La red Lightning de Bitcoin:
Pagos instantáneos escalables fuera de cadena

Joseph Poon joseph@lightning.network

Thaddeus Dryja rx@awsomnet.org

14 de enero de 2016
BORRADOR Versión 0.5.9.2

Resumen

El protocolo de Bitcoin puede abarcar el volumen global de transacciones financieras de todos los sistemas de pago electrónico actuales, sin que un único tercero custodio retenga fondos ni se requiera que los participantes dispongan de nada más que un ordenador con conexión de banda ancha. Se propone un sistema descentralizado en el que las transacciones se envían a través de una red de canales de micropagos (también conocidos como canales de pago o canales de transacción) cuya transferencia de valor se produce fuera de la cadena de bloques. Si las transacciones de Bitcoin pueden firmarse con un nuevo tipo de sighash que solucione la maleabilidad, estas transferencias pueden realizarse entre partes no confiables a lo largo de la ruta de transferencia mediante contratos que, en caso de participantes no cooperativos u hostiles, son ejecutables mediante difusión a través de la cadena de bloques de Bitcoin, mediante una serie de bloqueos temporales decrecientes.

1 El problema de escalabilidad de la cadena de bloques de Bitcoin

La cadena de bloques de Bitcoin[1] es muy prometedora para los libros de contabilidad distribuidos, pero la cadena de bloques como plataforma de pago, por sí sola, no puede abarcar el comercio mundial en un futuro próximo. La cadena de bloques es un protocolo de chisme mediante el cual todas las modificaciones del estado del libro mayor se transmiten a todos los participantes. Es a través de este «protocolo de chisme» como se alcanza el consenso sobre el estado, es decir, los saldos de todos. Si cada nodo de la red de Bitcoin debe conocer cada una de las

transacciones que se producen a nivel mundial, eso podría

suponer un lastre significativo para la capacidad de la red de abarcar todas las transacciones financieras globales. En cambio, sería deseable abarcar todas las transacciones de una manera que no sacrifique la descentralización y la seguridad que ofrece la red.

La red de pagos Visa alcanzó un pico de 47 000 transacciones por segundo (tps) en su red durante las vacaciones de 2013[2], y actualmente registra un promedio de cientos de millones al día. En la actualidad, Bitcoin admite menos de 7 transacciones por segundo con un límite de bloque de 1 megabyte. Si utilizamos una media de 300 bytes por transacción de Bitcoin y suponemos tamaños de bloque ilimitados, una capacidad equivalente al volumen máximo de transacciones de Visa de 47 000 tps sería de casi 8 gigabytes por bloque de Bitcoin, cada diez minutos de media. De forma continua, eso supondría más de 400 terabytes de datos al año.

Es evidente que alcanzar una capacidad similar a la de Visa en la red de Bitcoin no es factible hoy en día. Ningún ordenador doméstico del mundo puede funcionar con ese ancho de banda y esa capacidad de almacenamiento. Si Bitcoin fuera a sustituir a todos los pagos electrónicos en el futuro, y no solo a Visa, ello provocaría el colapso total de la red de Bitcoin o, en el mejor de los casos, una centralización extrema de los nodos y mineros de Bitcoin en manos de los únicos que podrían permitírselo. Esta centralización acabaría entonces con aspectos de la descentralización de la red que hacen que Bitcoin sea seguro, ya que la capacidad de las entidades para validar la cadena es lo que permite a Bitcoin garantizar la precisión y la seguridad del libro mayor.

Contar con menos validadores debido a bloques más grandes no solo implica que haya menos personas garantizando la exactitud del libro mayor, sino que también da lugar a que haya menos entidades capaces de validar la cadena de bloques como parte del proceso de minería, lo que fomenta la centralización de los mineros. Los bloques extremadamente grandes, por ejemplo, en el caso anterior de 8 gigabytes cada 10 minutos de media, implicarían que solo unas pocas partes podrían realizar la validación de bloques. Esto crea una gran posibilidad de que las entidades acaben confiando en partes centralizadas. Contar con partes privilegiadas y de confianza crea una trampa social en la que la parte central no actuará en interés de un individuo (problema principal-agente), por ejemplo, el rentismo mediante el cobro de comisiones más elevadas para mitigar el incentivo de actuar de forma deshonesta. En casos extremos, esto se manifiesta en que los individuos envían fondos a custodios centralizados de confianza que tienen la custodia total de los fondos de los clientes. Este tipo de acuerdos, tan comunes hoy en día, crean un grave riesgo de contraparte. Un requisito previo para evitar que se produzca ese tipo de centralización sería que el bitcoin pudiera ser validado por un único

ordenador de nivel doméstico a través de una conexión de banda ancha. Al garantizar que la validación completa pueda realizarse de forma económica, los nodos y mineros de Bitcoin podrán evitar la centralización y la dependencia extremas, lo que garantiza comisiones de transacción extremadamente bajas.

Si bien es posible que la Ley de Moore continúe indefinidamente y que en el futuro exista la capacidad computacional para que los nodos calculen de forma rentable bloques de varios gigabytes, no es una certeza.

Para alcanzar un número de transacciones muy superior a 47 000 por segundo utilizando Bitcoin, es necesario realizar las transacciones fuera de la propia cadena de bloques de Bitcoin. Sería aún mejor si la red de bitcoins admitiera un número casi ilimitado de transacciones por segundo con comisiones extremadamente bajas para los micropagos. Se pueden enviar muchos micropagos de forma secuencial entre dos partes para permitir pagos de cualquier cuantía. Los micropagos permitirían la desagregación, una menor necesidad de confianza y la mercantilización de los servicios, como los pagos por megabyte en el servicio de Internet. Sin embargo, para poder lograr estos casos de uso de micropagos, sería necesario reducir drásticamente la cantidad de transacciones que acaban transmitiéndose en la cadena de bloques global de Bitcoin.

Si bien es posible escalar a pequeña escala, es absolutamente imposible gestionar una gran cantidad de micropagos en la red o abarcar todas las transacciones globales. Para que Bitcoin tenga éxito, es necesario confiar en que, si llegara a ser extremadamente popular, sus ventajas actuales derivadas de la descentralización seguirán existiendo. Para que la gente de hoy crea que Bitcoin funcionará mañana, Bitcoin necesita resolver el problema de los efectos de centralización del tamaño de los bloques; los bloques grandes crean implícitamente custodios de confianza y comisiones significativamente más altas.

2 Una red de canales de micropagos puede resolver la escalabilidad

> «Si un árbol cae en el bosque y no hay nadie cerca para oírlo, ¿hace ruido?»

La cita anterior cuestiona la relevancia de los acontecimientos que pasan desapercibidos: si nadie oye caer el árbol, da igual si hizo ruido o no. Del mismo modo, en la cadena de bloques, si solo dos participantes se preocupan por una transacción cotidiana y recurrente, no es necesario que todos los demás nodos de la

red Bitcoin conozcan dicha transacción. En cambio, es preferible que solo haya la información mínima imprescindible en la cadena de bloques. Al posponer la notificación al mundo entero de cada transacción y realizar la liquidación neta de su relación en una fecha posterior, los usuarios de Bitcoin pueden llevar a cabo muchas transacciones sin sobrecargar la cadena de bloques ni crear confianza en una contraparte centralizada. Se puede lograr una estructura efectivamente sin confianza utilizando bloqueos temporales como componente del consenso global.

Actualmente, la solución para los micropagos y la escalabilidad consiste en delegar las transacciones a un custodio, con lo que se confía en terceros para que guarden las monedas y actualicen los saldos con otras partes. Confiar en terceros para que guarden todos los fondos genera riesgo de contraparte y costes de transacción.

En cambio, utilizando una red de estos canales de micropagos, Bitcoin puede escalar hasta miles de millones de transacciones al día con la potencia computacional disponible en un ordenador de sobremesa moderno hoy en día. Enviar muchos pagos dentro de un canal de micropagos determinado permite enviar grandes cantidades de fondos a otra parte de forma descentralizada. Estos canales no son una red de confianza separada sobre Bitcoin. Son transacciones reales de Bitcoin.

Los canales de micropagos[3][4] crean una relación entre dos partes para actualizar los saldos de forma continua, aplazando lo que se transmite a la cadena de bloques en una única transacción que compensa el saldo total entre esas dos partes. Esto permite que las relaciones financieras entre dos partes se aplacen sin necesidad de confianza hasta una fecha posterior, sin riesgo de incumplimiento de la contraparte. Los canales de micropagos utilizan transacciones reales de bitcoin, optando únicamente por aplazar la transmisión a la cadena de bloques de tal manera que ambas partes puedan garantizar su saldo actual en la cadena de bloques; no se trata de una red superpuesta de confianza: los pagos en los canales de micropagos son bitcoins reales que se comunican e intercambian fuera de la cadena.

2.1 Los canales de micropagos no requieren confianza

Al igual que la vieja pregunta de si un árbol que cae en el bosque hace ruido, si todas las partes están de acuerdo en que el árbol cayó a las 2:45 de la tarde, entonces el árbol realmente cayó a las 2:45 de la tarde. Del mismo modo, si ambas contrapartes están de acuerdo en que el saldo actual dentro de un canal es de 0,07 BTC para Alice y 0,03

BTC para Bob, entonces ese es el saldo real. Sin embargo, sin criptografía, se plantea un problema interesante: si la contraparte no está de acuerdo con el saldo actual de fondos (o con la hora a la que cayó el árbol), entonces es la palabra de uno contra la de otro. Sin firmas criptográficas, la cadena de bloques no sabrá quién es el propietario de qué.

Si el saldo en el canal es de 0,05 BTC para Alice y 0,05 BTC para Bob, y el saldo tras una transacción es de 0,07 BTC para Alice y 0,03 BTC para Bob, la red necesita saber qué conjunto de saldos es el correcto. Las transacciones de la cadena de bloques resuelven este problema utilizando el libro mayor de la cadena de bloques como un sistema de sellado de tiempo. Al mismo tiempo, es deseable crear un sistema que no utilice activamente este sistema de sellado de tiempo a menos que sea absolutamente necesario, ya que puede resultar costoso para la red.

En su lugar, ambas partes pueden comprometerse a firmar una transacción y a no difundirla. Así, si Alice y Bob depositan fondos en una dirección de multifirma 2 de 2 (en la que se requiere el consentimiento de ambas partes para realizar gastos), pueden ponerse de acuerdo sobre el estado actual del saldo. Alice y Bob pueden acordar crear un reembolso desde esa transacción 2 de 2 a sus propias cuentas, 0,05 BTC para cada uno. Este reembolso *no* se difunde en la cadena de bloques. Cualquiera de las partes
puede hacerlo, pero pueden optar por retener esa transacción, sabiendo
que pueden retirar los fondos cuando lo consideren oportuno. Al aplazar la difusión de esta transacción, pueden optar por modificar este saldo en una fecha futura.

Para actualizar el saldo, ambas partes crean un nuevo gasto desde la dirección de multifirma 2 de 2, por ejemplo, 0,07 para Alice y 0,03 para Bob. Sin un diseño adecuado, sin embargo, existe el problema del sellado de tiempo, ya que no se sabe qué gasto es el correcto: el nuevo gasto o el reembolso original.

La restricción sobre el sellado de tiempo y las fechas, sin embargo, no es tan compleja como el orden completo de todas las transacciones, como en la cadena de bloques de Bitcoin. En el caso de los canales de micropagos, solo se requieren dos estados: el saldo correcto actual y cualquier saldo antiguo obsoleto. Solo habría un único saldo correcto actual y, posiblemente, muchos saldos antiguos que están obsoletos.

Por lo tanto, en Bitcoin es posible diseñar un script de Bitcoin mediante el cual todas las transacciones antiguas se invaliden y solo sea válida la nueva transacción. La invalidación se aplica mediante un script de salida de Bitcoin y transacciones dependientes que obligan a la otra parte a entregar todos sus fondos a la contraparte del canal

contraparte. Al tomar todos los fondos como penalización para entregárselos a la otra parte, todas las transacciones antiguas quedan así invalidadas.

Este proceso de invalidación puede darse a través de un proceso de consenso de canal en el que, si ambas partes están de acuerdo con los estados actuales del libro mayor (y en la creación de nuevos estados), entonces se actualiza el saldo real. El saldo se refleja en la cadena de bloques solo cuando una de las partes no está de acuerdo. Conceptualmente, este sistema no es una red superpuesta independiente; se trata más bien de un aplazamiento del estado en el sistema actual, ya que la ejecución sigue teniendo lugar en la propia cadena de bloques (aunque se aplace a fechas y transacciones futuras).

2.2 Una red de canales

Por lo tanto, los canales de micropagos solo crean una relación entre dos partes. Exigir que todos creen canales con todos los demás no resuelve el problema de la escalabilidad. La escalabilidad de Bitcoin puede lograrse utilizando una gran red de canales de micropagos.

Si suponemos una gran red de canales en la cadena de bloques de Bitcoin, y todos los usuarios de Bitcoin participan en este grafo al tener al menos un canal abierto en la cadena de bloques de Bitcoin, es posible crear una cantidad casi infinita de transacciones dentro de esta red. Las únicas transacciones que se transmiten prematuramente en la cadena de bloques de Bitcoin son aquellas con contrapartes de canal no cooperativas.

Al gravar las salidas de las transacciones de Bitcoin con un hashlock y un timelock, la contraparte del canal no podrá robar fondos directamente y los bitcoins se podrán intercambiar sin que se produzca un robo directo por parte de la contraparte. Además, mediante el uso de tiempos de espera escalonados, es posible enviar fondos a través de múltiples intermediarios en una red sin el riesgo de que estos roben los fondos.

3 Canales de pago bidireccionales

Los canales de micropagos permiten un simple aplazamiento de la difusión del estado de una transacción a un momento posterior. Los contratos se hacen cumplir creando la responsabilidad de una de las partes de difundir las transacciones antes o después de determinadas fechas. Si la cadena de bloques es un sistema descentralizado de sellado de tiempo, es posible utilizar relojes como componente del consenso descentralizado[5] para determinar la validez de los datos, así como los estados actuales como método para ordenar los eventos[6].

Al crear marcos temporales en los que determinados estados pueden transmitirse y posteriormente invalidarse, es posible crear contratos complejos utilizando scripts de transacciones de bitcoin. Existen trabajos previos sobre canales de micropagos de tipo «hub-and-spoke»[7][8][9] (y redes de canales de pago de confianza[10][11]) que exploran la construcción de una red de este tipo en la actualidad. Sin embargo, el canal de micropagos bidireccional de Lightning Network requiere la bifurcación blanda de maleabilidad descrita en el Apéndice A para permitir una escalabilidad casi infinita, al tiempo que se mitigan los riesgos de incumplimiento de los nodos intermedios.

Al encadenar múltiples canales de micropagos, es posible crear una red de rutas de transacción. Las rutas pueden enrutarse utilizando un sistema similar al BGP, y el remitente puede designar una ruta concreta al destinatario. Los scripts de salida están protegidos por un hash, generado por el destinatario. Al revelar la entrada de ese hash, la contraparte del destinatario podrá retirar fondos a lo largo de la ruta.

3.1 El problema de la responsabilidad en la creación de canales

Para participar en esta red de pagos, es necesario crear un canal de micropagos con otro participante de la red.

3.1.1 Creando una transacción de financiación sin firmar

Se crea una transacción de financiación inicial del canal mediante la cual una o ambas contrapartes del canal financian las entradas de esta transacción. Ambas partes crean las entradas y salidas de esta transacción, pero no la firman. La salida de esta transacción de financiación es un único script de multisignatura 2-de-2

con ambos participantes en este canal, en lo sucesivo denominados Alice y Bob. Ninguno de los participantes intercambia firmas para la transacción de financiación hasta que hayan creado gastos a partir de esta salida 2 de 2, devolviendo el importe original a sus respectivos financiadores. El propósito de no firmar la transacción permite gastar de una transacción que aún no existe. Si Alice y Bob intercambian las firmas de la transacción de financiación sin poder difundir los gastos de la misma, los fondos pueden quedar bloqueados para siempre si Alice y Bob no cooperan (o puede producirse otra pérdida de monedas a través de situaciones de secuestro en las que uno paga por la cooperación de la contraparte).

Alice y Bob intercambian entradas para financiar la transacción de financiación

(para saber qué entradas se utilizan para determinar el valor total del canal), e intercambian una clave que utilizarán para firmar más adelante. Esta clave se utiliza para la salida 2 de 2 de la transacción de financiación; se necesitan ambas firmas para gastar desde la transacción de financiación; en otras palabras, tanto Alice como Bob deben estar de acuerdo en gastar desde la transacción de financiación.

3.1.2 Gasto de una transacción sin firmar

La Red Lightning utiliza una transacción SIGHASH NOINPUT para gastar desde esta salida de 2 de 2 de la transacción de financiación, ya que es necesario gastar desde una transacción para la que aún no se han intercambiado las firmas. SIGHASH NOINPUT, implementado mediante un soft fork, garantiza que las transacciones se puedan gastar antes de que sean firmadas por todas las partes, ya que las transacciones tendrían que firmarse para obtener un ID de transacción sin nuevas banderas sighash. Sin SIGHASH NOINPUT, las transacciones de Bitcoin no pueden gastarse antes de que puedan transmitirse; es como si no se pudiera redactar un contrato sin pagar primero a la otra parte. SIGHASH NOINPUT resuelve este problema. Véase el Apéndice A para obtener más información y detalles de implementación.

Sin SIGHASH NOINPUT, no es posible generar un gasto a partir de una transacción sin intercambiar firmas, ya que gastar la transacción de financiación requiere un ID de transacción como parte de la firma en la entrada de la transacción secundaria. Un componente del ID de transacción es la firma de la transacción principal (transacción de financiación), por lo que ambas partes deben intercambiar sus firmas de la transacción principal antes de que se pueda gastar la secundaria. Dado que una o ambas partes deben conocer las firmas de la transacción principal para gastar desde ella, eso significa que una o ambas partes pueden difundir la transacción principal (transacción de financiación) antes incluso de que exista la secundaria. SIGHASH NOINPUT soluciona esto permitiendo que la secundaria se gaste sin firmar la entrada. Con SIGHASH NOINPUT, el orden de las operaciones es el siguiente:

1. Crear el elemento principal (transacción de financiación)
2. Crea los elementos secundarios (operaciones de compromiso y todos los gastos derivados de las operaciones de compromiso)
3. Firmar los elementos secundarios
4. Intercambiar las firmas de los elementos secundarios

5. Firmar el elemento principal

6. Intercambiar las firmas del elemento principal

7. Difundir el elemento principal en la cadena de bloques

No es posible transmitir la transacción principal (paso 7) hasta que se haya completado el paso 6. Ninguna de las partes ha dado su firma para gastar desde la transacción de financiación hasta el paso 6. Además, si una de las partes falla durante el paso 6, la transacción principal puede gastarse para convertirse en la transacción principal o las entradas de la transacción principal pueden gastarse dos veces (de modo que toda esta ruta de transacción quede invalidada).

3.1.3 Transacciones de compromiso: estructura no ejecutable

Una vez creada la transacción de financiación sin firmar (y sin transmitir), ambas partes firman e intercambian una transacción de compromiso inicial. Estas transacciones de compromiso gastan desde la salida 2 de 2 de la transacción de financiación (padre). Sin embargo, solo la transacción de financiación se transmite a la cadena de bloques.

Dado que la transacción de financiación ya ha entrado en la cadena de bloques y la salida es una transacción de multifirma 2 de 2 que requiere el acuerdo de ambas partes para gastar, las transacciones de compromiso se utilizan para expresar el saldo actual. Si solo se intercambia una transacción de compromiso firmada 2-de-2 entre ambas partes, entonces ambas partes tendrán la certeza de que podrán recuperar su dinero una vez que la transacción de financiación entre en la cadena de bloques. Ninguna de las partes difunde las transacciones de compromiso en la cadena de bloques hasta que desean cerrar el saldo actual en el canal. Lo hacen difundiendo la transacción de compromiso actual.

Las transacciones de compromiso pagan los respectivos saldos actuales a cada parte. Una implementación ingenua (defectuosa) construiría una transacción no difundida en la que se produce un gasto 2-de-2 a partir de una única transacción que tiene dos salidas que devuelven todos los saldos actuales a ambas contrapartes del canal. Esto devolverá todos los fondos a la parte original al crear una transacción de compromiso inicial.

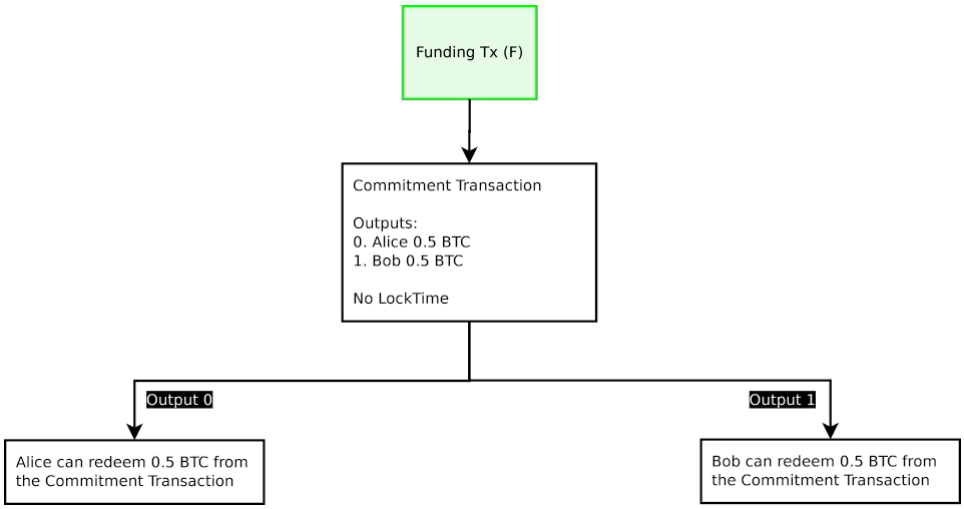

Figura 1: En este diagrama se describe una transacción de financiación ingenua y defectuosa. La transacción de financiación (F), designada en verde, se transmite en la cadena de bloques después de que se hayan firmado todas las demás transacciones. El resto de transacciones que gastan fondos de las transacciones de financiación aún no se han transmitido, por si las contrapartes desean actualizar su saldo. En este momento, solo se transmite la transacción de financiación en la cadena de bloques.

Por ejemplo, si Alice y Bob acuerdan crear una transacción de financiación con una única salida 2-de-2 por valor de 1,0 BTC (con una contribución de 0,5 BTC de cada uno), crean una transacción de compromiso en la que hay dos salidas de 0,5 BTC para Alice y Bob. Las transacciones de compromiso se firman primero y se intercambian las claves, de modo que cualquiera de las partes puede transmitir la transacción de compromiso en cualquier momento, siempre que la transacción de financiación se haya introducido en la cadena de bloques. En este punto, las firmas de la transacción de financiación pueden intercambiarse de forma segura, ya que cualquiera de las partes puede recuperar sus fondos transmitiendo la transacción de compromiso.

Sin embargo, esta estructura falla cuando se desea actualizar el saldo actual. Para actualizar el saldo, deben actualizar los valores de salida de su transacción de compromiso (la transacción de financiación ya ha entrado en la cadena de bloques y no se puede modificar).

Cuando ambas partes acuerdan una nueva transacción de compromiso e intercambian firmas para la nueva transacción de compromiso, cualquiera de las transacciones de compromiso puede transmitirse. Dado que la salida de la transacción de financiación solo puede canjearse una vez, solo una de esas transacciones será válida. Por ejemplo, si Alice y Bob acuerdan que el saldo del canal

es ahora de 0,4 para Alice y de 0,6 para Bob, y se crea una nueva transacción de compromiso para reflejarlo, cualquiera de las dos transacciones de compromiso puede transmitirse. En efecto, no sería posible restringir qué transacción de compromiso se transmite, ya que ambas partes han firmado e intercambiado las firmas para que se transmita cualquiera de los dos saldos.

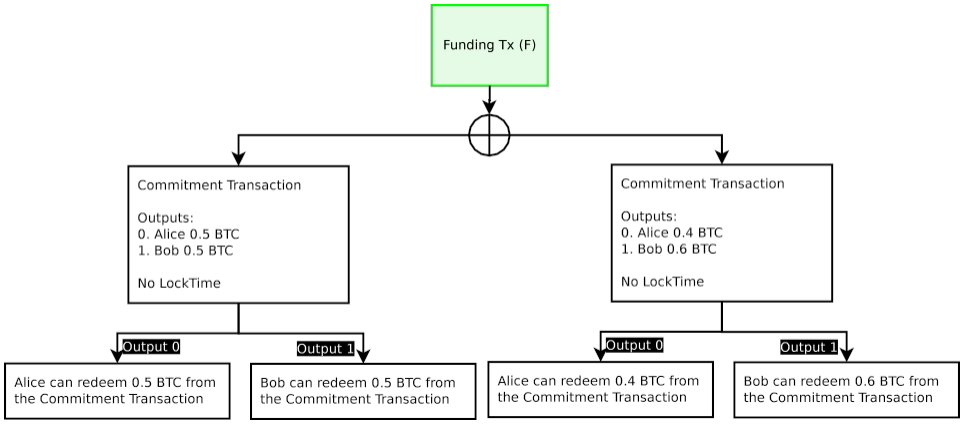

Figura 2: Cualquiera de las transacciones de compromiso puede ser difundida en cualquier momento por cualquiera de las partes, pero solo una gastará con éxito los fondos de la única transacción de financiación. Esto no puede funcionar porque una de las partes no querrá difundir la transacción más reciente.

Dado que cualquiera de las partes puede difundir la transacción de compromiso en cualquier momento, el resultado sería que, tras generarse la nueva transacción de compromiso, quien reciba menos fondos tendría un incentivo significativo para difundir la transacción que le reporta mayores beneficios en las salidas de la transacción de compromiso. Como resultado, el canal se cerraría inmediatamente y los fondos serían sustraídos. Por lo tanto, no es posible crear canales de pago bajo este modelo.

3.1.4 Transacciones de compromiso: atribución de culpa

Dado que cualquier transacción de compromiso firmada puede difundirse en la cadena de bloques, y solo una puede difundirse con éxito, es necesario evitar que se difundan transacciones de compromiso antiguas. No es posible revocar decenas de miles de transacciones en Bitcoin, por lo que se necesita un método alternativo. En lugar de una revocación activa impuesta por la cadena de bloques, es necesario construir el propio canal de manera similar a un seguro de fidelidad, en el que ambas partes asumen compromisos, y

el incumplimiento de estos compromisos se sanciona con penalizaciones. Si una de las partes incumple el acuerdo, perderá todo el dinero del canal.

Para este canal de pago, los términos del contrato establecen que ambas partes se comprometen a transmitir únicamente la transacción más reciente. Cualquier transmisión de transacciones anteriores supondrá un incumplimiento del contrato, y todos los fondos se entregarán a la otra parte como penalización.

Esto solo puede hacerse cumplir si se es capaz de atribuir la culpa de la transmisión de una transacción antigua. Para ello, debe ser posible identificar de forma única quién ha difundido una transacción anterior. Esto es posible si cada contraparte tiene una transacción de compromiso identificable de forma única. Ambas partes deben firmar las entradas de la transacción de compromiso que la otra parte es responsable de difundir. Dado que cada una tiene una versión de la transacción de compromiso firmada por la otra parte, solo puede difundirse la propia versión de la transacción de compromiso.

En la red Lightning, todos los gastos de la salida de la transacción de financiación, las transacciones de compromiso, tienen dos transacciones firmadas a medias. Una transacción de compromiso en la que Alice firma y se la entrega a Bob (C1b), y otra en la que Bob firma y se la entrega a Alice (C1a). Estas dos transacciones de compromiso se gastan desde la misma salida (transacción de financiación) y tienen contenidos diferentes; solo una puede difundirse en la cadena de bloques, ya que ambos pares de transacciones de compromiso se gastan desde la misma transacción de financiación. Cualquiera de las partes puede transmitir la transacción de compromiso que ha recibido firmando su versión e incluyendo la firma de la contraparte. Por ejemplo, Bob puede transmitir el compromiso C1b, ya que ha recibido la firma de C1b de Alice; incluye la firma de Alice y firma C1b él mismo. La transacción será un gasto válido de la salida 2 de 2 de la transacción de financiación, que requiere tanto la firma de Alice como la de Bob.

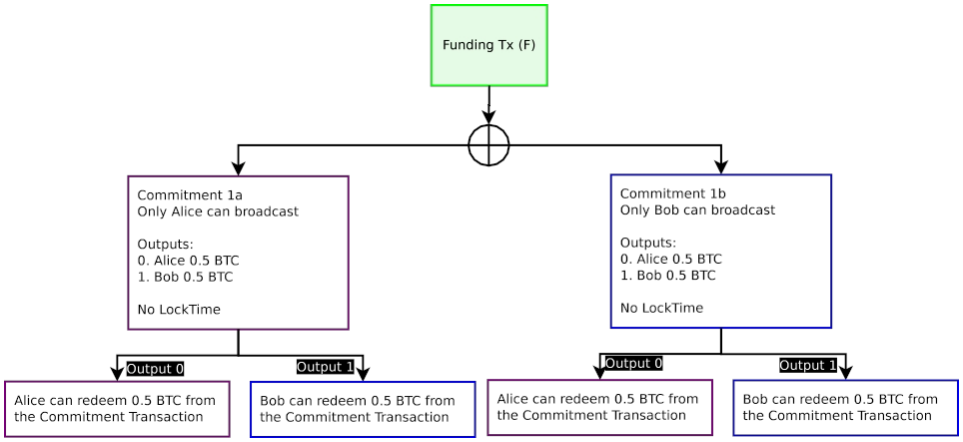

Figura 3: Los recuadros morados son transacciones no difundidas que solo Alice puede difundir. Los recuadros azules son transacciones no difundidas que solo Bob puede difundir. Alice solo puede difundir el compromiso 1a, Bob solo puede difundir el compromiso 1b. Solo se puede gastar una transacción de compromiso de la salida de la transacción de financiación. Se atribuye la culpa, pero cualquiera de las dos puede gastarse sin penalización.

Sin embargo, incluso con esta estructura, solo se ha asignado la responsabilidad. Todavía no es posible hacer cumplir este contrato en la cadena de bloques de Bitcoin. Bob sigue confiando en que Alice no transmitirá una transacción de compromiso antigua. En este momento, solo puede demostrar que Alice lo ha hecho mediante una prueba de transacción firmada a medias.

3.2 Creación de un canal con revocación del contrato

Para poder hacer cumplir realmente los términos del contrato, es necesario construir una transacción de compromiso (junto con sus gastos) en la que se pueda revocar una transacción. Esta revocación se consigue utilizando datos sobre cuándo una transacción entra en una cadena de bloques y utilizando la madurez de la transacción para determinar las rutas de validación.

3.3 Madurez de los números de secuencia

Mark Freidenbach ha propuesto que los números de secuencia puedan hacerse obligatorios mediante un vencimiento relativo del bloque de la transacción principal a través de un soft fork[12]. Esto permitiría una capacidad básica para garantizar algún tipo de bloqueo temporal relativo de la confirmación del bloque en el script de gasto. Además,

, un código de operación adicional, OP CHECKSEQUENCEVERIFY[13] (también conocido como OP RELATIVECHECKLOCKTIMEVERIFY)[14], permitiría capacidades adicionales, incluyendo una solución provisional antes de una solución más permanente para resolver la maleabilidad de las transacciones. Una versión futura de este artículo incluirá las soluciones propuestas.

En resumen, Bitcoin se lanzó con un número de secuencia que solo se aplicaba en el mempool de transacciones no confirmadas. El comportamiento original permitía la sustitución de transacciones al reemplazar las transacciones del mempool por otras más recientes si estas tenían un número de secuencia más alto. Debido a las reglas de sustitución de transacciones, esto no se aplica por los riesgos de ataques de denegación de servicio. Parece que el propósito previsto del número de secuencia es sustituir las transacciones no difundidas. Sin embargo, este comportamiento de sustitución por un número de secuencia más alto es inaplicable. No se puede garantizar que las versiones antiguas de las transacciones hayan sido sustituidas en el mempool y que un bloque contenga la versión más reciente de la transacción. Una forma de hacer cumplir las versiones de las transacciones fuera de la cadena es mediante compromisos de tiempo.

Una transacción revocable gasta desde una salida única en la que la transacción tiene un tipo único de script de salida. Esta salida de la transacción principal tiene dos rutas de canje: la primera puede canjearse inmediatamente, y la segunda solo puede canjearse si la transacción secundaria tiene un número mínimo de confirmaciones entre transacciones. Esto se consigue haciendo que el número de secuencia de la transacción secundaria requiera un número mínimo de confirmaciones de la transacción principal. En esencia, este nuevo comportamiento del número de secuencia solo permitirá que un gasto desde esta salida sea válido si el número de bloques entre la salida y la transacción de canje es superior a una altura de bloque especificada.

Una transacción puede revocarse con este comportamiento del número de secuencia creando una restricción con un número definido de bloques en el número de secuencia, lo que hará que el gasto solo sea válido después de que la transacción principal haya entrado en la cadena de bloques durante un número definido de bloques. Esto crea una estructura en la que la transacción principal con esta salida se convierte en un depósito garantizado, lo que certifica que no hay revocación. Existe un periodo de tiempo durante el cual cualquier persona en la
cadena de bloques puede refutar esta certificación emitiendo un gasto inmediatamente después de que se emita la transacción.

Si se desea permitir transacciones revocables con un retraso de 1000 confirmaciones, la construcción de la transacción de salida seguiría siendo una multisig 2 de 2:

2 <Alice 1 > <Bob1> 2 OP CHECKMULTISIG

Sin embargo, la transacción de gasto secundaria contendría un valor nSequence de 1000. Dado que esta transacción requiere la firma de ambas contrapartes para ser válida, ambas partes incluyen el número nSequence de 1000 como parte de la firma. Ambas partes pueden, a su discreción, acordar crear otra transacción que sustituya a esa transacción sin ningún número nSequence.

Esta construcción, un contrato de vencimiento de secuencia revocable (RSMC), crea dos vías, con condiciones contractuales muy específicas.

Las condiciones del contrato son:

1. Todas las partes realizan un pago a un contrato con una salida que hace cumplir este contrato

2. Ambas partes pueden acordar enviar fondos a algún contrato, con un periodo de espera (1000 confirmaciones en nuestro guion de ejemplo). Este es el saldo de salida revocable.

3. Una o ambas partes pueden optar por no difundir (ejecutar) los pagos hasta una fecha futura; cualquiera de las partes puede canjear los fondos en cualquier momento tras el periodo de espera.

4. Si ninguna de las partes ha difundido esta transacción (canjeado los fondos), pueden revocar los pagos anteriores si, y solo si, ambas partes acuerdan hacerlo incluyendo una nueva cláusula de pago en una transacción de pago sustitutiva. El nuevo pago de la transacción puede canjearse inmediatamente después de que el contrato se haga público (se difunda en la cadena de bloques).

5. En caso de que el contrato se divulgue y no se canjee la nueva estructura de pago, cualquiera de las partes podrá canjear los términos de pago revocados anteriormente (por lo que es responsabilidad de cualquiera de las partes hacer cumplir los nuevos términos).

La transacción secundaria pre-firmada puede canjearse una vez que la transacción principal haya entrado en la cadena de bloques con 1000 confirmaciones, debido a que el número nSequence de la secundaria en la entrada gasta la principal.

Para revocar esta transacción secundaria firmada, ambas partes solo tienen que acordar crear otra transacción secundaria con el campo predeterminado del número nSequence de MAX INT, que tiene un comportamiento especial que permite el gasto en cualquier momento.

Este nuevo gasto firmado sustituye al gasto revocable siempre y cuando el nuevo gasto firmado se incorpore a la cadena de bloques en un plazo de 1000 confirmaciones desde que la transacción principal se incorporó a la cadena de bloques. En la práctica, si Alice y Bob acuerdan supervisar la cadena de bloques para detectar una difusión incorrecta de las transacciones de compromiso, en el momento en que se difunde la transacción, pueden gastar inmediatamente utilizando la transacción sustitutiva. Para transmitir el gasto revocable (transacción obsoleta), que se gasta desde la misma salida que la transacción sustitutiva, deben esperar 1000 confirmaciones. Siempre que ambas partes vigilen la cadena de bloques, el gasto revocable nunca entrará en la transacción si cualquiera de las partes prefiere la transacción sustitutiva.

Utilizando esta estructura, cualquiera podría crear una transacción, no transmitirla difundir la transacción y, posteriormente, crear incentivos para que esa transacción nunca se difunda en el futuro mediante sanciones. Esto permite a los participantes de la red Bitcoin aplazar muchas transacciones para que nunca lleguen a la cadena de bloques.

3.3.1 Timestop

Para mitigar una avalancha de transacciones por parte de un atacante malintencionado se requiere una amenaza creíble de que el ataque fracasará.

Greg Maxwell propuso utilizar un «timestop» para mitigar una avalancha maliciosa en la cadena de bloques:

> Hay muchas formas de abordar este [riesgo de avalancha] que aún no se han explorado adecuadamente; por ejemplo, el reloj puede detenerse cuando los bloques estén llenos, convirtiendo el riesgo de seguridad en un mayor retraso de retención en caso de un ataque DoS.[15]

Esto puede mitigarse permitiendo que el minero especifique si el mempool actual (con tarifa pagada) se está viendo inundado de transacciones. Pueden introducir un valor «1» en el último bit del número de versión del encabezado del bloque. Si el último bit del encabezado del bloque contiene un «1», entonces ese bloque no contará para la madurez de altura relativa del valor nSequence y el bloque se designará como un bloque congestionado. Existe una altura de bloque no congestionado (que siempre es inferior a la altura de bloque normal). Esta altura de bloque se utiliza para el valor nSequence, que solo cuenta la madurez del bloque (confirmaciones).

Un minero puede optar por definir el bloque como congestionado o no. El código predeterminado podría establecer automáticamente el indicador de bloque congestionado como «1» si el

mempool supera un determinado tamaño y la tarifa media para ese tamaño establecido supera un determinado valor. Sin embargo, un minero tiene plena discreción para cambiar las reglas sobre lo que se establece automáticamente como un bloque congestionado, o puede optar por activar o desactivar permanentemente el indicador de congestión. Se espera que la mayoría de los mineros honestos utilicen el comportamiento predeterminado definido en su minero y no organicen un ataque del 51 %.

Por ejemplo, si la salida de una transacción principal es gastada por una secundaria con un valor de nSequence de 10, hay que esperar 10 confirmaciones antes de que la transacción sea válida. Sin embargo, si se ha activado el indicador de detención de tiempo, el recuento de confirmaciones se detiene, incluso con nuevos bloques. Si han transcurrido 6 confirmaciones (se necesitan 4 más para que la transacción sea válida) y el bloque de timestop se ha establecido en el séptimo bloque, ese bloque no cuenta para el requisito de nSequence de 10 confirmaciones; la transacción secundaria sigue estando en 6 bloques para el valor de confirmación relativo. Funcionalmente, esto se almacenará como una especie de altura de bloque de timestop auxiliar que se utiliza únicamente para el seguimiento del valor de timestop. Cuando se activa el bit de timestop, todas las transacciones que utilicen un valor de nSequence dejarán de contar hasta que se desactive el bit de timestop. Esto proporciona tiempo y espacio de bloque suficientes para que las transacciones en la altura actual del bloque de timestop auxiliar entren en la cadena de bloques, lo que puede impedir que los atacantes sistémicos logren atacar el sistema.

Sin embargo, esto requiere algún tipo de indicador en el bloque para designar si se trata de un bloque de parada de tiempo. Para una compatibilidad total con SPV (Verificación de Pago Simple; clientes ligeros), es deseable que esto se encuentre dentro del encabezado del bloque de 80 bytes en lugar de en la coinbase. Hay dos lugares que podrían ser adecuados para colocar este indicador en el encabezado del bloque: en la hora del bloque y en la versión del bloque. La hora del bloque podría no ser segura debido a que los últimos bits se utilizan como fuente de entropía para algunos mineros ASIC, por lo que podría ser necesario consumir un bit para los indicadores de timestop. Otra opción sería codificar de forma rígida la activación de timestop como una regla de consenso estricta (por ejemplo, a través del tamaño del bloque), aunque esto podría reducir la flexibilidad. Al establecer valores predeterminados razonables para las reglas de timestop, estas reglas pueden modificarse sin necesidad de bifurcaciones suaves de consenso.

Si se utiliza la versión del bloque como indicador, la información contextual debe coincidir con el ID de cadena utilizado en algunas monedas de minería combinada.

3.3.2 Transacciones de compromiso revocables

Al combinar la atribución de culpa con la transacción revocable, se puede determinar cuándo una parte no está cumpliendo los términos del contrato y aplicar sanciones sin necesidad de confiar en la contraparte.

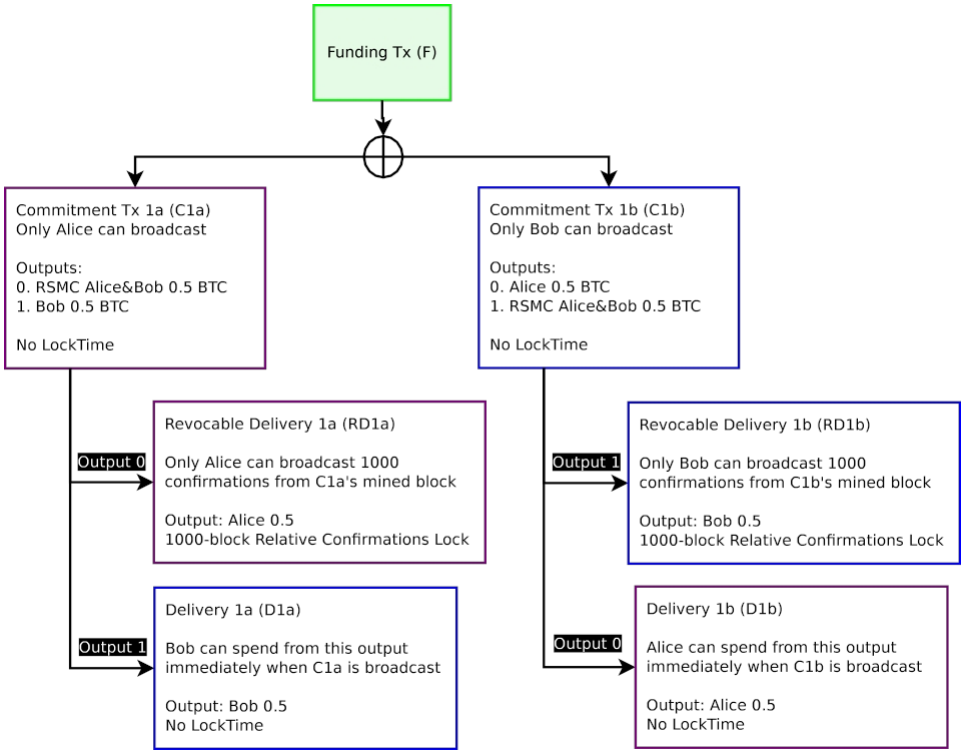

Figura 4: La transacción de financiación F, señalada en verde, se transmite a la cadena de bloques después de que se hayan firmado todas las demás transacciones. Todas las transacciones que solo Alice puede transmitir están en morado. Todas las transacciones que solo Bob puede transmitir están en azul. En este momento, solo se transmite la transacción de financiación a la cadena de bloques.

La intención de crear una nueva transacción de compromiso es invalidar todas las transacciones de compromiso antiguas al actualizar el nuevo saldo con una nueva transacción de compromiso. La invalidación de transacciones antiguas puede realizarse haciendo que una salida sea un contrato de vencimiento de secuencia revocable (RSMC). Para invalidar una transacción, ambas partes firmarán e intercambiarán una transacción sustitutiva que ceda todos los fondos a la contraparte en caso de que se transmita incorrectamente una transacción anterior. La transmisión incorrecta

se identifica creando dos transacciones de compromiso diferentes con las mismas salidas de saldo final; sin embargo, el pago a uno mismo queda gravado por un RSMC.

En efecto, hay dos transacciones de compromiso a partir de una única transacción de financiación con salidas 2 de 2. De estas dos transacciones de compromiso, solo una puede entrar en la cadena de bloques. Cada parte dentro de un canal tiene una versión de este contrato. Así que, si este es el primer par de transacciones de compromiso, la transacción de compromiso de Alice se define como C1a, y la transacción de compromiso de Bob se define como C1b. Al transmitir una transacción de compromiso, se solicita que el canal se cierre y finalice. Las dos primeras salidas de la transacción de compromiso incluyen una transacción de entrega (pago) del saldo actual no asignado a las contrapartes del canal. Si Alice difunde C1a, una de las salidas puede ser gastada por D1a, que envía fondos a Bob. Para Bob, C1b puede ser gastada por D1b, que envía fondos a Alice. La transacción de entrega (D1a/D1b) es inmediatamente canjeable y no está sujeta a ningún gravamen en caso de que se difunda la transacción de compromiso.

En la transacción de compromiso de cada parte, esta certifica que está transmitiendo la transacción de compromiso más reciente de la que es titular. Dado que certifican que este es el saldo actual, se supone que el saldo pagado a la contraparte es cierto, ya que nadie obtiene un beneficio directo al pagar fondos a la contraparte a modo de penalización.

Sin embargo, el saldo pagado a la persona que difunde la transacción de compromiso no está verificado. Los participantes en la cadena de bloques no tienen forma de saber si la transacción de compromiso es la más reciente o no. Si no transmiten su versión más reciente, serán penalizados con la retirada de todos los fondos del canal y su entrega a la contraparte. Dado que sus propios fondos están bloqueados en su propio RSMC, solo podrán reclamar sus fondos tras un número determinado de confirmaciones una vez que la transacción de compromiso haya sido incluida en un bloque (en nuestro ejemplo, 1000 confirmaciones). Si transmiten su transacción de compromiso más reciente, no debería haber ninguna transacción de revocación que sustituya a la transacción revocable, por lo que podrán recibir sus fondos tras un tiempo determinado (1000 confirmaciones).

Al saber quién transmitió la transacción de compromiso y al comprometer que los propios pagos queden bloqueados durante un período de tiempo predefinido,

ambas partes podrán revocar la transacción de compromiso en el futuro.

3.3.3 Cobro de fondos del canal: contrapartes cooperativas

Cualquiera de las partes puede rescatar los fondos del canal. Sin embargo, la parte que transmita la transacción de compromiso debe esperar el número predefinido de confirmaciones descrito en el RSMC. La contraparte que no haya transmitido la transacción de compromiso puede rescatar los fondos inmediatamente.

Por ejemplo, si la transacción de financiación se compromete con 1 BTC (la mitad para cada contraparte) y Bob transmite la transacción de compromiso más reciente, C1b, debe esperar 1000 confirmaciones para recibir sus 0,5 BTC, mientras que Alice puede gastar 0,5 BTC. Para Alice, esta transacción se da por completamente cerrada si Alice acepta que Bob ha difundido la transacción de compromiso correcta (C1b).

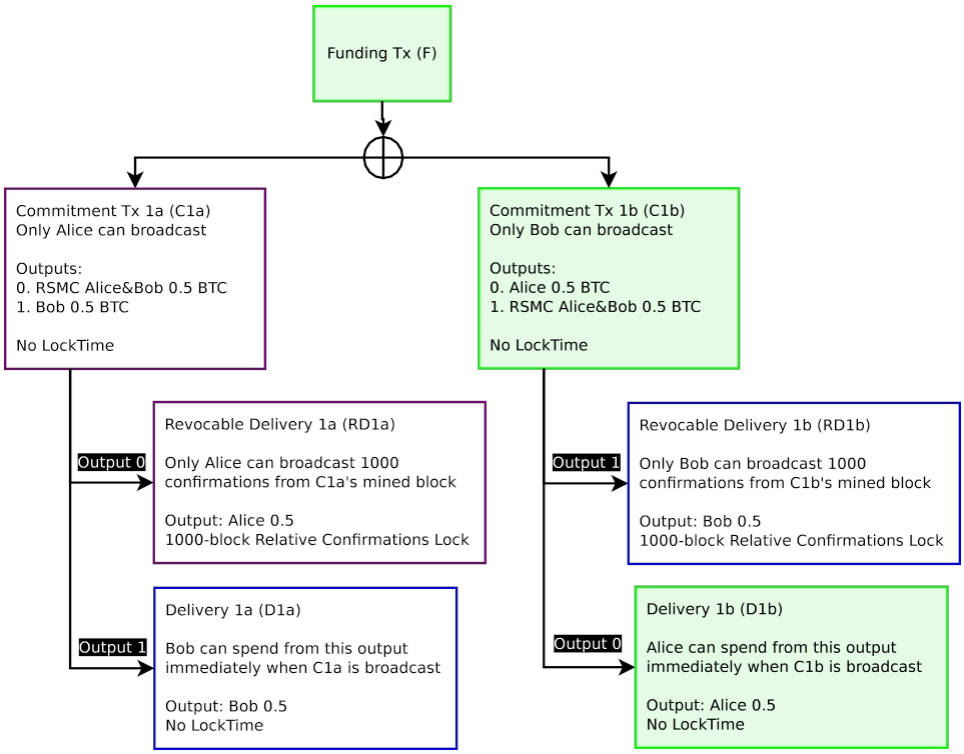

Figura 5: Cuando Bob transmite C1b, Alice puede canjear inmediatamente su parte. Bob debe esperar 1000 confirmaciones. Cuando el bloque se transmite inmediatamente, se encuentra en este estado. Las transacciones en verde son aquellas que se han comprometido en la cadena de bloques.

Una vez que la transacción de compromiso haya estado en la cadena de bloques durante 1000 bloques, Bob podrá entonces transmitir la transacción de entrega revocable. Debe esperar 1000 bloques para demostrar que no ha revocado esta transacción de compromiso (C1b). Tras 1000 bloques, la transacción de entrega revocable podrá incluirse en un bloque. Si una de las partes intenta incluir la transacción de entrega revocable en un bloque antes de que se hayan producido 1000 confirmaciones, la transacción será inválida hasta que hayan transcurrido 1000 confirmaciones (momento en el que se volverá válida si la salida aún no ha sido canjeada).

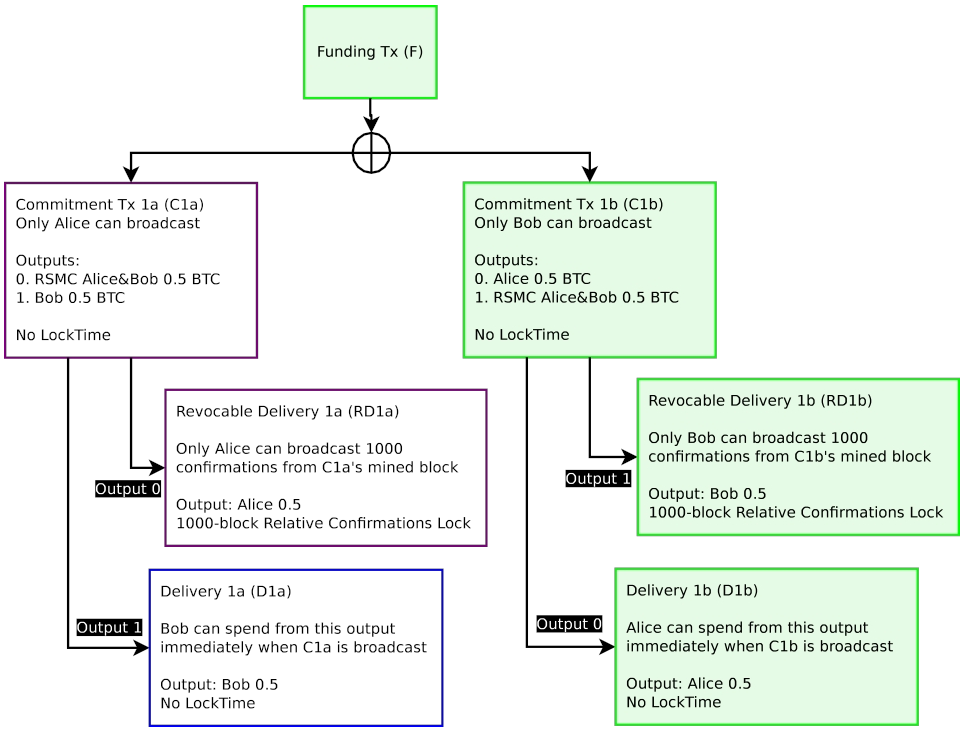

Figura 6: Alice acepta que Bob haya difundido la transacción de compromiso correcta y han transcurrido 1000 confirmaciones. Bob puede entonces difundir la transacción de entrega revocable (RD1b) en la cadena de bloques.

Una vez que Bob transmite la transacción de entrega revocable, el canal queda completamente cerrado tanto para Alice como para Bob; ambos han recibido los fondos que, según acuerdan, constituyen el saldo actual que cada uno posee en el canal.

Si, por el contrario, fuera Alice quien hubiera difundido la transacción de compromiso (C1a), sería ella quien tendría que esperar 1000 confirmaciones en lugar de Bob.

3.3.4 Creación de una nueva transacción de compromiso y revocación de compromisos anteriores

Aunque cada parte puede cerrar la transacción de compromiso más reciente en cualquier momento, también pueden optar por crear una nueva transacción de compromiso e invalidar la anterior.

Supongamos que Alice y Bob quieren ahora actualizar sus saldos actuales de 0,5 BTC cada uno a 0,6 BTC para Bob y 0,4 BTC para Alice.

Cuando ambos acuerdan hacerlo, generan un nuevo par de transacciones de compromiso.

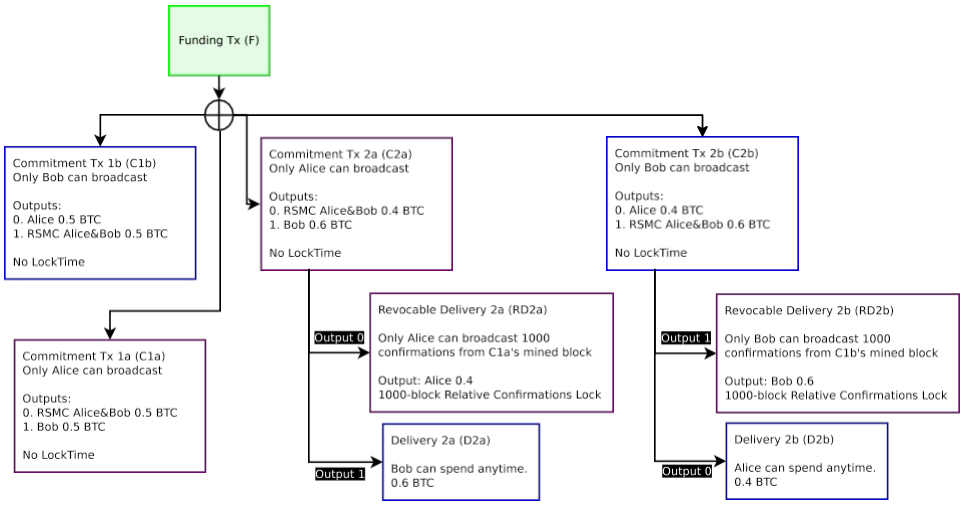

Figura 7: Pueden existir cuatro transacciones posibles: un par con los compromisos antiguos y otro par con los nuevos compromisos. Cada parte dentro del canal solo puede transmitir la mitad del total de compromisos (dos cada una). No existe ninguna medida explícita que impida la difusión de un compromiso concreto, salvo los gastos de penalización, ya que todos son gastos válidos no difundidos. El compromiso revocable sigue existiendo con el par C1a/C1b, pero no se muestra por brevedad.

Cuando se acuerda un nuevo par de transacciones de compromiso (C2a/C2b), ambas partes firmarán e intercambiarán firmas para la nueva transacción de compromiso, y luego invalidarán la antigua transacción de compromiso. Esta invalidación se produce cuando ambas partes firman una transacción de subsanación de incumplimiento (BR1), que sustituye a la transacción de entrega revocable (RD1). Cada parte entrega a la otra una revocación firmada a medias (BR1) de su propia entrega revocable (RD1), que es un gasto de la transacción de compromiso. La transacción de subsanación de incumplimiento enviará todas las monedas a la contraparte dentro del saldo actual del canal. Por ejemplo, si Alice y Bob generan ambos un nuevo par de transacciones de compromiso (C2a/C2b) e invalidan los compromisos anteriores (C1a/C1b), y más tarde Bob difunde incorrectamente C1b en la cadena de bloques, Alice puede quedarse con todo el dinero de Bob del canal. Alice puede hacer esto porque Bob le ha demostrado a Alice, mediante una penalización, que nunca transmitirá C1b, ya que en el momento en que transmita C1b, Alice podrá quedarse con todo el dinero de Bob en el canal. En efecto, al construir una

para la contraparte, se ha certificado que no se transmitirá ningún compromiso anterior. La contraparte puede aceptar esto, ya que obtendrá todo el dinero del canal cuando se incumpla este acuerdo.

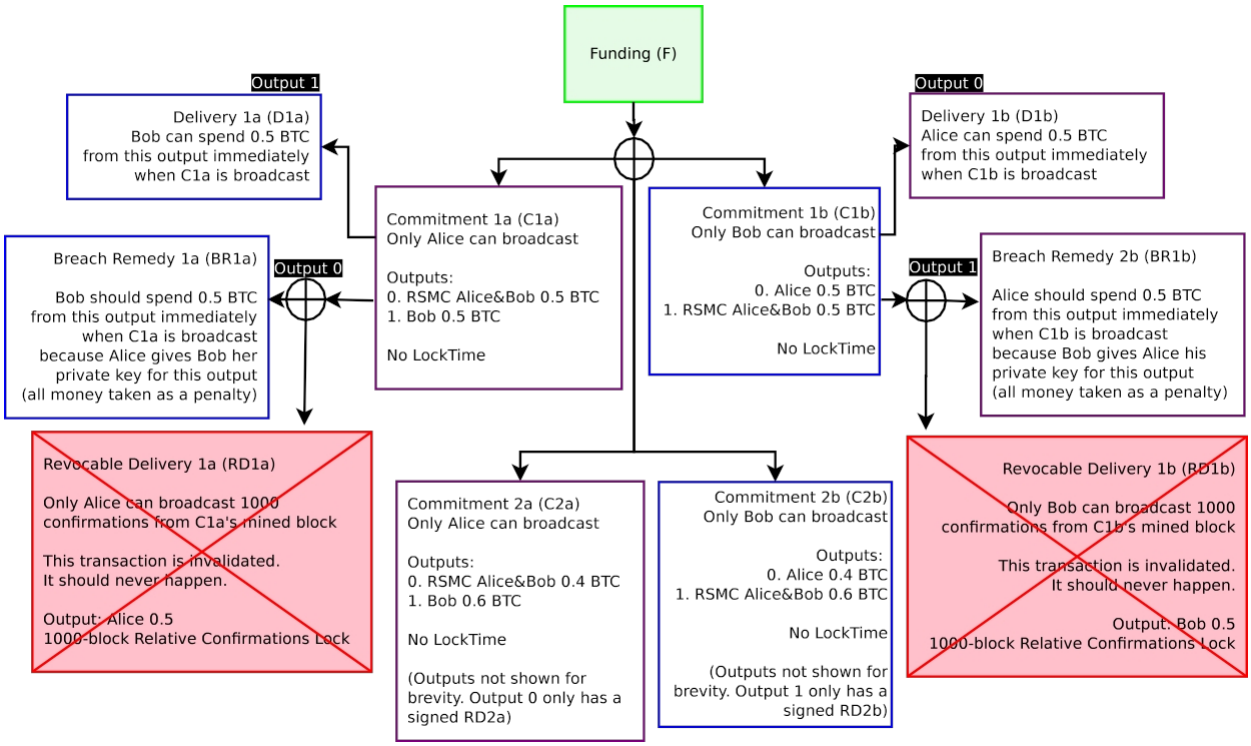

Figura 8: Cuando existen C2a y C2b, ambas partes intercambian transacciones de «Breach Remedy». Ambas partes tienen ahora un incentivo económico explícito para evitar transmitir antiguas transacciones de compromiso (C1a/C1b). Si cualquiera de las partes desea cerrar el canal, solo utilizará C2a (Alice) o C2b (Bob). Si Alice transmite C1a, todo su dinero irá a parar a Bob. Si Bob transmite C1b, todo su dinero irá a parar a Alice. Véase la figura anterior para las salidas de C2a/C2b.

Debido a esto, es probable que se eliminen todas las transacciones de compromiso anteriores cuando se haya enviado una transacción de subsanación de incumplimiento a la contraparte. Si se transmite una transacción de compromiso incorrecta (obsoleta e invalidada), todo el dinero irá a parar a la contraparte. Por ejemplo, si Bob transmite C1b, siempre que Alice observe la cadena de bloques dentro del número predefinido de bloques (en este caso, 1000 bloques), Alice podrá quedarse con todo el dinero de este canal transmitiendo RD1b. Incluso si el

saldo actual del estado de compromiso (C2a/C2b) sea de 0,4 BTC para Alice y 0,6 BTC para Bob, dado que Bob ha incumplido los términos del contrato, todo el dinero va a parar a Alice como penalización. Funcionalmente, la transacción revocable actúa como prueba ante la cadena de bloques de que Bob ha incumplido los términos del canal, y esto es resuelto programáticamente por la cadena de bloques.

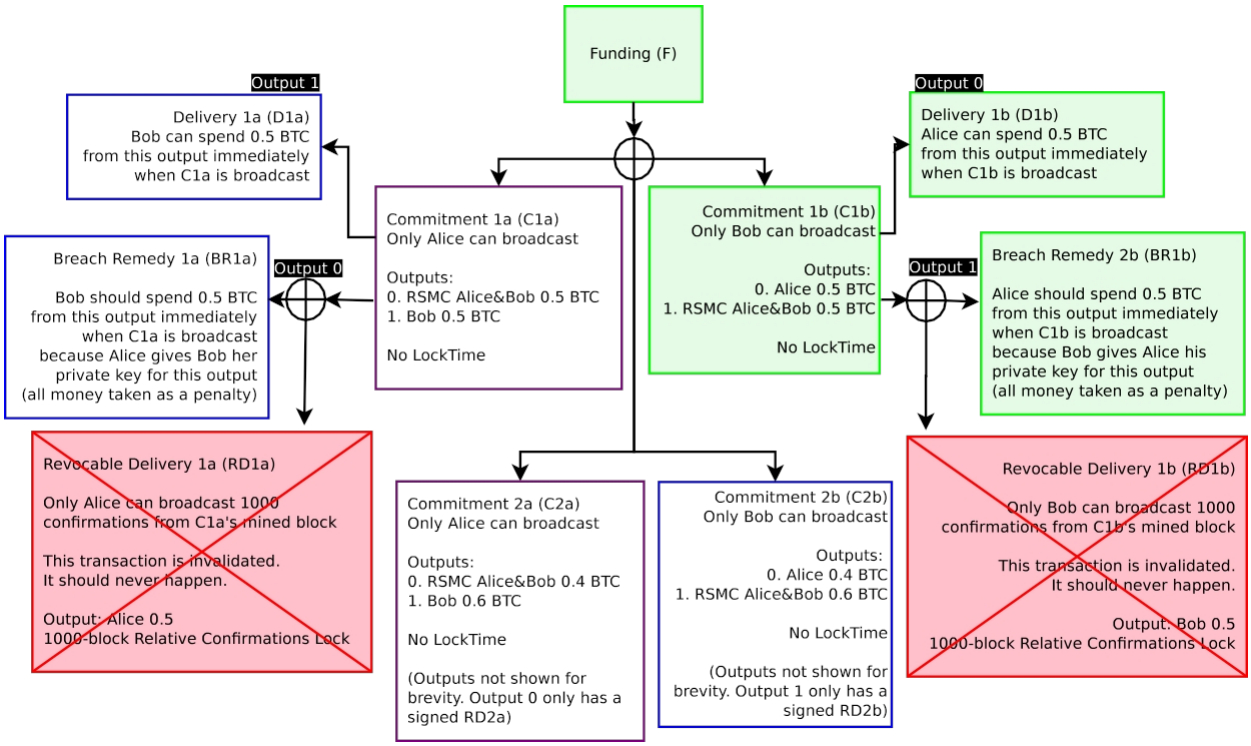

Figura 9: Las transacciones en verde se registran en la cadena de bloques. Bob transmite incorrectamente C1b (solo Bob puede transmitir C1b/C2b). Dado que ambos acordaron que el estado actual es el par de compromisos C2a/C2b, y han certificado ante cada parte que los compromisos anteriores quedan invalidados mediante transacciones de subsanación de incumplimiento, Alice puede transmitir BR1b y quedarse con todo el dinero del canal, siempre que lo haga en un plazo de 1000 bloques tras la transmisión de C1b.

Sin embargo, si Alice no transmite BR1b en un plazo de 1000 bloques, Bob podría robar parte del dinero, ya que su transacción de entrega revocable (RD1b) se vuelve válida tras 1000 bloques. Cuando se transmite una transacción de compromiso incorrecta, solo la transacción de subsanación de incumplimiento puede transmitirse durante 1000 bloques (o el número de confirmaciones que ambas

acuerden). Tras 1000 confirmaciones de bloques, tanto la transacción de subsanación del incumplimiento (BR1b) como la transacción de entrega revocable (RD1b) pueden emitirse en cualquier momento. Las transacciones de subsanación del incumplimiento solo tienen exclusividad dentro de este periodo de tiempo predefinido, y cualquier momento posterior a ese es, a efectos prácticos, la expiración del plazo de prescripción: según el consenso de la cadena de bloques de Bitcoin, el plazo para la disputa ha finalizado.

Por este motivo, se debe supervisar periódicamente la cadena de bloques para comprobar si la contraparte ha difundido una transacción de compromiso invalidada, o delegar en un tercero para que lo haga. Se puede delegar en un tercero simplemente cediéndole la transacción de subsanación de incumplimiento. Se les puede incentivar para que vigilen la cadena de bloques y difundan dicha transacción en caso de malicia de la contraparte, otorgándoles a estos terceros una comisión en la salida. Dado que el tercero solo puede actuar cuando la contraparte actúa de forma maliciosa, este tercero no tiene ningún poder para forzar el cierre del canal.

3.3.5 Proceso para crear transacciones de compromiso revocables

Para crear transacciones de compromiso revocables, es necesario construir adecuadamente el canal desde el principio y firmar únicamente transacciones que puedan transmitirse en cualquier momento en el futuro, al tiempo que se garantiza que no se sufrirán pérdidas debido a contrapartes poco cooperativas o maliciosas. Esto requiere determinar qué clave pública utilizar para los nuevos compromisos, ya que el uso de SIGHASH NOINPUT exige el uso de claves únicas para cada salida RSMC (y HTLC) de la transacción de compromiso. Utilizamos P para designar las claves públicas y K para designar la clave privada correspondiente utilizada para firmar.

Al generar la primera transacción de compromiso, Alice y Bob acuerdan crear una salida multisig a partir de una transacción de financiación con una única salida $multisig(P_{AliceF}, P_{BobF})$, financiada con 0,5 BTC de Alice y Bob para un total de 1 BTC. Esta salida es una transacción Pay to Script Hash[16], que requiere que tanto Alice como Bob acuerden gastar desde la transacción de financiación. Todavía no hacen que la transacción de financiación (F) sea gastable. Además, P_{AliceF} y P_{BobF} solo se utilizan para la transacción de financiación, no se utilizan para nada más.

Dado que la transacción de entrega es simplemente una salida P2PKH (direcciones de bitcoin que comienzan por 1) o una transacción P2SH (comúnmente reconocidas como direcciones que comienzan por 3) que las contrapartes designan de antemano,

Esto puede generarse como una salida de P_{AliceD}y P_{BobD}. Para simplificar, estas direcciones de salida permanecerán iguales a lo largo de todo el canal, ya que sus fondos quedan bajo el control total del destinatario designado una vez que la transacción de compromiso se registra en la cadena de bloques. Si se desea, aunque no es necesario, ambas partes pueden actualizar y modificar P_{AliceD}y P_{BobD}para futuras transacciones de compromiso.

Ambas partes intercambian las claves públicas que pretenden utilizar para el RSMC (y el HTLC descrito en secciones posteriores) para la transacción de compromiso. Cada conjunto de transacciones de compromiso utiliza sus propias claves públicas y nunca se reutilizan. Ambas partes pueden conocer ya todas las claves públicas futuras utilizando una construcción de monedero HD BIP 0032[17] mediante el intercambio de claves públicas maestras durante la construcción del canal. Si desean generar un nuevo par de transacciones de compromiso C2a/C2b, utilizan multisig($P_{AliceRSMC2}$, $P_{BobRSMC2}$) para la salida del RSMC.

Una vez que ambas partes conocen los valores de salida de las transacciones de compromiso, ambas crean el par de transacciones de compromiso,
p. ej., C2a/C2b, pero no intercambian firmas para las transacciones de compromiso. Ambos firman la transacción de entrega revocable (RD2a/RD2b) e intercambian las firmas. Bob firma RD1a y se la entrega a Alice (utilizando $K_{BobRSMC2}$), mientras que Alice firma RD1b y se la entrega a Bob (utilizando KAliceRSMC2).

Cuando ambas partes tienen la transacción de entrega revocable, intercambian intercambian las firmas de las transacciones de compromiso. Bob firma C1a utilizando K_{BobF} y se la entrega a Alice, y Alice firma C1b utilizando K_{AliceF} y se la entrega a Bob.

En este punto, tanto la transacción de compromiso anterior como la nueva transacción de compromiso pueden difundirse; tanto C1a/C1b como C2a/C2b son válidas. (Tenga en cuenta que los compromisos anteriores al compromiso anterior se invalidan mediante sanciones). Para invalidar C1a y C1b, ambas partes intercambian firmas de transacciones de subsanación de incumplimiento (BR1a/BR1b) para el compromiso anterior C1a/C1b. Alice envía BR1a a Bob utilizando $K_{AliceRSMC1}$, y Bob envía BR1b a Alice utilizando $K_{BobRSMC1}$. Una vez intercambiadas ambas firmas de subsanación de incumplimiento, el estado del canal se encuentra ahora en el compromiso actual C2a/C2b y los saldos quedan comprometidos.

Sin embargo, en lugar de revelar las firmas BR1a/BR1b, también es posible limitarse a revelar las claves privadas a la contraparte. Esto resulta más

eficaz, tal y como se describe más adelante en la sección de almacenamiento de claves. Se pueden revelar las claves privadas utilizadas en la propia transacción de compromiso. Por ejemplo, si Bob desea invalidar C1b, envía a Alice sus claves privadas utilizadas en C1b (*NO* revela sus claves utilizadas en C1a, ya que eso permitiría el robo de monedas). Del mismo modo, Alice revela todas sus salidas de clave privada en C1a a Bob para invalidar C1a.

Si Bob difunde incorrectamente C1b, entonces, dado que Alice tiene todas las claves privadas utilizadas en las salidas de C1b, puede quedarse con el dinero. Sin embargo, solo Bob puede difundir C1b. Para evitar este riesgo de robo de monedas, Bob debe destruir todas las transacciones de compromiso antiguas.

3.4 Cierre cooperativo de un canal

Ambas partes pueden enviar tantos pagos a su contraparte como deseen, siempre que dispongan de fondos en el canal, sabiendo que, en caso de desacuerdo, pueden publicar en la cadena de bloques el estado actual en cualquier momento.

En la gran mayoría de los casos, ninguna de las salidas de la transacción de financiación se transmitirá nunca a la cadena de bloques. Simplemente están ahí por si la otra parte no coopera, de forma muy similar a cómo rara vez se ejecuta un contrato en los tribunales. La capacidad demostrada de que el contrato se ejecute de manera determinista es un incentivo suficiente para que ambas partes actúen con honestidad. Cuando cualquiera de las partes desee cerrar un canal de forma cooperativa, podrá hacerlo poniéndose en contacto con la otra parte y gastando desde la transacción de financiación con una salida de la transacción de compromiso más reciente directamente, sin condiciones que lo impidan. No se pueden realizar más pagos en el canal.

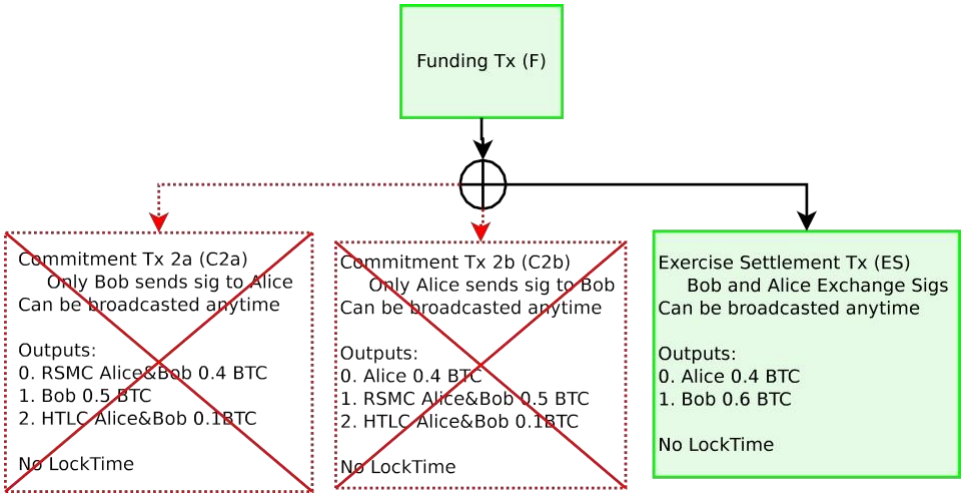

Figura 10: Si ambas contrapartes cooperan, toman los saldos de la transacción de compromiso actual y gastan desde la transacción de financiación mediante una transacción de liquidación de ejercicio (ES). Si, en cambio, se transmite la transacción de compromiso más reciente, el pago (menos las comisiones) será el mismo.

El objetivo del cierre cooperativo es reducir el número de transacciones que se producen en la cadena de bloques y que ambas partes puedan recibir sus fondos de forma inmediata (en lugar de que una de las partes tenga que esperar a que la transacción de entrega de revocación sea válida).

Los canales pueden permanecer indefinidamente hasta que decidan cerrar la transacción de forma cooperativa, o cuando una de las partes no coopere con la otra y el canal se cierre y se ejecute en la cadena de bloques.

3.5 Implicaciones y resumen de los canales bidireccionales

Al garantizar que los canales solo puedan actualizarse con el consentimiento de ambas partes, es posible construir canales que existan de forma perpetua en la cadena de bloques. Ambas partes pueden actualizar el saldo dentro del canal con los saldos de salida que deseen, siempre que sea igual o inferior al total de fondos comprometidos en la transacción de financiación; los saldos pueden moverse en ambas direcciones. Si una de las partes actúa de forma maliciosa, cualquiera de las partes puede cerrar inmediatamente el canal y transmitir el estado más actual a la cadena de bloques. Mediante el uso de una estructura de fianza de fidelidad (transacciones de entrega revocables), si una parte incumple los términos del canal, los fondos se enviarán a la contraparte,

siempre que la prueba de la infracción (transacción de subsanación de incumplimiento) se introduzca en la cadena de bloques a su debido tiempo. Si ambas partes cooperan, el canal puede permanecer abierto indefinidamente, posiblemente durante muchos años.

Este tipo de estructura solo es posible porque la resolución se produce de forma programada a través de la cadena de bloques como parte del consenso de Bitcoin, por lo que no es necesario confiar en la otra parte. Como resultado, la contraparte del canal no posee la custodia ni el control total de los fondos.

4 Contrato con bloqueo temporal hash (HTLC)

Un canal de pago bidireccional solo permite la transferencia segura de fondos dentro de un canal. Para poder realizar transferencias seguras utilizando una red de canales a través de múltiples saltos hasta el destino final se requiere una estructura adicional: un contrato con bloqueo temporal hash (HTLC).

El propósito de un HTLC es permitir un estado global a través de múltiples nodos mediante hash. Este estado global se garantiza mediante compromisos temporales y la liberación de recursos basada en el tiempo a través de la divulgación de preimágenes. El «bloqueo» transaccional se produce de forma global a través de compromisos; en cualquier momento, un único participante es responsable de revelar al siguiente participante si tiene conocimiento de la preimagen *R*. Esta estructura no requiere confianza de custodia en la contraparte del canal, ni en ningún otro participante de la red.

Para lograrlo, un HTLC debe poder crear ciertas transacciones que solo sean válidas después de una fecha determinada, utilizando nLockTime, así como la divulgación de información a la contraparte del canal. Además, estos datos deben ser revocables, ya que debe ser posible deshacer un HTLC.

Un HTLC es también un contrato de canal con la contraparte que es ejecutable a través de la cadena de bloques. Las contrapartes de un canal acuerdan los siguientes términos para un contrato con bloqueo temporal hash:

1. Si Bob puede proporcionar a Alice unos datos de entrada aleatorios desconocidos de 20 bytes *R* a partir de un hash conocido *H*, en un plazo de tres días, entonces Alice liquidará el contrato pagando a Bob 0,1 BTC.

2. Si han transcurrido tres días, la cláusula anterior queda sin efecto y el proceso de liquidación se invalida; ninguna de las partes debe intentar liquidar ni reclamar el pago una vez transcurridos los tres días.

3. Cualquiera de las partes podrá (y deberá) efectuar el pago de conformidad con las condiciones del presente contrato mediante cualquier método que elijan los participantes y rescindir el contrato anticipadamente, siempre y cuando ambas partes del contrato estén de acuerdo.

4. El incumplimiento de los términos anteriores acarreará una penalización máxima equivalente a los fondos bloqueados en este contrato, que se pagará a la contraparte que no haya incumplido como fianza de fidelidad.

Para mayor claridad en los ejemplos, utilizamos días para los HTLC y la altura del bloque para los RSMC. En realidad, el HTLC también debería definirse como una altura de bloque (por ejemplo, 3 días equivalen a 432 bloques).

En efecto, se desea construir un pago que dependa del conocimiento de *R* por parte del destinatario dentro de un plazo determinado. Transcurrido este plazo, los fondos se devuelven al remitente.

Al igual que en los RSMC, estas condiciones contractuales se aplican de forma programática en la cadena de bloques de Bitcoin y no requieren confianza en que la contraparte cumpla con los términos del contrato, ya que todas las infracciones se penalizan mediante fianzas de fidelidad aplicadas unilateralmente, que se construyen utilizando transacciones de penalización que gastan desde los estados de compromiso. Si Bob conoce *R* en un plazo de tres días, puede canjear los fondos emitiendo una transacción; Alice no puede retener los fondos de ninguna manera, ya que el script se considera válido cuando la transacción se gasta en la cadena de bloques de Bitcoin.

Un HTLC es una salida adicional en una transacción de compromiso con un script de salida único:

```
OP IF
        OP HASH160 <Hash160 (R)> OP EQUALVERIFY
        2 <Alice 2 > <Bob2> OP CHECKMULTISIG
OP ELSE
        2 <Alice 1> <Bob1> OP CHECKMULTISIG
OP ENDIF
```

Conceptualmente, este script tiene dos rutas posibles para gastar desde una única salida HTLC. La primera ruta (definida en el OP IF) envía fondos a Bob si Bob puede generar *R*. La segunda ruta se canjea mediante un reembolso con bloqueo temporal de 3 días a Alice. El bloqueo temporal de 3 días se aplica utilizando nLockTime de la transacción de gasto.

4.1 Construcción de HTLC no revocable

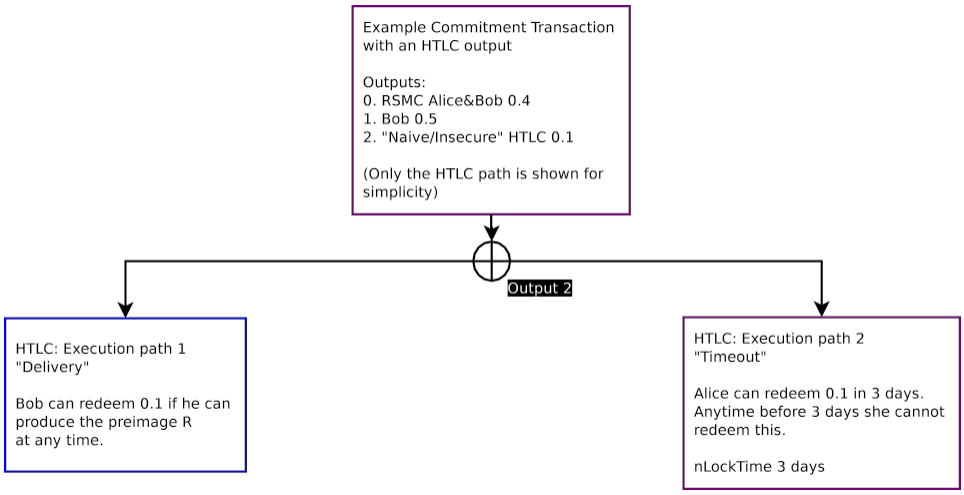

Figura 11: Esta es una implementación ingenua y no funcional de un HTLC. Solo se muestra la ruta HTLC de la transacción de compromiso. Tenga en cuenta que hay dos posibles gastos desde una salida HTLC. Si Bob puede producir la preimagen *R* en un plazo de 3 días, puede canjear la ruta 1. Pasados tres días, Alice puede transmitir la ruta 2. Una vez transcurridos los 3 días, cualquiera de las dos es válida. Este modelo, sin embargo, no funciona con múltiples transacciones de compromiso.

Si *R* se genera en un plazo de 3 días, Bob puede canjear los fondos difundiendo la transacción de «Entrega». Para que la transacción de «Entrega» sea válida, es necesario que *R* se incluya en la transacción. Si *R* no se incluye, la transacción de «Entrega» no es válida. Sin embargo, si han transcurrido 3 días, los fondos pueden devolverse a Alice difundiendo la transacción «Timeout». Cuando han transcurrido 3 días y se ha revelado *R*, cualquiera de las transacciones puede ser válida.

Es responsabilidad individual de ambas partes asegurarse de que pueden introducir su transacción en la cadena de bloques para garantizar que los saldos sean correctos. En el caso de Bob, para recibir los fondos, debe transmitir la transacción «Delivery» en la cadena de bloques de Bitcoin o, de lo contrario, liquidar la operación con Alice (cancelando al mismo tiempo el HTLC). En el caso de Alice, debe transmitir la transacción «Timeout» dentro de tres días para recibir el reembolso, o cancelar por completo el HTLC con Bob.

Sin embargo, este tipo de construcción simplista presenta problemas similares a los de una

una construcción incorrecta de un canal de pago bidireccional. Cuando se transmite una transacción de compromiso antigua, cualquiera de las partes puede intentar robar fondos, ya que ambas rutas pueden ser válidas a posteriori. Por ejemplo, si *R* se revela un año después y se transmite una transacción de compromiso incorrecta, ambas rutas son válidas y pueden ser canjeadas por cualquiera de las partes; el contrato aún no es ejecutable en la cadena de bloques. Cerrar el HTLC es absolutamente necesario, ya que para que Alice obtenga su reembolso, debe rescindir el contrato y recibir su reembolso. De lo contrario, cuando Bob descubra *R* tras haber transcurrido tres días, podría robar los fondos que deberían ir a parar a Alice. Con contrapartes poco cooperativas, no es posible rescindir un HTLC sin difundirlo a la cadena de bloques de bitcoin, ya que la parte poco cooperativa no está dispuesta a crear una nueva transacción de compromiso.

4.2 HTLC revocable fuera de cadena

Para poder rescindir este contrato fuera de cadena sin una difusión a la cadena de bloques de Bitcoin, es necesario incrustar RSMC en la salida, lo que tendrá una estructura similar a la del canal bidireccional.

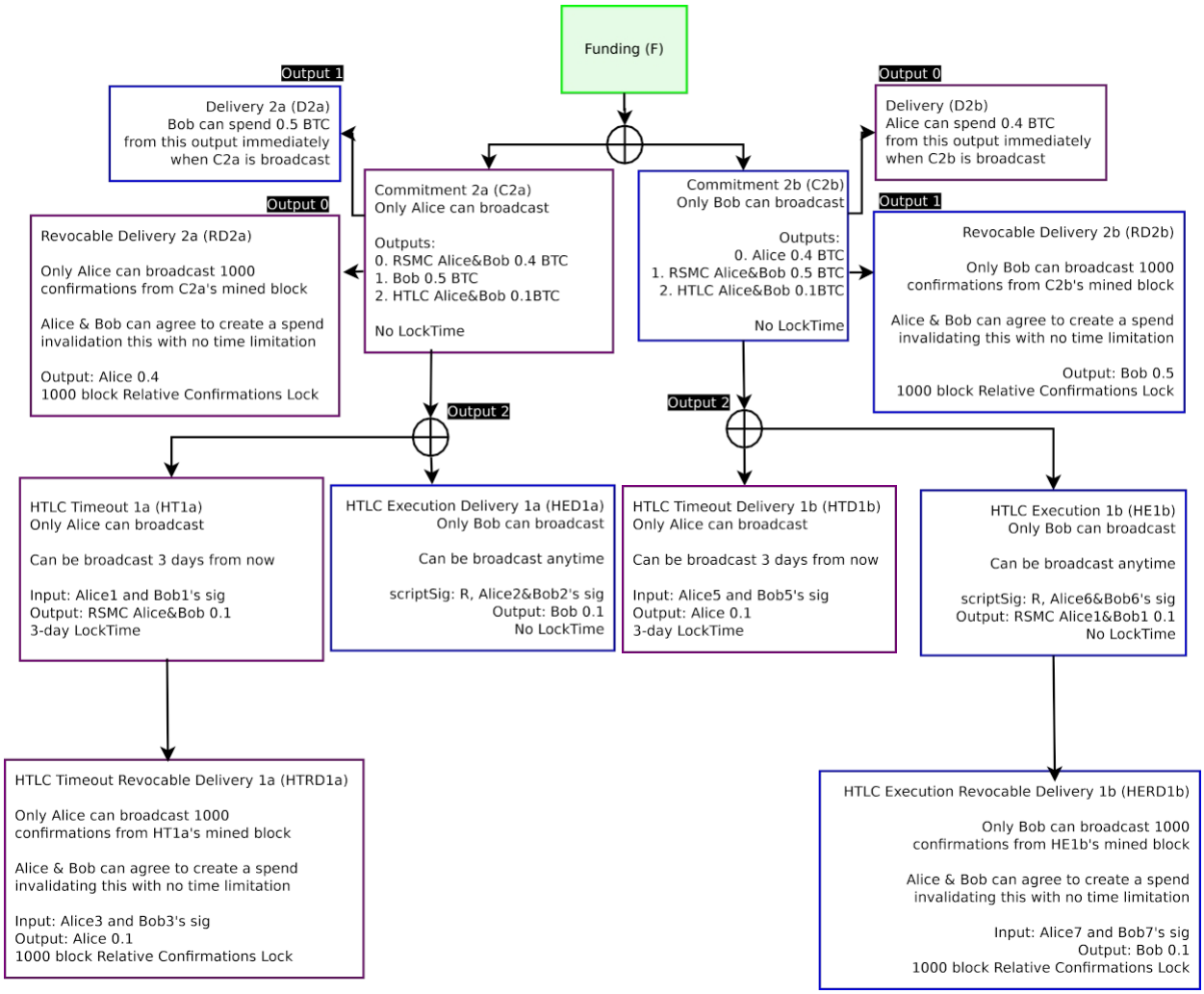

Figura 12: Si Alice difunde C2a, se ejecutará la mitad izquierda. Si Bob difunde C2b, se ejecutará la mitad derecha. Cualquiera de las partes puede difundir su transacción de compromiso en cualquier momento. El tiempo de espera del HTLC solo es válido tras 3 días. Las ejecuciones del HTLC solo pueden difundirse si se conoce la preimagen del hash *R*. Los compromisos anteriores (y sus transacciones dependientes) no se muestran por brevedad.

Supongamos que Alice y Bob desean actualizar su saldo en el canal en el Compromiso 1 con un saldo de 0,5 para Alice y 0,5 para Bob.

Alice desea enviar 0,1 a Bob a condición de que conozca *R* en un plazo de 3 días; transcurridos 3 días, quiere que le devuelvan su dinero si Bob no presenta *R*. La nueva transacción de compromiso incluirá un reembolso completo del saldo actual saldo actual a Alice y Bob (salidas 0 y 1), siendo la salida 2 el HTLC, que describe los fondos en tránsito. Como 0,1 quedará comprometido en un HTLC, el saldo de Alice se reduce a 0,4 y el de Bob permanece igual en

0,5.

Esta nueva transacción de compromiso (C2a/C2b) tendrá un HTLC salida con dos posibles gastos. Cada gasto es diferente en función de la versión de la transacción de compromiso de cada contraparte. Al igual que en el canal de pago bidireccional, cuando una de las partes difunde su compromiso, se asumirá que los pagos a la contraparte son válidos y no se invalidarán. Esto puede ocurrir porque, cuando se transmite una transacción de compromiso, se certifica que se trata de la transacción de compromiso más reciente. Si es la más reciente, también se certifica que el HTLC existe y no ha sido invalidado anteriormente, por lo que los posibles pagos a la contraparte deberían ser válidos.

Tenga en cuenta que los nombres de las transacciones HTLC (que comienzan con la letra H) empezarán por el número 1, cuyos valores no se correlacionan con las transacciones de compromiso. Se trata simplemente de la primera transacción HTLC. Las transacciones HTLC pueden persistir entre transacciones de compromiso. Cada HTLC tiene 4 claves por cada lado de la transacción (C2a y C2b), lo que suma un total de 8 claves por contraparte.

La salida HTLC en la transacción de compromiso tiene dos conjuntos de claves por contraparte en la salida.

Para la transacción de compromiso de Alice (C2a), el script de salida HTLC requiere *multisig*(P_{Alice2} , P_{Bob2}) gravada por la divulgación de *R*, así como *multisig*(P_{Alice1} , P_{Bob1}) sin gravamen.

Para la transacción de compromiso de Bob (C2b), el script de salida HTLC requiere *multisig*(P_{Alice6} , P_{Bob6}) gravado por la divulgación de *R*, así como *multisig*(P_{Alice5} , P_{Bob5}) sin gravamen.

Los estados de salida de HTLC varían en función de la transacción de compromiso que se transmita.

4.2.1 HTLC cuando el remitente difunde la transacción de compromiso

Para el remitente (Alice), la transacción de «Entrega» se envía como una transacción de Entrega de Ejecución HTLC (HED1a), que no está sujeta a ningún gravamen en un RSMC. Se asume que este HTLC nunca se ha cancelado fuera de la cadena, ya que Alice certifica que la transacción de compromiso difundida es la más reciente. Si Bob puede generar la preimagen *R*, podrá canjear los fondos del HTLC después de que la transacción de compromiso se haya difundido en la cadena de bloques.

Esta transacción consume *multisig*(P_{Alice2} , P_{Bob2}) si Alice difunde su compromiso C2a. Solo Bob puede difundir HED1a, ya que solo Alice le dio su firma para HED1a a Bob.

Sin embargo, si han transcurrido 3 días desde la creación del HTLC, Alice podrá transmitir una transacción de «tiempo de espera», la transacción de tiempo de espera del HTLC (HT1a). Esta transacción es un RSMC. Consume la salida *multisig*(P_{Alice1} , P_{Bob1}) sin requerir la divulgación de *R* si Alice transmite C2a. Esta transacción no puede entrar en la cadena de bloques hasta que hayan transcurrido 3 días. La salida de esta transacción es un RSMC con *multisig*(P_{Alice3} , P_{Bob3}) con una madurez relativa de 1000 bloques, y *multisig*(P_{Alice4} , P_{Bob4}) sin requisito de madurez de confirmación. Solo Alice puede transmitir HT1a, ya que solo Bob proporcionó su firma para HT1a a Alice.

Una vez que HT1a entra en la cadena de bloques y se producen 1000 confirmaciones de bloque, Alice puede transmitir una transacción HTLC de entrega revocable por tiempo de espera (HTRD1a) que consume *la firma múltiple* (P_{Alice3} , P_{Bob3}). Solo Alice puede transmitir HTRD1a 1000 bloques después de que se transmita HT1a, ya que solo Bob proporcionó su firma para HTRD1a a Alice. Esta transacción puede ser revocable cuando otra transacción sustituya a HTRD1a utilizando *multisig*(P_{Alice4} , P_{Bob4}), que no tiene ningún requisito de vencimiento de bloques.

4.2.2 HTLC cuando el receptor transmite la transacción de compromiso

Para el posible destinatario (Bob), el «tiempo de espera» de la recepción se reembolsa mediante una transacción HTLC de entrega por tiempo de espera (HTD1b). Esta transacción reembolsa directamente los fondos al remitente original (Alice) y no está sujeta a un RSMC. Se da por supuesto que este HTLC nunca se ha cancelado fuera de la cadena, ya que Bob certifica que la transacción de compromiso (C2b) difundida es la más reciente. Si han transcurrido 3 días, Alice puede transmitir HTD1b y recibir el reembolso. Esta transacción consume *multisig*(P_{Alice5} , P_{Alice5}) si Bob transmite C2b. Solo Alice puede transmitir HTD1b, ya que Bob le dio su firma para HTD1b a Alice.

Sin embargo, si HTD1b no se transmite (no han transcurrido 3 días) y Bob conoce la preimagen *R*, entonces Bob podrá transmitir la transacción de ejecución HTLC (HE1b) si puede producir *R*. Esta transacción es una

RSMC. Consume la salida *multisig*(P_{Alice6} , P_{Bob6}) y requiere la revelación de R si Bob transmite C2b. La salida de esta transacción es un RSMC con *multisig*(P_{Alice7} , P_{Bob7}) con una madurez relativa de 1000 bloques, y *multisig*(P_{Alice8} , P_{Bob8}), que no tiene ningún requisito de madurez de bloques. Solo Bob puede difundir HE1b, ya que solo Alice proporcionó su firma para HE1b a Bob.

Una vez que HE1b entra en la cadena de bloques y se producen 1000 confirmaciones de bloque, Bob puede transmitir una transacción HTLC de entrega revocable de ejecución (HERD1b) que consume *la firma múltiple* (P_{Alice7} , P_{Bob7}). Solo Bob puede transmitir HERD1b 1000 bloques después de que se haya transmitido HE1b, ya que solo Alice proporcionó su firma para HERD1b a Bob. Esta transacción puede ser revocable cuando otra transacción sustituya a HERD1b utilizando *multisig*(P_{Alice8} , P_{Bob8}), que no tiene ningún requisito de vencimiento de bloque.

4.3 Terminación fuera de cadena de HTLC

Una vez construido un HTLC, para terminarlo fuera de cadena es necesario que ambas partes se pongan de acuerdo sobre el estado del canal. Si el destinatario puede demostrar a la contraparte que conoce R, está demostrando que es capaz de cerrar inmediatamente el canal en la cadena de bloques de Bitcoin y recibir los fondos. En este punto, si ambas partes desean mantener el canal abierto, deben terminar el HTLC fuera de cadena y crear una nueva transacción de compromiso que refleje el nuevo saldo.

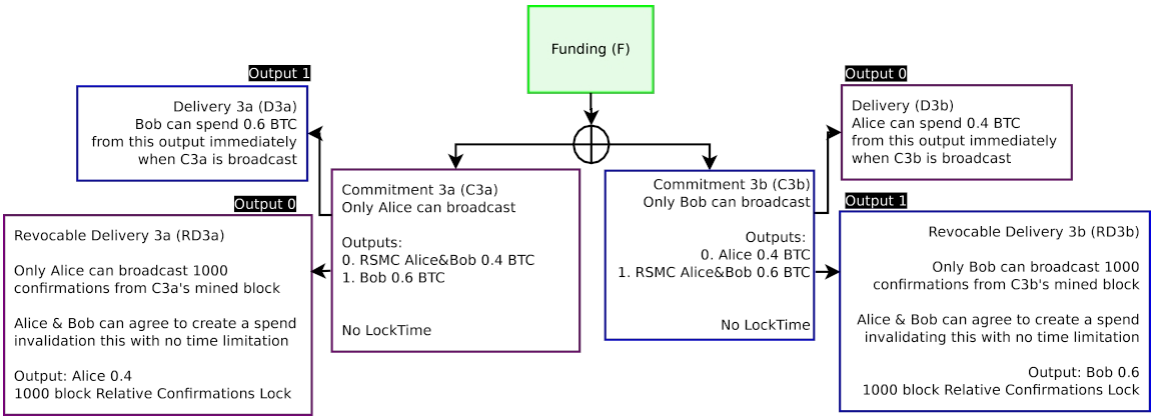

Figura 13: Dado que Bob ha demostrado a Alice que conoce *R* al revelarle *R*, Alice está dispuesta a actualizar el saldo con una nueva transacción de compromiso. El pago será el mismo tanto si se transmite C2 como C3 en este momento.

Del mismo modo, si el destinatario no es capaz de demostrar que conoce *R* revelando *R*, ambas partes deben acordar rescindir el HTLC y crear una nueva transacción de compromiso con el saldo del HTLC reembolsado al remitente.

Si las contrapartes no logran llegar a un acuerdo o dejan de responder, deben cerrar el canal transmitiendo las transacciones de canal necesarias en la cadena de bloques de Bitcoin.

Sin embargo, si cooperan, pueden hacerlo generando primero una nueva transacción de compromiso con los nuevos saldos y, a continuación, invalidando el compromiso anterior mediante el intercambio de transacciones de subsanación de incumplimiento (BR2a/BR2b). Además, si están rescindiendo un HTLC concreto, también deben intercambiar algunas de sus propias claves privadas utilizadas en las transacciones del HTLC.

Por ejemplo, si Alice desea rescindir el HTLC, revelará K_{Alice1} y K_{Alice4} a Bob. Del mismo modo, si Bob desea rescindir el HTLC, revelará K_{Bob6} y K_{Bob8} a Alice. Una vez reveladas las claves privadas a la contraparte, si Alice transmite C2a, Bob podrá retirar todos los fondos del HTLC inmediatamente. Si Bob transmite C2b, Alice podrá retirar todos los fondos del HTLC inmediatamente. Tenga en cuenta que, cuando se rescinde un HTLC, también debe revocarse la transacción de compromiso anterior.

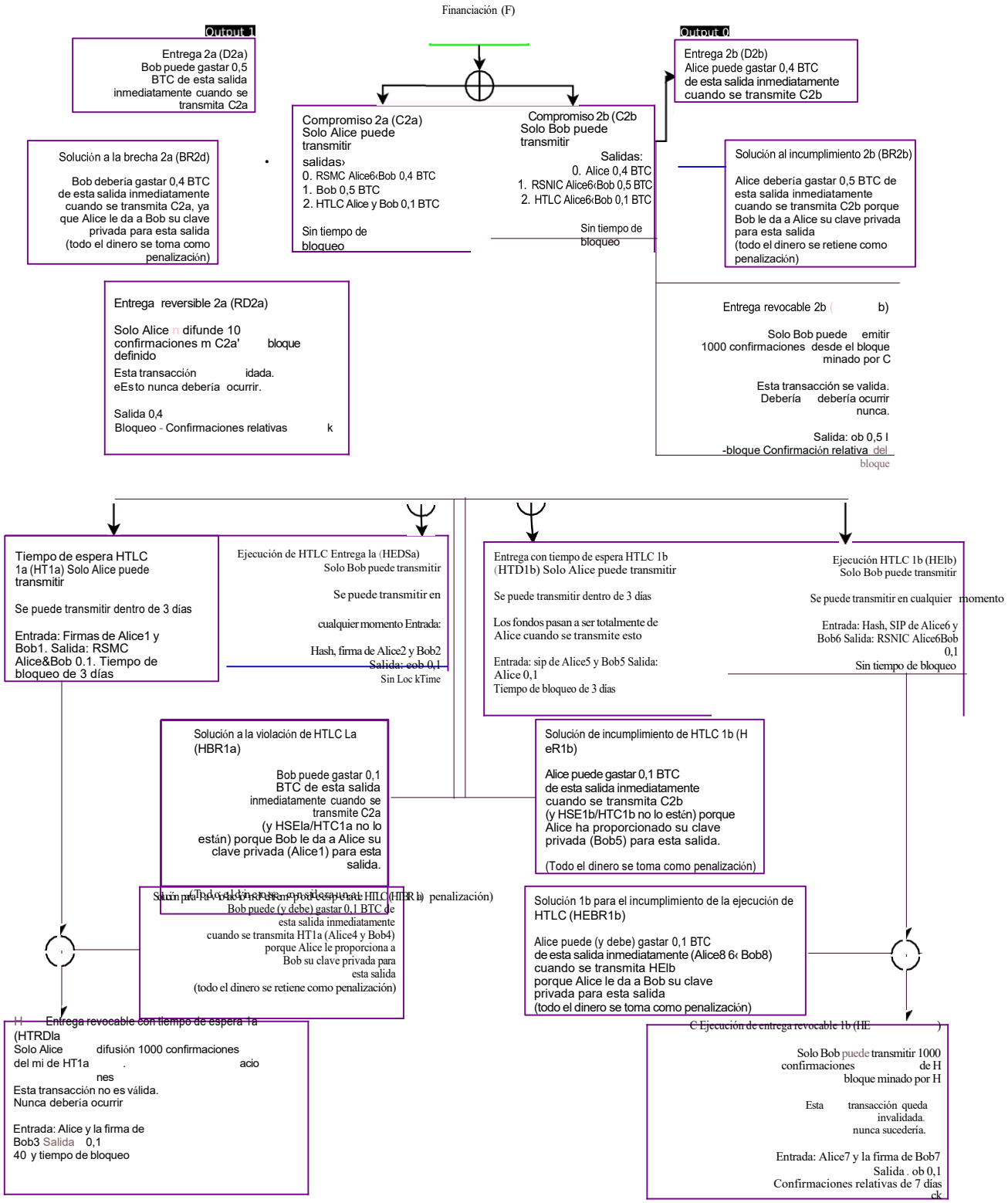

Figura 14: Una transacción de compromiso totalmente revocada y un HTLC rescindido. Si cualquiera de las partes transmite el Compromiso 2, perderá todo su dinero a favor de la contraparte. No se muestran otros compromisos (por ejemplo, si el Compromiso 3 es el compromiso actual) por motivos de brevedad.

Dado que ambas partes pueden demostrar el estado actual entre sí, pueden llegar a un acuerdo sobre el saldo actual dentro del canal. Dado que pueden transmitir el estado actual en la cadena de bloques, pueden llegar a un acuerdo sobre la compensación y la terminación del HTLC con una nueva transacción de compromiso.

4.4 Formación y orden de cierre de un HTLC

Para crear un nuevo HTLC, el proceso es el mismo que para crear una nueva transacción de compromiso, salvo que las firmas del HTLC se intercambian antes de las firmas de la nueva transacción de compromiso.

Para cerrar un HTLC, el proceso es el siguiente (de C2 a C3):

1. Alice firma y envía su firma para RD3b y C3b. En este punto, Bob puede optar por transmitir C3b o C2b (con el HTLC) con el mismo pago. Bob está dispuesto, tras recibir C3b, a cerrar C2b.

2. Bob firma y envía su firma para RD3a y C3a, así como sus claves privadas utilizadas para el Compromiso 2 y el HTLC que se está cerrando; envía a Alice $K_{BobRSMC2}$, K_{Bob5} y K_{Bob8} . En este punto, Bob solo debe difundir C3b y no debe difundir C2b, ya que perderá todo su dinero si lo hace. Bob ha revocado por completo C2b y el HTLC. Alice está dispuesta, tras recibir C3a, a cerrar C2b.

3. Alice firma y envía su firma para RD3b y C3b, así como sus claves privadas utilizadas para el Compromiso 2 y el HTLC que se está cancelando; envía a Bob $K_{AliceRSMC2}$, K_{Bob1} y K_{Bob4} . En este punto, ninguna de las partes debe transmitir el Compromiso 2; si lo hacen, sus fondos irán a parar a la contraparte. El antiguo Compromiso y el antiguo HTLC quedan ahora revocados y totalmente rescindidos. Solo queda el nuevo Compromiso 3, que no tiene un HTLC.

Cuando se ha cerrado el HTLC, los fondos se actualizan de modo que el saldo actual en el canal sea el que se habría producido si el contrato HTLC se hubiera completado y difundido en la cadena de bloques. En su lugar, ambas partes optan por realizar una novación fuera de cadena y actualizar sus pagos dentro del canal.

Es absolutamente necesario que ambas partes completen la novación fuera de cadena dentro del plazo designado. En cuanto al receptor (Bob), debe

conocer *R* y actualizar su saldo con Alice en un plazo de 3 días (o el tiempo que se haya seleccionado); de lo contrario, Alice podrá canjearlo en un plazo de 3 días. En cuanto a Alice, muy poco después de que su tiempo de espera sea válido, debe realizar la novación o transmitir la transacción HTLC Timeout. También debe realizar la novación o transmitir la transacción HTLC Timeout Revocable Delivery tan pronto como sea válida. Si la contraparte no está dispuesta a realizar la novación o está dando largas al asunto, entonces se debe transmitir el estado actual del canal, incluidas las transacciones HTLC, a la cadena de bloques de Bitcoin.

La flexibilidad temporal de estas ofertas de novación depende de las dependencias contingentes respecto al hashlock *R*. Si se establece un contrato según el cual el HTLC debe resolverse en el plazo de 1 día, entonces, si la transacción caduca, Alice deberá resolverla antes del día 4 (3 días más 1); de lo contrario, Alice corre el riesgo de perder los fondos.

5 Almacenamiento de claves

Las claves se generan mediante carteras determinísticas jerárquicas según el BIP 0032[17]. Ambas partes generan las claves por adelantado. Las claves se generan en un árbol de Merkle y se encuentran en niveles muy profundos del árbol. Por ejemplo, Alice genera por adelantado un millón de claves, cada una de las cuales es una «hija» de la clave anterior. Alice asigna qué claves utilizar de acuerdo con un método determinista. Por ejemplo, comienza con la hija más profunda del árbol para generar muchas subclaves para el día 1. Esta clave se utiliza como clave maestra para todas las claves generadas el día 1. Ella le da a Bob la dirección que desea utilizar para la siguiente transacción y le revela la clave privada a Bob cuando esta queda invalidada. Cuando Alice revela a Bob todas las claves privadas derivadas de la clave maestra del día 1 y no desea seguir utilizando esa clave maestra, puede revelar la clave maestra del día 1 a Bob. En este punto, Bob no necesita almacenar todas las claves derivadas de la clave maestra del día 1. Bob hace lo mismo con Alice y le da su clave del día 1.

Cuando se hayan intercambiado todas las claves privadas del día 2, por ejemplo, para el día 5, Alice revela su clave del día 2. Bob puede generar la clave del día 1 a partir de la clave del día 2, ya que la clave del día 1 también es una derivada de la clave del día 2.

Si una contraparte transmite la transacción de compromiso incorrecta, la clave privada que se debe utilizar en una transacción para recuperar fondos puede determinarse mediante un ataque de fuerza bruta o, si ambas partes están de acuerdo, pueden utilizar el número de identificación de secuencia

al crear la transacción para identificar qué conjuntos de claves se utilizan.

Esto permite a los participantes en un canal invalidar estados de salida (transacciones) anteriores por ambas partes sin utilizar apenas datos. Al revelar claves privadas preestablecidas en un árbol de Merkle, es posible invalidar millones de transacciones antiguas con solo unos pocos kilobytes de datos por canal. Los canales principales de la Red Lightning pueden realizar miles de millones de transacciones sin necesidad de incurrir en costes de almacenamiento significativos.

6 Comisiones de transacción de la cadena de bloques para canales bidireccionales

Es posible que cada participante genere diferentes versiones de las transacciones para atribuir la culpa de quién las ha difundido en la cadena de bloques. Al saber quién ha difundido una transacción y tener la capacidad de atribuir la culpa, se puede utilizar un servicio de terceros para retener las comisiones en un depósito en garantía multisig 2 de 3. Si se desea transmitir la cadena de transacciones en lugar de aceptar realizar un cierre de financiación o una sustitución con una nueva transacción de compromiso, se comunicaría con el tercero y se transmitiría la cadena a la cadena de bloques. Si la contraparte rechaza la notificación del tercero para cooperar, la penalización se asigna a la parte no cooperativa. En la mayoría de los casos, los participantes pueden mostrarse indiferentes ante las comisiones de transacción en caso de una contraparte no cooperativa.

Se debería elegir contrapartes en el canal que sean cooperativas, pero no es una necesidad absoluta para que el sistema funcione. Tenga en cuenta que esto no requiere confianza entre el resto de la red, y solo es relevante para las comisiones de transacción comparativamente menores. La parte menos confiable podría ser simplemente la responsable de las comisiones de transacción.

Es probable que las comisiones de Lightning Network sean significativamente más bajas que las comisiones de transacción de la cadena de bloques. Las comisiones se derivan en gran medida del valor temporal de bloquear fondos para una ruta concreta, así como del pago por la posibilidad de cierre del canal en la cadena de bloques. Estas deberían ser significativamente más bajas que las transacciones en cadena, ya que muchas transacciones en un canal de Lightning Network pueden liquidarse en una sola transacción de la cadena de bloques. Con una red suficientemente robusta e interconectada, las comisiones deberían acercarse asintóticamente a la insignificancia para muchos tipos de transacciones. Con comisiones baratas y transacciones rápidas, será posible crear micropagos escalables, incluso entre

sistemas de alta frecuencia, como las aplicaciones del Internet de las cosas o la microfacturación por unidad.

7 Pago por contrato

Es posible construir un contrato «Entrega contra pago» criptográficamente demostrable, o «pago contra contrato»[18], como prueba de pago. Esta prueba puede establecerse como el conocimiento de la entrada R a partir de hash(R) como pago de un valor determinado. Al incluir una cláusula en el contrato entre el comprador y el vendedor que establezca que conocer R es prueba de los fondos enviados, el destinatario de los fondos no tiene ningún incentivo para revelar R a menos que tenga la certeza de que recibirá el pago. Cuando la contraparte finalmente retira los fondos del comprador a través de su canal de micropagos, R se revela como parte de esa retirada de fondos. Se pueden diseñar documentos legales en papel que especifiquen que el conocimiento o la revelación de R implica el cumplimiento del pago. El remitente puede entonces formalizar un contrato firmado criptográficamente con conocimiento de las entradas para los hash, lo cual se considera cumplimiento del contrato en papel antes de que se produzca el pago.

8 La red Lightning de Bitcoin

Al disponer de un canal de micropagos con contratos sujetos a hashlocks y timelocks, es posible liquidar transacciones en una red de pagos de múltiples saltos utilizando una serie de timelocks decrecientes sin necesidad de cámaras de compensación centrales adicionales.

Tradicionalmente, los mercados financieros liquidan las transacciones transfiriendo la obligación de entrega a un punto central y las liquidan transfiriendo la propiedad a través de este centro. Los sistemas de transferencias bancarias y de fondos (como ACH y la red de tarjetas Visa), o las cámaras de compensación de valores (como la DTCC) operan de esta manera.

Dado que Bitcoin permite el dinero programable, es posible crear transacciones sin contactar con una cámara de compensación central. Las transacciones pueden ejecutarse fuera de la cadena sin que un tercero recaude todos los fondos antes de desembolsarlos; solo las transacciones con contrapartes de canal no cooperativas se adjudican automáticamente en la cadena de bloques.

La obligación de entregar los fondos al destinatario final se cumple mediante un proceso de delegación en cadena. Cada participante a lo largo del camino asume la obligación de entregar a un destinatario concreto. Cada participante transfiere esta obligación al siguiente participante en la ruta. La obligación de cada participante posterior a lo largo de la ruta, definida en sus respectivos HTLC, tiene un plazo de cumplimiento más corto en comparación con el participante anterior. De esta manera, cada participante tiene la seguridad de que podrá reclamar los fondos cuando la obligación se transmita a lo largo de la ruta.

La programación de transacciones de Bitcoin, una forma de lo que algunos denominan una implementación de «contratos inteligentes»[19], permite crear sistemas sin cámaras de compensación de custodia de confianza ni servicios de depósito en garantía.

8.1 Bloqueos temporales decrecientes

Supongamos que Alice desea enviar 0,001 BTC a Dave. Encuentra una ruta a través de Bob y Carol. La ruta de transferencia sería de Alice a Bob, de Bob a Carol y de Carol a Dave.

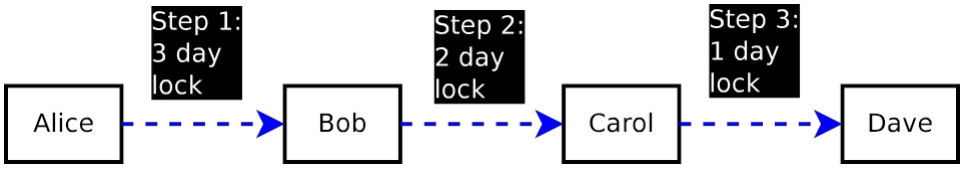

Figura 15: Pago a través de la Red Lightning utilizando HTLC.

Cuando Alice envía el pago a Dave a través de Bob y Carol, solicita a Dave el hash(R) para utilizarlo en este pago. A continuación, Alice cuenta el número de saltos hasta el destinatario y utiliza ese valor como fecha de caducidad del HTLC. En este caso, establece la caducidad del HTLC en 3 días. A continuación, Bob crea un HTLC con Carol con una caducidad de 2 días, y Carol hace lo mismo con Dave con una caducidad de 1 día. Dave ahora es libre de revelar R a Carol, y es probable que ambas partes acuerden una liquidación inmediata mediante novación con una transacción de compromiso sustitutiva. Esto se produce entonces paso a paso de vuelta a Alice. Tenga en cuenta que esto ocurre fuera de la cadena, y no se transmite nada a la cadena de bloques cuando todas las partes cooperan.

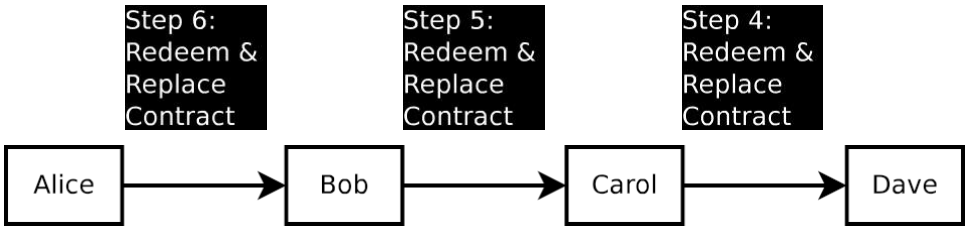

Figura 16: Liquidación de HTLC, los fondos de Alice se envían a Dave.

Se utilizan bloqueos temporales decrecientes para que todas las partes a lo largo de la ruta sepan que la revelación de R permitirá a la parte reveladora retirar fondos, ya que, en el peor de los casos, retirarán fondos después de la fecha en la que deben recibir R. Si Dave no entrega R a Carol en el plazo de 1 día, entonces Carol podrá cerrar el HTLC. Si Dave transmite R después de 1 día, entonces no podrá retirar fondos de Carol. La responsabilidad de Carol ante Bob se produce el día 2, por lo que Carol nunca será responsable del pago a Dave sin la capacidad de retirar fondos de Bob, siempre que actualice su transacción con Dave mediante transmisión a la cadena de bloques o mediante novación.

En caso de que R se revele a los participantes a mitad del plazo de vencimiento a lo largo de la ruta (por ejemplo, el día 2), es posible que algunas partes de la ruta se enriquezcan. El remitente podrá conocer R, por lo que, debido a «Pay to Contract», el pago se habrá cumplido aunque el destinatario no haya recibido los fondos. Por lo tanto, el receptor nunca debe revelar R a menos que haya recibido un HTLC de su contraparte del canal; se le garantiza recibir el pago de una de sus contrapartes del canal tras la revelación de la preimagen.

En caso de que una de las partes se desconecte por completo, la contraparte será responsable de transmitir el estado actual de la transacción de compromiso del canal a la cadena de bloques. Solo se cerrará en la cadena de bloques el estado del canal que haya fallado y no responda; el resto de canales deberán seguir actualizando sus transacciones de compromiso mediante novación dentro del canal. Por lo tanto, el riesgo de contraparte en las comisiones de transacción solo afecta a las contrapartes directas del canal. Si un nodo a lo largo de la ruta decide dejar de responder, los participantes que no estén conectados directamente a ese nodo solo sufrirán una disminución del valor temporal de sus fondos al no realizar la liquidación anticipada antes del cierre del HTLC.

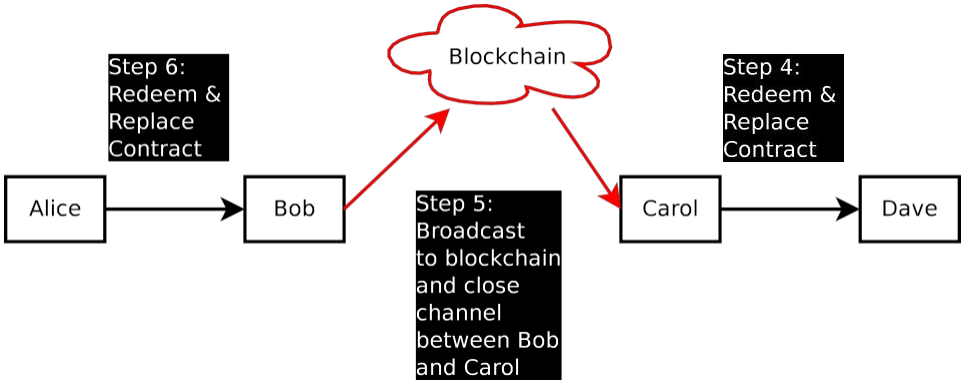

Figura 17: Solo los canales que no responden se transmiten en la cadena de bloques; todos los demás se liquidan fuera de la cadena mediante novación.

8.2 Importe del pago

Es preferible utilizar un pago pequeño por HTLC. No se debe utilizar un pago extremadamente alto, en caso de que el pago no llegue completamente a su destino. Si el pago no llega a su destino y uno de los participantes a lo largo del camino no coopera, es posible que el remitente deba esperar hasta el vencimiento antes de recibir un reembolso. La entrega puede tener pérdidas, similar a los paquetes en Internet, pero la red no puede robar fondos en tránsito. Dado que las transacciones no llegan a la cadena de bloques con contrapartes de canal cooperativas, se recomienda utilizar un pago lo más pequeño posible. Existe una disyuntiva entre bloquear las comisiones de transacción en cada salto y el deseo de utilizar un importe de transacción lo más pequeño posible (lo cual puede incurrir en comisiones totales más elevadas). Las transferencias más pequeñas con más intermediarios implican un porcentaje más alto pagado como comisiones de Lightning Network a los intermediarios.

8.3 Fallo de liquidación y redireccionamiento

Si una transacción no llega a su destino final, el receptor debe enviar un pago equivalente al remitente con el mismo hash, pero sin revelar R. Esto compensará la revelación del hash para el remitente, pero puede que no para el receptor. El receptor, que generó el hash, debe descartar R y no difundirlo nunca. Si no se puede contactar con un canal a lo largo de la ruta, los canales pueden optar por esperar hasta que la ruta caduque, lo que

probablemente cerrarán el HTLC como no liquidado sin ningún pago con una nueva transacción de compromiso.

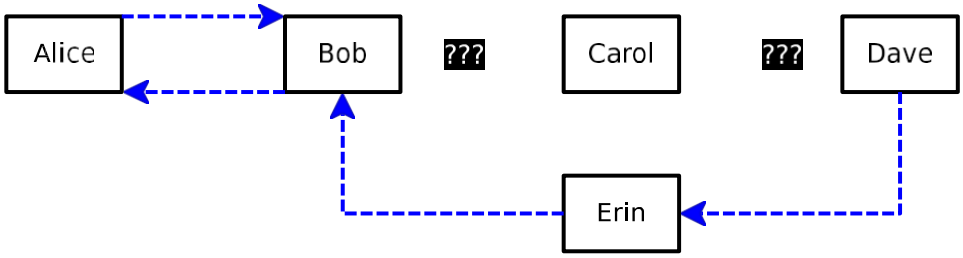

Figura 18: Dave crea una ruta de regreso hacia Alice después de que Alice no logre enviar fondos a Dave, debido a que Carol no coopera. Dave nunca transmite la entrada R de hash(R), porque Carol no completó sus acciones. Si se transmitiera R, Alice quedaría en equilibrio. Dave, quien controla R, nunca debe transmitirla porque es posible que no reciba fondos de Carol; debe dejar que los contratos caduquen. Alice y Bob también tienen la opción de saldar y cerrar el contrato anticipadamente, en este diagrama.

Si la ruta de reembolso es la misma que la ruta de pago, y no hay contratos firmados a medias por los que una de las partes pueda robar fondos, es posible cancelar directamente la transacción sustituyéndola por una nueva transacción de compromiso que comience con el nodo más reciente que participó en el HTLC.

También es posible liquidar un canal creando una ruta alternativa en la que el pago se realice en la dirección opuesta (con un saldo neto de cero) y/o creando una ruta totalmente alternativa para la ruta de pago. Esto creará un valor temporal del dinero por revelar entradas a los hash en la Red Lightning. Los participantes pueden especializarse en una alta conectividad entre nodos y ofrecer descargar hashlocks de contratos de otros nodos a cambio de una tarifa. Estos participantes aceptarán pagos que se compensan a cero (más tarifas), pero prestan bitcoins durante un período de tiempo determinado. Lo más probable es que estas entidades con baja demanda de recursos de canal sean usuarios finales que ya están conectados a múltiples nodos bien conectados. Cuando un usuario final se conecta a un nodo, este puede pedirle al cliente que bloquee sus fondos durante varios días en otro canal que el cliente haya establecido a cambio de una tarifa. Esto se puede lograr haciendo que las nuevas transacciones requieran un nuevo hash (Y) de la entrada Y, además del hash existente que puede ser generado por cualquier participante, pero que debe revelar Y solo después de que se establezca un círculo completo. El nuevo participante tiene la misma responsabilidad, así como los mismos bloqueos temporales

que el antiguo participante al que sustituye. También es posible que un nuevo participante sustituya a varios saltos.

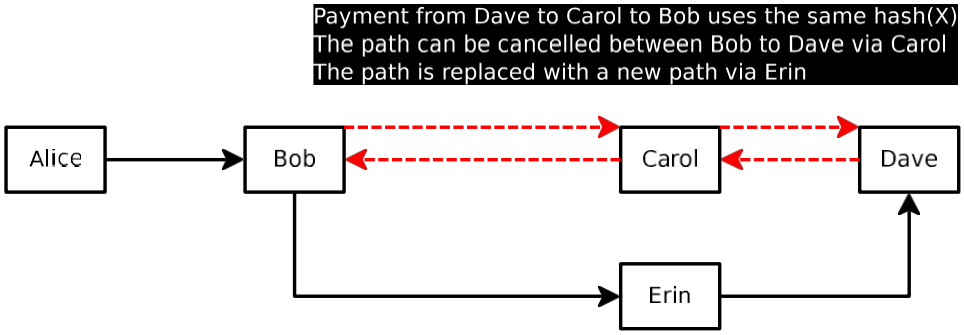

Figura 19: Erin está conectada tanto a Bob como a Dave. Si Bob desea liberar su canal con Carol, dado que ese canal está activo y es muy rentable, Bob puede descargar el pago a Dave a través de Erin. Dado que Erin tiene bitcoins adicionales disponibles, podrá cobrar una comisión por descargar el canal entre Bob y Carol, así como entre Carol y Dave. Los canales entre Bob y Carol, así como entre Carol y Dave, se deshacen y ya no tienen el HTLC, ni se ha producido ningún pago en esa ruta. El pago se realizará en la ruta en la que participa Erin. Esto se logra creando un nuevo pago de Dave a Carol y a Bob, condicionado a que Erin construya un HTLC. Los pagos en líneas discontinuas (rojas) se compensan a cero y se liquidan a través de un nuevo contrato de compromiso.

8.4 Enrutamiento de pagos

En teoría, es posible construir un mapa de rutas de forma implícita a partir de la observación de firmas múltiples 2-de-2 en la cadena de bloques para crear una tabla de enrutamiento. Sin embargo, cabe señalar que esto no es factible con salidas de transacción de pago por hash de script, que pueden resolverse fuera de banda del protocolo de bitcoin a través de un servicio de enrutamiento de terceros. La creación de una tabla de enrutamiento será necesaria para los grandes operadores (por ejemplo, BGP, Cjdns). Con el tiempo, gracias a las optimizaciones, la red se parecerá mucho a la red de bancos corresponsales o a los ISP de nivel 1. De manera similar a cómo los paquetes siguen llegando a su destino en la conexión de red de su hogar, no todos los participantes necesitan tener una tabla de enrutamiento completa. Las rutas centrales de nivel 1 pueden estar en línea todo el tiempo, mientras que los nodos en los extremos, como los usuarios promedio, estarían conectados de manera intermitente.

El descubrimiento de nodos puede ocurrir a lo largo de los bordes mediante la preselección y el ofrecimiento de rutas parciales a nodos conocidos.

8.5 Tarifas

Las tarifas de Lightning Network, que difieren de las tarifas de blockchain, se pagan directamente entre los participantes dentro del canal. Las tarifas cubren el valor temporal del dinero por el uso del canal durante un período máximo determinado, y el riesgo de contraparte por falta de comunicación.

El riesgo de contraparte para las tarifas solo existe con la contraparte directa del canal. Si un nodo a dos saltos de distancia decide desconectarse y su transacción se transmite en la cadena de bloques, las contrapartes directas no deberían transmitir en la cadena de bloques, sino continuar actualizándose mediante novación con una nueva transacción de compromiso. Consulte la entrada «Timelocks decrecientes» en la sección HTLC para obtener más información sobre el riesgo de contraparte.

El valor temporal de las comisiones paga por el tiempo consumido (por ejemplo, 3 días) y es conceptualmente equivalente a una tasa de arrendamiento de oro sin riesgo de custodia; es el valor temporal por agotar el acceso al dinero durante un período muy corto. Dado que ciertas rutas pueden volverse muy rentables en una dirección, es posible que las comisiones sean negativas para alentar que el canal esté disponible para esas rutas rentables.

9 Riesgos

Los principales riesgos están relacionados con la expiración del bloqueo de tiempo. Además, para que los nodos centrales y, posiblemente, algunos comerciantes puedan enrutar fondos, las claves deben mantenerse en línea para una menor latencia. Sin embargo, los usuarios finales y los nodos pueden mantener sus claves privadas protegidas por un firewall en almacenamiento en frío.

9.1 Bloqueos temporales inadecuados

Los participantes deben elegir bloqueos temporales con una duración suficiente. Si se asigna un tiempo insuficiente, es posible que las transacciones con bloqueo temporal que se creían inválidas se vuelvan válidas, lo que permitiría el robo de monedas por parte de la contraparte. Existe una disyuntiva entre los bloqueos temporales más largos y el valor temporal del dinero. Al escribir software de cartera y aplicaciones de la Red Lightning, es necesario asegurarse de que se asigne tiempo suficiente y de que los usuarios puedan hacer que sus transacciones ingresen a la cadena de bloques al interactuar con contrapartes de canales no cooperativas o maliciosas.

9.2 Spam por vencimiento forzado

La expiración forzada de muchas transacciones puede ser el mayor riesgo sistémico al usar Lightning Network. Si un participante malicioso crea muchos canales y fuerza que todos expiren a la vez, estos pueden saturar la capacidad de datos de los bloques, forzando la expiración y la transmisión a la cadena de bloques. El resultado sería spam masivo en la red de Bitcoin. El spam puede retrasar las transacciones hasta el punto en que otras transacciones con bloqueo de tiempo se vuelvan válidas.

Esto puede mitigarse permitiendo un reemplazo de transacción en todas las transacciones pendientes. Se puede utilizar el antispam permitiendo solo un reemplazo de transacción de un número de secuencia más alto por el inverso de un número par o impar. Por ejemplo, si se transmitió un número de secuencia impar, permitir un reemplazo por un número par más alto solo una vez. Las transacciones utilizarían el número de secuencia de manera ordenada para reemplazar otras transacciones. Esto mitiga el riesgo, suponiendo que los mineros sean honestos. Este ataque es de riesgo extremadamente alto, ya que la transmisión incorrecta de transacciones de compromiso conlleva una penalización total de todos los fondos en el canal.

Además, se podría intentar robar transacciones HTLC forzando que una transacción de tiempo de espera se procese cuando no debería. Esto se puede mitigar fácilmente haciendo que cada transferencia dentro del canal sea inferior al total de las comisiones de transacción utilizadas. Dado que las transacciones son extremadamente baratas y no llegan a la cadena de bloques con contrapartes de canal cooperativas, las transferencias de gran valor se pueden dividir en muchas transferencias pequeñas. Este intento solo puede funcionar si los bloques están completamente llenos durante mucho tiempo. Si bien es posible mitigarlo utilizando una duración de tiempo de espera de HTLC más larga, los tamaños de bloque variables pueden volverse comunes, lo que puede requerir medidas de mitigación.

Si este tipo de transacción se convierte en la forma dominante de transacciones incluidas en la cadena de bloques, podría ser necesario aumentar el tamaño del bloque y ejecutar una estructura de tamaño de bloque variable y banderas de detención de tiempo, tal como se describe en la sección siguiente. Esto puede generar sanciones y desincentivos suficientes para que resulte altamente poco rentable y fallido para los atacantes, ya que estos pierden todos sus fondos al transmitir la transacción incorrecta, hasta el punto de que nunca ocurrirá.

9.3 Robo de monedas mediante piratería informática

Dado que las partes deben estar en línea y utilizar claves privadas para firmar, existe la posibilidad de que, si la computadora donde se almacenan las claves privadas se ve comprometida, el atacante robe las monedas. Si bien puede haber métodos para mitigar la amenaza para el remitente y el destinatario, los nodos intermediarios deben estar en línea y es probable que procesen la transacción automáticamente. Por esta razón, los nodos intermediarios estarán en riesgo y no deberían mantener una cantidad sustancial de dinero en esta «billetera caliente». Los nodos intermedios que cuenten con mayor seguridad probablemente podrán superar a los demás a largo plazo y serán capaces de gestionar un mayor volumen de transacciones debido a las tarifas más bajas. Históricamente, uno de los componentes más importantes de las tarifas y los intereses en el sistema financiero proviene de diversas formas de riesgo de contraparte; en Bitcoin, es posible que el componente más importante de las tarifas se derive de las primas de riesgo de seguridad.

Una transacción de financiación puede tener múltiples salidas con múltiples transacciones de compromiso, con la clave de la transacción de financiación y algunas claves de las transacciones de compromiso almacenadas fuera de línea. Es posible crear un equivalente a una «cuenta corriente» y una «cuenta de ahorros» moviendo fondos entre las salidas de una transacción de financiación, con la «cuenta de ahorros» almacenada fuera de línea y requiriendo firmas adicionales de los servicios de seguridad.

9.4 Pérdida de datos

Cuando una de las partes pierde datos, es posible que la contraparte robe fondos. Esto se puede mitigar contando con un servicio de almacenamiento de datos de terceros al que se envían datos encriptados que la parte no puede descifrar. Además, se debe elegir contrapartes de canal que sean responsables y estén dispuestas a proporcionar el estado actual, con algunas pruebas periódicas de honestidad.

9.5 Olvidarse de transmitir la transacción a tiempo

Si no se transmite una transacción en el momento correcto, la contraparte podría robar fondos. Esto se puede mitigar contando con un tercero designado para enviar los fondos. Se puede agregar una tarifa de salida para crear un incentivo para que este tercero vigile la red. Además, esto también se puede mitigar implementando OP CHECKSEQUENCEVERIFY.

9.6 Incapacidad para realizar las bifurcaciones suaves necesarias

Son necesarios cambios en Bitcoin, como el soft fork de maleabilidad. Además, si este sistema se popularizara, será necesario que el sistema realice transacciones de forma segura con muchos usuarios y será deseable algún tipo de estructura como un «timestop» de altura de bloque. Este sistema asume que se producirán dichos cambios para permitir la existencia de Lightning Network en su totalidad, así como soft forks que garanticen una seguridad robusta frente a los atacantes. Si bien el sistema puede seguir operando solo con algunas bifurcaciones suaves de bloqueo de tiempo y maleabilidad, habrá bifurcaciones suaves necesarias en relación con los riesgos sistémicos. Sin una previsión adecuada por parte de la comunidad, la incapacidad de establecer una pausa temporal o una función similar permitirá que se produzcan ataques sistémicos y es posible que no se reconozca como imperativo hasta que realmente ocurra un ataque.

9.7 Ataques de mineros en colusión

Los mineros pueden optar por negarse a incluir transacciones específicas (por ejemplo, transacciones de subsanación de brechas) con el fin de facilitar el robo de monedas por tiempo de espera. Un atacante puede sobornar a todos los mineros para que se nieguen a incluir ciertas transacciones en su mem-pool y bloques. Los mineros pueden identificar sus propios bloques en un intento de demostrar su comportamiento al atacante que les paga.

Esto puede mitigarse animando a los mineros a evitar identificar sus propios bloques. Además, debe esperarse que este tipo de pago a los mineros sea una actividad maliciosa y que el contrato sea inaplicable. Los mineros pueden entonces aceptar el pago y minar un bloque de forma encubierta sin identificarlo ante el atacante. Dado que el atacante está pagando por esto, se quedará rápidamente sin dinero al perder la tarifa que paga al minero, además de perder todo su dinero en el canal. Este ataque es poco probable y bastante poco atractivo, ya que es demasiado difícil y requiere un alto grado de colusión con un riesgo extremo. El modelo de riesgo de que ocurra este ataque es similar al de los mineros que se confabulan para realizar ataques de reorganización: extremadamente improbable con muchos mineros descoordinados
mineros.

10 Aumentos del tamaño del bloque y consenso

Si suponemos que existe una red de pagos descentralizada y que un usuario realiza un promedio de 3 transacciones en la cadena de bloques al año, Bitcoin podrá

dar servicio a más de 35 millones de usuarios con bloques de 1 MB en circunstancias ideales (suponiendo 2000 transacciones/MB, o 500 bytes/Tx). Esto es bastante limitado, y podría ser necesario un aumento del tamaño de los bloques para dar servicio a todos los usuarios de Bitcoin en el mundo. Un simple aumento del tamaño de los bloques supondría un hard fork, lo que significa que todos los nodos tendrían que actualizar sus carteras si desean participar en la red con bloques más grandes.

Aunque pueda parecer que este sistema mitigará los aumentos del tamaño de bloque a corto plazo, si alcanza una escala global, será necesario un aumento del tamaño de bloque a largo plazo. Se vuelve imperativo crear una herramienta confiable que ayude a prevenir el spam en la cadena de bloques diseñado para provocar que las transacciones expiren.

Para mitigar las vulnerabilidades del spam de bloqueo de tiempo, las reglas de consenso de los no mineros y los mineros también pueden diferir si las reglas de consenso de los mineros son más restrictivas. Los no mineros pueden aceptar bloques de más de 1 MB, mientras que los mineros pueden tener diferentes límites flexibles en los tamaños de bloque. Si el tamaño de un bloque supera ese límite, entonces otros mineros lo consideran un bloque inválido, pero no los no mineros. Los mineros solo construirán la cadena sobre bloques que sean válidos de acuerdo con el límite flexible acordado. Esto permite a los mineros acordar aumentar el límite de tamaño de bloque sin requerir hard forks frecuentes de los clientes, siempre y cuando la cantidad aumentada por los mineros no supere el límite estricto de los clientes. Esto mitiga el riesgo de que las transacciones caduquen masivamente de una sola vez. Todas las transacciones que no se canjeen a través de la Liquidación de Ejercicio (ES) pueden tener una tarifa muy alta asociada, y los mineros pueden utilizar una regla de consenso por la cual esas transacciones quedan exentas del límite flexible, lo que hace muy probable que las transacciones correctas ingresen a la cadena de bloques.

Cuando las transacciones se ven como circuitos y contratos en lugar de paquetes de transacciones, los riesgos de consenso pueden medirse por la cantidad de tiempo disponible para cubrir el conjunto de UTXO controlado por partes hostiles. En efecto, el límite superior del tamaño de UTXO está determinado por las comisiones de transacción y el valor mínimo estándar de salida de la transacción. Si los mineros de bitcoins tienen un mempool determinista que prioriza las transacciones respetando un orden de tiempo local «débil» de las transacciones, podría resultar extremadamente poco rentable y poco probable que un ataque tenga éxito. Cualquier ataque de tiempo por spam de transacciones mediante la difusión de una transacción de compromiso incorrecta supone un riesgo extremadamente alto para el atacante, ya que requiere una inmensa cantidad de bitcoins y todos los fondos comprometidos en esas transacciones se perderán si el atacante fracasa.

11 Casos de uso

Además de ayudar a escalar Bitcoin, hay muchos usos para las transacciones en la Red Lightning:

- Transacciones instantáneas. Con Lightning, las transacciones de Bitcoin son ahora casi instantáneas con cualquier parte. Es posible pagar una taza de café con un pago directo e irrevocable en milisegundos o segundos.

- Arbitraje de intercambio. Actualmente existe un incentivo para mantener fondos en las plataformas de intercambio a fin de estar preparados para grandes movimientos del mercado debido a los tiempos de confirmación de 3 a 6 bloques. Es posible que la plataforma de intercambio participe en esta red y que los clientes muevan sus fondos hacia y desde la plataforma para realizar órdenes casi al instante. Si la plataforma de intercambio no tiene una gran profundidad de mercado y se compromete a permitir únicamente órdenes limitadas cercanas a la parte superior del libro de órdenes, entonces el riesgo de robo de monedas se reduce considerablemente. La bolsa, en efecto, ya no tendría necesidad de una cartera de almacenamiento en frío. Esto podría reducir sustancialmente los robos y la necesidad de custodios externos de confianza.

- Micropagos. Las comisiones de la cadena de bloques de Bitcoin son demasiado altas para aceptar micropagos, especialmente con los valores más pequeños. Con este sistema, serían posibles micropagos casi instantáneos utilizando Bitcoin sin un custodio externo. Permitiría, por ejemplo, pagar por megabyte por el servicio de Internet o por artículo para leer un periódico.

- Contratos inteligentes financieros y depósitos en garantía. Los contratos financieros son especialmente sensibles al factor tiempo y plantean mayores exigencias a la capacidad computacional de la cadena de bloques. Al trasladar la gran mayoría de las transacciones sin necesidad de confianza fuera de la cadena, es posible establecer condiciones contractuales de transacciones altamente complejas sin que estas lleguen a pasar por la cadena de bloques.

- Pagos entre cadenas. Siempre que existan funciones hash similares entre las cadenas, es posible que las transacciones se enruten a través de múltiples cadenas con diferentes reglas de consenso. El remitente no tiene que confiar en las otras cadenas ni siquiera conocerlas, ni siquiera la cadena de destino. Del mismo modo, el destinatario no tiene que saber nada sobre la cadena del remitente ni sobre ninguna otra cadena. Lo único que le importa al receptor es un pago condicional al conocer un secreto en su cadena.

El pago puede ser enrutado por participantes en ambas cadenas en el salto. Por ejemplo, si Alice está en Bitcoin, Bob está tanto en Bitcoin como en X-Coin y Carol está en una hipotética X-Coin, Alice puede pagarle a Carol sin comprender las reglas de consenso de X-Coin.

12 Conclusión

La creación de una red de canales de micropagos permite la escalabilidad de Bitcoin, micropagos hasta el satoshi y transacciones casi instantáneas. Estos canales representan transacciones reales de Bitcoin, utilizando los códigos de operación de scripting de Bitcoin para permitir la transferencia de fondos sin riesgo de robo por parte de la contraparte, especialmente con mitigaciones de riesgo de mineros a largo plazo.

Si todas las transacciones que utilizan Bitcoin estuvieran en la cadena de bloques, para permitir que 7 mil millones de personas realizaran dos transacciones al día, se necesitarían bloques de 24 GB cada diez minutos en el mejor de los casos (suponiendo 250 bytes por transacción y 144 bloques al día). Realizar todas las transacciones de pago globales en la cadena de bloques hoy en día implica que los mineros tendrán que realizar una cantidad increíble de cálculos, lo que limitaría gravemente la escalabilidad de Bitcoin y los nodos completos a unos pocos procesadores centralizados.

Si todas las transacciones con Bitcoin se realizaran dentro de una red de canales de micropagos, para permitir que 7 mil millones de personas realizaran dos canales al año con transacciones ilimitadas dentro del canal, se necesitarían bloques de 133 MB (suponiendo 500 bytes por transacción y 52 560 bloques al año). Las computadoras de escritorio de la generación actual podrán ejecutar un nodo completo con los bloques antiguos eliminados en 2 TB de almacenamiento.

Con una red de canales de micropagos confirmados al instante cuyos pagos están sujetos a bloqueos temporales y salidas con bloqueo de hash, Bitcoin puede escalar a miles de millones de usuarios sin riesgo de custodia ni centralización de la cadena de bloques cuando las transacciones se realizan de forma segura fuera de la cadena utilizando scripts de Bitcoin, con la aplicación de la no cooperación mediante la difusión de transacciones multifirma firmadas en la cadena de bloques.

13 Agradecimientos

Los canales de micropagos han sido desarrollados por muchas partes y se han discutido en bitcointalk, la lista de correo de Bitcoin y el IRC. La cantidad de

colaboradores de esta idea es inmensa y se ha dedicado mucho esfuerzo a esta capacidad. Se ha puesto empeño en citar y encontrar ideas similares, sin embargo, no está ni mucho menos completa. En particular, hay muchas similitudes con una propuesta de Alex Akselrod que utiliza el hashlocking como método para restringir un canal de pago de tipo «hub-and-spoke».

Gracias a Peter Todd por corregir un error significativo en el script HTLC, así como por optimizar el tamaño del código de operación.

Gracias a Elizabeth Stark por la revisión y las correcciones.

Gracias a Rusty Russell por revisar este documento y por sus sugerencias para hacer el concepto más comprensible, así como por trabajar en una construcción que podría proporcionar una solución provisional antes de una corrección a largo plazo de la maleabilidad (que se describirá en una versión futura).

Apéndice A: Resolución de la maleabilidad de los contratos de compromiso ( )

Para crear estos contratos en Bitcoin sin un servicio de confianza de terceros, Bitcoin debe solucionar el problema de la maleabilidad de las transacciones. Si las transacciones pueden mutarse, entonces las firmas pueden invalidarse, lo que invalida las transacciones de reembolso y los bonos de compromiso. Esto crea una oportunidad para que actores hostiles lo utilicen como táctica de negociación para robar monedas, en efecto, un escenario de secuestro.

Para mitigar la maleabilidad, es necesario realizar un cambio de bifurcación suave (soft fork) en Bitcoin. Los clientes más antiguos seguirían funcionando, pero los mineros tendrían que actualizar. Bitcoin ha tenido varias bifurcaciones suaves en el pasado, incluyendo pay-to-script-hash (P2SH).

Para mitigar la maleabilidad, es necesario cambiar qué contenidos firman los participantes. Esto se logra creando nuevos tipos de sighash. Para adaptarse a este nuevo comportamiento, se necesita un nuevo tipo de P2SH o un nuevo OP CHECKSIG para que sea un soft fork en lugar de un hard fork.

Si se definiera un nuevo P2SH, utilizaría un script de salida diferente, como por ejemplo:

OP_DUP OP_ HASH160 <hash de 20 bytes> OP _EQUALVERIFY

Dado que esto siempre se resolverá como verdadero siempre que se proporcione un redeemScript válido,

todos los clientes existentes devolverán verdadero. Esto permite que el sistema de scripts construya nuevas reglas, incluyendo nuevas reglas de validación de firmas. Se necesitaría que existiera al menos un nuevo sighash.

SIGHASH_NOINPUT no firmaría ningún ID de transacción de entrada ni el índice. Al utilizar SIGHASH NOINPUT, se puede estar seguro de que la contraparte no puede invalidar árboles completos de transacciones encadenadas de posibles estados de contrato que se acordaron previamente, utilizando la mutación del ID de transacción. Con los nuevos indicadores de sighash, es posible gastar desde una transacción principal aunque el ID de la transacción haya cambiado, siempre y cuando el script se evalúe como verdadero (es decir, una firma válida).

SIGHASH_NOINPUT implica un riesgo significativo con la reutilización de direcciones, ya que puede funcionar con cualquier transacción en la que el sigScript devuelva como válido, por lo que múltiples transacciones con las mismas salidas son canjeables (siempre que los valores de salida sean menores).

Además, y no menos importante, SIGHASH_NOINPUT permite a los participantes firmar gastos de transacciones sin conocer las firmas de la transacción que se está gastando. Al resolver la maleabilidad de la manera anterior, dos partes pueden crear contratos y realizar transacciones de gasto sin que ninguna de las partes tenga la capacidad de transmitir esa transacción original en la cadena de bloques hasta que ambas partes estén de acuerdo. Con el nuevo tipo de sighash, los participantes pueden crear posibles estados de contrato y posibles condiciones de pago, y acordar todos los términos antes de que el contrato pueda pagarse, difundirse y ejecutarse sin necesidad de un tercero de confianza.

Sin SIGHASH NOINPUT, no se pueden crear salidas antes de que la transacción pueda ser financiada. Es como si no se pudiera llegar a ningún acuerdo sin comprometer fondos sin saber a qué se está comprometiendo. SIGHASH_NOINPUT permite crear redención para transacciones que aún no existen. En otras palabras, se pueden formar acuerdos antes de financiar la transacción si la salida es una transacción de multifirma 2 de 2.

Para usar SIGHASH_NOINPUT, se crea una transacción de financiación y aún no se firma. Esta transacción de financiación no necesita usar SIGHASH NOINPUT si se gasta desde una transacción que ya se ha ingresado en la cadena de bloques. Sin embargo, para gastar desde una transacción de financiación con una salida de firma múltiple 2 de 2 que aún no se ha firmado ni transmitido, es necesario usar SIGHASH NOINPUT.

En un próximo artículo de Rusty Russell se describirá otra solución provisional que utiliza OP CHECKSEQUENCEVERIFY

o un uso menos óptimo de QP CHECKLOCKTIMEVERIFY se describirá en un futuro artículo de Rusty Russell. Una versión actualizada de este artículo también incluirá estas construcciones.

Referencias

[1] Satoshi Nakamoto. Bitcoin: Un sistema de efectivo electrónico peer-to-peer. https://bitcoin.org/bitcoin.pdf, octubre de 2008.

[2] Manny Trillo. La prueba de estrés prepara a VisaNet para la época más maravillosa del año. http://www.visa.com/blogarchives/us/2013/10/10/«Una prueba de estrés prepara a Visanet para la época más maravillosa del año», octubre de 2013.

[3] Bitcoin Wiki. Contratos: Ejemplo 7: Pagos (micro) ajustados rápidamente a una parte predeterminada. https://en.bitcoin.it/wiki/Contracts#Example_7:_Rapidly-adjusted_.28micro.29payments_to_a_pre-determined_party.

[4] bitcoinj. Trabajar con canales de micropagos. https://bitcoinj.github.io/working-with-micropayments.

[5] Leslie Lamport. El Parlamento a tiempo parcial. *ACM Transactions on Computer Systems*, 21(2):133–169, mayo de 1998.

[6] Leslie Lamport. El tiempo, los relojes y el orden de los eventos en un sistema distribuido. *Communications of the ACM*, 21(7):558–565, julio de 1978.

[7] Alex Akselrod. Borrador. https://en.bitcoin.it/wiki/User:Aakselrod/Borrador, marzo de 2013.

[8] Alex Akselrod. ESCHATON. https://gist.github.com/aakselrod/9964667, abril de 2014.

[9] Peter Todd. Transacciones con comisiones casi nulas mediante micropagos de tipo «hub-and-spoke». http://sourceforge.net/p/bitcoin/mailman/message/33144746/, diciembre de 2014.

[10] C.J. Plooy. Combinación de Bitcoin y Ripple para crear una red de pagos rápida, escalable, descentralizada, anónima y de baja confianza. http://www.ultimatestunts.nl/bitcoin/ripple_bitcoin_ draft_2.pdf, enero de 2013.

[11] BitPay. Impulse. http://impulse.is/impulse.pdf, enero de 2015.

[12] Mark Friedenbach. BIP 0068: Reemplazo de transacciones impuesto por consenso y señalado a través de números de secuencia (tiempo de bloqueo relativo). https://github.com/bitcoin/bips/blob/master/bip-0068. mediawiki, mayo de 2015.

[13] Mark Friedenbach BtcDrak y Eric Lombrozo. BIP 0112: CHECK-SEQUENCEVERIFY. https://github.com/bitcoin/bips/blob/ master/bip-0112.mediawiki, agosto de 2015.

[14] Jonas Schnelli. ¿Qué hace OP CHECKSEQUENCEVERIFY? http://bitcoin.stackexchange.com/a/38846, julio de 2015.

[15] Greg Maxwell (nullc). reddit. https://www.reddit.com/r/Bitcoin/ comments/37fxqd/it_looks_like_blockstream_is_working_on_ the/crmr5p2, mayo de 2015.

[16] Gavin Andresen. BIP 0016: Pago a hash de script. https://github. com/bitcoin/bips/blob/master/bip-0016.mediawiki, enero de 2012.

[17] Pieter Wuille. BIP 0032: Billeteras determinísticas jerárquicas. https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki , febrero de 2012.

[18] Ilja Gerhardt y Timo Hanke. Direcciones de pago homomórficas y el protocolo Pay-to-Contract. http://arxiv.org/abs/1212.3257, diciembre de 2012.

[19] Nick Szabo. Formalización y protección de las relaciones en redes públicas. http://szabo.best.vwh.net/formalize.html, septiembre de 1997.