# Blockchain Technology

Subba Rao Y V

University of Hyderabad

March 22, 2018

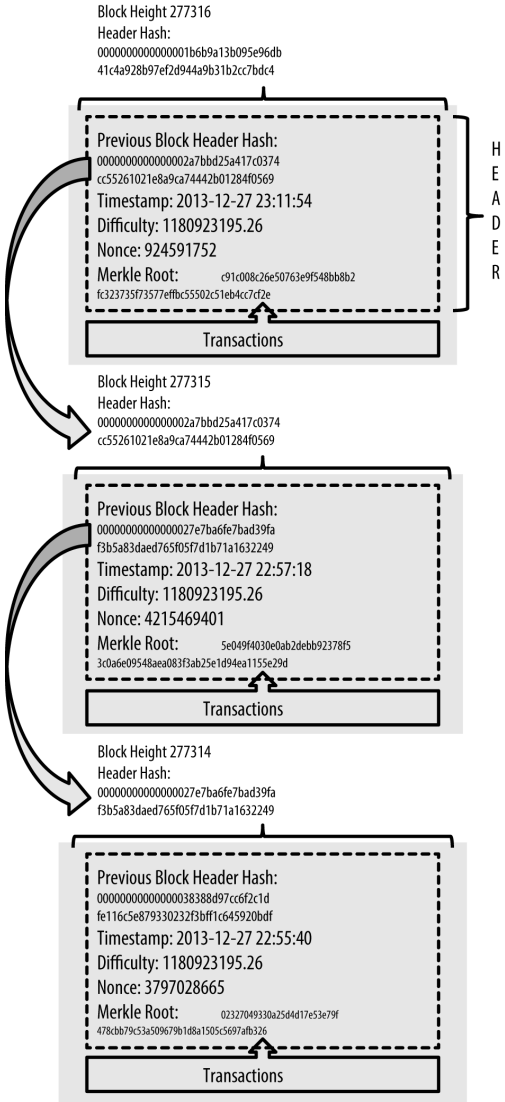
# Structure of the Block

| Size | Field | Description |
|---|---|---|
| 4 bytes | Block Size | The size of the block, in bytes, following this field |
| 80 bytes | Block Header | Several fields form the block header |
| 1–9 bytes (VarInt) | Transaction Counter | How many transactions follow |
| Variable | Transactions | The transactions recorded in this block |

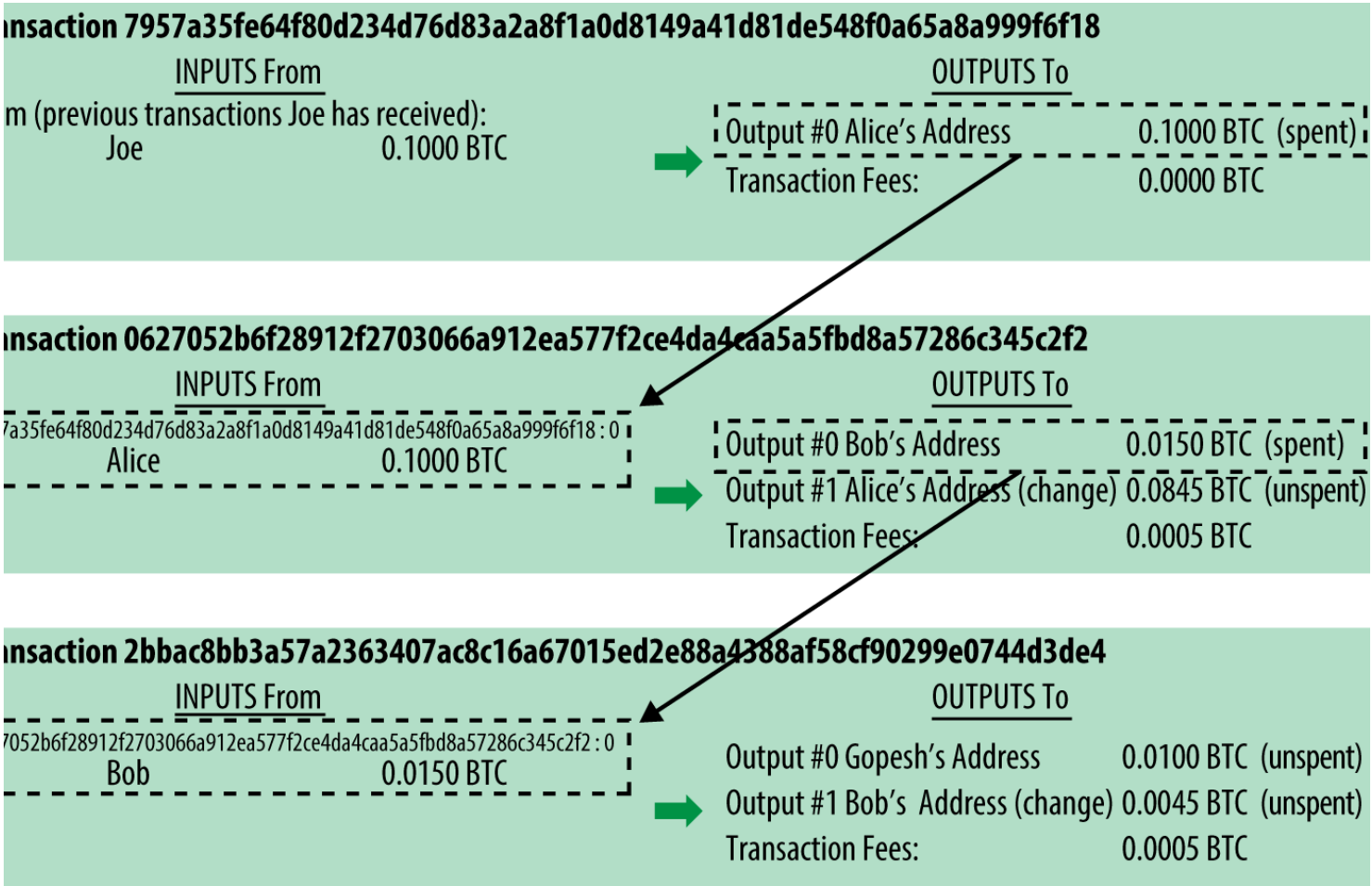
# Structure of the Header

| Size | Field | Description |
|---|---|---|
| 4 bytes | Version | A version number to track software/protocol upgrades |
| 32 bytes | Previous Block Hash | A reference to the hash of the previous (parent) block in the chain |
| 32 bytes | Merkle Root | A hash of the root of the merkle tree of this block's transactions |
| 4 bytes | Timestamp | The approximate creation time of this block (seconds from Unix Epoch) |
| 4 bytes | Difficulty Target | The Proof-of-Work algorithm difficulty target for this block |
| 4 bytes | Nonce | A counter used for the Proof-of-Work algorithm |

# Linking Blocks



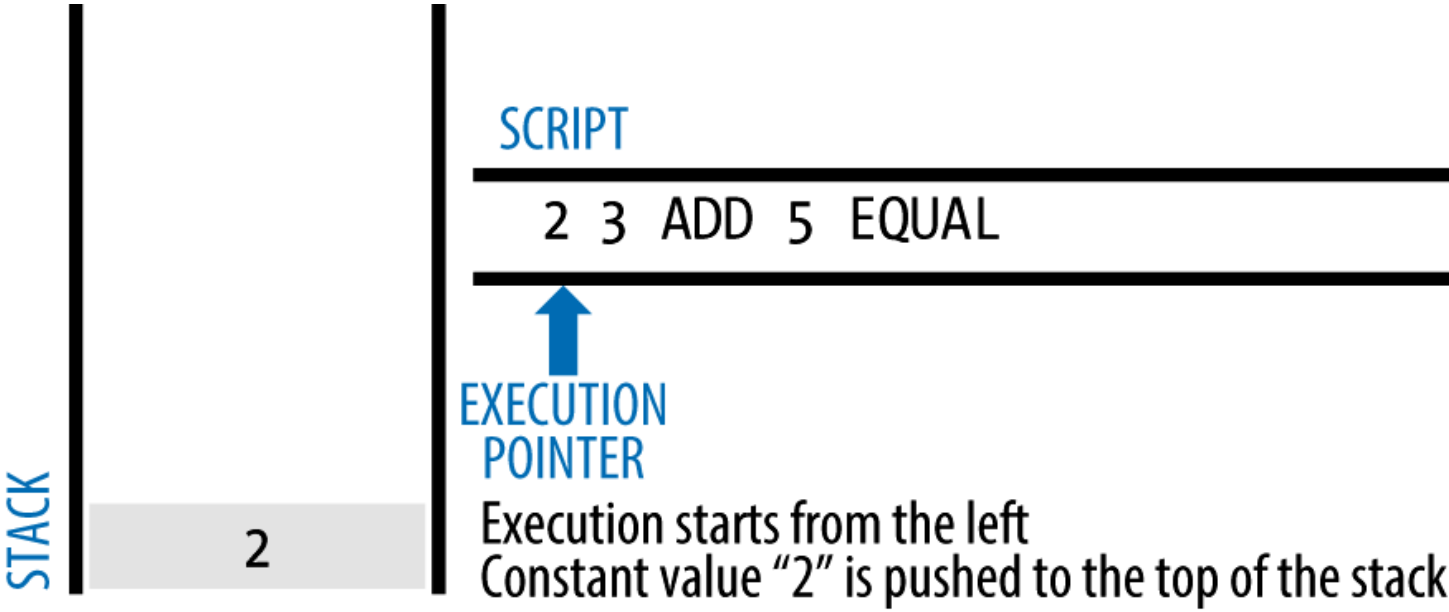
# Transaction Sequence



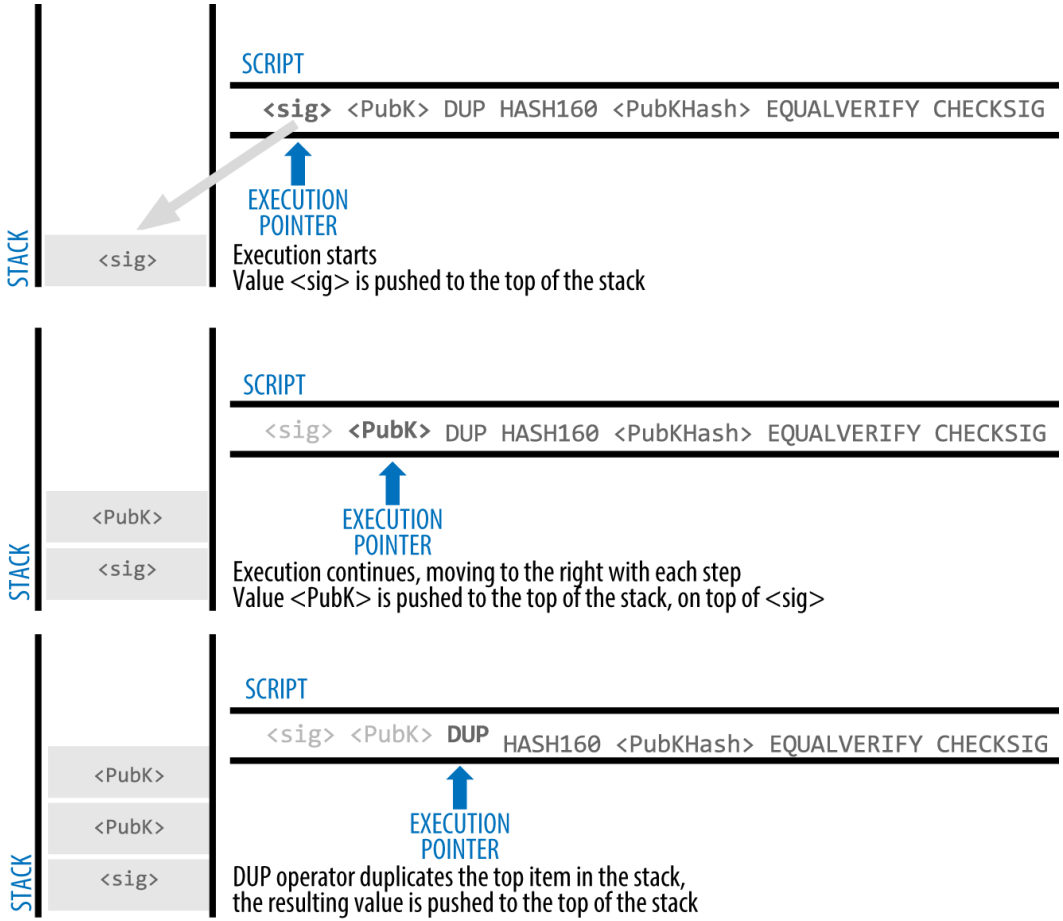
# Transaction


```
{
"version": 1,
"locktime": 0,
"vin": [
{
"txid":
"7957a35fe64f80d234d76d83a2a8f1a0d8149a41d81de548f0a65a8a999f6f18",
"vout": 0,
"scriptSig" :
"3045022100884d142d86652a3f47ba4746ec719bbfbd040a570b1deccbb6498c75c4ae24cb02204
b9f039ff08df09cbe9f6addac960298cad530a863ea8f53982c09db8f6e3813[ALL]
0484ecc0d46f1918b30928fa0e4ed99f16a0fb4fde0735e7ade8416ab9fe423cc5412336376789d1
72787ec3457eee41c04f4938de5cc17b4a10fa336a8d752adf",
"sequence": 4294967295
}
],
"vout": [
{
"value": 0.01500000,
"scriptPubKey": "OP_DUP OP_HASH160
ab68025513c3dbd2f7b92a94e0581f5d50f654e7 OP_EQUALVERIFY OP_CHECKSIG"
},
{
"value": 0.08450000,
"scriptPubKey": "OP_DUP OP_HASH160
7f9b1a7fb68d60c536c2fd8aeaa53a8f3cc025a8 OP_EQUALVERIFY OP_CHECKSIG",
}
]
}
```

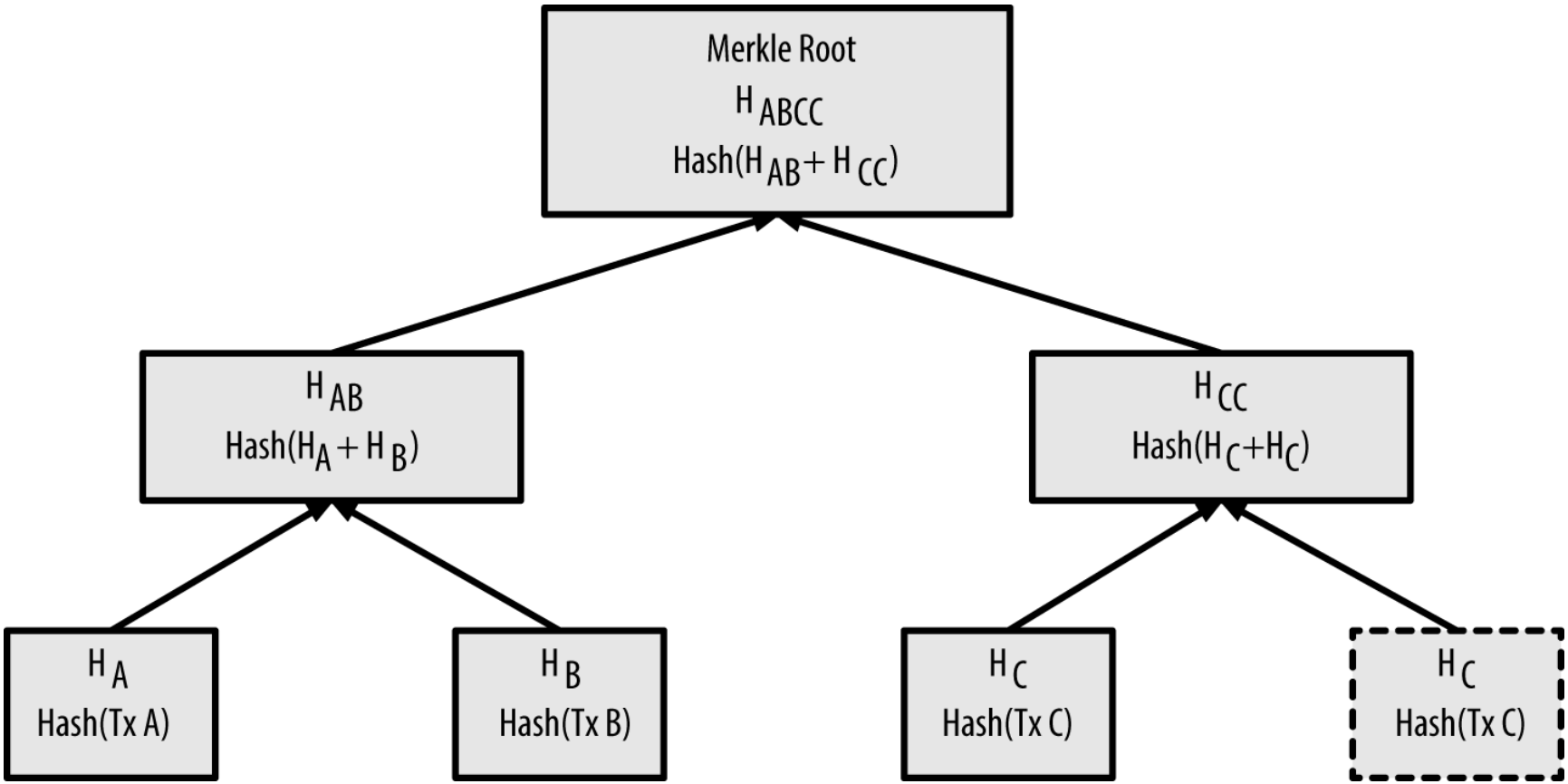
# A Simple Script



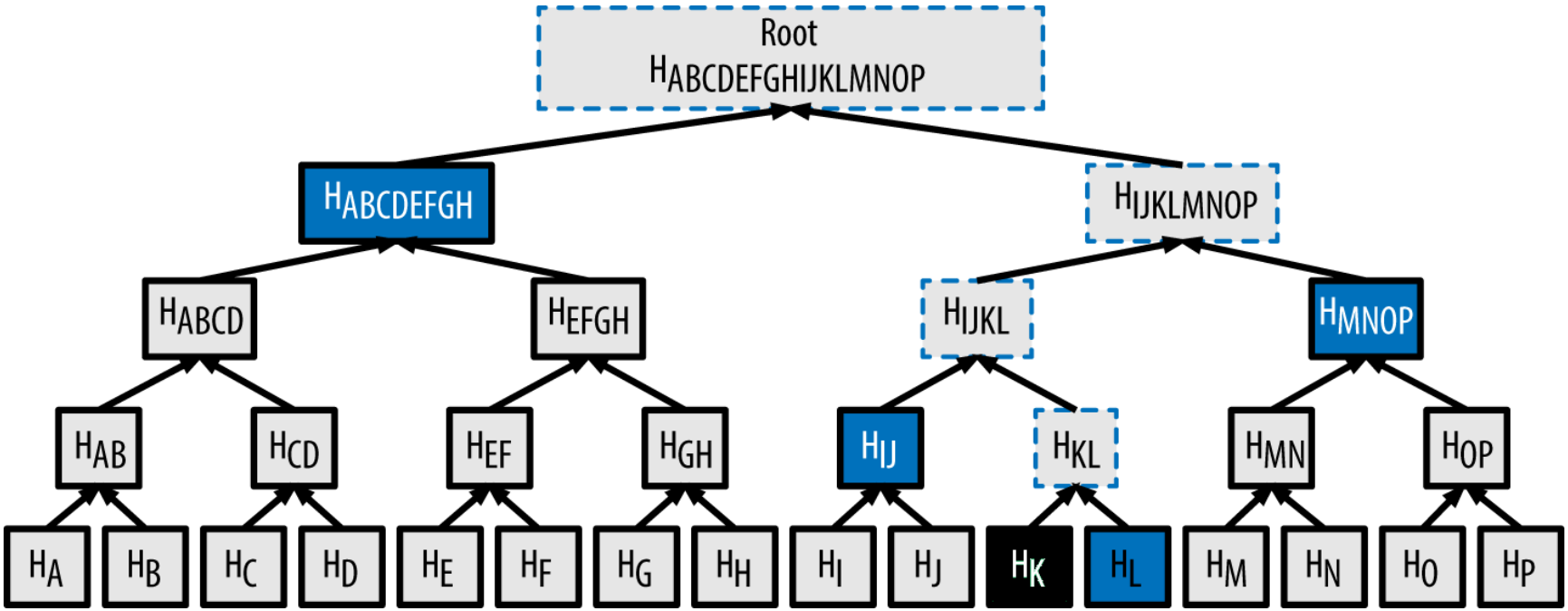
# A Sample Script



# Merkle Tree


Merkle Root
H ABCC
Hash(H AB + H CC)
H AB
Hash(H A + H B)
H CC
Hash(H C+H C)
H A
Hash(Tx A)
H B
Hash(Tx B)
H C
Hash(Tx C)
H C
Hash(Tx C)

# Merkle Path


Root
HABCDEFGHIJKLMNOP
HABCDEFGH
HIJKLMNOP
HABCD
HEFGH
HIJKL
HMNOP
HAB
HCD
HEF
HGH
HIJ
HKL
HMN
HOP
HA
HB
HC
HD
HE
HF
HG
HH
HI
HJ
HK
HL
HM
HN
HO
HP

# Thank you

# SHA-256 & ECDSA

Dr. Subba Rao Y. V.

March 2018

# Introduction - *SHA* – 256

- An *n*-bit hash is a map from an arbitrary length messages to *n*-bit *hash values*
- An *n*-bit cryptographic hash is an *n*-bit hash which is *One-way* and *Collision-resistant.*[1]
- *SHA* – 256 is a member of the *SHA* – 2 cryptographic hash functions designed by the NSA.[2].
- It is used as proof of work algorithm and in the creation of bitcoin addresses [2].

# Preprocessing [1]

- Pad the message as follows:
  - Let $M$ be message of length $l$ bits.
  - Append the bit "1" to the end of message followed by $k$ zero bits where $k$ is the smallest non-negative solution to the equation $l + 1 + k \equiv 448 \, mod \, 512$.
  - To this append 64−bit block which is equal to the number $l$ written in binary.
- Parse the message into $N$ 512−bit blocks $M^{(1)}, M^{(2)}, ..., M^{(N)}$.. The first 32 bits of message block $i$ are denoted $M_0^{(i)}$, the next 32 bits are $M_1^{(i)}$ and so on upto $M_{15}^{(i)}$. (big-endian convention is used)

# Preprocessing [1]

The **initial hash value** $H^{(0)}$ is the following sequence of 32−bit words.

$$H_1^{(0)} = 6a09e667$$
$$H_2^{(0)} = bb67ae85$$
$$H_3^{(0)} = 3c6ef372$$
$$H_4^{(0)} = a54ff53a$$
$$H_5^{(0)} = 510e527f$$
$$H_6^{(0)} = 9b05688c$$
$$H_7^{(0)} = 1f83d9ab$$
$$H_8^{(0)} = 5be0cd19$$

# Algorithm [1]

For $i = 1$ to $N$ ($N$ = number of blocks in the padded message)

- Initialize registers $a, b, c, d, e, f, g, h$ with the $(i-1)^{st}$ intemediate hash value (= the initial hash value when $i = 1$)
  $a \leftarrow H_1^{(i-1)}$
  $b \leftarrow H_2^{(i-1)}$
  .
  .
  $h \leftarrow H_8^{(i-1)}$
- Apply the **SHA-256 compression function** to update registers $a, b, ..., h$
  For $j = 0$ *to* $63$

  compute $Ch(e, f, g)$, $Ma(a, b, c)$,$\sum_0(a)$,$\sum_1(e)$ and $W_j$(Def in later slides)
  $T_1 \leftarrow h + \sum_1(e) + Ch(e, f, g) + K_j + W_j$
  $T_2 \leftarrow \sum_0(a) + Ma(a, b, c)$
  $h \leftarrow g$
  $g \leftarrow f$
  $f \leftarrow e$

$$e \leftarrow d + T_1$$
$$d \leftarrow c$$
$$c \leftarrow b$$
$$b \leftarrow a$$
$$a \leftarrow T_1 + T_2$$

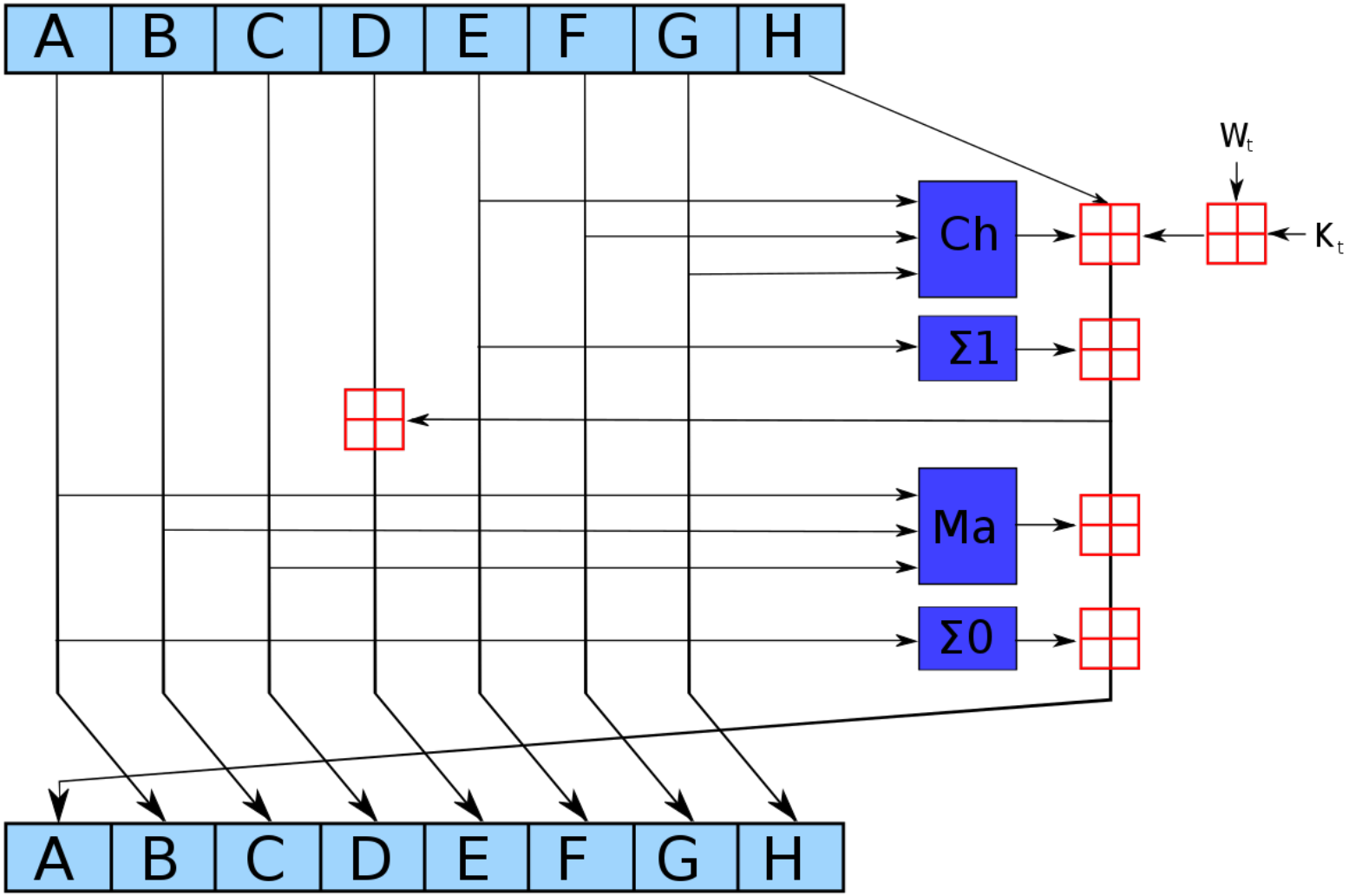
- Compute the $i^{th}$ intemediate hash vale $H^{(i)}$

$$H_1^{(i)} \leftarrow a + H_1^{(i-1)}$$
$$H_2^{(i)} \leftarrow b + H_2^{(i-1)}$$
.

.

$$H_1^{(i)} \leftarrow h + H_1^{(i-1)}$$

$H^{(N)} = (H_1^{(N)}, H_2^{(N)}, ..., H^{(N)})$ is the hash of $M$




SHA 256[3]

The red box is addition modulo $2^{32}$ for $SHA-256$

# Definitions

One iteration in $SHA-2$ family compression function [3].

- $Ch(E, F, G) = (E \wedge F) \oplus (\neg E \wedge G)$
- $Ma(A, B, C) = (A \wedge B) \oplus (A \wedge C) \oplus (B \wedge C)$
- $\sum_0(A) = (A \ggg 2) \oplus (A \ggg 13) \oplus (A \ggg 22)$
- $\sum_1(E) = (E \ggg 6) \oplus (E \ggg 11) \oplus (E \ggg 25)$

# Elliptic Curve Digital Signature Algorithm (ECDSA) [5]

ECDSA is an asymmetric key algorithm for digital signature. Asymmetric Key, also known as public key cryptography, uses public and private keys to encrypt and decrypt data. The security of ECDSA depends on the Elliptic curve discrete logarithm problem (ECDLP) over finite field [4].

**Definition:** Elliptic curve discrete logarithm problem (ECDLP):

- Let $A, B$ be two points on Elliptic curve $E$ defined over finite field $F_p$, for large prime $p$, such that point $A$ having prime order $q$ and $B = xA$; for some integer $x$ with $0 < x < q$;
- The Elliptic curve discrete logarithm problem (ECDLP) is to find $x$ where $x = \log_A B$ and $B, A$ are public and $x$ is secret key.

Since solving ECDLP involves solving DLP, ECDLP becomes significantly harder problem than DLP even for comparable key sizes[5].

# Phases of ECDSA

ECDSA has the following three phases:

- Key Generation Phase
- Signature Creation Phase
- Signature Verification Phase

# Phases of ECDSA

**Key Generation Phase:**

1. The given elliptic curve $y^2 = x^3 + ax + b \mod p$, has a point $A$ which generates a cyclic group of prime order $q$;
2. Select secret integer $x$ with $0 < x < q$;
3. Calculate $B = xA$;
4. Public keys are $= p, a, b, q, A$ and $B$;
5. Private key $= x$.

# Phases of ECDSA

**Signature Creation Phase**

1. Choose a secret random integer $k$ with $0 < k < q$;
2. Calculate $R = kA$;
3. Signature consists of a pair $(r, s)$,
   where $r = X_R$ and $s = (h(m) + xr)(k)^{-1} \mod q$.

**Signature Verification Phase**

1. Calculate $l = s^{-1} \mod q$;
2. Calculate $h = l * h(m) \mod q$;
3. Calculate $j = lr \mod q$;
4. Calculate $N = hA + jB$;
5. if $r \equiv X_N \mod q$ then signature is valid otherwise signature is invalid.

# Proof for Signature Verification

The Signature $(r, s)$ is valid if and only if it satisfies the condition

$$r \equiv X_N \mod q$$

$$\begin{aligned} s &= (h(m) + xr)(k)^{-1} \mod q \\ \implies k &= s^{-1}(h(m) + xr) \mod q \\ \implies k &= s^{-1}h(m) + s^{-1}xr \mod q \end{aligned} \tag{1}$$

we know the above that

$$l = s^{-1} \mod q$$

$$h = l * h(m) \mod q$$

$$h = s^{-1}h(m) \mod q$$

$$j = lr \mod q$$

;

$$j = s^{-1}r \mod q$$

# Proof contd..

Substitute value of $h$ and $j$ in equation 1

$$\implies k = h + xj \mod q$$

Multiplying point $A$ on both sides

$$\implies kA = (h + xj)A$$

$$\implies kA = hA + jB$$

we Know that

$$R = kA \quad and \quad N = hA + jB$$

Here $R = N$ resulting in above verification comparing $X$ coordinate of $R$ and $X$ coordinate of $N$.

# References


- http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf.
- https://en.bitcoin.it/wiki/SHA-256.
- Sha 256 wikipedia.
  https://en.wikipedia.org/wiki/SHA-2.
- Maike Massierer.
  The elliptic curve discrete logarithm problem.
- Jan Pelzl Christof Paar.
  *Understanding Cryptography*.
  Springer-Verlag Berlin Heidelberg, 2010.
  https://murdercube.com/files/Computers/Computer%20Security/Understanding%20Cryptography.pdf.