Bitcoin:
Programming the Future of Money

Topics in Computer Science - ITCS 4010/5010, Spring 2025

Dr. Christian Kümmerle

Lecture 14

Bitcoin Addresses

Some figures are taken from:

- “Mastering Bitcoin: Programming the Open Blockchain”, (Andreas Antonopoulos, David Harding), 3rd Edition, O’Reilly, 2023.

The Strategic Bitcoin Reserve

FROM A RECENT EXECUTIVE ORDER (MARCH 6)

- Creation of a **Bitcoin Strategic Reserve**
- Creation of a **Digital Asset Stockpile**

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered:

Section 1. Background. Bitcoin is the original cryptocurrency. The Bitcoin protocol permanently caps the total supply of bitcoin (BTC) at 21 million coins, and has never been hacked. As a result of its scarcity and security, Bitcoin is often referred to as "digital gold". Because there is a fixed supply of BTC, there is a strategic advantage to being among the first nations to create a strategic bitcoin reserve. The United States Government currently holds a significant amount of BTC, but has not implemented a policy to maximize BTC's strategic position as a unique store of value in the global financial system. Just as it is in our country's interest to thoughtfully manage national ownership and control of any other resource, our Nation must harness, not limit, the power of digital assets for our prosperity.

FROM A RECENT EXECUTIVE ORDER (MARCH 6)

Sec. 2. Policy. It is the policy of the United States to establish a Strategic Bitcoin Reserve. It is further the policy of the United States to establish a United States Digital Asset Stockpile that can serve as a secure account for orderly and strategic management of the United States' other digital asset holdings.

(c) The Secretary of the Treasury and the Secretary of Commerce shall develop strategies for acquiring additional Government BTC provided that such strategies are budget neutral and do not impose incremental costs on United States taxpayers. However, the United States Government shall not acquire additional Stockpile Assets other than in connection with criminal or civil asset forfeiture proceedings or in satisfaction of any civil money penalty imposed by any agency without further executive or legislative action.

A SOCIAL MEDIA POST FROM THE WEEK BEFORE...

WHITE HOSE CRYPTO SUMMIT ON MARCH 7

SOME QUESTIONS TO REFLECT ON

- What is the strategic benefit of having a national Bitcoin reserve?
- What are the dangers of having a national Bitcoin reserve?
- Why is the digital asset stockpile separated from the strategic Bitcoin reserve?

Recap: Encodings and Serialization

ENCODINGS AND SERIALIZATION

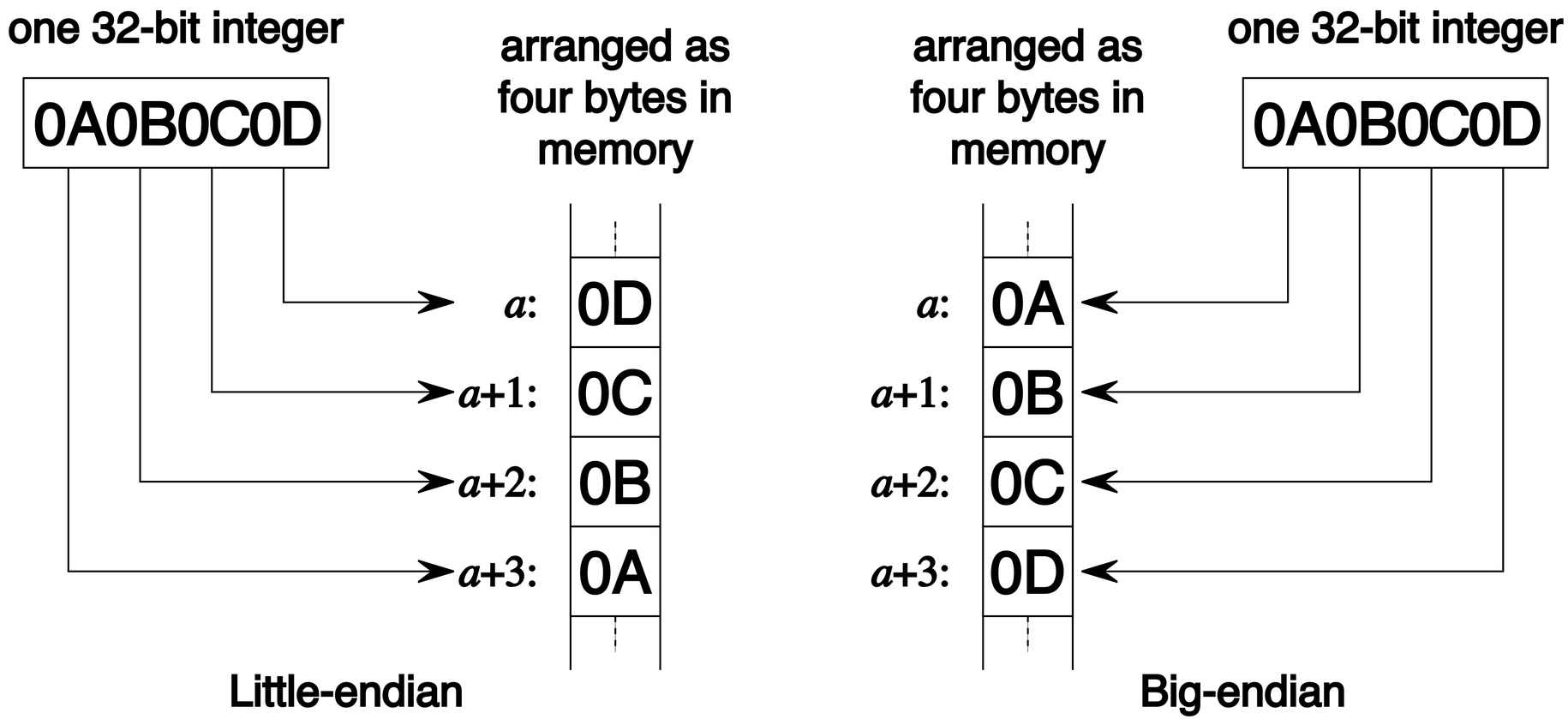
- What is the **big-endian encoding** and what is the **little-endian encoding** of bit strings?
- What is **Uncompressed SEC format** and the **Compressed SEC format** of ECDSA public keys? How long are these formats, respectively?
- What are **DER Signatures**?
- How are

  - ECDSA signatures

  - Schnorr Signatures

  serialized in the Bitcoin protocol?
- What is the purpose of **SIGHASH flags** and how do they appear in the serialization of ECDSA/Schnorr Signatures?

BIG-ENDIAN AND LITTLE-ENDIAN



What is a Bitcoin address?

BITCOIN ADDRESS TYPES

A **Bitcoin address** is a user-friendly encoding of an output (locking) script which specifies the conditions for corresponding coins to be spent.

Example:

- "Only somebody who is able to **present a valid signature** corresponding to the **public key** 04b0bd634234… can spend the associated coins."

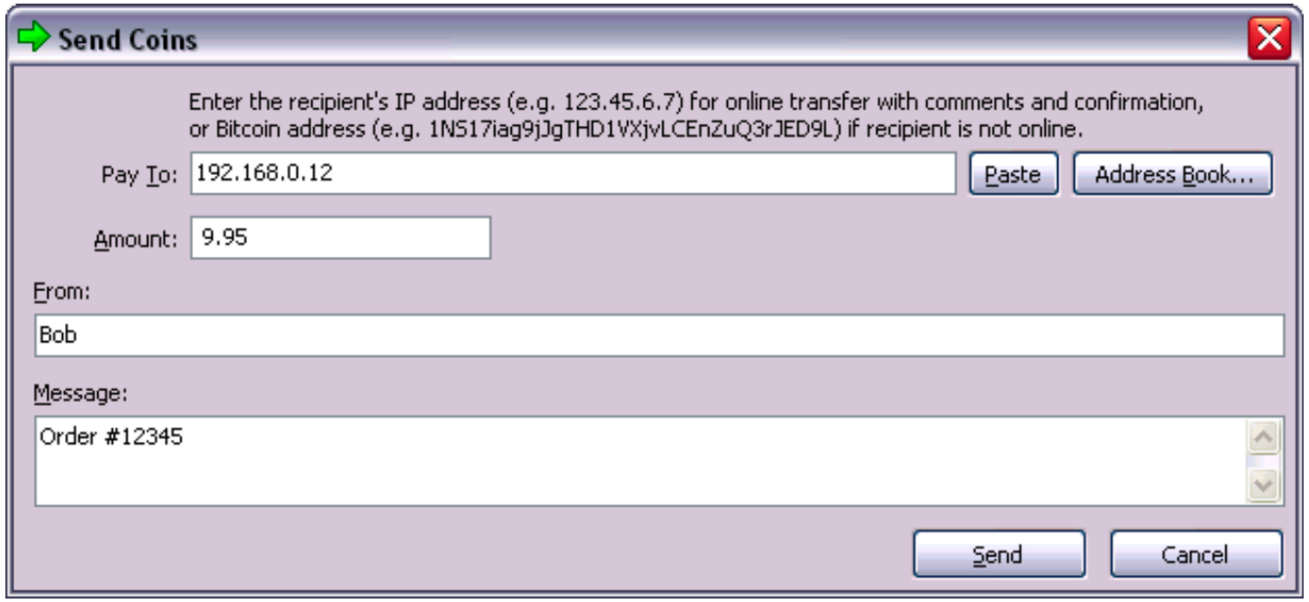
or:

- "Only somebody who is able to **present two valid signatures,** corresponding to **the public keys** 04b0bd634234… AND 04e8e37f1556b… can spend the associated coins."

BITCOIN ADDRESS TYPES

- **P2PK:** “Pay To Public Key”
  Legacy, only used commonly in 2010 era
- **P2PKH:** “Pay To Public Key Hash”
  Default until 2017 SegWit upgrade for standard single-signature spends
- **P2SH:** “Pay To Script Hash”
  Used for custom locking scripts, e.g., for multi-signature scripts before the 2017 SegWit upgrade
- **P2MS:** “Pay to Multisig”
  Can be used for multi-signature output scripts, but longer than P2SH.
- **P2WPKH:** “Pay to Witness Public Key Hash”
  Similar function as P2PKH before 2017 SegWit upgrade, different encoding
- **P2WSH:** “Pay to Witness Script Hash”
  Similar function as P2SH before 2017 SegWit upgrade, different encoding
- **P2TR:** “Pay to Taproot”
  Introduced by 2021 Taproot upgrade, uses Schnorr signatures, enhanced privacy and flexibility, covers use cases of both P2WPKH and P2WSH (or P2PKH/P2SH)

PAY TO PUBLIC KEY (P2PK)



- Originally designed to facilitate “send to IP address” feature
- Rarely used, as P2PKH is shorter and adds privacy
- Address format: Uncompressed SEC or Compressed SEC of a public key.

Example (Uncompressed SEC): (Block explorer)

```
04b0bd634234abbb1ba1e986e884185c61cf43e001f9137f23c2c409
273eb16e6537a576782eba668a7ef8bd3b3cfb1edb7117ab65129b8a
2e681f3c1e0908ef7b
```

PAY TO PUBLIC KEY HASH (P2PKH)

- Similarly to P2PK, encodes that only somebody with knowledge of a private key corresponding to a certain public key P can spend the funds.
- However, the address format contains an encoding of the output of a “hashed” public key A = RIPEMD160(SHA256(P)). This double hashing process is also called hash160.
  SHA-256: We know already.
  RIPEMD160: “RIPE Message Digest” with 160 bit output.
  - Developed in EU project 1992-1996
  - Worse cryptographic hash function properties than SHA-256

- Result (A) is encoded in **base58check encoding** which combines
  - Smaller number of digits than hex
  - Better human readability
  - Error detection capability

Example: `1E86A5E6ANEVPuayP2XLGVzsXjaxT5MbRm` (Block explorer)

BASE58 CHECK ENCODING

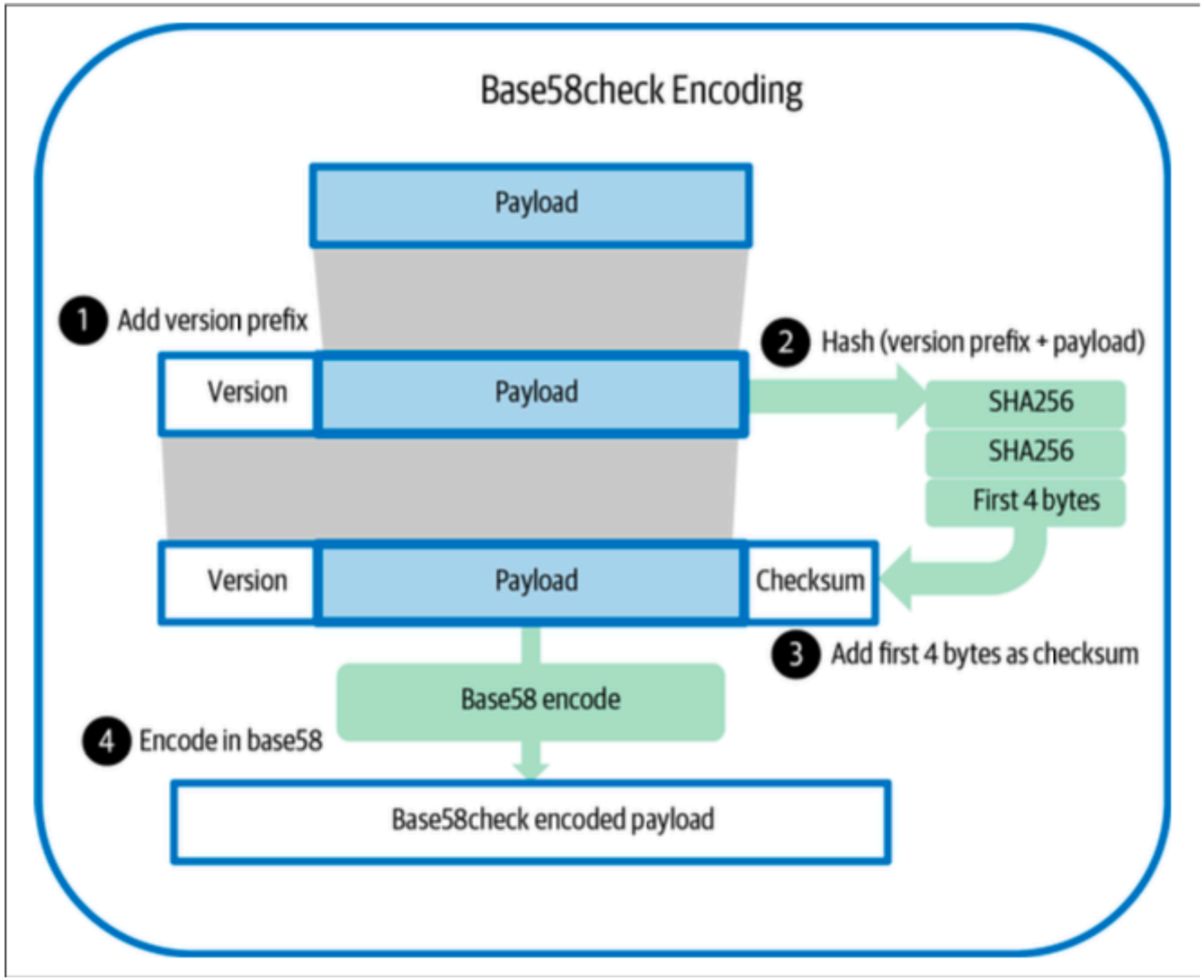
- Uses alphabet of lowercase and capital letters and digits 1-9: 123456789ABCDEFGHJKLMNPQRSTUVWXYZabcdefghijkmnopqrstuvwxyz

- Avoids usage of “0”, “O”, “I” and “l” (lowercase L) to mitigate potential confusion

- Appends checksum via double application of SHA256 to input prepended with prefix before encoding:

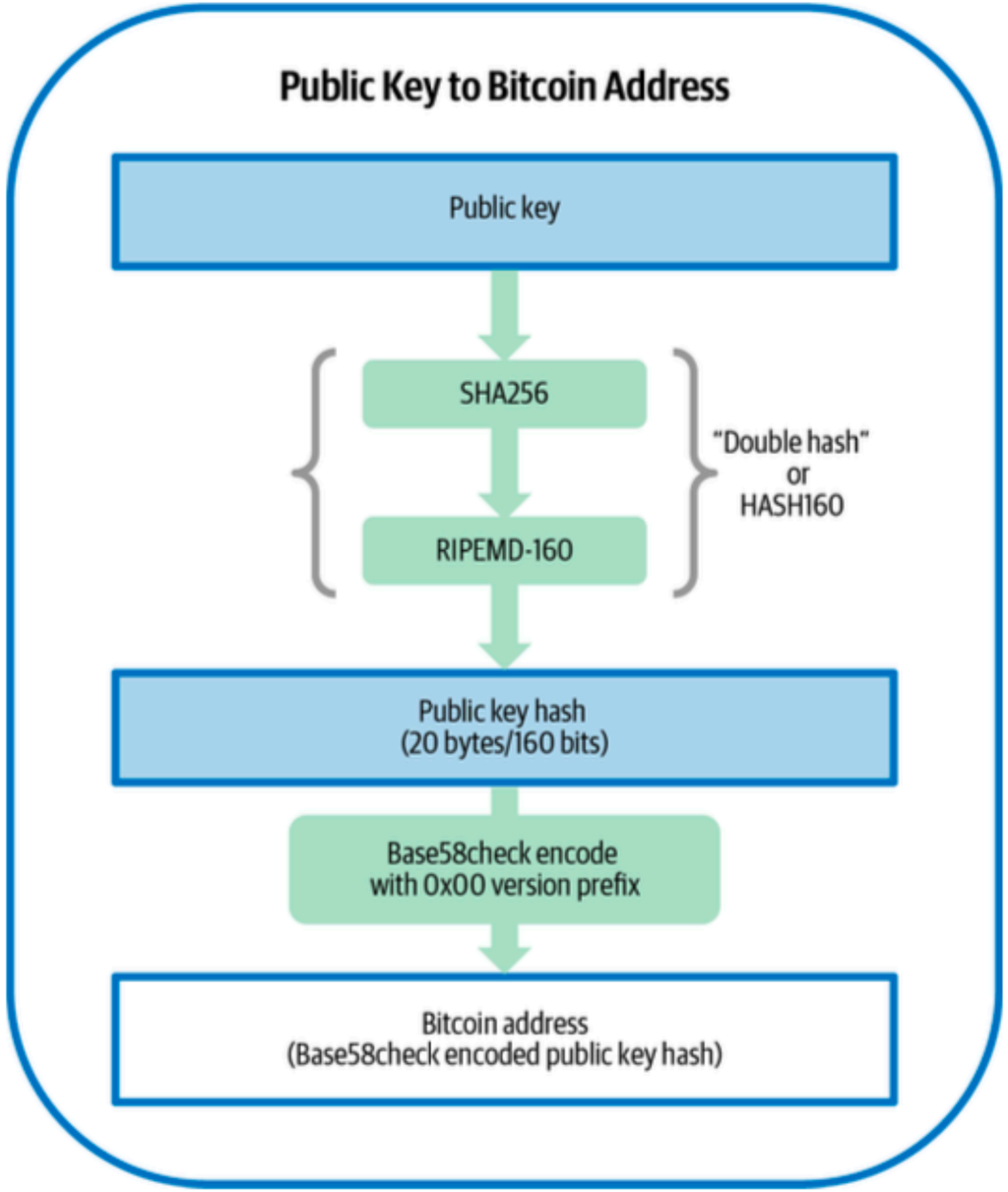
  checksum = $SHA256(SHA256(prefix||input))$

BASE58 CHECK ENCODING



Flow of Base58check encoding. Above, we had Payload = public key hash.

BASE58 CHECK ENCODING OF PUBLIC KEY (HASHES) IN P2PKH



Result has length of 26-35 characters.

P2SH ADDRESSES

- For “Pay to Script Hash” (P2SH), instead of the HASH160-output of a public key, the HASH160-output of a “script” is base58check-encoded.
- Script is written in the Bitcoin Script language. Example:

```
<public key 1> OP_CHECKSIGVERIFY <public key 2> OP_CHECKSIG
```

Table 4-1. Base58check version prefix and encoded result examples

Type	Version prefix (hex)	Base58 result prefix
Address for pay to public key hash (P2PKH)	0x00	1
Address for pay to script hash (P2SH)	0x05	3
Testnet Address for P2PKH	0x6F	m or n
Testnet Address for P2SH	0xC4	2
Private Key WIF	0x80	5, K, or L
BIP32 Extended Public Key	0x0488B21E	xpub

Example: 3CK4fEwbMP7heJarmU4eqA3sMbVJyEnU3V [see second output in this transaction]

BECH32 ADDRESS FORMATS

- **Issues of base58check encoded addresses:**
  - Mixed-case representation is inconvenient to read aloud
  - No error correction (only error detection)
  - Inconvenient to encode in QR codes

- **In 2017 SegWit and 2021 Taproot upgrades:**
  - Bech32 addresses (https://en.bitcoin.it/wiki/Bech32)
  - Only contains single-case letters and numbers
  - Uses BCH code correction algorithm (Bose–Chaudhuri–Hocquenghem codes) Guarantees error correction capability for up to 4
  - “Native SegWit format” defined in BIP 173 (for P2WPKH/P2WSH)
    - Starts with “bc1q…”, 42 characters (P2WPKH) or 62 characters (P2WSH)
  - “SegWit v1 format” / Taproot addresses (P2TR)
    - Starts with “bc1p…”, 62 characters

BECH32 ADDRESS FORMATS

- **“Native SegWit format” defined in BIP 173**
  - Introduced with 2017 SegWit upgrade
  - Starts with “bc1q…”
  - 42 characters (for P2WPKH) or 62 characters (for P2WSH)

Example: `bc1q34aq5drpuwy3wgl9lhup9892qp6svr8ldzyy7c` (Block explorer)

- **“SegWit v1 format” defined in BIP 350**
  - Introduced with November 2021 Taproot upgrade (correspond to P2TR)
  - Also called Bech32m addresses. Format due to one changed constant in encoding compared to Bech32
  - Starts with “bc1p…”, 62 characters

Example: `bc1p3fyk7j2c4tfvk99ezhmud5d4decperkhsdhqw0l55wy6dn6hpsgq8uahs2` (Block explorer)

OVERVIEW OF BITCOIN ADDRESS FORMATS

Type	First Seen	BTC Supply*	Use*	Encoding	Prefix	Characters
P2PK	Jan 2009	9% (1.7M)	Obsolete			
P2PKH	Jan 2009	43% (8.3M)	Decreasing	Base58	1	26 – 34
P2MS	Jan 2012	Negligible	Obsolete			
P2SH	Apr 2012	24% (4.6M)	Decreasing	Base58	3	34
P2WPKH	Aug 2017	20% (3.8M)	Increasing	Bech32	bc1q	42
P2WSH	Aug 2017	4% (0.8M)	Increasing	Bech32	bc1q	62
P2TR	Nov 2021	0.1% (0.02M)	Increasing	Bech32m	bc1p	62

Source: https://unchained.com/blog/bitcoin-address-types-compared/

Structure of Bitcoin transactions

EXPLORATIONS FOR UNDERSTANDING BITCOIN TRANSACTIONS

- Tool: Check on: https://learnmeabitcoin.com/technical/transaction/

THE STRUCTURE OF A BITCOIN TRANSACTION

Structure of “Legacy” transactions (without any P2WPKH/P2WSH/P2TR inputs or outputs, standard before 2016):

- Version (4 bytes, little-endian)
- Inputs (variable), see breakdown here
- Outputs (variable), see breakdown here
- Locktime (4 bytes, little-endian integer):
  If 499,999,999 or below: Indicates block height after which transaction can be mined
  If 500,000,000 or above: Indicates Linux time after which transaction can be mined