LURK Interim draft-erb-lurk-rsalg-01

Samuel Erb, Rich Salz

Akamai Technologies

Significant Updates

- Setup request/response
  - Server can request at any time
  - Response contains:
    - List of of certificates with “purpose” tag
    - supported signature & hash algorithms
    - ”state” tag
  - KeyOwner sends back consistent ”state” tag in each response, Server watches for changes
- Session ticket key request
  - Maintains the private key as an input to the session ticket key KDF

Setup

Request:

```
struct {
    lurk_msg_header  header;
    uint64           id;
} setup_request;
```

Response:

```
struct {
    uint8  purpose<32>;
    opaque ASN.1Cert<1..2^24-1>;
} certificate;
struct {
    lurk_msg_header  header;
    uint64           id;
    SignatureAndHashAlgorithm
                     supported_signature_algorithms<2..2^16-2>;
    certificate      certificate_list<0..2^24-1>;
    uint8            state<32>;
} setup_response;
```

Requests

LURK request:

```
enum {
    rsalg(0), server_kx(1), (255)
} ReqType
struct {
    lurk_msg_header  header;
    uint64           id;
    ReqType          op_type;
    uint8            cert<32>;
    uint16           client_version;
    uint16           server_version;
    uint8            client_random<32>;
    uint8            server_random<32>;
    SignatureAndHashAlgorithm sig_hash_alg;
    PRFHashAlgorithm          prf_hash_alg;
    opaque           data<0..2^16-1>;
} lurk_request;
```

Session ticket key request:

```
struct {
    lurk_msg_header  header;
    uint64           id;
    uint8            cert<32>;
    uint8            server_salt<48>;
} lurk_session_ticket_request;
```

Response

Common Response:

```
enum {
    success(0), invalidParameters(1), certUnavailable(2),
    permissionDenied(3), insufficentResources(4), (255)
} ResponseStatus
struct {
    lurk_msg_header  header;
    ResponseStatus   status;
    uint64           id;
    uint8            state<32>;
    opaque           data<0..2^16-1>;
} lurk_response;
```

Open issues

- The KeyOwner could choose the TLS server random. This makes RSALG even less likely to be useful as an oracle, but has turned out to be difficult to integrate into existing TLS/SSL libraries.
- Should the lurk_request and lurk_response messages be padded out to eight-byte alignment?
- Should we use variant for the different request/response payloads?

We are still looking for feedback!