DNS SEC Implementation

The function dnssec represents the dns sec implementation in the code. The code follows a flow of steps which are explained below with an example of **www.paypal.com**

1. Request the root server for DNSKEY using dns.message.make_query
   We get 4 things,
   1. DNSKEY of root (.)
   2. RRSIG of DNSKEY
   3. DS record of TLD zone (.com)
   4. RRSIG for the DS record signed with private ZSK of root

2. We perform 3 kinds of validation
   1. Decrypts RRSIG using Public KSK of root. This is done by the ***dns.dnssec.validate*** function
   2. We validate root server's public KSK with trust anchor. This is done by ***rootkey_verify*** function
   3. Decrypt DS record of next TLD (.com) using Public ZSK of the root
      This is done by the ***verify_record*** function.

3. We repeat this process in the next iteration by Requesting the .com server for its DNSKEYs
   We receive 4 RRs:
   1. DNSKEY of TLD (.com)
   2. RRSIG of the public and private KSK of TLD (.com)
   3. DS record of paypal.com's Authoritative Server
   4. RRSIG of the DS record signed with private ZSK of TLD (.com)

4. We perform the same validation as before

   1. We Decrypt RRSIG using Public KSK of TLD (.com)
   2. We Validate TLD server's public KSK with public KSK stored in the previous iteration
   3. We Decrypt DS record of Authoritative Name server (paypal.com) using Public ZSK of TLD (.com)

5. Next Step is the last iteration where
   We send a request paypal.com for DNSKEY
   We get 4 RRs:
   1. DNSKEY of Authoritative Name Server (paypal.com)
   2. RRSIG of the public and private KSK of paypal.com
   3. ' A ' record of paypal's Web Server
   4. RRSIG of the A record signed with private ZSK of Authoritative Name Server (paypal.com)

6. We perform 3 validations:
   1. We Decrypt RRSIG with Public KSK of Paypal.com
   2. We Validate paypal.com's pub KSK against pub KSK received in the previous request
   3. We Decrypt the A record using Public ZSK of paypal.com Authoritative Name Server.