zk−SNARKの理論

Kyosuke Tsubaki

自己紹介

椿　京介

東京理科大学　情報科学科４年

一年ほど前からGaudiyにインターンとして参画、現在はエンジニアとしてコミュニティアプリのフロントやバックエンドの開発を行っています。

背景課題

Gaudiyのサービスを例にzk-SNARKがどのように使われるか

IP毎に独自のコミュニティプラットフォームを提供

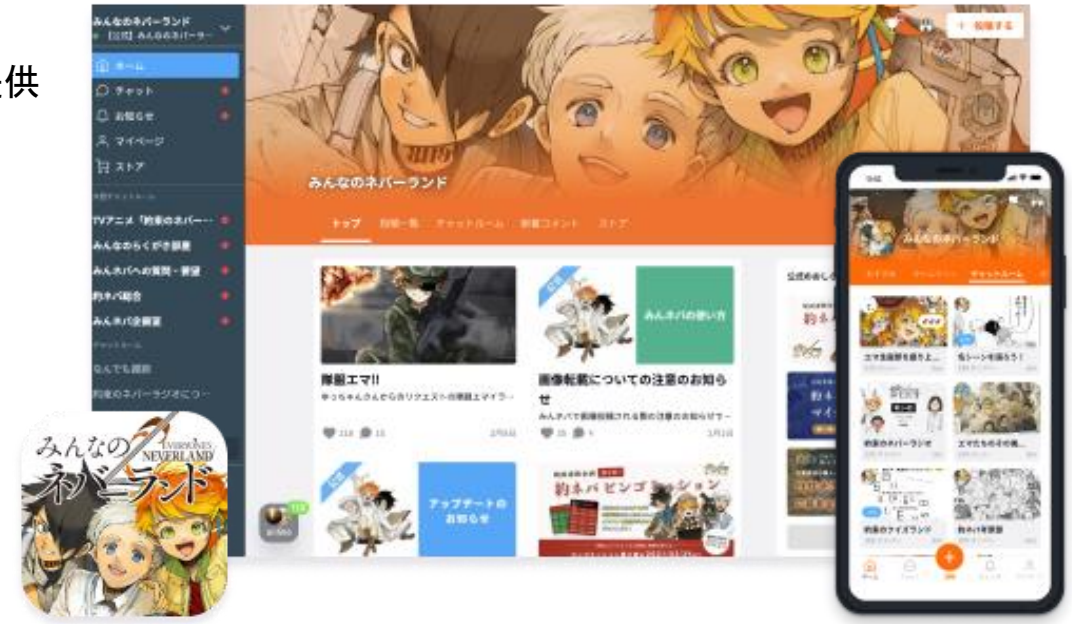
トークンエコノミーを実現するためにブロックチェーンを活用したファンコミュニティ

IP軸のコミュニティプラットフォームを提供

マンガ、アニメ、アイドル、ゲームなど
多様なエンタメジャンルに対応

ブロックチェーンの社会実装

マジックリンクの仕組みを利用して、秘密鍵やウォレットなどの一般ユーザーにはわかりにくい概念を理解しやすい形で提供



高いカスタマイズ性・多機能

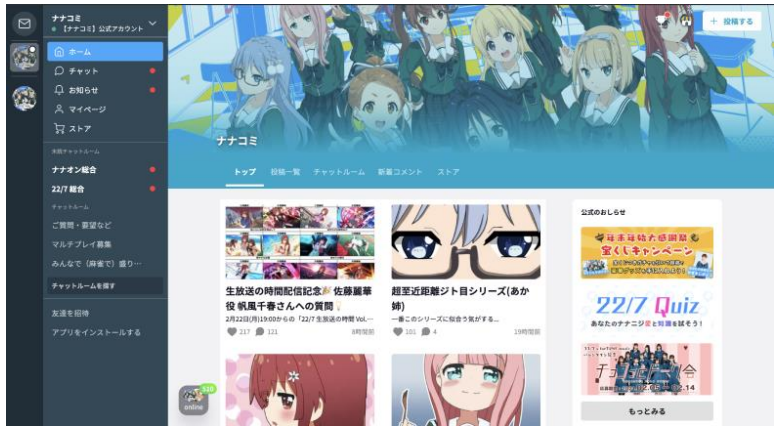
ブロックチェーン技術を活用して、決済システムやデジタルNFTトレカ、機能を展開

自社のコミュニティを起点に、ファンエンゲージメントとDXを促進

提供サービス例

DID（分散型ID）というブロックチェーン技術を活用して、IPを軸に異なるサービス（コミュニティ、ゲーム、ファンクラブetc.）を連携。

コミュニティ



チケット

TICKET

forTUNE music
フォーチュンミュージック

ファンクラブ

22/7
FANCLUB

ゲーム

22/7 ナナブンノニジュウニ 音楽の時間 ™

デジタル声優アイドル「22/7（ナナブンノニジュウニ）」経済圏

プライバシーの問題

外部プラットフォームとコミュニティがウォレットによって繋がっている

課題：
ブロックチェーンはパブリック性がある一方でプライバシーの問題がある

ゲーム

ある情報を持っているか確認したい

そのまま渡すと必要のない情報や機密情報まで知られてしまう

ファンクラブ

zk-SNARKというプライバシー強化を可能にしている技術を活用することで秘匿化された情報の整合性と正確性を保証しながら、ブロックチェーン上で検証可能。

zk-SNARKとは

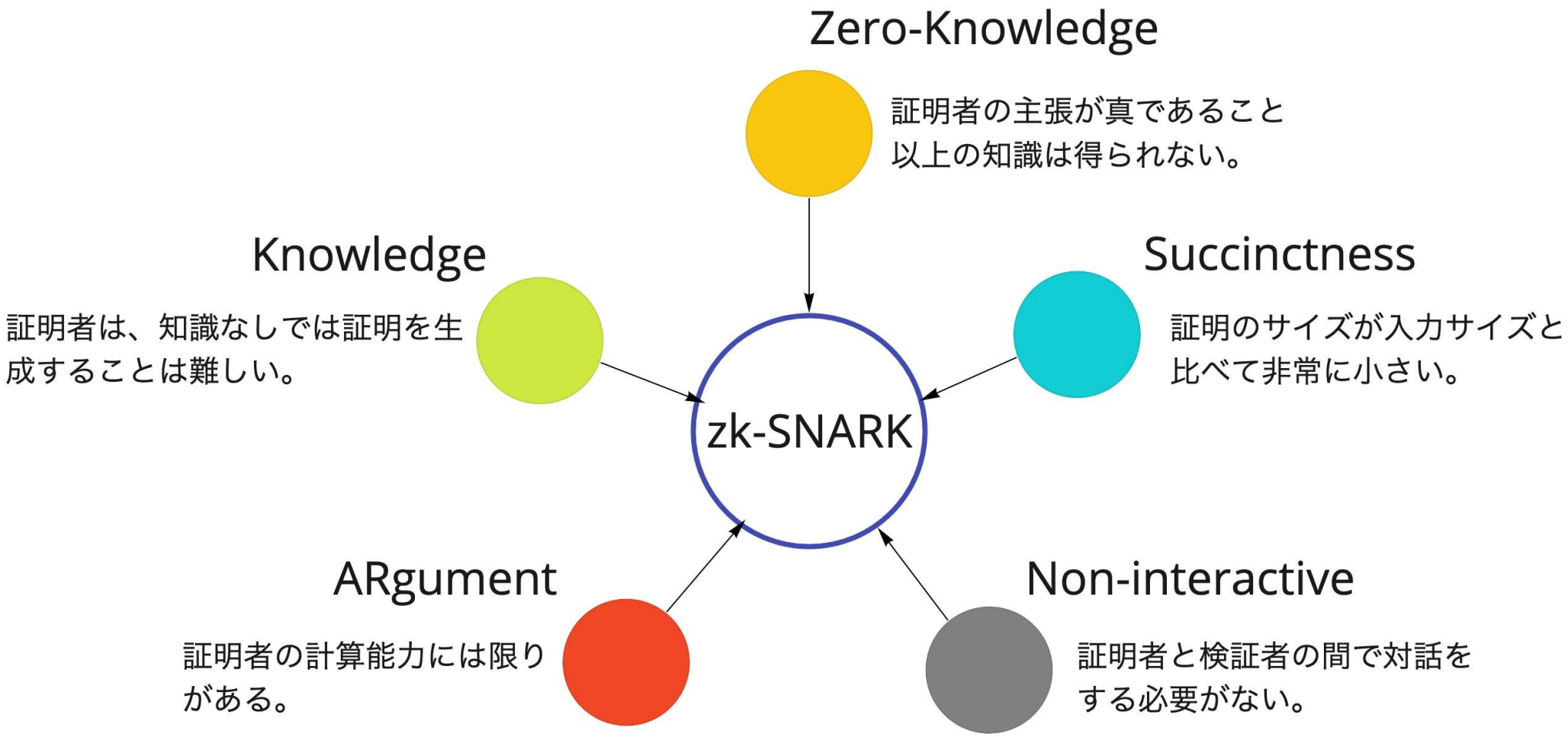
zk-SNARK（Zero Knowledge Succinct Non-interactive Arguments of Knowledge）とは非対話型ゼロ知識証明を構築している技術。

- ブロックチェーン におけるトランザクションの秘匿化
- イーサリアムのスケーリング問題のソリューション

などに応用されている。

また、zk-SNARKは「Pinocchio Protocol」と呼ばれる「暗号学的仮定のみに依存し、一般的な計算を効率的に検証するために構築されたシステム」のもとで構築されており、簡潔でありながら、強力なセキュリティを実現している。

zk-SNARKの性質

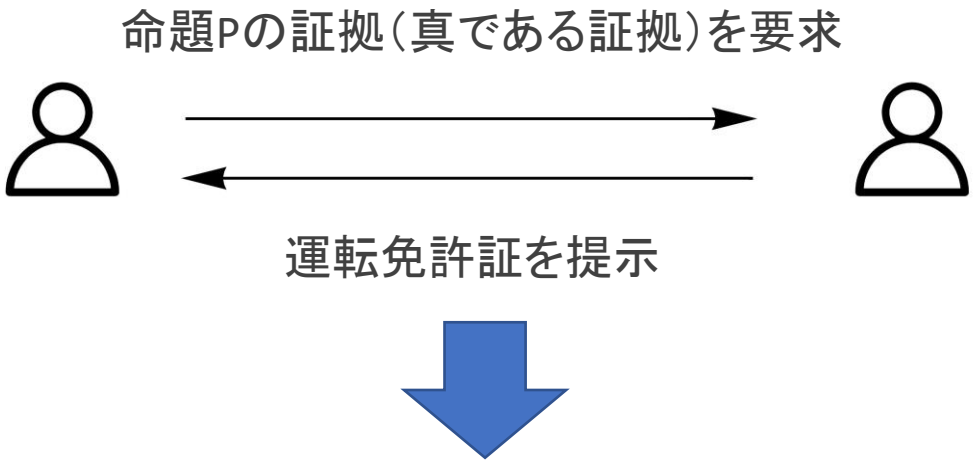

Zero-Knowledge
証明者の主張が真であること
以上の知識は得られない。
Knowledge
証明者は、知識なしでは証明を生
成することは難しい。
Succinctness
証明のサイズが入力サイズと
比べて非常に小さい。
zk-SNARK
ARgument
証明者の計算能力には限り
がある。
Non-interactive
証明者と検証者の間で対話を
する必要がない。

zk-SNARKを説明する上で一番重要なゼロ知識証明

ゼロ知識証明とは

ゼロ知識証明とは証明者が検証者に対して、ある情報が正しいことを、それが正しいこと以外の情報を明らかにせずに証明できる手法のこと。

命題P
20歳以上である



これだと個人情報まで知られてしまうので、20歳以上であることのみを証明したい。（ゼロ知識証明）

ゼロ知識証明の性質

- 完全性
  - 真であることを確認する側（検証者）は、証明する側（証明者）の持っている命題が真であるならば、真であることが必ずわかること。
- 健全性
  - 証明者の持つ命題が偽であるなら、検証者は高い確率でそれが偽であると見抜けること。
- ゼロ知識
  - 証明者の持つ命題が真であるなら、検証者が不正して証明者から知識を盗もうとしても「命題が真である」以外の何の知識も得られないこと。

zk-SNARKの構成を説明する前に前提となる知識

① 複雑性クラスNP

② NP言語L

複雑性クラス

- クラスP

「多項式時間で解ける判定問題」全体の集合

（多項式時間とは、現実的に計算可能な時間のことを意味する。）

- クラスNP

「多項式時間で検証できる判定問題」全体の集合

→ **ある判定問題に対する証拠wが与えられた時、その証拠wが正しいかどうかを多項式時間で判定できる**が、自らその問題の解を見つけるには指数関数的な時間を要するかもしれない。

（NP問題の例：巡回セールスマン問題、3彩色問題）

NP言語Lに対するゼロ知識証明

証明者が証明する命題は「NP言語Lとステートメントxに関して$x \epsilon L$が成り立つこと」

●NP言語Lの例

「正しく生成されたRSAの公開鍵の集合」

●ステートメントxの例

「特定の公開鍵」

●証拠wの例

「公開鍵の素因数」（秘密鍵）

証拠wからその公開鍵がRSAの公開鍵の集合に含まれるかどうかの確認が簡単にできる。しかし、公開鍵から素因数を特定することは難しい。

zk−SNARKの流れ

- 関数$f(x, w)$は入力が正しい場合のみ1を出力し、正しくない場合は0を出力する関数。
- 証明者は検証者に対して$f(x, w) = 1$となるような証拠wを持っていることをそれ自体を明らかにせずに証明したい。
- 証明者は「正しい証拠wを持っている」ことを証明する証明πを作り、検証者に送る。
- 検証者は受け取った証明πを検証し、それが正しい場合は受け入れ、正しくない場合は拒否する。

証明者と検証者の間に「信頼できる第三者」をおくことで、非対話のゼロ知識証明を構築し、効率的で安全性の高い証明が可能になる。

zk−SNARKの構成

- Generator

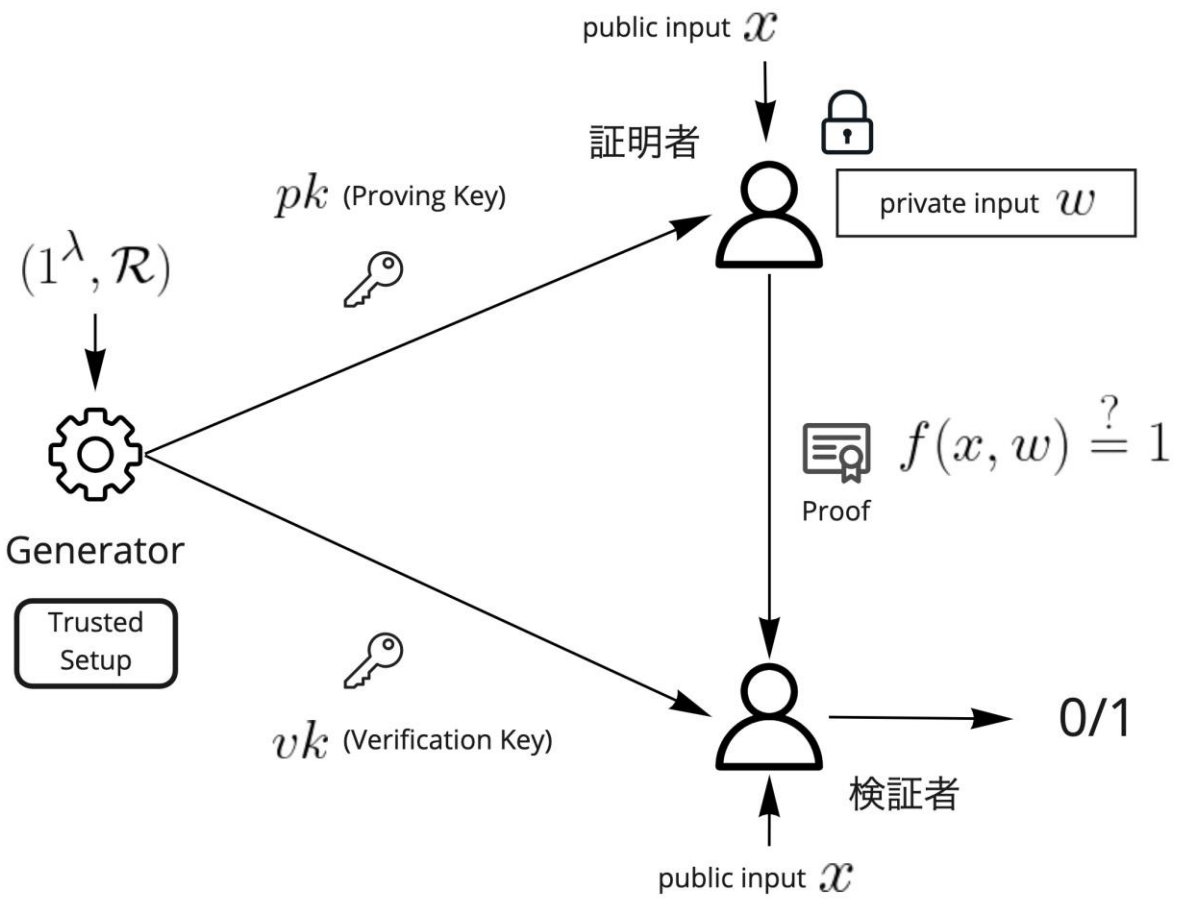
信頼できる第三者。セットアップとして事前にCRS(common reference string)を生成し、pkとvkをそれぞれ証明者と検証者に送る。

- 証明者

pkとxを受け取り、wから証明πを作成し、検証者に送る。

- 検証者

vkと証明πとxを受け取り、その証明が正しいものであれば受理し、そうでなければ拒否する。



「正しい証拠wを持っている」を「ある多項式を知っている」に変換する

代数的性質へ変換

「$f(x,w)=1$となる証拠wを持っている」 → 「$h(x)t(x)=p(x)$を満たす多項式$h(x)$を知っている」

- 関数fが加算、乗算ゲートからなる演算回路として表現できる。
- この演算回路から目的多項式$t(x)$と関数fを表す3つの多項式の組$\{v_0(x),\dots,v_m(x)\},\{w_0(x),\dots,w_m(x)\},\{y_0(x),\dots,y_m(x)\}$が得られる。（これらは左入力多項式、右入力多項式、出力多項式と呼ばれる。）
- この回路に入力を与えると$c_1,\dots,c_m$を得ることができる。この値は回路の入出力と各ゲートの出力値（中間値）を表す。
- この$c_1,\dots,c_m$と3つの多項式の組の線形結合から以下のような多項式$p(x)$を作ることができる。

$$p(x)=\left(v_0(x)+\sum_{k=1}^{m}c_k\cdot v_k(x)\right)\cdot\left(w_0(x)+\sum_{k=1}^{m}c_k\cdot w_k(x)\right)-\left(y_0(x)+\sum_{k=1}^{m}c_k\cdot y_k(x)\right)$$

- 回路に対する入力が「正しい」場合、$p(x)$は$t(x)$を割り切ることができ、$h(x)t(x)=p(x)$を満たすような$h(x)$が存在する。
- 「正しくない」場合は割り切ることができず、上式を満たすような$h(x)$は存在しない

これらの多項式が証明者と検証者
間でどのように表現されるか

多項式の検証

送られてきた多項式が正しいか検証

$$h(x)t(x) \stackrel{?}{=} p(x)$$

演算回路から得られた3つの多項式の組と$c_1, \ldots, c_m$の線形結合

$$p(x) = \left(v_0(x) + \sum_{k=1}^{m} c_k \cdot v_k(x)\right) \cdot \left(w_0(x) + \sum_{k=1}^{m} c_k \cdot w_k(x)\right) - \left(y_0(x) + \sum_{k=1}^{m} c_k \cdot y_k(x)\right)$$

$$p(x) = \big(v_0(x) + v(x)\big) \cdot \big(w_0(x) + w(x)\big) - \big(y_0(x) + y(x)\big)$$

$t(x), v_0(x), w_0(x), y_0(x)$

検証者

証明π= $h(x), v(x), w(x), y(x)$

$h(x)t(x) = p(x)$を満たす$h(x)$を持っているという証明

証明者

回路に入力を与えると得られる

$c_1, \ldots c_m$

証明者のみが知っている値

多項式のゼロ知識化

多項式を隠蔽する

一方向性関数E

$$E(x) = g^x$$

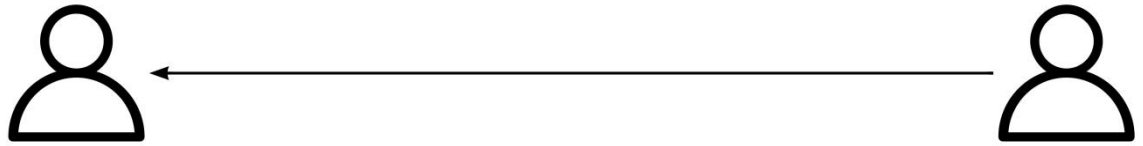
性質：

xからg^xを作ることは簡単だが、
g^xからxを特定するのは難しい
（離散対数問題）

一方向性関数$E(x)$で$h(x), v(x), w(x), y(x)$を隠蔽して検証者に送る。

このとき、これらの多項式の詳細を知ることは難しいため、**検証者は$c_1, \ldots c_m$の値を盗むことはできない**。$c_1, \ldots c_m$を知るということは回路を満たす入力を知っていることになる。

証明$\pi = E\big(h(x)\big), E\big(v(x)\big), E\big(w(x)\big), E\big(y(x)\big)$



検証者　　証明者

多項式が隠蔽された状態で検証

ペアリングによる検証

例えば、証明者は$x_1 x_2 + {x_3}^2 = 0$を満たす$(a_1 a_2 a_3)$を知っていること検証者にその値を明らかにせずに証明することができる。

$$e(g,g)^{a_1 a_2 + {a_3}^2} = \mathrm{e}(g^{a_1}, g^{a_2}) \cdot e(g^{a_3}, g^{a_3}) \stackrel{?}{=} e(g,g)^0$$

写像eが双線形性を持つとき、その写像をペアリングと呼ぶ。

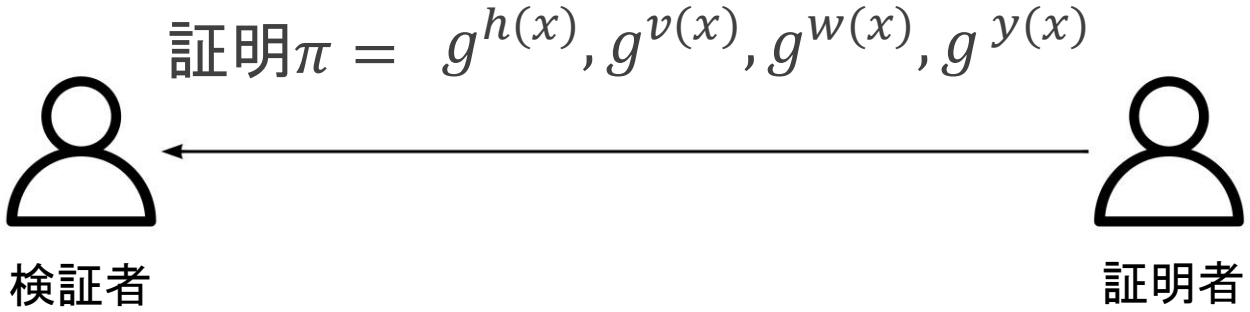
ペアリングにより「IDベース暗号」や「検索可能暗号」、「関数型暗号」などの技術を実現できる。

ペアリングによる多項式の検証

$$e\left(g^{h(x)}, g^{t(x)}\right) = e(g,g)^{h(x)t(x)}$$
$$= e(g,g)^{(v_0+v(x))\cdot(w_0+w(x))-(y_0+y(x))}$$
$$= e(g^{v_0(x)}g^{v(x)}, g^{w_0(x)}g^{w(x)})/e(g^{y_0(x)}g^{y(x)}, g)$$

多項式の詳細を知ることは難しいが、
$h(x)t(x) = p(x)$が成り立つかどうかの
検証は可能

$$t(x), v_0(x), w_0(x), y_0(x)$$



証明者と検証者の間に「信頼できる第三者」
をおき、非対話型のゼロ知識証明

非対話のゼロ知識証明

pk

$\{g^{v_0(x)}, \ldots, g^{v(x)}\}, \{g^{w_0(x)}, \ldots, g^{w(x)}\}, \{g^{y_0(x)}, \ldots, g^{y(x)}\}$

vk

$g^{t(x)}, g^{v_0(x)}, g^{w_0(x)}, g^{y_0(x)}$

$$g^{v(x)} = g^{v_0(x)+c_1 \cdot v_1(x)+\cdots+c_m \cdot v_m(x)}$$
$$= g^{v_0(x)} \cdot g^{c_1 \cdot v_1(x)} \cdot \ldots \cdot g^{c_m \cdot v_m(x)}$$
$$= (g^{v_0(x)})^1 \cdot (g^{v_1(x)})^{c_1} \cdot \ldots \cdot (g^{v_m(x)})^{c_m}$$

Generator

$$e(g^{h(x)}, g^{t(x)}) = e(g^{v_0(x)} g^{v(x)}, g^{w_0(x)} g^{w(x)}) / e(g^{y_0(x)} g^{y(x)}, g)$$

vk

pk

演算回路の入力から取得

$c_1, \ldots, c_m$

private input w

証明$\pi = g^{h(x)}, g^{v(x)}, g^{w(x)}, g^{y(x)}$

$0/1$

検証者

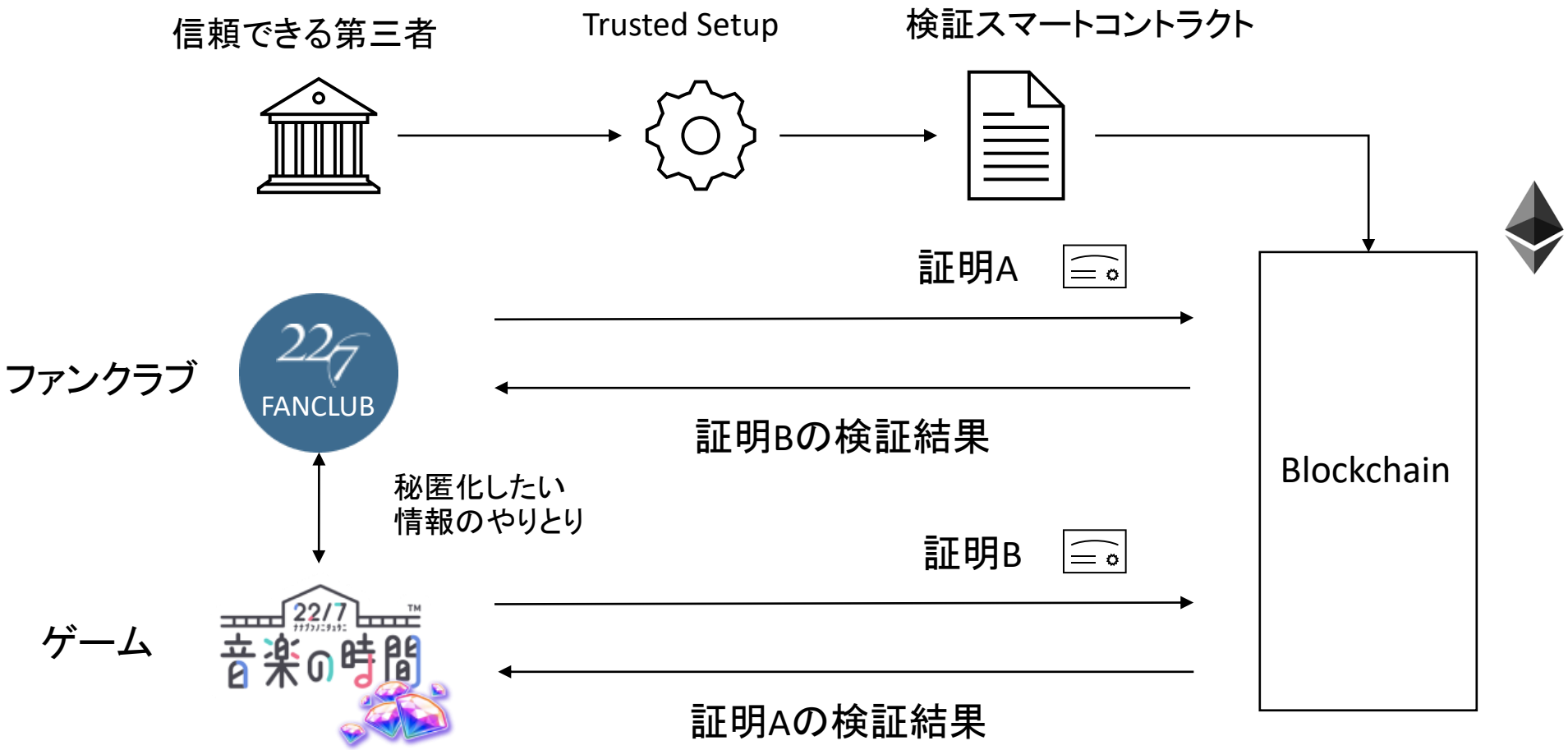
public input x

証明者

public input x

プライバシー問題の解決

zk-SNARKによる課題の解決



まとめ

zk-SNARKの簡単な全体の流れの説明を行った。

説明できていないこと

- 証明者に多項式時間の計算能力があるからといって、必ずしも計算結果を送ることを保証できていない。

  → Pinocchio Protocolのq-PKEの仮定より不正することを難しくしている。

- 多項式の検証の効率化

  →Schwartz-Zippelの補題により簡単に不正な多項式を見抜ける。

- 完全なゼロ知識証明化

  →証明の乱数化