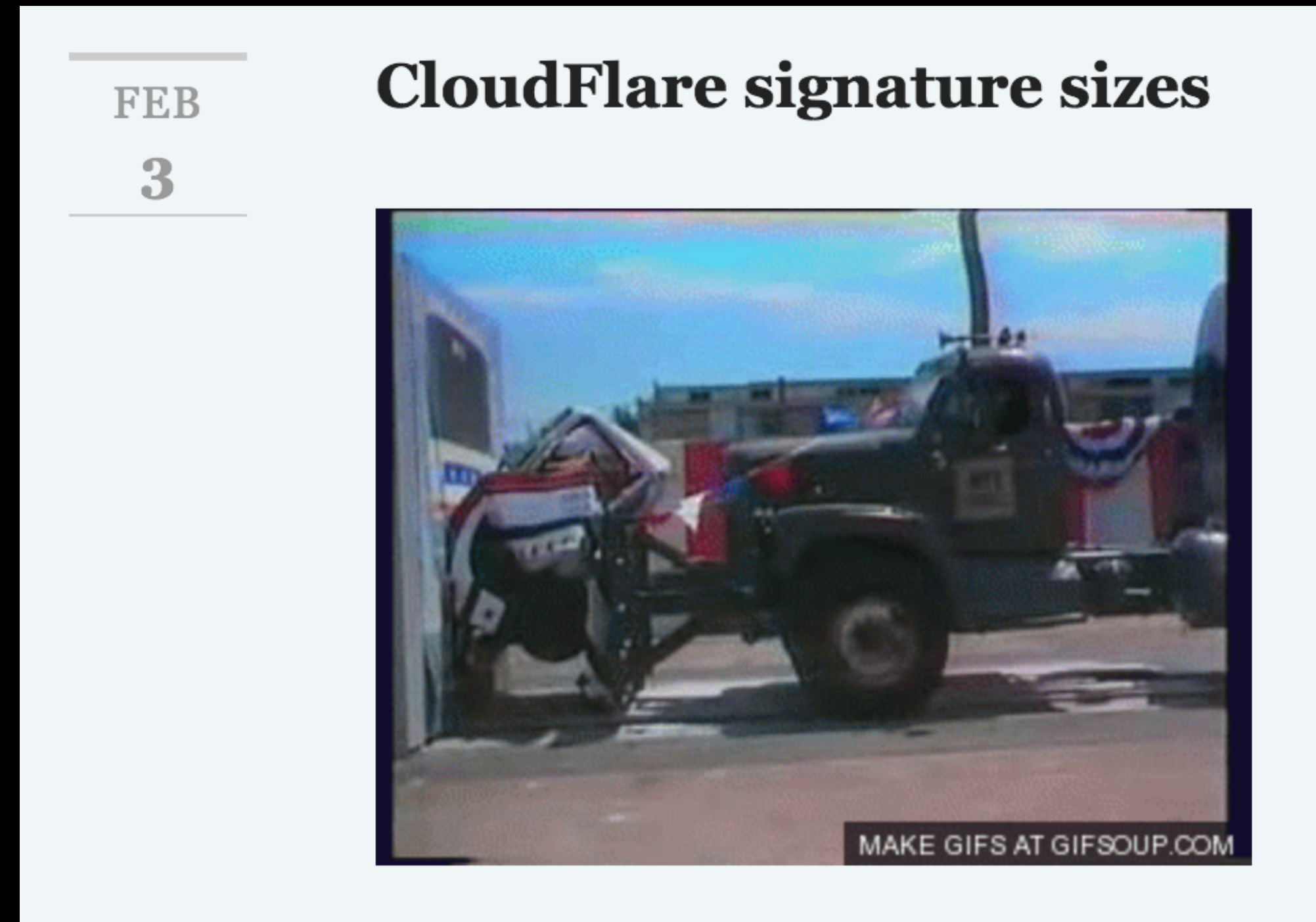
- Our solution: true lies. Just sign a NOERROR.
- Place a NSEC on the name, cover until the successor, set only the NSEC and RRSIG bits

```
missing.filippo.io.      3587      IN        NSEC      \003.missing.filippo.io. RRSIG NSEC
missing.filippo.io.      3587      IN        RRSIG     NSEC 13 3 3600 20150507190048 201505
05170048 35273 filippo.io. Fb/xInfArVCMJWBDBqsbBPxiKsC1ueUyBFGi5lAHbjRBGAGm8sKDJx/l
YA01bKYzJep3dRgQw5hS89JukD+m8w==
```


Solving negatives: "Black Lies"

```
missing.filippo.io. 3587 IN NSEC \003.missing.filippo.io. RRSIG NSEC  
missing.filippo.io. 3587 IN RRSIG NSEC 13 3 3600 20150507190048 201505  
05170048 35273 filippo.io. Fb/xInfArVCMJWBDBqsbBPxiKsC1ueUyBFGi51AHbjRBGAGm8sKDJx/1  
YA01bKYzJep3dRgQw5hS89JukD+m8w==
```

```
:: Query time: 0 msec  
:: SERVER: 127.0.0.1#53(127.0.0.1)  
:: WHEN: Wed May 06 19:01:01 BST 2015  
:: MSG SIZE rcvd: 363
```



Solving negatives: "Black Lies"

```
missing.filippo.io.      3587      IN        NSEC      \003.missing.filippo.io. RRSIG NSEC
missing.filippo.io.      3587      IN        RRSIG     NSEC 13 3 3600 20150507190048 201505
05170048 35273 filippo.io. Fb/xInfArVCMJWBDBqsbBPxiKsC1ueUyBFGi5IAHbjRBGAGm8sKDJx/1
YA01bKYzJep3dRgQw5hS89JukD+m8w==
```

- **1 signature op**, no db lookup or zone walking
- The entire answer fits 512 bytes (actually, < 400!)
- End-user behavior is unchanged

Solving negatives: “Black Lies”

```
missing.filippo.io.      3587      IN        NSEC      \003.missing.filippo.io. RRSIG NSEC
missing.filippo.io.      3587      IN        RRSIG     NSEC 13 3 3600 20150507190048 201505
05170048 35273 filippo.io. Fb/xInfArVCMJWBDBqsbBPxiKsC1ueUyBFGi5IAHbjRBGAGm8sKDJx/1
YA01bKYzJep3dRgQw5hS89JukD+m8w==
```

- We suggest to signal the difference between a NXDOMAIN and a empty non-terminal with a special RRType in the NSEC bitmap

<https://datatracker.ietf.org/doc/draft-ogud-fake-nxdomain-type/>

Solving negatives: the “NSEC shotgun”

- But. To answer a missing type on an existing name, we still need to query the database for the NSEC bitmap
- That’s not even always possible! (Dynamic answers)

```
filippo.io. 3600 IN NSEC \003.filippo.io.  
A NS SOA MX TXT AAAA RRSIG NSEC DNSKEY
```

Solving negatives: the “NSEC shotgun”

- Step back: what is a NSEC? A denial of existence.
- “The types not in the bitmap don’t exist”
- So, let’s make a “minimally covering” one.
By setting all possible bits in the bitmap!

```
filippo.io. 3600 IN NSEC \003.filippo.io.  
A NS SOA WKS HINFO MX TXT AAAA LOC SRV CERT SSHFP  
IPSECKEY RRSIG NSEC DNSKEY TLSA HIP OPENPGPKEY SPF
```

Solving negatives: the “NSEC shotgun”

- Asked for TXT and there’s no TXT? Set all the other bits that might exist.
- The NSEC is a valid denial for TXT, and is useless for an attacker that wants to replay it for other queries.

```
filippo.io. 3600 IN NSEC \003.filippo.io.  
A NS SOA WKS HINFO MX TXT AAAA LOC SRV CERT SSHFP  
IPSECKEY RRSIG NSEC DNSKEY TLSA HIP OPENPGPKEY SPF
```


Solving negatives: the “NSEC shotgun”

- Asked for TXT and there’s no TXT? Set all the other bits that might exist.
- No useless database lookups! Actually, no need to see the database from the signer at all.

```
filippo.io. 3600 IN NSEC \003.filippo.io.  
A NS SOA WKS HINFO MX TXT AAAA LOC SRV CERT SSHFP  
IPSECKEY RRSIG NSEC DNSKEY TLSA HIP OPENPGPKEY SPF
```

Solving keys: centralized DNSKEY sets

- It's live-signing, you need the ZSK at the edge (for now)
- Protect the KSK: keep it in a safe central auditable machine, distribute the signed DNSKEY sets to edges
- Short regular RRSIG validity, longer for DNSKEY
- Prepared to roll the ZSK fast at any time

Solving keys: global ZSK and KSK

- No reason to have millions of ZSKs and KSKs: all would be used/stored/rolled together
- Use a single KSK and a single ZSK with multiple names

```
filippo.io.          3600  IN  DNSKEY  256 3 13  
koPbw9wmYZ7ggcjnQ6ayHyhHaDNMYELKTqT+qRGrZpWSccr/lBcrm10Z  
1PuQHB3Azhii+sb0PYFkH1ruxLhe5g==
```

```
cloudflare-dnssec-auth.com. 3600  IN  DNSKEY  256 3 13  
koPbw9wmYZ7ggcjnQ6ayHyhHaDNMYELKTqT+qRGrZpWSccr/lBcrm10Z  
1PuQHB3Azhii+sb0PYFkH1ruxLhe5g==
```


Questions?

Filippo Valsorda
filippo@cloudflare.com



What an engineer at CloudFlare must feel like