DNSSec Implementation Explanation

I have implemented DNSSec using the help of the following methods –

1) **my_dnssec** – This is the primary method which gets called by the main method and the starting point of the DNSSec implementation. This method does 2 main things – First is the DNS resolution for a particular domain name and second is the DNSSec verification for the different records that gets returned along the way during DNS resolution. It takes the domain name as argument and returns different results according to the following scenarios
    a. If DNSSec is supported for that particular domain, it returns the IP address of that domain name.
    b. If DNSSec is not supported for that particular domain, it returns "DNSSEC not supported" as the output.
    c. If the DNSSec is supported for that particular domain and either of KSK verification or DNSKEY verification fails it returns "DNSSec verification failed" as the output.

    At first it verifies the Root, by comparing the DNSKEY records of the root to the public root signing keys which I obtained from the internet. Once that verification is complete, the program moves forward, otherwise the program exits with the output as "DNSSec verification failed".

2) **recursive_query_resolver** – This method gets called by the my_dnssec method to help the latter make the DNS queries for getting the responses for different domains. This method is responsible for analyzing the responses and parsing them to retrieve the DS record, RRSET, RRSIG and DNSKEY records for different domains. It also helps in tackling different edge cases like SOA records and Name Server IP resolution if the DNS query doesn't immediately return the IP for a domain. This method takes 5 arguments viz. searchDomain, queryType, targetServer, depth, maxDepth. It ultimately returns the IP address for the domain name in question to my_dnssec. If, during fetching the DS records or the DNSKEY records, those records are not returned in the response, then the program exits saying ""DNSSEC not supported" as the output.

3) **fetch_dnskey** – This method gets called by the recursive_query_resolver method for fetching the DNSKEY records for a particular domain. It takes the domain name and the targetServer as the arguments and makes a DNS query to fetch the DNSKEY records for that supplied domain name form the supplied targetServer. It ultimately returns RRSIG, ZSK, KSK & RRSET records corresponding to the domain in question as the output.

4) **verify_KSK** – This method gets called by the recursive_query_resolver method for verifying the KSK value for a particular domain by comparing its hashed value with the DS records for that domain which we retrieve from the parent domain. It takes the domain name, KSK and DS records as argument and ultimately returns True or False depending on whether the comparison was successful or not. If it returns False the program exits saying "DNSSec verification failed" as the output.

5) **verify_DNSKEY** - This method gets called by the recursive_query_resolver method for verifying the DNSKEY values for a particular domain by calling the dns.dnssec.validate() method from the dnspython library. It takes the domain name, RRSET and RRSIG records as argument and ultimately if the validation was successful the program continues. Otherwise, the program exists by giving “DNSSec verification failed” as the output.