Blockchain CheatSheet - Cryptography & Signatures

Read Time: 6 m

Table of Contents

§ Fundamentals

Cryptoanalysis

- **Definition**: The art of decryption, which is the analysis and overcoming of cryptographic systems.

Cryptography

- **Definition**: The art of encryption, which is the practice of protecting information using ciphers.

Ciphers

- **Definition**: Rules used to encrypt data.
  - **Symmetric**: Uses the same key for encryption and decryption.
  - **Asymmetric**: Uses a pair of keys, a public key to encrypt and a private key to decrypt.
- **Protocols**: Sets of rules that determine how encryption and decryption operations should be performed.
- **Properties of Valid Ciphers**:
  1. Easy to encrypt
  2. Easy to transmit
  3. Easy to decode
  4. Hard to decode if intercepted
  5. Source of data should be validated

§ Symmetric Cyphers

Monoalphabetic Symmetric Ciphers

- **Definition**: Use a single fixed substitution between plaintext and ciphertext.

Example of Cipher Alphabet (Inverse)

Alphabet	A	B	C	...	K	L	M	N	O	...	Z

| **Inverse** | **Z** | **Y** | **X** | **...** | **P** | **O** | **N** | **M** | **L** | **...** | **A** |

Example of Encryption

H	E	L	L	O
S	**V**	**O**	**O**	**L**

Polyalphabetic Symmetric Ciphers

Phrase to encrypt: "HELLO WORLD" Repeated key: "KEYKEYKEYKE"

Message	H	E	L	L	O	W	O	R	L	D
Repeated Key	K	E	Y	K	E	Y	K	E	Y	K
Message (numbers)	7	4	11	11	14	22	14	17	11	3
Key (numbers)	10	4	24	10	4	24	10	4	24	10
Sum mod 26	17	8	9	21	18	20	24	21	9	13
Encrypted	R	I	J	V	S	U	Y	V	J	N

---

§ Symmetric Digital Signatures

Symmetric key Exchange

- **Key Usage**: Uses a single key for both signing and verification.
- **Speed**: Generally faster because it uses simpler algorithms.
- **Key Management**: Key distribution can be challenging since the same key must be shared securely between parties.
- **Use Case**: Commonly used in scenarios where both parties already share a secret key, like within closed systems.

Diffie Hellman Key Exchange

Definition: Diffie-Hellman Key Exchange is a secret-sharing algorithm that returns the components needed for arithmetic operations to generate a shared secret key.

Process:

1. **Establish Public Components**:
    - **Modulus (M)**: A large prime number used as the mathematical dividend.

    - **Generator (G)**: A base number used for exponentiation.
2. **Private Keys**:
    - Each party generates their own private key (**PrK**).
3. **Arithmetic Operations**:
    - Each party performs the following operation using their private key: $\mathbf{G^{PrK} \bmod M}$
    - The remainder (R) from this operation is shared between the parties.
4. **Secret Unveiling**:
    - Each party then takes the received remainder (**R**) and performs the following operation using their private key: $\mathbf{R^{PrK} \bmod M}$
    - The final remainder (**LR**) will be the same for both parties and will serve as the common encryption and decryption key.

Security:

- No attacker can decipher the shared secret key (**LR**) by only knowing **G**, **M**, and **R** without access to the private keys (**PrK**) of the parties involved.

---

§ Asymmetric Digital Signatures

RSA (Rivest Shamir Adleman)

Key Generation:

- Generate two prime numbers A and B.
- Calculate Max = A × B.
- Calculate φ(Max) = (A-1) × (B-1).
- Choose a public exponent e.
- Calculate the private exponent d as the modular multiplicative inverse of e modulo φ(Max).

The security of RSA is based on the difficulty of factoring Max into A and B. Without the prime numbers A and B, it is very difficult to calculate the private key d if only Max is known.

Brute-force attacks to find d would require factoring Max, which is computationally difficult for sufficiently large numbers.

- **Decryption**: Uses the private key (d, Max).
- **Private Key Generation**: Requires the prime numbers A and B.
- **Factoring Attacks**: An attacker who wants to find d without knowing A and B must factor N, a problem known to be difficult.

Weaknesses:

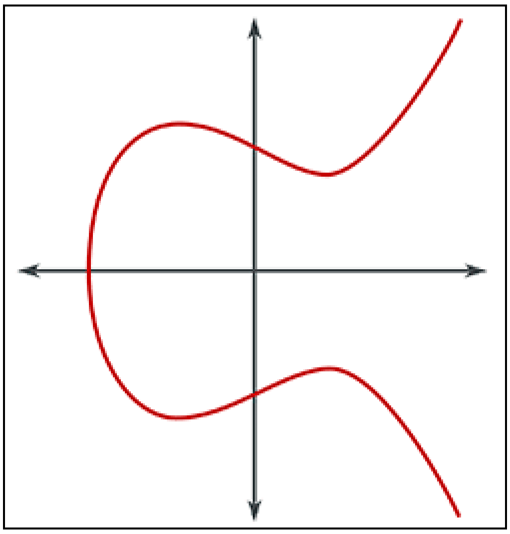
- Factoring Max is possible by dividing it by prime numbers in search of the original pair.

ECC Operations (Elliptic Curve Cryptography)

Comparisons:
To reach the level of security of a 256-bit key with ECC, you would need a 3072-bit key with RSA. In real use cases, a government top secret level of security implies a 384-bit key with ECC, which would require a 7680-bit key with RSA.

Formula:

$Y^2 = X^3 + ax + b$



Summary

1. **Intersection of Points on an Elliptic Curve**:
    - Drawing a straight line that intersects two points on the elliptic curve (A and B), this line will inevitably touch a third point (C) on the curve.
2. **Symmetry with respect to the X-axis**:
    - Reflecting the third point (C) with respect to the X-axis, a new point (D) is obtained on the curve.
3. **Repetition of the Operation (Point Addition)**:

- Repeating this point addition operation N times generates a sequence of points on the curve.

4. **Private Key**:
   - The number of point addition operations performed (N) will be our private key.

> BTC uses $Y^2 = X^3 + 0 * x + 7 = \mathbf{X^3 + 7}$

ECDSA (Elliptic Curve Digital Signature Algorithm)

Public key : we take a private key or in other words a Secret Signing key then we generate a liked public key though Elliptic Curve operations in mathematical way as coordinates (x_1 , y_1)

The signature : We use the data a Nonce(Random number) and the private key and we use the in elliptic curve operation that will return a digital signatures in coordinates (r , s) which are public

Verification of the signatures : We use the data, the coordinates as signatures and the public key we use them with elliptic curve operations which gives us two new coordinates (x_2 , y_2) then we do a modulus using x_2 as a base if we get x_1 the signature is verified.

> Notice: This elegantly proves that the person with the private key generated the data. And the signature is always different based on the entity of the data.

Suggested Follow-up

Blockchain CheatSheet - Technical Use

Author: Kenneth Boldrini