Bitcoin で CTF

Bitcoin based CTF

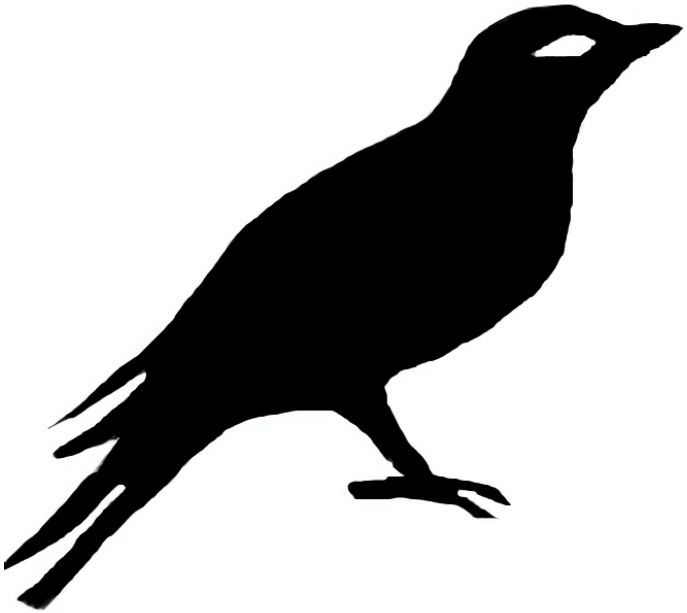
吉村 優
Hikaru Yoshimura

リクルートマーケティングパートナーズ
hikaru_yoshimura@r.recruit.co.jp
yyu@mental.poker

March 27, 2018

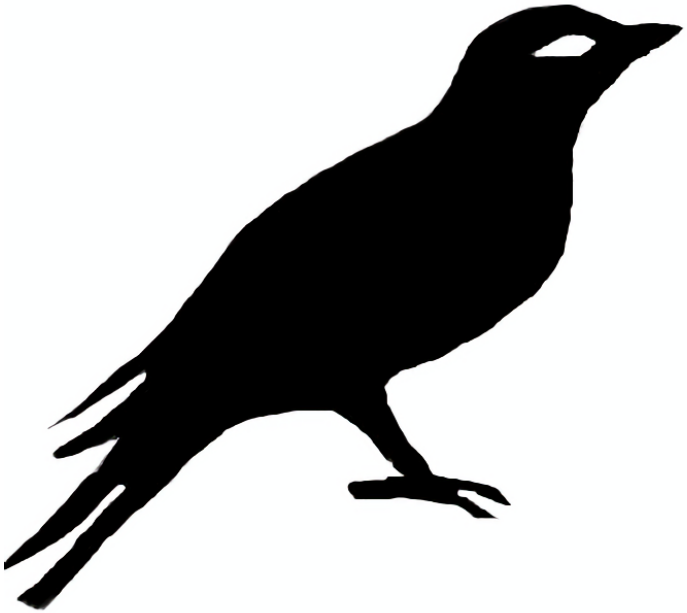
(Git Commit ID: 8c1ece8)

自己紹介



Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

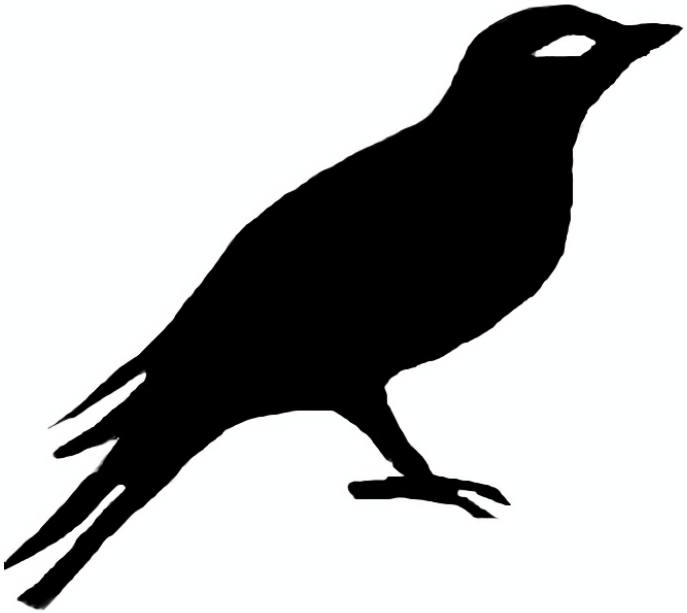
自己紹介



- 筑波大学 情報科学類卒（学士）

Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

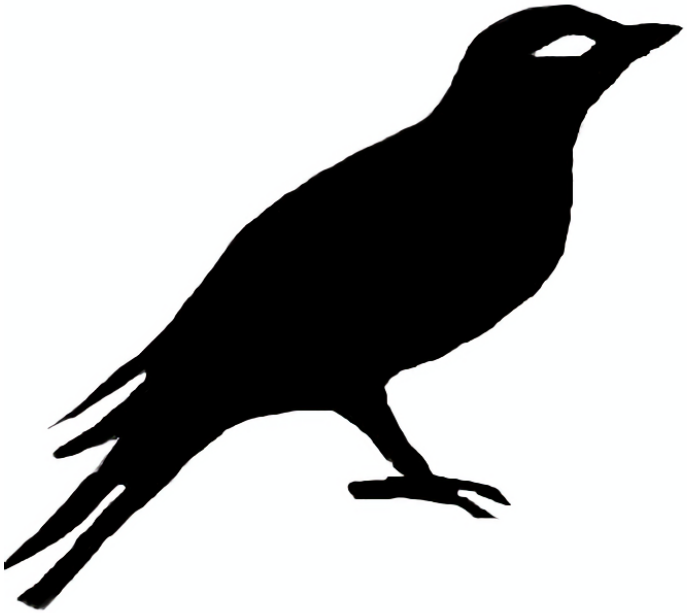
自己紹介



- 筑波大学 情報科学類卒（学士）
- リクルートマーケティングパートナーズ（中途）

Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

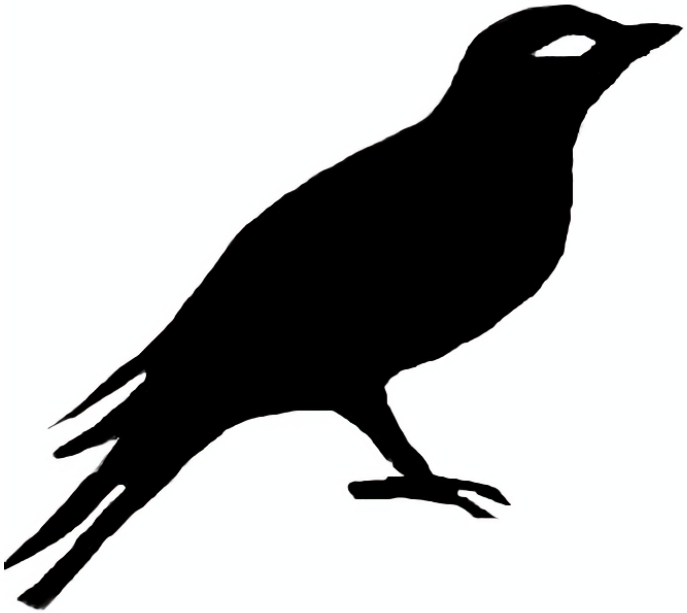
自己紹介



- 筑波大学 情報科学類卒（学士）
- リクルートマーケティングパートナーズ（中途）
- NB 本部 プロディベ部 英語学習 G

Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

自己紹介



- 筑波大学 情報科学類卒（学士）
- リクルートマーケティングパートナーズ（中途）
- NB 本部 プロディベ部 英語学習 G
- CTF Team: urandom
  - https://urandom.team/

Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

自己紹介

- SECCON 2014 オンライン予選 優勝
- IWSEC Cup 2015 Gold Prize
- SECCON 2015 x CEDEC CHALLENGE ゲームクラッキング＆チートチャレンジ 優勝
- サイバーコロッセオ x SECCON 2016 準優勝

Twitter	@_yyu_
Qiita	yyu
GitHub	y-yu

CTF とは？

CTF とは？

- “Capture The Flag” の略でセキュリティ系の競技のこと

CTF とは？

- “Capture The Flag” の略でセキュリティ系の競技のこと
- ただし、この発表では *jeopardy* 形式を前提とする

CTF とは？

- “Capture The Flag” の略でセキュリティ系の競技のこと
- ただし、この発表では *jeopardy* 形式を前提とする

CTF (jeopardy)

- 脆弱性を攻撃するなどして**フラッグワード**と呼ばれる文字列を得る
- フラッグワードによって相応のポイントが得られ、最終的に最もポイントを獲得したチームが勝利する

CTF とは？

- “Capture The Flag” の略でセキュリティ系の競技のこと
- ただし、この発表では *jeopardy* 形式を前提とする

CTF (jeopardy)

- 脆弱性を攻撃するなどして**フラッグワード**と呼ばれる文字列を得る
- フラッグワードによって相応のポイントが得られ、最終的に最もポイントを獲得したチームが勝利する

- 国際的な CTF は賞金がもらえる

  Google CTF 2017 (Final)　1 位に 13,337 USD（約 150 万円）
  Codegate CTF Finals 2017　1 位に 30,000,000 KRW（約 307 万円）

従来の CTF の課題

従来の CTF の課題

- 勝利したチームが賞金を本当に得られるのかが不明である
  - たとえば脆弱性の情報が欲しいので CTF を開催する

従来の CTF の課題

- 勝利したチームが賞金を本当に得られるのかが不明である
  - たとえば脆弱性の情報が欲しいので CTF を開催する
- 問題を解いたチームが時間内に解答したという証拠がない
  - Write-up を書くという手もあるが、CTF が終ってから解いたという可能性もある

提案するCTFの特徴

提案するCTFの特徴

- 正しいフラッグワードを提出した場合、参加者は直ちにその問題に対応する賞金が得られる

提案する CTF の特徴

- 正しいフラッグワードを提出した場合、参加者は直ちにその問題に対応する賞金が得られる
  - 従来の CTF ではポイントの多い順に賞金が決まるが、提案する CTF では賞金が多い順に順位が決まる

提案する CTF の特徴

- 正しいフラッグワードを提出した場合、参加者は直ちにその問題に対応する賞金が得られる
  - 従来の CTF ではポイントの多い順に賞金が決まるが、提案する CTF では賞金が多い順に順位が決まる
- 賞金は全て Bitcoin で支払われる

提案する CTF の特徴

- 正しいフラッグワードを提出した場合、参加者は直ちにその問題に対応する賞金が得られる
  - 従来の CTF ではポイントの多い順に賞金が決まるが、提案する CTF では賞金が多い順に順位が決まる
- 賞金は全て Bitcoin で支払われる
- 基本的に問題を最初に解答したチーム以外には賞金が支払われない

提案するCTFの特徴

- 誰もが閲覧できるBitcoinのブロックチェーンを利用するので、どのチームが問題を時間内に解答したかが誰にとっても明らかである

提案するCTFの特徴

- 誰もが閲覧できるBitcoinのブロックチェーンを利用するので、どのチームが問題を時間内に解答したかが誰にとっても明らかである
- 従来のCTFとは異なり開始時刻や終了時刻がJSTなどではなく、Bitcoinのブロックチェーンの長さがnに達した時に開始であり、m $(m > n)$に達した時に終了である
  - Bitcoinのブロックチェーンは1ブロックの作成に約10分必要である。よって、このCTFの制限時間は約10分刻みで指定できるため、実用上の問題はないと考えられる

Bitcoin とスクリプト

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

トランザクション ひとつの送金を表し、送金者が作成する

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

トランザクション ひとつの送金を表し、送金者が作成する

ブロック トランザクションを複数集めたものを表し、マイナーが作成する。ある i 番目のブロックは $i-1$ 番目のブロックの情報を参照する

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

トランザクション ひとつの送金を表し、送金者が作成する

ブロック トランザクションを複数集めたものを表し、マイナーが作成する。ある i 番目のブロックは $i-1$ 番目のブロックの情報を参照する

ブロックチェーン ブロックを複数集めたもの

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

トランザクション ひとつの送金を表し、送金者が作成する

ブロック トランザクションを複数集めたものを表し、マイナーが作成する。ある i 番目のブロックは $i-1$ 番目のブロックの情報を参照する

ブロックチェーン ブロックを複数集めたもの

- ハッシュ値の計算によって “長い” ブロックチェーンを作ることを難しくし、ブロックをブロックチェーンへ入れたマイナーに報酬を与えている

Bitcoin とスクリプト

- まずは Bitcoin に関する用語を整理する

トランザクション ひとつの送金を表し、送金者が作成する

ブロック トランザクションを複数集めたものを表し、マイナーが作成する。ある i 番目のブロックは $i-1$ 番目のブロックの情報を参照する

ブロックチェーン ブロックを複数集めたもの

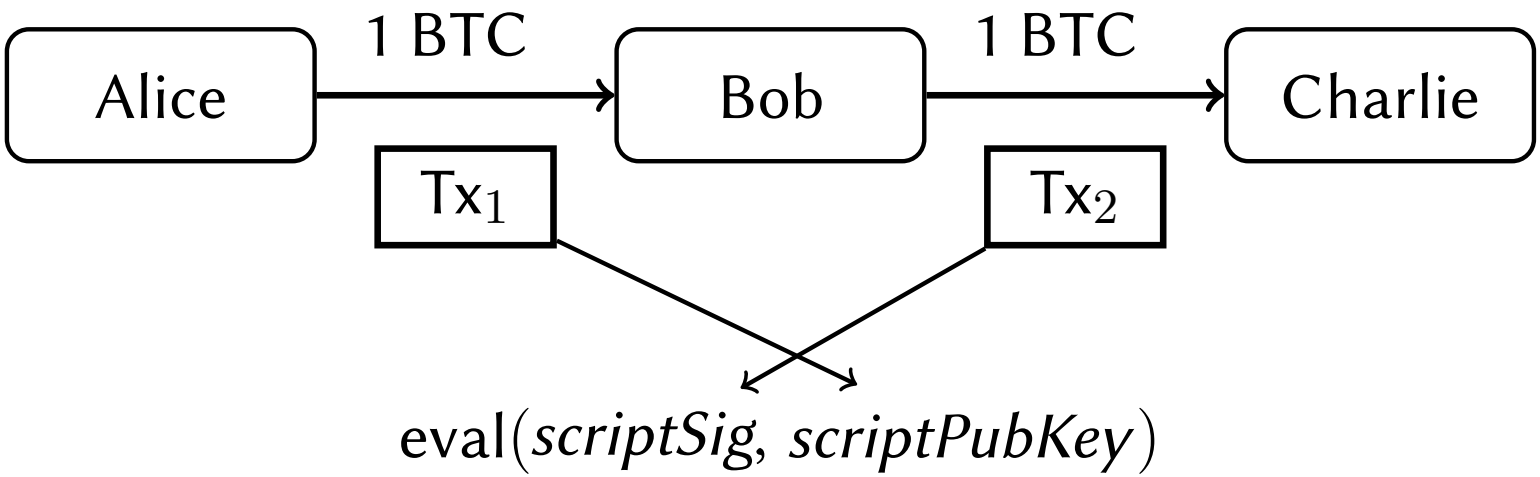
- ハッシュ値の計算によって “長い” ブロックチェーンを作ることを難しくし、ブロックをブロックチェーンへ入れたマイナーに報酬を与えている
- マイナーはどのようにブロックに含めるべきトランザクションを検証しているのか？
  - 残高よりも大きいお金のトランザクションといった不正なトランザクションをどのように検出する？

Bitcoin とスクリプト

- Bitcoin のトランザクションには *scriptSig* と *scriptPubKey* という 2 つの場所に**スクリプト**と呼ばれる非チューリング完全なスタックベースのプログラムを書き込める

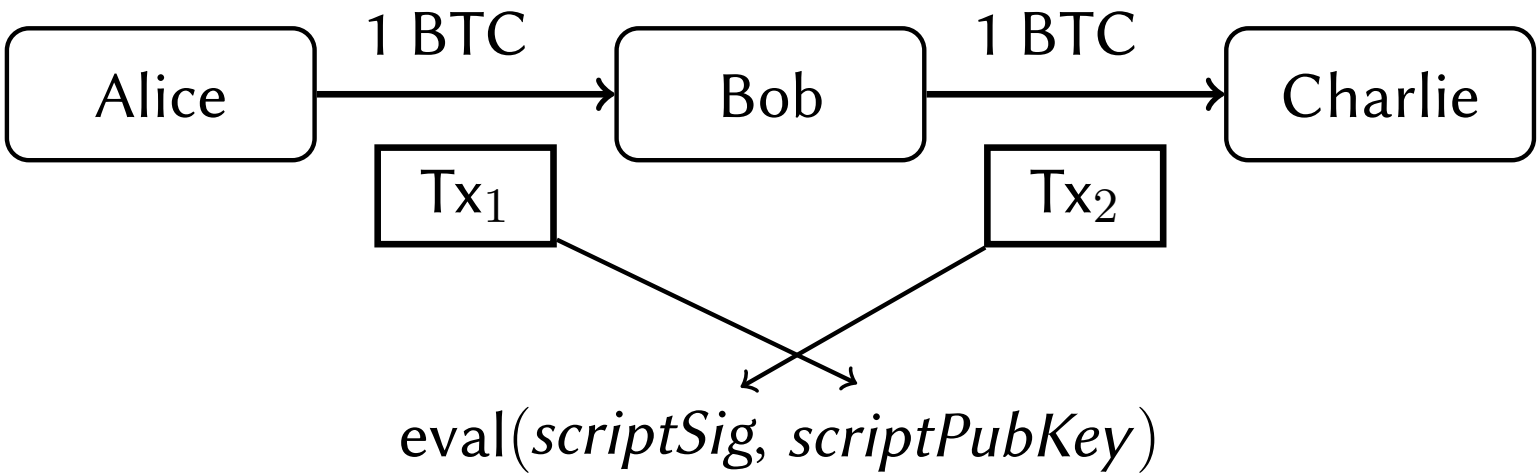
Bitcoin とスクリプト

- Bitcoin のトランザクションには *scriptSig* と *scriptPubKey* という 2 つの場所に**スクリプト**と呼ばれる非チューリング完全なスタックベースのプログラムを書き込める
- マイナーは次のようにトランザクションのスクリプトを実行する



Bitcoin とスクリプト

- Bitcoin のトランザクションには *scriptSig* と *scriptPubKey* という 2 つの場所に**スクリプト**と呼ばれる非チューリング完全なスタックベースのプログラムを書き込める
- マイナーは次のようにトランザクションのスクリプトを実行する



- マイナーは eval の結果が $\boxed{0}$ 以外ならトランザクション Tx_2 を受理し、$\boxed{0}$ ならば拒否する

CTF の開催前

CTF の開催前

1. 参加チーム T_i は運営に Bitcoin の公開鍵 $\mathcal{T}_i$ を提出する

CTF の開催前

1. 参加チーム T_i は運営に Bitcoin の公開鍵 $\mathcal{T}_i$ を提出する
2. 運営は参加登録をブロックチェーンの長さが n となる前に締め切る

CTF の開催前

1. 参加チーム T_i は運営に Bitcoin の公開鍵 $\mathcal{T}_i$ を提出する
2. 運営は参加登録をブロックチェーンの長さが n となる前に締め切る
3. 問題 j に対応するフラッグワードを F_j として、またこの問題 j を解答した際に得られる賞金を B_j BTC とし、さらに、$ans_{ij} := H\left(H(F_j \,||\, i)\right)$ とする運営は問題 j に対して、後述のような *scriptPubKey* を持つ B_j BTC のトランザクション Tx_j を作成する
   - H はハッシュ関数 SHA-256 であり、また $||$ は文字列の結合である

CTF の開催前

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   OP_DUP
   2
   OP_EQUAL
   ⋮
OP_ENDIF
OP_CHECKSIG
```

CTF の開催前

4. 運営は全てのトランザクション Tx_j をブロックチェーンへ送信する
   - ただし、トランザクションをブロックチェーンへ送信する時間はあらかじめ全参加チームに告知する

CTF の開催前

4. 運営は全てのトランザクション Tx_j をブロックチェーンへ送信する
   - ただし、トランザクションをブロックチェーンへ送信する時間はあらかじめ全参加チームに告知する
5. 運営は全てのトランザクション Tx_j のトランザクション ID を CTF の問題ページに記載する

CTF の実施中

CTF の実施中

1. 今、チーム T_i が問題 j のフラッグワード F_j を得たとする。チーム T_i はトランザクション Tx_j を入力に持ち、次のような *scriptSig* を持つトランザクション Tx_{ij} を作成する。ただし、$h_{ij} := H(F_j \mid\mid i)$ であり、S_i はチーム T_i の Bitcoin の公開鍵 $\mathcal{T}_i$ に対応する秘密鍵によって作成された署名である

Listing: Tx_{ij} の *scriptSig*

S_i
h_{ij}
i

CTF の実施中

1. 今、チーム T_i が問題 j のフラッグワード F_j を得たとする。チーム T_i はトランザクション Tx_j を入力に持ち、次のような *scriptSig* を持つトランザクション Tx_{ij} を作成する。ただし、$h_{ij} := H(F_j \,||\, i)$ であり、S_i はチーム T_i の Bitcoin の公開鍵 $\mathcal{T}_i$ に対応する秘密鍵によって作成された署名である

Listing: Tx_{ij} の *scriptSig*

S_i
h_{ij}
i

2. チーム T_i はトランザクション Tx_{ij} を Bitcoin のブロックチェーンへ送信する

CTF の実施中

1. 今、チーム T_i が問題 j のフラッグワード F_j を得たとする。チーム T_i はトランザクション Tx_j を入力に持ち、次のような *scriptSig* を持つトランザクション Tx_{ij} を作成する。ただし、$h_{ij} := H(F_j \,||\, i)$ であり、S_i はチーム T_i の Bitcoin の公開鍵 $\mathcal{T}_i$ に対応する秘密鍵によって作成された署名である

Listing: Tx_{ij} の *scriptSig*

S_i
h_{ij}
i

2. チーム T_i はトランザクション Tx_{ij} を Bitcoin のブロックチェーンへ送信する
3. 問題 j がまだ解かれていないかつフラッグワードが正しい場合、チーム T_i は B_j BTC を獲得する

CTF の終了後

CTF の終了後

1. 運営は Bitcoin のブロックチェーンの長さが $m+3$ に達したときのブロックチェーンについて、チームに対応する Bitcoin の公開鍵を用いて問題を解答することで獲得した Bitcoin の量を計測する
   - $m+3$ は目安なので、$m+1$ や $m+6$ などでもよい

CTF の終了後

1. 運営は Bitcoin のブロックチェーンの長さが $m+3$ に達したときのブロックチェーンについて、チームに対応する Bitcoin の公開鍵を用いて問題を解答することで獲得した Bitcoin の量を計測する
   - $m+3$ は目安なので、$m+1$ や $m+6$ などでもよい
2. Bitcoin を獲得した量でチームの順位付けを行う

フラッグワードが正しい場合の挙動

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

Listing: Tx_{1j} の *scriptSig*

```
S_1
h_{1j}
1
```

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

Listing: Tx_{1j} の *scriptSig*

```
S_1
h_{1j}
1
```

- 左が運営が作成したトランザクション Tx_j の *scriptPubKey*

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

Listing: Tx_{1j} の *scriptSig*

```
S_1
h_{1j}
1
```

- 左が運営が作成したトランザクション Tx_j の *scriptPubKey*
- 右がチーム T_1 が作成したトランザクション Tx_{1j} の *scriptSig*

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

1. チーム T_1 がトランザクション Tx_{1j} を送信したとする。その *scriptSig* から、スタックは $\boxed{1 \mid h_{1j} \mid S_1}$ となる。ここから Tx_j の *scriptPubKey* を実行する

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

1. チーム T_1 がトランザクション Tx_{1j} を送信したとする。その *scriptSig* から、スタックは $\boxed{1 \mid h_{1j} \mid S_1}$ となる。ここから Tx_j の *scriptPubKey* を実行する
2. スタックの先頭を複製して 1 を載せる $\boxed{1 \mid 1 \mid 1 \mid h_{1j} \mid S_1}$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

1. チーム T_1 がトランザクション Tx_{1j} を送信したとする。その *scriptSig* から、スタックは $\boxed{1 \mid h_{1j} \mid S_1}$ となる。ここから Tx_j の *scriptPubKey* を実行する
2. スタックの先頭を複製して 1 を載せる $\boxed{1 \mid 1 \mid 1 \mid h_{1j} \mid S_1}$
3. スタックの先頭から 2 つを取り出し、それらを比較する（等しいので 1 が積まれる） $\boxed{1 \mid 1 \mid h_{1j} \mid S_1}$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

1. チーム T_1 がトランザクション Tx_{1j} を送信したとする。その *scriptSig* から、スタックは $\boxed{1 \mid h_{1j} \mid S_1}$ となる。ここから Tx_j の *scriptPubKey* を実行する
2. スタックの先頭を複製して 1 を載せる $\boxed{1 \mid 1 \mid 1 \mid h_{1j} \mid S_1}$
3. スタックの先頭から 2 つを取り出し、それらを比較する（等しいので 1 が積まれる） $\boxed{1 \mid 1 \mid h_{1j} \mid S_1}$
4. スタックの先頭を取り除き、それが 1 なので OP_IFから OP_ELSEを実行する $\boxed{1 \mid h_{1j} \mid S_1}$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

5. スタックの先頭を捨てる

h_{1j}	S_1

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

5. スタックの先頭を捨てる $\boxed{h_{1j} \mid S_1}$
6. スタック先頭に SHA-256 を適用し結果をスタックの先頭に追加する $\boxed{H(h_{1j}) \mid S_1}$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

5. スタックの先頭を捨てる $\boxed{h_{1j} \mid S_1}$
6. スタック先頭に SHA-256 を適用し結果をスタックの先頭に追加する
   $\boxed{H(h_{1j}) \mid S_1}$
7. ans_{1j} をスタックの先頭に追加する
   $\boxed{ans_{1j} \mid H(h_{1j}) \mid S_1}$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

5. スタックの先頭を捨てる $\boxed{h_{1j} \mid S_1}$
6. スタック先頭に SHA-256 を適用し結果をスタックの先頭に追加する $\boxed{H(h_{1j}) \mid S_1}$
7. ans_{1j} をスタックの先頭に追加する $\boxed{ans_{1j} \mid H(h_{1j}) \mid S_1}$
8. スタックの先頭から 2 つを取り出し、それらを比較する。等しくない場合は直ちに失敗となる $\boxed{S_1}$
   - $h_{1j} = H(F_j \,||\, 1)$
   - $ans_{1j} = H\left(H(F_j \,||\, 1)\right)$

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

9. チーム T_1 の公開鍵 $\mathcal{T}_1$ をスタックの先頭に追加する

$\mathcal{T}_1$	S_1

フラッグワードが正しい場合の挙動

Listing: Tx_j の *scriptPubKey*

```
OP_DUP
1
OP_EQUAL
OP_IF
   OP_DROP
   OP_SHA256
   ans_{1j}
   OP_EQUALVERIFY
   T_1
OP_ELSE
   ⋮
OP_ENDIF
OP_CHECKSIG
```

9. チーム T_1 の公開鍵 $\mathcal{T}_1$ をスタックの先頭に追加する $\boxed{\mathcal{T}_1 \mid S_1}$
10. スタックの先頭にあるデータを公開鍵として、スタックの先頭から 2 番目にあるデータとして署名としてそれらを検証する $\boxed{1}$

進んだ話題

進んだ話題

- ブロックチェーンの長さが m となったら運営が賞金を回収するために、トランザクションを次のようにする

Listing: 改良した Tx_j の *scriptPubKey* の一部

```
⋮
OP_ELSE
   OP_DROP
   m
   OP_CHECKLOCKTIMEVERIFY
   OP_DROP
   T_master
OP_ENDIF
⋮
OP_CHECKSIG
```

  - ただし $\mathcal{T}_{master}$ は運営の Bitcoin 公開鍵とする

進んだ話題

- 問題とチームのそれぞれごとにトランザクションを用意することで、従来の CTF のように解答順序によらずに賞金を与えることもできる？

進んだ話題

- 問題とチームのそれぞれごとにトランザクションを用意することで、従来の CTF のように解答順序によらずに賞金を与えることもできる？
  - ただし、そうするとメンバーを同じにしたチームをたくさん作ることで分割した賞金を不正に取得する攻撃ができる

進んだ話題

- 問題とチームのそれぞれごとにトランザクションを用意することで、従来の CTF のように解答順序によらずに賞金を与えることもできる？
  - ただし、そうするとメンバーを同じにしたチームをたくさん作ることで分割した賞金を不正に取得する攻撃ができる
  - 従って最初に解いたチーム以外に賞金を与えることはできない

まとめ

まとめ

- Bitcoin を利用した CTF について考えることで、より透明で公平な CTF を構成できた

まとめ

- Bitcoin を利用した CTF について考えることで、より透明で公平な CTF を構成できた
- Bitcoin のリファレンス実装は、提案する CTF で利用するようなトランザクションを受け付けない。よって現実的には Ethereum など別の暗号通貨を利用しなければならないかもしれない

まとめ

- Bitcoin を利用した CTF について考えることで、より透明で公平な CTF を構成できた
- Bitcoin のリファレンス実装は、提案する CTF で利用するようなトランザクションを受け付けない。よって現実的には Ethereum など別の暗号通貨を利用しなければならないかもしれない
  - @lotz さんが Ethereum で実装してくださった

まとめ

- Bitcoin を利用した CTF について考えることで、より透明で公平な CTF を構成できた
- Bitcoin のリファレンス実装は、提案する CTF で利用するようなトランザクションを受け付けない。よって現実的には Ethereum など別の暗号通貨を利用しなければならないかもしれない
  - @lotz さんが Ethereum で実装してくださった
  - http://lotz84.hatenablog.com/entry/2018/01/02/134056

まとめ

- Bitcoin を利用した CTF について考えることで、より透明で公平な CTF を構成できた
- Bitcoin のリファレンス実装は、提案する CTF で利用するようなトランザクションを受け付けない。よって現実的には Ethereum など別の暗号通貨を利用しなければならないかもしれない
  - @lotz さんが Ethereum で実装してくださった
  - http://lotz84.hatenablog.com/entry/2018/01/02/134056
- 賞金が集まれば、提案する方法で CTF を実施してみたい

目次

Thank you for your attention!