Paymail WG Space: Amending the 2019 Specification

Subject	Amending the 2019 Specification
Date	28 May 2021
Approver	@ Andy Mee (Unlicensed) @ Miguel Duarte
Contributors	@ Andy Mee (Unlicensed) @ Miguel Duarte
Outcome	The 2019 BSV Alias specification will not be put forward for standardisation as-is. The following changes were agreed: • Section 1 (BRFC numbering) becomes its own separate specification • Section 2 (Service Discovery) becomes the core standard • Sections 3 and 5 (PKI, Verify PK Owner) are moved to a Paymail extension • Section 4 (payment addressing) is moved to a Paymail extension • The host entity is introduced as an addressable entity in its own right (just *domain.tld* rather than *alias@domain.tld*) • Message signing, currently described in section 4.1 of the 2019 specification, is promoted to a core concept • Formal separation of mainnet, testnet, STN, regtest aliases to be introduced
Rationales for the decision	The BSV Alias core, as scoped above, presents a secure base to build application and business specific flows on top of. It codifies a security and trust model and peer-to-peer message flow that is agnostic to any specific flow. BRFC numbering was included in the original document only because it was not publicly available anywhere else, however it has never been part of the BSV Alias specification. Sections 3 and 5 together form a PKI infrastructure, which whilst useful, is not core to the concepts that remain in the base. Alternative PKI infrastructure, for example alternative public key schemes, may be of use and should not be “second class” to the PKI that debuted in 2019. Section 4 was a direct response to the Genesis upgrade and the deprecation of P2SH. Whilst it serves this purpose it is already perceived as legacy and will likely be deprecated soon, as four-phase payments (purchase order invoice payment receipt) become more commonplace, and as regulatory factors such as the Travel Rule require additional steps to be taken. Message signing is promoted from section 4 to a core concept so that it can be addressed at a core level, rather than a flow-by-flow level. This promotes consistency across the entire API surface. In addition, message signing will be revisited, following the discovery of a low-impact, low-severity security vulnerability in the message signing process discovered as part of a third-party security audit of the 2019 specification commissioned by nChain. The introduction of the host as an addressable entity is in recognition of the different approaches to key custody and management that may be taken by implementations. In some scenarios the service provider manages keys and may be authorised to sign on behalf of aliased individuals. This may be directly by using a key whose beneficial owner is the service provider’s customer, or may be by using a key connected with the identity of the service provider. This distinction is not made in the 2019 specification, so in some cases the audit trail of signed messages may be ambiguous as to the identity of the signatory. Introducing a formal separation between mainnet and other networks has been an often-requested feature. Handling this at the core allows for fewer changes, branches, switches and conditionals through the codebase of implementations, leading to fewer defects.
Comments	

Background

The BSV Alias specification was produced in 2019 by nChain and Money Button in collaboration with a number of industry participants. At the time the imminent Genesis upgrade left P2SH users without a clear alternative mechanism for exchanging complex payment destinations. This concern was emphasised too strongly against the other concerns addressed by the 2019 specification, including enabling secure peer-to-peer application flows. This decision clarifies the role of the core BSV Alias specification and the elements of it that should always have been extensions.

Following two years of implementation and adoption, and the publication of a number of interesting third party extensions to the core, the role and purpose of the core as standardised by this working group has become clearer, and the modifications contained within this standardisation effort reflect this.

Risks (where applicable)

The 2019 specification introduced an extension mechanism whereby feature negotiation was performed via BRFC ID, whilst core-to-2019 sections were named explicitly. In moving previously core sections to extensions, an exemption to the BRFC rule must be carried forward in order to prevent breaking changes to existing implementations.

Fixing the low-priority, low-severity security vulnerability in message signing is a breaking change. A number of implementation teams have already confirmed that they either do not use message signing and verification, or that they are prepared to make changes to fix the vulnerability. Informally, nobody has objected to the proposal to modify the signing process to fix the known vulnerability.

Options considered

On scope:

	Option 1:	**Option 2:**
Description	Leave the 2019 scope as is	Redefine the scope as part of standardisation
Pros and cons	➕ Less work writing the standard ➖ Leaving 2019 sections identified as extensions not core as part of the core spec promotes them as the “blessed” solution to the problems they solve and discourages innovation in the problem space they occupy	➕ Clearer definition of the role of BSV Alias and the purpose of extensions ➖ Minor ugliness in carried exemption to named extensions during feature negotiation
Reasons for selecting or discarding the option	Section 4 is already planned for obsolescence	Smaller core plus extensions allows for greater innovation in areas of industry need

Options considered

On fixing the signing process:

	Option 1:	**Option 2:**
Description	Leave the 2019 scope as is	Adjust the message signing process
Pros and cons	➕ No risk of breaking existing implementations ➖ A known vulnerability exists	➕ Fixes a known vulnerability Existing implementations have responded positively to adopting any breaking change. ➖ Could break implementations that we are not aware of
Reasons for selecting or discarding the option	Leaving security vulnerabilities, however minor, in wallet software is borderline negligent. At the very least this impacts the perception of the security strengths of the standard negatively.	Parallel efforts have produced the JSON Envelope specification for signing messages, as seen in MAPI, whilst another option exists using HTTP headers to sign payloads. These may now be explored for the most suitable option to standardise on. No implementations have objected to adopting breaking changes.

Options considered

On supporting mainnet and test network separation at the core:

	Option 1:	**Option 2:**
Description	Do nothing	Support separation

Pros and cons	➕ None perceived ➖ This has been requested on several occasions; failing to respond to an industry need diminishes the value of the standard	➕ Standardising support for network separation allows implementations to test interoperability safely without risking mainnet funds ➖ None perceived
Reasons for selecting or discarding the option	There is no upside to this option	Separation can be worked into the standard in a purely additive way, without breaking any implementation today, whilst addressing a frequently-identified industry need.

Supporting information (optional)

N/A.