Discreet Log Contracts

Thaddeus Dryja

MIT Digital Currency Initiative

要旨

スマートコントラクト [1] は、Bitcoin などの暗号通貨システムでしばしば宣伝されている機能ですが、財務上での利用はまだ広がっていません。実装と採用の最大のハードルの 2 つは、スマートコントラクトのスケーラビリティと、通貨システム外のデータをスマートコントラクトに入れることの難しさです。これまで、契約のプライバシーは別の問題でした。 Discreet Log Contracts は、スケーラビリティとプライバシーの問題に対処し、外部データを提供するオラクルに必要な信頼を最小限に抑えるシステムです。この契約は、外部の観察者がトランザクションログ内で契約の存在を検出できないという点で”用心深い”です。 Discreet Log Contracts は離散 (*discrete*) 対数の知識に依存しています。これは良いことです。

モデル

契約プロセスには、アリス、ボブ、オリビアの 3 者が参加しています。アリスとボブは契約相手、オリビアはオラクルです。アリスとボブはお互いを信用しておらず、互いの法的な識別情報を知る必要はありませんが、認証されたチャネルを介して通信できる必要があり、互いを永続的に認識できる必要があります。 アリスとボブはオリビアから署名されたブロードキャストメッセージを受信できる必要があります。 オリビアはアリスとボブを意識する必要はなく、理想的には情報をブロードキャストすること以外の接点はありません。この情報は Bitcoin ネットワーク自体でブロードキャストが行われるほどコンパクトですが、これは必須ではありません。

DLC プロトコルは、多種多様な契約に使用することができ、当事者間の支払いが未来の公的に知られている数字に依存するほとんどのケースをカバーします。 この例では、アリスとボブが通貨の将来の契約を作成して実行します。 契約は水曜日に始まり、その時日本円は 1000satoshi の価値があるもの

とします。契約は金曜日に終了し、その時点で円は 1050satoshi の価値となるものとします。

コミットされた点 R の署名

Discreet Log Contracts はよく知られている Schnorr 署名 [2] を使用しますが、通常とは異なります。 通常 (の Schnorr 署名では)、ユーザはプライベートスカラー a を作成し、群の生成元 G を繰り返し追加して $A = aG$ を計算し、それを公開鍵 A として公開します。 メッセージに署名するとき、彼らは別のランダムなプライベートスカラー k を作成し、 $R = kG$ を計算し、次に以下を計算します。

$$s = k - h(m, R)a$$

ここで、 $h()$ はハッシュ関数であり、 m は署名すべきメッセージです。署名は (R, s) です。

検証者は与えられた (A, m, R, s) を使って以下を計算します。

$$sG = R - h(m, R)A$$

$$= kG - h(m, R)aG$$

(一方)Discreet Log Contracts では、ペア (A, R) を公開鍵と呼び、署名として s のみを残します。 方程式は同じですが、 R は署名の一部としてではなく公開鍵の一部として再分類されています。二重支出を防ぐための同様の構成を使う事例が、最近 [3] で公開されています。

オリビアは公開鍵 vG であるポイント V を公開します。オリビアはこの公開鍵を複数回使用しますが、 v を安全に保つ必要があります。 v は、異なる当事者が保持する異なる鍵で構成することができます。オリビアは別のポイント $R = kG$ も公開しています。k はランダムなノンスであり、R は 1 回限りの署名鍵です。これは、通常の Schnorr 署名で使用されている k と R と同じですが、 R 値は署名が計算される前にコミットされています。署名されるメッセージはまだ分かっていませんが、ノンスが選択され、ポイント R が公開されています。(これは、RFC6979 のような決定論的なノンスのスキームの使用を防ぎます)

オリビアは、 R に関連付けられた説明メタデータも公開しています。すべての R には、アセットタイプとそれに関連したクローズ時間をもっています。 この例では、 R は金曜日の市場終値での日本円の価格に関連付けられています。 R は 33 バイトの長さであるため、メタデータは R 自体よりも大きい可能性があります。

R は既知であるため、署名者が s を計算する前に、任意の m について sG を計算できます。

契約の作成

契約は、ブロックチェーン上に単一の出力として存在し、契約期間中に保持され、契約実行時に支払われるすべての資金が存在します。（この出力は多くの場合最適化できます。下記の「最適化」を参照してください）契約の設定が始まる前に、アリスとボブはまず互いを見つけ、契約の条件に同意する必要があります。アリスは金曜日の配達のために円を “買い”、ボブは円を “売り”ます。

契約を結ぶには、両方の当事者が共用のマルチシグアドレスに資金を持たなければいけません。この資金の確立は、ライトニングネットワークのチャネル資金調達のプロセスと実質的には同じです。アリスとボブは最初、ファンディングトランザクションの txid とインデックスに同意します。そのため、ネットワークにブロードキャストされてブロックで承認される前に、この出力を消費する子トランザクションを作成することがあります。

アリスとボブは、資金調達のアウトプットを消費する多数の取引からなる契約の作成に移ります。 出力は 1 回しか使用できないため、契約を構成するトランザクションの 1 つだけがブロックチェーンに現れます。 アリスとボブはどれが現れるかは知らないので、複数のトランザクションを両方とも署名して保存する必要があります。

各クロージングトランザクションは、クローズ時に異なる可能な価格に基づいて作成されます。この例では、価格は日本円の値段であり、satoshi 単位で表示されます。 アリスとボブは、それぞれ異なる終値でたくさんのトランザクションを作成します。

現在開発中の Lightning Network ソフトウェア [4] と同様に、当事者は契約状態に同意しますが、同じトランザクションのバリエーションを保持します。アリスは、2 つのアウトプットを持つ、ボブが署名したトランザクションを保持します。そのうち 1 つは、ボブに直接支払うもので、もう一つはアリスまたはボブのどちらかに支払うものです。ボブはその逆を保持しています：トランザクションはアリスによって署名され、アリスに直接支払うアウトプットと、ボブまたはアリスのいずれかが払い戻しするスクリプトに支払うアウトプットを持ちます。

Lightning Network では、これらのスクリプトを使用してペイメントチャネルの一貫性を維持し、いずれかの当事者が古い状態をブロードキャストした場合に、相手の当事者が両方の出力からすべての資金を取ることができるようにします。 アリスとボブはお互いに秘密を明らかにしません。 むしろ誰にでも秘密を明らかにするのはオリビアです。

契約の中のトランザクション

アリスとボブにはたくさんの署名済みの契約実行トランザクションがあり、それらは彼らのコンピュータに保存されています。これらの CET のそれぞれは、契約の資金調達のアウトポイントから支出し、2 つのアウトプットに送るものです。取引相手の公開鍵のハッシュ、および独自のスクリプトハッシュです。 このスクリプトは、Lightning Network チャネルで使用されているのと同じ OPCODE の連なりです。アリスは

$$Pub_{A_i} \vee (Pub_B \wedge TimeDelay)$$

を保持し、一方、ボブは

$$Pub_{B_i} \vee (Pub_A \wedge TimeDelay)$$

を保持します。

後者は、キー $Priv_{B_i}$ を持つユーザーがすぐに出力を費やし、キー $Priv_A$ を持つユーザーがしばらくして出力を費やすことを許可します。 Lightning Network では、これは前の有効なチャネル状態を強制的に取り消すために使用されますが、ここでは、ユーザーはオラクルが間接的に正しい状態として承認したトランザクションだけをブロードキャストすることを強制しています。

アリスは、

$$Pub_{Ai} = Pub_{Alice} + s_iG$$

に送信するトランザクション $TX_1...TX_n$ を保持します。ここで、

$$s_iG = R - h(i, R)V$$

です。 Pub_B は単にボブの公開鍵です。

ボブはその逆を持ちます。アリスの公開鍵 Pub_A に時間遅れで送り、

$$Pub_{Bi} = Pub_{Bob} + s_iG$$

を時間遅れなしで送るトランザクションです。

検証者はポイント s_iG を計算することができますが、任意の与えられた i に対して s_i が何であるかは知らない。 オリビアが署名すると、彼女はアリスとボブがすでに計算したポイントの離散対数を明らかにします。

オラクルによる署名

オリビアの仕事は簡単です：金曜日の市場終値まで待って、円の終値を観察し、その数値を事前にコミットしたノンスで署名します。

オラクルは、m を観測した価格 (この例では、1 円あたり 1050satoshi) に設定します。 オラクルは

$$s = k - h(1050, R)v$$

を計算します。 ここで、円の価格は 1 円あたり 1050satoshi に上昇したので、 $m = 1050$ です (実際には、メッセージ m はハッシュ出力ですが、ここでは明確にするために、署名された生の数値として残しています)。

これは、アリスとボブが以前にトランザクションの鍵を導出するために使用した s_{1050} と同じものを示しています。

$$s_{1050}G = R - h(1050, R)V$$

オラクルが s_{1050} を明らかにしてしまうと、 $Pub_{B_{1050}}$ のプライベートスカラーは $b + s_{1050}$ なのでボブに完全に知られます。 アリスも同様に $Pub_{A_{1050}}$ のスカラーを知ります。

契約の実行

アリスとボブの両方は TX_{1050} をブロードキャストすることで、正しい状態で契約を一方的にクローズすることができるようになりました。ブロードキャスト直後に、彼らは sighash 出力 (Script Hash Output の誤り?) を消費して、完全に制御できるアドレスにそれを送ります。この結果として、2 つのオンチェーントランザクションが行われるため、契約実行トランザクションと同じ金額で両方の当事者に対してスクリプトハッシュ出力に直接送信する新しいトランザクション TX_{gg} を作成することに同意する方が効率的です。また、スクリプトハッシュ出力をすぐに使うよう注意する必要があります。 遅延期間よりも長く待った場合、それを費やす前に、相手方はその資金を請求することができます。

いずれかの当事者が実行トランザクションを早期にブロードキャストしたり、間違ったトランザクション（ブロードキャスト TX_{950} など）をブロードキャストすると、スクリプトハッシュ出力を費やすことができなくなります。彼らの取引相手は、遅延時間が経過した後、スクリプトハッシュ出力を要求することができます。 彼らは P2PKH アウトプットへの単独アクセス権を持っているので、コントラクトの価値全体を請求することができます。このようにして、契約の規則に違反すると、相手方との契約におけるすべての価値が失われます。

信頼されたオラクルのリスク

オラクルであるオリビアは、価格を誤って報告する可能性があります。 オラクルが誤って報告すると、システムのすべてのユーザーがエラーを識別し、オラクルの使用を停止できます。(契約において取引相手の役割を担い、真の結果にかかわらず賭けを “勝利” させるために) オリビアが 2 つの異なる価格を公に報告しようとすると、彼女は自身の秘密鍵を漏らすことになります。これは、彼らが 2 重に報告しようとした場合の特定の契約に対する k の値と同様です。オリビア自身が契約の相手（例えば、アリスがオリビア）の場合、彼女は秘密鍵を公開せずに任意の方法で契約を実行させることができます。これは、所望の結果である TX2 を放送し、次にオリビア自身が契約の相手（例えば、アリスがオリビア）の場合、彼女は秘密鍵を公開せずに任意の方法で契約を実行させることができます。これは、所望の結果である TX_2 をブロードキャストし、次に

$$Priv_{A_2} = Priv_A + s_2$$

を計算し、 Pub_{A_2} への出力に署名することによって可能です。オリビアは s_2 を公に明かすことなく、秘密鍵 v を保持したまま署名します。これは検出可能です。不正な当事者ボブは、詐欺のコンパクトな証明を提供して、他のすべてのユーザーがオリビアのコミットメントと署名の使用を止めさせることができます。 オリビアは参加している契約を欺く機会が 1 回ありますが、すぐにすべてのユーザーの信頼を失うでしょう。

オリビアは、 R 値を公開した後でも、値段や出来事を報告するために署名するよりも前に消えるかもしれません。 この可能性に対処するために、契約は、 s の値が到着すると予想されてから数日後にいずれかの当事者によってブロードキャストされる払い戻しのトランザクションを有することができます。 払い戻しトランザクションによる返金は、デフォルトでは当事者に最初の預かり金を返却します。

複数のオラクルを契約の相手方が使用することができます。 2 つのオラクル署名が必要な場合、取引相手は単純にオラクルのポイント sG を追加してから公開鍵に追加します。 これは、不正確なオラクルのリスクを実質的に減らすことができます。ただし、正確な合意が得られない可能性のあるデータとなるリスクがあるということを犠牲にします。 オラクル 1 が 1050 と報告し、オラクル 2 が 1049 と報告した場合、実行トランザクションは安全に使用されず、事前に作っておいたタイムアウト・トランザクションが最終的に有効になり、両者に払い戻しとなる可能性があります。

もう 1 つのリスクは、契約の全部または実質的にすべてを失った当事者が、資金の正当な所有者を遅らせるために無効な契約実行のトランザクションをブロードキャストする可能性があることです。最終的に、資金は正しい当事

者によって回収されますが、資金にアクセスする前にタイムアウト期間を待たなければなりません。 これは迷惑なことですが、契約に超過担保を課すことによって緩和することができるため、実際にはまれであると思われます。

最適化

いくつかの最適化は、計算とデータの要件を減らすことができます。システムを実装する際に間違いなく多くの最適化が見つかるはずですが、いくつかの基本的なアイデアがここに列挙します。

R 値の基数と指数

アリスとボブが契約を結ぶと、オリビアが署名する可能性のあるすべてのメッセージ m_i の署名を送信して保存する必要があります。アセットには多くの可能な価格があり、数千もの署名を検証して保存するかもしれません。アリスとボブに対する配当が異なるトランザクションは有用であり、精度を提供します。同じ支払い額となるにもかかわらず予想しておかなければいけない、多くの異なる価格 m1、m2、...、mq が存在する可能性がある。おそらく、“ノックイン (knock-in)”と “ノックアウト (knock-out)”価格が、それを下回ったり、それを上回ると、一人の当事者が契約のすべての金額を受け取ることになります。たとえば、円の価格が 10satoshi 以下になった場合、アリスとボブは、最終価格が 4satoshi または 5satoshi であるかどうかにかかわらず、ボブがすべてのお金を得ることに同意します。同様に、アリスは、6,000 か 77,000 かにかかわらず、価格が 5,000 を超えるとすべてのお金を得ます。

アリスとボブは、価格がさまざまなオーダーに及ぶ可能性があるため、こうした極端な範囲でのトランザクションを少なくすることによって、ノックインとノックアウト価格を最適化することができます。 これは、オリビアが 1 つの R 値の代わりに $R_{mantissa}$ と $R_{exponent}$ という 2 つの R 値をコミットする場合に可能となります。 次にオリビアは、価格を表す 2 つのメッセージに署名することを約束します。“1050” に署名する代わりに、 $R_{mantissa}$ を使用して “.050”に署名し、 $R_{exponent}$ を使用して 3 に署名します。 小数点の基数と仮数部の先頭の 1 は省略されます。 $1.050 * 10^3 = 1050$

アリスとボブは、 $s_{mantissa}G$ と $s_{exponent}G$ 点の合計を計算することによって同じようにトランザクションを構築できます。オリビアが仮数部と基数のメッセージのペアに署名するとき、彼女は s 値を明らかにし、必要なスカラーを得るためにそれを一緒に加算します。最適化は、指数が 4 の場合、アリスとボブは精度を必要としないという事実にあります。 $R_{exponent}$ が 4,5,6 などのときには $R_{mantissa}$ を無視し、可能性の低い結果を扱うトランザクションのほんの一握りにしか署名しません。

これをさらに 3 つまたは 4 つの R ポイントに拡張することで、より詳細な粒度を可能にすることができます。また、10 は累乗のための最適な基底ではありません。 おそらく 2 が良い選択です。

チャネル内の契約

コントラクトごとに個別のアウトポイントと p2wsh アドレスがある場合、すべての契約はブロックチェーン上の領域を占有します。代わりに、既存の Lightning Network チャネル内で契約を作成することができます。チャネルのファンディングトランザクションの出力を消費しているコミットメントトランザクションの中に存在する直接および HTLC 出力に加えて、独自のチャネルまたはこの場合は契約である 2of2 のマルチシグ出力が複数含めることができます。このようにして、両方の当事者がオンラインで契約の結果を承認した場合、契約の出力をチャネルから削除し、両方の残高を更新して契約で生じた変更を反映させることができます。いずれかのパーティが非協力的である場合、相手方は親チャンネルを閉じて、オラクルが提供する s の値を使用してすぐに契約を閉じることができます。

関連研究

更改

先物取引は有用ですが、通常、取引者は取引終了時刻前にポジションに入ったりポジションを終了したりしたいと考えています。両当事者がオンラインで同意する場合、将来の価格ではなく現在の価格に基づいて契約の終了を交渉することができます。一方の当事者は、かなり合理的に、契約を短くすることに同意しないかもしれません。ボブが早期に離れることを望んでいるが、アリスが契約を最後まで見たい場合、ボブはボブの保有しているポジションを取ろうとしている別のユーザー、キャロルとポジションを入れ替えることができるかもしれません。ボブは、アリスの契約について、アリスの取引相手の公開鍵を除いては何も変えないようにするためにアリスに手数料を請求するかもしれません。これは、3 者すべてが同時にオンラインになることを必要とするようであり、興味深い研究分野と思われます。

分散型マッチング

アリスとボブが契約を結ぶ前に、互いを見つける必要があります。はじめのうちは、これは集中型のマッチングエンジンで行うことができますが、これはユーザーの資金を保護しません。公平な分散されたピアの発見とマッチングは、将来の研究のための別の話題です

結論

ユーザーがさまざまな資産の先物取引を慎重に契約できるようにして、正しい価格に署名するだけのオラクルを信頼することによって、Discreet Log Contracts は、Bitcoin やその他の暗号通貨ネットワークのユースケースを強化する潜在的可能性を秘めています。トランザクションは Lightning Network のトランザクションと同じように見えるため、ネットワーク全体で DLC の総量を見積もることは困難であり、グローバルネットワークに過度の課税を課すことなく広範で複雑なスマートコントラクトを実現する必要があります。

References

[1] Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997.

[2] Claus Schnorr. Efficient identification and signatures for smart cards. pages 239–252, 1990.

[3] Cristina Prez-Sol, Sergi Delgado-Segura, Guillermo Navarro-Arribas, and Jordi Herrera-Joancomart. Double-spending prevention for bitcoin zero-confirmation transactions. Cryptology ePrint Archive, Report 2017/394, 2017. http://eprint.iacr.org/2017/394.

[4] Thaddeus Dryja Joseph Poon. The bitcoin lightning network: Scalable off-chain instant payments. Technical Report (draft), 2015.