Blockchain Technology

Open Elective @ VJTI - Fall 2019

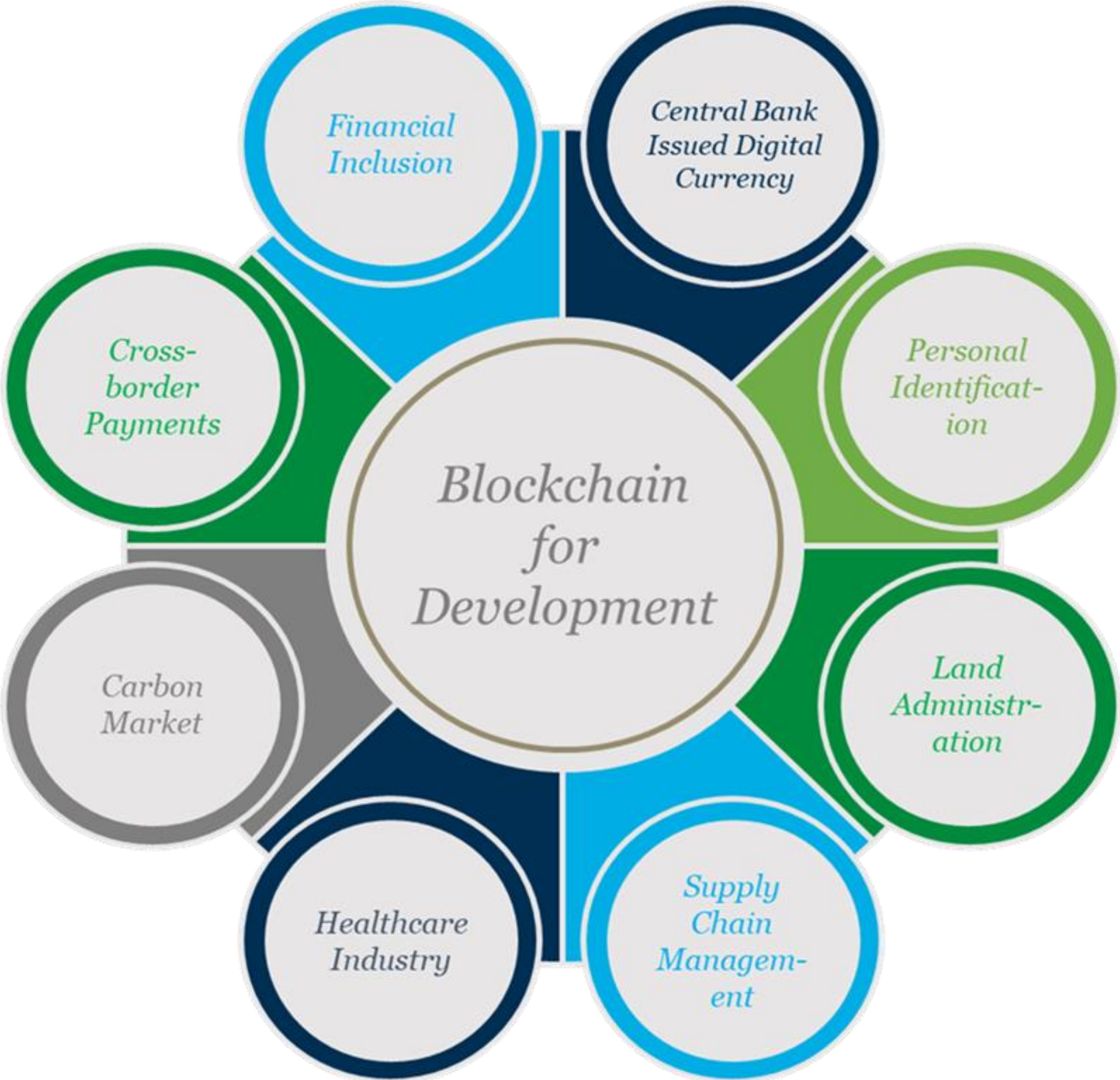
Lecture#2 and 3 (25 and 29 July 2019)

Dhiren Patel
VJTI Mumbai

- Pre-test
- Quick review
- Few more Slides
- Goals - Crypto foundations

Bitcoin Core issues

- Volatility, Usability
- No Ease of use (private key management)
- User community, Developer community
- Incentive to mine and validate?? (Energy cost, waste)
- Nothing for small User
- Trading interest
- Gaming hopes: Crypto Kitties – collecting kitties – digital assets – coin ether
- Prepaid small crowd funding – converted into lottery – treasure hunt


Financial Inclusion
Central Bank Issued Digital Currency
Cross-border Payments
Personal Identificat-ion
Blockchain for Development
Carbon Market
Land Administr-ation
Healthcare Industry
Supply Chain Managem-ent

Course Micro details (flexible)

- Introduction and Crypto foundations: Elliptic curve cryptography, ECDSA, Cryptographic hash functions, SHA-256, Merkle Trees, Crytpocurrencies (4 hrs)
- Bitcoin: Bitcoin addresses, Bitcoin's blockchain, block header, mining, proof of work (PoW) algorithms, difficulty adjustment algorithm, mining pools, transactions, double spending attacks, the 51% attacker, block format, pre-SegWit transaction formats, Bitcoin script, transaction malleability, SegWit transaction formats, smart contracts (escrow, micropayments, decentralized lotteries), payment channels, Lightning network (8-10 hrs)
- Ethereum: Overview of differences between Ethereum and Bitcoin, block format, mining algorithm, proof-of-stake (PoS) algorithm, account management, contracts and transactions, Solidity language, decentralized applications using Ethereum (4-6 hrs)
- Smart Contracts (4-6 hrs)
- Different Blockchains and Consensus mechanisms (4-6 hrs)
- Blockchain and Security: Attacks and countermeasures (4-6 hrs)
- R3, CORDA and Hyperledger System architecture, ledger format, chaincode execution, transaction flow and ordering, private channels, membership service providers, case studies (4-6 hrs)
- dApps – (6 hrs)
- Blockchain use cases and advanced topics (4-6 hrs)

Cryptography

- **Cryptography** is the **art and science of keeping information secure** from unintended audiences
- If you want to keep information **secret**, you have two possible strategies: **hide the existence of the information**, or **make the information unintelligible**.
- Cryptography - encrypting information
- Steganography – concealing (hiding) information
- Conversely, **cryptanalysis** is the **art and science of breaking encoded data**.
- The branch of mathematics encompassing both cryptography and cryptanalysis is cryptology.

Security objectives

- Cryptography, Information Security, Cyber Security, Network Security, Web Security
- Data Confidentiality (Encryption Algorithms)
- Data hiding (Steganography)
- Data Integrity (Hash functions)
- Authentication (Identity and Access Management)
- Non-repudiation (Digital signature)
- Security Policy, Vulnerability Assessment and Penetration Testing…

Attacks (pictorial on next slide)

- Theft of sensitive information
- Disruption of service
- Illegal access to resources
- E.g. Stealing Credit card details
- E.g. Ransomware
- E.g. Resource (Compute) Hijacking for Cryptocurrency Mining

Attacks

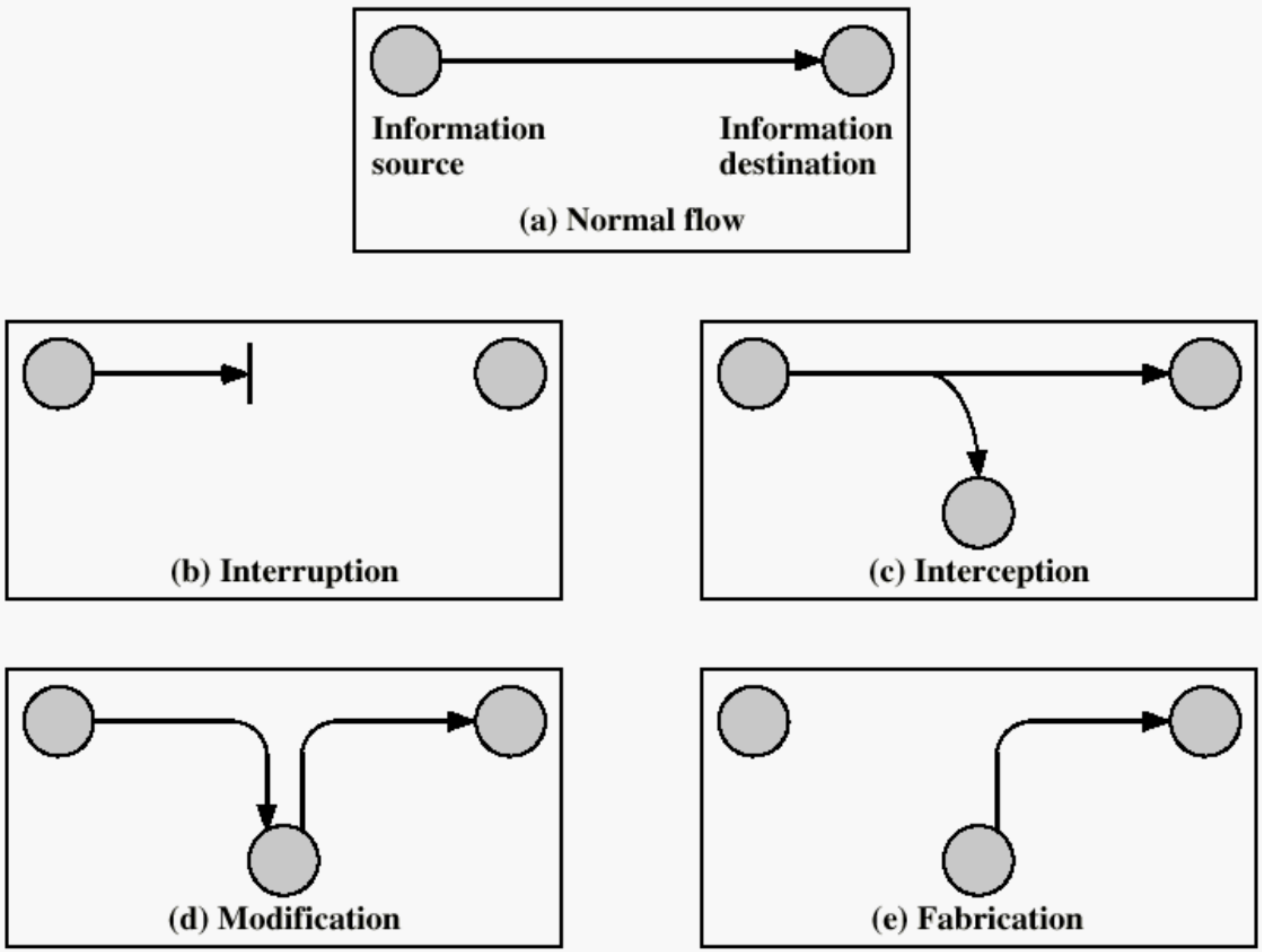


Figure 1.1 Security Threats

Crypto algorithms (Ciphers)

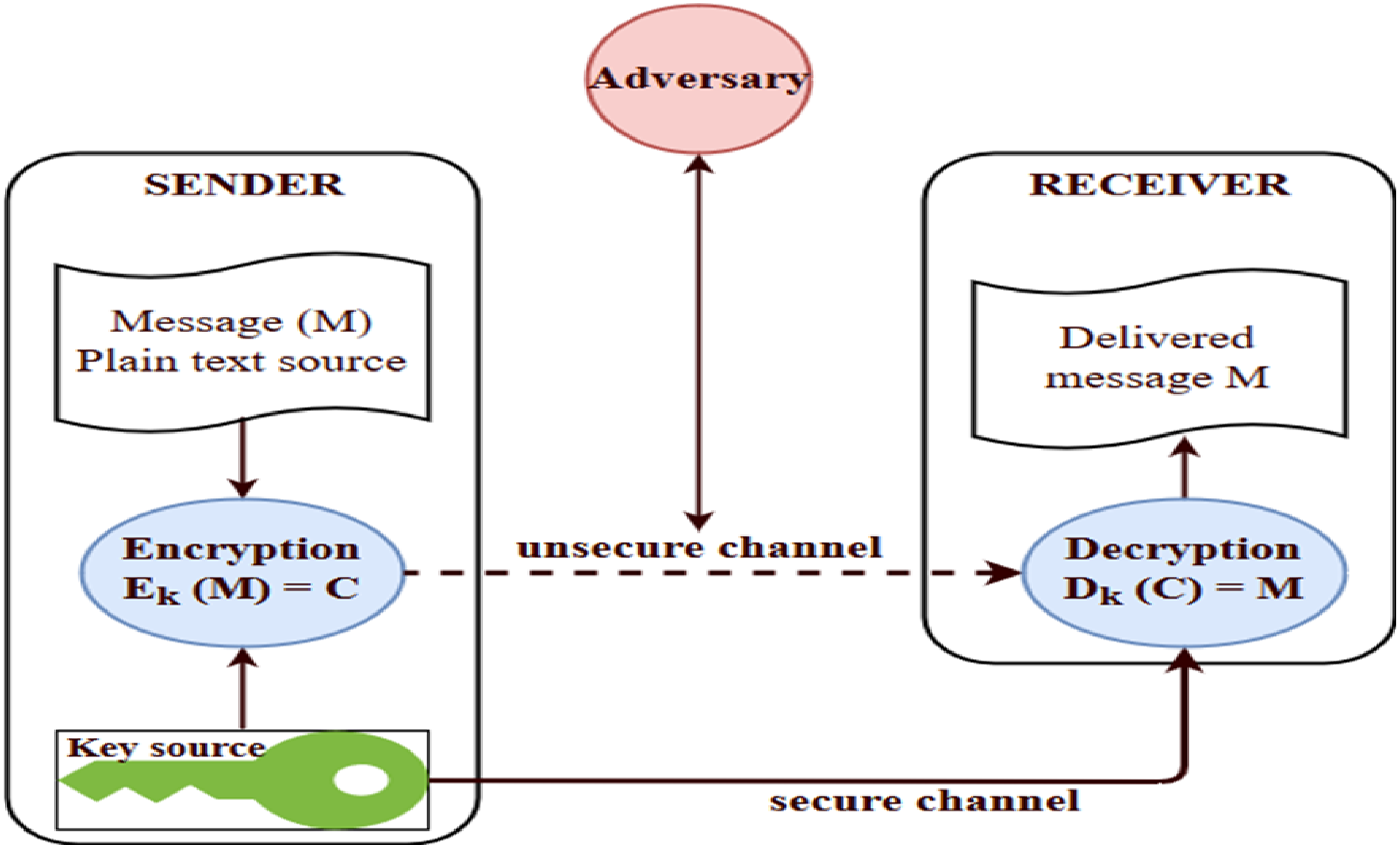
- symmetric key algorithm: the same key deciphers and encyphers the message
- $\mathbf{m = d_{key}(c_{key}(m))}$
- asymmetric key algorithm: different keys for encryption and decryption are required
- $\mathbf{m = d_{key2}(c_{key1}(m))}$
  - one key can be made public (public key), the other must be kept private (private key)
- Public-key cryptography (RSA Crypto and Elliptic Curve Crypto)

Cryptosystem: generic Definition

A CRYPTOSYSTEM is a 5-tuple (P,C,K,E,D) satisfying

1. P is a finite set of possible plaintexts
2. C is a finite set of possible ciphertexts
3. K is a finite set of possible keys
4. E is a finite set of encryption rules indexed by K so for each K there is a function $e_K : P \rightarrow C$
5. D is a finite set of decryption rules indexed by K so for each K there is a function $d_K : C \rightarrow P$

Symmetric key cryptography



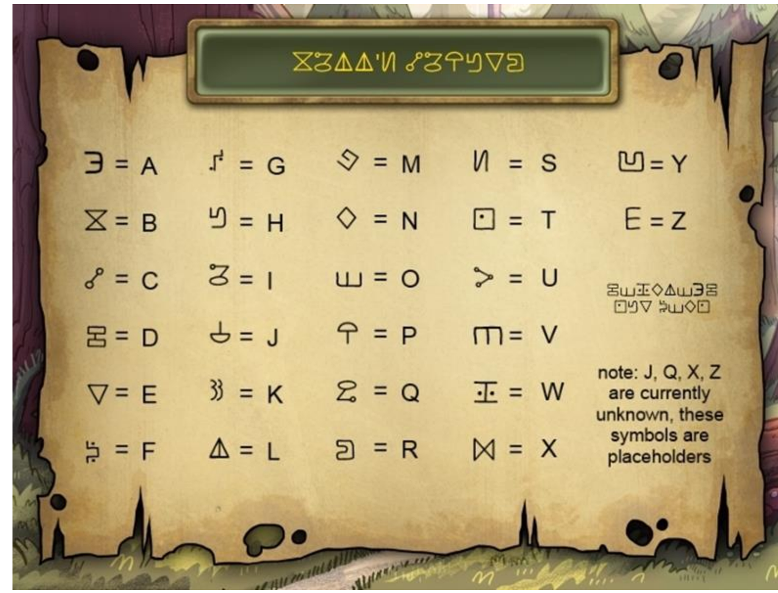
E.g. Symmetric key cipher: Shift Cipher

The **shift cipher** is the cryptosystem defined by taking

- $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$
- $e_K(x) = (x + K) \bmod 26$
- $d_K(y) = (y - K) \bmod 26$

Letters are identified with numbers: A=0, B=1, ..., Z=25

$\mathbb{Z}_{26}$ denotes the set {0, 1, ... , 25} with addition and multiplication taken modulo 26



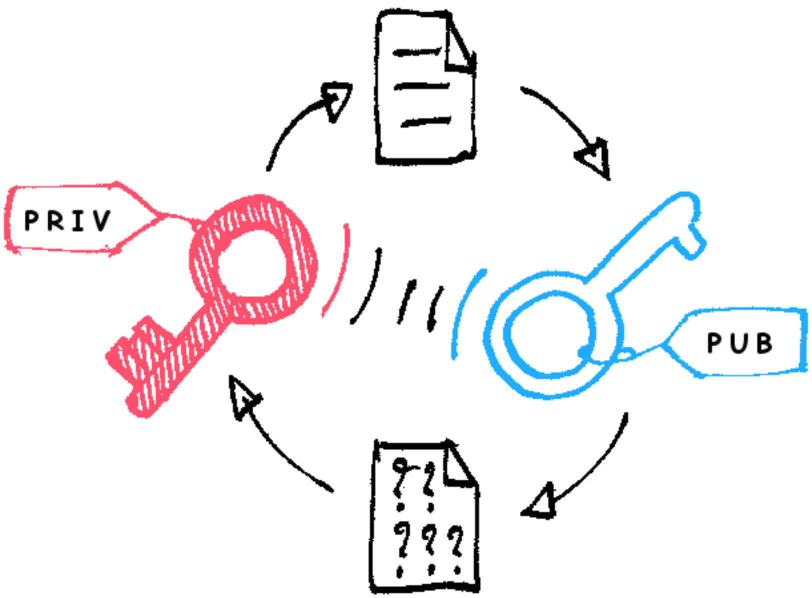
Caesar Cipher

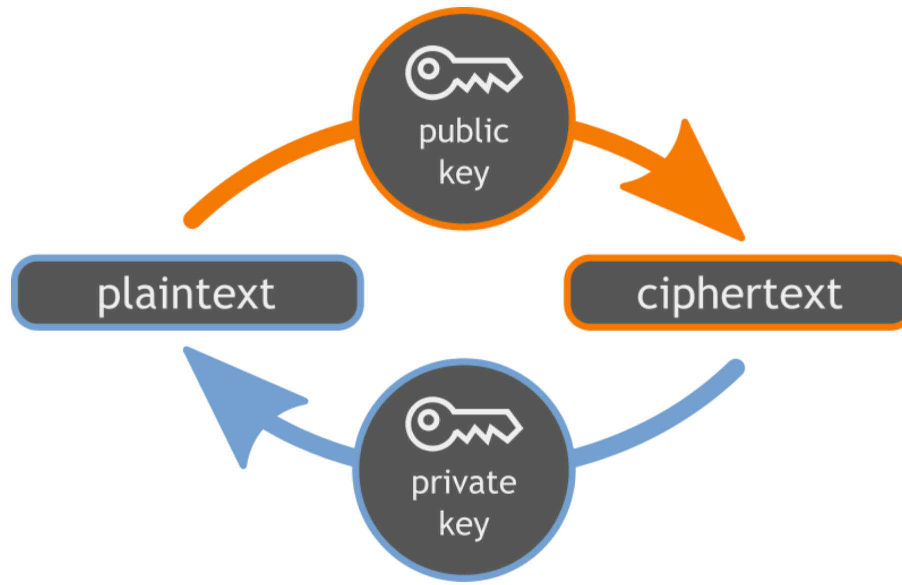
Block cipher

- A block cipher is a function which maps n-bit plaintext blocks to n-bit ciphertext blocks; n is called the **block length**.
  - E: $\{0,1\}^n \times \{0,1\}^k$ $\{0,1\}^n$
- Use of plaintext and ciphertext blocks of equal size avoids data expansion.
- The function is parameterized by a k-bit key.
- To allow unique decryption, the encryption function must be one-to-one (i.e., invertible)
- For n-bit plaintext and ciphertext blocks and a fixed key, the encryption function s a bijection (1-to-1 and on-to), defining a permutation on n-bit vectors.
- E.g. DES (Data Encryption Standard) – 64 bit symmetric key cipher, AES (Advance Encryption Standard) – 128 bit symmetric key cipher

Public key cryptography

(each user has a key-pair)
Public key is published – known to all
Private key is known to user himself/herself





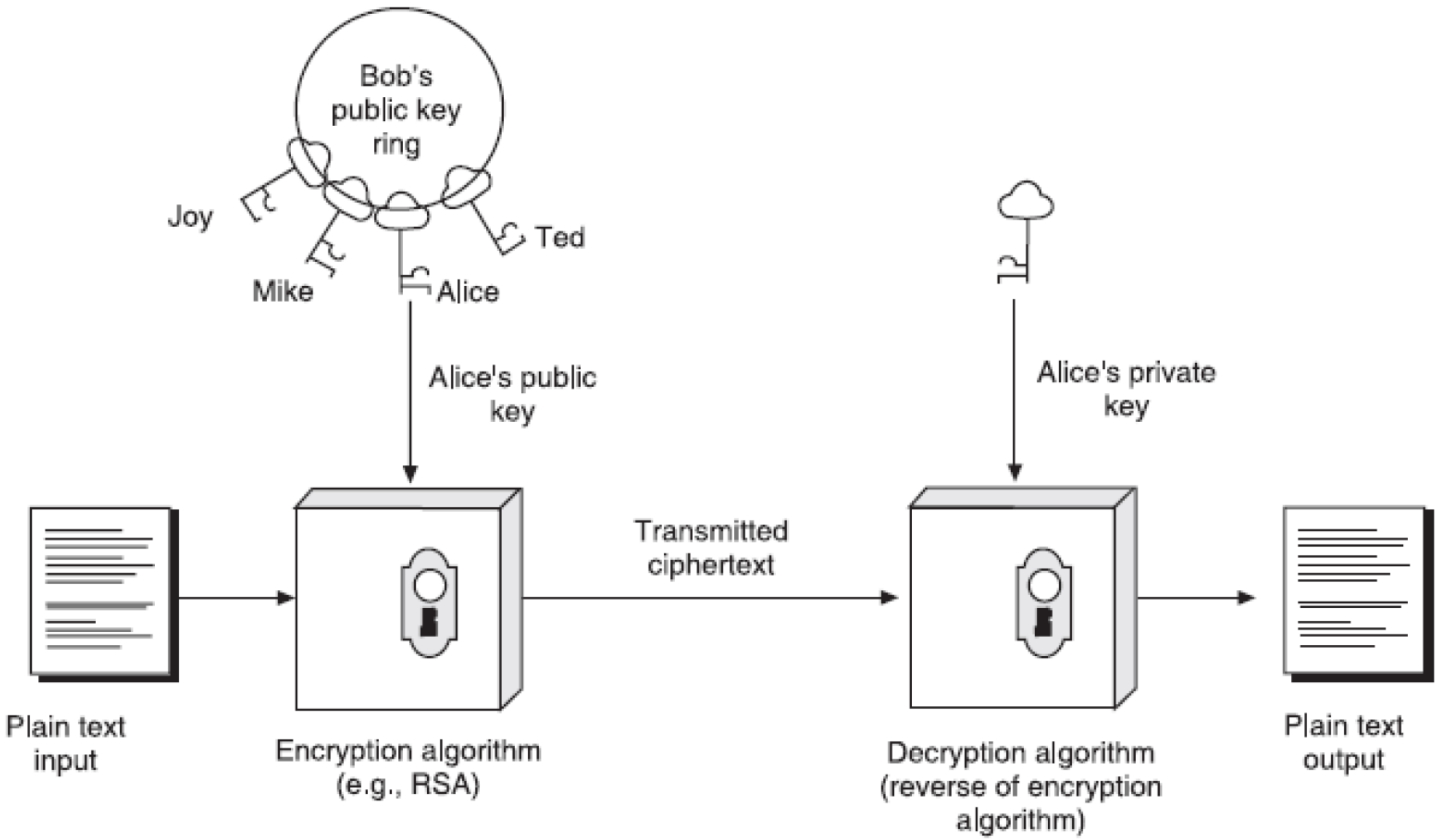
Confidentiality (encrypt with some one’s public key)
Entity Authentication (encrypt with own private key)

Key management issues

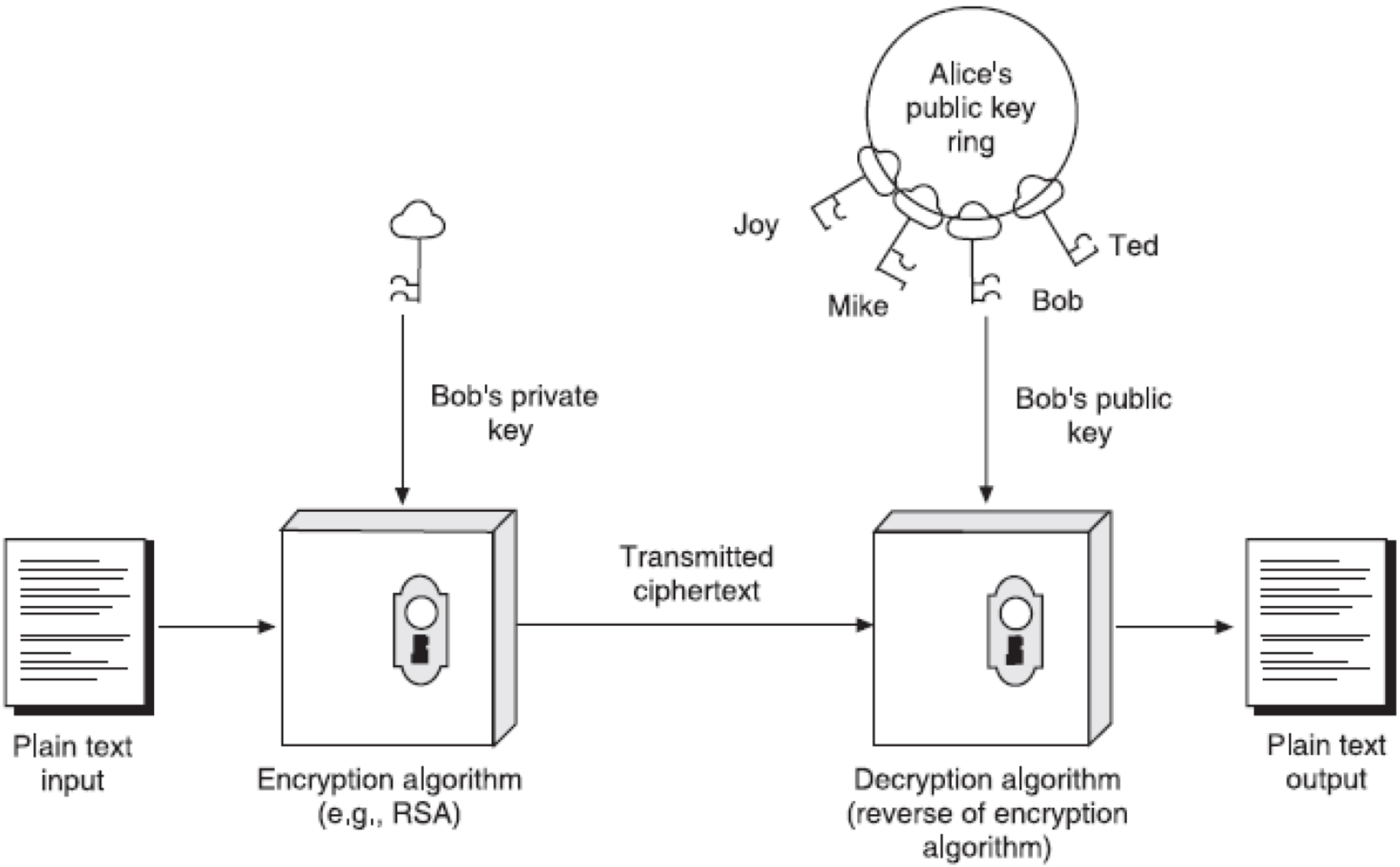
- SKC (Symmetric key cryptography)
- 1 user wishes to communicate with another user – requires 1 key (**secure key sharing??)**
- 1 user wishes to communicate with 2 users – requires 2 keys
- N users – wish to communicate to each other – require N*(N-1) keys!!! → O(n^2)
- PKC (Public Key Cryptography)
- N users require – 2N keys (each has pair of keys)!!
- (**un-secure key sharing**, public keys!!!)
- Key generation, Key exchange (distribution), Key management (expiry, revoking)

Public key ring

PKC Encryption

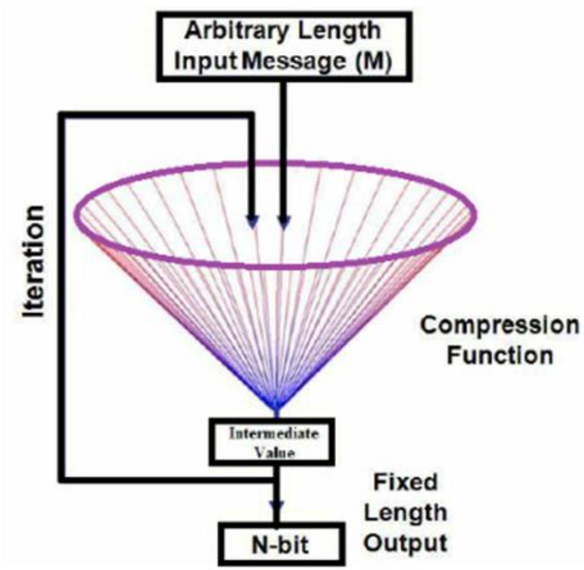


PKC in Authentication



Cryptographic hash functions

- **Hash Function**: takes input (of variable-length) and returns a fixed size output string h (usually much smaller than input)
  $H: \{0,1\}^* \rightarrow \{0,1\}^n, \quad h = H(M)$
- One way



- A block cipher is a function wh
  blocks to n-bit ciphertext block
  - $E: \{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$
- To allow unique decryption, the encryption function must be one-to-one

Properties of crypto Hash function

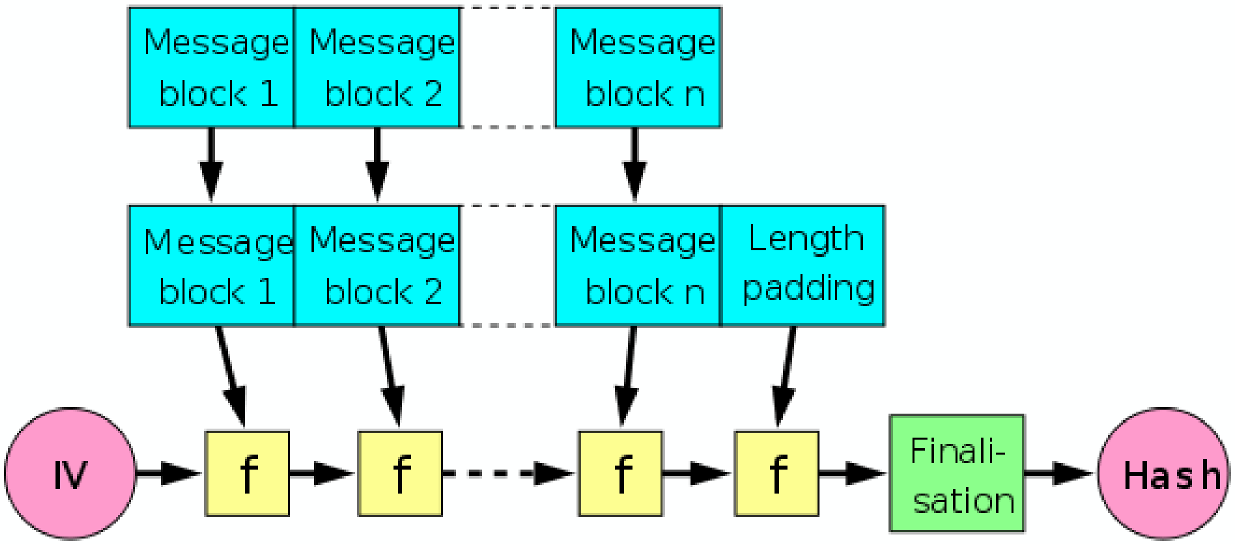
- H() should work on any input length
- H() should produce output of fixed size
- H() should be easy to compute
- Additionally one should understand
  - Compression ➔ leading to collisions (in theory)
  - Sparse (existence of collisions) over large input space
  - More bits – output lookup table too large
  - Weak collision resistance
  - Strong collision resistance

Hash Function construction

<u>Merkle-Damgard</u>:

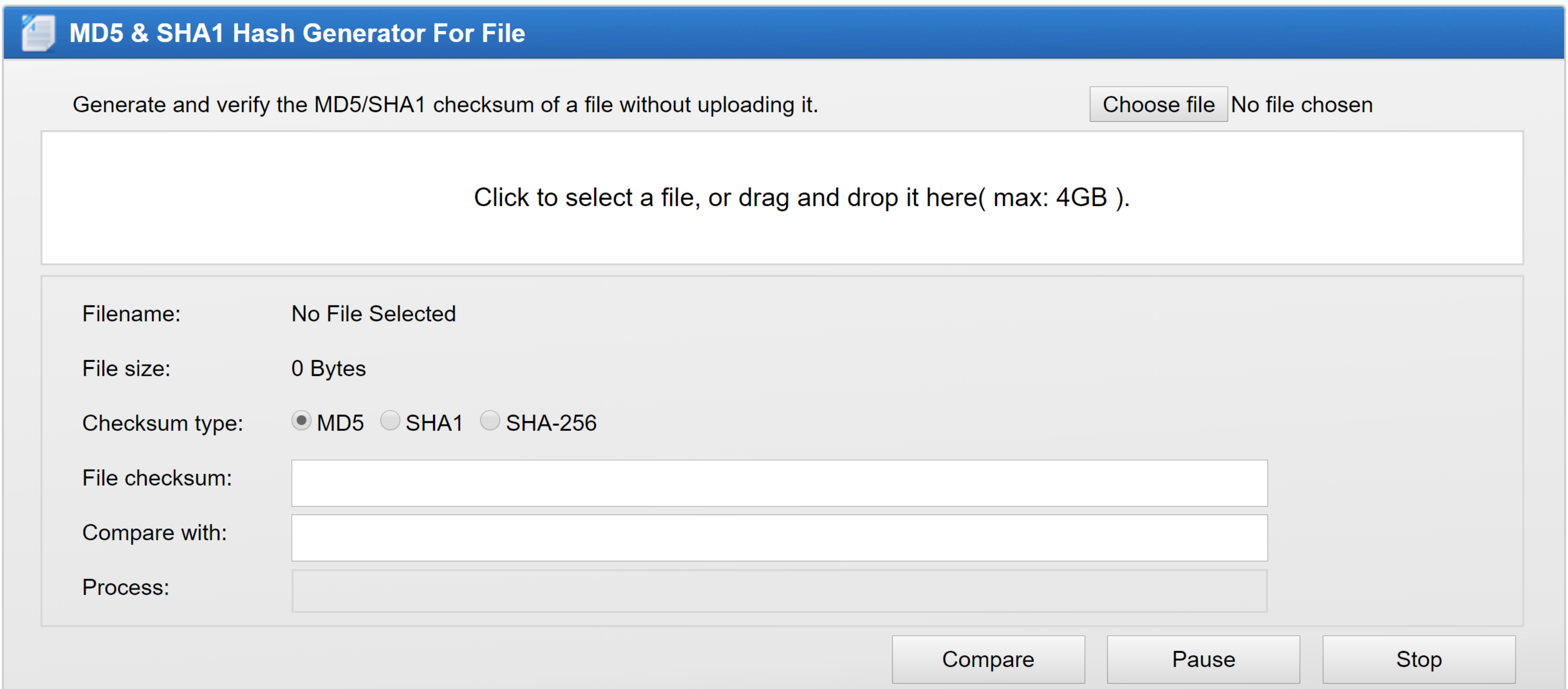
iterative application of compression function

- <u>MD-strengthening</u> → The procedure of fixing the *IV* and adding a representation of the length of input.



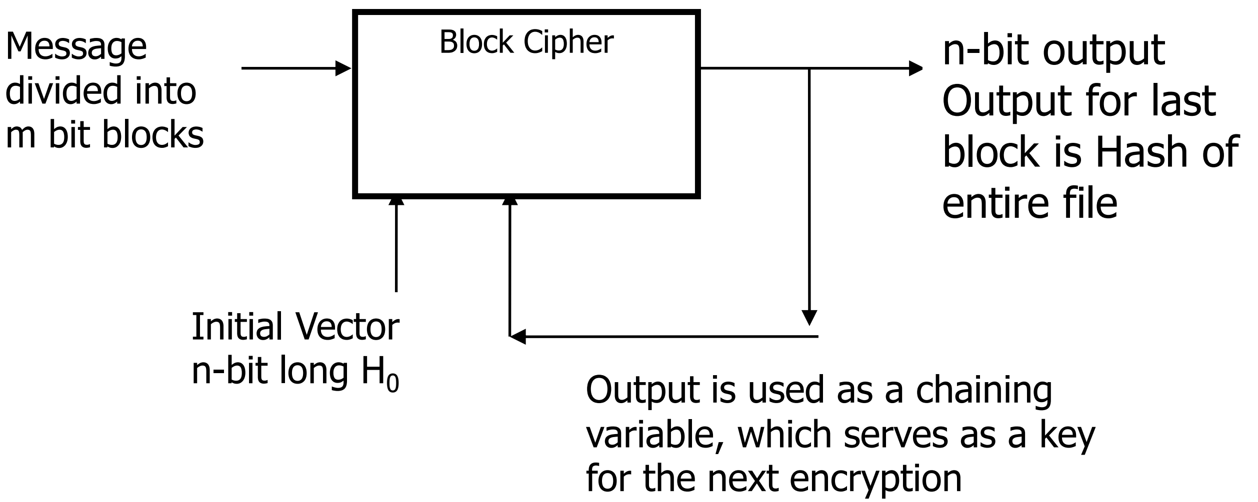
Hash function (demo – online)

- MD5, SHA-1, SHA-256 ... (demo) – observe change in hash code while making a single bit change in input file
- E.g. http://onlinemd5.com



HF construction – *using Block Cipher*

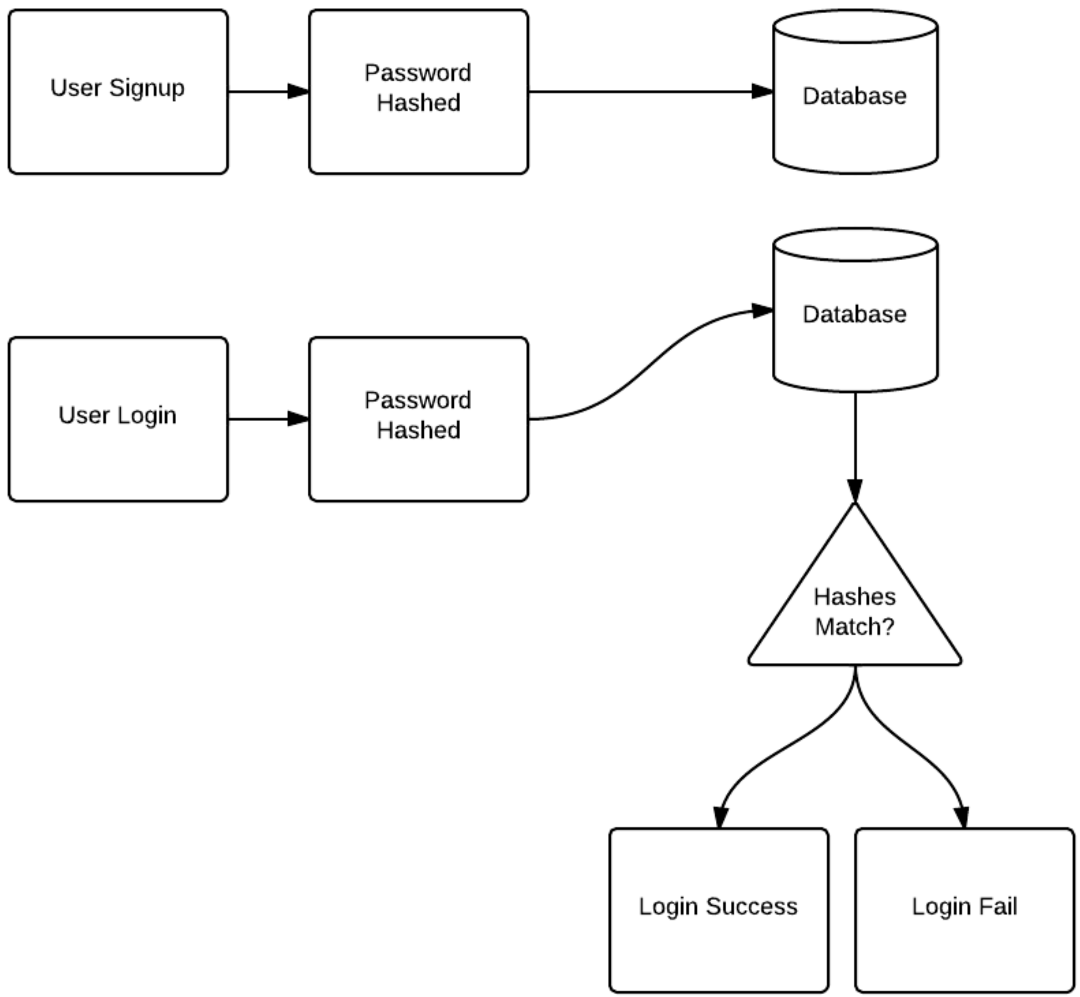
- Block cipher (standard or dedicated) in CBC mode
- Hash function H: $\{0,1\}^* \rightarrow \{0,1\}^n$
- Block cipher encryption E: $\{0,1\}^n \times \{0,1\}^k \rightarrow \{0,1\}^n$
- $H_i = H_{i-1} \oplus M_i$



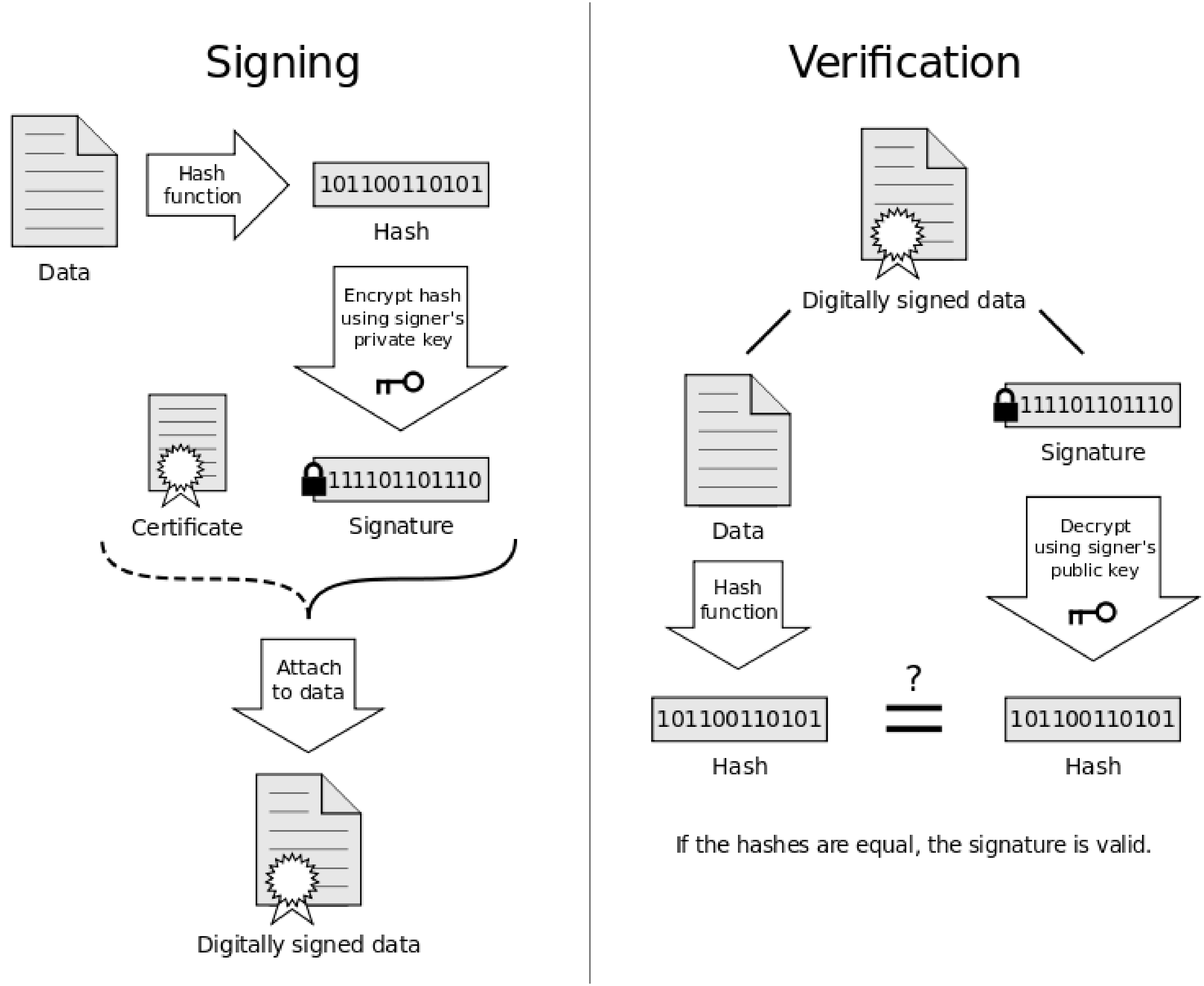
$$H_i = E_{H_{(i-1)}}(B_i),$$

$$H(M) = E_{H_{(n-1)}}(B_n)$$

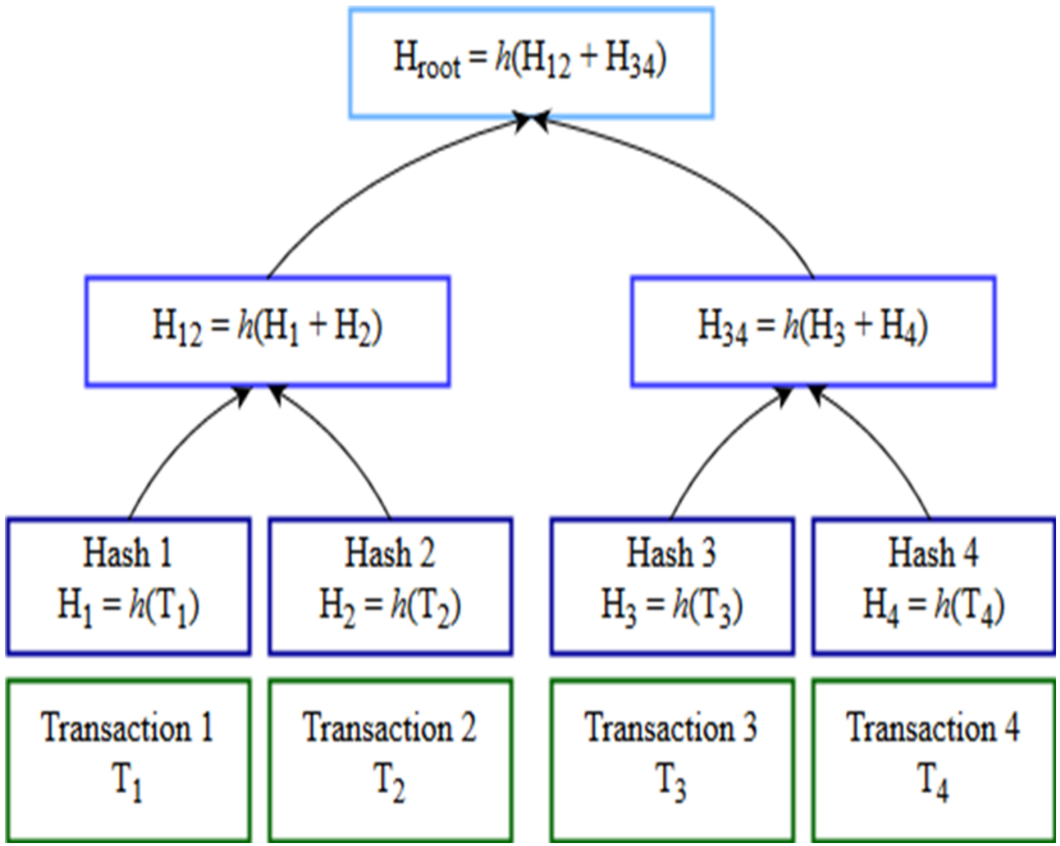
Password Protection (using HF)



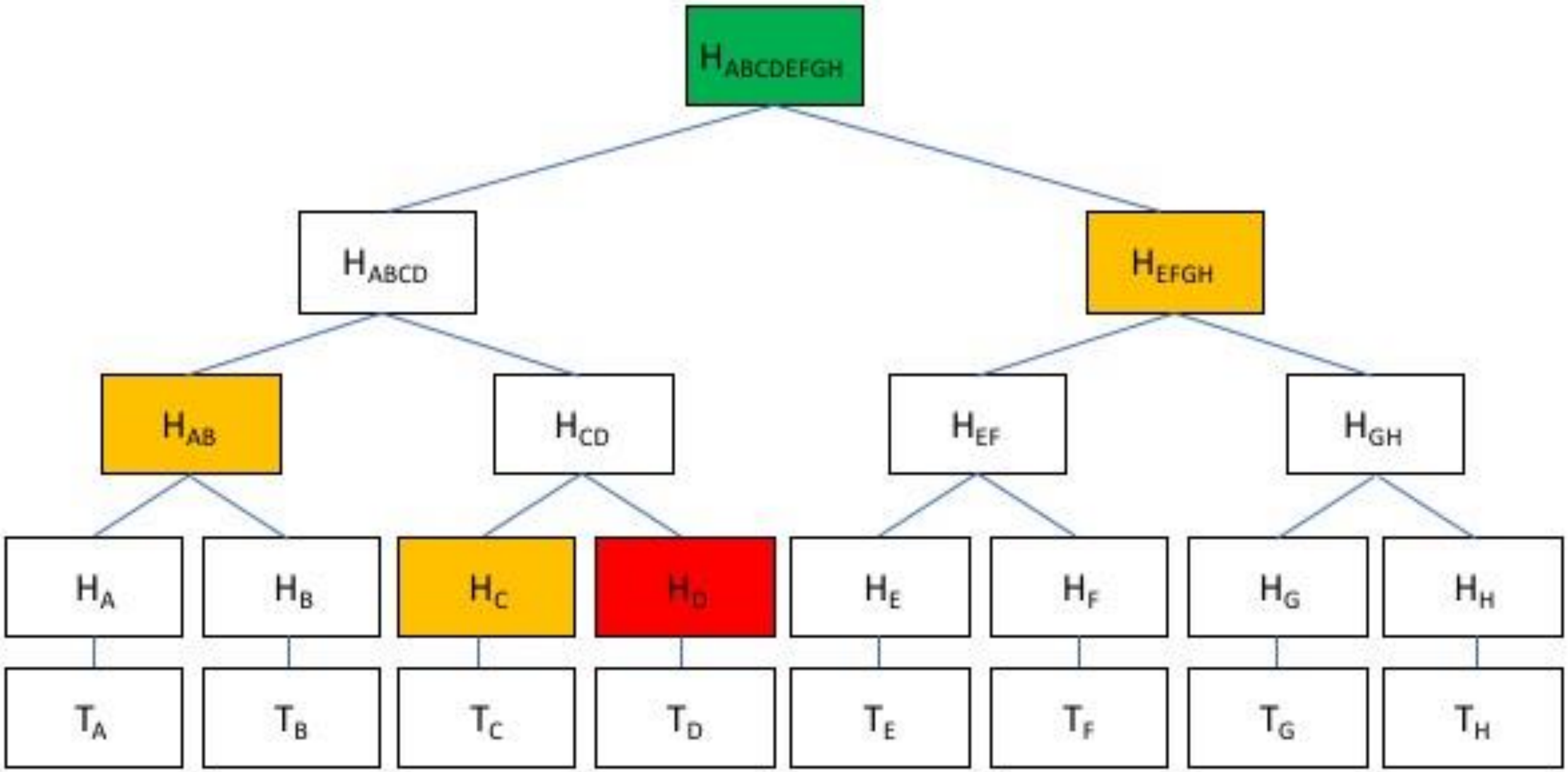
Digital Signature



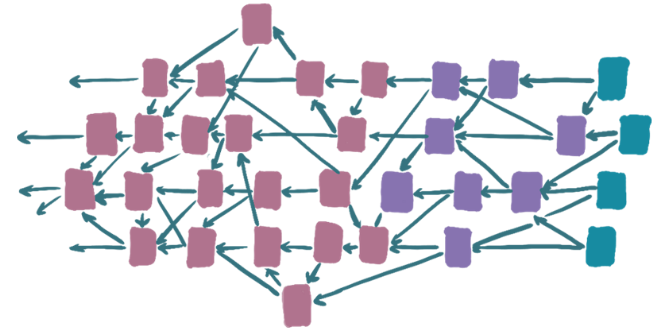
Compressing transactions by hashing



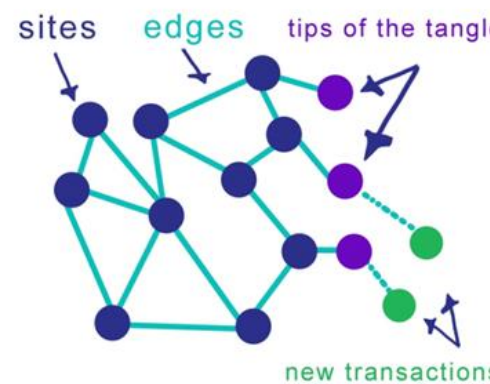
Merkle Tree (bitcoin core)



Moving forward beyond Cryptocurrencies



- Blockchain - a data structure of back-linked list of blocks of transactions, ordered with resect to time and provides tamper evident log
- Immutable records, Supply Chain Management
- Blockchain ecosystems (rather than coins)
- Day-to-day use
- Smart contracts (Ethereum , Solidity)
- Specialized use – IOTA
- dAPPs (beyond BitTorrent…)



Concluding Remarks

- Security: Looking Back
- The historical focus has been to try to build a "wall of protection" around the system or network to protect it from external threats
- this approach worked when organizations were more centralized
- Today – highly connected world!!!!!