The Curious Case of the Crippling DS record

Public Safety Notice

Roy Arends

DNS-OARC
8-9 March 2018

ICANN

Overview

- Recent validation failures during KSK rollovers
- High level key rollover overview (double DS)
- When double DS fails
- What the standards say
- What do the DNSSEC Operational Practices say? (RFC6781)
- Notes on chains of trust
- What is this trying to prevent
- The missing advice

Recent validation failures during KSK rollovers

- Recently, a few top level domains were temporarily unresolvable.
  - There was no chain of trust
- Manually checking showed that there was a chain of trust:
  - There was a DS record, referring to a KSK, which in turn had signed the DNSKEY set.
- This happened when DS records were added
  - Which was part of a KSK rollover event.
- This failure was ”protocol compliant”
  - But completely unexpected.

High level key rollover overview (double DS)

- Reminder, always keep a chain of trust between parent and child:
  - DNSKEY is present in the child
  - DS record in parent contains a hash over a DNSKEY in child
  - DNSKEY signs the DNSKEY RRset in the child

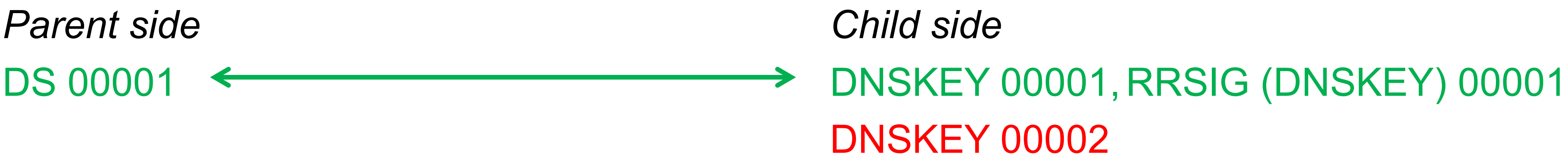
- No sudden moves:
  - Add new DNSKEY in child
  - Add DS record with hash over new DNSKEY in parent
  - Sign DNSKEY RRset with new DNSKEY instead of old DNSKEY

- Clean up:
  - Remove DS record in parent that contains a hash over old DNSKEY
  - Remove old DNSKEY from from child

High level key rollover overview (double DS)

Parent side		*Child side*
DS 00001	↔	DNSKEY 00001, RRSIG (DNSKEY) 00001

High level key rollover overview (double DS)



Parent side

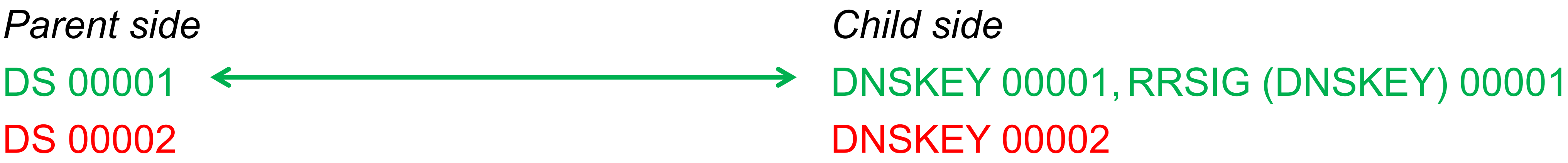
DS 00001

Child side

DNSKEY 00001, RRSIG (DNSKEY) 00001

DNSKEY 00002

High level key rollover overview (double DS)



Parent side

DS 00001

DS 00002

Child side

DNSKEY 00001, RRSIG (DNSKEY) 00001

DNSKEY 00002

High level key rollover overview (double DS)

Parent side		*Child side*
DS 00001		DNSKEY 00001, ~~RRSIG (DNSKEY) 00001~~
DS 00002	↔	DNSKEY 00002, RRSIG (DNSKEY) 00002

High level key rollover overview (double DS)

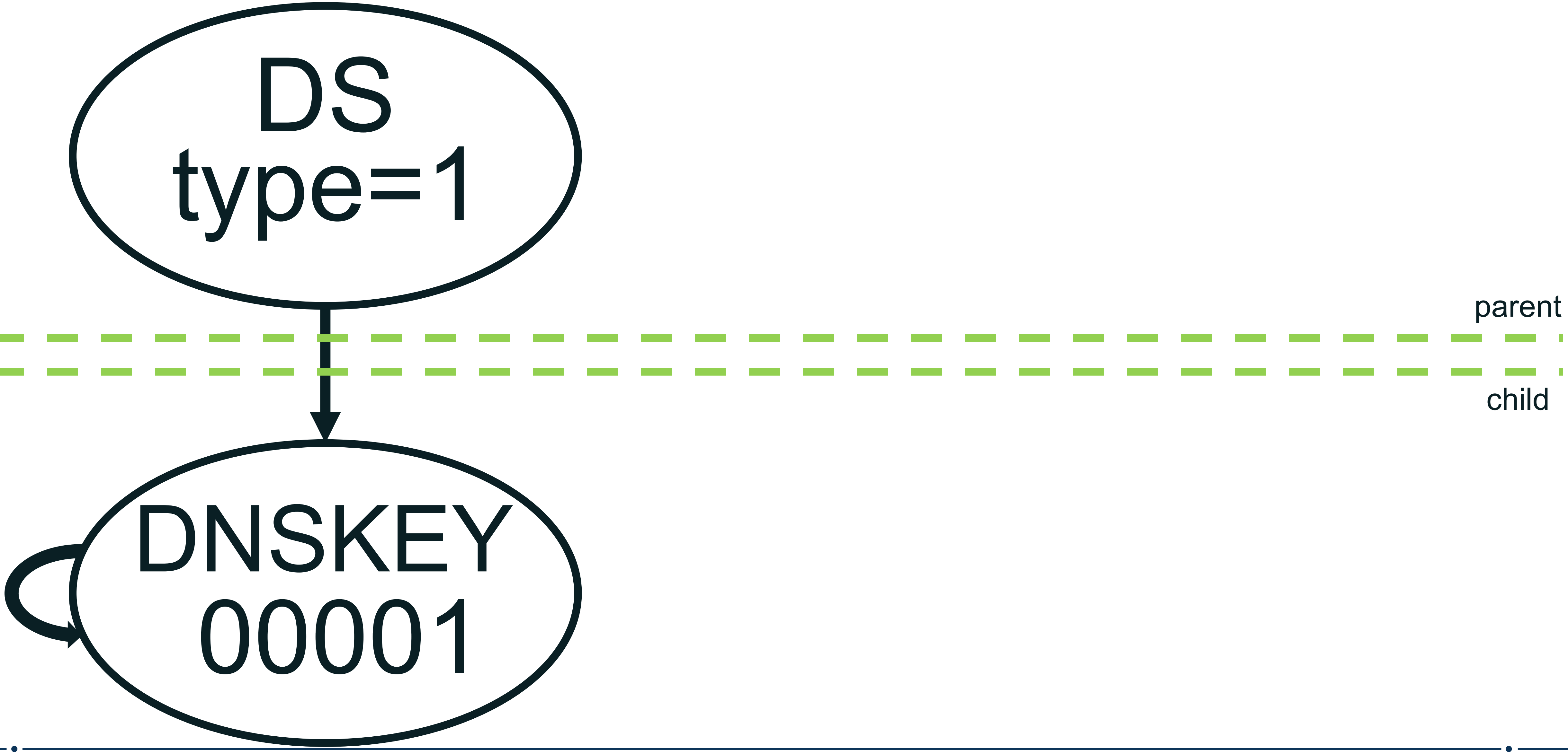
Parent side		*Child side*
~~DS 00001~~		~~DNSKEY 00001~~
DS 00002	↔	DNSKEY 00002, RRSIG (DNSKEY) 00002

High level key rollover overview (double DS)

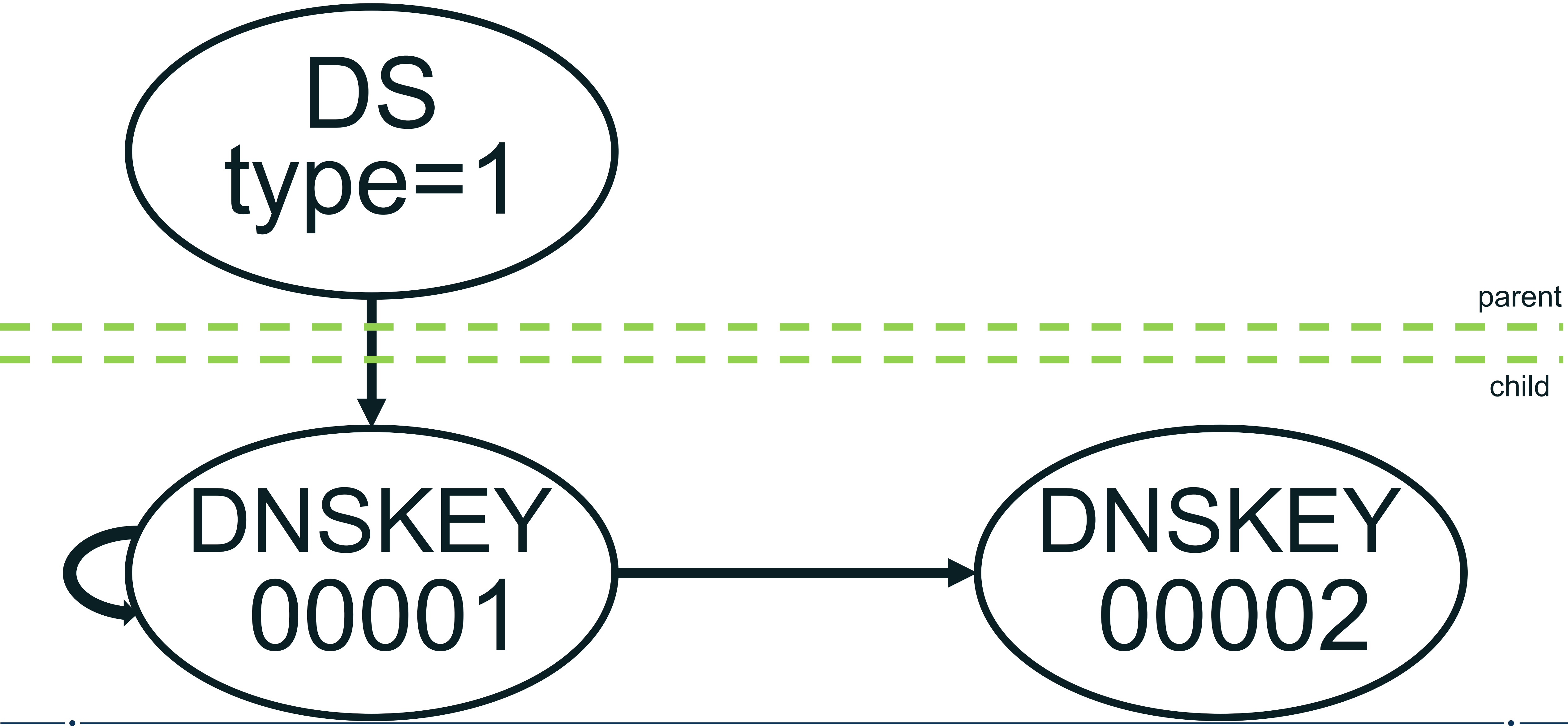
Parent side *Child side*

DS 00002 ↔ DNSKEY 00002, RRSIG (DNSKEY) 00002

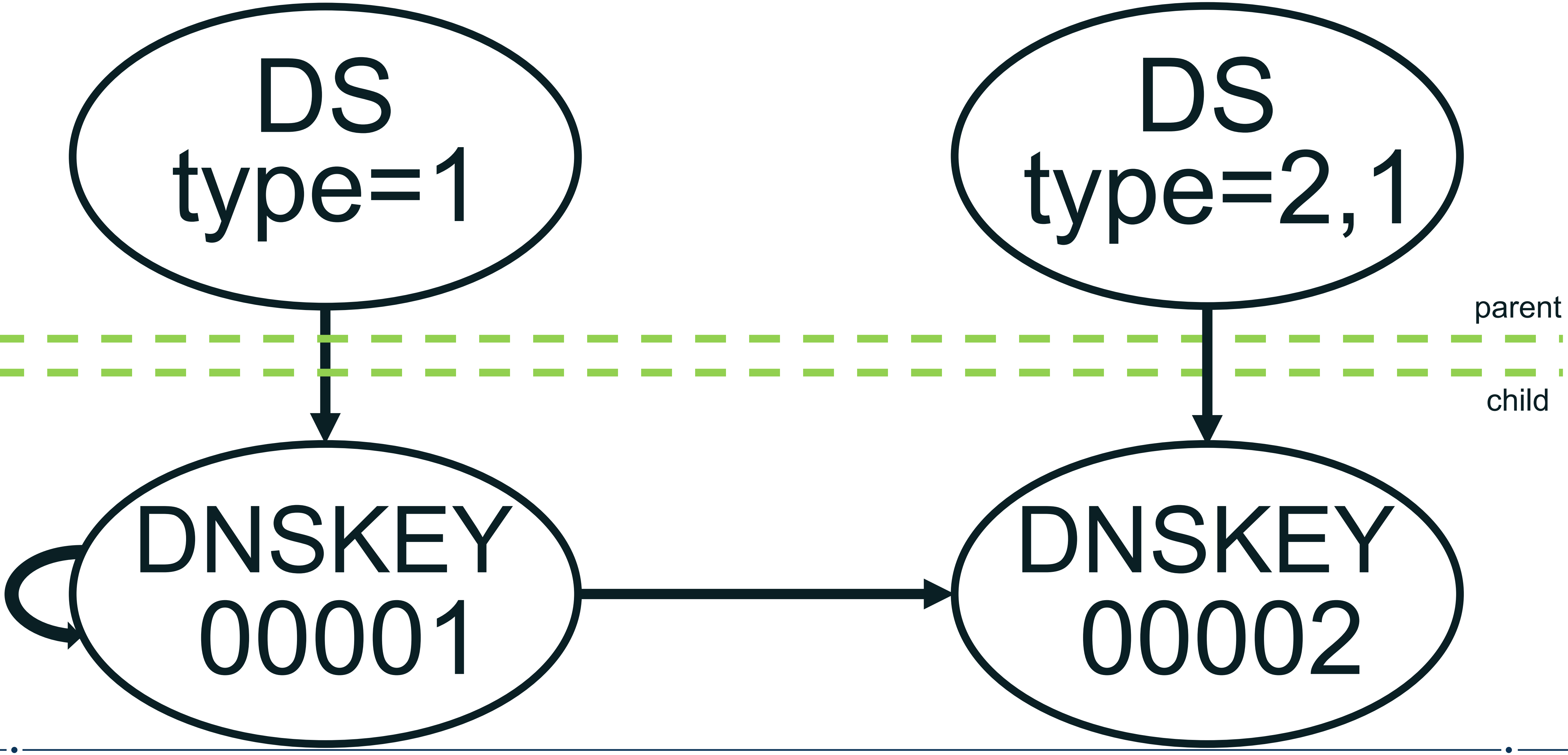
When double DS fails



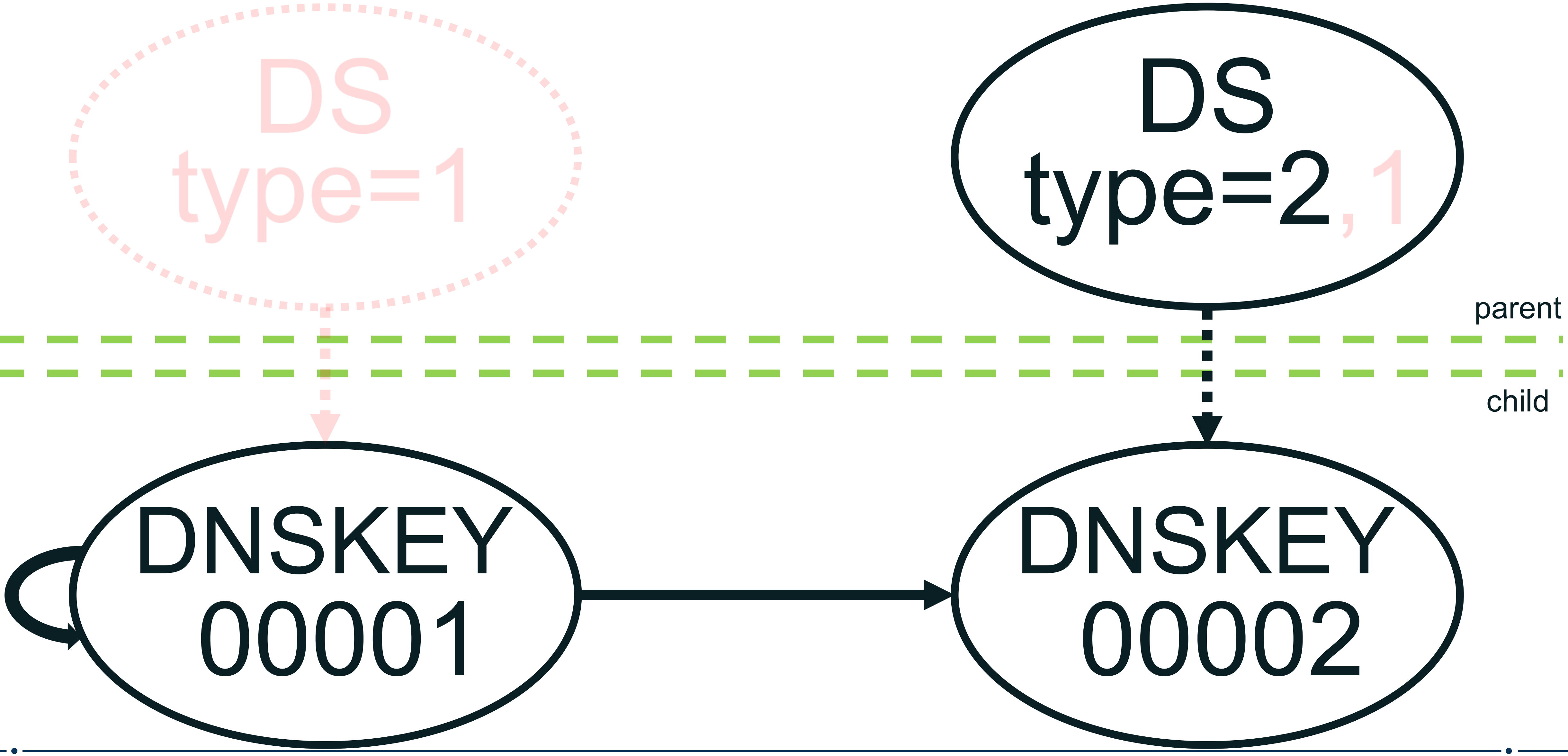
When double DS fails



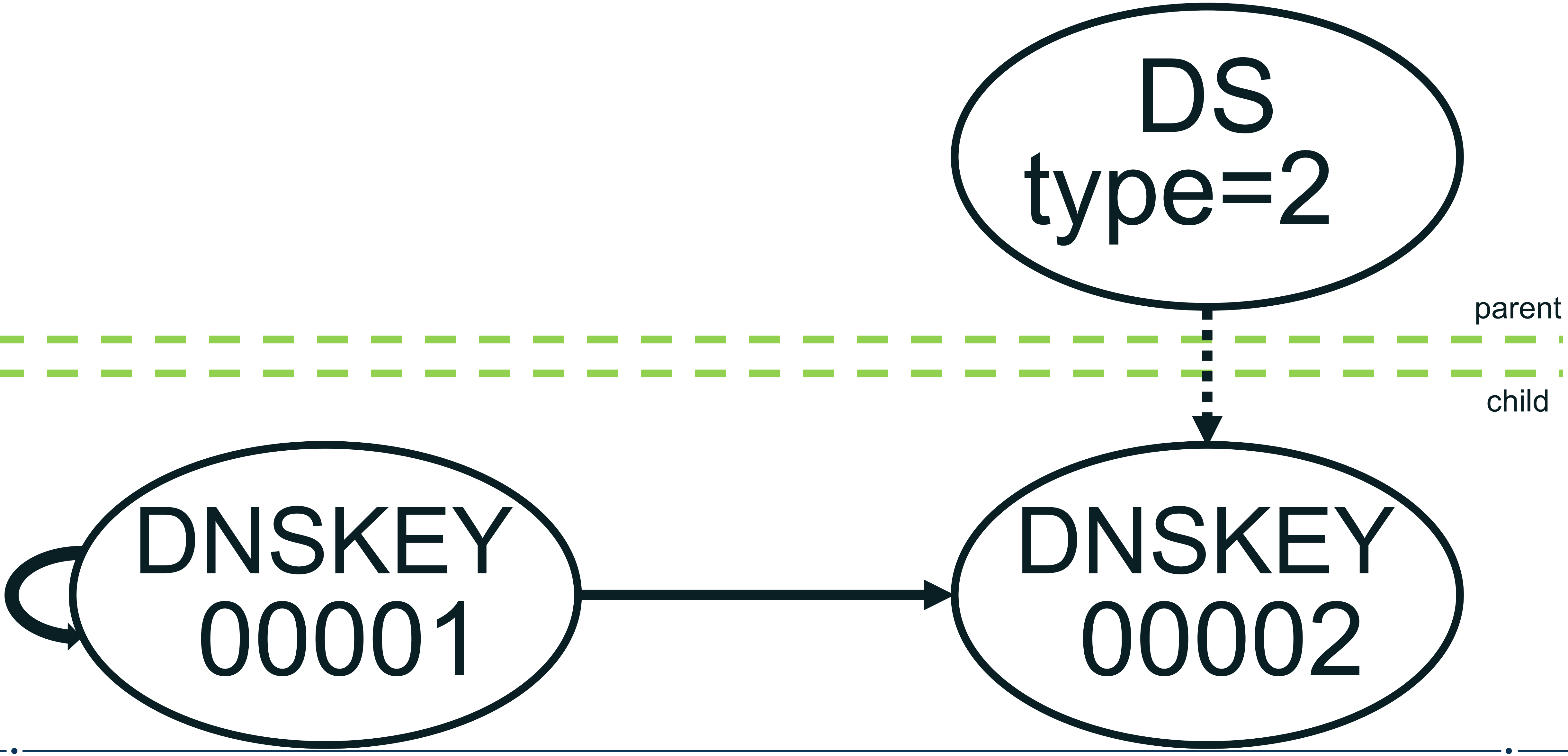
When double DS fails



When double DS fails



When double DS fails



What the standards say

- RFC4509: SHA-256 in DS records

  > 3. Implementation Requirements
  >
  > Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

What the standards say

- RFC4509: SHA-256 in DS records

> 3. Implementation Requirements
>
> Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

UPDATES 4035

What the standards say

- RFC4509: SHA-256 in DS records

3. Implementation Requirements

Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

UPDATES 4035

4. Deployment Considerations

…zone operators should consider deploying both SHA-1 and SHA-256 based DS records. This should be done for every DNSKEY for which DS records are being generated.

What the standards say

- RFC4509: SHA-256 in DS records

3. Implementation Requirements

Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

UPDATES 4035

4. Deployment Considerations

…zone operators should consider deploying both SHA-1 and SHA-256 based DS records. This should be done for every DNSKEY for which DS records ***are being*** generated.

What the standards say

- RFC4509: SHA-256 in DS records

3. Implementation Requirements

Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

UPDATES 4035

4. Deployment Considerations

…zone operators should consider deploying both SHA-1 and SHA-256 based DS records. This should be done for every DNSKEY for which DS records ***are being*** generated. Whether to make use of both digest types and for how long is a policy decision that extends beyond the scope of this document.

What the standards say

- RFC4509: SHA-256 in DS records

3. Implementation Requirements

Validator implementations SHOULD ignore DS RRs containing SHA-1 digests if DS RRs with SHA-256 digests are present in the DS RRset.

UPDATES 4035

4. Deployment Considerations

…zone operators should consider deploying both SHA-1 and SHA-256 based DS records. This should be done for every DNSKEY for which DS records ***are being*** generated. Whether to make use of both digest types and for how long is a policy decision that extends beyond the scope of this document.

OPERATIONAL ADVICE

What do the DNSSEC Operational Practices say? (RFC6781)

- Either use Double Signatures:
  - KSK old and new both sign the DNSKEYset
  - old DS is then replaced by new DS
- Or use Double DS:
  - Both old and new DS are in the parent
  - Old DNSKEY is then replaced by new DNSKEY
- No prescription of the prevention of the failure mode where DS with SHA1 is ignored in the presence of SHA2

What do the DNSSEC Operational Practices say? (RFC6781)

- Either use Double Signatures:
  - KSK old and new both sign the DNSKEYset
  - old DS is then replaced by new DS
- Or use Double DS:
  - Both old and new DS are in the parent
  - Old DNSKEY is then replaced by new DNSKEY
- No prescription of the prevention of the failure mode where DS with SHA1 is ignored in the presence of SHA2

ABSOLUTELY NOTHING

Notes on chains of trust

- Some top level domains have two DS records per DNSKEY in the root zone.
  - Using different Digest Algorithms
- Highest recorded number of DS records for a single TLD since the root was signed:
  - 8 .US DS records, referring to 4 DNSKEYs, using 2 Digest Algorithms
- Current highest number of unique keytags:
  - 3 DS records, all unique keys, same Digest algorithm
- Some stats:
  - 1398 TLDS with chains of trust
  - 184 TLDS with self signed KSKs that do not have DS records.
  - 202 TLDS with KSKs that do have DS records, but are not self-signed.
  - 81 TLDS with DS, but no keys.

What is this trying to prevent

- To prevent a on-path downgrade attack in the following scenario:
  - DS records with SHA1 and SHA256 point to KSK
  - Attacker has a second pre-image for DS SHA1
    - (the second pre-image is a working alternative KSK)
  - Validator accepts DS SHA1 and alternative KSK
    - (DS SHA256 and alternative KSK are no match so will not be considered)

- Multiple variations of this exist, but they all have two things in common:
  - On-path attack
    - (The attacker is a Man-in-the-Middle)
  - The attacker is able to generate a working DNSKEY that has the same digest and keytag as the victim KSK (aka a second pre-image)
    - (This is not the ”shattered” attack where a SHA1 collision was found)

The Missing Advice

- Be consistent is using digest types in DS records
  - Use the same digest type(s) for every KSK.
- Don’t rely on your parent to figure it out for you.
  - Often Garbage-In, Garbage-Out
- Its 2018. You don’t have to use SHA1, you can safely use SHA256.
- Do not roll the KSK and the DS digest type at the same type
  - Either roll the KSK OR roll the DS digest type
- If there is a DNSSEC Best Current Practises 3, this should be added.

Engage with ICANN

Thank You and Questions

Visit us at **icann.org**
Email: email@icann.org

- @icann
- facebook.com/icannorg
- youtube.com/icannnews
- flickr.com/icann
- linkedin/company/icann
- slideshare/icannpresentations
- soundcloud/icann