Bulletproofs の理論

Bulletproofs とは、トラステッドセットアップを必要としない非対話型ゼロ知識証明スキームである.主な用途はレンジプルーフであり、秘密の値が所定の範囲内にあることを証明するために用いられる.

Bulletproofsの全体の流れ

1. コミットメント生成**:**
    - Pedersenコミットメントで秘密値 vv を隠蔽
2. ビット分解**:**
    - v をビットに分解し、aL と aRを設定
3. 内積論証**:**
    - 再帰的圧縮により内積の正しさを証明
4. 非対話性**:**
    - Fiat-Shamirヒューリスティックで証明を非対話型に

1.Pedersen コミットメントの性質

- 目的:
  - 秘密値v を隠蔽しつつ固定する
- 仕組み::
- $C = v \cdot G + r \cdot H$　(←コミットメント)
  - v: 証明対象の秘密値（例: 取引額）
  - r: ランダムなブラインディング係数
  - G, H: 公開されたジェネレーター
    ※ 特に H は G との間に既知の離散対数関係が存在しないように設定
- 完全な隠蔽性 :コミットメント C から、v や r を復元することは計算上不可能である（情報理論的隠蔽性）.
- 束縛性 :離散対数問題の困難性に基づき、後から別の値 v' に変更できないことが保証

2.内積論証

1. ビット分解と補助ベクトルの定義

- ビット分解:
  秘密値 v を2進数で分解して、ビット列ベクトル
  $\mathbf{a} = (a_1, a_2, \ldots, a_n)$を生成　※各 $a_i \in \{0, 1\}$
  よって$v = \langle \mathbf{a}, \mathbf{w} \rangle = \sum_{i=1}^{n} w_i a_i$,と表せる
  通常、重みベクトルは $\mathbf{w} = (2^0, 2^1, \ldots, 2^{n-1})$

- 補助ベクトル:
  ビット性質を確認するために、
  $\mathbf{b} = \mathbf{a} - \mathbf{1} = (a_1 - 1,\ a_2 - 1,\ \ldots,\ a_n - 1)$と定義
  これにより、各要素について
  $a_i(a_i - 1) = 0$となり、各 a_i が 0 か 1 であることが保証

2. 内積論証の概要

Bulletproofs では、内積論証を用いて、以下の2点を同時に証明する

1. レンジ条件の証明:
   秘密値 v が、ビット分解 $\mathbf{a}$ によって正しく表現される、すなわち
   $$v = \langle \mathbf{a}, \mathbf{w} \rangle.$$
2. ビット性の確認:
   各 a_i が 0 または 1 であること、すなわち
   $$\langle \mathbf{a}, \mathbf{b} \rangle = \sum_{i=1}^{n} a_i(a_i - 1) = 0.$$

3. 内積論証の再帰的プロトコル

内積論証は、長いベクトル $\mathbf{a}$ と $\mathbf{b}$ の内積が 0 である（または任意の値 c になる）ことを、対話的なプロトコル（Fiat–Shamir ヒューリスティックにより非対話型化）で短い証明に圧縮する仕組みである

以下は、再帰的な圧縮ステップの概要である

1. ベクトルの分割:
   ベクトル $\mathbf{a}$ と $\mathbf{b}$ をそれぞれ左右に分割する
   $$\mathbf{a} = (\mathbf{a_L}, \mathbf{a_R}), \quad \mathbf{b} = (\mathbf{b_L}, \mathbf{b_R}).$$
2. 補助コミットメントの生成:
   分割したベクトルに対して、以下のような値 L と R を計算する
   $$L = \mathrm{Com}\big(\mathbf{a_L}, \mathbf{b_R}\big), \quad R = \mathrm{Com}\big(\mathbf{a_R}, \mathbf{b_L}\big).$$
   ここで $\mathrm{Com}(\cdot,\cdot)$ は、適切なペダースコミットメントなどを意味する

3. チャレンジ値の生成:
   非対話型にするため、Fiat–Shamir ヒューリスティックにより
   $x = H(L, R)$
   とチャレンジ値 x を計算する

4. ベクトルの圧縮:
   チャレンジ x を用いて、新しいベクトルを次のように定義する

$$\mathbf{a}' = \mathbf{a_L} + x \cdot \mathbf{a_R}, \quad \mathbf{b}' = \mathbf{b_L} + x^{-1} \cdot \mathbf{b_R}.$$

   この操作により、内積は保存される：

$$\langle \mathbf{a}', \mathbf{b}' \rangle = \langle \mathbf{a_L}, \mathbf{b_L} \rangle + \langle \mathbf{a_R}, \mathbf{b_R} \rangle.$$

5. 再帰:
   この圧縮ステップを、ベクトルの次元が 1 になるまで繰り返す
   最終的には、1 つのスカラー値に圧縮され、その値が内積の結果（ここでは 0）であることを検証する

Fiat–Shamir ヒューリスティックと非対話型化

- 元々は対話型プロトコルであるが、 チャレンジ値をハッシュ関数で生成することで、非対話型ゼロ知識証明（NIZK）に変換される.
- ハッシュ関数により、検証者のチャレンジがランダムに生成され、対話なしでも安全な証明が可能となる.
- これにより、証明の再現性および検証の自動化が実現される

3. 全体の証明と検証

- 証明の生成:
  証明は、各再帰ラウンドで生成される補助コミットメント
  $(L_1, R_1), (L_2, R_2), \ldots, (L_k, R_k)$ と、最終的なスカラー値 a', b' から構成される

- 検証:
  検証者は、公開情報と証明から各ラウンドの $x_i = H(L_i, R_i)$ を再計算し、圧縮されたコミットメントが正しく更新されているか、

$$P \stackrel{?}{=} \left(\prod_{i=1}^{k} L_i^{x_i^2} \cdot R_i^{x_i^{-2}} \right) \cdot g^{a'} h^{b'},$$

  そして内積の最終検証 $a' \cdot b' = c$ を確認する

トラステッドセットアップ※1が不要な理由

- Bulletproofs では、ジェネレーター G と H を「nothing-up-my-sleeve」方式で生成する.
- この方法により、特定のトラステッドセットアップを必要としません.
  - 例えば、H は Gからハッシュ関数を使って導出されるなど、信頼できる初期設定が不要な設計となっている.

※1トラステッドセットアップ:システム全体で使われる公開パラメータを生成する初期プロセス

Bulletproofs の応用例

- レンジプルーフ
  - 秘密値が $0 \leq v < 2^n$ の範囲にあることを証明する.
  - Confidential Transactions で取引額を隠しながら正当性を保証する.
- 複数証明の集約
  - 複数のレンジプルーフを $O(\log m)$ の追加サイズで集約可能である.
- シャッフラーへの応用
  - 各参加者のコインは Pedersen コミットメントで表現される.
  - 各コミットメントに対してレンジプルーフが付与され、総額が変わらないことが検証される.
  - これにより、混合後も不正なコインの生成や改竄がないことがゼロ知識で保証されるのである.