KAPITEL 9

Transaktionsgebühren

Die von Alice in Kapitel 8 erzeugte digitale Signatur beweist nur, dass sie ihren privaten Schlüssel kennt und die Bob bezahlende Transaktion bestätigt. Sie könnte noch eine weitere Signatur erzeugen, die stattdessen eine Carol bezahlende Transaktion bestätigt – eine Transaktion, die den gleichen Output (Bitcoins) einlöst, den sie auch für die Zahlung an Bob genutzt hat. Wir haben es nun mit *kollidierenden Transaktionen* zu tun, weil eine Transaktion, die einen bestimmten Output einlöst, in die gültige Blockchain mit dem höchsten Proof-of-Work aufgenommen wird. Diese Blockchain wird von Full Nodes verwendet, um zu ermitteln, welche Schlüssel welche Bitcoins kontrollieren.

Um sich selbst vor kollidierenden Transaktionen zu schützen, wäre Bob gut beraten, abzuwarten, bis Alice' Transaktion mit ausreichender Tiefe in die Blockchain aufgenommen wurde, bevor er das empfangene Geld als sein Eigentum betrachtet (siehe »Bestätigungen« auf Seite 40). Damit Alice' Transaktion in die Blockchain aufgenommen wird, muss sie in einen *Block* von Transaktionen eingetragen werden. In einer bestimmten Zeitspanne kann nur eine bestimmte Anzahl von Blöcken erzeugt werden, und jeder Block hat nur begrenzt Platz. Lediglich der Miner, der den Block erzeugt, entscheidet, welche Transaktionen darin aufgenommen werden. Miner können Transaktionen nach ihren eigenen Kriterien auswählen. Sie könnten sich sogar entscheiden, alle Transaktionen abzulehnen.

Wenn wir in diesem Kapitel von »Transaktionen« sprechen, meinen wir jede Transaktion in einem Block außer der ersten Transaktion. Die erste Transaktion in einem Block ist die *Coinbase-Transaktion*, beschrieben in »Coinbase-Transaktionen« auf Seite 163, die es dem Miner des Blocks erlaubt, die Belohnung für die Erzeugung dieses Blocks einzuziehen. Im Gegensatz zu anderen Transaktionen gibt eine Coinbase-Transaktion nicht den Output einer vorherigen Transaktion frei. Auch verschiedene andere Regeln für normale Transaktionen gelten für Coinbase-Transaktionen nicht. Sie zahlen keine Transaktionsgebühren, müssen nicht durch Gebühren angeschoben werden, Transaktions-Pinning betrifft sie nicht, und sie sind für die folgende Diskussion über Gebühren größtenteils uninteressant – deshalb werden wir sie in diesem Kapitel ignorieren.

Das Kriterium, nach dem die meisten Miner die Transaktionen wählen, die in ihre Blöcke aufgenommen werden, ist die Höhe des Umsatzes. Bitcoin unterstützt das mit einem Mechanismus, der es einer Transaktion erlaubt, dem Miner Geld zu geben, der die Transaktion in einen Block aufnimmt. Wir nennen diesen Mechanismus *Transaktionsgebühren*, obwohl es sich dabei nicht um eine Gebühr im eigentlichen Sinne handelt. Sie ähnelt eher dem Gebot bei einer Auktion. Das gekaufte Gut ist dabei ein Teil des beschränkten Platzes in einem Block, den eine Transaktion verbraucht. Miner wählen eine Reihe von Transaktionen aus, die ihnen den größten Umsatz bringen.

In diesem Kapitel wollen wir verschiedene Aspekte dieser Gebote – Transaktionsgebühren – betrachten und wie sie die Erzeugung und Verwaltung von Bitcoin-Transaktionen beeinflussen.

Wer zahlt die Transaktionsgebühr?

Die meisten Zahlungssysteme verlangen für die Abwicklung irgendeine Form von Gebühr, auch wenn diese dem typischen Käufer meist verborgen bleibt. Zum Beispiel könnte ein Händler einen Artikel zum gleichen Preis anbieten, unabhängig davon, ob man bar oder mit Kreditkarte bezahlt, obwohl der Zahlungsdienstleister für die Verarbeitung der Kreditkarte höhere Gebühren verlangt als die Bank für die Einzahlung von Bargeld.

Bei Bitcoin muss jede Zahlung authentifiziert werden (üblicherweise durch eine Signatur), sodass eine Transaktion keine Gebühr zahlen kann, ohne dass der Zahlungspflichtige dem zustimmt. Der Zahlungsempfänger kann eine Gebühr in einer anderen Transaktion zahlen – was wir später noch sehen werden –, doch wenn eine Transaktion ihre eigene Gebühr zahlen soll, muss sie mit dem Zahlungspflichtigen in irgendeiner Form abgesprochen sein. Versteckte Gebühren sind nicht möglich.

Bitcoin-Transaktionen sind so aufgebaut, dass der Zahlungspflichtige keinen zusätzlichen Raum in der Transaktion benötigt, um die Gebühr zu bestätigen. Das bedeutet, dass es zwar möglich ist, die Gebühr in einer anderen Transaktion zu bezahlen, dass es aber effizienter (und daher günstiger) ist, die Gebühr in einer einzelnen Transaktion zu bezahlen.

Bei Bitcoin ist die Gebühr ein (An-)Gebot, und die Höhe des Gebots bestimmt zu einem großen Teil, wie lange die Bestätigung einer Transaktion dauert. Der Zahlende und der Empfänger sind beide daran interessiert, dass die Bestätigung schnell erfolgt, weshalb es ein Problem sein kann, wenn nur der Zahlende die Gebühr wählt. Eine Lösung dieses Problems sehen wir uns in »Child-Pays-for-Parent (CPFP)« auf Seite 232 an. Allerdings sind bei den üblichen Zahlungsströmen die Parteien, die eine Transaktion schnell bestätigt sehen wollen (und daher auch eher bereit sind, höhere Gebühren zu bezahlen), die Zahlungspflichtigen.

Aus diesen Gründen ist es bei Bitcoin üblich, dass der Zahlungspflichtige die Gebühr zahlt. Es gibt aber auch Ausnahmen, etwa wenn ein Händler unbestätigte Transaktionen akzeptiert oder wenn Protokolle Transaktionen nicht sofort veröf-

fentlichen, nachdem sie signiert wurden. (Der Zahlungspflichtige hat dann keine Möglichkeit, eine angemessene Gebühr zu wählen.) Diese Ausnahmen sehen wir uns später noch an.

Gebühren und Gebührensätze

Jede Transaktion zahlt nur eine Gebühr. Wie groß die Transaktion ist, spielt dabei keine Rolle. Doch je größer eine Transaktion ist, desto weniger kann der Miner in einen Block aufnehmen. Aus diesem Grund bewerten Miner Transaktionen so ähnlich, wie Sie gleichwertige Produkte im Supermarkt auswählen: Sie teilen den Preis durch die Menge.

Während Sie verschiedene Packungen Reis durch das jeweilige Gewicht teilen, um den niedrigsten Preis pro Kilogramm zu bestimmen (bester Preis), teilen Miner die Gebühr einer Transaktion durch deren Größe (die auch *Gewicht*, *Weight*, genannt wird), um die höchste Gebühr pro Gewicht (den höchsten Umsatz) zu bestimmen. Bei Bitcoin verwenden wir den Begriff *Gebührensatz* (engl. *Fee Rate*) für die Größe einer Transaktion geteilt durch ihr Gewicht. Über die Jahre hat Bitcoin verschiedene Änderungen durchlaufen, weshalb der Gebührensatz in verschiedenen Einheiten ausgedrückt werden kann:

- BTC/Bytes (veraltet, wird heute nur noch selten verwendet)
- BTC/Kilobytes (veraltet, wird heute nur noch selten verwendet)
- BTC/Vbytes (wird selten verwendet)
- BTC/Kilo-vbyte (wird hauptsächlich von Bitcoin Core genutzt)
- Satoshi/Vbyte (wird heutzutage am häufigsten genutzt)
- Satoshi/Weight (wird heutzutage ebenfalls häufig genutzt)

Wir empfehlen entweder *sat/vbyte* oder *sat/weight* für die Darstellung von Gebührensätzen.

Vorsicht bei Vorschlägen für Gebührensätze. Kopiert ein Nutzer eine Gebührenrate mit einem bestimmten Nenner in ein Feld mit einem anderen Nenner, kann er die Gebühr bis tausendfach überzahlen. Ändert er stattdessen den Nenner, kann er die Gebühr theoretisch millionenfach überzahlen. Wallets machen es dem Nutzer schwer, exzessive Gebühren zu bezahlen, und wollen vom Nutzer eine Bestätigung, wenn die Gebührenrate nicht über eine vertrauenswürdige Datenquelle berechnet wurde.

Eine exzessive Gebühr, auch *absurde Gebühr*, ist jede Gebührenrate, die deutlich über der Gebührenschätzung liegt, die momentan als notwendig erachtet wird, um eine Transaktion in den nächsten Block aufzunehmen. Wallets sollen es Nutzern nicht unmöglich machen, absurde Gebührensätze zu wählen, sollen es aber erschweren, solche Gebührensätze versehentlich zu wählen. In seltenen Fällen gibt es für den Nutzer legitime Gründe, Gebühren zu überzahlen.

Angemessene Gebührenraten bestimmen

Sie können eine geringe Gebührenrate zahlen, wenn Sie bereit sind, länger auf die Bestätigung Ihrer Transaktion zu warten. Ist die Gebühr allerdings zu gering, kann es passieren, dass die Transaktion nie bestätigt wird. Da die Gebührenraten in einer offenen Auktion Gebote für den Platz in einem Block darstellen, kann man nicht genau vorhersagen, welche Gebühr gezahlt werden muss, um die Transaktion innerhalb einer bestimmten Zeit zu bestätigen. Allerdings kann anhand der Gebührenraten früherer Transaktionen eine grobe Schätzung vorgenommen werden.

Eine Full Node hält drei Informationen zu jeder Transaktion fest: die Zeit (Blockhöhe), zu der die Transaktion eingegangen ist, die Blockhöhe, bei der die Transaktion bestätigt wurde, und die für die Transaktion gezahlte Gebührenrate. Indem wir Transaktionen zusammenfassen, die bei einer ähnlichen Höhe eingegangen sind, bei einer ähnlichen Höhe bestätigt wurden und vergleichbare Gebühren gezahlt haben, können wir berechnen, wie viele Blöcke es dauert, Transaktionen zu bestätigen, für die wir eine bestimmte Gebühr bezahlen. Wir können dann davon ausgehen, dass eine ähnliche Gebührenrate eine vergleichbare Anzahl von Blöcken für die Bestätigung benötigt. Bitcoin Core enthält einen Gebührenkalkulator, der auf diesen Prinzipien basiert. Sie können ihn über den `estimatesmartfee`-RPC aufrufen, wobei Sie als Parameter festlegen, wie viele Blöcke Sie zu warten bereit sind, bevor eine Transaktion höchstwahrscheinlich bestätigt wird (144 Blöcke entsprechen beispielsweise etwa einem Tag):

```
$ bitcoin-cli -named estimatesmartfee conf_target=144
{
  "feerate": 0.00006570,
  "blocks": 144
}
```

Viele webbasierte Dienste bieten über ein API ebenfalls Gebührenschätzungen an. Eine aktuelle Liste finden Sie unter *https://oreil.ly/TB6IN*.

Wie bereits erwähnt, kann die Gebührenschätzung niemals perfekt sein. Ein typisches Problem besteht darin, dass sich der grundlegende Bedarf ändern kann, wodurch sich das Gleichgewicht entweder in Richtung höherer Preise (Gebühren) oder in Richtung Minimum bewegt. Sinkt die Gebührenrate, zahlt eine Transaktion mit der bisher üblichen Gebühr plötzlich eine höhere Gebührenrate und wird früher als erwartet bestätigt. Es gibt keine Möglichkeit, die Gebühr einer bereits gesendeten Transaktion zu verringern, d. h., die höhere Gebühr bleibt an Ihnen hängen. Doch wenn die Gebühren steigen, muss es die Möglichkeit geben, die Gebührenraten für diese Transaktionen zu erhöhen. Dieses »Anstoßen« der Gebühren nennt man *Fee-Bumping*. Bei Bitcoin sind zwei Arten von Fee-Bumping üblich: RBF (*Replace-by-Fee*) und CPFP (*Child-Pays-for-Parent*).

Replace-By-Fee (RBF)

Um die Gebühr einer Transaktion per RBF (*Replace-By-Fee*, deutsch etwa »Ersetze durch Gebühr«) zu erhöhen, erzeugen Sie eine kollidierende Version der Transaktion, die eine höhere Gebühr zahlt. Zwei oder mehr Transaktionen sind *kollidierende Transaktionen*, wenn nur eine in einen gültigen Block aufgenommen werden kann. Der Miner muss also entscheiden, welche der beiden er aufnimmt. Solche Kollisionen treten auf, wenn zwei oder mehr Transaktionen den gleichen UTXO ausgeben wollen, also wenn sie einen Input enthalten, der auf den gleichen Outpoint (den Output einer vorherigen Transaktion) verweist.

Um zu verhindern, dass jemand große Mengen an Bandbreite verbraucht, indem er eine unbeschränkte Anzahl kollidierender Transaktionen erzeugt und sie durch das Netzwerk weiterleitender Full Nodes sendet, verlangen Bitcoin Core und andere die Ersetzung von Transaktionen unterstützende Full Nodes, dass jede Ersatztransaktion eine höhere Gebühr zahlt als die zu ersetzende Transaktion. Bitcoin Core verlangt momentan außerdem, dass die Ersatztransaktion eine höhere Gesamtgebühr zahlt, was aber zu ungewollten Nebeneffekten führen kann. Während wir diese Zeilen schreiben, suchen Entwicklerinnen und Entwickler nach Möglichkeiten, dieses Problem zu beheben.

Bitcoin Core unterstützt momentan zwei RBF-Varianten:

Opt-in-RBF
: Eine unbestätigte Transaktion kann Minern und anderen Full Nodes signalisieren, dass der Ersteller der Transaktion sie durch eine Version mit einer höheren Gebühr ersetzen möchte. Das Signal und die dafür notwendigen Regeln sind in BIP125 spezifiziert. Während wir diese Zeilen schreiben, ist das bei Bitcoin Core seit Jahren standardmäßig aktiviert.

Vollständiges (Full) RBF
: Jede unbestätigte Transaktion kann durch eine Version mit einer höheren Gebührenrate ersetzt werden. Während wir dies schreiben, kann das bei Bitcoin Core optional aktiviert werden (ist standardmäßig aber deaktiviert).

Warum gibt es zwei RBF-Varianten?

Der Grund für zwei verschiedene RBF-Varianten ist die Kontroverse um vollständiges RBF. Frühe Bitcoin-Versionen erlaubten das Ersetzen von Transaktionen, doch dieses Verhalten wurde für verschiedene Releases deaktiviert. Während dieser Zeit ersetzte ein diese Software (die heute Bitcoin Core genannt wird) nutzender Miner oder eine Full Node die erste Version einer unbestätigten Transaktion nicht, wenn sie eine andere Version empfingen. Einige Händler verließen sich auf dieses Verhalten. Sie gingen davon aus, dass eine unbestätigte Transaktion, die eine angemessene Gebühr bezahlt hat, letztlich bestätigt werden würde, und gaben ihre Güter oder Leistungen kurz nach dem Empfang einer unbestätigten Transaktion frei.

Allerdings kann das Bitcoin-Protokoll nicht garantieren, dass eine unbestätigte Transaktion irgendwann bestätigt wird. Wie bereits weiter oben in diesem Kapitel erläutert, entscheidet jeder Miner für sich, welche Transaktion er bestätigt, und das gilt auch für die Version der Transaktion. Bitcoin Core ist Open-Source-Software, sodass jeder im Quellcode das Ersetzen von Transaktionen hinzufügen (oder entfernen) kann. Selbst wenn Bitcoin Core nicht Open Source wäre, so ist Bitcoind doch ein offenes Protokoll, und jeder ausreichend kompetente Programmierer kann es implementieren. Diese Reimplementierung kann Ersatztransaktionen erlauben oder auch nicht.

Das Ersetzen von Transaktionen zerstört die Erwartungen einiger Händler, die sich darauf verlassen, dass unbestätigte Transaktionen letztlich bestätigt werden. Eine alternative Version einer Transaktion kann die gleichen Outputs bezahlen wie die Ursprungsversion, muss sie aber nicht. Zahlt die erste Version einer Transaktionen einen Händler aus, muss das bei der zweiten Version also nicht der Fall sein. Stellt der Händler basierend auf der ersten Version seine Güter oder Dienste zur Verfügung und wird die zweite Version bestätigt, wird der Händler nicht bezahlt.

Einige Händler (und ihre Unterstützer) haben gefordert, das Ersetzen von Transaktionen in Bitcoin Core nicht wieder zu aktivieren. Andere haben darauf hingewiesen, dass das Ersetzen von Transaktionen Vorteile bietet, einschließlich der Möglichkeit des Fee-Bumpings für Transaktionen mit ursprünglich zu geringen Gebühren.

Letztlich haben die an Bitcoin Core arbeitenden Entwicklerinnen und Entwickler einen Kompromiss implementiert: Statt die Ersetzung jeder unbestätigten Transaktion zu erlauben (vollständiges RBF), wurde Bitcoin Core so programmiert, dass nur Transaktionen ersetzt werden dürfen, die die Ersetzung erlauben wollen (Opt-in-RBF). Händler können eingehende Transaktion auf ein entsprechendes Opt-in-Signal prüfen und solche Transaktionen anders handhaben.

Das fundamentale Problem wird dadurch aber nicht gelöst: Jeder kann sich eine Kopie von Bitcoin Core beschaffen und eine eigene Version implementieren, um vollständige RBFs zu erlauben. Einige Entwickler haben das sogar getan, doch nur wenige Menschen haben ihre Software verwendet.

Nach einigen Jahren haben die an Bitcoin Core arbeitenden Entwickler diesen Kompromiss leicht abgewandelt. Neben dem standardmäßig aktiven Opt-in-RBF haben sie eine Option eingefügt, die die Aktivierung vollständiger RBFs erlaubt. Bei ausreichender Mining-Hashrate und genug Full Nodes, die diese Option aktivieren, kann irgendwann jede unbestätigte Transaktion durch eine Version mit einer höheren Gebühr ersetzt werden. Während wir dies schreiben, ist noch nicht klar, ob das bereits passiert ist.

Wenn Sie als Nutzer Fee-Bumping nach RBF nutzen wollen, müssen Sie zuerst eine Wallet wählen, die das unterstützt, etwa eine der in *https://oreil.ly/IhMzx* aufgeführten Wallets, bei denen »Sending support« aktiv ist.

Wollen Sie als Entwickler Fee-Bumping nach RBF implementieren, müssen Sie zuerst entscheiden, ob Sie Opt-in- oder vollständiges RBF unterstützen wollen. Während wir dies schreiben, ist Opt-in-RBF die einzige Methode, die garantiert funktio-

niert. Selbst wenn vollständige RBFs verlässlich arbeiten, werden Opt-in-RBFs auf Jahre hinaus immer noch etwas schneller bestätigt werden als Ersetzungen durch vollständige RBFs. Wenn Sie sich für Opt-in-RBF entscheiden, müssen Sie die Signalisierung nach BIP125 implementieren. Dabei handelt es sich um eine einfache Modifikation eines der Sequenzfelder einer Transaktion (siehe »Sequenz« auf Seite 150). Wenn Sie sich für vollständiges RBF entscheiden, müssen Sie Ihre Transaktionen nicht um eine Signalisierung ergänzen. Alle anderen Aspekte von RBF sind bei beiden Ansätzen identisch.

Braucht eine Transaktion einen Fee-Bump, erzeugen Sie einfach eine neue Transaktion mit mindestens einem UTXO der zu ersetzenden Transaktion. Die Outputs sollten gleich bleiben, um den oder die gleichen Empfänger zu erreichen. Sie können die Gebühr erhöhen, indem Sie Ihr Rückgeld reduzieren oder indem Sie zusätzliche Inputs in die Transaktion einfügen. Entwickler sollten eine Fee-Bumping-Schnittstelle zur Verfügung stellen, die dem Nutzer die gesamte Arbeit abnimmt und ihn einfach fragt (oder ihm empfiehlt), wie hoch die Gebührenrate sein soll.

Vorsicht bei der Erzeugung von mehr als einer Ersatztransaktion. Sie müssen sicherstellen, dass alle Versionen dieser Transaktion miteinander kollidieren. Ist das nicht der Fall, können mehrere separate Transaktionen bestätigt werden, und der Empfänger bekommt mehr, als ihm zusteht. Ein Beispiel:

- Transaktion Version 0 enthält Input *A*.
- Transaktion Version 1 enthält die Inputs *A* und *B* (d.h., Sie haben Input *B* eingefügt, um die zusätzlichen Gebühren zu bezahlen).
- Transaktion Version 2 enthält die Inputs *B* und *C* (d.h., Sie haben Input *C* eingefügt, um die zusätzlichen Gebühren zu bezahlen, doch *C* war groß genug, sodass Input *A* nicht länger benötigt wurde).

Bei diesem Szenario kann jeder Miner, der Version 0 dieser Transaktion gespeichert hat, sowohl diese als auch Version 2 der Transaktion bestätigen. Zahlen beide Versionen den gleichen Empfänger aus, wird er zweimal bezahlt (und der Miner erhält die Gebühren für zwei verschiedene Transaktionen).

Eine einfache Methode, dieses Problem zu vermeiden, besteht darin, in die Ersatztransaktion immer die gleichen Inputs aufzunehmen wie in die vorherige Version der Transaktion.

Der Vorteil des RBF-Fee-Bumpings gegenüber anderen Fee-Bumping-Arten ist die sehr effiziente Nutzung des Platzes im Block. Häufig hat die Ersatztransaktion die gleiche Größe wie die Transaktion, die sie ersetzt. Selbst wenn sie größer ist, hat sie häufig die gleiche Größe wie die Transaktion, die der Nutzer erzeugt hätte, wenn er von vornherein die höhere Gebühr verwendet hätte.

Der grundlegende Nachteil von RBF-Fee-Bumping ist, dass es normalerweise nur der Urheber der Transaktion (also die Personen, die Signaturen oder andere Authentifizierungsdaten für diese Transaktion bereitstellen mussten) veranlassen kann. Eine Ausnahme bilden diejenigen Transaktionen, die über Sighash-Flags das Einfü-

gen zusätzlicher Inputs erlauben (siehe »Arten von Signatur-Hashes (SIGHASH)« auf Seite 207), doch das stellt einen vor ganz eigene Herausforderungen. Wenn Sie der Empfänger einer unbestätigten Transaktion sind und diese beschleunigen wollen, gilt generell, dass Sie keine RBF-Fee-Bumps nutzen können. Sie müssen eine andere Methode verwenden.

Es gibt weitere Probleme mitRBF, die wir uns in »Transaktions-Pinning« auf Seite 234 noch genauer ansehen.

Child-Pays-for-Parent (CPFP)

Der Empfänger einer unbestätigten Transaktion kann Miner zur Bestätigung dieser Transaktion animieren, indem er den entsprechenden Output einlöst. Die Transaktion, die Sie bestätigt haben wollen, wird *Eltern-(Parent-)Transaktion*) genannt. Die Transaktion, die einen Output der Parent-Transaktion einlöst, nennt man *Kind-(Child-)Transaktion*.

Wie wir in »Outpoint« auf Seite 148 gelernt haben, muss jeder Input einer bestätigten Transaktion den nicht eingelösten Output einer Transaktion referenzieren, der an einer früheren Stelle der Blockchain steht (sei es im gleichen oder in einem vorherigen Block). Das bedeutet, dass der Miner, der eine Child-Transaktion bestätigen will, sicherstellen muss, dass auch die Parent-Transaktion bestätigt ist. Ist die Parent-Transaktion noch nicht bestätigt, die Gebühr der Child-Transaktion aber hoch genug, kann es für den Miner profitabel sein, beide im gleichen Block zu bestätigen.

Um die Profitabilität des Minings einer Parent- und einer Child-Transaktion zu ermitteln, betrachtet der Miner sie als *Transaktionspaket* mit aggregierter Größe und Gebühr. Die Gebühr kann dann durch die Größe geteilt werden, um eine *Paket-Gebührenrate* (*Package Fee Rate*) zu berechnen. Der Miner sortiert alle ihm bekannten Transaktionen und Transaktionspakete und nimmt die profitabelsten in den Block auf, den er zu schürfen versucht, wobei er die maximale Größe (Gewicht) berücksichtigt, die in den Block eingefügt werden darf. Um noch profitablere Pakete aufzuspüren, kann der Miner sogar Pakete über mehrere Generationen hinweg analysieren (z.B. eine unbestätigte Eltern-Transaktion in Kombination mit dessen Kind und Enkel). Das wird als *Ancestor-Fee-Rate-Mining* bezeichnet (zu Deutsch etwa *vorfahrenbasiertes Gebührenraten-Mining*).

Bitcoin Core hat Ancestor-Fee-Rate-Mining seit vielen Jahren implementiert, und man geht davon aus, dass nahezu alle Miner es nutzen, während diese Zeilen geschrieben werden. Es ist den Wallets also möglich, dieses Feature für das Fee-Bumping einer eingehenden Transaktion zu nutzen, indem man mit einer Child-Transaktion für dessen Parent bezahlt (CPFP).

CPFP hat gegenüber RBF mehrere Vorteile. Jeder, dem ein Output einer Transaktion zusteht, kann CPFP nutzen – sowohl der Zahlungsempfänger als auch der Zahlende (wenn ihm Wechselgeld zusteht). Darüber hinaus muss die ursprüngliche Transaktion nicht ersetzt werden, wodurch es für einige Händler besser geeignet ist als RBF.

Der wesentliche Nachteil von CPFP gegenüber RBF ist, dass CPFP üblicherweise mehr Platz in einem Block benötigt. Bei RBF hat eine Fee-Bump-Transaktion meist die gleiche Größe wie die Transaktion, die sie ersetzt. Bei CPFP fügt der Fee-Bump eine weitere Transaktion ein. Dieser zusätzliche Platz im Block verlangt über die Kosten des Fee-Bumps hinaus zusätzliche Gebühren.

CPFP stellt einen vor verschiedene Herausforderungen, von denen wir uns einige in »Transaktions-Pinning« auf Seite 234 ansehen werden. Ein weiteres Problem, das wir explizit ansprechen müssen, ist die minimale Relay-Gebührenrate, die wir im nächsten Abschnitt diskutieren.

Paketweiterleitung (Package Relay)

Bei früheren Versionen von Bitcoin Core gab es für die Mempools keine Beschränkung der Anzahl unbestätigter Transaktionen für die spätere Weiterleitung und das Mining (siehe »Mempools und Waisenpools« auf Seite 264). Doch natürlich haben Computer physikalische Grenzen, sei es Arbeitsspeicher (RAM) oder Plattenplatz. Eine Full Node kann unmöglich eine unbeschränkte Anzahl unbestätigter Transaktionen speichern. Spätere Versionen von Bitcoin Core schränkten die Größe des Mempools auf etwa die Tagesmenge an Transaktionen ein und speicherten nur die Transaktionen und Pakete mit der höchsten Gebührenrate.

Das funktioniert in den meisten Fällen ausgezeichnet, führt aber zu einem Abhängigkeitsproblem. Um die Gebührenrate eines Transaktionspakets berechnen zu können, benötigen wir die Parent-Transaktion sowie alle Transaktionen der Nachkommen. Doch wenn die Parent-Transaktion keine ausreichend hohe Gebührenrate zahlt, wird sie nicht im Mempool der Node vorgehalten. Empfängt die Node eine Child-Transaktion, ohne auf deren Parent zugreifen zu können, kann sie mit der Transaktion nichts anfangen.

Die Lösung dieses Problems besteht darin, die Transaktionen als Paket zu versenden, was man als *Package Relay* bezeichnet. Die empfangende Node kann die Gebührenrate des gesamten Pakets untersuchen, bevor sie an einzelnen Transaktionen arbeitet. Während diese Zeilen geschrieben werden, haben die Bitcoin-Core-Entwicklerinnen und -Entwickler deutliche Fortschritte bei der Implementierung der Paketweiterleitung gemacht. Eine frühe Version wird wohl verfügbar sein, wenn dieses Buch veröffentlicht wird.

Package Relay ist für Protokolle besonders wichtig, die auf zeitkritischen, vorsignierten Transaktionen basieren, etwa das Lightning Network (LN). In einigen nicht kooperativen Fällen ist ein Fee-Bumping per RBF für Transaktionen nicht möglich, weshalb sie von CPFP abhängig sind. Bei diesen Protokollen können einige Transaktionen auch sehr viel früher erzeugt werden, als sie veröffentlicht werden, daher ist es unmöglich, eine angemessene Gebührenrate zu ermitteln. Zahlt eine vorsignierte Transaktion eine zu niedrige Gebühr, schafft sie es nicht in den Mempool, und es gibt für eine Child-Transaktion keine Möglichkeit des Fee-Bumpings. Wird dadurch die rechtzeitige Bestätigung der Transaktion verhindert, kann ein ehrlicher Nutzer Geld verlieren. Package Relay ist die Lösung dieses ernsten Problems.

Transaktions-Pinning

Obwohl das Fee-Bumping bei RBF und CPFP in den grundlegenden Fällen wie beschrieben funktioniert, gibt es bei beiden Methoden Regeln, die Denial-of-Service-Angriffe auf Miner und weiterleitende Full Nodes verhindern sollen. Ein unglücklicher Nebeneffekt dieser Regeln ist, dass sie das Fee-Bumping manchmal verhindern. Das Fee-Bumping einer Transaktion unmöglich zu machen oder zu erschweren, wird *Transaktions-Pinning* genannt.

Eine der wesentlichen Sorgen in Bezug auf Denial-of-Service-Angriffe ergibt sich aus den Beziehungen zwischen den Transaktionen. Sobald der Output einer Transaktion eingelöst wird, wird die Transaktionskennung (txid) durch die Child-Transaktion referenziert. Wird eine Transaktion aber ersetzt, ändert sich auch die txid. Und wird die Ersatztransaktion bestätigt, kann keine ihrer Nachfolgetransaktion in die gleiche Blockchain aufgenommen werden. Es ist möglich, die Nachfolgetransaktionen neu zu erzeugen und zu signieren, doch dafür gibt es keine Garantie. Das hat für RBF und CPFP ähnliche, aber doch voneinander abweichende Auswirkungen:

- Wenn Bitcoin Core im RBF-Kontext eine Ersatztransaktion akzeptiert, vergisst es einfach die Originaltransaktion und alle Nachfolgetransaktionen, die vom Original abhängen. Um sicherzustellen, dass es für Miner profitabel ist, Ersatztransaktionen zu akzeptieren, lässt Bitcoin Core nur Ersatztransaktionen zu, die höhere Gebühren zahlen als alle zu vergessenden Transaktionen zusammen.

  Der Nachteil dieses Ansatzes ist, dass Alice eine kleine Transaktion erzeugen kann, um Bob zu bezahlen, und Bob dann diesen Output nutzt, um eine große Child-Transaktion zu erzeugen. Möchte Alice die Originaltransaktion ersetzen, muss sie eine Gebühr bezahlen, die höher ist als die, die sie und Bob ursprünglich bezahlt haben. Lag Alice' Originaltransaktion beispielsweise bei etwa 100 vbytes und Bobs Transaktion bei etwa 100.000 vbytes und haben beide die gleiche Gebührenrate genutzt, muss Alice bei RBF nun tausendmal mehr für das Fee-Bumping ihrer Transaktion zahlen.
- Erwägt eine Node im CPFP-Kontext, ein Paket in einen Block aufzunehmen, muss sie die Transaktionen in diesem Paket aus allen anderen Paketen entfernen, die für den gleichen Block infrage kommen. Zahlt eine Child-Transaktion beispielsweise für 25 Vorfahren und hat jeder dieser Vorfahren 25 andere Children, verlangt das Einfügen des Pakets in den Block die Aktualisierung von etwa 625 Paketen (25^2). Gleiches gilt für eine Transaktion mit 25 Nachkommen, die aus dem Mempool einer Node entfernt werden soll (z. B. weil sie in einen Block aufgenommen wird). Hat jeder dieser Nachkommen ebenfalls 25 Nachkommen, müssen weitere 625 Pakete aktualisiert werden. Und wenn wir die Zahl der Parameter verdoppeln (z. B. von 25 auf 50), dann muss die Node viermal so viel Arbeit erledigen.

  Darüber hinaus ist es nicht sinnvoll, eine Transaktion und ihre Nachkommen langfristig im Mempool vorzuhalten, wenn eine alternative Version dieser

Transaktion gemint wurde. Keine dieser Transaktionen kann noch bestätigt werden (es sei denn, es kommt zu einer seltenen Reorganisation der Blockchain). Bitcoin Core entfernt alle Transaktionen aus dem Mempool, die in der aktuellen Blockchain nicht mehr bestätigt werden können. Im schlimmsten Fall kann das bei der Node eine enorme Bandbreite verbrauchen und möglicherweise die korrekte Propagation von Transaktionen verhindern.

Um solche (und ähnliche) Probleme zu verhindern, schränkt Bitcoin Core eine Parent-Transaktion auf maximal 25 Vorfahren und Nachfolger im Mempool ein und beschränkt die Gesamtgröße dieser Transaktionen auf 100.000 vbytes. Der Nachteil dieses Ansatzes besteht darin, dass Nutzer keine CPFP-Fee-Bumps erzeugen können, wenn die Transaktion bereits zu viele Nachfahren hat (oder wenn sie und ihre Nachfahren zu groß sind).

Zum Transaktions-Pinning kann es versehentlich kommen, doch es stellt für zeitkritische Multipart-Protokolle wie LN auch eine ernste Schwachstelle dar. Kann eine Gegenpartei verhindern, dass eine ihrer Transaktionen rechtzeitig bestätigt wird, kann sie Ihr Geld stehlen.

Protokollentwickler arbeiten seit Jahren daran, Probleme mit Transaktions-Pinning zu beheben. Eine Teillösung wird in »CPFP-Carve-out und Anker-Outputs« auf Seite 235 beschrieben. Verschiedene andere Lösungen wurden vorgeschlagen, und zumindest eine Lösung wird aktiv entwickelt, während wir diese Zeilen schreiben – kurzlebige Anker (*https://oreil.ly/300dv*).

CPFP-Carve-out und Anker-Outputs

2018 standen die Entwickler von LN vor einem Problem. Ihr Protokoll nutzte Transaktionen, die von zwei verschiedenen Parteien signiert werden mussten. Keine Partei wollte der anderen vertrauen, deshalb mussten die Transaktionen an einem Punkt signiert werden, an dem Vertrauen nicht nötig war. Beide Parteien konnten diese Transaktionen veröffentlichen, wenn die Gegenpartei ihre Pflichten nicht erfüllen wollte (oder konnte). Da keine Partei der anderen vertrauen will, müssen Transaktionen signiert werden. Das Problem mit diesem Ansatz besteht darin, dass die Transaktionen zu einem unbekannten (weit in der Zukunft liegenden) Zeitpunkt veröffentlicht werden müssen. Eine vernünftige Schätzung der Gebührenrate ist daher unmöglich.

Theoretisch hätten die Entwickler ihre Transaktionen so entwerfen können, dass ein Fee-Bumping sowohl per RBF (mittels spezieller Sighash-Flags) als auch per CPFP möglich ist, doch beide Protokolle sind für Transaktions-Pinning anfällig. Da die involvierten Transaktionen zeitkritisch sind, ist die Möglichkeit einer Partei, durch Transaktions-Pinning die Bestätigung zu verzögern, eine reproduzierbare Sicherheitslücke, die bösartige Parteien ausnutzen können, um ehrlichen Parteien Geld zu stehlen.

Der LN-Entwickler Matt Corallo hat eine Lösung vorgeschlagen, die die CPFP-Fee-Bumping-Regeln um eine Ausnahme erweitert: das sogenannte *CPFP-Carve-out*.

Die normalen CPFP-Regeln verbieten die Aufnahme eines weiteren Nachkommens, wenn die Parent-Transaktion dann 26 oder mehr Nachfahren hat oder wenn ein Parent und all seine Nachfahren die Größe von 100.000 vbytes übersteigen. Beim CPFP-Carve-out kann einem Paket zusätzlich eine einzelne Transaktion mit einer Größe von bis zu 1.000 vbytes hinzugefügt werden (selbst wenn alle anderen Grenzen überschritten wurden), solange es sich um eine direkte Child-Transaktion einer unbestätigten Transaktion ohne unbestätigte Nachfolger handelt.

Beispielsweise signieren Bob und Mallory beide eine Transaktion mit zwei Outputs (einen für jeden von ihnen). Mallory veröffentlicht diese Transaktion und nutzt ihren Output, um entweder 25 Child-Transaktionen oder eine kleinere Zahl von Transaktionen mit einer Größe von 100.000 vbytes zu erzeugen. Ohne Carve-out wäre Bob nicht in der Lage, das CPFP-Fee-Bumping mit einer weiteren Child-Transaktion anzustoßen. Mittels Carve-out kann er einen der beiden Outputs der Transaktion (der ihm gehört) einlösen, solange die Child-Transaktion kleiner als 1.000 vbytes ist (was mehr als ausreichend ist).

Ein CPFP-Carve-out darf nur einmal genutzt werden, deshalb funktioniert es nur bei Zwei-Parteien-Protokollen. Es gab Vorschläge, das auf Protokolle mit mehreren Teilnehmenden auszuweiten, doch der Bedarf dafür ist gering, und die Entwicklerinnen und Entwickler konzentrieren sich lieber auf eine allgemeine Lösung für Transaktions-Pinning-Angriffe.

Während wir dies schreiben, nutzen die meisten beliebten LN-Implementierungen eine Transaktionsvorlage namens *Anker-Outputs* (engl. *Anchor Outputs*), die für den CPFP-Carve-out entwickelt wurde.

Gebühren in Transaktionen einfügen

Die Datenstruktur für Transaktionen kennt kein Feld für Gebühren. Sie werden vielmehr implizit als die Differenz der Summe von Inputs und Outputs aufgefasst. Jeder Betrag, der nach Abzug aller Outputs von allen Inputs übrig bleibt, ist die Gebühr, die der Miner einbehält:

$$Gebühren = Summe(Inputs) - Summe(Outputs)$$

Dieser Aspekt bei Transaktionen verwirrt ein wenig, und es ist wichtig, dass Sie ihn verstehen, da Sie beim Aufbau eigener Transaktionen sicherstellen müssen, nicht versehentlich zu viele Gebühren zu bezahlen, weil Sie die Inputs falsch angesetzt haben. Sie müssen also alle Inputs berücksichtigen und bei Bedarf Wechselgeld erzeugen, sonst geben Sie den Minern ein sehr hohes Trinkgeld!

Wenn Sie beispielsweise einen 20-Bitcoin-UTXO für eine Zahlung von einem Bitcoin verwenden, muss ein Wechselgeld von 19 Bitcoin an Ihre Wallet zurückgehen. Ansonsten gilt das »Überbleibsel« von 19 Bitcoin als Transaktionsgebühr und wird von dem Miner eingesackt, der Ihre Transaktion in einen Block schreibt. Zwar ist die Verarbeitungspriorität sehr hoch, und Sie machen einen Miner sehr glücklich, in Ihrem Sinne ist das aber sicher nicht.

Wenn Sie vergessen, den Change-Output (also das »Wechselgeld«) in Ihre händisch erzeugte Transaktion einzutragen, zahlen Sie dieses Wechselgeld als Transaktionsgebühr. »Der Rest ist für Sie« ist aber sicher nicht ganz das, was Sie beabsichtigt haben.

Timelock-Schutz gegen Fee-Sniping

Fee-Sniping, zu Deutsch etwa »Gebühren schneiden«, ist ein theoretisches Angriffsszenario, bei dem Miner versuchen, zurückliegende Blöcke neu zu schreiben und Transaktionen mit höheren Gebühren aus zukünftigen Blöcken einzufügen, um die Profitabilität zu steigern.

Nehmen wir zum Beispiel an, der höchste existierende Block sei #100000. Statt nun Block #100001 zu minen, um die Kette zu erweitern, könnten einige Miner versuchen, Block #100000 erneut zu schürfen. Diese Miner könnten jede gültige Transaktion (die noch nicht gemint wurde) in ihren Kandidatenblock #100000 aufnehmen. Sie müssen den Block nicht mit den gleichen Transaktionen erneut schürfen. Vielmehr versuchen sie, die profitabelsten Transaktionen (mit den höchsten Gebühren pro Kilobyte) in ihren Block aufzunehmen. Sie können jegliche Transaktion aus dem »alten« Block #100000 aufnehmen sowie jede Transaktion aus dem aktuellen Mempool. Prinzipiell haben sie also die Möglichkeit, Transaktionen aus der »Gegenwart« in die »Vergangenheit« zu schreiben, indem sie den Block #100000 neu schreiben.

Heutzutage ist dieser Angriff nicht sehr lukrativ, weil die Belohnung für den Block wesentlich höher ist als die Gebühren. Doch irgendwann in der Zukunft werden Transaktionsgebühren den Großteil der Belohnung (oder sogar die einzige Belohnung) darstellen. Zu diesem Zeitpunkt wird das Szenario unausweichlich.

Um dieses »Fee-Sniping« zu unterbinden, erzeugen Wallets Transaktionen mit einer Locktime, die sie standardmäßig auf den nächsten oder einen späteren Block beschränkt. In unserem Beispiel würde die Wallet die Locktime bei jeder erzeugten Transaktion auf 100001 setzen. Unter normalen Bedingungen hat diese Locktime keinen Einfluss, da die Transaktionen sowieso nur in Block #100001, also den nächsten Block, aufgenommen werden.

Bei einem Reorganisationsangriff wären die Miner nun aber nicht mehr in der Lage, die Transaktionen mit hohen Gebühren aus dem Mempool zu ziehen, weil diese Transaktionen über den Timelock an Block #100001 gekoppelt sind. Sie könnten nur Block #100000 mit den zu dieser Zeit gültigen Gebühren neu minen, ohne zusätzliche Gebühren abgreifen zu können.

Dies verhindert das Fee-Sniping nicht vollständig, aber es macht es in einigen Fällen weniger profitabel und kann helfen, die Stabilität des Bitcoin-Netzwerks zu erhalten, wenn die Blockbelohnung abnimmt. Wir empfehlen allen Wallets, Maßnahmen gegen das Fee-Sniping zu implementieren, wenn es anderen Anwendungsfällen des Locktime-Felds nicht widerspricht.

Während Bitcoin weiter reift und die Belohnung kontinuierlich sinkt, werden Gebühren für Bitcoin-Nutzer immer wichtiger. Das gilt für den täglichen Einsatz, um Transaktionen schnell zu bestätigen, aber auch um für die Miner einen Anreiz zu schaffen, Bitcoin-Transaktionen auch weiterhin durch neuen Proof-of-Work abzusichern.

Inhalt