DNSSec Implementation:

On a high level, the resolution using the DNSSEC resolver returns one of the 3 statuses - “SUPPORTED”, “NOT_SUPPORTED”, “VERIFICATION_FAILED” for a domain when querying for one of the 3 DNS query types (“A”, “MX”, “NS”)

The resolver iteratively queries the nameservers for the DNS rrsets based on the IP using the following steps:

- First we perform a DNS query on the root server using the required query type.
- Starting with the root server, we first query the DNSKEY records using the root server IP.

  ```
  self.query('.', dns.rdatatype.DNSKEY, server_ipv4)
  ```
- Next we verify the RRSIG of the DNSKEY RRsets. This helps us validate the public ZSK of the current zone.

  ```
  self.verify_dnskey(parent_zone_dnskey_response)
  ```
- Next we establish the chain of trust for the root server:
  - To do so, we first validate the RRSIG of the DS records and verify the root zone by hashing its public Key-signing Key and comparing it with the DS record parent server, which in the case of root doesn’t exist. We hard code these DS record hashes (root anchors) provided on the ICANN website
- Repeat the above steps assigning the child zone to be the next nameserver's parent zone until we get a response in the answer section - we can break out of this loop and verify the authoritative nameserver’s query RRset (eg: A record RRSIG) and also finally verify the zone (helps validate the current zone’s public KSK).

  ```
  self.verify_zone(child_zone_dnskey_response, parent_zone_query_response)
  ```
- If any of the above verification fails, we return a `DNSSEC_STATUS_CODES.VERIFICATION_FAILED` status code stating DNSSec verification failed.
- A `DNSSEC_STATUS_CODES.NOT_SUPPORTED` is returned when no DS record is found in the answer section of a DNS query for the domain.
- A successful DNSSec resolution would return a `DNSSEC_STATUS_CODES.SUPPORTED` status code.