BitcoinCore

BIP324

Jonas Schnelli / dev@jonasschnelli.ch / Breaking Bitcoin 8th June 2019
PGP: CA1A2908DCE2F13074C62CDE1EB776BB03C7922D

V2 MESSAGE TRANSPORT PROTOCOL

Goals of the v2 proposal

- Opportunistic Encryption
  - Eliminate passive non detectable observing
  - Eliminate non detectable message manipulation
- Optimize protocol
- Extendable with various authentication schemes

Why V2?

- People did start implementing BIP151
- Major differences to BIP151
- New message structure
- New service flag
- Short Command IDs
- Opportunistic encryption

Why encryption?

- The blockchain data is public, the general traffic of the Bitcoin network is not

- The Bitcoin network is under active surveillance

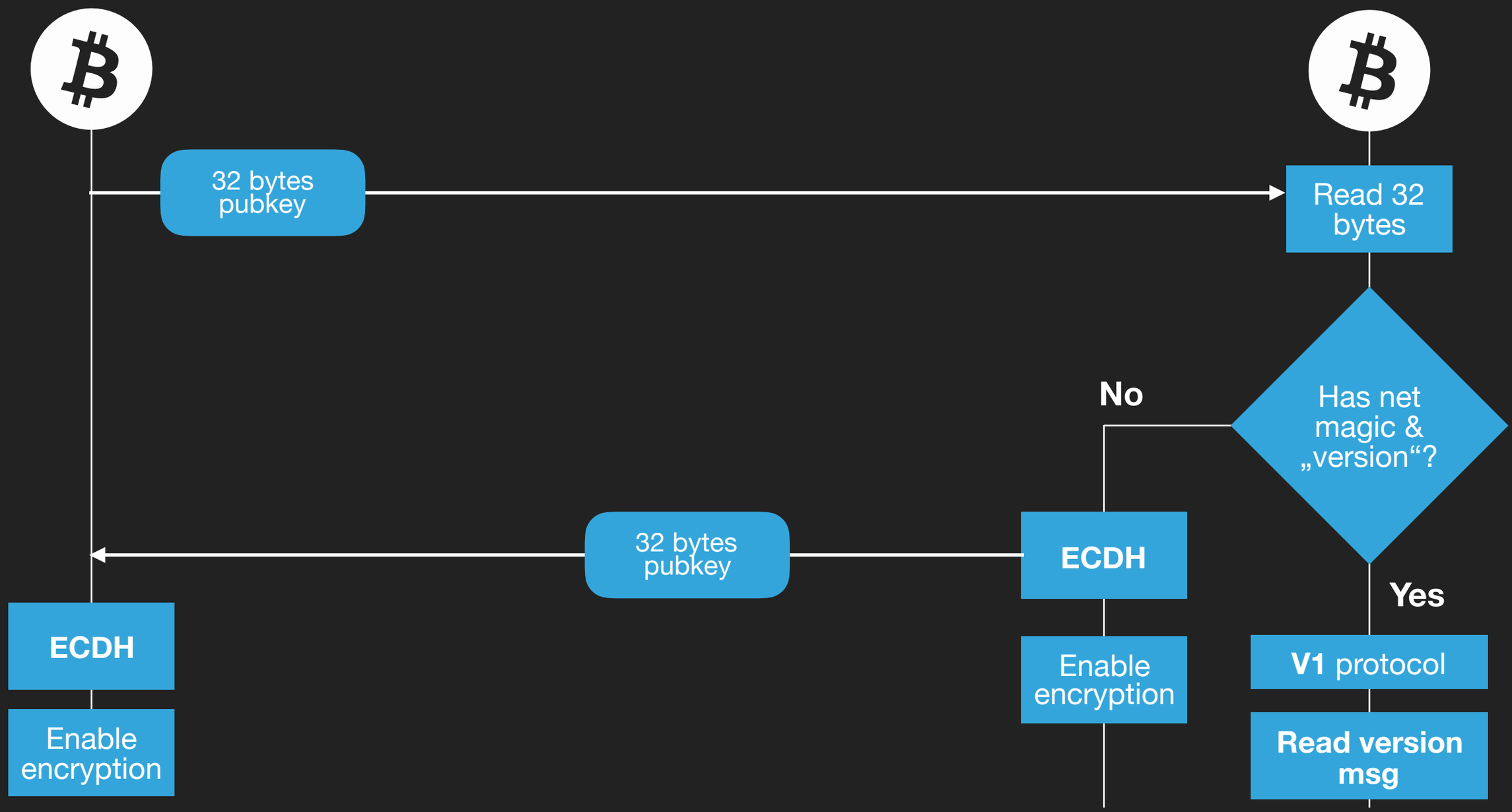
- Eliminate passive non detectable observing
- Eliminate non detectable message tempering

- **Building block for secure connections**

Required crypto primitives

- ECDH secp256k1
- HKDF SHA256 L32
- ChaCha20
- Poly1305

Handshake


Initiator (V2)
Responder (V2)
32 bytes pubkey
Read 32 bytes
Has net magic & „version“?
No
Yes
32 bytes pubkey
ECDH
ECDH
Enable encryption
Enable encryption
V1 protocol
Read version msg

Handshake

Initiator

Responder

x, X := Generate Secp256k1 Keypair
(only **odd** pub keys are allowed)

send pubkey **X** (no message structure, pure 32 bytes)

y, Y := Generate Secp256k1 Keypair
(only **odd** pub keys are allowed)

SS := Secp256k1_ECDH(X, y)

send pubkey **Y** (no message structure, pure 32 bytes)

SS := **Secp256k1_ECDH**(Y, x)

Handshake

- No message structure
- Pure 32byte handshake payload
- Only ODD pubkeys
- Only pubkeys **not** starting with the V1 network magic
- V1 Compatibility: fallback option to a version msg

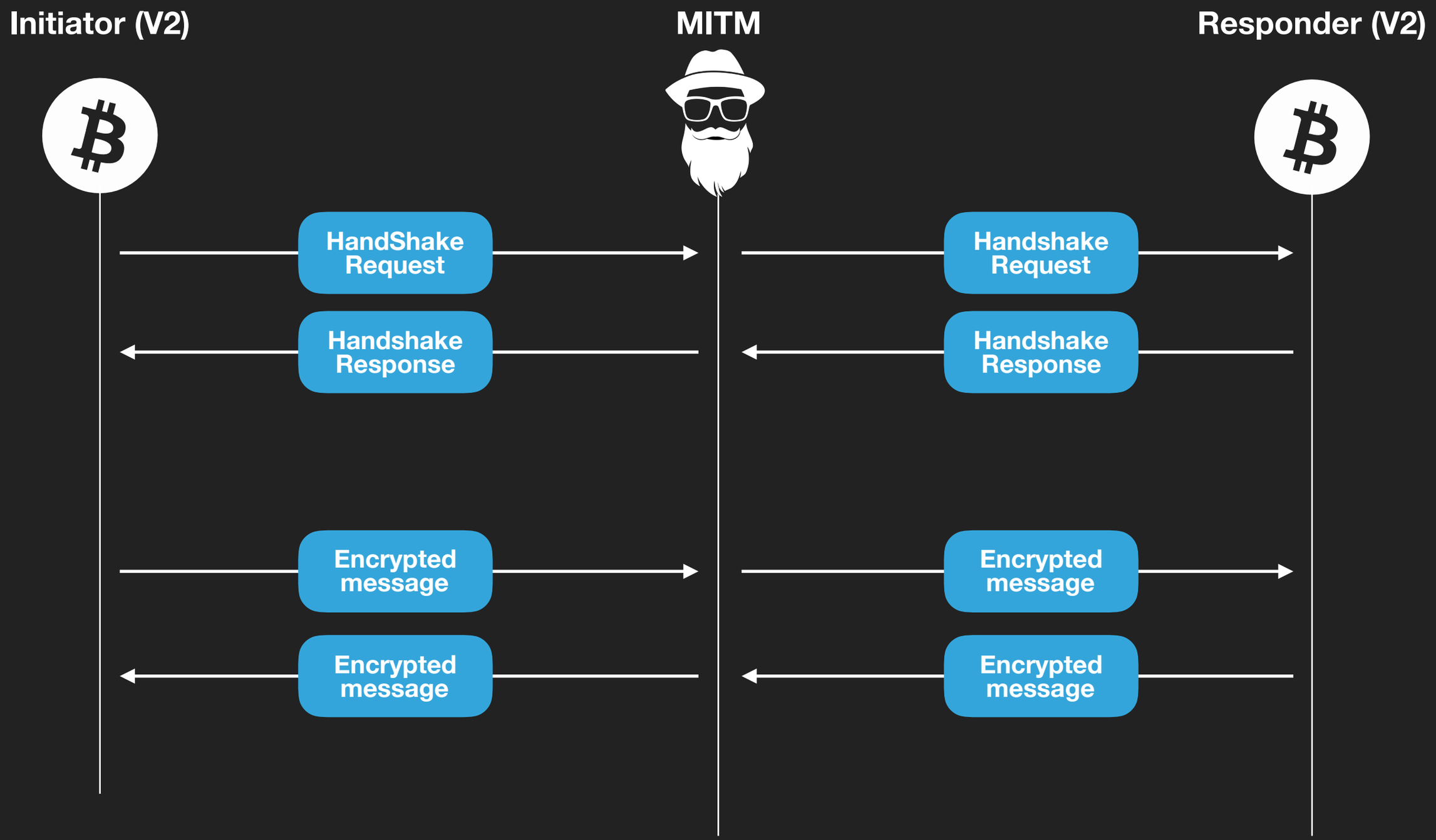
Session ID and keys

PRK = HKDF_SHA256_L32(**BitcoinSharedSecret || INITIATOR_32BYTES_PUBKEY||RESPONDER_32BYTES_PUBKEY**)

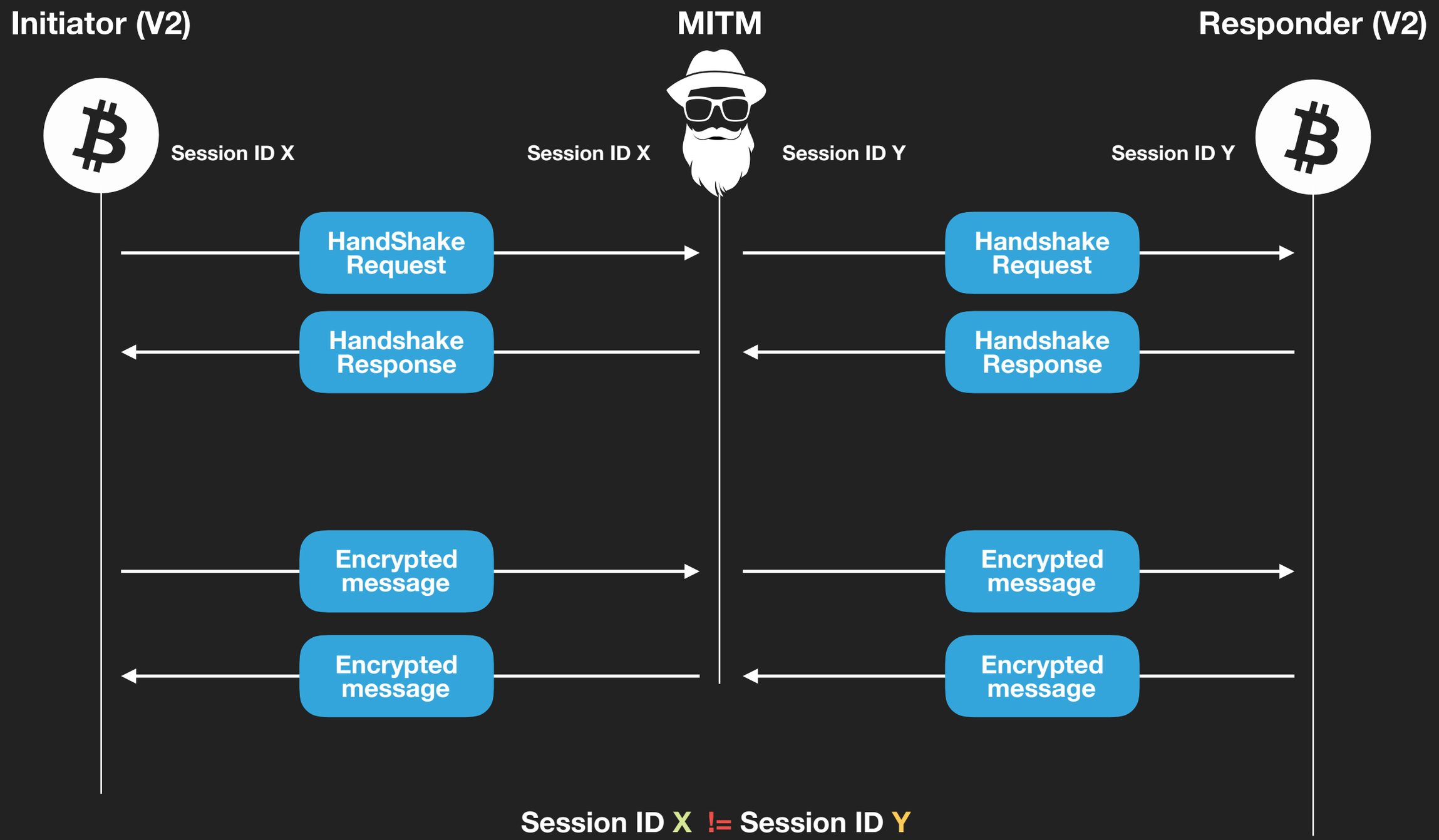
k1-k4 = HKDF_SHA256_L32_EXPAND(PRK, „**BitcoinK_1 - K4**“)

SessionID = HKDF_SHA256_L32_EXPAND(PRK, „**BitcoinSessionID**“)

Handshake MITM



Handshake

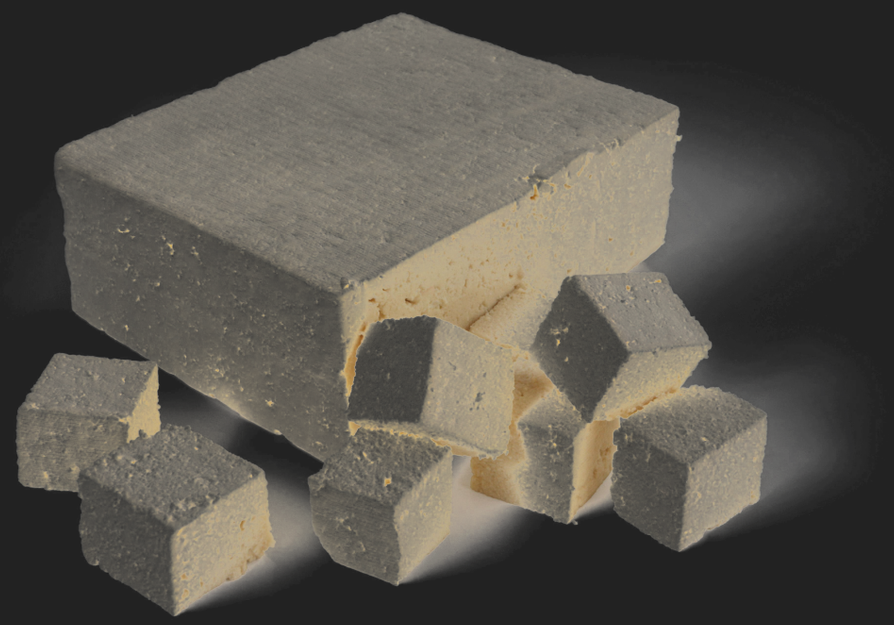

Initiator (V2)
MITM
Responder (V2)
Session ID X
Session ID X
Session ID Y
Session ID Y
HandShake Request
Handshake Request
Handshake Response
Handshake Response
Encrypted message
Encrypted message
Encrypted message
Encrypted message
Session ID X != Session ID Y

Authentication / MITM

«Bip151 provides excellent defence against government attackers with MITM capability: you can detect such attacks, and change behaviour. This is a huge improvement over the status quo of having no way of knowing if we're being attacked.»

Peter Todd

Authentication?



NO TOFU, no CA

TRUST **O**N **F**IRST **U**SE
Certificate **A**uthorities

CA

MOZILLA
Included CA Certificate List

As of June 4, 2019 (155 records displayed)

Owner	Certificate Issuer Organization	Certificate Issuer Organizational Unit	Common Name or Certificate Name	Certificate Serial Number	SHA-256 Fingerprint (Click to download .crt files)	Subject + SPKI SHA256
AC Camerfirma, S.A.	AC Camerfirma SA CIF A82743287	http://www.chambersign.org	Chambers of Commerce Root	00	0C258A12A5674AEF25F28BA7DCFAECEEA348E541E6F5CC4EE63B71B361606AC3	BC2FD9EA61581CB22BB859690D61430E7D222D1119E8C41649B9B1D556D439A4
AC Camerfirma, S.A.	AC Camerfirma S.A.		Chambers of Commerce Root - 2008	00A3DA427EA4B1AEDA	063E4AFAC491DFD332F3089B8542E94617D893D7FE944E10A7937EE29D9693C0	849AD3279D9B805A288339468C417744AC1CE2758A6E283A446685384D5D6CD2
AC Camerfirma, S.A.	AC Camerfirma SA CIF A82743287	http://www.chambersign.org	Global Chambersign Root	00	EF3CB417FC8EBF6F97876C9E4ECE39DE1EA5FE649141D1028B7D11C0B2298CED	F33BD6DDC7576A7AC9BBA464D7425F26D58FD2C51A3A1F89878D5262BDD485E0
AC Camerfirma, S.A.	AC Camerfirma S.A.		Global Chambersign Root - 2008	00C9CDD3E9D57D23CE	136335439334A7698016A0D324DE72284E079D7B5220BB8FBD747816EEBEBACA	7DB0C6863B891672A51E9D28B42610F1DB9F5651C8C87736B937536F8453F947
Actalis	Actalis S.p.A./03358520967		Actalis Authentication Root CA	570A119742C4E3CC	55926084EC963A64B96E2ABE01CE0BA86A64FBFEBCC7AAB5AFC155B37FD76066	362588CB736FFCA8E8CA1A485F8D0FD3FE4A32A1EDB3606E71A5F2F7986A4B34

SSH tofu

```
ECDSA key fingerprint is SHA256:jkhd+oTybtJwDoMqPwLThFjgIZfO56IukmqMfN2TUq8.

Are you sure you want to continue connecting (yes/no)?
```

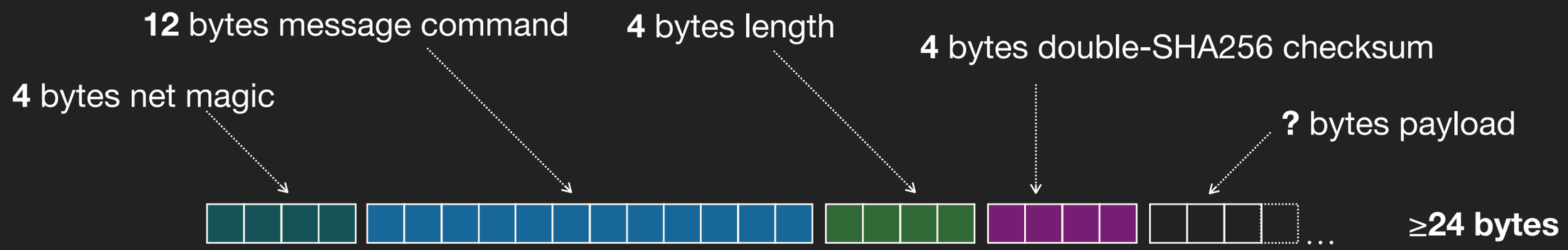
Make v2 **faster**

and **smaller**!

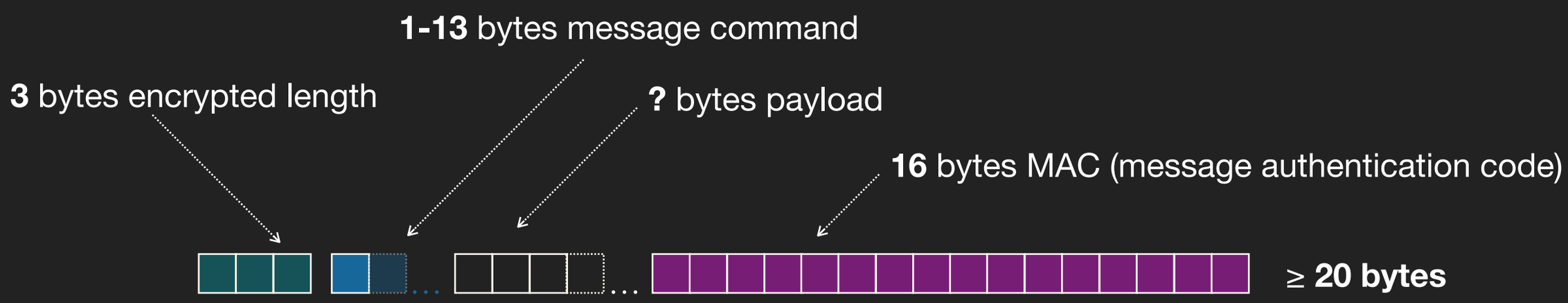
V2 Message Structure

V1 vs V2 Message Structure

V1



V2



V1 vs V2 Message Structure

V1 - message command string (pchCommand)

The message command space is always 12 bytes

examples:

INV000000000

BLOCK0000000

V2 - message command string (or short id)

First byte is size or short command id

The threshold is 12 (<=12 it's a length, >12 is a **short id**)

examples:

3INV

23

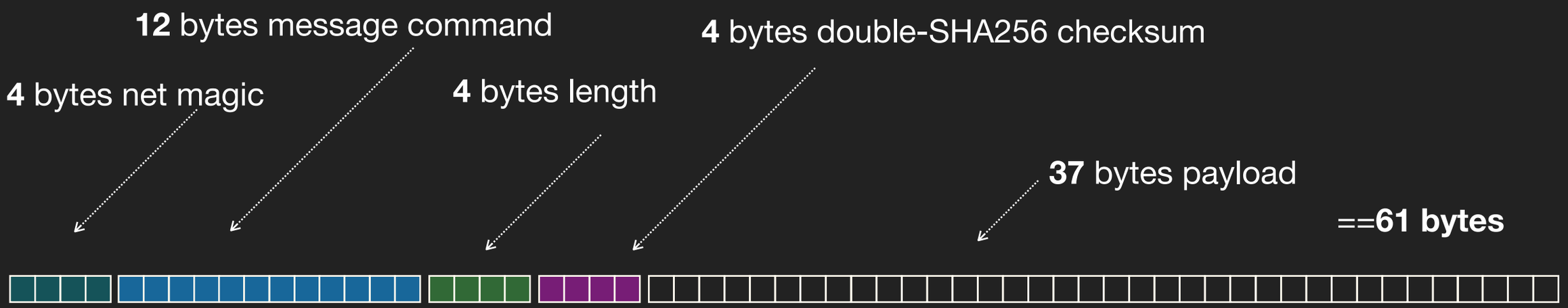
10GIGAMEGBLK

V1 vs V2 Message Structure

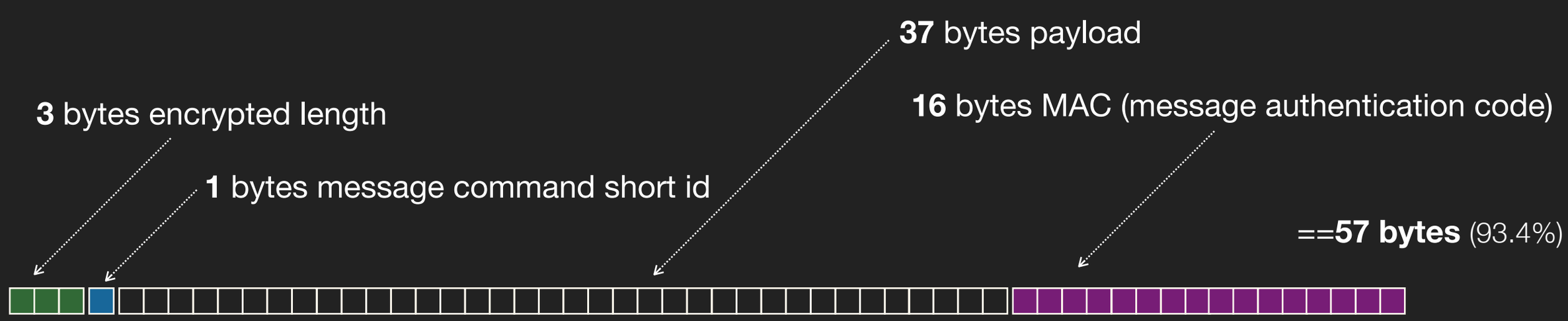
Number	Command		Number	Command
13	ADDR		21	GETHEADERS
14	BLOCK		22	HEADERS
15	BLOCKTXN		23	INV
16	CMPCTBLOCK		24	MERKLEBLOCK
17	FEEFILTER		25	NOTFOUND
18	GETADDR		26	PING
19	GETBLOCKTXN		27	PONG
20	GETDATA		28	SENDCMPCT
21	GETHEADERS		29	SENDHEADERS
22	HEADERS		30	TX
23	INV		31	VERACK
30	TX		32	VERSION

V1 vs V2 Message Structure

V1 - INV (single)

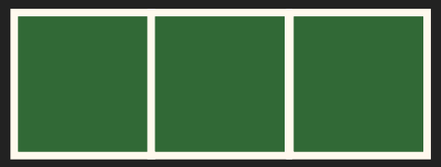


V2 - INV (single)



V1 vs V2 Message Structure

V2 length field



3 bytes == 24 bits
23 bits for length + 1 rekey trigger bit

Maximal Message length is 0x7FFFFF = ~8MB

Longer messages (e.g. gigamegblocks) could use **multi-part messages**.

Rekey

- Can be triggered by setting the most significant bit in the 3 bytes length field
- MUST not encrypt more then 1GB of data with the same key
- Avoid nonce reuse
- Rekey is **SHA256(SHA256(session ID ‖ old_symmetric_cipher_key))**

Custom AEAD construct:

ChaCha20Poly1305@**Bitcoin**

stream cipher

MAC

ChaCha20

- Faster on systems without AES NI
- Not vulnerable to cache-timing attacks (ARX)
- randomly accessible output stream (parallelizable)

ChaCha20Poly1305**@Bitcoin**

- Based on ChaCha20Poly1305@**OpenSSH** which is. based on ChaCha20Poly1305 IETF RFC 7539

ChaCha20Poly1305 **IETF RFC 7539**

↓ Change AD, Encrypt Length

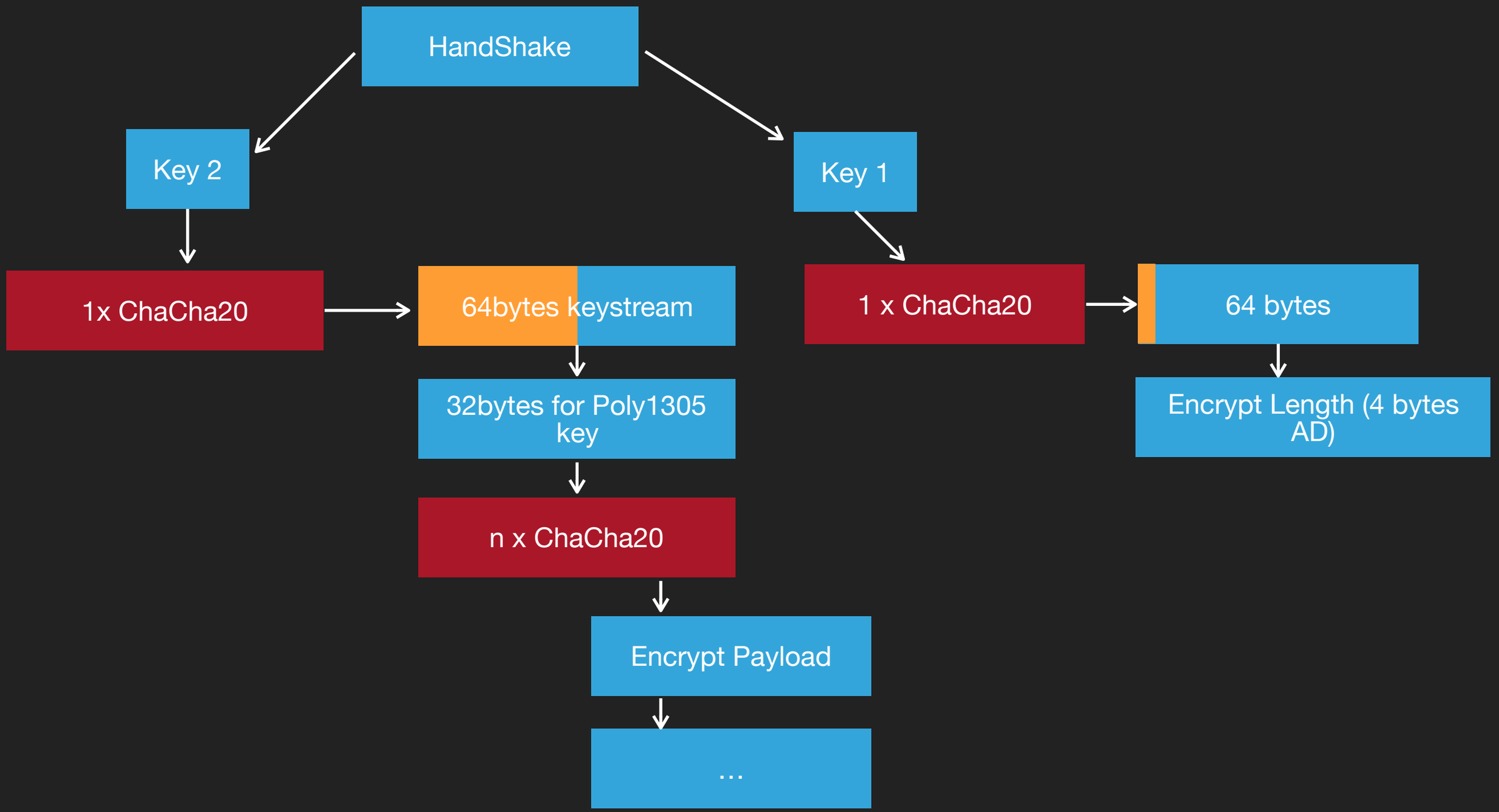
ChaCha20Poly1305**@OpenSSH**

↓ More efficient AD encryption

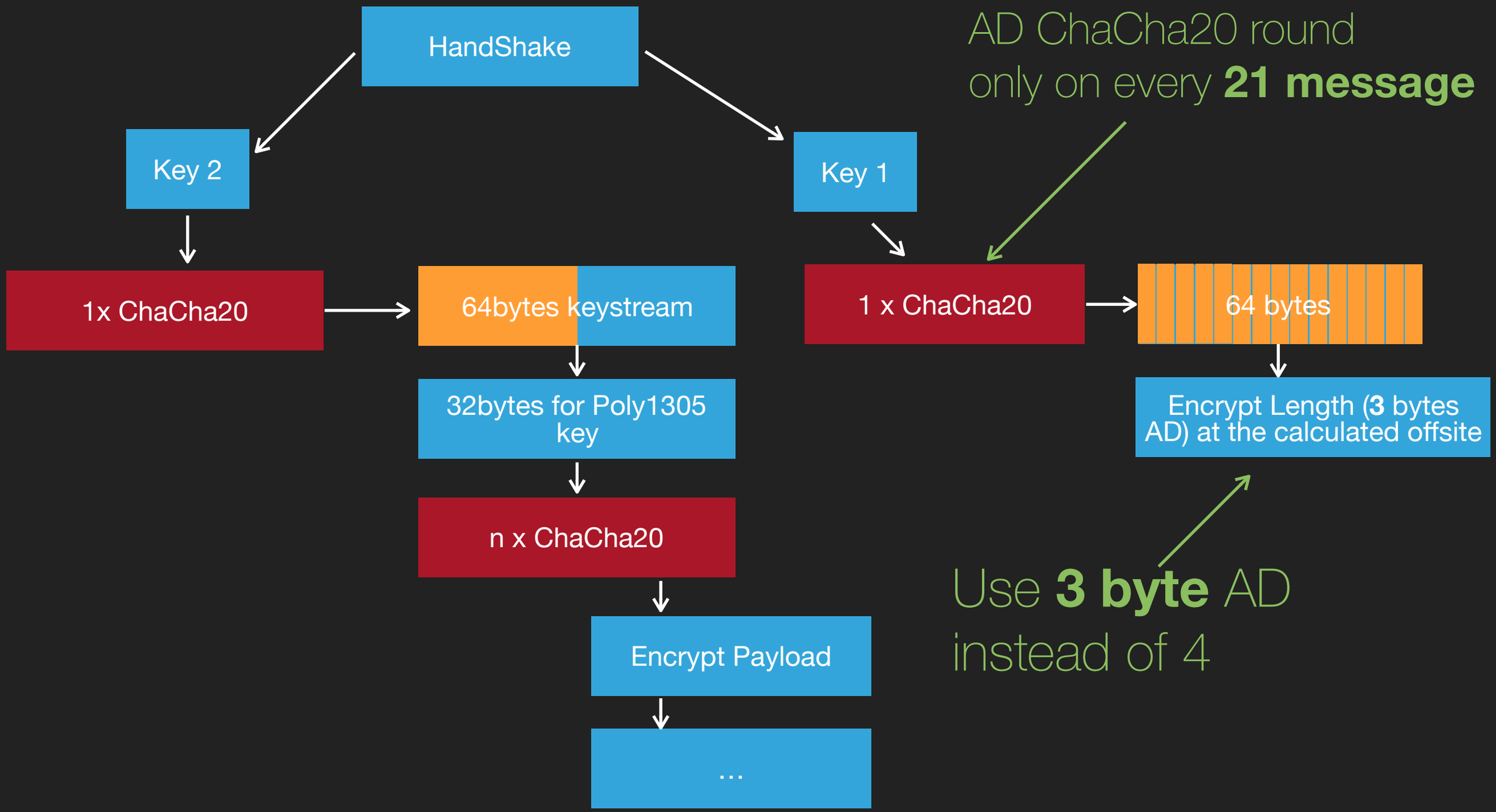
ChaCha20Poly1305**@Bitcoin**

Optimized for bitcoins traffic

ChaCha20Poly1305**@OpenSSH**



ChaCha20Poly1305@Bitcoin



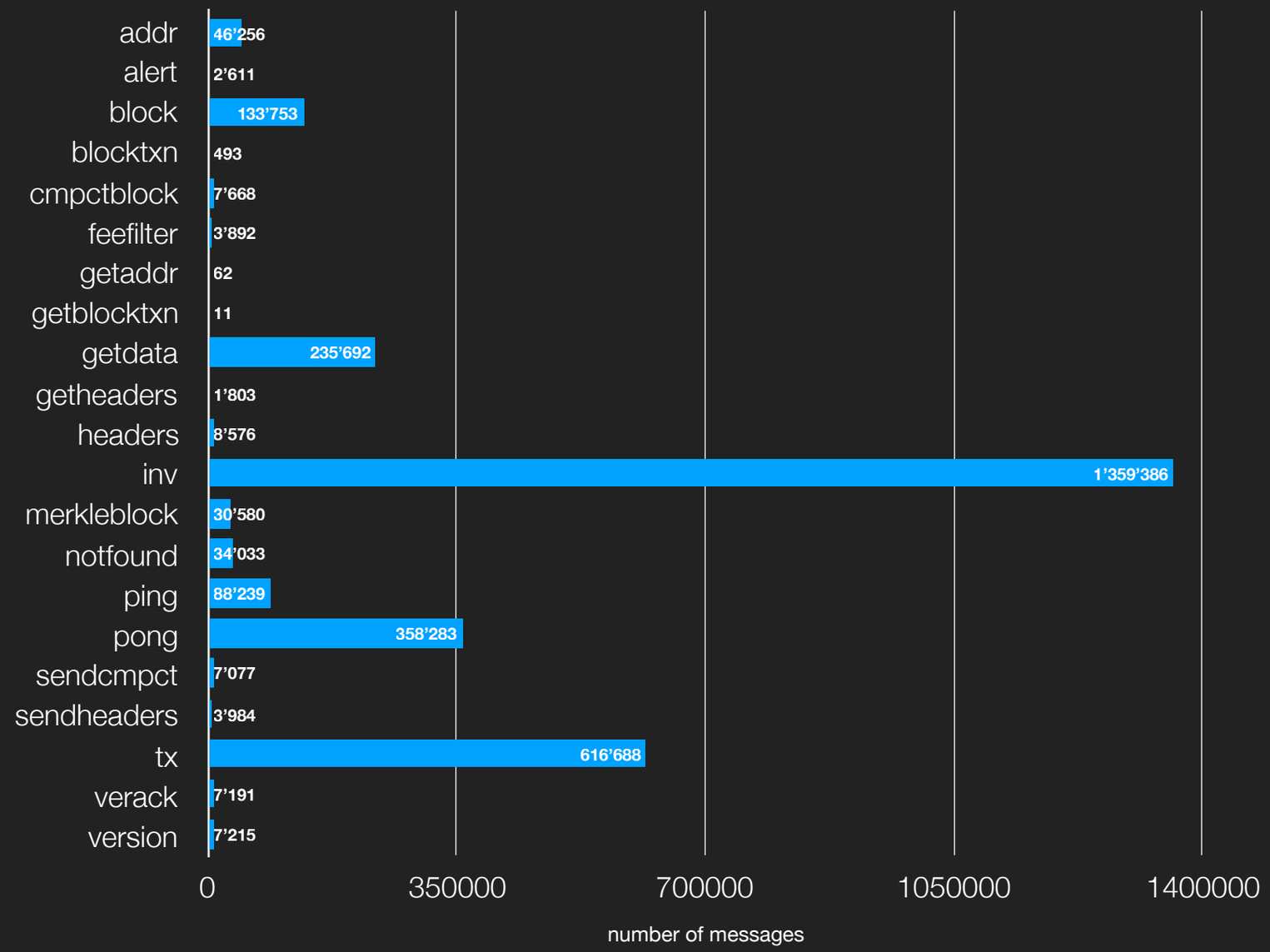
ChaCha20Poly1305**@Bitcoin**

>= ~2.048 ChaCha20 „rounds“ per message

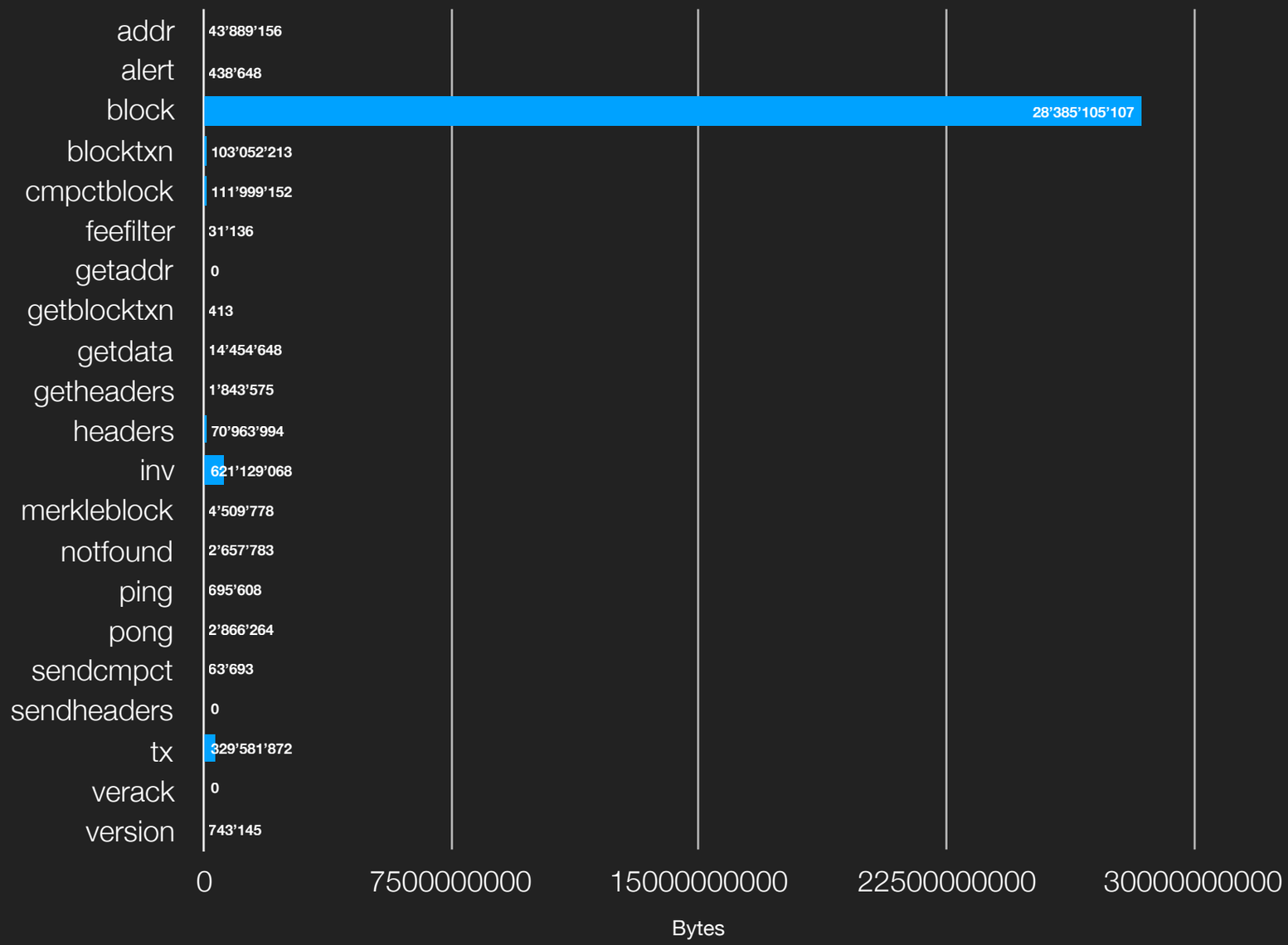
ChaCha20Poly1305**@OpenSSH**

>= 3 ChaCha20 „rounds“ per message

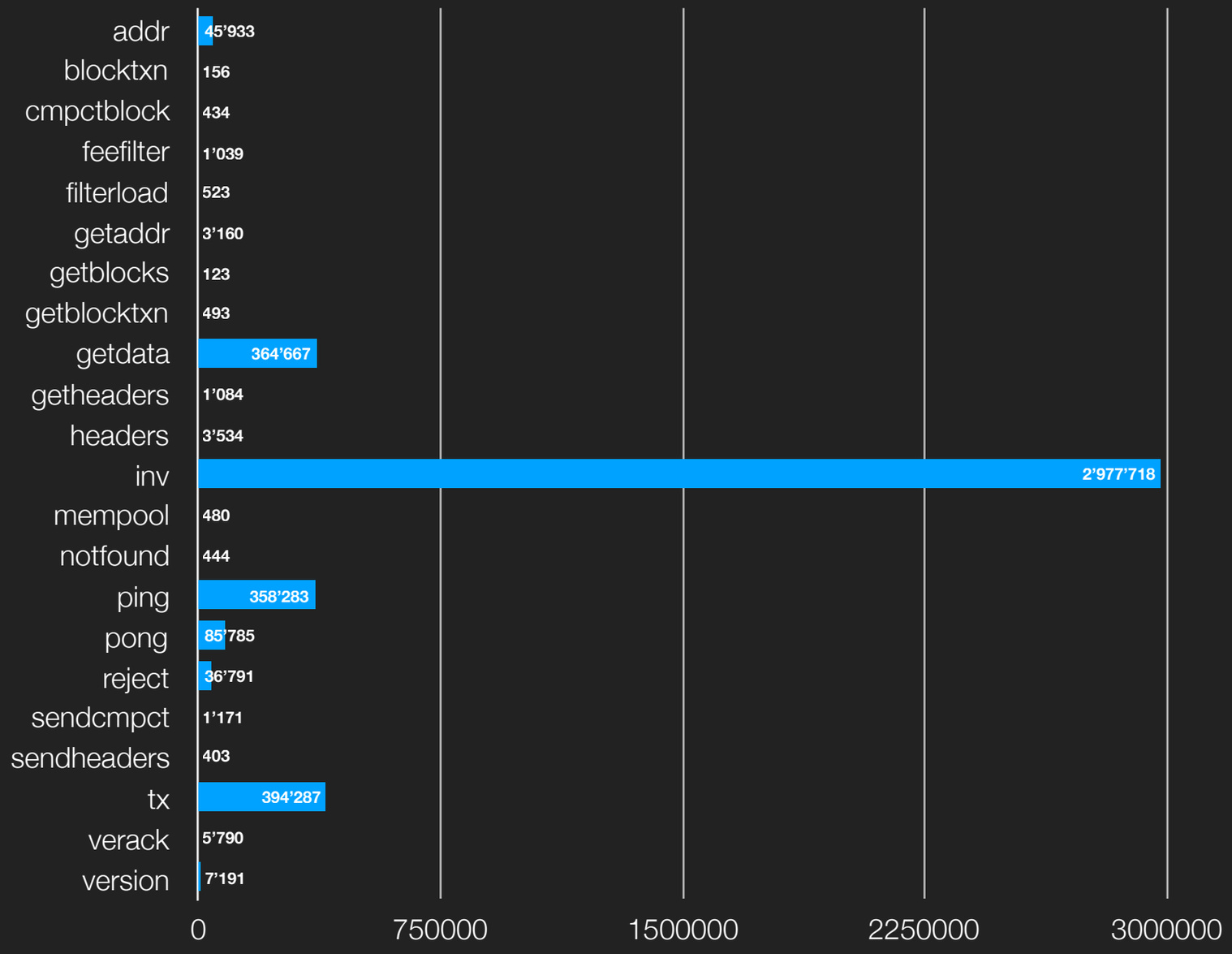
Bitcoin Core **send** message **count** with standard settings, random 24h



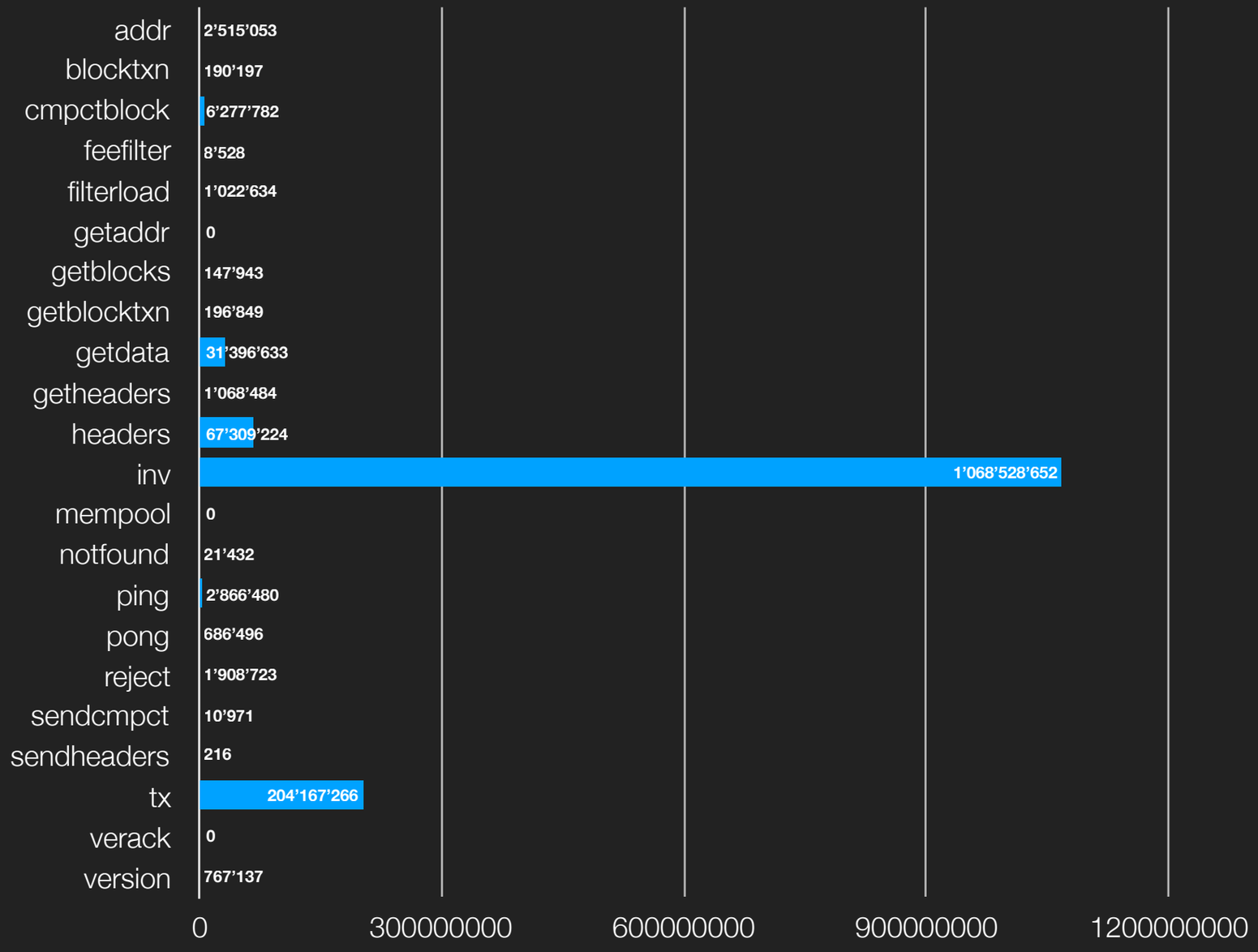
Bitcoin Core **send** message **bytes** with standard settings, random 24h



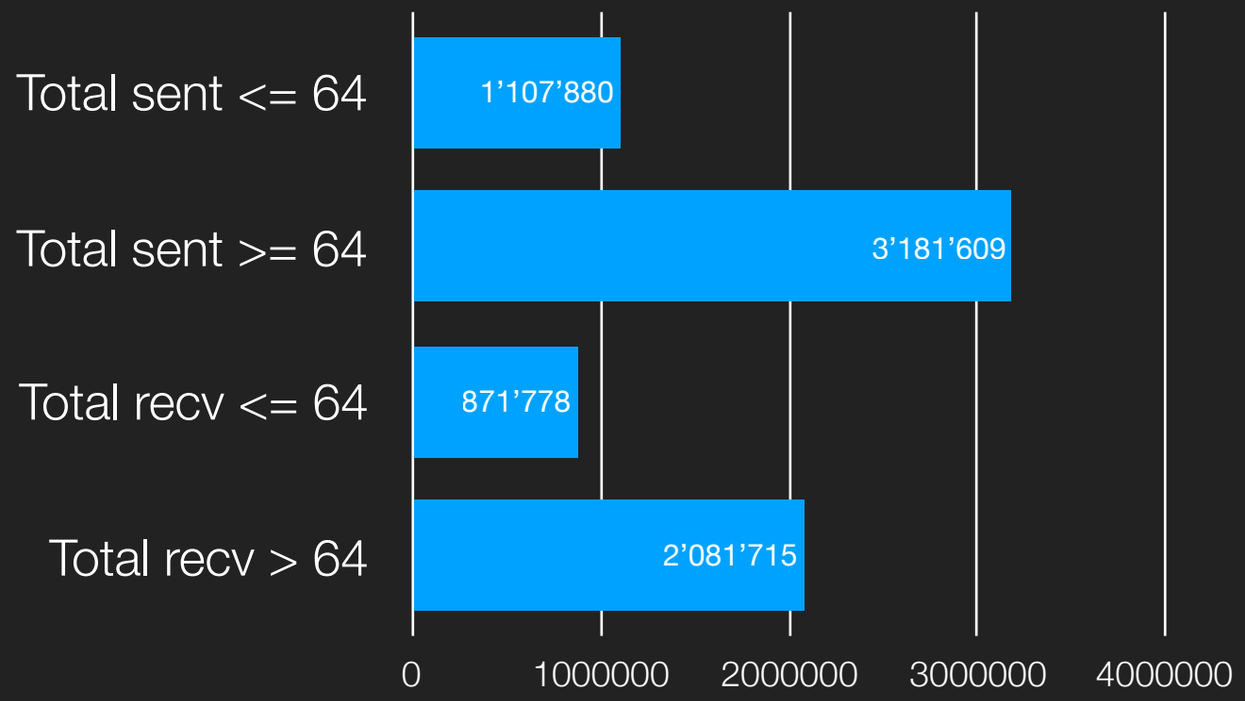
Bitcoin Core **recv** message **count** with standard settings, random 24h



Bitcoin Core **recv** message **bytes** with standard settings, random 24h

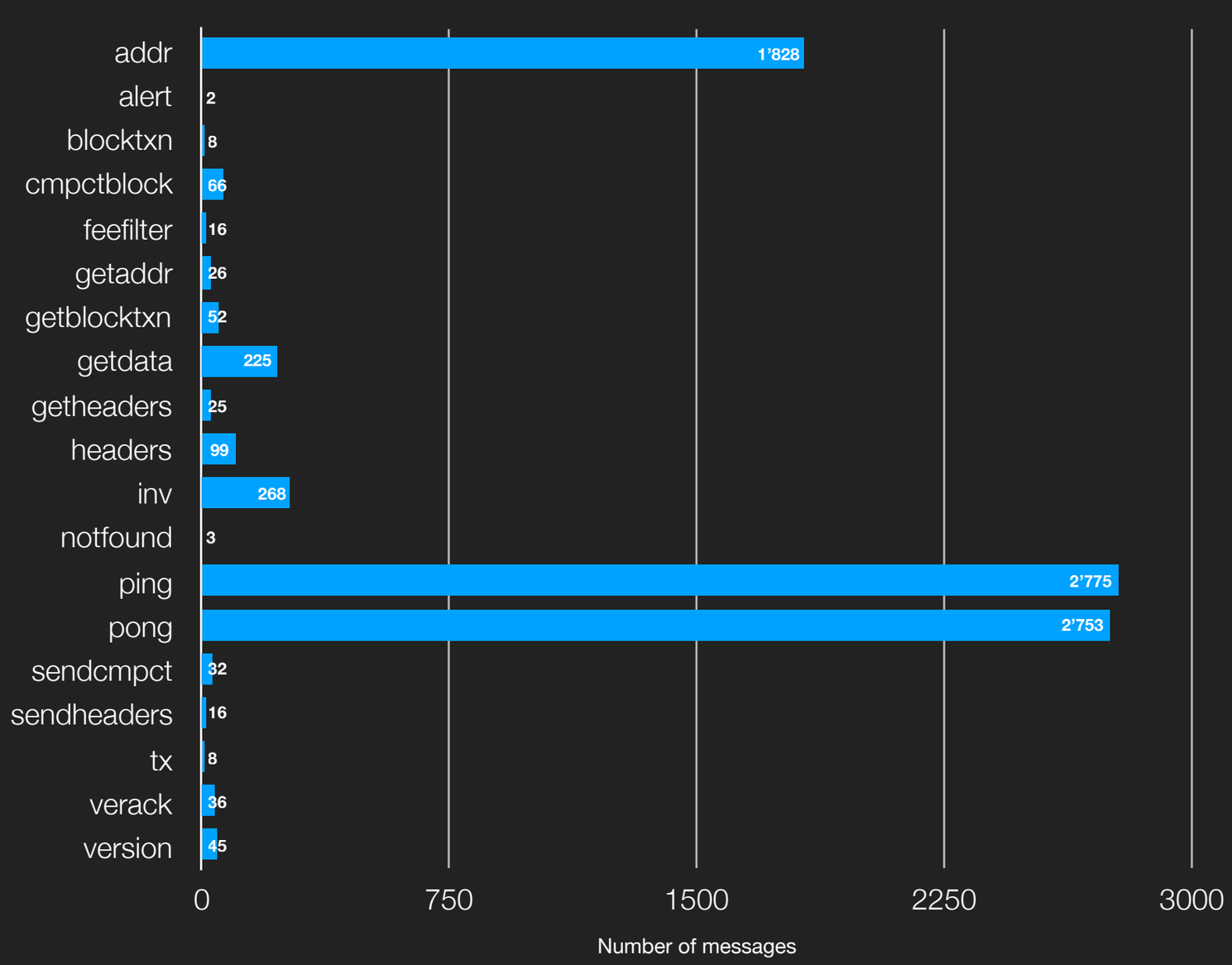


Total amount of messages <= 64 bytes payload



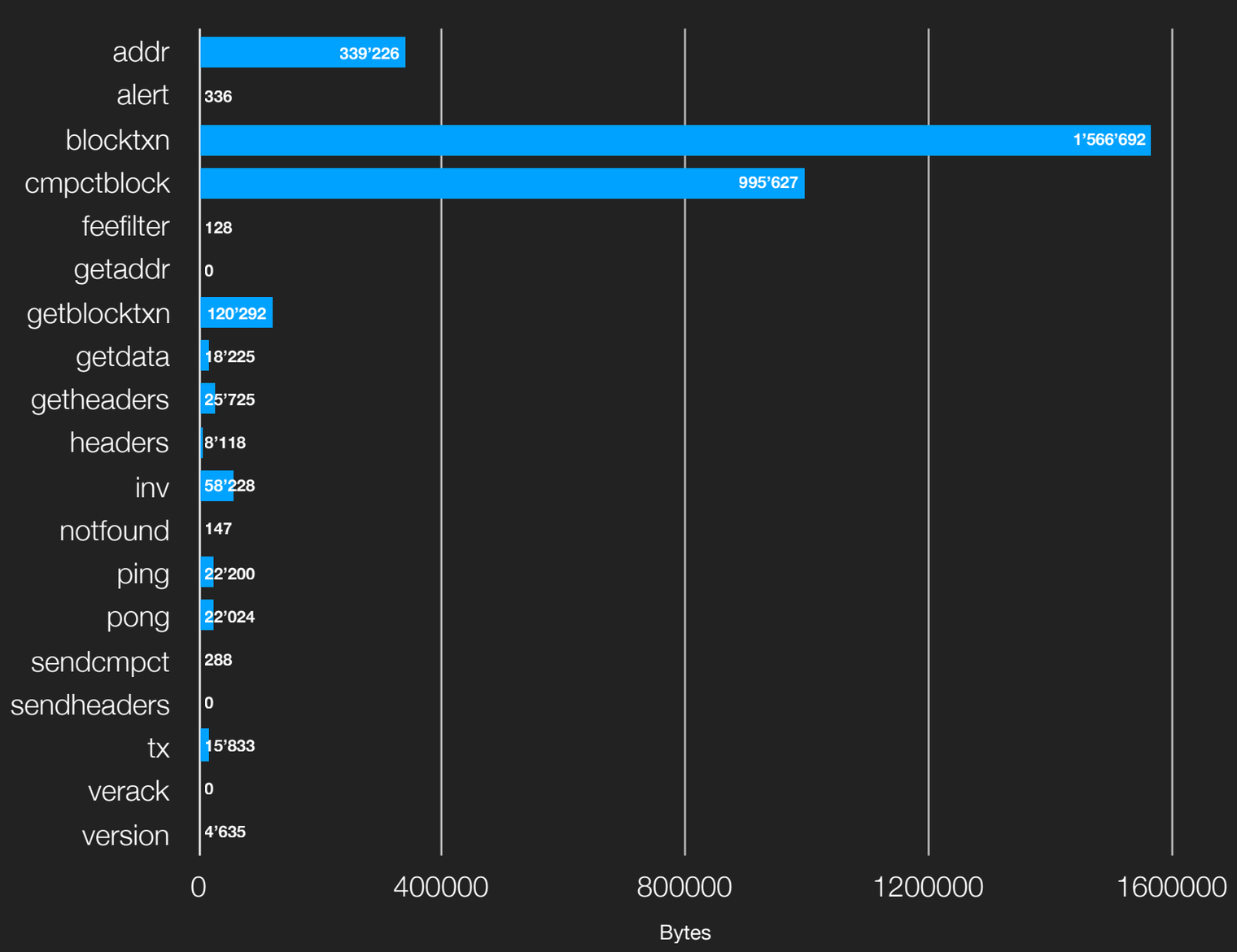
Bitcoin Core **send** message **count** with **prune**, random 9h

PRUNE

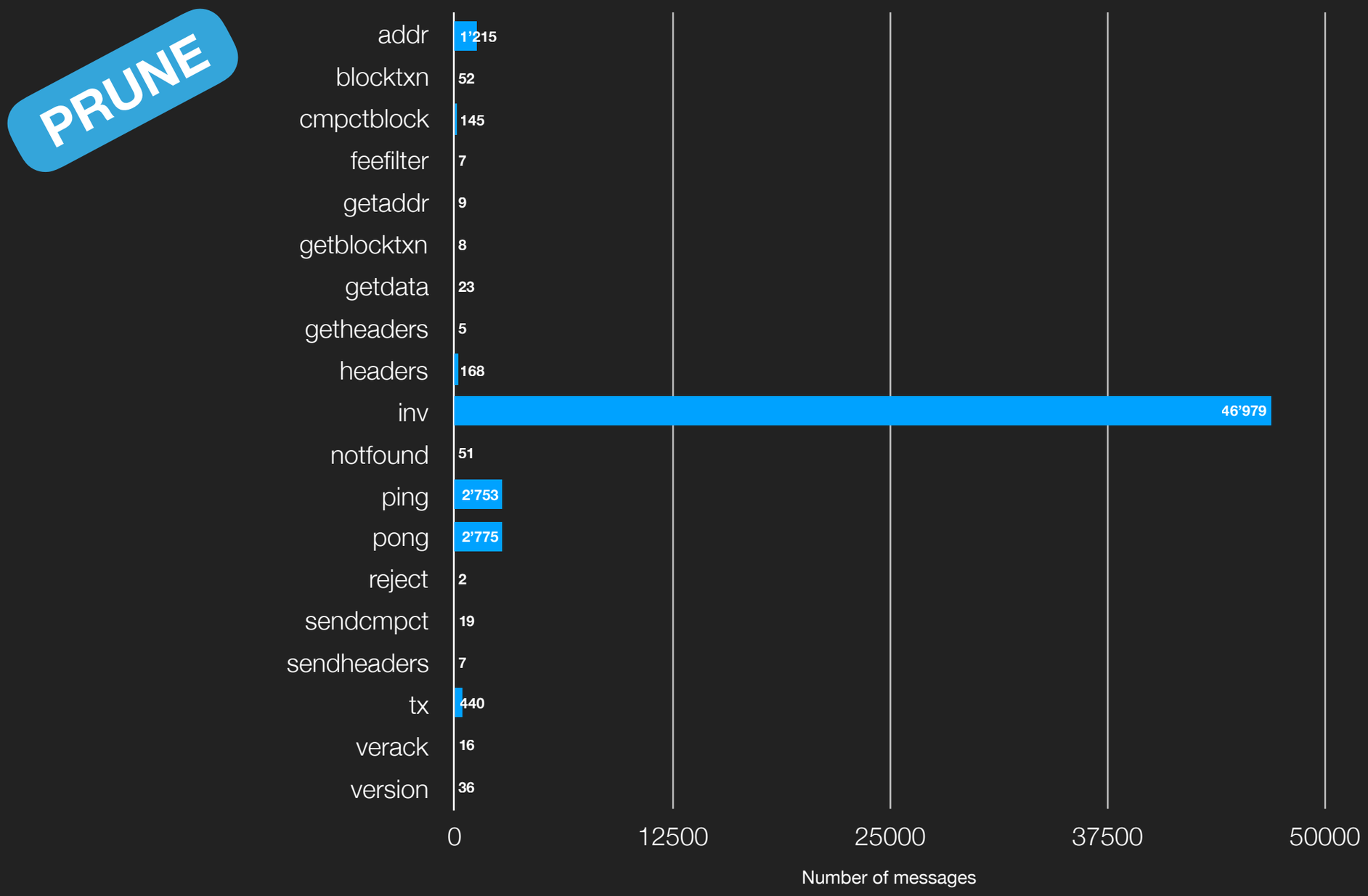


Bitcoin Core **send** message **bytes** with **prune**, random 9h

PRUNE

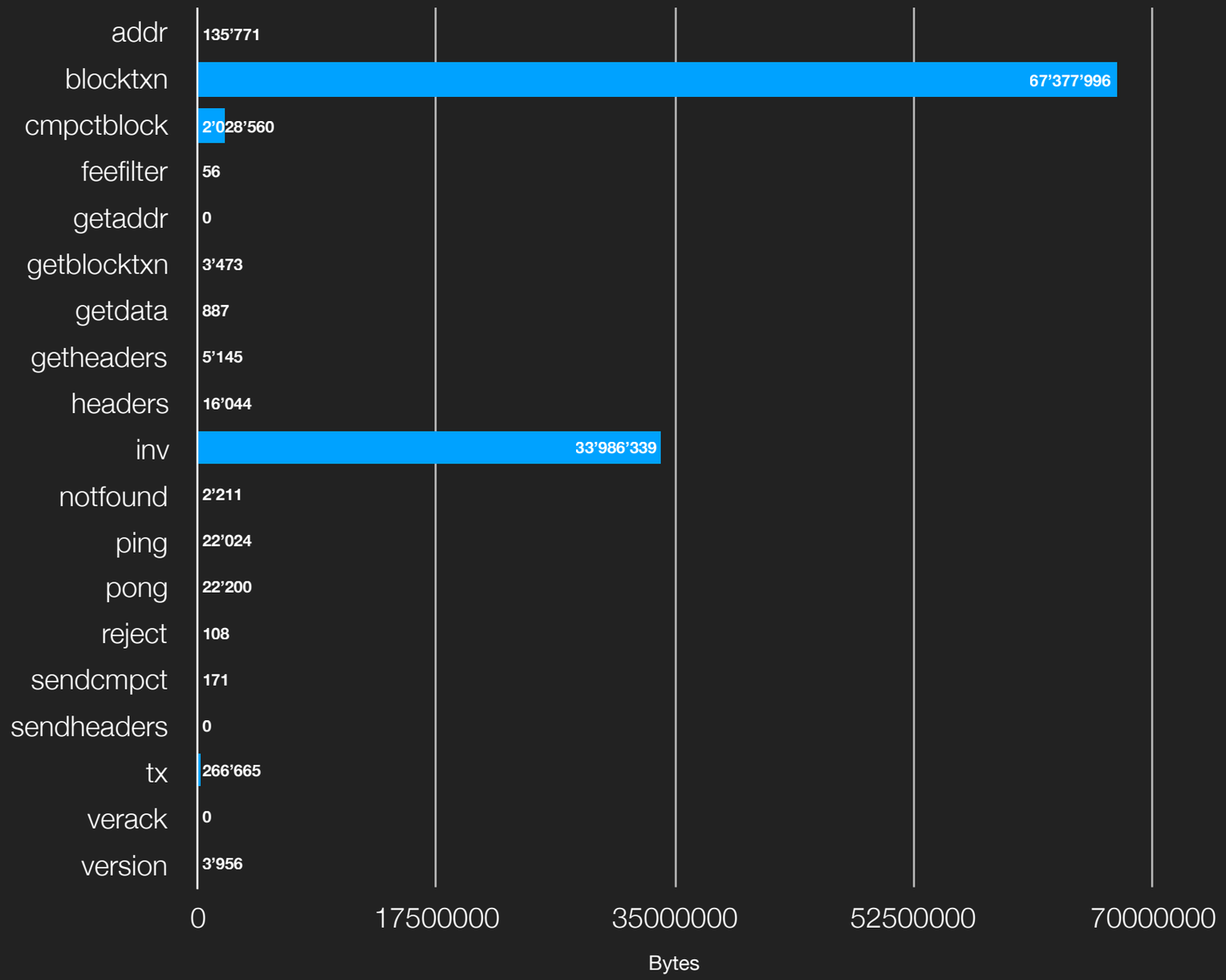


Bitcoin Core **recv** message **count** with **prune**, random 9h

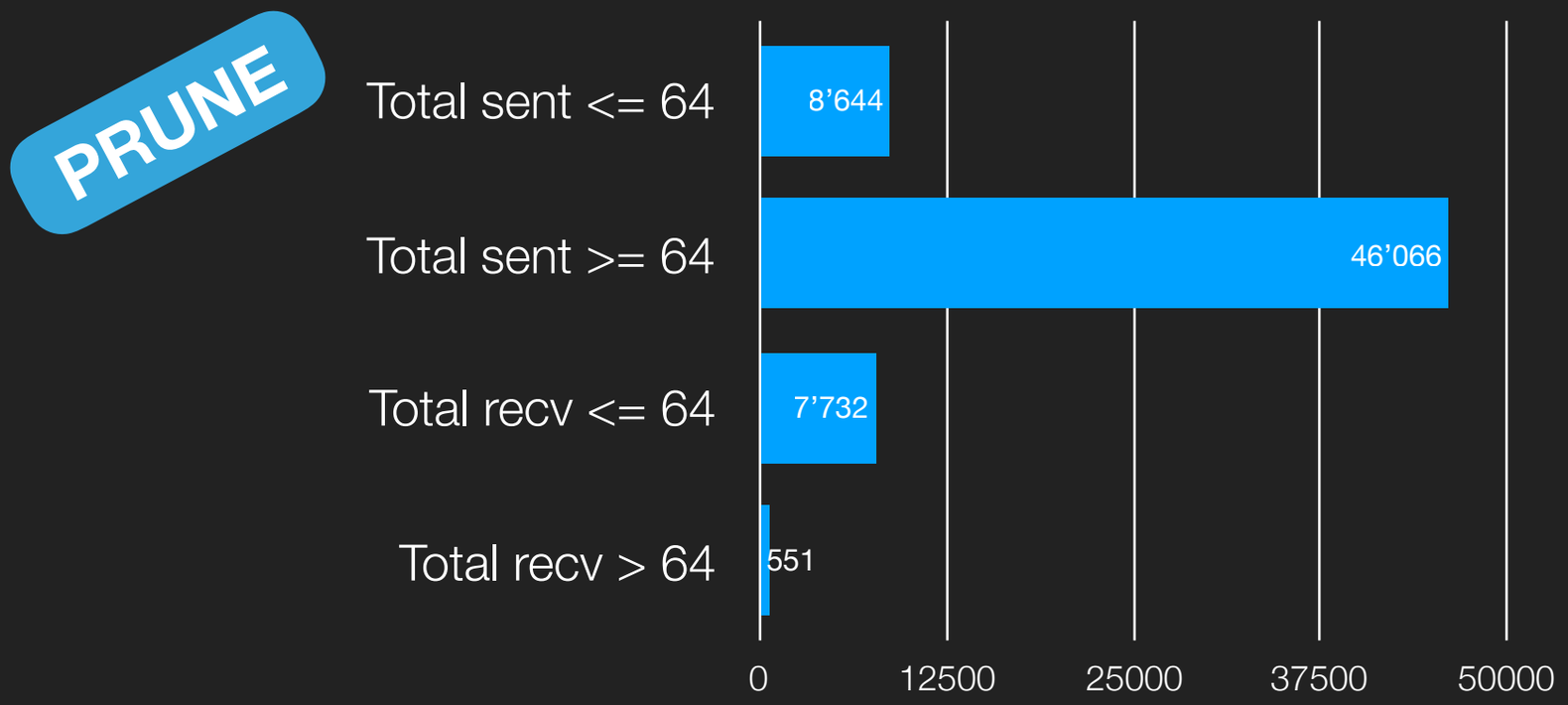


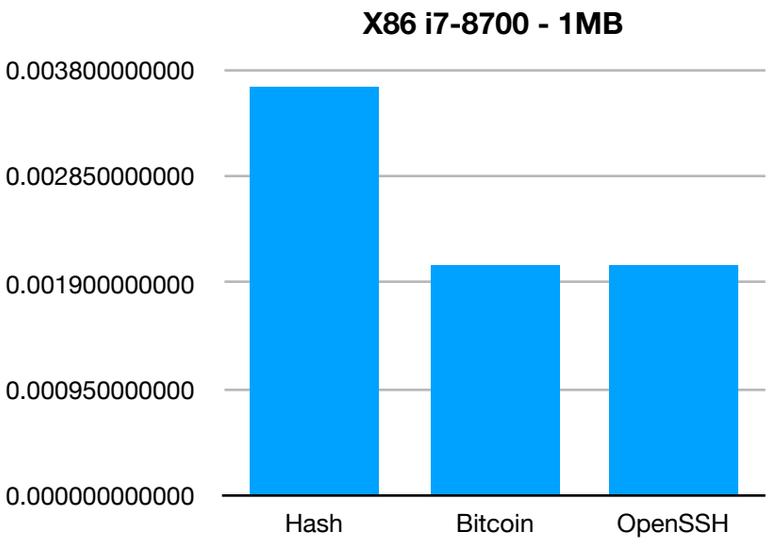
Bitcoin Core **recv** message **bytes** with **prune**, random 9h

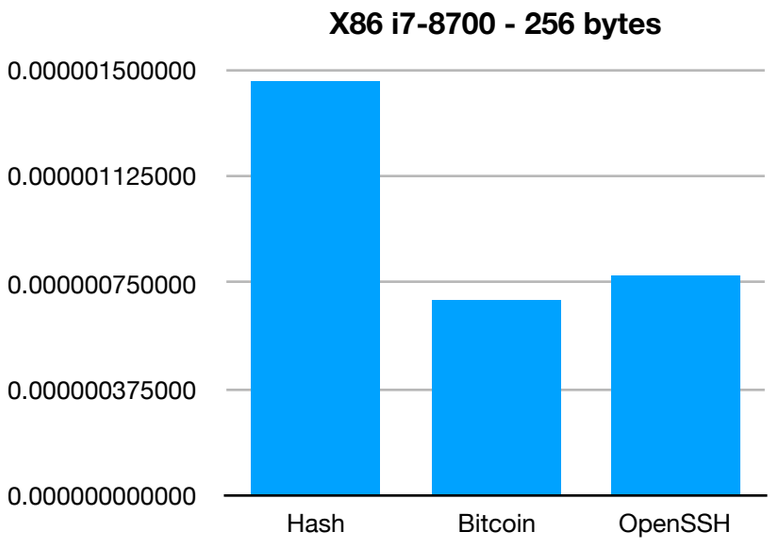
PRUNE

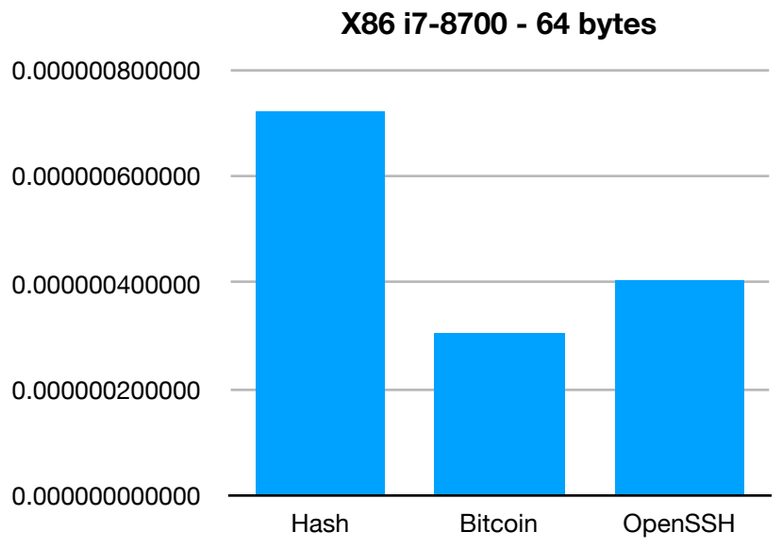


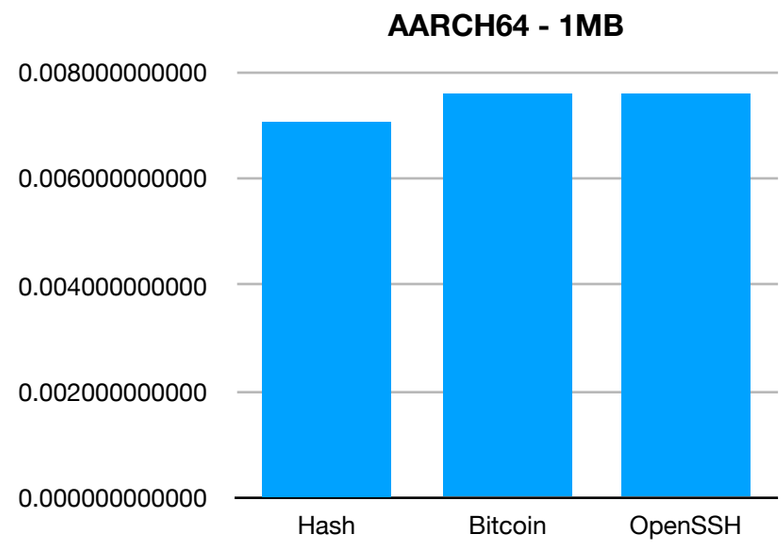
Total amount of messages <= 64 bytes payload

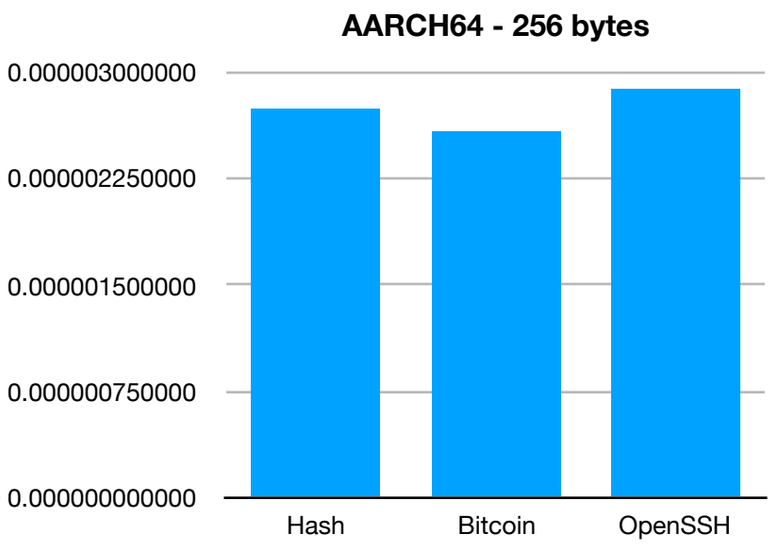


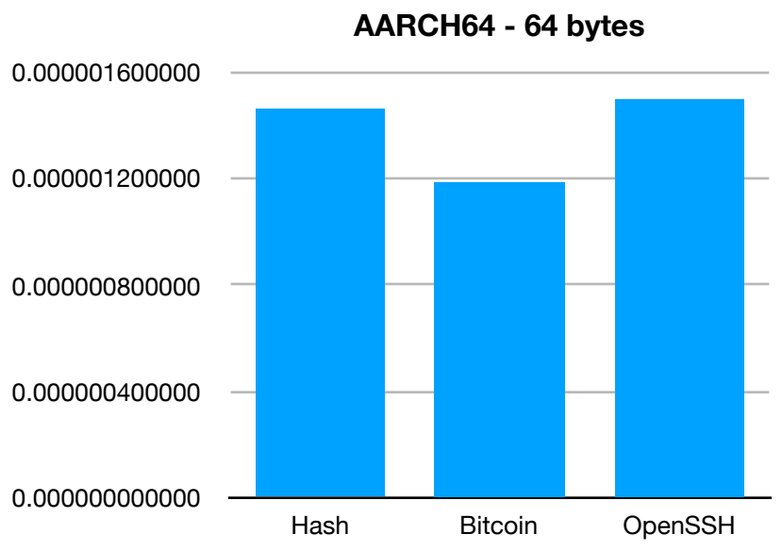

X86 i7-8700 - 1MB
0.003800000000
0.002850000000
0.001900000000
0.000950000000
0.000000000000
Hash
Bitcoin
OpenSSH


X86 i7-8700 - 256 bytes
0.000001500000
0.000001125000
0.000000750000
0.000000375000
0.000000000000
Hash
Bitcoin
OpenSSH


X86 i7-8700 - 64 bytes
0.000000800000
0.000000600000
0.000000400000
0.000000200000
0.000000000000
Hash
Bitcoin
OpenSSH


AARCH64 - 1MB
0.008000000000
0.006000000000
0.004000000000
0.002000000000
0.000000000000
Hash
Bitcoin
OpenSSH


AARCH64 - 256 bytes
0.000003000000
0.000002250000
0.000001500000
0.000000750000
0.000000000000
Hash
Bitcoin
OpenSSH


AARCH64 - 64 bytes
0.000001600000
0.000001200000
0.000000800000
0.000000400000
0.000000000000
Hash
Bitcoin
OpenSSH

Next steps

- ChaCha20 / Poly1305 has been merged
- More review of the proposal
- The AEAD is an open PR ready to review
- The complete implementation is open as PR for conceptual review
- Deploy it as optional experimental feature

Thanks, **Q&A**?

dev@**jonasschnelli**.ch

PGP: CA1A2908DCE2F13074C62CDE1EB776BB03C7922D

jonasschnelli

github.com/jonasschnelli