ECDSA P-256 support in DNSSEC-validating Resolvers

Geoff Huston
APNIC Labs
March 2016

ECDSA

- Elliptic Curve Cryptography allows for the construction of “strong” public/private key pairs with key lengths that are far shorter than equivalent strength keys using RSA

  “256-bit ECC public key should provide comparable security to a 3072-bit RSA public key” *

- And the DNS protocol has some sensitivities over size when using UDP
  – UDP fragmentation has its issues in both V4 and V6

* http://en.wikipedia.org/wiki/Elliptic_curve_cryptography

ECDSA vs RSS

```
$ dig +dnssec u5221730329.s1425859199.i5075.vcf100.5a593.y.d

; <<>> DiG 9.9.6-P1 <<>> +dnssec u5221730329.s1425859199.i50
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 61126
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADI

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;u5221730329.s1425859199.i5075.vcf100.5a593.y.dotnxdomain.ne

;; ANSWER SECTION:
u5221730329.s1425859199.i5075.vcf100.5a593.y.dotnxdomain.net
u5221730329.s1425859199.i5075.vcf100.5a593.y.dotnxdomain.net

;; AUTHORITY SECTION:
ns1.5a593.y.dotnxdomain.net. 1      IN      NSEC    x.5a593.y
ns1.5a593.y.dotnxdomain.net. 1      IN      RRSIG   NSEC 13 5
5a593.y.dotnxdomain.net. 3598 IN    NS      ns1.5a593.y.dotn
5a593.y.dotnxdomain.net. 3600 IN    RRSIG   NS 13 4 3600 202

;; Query time: 1880 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Mar 12 03:59:42 UTC 2015
;; MSG SIZE  rcvd: 527
```

ECDSA signed response – 527 octets

```
$ dig +dnssec u5221730329.s1425859199.i5075.vcf100.5a593.z.dotnxdomain.ne

; <<>> DiG 9.9.6-P1 <<>> +dnssec u5221730329.s1425859199.i5075.vcf100.5a
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 25461
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 2, AUTHORITY: 4, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;u5221730329.s1425859199.i5075.vcf100.5a593.z.dotnxdomain.net. IN A

;; ANSWER SECTION:
u5221730329.s1425859199.i5075.vcf100.5a593.z.dotnxdomain.net. 1    IN A 1
u5221730329.s1425859199.i5075.vcf100.5a593.z.dotnxdomain.net. 1    IN RRS

;; AUTHORITY SECTION:
33d23a33.3b7acf35.9bd5b553.3ad4aa35.09207c36.a095a7ae.1dc33700.103ad556.
33d23a33.3b7acf35.9bd5b553.3ad4aa35.09207c36.a095a7ae.1dc33700.103ad556.
5a593.z.dotnxdomain.net. 3599 IN    NS      nsz1.z.dotnxdomain.net.
5a593.z.dotnxdomain.net. 3600 IN    RRSIG   NS 5 4 3600 20200724235900 2

;; Query time: 1052 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Thu Mar 12 03:59:57 UTC 2015
;; MSG SIZE  rcvd: 937
```

RSA signed response – 937 octets

So lets use ECDSA for DNSSEC

Yes!

So lets use ECDSA for DNSSEC

Or maybe we should look before we leap...

– Is ECDSA a "well supported" crypto protocol?
– If you signed using ECDSA would resolvers validate the signature?

The Test Environment

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

We use 5 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 ECDSA P-256 algorithm
5. DNSSEC signature using broken ECDSA P-256 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)

m.t10000.u2045476887.s1412035201.i5053.vne0001.4f167.y.dotnxdomain.net ECDSA-Signed

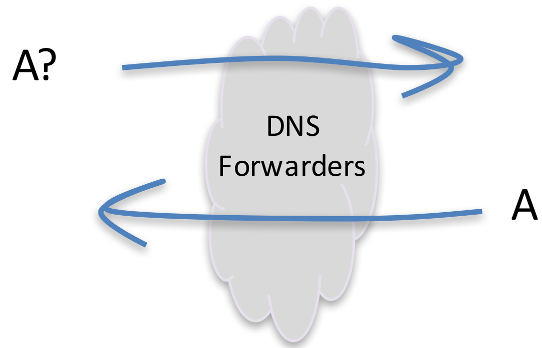
n.t10000.u2045476887.s1412035201.i5053.vne0001.4f168.y.dotnxdomain.net ECDSA-Signed (bad!)

Mapped to a wildcard in the zone file

Unique Signed Zone

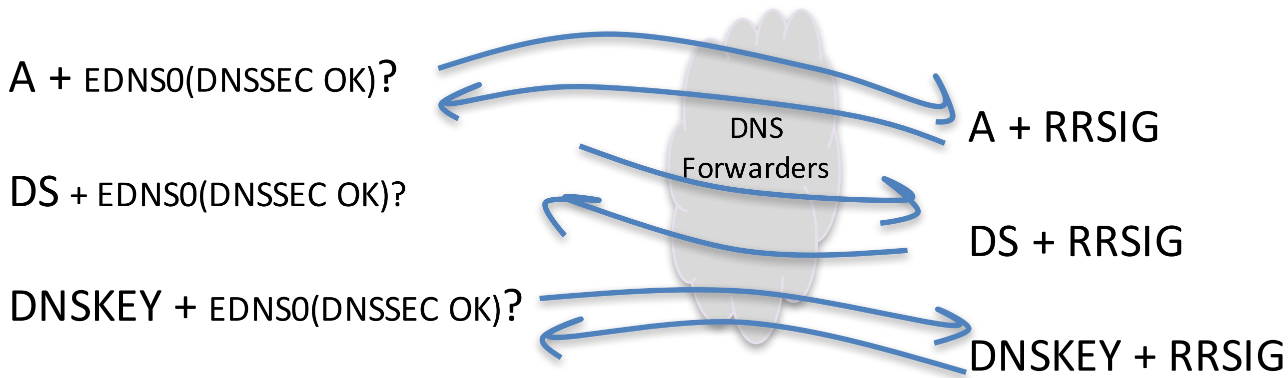
A Naive View of the DNS in Operation

A non-DNSSEC-validating resolver query:



Seen: Single A Query

A DNSSEC-Validating resolver query:



Seen: A, DS, DNSKEY Queries

Theory: DNSSEC Validating Queries

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

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

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

2. Query the parent domain for the **DS** resource record

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

3. Query for the **DNSKEY** resource record

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

Practice: The DNS is "messy"

- Clients typically use multiple resolvers, and use local timeouts to repeat the query across these resolvers
- Resolvers may use slave farms, so that queries from a common logical resolution process may be presented to the authoritative name server from multiple resolvers, and each slave resolver that directs queries to servers 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

DNS Mess!

Queries for a single badly signed (RSA) name:

```
Resolver        Queries
200.55.224.68: A#,K#,D#
74.125.19.147: A#,D#,K#,D#,D#
74.125.19.145: K#,K#
200.55.224.67: A#,A#,A,K#,K#,D#
74.125.19.148: D#
```

Queries via ISP resolver set

Via Google PDNS Slave Resolvers

What is going on here?

#: EDNS(0) DNSSEC OK flag set

DNS Mess!

Queries for a single badly signed (RSA) name:

```
Resolver         Queries
200.55.224.68:  A#,K#,D#
74.125.19.147:  A#,D#,K#,D#,D#
74.125.19.145:  K#,K#
200.55.224.67:  A#,A#,A,K#,K#,D#
74.125.19.148:  D#
```

Failed validation (SERVFAIL) from the initial query to ISP resolver causes client to ask Google PDNS resolver

Failed validation appears to cause client to repeat the query to Google PDNS 2 further times

Failed validation appears to cause client to repeat the query to ISP's resolver 2 (or 3?) further times

No clue why this is an orphan DS query!

#: EDNS(0) DNSSEC OK flag set

DNS resolver failure modes for an unknown signing algorithm

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

- ❑ Immediately stop resolution and return a status code of SERVFAIL?
- ❑ Fetch the DNSKEY RR and then return a status code of SERVFAIL?
- ❑ Abandon validation and just return the unvalidated query result?

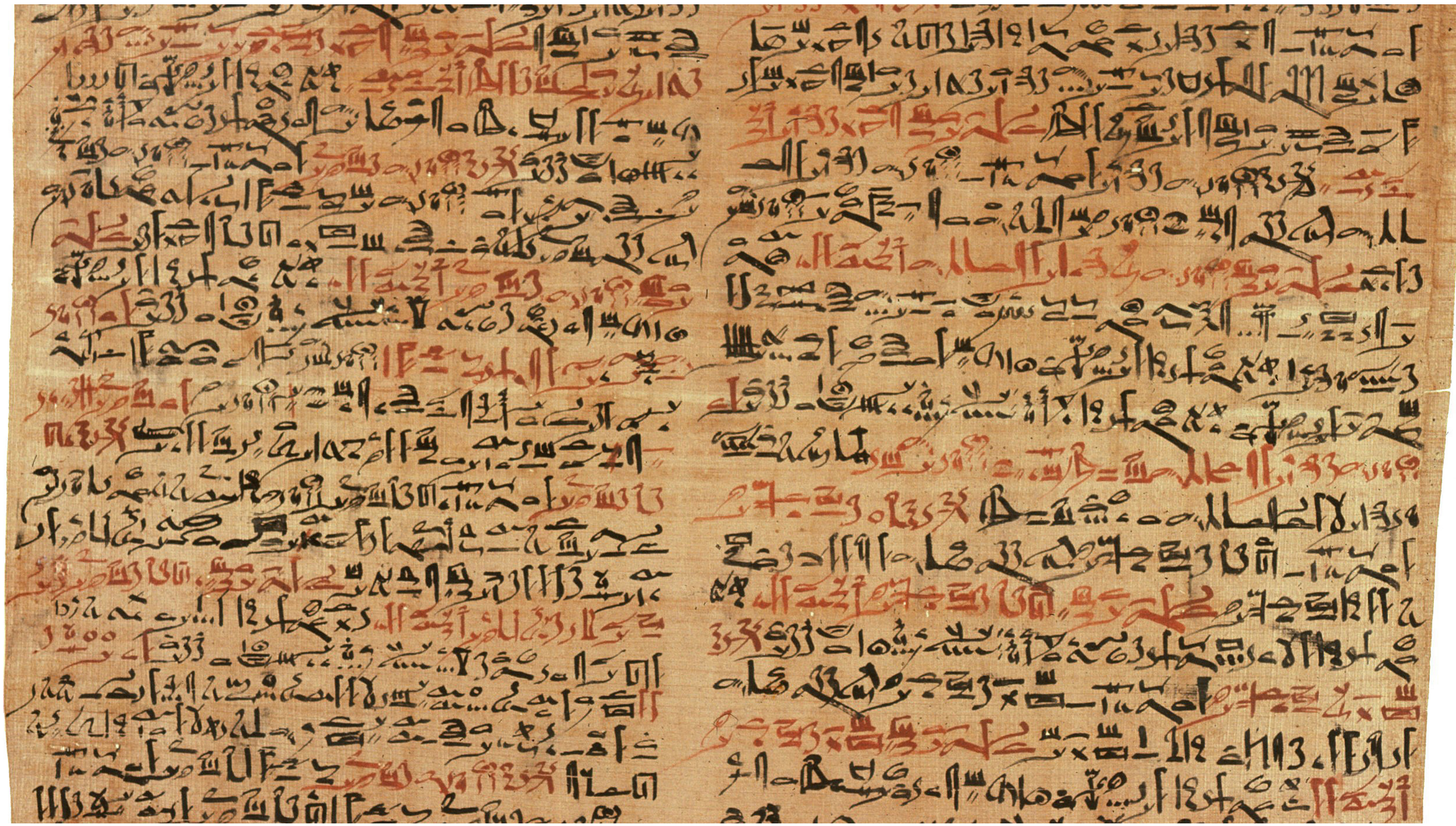
DNS resolver failure modes for an unknown signing algorithm

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

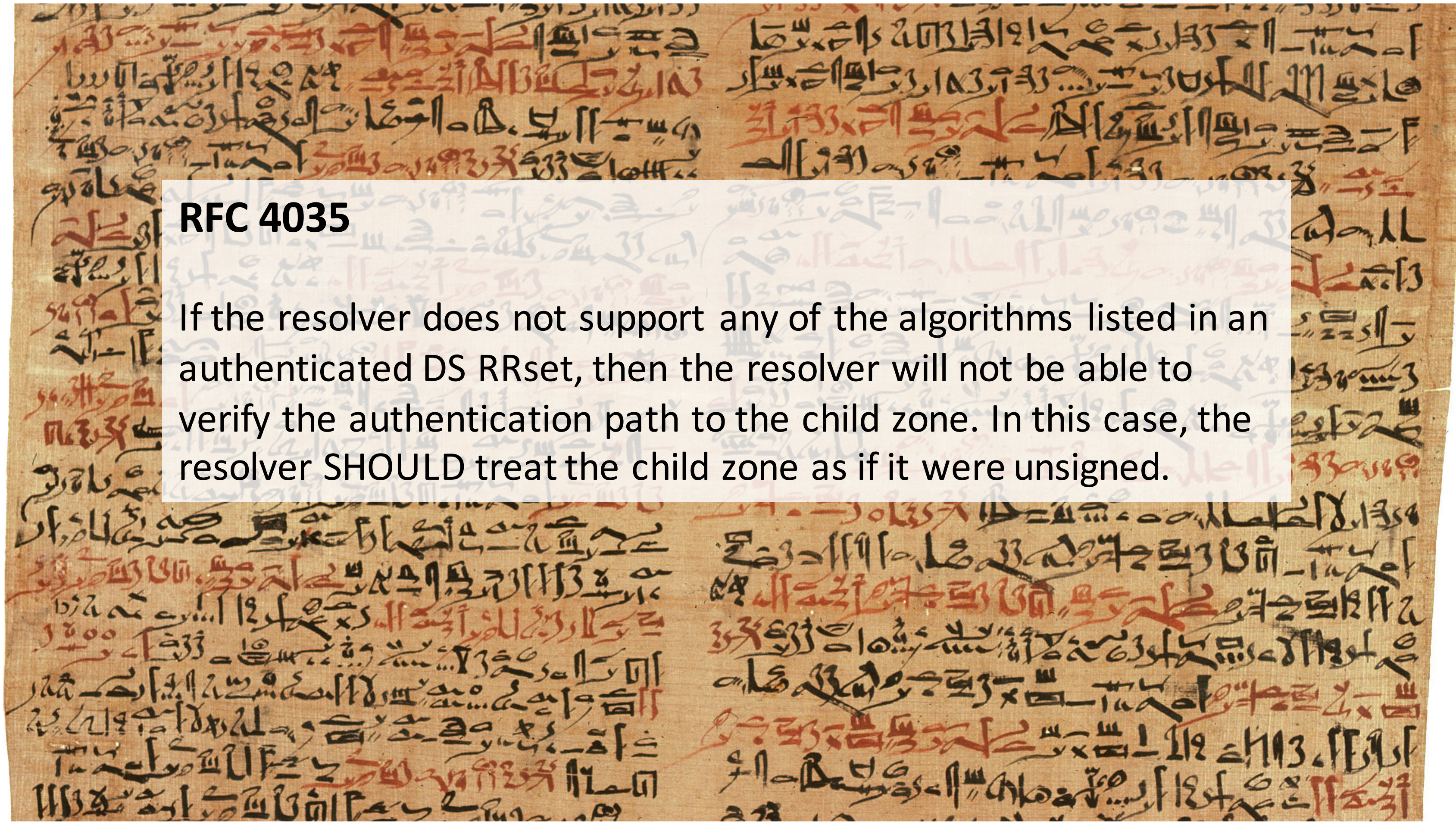
- ❑ Immediately stop resolution and return a status code of SERVFAIL?
- ❑ Fetch the DNSKEY RR and then return a status code of SERVFAIL?
- ☒ Abandon validation and just return the unvalidated query result?

So if the resolver doesn't recognize the protocol in the authenticated DS record then there is no point in pulling the DNSKEY record

The Words of the Ancients



The Words of the Ancients



RFC 4035

If the resolver does not support any of the algorithms listed in an authenticated DS RRset, then the resolver will not be able to verify the authentication path to the child zone. In this case, the resolver SHOULD treat the child zone as if it were unsigned.

First Approach to answering the ECDSA 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: 2014

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%**)

1 in 3 experiments that fetched the DNSKEY in RSA did not fetch the ECDSA-signed DNSKEY

And then we changed things…

We changed the Test Rig

- We were using a setup of:
  - cycling through 250,000 unique signed domains, with a 3 minute TTL
  - And serving 500,000 ads per day
  - All over port 80

- Now we need to cope with 10 – 20 M ads per day, and allow for secure access to essentially an unbounded namespace of signed subdomains

The RSA DNSSEC Validator Test Rig

Authoritative server for RSA-signed zone

- EVL DNS implementation (*)
- Acts as if there is a wildcard signed delegated child zone
- But the contents of the synthetic delegated zone is just the origin name
- A single authoritative server instance serves both child and parent zones

* Thanks to Ray Bellis and Nominet and ISC

Then we changed it again!

The ECDSA DNSSEC Validator Test Rig

For ECDSA we use a second implementation* of this synthetic wildcard subdomain using three distinct authoritative servers:

- The parent and child servers are separate servers
- And the glue records of the delegation are only accessible from a separate glue zone server
- NS records are not validated, so the glue zone query logs are not used for this particular test
- This "glueless" form of delegation and the explicit separation of parent and child might alter some resolver behaviour with respect to validation queries

* Thanks to Ray Bellis and Nominet and iSC

Hmmm

Did we tickle unanticipated resolver behaviour by using a glueless structure of synthetic signed subdomains?

Let’s check by using an experiment that has both glue and glueless RSA-signed records

RSA - Glue vs Glueless

Validated RSA with Glue:	2,889,062
– Saw both Glue and Glueless Queries:	2,355,369
– Validated Glueless with RSA	2,258,026 (96%)

It appears that the shift from Glued to a Glueless delegation does not have a major impact on DNS resolver behaviour

So now let's check RSA vs ECDSA

Results

Over 45 days in December 2015 – January 2016 we saw:

765,257,019 completed experiments

208,980,333 experiments queried for the DNSKEY RR of a validly signed (RSA) domain **(27.3%**)

183,240,945 experiments queried for the DNSKEY RR of a validly signed (ECDSA) domain (**23.9%**)

Results

Over 45 days in December 2015 – January 2016 we saw:

765,257,019 completed experiments

208,980,333 experiments queried for the DNSKEY RR of a validly signed (RSA) domain **(27.3%)**

183,240,945 experiments queried for the DNSKEY RR of a validly signed (ECDSA) domain (**23.9%**)

If we assume that the DNSKEY query indicates that the resolver "recognizes" the protocol, then it appears that there is a fall by 19.5% in validation when using the ECC protocol

1 in 5 RSA experiments that fetched the DNSKEY did not fetch the ECC DNSKEY

Results: 2016

Over 45 days in December 2015 – January 2016 we saw:

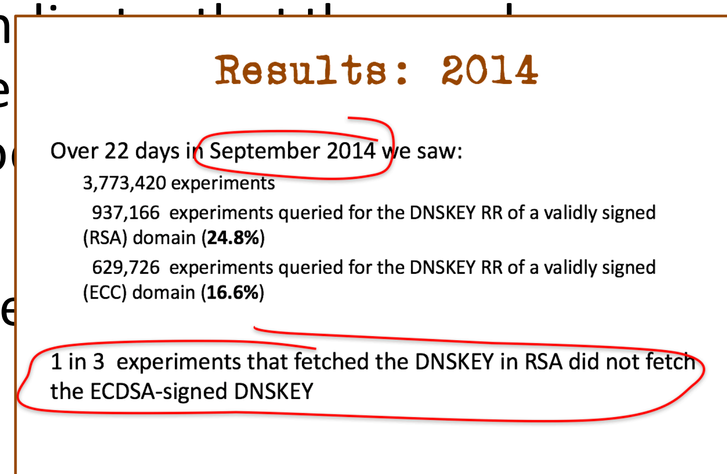
765,257,019 completed experiments

208,980,333 experiments queried for the DNSKEY RR of a validly signed (RSA) domain **(27.3%)**

183,240,945 experiments queried for the DNSKEY RR of a validly signed (ECDSA) domain (**23.9%**)

If we assume that the DNSKEY query in
"recognizes" the protocol, then it appe
in validation when using the ECC protoc

1 in 5 RSA experiments that fetched the
DNSKEY


Results: 2014

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%**)

1 in 3 experiments that fetched the DNSKEY in RSA did not fetch the ECDSA-signed DNSKEY

Second Approach to answering the ECC question - DNS + WEB

Data collection: 1/1/16 – 16/2/16

64,948,234 clients who appear to be exclusively using RSA DNSSEC-Validating resolvers

ECC Results:

Success:	82%	53,514,518	Saw fetches of the ECC DNSSEC RRs and the well-signed named URL, but not the badly signed named URL

Failure (fetched both URLs):

Mixed Resolvers	1.9%	1,218,240	Used both ECDSA-Validating and non-validating resolvers
NO ECC	13.0%	8,461,551	Saw A, DS, no DNSKEY, fetched both URLs
Mixed	0.5%	352,914	Saw some DNSSEC queries, fetched both URLs
No Validation	2.2%	1,401,011	Did not fetch any DNSSEC RRs

Apparent Fail: 17.6% 11,433,716

1 in 6 clients that use resolvers that perform DNSSEC validation with RSA fail to validate with ECDSA

Results

- These results show that 82% of clients who appeared to exclusively use RSA DNSSEC-Validating resolvers were also seen to perform validation using ECDSA
- Two thirds of the the remaining clients fetched both objects (13% of the total), but did not fetch any DNSKEY RRs.
- Of the remainder (5%), most were using a validating resolver (which returned SERVFAIL for the badly signed object), and then the client failed over to a non-validating resolver *

* This is curious, because these clients did not failover to a non-validating resolver on a badly signed RSA structure

Where?

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

Rank	CC	Failure	Samples	Country Name
1	DM	98.44	25,468	Dominica
2	AI	95.51	15,939	Anguilla
3	YT	95.37	1,748	Mayotte
4	BB	94.67	195,691	Barbados
5	AD	94.50	101,874	Andorra
6	LU	91.62	77,433	Luxembourg
7	AG	89.80	74,758	Antigua and Barbuda
8	MT	89.50	69,632	Malta
9	TJ	89.26	14,595	Tajikistan
10	BY	81.02	220,418	Belarus
11	PS	78.84	617,909	Occupied Palestinian Territory
12	ZA	75.60	66,205	South Africa
13	BM	75.04	16,371	Bermuda
14	MV	74.56	57,964	Maldives
15	GE	73.97	173,639	Georgia
16	LY	72.07	83,420	Libya
17	NZ	70.00	287,090	New Zealand
18	SI	69.78	1,650,816	Slovenia
19	KE	68.41	120,764	Kenya
20	VC	66.57	3,715	Saint Vincent and the Grenadines
21	AM	65.44	170,124	Armenia
22	MW	62.92	15,150	Malawi
23	LR	62.07	7,324	Liberia
24	MK	55.11	389,592	The former Yugoslav Republic of Macedonia
25	BA	54.97	192,2461	Bosnia and Herzegovina

Which AS?

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

	AS	Fail Rate	Samples	AS Description
1	AS57481	99.97	3,235	ASMULTISOL Multiservice Ltd., BY
2	AS22252	99.91	1,142	AS22252 – The City of New York, US
3	AS30852	99.85	5,838	VIS OJSC Volgainformnet, RU
4	AS10297	99.73	514,003	ENET-2 – eNET Inc., US
5	AS25	99.64	3,296	UCB – University of California at Berkeley, US
6	AS54934	99.63	1,093	JC-39-AS – JEFFERSON CO. CABLE, INC., US
7	AS59815	99.54	10,304	TRK-METRO-AS TRK Metro LLC, UA
8	AS25031	99.51	33,646	NOVARTIS-CH Novartis, CH
9	AS11596	99.50	5,774	BESTBUY – Best Buy Co., Inc., US
10	AS16299	99.37	36,497	XFERA Xfera Moviles SA, ES
11	AS17071	99.37	1,103	UBSW-STAMFORD – UBS AG, US
12	AS63089	99.36	1,873	SST – Salina Spavinaw Telephone Company, Inc, US
13	AS57990	99.35	1,227	ASALIEV PE Aliev Murad Ahmedovich, RU
14	AS58600	99.34	7,865	FLIP-AS-AP Flip Services Limited, NZ
15	AS33067	99.30	997	CLASSICSOUTHCOMM – Classic South Communications, L.L.C., US
16	AS31286	99.26	2,685	INTELSET-AS MTS PJSC, RU
17	AS8416	99.18	12,068	INFOLINE-AS Infoline Ltd., RU
18	AS17253	99.15	4,246	COMMUNIGROUP – TEC of Jackson, Inc., US
19	AS42082	99.15	23,525	GEOCELL GEOCELL Ltd, GE
20	AS394111	99.14	6,783	FRTCCNET – Foothills Rural Telephone Cooperative Corporation, Inc., US
21	AS51158	99.12	6,821	MTREND-AS Mobile Trend Ltd, RU
22	AS21310	99.08	19,401	ASN-SATELLITE Satellite Ltd, UA
23	AS40091	99.03	1,030	WVVANET – WVVA.net Inc., US
24	AS20879	99.01	1,110	MICRONET SC Servicii Micronet SRL, RO
25	AS4385	98.95	1,722	RIT-ASN – Rochester Institute of Technology, US

Which Resolver?

This filter involves:

- pick out those experiments where the invalidly-signed URL was retrieved (i.e. either no DNSSEC Validation is being performed OR the validator does not recognize ECDSA
- pick out those resolvers that asked for the A and DS RRs' but NOT the DNSKEY for this experiment
- note if the resolver asked for the DNSKEY RR

- pick out those resolvers that asked for A and DS every time they were used

Which Resolver?

Most intensively used RSA-validating resolvers that appear to lack support for ECDSA

Rank	Resolver	Use	AS	AS Description
1	195.222.32.20	308,779	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
2	80.65.92.113	266,115	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
3	122.2.166.129	256,126	AS9299	IPG-AS-AP Philippine Long Distance Telephone Company, PH
4	84.20.224.66	244,499	AS33929	MASICOM-AS Telemach d.o.o., SI
5	193.189.177.55	240,733	AS5603	SIOL-NET Telekom Slovenije d.d., SI
6	80.65.92.61	238,450	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
7	93.91.200.207	227,153	AS21277	NWRZ Newroz Telecom Ltd. AS Number, IQ
8	195.222.60.60	224,325	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
9	78.87.0.195	219,196	AS6866	CYTA-NETWORK Cyprus Telecommunications Authority, CY
10	82.102.232.202	218,936	AS15975	HADARA-AS Hadara Technologies Private Shareholding Company, PS
11	192.116.18.3	211,441	AS15975	HADARA-AS Hadara Technologies Private Shareholding Company, PS
12	195.222.60.40	202,489	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
13	209.190.123.3	201,629	AS10297	ENET-2 - eNET Inc., US
14	209.190.123.4	201,583	AS10297	ENET-2 - eNET Inc., US
15	209.190.123.2	201,347	AS10297	ENET-2 - eNET Inc., US
16	193.189.177.53	197,740	AS5603	SIOL-NET Telekom Slovenije d.d., SI
17	62.240.32.5	181,917	AS21003	GPTC-AS, LY
18	124.106.6.109	180,466	AS9299	IPG-AS-AP Philippine Long Distance Telephone Company, PH
19	213.226.131.131	176,691	AS13194	BITE UAB "Bite Lietuva", LT
20	195.222.33.216	170,510	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
21	124.106.6.107	168,941	AS9299	IPG-AS-AP Philippine Long Distance Telephone Company, PH
22	192.116.18.2	162,807	AS15975	HADARA-AS Hadara Technologies Private Shareholding Company, PS
23	195.222.32.10	147,196	AS9146	BIHNET BH Telecom d.d. Sarajevo, BA
24	193.189.177.54	124,610	AS5603	SIOL-NET Telekom Slovenije d.d., SI
25	192.235.48.68	122,836	AS14813	BB-COLUMBUS - Columbus Telecommunications (Barbados) Limited, BB

Why?

- These resolvers all generate queries for the A record and the DS record, but did not query for the DNSKEY record when the signing algorithm was ECDSA
- It appears that these resolvers who do not perform the DNSKEY query do not have local support for ECDSA
  - Resolvers do not, in general use a custom crypto library
  - As we saw with the Heartbleed bug, there is a preponderance of use of OpenSSL
  - So perhaps the question is: why doesn't OpenSSL support ECDSA?

ECC patents

From Wikipedia, the free encyclopedia

Patent-related uncertainty around elliptic curve cryptography (ECC), or **ECC patents**, is one of the main factors limiting its wide acceptance. For example, the OpenSSL team accepted an ECC patch only in 2005 (in OpenSSL version 0.9.8), despite the fact that it was submitted in 2002.

According to Bruce Schneier as of May 31, 2007, "Certicom certainly can claim ownership of ECC. The algorithm was developed and patented by the company's founders, and the patents are well written and strong. I don't like it, but they can claim ownership."[1] Additionally, NSA has licensed MQV and other ECC patents from Certicom in a US$25 million deal for NSA Suite B algorithms.[2] (ECMQV is no longer part of Suite B.)

However, according to RSA Laboratories, "*in all of these cases, it is the implementation technique that is patented, not the prime or representation, and there are alternative, compatible implementation techniques that are not covered by the patents.*"[3] Additionally, Daniel J. Bernstein has stated that he is "not aware of" patents that cover the Curve25519 elliptic curve Diffie–Hellman algorithm or its implementation.[4] RFC 6090, published in February 2011, documents ECC techniques, some of which were published so long ago that even if they were patented any such patents for these previously published techniques would now be expired.

Contents [hide]

1 Known patents
2 Certicom's lawsuit against Sony
3 See also
4 References
5 External links

Why?

- OpenSSL added ECDSA support as from 0.9.8
- Other bundles and specific builds added ECDSA support later
- But deployed systems often lag behind the latest bundles, and therefore still do not include ECC support in their running configuration

Why?

- One further observation – most of these wayward non-ECDSA resolvers are housed in telephone service entities
- One possible explanation is that they are running a “packaged” data service for a mobile system as a black box
- And updates applied to this black box are infrequent

Is ECDSA a viable crypto algorithm for DNSSEC?

If the aim is to detect efforts to compromise the DNS for the signed zone, then signing a zone with ECDSA limits the number of DNS resolvers who will validate the signature

Which is a shame, because the shorter key lengths could be attractive for DNS over UDP

ECDSA in the (semi-)wild

```
$ dig +dnssec www.cloudflare-dnssec-auth.com

; <<>> DiG 9.9.6-P1 <<>> +dnssec www.cloudflare-dnssec-auth.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 7049
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 6, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags: do; udp: 4096
;; QUESTION SECTION:
;www.cloudflare-dnssec-auth.com.  IN    A

;; ANSWER SECTION:
www.cloudflare-dnssec-auth.com.   300 IN     A     104.20.23.140
www.cloudflare-dnssec-auth.com.   300 IN     A     104.20.21.140
www.cloudflare-dnssec-auth.com.   300 IN     A     104.20.19.140
www.cloudflare-dnssec-auth.com.   300 IN     A     104.20.22.140
www.cloudflare-dnssec-auth.com.   300 IN     A     104.20.20.140
www.cloudflare-dnssec-auth.com.   300 IN     RRSIGA 13 3 300 20150317021923 20150315001923 35273
cloudflare-dnssec-auth.com. pgBvfQkU4Il8ted2hGL9o8NspvKksDT8/jvQ+4o4h4tGmAX0fDBEoorb
tLiW7mcdOWYLoOnjovzYh3QOOdu0Xw==

;; Query time: 237 msec
;; SERVER: 127.0.0.1#53(127.0.0.1)
;; WHEN: Mon Mar 16 01:19:24 UTC 2015
;; MSG SIZE  rcvd: 261
```

Thanks!