TS 2021.010 | Paymail: Mutual Party Authentication

Subject	Mutual Party Authentication
Date	06 Apr 2022
Approver	@ Andy Mee (Unlicensed)
Contributors	@ Miguel Duarte @ Curtis Ellis @ Austin (MrZ) Rappaport @ darrkellen
Outcome	To include the high level flow, but not the message enveloping or signature scheme, in the core standard
Rationales for the decision	Mutual Party Authentication is a valuable feature of the Paymail standard. Including it in the base allows extension authors to defer entirely to the base standard and not have to consider MPA with respect to their specific extension; it is an out-the-box feature. Separating the specifics of message enveloping and signature scheme into extensions allows for these to be selected and plugged in to the core in a way that is appropriate for individual use-cases.
Comments	

Background

The following considerations were made when designing the authorisation requirements structure:

- Mutual Party Authentication is core to the Paymail standard. It is considered desirable to define an authenticating host's requirements once, rather than encumber every extension capability with solving for their own authentication concerns. Extension authors who mandate or recommend the use of Mutual Party Authentication need only refer back to this base Paymail standard to address the need.
- This standard defines the domain owner as the trust anchor in the Paymail system. The domain owner carries responsibility for ensuring the trustworthiness of any Paymail host acting on their behalf, regardless of the use of Authoritative of Delegated domain configuration. It is therefore appropriate to rely on the client's representative Paymail host for client-authoritative declarations.
- Cryptographic agility has been a source of vulnerabilities. The ability for clients and servers to negotiate supported cryptographic primitives such as cipher suite, algorithm, or key length, has led to downgrade attacks, where clients deliberately negotiate servers down to the weakest combination offered. Paymail Mutual Party Authentication allows only for an authenticating host to declare a security construction that it will accept, and does not offer any negotiation on these points. Downgrade attacks have been seen on everything from SSL/TLS to JWTs with the "none" algorithm. By not allowing negotiation, the authenticating host is protected from this scenario.
- Capabilities are free to define message schema, data types, and wire encodings as appropriate for the problem that they solve. It is therefore not desirable for this standard to mandate a single signature scheme/message envelope pair for all endpoints. For example, a message envelope that relied on the JSON Envelope Specification to carry the Paymail alias, message signature and public key may be considered inappropriate for capabilities that define non-JSON messages, such as those encoded via CBOR, ProtoBufs, or Bitcoin Binary Serialization. Therefore, each authenticated capability may specify a different scheme/envelope pair (discussed below).
- Resistance to information disclosure. By designing the client-authoritative host's key-alias endpoint to only confirm a correct pairing of Paymail alias and signing key, casual querying of the endpoint cannot reveal information regarding the validity of an alias, a key, or the correlation between them. A valid alias with an invalid key is indistinguishable from an invalid alias. Given the key space of modern cryptographic schemes, it is infeasible to brute-force this endpoint to reveal a list of valid aliases and known public signing keys from the client-authoritative host.