Single-use-seals and their applications

Dr Maxim Orlovsky,
Pandora Core AG

* Single-use-seals standards and reference implementation are maintained by **LNP/BP Standards Association**

What is single-use-seal?

- Form of applied cryptographical commitment
- more advanced than
  - simple commitments
  - timestamps
- Requires “proof-of-publication medium”, i.e. medium preventing ability to “double spend” or “double commit” to certain data.
  This can be a medium with global consensus (like blockchain) but not necessarily decentralized (can be a centralized printed medium coming from a trusted party).
- Proposed as a cryptographic primitive by Peter Todd, following his development of timestamps

Proof of publication medium by Peter Todd

Proof-of-publication is what solves the double-spend/comit problem.

- Prove that every member *p* in of audience *P* has received message *m*. *A real world analogy is a legal notice being published in a major newspaper - we can assume any subscriber received the message and had a chance to read it.*

- Prove that message *m* has not been published.
  Extending the above real world analogy the court can easily determine that a legal notice was not published when it should have been by examining newspaper archives.

- Prove that some member *q* is in the audience *P*.
  Using newspaper analogy, because we know what today's date is, and we trust the newspaper never to publish two different editions with the same date we can be certain we have searched all possible issues where the legal notice may have been published.

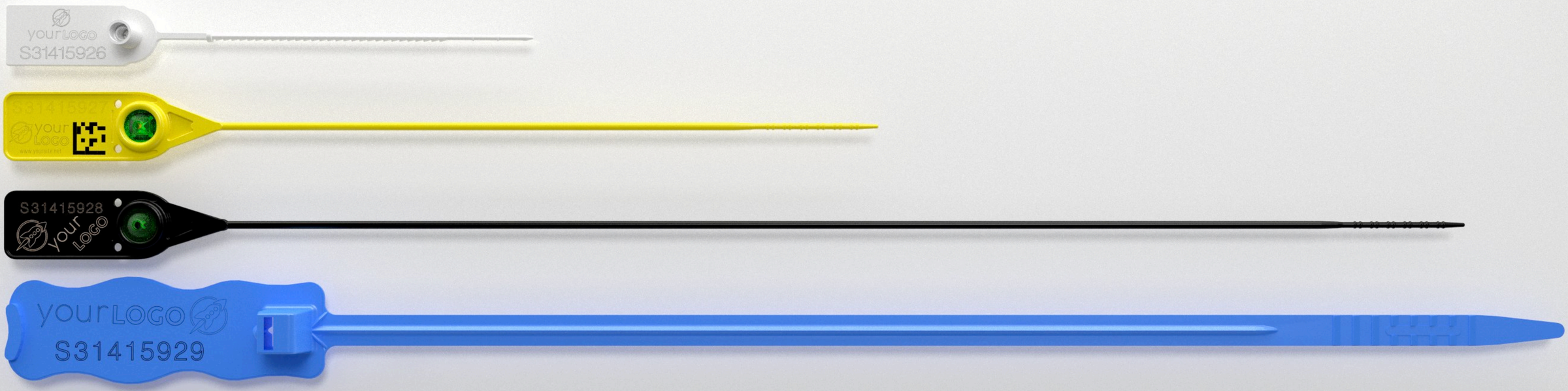
Single-use-seal is
a commitment (formal promise) to commit
to a (yet) unknown message in the future,
once and only once,
such that commitment fact will be
provably known to all members of a certain audience

Single-use-seals vs other commitment schemes

	Simple commitment (digest/hash)	Timestamps	Single-use-seals
Commitment publication does not reveal the message	Yes	Yes	Yes
Proof of the commitment time / message existence before certain date	Not possible	Possible	Possible
Prove that no alternative commitment can exist	Not possible	Not possible	Possible

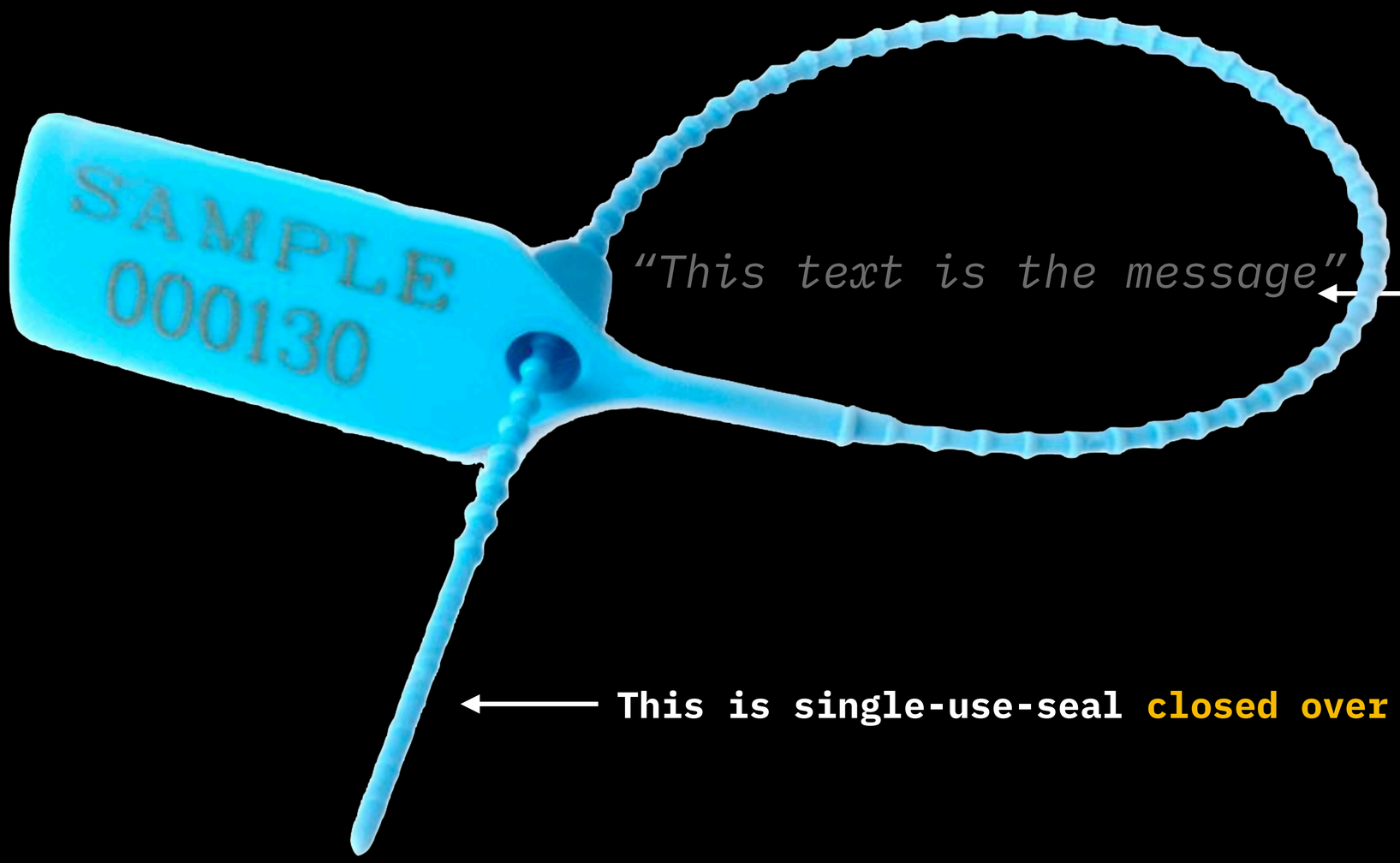
Constructing single-use-seals

Intuition behind single-use-seal terminology



Container is the **message** over which the **seal** is **closed**

Closed seal



This is a **message**

This is single-use-seal **closed over the message**

Two-staged process

1. Seal definition:

Top right corner of “Old World News Paper”

That will be issued on the 1st of July 2021

Two-staged process

2. Closing the seal

The seal is closed over message “ERAT ZRIL DISSENTIET …”

The date has come

Single-use seal WTF

- A promise by Alice (public or private) to Bob
- to create a commitment to some message
- at the well-defined point (in time, space, or any other form of phase space)
- This point is named “seal” and the process of its definition via Alice’s promise is a “seal definition”

Single-use seal WTF

- When Alice creates a commitment to that message at that defined point, she *closes the seal over a message*, producing the *witness* (of the commitment)

Single-use seal

- Two parties (Alice, Bob), three methods:
  - seal <- Define()
    done by Alice, accepted by Bob
  - witness <- Close(seal, message)
    close a seal over a message, done by Alice
  - true | false <- Verify(seal, witness, message)
    verify that the seal was closed, done by Bob

Single-use seal data structures

- Seal
- Message (any data over which the seal is closed)
- Witness (proves that the seal was indeed closed)
- Medium (a medium where the seals are defined and witnesses exist)

New York Times as single-use-seal medium

- Medium: printed newspaper
- Seal definition: “First column in the issue of NYT that will come live on the 9th of June 2022” that we agreed upon
- Seal: publishing slot of NYT on the 9th of June 2022 (abstract)
- Witness: the actual issue of NYT on the 9th of June 2022
- Message: content of the first column of that issue

"All the News That's Fit to Print"

The New York Times

LATE CITY EDITION

Weather: Rain, warm today; clear tonight. Sunny, pleasant tomorrow. Temp. range: today 80-66; Sunday 71-64. Temp.-Hum. Index yesterday 88. Complete U.S. report on P. 86.

VOL. CXVIII . No. 40,321 NEW YORK, MONDAY, JULY 21, 1969 10 CENTS

MEN WALK ON MOON

ASTRONAUTS LAND ON PLAIN; COLLECT ROCKS, PLANT FLAG

Voice From Moon: 'Eagle Has Landed'

EAGLE (the lunar module): Houston, Tranquility Base here. The Eagle has landed.

HOUSTON: Roger, Tranquility, we copy you on the ground. You've got a bunch of guys about to turn blue. We're breathing again. Thanks a lot.

TRANQUILITY BASE: Thank you.

HOUSTON: You're looking good here.

TRANQUILITY BASE: A very smooth touchdown.

HOUSTON: Eagle, you are stay for TI. [The first step in the lunar operation.] Over.

TRANQUILITY BASE: Roger. Stay for TI.

HOUSTON: Roger and we see you venting the ox.

A Powdery Surface Is Closely Explored

By JOHN NOBLE WILFORD

HOUSTON, Monday, July 21—Men have landed and walked on the moon.

Two Americans, astronauts of Apollo 11, steered their fragile four-legged lunar module safely and smoothly to the historic landing yesterday at 4:17:40 P.M., Eastern daylight time.

Neil A. Armstrong, the 38-year-old civilian commander, radioed to earth and the mission control room here:

"Houston, Tranquility Base here. The Eagle has landed."

The first men to reach the moon—Mr. Armstrong and his co-pilot, Col. Edwin E. Aldrin Jr. of the Air Force—

Single-use seal gotchas

- We can’t prove that the seal was not closed,
  we can just prove that the seal is closed
- i.e. while there is a method verify(seal, message, witness)
  there is no method is_closed(seal, message)
- so we have no concept of the “opened seal”, only a
  “defined seal” or “closed seal [over a message]”
- There is nothing Bitcoin or blockchain-specific
  at the core of single-use seal concept

Single-use seals summary

“Commitment to commit” (a promise of future commitment):

- Alice deterministically defines a point (*single-use seal*) at which a commitment to the future (non-existing yet) message will be made
- Alice commits to that point (“commitment to commit”)

define

- Later on, Alice creates a message and commits to it at the defined point (*closes the seal over a message*)

close

- The fact of the message commitment can be proved by some private data block (*witness*)

verify

Beyond single-use-seal

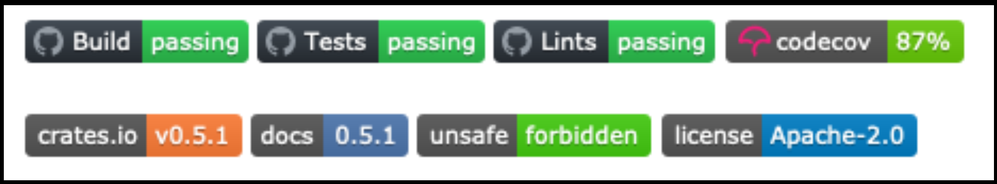
- With single-use-seal you can build **client-side-validated** systems which does not require global consensus (“blockchain”) storing all of the data.
  - This gives scalability and privacy
  - Single-use-seal definition made on the client side, not necessary in the global consensus medium
- You can’t prove that the seal is not defined even if you a member of audience *P*.
  - To fix this, we need a “chain” of single-use-seals, where the message of previous seal defines the next seal(s)
  - This is what **RGB** (at first level of inference) is

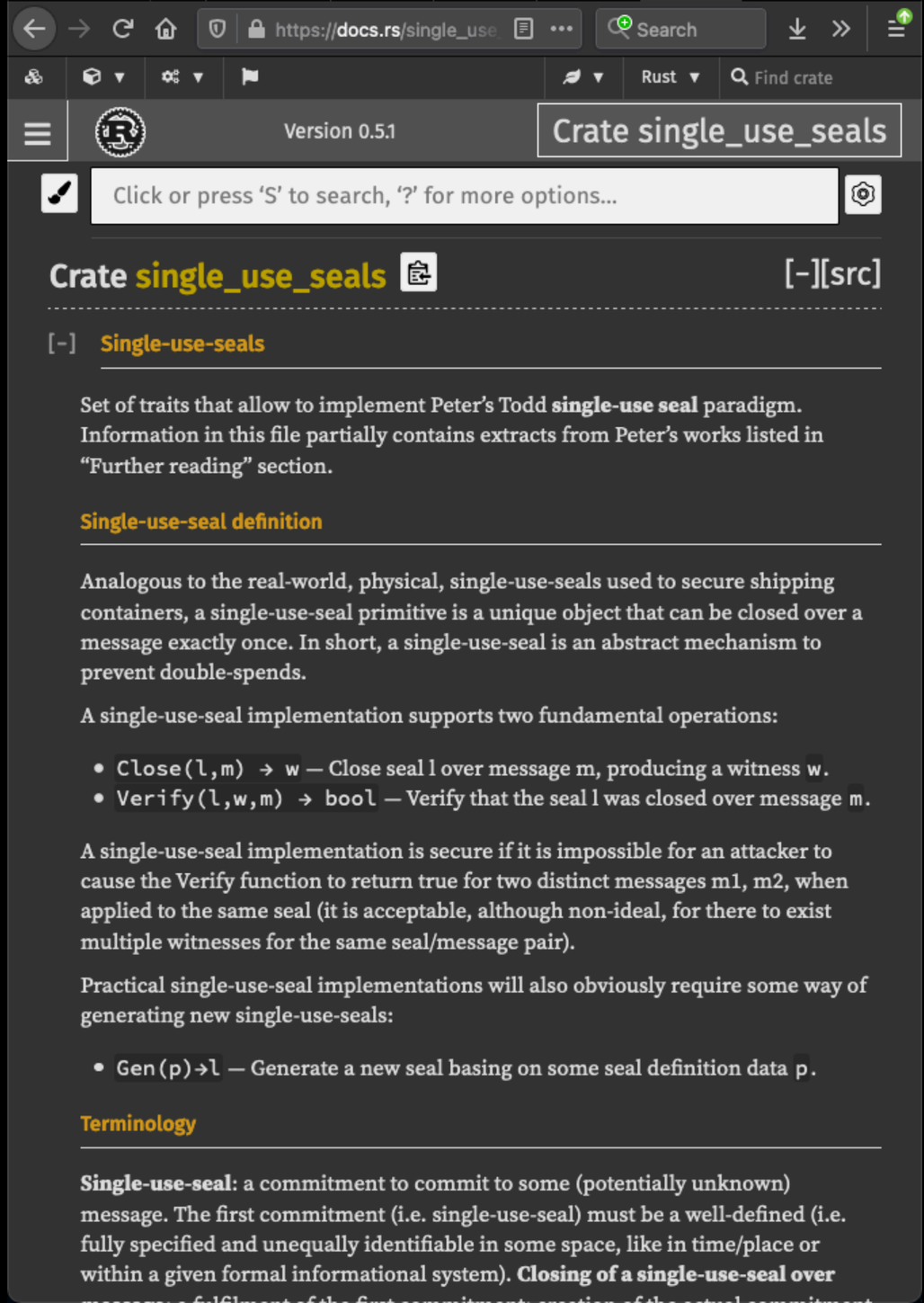
Designing single-use-seal implementation

Single-use-seal cryptographic primitive may be implemented on different proof-of-publication mediums

API library

- Pure opensource rust implementation of single-use-seal primitive from LNP/BP Standards Association
- Independent from a proof-of-publication medium
- Best practices to implement single-use-seals for a specific medium defined as API
- This is a pre-production release which will become v1.0 without changes other than bugfixes upon RGB release



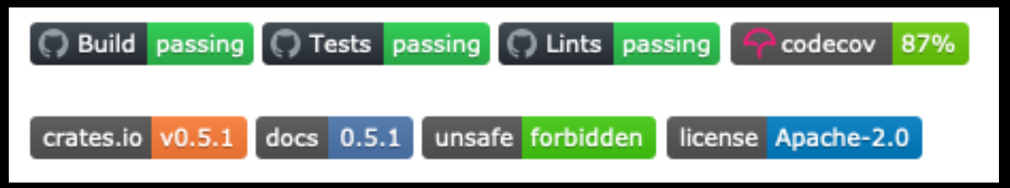


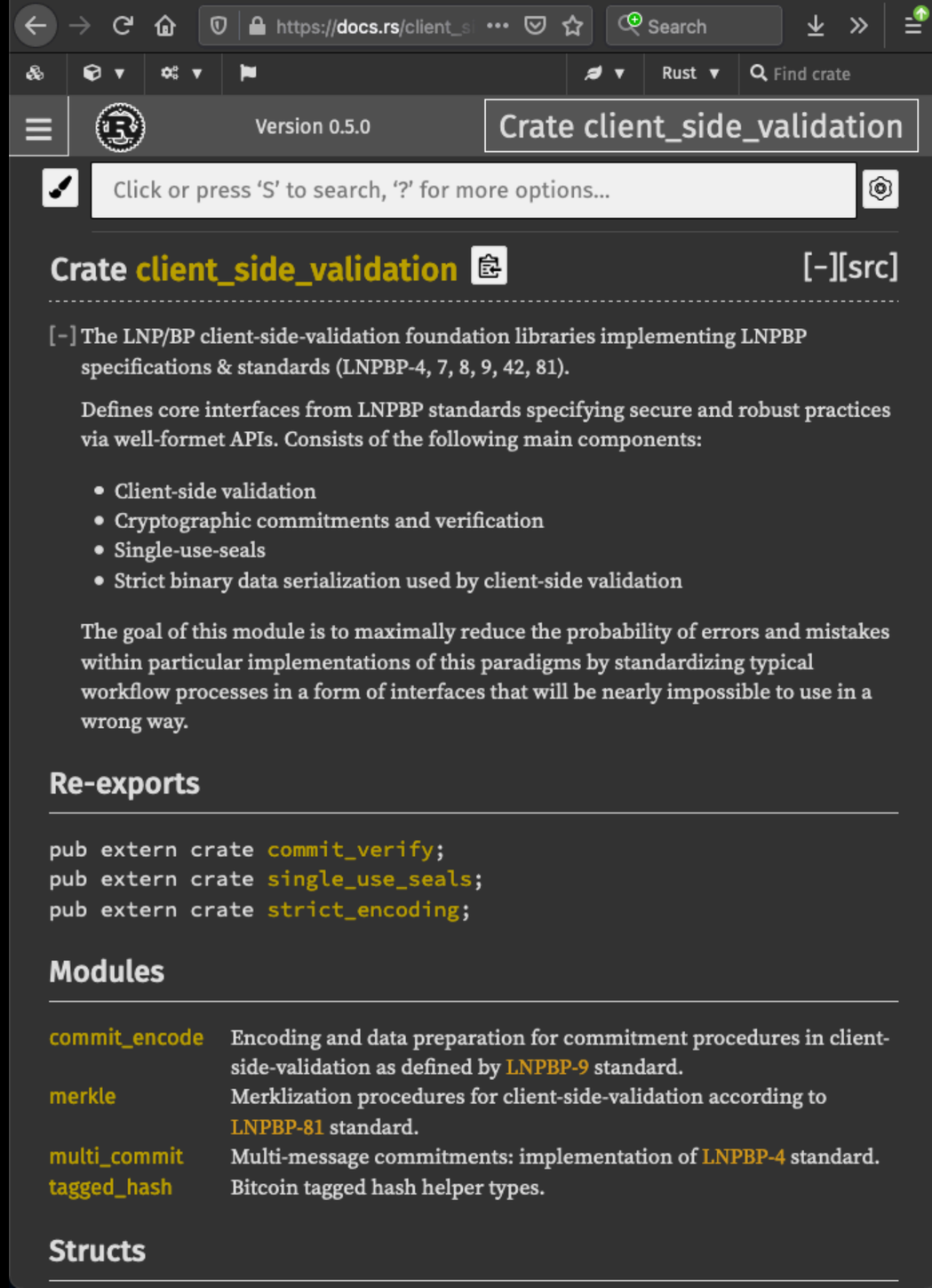
Client-side-validation libs

- Pure opensource rust implementation of stack for client-side-validation from LNP/BP Standards Association

- Includes
  - single-use-seals
  - strict encoding for deterministic message commitments
  - commit-verify generic APIs
  - trust resolution for proof-of-publication mediums

- This is a pre-production release which will become v1.0 without changes other than bugfixes upon RGB release





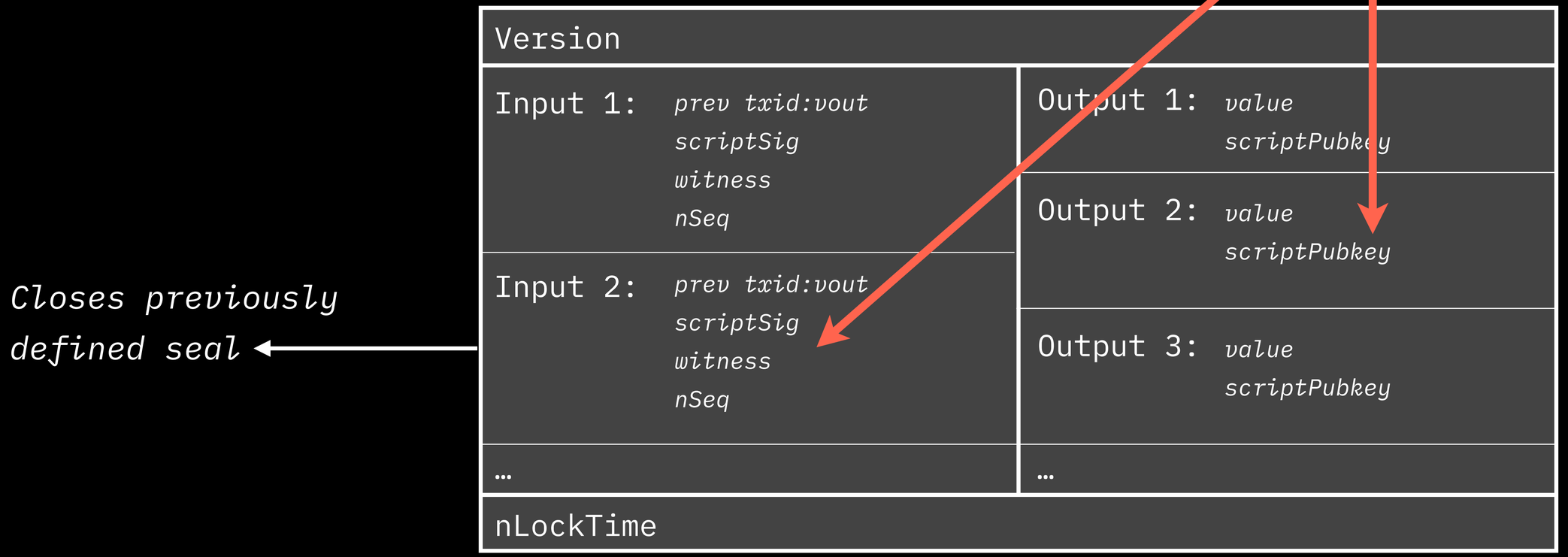
Defining seals with bitcoin

What can be a seal definition?

- Public key or address
  - we close it with its first use,
    commit to new data with signature (sign-to-contract scheme)
- Bitcoin transaction output
  - we close it with spending,
    commit to new data with either S2C or pay-to-contract scheme

Closing seals with bitcoin transactions

Single-use-seal closing message commitment can go in here



Fitting commitment data into transaction

- **“Good practices”:**
  - Signature “tweaking” (*r* component): sign-to-contract scheme (S2C)
    - *Saves space in client-side-validated data (no need to keep the original pubkey)*
    - *Last signer gets unfair advantage of failing commitment without loosing bitcoins (solved by Musig)*
    - *Still non-trivial to combine with (yet unknown) final musig scheme*
  - Public key “tweaking”: pay-to-contract scheme (P2C)
    - *How to define specific output which should contain the commitment?*
- **“Bad practice”:**
  - OP_RETURN (R2C)
    - *Increase onchain data size*
    - *May be required because of some old hardware wallets not supporting S2C/P2C*

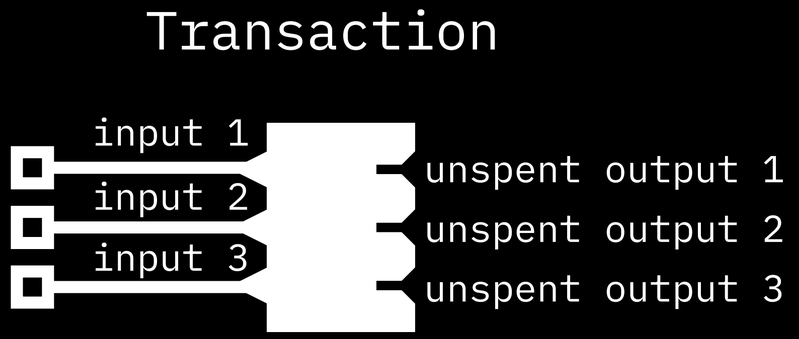
Comparing bitcoin commitment schemes

	Pay-to-contract (P2C)	Sign-to-contract (S2C)	OP_RETURN-to-contract (R2C)
Onchain space consumption	0 bytes	0 bytes	42 bytes
Client-side space consumption	33 bytes	0 bytes	0 bytes
Can be combined with SegWit v0 outputs	Yes	Yes	No
Can be combined with SegWit v1 (Taproot) outputs	Yes, with modifications (T2C)	Yes	No
Can be used by legal hardware wallets	No	No	Yes
Multisig vulnerability	No	Yes, on non-taproot single-use-seals	No

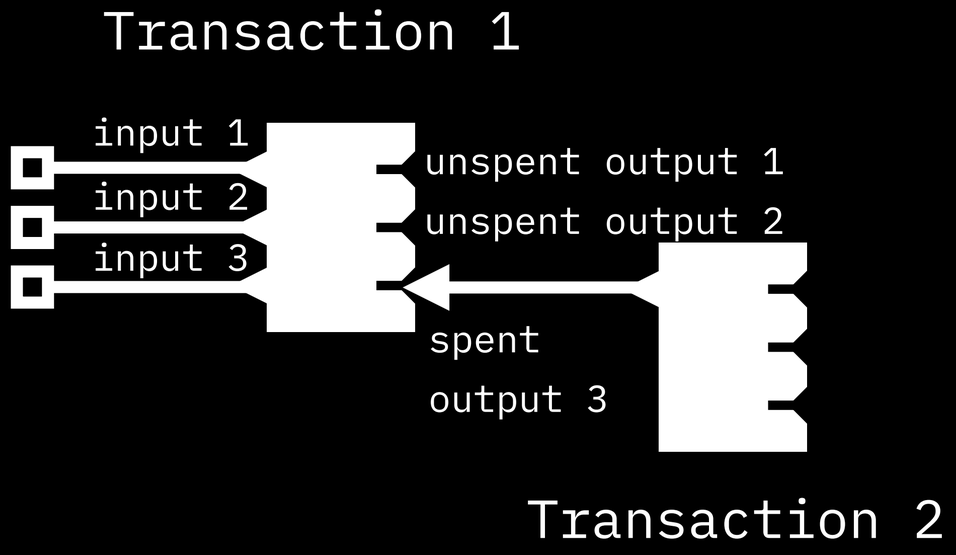
Bitcoin single-use-seal schemes

Scheme name	Seal definition	Seal closing	Used commitment schemes	Additional requirements	Main application
PkO	Public key value	Transaction output	P2C, T2C, R2C		Unknown
TxO2	Transaction output	Transaction output		Requires deterministic bitcoin commitments	RGBv1 (universal)
PkI	Public key value	Transaction input	S2C	Taproot-only & doesn’t work with legacy wallets	Bitcoin-based identities
TxOI	Transaction output	Transaction input			RGBv2 (future, taproot-only)

How we prevent single-use-seals “double spend”

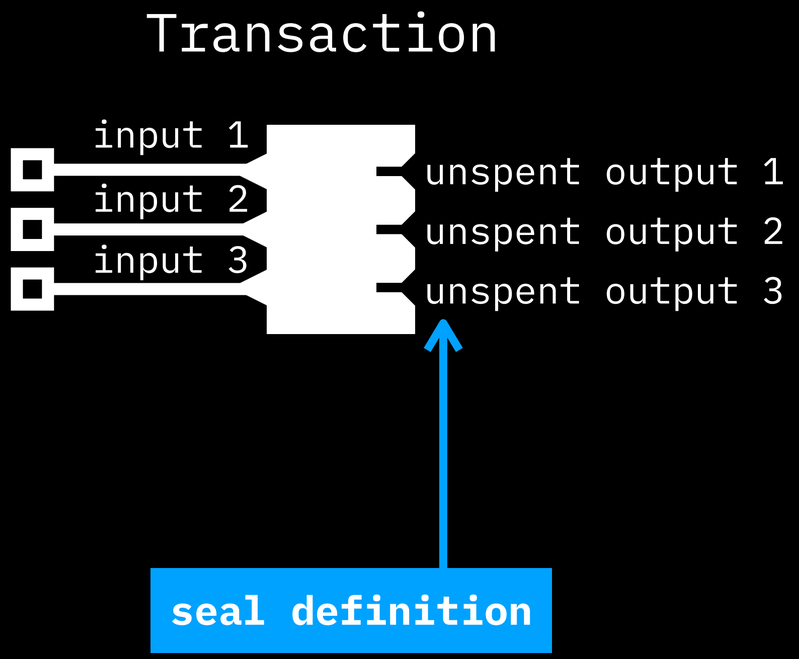


How we prevent single-use-seals “double spend”

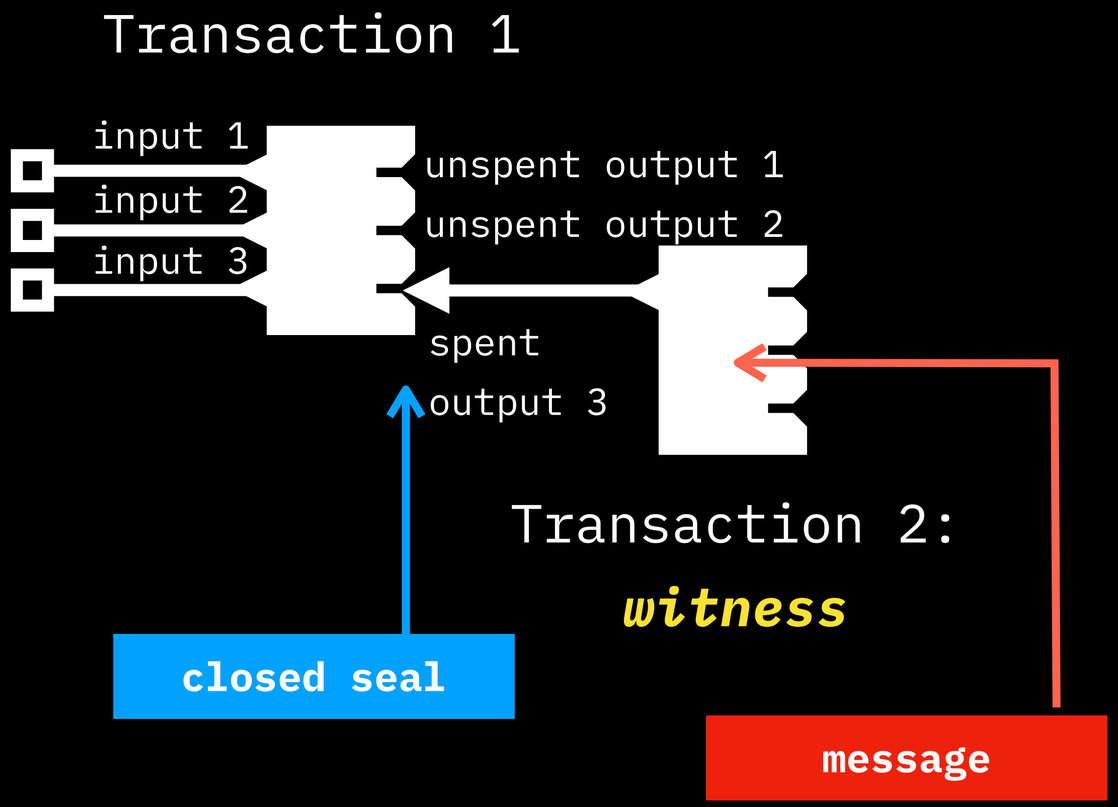


A unique event

Tx02 single-use-seals



Tx02 single-use-seals



Tx02 single-use-seals

Define:

Close, verify:

Public
(onchain or a part of state channel):

Seal: Tx OutPoint

Private
(offchain, owned):

Seal definition data

Witness:

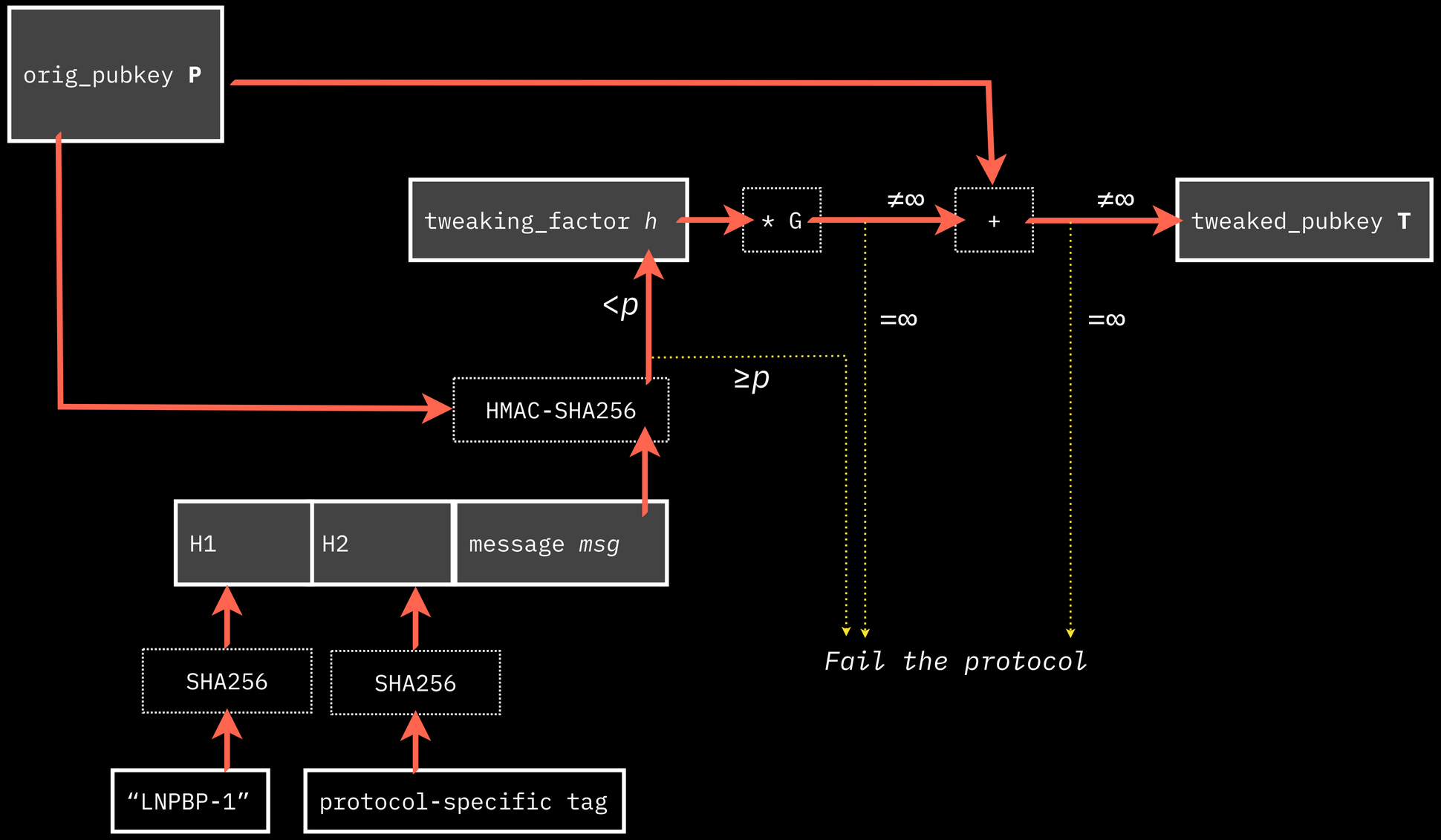
Spending tx

Extra-transaction proof (ETP)

Required only for P2C

Message

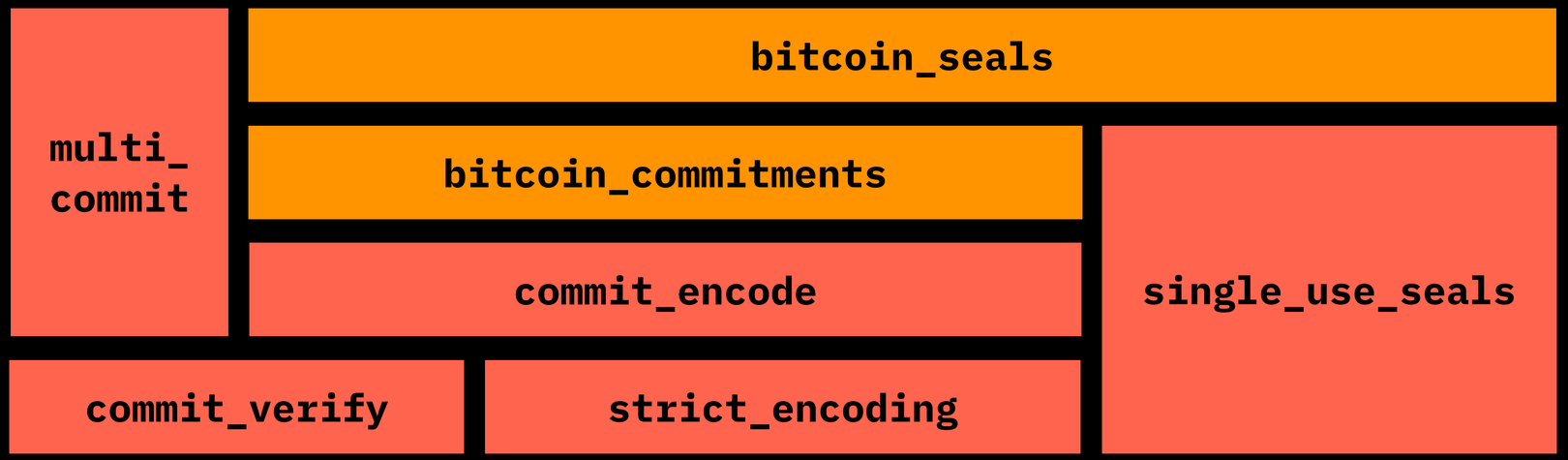
Modified pay-to-contract commitment: LNPBP-1



Deterministic bitcoin commitments

- Required by TxO2 single-use-seal scheme to answer the question: “where in bitcoin transaction we need to put commitment to the message when the seal is closed”?
- Addressed with LNPBP-2 and LNPBP-3 standards:
  - LNPBP-2: How to put commitment into bitcoin scripts & taproot
    uses miniscript
  - LNPBP-3: Which Tx output must contain commitment
    chainanalysis protection by combining onchain (fee) and offchain (protocol id) factors modulo number of transaction outputs

Foundations of client-side-validation on Bitcoin



client_side_validation library

bp-core library

Can you do single-use-seals outside bitcoin?

Yes! But te question is though “why would you need it”:

- for centralized single-use-seals you do not need client-side-validation or opened standard
- for decentralized, there is no more decentralized and trustless proof-of-publication medium than bitcoin

Still, there are future prospect for single-use-seals outside bitcoin:

creation of “blockchainless”/“coinless” scalable proof-of-publication commitment medium

There is an ongoing work in progress on this:

- Peter Todd’s work “proofmarshal” – github.com/petertodd/proofmarshal
- Pandora Core AG work on “sigchain” – github.com/pandora-network/sigchain

Applications

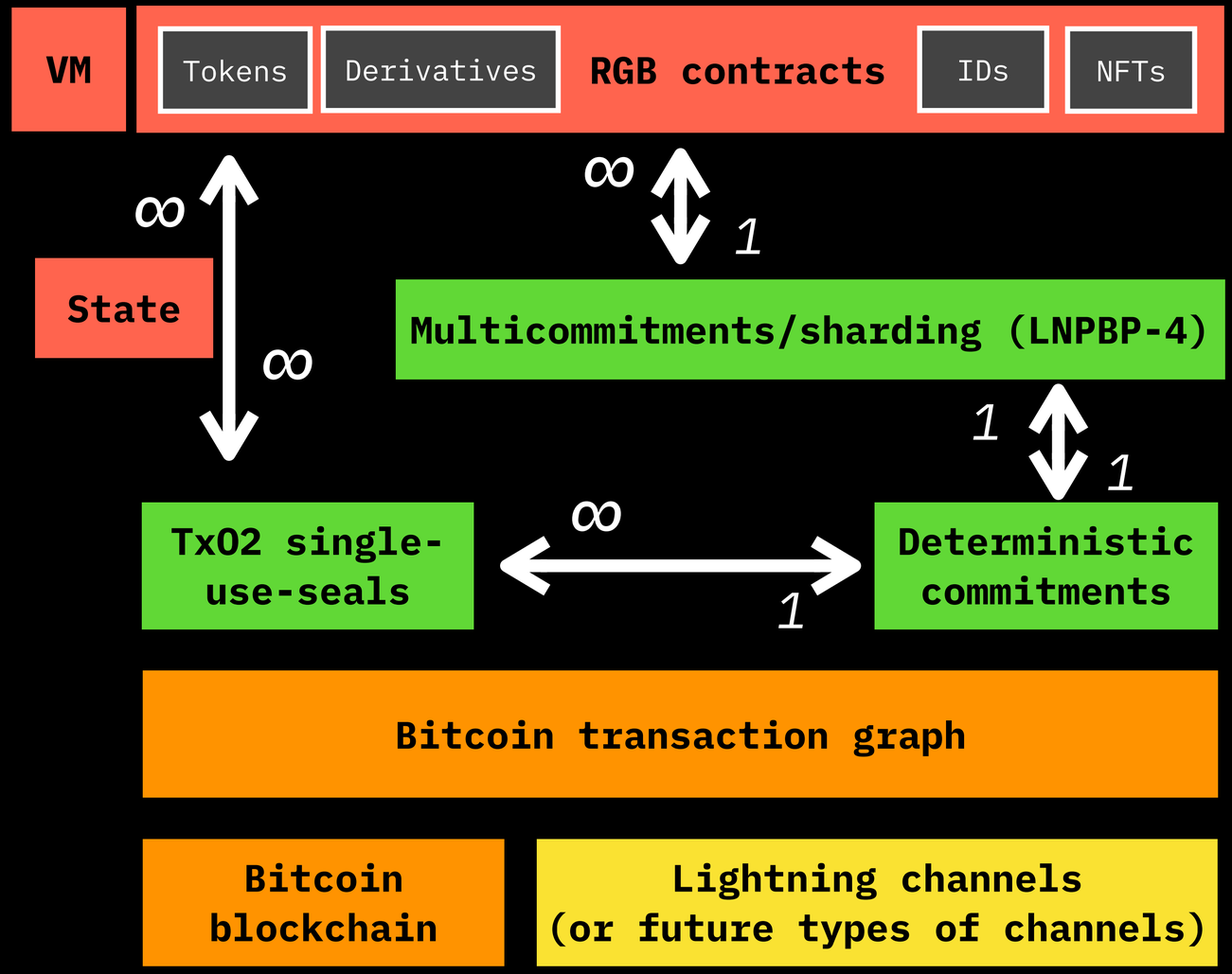
1. RGB

- Confidential & scalable smart-contract system based on client-side-validation
- Works with
  - Bitcoin onchain
  - Lightning network
  - (potentially) other bitcoin sidechains
  - (potentially) future coin-less proof-of-publication mediums

RGB != single-use-seals

Based on single-use-seals, but requires much more:

- deterministic bitcoin commitments
- concept of state
- state evolution schema, scripting with VMs
- multi-contract commitments (contracts sharding)
- confidentiality with Pedersen commitments & bulletproofs (“confidential amounts”)



2. Bitcoin-based identity

- Problem: how to announce identity revocation in unique way?
- Solution: single-use-seals

2. Bitcoin-based identity

- Identity definition:
  extended public key + revocation UTXO (Tx02 single-use-seal definition)
- KYC-less identity authentication:
  - Login: public key + chaincode + unhardened derivation path from identity xpub
  - Password: signature with the public key
- Identity revocation:
  spending UTXO = closing single-use-seal over a message which is the new identity xpub stored as pubkey in the signature *R*-value
  (the chaincode value is given during authentication)

New BIP-43 purpose covering BIP340 & identities

Proposal for BIP-??? standard
lists.linuxfoundation.org/pipermail/bitcoin-dev/2021-February/018381.html

m/ **signature_algo'** / **blockchain'** / **identity'** / **scope'** / **usecase** / **index**

signature_algo'	blockchain'	scope'	usecase
???' ECDSA	0' mainnet	New one for each website, multisig etc	0 normal
340' BIP340	1' testnets		1 change
	2' liquid		2 tweaked
	...		+827166 RGB