DNS TKEY & SIG(0)

draft-eastlake-dnsop-rfc2930bis-tkey-01
draft-eastlake-dnsop-rfc2931bis-sigzero-00

Donald Eastlake 3rd d3e3e3@gmail.com

Mark Andrews marka@isc.org

Two Types of DNS Security

- DNSSEC Data Security provides authentication of data RRs or authenticated denial of their existence cryptographically linked to the zone owner.
- DNS Transaction Security provides authentication of DNS requests and DNS transactions (concatenation of request and response) cryptographically linked to the resolver and server or to the authority being invoked by the request.
- This presentation is about Transaction Security.

Historical RR Type Note

- Both DNS transaction security and DNSSEC data security originally used the
  - SIG (type = 24) and
  - KEY (type = 25) RRs [RFC2535].
- DNSSEC was changed to use the
  - RRSIG (type = 46) and
  - DNSKEY (type = 48) RRs [RFC4034].
- The corresponding RRs have the same field structure as each other. Transaction security continues to use the SIG and KEY RRs.

TSIG and TKEY

- TSIG (Transaction Signature, RFC 8945) is a meta RR providing efficient DNS request and transaction authentication based on a keyed hash algorithm and shared secret key.
- TSIG provides no way to set up such keys or to agree on the keyed hash algorithm other than configuration.
- The TKEY (Transaction Key, RFC 2930) meta RR provides a mechanism for a resolver and server to agree on a secret key and keyed hash algorithm for that key.

More on RFC 2930 TKEY

- TKEY sent by resolver in the additional information field of a query for type TKEY. Response TKEY is in the answer part of the reply.
- TKEY RR has a “mode” field for establishing or deleting shared secret keys. RFC 2930 specifies five modes:
  - 1: Server Assigned
  - **2: Diffie-Hellman exchange**
  - **3: GSS-API negotiation**
  - 4: Resolver Assigned
  - 5: Key Deletion

More on RFC 2930 TKEY

- (GSS-API mode TKEY RRs just provide a tunnel for sending GSS-API tokens which have their own authentication and encryption. (see RFC 3645))
- For other modes:
  - Messages must be authenticated with either TSIG or SIG(0).
  - Names, which must be locally unique across the resolver and server, designate a secret key and algorithm for TKEY and TSIG.
  - Keys have inception and expiration times.

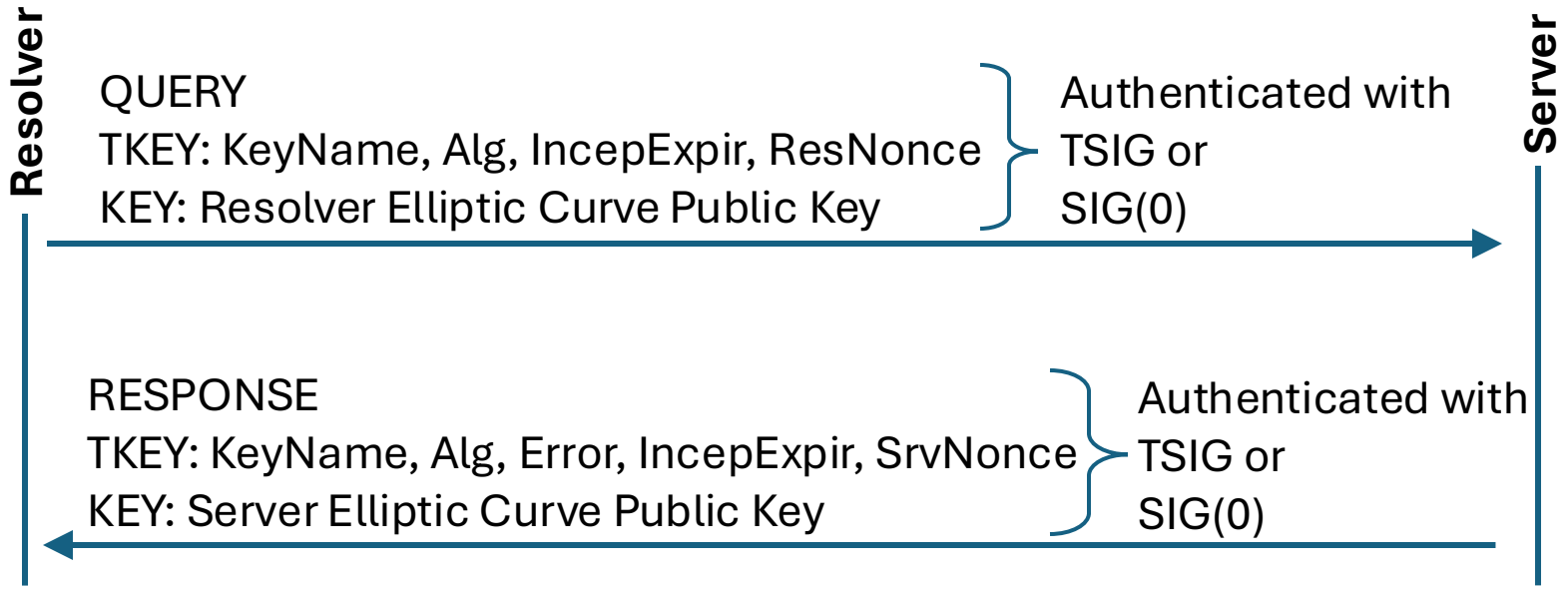
Problems with TKEY

- “Diffie-Hellman exchanged keying” uses XOR for mixing and the deprecated MD5 [RFC 6151] algorithm for hashing and was written assuming RSA public/private keys.
- Not widely supported.
- Although TSIG RFC 8945 has provisions for multi-hop propagation, TKEY RFC 2930 has no such explicit provisions.

Changes in rfc2930bis

- Mode changes:
  - Deprecates “Diffie-Hellman exchanged keying” mode which uses MD5 hashing and XOR mixing.
  - Adds “ECDH exchanged keying” (Elliptic Curve Diffie-Hellman) mode which uses SHA-256 [RFC6234] hashing, and HKDF [RFC5869] for mixing.
  - Adds a “Ping” mode to test basic TKEY plumbing and to check the synchronization of the resolver and server clocks.
  - Adds a reserved “Documentation” mode number for use in examples.

TKEY ECDH Flow



- Input Keying Material = shared secret from ECDH with resolver and server keys
- salt = "IETF-TKEY-ECDH”
- info = Res-IP-address | Srv-IP-address | ResNonce | SrvNonce

- Output KM = HKDF-Expand(HMAC-SHA256(salt, IKM), info, L)

SIG(0) RFC 2931

- Reuses the SIG RR but with a “Type covered” field of zero.
- SIG(0) signs requests and transactions using a public key for which the authenticator has the private key.
  - Can authenticate general requests and replies if public key associated with querier/server.
  - Can authorize UPDATE or the like if public key associated with zone or other authority.
  - Public keys are stored in the DNS with the KEY RR.

Problems with SIG(0)

- The SIG RR was not designed for this purpose. Many SIG fields are not used.
- Has no Error field.
- Has no Original ID field so multi-hop authentication not obviously supported.
  - BIND supports multi-hop SIG(0) by forwarding with TCP with the same ID and maintaining a separate ID space per TCP connection.
- Has one byte Algorithm field rather than a domain name Algorithm field.
- Not extensible. (TSIG and TKEY have an “Other Data” field.)

Changes in rfc2931bis

- Remove statement that TCP support of SIG(0) is optional.
- Change some implementation requirements to reduce the variability in SIG(0) RRs.
- Add section on considerations for forwarding servers.
- Update to current draft format standards and update references.

Some Questions

- Should “SIG(0)” be replaced with a new RR (PKTSIG (Public Key TSIG)?) better designed for public key authentication of DNS requests and transactions?
- Should it be convenient to agree on TSIG secret keying and algorithm between a querier and a responder multi-hop using TKEY or something else? I.E., with forwarder(s) in between?
- Should the TKEY “Resolver Assigned” and “Server Assigned” keying modes of TKEY be deprecated?
- Should multiple simultaneous TSIGs and/or public key based DNS transaction signatures be allowed?

END

Donald Eastlake 3rd d3e3e3@gmail.com

Mark Andrews marka@isc.org