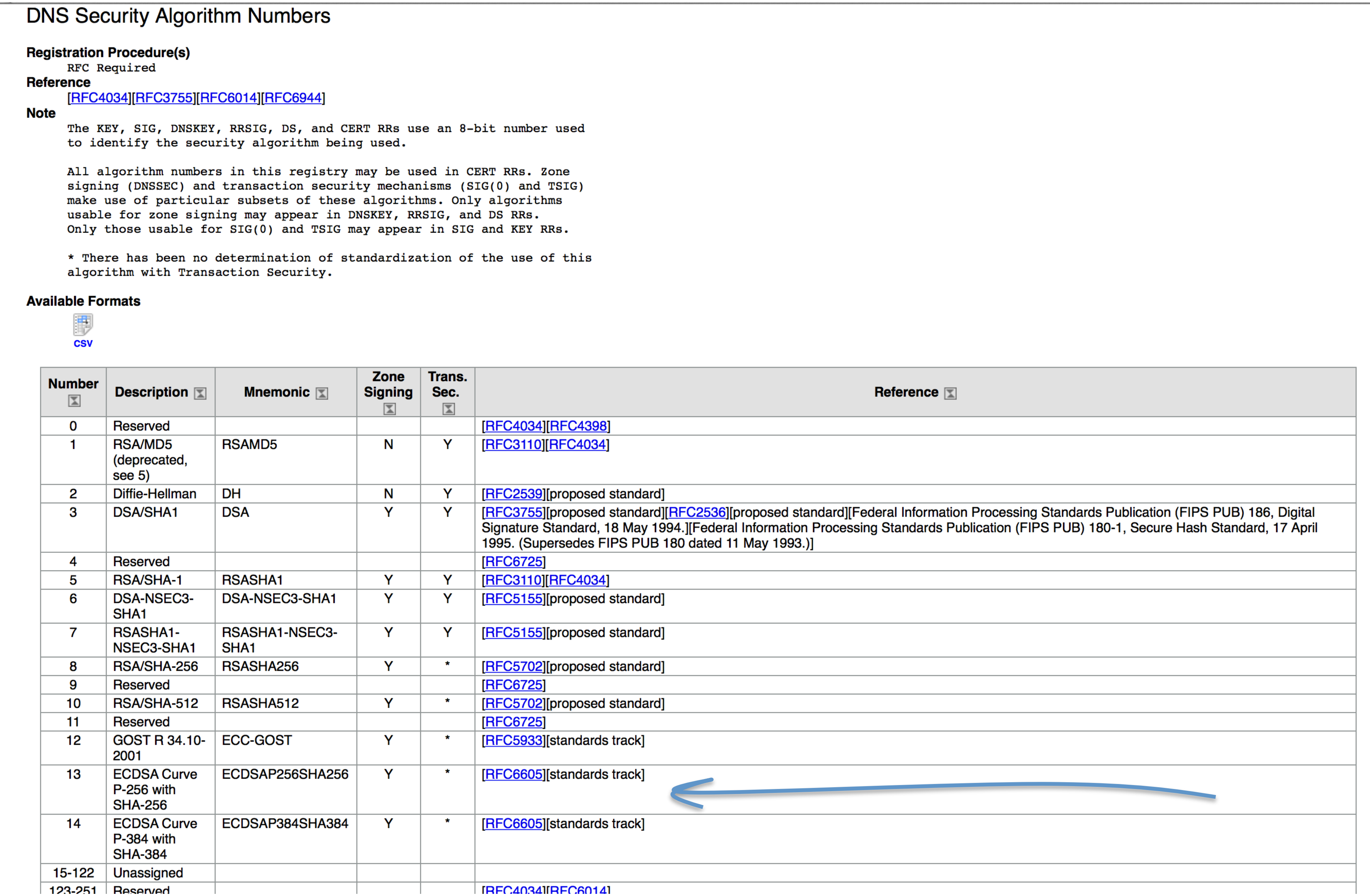
ECC support in DNSSEC-validating Resolvers

Geoff Huston, George Michaelson

APNIC Labs

October 2014



DNS Security Algorithm Numbers

Registration Procedure(s)

RFC Required

Reference

[RFC4034][RFC3755][RFC6014][RFC6944]

Note

```
The KEY, SIG, DNSKEY, RRSIG, DS, and CERT RRs use an 8-bit number used
to identify the security algorithm being used.

All algorithm numbers in this registry may be used in CERT RRs. Zone
signing (DNSSEC) and transaction security mechanisms (SIG(0) and TSIG)
make use of particular subsets of these algorithms. Only algorithms
usable for zone signing may appear in DNSKEY, RRSIG, and DS RRs.
Only those usable for SIG(0) and TSIG may appear in SIG and KEY RRs.

* There has been no determination of standardization of the use of this
algorithm with Transaction Security.
```

Available Formats

CSV

Number	Description	Mnemonic	Zone Signing	Trans. Sec.	Reference
0	Reserved				[RFC4034][RFC4398]
1	RSA/MD5 (deprecated, see 5)	RSAMD5	N	Y	[RFC3110][RFC4034]
2	Diffie-Hellman	DH	N	Y	[RFC2539][proposed standard]
3	DSA/SHA1	DSA	Y	Y	[RFC3755][proposed standard][RFC2536][proposed standard][Federal Information Processing Standards Publication (FIPS PUB) 186, Digital Signature Standard, 18 May 1994.][Federal Information Processing Standards Publication (FIPS PUB) 180-1, Secure Hash Standard, 17 April 1995. (Supersedes FIPS PUB 180 dated 11 May 1993.)]
4	Reserved				[RFC6725]
5	RSA/SHA-1	RSASHA1	Y	Y	[RFC3110][RFC4034]
6	DSA-NSEC3-SHA1	DSA-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
7	RSASHA1-NSEC3-SHA1	RSASHA1-NSEC3-SHA1	Y	Y	[RFC5155][proposed standard]
8	RSA/SHA-256	RSASHA256	Y	*	[RFC5702][proposed standard]
9	Reserved				[RFC6725]
10	RSA/SHA-512	RSASHA512	Y	*	[RFC5702][proposed standard]
11	Reserved				[RFC6725]
12	GOST R 34.10-2001	ECC-GOST	Y	*	[RFC5933][standards track]
13	ECDSA Curve P-256 with SHA-256	ECDSAP256SHA256	Y	*	[RFC6605][standards track]
14	ECDSA Curve P-384 with SHA-384	ECDSAP384SHA384	Y	*	[RFC6605][standards track]
15-122	Unassigned				
123-251	Reserved				[RFC4034][RFC6014]

Background Questions

- – Is ECC a “well supported” crypto protocol?
- – Is it a reasonable candidate crypto protocol for use as the signing algorithm for the root key of the DNS?
- – Is ECC as widely supported as RSA?

The ECC Question

Is there a clear signal of a set of DNS resolvers who are evidently performing DNSSEC validation using RSA-based crypto algorithms, but fail to understand ECC?

The Test Environment

We used the Google Ad network to deliver a set of DNS tests to clients to determine whether (or not) they use DNSSEC validating resolvers

We used 4 tests:

1. no DNSSEC-signature at all
2. DNSSEC signature using RSA-based algorithm
3. DNSSEC signature using broken RSA-based algorithm
4. DNSSEC signature using ECC-based algorithm

The Test Environment

d.t10000.u2045476887.s1412035201.i5053.vne0001.4f167.z.dashnxdomain.net Unsigned

e.t10000.u2045476887.s1412035201.i5053.vne0001.4f167.z.dotnxdomain.net RSA Signed

f.t10000.u2045476887.s1412035201.i5053.vne0001.4f168.z.dotnxdomain.net RSA signed (Badly)

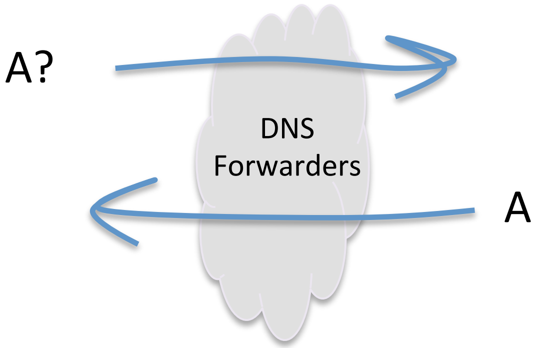
g.t10000.u2045476887.s1412035201.i5053.vne0001.4f167.y.dotnxdomain.net ECC-Signed

Mapped to a wildcard in the zone file

Unique Signed Zone

A Naïve View

A non-DNSSEC-validating resolver query:



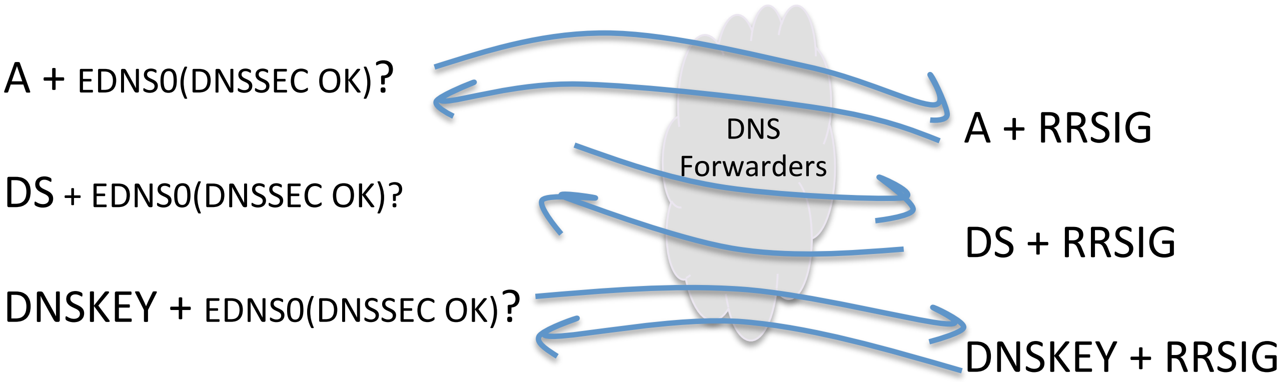
A?

DNS Forwarders

A

Single A Query

A DNSSEC-Validating resolver query:



A + EDNS0(DNSSEC OK)?

DS + EDNS0(DNSSEC OK)?

DNSKEY + EDNS0(DNSSEC OK)?

DNS Forwarders

A + RRSIG

DS + RRSIG

DNSKEY + RRSIG

A, DS, DNSKEY Queries

DNSSEC Validation Queries

```
e.t10000.u2045476887.s1412035201.i5053.vne0001.4f167.z.dotnxdomain.net
```

Query for the A resource record with EDNS0, DNSSEC-OK

    query: e.t10000.u204546887.s1412035201.i5053.vne0001.4f167.z.dotnxdomain.net IN A +ED

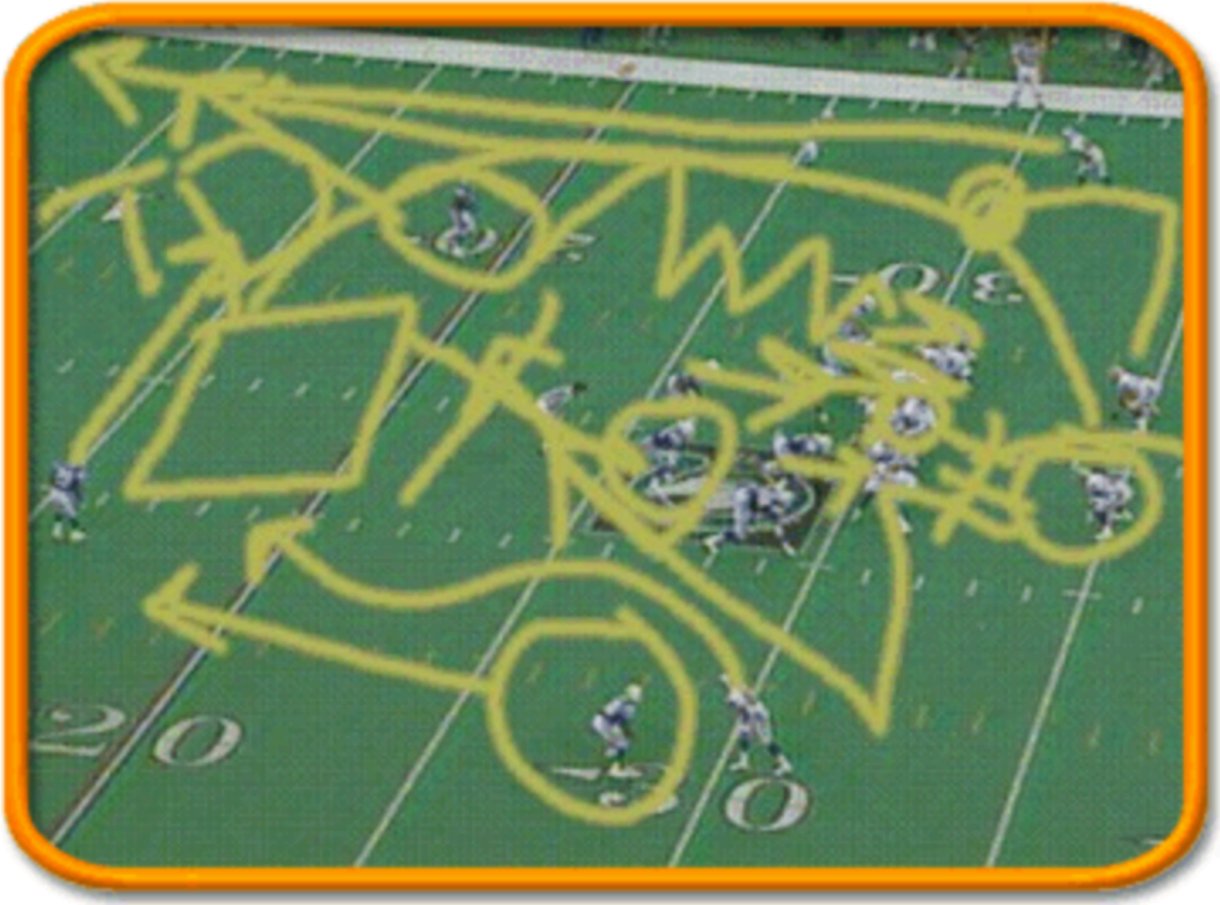
Query the parent domain for the DS resource record

    query: 2f7b3.z.dotnxdomain.net): query: 4f167.z.dotnxdomain.net IN DS +ED

Query for the DNSKEY resource record

    query: 2f7b3.z.dotnxdomain.net): query: 4f167.z.dotnxdomain.net IN DNSKEY +ED

What We See



The DNS is (really) messy

- The DNS is a non-deterministic environment, and the signals one sees at resolvers and servers can be incredibly confusing
- We are priming each client with a unique DNS name, and watching the DNS query traffic that appears on the only authoritative server for that name
- What we see are a variety of query patterns that reach our authoritative name server

What We See (a small random sample)

```
3fb0f.1410333 queries:AADD
5c323.1410361 queries:a
660e3.1410366 queries:aA
415ad.1410383 queries:A
733e3.1410317 queries:AADK
6dce7.1410371 queries:A
3d2c5.1410325 queries:A
5b739.1410360 queries:A
5be73.1410361 queries:A
557e1.1410350 queries:AAa
46693.1410334 queries:a
702b7.1410373 queries:a
3f1ab.1410332 queries:ADK
70b99.1410314 queries:AA
6d4dd.1410372 queries:AADDKK
585b3.1410359 queries:AA
49d2f.1410393 queries:ADAAADAADAADADAAAA
```

```
3f731.1410329 queries:ADK
4cc9d.1410339 queries:A
47877.1410338 queries:a
489f5.1410337 queries:A
4b439.1410349 queries:A
77829.1410325 queries:ADK
5ebf5.1410360 queries:A
5f6f1.1410362 queries:A
49261.1410337 queries:a
4e5ff.1410341 queries:A
413db.1410332 queries:a
5a5cd.1410357 queries:ADKADK
73129.1410375 queries:A
78a73.1410385 queries:A
47459.1410336 queries:A
48a8f.1410337 queries:AKD
72fed.1410317 queries:AAa
5b6cb.1410362 queries:AAaA
76bf9.1410324 queries:aA
```

A = A + DNSSEC-OK   a = A without DNSSEC-OK

Why Do We See What We See?

The DNS has no “trace” in its queries to help diagnosis

- Clients use multiple name servers, and use local timeouts to repeat the query
- Resolvers may use server farms, so that queries from a common logical resolution process may be presented to the authoritative name server from multiple resolvers, and each resolver may present only a partial set of validation queries
- Resolvers may use forwarding resolvers, and may explicitly request checking disabled to disable the forwarding resolver from performing validation itself
- Clients and resolvers have their own independent retry and abandon timers

First Approach to answering the ECC question – Statistical Inference

- A DNSSEC-aware resolver encountering a RR with an attached RRSIG that uses a known algorithm will query for DS and DNSKEY RRs

- A DNSSEC-aware resolver encountering a RR with an attached RRSIG that uses an unknown/unsupported crypto algorithm appears ***not*** to query for the DNSKEY RRs

Results

Over 22 days in September 2014 we saw:

3,773,420 experiments

937,166 experiments queried for the DNSKEY RR of a validly signed (RSA) domain (24.8%)

629,726 experiments queried for the DNSKEY RR of a validly signed (ECC) domain (16.6%)

Results

Over 22 days in September 2014 we saw:

3,773,420 experiments

937,166 experiments queried for the DNSKEY RR of a validly signed (RSA) domain (24.8%)

629,726 experiments queried for the DNSKEY RR of a validly signed (ECC) domain (16.6%)

If we assume that the DNSKEY query indicates that the resolver "recognises" the protocol, then it appears that there is a fall by 8.2% in validation when using the ECC protocol

1 in 3 RSA experiments that fetched the DNSKEY did not fetch the ECC DNSKEY

Hmmm

- How does this relate to affected users?
- How do validating resolvers manage an unrecognised algorithm failure?

- Lets try again and look at both DNS query and web log data

DNS resolver failure modes for an unknown signing algorithm

If a DNSSEC-Validating resolver receives a response RRSIG with an unknown crypto algorithm does it:

- Immediately stop resolution and return a status code of SERVFAIL?
- Fetch the DS RR and then return a status code of SERVFAIL?
- Fetch the DS and DNSKEY RRs and then return a status code of SERVFAIL?
- Or does it abandon validation, query without the DO bit and just return the unvalidated query result?

Second Approach to answering the ECC question – DNS + WEB

```
Data collection: 10/9/14 – 4/10/14

552,104 clients who appear to be exclusively using RSA DNSSEC-Validating resolvers

ECC Results:
  Success:       76.45%   361,698  Saw fetch of the DNSSEC RRs and the URL

Fetched the URL but appeared not to validate
  Failure (1)    19.64%   108,411   Did not see query of DNSKEY, but fetched the URL
  Failure (2)     1.47%     8,121   Saw only A queries, but fetched the URL
  Failure (3)     0.84%     4,615   Saw queries with DO set and not set, fetched the URL

Did not fetch the URL
  Failure (4)     1.07%     5,927   Saw query of the DNSSEC RRs, NOT URL
  Failure (5)     0.34%     1,875   Saw query of A, DS, not DNSKEY, NOT URL
  Failure (6)     0.12%       655   Saw only A queries, NOT URL
  Failure (7)     0.08%       436   Saw queries with DO set and not set, NOT URL

  Apparent Fail: 23.55%   130,040
```

Results

- These results show that 76% of clients who appeared to exclusively use RSA DNSSEC-Validating resolvers were also seen to perform validation using ECC
- 22% of the the remaining clients fetched the object, even though the DNS queries showed that there was not a complete DNSSEC validation pass being performed
- Just 1.6% of clients did NOT fetch the URL

What?

- 23.6% ECC validation failure is very surprising
  - Don’t forget that the subsection of users’ resolvers being polled here already did RSA validation and appeared to correctly return SERVFAIL when the DNSSEC crypto was broken

- The fact that most of the failures result in a fetch of the URL is even more surprising
  - The expectation was that we would see far more SERVFAIL and far higher URL fail-to-fetch rates
  - It seems that the resolvers involved in this behaviour appear to be tagging the domain as “not validatable” and passing back an “insecure” outcome

Where?

ECC failure rates – the % of users in each country who use RSA DNSSEC validating resolvers, but fail to validate when the DNSSEC crypto algorithm is ECC. Top 24 countries, ranked by Observed ECC Validation failure rates

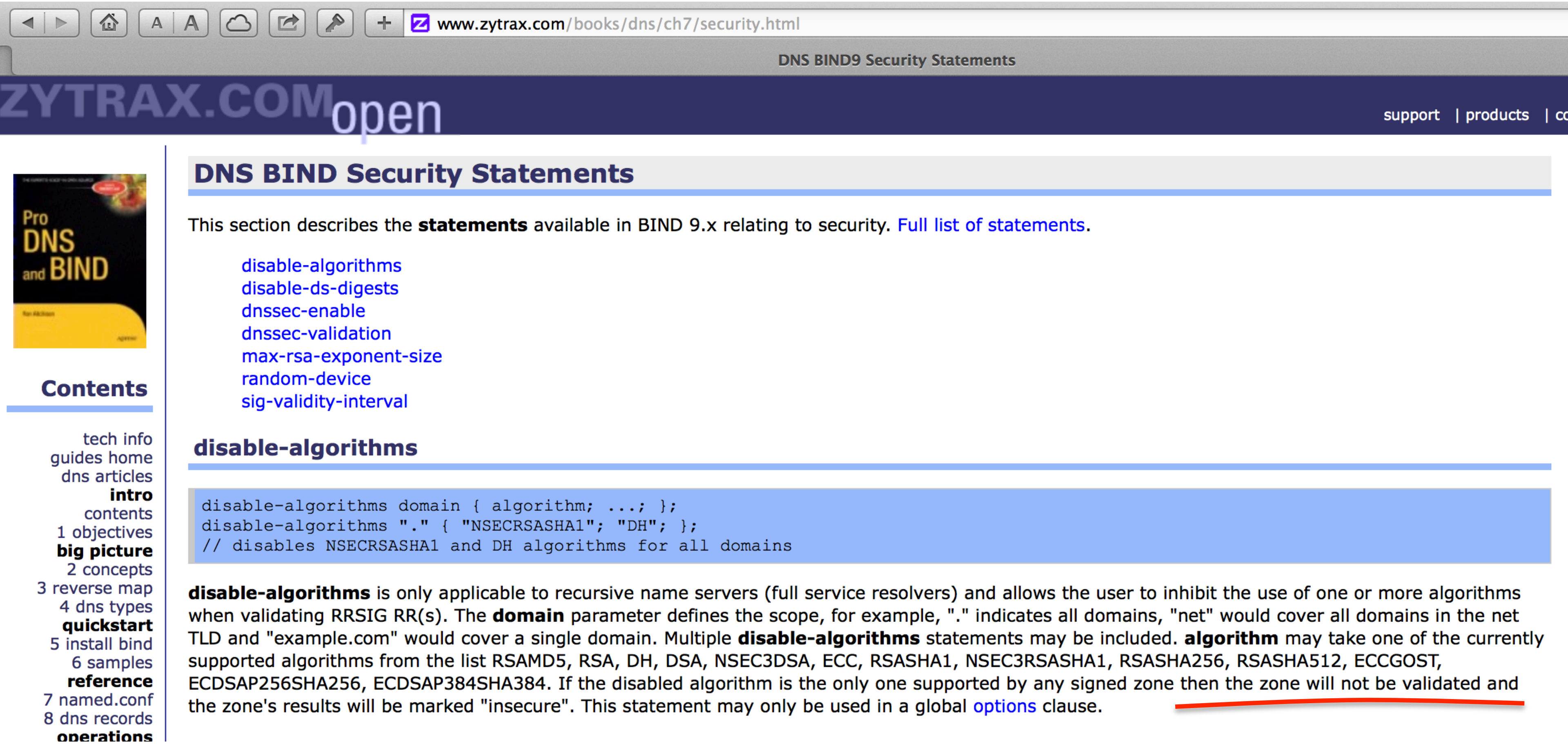
1	MN	96.82	Mongolia
2	MT	96.68	Malta
3	FI	95.75	Finland
4	AD	93.41	Andorra
5	CY	92.61	Cyprus
6	BB	90.59	Barbados
7	FJ	89.93	Fiji
8	ZA	85.94	South Africa
9	AG	84.51	Antigua and Barbuda
10	LU	83.28	Luxembourg
11	AU	79.93	Australia
12	SI	79.51	Slovenia
13	NO	78.91	Norway
14	LY	77.13	Libya
15	YE	75.81	Yemen
16	GR	69.64	Greece
17	KW	68.69	Kuwait
18	RW	66.67	Rwanda
19	BY	63.38	Belarus
20	UA	62.15	Ukraine
21	KE	60.57	Kenya
22	BA	56.35	Bosnia and Herzegovina
23	JP	56.06	Japan
24	KZ	49.50	Kazakhstan

Who?

ECC failure rates – the % of users in each AS who use RSA DNSSEC validating resolvers, but fail to validate when the DNSSEC crypto algorithm is ECC – top 25 Ases ranked by ECC failure rate

	AS	Fail Rate	Samples	AS Description
1	7155	100.00	202	WB-DEN2 - Viasat Communications Inc.,US
2	44143	100.00	662	VIPMOBILE-AS Vip mobile d.o.o.,RS
3	22363	100.00	157	PHMGMT-AS1 - Powerhouse Management, Inc.,US
4	12638	99.53	215	AS12638 E-Plus Mobilfunk GmbH & Co. KG,DE
5	33929	99.39	164	MASICOM-AS Telemach d.o.o.,SI
6	37457	99.36	933	Telkom-Internet,ZA
7	16014	99.25	398	EE-EMT AS EMT,EE
8	10219	99.17	362	SKYCC-AS-MAIN SKY C&C LLC,MN
9	7679	99.11	450	QTNET Kyushu Telecommunication Network Co.,Inc.,JP
10	1759	98.98	2,644	TSF-IP-CORE TeliaSonera Finland IP Network,FI
11	11815	98.97	291	Cooperativa Telefonica de V.G.G. Ltda.,AR
12	16232	98.79	1,238	ASN-TIM TIM (Telecom Italia Mobile) Autonomous System,IT
13	5603	98.77	5,039	SIOL-NET Telekom Slovenije d.d.,SI
14	17711	98.71	155	NDHU-TW National Dong Hwa University,TW
15	4804	98.70	1,456	MPX-AS Microplex PTY LTD,AU
16	12644	98.60	930	TELEMACH Telemach Autonomous System,SI
17	15735	98.58	1,059	DATASTREAM-NET GO p.l.c.,MT
18	53142	98.57	210	Friburgo Online LTDA ME,BR
19	41164	98.13	267	GET-NO GET Norway,NO
20	7992	97.94	679	COGECOWAVE - Cogeco Cable,CA
21	44489	97.31	335	STARNET Starnet s.r.o.,CZ
22	39651	96.82	943	COMHEM-SWEDEN Com Hem Sweden,SE
23	27813	96.70	485	Teledifusora S.A.,AR
24	47956	96.50	371	XFONE XFONE COMMUNICATION LTD,IL
25	52263	96.14	233	Telecable Economico S.A.,CR

Is this old resolver code and/or a design choice?



DNS BIND Security Statements

This section describes the **statements** available in BIND 9.x relating to security. Full list of statements.

- disable-algorithms
- disable-ds-digests
- dnssec-enable
- dnssec-validation
- max-rsa-exponent-size
- random-device
- sig-validity-interval

disable-algorithms

```
disable-algorithms domain { algorithm; ...; };
disable-algorithms "." { "NSECRSASHA1"; "DH"; };
// disables NSECRSASHA1 and DH algorithms for all domains
```

disable-algorithms is only applicable to recursive name servers (full service resolvers) and allows the user to inhibit the use of one or more algorithms when validating RRSIG RR(s). The **domain** parameter defines the scope, for example, "." indicates all domains, "net" would cover all domains in the net TLD and "example.com" would cover a single domain. Multiple **disable-algorithms** statements may be included. **algorithm** may take one of the currently supported algorithms from the list RSAMD5, RSA, DH, DSA, NSEC3DSA, ECC, RSASHA1, NSEC3RSASHA1, RSASHA256, RSASHA512, ECCGOST, ECDSAP256SHA256, ECDSAP384SHA384. If the disabled algorithm is the only one supported by any signed zone then the zone will not be validated and the zone's results will be marked "insecure". This statement may only be used in a global options clause.

Is ECC a viable crypto algorithm for the Root?

These results don’t look especially promising for the use of ECC in this context

Next steps

- If we used ECC in the validation path, not at the terminal zone, would we see a similar outcome?
- What are the behaviours of resolvers when encountering an unknown crypto algorithm? Should we experiment with other algorithm code values?

Questions?