

ZK-SNARKの仕組み ～Pinocchio編～

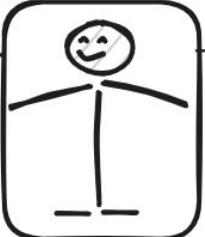
RG@SFC Delight B2 inaridiy

※終始インフォーマインフォーマルです。間違いも多分あります

ゼロ知識証明プロトコル(ZKP)の概要

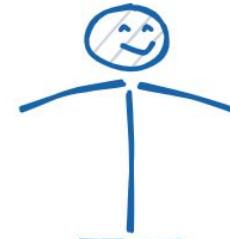
証明者が検証者に、ある共有された命題が真であると証明する際、
真であること以外の何の知識も伝えることなく証明できるような
やりとりの手法

本人確認カード

Name: CSS2025 太郎	
Address: 日本-岡山	
平成32年 2月30日 生	
.....	
.....	
.....	
2025年10月26日まで有効	

個人情報の塊

私は20歳以上です！

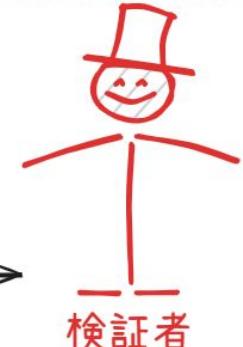


証明者

Name: [REDACTED]	
Address: [REDACTED]	
平成32年 2月30日 生	
.....	
.....	
.....	
[REDACTED] まで有効	

生年月日以外、隠して提示する

生年月日を確認できた
他はわからないけど



検証者

ゼロ知識証明プロトコルの性質

- **ゼロ知識性(Zero-Knowledge):** 証明を見ても「それが真実」のみ分かる
 - 年齢以外の個人情報を明かさず、年齢が20歳以上であると証明できる
- **完全性(Completeness):** 本当のことは必ず証明できる
 - 年齢が20歳以上なら、検証者は必ず納得できる
- **健全性(Soundness):** 証明されたことは、必ず本当
 - 検証者が証明を検証出来たら、証明者は必ず20歳以上

完全性、健全性、ゼロ知識性の程度はプロトコルによって異なる

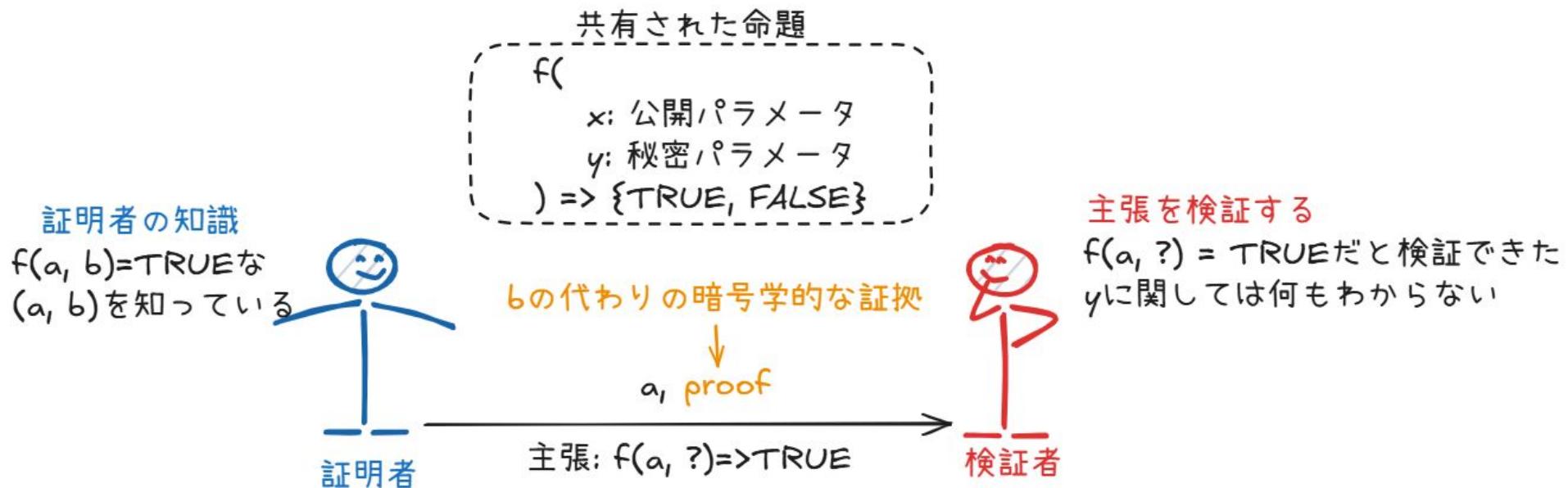
ZK-SNARK: 実用的なゼロ知識証明プロトコル

SNARK: Succinct Non-Interactive ARguments of Knowledge

- 簡潔性(Succinct)<=とてもすごい
 - 証明サイズが一定で、検証コストは入力サイズに比例し、命題の複雑さに対して検証コストは一定
- 非対話型(Non-interactive)<=かなり便利
 - 証明者から検証者に対する通信は一方方向
 - 一般的に、公的認証可能性を備える

ZK-SNARKの問題設定

共有された命題として、ある関数 $f(x, y)$ を考える



ZK-SNARKとVerifiable Computation(VC)

ZK-SNARKはしばしばVerifiable Computationとして用いられる

- **Verifiable Computation**とは
 - ZK-SNARKからゼロ知識性を引き、計算の検証可能性に特化したもの
 - 対象となる計算は、多項式時間で検証可能な任意の計算
- **Verifiable Computation**が利用される状況
 - 計算資源が豊富な主体と、計算資源が少ない主体がいて、少ない主体が、豊富な主体に計算 ($F(\text{input}) \Rightarrow \text{output}$) の代行をさせる
 - 計算資源の少ない主体は、豊富な主体から受け取った計算結果が、本当に正しく実行されたか物か、**計算を直接実行するよりも効率的に検証したい**

Pinocchio: Nearly Practical Verifiable Computation

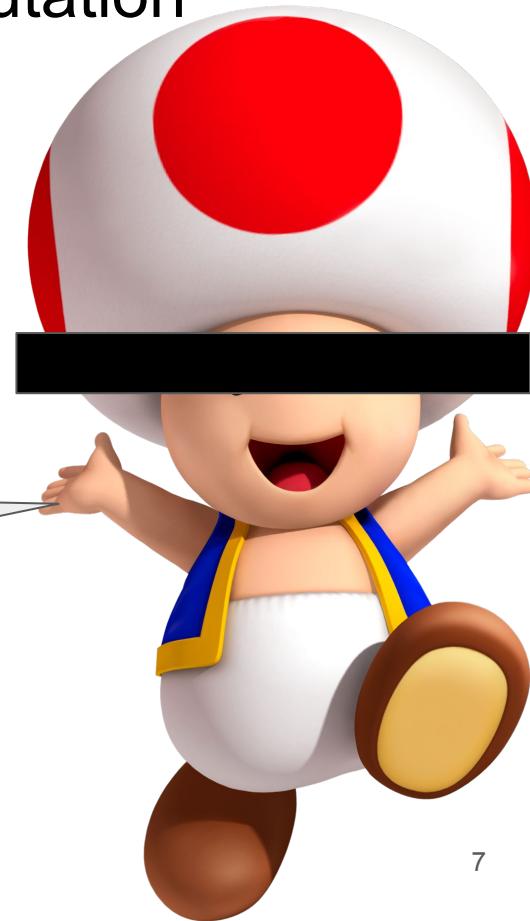
2013年公開・被引用数 1672回

<https://eprint.iacr.org/2013/279.pdf>

初の実用的なZK-SNARK兼VC

現在最も使われているGroth16と大体同じ

マンマ・ミーア



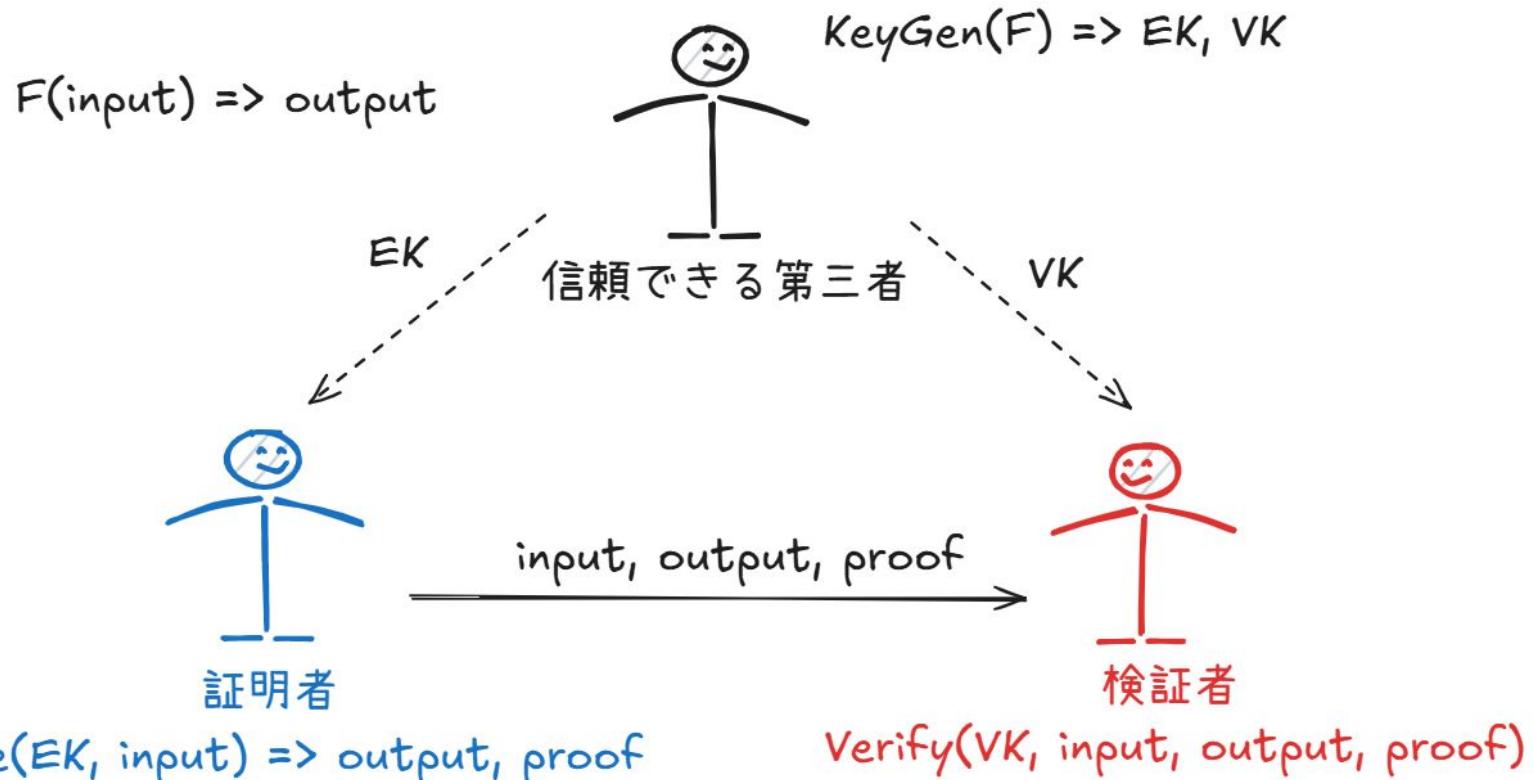
Pinocchio解説の流れ

1. プロトコルの全体像
2. 命題の変形、Gate=>R1CS=>QAP
3. 巡回群・ペアリング
4. Schwartz–Zipperの補題とペアリング
5. Pinocchioプロトコル???
6. Knowledge of Exponent Assumption
7. Pinocchioプロトコル
8. ゼロ知識化 $<=$!?!?なんで最後なの？！
9. こんくる一じょん

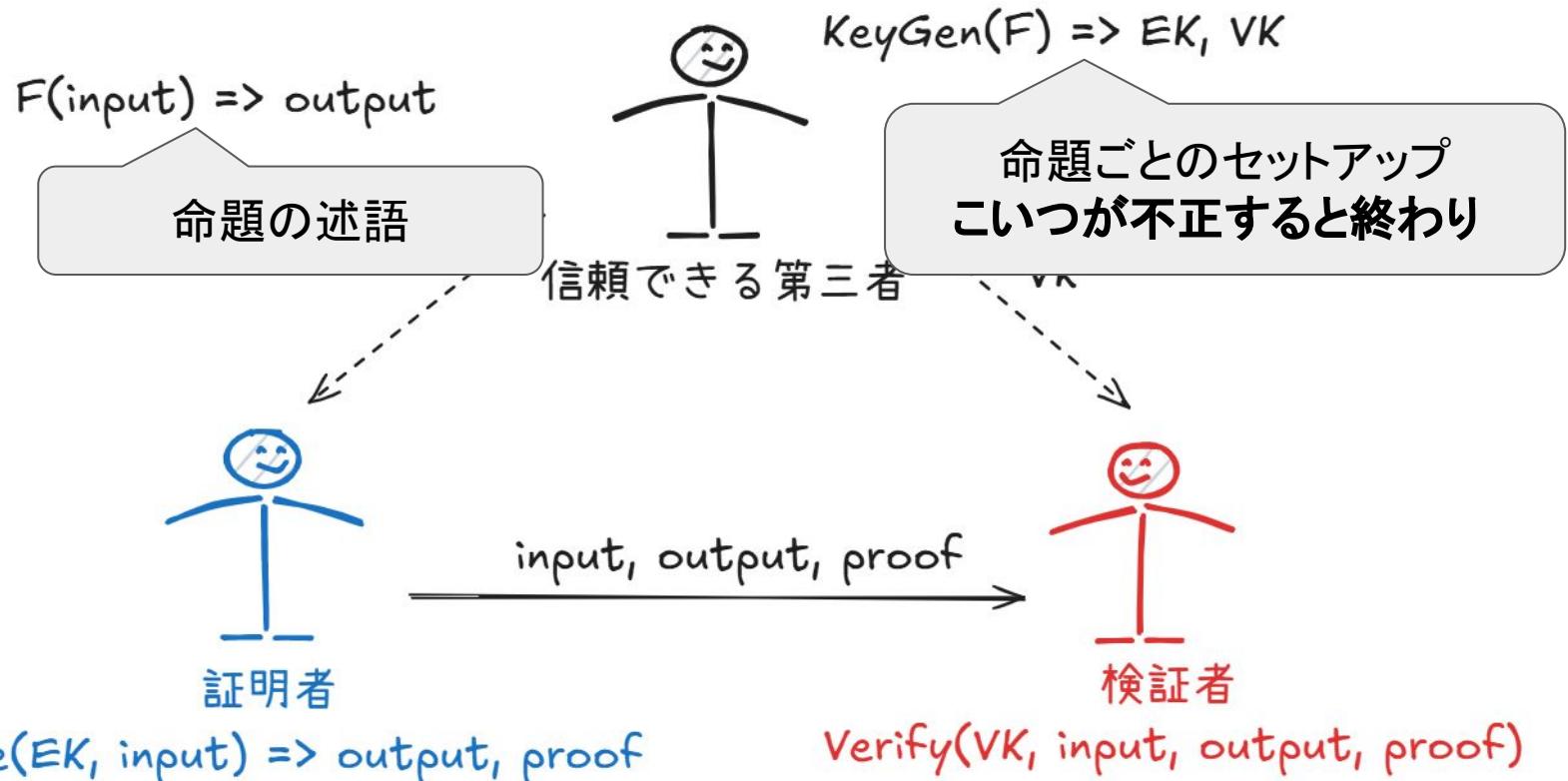


Pinocchioプロトコルの全体像

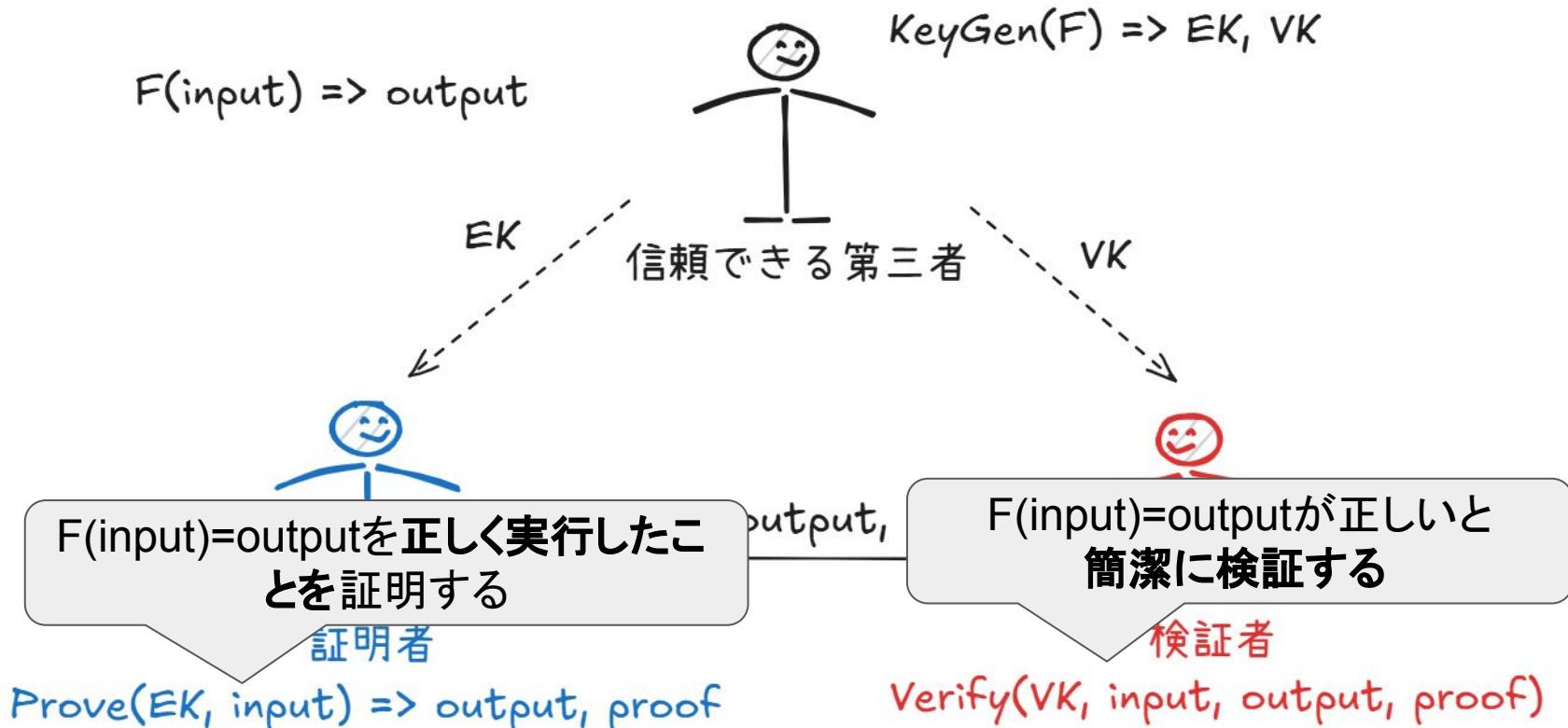
VCとしてのPinocchioプロトコル※ゼロ知識化ナシ



VCとしてのPinocchioプロトコル※ゼロ知識化ナシ



VCとしてのPinocchioプロトコル※ゼロ知識化ナシ



命題を変形しまくる。R1CS=>QAPへ

命題を変形しまくる。序章

方程式 $5x^3 + 3x = 46$ の解($x = 2$)を知っていることを命題とする

しかし、これを直接証明することは難しいので、

任意の命題 =>Flattening=>

R1CS(Rank1 Constraint System)=>QAP(Quadratic Arithmetic Program)
の順に命題を変形していき、数学的なパートを使いややすくする



方程式をFlatteningしてゲートの列にする

方程式を $c = a \text{ (op)} b$ という形式(ゲート)の列に変換する。(opは+や*など)
 $x=2$ なので、各変数は右のようになる

命題の術語

$$5x^3 + 3x$$

Flattening

ゲートの列

$$\begin{aligned} i1 &= 5x * x \\ i2 &= i1 * x \\ \text{out} &= i2 + 3x \end{aligned}$$

$x=2$ で評価

$x=2$ 時の各変数

$$\begin{aligned} x &= 2 \\ i1 &= 5x * x = 20 \\ i2 &= i1 * x = 40 \\ \text{out} &= i2 + 3x = 46 \end{aligned}$$

一旦各変数をベクトルに詰めてみる

とりあえず、定数1と各変数をベクトルwに詰める。

命題の術語

$$5x^3 + 3x$$

Flattening

ゲートの列

$$\begin{aligned} i1 &= 5x * x \\ i2 &= i1 * x \\ \text{out} &= i2 + 3x \end{aligned}$$

$x=2$ で評価

$x=2$ の時の各変数

$$\begin{aligned} x &= 2 \\ i1 &= 5x * x = 20 \\ i2 &= i1 * x = 40 \\ \text{out} &= i2 + 3x = 46 \end{aligned}$$

{1, x , out, i1, i2}の順に
詰めたベクトルを考える

$$w = \begin{pmatrix} \text{one} \\ x \\ \text{out} \\ i1 \\ i2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix}$$

一つ目のゲートを変形する。その1

一つ目のゲートを $x \text{ (op)} b - c = 0$ という形式に変換する

命題の術語

$$5x^3 + 3x$$

Flattening

ゲートの列

$$\begin{array}{|l} i1 = 5x * x \\ i2 = i1 * x \end{array}$$

$$out = i2 + 3x$$

$x=2$ で評価

↓ 一列目を $x \text{ (op)} b - c = 0$ に変形する

$$5x * x - i1 = 0$$

$x=2$ の時の各変数

$$\begin{aligned} x &= 2 \\ i1 &= 5x * x = 20 \\ i2 &= i1 * x = 40 \\ out &= i2 + 3x = 46 \end{aligned}$$

$\{1, x, out, i1, i2\}$ の順に
詰めたベクトルを考える

$$w = \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix}$$

一つ目のゲートを変形する。その2

ゲートが、 w においてどの位置の変数を使ったか、に着目してみる

命題の術語

$$5x^3 + 3x$$

Flattening

ゲートの列

$$\begin{cases} i1 = 5x * x \\ i2 = i1 * x \end{cases}$$

$$out = i2 + 3x$$

$x=2$ で評価

↓ 一列目を x (op) $b - c = 0$ に変形する

$$5x * x - i1 = 0$$

$w[1]$ は x と対応している

↓ w の中で何を何個使ったか表現する

$$(5 * w[1]) * (1 * w[1]) - (1 * w[3]) = 0 \leftarrow w =$$

$x=2$ の時の各変数

$$x = 2$$

$$i1 = 5x * x = 20$$

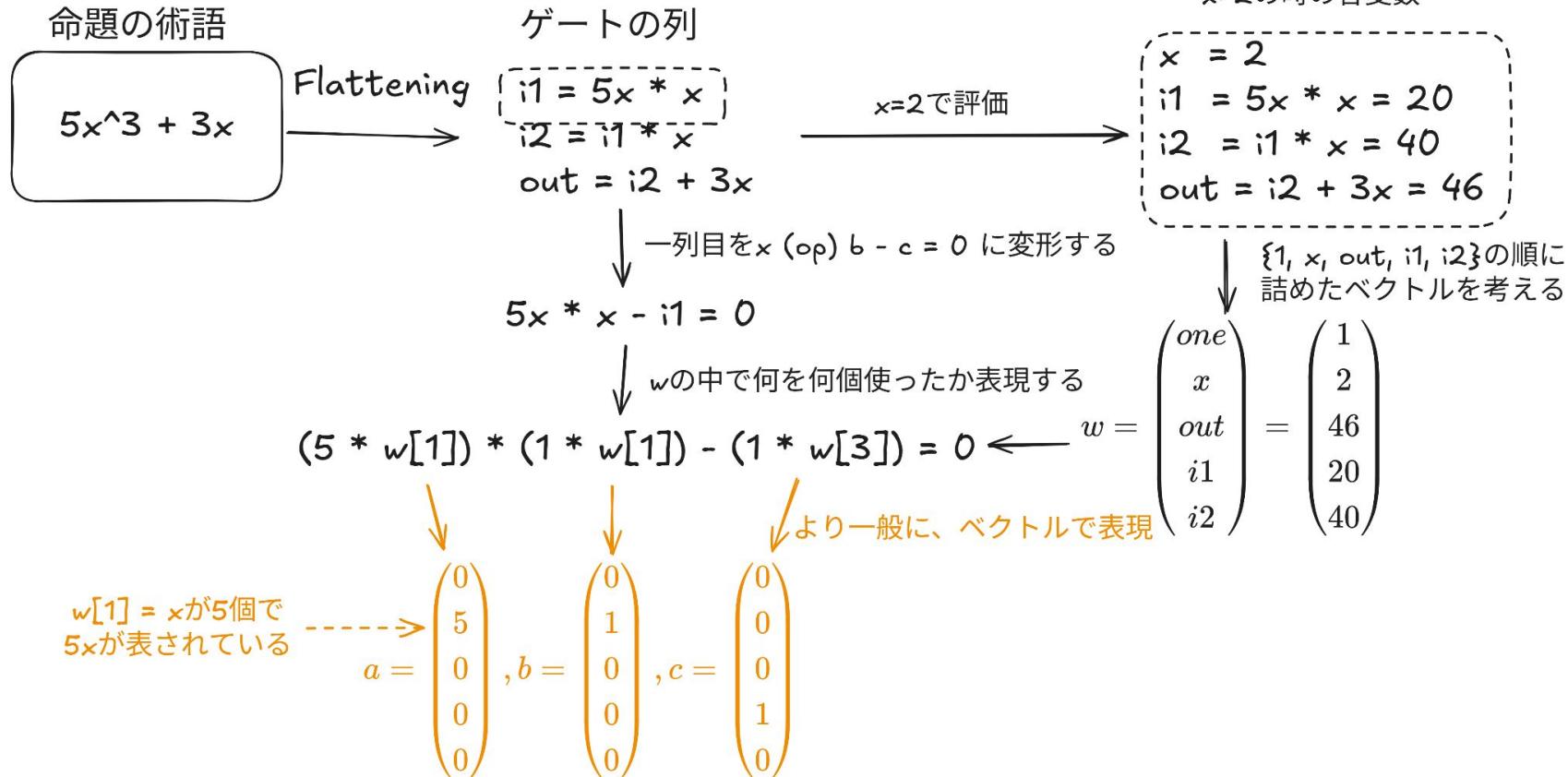
$$i2 = i1 * x = 40$$

$$out = i2 + 3x = 46$$

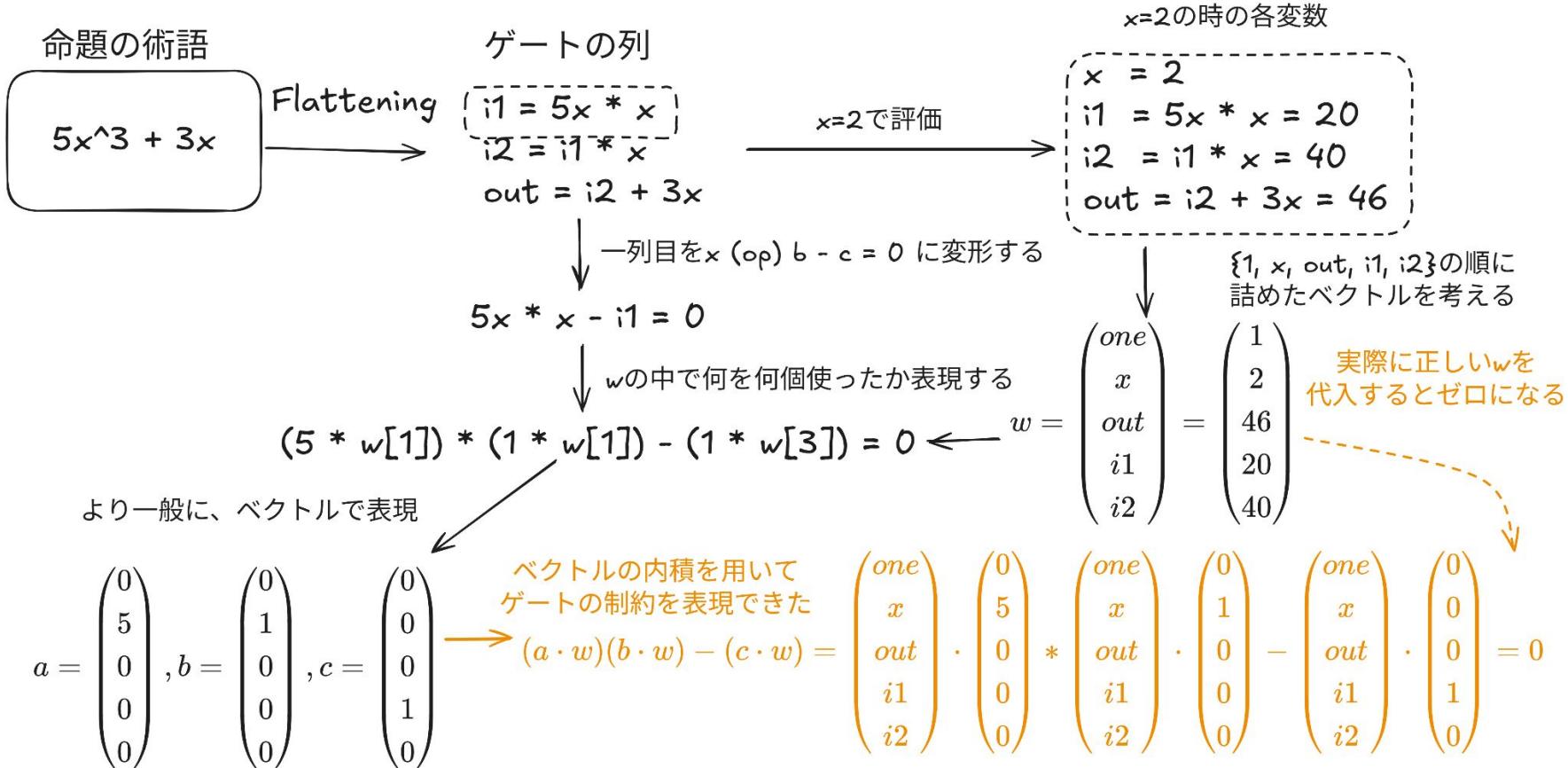
↓ $\{1, x, out, i1, i2\}$ の順に詰めたベクトルを考える

$$\begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix}$$

ゲートもベクトルで表せるんじゃね！？



ゲートを三つのベクトルとwとの内積で表現できた！！



全部を束ねると、こうなる。これがR1CS

abcを束ねて行列にする。この形式をR1CSと呼ぶ

[
i1 = 5x * x
i2 = i1 * x
out = i2 + 3x

$$a = \begin{pmatrix} 0 \\ 5 \\ 0 \\ 0 \\ 0 \end{pmatrix}, b = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}, c = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}$$

$$A = \begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

全部を束ねると、こうなる。これがR1CS

命題が正しいとは、述語行列A,B,Cに対する正しい変数ベクトルwを持つ、
 $(A_k \cdot w)(B_k \cdot w) - C_k \cdot w = 0$ ($k=1,2,3$)がすべて成り立つとき、と言える

$$A = \begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

$(one, x, out, i1, i2)$ な正しいwを持つとき、

$$(A_k \cdot w)(B_k \cdot w) - C_k \cdot w = 0 \quad k \in \{1, 2, 3\}$$

が成り立つ！！

ところで、ラグランジュ補完って知ってる？

$(x_0, y_0), (x_1, y_1), \dots, (x_n, y_n)$ をすべて通る多項式を決定的に導ける

$$P(x) = \sum_{j=0}^n y_j \prod_{\substack{k=0 \\ k \neq j}}^n \frac{x - x_k}{x_j - x_k}$$

ベクトル $(3, 0, 5)$ を、添え字を加えた点 $(1, 3), (2, 0), (3, 5)$ として多項式に埋め込むと

$$\begin{aligned} P(x) &= y_0 \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + y_1 \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + y_2 \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \\ &= 3 \cdot \frac{(x - 2)(x - 3)}{(1 - 2)(1 - 3)} + 0 \cdot \frac{(x - 1)(x - 3)}{(2 - 1)(2 - 3)} + 5 \cdot \frac{(x - 1)(x - 2)}{(3 - 1)(3 - 2)} \\ &= \frac{3}{2}(x - 2)(x - 3) + \frac{5}{2}(x - 1)(x - 2) = 4x^2 - 15x + 14 \end{aligned}$$

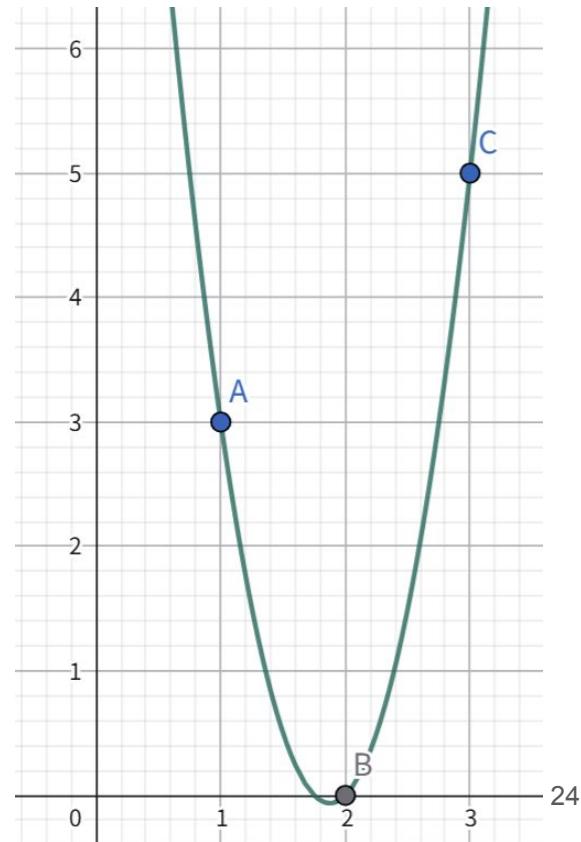
←---検算してみよう

ベクトル(3,0,5)を埋め込んだ多項式の検算

ベクトル(3,0,5)を、点(1,3),(2,0),(3,5)として、ラグランジュ補完で埋め込んだ多項式

$$f(x)=4x^2-15x+14$$

をプロットすると、確かに三つの点を通っていると確認できる



R1CSの行列を多項式にしてみよう！！

R1CSの行列を列ごとに(=同じ変数を使う物毎に)ラグランジュ補完する
結果、多項式のベクトルができる

変数 x にかかわる列を多項式にした $A_x(x)$

$$A = \begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}$$

\Rightarrow

$$AP(x) = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix}$$

変数 $i2$ に関わる列を多項式にした $A_{i2}(x)$

できた！これがQAPだ！！

前述の処理をR1CSの行列A,B,Cに行うと、次の形式になり、これがQAP

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix} \quad BP = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{3}{2}x - \frac{1}{2}x^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$CP = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{1}{2}x^2 - \frac{5}{2}tx + 3 \\ -x^2 + 4x - 3 \end{pmatrix}$$

....?で、何がうれしいの？

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix}$$

$$BP = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{3}{2}x - \frac{1}{2}x^2 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$CP = \begin{pmatrix} C_1(x) \\ C \\ C_c \\ C \\ C \end{pmatrix}$$

とりあえず変数ベクトルと内積を取ってみる

個別のゲートを扱ったときのように、変数ベクトルと内積を取ってみる

$$A(x) = AP \cdot w, \quad B(x) = BP \cdot w, \quad C(x) = CP \cdot w$$

とりあえず変数ベクトルと内積を取ってみる

個別のゲートを扱ったときのように、変数ベクトルと内積を取ってみる

$$A(x) = AP \cdot w, \quad B(x) = BP \cdot w, \quad C(x) = CP \cdot w$$

$$P(x) = A(x) * B(x) - C(x) = (AP \cdot w)(BP \cdot w) - (CP \cdot w)$$

$$= \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} * \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} - \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix}$$

とりあえず変数ベクトルと内積を取ってみる

個別のゲートを扱ったときのように、変数ベクトルと内積を取ってみる

$$A(x) = AP \cdot w, \quad B(x) = BP \cdot w, \quad C(x) = CP \cdot w$$

$$P(x) = A(x) * B(x) - C(x) = (AP \cdot w)(BP \cdot w) - (CP \cdot w)$$

$$\begin{aligned}
 &= \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} * \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} - \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} \\
 &= \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix} * \begin{pmatrix} \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{3}{2}x - \frac{1}{2}x^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{1}{2}x^2 - \frac{5}{2}tx + 3 \\ -x^2 + 4x - 3 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix}
 \end{aligned}$$

$$= -4x^4 + 19x^3 - 14x^2 - 31x + 30 \quad \text{なんか多項式ができた}$$

$P(x)$ をとりあえず $P(1)$ で評価してみると…?

$$\begin{aligned}
 P(1) &= \begin{pmatrix} A_1(1) \\ A_x(1) \\ A_{out}(1) \\ A_{i1}(1) \\ A_{i2}(1) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} * \begin{pmatrix} B_1(1) \\ B_x(1) \\ B_{out}(1) \\ B_{i1}(1) \\ B_{i2}(1) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} - \begin{pmatrix} C_1(1) \\ C_x(1) \\ C_{out}(1) \\ C_{i1}(1) \\ C_{i2}(1) \end{pmatrix} \cdot \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} \\
 &= \boxed{\begin{pmatrix} 0 \\ 5 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix} * \boxed{\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix} - \boxed{\begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{pmatrix}} \cdot \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix} = 0
 \end{aligned}$$

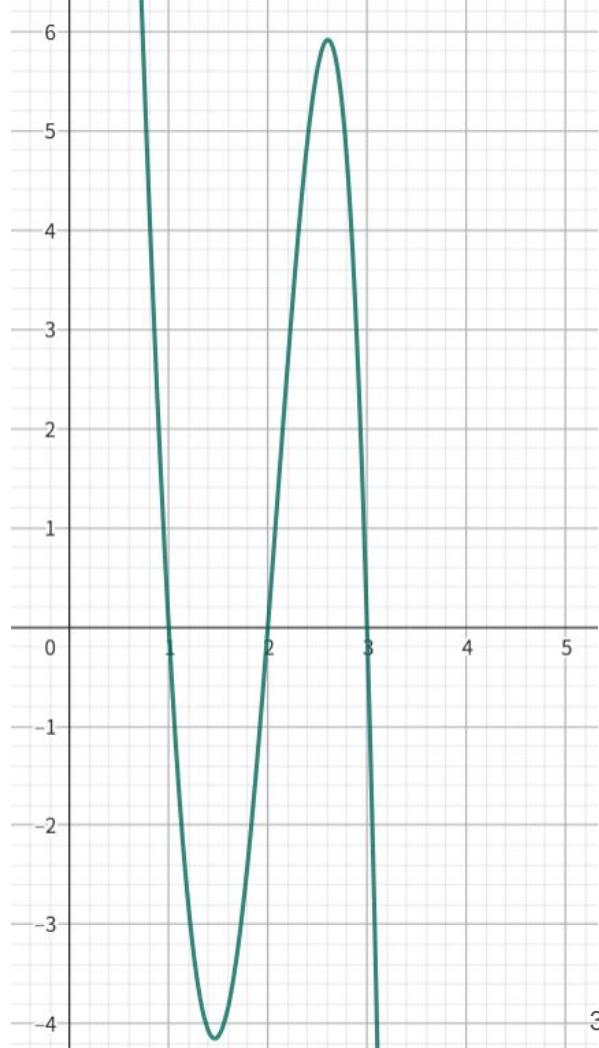
i1 = 5x * x のゲートのベクトルa,b,cと同じになったぞ！？

この多項式すごいんじゃね！？

ん？さっき個別に内積取ってたのと同じじゃね！？

$P(1) \cdot P(2) \cdot P(3)$ は、それぞれ1・2・3番目のゲートの制約をチェックすることと同義になる！！

$P(1) = P(2) = P(3) = 0$ なら、QAPの述語AP・BP・CPに対して正しい変数ベクトル w で $P(x)$ を作ったといえるぜ！！！！

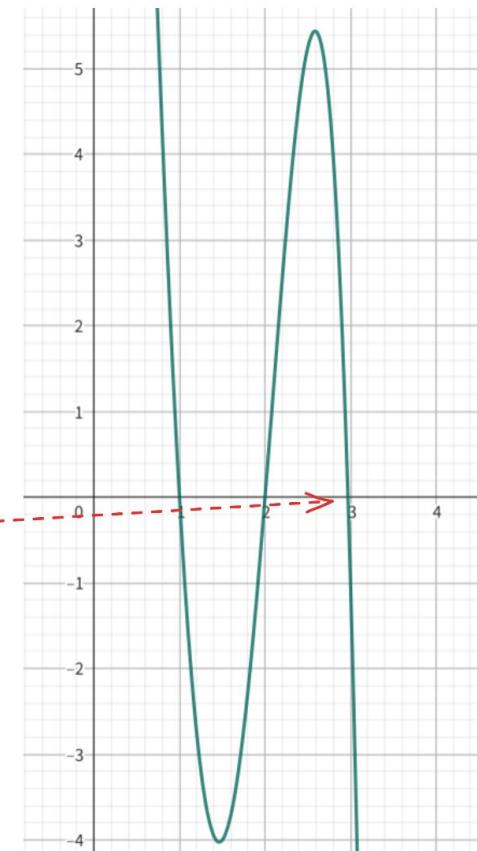


一応間違った変数ベクトルで $P(x)$ を作つてみると…

間違った変数割り当て $w_{bad} = (1, 2, 47, 20, 40)$
 $f(x) = 5x^3 + 3x$ で、本当は46なのに $f(2) = 47$ と主張する

$$\begin{aligned}P_{bad}(x) &= (AP \cdot w_{bad})(BP \cdot w_{bad}) - (CP \cdot w_{bad}) \\&= -4t^4 + 19t^3 - \frac{29}{2}t^2 - \frac{59}{2}t + 29\end{aligned}$$

$P_{bad}(3) = -1$ で、ギリ0を通つてない



すごいぞQAP！！まとめて命題を証明できる

正しい変数 w を知っているなら、 $P(1) = P(2) = P(3)=0$ になる

なら、因数定理より、多項式 $P(x)$ は $(x-1), (x-2), (x-3)$ を因子に持つ

なら、 $P(x)$ は多項式 $t(x) = (x - 1)(x - 2)(x - 3)$ で割り切れる

命題が正しいかを $P(x)$ が $t(x)$ で割り切れるか否かで判定できるようになった

各ゲートを同時にチェックできる！！！すごい

つまり、 $P(x)$ は適当な多項式 $h(x)$ を用いて、 $P(x)=h(x)t(x)$ と表せる

巡回群とペアリング

群(可換群)とは集合にいい感じの二項演算を定めたもの

集合 G 上に、二項演算 $*$ ($G \times G \Rightarrow G$) が定義されていて、

1. 結合法則: $(a * b) * c = a * (b * c)$
2. 単位元の存在: $a * e = e * a = a$ (加法におけるゼロ、乗法において1)
3. 逆元の存在: $a * a^{-1} = e$ となるような逆元が常に存在する

が成立するとき、 $(G, *)$ を群という

4. 交換法則: $a * b = b * a$

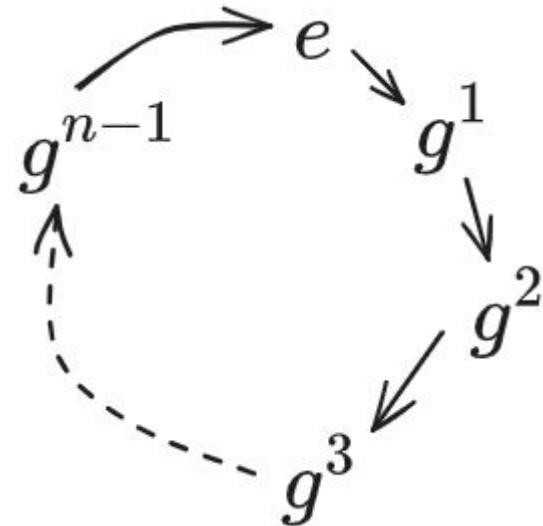
が成り立つとき、可換群という。例: 整数全体の集合 Z 上の加算 $(Z, +)$

巡回群ってなんだ

群**G**が巡回群とは、

- $g^1 = g$
- $g^2 = g * g$
- $g^3 = g^2 * g = (g * g) * g$

のように、ある適当な元 g のべき乗(群演算を繰り返した物)を
繰り返すと、群**G**のすべての元を作り切れるような可換群



モジュラ乗算を用いた巡回群の例

群Gの元: $\{1, 2, 3, 4, 5, 6\}$

演算: 二項を掛け合わせて、素数7で割ったあまり

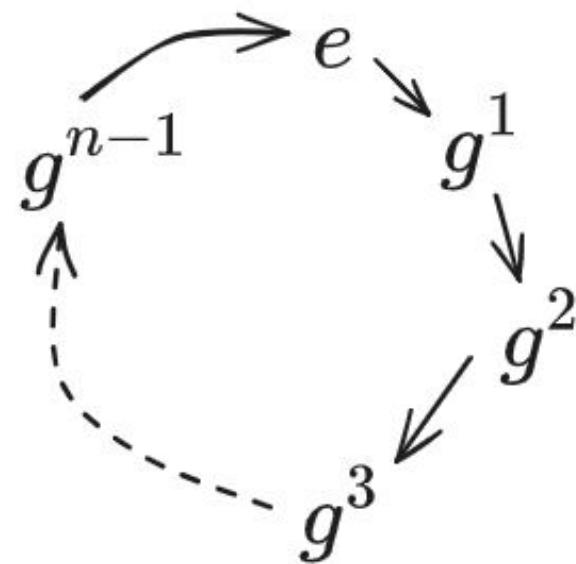
生成元 $g = 3$

$$g^1 \equiv 3 \pmod{7} \quad g^4 \equiv 4 \pmod{7}$$

$$g^2 \equiv 2 \pmod{7} \quad g^5 \equiv 5 \pmod{7}$$

$$g^3 \equiv 6 \pmod{7} \quad g^6 \equiv 1 \pmod{7}$$

生成元 g ですべての元を生成できているので、これは巡回群
※逆元の存在等の説明は省く



巡回群における離散対数問題とDiffie-Hellman問題

群Gを巡回群、gを生成元とする

- **離散対数問題 (DLP)**
 - g と自然数 n から、 g^n は簡単に求められる
 - g と g^n から、 n を求めるることは現実的に困難
- **Diffie-Hellman(DH)問題**
 - g^a と g^b から、 $g^{(a+b)}$ は簡単に求められる
 - しかし、 g^a と g^b から $g^{(ab)}$ を求めるることは困難

DH鍵交換・DSA署名など、多くの暗号プロトコルがこれらの困難性に基づく

ペアリング～魔法の写像～

巡回群 $G_1 \cdot G_2 \cdot G_3$ と、双線形性を持つ写像 $en(G_1 \times G_2 \Rightarrow G_3)$ を考える

$$en(g1^a, g2^b) = en(g1, g2^{ab}) = en(g1^{ab}, g2) = en(g1, g2)^{ab}$$

が成り立つことを双線形性といい、双線形性を持つ写像をペアリングという。

※驚くべきことに、ペアリングの計算コストは指数部に対して $O(1)$

ペアリング～魔法の写像～

巡回群 $G_1 \cdot G_2 \cdot G_3$ と、双線形性を持つ写像 $en(G_1 \times G_2 \Rightarrow G_3)$ を考える

$$en(g1^a, g2^b) = en(g1, g2^{ab}) = en(g1^{ab}, g2) = en(g1, g2)^{ab}$$

が成り立つことを双線形性といい、双線形性を持つ写像をペアリングという。

※驚くべきことに、ペアリングの計算コストは指数部に対して $O(1)$

え！？DH問題壊れてない！？いいの！？

ペアリング～魔法の写像～

巡回群G1・G2・G3と、双線形性を持つ写像 $en(G1 \times G2 \Rightarrow G3)$ を考える

$$en(g1^a, g2^b) = en(g1, g2^{ab}) = en(g1^{ab}, g2) = en(g1, g2)^{ab}$$

が成り立つことを双線形性といい、双線形性を持つ写像をペアリングという。

※驚くべきことに、ペアリングの計算コストは指数部に対してO(1)

え！？DH問題壊れてない！？いいの！？

=> 終域が別の群 G3なので、いいんです！！！ (雑な説明)

ペアリング仕組みは複雑すぎるので、性質を受け入れてください



ここでは簡便さのため、ペアリング=Type3ペアリングとする

ペアリングとSchwartz–Zippelの補題

Schwartz–Zippelの補題

n 次の多項式 $f(x)$ と $g(x)$ が同じか違うか判定したいなら、

ランダムな値 s を選び、 $f(s) = g(s)$ なら同じ、 $f(s) \neq g(s)$ なら違うと見なしていい

Schwartz–Zippeの補題

n 次の多項式 $f(x)$ と $g(x)$ が同じか違うか判定したいなら、

ランダムな値 s を選び、 $f(s) = g(s)$ なら同じ、 $f(s) \neq g(s)$ なら違うと見なしていい

=>ほんまか？



Schwartz–Zippelの補題

n 次の多項式 $f(x)$ と $g(x)$ が同じか違うか判定したいなら、

ランダムな値 s を選び、 $f(s) = g(s)$ なら同じ、 $f(s) \neq g(s)$ なら違うと見なしていい

$h(x) = f(x) - g(x)$ としたとき、 $h(x)$ は高々 n 次の多項式のため、
方程式 $h(x) = 0$ の解は高々 n 個しかない

つまり、 s を非常に大きな集合 (例: 2^{256} 要素など) から選ぶなら、
偽陽性を出す確率は $n/2^{256}$ となり、無視できる

閑話休題、 $g^{f(s)}$ を s を用いずに計算できるか！？

2次の $f(x) = ax^2 + bx + c$ について、 $g^{f(s)}$ を計算する場合を考える
未知の s に対して、 g, g^s, g^{s^2} が与えられたとき、 $g^{f(s)}$ を

$$g^{f(s)} = ag^{s^2} * bg^s * cg$$

として計算することができる

信頼できる第三者(Trusted Third-Party; TTP)がランダムな s を生成した後、
 $crs_s := g, g^s, g^{s^2} \dots g^{s^n}$ を配布し、 s を破棄する。

crs_s を用いると、誰でも $g^{f(s)}$ を s を知らずに計算できる

Schwartz–Zipperの補題とペアリングのコンボでbingo

証明者が検証者に、多項式 $p(x), h(x), t(x)$ が

$p(x) = h(x)t(x)$ の関係性であると簡潔に証明したい、場合を考える

Schwartz–Zippelの補題とペアリングのコンボでbingo

証明者が検証者に、多項式 $p(x), h(x), t(x)$ が

$p(x) = h(x)t(x)$ の関係性であると簡潔に証明したい、場合を考える

- 前述の補題より、ランダムな s を選び、 $p(s) = h(s)t(s)$ を検証できればいい

Schwartz–Zipperの補題とペアリングのコンボでbingo

証明者が検証者に、多項式 $p(x), h(x), t(x)$ が

$p(x) = h(x)t(x)$ の関係性であると簡潔に証明したい、場合を考える

- 前述の補題より、ランダムな s を選び、 $p(s) = h(s)t(s)$ を検証できればいい
- TTP が crs_s を計算し、配布する
- 証明者は crs_s で $g^{p(s)}, g^{h(s)}, g^{t(s)}$ を計算し、検証者に送信する
- 検証者は $en(g^{p(s)}, g) = en(g^{h(s)}, g^{t(s)})$ で、主張を検証できる

Schwartz–Zippeの補題とペアリングのコンボでbingo

証明者が検証者に、多項式 $p(x), h(x), t(x)$ が

$p(x) = h(x)t(x)$ の関係性であると簡潔に証明したい、場合を考える

- 前述の補題より、ランダムな s を選び、 $p(s) = h(s)t(s)$ を検証できればいい
- TTP が crs_s を計算し、配布する
- 証明者は crs_s で $g^{p(s)}, g^{h(s)}, g^{t(s)}$ を計算し、検証者に送信する
- 検証者は $en(g^{p(s)}, g) = en(g^{h(s)}, g^{t(s)})$ で、主張を検証できる

このプロトコルは簡潔(ペアリング2回)に検証できる！！！！！

=> Schwartz–Zippeの補題 & CRSの生成 & ペアリングでパズルがハマりだす！！

Schwartz–Zippelの補題とペアリングのコンボでbingo

証明者が検証者に、多項式 $p(x), h(x), t(x)$ が

$p(x) = h(x)t(x)$ の関係性であると簡潔に証明したい、場合を考える

1. 前述の補題より、ランダムな s を選び、 $p(s) = h(s)t(s)$ を検証できればいい
2. TTP が crs_s を計算し、配布する
3. 証明者は crs_s で $g^{p(s)}, g^{h(s)}, g^{t(s)}$ を計算し、検証者に送信する
4. 検証者は $en(g^{p(s)}, g) = en(g^{h(s)}, g^{t(s)})$ で、主張を検証できる

このプロトコルは簡潔(ペアリング2回)に検証できる！！！！！

=> Schwartz–Zippelの補題 & CRSの生成 & ペアリングでパズルがハマりだす！！

=> ちょま、証明者が正しく $g_{p(s)}$ などを計算した保証なくね？まあ,,, いったんパス

Pinocchioプロトコル作れそうじゃね？

Pinocchioプロトコル作れそうじゃね？

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

Pinocchioプロトコル作れそうじゃね？

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
2. 証明者は自身が持つ変数ベクトル w を用いて
 $A(x) = AP \cdot w, B(x) = BP \cdot w, C(x) = CP \cdot w$ を計算する
そして、 $P(x) = A(x)B(x) - C(x)$ を計算し、
 $h(x) = P(x)/(x - 1)(x - 2)(x - 3)$ となるような、 $h(x)$ を求める

さらに、QAPの中間変数 $(i1, i2)$ の部分だけ個別に計算する

証明者: 中間変数にかかる多項式だけ束ねる

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

入出力以外、中間変数部分だけ取り出す

↗ $AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$

証明者: 中間変数にかかる多項式だけ束ね続ける

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

入出力以外、中間変数部分だけ取り出す

A diagram illustrating the extraction of intermediate variable components. A dashed orange box encloses the bottom two rows of the vector AP: \$A_{i1}(x)\$ and \$A_{i2}(x)\$. An orange arrow points from this box to the vector \$AP_{mid}\$.

$$AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

$$w = \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix}$$

変数ベクトルに関しても同様

A diagram illustrating the extraction of variable components. A dashed orange box encloses the bottom two rows of the vector w: \$i1\$ and \$i2\$. An orange arrow points from this box to the vector \$w_{mid}\$.

$$w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix}$$

証明者: 中間変数にかかる多項式だけ束ね続ける

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

入出力以外、中間変数部分だけ取り出す

$$AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

$$w = \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix}$$

BP、CPも同様

$$w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix}$$
$$BP_{mid} = \begin{pmatrix} B_{i1}(x) \\ B_{i2}(x) \end{pmatrix}$$
$$CP_{mid} = \begin{pmatrix} C_{i1}(x) \\ C_{i2}(x) \end{pmatrix}$$

証明者: 中間変数にかかる多項式だけ束ね続ける

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ \vdots \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

入出力以外、中間変数部分だけ取り出す

$$AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix}$$

BP、CPも同様

$$w = \begin{pmatrix} one \\ x \\ out \\ \vdots \\ i1 \\ i2 \end{pmatrix}$$

$$w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix}$$

$$BP_{mid} = \begin{pmatrix} B_{i1}(x) \\ B_{i2}(x) \end{pmatrix}$$

$$CP_{mid} = \begin{pmatrix} C_{i1}(x) \\ C_{i2}(x) \end{pmatrix}$$

中間変数だけで多項式を束ねる

$$A_{mid}(x) = AP_{mid} \cdot w_{mid}$$

$$B_{mid}(x) = BP_{mid} \cdot w_{mid}$$

$$C_{mid}(x) = CP_{mid} \cdot w_{mid}$$

証明者: 必要な多項式を巡回群に乗せる

$$A(x) = AP \cdot w$$

$$B(x) = BP \cdot w$$

$$C(x) = CP \cdot w$$

$$P(x) = A(x)B(x) - C(x)$$

$$h(x) = P(x)/(x-1)(x-2)(x-3)$$

$$A_{mid}(x) = AP_{mid} \cdot w_{mid}$$

$$B_{mid}(x) = BP_{mid} \cdot w_{mid}$$

$$C_{mid}(x) = CP_{mid} \cdot w_{mid}$$

crs_s を用いて必要な多項式を巡回群に乗せる

$$g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$$

証明者の手続きをまとめる

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
2. 証明者は自身が持つ変数ベクトル w と crs_s を用いて
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ を計算し、
上記の4つの元と、入出力の変数 $w_{io} = (1, 2, 46)$ を検証者に送る

さあ検証開始だ！！！

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
2. 証明者は自身が持つ変数ベクトル w と crs_s を用いて
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ を計算し、
上記の4つの元と、入出力の変数 $w_{io} = (1, 2, 46)$ を検証者に送る
3. 検証者は受け取った入出力 w_{io} と $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ が
QAPを満たすことを、ペアリングを用いていい感じに検証する

検証者: 貰った入出力でIO部分の多項式をまとめる

AP, BP, CPの入出力の部分だけ取り出す

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} \quad AP_{io} = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \end{pmatrix} \quad BP_{io} = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \end{pmatrix}$$
$$CP_{io} = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \end{pmatrix}$$

証明者から受け取った w_{io} で、
入出力にまつわる部分をまとめる

$$A_{io}(x) = AP_{io} \cdot w_{io}$$

$$B_{io}(x) = BP_{io} \cdot w_{io}$$

$$C_{io}(x) = CP_{io} \cdot w_{io}$$

crs_sを用いて計算
----->

$$g^{A_{io}(s)}, g^{B_{io}(s)}, g^{C_{io}(s)}$$

検証者: 検証者がそろえた情報の整理

TTPが公開したやつ

$$t(x) = (x - 1)(x - 2)(x - 3), \quad g^{t(s)}$$

証明者から受け取ったやつ

$$g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$$

検証者が計算したやつ

$$g^{A_{io}(s)}, g^{B_{io}(s)}, g^{C_{io}(s)}$$

検証者: QAPの多項式を復元する

TPPが公開したやつ

$$t(x) = (x - 1)(x - 2)(x - 3), \quad g^{t(s)}$$

証明者から受け取ったやつ

$$g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$$

検証者が計算したやつ

$$g^{A_{io}(s)}, g^{B_{io}(s)}, g^{C_{io}(s)}$$

$$g^{A(s)} = g^{A_{io}(s)+A_{mid}(s)} = g^{A_{io}(s)} * g^{A_{mid}(s)}$$

$$g^{B(s)} = g^{B_{io}(s)+B_{mid}(s)} = g^{B_{io}(s)} * g^{B_{mid}(s)}$$

$$g^{C(s)} = g^{C_{io}(s)+C_{mid}(s)} = g^{C_{io}(s)} * g^{C_{mid}(s)}$$

IO部分と中間部分を足し合わせて、
多項式全体を計算する

検証者: QAPが満たされるか、ペアリングで検証！！

TTPが公開したやつ

$$t(x) = (x - 1)(x - 2)(x - 3), \quad g^{t(s)}$$

証明者から受け取ったやつ

$$g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$$

検証者が計算したやつ

$$g^{A_{io}(s)}, g^{B_{io}(s)}, g^{C_{io}(s)}$$

$$g^{A(s)} = g^{A_{io}(s)+A_{mid}(s)} = g^{A_{io}(s)} * g^{A_{mid}(s)}$$

$$g^{B(s)} = g^{B_{io}(s)+B_{mid}(s)} = g^{B_{io}(s)} * g^{B_{mid}(s)}$$

$$g^{C(s)} = g^{C_{io}(s)+C_{mid}(s)} = g^{C_{io}(s)} * g^{C_{mid}(s)}$$

IO部分と中間部分を足し合わせて、
多項式全体を計算する

QAPの検証！！！！

$$en(g^{A(s)}, g^{B(s)}) \stackrel{?}{=} en(g^{h(s)}, g^{t(s)}) * en(g^{C(s)}, g)$$

検証者: 正しく検証できているか、式変形してみる

QAPにまつわる多項式の復習

$$\begin{aligned} A(x) &= AP \cdot w, & B(x) &= BP \cdot w, & C(x) &= CP \cdot w \\ P(x) &= A(x) * B(x) - C(x), & t(x) &= (x-1)(x-2)(x-3) \\ P(x) &= h(x)t(x) \end{aligned}$$

-----P(x)がt(x)で割り切れたら、QAPを満たす

$$en(g^{A(s)}, g^{B(s)}) \stackrel{?}{=} en(g^{h(s)}, g^{t(s)}) * en(g^{C(s)}, g)$$

検証者: 正しく検証できているか、式変形してみる

QAPにまつわる多項式の復習

$$\begin{aligned} A(x) &= AP \cdot w, & B(x) &= BP \cdot w, & C(x) &= CP \cdot w \\ P(x) &= A(x) * B(x) - C(x), & t(x) &= (x-1)(x-2)(x-3) \\ P(x) &= h(x)t(x) \end{aligned}$$

----- P(x) が t(x) で割り切れたら、QAP を満たす

$$en(g^{A(s)}, g^{B(s)}) \stackrel{?}{=} en(g^{h(s)}, g^{t(s)}) * en(g^{C(s)}, g)$$

$$\begin{aligned} en(g^{A(s)}, g^{B(s)}) &= en(g, g)^{A(s)B(s)} && \text{----- } P(x) + C(x) = A(x) + B(x) \\ &= en(g, g)^{P(s)+C(s)} \\ &= en(g, g)^{h(s)t(s)+C(s)} \\ &= en(g, g)^{h(s)t(s)} * en(g, g)^{C(s)} \\ &= en(g^{h(s)}, g^{t(s)}) * en(g^{C(s)}, g) \end{aligned}$$

----- 右辺と一致するぜ

できた！！Pinochhio！！

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
2. 証明者は自身が持つ変数ベクトル w と crs_s を用いて
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ を計算し、
上記の4つの元と、入出力の変数 $w_{io} = (1, 2, 46)$ を検証者に送る
3. 検証者は受け取った元と、 $t(x)$ をもとにペアリングを用いて、
証明者の入出力が正しい事を検証する。

簡潔だぜ！！これがPinocchioだ！！

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
2. 証明者は自身が持つ変数ベクトル w と crs_s を用いて
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ を計算し、
上記の4つの元と、入出力の変数 $w_{io} = (1, 2, 46)$ を検証者に送る
3. 検証者は受け取った元と、 $t(x)$ をもとにペアリングを用いて、
証明者の入出力が正しい事を検証する。

証明者から受け渡される情報は、入出力のほかに常に元4つで、
検証コストも入出力サイズに依存し、ゲート数に対しては一定！！！！！

実はできない、Pinocchio。健全性がない

証明者が検証者に、多項式 $5x^3 + 3x$ に対応する
QAPの多項式の列 AP, BP, CP に対し、
入出力 $x = 2, out = 46$ が正しいと簡潔に証明したい

1. TTPが crs_s と $t(x) = (x - 1)(x - 2)(x - 3)$ に対する $g^{t(s)}$ を配布する
 2. 証明者は自身が持つ変数ベクトル w と crs_s を用いて
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ を計算し、
上記の4つの元と、入出力の変数 $w_{io} = (1, 2, 46)$ を検証者に送る
 3. 検証者は受け取った元と、 $t(x)$ をもとにペアリングを用いて、
証明者の入出力が正しい事を検証する。
- 正しく元を
計算した保証がない

証明者から受け渡される情報は、入出力のほかに常に元4つで、
検証コストも入出力サイズに依存し、ゲート数に対しては一定！！！！！

現時点で証明者ができうる不正

$$A_{fake}(x) = A_{io}(x) + A_{fake_mid}(x)$$

$$B_{fake}(x) = B_{io}(x) + B_{fake_mid}(x)$$

$$C_{fake}(x) = C_{io}(x) + C_{fake_mid}(x)$$

証明者は

※黄枠は証明者が提出者に提出するもの

$$A_{invaded}(x)B_{invaded}(x) - C_{invaded}(x) = h_{fake}(x)t(x)$$

となるような、 AP, BP, CP と関係ない適当な

$$A_{fake_mid}(x), B_{fake_mid}(x), C_{fake_mid}(x), h_{fake}(x)$$

を求め、 crs_s で元に埋め込み検証者に送ることができる。

前述のプロトコルでは、この不正に検証者は気づけない！！

どんなチェックを追加すればいいか考える

でも、先述のプロトコルはいい線行ってそう。健全性を満たすため、
 $g^{h(s)}, g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ に対するチェックを追加したい

正しい $A(x), B(x), C(x)$ とは、

それぞれ AP, BP, CP と適当なベクトルとの内積で作られる

=>つまり、 **AP, BP, CP の成分の線形結合に制約される**

※ここで言う線形結合とは、

多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ が与えられた際、係数 a_1, \dots, a_n を用いて、
 $p(x) = a_1u_1(x) + a_2u_2(x) + \dots + a_nu_n(x)$ のように表せる多項式 $p(x)$ のこと

$g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ の指數 $A_{mid}(s), B_{mid}(s), C_{mid}(s)$ が、
それぞれ $AP_{mid}, BP_{mid}, CP_{mid}$ の成分の線形結合か検証できれば、
不正な多項式を埋め込んだ元の提出を防げそう？？

まだできる証明者の不正と、さらなる追加のチェック

前述のチェックを追加できれば、

$g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ がAP, BP, CP由来だと検証できる

一方で、QAPを正しく機能させるには、

$$A_{mid}(x) = AP \cdot w_{a_mid}, B_{mid}(x) = BP \cdot w_{b_mid}, C_{mid}(x) = CP \cdot w_{c_mid}$$
$$P_{mid}(x) = A_{mid}(x)B_{mid}(x) - C_{mid}(x)$$

の変数ベクトル $w_{a_mid}, w_{b_mid}, w_{c_mid}$ がすべて同じ必要があるが、
前述のチェックのみではこれは保証でない

$g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ の指数 $A_{mid}(s), B_{mid}(s), C_{mid}(s)$ が、
それぞれ $AP_{mid}, BP_{mid}, CP_{mid}$ と同一のベクトルと内積を取った結果、
つまり、線形結合の係数が同じことを検証できれば良さそう

なんちやってPinocchoのまとめ

1. 完全性を満たし、簡潔なVCはできた
2. しかし、健全性を満たせておらず次の二つのチェックの追加が必要
 - a. **線形結合チェック (α -チェック):**
 $g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)}$ が、
それぞれ AP, BP, CP の成分の線形結合であること
 - b. **係数の一貫性チェック (β -チェック):**
上記の線形結合の指数が、すべて同じであること

Knowledge of Exponent Assumption

直感的なKnowledge of Exponent Assumption (KEA1)

$A = g^s, B = A^\alpha = g^{\alpha s}$ となるような値のペア(A, B)を、
 α が分からないまま与えられたとき、
 $D = C^\alpha$ の関係性の新しいペア(C, D)を求めることはできるか！？

単純に、任意の自然数 k を用意し、 $C = A^k, D = B^k$ を計算すれば、
 $D = g^{\alpha sk}, C = g^{sk}$ なので、 $D = C^\alpha$ を満せる！！

直感的なKnowledge of Exponent Assumption (KEA1)

$A = g^s, B = A^\alpha = g^{\alpha s}$ となるような値のペア(A, B)を、
 α が分からないまま与えられたとき、
 $D = C^\alpha$ の関係性の新しいペア(C, D)を求めることはできるか！？

単純に、任意の自然数 k を用意し、 $C = A^k, D = B^k$ を計算すれば、
 $D = g^{\alpha sk}, C = g^{sk}$ なので、 $D = C^\alpha$ を満せる！！

KEA1 (Knowledge of Exponent Assumption)

「未知の α, s に対し、 $A = g^s, B = A^\alpha$ から $D = C^\alpha$ となるペアを作るには、
上記のように $C = A^k, D = B^k$ という形で作る以外方法はない。
つまり、ペアを作れたということは、その作者は指数 k を知っているハズだ」という仮定。(※非常にインフォーマル)

ペアに対するKEAと一般化したq-PKE

未知の自然数 α, s に対し、指數が α 倍の二つのペア $(g^s, g^{\alpha s}), (g^{s^2}, g^{\alpha s^2})$ が与えられた際、
 $g^{3s^2+2s} = (g^{s^2})^3 * (g^s)^2, \quad g^{\alpha(3s^2+2s)} = (g^{\alpha s^2})^3 * (g^{\alpha s})^2$
のようなペアを定数倍して足し合わせた形で、新たな指數が α 倍のペアは作れる

しかし、 $g^{\alpha s^3}$ は、DH問題により $g^{\alpha s}, g^{\alpha s^2}$ 等から作ることが困難で、ペアも作れない

ペアに対するKEAと一般化したq-PKE

未知の自然数 α, s に対し、指数が α 倍の二つのペア $(g^s, g^{\alpha s}), (g^{s^2}, g^{\alpha s^2})$ が与えられた際、
 $g^{3s^2+2s} = (g^{s^2})^3 * (g^s)^2, \quad g^{\alpha(3s^2+2s)} = (g^{\alpha s^2})^3 * (g^{\alpha s})^2$
のようなペアを定数倍して足し合わせた形で、新たな指数が α 倍のペアは作れる

しかし、 $g^{\alpha s^3}$ は、DH問題により $g^{\alpha s}, g^{\alpha s^2}$ 等から作ることが困難で、ペアも作れない

=> $g^p, g^{\alpha p}$ のような指数が α 倍のペアを作る際、

KEA1とDH問題により、 p は指数の線形結合($p = as^2 + bs$)に制限される

ペアに対するKEAと一般化したq-PKE

未知の自然数 α, s に対し、指数が α 倍の二つのペア $(g^s, g^{\alpha s}), (g^{s^2}, g^{\alpha s^2})$ が与えられた際、
 $g^{3s^2+2s} = (g^{s^2})^3 * (g^s)^2, \quad g^{\alpha(3s^2+2s)} = (g^{\alpha s^2})^3 * (g^{\alpha s})^2$
のようなペアを定数倍して足し合わせた形で、新たな指数が α 倍のペアは作れる

しかし、 $g^{\alpha s^3}$ は、DH問題により $g^{\alpha s}, g^{\alpha s^2}$ 等から作ることが困難で、ペアも作れない

=> $g^p, g^{\alpha p}$ のような指数が α 倍のペアを作る際、

KEA1とDH問題により、 p は指数の線形結合($p = as^2 + bs$)に制限される

q-PKE (q-Power Knowledge of Exponent) ※非常にインフォーマル

q-PKEは上記の議論を一般化し、
「未知の整数 α, s に対し、 $(g^s, g^{\alpha s}), (g^{s^2}, g^{\alpha s^2}), \dots, (g^n, g^{\alpha s^n})$ が与えられた際、
もし誰かが $g^p, g^{\alpha p}$ のような指数が α 倍のペアを作れるなら、
その指数 p は s, s^2, \dots, s^n の線形結合($p = \sum_{k=1}^n c_k s^k$)であり、
その誰かは係数 c_1, c_2, \dots, c_n を知っているだろう」という仮定

$g^{\{p\}}$ と $g^{\{\alpha p\}}$ の指数部分のチェックについて

さっきから $g^p, g^{\alpha p}$ を計算できたなら、って言ってるけど、
ある二つの元 $A = g^?, B = g^?$ の指数が α 倍の関係かどうか検証すればいい??

まず、追加で g^α が既知だとする(もちろん α は未知)。そして、

$$en(A, g^\alpha) \stackrel{?}{=} en(B, g)$$

のチェックに成功したら、 $B = A^\alpha$ であると確証できる

式変形: $en(g^p, g^\alpha) = en(g, g)^{\alpha p} = en(g^{\alpha p}, g)$

q-PKEを応用した、多項式の列に対する線形結合チェック

十分に独立した既知の多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ に対して、
未知のランダムな整数 α, s を用いて計算された、
 $(g^{u_1(s)}, g^{\alpha u_1(s)}), (g^{u_2(s)}, g^{\alpha u_2(s)}), \dots, (g^{u_n(s)}, g^{\alpha u_n(s)})$ が既知だとする

q-PKEを仮定すると、 $g^p, g^{\alpha p}$ を作る際、 p は $u_1(s), u_2(s), \dots, u_n(s)$ の線形結合、つまり p を適当な多項式 $P(x)$ を用いて $p = P(s)$ と表現した際、 $P(x)$ は多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ の線形結合に制限される

q-PKEを応用した、多項式の列に対する線形結合チェック

十分に独立した既知の多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ に対して、未知のランダムな整数 α, s を用いて計算された、 $(g^{u_1(s)}, g^{\alpha u_1(s)}), (g^{u_2(s)}, g^{\alpha u_2(s)}), \dots, (g^{u_n(s)}, g^{\alpha u_n(s)})$ が既知だとする

q-PKEを仮定すると、 $g^p, g^{\alpha p}$ を作る際、 p は $u_1(s), u_2(s), \dots, u_n(s)$ の線形結合、つまり p を適当な多項式 $P(x)$ を用いて $p = P(s)$ と表現した際、 $P(x)$ は多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ の線形結合に制限される

既知の多項式の列 $U = \{(u_1(x), \dots, u_n(x))\}$ に対して、TTP がランダムな α, s を用いて、各多項式に対応するペア $crs_{\alpha U(s)} := \{(g^{u_1(s)}, g^{\alpha u_1(s)}), (g^{u_2(s)}, g^{\alpha u_2(s)}), \dots, (g^{u_n(s)}, g^{\alpha u_n(s)})\}$

と g^α を計算し、配布する。もし誰かが $en(A, g^\alpha) = en(B, g)$ となるような、 $A = g^p, B = g^{\alpha p}$ を計算できたなら、 p に対応する多項式 $P(x)$ は、多項式の列 U の成分の線形結合であることが保証される。



なんちゃって Pinocchio で欠けていた、
線形結合チェック (α -チェック) ができた！！！！！！

q -PKEを仮定すると、 $g^p, g^{\alpha p}$ を作る際、 p は $u_1(s), u_2(s), \dots$ 、つまり p を適当な多項式 $P(x)$ を用いて $p = P(s)$ と表現した際、 $P(x)$ は多項式の列 $u_1(x), u_2(x), \dots, u_n(x)$ の線形結合に制限される

既知の多項式の列 $U = \{(u_1(x), \dots, u_n(x))\}$ に対して、
TTPがランダムな α, s を用いて、各多項式に対応するペア
 $crs_{\alpha U(s)} := \{(g^{u_1(s)}, g^{\alpha u_1(s)}), (g^{u_2(s)}, g^{\alpha u_2(s)}), \dots, (g^{u_n(s)}, g^{\alpha u_n(s)})\}$

と g^α を計算し、配布する。もし誰かが $en(A, g^\alpha) = en(B, g)$ となるような、
 $A = g^p, B = g^{\alpha p}$ を計算できたなら、 p に対応する多項式 $P(x)$ は、
 多項式の列 U の成分の線形結合であることが保証される。

$u_n(x)$ に対して、



KEAを用いた、係数の一貫性チェックの兆し

未知のランダムな値 s_1, s_2, s_3 に対し、 $g^{s_1}, g^{s_2}, g^{s_3}$ が既知であるとする

誰かがある元 $A = g^{s_1 k_1}, B = g^{s_2 k_2}, C = g^{s_3 k_3}$ を提出し、
指数が同じ ($k := k_1 = k_2 = k_3$) であると主張した際、それを k ナシで検証できるか？

KEAを用いた、係数の一貫性チェックの兆し

未知のランダムな値 s_1, s_2, s_3 に対し、 $g^{s_1}, g^{s_2}, g^{s_3}$ が既知であるとする

誰かがある元 $A = g^{s_1 k_1}, B = g^{s_2 k_2}, C = g^{s_3 k_3}$ を提出し、
指数が同じ ($k := k_1 = k_2 = k_3$) であると主張した際、それを k ナシで検証できるか？

ここで新たに未知でランダムな値 β に対し、
バンドル値 $K = (g^{s_1} * g^{s_2} * g^{s_3})^\beta = g^{\beta(s_1+s_2+s_3)}$ が与えられたとする

もしその誰かが、 A, B, C に加え、 $Z = ?$ $(A * B * C)^\beta$ を満たすような
 Z を提出できたなら、KEAにより k を知っていて $Z = K^k$ を計算したとみなせる

KEAを用いた、係数の一貫性チェックの兆し

未知のランダムな値 s_1, s_2, s_3 に対し、 $g^{s_1}, g^{s_2}, g^{s_3}$ が既知であるとする

誰かがある元 $A = g^{s_1 k_1}, B = g^{s_2 k_2}, C = g^{s_3 k_3}$ を提出し、
指数が同じ ($k := k_1 = k_2 = k_3$) であると主張した際、それを k ナシで検証できるか？

ここで新たに未知でランダムな値 β に対し、
バンドル値 $K = (g^{s_1} * g^{s_2} * g^{s_3})^\beta = g^{\beta(s_1 + s_2 + s_3)}$ が与えられたとする
もしその誰かが、 A, B, C に加え、 $Z = (A * B * C)^\beta$ [?]を満たすような
 Z を提出できたなら、KEAにより k を知っていて $Z = K^k$ を計算したとみなせる

したがって、上述のテストを満たす Z を作れたということは、
 A, B, C も共通の k を使って作られたと見なせる

直感的には、ランダムな β で「束ねられた」 K に対して $Z = (A * B * C)^\beta$ を満たすには、
三つの元 $g^{s_1}, g^{s_2}, g^{s_3}$ に共通の指数 k を掛ける形で A, B, C を作るしかない

$\$Z = (A^*B^*C)^\beta$ の検証もペアリングができる

では、 $Z = (A * B * C)^\beta$ だとどう検証すればいい？

はい、察しの通りペアリングです

線形結合チェックと同様、追加で g^β が与えられ、

$$en(Z, g) \stackrel{?}{=} en(A * B * C, g^\beta)$$

のチェックに成功すれば、 $Z = (A * B * C)^\beta$ だと検証できる

式変形

$$\begin{aligned} en(Z, g) &= en(K^k, g) = en(g^{\beta k(s_1+s_2+s_3)}, g) = en(g, g)^{\beta k(s_1+s_2+s_3)} \\ &= en(g^{k(s_1+s_2+s_3)}, g^\beta) = en(g^{s_1 k} * g^{s_2 k} * g^{s_3 k}, g^\beta) \\ &= en(A * B * C, g^\beta) \end{aligned}$$

多項式の列の線形結合に対する係数の一貫性チェック

これまでの考察を踏まえ、係数の一貫性チェックの方法をまとめる

十分に独立した既知の多項式列のセットが三つあり、

$$VP = \{v_1(x), \dots, v_n(x)\} \quad WP = \{w_1(x), \dots, w_n(x)\}$$

$$YP = \{y_1(x), \dots, y_n(x)\}$$

それに対しTPPがランダムな β, s を用いて、バンドル値

$$K_{VP,WP,YP} := \{g^{\beta(v_1(s)+w_1(s)+y_1(s))}, \dots, g^{\beta(v_n(s)+w_n(s)+y_n(s))}\}$$

を計算し、配布する

もし誰かが $en(Z, g) = en(A * B * C, g^\beta)$ となるような A, B, C, Z を計算でき、
 A, B, C が AP, BP, CP の線形結合であると保証されているなら、
その線形結合に使われた係数 k_1, k_2, \dots, k_n (内積の相手)は、三つの間で同一である。



多項式の列の線形結合に対する係数の

これまでの考察を踏まえ、依然

十八

なんちゃって Pinocchio で欠けていた、
係数の一貫性チェック (β -チェック) ができた!!!!

$$WP = \{w_1(x), \dots, w_n(x)\}$$

$$v_1(x), \dots, y_n(x)\}$$

それに対し TTP がランダムな β, s を用いて、バンドル値

$$K_{VP, WP, YP} := \{g^{\beta(v_1(s) + w_1(s) + y_1(s))}, \dots, g^{\beta(v_n(s) + w_n(s) + y_n(s))}\}$$

を計算し、配布する

もし誰かが $en(Z, g) = en(A * B * C, g^\beta)$ となるような A, B, C, Z を計算でき、
 A, B, C が AP, BP, CP の線形結合であると保証されているなら、
その線形結合に使われた係数 k_1, k_2, \dots, k_n (内積の相手) は、三つの間で同一である。⁹¹



そろそろさすがに疲れてきましたね。

ただ、本章では

1. 線形結合チェックの方法
2. 係数の一貫性チェックの方法

がわかつてきました。と、いうことは？



The Pinocchioプロトコル



復習: 多項式 => R1CS => QAP

命題の術語

$$5x^3 + 3x$$

Flattening

ゲートの列

$$i1 = 5x * x$$

$$i2 = i1 * x$$

$$out = i2 + 3x$$

$x=2$ で評価

$x=2$ 時の各変数

$$x = 2$$

$$i1 = 5x * x = 20$$

$$i2 = i1 * x = 40$$

$$out = i2 + 3x = 46$$

ベクトルに詰める

$$w = \begin{pmatrix} one \\ x \\ out \\ i1 \\ i2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 46 \\ 20 \\ 40 \end{pmatrix}$$

R1CS

$$A = \begin{pmatrix} 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 3 & 0 & 0 & 1 \end{pmatrix}, \quad B = \begin{pmatrix} 0 & 1 & 0 & 10 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \quad C = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

QAP

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix} \quad BP = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{3}{2}x - \frac{1}{2}x^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad CP = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{1}{2}x^2 - \frac{5}{2}tx + 3 \\ -x^2 + 4x - 3 \end{pmatrix}$$

青枠: 入出力にまつわる部分

緑枠: 中間変数にまつわる部分

復習: 多項式の列、変数の列を分割する

QAP

$$AP = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 4x^2 - 17x + 18 \\ 0 \\ -x^2 + 4x - 3 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \end{pmatrix} \quad BP = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} = \begin{pmatrix} \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{3}{2}x - \frac{1}{2}x^2 \\ 0 \\ 0 \\ 0 \end{pmatrix} \quad CP = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{2}x^2 - \frac{3}{2}x + 1 \\ \frac{1}{2}x^2 - \frac{5}{2}tx + 3 \\ -x^2 + 4x - 3 \end{pmatrix}$$

諸々分けたやつ

$$\begin{array}{lll} AP_{io} = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \end{pmatrix} & BP_{io} = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \end{pmatrix} & CP_{io} = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \end{pmatrix} \quad w_{io} = \begin{pmatrix} one \\ x \\ out \end{pmatrix} \\ AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} & BP_{mid} = \begin{pmatrix} B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} & CP_{mid} = \begin{pmatrix} C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} \quad w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix} \end{array}$$

青枠: 入出力にまつわる部分

緑枠: 中間変数にまつわる部分

復習: 分割したやつをくっつけて、QAPチェック

諸々分けたやつ

$$\begin{array}{ll}
 AP_{io} = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \end{pmatrix} & BP_{io} = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \end{pmatrix} \\
 CP_{io} = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \end{pmatrix} & w_{io} = \begin{pmatrix} one \\ x \\ out \end{pmatrix} \\
 AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{j2}(x) \end{pmatrix} & BP_{mid} = \begin{pmatrix} B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} \\
 CP_{mid} = \begin{pmatrix} C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} & w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix}
 \end{array}$$

分けたやつから
多項式の組み立て

$$\begin{array}{l}
 A_{io}(x) = AP_{io} \cdot w_{io}, \quad A_{mid}(x) = AP_{mid} \cdot w_{mid}, \quad A(x) = A_{mid}(x) + A_{io}(x) \\
 B_{io}(x) = BP_{io} \cdot w_{io}, \quad B_{mid}(x) = BP_{mid} \cdot w_{mid}, \quad B(x) = B_{mid}(x) + B_{io}(x) \\
 C_{io}(x) = CP_{io} \cdot w_{io}, \quad C_{mid}(x) = CP_{mid} \cdot w_{mid}, \quad C(x) = C_{mid}(x) + C_{io}(x)
 \end{array}$$

QAP Check

$$\begin{aligned}
 P(x) &= A(x)B(x) - C(x), \quad t(x) = (x-1)(x-2)(x-3) \\
 P(x) &= h(x)t(x)
 \end{aligned}$$

青枠: 入出力にまつわる部分

緑枠: 中間変数にまつわる部分

TTPによるトラステッドセットアップ！！

1. ランダムな値 $s, \alpha_a, \alpha_b, \alpha_c, \beta$ を選択する
2. 線形結合チェックのための以下のcrsを作成する

$$crs_{\alpha_a AP_{mid}(s)} = \{(g^{A_{i1}(s)}, g^{\alpha A_{i1}(s)}), (g^{A_{i2}(s)}, g^{\alpha A_{i2}(s)})\}$$

$$crs_{\alpha_b BP_{mid}(s)} = \{(g^{B_{i1}(s)}, g^{\alpha B_{i1}(s)}), (g^{B_{i2}(s)}, g^{\alpha B_{i2}(s)})\}$$

$$crs_{\alpha_c CP_{mid}(s)} = \{(g^{C_{i1}(s)}, g^{\alpha C_{i1}(s)}), (g^{C_{i2}(s)}, g^{\alpha C_{i2}(s)})\}$$

3. 係数の一貫性チェックのため、次のバンドル値の列を作成する

$$K_{AP_{mid}, BP_{mid}, CP_{mid}} = \{g^{\beta(A_{i1}(s)+B_{i1}(s)+C_{i1}(s))}, g^{\beta(A_{i2}(s)+B_{i2}(s)+C_{i2}(s))}\}$$

4. 証明者の $h(s)$ 計算や、検証者の $AP_{io}(s)$ 等計算用に crs_s を作成する

$$crs_s = \{g, g^s, g^{s^2}, \dots, g^{s^n}\}$$

5. 下記のパラメータをすべて公開し、 $s, \alpha_a, \alpha_b, \alpha_c, \beta$ は破棄する

$$g, g^{\alpha_a}, g^{\alpha_b}, g^{\alpha_c}, g^{\beta}, g^{t(s)}, crs_s$$

$$crs_{\alpha_a AP_{mid}(s)}, crs_{\alpha_b BP_{mid}(s)}, crs_{\alpha_c CP_{mid}(s)}, K_{AP_{mid}, BP_{mid}, CP_{mid}}$$

証明者による証明

1. QAPを充足する変数ベクトル w (例: $(1, 2, 46, 20, 40)$)を用意する
2. 次のように、 $h(x)$ を求め、 crs_s を用いて $g^{h(s)}$ も求める

$$h(x) = \frac{(AP \cdot w)(BP \cdot w) - CP \cdot w}{(x-1)(x-2)(x-3)}$$

3. $w_{mid} = (20, 40)$ と $crs_{\alpha_a AP_{mid}(s)}, crs_{\alpha_b BP_{mid}(s)}, crs_{\alpha_c CP_{mid}(s)}$ を用いて、

$$\boxed{g^{A_{mid}(s)}, g^{\alpha_a A_{mid}(s)}, g^{B_{mid}(s)}, g^{\alpha_b B_{mid}(s)}, g^{C_{mid}(s)}, g^{\alpha_c C_{mid}(s)}}$$

を計算する

線形結合チェック用

4. $K_{AP_{mid}, BP_{mid}, C_{mid}}$ を用いて、元 g^Z を計算する

$$\boxed{g^Z = g^{\beta(A_{mid}(s) + B_{mid}(s) + C_{mid}(s))}}$$
 係数の一貫性チェック用

5. 証明 π と入出力 w_{io} を検証者に送る

$$\boxed{\pi = (g^{h(s)}, g^{A_{mid}(s)}, g^{\alpha_a A_{mid}(s)}, g^{B_{mid}(s)}, g^{\alpha_b B_{mid}(s)}, g^{C_{mid}(s)}, g^{\alpha_c C_{mid}(s)}, g^Z)}$$

$$\boxed{w_{io} = (1, 2, 46)}$$
 ゲート数に関係なく、証明は常に元8個！！簡潔！！

検証者による、簡潔な検証！！！

1. 証明者から π と $w_{io} = (1, 2, 46)$ を受け取る

$$\pi = (g^{h(s)}, g^{A_{mid}(s)}, g^{\alpha_a A_{mid}(s)}, g^{B_{mid}(s)}, g^{\alpha_b B_{mid}(s)}, g^{C_{mid}(s)}, g^{\alpha_c C_{mid}(s)}, g^Z)$$

2. 入出力部分の元 $g^{A_{io}(s)}, g^{B_{io}(s)}, g^{C_{io}(s)}$ を crs_s で計算する
3. ペアリングでQAPチェックをする

$$en(g^{A_{io}(s)} * g^{A_{mid}(s)}, g^{B_{io}(s)} * g^{B_{mid}(s)}) \stackrel{?}{=} en(g^{t(s)}, g^{h(s)}) * en(g^{C_{io}(s)} * g^{C_{mid}(s)}, g)$$

4. ペアリングで線形結合チェックする

$$en(g^{\alpha_a A_{mid}(s)}, g) \stackrel{?}{=} en(g^{A_{mid}(s)}, g^{\alpha_a}) \quad en(g^{\alpha_b B_{mid}(s)}, g) \stackrel{?}{=} en(g^{B_{mid}(s)}, g^{\alpha_b})$$

$$en(g^{\alpha_c C_{mid}(s)}, g) \stackrel{?}{=} en(g^{C_{mid}(s)}, g^{\alpha_c})$$

ゲート数がどれだけ増えても、常にペアリング11回！！
簡潔！！！！！！

5. ペアリングで係数の一貫性チェックをする

$$en(g^Z, g) \stackrel{?}{=} en(g^{A_{mid}(s)} * g^{B_{mid}(s)} * g^{C_{mid}(s)}, g^\beta)$$

6. 全部のチェックに通過したら、 $w_{io}(1, 2, 46)$ が正しい充足の一部だと確信できる！！⁹⁹！



軽く一般化する

ここまでプロトコルは、 $5x^3 + 3x$ に対するもの。

しかし、R1CSで命題を記述できたら、何でもこのプロトコルに突っ込める
また、入出力ベクトルの長さや配置も好き放題いじることができる
例えば、 $w_{\{io\}}=(one,x,y,z,\dots,out1,out2,\dots)$ のように

また、R1CSはNPの計算クラスを表現できると証明されている
つまり、多項式時間で検証可能な任意の計算をエンコードできる

そのため、初めの問題設定で述べたVCとしてのPinocchioは達成した

ゼロ知識化！！！

隠したい入出力を中間変数に移す

問題設定：

$5x^3 + 3x = 46$ の解を知っていることを、 x を隠したまま証明したい

隠したい入出力を中間変数に移す

単純に、 x 部分を中間変数部分に移せば、 $g^{\{A_mid(x)\}}$ 等に埋め込まれ、 x の推測はかなり難しくなる。もちろん、QAPチェックや線形結合チェックは通ったまま

諸々分けたやつ

$$\begin{array}{llll} AP_{io} = \begin{pmatrix} A_1(x) \\ A_x(x) \\ A_{out}(x) \end{pmatrix} & BP_{io} = \begin{pmatrix} B_1(x) \\ B_x(x) \\ B_{out}(x) \end{pmatrix} & CP_{io} = \begin{pmatrix} C_1(x) \\ C_x(x) \\ C_{out}(x) \end{pmatrix} & w_{io} = \begin{pmatrix} one \\ x \\ out \end{pmatrix} \\ AP_{mid} = \begin{pmatrix} A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} & BP_{mid} = \begin{pmatrix} B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} & CP_{mid} = \begin{pmatrix} C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} & w_{mid} = \begin{pmatrix} i1 \\ i2 \end{pmatrix} \end{array}$$

x の部分を
中間変数部分に
移しちゃう

$$\begin{array}{llll} AP_{io} = \begin{pmatrix} A_1(x) \\ A_{out}(x) \end{pmatrix} & BP_{io} = \begin{pmatrix} B_1(x) \\ B_{out}(x) \end{pmatrix} & CP_{io} = \begin{pmatrix} C_1(x) \\ C_{out}(x) \end{pmatrix} & w_{io} = \begin{pmatrix} one \\ out \end{pmatrix} \\ AP_{mid} = \begin{pmatrix} A_x(x) \\ A_{i1}(x) \\ A_{i2}(x) \end{pmatrix} & BP_{mid} = \begin{pmatrix} B_x(x) \\ B_{i1}(x) \\ B_{i2}(x) \end{pmatrix} & CP_{mid} = \begin{pmatrix} C_x(x) \\ C_{i1}(x) \\ C_{i2}(x) \end{pmatrix} & w_{mid} = \begin{pmatrix} x \\ i1 \\ i2 \end{pmatrix} \end{array}$$

さらに強固にゼロ知識化する

QAP Checkは $P(x) = A(x)B(x) - C(x)$, $P(x) = h(x)t(x)$ 。

そのため、 $A(x), B(x), C(x)$ に $t(x)$ の倍数を足しても、 $P(x)$ は $t(x)$ で依然割れる

そこで、証明者は新たにランダムな $\delta_a, \delta_b, \delta_c$ を用意し、

$$A'_{mid}(x) = A_{mid}(x) + \delta_a t(x)$$

$$B'_{mid}(x) = B_{mid}(x) + \delta_b t(x)$$

$$C'_{mid}(x) = C_{mid}(x) + \delta_c t(x)$$

を $(g^{A_{mid}(s)}, g^{B_{mid}(s)}, g^{C_{mid}(s)})$ の代わりに元に埋めて検証者に送る

これにより、検証者に渡す情報と w_{mid} は完全に相関しなくなる！！！

もちろん、検証者のペアリングを用いたQAPチェックは通ったまま！！

ただ、 α -Checkや β -Check用の元は、

現状のTTPが公開するパラメータから計算することはできない

ので、TTPのパラメータも少し変わるが割愛

終わりに

終わりに

- 今回解説したのは、**Pinocchio-Groth16のパラダイムだけ！！**
 - R1CSベースのまま、トラステッドセットアップを不要にしたヤツ
 - R1CSすら使わない、別の述語に対するZK-SNARK
 - 再帰的にZK-SNARKできるやつ
 - ...etc
- **ZK-SNARKの応用もたくさんあるよ！！**
 - ZK-VM、ZK-ML、匿名投票、
- あと今回のプロトコルは軽く簡略化しているので、気持ち健全性欠けてます
 - 基底ずらす部分とか...

参考文献

1. Parno, Howell, Gentry, Raykova,
“Pinocchio: Nearly Practical Verifiable Computation”, ePrint 2013/279.
<https://eprint.iacr.org/2013/279.pdf>
2. minaminao, “Tornado Cats”
<https://minaminao.github.io/tornado-cats>
3. nuno (Zenn), “Pinocchioの原理”
<https://zenn.dev/qope/articles/f94b37ff2d9541>
4. Leona Hioki (Medium) , “楕円曲線の夢の国に住もう！！”
<https://leonahioki.medium.com/椭圆曲线的梦之国-12dcc675995a>