CFRG
Internet-Draft
Intended status: Informational
Expires: 7 March 2026

V. Kalos
MATTR
G. Bernstein
Grotto Networking

3 September 2025

BBS per Verifier Linkability
draft-irtf-cfrg-bbs-per-verifier-linkability-02

Abstract

The BBS Signatures scheme defined in [I-D.irtf-cfrg-bbs-signatures], describes a multi-message digital signature, that supports selectively disclosing the messages through unlinkable presentations, built using zero-knowledge proofs. Each BBS proof reveals no information other than the signed messages that the Prover chooses to disclose in that specific instance. As such, the Verifier (i.e., the recipient) of the BBS proof, may not be able to track those presentations over time. Although in many applications this is desirable, there are use cases that require the Verifier be able to track the BBS proofs they receive from the same Prover. Examples include monitoring the use of access credentials for abnormal activity, assertion of pseudonymous identity, monetization, etc.. This document provides a mechanism for binding prover secret material for pseudonym creation to a BBS signature and shows how to use this bound information for the creation of context dependent pseudonyms in BBS proofs.

Table of Contents

1. Introduction

The BBS Signature Scheme, originally described in the academic work by Dan Boneh, Xavier Boyen, and Hovav Shacham [BBS04], is a signature scheme able to sign multiple messages at once, allowing for selectively disclosing those message while not revealing the signature it self. It does so by creating unlinkable, zero-knowledge proofs-of-knowledge of a signature value on (among other) the disclosed set of messages. More specifically, the BBS Prover, will create a BBS proof that if validated by the Verifier, guarantees that the prover knows a BBS signature on the disclosed messages, guaranteeing the revealed messages authenticity and integrity.

The BBS Proof is by design unlinkable, meaning that given two different BBS proofs, there is no way to tell if they originated from the same BBS signature. This means that if a Prover does not reveal any other identifying information (for example if they are using proxies to hide their IP address etc.), the Verifier of the proof will not be able "track" or "correlate" the different proof presentations or the Provers activity via cryptographic artifacts. This helps enhance user privacy in applications where the Verifier only needs to know that the Prover is in possession of a valid BBS signature over a list of disclosed messages.

In some applications, however, a Verifier needs to track the presentations made by the Prover over time, as to provide security monitoring, monetization services, configuration persistance etc.. In other applications a Prover may wish to assert a pseudonymous identity associated with a signature from an issuer. To promote privacy, the Prover should not be forced to reveal or be bound to a unique identifier that would remain constant across proof presentations to different Verifiers and which could be used to link a Provers interactions with different Verifiers.

The goal of this document is to provide a way for a Verifier to track the proof presentations that are intended for them, while at the same time not allowing the tracking of the Prover’s activities with other Verifiers. This is done through the use of a cryptographic pseudonyms.

A cryptographic pseudonym, or pseudonym for short, as defined by this document, is a value that will be computed from two parts. One part is the pseudonym secret value (nym_secret) which is known only to the prover and a context value (context_id) that must be known by both prover and verifier. The pseudonym value is computed in such a way that it is computationally infeasable to link to pseudonyms to the same pseudonym secret, i.e., holder for two different context values.

1.1. Pseudonyms Bound to BBS Signatures

The BBS signature scheme is based on a three party model of _signer_ (aka issuer), _prover_ (aka user or holder), and _verifier_. A _prover_ obtains a BBS signature from a _signer_ over a list of _messages_ and presents a BBS proof (of signature possession) along with a selectively disclosed subset of the BBS _messages_ to a verifier. Each BBS proof generated is unlinkable to other BBS proofs derived from the same signature and from the BBS signature itself. If the disclosed subset of BBS _messages_ are not linkable then the presentations cannot be linked.

BBS pseudonyms extend the BBS signature scheme to "bind" a "pseudonym secret" to a BBS signature retaining all the properties of the BBS signature scheme: (a) a short signature over multiple messages, (b) selective disclosure of a subset of messages from _prover_ to _verifier_, (c) unlinkable proofs.

In addition BBS pseudonyms provide for:

1. A essentially unique identifier, a pseudonym, bound to a proof of signature whose linkability is under the control of the _prover_ in conjunction with a _verifier_ via the selection of a "context". Such a pseudonym can be used when a _prover_ revisits a _verifier_ to allow a _verifier_ to recognize the prover when they return or for the _prover_ to assert their pseudononous identity when visiting a _verifier_
2. Assurance of per _signer_ uniqueness of the "pseudonym secret", i.e., the _signer_ assures that the pseudonyms that will be guaranteed by the signature have not been used with any other signature issued by the signer (unless a signature is intentionally reissued).
3. The _signer_ cannot track the _prover_ presentations to _verifiers_ based on pseudonym values.
4. Colluding _verifiers_ sharing BBS proofs with pseudonyms cannot link proofs or pseudonyms across "contexts".

To realize the above feature set we embed a two part pseudonym capability into the BBS signature scheme. The pseudonym’s cryptographic value will be computed from a secret part, which we call the _nym_secret_ and a part that is public or at least shared between the _prover_ and one or more _verifiers_. The public part we call the _context_id_. The pseudonym is calculated from these two pieces using discrete exponentiation. This is similar to the computations in [Lys00] and [ABC14]. The pseudonym is presented to the _verifier_ along with a ZKP that the _prover_ knows the _nym_secret_ and used it and the _context_id_ to compute the pseudonym value. A similar proof mechanism was used in [Lys00].

To bind a pseudonym to a BBS signature we have the _signer_ utilzed Blind BBS signatures and essentially sign over a commitment to the _nym_secret_. Hence only a prover that knows the _nym_secret_ can generate a BBS proof from the signature (and also generate the pseudonym proof).

As in [Lys00] we are concerned with the possibility of a dishonest user and hence require that the _nym_secret_ = _prover_nym_ + _signer_nym_entropy_ be the sum of two parts where the _prover_nym_ is a provers secret and only sent to the _signer_ in a binding and hiding commitment. The _signer_nym_entropy_ is "blindly added" in by

the _signer_ during the signing procedure and sent back to the _prover_ along with the signature. Note the order of operations. The _prover_ chooses their (random) _prover_nym_ and commits to it. They then send the commitment along with a ZKP proof that the commitment is correctly calculated. The _signer_ verifies the commitment to the _prover_nym_ then generates the _signer_nym_entropy_ and "blindly adds" it to the _prover_nym_ during the signature process. Note that this can be done since we sign over the commitment and we know the generator for the commitment.

This document will define new BBS Interfaces for use with pseudonyms, however it will not define new ciphersuites. Rather it will re-use the ciphersuites defined in Section 6 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-03.html#name-ciphersuites) of [I-D.irtf-cfrg-bbs-signatures]).

1.2. Cryptographic Pseudonyms: A Short History

The discussion of cryptographic pseudonyms for privacy preservation has a long history, with Chaum’s 1985 popular article “(U+201C)Security without identification: transaction systems to make big brother obsolete” (U+201D) [Chaum85] addressesing many of the features of such systems such as unlinkability and constraints on their use such as one pseudonym per organization and accountability for pseudonym use. Although Chaum’s proposal makes use of different cryptographic primitives than we will use here, one can see similarities in the use of both secret and "public" information being combined to create a cryptographic pseudonym.

Lysyanskaya’s 2000 paper [Lys00] also addresses the unlinkable aspects of pseudonyms but also provides protections against dishonest users. In addition they provide practical contructions similar to those used in our draft based on discrete logarithm and sigma protocol based ZKPs. Finally as part of the ABC4Trust project [ABC14] three flavors of pseudonyms were defined:

1. _Verifiable pseudonyms_ are pseudonyms derived from an underlying secret key.
2. _Certified pseudonyms_ are verifiable pseudonyms derived from a secret key that also underlies an issued credential.
3. _Scope-exclusive pseudonyms_ are verifiable pseudonyms that are guaranteed to be unique per scope string and per secret key.

The BBS based pseudonyms in our draft are aimed primarily at providing the functionality of the pseudonym flavors 2. and 3. above.

1.3. BBS Pseudonym Example Applications

1.3.1. Certifiable Pseudonyms

In this case the _prover_ gets to choose and assert a "pseudonymous identity" bound to a signature (credential) from an _issuer_.

The _prover_ can choose this "pseudonymous identity" through its choice of a _context_id_ that will then be shared with along with the cryptographic pseudonym with a _verifier_. Note that the combination of (_context_id_, _cryptographic_pseudonym_) forms the "psedonymous identity" that is bound to the BBS signature. By changing the _context_id_ the _prover_ can choose a new "pseudonymous identity" however, within the cryptographic limitations of BBS and the pseudonym computaions, no other prover should be able to assert this "pseudonymous identity". This is confirmed by the _verifier_ during BBS pseudonym proof validation and utilizes the _signers_ public key.

The mechanims in this draft permit the _issuer_ to guarantee that the _nym_secret_ is essentially unique to and by the issuer, though not known to the issuer. Further enhancing the "pseudonymous identity".

The _prover_ and no one else without the _nym_secret_ and signature can produce a proof that they "own" the "pseudonymous identity".

1.3.2. Scope Exclusive Pseudonyms

In this case a _verifier_ or group of _verifiers_ needs to know if the same _prover_ is presenting a BBS proof of signature along with some selectively disclosed information (BBS messages) on subsequent presentations.

We assume that no linkable information is contained in the disclosed messages or information that can be obtained from other layers in the application/communications stack.

To do this a _verifier_ requires that the _prover_ use a _context_id_ that the _verifier_ specifies. In this case the _verifier_ can use the _cryptographic pseudonym_ to link subsequent proof presentations from the same _prover_. However, _cryptographic pseudonyms_ from _verifiers_ that specify different _context_ids_ cannot, within the cryptographic assumptions of the pseudonym computation be linked to each other. This provides the _verifier_ with limited linkability to a _prover_ and a _prover_ unlinkability across _verifiers_ using *different* _context_ids_.

1.3.3. Scope Exclusive Pseudonyms with Monitoring

In this case third party monitoring of interactions between _prover_ and _verifier_ is required.

For example (completely ficticious) suppose the signature (credential) certifies that the _prover_ is qualified to purchase and store some type of controlled substance, e.g., a class of potentially hazardous chemicals. To avoid price fixing or leakage of secret chemical formulas the _prover_ purchases these chemicals under a _verifier_ (vendor) specific pseudonym. Which prevents the different vendors from colluding on prices or seeing all the chemicals being purchase by a given prover.

However for public safety, hording prevention, etc... verifiers (vendors) are required to report all purchase to a 3rd party monitor along with the pseudonym under which the purchases were made (and the _context_id_ of the vendor). To allow the third party monitor to link these pseudonyms to a _prover_, the prover would be required to reveal the _nym_secret_ associated with this credential only to the _monitor_.

Note that this is why this specification separates _nym_secrets_ from other secrets (blind BBS messages) that might be used to "bind" a credential to a _prover_.

1.4. Terminology

The following terminology is used throughout this document:

SK The secret key for the signature scheme.
PK The public key for the signature scheme.
L The total number of signed messages.
R The number of message indexes that are disclosed (revealed) in a proof-of-knowledge of a signature.
U The number of message indexes that are undisclosed in a proof-of-knowledge of a signature.
scalar An integer between 0 and r-1, where r is the prime order of the selected groups, defined by each ciphersuite (see also Notation (#notation)).
generator A valid point on the selected subgroup of the curve being used that is employed to commit a value.
signature The digital signature output.
presentation_header (ph) A payload generated and bound to the context of a specific spk.
INVALID, ABORT Error indicators. INVALID refers to an error encountered during the Deserialization or Procedure steps of an operation. An INVALID value can be returned by a subroutine and handled by the calling operation. ABORT indicates that one or more of the initial constraints defined by the operation are not met. In that case, the operation will stop execution. An operation calling a subroutine that aborted must also immediately abort.

1.5. Notation

The following notation and primitives are used:

a || b Denotes the concatenation of octet strings a and b.

I \ J For sets I and J, denotes the difference of the two sets i.e., all the elements of I that do not appear in J, in the same order as they were in I.

X[a..b] Denotes a slice of the array X containing all elements from and including the value at index a until and including the value at index b. Note when this syntax is applied to an octet string, each element in the array X is assumed to be a single byte.

X[-n] Denotes the last n element of the array X

range(a, b) For integers a and b, with a <= b, denotes the ascending ordered list of all integers between a and b inclusive (i.e., the integers "i" such that a <= i <= b).

length(input) Takes as input either an array or an octet string. If the input is an array, returns the number of elements of the array. If the input is an octet string, returns the number of bytes of the inputted octet string.

Terms specific to pairing-friendly elliptic curves that are relevant to this document are restated below, originally defined in [I-D.irtf-cfrg-pairing-friendly-curves].

E1, E2 elliptic curve groups defined over finite fields. This document assumes that E1 has a more compact representation than E2, i.e., because E1 is defined over a smaller field than E2. For a pairing-friendly curve, this document denotes operations in E1 and E2 in additive notation, i.e., P + Q denotes point addition and x * P denotes scalar multiplication.

G1, G2 subgroups of E1 and E2 (respectively) having prime order r.

GT a subgroup, of prime order r, of the multiplicative group of a field extension.

e G1 x G2 -> GT: a non-degenerate bilinear map.

r The prime order of the G1 and G2 subgroups.

BP1, BP2 base (constant) points on the G1 and G2 subgroups respectively.

Identity_G1, Identity_G2, Identity_GT The identity element for the G1, G2, and GT subgroups respectively.

hash_to_curve_g1(ostr, dst) -> P A cryptographic hash function that takes an arbitrary octet string as input and returns a point in G1, using the hash_to_curve operation defined in [I-D.irtf-cfrg-hash-to-curve] and the inputted dst as the domain separation tag for that operation (more specifically, the inputted dst will become the DST parameter for the hash_to_field operation, called by hash_to_curve).

point_to_octets_g1(P) -> ostr, point_to_octets_g2(P) -> ostr returns

the canonical representation of the point P for the respective subgroup as an octet string. This operation is also known as serialization.

octets_to_point_g1(ostr) -> P, octets_to_point_g2(ostr) -> P returns the point P for the respective subgroup corresponding to the canonical representation ostr, or INVALID if ostr is not a valid output of the respective point_to_octets_g* function. This operation is also known as deserialization.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Key Concepts

A _pseudonym_ will be cryptographically generated for each prover-context of usage pair. Its value is dependent on a pseudonym secret (nym_secret) and a context identifier (context_id).

3.1. Context Identifier

The Context Identifier (context_id) is an octet string that represents a specific context of usage, within which, the pseudonym will have a constant value. Context Identifiers can take the form of unique Verifier Identifiers, Session Identifiers etc., depending on the needs of the application. Verifiers will be able to use the pseudonym values to track the presentations generated by a Prover, using the same signature, for that specific context.

3.2. Prover Pseudonym Secret

The _Prover_ pseudonym secret (nym_secrets) is a vector of one or more secret scalars used in the pseudonym calculation procedure of Section 4. The _Prover_ needs to keep this information secret as its name indicates. For a BBS signature that supports pseudonyms to be generated, the blind issuance procedure described in BBS Blind Signatures ([BlindBBS]) will be used, where the _Prover_ will send a commitment to the nym_secrets vector to the _Signer_. The last value of the nym_secrets list will be updated with entropy chosen by the _Signer_ (called signer_nym_entropy). This prevents malicious actors from re-using a stolen nym_secrets vector or _Prover_ commitment.

The _prover_nyms_ is a a vector of prover secret scalars and is only sent to the _signer_ in a binding and hiding commitment. The _signer_nym_entropy_ is "blindly added" in by the _signer_ during the signing procedure of Section 6.1.2 and sent back to the _prover_ along with the signature.

3.3. Pseudonyms

The _pseudonym_ is a cryptographic value computed by the prover based on the nym_secrets and the context_id. At a high level this is computed as a function of the context_id and the nym_secrets value. See Section Section 4 for details. The pseudonym is sent to a verifier along with the BBS proof.

This document defines a pseudonym as point of the G1 group different from the Identity (Identity_G1) or the base point (BP1) of G1. A pseudonym remains constant for the same context, when combined with the same signature, but is unique (and unlinkable) across different contexts. In other words, when the Prover presents multiple BBS proofs with a pseudonym to a Verifier, the pseudonym value will be constant across those presentations, if the same context_id value is used. When presenting a BBS proof with a pseudonym to a different context, the pseudonym value will be different. Note that since pseudonyms are group points, their value will necessarily change if a different a ciphersuite with a different curve will be used. Serialization and deserialization of the pseudonym point MUST be done using the point_to_octets_g1 and octets_to_point_g1 defined by the BBS ciphersuite used (see Section 6 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-03.html#name-ciphersuites) of [I-D.irtf-cfrg-bbs-signatures]).

This document specifies pseudonyms to be BBS Interface specific (see Section TBD of [I-D.irtf-cfrg-bbs-signatures] for the definition of the BBS Interface). It is outside the scope of this document to provide a procedure for "linking" the pseudonyms that are used by different Interfaces or that are based on different ciphersuites. An option is for the Prover to present both pseudonyms with the relevant BBS proofs to the Verifier, and upon validation of both, the Verifier to internally link the 2 pseudonyms together.

3.4. Mapping Messages to Scalars

Each BBS Interface defines an operation that will map the inputted messages to scalar values, required by the core BBS operations. Each Interface can use a different mapping procedure, as long as it comforts to the requirements outlined in [I-D.irtf-cfrg-bbs-signatures]. For using BBS with pseudonyms, the mapping operation used by the interface is REQUIRED to additionally

adhere the following rule;

For each set of messages and separate message msg',
if C1 = messages_to_scalars(messages.push(msg')),
and msg_prime_scalar = messages_to_scalars((msg')),
and C2 = messages_to_scalars(messages).push(msg_prime_scalar),
it will always hold that C1 == C2.

Informally, the above means that each message is mapped to a scalar independently from all the other messages. For example, if a = messages_to_scalars((msg_1)) and b = messages_to_scalars((msg_2)), then (a, b) = messages_to_scalars((msg_1, msg_2)). Its trivial to see that the messages_to_scalars operation that is defined in Section TBD of [I-D.irtf-cfrg-bbs-signatures], has the required property. That operation will be used by the Interface defined in this document to map the messages to scalars. Note that the above operation (and hence the defined by this document Interface), only accepts messages that are octet strings.

4. Pseudonym Calculation Procedure

The following pseudo code describes how the pseudonym is calculated from the nym_secrets held by the Prover and the public context identifier. The pseudonym will be unique for different contexts (e.g., unique Verifier identifiers) and constant under constant inputs (i.e., the same context_id and nym_secrets). The context_id is an octet string representing the unique identifier of the context in which the pseudonym will have the same value. The nym_secrets value is a vector of scalars calculated from secret input provided by the Prover and random (but not secret) input provided by the Signer. This will guarantee uniqueness of the nym_secrets between different signatures and users.

```
OP = hash_to_curve_g1(context_id) // a point in the curve group G1
z = hash_to_scalar(context_id)
poly = sum i = 0 to N-1 of nym_secrets[i]*z^i // in the scalar field
pseudonym = OP*poly // in the curve group G1
```

Additionally, the nym_secrets value will be signed by the BBS Signature. This will bind the pseudonym to a specific signature, held by the Prover. During proof generation, along the normal BBS proof, the Prover will generate a proof of correctness of the pseudonym, i.e., that it has the form described above, and that it was constructed from the nym_secrets signed by the BBS signature used to generate that proof.

5. High Level Procedures and Information Flows

To prevent forgeries in all cases all BBS messages are signed with the inclusion of some form of the provider pseudonym secret (nym_secret). In addition the pseudonym is always computed by the prover and sent with the proof to the verifier. While two different variations of signature and proof generation are given below based on the previously discussed unlinkability requirements there MUST be only one verification algorithm for the verifier to use.

1. The Prover computes their input for the nym_secrets (called prover_nyms) and retained for use when calculating the nym_secrets value.
2. The Prover will wrap up in a cryptographic commitment using the _CommitWithNym_ procedures of Blind BBS the messages they want to include in the signature (committed_messages) and the prover_nyms value, generating a commitment_with_proof and a secret_prover_blind.
3. The commitment_with_proof is conveyed to the signer which then uses the signing procedures in Section Section 6.1 to create a BBS signature and their input for the nym_secrets value, called signer_nym_entropy. They will convey both to the Prover.
4. On receipt of the signature and the signer_nym_entropy value, the Prover verifies the signature using the procedure of section Section 6.1 and calculates the nym_secrets value by adding their prover_nyms secret and the provided signer_nym_entropy values as detailed in later sections.
5. The Prover computes the _pseudonym_ based on the nym_secrets and the pseudonym’s context identifier context_id.
6. The Prover generates a proof using nym_secrets, secret_prover_blind, signature, messages, committed_messages and the indexes of the messages to be reveled from those two lists (i.e., disclosed_indexes and disclosed_committed_indexes) using the procedures of Section Section 6.2.
7. The Prover conveys the proof and pseudonym to the verifier. The verifier uses the procedure of Section Section 6.3 to verify the proof.

6. BBS Pseudonym Interface

The following section defines a BBS Interface that will make use of per-origin pseudonyms where the nym_secrets value is only known to the prover. The identifier of the Interface, api_id, is defined as ciphersuite_id || H2G_HM2S_PSEUDONYM_, where ciphersuite_id the unique identifier of the BBS ciphersuite used, as is defined in Section 6 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-03.html#name-ciphersuites) of [I-D.irtf-cfrg-bbs-signatures]).

The prover create a prover_nyms value and keeps it secret. Only sending a commitment on the prover_nyms, with the corresponding proof of correctness, that the signer will use when creating the signature.

6.1. Signature Generation and Verification with Pseudonym

6.1.1. Commitment

This section will describe the steps with which the Signer will generate a blind signature over an array of messages provided (and committed) by the Prover (committed_messages) and a pseudonym secret prover_nym, also chosen by the Prover. During signature generation, the Signer will provide their own randomness into the pseudonym secret. This will ensure that the pseudonym secret will always be unique, among different signature generation events.

This section will provide a high level description of the required operations, by detailing the modifications required in the relevant BBS blind signature operations, to also consider the use of pseudonyms. The full formal description of the operation can be seen at Appendix. We will reference those operations where appropriate in this section.

Initially, the Prover will chose a set of messages committed_messages that they want to be included in the signature, without reveling them to the Signer. They will also choose their part of the pseudonym secret prover_nym as a random scalar value.

```
(commitment_with_proof, secret_prover_blind) = CommitWithNym(
                                                  committed_messages,
                                                  prover_nyms,
                                                  api_id)
```

Inputs:

- committed_messages (OPTIONAL), a vector of octet strings. If not supplied it defaults to the empty array ("()").
- prover_nyms (REQUIRED), a vector of random scalar values.
- api_id (OPTIONAL), octet string. If not supplied it defaults to the empty octet string ("").

Outputs:

- (commitment_with_proof, secret_prover_blind), a tuple comprising from an octet string and a random scalar in that order.

Procedure:

```
1. committed_message_scalars = BBS.messages_to_scalars(
                                          committed_messages, api_id)
2. committed_message_scalars.append(prover_nyms)

3. blind_generators = BBS.create_generators(
                                  length(committed_message_scalars) + 1,
                                  "BLIND_" || api_id)

4. return Blind.CoreCommit(committed_message_scalars,
                                            blind_generators, api_id)
```

6.1.2. Blind Issuance

The Signer generate a signature from a secret key (SK), the commitment with proof, the length_nym_vector, the signer_nym_entropy and optionally over a header and vector of messages using the BlindSignWithNym procedure shown below. The length of the nym vector parameter MUST be furnished by the prover along with the commitment with proof. See the security considerations section for the need for this parameter.

Typically the signer_nym_entropy will be a fresh random scalar, however in the case of "reissue" of a signature for a prover who wants to keep their same pseudonymous identity this value can be reused for the same prover if desired.

```
BlindSignWithNym(SK, PK, commitment_with_proof, length_nym_vector,
                 signer_nym_entropy, header, messages)

Inputs:

- SK (REQUIRED), a secret key in the form outputted by the KeyGen
                 operation.
- PK (REQUIRED), an octet string of the form outputted by SkToPk
                 provided the above SK as input.
- commitment_with_proof (OPTIONAL), an octet string, representing a
                                    serialized commitment and
                                    commitment_proof, as the first
                                    element outputted by the
                                    CommitWithNym operation. If not
                                    supplied, it defaults to the empty
                                    string ("").
- length_nym_vector (REQUIRED), the length of the prover_nyms secret
                                vector.
- signer_nym_entropy (REQUIRED), a scalar value.
- header (OPTIONAL), an octet string containing context and application
                     specific information. If not supplied, it defaults
                     to an empty string ("").
- messages (OPTIONAL), a vector of octet strings. If not supplied, it
                       defaults to the empty array ("()").

Deserialization:

1. L = length(messages)

// calculate the number of blind generators used by the commitment,
// if any.
2. M = length(commitment_with_proof)
3. if M != 0, M = M - octet_point_length - octet_scalar_length
4. M = M / octet_scalar_length
5. if M < 0, return INVALID

Procedure:

1.  generators = BBS.create_generators(L + 1, api_id)
2.  blind_generators = BBS.create_generators(M, "BLIND_" || api_id)

3.  commit = Blind.deserialize_and_validate_commit(
                       commitment_with_proof, blind_generators, api_id)
4.  if commit is INVALID, return INVALID

5.  message_scalars = BBS.messages_to_scalars(messages, api_id)

6.  B = Blind.B_calculate(generators, commit, message_scalars)
```

```
7.  if B is INVALID, return INVALID
8.  B = B + blind_generators[-1] * signer_nym_entropy

9.  combined_header = header || I2OSP(length_nym_vector, 8)
10. blind_sig = Blind.FinalizeBlindSign(SK,
                                        PK,
                                        B,
                                        generators,
                                        blind_generators,
                                        combined_header,
                                        api_id)

11. if blind_sig is INVALID, return INVALID
12. return blind_sig
```

6.1.3. Verification and Finalization

The following operation both verifies the generated blind signature, as well as calculating and returning the final nym_secret, used to calculate the pseudonym value during proof generation.

This operation uses the Blind.Verify function as defined in Section 4.2.2 (https://www.ietf.org/archive/id/draft-kalos-bbs-blind-signatures-01.html#name-blind-signature-verificatio) of the Blind BBS document [BlindBBS]

```
nym_secret = VerifyFinalizeWithNym(PK,
                                   signature,
                                   header,
                                   messages,
                                   committed_messages,
                                   prover_nyms,
                                   signer_nym_entropy,
                                   secret_prover_blind)
```

Inputs:

- PK (REQUIRED), an octet string of the form outputted by the SkToPk operation.
- signature (REQUIRED), an octet string of the form outputted by the Sign operation.
- header (OPTIONAL), an octet string containing context and application specific information. If not supplied, it defaults to an empty string.
- messages (OPTIONAL), a vector of octet strings. If not supplied, it defaults to the empty array "()".
- committed_messages (OPTIONAL), a vector of octet strings. If not supplied, it defaults to the empty

```
                                  array "()".
- prover_nyms (REQUIRED), a vector of scalar values.
- signer_nym_entropy (OPTIONAL), a scalar value. If not supplied, it
                                  defaults to the zero scalar (0).
- secret_prover_blind (OPTIONAL), a scalar value. If not supplied it
                                   defaults to zero "0".

Outputs:

- nym_secrets, a vector of scalar values; or INVALID.

Procedure:

1. (message_scalars, generators) = Blind.prepare_parameters(
                                     messages,
                                     committed_messages,
                                     length(messages) + 1,
                                     length(committed_messages) + N + 1,
                                     secret_prover_blind,
                                     api_id)

2. nym_secrets = prover_nym.copy()
3. nym_secrets[-1] = prover_nyms[-1] + signer_nym_entropy // in scalar
                                                          // field
4. message_scalars.append(nym_secrets)
5. combined_header = header || I2OSP(length(prover_nyms), 8)
6. res = BBS.CoreVerify(PK, signature, generators, combined_header,
                                                 message_scalars, api_id)

7. if res is INVALID, return INVALID
8. return nym_secrets
```

6.2. Proof Generation with Pseudonym

This section defines the ProofGenWithNym operations, for calculating a BBS proof with a pseudonym. The BBS proof is extended to include a zero-knowledge proof of correctness of the pseudonym value, i.e., that is correctly calculated using the (undisclosed) pseudonym secret (nym_secrets), and that is "bound" to the underlying BBS signature (i.e., that the nym_secrets value is signed by the Signer).

Validating the proof (see ProofVerifyWithNym defined in Section 6.3), guarantees authenticity and integrity of the header, presentation header and disclosed messages, knowledge of a valid BBS signature as well as correctness and ownership of the pseudonym.

To support pseudonyms, the ProofGenWithNym procedure takes the pseudonym secret nym_secret, as well as the context identifier context_id, which the pseudonym will be bounded to.

```
(proof, pseudonym) = ProofGenWithNym(PK,
                                     signature,
                                     header,
                                     ph,
                                     nym_secrets,
                                     context_id,
                                     messages,
                                     committed_messages,
                                     disclosed_indexes,
                                     disclosed_commitment_indexes,
                                     secret_prover_blind)
```

Inputs:

- PK (REQUIRED), an octet string of the form outputted by the SkToPk operation.
- signature (REQUIRED), an octet string of the form outputted by the Sign operation.
- header (OPTIONAL), an octet string containing context and application specific information. If not supplied, it defaults to an empty string.
- ph (OPTIONAL), an octet string containing the presentation header. If not supplied, it defaults to an empty string.
- nym_secrets (REQUIRED), the vector of nym secret scalars, known to Prover.
- context_id (REQUIRED), an octet string.
- messages (OPTIONAL), a vector of octet strings. If not supplied, it defaults to the empty array "()".
- committed_messages (OPTIONAL), a vector of octet strings. If not supplied, it defaults to the empty array "()".
- disclosed_indexes (OPTIONAL), vector of unsigned integers in ascending order. Indexes of disclosed messages. If not supplied, it defaults to the empty array "()".
- disclosed_commitment_indexes (OPTIONAL), vector of unsigned integers in ascending order. Indexes of disclosed committed messages. If not supplied, it defaults to the empty array "()".
- secret_prover_blind (OPTIONAL), a scalar value. If not supplied it defaults to zero "0".

Parameters:

- api_id, the octet string ciphersuite_id || "BLIND_H2G_HM2S_", where
  ciphersuite_id is defined by the ciphersuite and
  "BLIND_H2G_HM2S_"is an ASCII string composed of 15 bytes.

Outputs:

- proof, an octet string; or INVALID.

Deserialization:

```
1. L = length(messages)
2. M = length(committed_messages)
3. if length(disclosed_indexes) > L, return INVALID
4. for i in disclosed_indexes, if i < 0 or i >= L, return INVALID
5. if length(disclosed_commitment_indexes) > M, return INVALID
6. for j in disclosed_commitment_indexes,
                              if i < 0 or i >= M, return INVALID
```

Procedure:

```
1. (message_scalars, generators) = Blind.prepare_parameters(
                                            messages,
                                            committed_messages,
                                            L + 1,
                                            M + length(nym_secrets) + 1,
                                            secret_prover_blind,
                                            api_id)

2. indexes = ()
3. indexes.append(disclosed_indexes)
4. for j in disclosed_commitment_indexes: indexes.append(j + L + 1)

5. (proof, pseudonym) = CoreProofGenWithNym(
                       PK,
                       signature,
                       generators.append(blind_generators),
                       header,
                       ph,
                       context_id,
                       message_scalars.append(nym_secrets)
                       indexes,
                       length(nym_secrets),
                       api_id)
6. return (proof, pseudonym)
```

6.3. Proof Verification with Pseudonym

This operation validates a BBS proof with a pseudonym, given the Signer's public key (PK), the proof, the pseudonym, the context identifier that was used to create it, a header and presentation header, the disclosed messages and committed messages as well as the, the indexes those messages had in the original vectors of signed messages. Validating the proof also validates the correctness and ownership by the Prover of the received pseudonym.

```
result = ProofVerifyWithNym(PK,
                            proof,
                            header,
                            ph,
                            pseudonym,
                            context_id,
                            length_nym_vector,
                            L,
                            disclosed_messages,
                            disclosed_committed_messages,
                            disclosed_indexes,
                            disclosed_committed_indexes)
```

Inputs:

- PK (REQUIRED), an octet string of the form outputted by the SkToPk operation.
- proof (REQUIRED), an octet string of the form outputted by the ProofGen operation.
- header (OPTIONAL), an optional octet string containing context and application specific information. If not supplied, it defaults to the empty octet string ("").
- ph (OPTIONAL), an octet string containing the presentation header. If not supplied, it defaults to the empty octet string ("").
- pseudonym (REQUIRED), the pseudonym (element of group G1).
- context_id (REQUIRED), an octet string.
- length_nym_vector (REQUIRED), integer.
- L (OPTIONAL), an integer, representing the total number of Signer known messages if not supplied it defaults to 0.
- disclosed_messages (OPTIONAL), a vector of octet strings. If not supplied, it defaults to the empty array ("()").
- disclosed_indexes (OPTIONAL), vector of unsigned integers in ascending order. Indexes of disclosed messages. If not supplied, it defaults to the empty array ("()").

Parameters:

- api_id, the octet string ciphersuite_id || "H2G_HM2S_", where ciphersuite_id is defined by the ciphersuite and "H2G_HM2S_"is an ASCII string comprised of 9 bytes.
- (octet_point_length, octet_scalar_length), defined by the ciphersuite.

Outputs:

- result, either VALID or INVALID.

Deserialization:

```
1. proof_len_floor = 3 * octet_point_length + 4 * octet_scalar_length
2. if length(proof) < proof_len_floor, return INVALID
3. U = floor((length(proof) - proof_len_floor) / octet_scalar_length)
4. total_no_messages = length(disclosed_indexes) +
                            length(disclosed_committed_indexes) + U - 1
5. M = total_no_messages - L
```

Procedure:

```
1. (message_scalars, generators) = Blind.prepare_parameters(
                                          disclosed_messages,
                                          disclosed_committed_messages,
                                          L + 1,
                                          M + 1,
                                          NONE,
                                          api_id)

2. indexes = ()
3. indexes.append(disclosed_indexes)
4. for j in disclosed_commitment_indexes: indexes.append(j + L + 1)

5. result = CoreProofVerifyWithNym(PK,
                                   proof,
                                   pseudonym,
                                   context_id,
                                   length_nym_vector,
                                   generators,
                                   header,
                                   ph,
                                   message_scalars,
                                   indexes,
                                   api_id)
6. return result
```

7. Core Operations

7.1. Core Proof Generation

This operations computes a BBS proof and a zero-knowledge proof of correctness of the pseudonym in "parallel" (meaning using common randomness), as to both create a proof that the pseudonym was correctly calculated using an undisclosed value that the Prover knows (i.e., the nym_secrets value), but also that this value is "signed" by the BBS signature (the last undisclosed message). As a result, validating the proof guarantees that the pseudonym is correctly computed and that it was computed using the Prover identifier that was included in the BBS signature.

The operation uses the BBS.ProofInit and BBS.ProofFinalize operations defined in Section 3.7.1 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-07.html#name-proof-initialization) and Section 3.7.2 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-07.html#name-proof-finalization) correspondingly of [I-D.irtf-cfrg-bbs-signatures], the PseudonymProofInit operation defined in Section 7.3.1 and the ProofWithPseudonymChallengeCalculate defined in Section 8.1.

```
(proof, pseudonym) = CoreProofGenWithNym(PK,
                                         signature,
                                         pseudonym,
                                         verifier_id,
                                         generators,
                                         header,
                                         ph,
                                         messages,
                                         disclosed_indexes,
                                         length_nym_secrets
                                         api_id)
```

Inputs:

- PK (REQUIRED), an octet string of the form outputted by the SkToPk operation.
- signature (REQUIRED), an octet string of the form outputted by the Sign operation.
- pseudonym (REQUIRED), A point of G1, different from the Identity of G1, as outputted by the CalculatePseudonym operation.
- context_id (REQUIRED), an octet string, representing the unique proof Verifier identifier.
- generators (REQUIRED), vector of points in G1.
- header (OPTIONAL), an octet string containing context and application

specific information. If not supplied, it defaults to an empty string.
- ph (OPTIONAL), an octet string containing the presentation header. If not supplied, it defaults to an empty string.
- message_scalars (OPTIONAL), a vector of scalars representing the messages. If not supplied, it defaults to the empty array "()" must include the nym_secret scalar as last element.
- disclosed_indexes (OPTIONAL), vector of unsigned integers in ascending order. Indexes of disclosed messages. If not supplied, it defaults to the empty array "()".
- length_nym_secrets (REQUIRED), the length of the nym_secrets vector.
- api_id (OPTIONAL), an octet string. If not supplied it defaults to the empty octet string ("").

Parameters:

- P1, fixed point of G1, defined by the ciphersuite.

Outputs:

- proof, an octet string; or INVALID.

Deserialization:

```
1.  signature_result = octets_to_signature(signature)
2.  if signature_result is INVALID, return INVALID
3.  (A, e) = signature_result
4.  L = length(message_scalars)
5.  R = length(disclosed_indexes)
6.  (i1, ..., iR) = disclosed_indexes
7.  if R > L - 1, return INVALID, Note: we never reveal the nym_secret.
8.  U = L - R

// Note: nym_secret is last message and is not revealed.
9.  undisclosed_indexes = (0, 1, ..., L - 1) \ disclosed_indexes
10. (i1, ..., iR) = disclosed_indexes
11. (j1, ..., jU) = undisclosed_indexes
12. disclosed_messages = (message_scalars[i1], ..., message_scalars[iR])
13. undisclosed_messages = (message_scalars[j1], ...,
                                                    message_scalars[jU])
```

ABORT if:

```
1. for i in disclosed_indexes, i < 0 or i > L - 1, // Note: nym_secret
                                                   // is the Lth message
                                                   // and not revealed.
```

Procedure:

```
1.  random_scalars = calculate_random_scalars(5+U)
2.  combined_header = header || I2OSP(length_nym_vector, 8)
3.  init_res = BBS.ProofInit(PK,
                             signature_res,
                             combined_header,
                             random_scalars,
                             generators,
                             message_scalars,
                             undisclosed_indexes,
                             api_id)
4.  if init_res is INVALID, return INVALID

5.  pseudonym_init_res = PseudonymProofInit(
                                  context_id,
                                  message_scalars[-length_nym_secrets],
                                  random_scalars[-length_nym_secrets])

6.  if pseudonym_init_res is INVALID, return INVALID
7.  pseudonym = pseudonym_init_res[0]

8.  challenge = ProofWithPseudonymChallengeCalculate(init_res,
                                                     pseudonym_init_res,
                                                     disclosed_indexes,
                                                     disclosed_messages,
                                                     ph,
                                                     api_id)
9.  proof = BBS.ProofFinalize(init_res, challenge, e_value,
                                  random_scalars, undisclosed_messages)
10. return (proof, pseudonym)
```

7.2. Core Proof Verification

This operation validates a BBS proof that also includes a pseudonym. Validating the proof, other than the correctness and integrity of the revealed messages, the header and the presentation header values, also guarantees that the supplied pseudonym was correctly calculated, i.e., that it was produced using the Verifier’s identifier and the signed (but undisclosed) Prover’s identifier, following the operation defined in Section 4.

The operation uses the BBS.ProofVerifyInit operation defined Section 3.7.3 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-07.html#name-proof-verification-initiali) of [I-D.irtf-cfrg-bbs-signatures], the PseudonymProofVerifyInit operation defined in Section 7.3.2 and the ProofWithPseudonymChallengeCalculate operation defined in Section 8.1.

```
result = CoreProofVerifyWithNym(PK,
                                 proof,
                                 pseudonym,
                                 context_id,
                                 length_nym_vector,
                                 generators,
                                 header,
                                 ph,
                                 disclosed_messages,
                                 disclosed_indexes,
                                 api_id)
```

Inputs:

- PK (REQUIRED), an octet string of the form outputted by the SkToPk operation.
- proof (REQUIRED), an octet string of the form outputted by the ProofGen operation.
- pseudonym (REQUIRED), A point of G1, different from the Identity of G1, as outputted by the CalculatePseudonym operation.
- context_id (REQUIRED), an octet string, representing the unique proof Verifier identifier.
- length_nym_vector (REQUIRED), the length of the nym secrets vector from the Prover.
- generators (REQUIRED), vector of points in G1.
- header (OPTIONAL), an optional octet string containing context and application specific information. If not supplied, it defaults to an empty string.
- ph (OPTIONAL), an octet string containing the presentation header. If not supplied, it defaults to an empty string.
- disclosed_messages (OPTIONAL), a vector of scalars representing the messages. If not supplied, it defaults to the empty array "()".
- disclosed_indexes (OPTIONAL), vector of unsigned integers in ascending order. Indexes of disclosed messages. If not supplied, it defaults to the empty array "()".
- api_id (OPTIONAL), an octet string. If not supplied it defaults to the empty octet string ("").

Parameters:

- P1, fixed point of G1, defined by the ciphersuite.

Outputs:

- result, either VALID or INVALID.

Deserialization:

1. proof_result = octets_to_proof(proof)
2. if proof_result is INVALID, return INVALID
3. (Abar, Bbar, r2^, r3^, commitments, cp) = proof_result
4. W = octets_to_pubkey(PK)
5. if W is INVALID, return INVALID
6. R = length(disclosed_indexes)
7. (i1, ..., iR) = disclosed_indexes

ABORT if:

1. for i in disclosed_indexes, i < 1 or i > R + length(commitments) - 1

Procedure:

```
1. combined_header = header || I2OSP(length_nym_vector, 8)
2. init_res = BBS.ProofVerifyInit(PK,
                                  proof_result,
                                  combined_header,
                                  generators,
                                  messages,
                                  disclosed_indexes,
                                  api_id)

3. pseudonym_init_res = PseudonymProofVerifyInit(
                                      pseudonym,
                                      context_id,
                                      commitments[-length_nym_vector],
                                      cp)
4. if pseudonym_init_res is INVALID, return INVALID

5. challenge = ProofWithPseudonymChallengeCalculate(init_res,
                                                    pseudonym_init_res,
                                                    disclosed_indexes,
                                                    messages,
                                                    ph,
                                                    api_id)
6. if cp != challenge, return INVALID
7. if e(Abar, W) * e(Bbar, -BP2) != Identity_GT, return INVALID
```

8. return VALID

7.3. Pseudonym Proof Generation Utilities

7.3.1. Pseudonym Proof Generation Initialization

```
pseudonym_init_res = PseudonymProofInit(context_id,
                                        nym_secrets, random_scalars)
```

Inputs:

- context_id (REQUIRED), an octet string
- nym_secrets (REQUIRED), a scalar value
- random_scalars (REQUIRED), a scalar value

Outputs:

- a tuple consisting of three elements from the G1 group, or INVALID.

Procedure:

```
1.  OP = hash_to_curve_g1(context_id, api_id)
2.  z = hash_to_scalar(context_id, api_id || 'VECT_NYM_SECRETS')

// Polynomial evaluation over nym_secrets and random_scalars, this can
// be done in any way desired, e.g., Horner's rule.
3.  poly_eval_pseudo = nym_secrets[0]
4.  poly_eval_proof = random_scalars[0]
5.  z_n = z;
6.  for(let i = 1; i < nym_secrets.length; i++) {
        poly_eval_pseudo += nym_secrets[i] * z_n // in scalar field
        poly_eval_proof += random_scalars[i] * z_n // in scalar field
        z_n *= z; // in scalar field
    }
7.  pseudonym = OP * poly_eval_pseudo // in group G1
8.  Ut = OP * poly_eval_proof // in group G1
9.  if pseudonym == Identity_G1 or Ut == Identity_G1, return INVALID
10. return (pseudonym, context_id, Ut)
```

7.3.2. Pseudonym Proof Verification Initialization

```
pseudonym_init_res = PseudonymProofVerifyInit(pseudonym,
                                              context_id,
                                              nym_secret_commitments
                                              proof_challenge)
```

Inputs:

- pseudonym (REQUIRED), an element of the G1 group.
- context_id (REQUIRED), an octet string.
- nym_secret_commitments (REQUIRED), a vector of scalar values.
- proof_challenge (REQUIRED), a scalar value.

Outputs:

- a tuple consisting of three elements from the G1 group, or INVALID.

Procedure:

```
1. OP = hash_to_curve_g1(context_id, api_id)
2. z = hash_to_scalar(context_id, api_id || 'VECT_NYM_SECRETS');

// Polynomial evaluation over nym_secret_commitments, this can be
// done in any way desired, e.g., Horner's rule.
3. poly_eval_proof = nym_secret_commitments[0]
4. z_n = z;
5. for(let i = 1; i < nym_secret_commitments.length; i++) {
        poly_eval_proof += nym_secret_commitments[i] * z_n // in scalar
                                                           // field
        z_n *= z; // in scalar field
    }
6. Uv = OP * poly_eval_proof - pseudonym * proof_challenge // in G1
7. if Uv == Identity_G1, return INVALID
8. return (pseudonym, context_id, Uv)
```

8. Utility Operations

8.1. Challenge Calculation

```
challenge = ProofWithPseudonymChallengeCalculate(init_res,
                                                 pseudonym_init_res,
                                                 i_array,
                                                 msg_array,
                                                 ph, api_id)
```

Inputs:
- init_res (REQUIRED), vector representing the value returned after initializing the proof generation or verification operations, consisting of 5 points of G1 and a

scalar value, in that order.
- pseudonym_init_res (REQUIRED), vector representing the value returned after initializing the pseudonym proof, consisting of (pseudonym, context_id, Ut).
- i_array (REQUIRED), array of non-negative integers (the indexes of the disclosed messages).
- msg_array (REQUIRED), array of scalars (the disclosed messages after mapped to scalars).
- ph (OPTIONAL), an octet string. If not supplied, it must default to the empty octet string ("").
- api_id (OPTIONAL), an octet string. If not supplied it defaults to the empty octet string ("").

Outputs:

- challenge, a scalar.

Definitions:

1. challenge_dst, an octet string representing the domain separation tag: api_id || "H2S_" where "H2S_" is an ASCII string comprised of 4 bytes.

Deserialization:

1. R = length(i_array)
2. (i1, ..., iR) = i_array
3. (msg_i1, ..., msg_iR) = msg_array
4. (Abar, Bbar, D, T1, T2, domain) = init_res
5. (pseudonym, context_id, Ut) = pseudonym_init_res

ABORT if:

1. R > 2^64 - 1 or R != length(msg_array)
2. length(ph) > 2^64 - 1

Procedure:
1. c_arr = (R, i1, msg_i1, i2, msg_i2, ..., iR, msg_iR, Abar, Bbar, D, T1, T2, pseudonym, Ut, domain)
2. c_octs = serialize(c_arr) || I2OSP(length(ph), 8) || ph || I2OSP(length(context_id), 8) || context_id
3. return hash_to_scalar(c_octs, challenge_dst)

9. Privacy Considerations

9.1. Limited Everlasting Unlinkability

BBS pseudonyms are computationally unlinkable under the assumption of the hardness of the discrete log problem. In the presence of a Cryptographically Relevant Quantum Computer (CRQC) BBS pseudonyms can provide "limited" everlasting unlinkability. This is similar to how BBS proofs provide everlasting unlinkability in the presence of a CRQC.

Let N be the length of the secret prover_nyms vector, let M be the number of pseudonyms gathered from colluding verifiers each possessing a unique context_id. In the presence of a CRQC if N > M the pseudonyms will be unlinkable. In other words, as long as the _Prover_ does not generate (different) pseudonyms for more than N unique context identifiers (i.e., N different context_id values, for example, for N different Verifiers), unlinkability of the pseudonyms will hold, even in the presence of a CRQC.

TODO: Proof/Discussion to be furnished

10. Security Considerations

10.1. Preventing Impersonation Attacks

Assuming an honest issuer, to prevent impersonation attacks by a malicious prover that has obtained the prover_nyms from another prover, the issuer SHOULD use a different signer_nym_entropy for each different prover that it provides a pseudonym backed signature.

10.2. Preventing Sybil Attacks

Assuming an honest issuer, to prevent sybil attacks by a malicious prover, we require that the prover use the same set of nym_secrets in computing pseudonyms. In particular, the prover declares the length of the prover_nyms/nym_secrets vectors, N, in their commitment with proof and this value gets bound to the signature from the issuer. In verifying the proof of the pseudonym the verifier will be checking that the N values of the nym_secrets vector was actually used in the computation.

TODO Security

11. Ciphersuites

This document does not define new BBS ciphersuites. Its ciphersuite defined in Section 6 (https://www.ietf.org/archive/id/draft-irtf-cfrg-bbs-signatures-03.html#name-ciphersuites) of [I-D.irtf-cfrg-bbs-signatures]) can be used to instantiate the operations of the described scheme.

12. Test Vectors

12.1. BLS12-381-SHA-256

12.1.1. Generators

```
api_id = "BBS_BLS12381G1_XMD:SHA-256_SSWU_RO_H2G_HM2S_PSEUDONYM_"

P1 = "a8ce256102840821a3e94ea9025e4662b205762f9776b3a766c872b948f1fd225e
      7c59698588e70d11406d161b4e28c9"
Q1 = "a87fa55cfc29d0d0ef43b7816018c6162b9c4a5ddd5239ed24d9799f8e105c267d
      81ccb22f6379853c4070c28c71f13c"

Generators = {

H_0 = "8c6de69580b83b7c6d773857ae64b4495955eb06e67ebc5855af89c72cd8d9bea
       9fd7f71eca20c6a3388dfa67b1e7ccf"
H_1 = "aed1785c1c00d00893413e5011ecdc98706958a2ccf175be8a42afa56ef19c86c
       a6c14afe7e74a72596704fe34b6611d"
H_2 = "b5b5142c6314a918882439b634adb926cf42a55da2962865bcf09fe746554851a
       e075d9a4a03add64a0bec997eb708a8"
H_3 = "b54f25df3ba79539da4ceb375522625518590eebd52211d40cf1083f8a9e8c1bb
       19212f1f31711fd333678002a362830"
H_4 = "8fcb548cd1e5cddba514a7f90e3b0ebd00bd82bd66158f70cc7fffdb09f14ac8f
       c56ef9587cc41a82614533444494e75"
H_5 = "94a1d50478150c469711ccd09fa4544a590a1903f16445a4a5bc0ab639f1a4085
       80f2464972198d128f1bb4a4fa41b0b"
H_6 = "b36ceb6c0cc0850fd3a2e64fa534a1c15566f99688ec6134c5223a33de83ce553
       4d43c0973a2769ce887d5bac8481519"
H_7 = "a4e6dff6038ab2e8265d9c177d110c742bc97f3a32bd70123ecd67176181b2068
       a0ae8323db6e061e4d8e62db6f283ad"
H_8 = "90977c482711c97318d1e4c4205308847727ca7dbf3ca7d1c55f1906aeca21aae
       22b7f43e73feae41c9a9be75319015d"
H_9 = "a435ee46442dd320426a1eb163176154bb144a7f829900d0e14ec7c28d882572a
       cc1b4f670ef7cf5b41a4bea2efae6c6"

}
```

12.1.2. Blind Generators

```
api_id = "BLIND_BBS_BLS12381G1_XMD:SHA-256_SSWU_RO_H2G_HM2S_PSEUDONYM_"

P1 = "a8ce256102840821a3e94ea9025e4662b205762f9776b3a766c872b948f1fd225e
      7c59698588e70d11406d161b4e28c9"
Q1 = "a264ef107598f1caaeb323b65164bcea80e88814810efc61ea27412e879c7cb934
      4b1b513118d3cf5c79bfa81268ef36"

Blind Generators = {

J_0 = "8af923aaeaad46bf889049b2e5de19ff17778343114e589d716cde6eaa553c9e5
       4fd6805afb244e445be2939ac789b35"
J_1 = "aa6c94da21fafb4cd604029cf599df139aa88ca1cf3676fb7da1e12ec6a8dc83c
       3d7fdbf33a79e760d810c4fbac37f6a"
J_2 = "8f65ebef29b60b81447821ea2d5a201d339b0c092021bd71eeee2d1f39d4972d3
       688c98c21831490583285c12f6da579"
J_3 = "b94e4549a9ecbec9b83c004d86649f0aa6510ba292a2e68d982e79ad4de0e5bd2
       972313a95170f4c5881d7a5b790c205"
J_4 = "a1d1d4460ee7475aa66fcd4c803b11cef74a75b9d4bfe9924de20434e01f35707
       855299c9d4ead6af5b93f57d9392d56"
J_5 = "8852ab63577b0a382df12320c5fc900bce57680d47e371ced873399bd9c5adc79
       3ee890a919fb9c293e55acb4ab0312b"

}
```

12.1.3. Commit

Mocked random scalar parameters

```
seed = "3.14159265358979323846264338327 9"
dst = "BBS_BLS12381G1_XMD:SHA-256_SSWU_RO_H2G_HM2S_COMMIT_MOCK_RANDOM_SC
       ALARS_DST_"
```

12.1.3.1. valid no committed messages commitment with proof

```
committedMessages = "[  ]"
proverNym = "undefined"
proverBlind = "3ba0a2583bc7229fa9f2ae3a6697091032947c3a48f302b7fd2b08ca9
               d193041"

Trace:

s_tilde = "3a3b481c984f4396a13b1f65368aa393d08455fbfd351ab80f593aa5de8b4
           b1d"
m_tildes = "[ 5e82a40ae25e65fb04d7722f36ecd62fa4f07c8815e74f0a14a7e0a654
            7a36ce ]"

commitmentWithProof = "b989fc492e2047f602504eb3e236c0acb04224c77ad0d4cbd
                       31c887b9eb05a1f27d7acfb266fe0ae062914bfa060984c5c
                       2ac3247080eb71fefc7e9622ffae372425a699a298ba991a0
                       bc5c6a3d9211347d0ce98d5c0550667269df1fb81f8fa30c0
                       7d4917c7c0786411ee5c05b00b9d501d3f8e244b860b7b111
                       40cddc9787a3ab54ec7fd0a8950dae339f396f2641b"
```

12.1.3.2. valid multiple committed messages commitment with proof

```
committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"
proverNym = "undefined"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"


Trace:

s_tilde = "691b0c56dff95cd15fc221a7d66ec71742fa8161a435ac51ffaa0f593b059
           89a"
m_tildes = "[ 2df678f035e3b5c2628d40645c3b53d30b77b992b4d1663aa313892d08
            a78e85, 2c0add8de9779bf9e3ba6ef2a863cec5e0375b66c44d326f3019
            14eb73cabb46, 57ea3273104c990cba7c65f88c766b013c326857be408a
            55fefea46c71f51a48, 4ffdcbebe564f0aeac3e40c58cc42964b1948b58
            1671070f85bf003ba61caafe, 19fbc9539129d0fe065c6a19d2df158820
            7232d163e098f127b270c3ad25fa08, 682afdc2c093d95b88e5e1455147
            44d9a254cace1ecd92f20cde388da9adc20f ]"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
```

12.1.4. Signature

12.1.4.1. valid no prover committed messages, no signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[  ]"

committedMessages = "[  ]"

commitmentWithProof = "b989fc492e2047f602504eb3e236c0acb04224c77ad0d4cbd
                       31c887b9eb05a1f27d7acfb266fe0ae062914bfa060984c5c
                       2ac3247080eb71fefc7e9622ffae372425a699a298ba991a0
                       bc5c6a3d9211347d0ce98d5c0550667269df1fb81f8fa30c0
                       7d4917c7c0786411ee5c05b00b9d501d3f8e244b860b7b111
                       40cddc9787a3ab54ec7fd0a8950dae339f396f2641b"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "3ba0a2583bc7229fa9f2ae3a6697091032947c3a48f302b7fd2b08ca9
               d193041"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "8b74e51a16d305b01d3ca60329e697a3cbc8f3272cd6d65d398b529656b5159f958
     9293b1ba4507d8e7eec9f2d4d1a79s"
domain = "01b0b85ea47afe36f772bbce626fb8064f85a3aa6233c33776194a170f45fa
          61"

signature = "aabc3014c598f3cd8fcc162950ff9aa9ac93c0877d33d1cc0b71b31964e
             3b109715d5af307e580b498b0ec8c0b8f848028ba9d881be84bf405295f
             27f02131028498c50c4fa3f6bb93483bf676ef1f1c"
```

12.1.4.2. valid multi prover committed messages, no signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"
```

```
header = "11223344556677889900aabbccddeeff"

messages = "[  ]"

committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "8aa0835565a69418b9010e4e2cb82757a97c729d26ca8227863941659a9a37a1472
     8461dd0a6f5338e2acdcb34498c84"
domain = "3f7830ef29ea1742aa66c15c4b9748ea8b1bd40a83f2204d419c4baabdf8b3
          1f"

signature = "88b2a07b490f81f8be334fe30b4034f90bbf77d7ccacc488fa8bfd7d989
             96f95ca7a02bfa5fef4983240f80e5956e7836b4630d6bc54a0a28b246b
             ed38f83b0c4bb378ef315e51b581abd6d8f3a6fded"
```

12.1.4.3. valid no prover committed messages, multiple signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[ 9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4
            a45f02, c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811
            f5fb075f9b80, 7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932a
            a348e07b73, 77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c
            , 496694774c5604ab1b2544eababcf0f53278ff50, 515ae153e22aae04
            ad16f759e07237b4, d183ddc6e2665aa4e2f088af, ac55fb33a75909ed
            , 96012096,  ]"

committedMessages = "[  ]"

commitmentWithProof = "b989fc492e2047f602504eb3e236c0acb04224c77ad0d4cbd
                       31c887b9eb05a1f27d7acfb266fe0ae062914bfa060984c5c
                       2ac3247080eb71fefc7e9622ffae372425a699a298ba991a0
                       bc5c6a3d9211347d0ce98d5c0550667269df1fb81f8fa30c0
                       7d4917c7c0786411ee5c05b00b9d501d3f8e244b860b7b111
                       40cddc9787a3ab54ec7fd0a8950dae339f396f2641b"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "3ba0a2583bc7229fa9f2ae3a6697091032947c3a48f302b7fd2b08ca9
               d193041"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "b6c39d33218bc3adaa6cd9d5539f51c66c75c30ee129d7f981e135c0ee5716d60cb
     5ee82f709224e0c8d9efefa778a38"
domain = "55891afaaadc4df689ca0d112e8aef3ea38b4256db93226ede05546eb8f1da
          f2"

signature = "9737d3d2ae17d170b3320329df8af1639b41ef2251e07437908786fd642
             1465ac46f98ff8091455d5bfd9394262a818631b7034648ef8a6c940a0b
             8232e7b160e4e71d8c676958b2d587da285bbf890a"
```

12.1.4.4. valid multiple signer and prover committed messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[ 9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4
            a45f02, c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811
            f5fb075f9b80, 7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932a
            a348e07b73, 77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c
            , 496694774c5604ab1b2544eababcf0f53278ff50, 515ae153e22aae04
            ad16f759e07237b4, d183ddc6e2665aa4e2f088af, ac55fb33a75909ed
            , 96012096,  ]"

committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "b677b21f402d69919483418900e0647b1a73aada9e081808b313cf5f83c43f0522b
     8682857659aa7920bb511ef4a477f"
```

```
domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"

signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"
```

12.1.5. Proof

Mocked random scalar parameters

```
seed = "3.141592653589793238462643383279"
dst = "BBS_BLS12381G1_XMD:SHA-256_SSWU_RO_H2G_HM2S_PROOF_MOCK_RANDOM_SCA
       LARS_DST_"
```

12.1.5.1. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
```

```
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
1: "c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811f5fb075f9b80"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
3: "77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
5: "515ae153e22aae04ad16f759e07237b4"
6: "d183ddc6e2665aa4e2f088af"
7: "ac55fb33a75909ed"
8: "96012096"
9: ""


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
1: "a75d8b634891af92282cc81a675972d1929d3149863c1fc0"
2: "835889a40744813a892eff9deb1edaeb"
3: "e1ca9729410dc6ba"
4: ""

Trace:

random_scalars:

r_1 = "57af863d7be8df38f5431df51734fcc8b070d4fff721eab0737be707a0479747"
r_2 = "64e66eecfa127d5942a5f80e2482bb283080fa337eee066d9b4ee79e6e3c3fa2"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 177e9bc4a3681ab187a037fc218fb5a401a2f253c1e7a76f095
                    32a8f9be75508, 398d3a444831eeb55b89946106c9f35b7296c7
                    ce0e2e2fa66a4318c10bbb98cc ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "10343839f7ad6c221366c10e96eb67949f2bcfc83614232ee7a5f9f564f
             e2499"


L = "10"
```

```
proof = "8b461b6d894ca153a2e1c05ac10c1bf21778b4ba08e9ca80949525afd86d533
         bf4b4f53ae3f7db67b9dcf55b5c4d3816b80b033b140c3bab14da11a54bb7af
         eb32c357cf6a1b73f100cbf1cb4e1c3fa1376a57d3be7e2f0395ec59b9e2c39
         c6ba744a214e5cec73752d3aa6ca1461cc38b4f69397282e8c9552b8f2add6e
         878f4edb8370003e141bacca3c3131bdbe016a02395e38459b716da68c90eed
         f33e13d01684d271148dc05c11f934a11986c40664e63c3eddd2a7f84edac4b
         092dfa6eb0bc58b8ae5c44b7b4392b288e700f59c56be0674865eb7e89069c2
         f39fd0a2a61379d615db25d33473774ff72033304a8a62dbf5515d4475808a5
         f9fae6052f5031d741535af95294195a97e9f87336fa53bf566b1e88bc8987b
         6850b0f06fc7423d92910970ac6cf33a8a53d1fad10343839f7ad6c221366c1
         0e96eb67949f2bcfc83614232ee7a5f9f564fe2499"
```

12.1.5.2. valid half prover committed messages and all signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
```

```
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
1: "c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811f5fb075f9b80"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
3: "77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
5: "515ae153e22aae04ad16f759e07237b4"
6: "d183ddc6e2665aa4e2f088af"
7: "ac55fb33a75909ed"
8: "96012096"
9: ""


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "3f7f8e5116d83e95bcc525249ad54ff216e7c4c15aad548320db23d7a7fd72cb"
r_2 = "423372f6da6b802f861ac89bb679e6725cf996040fe214fb9f98883b853ff541"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 192287eaaa219042b2217fc6fc8d39e8755bee5d61e441912f6
                    a1399cfc12d17, 27de130fd1b0a7e113950fd909bd7ad0ef5285
                    fdc2a5e098f8038018a62af4d3, 464edb5824e1a70d5d493e98b
                    ccd23abf619476e659c5de9ccf37506c48f5e7e, 005f216a7074
                    70dacafc62ac2e5a99bf9fa42373a8c8f7531d667e4ffdd65650 ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "06100cc974c305ddf7c74e7eea17b9df62237f770e916b19973b73dca07
             10aa9"


L = "10"

proof = "a9067eac3cb7bafbb54e639890583ecef49e646aeec069a8928a04cdc68ddb0
         6236678db77c07f400fb6882ab5aae6bf93f1919bfece83649f9717f5976e75
```

```
          a4d35a168ec81cd09a890fbb0ab83625b2bf5cb37e707bc7b1867ddb42538de
          5cda84e99d1a905304d8e94455054fe91f90bbf4387a65328b370336fdb75b6
          f274f0096d61363c58a07a2764254cf0f7d22acd95eaad1ff589bbe83de5905
          44156221867a61a1e3b386f84b2073e6f9d9b2b6a60ffb0efc6c0e3e80126b8
          00e7fe577f0817313bcd3e1bf1b494bde4134a035c9f347e6a98465e551975d
          1451c35d45856c2f3949165c8d929c3279e8fc92bd3ddb0e515845e18bed266
          a92cf820d9c0e3f39fc83f07aa47ad4674caabe307a5b5dd551830561162d6f
          e0b05793444a9fdd36c9fadf8d21057537654ffd5016f8d5d95fadb17884d19
          c6a0f5ec4ae14280f2a6581a99fcfd7ca292a71cf21885cb85f3390b28a7f54
          7f2ace9e44f49da38993bfb00734aa5e1117f2f452c06100cc974c305ddf7c7
          4e7eea17b9df62237f770e916b19973b73dca0710aa9"
```

12.1.5.3. valid all prover committed messages and half signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
```

```
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"

revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
1: "a75d8b634891af92282cc81a675972d1929d3149863c1fc0"
2: "835889a40744813a892eff9deb1edaeb"
3: "e1ca9729410dc6ba"
4: ""

Trace:

random_scalars:

r_1 = "4b323b402acbd18f2109d611190669d66af7d4edc0ff899d308b77cbf54a1629"
r_2 = "4224bfcae26523b6a34fc37637e74cd83cb9a61f78b6be0b15414993ccc3f057"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 03da9778016b40f396a9f237e56c3ae5cfc6d5001f328f3db98
                    448e5e1115408, 5863a317cec4996d232afdb379aaf96cdb14e1
                    02fe00b1667b1aa68feacc5099, 3f043505ceae96a3db8f12615
                    52bea2053bfd21a872408ba8e19d0fa3201dc74, 726be9c9f6af
                    8407403c731b1a7282955dc1e563ce08b7d5705ed2cb20b04040,
                     0bffdf4cf02093b2473b6d129636b21d70542048989bf92471aa
                    f7a567e5ee0a, 46782785ca90e9b6751e9301a320c5b1b299a20
                    c1989b9ee0706525dfc7ac45a, 71e38b786f98c5c77b1dac4f08
                    8aad694408882745fa24efa020549e81091d1c ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "190e55a6926442a97ceed5c8e44fbc4e77110d91280c8658896e5d0eb9c
             1ab8b"


L = "10"

proof = "91bc88e3ca7d2f784c0e9a049a2c28edae23a9b23f8cfbfd04a61bf615f39fa
         dd079cca6b00eb864775f579042cbc52a8431670805c77cae96b9b859c7360d
```

```
          2aec6f177dc2a7156738904f82497c043dd87ce896b9513290e346a46b8d2f3
          0f983e4d3ae58965da888bac7e9a43bb54c8e8082eda0fea4f8158dcc9adccc
          209d8531414f9da9afaa50887f79c4de772e6559f6cd242865b97e5c5bd7cdf
          1de6e94b6efd70895c8778977053aaf7f8bd60c2fe70a00563b17dda2edf57f
          7119ede5423f8d8192ed1038b04ae070bb5db95c458052c07a909effa2fd858
          22876457e4ba4d2f6f8a5321f2bf81f378ed9ee06bd12afbb65911a5286f8b3
          eba84abbb36cb477bd43b3a23cab561b51ed5fc36477d163dda604560cc1e05
          dcd7b324bc485f9c69c60031a7c9cd543d10631035ad2480597b9f6c56638bc
          adb18fa4dee893f5b898970d0c47673e21325f126261463907e6911126a7f6a
          2fe94e7f0957a77297255afe329d7d4919f05b565a507b9af3d831c4c6c90e3
          fb80385b244d6c5dd20a05e49dcf84677446c456443655faa345d6393bfe140
          b7fd21534d6f3cef8b638395571c72fcb48108c1b907008a8b8d23851d7a296
          ed0722325e3ba920947f695cfe94aa27c56ac60f90545c190e55a6926442a97
          ceed5c8e44fbc4e77110d91280c8658896e5d0eb9c1ab8b"
```

12.1.5.4. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
```

```
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "54cd5cce27924327e7b102c514e3ef1ba9f1f9baad1117804203ab8825dfec30"
r_2 = "5f24156b67c03f1d3bf0b9c6f465b21768a84b0f08a75545d9cae648647cc5f9"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 1ba2fb229bd500bb65ed4fb9c44eff2a0bfb3b335bb28548e72
                   9ca10fdd34849, 0bc5a2dde5a559d01f61a4b291391e3f238735
                   726a723dd69954649d84a47ed1, 07c41f1b4ea44a3bafab37e42
                   76d7585b17ce82afd1dd099071d63677dd49298, 737db4fb5754
                   1bea9d125581faa7f2ccea4de7c4f101456b8bae7ae6bda72ea7,
                    0d92794b70ef2b6f56cf03192b9a3714df9503629081053f13f8
                   d12d9e7d79a7, 269df3a0585467568b6f88abedaa986297f33f4
                   aa069f13cd392ec49426866b8, 4a6c5ac852ddc2ef9f735647f9
                   59416dd8f2b78c56580dc1d4bf57993611b970, 1f167fd08394e
                   520dba66007180fbec6ea5b88ad87a4bdc21acdd80ccfa3e5e0,
                   3a870cc60dc5b1083cdd510fcfda6d23c8c9bde556584ddc9f703
                   78eb7515b0f ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "677faf57b5a9705e49685cbd34dcc0beee1439f552d1509c77eef55b91f
             a6545"
```

```
L = "10"

proof = "90b73086a74ee38b517bf6ebecadcfc2b2ea15b92dbb8debe6eb1417dd525db
         380f824e9b91377ff0af9bcd939d835b0accaaa10d0348bae5c05e14ee1a46b
         cfa62678d8927ff3ba52c2a5fca2fc8dac36eb44c657d9112b48931dc46a1a7
         deea3f82bce989c2c49bfbd8c8a3eda5a5302e606bc59da89ae9e2d9a56f2f6
         ab682a8ac453c9563f62fdd9cff3924f8a24373f1f9fd1fef2149c8207c1252
         b6fa1d235e9841921541da7285067507fe3f918e43f003d76d9ffafd1eb92ff
         7a42da802a578fc8c3e8f75b4925f7c9ac679c46cdac8861026c89fb20eec85
         b2872167ee146b0cb2d50418a9e01d724c1234446cb0de1023459c349c748ad
         1396c1df2670a2b5091b9e01671d72463c0fe83947acd2e6e6719e3c6566d23
         91d995bb2c0523b8af810845a90f9356f86acb0b80784ec2181757b8641c786
         8723f5dcf1dedf6da73669b1e61832b201955789d618e34c3c05230565e6575
         969f68c60c2cc3204fa9fdeab00e71ec986d431782766a2ddeb078c7fd2eb58
         554e0aafef57ddcab973c95d8565bb54847553a39cfe001f79da96ad6a42d60
         5ddb047009eae14e3e64f0742232dff7caebd6b62334601f86308ae01476e3f
         efe1b43069be08ea7c597231a4e9c56e9f460775a60a2254673a5fb004c8b4d
         04c8309574f0e9aca607edb64d3722d6a219e219c71067952cdf7e10ce295a3
         b7353aa45e536c7cc17025bfa75568272f1582c6643aaa6e677faf57b5a9705
         e49685cbd34dcc0beee1439f552d1509c77eef55b91fa6545"
```

12.1.5.5. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"
```

```
header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"

revealedCommittedMessages  = {}


Trace:

random_scalars:

r_1 = "3225bff980c15ebad66698a30058f01c7f1dc6c6cc2b959e974c11f2a1aecc15"
r_2 = "57ced773ccd372849c758db968e99c36495478640309379d6cab21383a965ce5"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 60154b04a4c98255e7aef7b9d08452445b613cd407cbe2ae9d4
                    2371e89b9a95b, 70cc1da93eabc0785d2847d89eb73b723e34c3
                    005acd39a41ff1e3954a10fa70, 19521e59845a027b89750643f
                    be7779c70965a624b0be492eef6f9f305442184, 4cce86548146
                    37741ad543a7e8844b6e737b35e7e3c5830cf171fb5bcb8e461e,
                     27b95c59ffe0163b480e802a8d713f5da75a6ead37d5cff34047
                    70777518849b, 3d4295c7724de36f339340e98216e78bfefa4be
                    8172e6c7c3bc024868c8aa8d7, 1e0f8fd41c7562a915cc9069d7
                    7f91afe2a6dea1168e06c53dc2c992f653225b, 3017a9528ac1b
                    f01ee7fcc37477ebc88239be178b953c7165b15bb76998e7653,
                    52b7d08b79d82945295d25de22e7a604c2967c910458eddfbcd9f
                    c60e09169ba, 334ad48e9043501e7c8893ab351f4091b2e7f869
                    25e814a69375d3b9fc6aa520, 2844fcf0fce4ac751997a092e92
                    1aef4c121b3fac72c16213e4f214315279fe6, 365dd49379b57e
                    ddc85c5665b2f4f1115686b4d94674f710d76eeaf91156c6b4 ]"
```

```
domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "55811031fa2473b3b8d5b72513a0edf519d91d961b468056514f5f997d4
             d8f0f"


L = "10"

proof = "817a20f821e7d168b47a21814beefafe5d782fea324653d21b1a61f32d46cd6
         4e25a36be1ccd8e7efe9c1242eb3c3f2aa799d99c59225f5f3bd1d8311ee0b3
         aae7bc4995eedcd38f48f91aecd310a4a21455220dbd703af459111b96b521a
         24595d59bacbe43a2b93cb90a4ac6d67da6fb5e121d1fa7ddc9dfbba670b891
         1e6a2888bd82dbfa48a2db892764467072f645d320f0c6785213a982b3c192b
         80ad345a656326127eb3d654289d9ca51ba601c81dce32e8c5153f1a50aa0f8
         6c7a96a2e150f4f585a0cc86ca452f890508e05ce208461984887ea0449953c
         06088cd060890a19a18276b48487a43bfd366e6501fe7c831fe7a97850fcffe
         6a6796581781b42f33e8c86c89c4c7cbdd01b82f17a251cdac6e3d30de4d60e
         a708901f5b6c2bddeeaea77abc772d5df66cfc38e60d64402c33e4dd1accc79
         ff74e57a473536477000b51ed64f46bbf696cd4f9f24cb8d94439e498cb7e51
         d181e34e7c9e1a1e398cd81361355d215ea1f12332443d8850dec1c792bb1c0
         bbba52c25e686d3be3871686ea5b1e3f805774cb122b5c81206df19a77355c8
         56b8a2c7fe04a628c943be502d5a0e12cc270f43004df1b9132a67293045a5d
         c6442b9661b07d2d1276603c3ad8cb05ac71271843f1dc43b064d2a17497253
         10abc8722891f8fc9c8cd4959cb21be92666de03cbd86501380eae938c12ab9
         1375daa5174b11692539788fbb9641e448a593a272f763a201273c6b84f3c78
         20dfb7908a2b42406a132944bd5b101573ac596ce7c0065ec446425a4e9210f
         658349dc22ae72cdff870328d4a5dd30891c47312314da2d762ab4c3982bdc8
         1f3c08322ed77bf6f5122e87131efa7456d7070f73e2222d3ae55811031fa24
         73b3b8d5b72513a0edf519d91d961b468056514f5f997d4d8f0f"
```

12.1.5.6. valid half prover committed messages and no signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
```

```
                        f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                        45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                        9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                        833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                        fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                        59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages = {}

revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "13c56cabf679ffcd59c41f3fe18ee4993ac4accba74037e01aafc5c0e79d8524"
r_2 = "2ad00067178fed38af012a1db49065e0135f7514acb184b15adc2f3bb3439c68"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 55cd78757fab58eb75a8260d4a990233ab996a1be803cfec107
                    1981b037fbfa8, 5f22ca14b29e8217c2d80bffadba8fbe0e9a72
                    b08676d8805f494b01ee0a322d, 18916a46f8255442283f13390
                    2d5f16f4cd583f6b8e68b4f2651ae52792485af, 39c6828e74bc
                    304f475a70831781de4be5e876fd5a34a34df553666c14c1b600,
                     54715e8092608f37043d85c2829eab964d0c8332ee8c0e1d3199
                    1394e367adab, 377db8d23e45a90c39b57b9042584b87e0e870d
                    00002aabd2e2b7432b1ee81fe, 456503806bf43b0766c629e5fe
                    e11956ed46232049e7a4fd29ec9f50212ceb88, 1c79c4fc9e4f6
                    96101ae41ba5728a14a54fef180e92c71471e964c8dfc1813ab,
```

```
                  4c7fe9d2f777bfcd1cc185d8c252f75c4512c3ad796a665d8737c
                  1fbccdf5246, 58e4f89934eca37018857c9967661294a68cfb43
                  12214696a9e060d96a6ce3e3, 64b1632f96a49fb49f00d928e84
                  0709b24f33706bab1d72a9243b6ea011d3fd6, 649304c1378ee2
                  0264a7b95c230b85bac9d18fd379b4e51dec087120e7e53201, 2
                  79980776234a84dba380ae7e83241754b68e55c1fd9ba31b75156
                  f8fcc6cb2d, 2d11b81789c8bfaeb419a1f436dc6b7ebc82e6cff
                  7994973ae549ee014c715e5 ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "0cdc48124e45bceca5c3d6e49eaba2d726948b181da772c74026af8320b
             f409f"


L = "10"

proof = "aa1d03daf3d029fe663162065bf6c56a71261409c0ddf3934e7c2e1969a5ab7
         10f106bd66503dcd453e6f05e2774d4288b6efd014709a1f1f0f4e2b365c663
         fba2683df462a57eb28d46344200b836e7519a7efaa8e8fb88ff47bd75d8064
         01494914bc998341c1bd56721e0626bd0dcdaddbe4b4f84d4798143905b81da
         c1d5c0b2c12d949cb6992c1fc24260b43a544b8ae8c3c955b110f48019a9658
         4a33c77d7f28e05d1c66937187ce001a3b9fb623fa9d8bb811e9d2f7c794caf
         4e57ac152207d204a1d722939d86b8953055171d93177df58988822e5372000
         3aa67d30795c119cc5ada3ecb4f2699338ea2bc6216a3c0fb353d31d3e5b192
         2a00b19986548c2c15509c4c786c8f12347906c90061db567bf996d2589c437
         d026c721eb2b5c7734ed7d538732011c8d95b1e5f4f430ce07542c59f9a7c05
         034512c82ba7b983c05932e83a9c464082896e8e42306809d030622f14037ad
         dba9a7cbe0fdcae68852655e2a6f48687c06c69ae364a67b0152dbabd36671d
         e5ab8663e05d906eedc5b1c19ea8c272d8bdf5dad7fd3f46cf989339c1032b5
         17dc43a9bc5980dc98f1bc9097b7644ed870402ff919d1bf5048d92d6e147a7
         ca28fae3dd5678c1855bded20cef05b21c3a4dac94dd4a26f08ade27ddab9ce
         45fff46fd42954987b21f38beef1a9a16b2776efbb3823142b27a6ce3702e3e
         d6c4df18129f3800017df4909d2f66b9601e66858883f123478c04f9878af7a
         540240877940a22ef39899606527d8f228fe32161cccc095c45f8907e4bb96c
         c7199c56c9dd9a866a854218c1e4733309284ef63aa527c8cd5db80edc85b3d
         e5607a97f406f7fd3b91bd3e84275d0d1b976c8186ef1b23ed032504327d35a
         bb3047258579fc36897696cb88d801ceac40c373f07810b302f141d8d59261f
         a92f513658def452f62613ca374357c3f681e965d6868d87bbcd20cdc48124e
         45bceca5c3d6e49eaba2d726948b181da772c74026af8320bf409f"
```

12.1.5.7. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "818f434f737d58ed13b7cbb53885b7a19fe9b4b7d7dc34d8fcc53ca1bfe
             376bd569053d8733a89b97fed23da4a04833c57ce2b42cfd0d60e1b862f
             7774431e80b0ed910a217f37837ab90a94dc1253bb"

commitmentWithProof = "99efccc0ccd91efabb8821ee33edacb823b1dd999682aaa54
                       f38a9c4585e7e7aa746357b2842d38c008f6d732dd501c70e
                       ed41caf3eafdd4bb6151ce2c0289401c7d13381e7db90137d
                       7aa2a64224aa2499a4548b2654481a2f0dd16d799116fe41d
                       b7b7a5c3ae8b1c64bef6a89a46f5040a5178d2e1126f7f351
                       89f0f6cea3803e679ce92eff73856b164425ac4ff8405a934
                       f65ada8ccbe21558ab66db113662ea17ce0c9aa0280db20dc
                       f79301c61269ddfdbdcc22025b85f7089c4ebebc224a938b7
                       45daae833ac4698d9d32bfa8382b4bbb2679ae232d2f6e8e1
                       9239e6ea919665ea736b45a61bbd0e4f4d7431f3038c3db25
                       833b9a0cc1a7709419ac241fb6f02ee13e51101743f1983d3
                       fa69b5d344b984c48a265ee6a7b0df8450004ceec7c1997b8
                       59be16af624e3da2cf44"
proverBlind = "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147
               ebf22b8"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "15494ae70742a6a4f420106c79ee405c138557385f3f6f7256449d147ebf22b8"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "b04bd002c85e31d2735ee2e6b36aea85147cbf197934f99ae26a7da73b98ebc34561848
 426aded0967e07fb333f79487"

revealedMessages = {}

revealedCommittedMessages  = {}

Trace:

random_scalars:

r_1 = "3073243f29432d46e7f2c4d2a7d87bb15310d624809af1b84187d93d30835bf1"
r_2 = "3bfe24fd593ffdf3d01853eeda787901f8a40b01180d93c3ab47034d74582c4c"
e_tilde = "undefined"
r1_tilde = "undefined"
```

```
r3_tilde = "undefined"
m_tilde_scalars = "[ 167266e78b1e98792faa161d1004860d9513390c951eaa0c38b
                    fdd8139ded981, 11b2f012310127e20e2ba5ccdd795832dae26f
                    337a2ac917246597c51ba0e814, 2a02495eb38ba3648b47a5401
                    07223ec68c8adc407495286e158391bb71830cd, 359e618bf603
                    979957ab4eb1d8a15f3dfad16d2131f74104023a58aa4824668e,
                     0d9a9076dd68a89f31587dcf07975ec95e38f54dcee6a8031020
                    10ef2b7ecfc9, 19762e5c3b76312bca412852d98967d5ec31ad4
                    83db5c4dc54301a48c35e12b5, 0f86ce1283a3b533eb7067a4f3
                    7e3d4f119e56ef94eeec42ca18eed0d84d611e, 29f8e06dfe2c1
                    173c14aa396065573168c89b177d8a2d7ca174ee5d9289760a2,
                    279f39de04874cc0abffe26fedbce6a940fa84f6d66b0ce7ad097
                    6797139b26a, 62ec7df765cd26dec3cdee13c42e24f0f8f8f46e
                    5606c598e82cfe20a46847c4, 40eaf6a1bacfd7d2c9dfd2b5d89
                    f661bfdc5d6a81b381aaf474997312a03fd61, 135e144adc59c8
                    7a6bfea6e571e7c91af3cdc15a97ed12a1171e168f2e2b9bf7, 0
                    844c5e30518175069417b37ff58748e96dcf5e9405d385be3ca27
                    6a331e38ee, 73c0dd04f96c6539deadb2ca61c272bcd154566aa
                    8bcb4ecddf6388f575c6db8, 25fe71c44d29832409b116262738
                    66d4372df93669e76af3d3e284bade276ac8, 22b7f7c40ded108
                    e6678604d3d937c53c7c4cf81e36ff51a89dbf4af573306d1, 07
                    3b68cea44ee1fe91040d61af84e9e665200ed34be657147171c71
                    31db53a78 ]"

domain = "18a554af90e12ae7a81bd511901abfe1cf882387033796cc47df19b244a158
          94"
challenge = "351176a5ed9c6e586d7e2b24c5fcb8a13287dfd1e3681cddf24476a189b
             13fba"


L = "10"

proof = "87162841fd4ce826997d4fb3a416f8522e5200df18c580c5282efab161b500b
         a9d0a132a2b8659633757834c9e7b848baebe49b81da59ee628b8a0be8a89aa
         bffbe245eddbbb7248ed0d5491ef9e2707a19754233fd4de25cff255aa7b896
         ca48cbacbc148a96779b3c38ae79469ca7ecb6fafabfc48e00d3091357e5e7b
         09ae3e3a25d594ff25081cbe472faf1ce03738ea41d1cb076e3c6e47e2693a7
         ff348d4ae67bbd58ff079b9ed4e1f66a637a1150e88a5d70c04770382e8d48f
         b8401ed0816e04ecb0fda641896bd8b3bde9e238531b0805d573aa878e46209
         7cefb974fcc3721d9d179657c75ba59aada49174e06ec4760394c65ef5ee173
         3379ae1c0160aae2936fbb6d86b5dab7f68f6a765c16b431571a7e9cfe11973
         bff0b831702d9f672e7efe8c36e2dbbb66ef54bba3597faefea1b8f1907dc1b
         141769fc84e55cc249f45067e83e8eaf73b3d10948253df71233dde2de3f1e1
         091fd6e68b2cb801efe3b1329b383565c9a30f85ef50e416c1d7640b8e25ae1
         d13f66cba9b9489297f59836cd8d98a66b3c1ed87fb9476e4400f8601a0940e
         8b60a872e165cdba78d1dbcc1c91b792b96aa2d744fd14cb1cd260df872035b
         43618a16df1e9487464592ff46e1a4b17821e34c8362df475c8ae4d625ebe65
         b4e50bde56260facf410974c2dca8b2777f5a79c51404071eb125a069d9ed71
```

```
          1821340dba34554d51a89fddf9ecd4434a3e3a04bd74b1063036fe4d36d274f
          c147fc637c92fd47a563b32dc23a7401504e2a3f6a1e45e9244834d7902a254
          5f88c1684f5398eaf787892ea44db69292df28b790e336deff1844ba30ce5b3
          d2df4910f112e99ef6ddb7978c124cced5a66cc2c79eeae8db2527e490420b7
          d1b0d613d3d92ac44c2f603facd533a77c2634d646d7910a6e9f601da71a5bf
          426a22e32f735837948279b755cf65985be4ccbc88481b9f828f6322b8ebfa5
          f842fe530191fc0d96ab61e9021e7b884fdf59def27bc1f0ade5eb63f01f6d0
          88fdbf3ea6410794c3703e278c4b1f2b10ecb95a6919a8454af1da06c6b1c92
          234017add2cadab0c8669a42296fc9b1b086dc855e593c949d3b2e74351176a
          5ed9c6e586d7e2b24c5fcb8a13287dfd1e3681cddf24476a189b13fba"
```

12.2. BLS12-381-SHAKE-256

12.2.1. Generators

```
api_id = "BBS_BLS12381G1_XOF:SHAKE-256_SSWU_RO_H2G_HM2S_PSEUDONYM_"

P1 = "8929dfbc7e6642c4ed9cba0856e493f8b9d7d5fcb0c31ef8fdcd34d50648a56c79
      5e106e9eada6e0bda386b414150755"
Q1 = "8c6f8b5efd544cb72ffc140a4585031ebbb8f25acb881ff559c42b94b8ba867be2
      3b183069032ea18c50910c9b7d3fcf"

Generators = {

H_0 = "896962df2851d1b83640182052fc49d07e9492347aee5ba8cbbf6414249367a17
       5d3e09f812dc2ff7d22618e7f0cb630"
H_1 = "80562c843a305661c2588da3ae2e3b96a5faab147fe6a58ff456648b42407af5b
       eddc2009b4078288e2a8e6a73d4ae4b"
H_2 = "896dabeba9bb98ef48d665cfaa894857cac7ed41f2c4b55bb64fd318dde0a8b03
       50578d9c37010dd266629d6bc9f8e70"
H_3 = "b3941ebfc0a011c442c8902d1e654b19d184385691aba1aeec60d87bfbedbcbf5
       d23ea3075126842522638613921240c"
H_4 = "8619d28414b303bc8accde9989b3caf9c9036303c9b8178d25a6fcd738b74c779
       b0a27c867e1a7f0e9b80c5cae3f4e7a"
H_5 = "85d6077858f8ad500df7e928a25cd0e5dbfc43aff4d761852d42feea68123212f
       f7d41978280be66e56724e98f776c9c"
H_6 = "adc477c6cf52b8aaaf055734c12dca89305937001e10e34e9007accb374d16d54
       0c97fa3ba026d6020f64319ce8d52c2"
H_7 = "ad24e98787eb7318cf04fd58793e77d41707e95fb5ab357237f88e2c9566400cb
       085748a3593e2f838caaf3a5223a7ac"
H_8 = "879db6d14a3ead2a81763f4909a6c4633cbd4e20c602e344e1f7d9891b517a329
       f72df0b516c4bbfa4f05c6204c242af"
H_9 = "8f01fc380b30f177090dba078af493e76c51e867f9f5f8f23b1ac162149d58264
       b1e262fbeaaa2f26624de592cc1ae4c"

}
```

12.2.2. Blind Generators

```
api_id = "BLIND_BBS_BLS12381G1_XOF:SHAKE-256_SSWU_RO_H2G_HM2S_PSEUDONYM_
         "

P1 = "8929dfbc7e6642c4ed9cba0856e493f8b9d7d5fcb0c31ef8fdcd34d50648a56c79
      5e106e9eada6e0bda386b414150755"
Q1 = "986e83f847c8c3fe1ad9d3efd0265b66268fc80f4add90b3e96192616364016bfe
      73a4005d2d86f841806a3132a0f544"

Blind Generators = {

J_0 = "9536711b4ff6e1038102b1473bd1be23b77ac6e85684662c7f340ca522f4e5fb5
       c02d7cb2c31c712324b29c540c9d7dc"
J_1 = "91611180da0248d8f7279a962c32472fb1e57b21fc41c09e6ab8aad61fcab5bbb
       51a4095aee80b070d8ca1e80f725339"
J_2 = "ac4f5344e62eafd1e96fc95539db6f568a3cd3dfd8c5cdafdc0bd2f95572c1083
       800f0f4538449051dd3aa362d33b718"
J_3 = "b823809960dafeb4d405c95d44b38cf868efe320b3c1d995daee411507e672a45
       a050f0b3a73c0175fa521f549dfac04"
J_4 = "9319ad949d6c5a368ed996732f0a665551604ee4a57cbadcbdd3f538ef2391f44
       ceef6f3509ead912cf64623da7e12ad"
J_5 = "b755da890d37cb97fc623b228dec163a6138489ff382292f608ac7adabe15856b
       74a5bed22364744d076b39cfda85faa"

}
```

12.2.3. Commit

Mocked random scalar parameters

```
seed = "3.14159265358979323846264338327 9"
dst = "BBS_BLS12381G1_XOF:SHAKE-256_SSWU_RO_H2G_HM2S_COMMIT_MOCK_RANDOM_
       SCALARS_DST_"
```

12.2.3.1. valid no committed messages commitment with proof

```
committedMessages = "[  ]"
proverNym = "undefined"
proverBlind = "643a0c0bc86a50e0d8c00bfe6c8debd85373597e1aef6cc912838bf7d
               c376e48"

Trace:

s_tilde = "40e7b7bc3a17cbd4fa61f81728b6f1224a934a34f8cd57000c360f1b30169
           0b8"
m_tildes = "[ 43a77228890e6cf2c297292b8989751a6e0c9713caa592f39e61e23a99
            7321cb ]"

commitmentWithProof = "990c1837a8af86843213e5b12fbfc962efcaf8fd0e5812a62
                       37b91b00a47b5a34714a60b4c365f72b47a4d9b656dde4753
                       a18a8286aca2bf58e8bb9a3d77a3e0052aefc427e5e47b666
                       255e53cfcaa7d34d36adc13da01798b8eb041652a57c3b595
                       ace54ed5eee43370c1697eb5ce996020d88ca5d811c011cde
                       10c6c07dc2f4acbc89bd5652414d5b8823a250ed40b"
```

12.2.3.2. valid multiple committed messages commitment with proof

```
committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"
proverNym = "undefined"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"


Trace:

s_tilde = "4cdc5d3fdbe932953dd181851ebf6c134103666761013ff3db4e6dbe47d39
           92a"
m_tildes = "[ 3aca8b66d624ae8974e93fd1f654ddc5f071c9b026eb6eb116401a4cce
            87d699, 0eb04c03f3571cc6cfaf29f19126d032b85bc1e9ac0af917ec5d
            c8ba61ce2d28, 0824fe0cfae8bdb1d2c88cd0d8a4c1b432a48f7f12e35a
            fe5494400a3caaa974, 05faf4555ddc6450e9f4b26ac7ed56ae57998c52
            9d3a898f93f72406d9c63990, 253782ab563a180dcdb220d0b75ad1499c
            70c8e7da183c2720f313368cf001a3, 58ca8d9150a51f432c32e41bbfc4
            b630333ccd19fd8daa6d581ff651392dbece ]"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
```

12.2.4. Signature

12.2.4.1. valid no prover committed messages, no signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[  ]"

committedMessages = "[  ]"

commitmentWithProof = "990c1837a8af86843213e5b12fbfc962efcaf8fd0e5812a62
                       37b91b00a47b5a34714a60b4c365f72b47a4d9b656dde4753
                       a18a8286aca2bf58e8bb9a3d77a3e0052aefc427e5e47b666
                       255e53cfcaa7d34d36adc13da01798b8eb041652a57c3b595
                       ace54ed5eee43370c1697eb5ce996020d88ca5d811c011cde
                       10c6c07dc2f4acbc89bd5652414d5b8823a250ed40b"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "643a0c0bc86a50e0d8c00bfe6c8debd85373597e1aef6cc912838bf7d
               c376e48"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "8d8c93a08cad41749cbd944e778027984498382efe5fd6a110ff9cc741ae65b1d50
     87d9bd0edffaefa492d8cffc1be3a"
domain = "65f5322d0c1035ffcc8a93ded3cf56ab258257d5169d6cef81caab0cbebe5b
          c4"

signature = "8c184a9844d7220ac2d65ac2ea9319f8a9fbe56e59e58e8c89e4c095a2f
             2c63675c85aa04e368e2f2cd451af94558c390660c636807b1f74412310
             271761d398e7cb48719aaec0d21043cbdb94d45f2a"
```

12.2.4.2. valid multi prover committed messages, no signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"
```

```
header = "11223344556677889900aabbccddeeff"

messages = "[  ]"

committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "a3e9e31869a174bd298fdb5510dfa387362aa26a91ebcfeb2290e75a6eb844a2fca
     f874cd75b74e242e59fc25b2ff5ce"
domain = "347edba7f30d3b0fe44611797091bcfc61c118b246125050fd609ecabef1b9
          08"

signature = "a861c09a27a58197416e8df99c55d6500eeb01b007df418c5871e0da3cd
             9741c3e80e8a83c7ccb2ff697bbee1c22953a4adcc9627ecb16654b4a9b
             19c0346c5d5fa79d20c8b77208f4bc4deceff065ba"
```

12.2.4.3. valid no prover committed messages, multiple signer messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[ 9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4
            a45f02, c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811
            f5fb075f9b80, 7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932a
            a348e07b73, 77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c
            , 496694774c5604ab1b2544eababcf0f53278ff50, 515ae153e22aae04
            ad16f759e07237b4, d183ddc6e2665aa4e2f088af, ac55fb33a75909ed
            , 96012096,  ]"

committedMessages = "[  ]"

commitmentWithProof = "990c1837a8af86843213e5b12fbfc962efcaf8fd0e5812a62
                       37b91b00a47b5a34714a60b4c365f72b47a4d9b656dde4753
                       a18a8286aca2bf58e8bb9a3d77a3e0052aefc427e5e47b666
                       255e53cfcaa7d34d36adc13da01798b8eb041652a57c3b595
                       ace54ed5eee43370c1697eb5ce996020d88ca5d811c011cde
                       10c6c07dc2f4acbc89bd5652414d5b8823a250ed40b"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "643a0c0bc86a50e0d8c00bfe6c8debd85373597e1aef6cc912838bf7d
               c376e48"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "8a2d9aced02797ca4a20dd7655dba6e27a442d482225af27a9ed7da592d196618c4
     1ea235f3774b5656ecd7d3f4813e1"
domain = "0777a5e4e6f3a1c64efe741339dc9c68a50aebaf279b5c0138e70c874e9795
          9f"

signature = "ad0a0326d2d8196fb7942f3d0c5dbdc1d7e7277e5cba6ab3ce6bc979485
             5f2242b1eb198228c78f4aaa20725ffda015438f11e13cd7fd21dc22478
             44c26ce34e82264ca2554ef337648ddd66d75c8cf5"
```

12.2.4.4. valid multiple signer and prover committed messages signature

```
secretKey = "60e55110f76883a13d030b2f6bd11883422d5abde717569fc0731f51237
             169fc"
publicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bbaa8fa1
             36f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632171d91aa8d46
             0acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db75c845d649ef3c4f63a
             ebc364cd55ded0c"

header = "11223344556677889900aabbccddeeff"

messages = "[ 9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4
            a45f02, c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811
            f5fb075f9b80, 7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932a
            a348e07b73, 77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c
            , 496694774c5604ab1b2544eababcf0f53278ff50, 515ae153e22aae04
            ad16f759e07237b4, d183ddc6e2665aa4e2f088af, ac55fb33a75909ed
            , 96012096,  ]"

committedMessages = "[ 5982967821da3c5983496214df36aa5e58de6fa25314af4cf
                     4c00400779f08c3, a75d8b634891af92282cc81a675972d192
                     9d3149863c1fc0, 835889a40744813a892eff9deb1edaeb, e
                     1ca9729410dc6ba,  ]"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"

signer_nym_entropy = "3d40961fce6c09eec24a371322732932503b458d7a4cf7891b
                      daa765b30027c5"

proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"
proverNym = "undefined"
nym_secret = "undefined"

Trace:

B = "8df28b59593bf5e65a4c3785c0bddc06958b18ae9376bdc2a973c86a9c91c3dcc6d
     6a8af8391f21f6352285df948123f"
```

```
domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"

signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"
```

12.2.5. Proof

Mocked random scalar parameters

```
seed = "3.14159265358979323846264338327 9"
dst = "BBS_BLS12381G1_XOF:SHAKE-256_SSWU_RO_H2G_HM2S_PROOF_MOCK_RANDOM_S
       CALARS_DST_"
```

12.2.5.1. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
```

```
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
1: "c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811f5fb075f9b80"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
3: "77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
5: "515ae153e22aae04ad16f759e07237b4"
6: "d183ddc6e2665aa4e2f088af"
7: "ac55fb33a75909ed"
8: "96012096"
9: ""


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
1: "a75d8b634891af92282cc81a675972d1929d3149863c1fc0"
2: "835889a40744813a892eff9deb1edaeb"
3: "e1ca9729410dc6ba"
4: ""

Trace:

random_scalars:

r_1 = "63a03d9b47b688aae279fb7a53bf1e27f3be6c718e5bc7c3388bfc859a3fff92"
r_2 = "00dc935c7b744dd702e49d388a587e7a974336e5392cde39d84a0484c6a0c895"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 516259fc7668cceb1f7e68181daa3697df471f8354cb22ebdab
                    01e00336441b9, 2f5e1e67418fb18d0d7359dac450ca67ce4608
                    defa9c80d7edcf85278a929130 ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "4774b210d213ba4104d0ba1c8c7f3161509084de363ca223708af76ec5b
             51e10"


L = "10"
```

```
proof = "b77dafdcaf6bc6e6340c2584d3fc2ba43fb3d3efa04a4f08d89f8fb7f5dab12
         454eca51f82b9723765bc3a3d95ee8a1eb5dc57e7bc4496e516261f7e0a010f
         954fb3d3e221518676eab1fc433a131e82437332e57fee380cf8adb1e440bb7
         23ab007dbe2eded5167f4e8f15f5c7dbb4055ec573db2a171597410a903dfca
         a61487297127a14faf23885abe134410b09602697599007d6bd605a15fb8b18
         ed7172be7dec6468204af3ba292f64406137309ee11b05a2e4517077de9851e
         f5abdf21df6a1566373a8cc967445fc89b0922714a1779efcdb259ee9ee71b9
         f6db592458c570279ad9c284f5ba3e400f8793f50dbf5ac0d1bc06a8ea48c4f
         92c8b4931441340042876 20d1aea4ec4e26402cc5e0fee3a11e3a5054841896
         f5bc0e590edc4a27b4c69caf8c628b96bdf002fb14774b210d213ba4104d0ba
         1c8c7f3161509084de363ca223708af76ec5b51e10"
```

12.2.5.2. valid half prover committed messages and all signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
```

```
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
1: "c344136d9ab02da4dd5908bbba913ae6f58c2cc844b802a6f811f5fb075f9b80"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
3: "77fe97eb97a1ebe2e81e4e3597a3ee740a66e9ef2412472c"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
5: "515ae153e22aae04ad16f759e07237b4"
6: "d183ddc6e2665aa4e2f088af"
7: "ac55fb33a75909ed"
8: "96012096"
9: ""


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "37fd1e468e9836a372c056fcc1aaf2a0faf7e3192391d2119525cadd1a21cc84"
r_2 = "3bafe2de79f657cc90b66b65c43af8ab0362841b2b5f1358aad2c861266b12c3"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 5cc47d2ef4f1fd7acacbd9f21708564c024e71d63f0a1dc9cf4
                    2844654777716, 17db681847018b04fbe5cab8c5a3aa6c902961
                    04a0b7af985f8e9ee66300ac7a, 0ad1b39daa6b6abe354992dd2
                    601e8997c23a593c116040800e1e9619d675a7d, 3557de528565
                    c4a0d75aeebf63bfa459becad78d8b2aecdcb289adc047f2b083 ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "1a5d0bb33157806e5a9d60bfd28c3ee2d08231b228fe08eb26378729dcc
             a5c10"


L = "10"

proof = "a6f95f58c37762de5818c98f0432f1cf878ed4703e6653c6a008dd5b7ec73b2
         b9f82b8075716ed55172c3904c44ba819a6e6238af6bfd430c208c7b6c2ad45
```

```
        a13bf5df561dc23fbf385cbb51ca80689f4505d138739f3f7ca95b6f1a473be
        050a20c55d9a0bccbbbbb78a52f7b56938847b97504f29a703a93e23b11c014
        ad90de3dff1ee97bb4c14bdebd33b8a874252a3ee50e8e126064127a7080591
        a00791ab767a9e3431321e5bd88169813958746901f84c263f4505196cd1921
        48bee47a493871941cdd8a18cb3aac4d27f54f43b92ce82b4fc8b9a99b45c72
        187d77d86675fb5387c4f24ddbb802df1902c0d09ce610dbfa9cbb93f0edaf3
        1bff55691d4c9f6f133ea6e2bf3e547eab929f0902a6b33123823637ab1e87c
        33daa9db1ec8064b7332963142abdf3af8a4d96da7171e87fc8b73f1a6a7982
        95e2414b31249019ecba5ff78b41c1a0d1fb004f176adc5df643264cdd4434c
        e30058a5a3cc298b2f6cc8ab1887d253735185acb8c1a5d0bb33157806e5a9d
        60bfd28c3ee2d08231b228fe08eb26378729dcca5c10"
```

12.2.5.3. valid all prover committed messages and half signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
```

```
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"

revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
1: "a75d8b634891af92282cc81a675972d1929d3149863c1fc0"
2: "835889a40744813a892eff9deb1edaeb"
3: "e1ca9729410dc6ba"
4: ""

Trace:

random_scalars:

r_1 = "7313e18d146909fcb0768f65bc7b40d1470bf6c7eaaf32cabe9d4efef57b4d6c"
r_2 = "43241d9968e2960809d9f4c7b0caad95903aeb236889723182dd66808b1447b9"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 69e1449113b7cccb77270a88081447496f292f3fdaff5063125
                    a2d4a2dfe90d3, 175efe5d253a12dcae58d30b190b421b43eebf
                    c11a1e0a375179712c053bbb5f, 334b6f76a164abfad24247570
                    01a06c32e5780ee1b46dcf641c08753d26731a7, 3d49a5055e25
                    4a8f7bd0559d7f8af12c61e12141c3582b938dbb1dd11bc485fe,
                     4d1d2808c702b371de6834b4e9d3b89ed246c06ded576a9b52b0
                    ba9e969ba4c0, 31e82a5bff25fb098a63b15151a4a4cb0ecbc8b
                    0ea2c2f90d160390600c4998b, 5ec90143423471c547ef946907
                    b2f7d52c2c8aac3444090191eddbde1557c2ba ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "385a2515cfae58be534fdf1d60a897ba22041c3a97eeae3fea8cbd14e03
             0efd1"


L = "10"

proof = "a60ac69442fb739e4e3bbfdfd6538a199c5a194f01c60b748d7df295a990d7c
         300c7faacde9157f6c78f519c3fec0167b432999917d646de19a0cc0ea4b7dd
```

```
          9ee7eb3488e0f7173386ea6880a4bee3751cab8cd2f3b0648f1de650715e46d
          569ac09cd3b1aa69af38ed503859f47db3e6458e0e94b28eecfc1f0989e1454
          d87b2ed5c516fe054424f8348f13678b61fb263c60463633fc23ad684269fb7
          f76e0d9f81ccd977ae04fc7c73d89d3d95b3e4e7c0ccf488b44b3ea5160c14a
          2a472a7ee796b1089d96f1225f49bbafeffe752fe86e6f9df6fc4de968053bb
          adda5f79e0531cdf48a8dfeadc33da2e364414f454146aaec6437b360d59dc1
          b70b66521dde547dfaa17027919cbe3b714cb331295e6048786b6e84b8ed0c5
          53ef90dcb94bb48b9cabcb34cedf50ff72036076a6b965cca0d68aacd2ec050
          843f82bedddcc27839aba9e1c07824b6cb2332b10b2ef6920d693b8b632d63a
          4b74fbe98d6a90b2a6be8d5bed00acd2b9459407dd4271952f75f9b19c73266
          a098d450613c7d2d979e18efd53357389674981dd3772aea9b94ef7cc998323
          fe782c9cae3930c5a4fc2b200b4d9c55ced0b76a02cab6348d850ad29aec666
          fcd2d8d445ff4f41d7bbd4b172df0688d909dabbbfd201385a2515cfae58be5
          34fdf1d60a897ba22041c3a97eeae3fea8cbd14e030efd1"
```

12.2.5.4. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
```

```
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"


revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "645e79d9cda4c43c9379b57bcf0fd3bf91551d7d3506cdc5ddf0ecd6f1d62ef8"
r_2 = "4651e4a1ca07e8b9027be1593efbe7b3739c689f4fa35f15e08bfd6e1fb403a9"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 19eff867cb51d34a5dcfd09842babd26246fb0f31e9d6495547
                    dfca626ea6218, 4a453094c95378613b7883d8c6134050eac731
                    55047406d9d9bc14c13f1ca43c, 3305b0eb13b29fde6433e5a7e
                    6397973304320e4538d530718865003691fdec1, 20d1e7865a12
                    778a333d5a8629250f552b62a8b5ee72392d8dea199af3d99af4,
                     0fffec53443eee11097ee9f1ca1629f0ada14f58e834027bc519
                    285708978a2a, 1f4bbeb4fa9afd52feae084cc8957cfcaf897b6
                    056b57341fabbd7b2da0bc8dc, 30172f3663cc33948d1f6d290e
                    b86e39ed91feebe4f89578056b627513db4388, 2354405c3a0d0
                    a2836fa077ac742c86fe908c7babacaefc915af6249024442f2,
                    0fc44c488c657f7b3b299769713c611a1b8284750c085334b0334
                    5865912dfff ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "2347f6c339f712af4456b3488362d6e9700ec0fcfdb73f49acdc1061e43
             91f61"
```

```
L = "10"

proof = "a1f1f6f051a401e823b8fa554a007db04db7140071735d7f9a9c3c5378ffeb2
         5fd4566a2b752048136bb850dd322aec496d1f2dc4c0285efbb00145e8a9dce
         4481b9b1497b16ea12a2e6679874b734d5bca1e997f9f64a17ba9333b330e53
         1188870f64ad87d45ab86d56e4972f1dd55bc534b575759cebf75413de2ac0e
         91befc5278db9d99bbfd039ce75413a750693266fb28b35f69a9a5215194279
         43cc1f22df6d111d1ba8e6df00baecf94ad2d63ee2520eb718e73a10c0d79ad
         205dadfb2d782ba75b931a522f0f2122f8fa903f7ec8266e8ebe4bd3c82e848
         12b25570b462f415e7db6436ad82729fbcd0ee65f9b83a3b0bf0769090e24c5
         8e4c339bc12e81ed66433faa3ff71c6a2e7080876729b9dad11eb583fb8035a
         4182e5aa80193c77124fbd8153e99dad48fd65907095316313883b0726cb99a
         473f2e9c9d61e99193b3b476cc40d0902e1f81841439c571b7b38167986872a
         abb205a10eefebf36a2a003936d263cc2b3eded8fec3968433472652bcee7f1
         509dba33684bef308d9a560be2e87b0c4048248bb2c038852cfdf4d3a9b525c
         1baaf177f94b5a17b482394bd5688a47215096e5e4b08167adab1c231c05dae
         e205a69b805a3658311a7c3f3a229e398781196f1dd1f336476d6c2783564b4
         16a8bc7351f4c968083da142b44886b783ce4a97871a20c5f3c2a69ddd09d4f
         721e7e6f3c2ce64f3964b18de02271021a7ac266594eeacc2347f6c339f712a
         f4456b3488362d6e9700ec0fcfdb73f49acdc1061e4391f61"
```

12.2.5.5. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f014945746871b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"
```

```
header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages =

0: "9872ad089e452c7b6e283dfac2a80d58e8d0ff71cc4d5e310a1debdda4a45f02"
2: "7372e9daa5ed31e6cd5c825eac1b855e84476a1d94932aa348e07b73"
4: "496694774c5604ab1b2544eababcf0f53278ff50"
6: "d183ddc6e2665aa4e2f088af"
8: "96012096"

revealedCommittedMessages  = {}


Trace:

random_scalars:

r_1 = "04ba5026d055fb29e48748ab435e4b61aa60fc480a826be9cdfa9e2aed3648fb"
r_2 = "1fcac43a8b494879ad6717566080583245f11776fbe555bd974fa99b1dc68353"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 637517351b2e91be4cab8d8cba3ce605d6b5bf4709fcec17ab9
                    c7302d188f7e4, 0d2857e8286db4c060f05875b34159b7a3b234
                    2868f4da8e421f1a1362a07030, 40af4c858f59b35c74d2b3353
                    9e29ba7ac642cc90482eca5615b3c915cf9f0f9, 71840e81d029
                    4ac5b3ad0ba9c0b0136f078d8cdac595a64f0d9e63b3fca8a2da,
                     22979e0e072479f80a81fe43c7b5ce6c21c0064b6039d8637950
                    212a7b90c192, 3671cc511ee5093111052d78af04270df346315
                    f6fe3f9a7a32c8f5a92ef6d4c, 1e096373f281d8a87f63a3ff0c
                    925017cd825f2dc97d34f39eff08362e4db6c6, 44ea9cdb9c05e
                    4b5119f2500f4648481fb07d12a1cc75a5b2957e95367b7d681,
                    4da405d28b04edc0adb8043ec95053543d879a067f27b1051dc27
                    e268340d77a, 1ceb3358e8e4617ad361b95563cbf3701a9173ed
                    00b5e303c207879c8f294523, 6d216a3f24966d7baa9049cc320
                    765ef18bfa5162f2d60cac425845ca11f60f0, 2d679ebc3e75c7
                    2b60e7ba485503789542e5d20f923ddf346fea3399269c2e0a ]"
```

```
domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "001f8fc7911ddbd368e4dd2272368e711f85b0e8ce84f3446647e188d01
             bce2b"


L = "10"

proof = "853619679818ae01ced0c601c52146f85364d3d11e8e595217b83a733ac9253
         a432b36b6ee574f450e3d3107d31ccf90828c11ae50d5717871d58e6abe796c
         32517943b470b8e20cdb92adb6237b5d2e08e1ae8b6705134e1229d7e317447
         6a1b709eb010bb73c2429aae3aa1a321192e477644a414e991b56a139b03a57
         b123d366af06658e2daeeca583da28a405b16f8ac6221a9372f8f556fdf422d
         e90d9c8449d995c8208be5bd02d3a497418fa1bc23021af3f9b2116d8461838
         5daf295498aadc3b892bdc128d8ee32b1477830ee1d78409b23d86ea61fc07c
         3959d8d6d4894b6f446d2fd0181338a47d802484abbe9d59a397d9839ac45c5
         cd22ce82d8cd125040cf919ae967fe6d35943df1502bf1a576a34fb5fde1174
         a7e556cb6bb5747f93415b9b06a5414fe6cbe569f3cb44eb2d36f0d07bed0f8
         c1e3bf552cdfc3bbb7ff795653ce1c255233ee4d0935af29032545ffdbd3ec4
         007b803668acf79f1f06d015eecdbd52a08d0d2e17a545bcc4c49d5a89125e0
         40a69d07e2b185041bc3c143729370a06fb768af94f45a031020b992ac9d530
         9ccc8a88bb76296ce8575cea2cbe7e675f61735bf7dd86538088c3c7dced1b4
         d6d3cfd5731f8a55f6c4d918aa92811900aea61744a01b1aa0ee25baa2286ab
         af5d44b69d650ff005c5e59cf5df8a895b73bb9f2a3be5507ad690c76c77016
         67e21650b3ba0c9122c23de8f930825ccdf5cfdbcb44a95e51faa0f9a390973
         bd3553cbd4a9980341627fe13243c5a4cc1362f64a896378c2af7f12a3daa1e
         cc92b4b426fdb7a22f2846168b90389efbbb75d2ea8e3e345150b9adaab4a41
         22a2fc7e6d0cf0608f1daa1ee8f7d727654af3e2b99f05d17d7001f8fc7911d
         dbd368e4dd2272368e711f85b0e8ce84f3446647e188d01bce2b"
```

12.2.5.6. valid half prover committed messages and no signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f01494574687 1b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
```

```
                        07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                        507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                        c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                        87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                        f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                        2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages = {}

revealedCommittedMessages  =

0: "5982967821da3c5983496214df36aa5e58de6fa25314af4cf4c00400779f08c3"
2: "835889a40744813a892eff9deb1edaeb"
4: ""

Trace:

random_scalars:

r_1 = "2782e9be32645456d20a01a8ce5dd233b8049efe8d25eafdb8576f4e8d85826a"
r_2 = "180ea80200b800d1cb672bc1aa52e2e6ecf0acaf794b810f3c16e8b07c413f5e"
e_tilde = "undefined"
r1_tilde = "undefined"
r3_tilde = "undefined"
m_tilde_scalars = "[ 11f580b98813f609f6099d00f35412ae67aa1aa5c35cec654da
                    7fd04eb023616, 3d33bc719b4b5ff24fe54b281a361858912006
                    eff78184deffbf6769ca87b9a4, 2a2fe081ccd47b8a669e86232
                    470b13a5157931bfb3b770dac1080bcccd28d81, 6d0e8353ed62
                    e613e6223334709ad0e146efb9c522633daf44840a2b5424195c,
                     4a1742ed19312608a1095a476a5fc2533b3df79d065c62494f7a
                    0fe9aef2f455, 58d11e68c4ee775d999f69bc33efc83c7557423
                    420b48cf915d772c975c9fcab, 50886691287e055accbec2696c
                    2fb20e9072d704461dc358261d30116f78eca2, 080126141ef0d
                    e78e1454319dc179a72f95d98e8e1c5912d71a09f6113bfd6a9,
```

```
                  3983a595dde792593983b34198422b2d2f13215cb4803b510cd4e
                  59ca0297f0f, 34c8e1e41b5718d9dc82232ea9fd6895601c5366
                  dfdc65a1f58e9b0bc40472cc, 6f9ea2c849ece4b9e6271e0e017
                  75fc358e18b8c0974bbe3fdd7d58a04b6d4dd, 3eb917340ca578
                  8ce9266b29b984756addc2e9d9574e3140ad90ee1a1e668e37, 0
                  5aa05f0dee70e3b5905f5d894be11d37a2e1bf98a838130a5a136
                  968d8c4811, 4e66ec908e4379b5298f7ab5ab8a3690677297690
                  444183674a694ad3a0e5639 ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "25e05527d98eccabb98ac160f65f2584dda742417bc6623b2d85fcff118
             a9ecb"


L = "10"

proof = "a7ef58d02fc849635b0e64fb23ccbde361e4f76354c34f75b53c3ae88612e42
         52accc3944a59293c6473b04f76acb1578b40f25f0d5878c8419e3d20c60527
         92f7b99ded10e8651135c39425c4d461020eb6f8014d372b0ad7169b83dd15b
         5db861c5d86ec8d0b93e8f7e57df7126dfb4f4c26c9e46b41b508bd20a48d76
         d0a3171623aec71bd4d77dff14f2752c26ea123a501f6f78d2f7d5390923a45
         dc2e9d906ffd986bef6c23538b4abbd9d6a7b3799600723f0054e45a0f0e630
         fc74d90452cb0808e46420c2637c9fdd039e072929d62a3ba48d11911d6140c
         8ce06a7bb7799efdc01a4b1b92957f60a2c436a29593b2e0cc9ddc36589538e
         812a75f505c8e7bb8e09fc2ed95082a4033560922946 8b8c7feb66fecc4d27e
         1df788f7c8a130318a0476beeb935708c4193e264296dd2ee994f6c1c312059
         278603601a910129d118c3c1ef74b843182eea1cf123ffd5184ef146a11cf91
         b2dead002d5c761d028a1c7ac54fae2ed0ee5d9bbcb12407f71040bdb1bc5fc
         f565607c40b7ad31c45ed961d0a5cd186922164c4d2e14dfe14d3d792329560
         abb657b480dd45128972420efba5be67f954a7cf2b6b6737f1ee9d00b89c216
         1d7fb844819bbf55a0c2c7e392b9b0d7a51fca325070f55fa240e1a683ff669
         b83308ec9731779b35fbb9e7e8dfc5157dfcf1168b63ab35189b6657eab0fa2
         c3667d22e3e84de8213536289569089b75f007e6774b54a742feed5f6d30cad
         ea9896f79b219debbb93a4a5ee81f878def938a4d6256ea336dd8c3cd2fa656
         423e39ada0c8411adb44694c0ebb61d8ad8790d10e35c51cee49f8e626cd345
         dd242f7d8c57909754a5596d46fca25fcc15fcd0c9ad69d80b5028e638af418
         343a578b6e9aa2750a24633fba37c48746422233366361c813a75e941030c7b
         13261a2b82642d21abdde098f1ea86df00d25c245d876f2089a8325e05527d9
         8eccabb98ac160f65f2584dda742417bc6623b2d85fcff118a9ecb"
```

12.2.5.7. valid all prover committed messages and signer messages revealed proof

```
signerPublicKey = "a820f230f6ae38503b86c70dc50b61c58a77e45c39ab25c0652bb
                   aa8fa136f2851bd4781c9dcde39fc9d1d52c9e60268061e7d7632
                   171d91aa8d460acee0e96f1e7c4cfb12d3ff9ab5d5dc91c277db7
                   5c845d649ef3c4f63aebc364cd55ded0c"
signature = "b6bb95cb52f5f44ca1ff76ae305b03f01494574687 1b057ea08c2e45c24
             846bccd26f14858afdcb942896630cc16439002f802e700bc2c83347064
             b3ff69bfdc8552119ab13b07b52e233d908f859237"

commitmentWithProof = "a9577c3e2f15081c03d2e86789c1d9208bc04409b1ca33c25
                       d06017c8fef5d139aee028ac96b9c09636a45846e9a5ee51f
                       83bfd55f12193061e3f707d11d9993d6e08293de7f3dd0a29
                       8c21f369208b43b7b401706a9a0a5dcfa12d28d5a59b09da3
                       37b435cf4aa2a869842c8e1409004865ce6ff78d345e5c814
                       2c9c440b677824ce06a8f70c50bbbb01838a91eb0041fd853
                       c2005109d3aec272dd03346f37fc90828490fbedc4fc88e73
                       07662b785653aba1a28a45bca913b7dd778e8bd141652e6f0
                       507c3f836c8852b8ddbf2c62659dbd7b83f096e7b351f2f0d
                       c6046bce3c8d0c5bb892a7a3d76d6bac899b3d356b099f882
                       87ac25e6879d5808f832927c8e28acae41ab3699b5c0f9da4
                       f58bf67d7e87c5ddb6dadd80fe281e158cc7a24bc398f8402
                       2dc0dc3a123971f7546c"
proverBlind = "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06
               fea89c3"

header = "11223344556677889900aabbccddeeff"
presentationHeader =  "bed231d880675ed101ead304512e043ade9958dd0241ea70b4b3957fba941501"

signer_nym_entropy =  "3d40961fce6c09eec24a371322732932503b458d7a4cf7891bdaa765b30027c5"
proverNym =  "undefined"
nym_secret =  "undefined"
proverBlind =  "1ade8b27cccac993dfe3d57be0cd1a200a5cae52d9ea525f106c94f06fea89c3"

context_id =  "bbb4750cdce6d2122bb4c4f039b6ad5a79f028eb448013a38636a95d63af360a"
pseudonym =  "8ef7b8516387badcdf24eda35553031d01c392b93fb943445ae90979d7285d877ba6509
 cec3a3520f46128e97ecbd136"

revealedMessages = {}

revealedCommittedMessages  = {}

Trace:

random_scalars:

r_1 = "326a49e2d8d7913701012fee5c5d928455d70a150535ee83ba8f838289023b1f"
r_2 = "056572994ba90ba03e44b30acc945aca792335b27c5f5ea05a0bf9c4c360603a"
e_tilde = "undefined"
r1_tilde = "undefined"
```

```
r3_tilde = "undefined"
m_tilde_scalars = "[ 19c6e4aacf97f53bff27c7de44179fd91380cdb164405b65189
                    94a92e78a2066, 4aa2ce863a88a4ba2d0d3b7abf4faa42ffa774
                    25d1226f8649855d3b07ab0098, 325d5fdfb63d63a1ed4821521
                    d5e8dd7bdbd3a82d033b0d607f9bc2f263d487c, 4ce828b81b72
                    bed30f74f5f32c6c0e1cd744f03faf0ef901a2bdb1d6952fe523,
                     05601ec295d3a2675f3fbce27e5657d1082f5f1c38c4b6e157d4
                    b66c64846453, 6df288dfa867d51ac44ac185aaa92f7e097a1b4
                    05178d88502066298e2e35e49, 1bf8f8de4f3a15233bb4cf3d9b
                    e23a5e022b0a3b55a492e84dfbbc093a0d0459, 1f501b827c4b1
                    3e3381e26997894aa3f4559062bee1c56d2f79535934c07ffca,
                    46391415dde3df672c839ef8b3d8316d01b8a08265041872f49d2
                    67788851990, 2db4ff5a39bdedfec9bca340748c55b825d9809d
                    d2ab58dec9f27d9b54dde60d, 07ad541e1dad6534c92c5707d16
                    8a5c08dc4cb916b22b63d174b78261fcc0728, 60d1c9f4285a4c
                    69a900d97bd22cc3788272d0dcdec5f9e1e6d64be10a01a0ed, 1
                    d21cf6835ee4d6729c0520135afa68e3125430a97d9502dd86e9d
                    dde148c298, 687c44782b9bdd724a3402d1e929c9f74febc3cf9
                    b91a55431790b3460b6f02b, 09abd129e4fb8d7e3a817bd65f38
                    9d53c3eb8df45d03325461756c5932d32a8c, 2cf3011ae8b91a1
                    4c5a058a43385885324cca7c78da2aaf1b2cc15cc751962f1, 15
                    4099a6f2a3df481fc6bc052b7ece804de2acdbb34b5cc945570a3
                    0416878cb ]"

domain = "28ef090980cdc152a3c1e56f778a09a54eeffb4117d051892df580f38e362a
          fd"
challenge = "6da88a0785efb7ce0d795834a85acdcf4e3a0ef413f4c613229ed96ccc7
             3cf93"


L = "10"

proof = "8c5508a7262fe9aee86720f7f2ade66289179468b872e4bbe854a94c27dcc0b
         8af2b8a0f6e2732d110de3c8bb7a24b84aea28faf71f94682feeaab2d6d8eaf
         17286cec952f0b406174cc63839f9ea993d8a0dca5ac772ae8f2b5841d1d477
         7ad8c0f87e5108206ee1c2ecfbfcf563827b591c2585116fbf6525d14498db7
         75f7e1e3aea90f0c4ccff01ba7a2f6f415fd1adb6ab79b43201239cf0c14f8e
         e939518f5040a27819eac726ea42cc0ecc6ca4a2aed4f95933abc68ef640151
         bf94037e78335e88dca0cf305d158483c5db37605b8123f45f62a7d01b1ea74
         2354f3770b826d434057caa769585334677245511334642462b86a10ec1c5c8
         4e2195750eecff0d7bd160791ee608b27505506e5ba6b212844dcca750485af
         44b2dc667b71a638e54c21f0dca1c47c18d3c9201303ca80d4b9657ce1a040c
         ea234843ff14104fafaba915ccffe8fd6e9fdfa3e91eacae0706718c4184874
         f14f16346488c234196fe5d348aa86513eacdfe78c04f53ae31b7313d8f285b
         91b9af2e673a5fc9545235bef034c1caa6958f28d465168bf5e4806f2c29597
         9da5cacfa0ad2a1b4d17416bde66807a3d9ccd4706b71508358a587dc912a7f
         5cc8d96e9e62a6defdcc1b357ff88a3c74a999ea9c340b2fc32c03cf4b45d4b
         37690d9a0c3321663fba714d6be1865996e2edceb7532590f046d8120577958
```

506268ba726bc4d1b6783daeffe1d751601a7b63d1a8fce6101ad733b6dae03
f33d6426dfbc3883eddb021c79fe4284ac5837a83a098b704253bea009182ed
77c891892ebec571bd06c13f2e3f64197e599614811f1e450a53a084cc27b33
9836b6ca13f6be62bb73e66a6160298caf3db3ffe11db48cf2a28f89433aca1
0bee9850e0c13d6cbe8730f4ce01f8c8c3799e4910b67a1c0d5b3a474900524
718ad0b8def6de1df6d6d4b27571986b04ad3e20c5fc96603beb31b6c2d1d5e
1fdf5ad0f0f4a0ab571a44451c1dca1f64f4b51345e2b8352e9a380f58d8f46
5d60c552fb9f7d1babcbad9dca348f15adb7a5dae5d12b4c0d433e81e6a98ba
53f70624e433bcabe1d22466ec44db7b620684cf87579d473c4570346da88a0
785efb7ce0d795834a85acdcf4e3a0ef413f4c613229ed96ccc73cf93"

13. IANA Considerations

This document has no IANA actions.

14. Normative References

[I-D.irtf-cfrg-bbs-signatures]
Looker, T., Kalos, V., Whitehead, A., and M. Lodder, "The BBS Signature Scheme", Work in Progress, Internet-Draft, draft-irtf-cfrg-bbs-signatures-09, 7 July 2025, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-bbs-signatures-09>.

[I-D.irtf-cfrg-hash-to-curve]
Faz-Hernandez, A., Scott, S., Sullivan, N., Wahby, R. S., and C. A. Wood, "Hashing to Elliptic Curves", Work in Progress, Internet-Draft, draft-irtf-cfrg-hash-to-curve-16, 15 June 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-hash-to-curve-16>.

[I-D.irtf-cfrg-pairing-friendly-curves]
Sakemi, Y., Kobayashi, T., Saito, T., and R. S. Wahby, "Pairing-Friendly Curves", Work in Progress, Internet-Draft, draft-irtf-cfrg-pairing-friendly-curves-11, 6 November 2022, <https://datatracker.ietf.org/doc/html/draft-irtf-cfrg-pairing-friendly-curves-11>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

15. Informative References

[ABC14] Bichsel, P. and , "D2.2 - Architecture for Attribute-based Credential Technologies - Final Version,", 2014, <https://abc4trust.eu/download/Deliverable_D2.2.pdf.>.

[BBS04] Boneh, D., Boyen, X., and H. Shacham, "Short Group Signatures", In Advances in Cryptology, pages 41-55, 2004, <https://link.springer.com/chapter/10.1007/978-3-540-28628-8_3>.

[BlindBBS] IETF, "Blind BBS Signatures", <https://datatracker.ietf.org/doc/draft-kalos-bbs-blind-signatures/>.

[Chaum85] Chaum, D., "Security without identification: transaction systems to make big brother obsolete", In Commun. ACM, vol 10, pages 1030-1044, 1985, <https://dl.acm.org/doi/pdf/10.1145/4372.4373>.

[Lys00] Lysyanskaya, A., Rivest, R. L., Sahai, A., and S. Wolf, "Pseudonym Systems", In Selected Areas in Cryptography, vol 1758, pages 184-199, 2000, <https://link.springer.com/chapter/10.1007/3-540-46513-8_14>.

Appendix A. Acknowledgments

TODO acknowledge.

Appendix B. Document History

-00

* Initial Version

-01

* Editorial updates
* First set of test vectors

-02

* Added n-use unlinkable pseudonyms against a CRPQC based on polynomial commitments
* Updated test vectors

Authors' Addresses

Vasilis Kalos
MATTR

Email: vasilis.kalos@mattr.global

Greg Bernstein
Grotto Networking
Email: gregb@grotto-networking.com