Blockchain CheatSheet - Kryptographie & Signaturen

🕓 **Lesezeit: 6 Min**

Inhaltsverzeichnis

§ Grundlagen

Kryptoanalyse

- **Definition**: Die Kunst der Entschlüsselung, also die Analyse und Überwindung von kryptographischen Systemen.

Kryptographie

- **Definition**: Die Kunst der Verschlüsselung, also die Praxis des Schutzes von Informationen durch Chiffren.

Chiffren

- **Definition**: Regeln, die verwendet werden, um Daten zu verschlüsseln.
  - **Symmetrisch**: Verwendet denselben Schlüssel für Verschlüsselung und Entschlüsselung.
  - **Asymmetrisch**: Verwendet ein Paar von Schlüsseln, einen öffentlichen Schlüssel zum Verschlüsseln und einen privaten Schlüssel zum Entschlüsseln.
- **Protokolle**: Regelwerke, die bestimmen, wie Verschlüsselungs- und Entschlüsselungsoperationen durchgeführt werden sollen.
- **Eigenschaften gültiger Chiffren**:
  1. Einfach zu verschlüsseln
  2. Einfach zu übertragen
  3. Einfach zu entschlüsseln
  4. Schwer zu entschlüsseln, wenn abgefangen
  5. Quelle der Daten sollte validiert werden

---

§ Symmetrische Chiffren

Monoalphabetische Symmetrische Chiffren

- **Definition**: Verwendet eine feste Substitution zwischen Klartext und Chiffriertext.

Beispiel für das Chiffren-Alphabet (Inverse)

Alphabet	A	B	C	...	K	L	M	N	O	...	Z
Inverse	Z	Y	X	...	P	O	N	M	L	...	A

Beispiel für Verschlüsselung

H	E	L	L	O
S	V	O	O	L

Polyalphabetische Symmetrische Chiffren

Zu verschlüsselnde Nachricht: "HELLO WORLD" Wiederholter Schlüssel: "KEYKEYKEYKE"

Nachricht	H	E	L	L	O	W	O	R	L	D
Wiederholter Schlüssel	K	E	Y	K	E	Y	K	E	Y	K
Nachricht (Zahlen)	7	4	11	11	14	22	14	17	11	3
Schlüssel (Zahlen)	10	4	24	10	4	24	10	4	24	10
Summe mod 26	17	8	9	21	18	20	24	21	9	13
Verschlüsselt	R	I	J	V	S	U	Y	V	J	N

---

§ Symmetrische Digitale Signaturen

Symmetrischer Schlüsselaustausch

- **Schlüsselverwendung**: Verwendet einen einzigen Schlüssel für sowohl Signierung als auch Verifizierung.
- **Geschwindigkeit**: Im Allgemeinen schneller, da einfachere Algorithmen verwendet werden.
- **Schlüsselmanagement**: Schlüsselverteilung kann schwierig sein, da derselbe Schlüssel sicher zwischen den Parteien geteilt werden muss.
- **Anwendungsfall**: Wird häufig in Szenarien verwendet, in denen beide Parteien bereits einen geheimen Schlüssel teilen, wie in geschlossenen Systemen.

Diffie-Hellman-Schlüsselaustausch

Definition: Der Diffie-Hellman-Schlüsselaustausch ist ein Geheimnis-Teilungsalgorithmus, der die Komponenten liefert, die für die arithmetischen Operationen zur Generierung eines gemeinsamen geheimen Schlüssels erforderlich sind.

Prozess:

1. **Öffentliche Komponenten Festlegen**:
   - **Modulus (M)**: Eine große Primzahl, die als mathematischer Dividend verwendet wird.
   - **Generator (G)**: Eine Basiszahl, die für die Exponentiation verwendet wird.
2. **Private Schlüssel**:
   - Jede Partei generiert ihren eigenen privaten Schlüssel (**PrK**).
3. **Arithmetische Operationen**:
   - Jede Partei führt die folgende Operation unter Verwendung ihres privaten Schlüssels durch: $\mathbf{G^{PrK} \bmod M}$
   - Der Rest (R) dieser Operation wird zwischen den Parteien geteilt.
4. **Geheimnis Offenlegen**:
   - Jede Partei nimmt dann den erhaltenen Rest (**R**) und führt die folgende Operation unter Verwendung ihres privaten Schlüssels durch: $\mathbf{R^{PrK} \bmod M}$
   - Der endgültige Rest (**LR**) wird für beide Parteien gleich sein und dient als gemeinsamer Verschlüsselungs- und Entschlüsselungsschlüssel.

Sicherheit:

- Kein Angreifer kann den gemeinsamen geheimen Schlüssel (**LR**) entschlüsseln, nur indem er **G**, **M** und **R** kennt, ohne Zugriff auf die privaten Schlüssel (**PrK**) der beteiligten Parteien zu haben.

---

§ Asymmetrische Digitale Signaturen

RSA (Rivest Shamir Adleman)

Schlüsselgenerierung:

- Zwei Primzahlen A und B generieren.
- Max = A × B berechnen.
- φ(Max) = (A-1) × (B-1) berechnen.
- Einen öffentlichen Exponenten e wählen.
- Den privaten Exponenten d als den modularen multiplikativen Inversen von e modulo φ(Max) berechnen.

Die Sicherheit von RSA basiert auf der Schwierigkeit, Max in A und B zu faktorisieren. Ohne die Primzahlen A und B ist es sehr schwierig, den privaten Schlüssel d zu berechnen, wenn nur Max bekannt ist.

Brute-Force-Angriffe zur Findung von d erfordern die Faktorisierung von Max, was bei ausreichend großen Zahlen rechnerisch schwierig ist.

- **Entschlüsselung**: Verwendet den privaten Schlüssel (d, Max).
- **Privatschlüsselgenerierung**: Erfordert die Primzahlen A und B.
- **Faktorisierungsangriffe**: Ein Angreifer, der d finden möchte, ohne A und B zu kennen, muss N faktorisieren, ein als schwierig bekanntes Problem.

Schwächen:

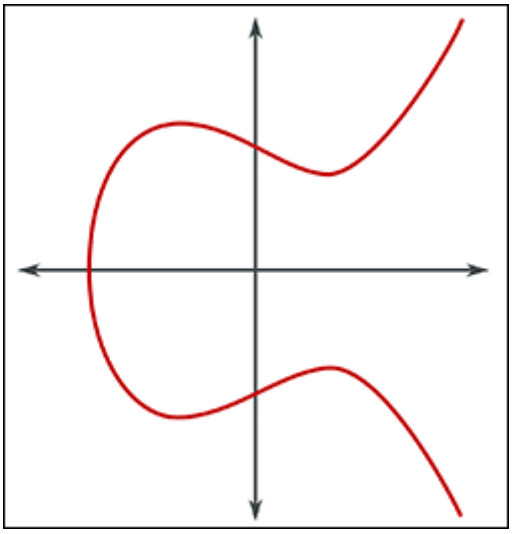
- Die Faktorisierung von Max ist möglich, indem man Max durch Primzahlen teilt, um das ursprüngliche Paar zu finden.

ECC-Operationen (Elliptische Kurven-Kryptographie)

Vergleiche: Um das Sicherheitsniveau eines 256-Bit-Schlüssels mit ECC zu erreichen, benötigen Sie einen 3072-Bit-Schlüssel mit RSA. In der Praxis bedeutet ein Regierungsgeheimnis-Level von Sicherheit eine 384-Bit-Schlüssel mit ECC, was einen 7680-Bit-Schlüssel mit RSA erfordert.

Formel:

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



Wenn Sie eine gerade Linie von einem Punkt auf der Kurve (A) im positiven Y-Plan ziehen, um einen anderen Punkt auf derselben Kurve (B) zu treffen, wird die Linie zwangsläufig einen dritten Punkt (C) berühren.

Durch die Nutzung der Symmetrie der elliptischen Kurve in Bezug auf die X-Achse, indem der dritte Punkt (C) in den negativen Y-Plan projiziert und mit dem ursprünglichen Punkt (A) verbunden wird, erhalten Sie einen vierten Punkt (D).

Durch wiederholtes Ausführen dieser letzten Operation (skalar Multiplikation) N Mal, wird die Anzahl der durchgeführten skalar Multiplikationsoperationen unser privater Schlüssel sein!

Zusammenfassung:

1. **Schnittpunkt von Punkten auf einer elliptischen Kurve**:
   - Eine gerade Linie, die zwei Punkte auf der elliptischen Kurve (A und B) schneidet, wird zwangsläufig einen dritten Punkt (C) auf der Kurve berühren.
2. **Symmetrie in Bezug auf die X-Achse**:
   - Durch Reflektion des dritten Punktes (C) bezüglich der X-Achse wird ein neuer Punkt (D) auf der Kurve erhalten.
3. **Wiederholung der Operation (Punktaddition)**:
   - Das wiederholte Hinzufügen von Punkten erzeugt eine Sequenz von Punkten auf der Kurve.
4. **Privater Schlüssel**:
   - Die Anzahl der durchgeführten Punktaddition-Operationen (N) wird unser privater Schlüssel sein.

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

ECDSA (Elliptischer Kurven Digitale Signatur Algorithmus)

Öffentlicher Schlüssel: Wir nehmen einen privaten Schlüssel oder, anders gesagt, einen geheimen Signaturschlüssel und generieren dann einen entsprechenden öffentlichen Schlüssel durch elliptische Kurvenoperationen als Koordinaten (x_1, y_1).

Die Signatur: Wir verwenden die Daten, eine Nonce (zufällige Zahl) und den privaten Schlüssel und wenden diese auf elliptische Kurvenoperationen an, die eine digitale Signatur in den Koordinaten (r, s) zurückgeben, die öffentlich sind.

Verifikation der Signaturen: Wir verwenden die Daten, die Koordinaten als Signaturen und den öffentlichen Schlüssel, und führen elliptische Kurvenoperationen durch, um zwei neue Koordinaten (x_2, y_2) zu erhalten. Dann verwenden wir x_2 als Basis für ein Modul. Wenn wir x_1 erhalten, ist die Signatur verifiziert.

> Hinweis: Dies beweist elegant, dass die Person mit dem privaten Schlüssel die Daten generiert hat. Und die Signatur ist immer unterschiedlich, basierend auf den Daten.

Vorgeschlagene Folge

Blockchain CheatSheet - Technische Nutzung

Autor: Kenneth Boldrini