CLOAK

ENIGMA V2.1

Whitepaper (Revised)

February 2018

ENIGMA

DAS PRIVATE, SICHERE UND NICHT ZURÜCKVERFOLGBARE TRANSAKTIONSSYSTEM VON CLOAKCOIN

1. ZUSAMMENFASSUNG

Die Kryptowährung Cloak ist darauf ausgelegt, private, sichere und dezentralisierte Überweisungen mit Enigma durchzuführen. Cloak ist ein dualer PoW/PoS-Coin (Proof of Work/Proof of Stake), der zurzeit zinstragend ist (PoS). Ein wichtiger Bestandteil von Cloak ist Enigma, ein datenschutzgesteuertes und nicht zurückverfolgbares Transaktionssystem, welches die dezentralisierten Anwendungen des CloakCoin-Netzwerks möglich macht und die Grundlage für zukünftige Weiterentwicklungen darstellt – denn Datenschutz ist heutzutage wichtiger als je zuvor. Der rasante technologische Fortschritt der vergangenen Jahre hat unsere Horizonte drastisch erweitert und dafür gesorgt, dass die Welt mehr verbunden ist als jemals zuvor. Dank der Einführung des Bitcoin im Jahr 2009 wurden Kryptowährungen immer populärer und mithilfe der Blockchain-Technologie können digitale Transaktionen mittlerweile sicher und unmittelbar auf der ganzen Welt durchgeführt werden. Aufgrund der stetig steigenden Verwendung von Kryptowährungen, sind verschärfte Vorschriften unabdingbar. Es bleibt abzuwarten inwieweit diese Vorschriften den alltäglichen Umgang mit Kryptowährungen tatsächlich beeinflussen werden, jedoch befürchten viele, dass die individuelle Handlungsfreiheit längerfristig eingeschränkt werden könnte.

ENIGMA

Enigma bietet im Grunde genommen einen dezentralisierten, off-blockchain Transaktionsservice, der es allen Nutzern des CloakCoin-Netzwerks erlaubt, ihre Cloaks privat und sicher zu transferieren. Das Konzept von Enigma basiert auf einem Mischprozess, der so ausgelegt ist, dass Transaktionen nicht von Dritten verfolgt werden können und somit geschützt sind. Es wird sichergestellt, dass Cloaks, die sich im Umlauf befinden, geschützt werden und weder Sender noch Empfänger identifiziert werden können. Des Weiteren werden CloakCoins auch niemals über eine zwischengeschaltete Station transferiert. Wir haben intensiv daran gearbeitet, dass das Enigmasystem diejenigen Nutzer belohnt, die bei Cloak-Transaktionen unterstützend mitwirken. Außerdem versuchen wir den Prozess kontinuierlich weiter zu verbessern und noch mehr Anreize für aktive Transaktionsteilnehmer zu schaffen. Jeder Besitzer von CloakCoins kann an Cloaking-Aktivitäten teilnehmen und sein Wallet im Staking-/Cloaking-Modus lassen. Dadurch wird ermöglicht, Cloak-Transaktionen auf passive Art und Weise zu unterstützen und erhebliche Belohnungen zu verdienen.

2. ENIGMA V1.0 – ÜBERSICHT

Enigma ist die erste öffentliche Iteration des privaten, sicheren und nicht zurückverfolgbaren Zahlungssystems von Cloak. Enigma-Transaktionen werden von anderen Usern „gecloaked“, wofür sie anschließend eine Belohnung erhalten. Außerdem ist es unmöglich die tatsächlichen Sender und Empfänger der Cloaks zu identifizieren, da die User Inputs und Outputs für die Enigma-Transaktionen liefern. Mithilfe von CloakShield kann der Empfänger sicherstellen, dass seine Daten beim Cloak-Transfer geschützt sind, da alle Nachrichten und Informationen im Netzwerk zerlegt und verschlüsselt werden. Für weitere Informationen zum Thema „CloakShield“ siehe Abschnitt 3. dieses Artikels.

2.1. DER ENIGMA-PROZESS (FÜR AKTIVE NODES)

ANKÜNDIGUNGEN VON ENIGMA

Enigma-Nodes kommunizieren innerhalb des Cloak-Netzwerks und behalten die Aktivitäten anderer Nodes im Auge. Der „Enigma Announcement Broadcast“ informiert andere Enigma-Nodes über unseren öffentlichen Session-Key sowie das aktuelle Cloak Guthaben.

CLOAKING-ANFRAGEN VON ENIGMA

Wenn ein User eine Cloak-Transaktion durchführen möchte, so werden eine Reihe von Enigma-Nodes (mit ausreichendem Enigma Guthaben) ausgewählt, die beim Cloaken unterstützen können. Ein Enigma-Node hat dann die Option, der Anfrage zuzustimmen oder sie abzulehnen. Erhält der Anforderer eine Ablehnung bzw. überhaupt keine Antwort, so wird innerhalb kurzer Zeit ein alternativer Node kontaktiert.

Der DDoS-Schutz (distributed denial of service) setzt alle negativ auffallenden Nodes für den Rest einer Session auf die schwarze Liste. Dazu kommt es, wenn sich ein Node mehrmalig weigert eine Enigma-Transaktion zu unterzeichnen bzw. Enigma-Nachrichten weiterzuleiten. Die Cloaking-Nodes von Enigma nutzen einen Eliptic Curve Diffie Hellman-Schlüsselaustausch (ECDH) mit dem Ziel einer Wissensteilung mit dem initiierenden Enigma-Node („shared secret"). Folglich wird ein geheimer gemeinsamer Schlüssel für symmetrische RSA-256 Datenverschlüsselungen zwischen dem „cloakenden" Node und dem Sender-Node generiert.

ENIGMA CLOAKING-ZUSTIMMUNG

Wenn ein Enigma-Node eine Cloaking-Anfrage akzeptiert, so liefert dieser Node eine Liste mit Inputs und Outputs, die für die Enigma-Transaktion genutzt werden können. Die Inputmenge, die von einem Node geliefert wird, muss größer oder gleich dem Wert sein, der tatsächlich von Enigma gesendet wird (jegliche Gebühren werden zusätzlich gezählt). Outputs werden sorgfältig ausgewählt, sodass sie mit den tatsächlichen Outputs der Enigma-Transaktion so weit wie möglich übereinstimmen. Wenn die Output-Adresse von Enigma vorher noch nicht benutzt wurde, so wird durch den „Cloaker" eine neue, geänderte Adresse generiert. Wenn die Output-Adresse von Enigma vorher bereits Coins erhalten hat, so wird eine bestehende Adresse mit einer ähnlichen Aktivität von dem „Cloaker" gewählt, um die eingesetzten Coins zurückzuzahlen und die Enigma „Cloaking"-Belohnung zu erhalten.

DIE „CLOAKED" ENIGMA-TRANSAKTION

Sender können eine "verhüllte" Transaktion durchführen, indem sie die von den Cloaker-Nodes bereitgestellten Inputs und Outputs verwenden. Anschließend fügen die Sender ihre eigenen In- und Ouputs zu der Transaktion hinzu und leiten einen Mischprozess ein, der das „Cloaking" ermöglicht. Im nächsten Schritt wird die verhüllte Transaktion (mithilfe von CloakShield) verschlüsselt und an alle teilnehmenden Cloaker gesendet.

Die Cloaker prüfen dann, ob die von ihnen beigesteuerten In-und Outputs in der aktuellen Transaktion vorhanden sind und ob mindestens einer ihrer Outputs mit ausreichenden Gebühren belohnt wurde. Verlaufen die Überprüfungen positiv, so wird die Transaktion unterschrieben (SIGHASH_ALL+SIGHASH_ANYONECANPAY), verschlüsselt und an den Enigma-Sender zurückgeleitet. Sobald alle Cloaker unterschrieben haben, wird die Transaktion noch seitens des Senders unterschrieben und für gültig erklärt. Nun kann die „verhüllte" Transaktion in das Netzwerk geleitet werden.

2.2.1. TRACKING VON ENIGMA CLOAKING-NODES

Aktive Enigma-Nodes versenden Nachrichten an andere Nodes im Cloak-Netzwerk. Diese Nachrichten enthalten die öffentliche ec-Schlüssel-ID des Nodes sowie das momentan verfügbare Guthaben der Enigma Cloaking-Vorgänge. Nodes pflegen eine Liste mit anderen aktiven Nodes im Netzwerk, sodass sie für Cloaking-Zwecke miteinander kommunizieren können. Die IDs der Nodes werden Session für Session generiert; das Neustarten des Clients führt zu einer Aktualisierung der aktuellen ID.

1. Jedes Wallet erzeugt sowohl einen öffentlichen als auch einen geheimen Schlüssel (secp256k1) beim Beginn einer neuen Session.

2. Das Wallet teilt den anderen Nodes innerhalb des Netzwerks regelmäßig den öffentlichen Schlüssel sowie das Cloaking-Guthaben der Session mit.

3. Die Nodes verfolgen andere aktive Enigma-Cloaking-Nodes und haben dabei die Möglichkeit, mit ihnen direkt oder indirekt (via Cloakshield Onion Routing) zu kommunizieren.

2.2.2. EINE ENIGMA-TRANSAKTION STARTEN

Alice möchte 10 Cloaks an Bob mithilfe von 5 Mixer-Nodes transferieren.

1. Alice sendet eine Enigma-Anfrage an das Netzwerk, welche ihren öffentlichen Enigma-Session-Schlüssen sowie die gewünschte Menge an Cloaks, die sie transferieren möchte, beinhaltet. Ihre Anfrage wird dann an eine Reihe von Enigma-Nodes geleitet, damit sie als Antragstellerin nicht erkannt werden kann.

2. Catherine aktiviert den „Cloaking Mode" und kann somit einen geschützten ClaokShield-Verschlüsselungskanal generieren, der eine sichere Kommunikation mit Alice ermöglicht. Anschließend erstellt Catherine ein Enigma Antwortpaket und sendet es sicher an Alice. Dieses Antwortpaket enthält eine Liste von Catherine´s In- und Outputs, die Alice verwenden wird, um ihre Transaktion zu verschleiern.

3. Alice verschlüsselt und verarbeitet Catherine´s Antwort und erzeugt eine Enigma-Transaktion basierend auf ihren eigenen In-und Outputs, die mit Catherine´s In-und Outputs vermischt werden. Dieser Vorgang wird verschlüsselt und dann an Catherine zur Unterzeichnung gesendet.

4. Catherine entschlüsselt die Enigma-Transaktion und führt eine Reihe von Integritäts-Checks durch. Dabei überprüft sie, ob ihre eigenen In-und Outputs entsprechend verwendet wurden und sie eine ausreichende Belohnung erhalten hat. Verlaufen diese Tests erfolgreich, so übermittelt Catherine die von ihr unterzeichnete und verschlüsselte Transaktion an Alice.

5. Alice führt weitere Überprüfungen der Transaktion durch und unterzeichnet dann ebenfalls. Anschließend wird die Transaktion sicher an das Netzwerk bzw. andere Enigma-Nodes geleitet.

6. Wenn die Transaktion abgeschlossen ist, erhält Bob die Cloaks von Alice. Catherine hingegen erhält eine „Cloaking"-Belohnung für ihre Unterstützung bei der Verschleierung der Enigma-Transaktion.

7. Dank der In-und Outputs von Catherine, die ein Spiegelbild der In-und Outputs von Alice darstellen, kann weder der Sender noch der Empfänger der Enigma-Transaktion identifiziert werden.

BEISPIEL

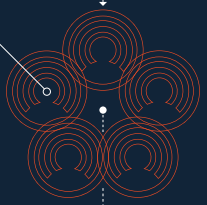
Kathrin möchte CloakCoins an Bob transferieren, und zwar vollkommen anonym.

KATHRIN

−10.0992 CLOAK

(−10 CLOAK) Empfängersumme +
(−0.0992 CLOAK) ENIGMA-Gebühr

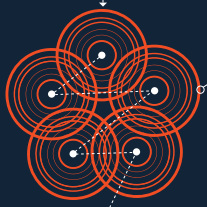
Die Kommunikation zwischen den ENIGMA Mixer-Nodes beginnt.



Jeder CloakCoin-Besitzer hat die Möglichkeit, als Mixer-Node bzw. "Cloaker" zu fungieren.

Jeder Teilnehmer kommuniziert durch einen verschlüsselten Kanal anonym.

Kathrins Wallet ist nun mit den Mixer-Nodes verbunden.



Jeder Mixer-Node hilft Kathrin dabei die Transaktion zu verschleiern.

Durch dieses Netzwerk von Nodes entsteht eine dezentrale Anonymisierung, ähnlich wie beim TOR-Onion-Routing System.

Mixer-Nodes werden für das Cloaken von Kathrins Transaktion belohnt.

+0.0992 CLOAK

Die anfallende ENIGMA-Gebühr des Transaktionswertes wird auf alle teilnehmenden Cloaker verteilt.

Dieses System arbeitet nahtlos und stellt sicher, dass vollkommene Anonymität sowie Datenschutz gewährleistet sind.

Bob erhält abschließend Kathrin verschlüsselte, anonyme Zahlung.

BOB

+10 CLOAK

3. CLOAKSHIELD

Cloakshield ermöglicht eine sichere Kommunikation zwischen den Nodes im Cloak-Netzwerk, indem es auf eine symmetrische RSA-Verschlüsselung zurückgreift, die durch einen Diffie-Hellmann Schlüsselaustausch (ECDH) unterstützt wird. Infolgedessen können Nodes ihre Daten sicher austauschen ohne dabei von Mittelmännern oder Betrügern gefährdet zu sein. Cloakshield wurde so entwickelt, dass sowohl Enigma als auch dezentralisierte CloakCoin-Anwendungen geschützt werden. Zusätzlich wird gewährleistet, dass alle Daten so privat und persönlich wie möglich bleiben. Mit Cloakshield können Daten verschlüsselt zu einem oder mehreren Empfängern gesendet werden. Gibt es nur einen Empfänger, so wird die Payload mithilfe des ECDH und des daraus resultierenden gemeinsamen geheimen Schlüssels RSA-verschlüsselt („shared secret“). Wenn mehrere Empfänger involviert sind, so wird die Payload mithilfe eines Einmalschlüssels verschlüsselt. Dabei wird der Einmalschlüssel für jeden einzelnen Empfänger mithilfe der ECDH/RSA-Methode verschlüsselt.

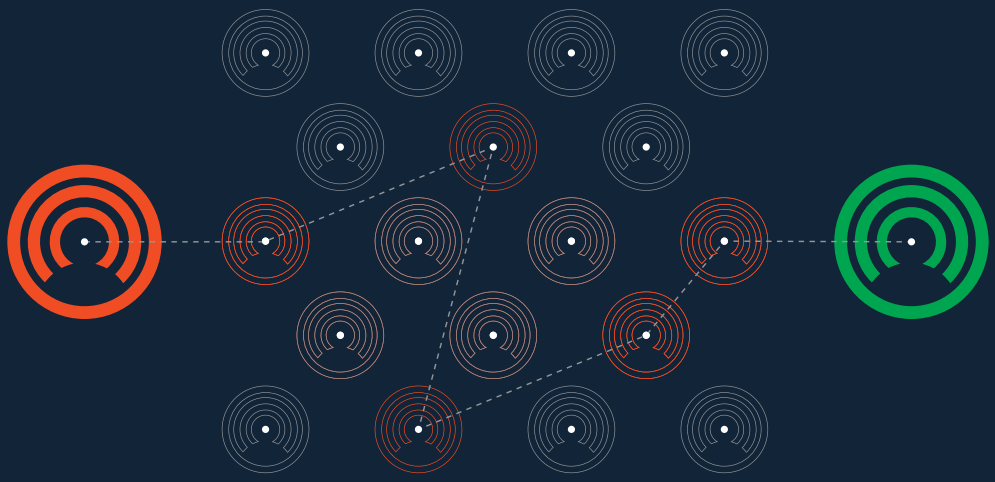
ERZEUGUNG EINER GEMEINSAMEN VERSCHLÜSSELUNG

Damit Alice und Bob sicher kommunizieren können, müssen sie einer gemeinsamen Verschlüsselung zustimmen. Dies wird mithilfe von Cloakshield sowie des ECDH möglich, wie folgendes Beispiel zeigen soll:

Alice besitzt einen privaten Enigma-Schlüssel „dA“ und einen öffentlichen Enigma-Schlüssel „QA=dAG“. Bob hingegen hat einen privaten Enigma-Schlüssel „dB“ und einen öffentlichen Enigma-Schlüssel „QB=dBG“. Aufgrund der Mitteilungen, die Bob an das Enigma-Netzwerk sendet, kommuniziert er seine Bereitschaft an Verschleierungsprozessen („Cloaking“) teilzunehmen. Außerdem wird sein öffentlicher Enigma-Schlüssel dadurch für Alice zugänglich. Alice nutzt ihren eigenen privaten Schlüssel zusammen mit dem öffentlichen Schlüssel von Bob, um ein geteiltes Geheimnis „dAQB=dAdBG“ zu kalkulieren (ECDH berechnet den Schlüssel in OpenSSL). Daraufhin generiert Alice einen SHA256-Hash und übermittelt den Hash an die OpenSSLEVP_BytesToKey-Methode, um eine Verschlüsselung abzuleiten mit der die Daten von Bob verschlüsselt werden (symmetrische RSA-Verschlüsselung). Alice kann nun Nachrichten an Bob senden, die durch Cloakshield geschützt sind. Sobald Bob eine Nachricht von Alice erhält, die durch Cloakshield geschützt ist, sieht er ihren öffentlichen Schlüssel in der Kopfzeile. Nun erzeugt er – auf die gleiche Weise wie Alice dies getan hat - das gleiche geteilte Geheimnis wie Alice (mit seinem eigenen privaten Schlüssel). Das Cloak-Wallet beinhaltet eine Liste von aktiven Cloakshield-Schlüsseln und überprüft diese Liste jedes Mal nach bereits vorhandenen Schlüsseln, bevor ein neuer Schlüssel generiert wird.

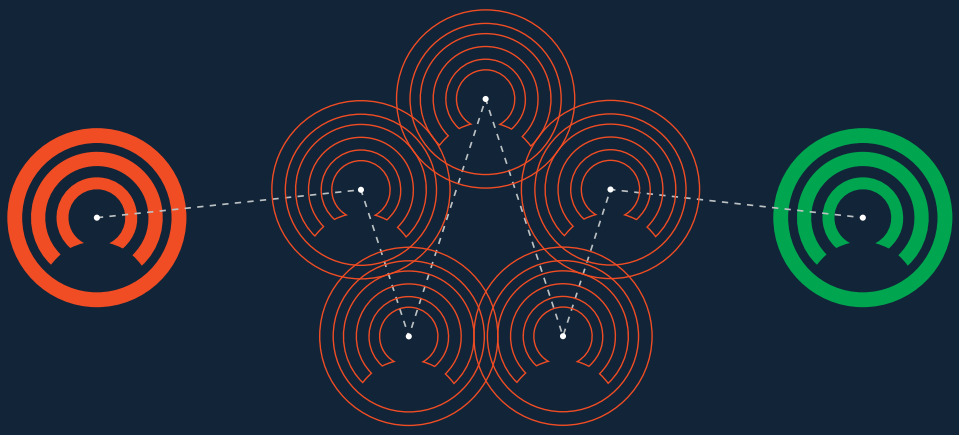
CLOAKSHIELD-DATEN

Claokshield ermöglicht es jegliche Datenobjekte zu ordnen und anschließend sicher an einen oder mehrere Empfänger zu senden. Ein Daten-Packet-Header von Cloakshield beinhaltet den öffentlichen Enigma-Schlüssel des Senders sowie die Hashes der öffentlichen Schlüssel der Empfänger. Darüber hinaus haben Cloakshield-Header einen Verifizierungs-Hash, der generiert wird, wenn der öffentliche Schlüssel des Senders zusammen mit den rohen und unverschlüsselten Daten benutzt wird. Dieser Hash wird während der Entschlüsselung von Cloakshield-Daten verifiziert, um zu gewährleisten, dass die Empfängerinformationen im Header mit der Verschlüsselung übereinstimmen. Außerdem kann so sichergestellt werden, dass die Daten nicht geändert wurden.



CLOAKSHIELD ONION ROUTING

Onion Routing ist eine Technik, die bei TOR genutzt wird und anonyme Kommunikation innerhalb eines Computernetzwerks ermöglicht. In einem Onion-Netzwerk sind Nachrichten von Verschlüsselungsschichten umgeben. Die verschlüsselten Daten werden von einer Reihe von Netzwerk-Nodes („Onion Routers“) übermittelt, wobei jeder Onion Router eine der Verschlüsselungsschichten „schält“. Dadurch wird der nächste Zielort der Daten sichtbar. Wenn die letzte Schicht entschlüsselt ist erreicht die Nachricht ihren endgültigen Zielort. Dabei bleibt der Sender anonym, da jede Zwischenperson lediglich die Standorte der unmittelbar vorangeschalteten bzw. nachfolgenden Nodes kennt.



ONION ROUTING ANALOGIE

Durch die Einführung der Onion-Routing-Funktion in das Enigma-Netzwerk (welches Cloakshield benutzt) können Nodes indirekt miteinander kommunizieren und Datenverkehrsanalysen umgehen. Folglich wird die Identifizierung der Nodes, die miteinander kommunizieren bzw. Transaktionen an das CloakCoin-Netzwerk senden, erschwert. Wenn ein Node mit einem anderen Node kommunizieren möchte, so wählt dieser Node eine Reihe von weiteren Nodes aus, die als Verbindungsstellen für die Kommunikation fungieren. Jede verschlüsselte Schicht kann lediglich von der dafür vorgesehenen Verbindungsstelle (also dem dafür vorgesehenen Node) entschlüsselt werden. Nachdem eine Schicht entschlüsselt wurde, werden die Daten an den nächsten Node weitergegeben. Dieser Ablauf setzt sich so lange fort, bis alle Schichten von den entsprechenden Nodes entschlüsselt wurden und die Daten den angestrebten Empfänger erreichen. Aufgrund der autonomen Natur des Enigma-Netzwerks, sind keine Exit-Nodes erforderlich. Außerdem wird durch CloakShield garantiert, dass nur der finale Empfänger auf die gesendeten Daten zugreifen kann.

4. STEALTH ADRESSEN

Cloak verwendet das Enigma-System, um Transaktionen einfach und sicher zu machen.

CLOAKSHIELD – KOMMUNIKATION ZWISCHEN VERSCHIEDENEN NODES

Zu Beginn generiert jedes Wallet ein [NID_secp256k1] Schlüsselpaar (Cloaking Encryption Key / CEK), um so Ad-hoc-Geheimnisse mithilfe eines ECDH ableiten zu können. Dabei wird der private Schlüssel des Senders sowie der öffentliche Schlüssel des Empfängers benutzt. Dieser Prozess stellt die Grundlage für jede Node-zu-Node-Kommunikation im Enigma-Netzwerk dar. Unter 'src/enigma/cloakshield.h/.cpp' finden Sie weitere Informationen zu diesem Thema. Die ECDH-basierte, verschlüsselte Kommunikation findet ebenso beim Onion-Routing-System von CloakShield Anwendung.

Wenn Onion-Routing eingeschaltet ist, versucht der Client einen gültigen Weg („Onion Route") für die Daten herzustellen, indem er auf eine Liste von bekannten Enigma-Peers zurückgreift. Der Node muss dabei keine direkte Verbindung zu den Enigma-Peers haben, da CloakData-Pakete (für das Routing von CloakShield gepackte Daten) von Peer zu Peer weitergegeben werden. Eine Onion-Route besteht in der Regel aus drei unterschiedlichen Routen, die alle zum Ziel-Node führen (mit 3 Node-Hops pro Route). Dadurch kann der Ausfall eines Routing-Nodes jederzeit kompensiert werden.

Nodes senden regelmäßig Enigma-Mitteilungen (src/enigma/ enigmaann.h) an Peers und machen damit deutlich, dass sie am Onion-Routing teilnehmen können. Andere Nodes im Netzwerk speichern diese Mitteilungen (bis sie ablaufen bzw. Updates vorliegen), um Onion Routen herzustellen.

STEALTH ADRESSEN: EIN TRANSAKTIONSBEISPIEL

Wenn ein Node eine Enigma-Transaktion an eine Stealth-Adresse sendet, passiert folgendes:

1. Der Sender erzeugt Inputs, die den gesendeten Betrag, die Enigma-Belohnung sowie die Netzwerkgebühr (1% bei 0 Coins und 0,2% bei >1000 Coins) abdeckt.
2. Der Sender generiert ein Objekt für eine Claoking-Anfrage, welches eine einzigartige Stealth für diese Anfrage enthält.
3. Der Sender generiert 2 bis 4 einmalige Stealth-Zahlungsadressen. Dafür benutzt er die Stealth-Adresse des Empfängers und teilt den gesendeten Betrag zufällig auf die Zahlungsadressen auf.
4. Der Sender bestimmt die Anzahl der Teilnehmer (er kann zwischen 5 und 25 Teilnehmer wählen). Jeder Teilnehmer erhält 80-120% einer gleichermaßen aufgeteilten Enigma-Gebühr.
5. Der Sender nutzt Onion-Routing, um die Cloaking-Anfrage an das Netzwerk zu leiten. Die Anfrage beinhaltet den gesendeten Betrag, sodass alle "Cloaker" wissen, wie viele Cloaks sie aufheben müssen.
6. Der Cloaker entscheidet sich dazu am „Cloaking" teilzunehmen und nimmt die Anfrage an.
7. Der Cloaker liefert X Inputs an den Sender sowie eine Stealth-Adresse und einen Stealth-Hash (für den Austausch).
8. Der Cloaker sendet eine „Cloaking-Accept-Response" an den Sender. Diese enthält die Stealth-Adresse, Stealth-Nonce und die Transaktionsinputs.
9. Der Sender wartet bis genügend Cloaker die Anfrage akzeptiert haben.
10. Der Sender erstellt eine Enigma-Transaktion, indem er seine eigenen Inputs sowie die Inputs der Cloaker benutzt. Alle Inputs werden hierbei gemischt.
11. Der Sender erstellt die Transaktionsoutputs für alle Cloaker. Die Outputs werden zufällig aufgeteilt und an die Cloaker zurückgesendet. In diesem Zuge erhalten die Cloaker auch ihre Belohnung.

12. Der Sender erstellt seine eigene Ausgleichszahlung für die Enigma-Transaktion. Für die Zahlung werden einmalige Stealth-Adressen verwendet.

13. Der Sender berechnet die Gebühren der Netzwerk-Transaktion und subtrahiert diese von seiner eigenen Ausgleichszahlung.

14. Der Sender leitet die Enigma-Transaktion an alle teilnehmenden Cloaker zur Unterschrift weiter.

15. Die Cloaker überprüfen, ob ihre Inputs in der Transaktion enthalten und korrekt sind. Außerdem überprüfen sie, ob einmalige Zahlungsadressen mit einer ihrer Stealth-Adressen verknüpft sind. Dabei ist wichtig, dass die Zahlung den Input übersteigt.

16. Der Cloaker stimmt der Transaktion zu bzw. lehnt diese ab und sendet die entsprechende Signatur an den Sender.

17. Der Sender sammelt alle Signaturen und übermittelt die finale, signierte Transaktion an das Netzwerk.

18. Die Nodes scannen alle eingehenden Transaktionen hinsichtlich der Stealth-Zahlungen sowie Enigma-Zahlungen und decken alle Zahlungen oder Änderungen auf. Die Schlüsselpaare und Adressen werden für alle übereinstimmenden Zahlungen generiert und anschließend im lokalen Wallet gespeichert.

5. DIE ZUKUNFT VON ENIGMA – WEITERE ENTWICKLUNGEN

Enigma bildet das Herzstück von CloakCoin und wird kontinuierlich verbessert und weiterentwickelt. Im Folgenden werden einige der geplanten Features vorgestellt:

VERBESSERTER PROOF-OF-STAKE ALGORITHMUS

Die Methode „Proof of Stake“ (PoS) zielt darauf ab, dass Nutzer innerhalb eines Krypto-Netzwerks nachweisen, dass sie Coins besitzen, um neue Blocks signieren zu können. Auf lange Sicht ist die Wahrscheinlichkeit Blocks zu signieren proportional zu der Menge an Coins, die man besitzt. Wenn jemand beispielsweise 1% des gesamten Coin-Bestands besitzt, so wird er (sie) 1% aller PoS-Blocks signieren können. Im Vergleich zu „Proof-of-Work“ (PoW), so ist für die PoS-Methode wesentlich weniger Rechenleistung sowie Energieaufwand erforderlich.

COIN-AGE UND LINEARER PROOF-OF-STAKE

Grundvoraussetzung für die meisten Implementierungen von PoS, inklusive ClokCoin, ist das Coin-Age Konzept. In erster Linie handelt es sich bei Coin-Age um eine Maßeinheit, die aufzeigt, wie lange ein Coin-Inhaber im Besitz sein Coins ist ohne diese auszugeben bzw. zu bewegen. Coin-Age steigt bei allen Coins ab dem Zeitpunkt des Abschlusses einer Transaktion (beginnend bei null). Die einfachste Form des Coin-Age ist das lineare Coin-Age. Dabei kumulieren Coins kontinuierlich Minuten, Stunden, Tage oder Jahre von Coin-Age. Beispiel: Wenn eine Person 100 Coins für 365 Tage hält, so häufen die Coins 36.500 „Coin-Tage“ bzw. 100 „Coin-Jahre“ an (ein Coin-Jahr ist auf ein Schaltjahr ausgelegt und beträgt deshalb in etwa 365,24 Tage).

Lineare PoS-Modelle wurden in der Vergangenheit aufgrund des Coin-Age Konzepts kritisiert. Den Kritikern zufolge fördert lineares PoS die Hortung von Coins und beeinträchtigt somit das Handels- und Transfervolumen. Ein weiterer Kritikpunkt bezieht sich auf die Auswirkungen von linearem PoS auf die Netzwerksicherheit. Die Sicherheit des Netzwerks leidet nämlich darunter, wenn sich ein Nutzer regelmäßig mit dem Netzwerk verbindet, um seine Coins zu staken und – sobald das gesamte Coin-Age verloren gegangen ist - die Verbindung umgehend wieder trennt.

Diesen Verbinden-Staken-Trennen-Prozess wiederholt der Nutzer immer wenn sich das Coin-Age wieder aufgefüllt hat. Ein solches Verhalten beeinträchtigt nicht nur die Netzwerksicherheit, sondern resultiert auch in der Annahme, dass ein PoS-Algorithmus, der häufiges und konstantes Staken belohnt, den größten Vorteil für CloakCoin und ähnliche Kryptowährungen mitbringen würde.

Um sicherzustellen, dass Cloaker so ausgiebig wie möglich belohnt werden, sollte Coin-Age aus dem PoS-Algorithmus entfernt werden. Folglich könnten Cloaker sowohl für das Staking als auch für das Cloaking in vollem Umfang belohnt werden. Des Weiteren würde eine neue und für das Berechnen von Staking-Belohnungen entwickelte Geschwindigkeitskomponente dazu führen, dass aktive Cloaking-Nodes im Enigma-Netzwerk zusätzlich belohnt werden. Dies würde Nutzer dazu animieren, verstärkt an Cloaking-Prozessen teilzunehmen, um sowohl ihre erhaltenen Cloaking-Belohnungen als auch ihre Zinserträge zu erhöhen.

Ein verbesserter PoS-Algorithmus würde nicht nur größere Belohnungen für aktiv teilnehmende Nutzer ermöglichen, sondern auch förderlich für die obenerwähnte Netzwerksicherheit sein.

ZUSAMMENFÜGEN UND SPLITTEN VON ENIGMA-TRANSAKTIONEN

Enigma erstellt momentan jeweils eine einzelne, verschleierte („cloaked“) Transaktion pro Überweisung. Wir arbeiten zurzeit an einem Update des Enigma-Frameworks, welches ermöglichen soll, dass mehrere verschleierte Enigma-Transaktionen in einer „Super-Transaktion“ zusammengefasst werden. Infolgedessen würde die Anonymität der Cloak-Nutzer weiter ansteigen. Diese Erweiterung verfolgt das Ziel, dass Nutzer die Anzahl gemeinsamer Enigma-Transaktionen auswählen können, die sie zusätzlich zu der Anzahl an Cloakern benötigen.

Dieses Update bleibt weiterhin vollkommen dezentral, privat und sicher. Eine weitere Verbesserung der Enigma-Sendungen, die momentan vom Cloak-Team ausgearbeitet wird, zielt darauf ab, eine große Menge von CloakCoins mithilfe einer Reihe von kleineren Enigma-Transaktionen „cloaken“ zu können. Um dies umsetzen zu können, muss der Nutzer zunächst festlegen, wie viele „gecloakte“ Coins er an eine bestimmte Adresse senden möchte. Im Hintergrund würde CloakCoin dann eine entsprechende Anzahl kleinerer, verschleierter Enigma-Transaktionen erstellen und sie innerhalb eines bestimmten Zeitraums an das Cloak-Netzwerk leiten. Dieser Prozess liefert zusätzlichen Schutz für Cloak-Transfers, da er mit kombinierten Enigma-Transaktionen kompatibel ist.

6. FAQ

FRAGE: WIE WIRKEN CLOAKER UNTERSTÜTZEND BEI EINER ENIGMA-TRANSAKTION MIT?

Cloaker steuern einen oder mehrere Inputs bei, die benutzt werden, um den Input des Senders zu „cloaken“, also zu verschleiern). Außerdem liefern Cloaker eine Reihe von Rücksendeadressen, sodass sie im Nachhinein ihren Input sowie das Belohnungsentgelt erhalten. Die Rücksendeadressen werden sorgfältig ausgewählt, wobei aktive Adressen priorisiert werden. Folglich ist es wesentlich schwieriger Blockchainanalysen durchzuführen und den tatsächlichen Output einer Enigma-Transaktion aufzudecken. Das Enigma-System überprüft ebenfalls die Zieladresse, sodass der tatsächliche Output weitestgehend mit den „gecloakten“ Outputs übereinstimmt.

FRAGE: WIE LANGE DAUERT ES, BIS EINE ENIGMA-TRANSAKTION ABGESCHLOSSEN IST?

Derzeit benötigen Enigma-Transaktionen eine Minute, bis sie abgeschlossen sind. Cloaking-Nodes helfen dabei eine Enigma-Transaktion zu „cloaken“ und behalten die erforderlichen finanziellen Mittel (Coins) ein, bis die

Enigma-Transaktion abgeschlossen bzw. die vorgesehene Zeit überschritten ist. Wird eine Enigma-Transaktion abgebrochen bzw. im vorgesehen Zeitrahmen nicht abgeschlossen, so werden die einbehaltenen Coins zur Wiederverwendung freigegeben.

FRAGE: WIE BEEINFLUSST ENIGMA DEN STAKING-PROZESS?

Bei allen Coins, die in einer Enigma-Transaktion sowohl vom Sender als auch vom Cloaker verwendet werden, wird das Coin-Age auf null zurückgesetzt. Es ist allerdings zu beachten, dass bei der Teilnahme am Cloaking wesentlich höhere Erträge zu erwarten sind als beim reinen Staking. Das Cloak-Team überarbeitet momentan den Enigma-Algorithmus für die anstehende „Hard-Fork"-Veröffentlichung (Enigma 1.1). Weitere Details zu diesem Thema finden Sie unter Punkt 5 „Die Zukunft von Enigma – weitere Entwicklungen".

FRAGE: BENÖTIGT MAN EINE BESTIMMTE ANZAHL AN CLAOKS IM EIGENEN WALLET, UM ALS ENIGMA-CLOAKER FUNGIEREN ZU KÖNNEN?

Man kann an Cloaking-Prozessen teilnehmen, ohne dabei über ein bestimmtes Guthaben im CloakCoin-Wallet verfügen zu müssen. Sobald Cloaking aktiviert wird, wird ein Teil der im Wallet vorhandenen Coins automatisch einbehalten und für die Teilnahme am Cloaking verwendet (Cloaker erhalten für ihre Unterstützung eine Belohnung). In der Regel werden 50% des Coin-Bestandes für Cloaking-Prozesse einbehalten, wobei dieser Standardwert vom Nutzer individuell angepasst werden kann. In einem solchen Fall wird der ausgewählte Wert randomisiert, sodass keine Rückschlüsse auf die Enigma-Transaktion gemacht werden können.

Es ist anzumerken, dass Wallets mit einem höheren Guthaben auch eine größere Chance haben für Cloaking-Prozesse ausgewählt zu werden.
Dies kann damit begründet werden, dass solche Wallets mit größerer Wahrscheinlichkeit über ein Cloak-Guthaben verfügen, welches für größere Enigma-Transaktionen erforderlich ist.

FRAGE: WIE KANN MAN SICH VOR EINEM ANGRIFF SCHÜTZEN, BEI DEM JEMAND GEZIELT DIE BLOCKCHAIN NACH IDENTISCHEN IN- UND OUTPUTS UNTERSUCHT?

Enigma-Transaktionen gruppieren die Outputs, wobei immer mehrere passende Output-Beträge vorhanden sind. So kann der tatsächliche Output des Empfängers „gecloakt“ werden.

FRAGE: IST ES MÖGLICH DEN ERZEUGER EINER ENIGMA-TRANSAKTION AUSFINDIG ZU MACHEN, INDEM DAS SIGNATUREN-SCRIPT UNTERSUCHT UND FOLGLICH DIE REIHENFOLGE DER SIGNATUREN IDENTIFIZIERT?

Nein. Während des Signierens wird die Reihenfolge der Signaturen im Script zufällig festgelegt. Dies tut der Sender in Zusammenarbeit mit den teilnehmenden Cloaking-Nodes.

FRAGE: KANN EIN EAVESDROPPER („LAUSCHER“) DAS NETZWERK ÜBERWACHEN, UM AUSGEHENDE ENIGMA-TRANSAKTIONEN AUSFINDIG ZU MACHEN UND SO DEN SENDER ZU IDENTIFIZIEREN?

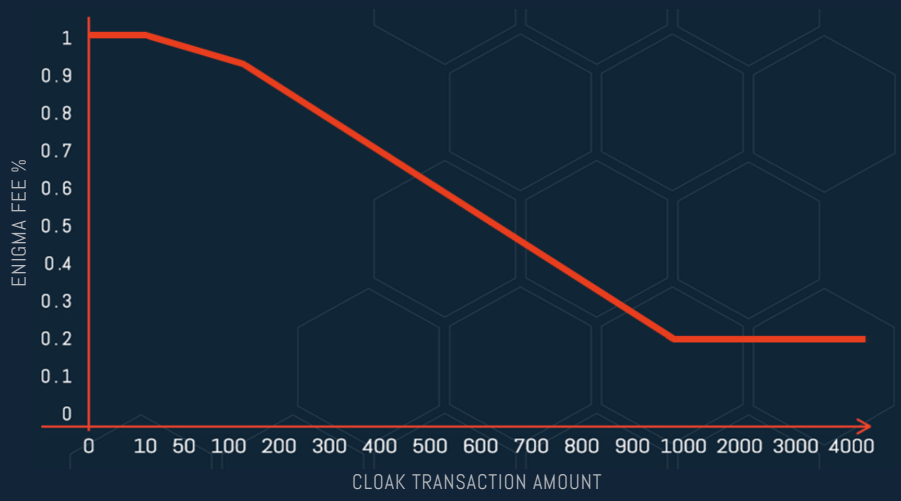
Nein. Alle Parteien senden ihre Transaktionen in zufälliger Reihenfolge an das Netzwerk. Somit kann Angriffen von Eavesdroppern entgegengewirkt werden.

FRAGE: WIE HOCH IST DIE GEBÜHR EINER ENIGMA-TRANSAKTION?

Die Gebühr beträgt 1% bei 0 Coins und bis zu 0,2% bei 1000 und mehr Coins. Sie wird dafür verwendet, um Enigma-Nodes für ihre Unterstützung beim Cloaking von Transaktionen zu belohnen. Dabei wird die Gebühr mit der Transaktion vermischt und anschließend unter den Cloakern aufgeteilt. Es ist anzumerken, dass die Gebühren nicht nur als Belohnung für die Teilnehmer dienen, sondern es auch so gut wie unmöglich machen den Transaktionsbetrag ausfindig zu machen. Jeder Teilnehmer erhält 80% - 120% einer gleichmäßig aufgeteilten Enigma-Transaktion.

FRAGE: WIE WIRD DIE ENIGMA-GEBÜHR FESTGELEGT?

Die Prozentzahl (%) der Enigma-Gebühr lässt sich wie folgt ermitteln:



TX AMOUNT	ENIGMA FEE %	CLOAK FEE
0	1.00	0
10	0.992	0.0992
50	0.96	048
100	0.92	0.92
200	0.84	1.68
300	0.76	2.28
400	0.68	2.72
500	0.60	3.00
600	0.52	3.12
700	044	3.08
800	0.36	2.88
900	0.28	2.52
1000	0.20	2.00
2000	0.20	4.00
3000	0.20	6.00
4000	0.20	8.00

FRAGE: BENÖTIGT DAS CLOAK-NETZWERK FÜR ENIGMA EINEN HARD-FORK?

Nein. Ältere CloakCoin-Clients können Enigma-Transaktionen problemlos handhaben, jedoch können sie keine eigene Transaktion erstellen bzw. „cloaken“. Aufgrund der kommenden Veränderungen des Proof-of-Stake Algorithmus sowie zusätzlicher Script-Opcodes für Markt Features (z.B. Block Escrow) wird für die nächste Enigma-Überarbeitung allerdings ein Hard-Fork erforderlich sein.

FRAGE: WIE VIELE CLOAKER KÖNNEN MAXIMAL BEI EINER ENIGMA-TRANSAKTION UNTERSTÜTZEN?

Maximal können 25 Cloaker bei einer Enigma-Transaktion mitwirken. Das Enigma-System ist jedoch flexibel, weshalb dieses Maximum problemlos erweitert werden kann.

FRAGE: WIE SCHÜTZT SICH ENIGMA VOR „BAD ACTORS“?

Das Enigma-System beinhaltet einen umfangreichen DDoS-Schutz, um betroffene Nodes für die Dauer einer Session auf eine Blacklist zu setzen. Wenn sich ein Node mehrmals weigert eine abgeschlossene Transaktion zu signieren, so wird dieser Node für den Rest der laufenden Session keine weiteren Cloaking-Einladungen erhalten. Wir sind momentan dabei weitere Methoden für die Bestrafung nicht-kooperativer Nodes zu untersuchen. Wahrscheinlich werden wir ein System implementieren, welches erfordert, dass Cloaker eine Kaution (in Form einer nominalen und zurückerstattbaren Gebühr) bei einem Dritten hinterlegen müssen. Sollte ein Cloaker dann eine Enigma-Transaktion blockieren, indem er sich weigert zu signieren, so würde er diese Gebühr nicht zurückerstattet bekommen. Es ist anzumerken, dass böswillige Nodes, die daran interessiert sind Enigma-Transaktionen zu behindern, niemals Kapital stehlen oder unterschlagen können.

FRAGE: WIE WERDEN STEALTH- UND ENIGMA-TRANSAKTIONEN ENTDECKT BZW. EMPFANGEN?

Alle eingehenden Transaktionen werden gescannt. Dabei werden Stealth-Transaktionen immer als erstes gescannt, indem der Einweg-PubKey benutzt wird, welcher in einem zufälligen OP_RETURN-Transaktionsoutput enthalten ist. Anschließend werden Enigma-Transaktionen gescannt, wobei ebenfalls der standardmäßige Einweg-PubKey verwendet wird. Zahlungen hingegen erfordern einen zusätzlichen Schritt, bei dem ein weiterer abgeleiteter Schlüssel benutzt wird. Enigma-Outputs werden erzeugt, indem ein Hash des Einweg-Pubkeys, eine private Hash-Stealth-Adresse und der Output-Index verwendet werden.

Wenn im Netzwerk nach Enigma-Transaktionen gesucht wird, so werden Zero-Index Bezahlungsadressen für jede vorhandene Stealth-Adresse generiert [HASH(ephemeral_pubkey, hash_stealth_secret, 0). Wird ein Treffer für den Zero-Index einer Stealth-Adresse gefunden, so werden zusätzliche Adressen für die restlichen Indizes [num_tx_outputs] generiert und anschließend gescannt, um Zahlungen aufzudecken. Weitere Informationen dazu finden Sie unter wallet.cpp („FindEnigmaTransactions“).

Eine ähnliche Scan-Methode wird von Cloakern angewandt, bevor sie eine Enigma-Transaktion signieren (um rückvergütet zu werden). Weitere Informationen dazu finden Sie unter wallet.cpp („GetEnigmaOutputsAmounts").

7. VERWEISE

[01] http://bitcoin.org

[02] https://en.bitcoin.it/wiki/Category:Mixing Services

[03] https://wiki.openssl.org/index.php/Elliptic_Curve_Diffie_Hellman

[04] http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization

[05] https://bitcointalk.org/index.php?topic=279249.0
(CoinJoin: Bitcoin Privacy for the Real World)

[06] https://bitcointalk.org/index.php?topic=27787.0
(Proof of Stake Instead of Proof of Work)

[07] https://en.bitcoin.it/wiki/Proof_of_Stake

[08] https://en.bitcoin.it/wiki/Deterministic_wallet

[09] https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki

[10] http://www.onion-router.net

www.cloakcoin.com

https://chat.cloakcoin.com

www.twitter.com/CloakCoin