A Story on Unsupported DNSSEC Algorithms

Matthijs Mekking, ISC
rm unsupported algorithms

- RFC 6944
- draft-ietf-dnsop-algorithm-update
- Removed in BIND 9:
  - RSAMD5
  - GOST
  - DSA
  - DSA-NSEC3-SHA1
What to expect?

Bogus:
The validating resolver has a trust anchor and a secure delegation indicating that subsidiary data is signed, but the response fails to validate for some reason: missing signatures, expired signatures, signatures with unsupported algorithms, signatures with missing NSEC RR data missing that the relevant NSEC RR data says should be present, and so forth.

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.

If the validator does not support any of the algorithms listed in an authenticated DS RRset, then the resolver has no supported authentication path leading from the parent to the child. The resolver should treat this case as it would the case of an authenticated NSEC RRset proving that no DS RRset exists, as described above.
Write tests

- Key creation
- Signing
- Publication
- Validation
- Trust anchor
- DLV
- 5011

Belfort, the bell tower of Bruges
Testing DNS software

- BIND 9.13
- Unbound 1.9
- Knot DNS 2.7.6
- Knot Resolver 3.2.1
- PowerDNS 4.1.6
- PDNS Recursor 4.1.12
- OpenDNSSEC 2.1.3
Observations

- The Good
  Most test pass

- The Bad
  Some quirky behavior

- The Ugly
  Found some crashes
Test signers

- dnssec-keygen, etc
  - example.com
  - RSAMD5, 255
- dnssec-signzone
- Automated DNSSEC
- Publish 255 DNSKEY
[1] Key creation should fail

$ dnssec-keygen -a RSAMD5 example.com.
dnssec-keygen: fatal: unsupported algorithm: 1

$ sudo keymgr -c knot.sample.conf example.com. generate algorithm=255
Error (invalid key algorithm)

$ keymgr -c knot.sample.conf example.com. generate algorithm=RSAMD5
Unknown algorithm: RSAMD5
Error (invalid parameter)

pdnsutil generate-zone-key
Syntax: pdnsutil generate-zone-key zsk|ksk [rsasha1|rsasha256|rsasha512|gost|ecdsa256|ecdsa384] [bits]
pdnsutil generate-zone-key ksk rsamd5 768
Generating a KSK with algorithm = 1
Requesting specific key size of 768 bits
Error: Request to create key object for unknown algorithm number 1

ods-enforcerd: 1 zone(s) found on policy "rsamd5"
ods-enforcerd: [hsm_key_factory_generate] 53 keys needed for 1 zones
covering 31536000 seconds, generating 53 keys for policy rsamd5
ods-enforcerd: 53 new ZSK(s) (768 bits) need to be created.
ods-enforcerd: [hsm_key_factory_generate] key generation failed
[1] Key creation should fail

$ dnssec-keygen -a RSAMD5 example.com.
dnssec-keygen: fatal: unsupported algorithm: 1

$ sudo keymgr -c knot.sample.conf example.com. generate algorithm=255
Error (invalid key algorithm)
$ keymgr -c knot.sample.conf example.com. generate algorithm=RSAMD5
Unknown algorithm: RSAMD5
Error (invalid parameter)

pdnsutil generate-zone-key
Syntax: pdnsutil generate-zone-key zsk|ksk [rsasha1|rsasha256|rsasha512|gost|ecdsa256|ecdsa384] [bits]
pdnsutil generate-zone-key ksk rsamd5 768
Generating a KSK with algorithm = 1
Requesting specific key size of 768 bits
Error: Request to create key object for unknown algorithm number 1

ods-enforcerd: 1 zone(s) found on policy "rsamd5"
ods-enforcerd: [hsm_key_factory_generate] 53 keys needed for 1 zones covering 31536000 seconds, generating 53 keys for policy rsamd5
ods-enforcerd: 53 new ZSK(s) (768 bits) need to be created.
ods-enforcerd: [hsm_key_factory_generate] key generation failed
[2] Signing should fail

$ dnssec-signzone foo.example.db Kfoo.example.+001+53634
dnssec-signzone: fatal: cannot load dnskey Kexample.+001+53634: algorithm is unsupported

Configuration excerpt:

```plaintext
zone "foo.example" IN {
    type master;
    file "db/foo.example.db";
    key-directory "keys/";
    auto-dnssec maintain;
}

dns_dnssec_findmatchingkeys: error reading key file Kfoo.example.+001+53634.private: algorithm is unsupported
```

```plaintext
pdnsutil add-zone-key foo.example. ksk 768 active rsamd5
Error: Request to create key object for unknown algorithm number 1
```
[2] Signing should fail

Configuration excerpt:

```
policy:
  - id: unsupported
    algorithm: 255

erro`r: config, file 'knot.conf', line 20, item'algorithm', value '255'
(invalid parameter)

policy:
  - id: dsa
    algorithm: RSAMD5

erro`r: config, file 'knot.conf', line 26, policy 'rsamd5' (RSAMD5
algorithm no longer supported)
```

See [1]
<table>
<thead>
<tr>
<th>Domain</th>
<th>TTL</th>
<th>Type</th>
<th>Algorithm</th>
<th>Key Tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>example.com</td>
<td>3600</td>
<td>DNSKEY</td>
<td>257 3 13</td>
<td>...</td>
</tr>
<tr>
<td>example.com</td>
<td>3600</td>
<td>DNSKEY</td>
<td>257 3 255</td>
<td>...</td>
</tr>
<tr>
<td>example.com</td>
<td>3600</td>
<td>RRSIG</td>
<td>DNSKEY</td>
<td>13 2 3600</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(20190319133330 20190219133330 9826 example.com. ... )</td>
<td></td>
</tr>
</tbody>
</table>
Test validators

- Basic query
- Trust anchors
- RFC 5011
- DLV
[4] Validation status insecure

S dig test.foo.example TXT

example. SOA ...
foo.example. IN NS ns.foo.example.
foo.example. IN DS 303 1 2 ...

foo.example. SOA ...
foo.example. IN DNSKEY 257 3 1 ...
...
test.foo.example. IN TXT ...
test.foo.example. IN RRSIG TXT 1 3 ...
[4] Validation status insecure

;; ANSWER SECTION:
test.foo.example. 60 IN TXT "I am a zone that uses an unsupported DNSSEC algorithm"

validating test.foo.example/TXT: no supported algorithm/digest (example.com/DS)
validating test.foo.example/TXT: marking as answer (proveunsecure (5))

info: Verified that response is INSECURE

AD: request NOT classified as SECURE

Starting validation of answer to test.foo.example|TXT for 127.0.0.1:42959
Answer to test.foo.example|TXT for 127.0.0.1:42959 validates as Insecure
[5] Trust anchors are ignored

$ dig test.foo.example TXT

e x a m p l e.

S

example. SOA ...
...
foo.example. IN NS ns.foo.example.
foo.example. IN DS 303 1 2 ...

foo.example. SOA ...
foo.example. IN DNSKEY 257 3 1 ...
...
test.foo.example. IN TXT ...
test.foo.example. IN RRSIG TXT 1 3 ...

trust-anchor:
foo.example. DNSKEY 257 3 1
Trust anchors are ignored

skipping trusted key for 'foo.example.': algorithm is unsupported
skipping managed key for 'foo.example.': algorithm is unsupported

warning: unsupported algorithm for trust anchor foo.example. DNSKEY IN
warning: trust anchor foo.example. has no supported algorithms, the anchor is ignored (check if you need to upgrade unbound and openssl)

> trust_anchors.add_file('trustanchors.conf')
nil

> PANIC: unprotected error in call to Lua API (/usr/local/lib/kdns_modules/trust_anchors.lua:169: invalid RR: foo.example. 60 DNSKEY 257 3 1 ...: invalid key algorithm)

$ rec_control add-ta foo.example 1822 1 1 ...
Added Trust Anchor for foo.example with data 1822 1 1 ...

S dig test.foo.example TXT

example. SOA ...
... foo.example. IN NS ns.foo.example.
foo.example. IN DS 303 13 2 ...

managed-key:
foo.example. DNSKEY 257 3 13

foo.example. SOA ...
foo.example. IN DNSKEY 257 3 13 ...
foo.example. IN DNSKEY 257 3 255 ...
...
test.foo.example. IN TXT ...
test.foo.example. IN RRSIG TXT 13 3 ...
info: resolving foo.example. DNSKEY IN
info: trust point has unsupported algorithm at foo.example. DNSKEY IN
... info: trust point was revoked foo.example. DNSKEY IN

zone.c:9600: fatal error!
RUNTIME_CHECK(result == 0)

PANIC: unprotected error in call to Lua AP
(/usr/local/lib/kns_modules/lua/trust anchors.lua:208: invalid RR:
foo.example. 60  DNSKEY 257 3 255 ... (invalid key algorithm)

info: resolving foo.example. DNSKEY IN
info: trust point has unsupported algorithm at foo.example. DNSKEY IN
... info: trust point was revoked foo.example. DNSKEY IN

zone.c:9600: fatal error:
RUNTIME_CHECK(result == 0) failed

result = compute_tag(keyname, &dnskey, mctx, &keytag);
RUNTIME_CHECK(result == ISC_R_SUCCESS);

PANIC: unprotected error in call to Lua AP
(/usr/local/lib/kdns_modules/trust_anchors.lua:208: invalid RR:
foo.example. 60 DNSKEY 257 3 255 ... (invalid key algorithm)
[7] DLV records are ignored

S dig test.foo.example TXT

dlv-anchor: dlv.example. DS 257 3 13

dlv.example. SOA ...
...
foo.example. IN NS ns.foo.example. foo.example. IN DS 303 1 2 ...

foo.example. SOA ...
foo.example. IN DNSKEY 257 3 1 ...
...
test.foo.example. IN TXT ...
test.foo.example. IN RRSIG TXT 1 3 ...

dlv.example. SOA ...
...
foo.example.dlv.example IN DLV 303 1 2 ...
[7] DLV records are ignored

validating test.foo.example/TXT: looking for DLV foo.example.dlv.example
validating test.foo.example/TXT: DLV found with no supported algorithms
validating test.foo.example/TXT: marking as answer (dlvfetched (2))

info: resolving foo.example.dlv.example. DLV IN
info: resolving foo.example. DNSKEY IN
info: Verified that response is INSECURE
[8] dlv trust anchors?

S dig test.foo.example TXT

dlv-anchor: dlv.example. DS 257 3 1

dlv.example. SOA ...
...
foo.example. IN NS ns.foo.example.
foo.example. IN DS 303 13 2 ...

foo.example. SOA ...
foo.example. IN DNSKEY 257 3 13 ...
...
test.foo.example. IN TXT ...
test.foo.example. IN RRSIG TXT 13 3 ...
ignoring trusted key for 'dlv.example.': **algorithm is unsupported**

validating test.foo.example/TXT: keyvalidated: **got broken trust chain**
client @0x7f135c0109c0 10.53.0.1#34299 (test.foo.example): **query failed** (**broken trust chain**)
for test.foo.example/IN/TXT at query.c:6784
fetch completed at resolver.c:5492 for test.foo.example/TXT in 0.018430: **broken trust chain/broken trust chain** [domain:foo.example,referral:0...]

**info:** warning: **unsupported algorithm** for trust anchor dlv.example. DNSKEY IN
warning: trust anchor dlv.example. has no supported algorithms, the anchor is ignored (check if you need to upgrade unbound and openssl)

...  
**info:** NSEC3s for the referral proved no DS.
**info:** Verified that response is **INSECURE**
[8] dlv trust anchors?

- **BIND 9**
  - `dnssec-lookaside "foo.example." \ trust-anchor "dlv.example";`
  - "foo.example" is an explicit trust point
  - Unsupported algorithm is ignored, trust point is empty

- **Unbound**
  - "The DLV configured is used as a root trusted DLV, this means that it is a lookaside for the root."
# Test results

<table>
<thead>
<tr>
<th>Unsupported algorithm</th>
<th>BIND 9</th>
<th>Knot</th>
<th>PowerDNS</th>
<th>Unbound</th>
<th>OpenDNSSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] Key creation should fail</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>-</td>
<td>V*</td>
</tr>
<tr>
<td>[2] Signing should fail</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>-</td>
<td>V</td>
</tr>
<tr>
<td>[3] Publication is allowed</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>-</td>
<td>V</td>
</tr>
<tr>
<td>[4] Validation status insecure</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>[5] Trust anchors are ignored</td>
<td>V</td>
<td>X**</td>
<td>X***</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>[6] 5011 rollover go to insecure</td>
<td>X*</td>
<td>X**</td>
<td>-</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>[7] DLV records are ignored</td>
<td>V</td>
<td>-</td>
<td>-</td>
<td>V</td>
<td>-</td>
</tr>
<tr>
<td>[8] DLV anchors?</td>
<td>?</td>
<td>-</td>
<td>-</td>
<td>?</td>
<td>-</td>
</tr>
</tbody>
</table>

* **BIND 9**: Fixed in 9.11.5-P4, 9.12.3-P4, 9.13.6

** **Knot Resolver**: Fixed in 4.0.0

*** **PowerDNS Recursor**: Issue reported

** *OpenDNSSEC**: Issue reported
Why this story?

- Improve BIND 9 tests
- Improve the code
- Learn about weird cases
- Show boring results
- Share thoughts & experiences
# Implementation status

<table>
<thead>
<tr>
<th></th>
<th>BIND 9</th>
<th>Knot</th>
<th>PowerDNS</th>
<th>Unbound</th>
<th>OpenDNSSEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>RSAMD5</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td><strong>DSA</strong>*</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>RSASHA1</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td><strong>DSA-NSEC3-SHA1</strong></td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>RSASHA1-NSEC3-SHA1</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>RSASHA256</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>RSASHA512</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td><strong>ECC-GOST</strong>*</td>
<td>NO</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>ECDSAP256SHA256</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>ECDSAP384SHA384</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>ED25519</td>
<td>NO</td>
<td>YES</td>
<td>YES (4.2)</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>ED448</td>
<td>NO</td>
<td>NO</td>
<td>YES (4.2)</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
Recap

- Unsupported algorithms handling
  - Overall quite good
  - A bit of quirkiness
  - Hard to exploit crashes
- Some undefined cases
  - Clarifications needed?
- Implementation guidelines change
  - Bye bye DSA, GOST
Thank you for your attention

- Plans for BIND 9.15
  - DNSSEC Made Easy
    - Sane algorithms
    - Fully automated
    - Offline KSK
    - CSK support
    - ...
  - Come talk to me :)
    - matthijs@isc.org