(資料公開版)

日本スマートフォンセキュリティ協会(JSSEC)
フィンテックセミナー

ビットコインの仕組みとセキュリティ技術

2017年7月10日(月) 技術セミナー1/13:00-13:50
於：大手町 三井住友銀行東館ライジングスクエア

富士ゼロックス株式会社　漆嶌 賢二

FUJI Xerox

自己紹介: 漆嶌 賢二(うるしま), CISSP

- 経歴
  - 富士ゼロックス(2010~), エントラスト(2005~), セコム(1988~)
- 興味：

  PKI, TLS暗号通信, 電子署名, SSO, 認証, 暗号,
  CSIRT, 商品系CSIRT(PSIRT), インシデント対応,
  脆弱性検査, フォレンジック, ビットコイン
- 委員、標準化、普及啓蒙
  - JNSA, CRYPTREC, 日本データ通信協会, ECOM, PKI-J, 欧ETSI 委員
    IPA セキュキャン講師
- 近年の主な講演/講師歴
  - 2014/06 JNSAビットコイン勉強会 ビットコインの技術を理解する
  - 2014/11 Internet Week SSL/TLS設定のツボ
  - 2015/04 JNSA PKI Day SSL生誕20年
  - 2015/08 IPAセキュリティキャンプ ビットコインの技術解説
  - 2015/11 Internet Week SSL/TLS最新動向
  - 2017/08 IPAセキュリティキャンプ 暗号運用技術

FUJI Xerox

今日のテーマ

暗号通貨と呼ばれるビットコインの、
使われている暗号技術、データ構造を解説し
その仕組みを知っていただく

FUJI Xerox

2014年6月の勉強会での講演内容がベース

データ構造、仕組みについて踏み込んだ、おそらく日本初の資料

リンク： https://www.slideshare.net/kenjiurushima/20140602-bitcoin1-201406031222

FUJI xerox

本日のアジェンダ

- ビットコインとは？
- ビットコインで使われる暗号
- ビットコインの技術的な説明

FUJI Xerox

ビットコインってなに？

FUJI Xerox

ビットコインとは？

- 政府や企業が一極発行したものではない
- インターネットで十分な市場を確立した
- 暗号技術を使った
- これまでにないアイディアの仮想通貨

FUJI Xerox

ビットコインの時価総額(2017年7月現在 約1千億米ドル)

約1000億米ドル
≒11兆円
≒東京都の予算13兆円
≒Facebook社の時価総額
≒Apple社の時価総額の1/5

出典：https://coinmarketcap.com/charts/

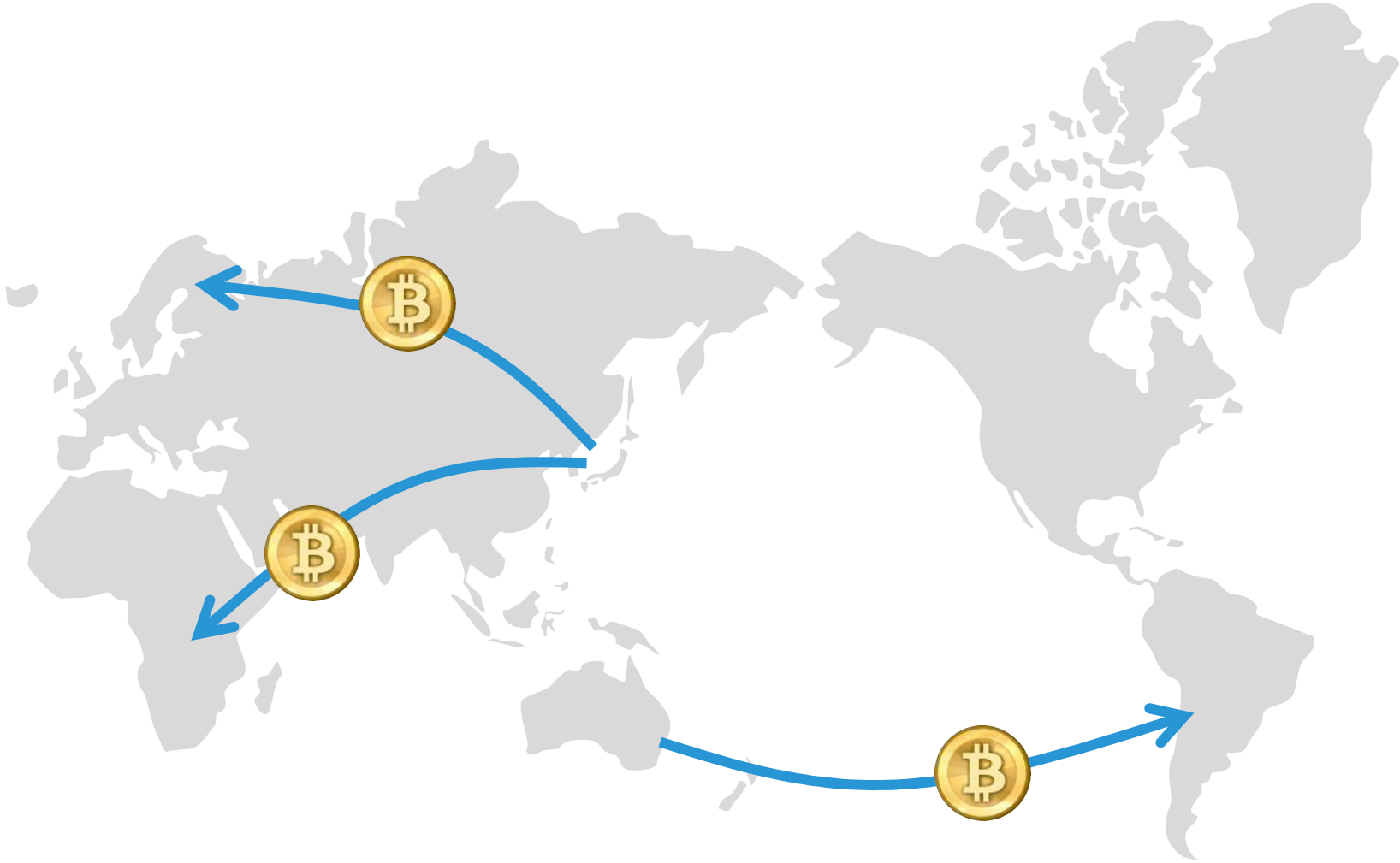
FUJI Xerox

(参考) ビットコイン(BTC)/日本円 レート

出典：http://ビットコイン相場.com/

FUJI Xerox

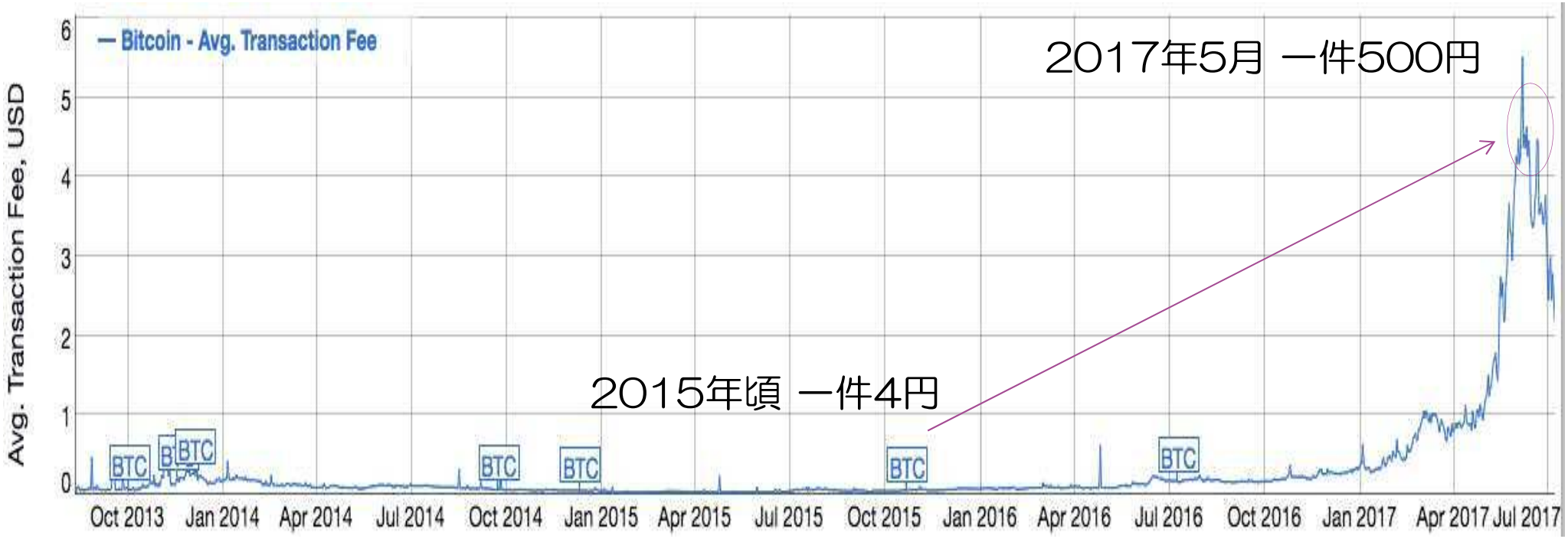
送金先文字列さえ知ってれば
世界中の誰にでも送れ、手数料は安かった
(今はそれなり)



FUJI Xerox

平均送金手数料の高騰(5円→最高500円)

2017年5月頃、少額決済が大量に流れ込み取引が大量停滞し、手数料が高騰
今は、やや沈静化したか？

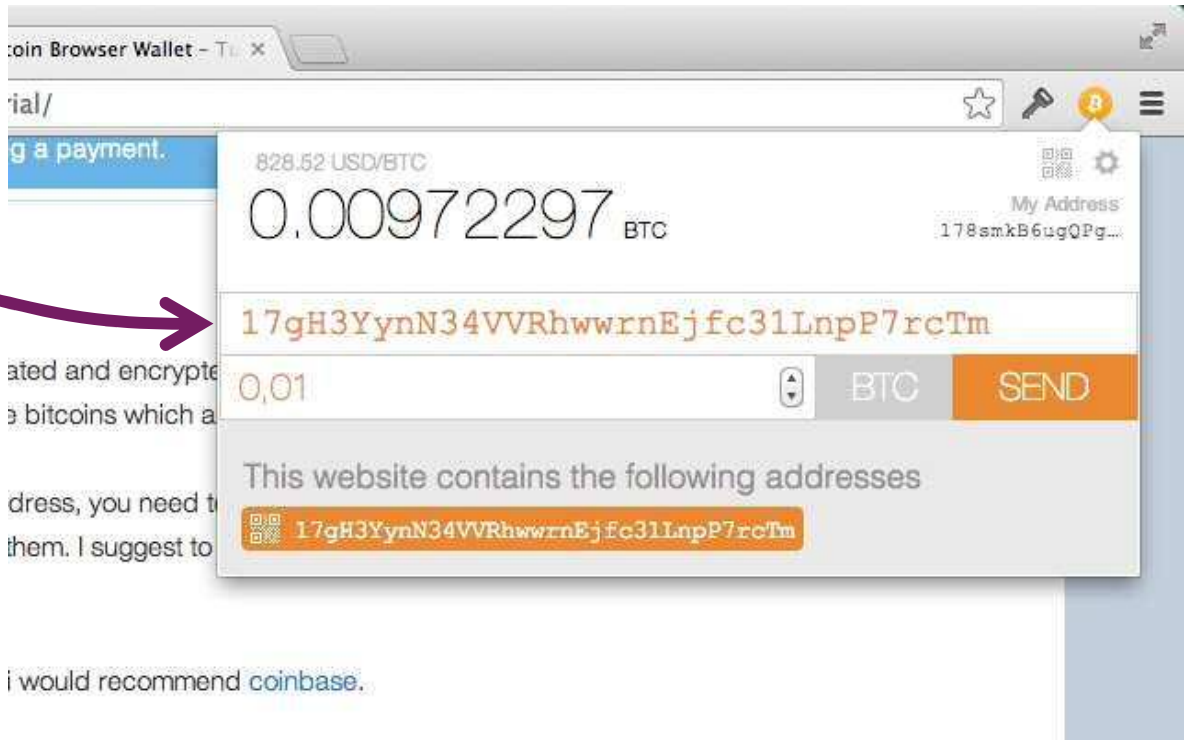


出典：https://bitinfocharts.com/comparison/bitcoin-transactionfees.html

FUJI Xerox

ビットコインはとても使い方が簡単

バーコードで読んだ
1から始まる34文字の英数字と送金額を入れれば、世界中誰にでもお金が払える

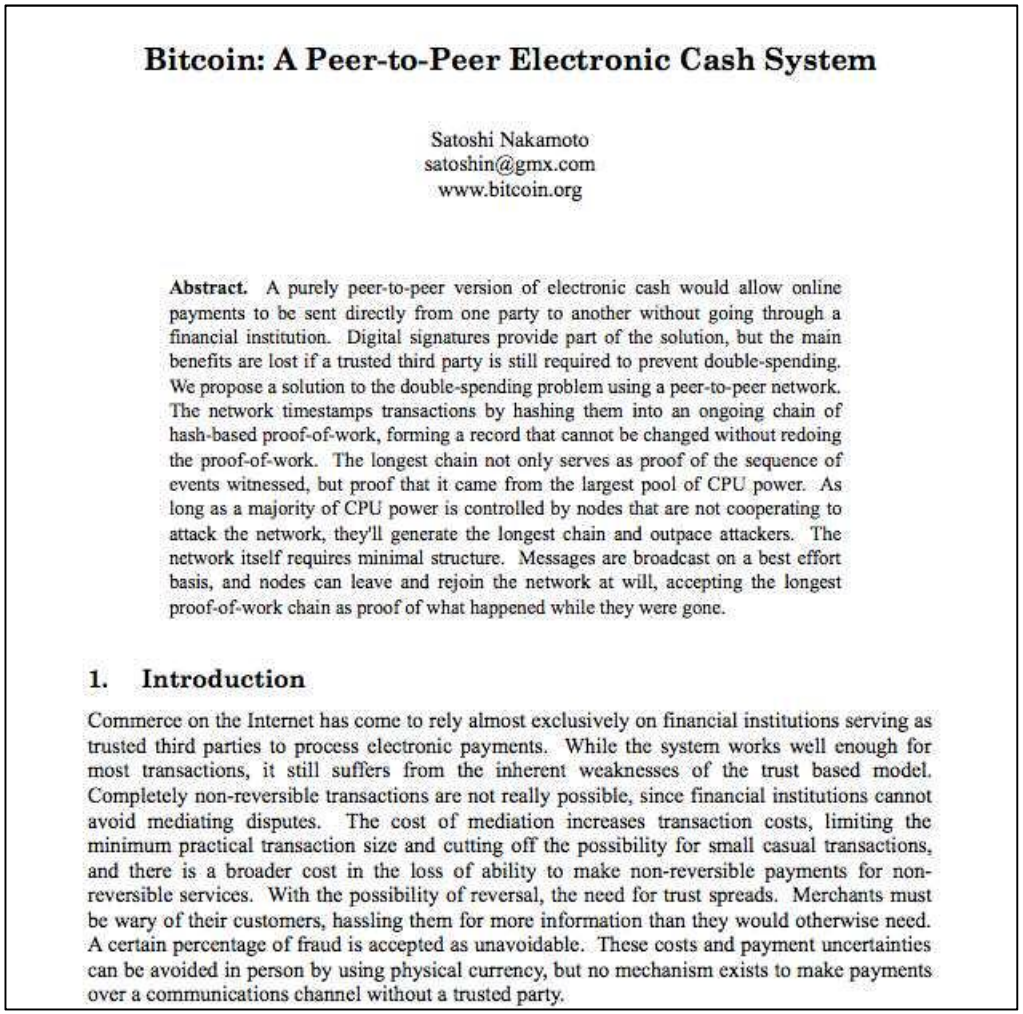


スマホからだって
送金や受取可能

FUJI Xerox

ビットコインは2008年の謎の論文からはじまった

- 中本哲史(ナカモトサトシ)という謎の人物が2008年にMetzger, Dowdeswell & Co (metzdowd.com) の暗号理論のメーリングリストに投稿した論文が始まり
- 2009年にはオープンソース実装 Bitcoin-Qtが開発され、最初のビットコイン通貨が発行された



Bitcoin: A Peer-to-Peer Electronic Cash System

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org

Abstract. A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution. Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network. The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

1. Introduction

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party.

出典：https://bitcoin.org/bitcoin.pdf

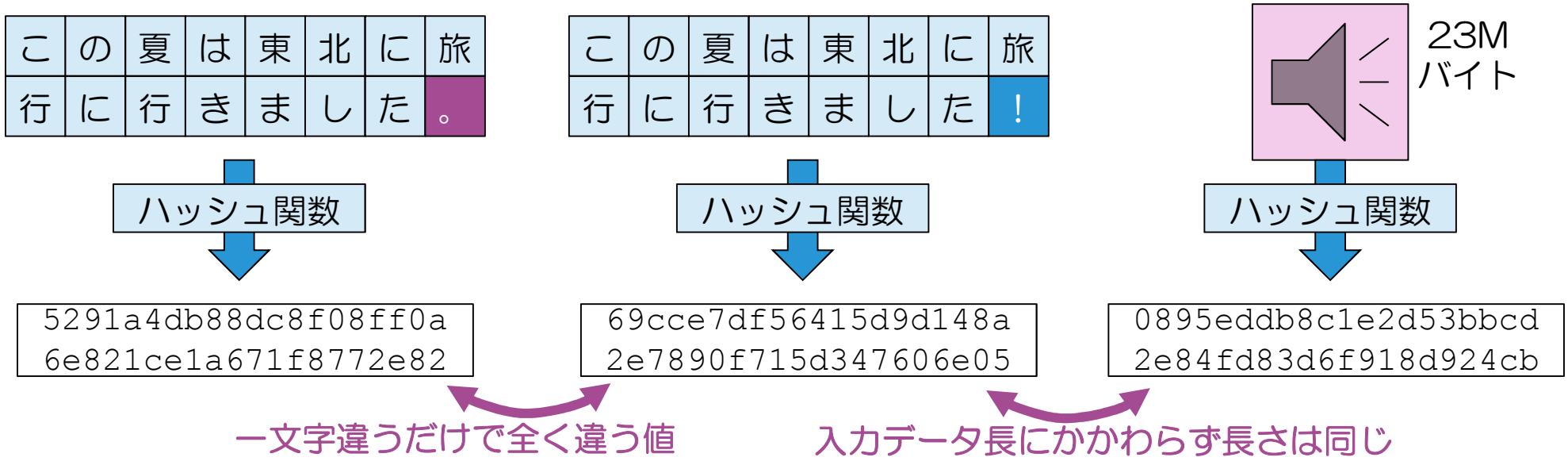
FUJI Xerox

ビットコイン技術を理解するのに必要な暗号技術(1/2)
ハッシュ関数

FUJI Xerox

(セキュア)ハッシュ関数

データの内容を識別するための固定の長さのデータを作る関数。
入力値の僅かな違いでも値は大きく異なり、大きなデータでも同じ固定の長さ



- ✓ 入力データに変更や改竄がないか簡単に確認できる
- ✓ 電子署名や暗号通信などにも用いられる
- ✓ ビットコインではデータのIDなどいろんな用途に使われる
- ✓ SHA-1, MD5などが有名

FUJI Xerox

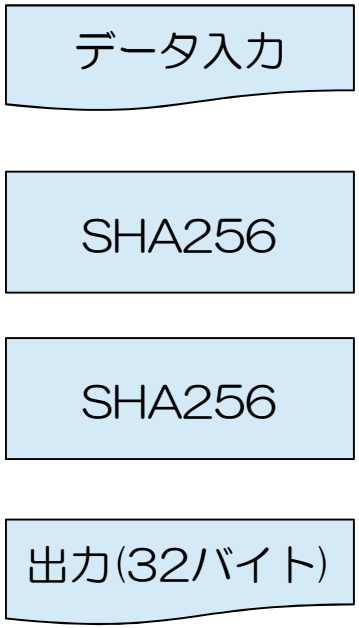
ビットコインで使われるハッシュ関数

ビットコインでは2種類のハッシュ関数しか使われない

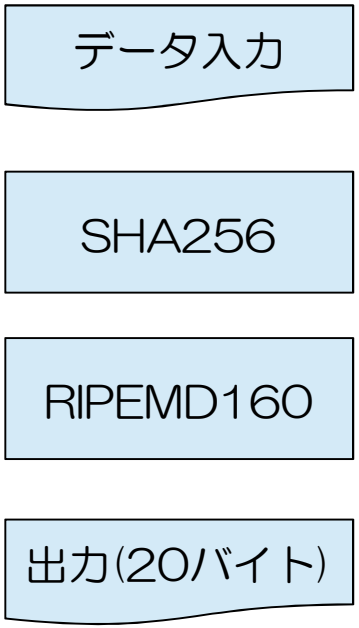
ハッシュ関数	出力長	説明
SHA-256	32バイト	米NSAが2001年に開発した米国家標準関数
RIPEMD-160	20バイト	独、ベルギーで1996年開発、ISO/IEC10118-3

ビットコインでは必ずハッシュ関数を二重にかける2通りの方法でしか使わない

便宜上
本講演では
sha256d
と呼ぶ
(doubleのd)

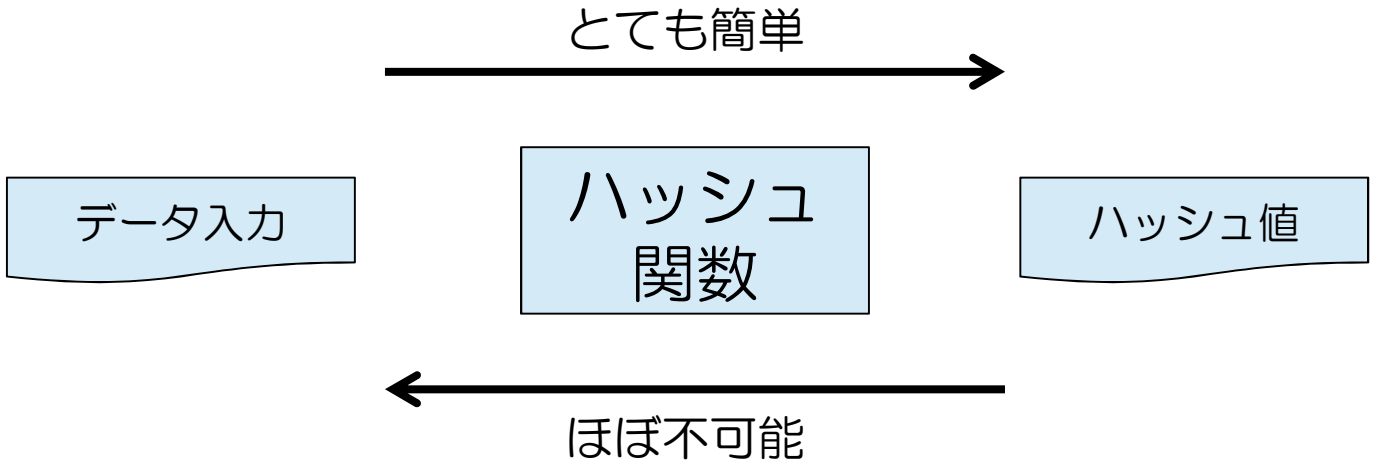


便宜上
本講演では
hash160
と呼ぶ

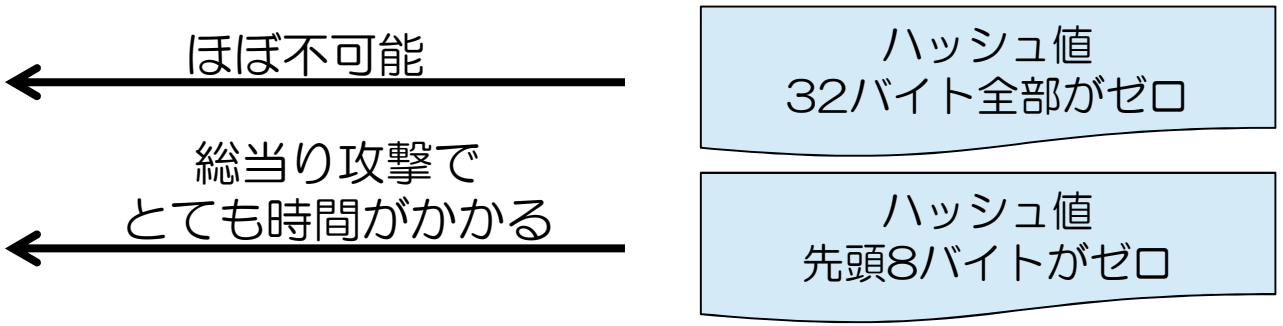


FUJI Xerox

ハッシュ関数の大切な特徴(一方向性関数)

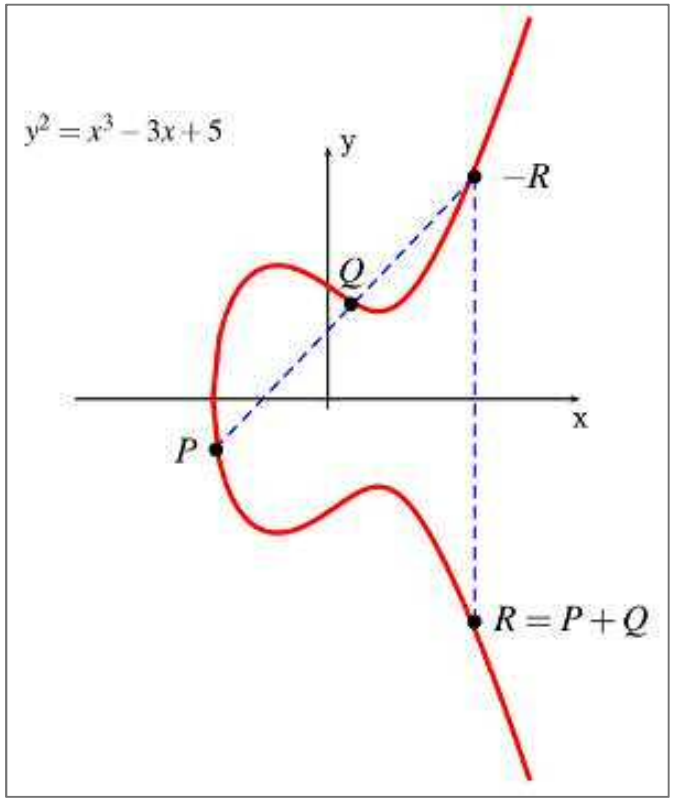


例えば？



FUJI Xerox

ビットコイン技術を理解するのに必要な暗号技術(2/2)
楕円曲線公開鍵暗号

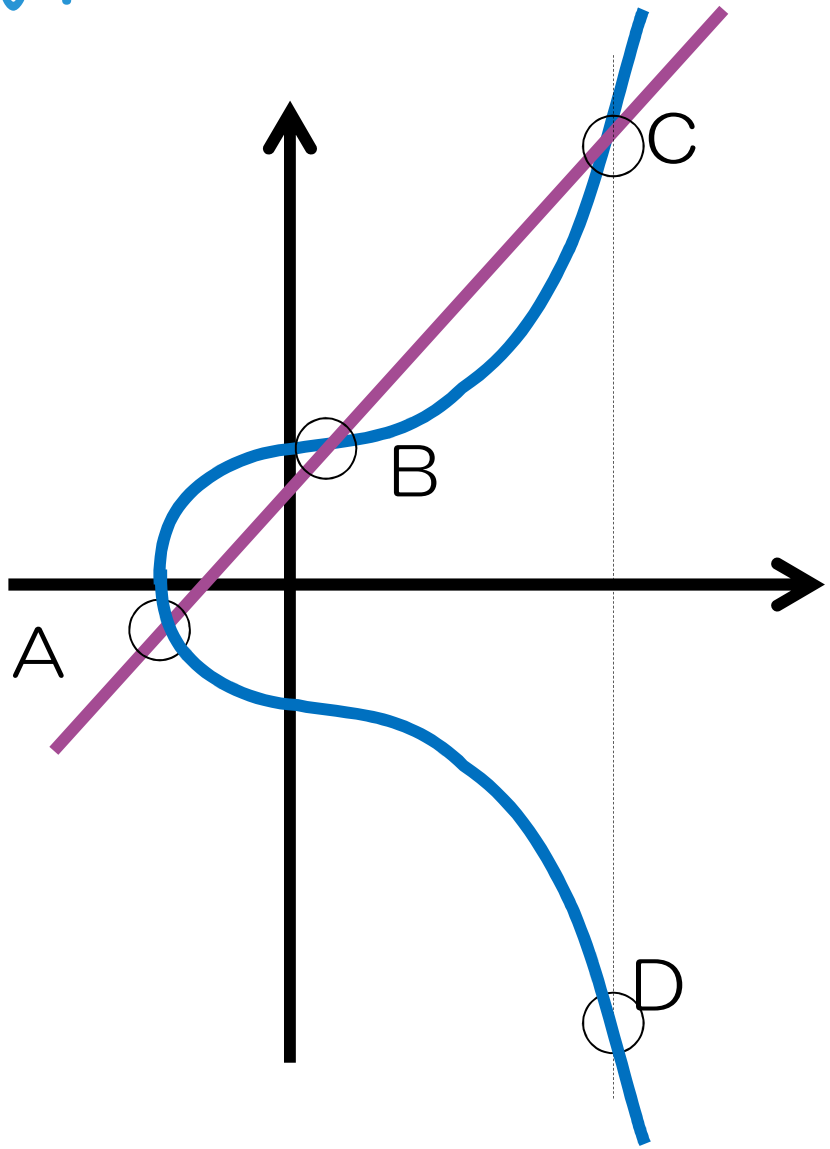


FUJI Xerox

楕円曲線公開鍵暗号(1)
楕円曲線は楕円ではない

$$y^2 = x^3 + ax + b$$

を満たす曲線を
「楕円曲線」という
注：楕円じゃない



(性質)

直線を引くと多くの場合
3つの点A, B, Cで交わる

X軸で線対称

FUJI Xerox

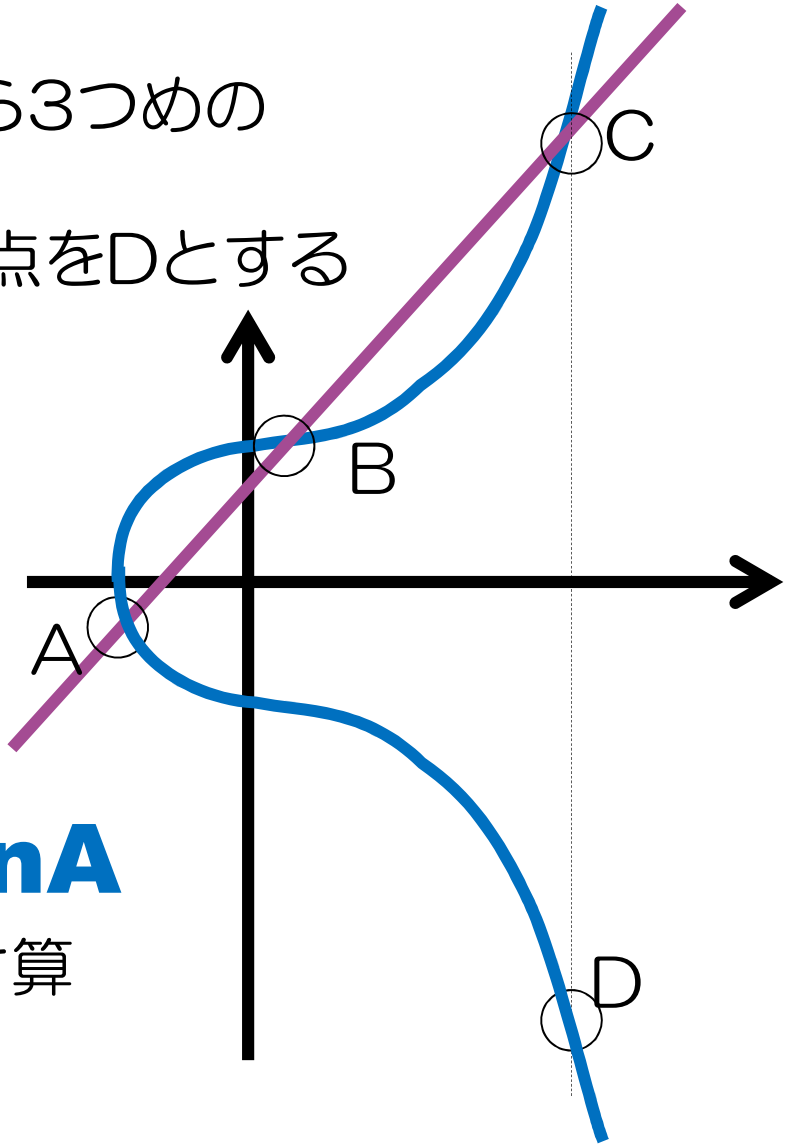
楕円曲線公開鍵暗号(2)
点の足し算=2つの点から３つ目を見つけること

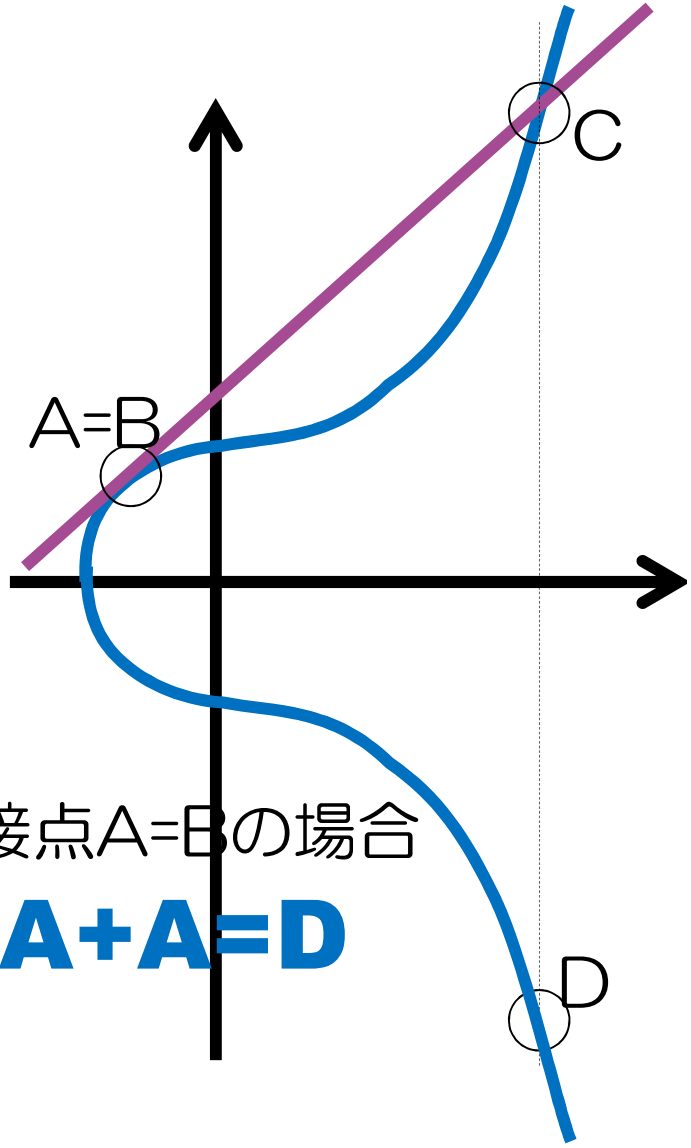
2つの点A,Bから3つめの
交点Cを見つけ
そのX軸対称の点をDとする

これを
A+B=D
と足し算で表す

すると
A+A+...=nA

整数と点の掛け算
を定義できる



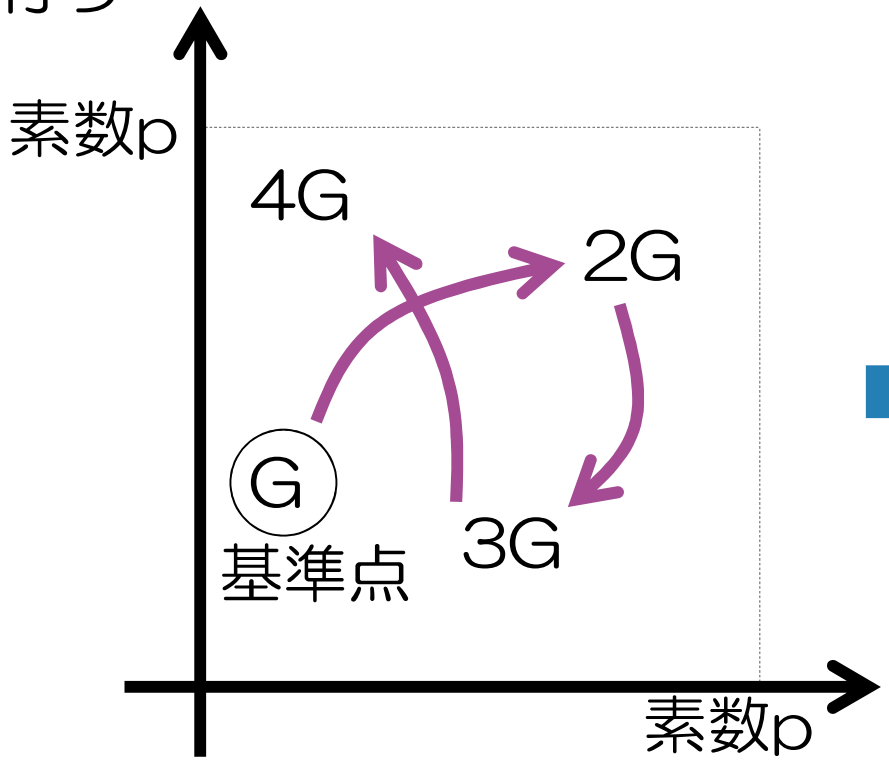


接点A=Bの場合
A+A=D

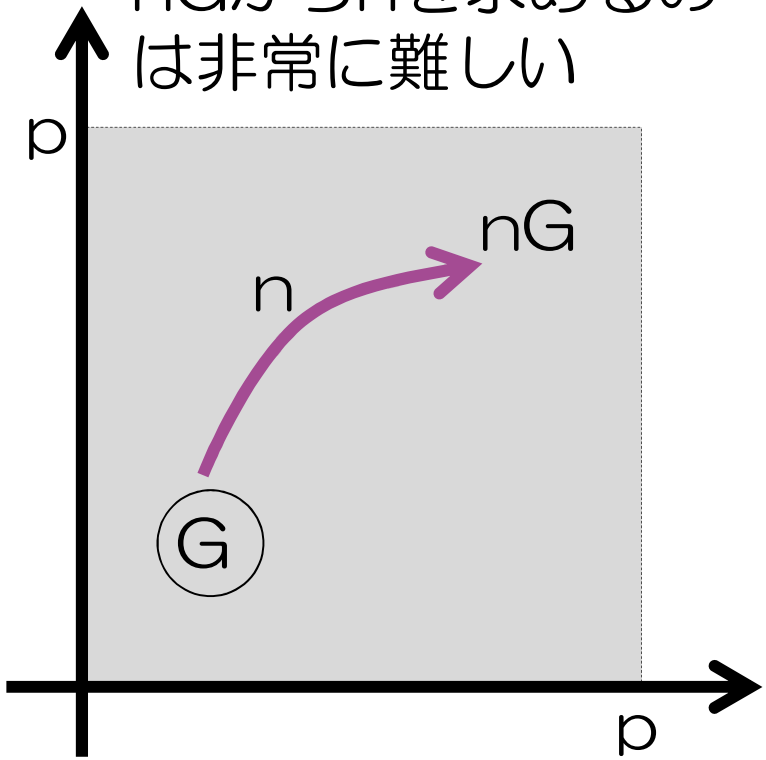
FUJI Xerox

楕円曲線公開鍵暗号(3)
実数ではなく整数Iを素数pで割った余りでやる

曲線上の点のx、y座標が「整数Iを素数pで割った余り」で行う

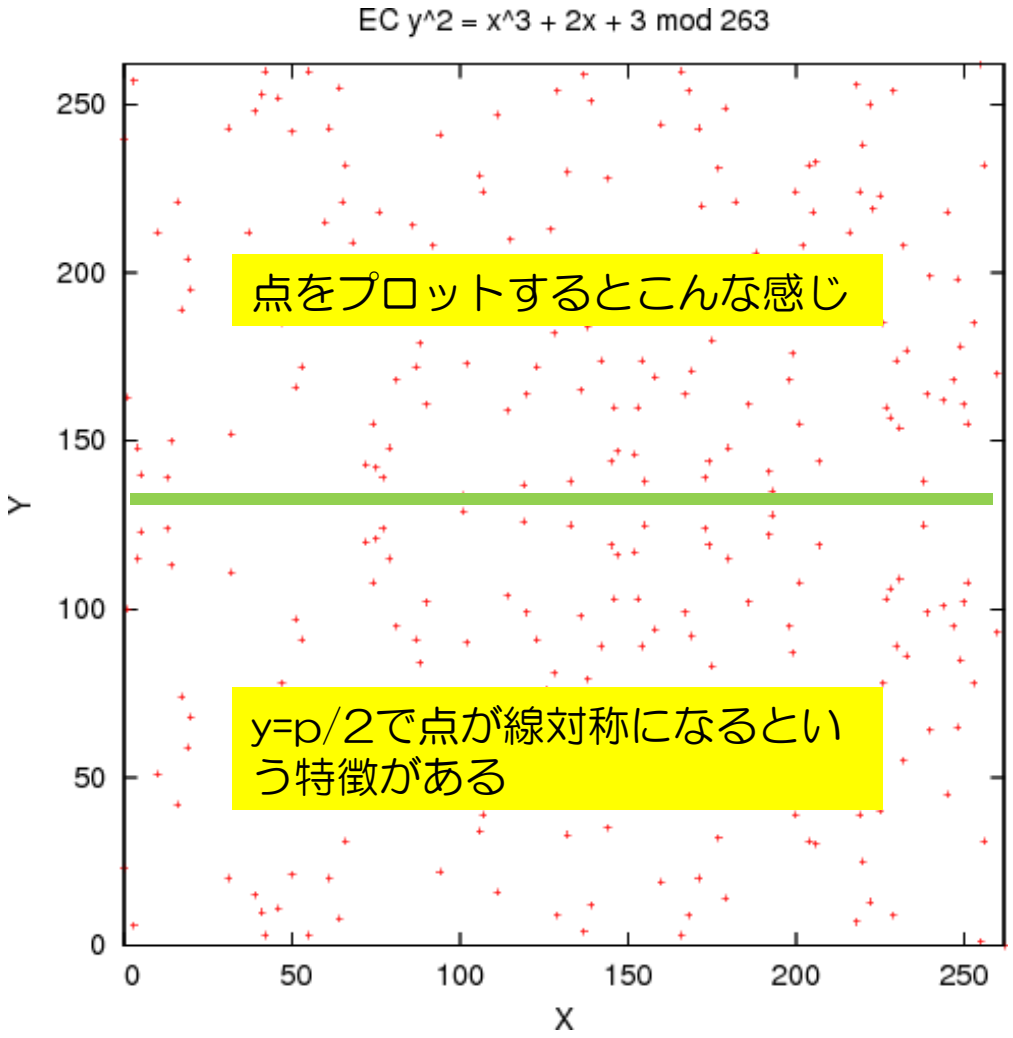


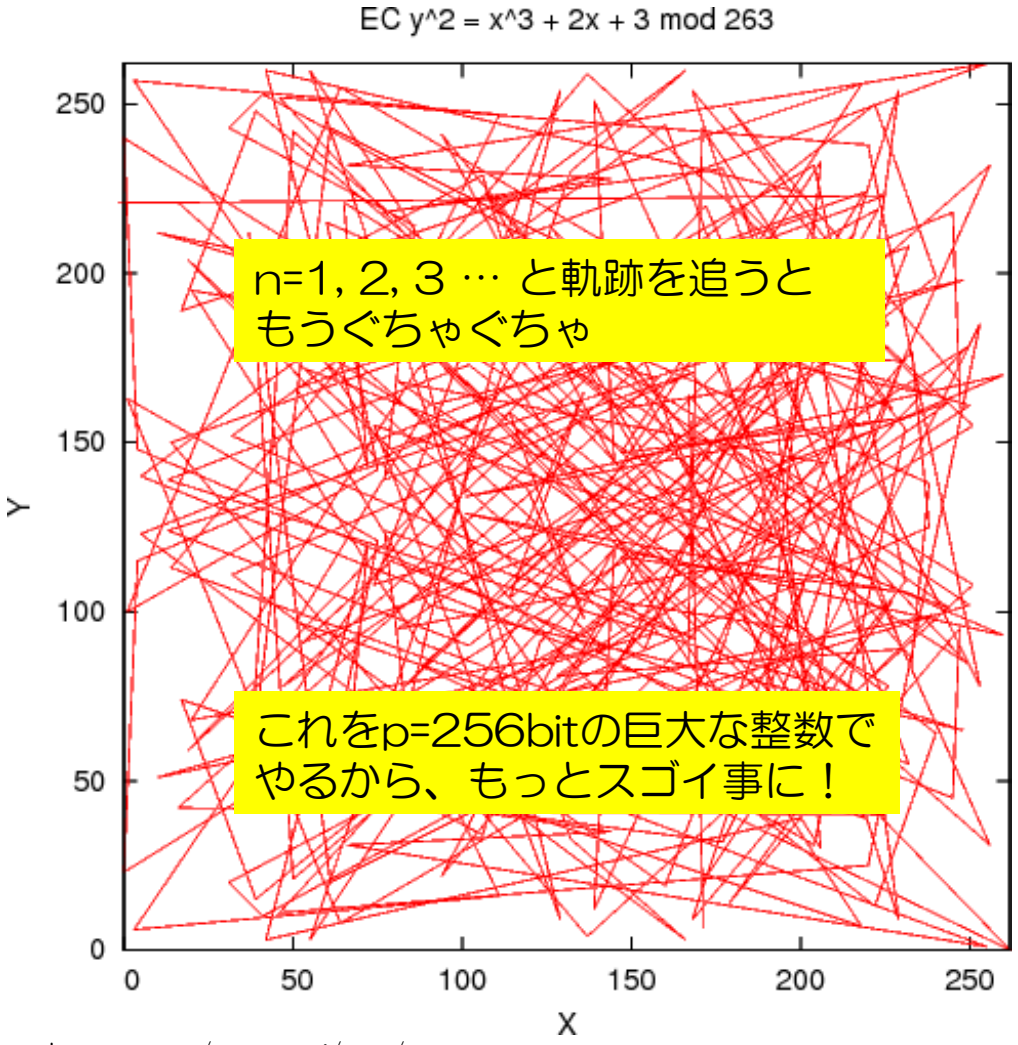
ある点nGが与えられた時、nが大きければnGからnを求めるのは非常に難しい



FUJI Xerox

(参考) 例えば素数p=263の場合





出典： http://www.johannes-bauer.com/compsci/ecc/

FUJI Xerox

楕円曲線公開鍵暗号(4)
楕円曲線を使った公開鍵暗号

パラメータは任意で良いのではなく、いくつかの団体で安全なパラメータが数十程度定義されている

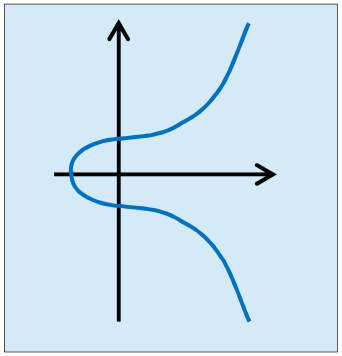
有名な所では
NIST P-256, 384, 521
secp256p1, 521p1等

曲線パラメータ
基準点G(x,y)
剰余の素数 p
曲線定数 a
曲線定数 b

ビットコインでは
secp256k1固定で使う

gx=x79be667ef9dc…
gy=x483ada7726a3…
p=xffffffff…
a=0
b=7

秘密鍵
n



公開鍵
nG

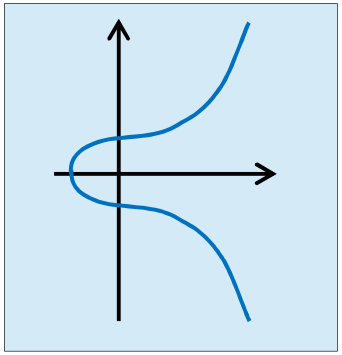
公開鍵nGとパラメータG, p, a, bから秘密鍵nを見つけるのは困難
→ 公開鍵暗号として使える！！

(特徴)データが小さい(256bなら公開65, 秘密32, パラ数byte以下

FUJI Xerox

楕円曲線公開鍵暗号(5)
公開鍵データの表現方法(非圧縮/圧縮)

秘密鍵
n



公開鍵
nG

公開鍵は二次元座標の点

表現方法は2種類あ

非圧縮フォーマット

注)例は256bit鍵の場合

04	公開鍵nGのX座標(32バイト)	公開鍵nGのY座標(32バイト)

圧縮フォーマット

02	公開鍵nGのX座標(32バイト)	Yが偶数の時
03	公開鍵nGのX座標(32バイト)	Yが奇数の時

XからYはすぐ計算可能：$y^2=x^3+ax+b$

FUJI Xerox

楕円曲線公開鍵暗号(6)
ECDSA署名の生成と検証

生成 秘密鍵d、パラメータ(生成点G, 剰余の素数n)があるとし、署名の度に乱数　k (0≦k≦n)を生成し、Gからk回移動した点Rを定義し(R=kG=(r_x, r_y))、署名値 r, sを求める。

メッセージハッシュe

$$e = HASH(m)$$

Gから乱数k回移動した点R

$$R = kG = \begin{pmatrix} r_x \\ r_y \end{pmatrix}$$

$$r = r_x \pmod n$$

署名値 r, s

$$s = \frac{e + dr}{k} \pmod n$$

検証 公開鍵Q=dG、パラメータ(生成点G, 剰余の素数n)、署名値(r,s)があるとし、以下の点Pを計算する。

等しければ検証成功

$$P = \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \frac{e}{s}G + \frac{r}{s}Q$$

このとき、$p_x = r_x \pmod n$ ならば署名値は正しい

FUJI Xerox

楕円曲線公開鍵暗号(7)
なぜECDSA署名の検証はこれでよいのか？

署名値rsとメッセージ
ハッシュeと公開鍵Qが
正しければ、
点Pと点Rが等しいので
$p_x = r_x$ であることを示そう。

点Pの定義より

$$P = \begin{pmatrix} p_x \\ p_y \end{pmatrix} = \frac{e}{s}G + \frac{r}{s}Q$$

公開鍵Q=dGなので

$$= \frac{e}{s}G + \frac{r}{s}dG$$

共通項を括って

$$= \frac{1}{s}(e + rd)G$$

署名sの定義より

$$= \left(\frac{k}{e + rd}\right)(e + rd)G$$

約分してR定義より

$$= kG = R$$

点P=点R が示されたので $p_x = r_x$ となる。

FUJI Xerox

ビットコインの用語の説明(1)

① ウォレットとアドレスと鍵
② アドレス
③ トランザクション

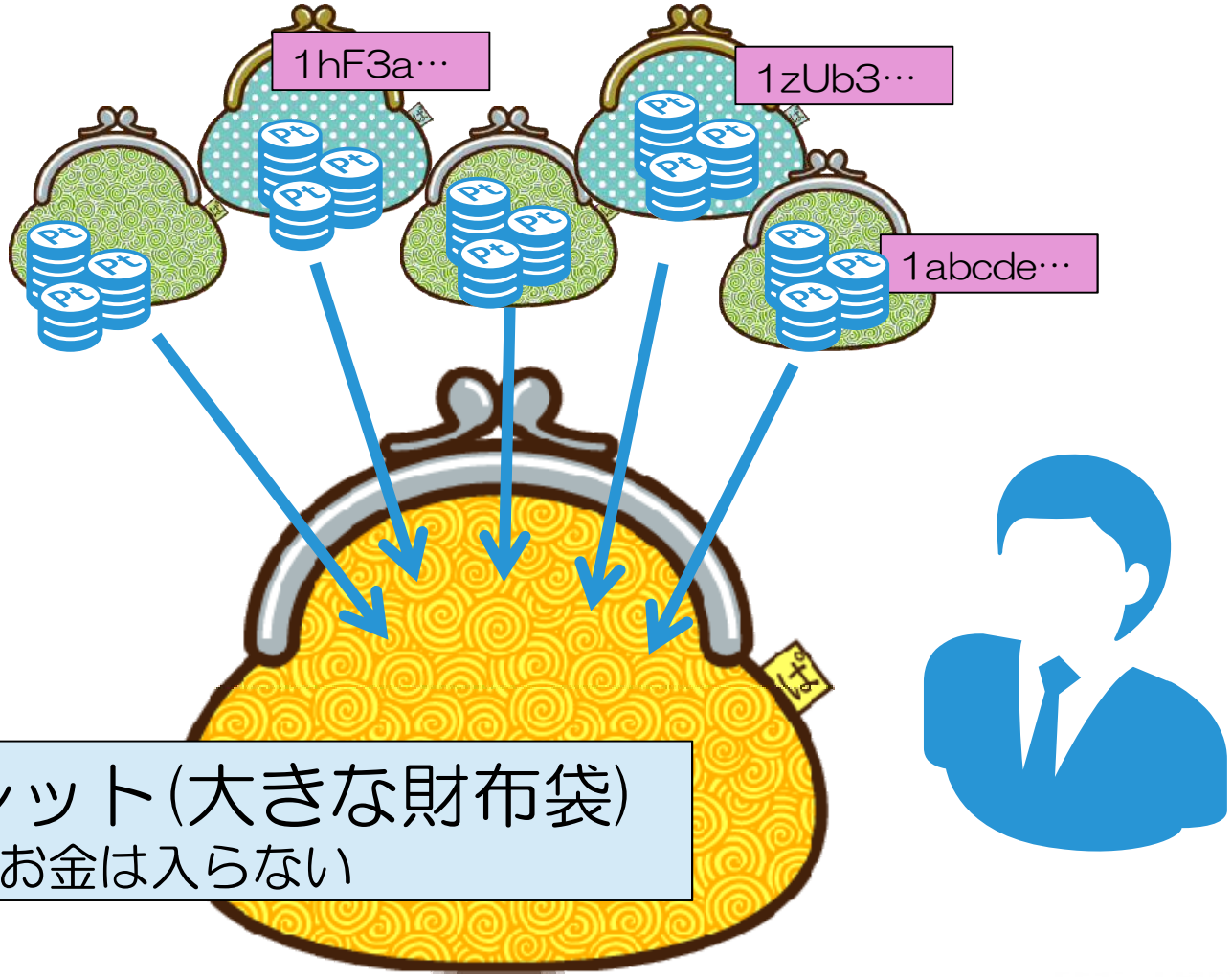
FUJI Xerox

Bitcoinウォレットと
アドレスと
鍵

FUJI Xerox

Bitcoinウォレット(財布)と Bitcoinアドレスのイメージ

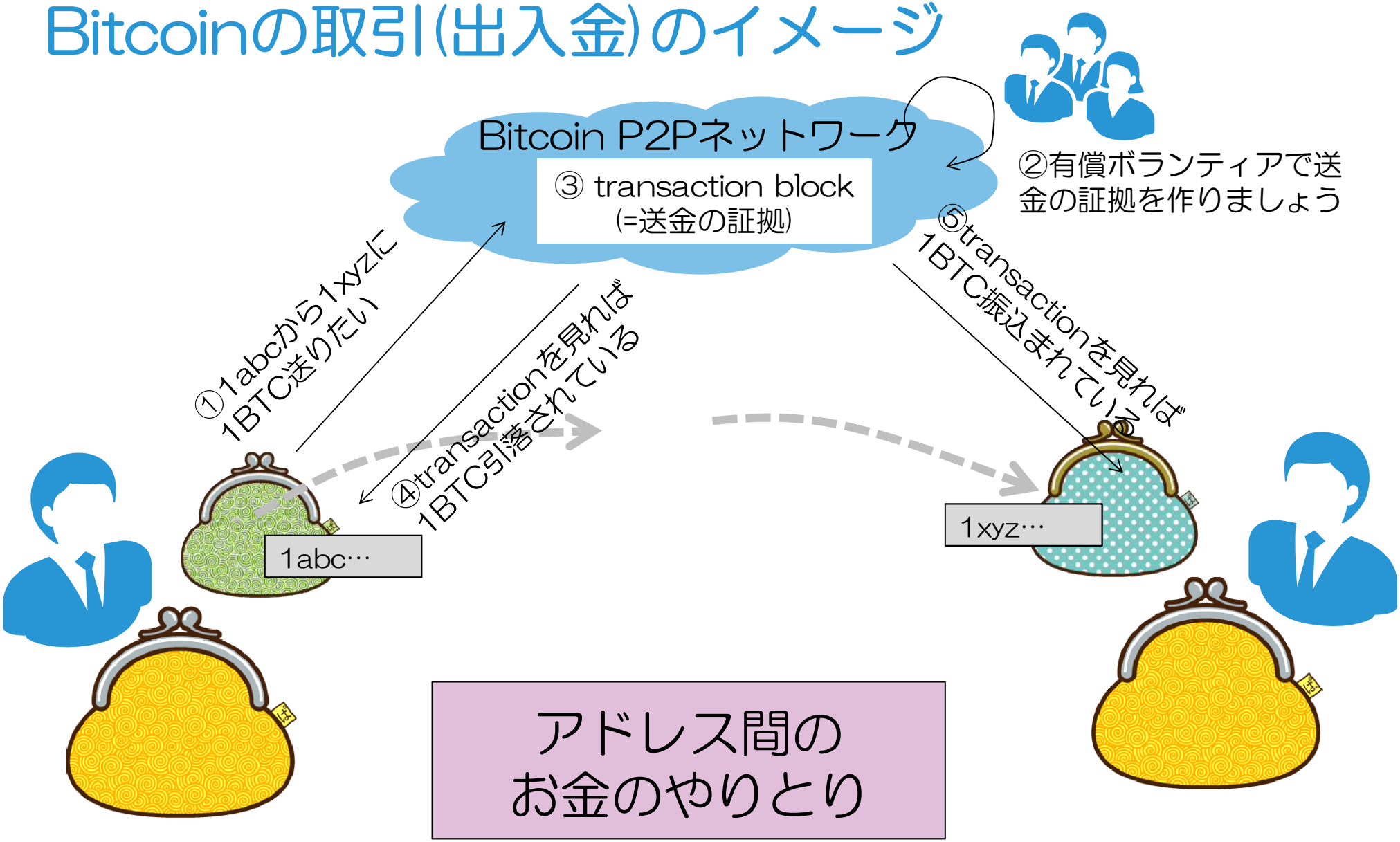
Bitcoinアドレス(小さな財布)
個々にお金が入る・口座番号(アドレス)を持つ



Bitcoinウォレット(大きな財布袋)
直接はお金は入らない

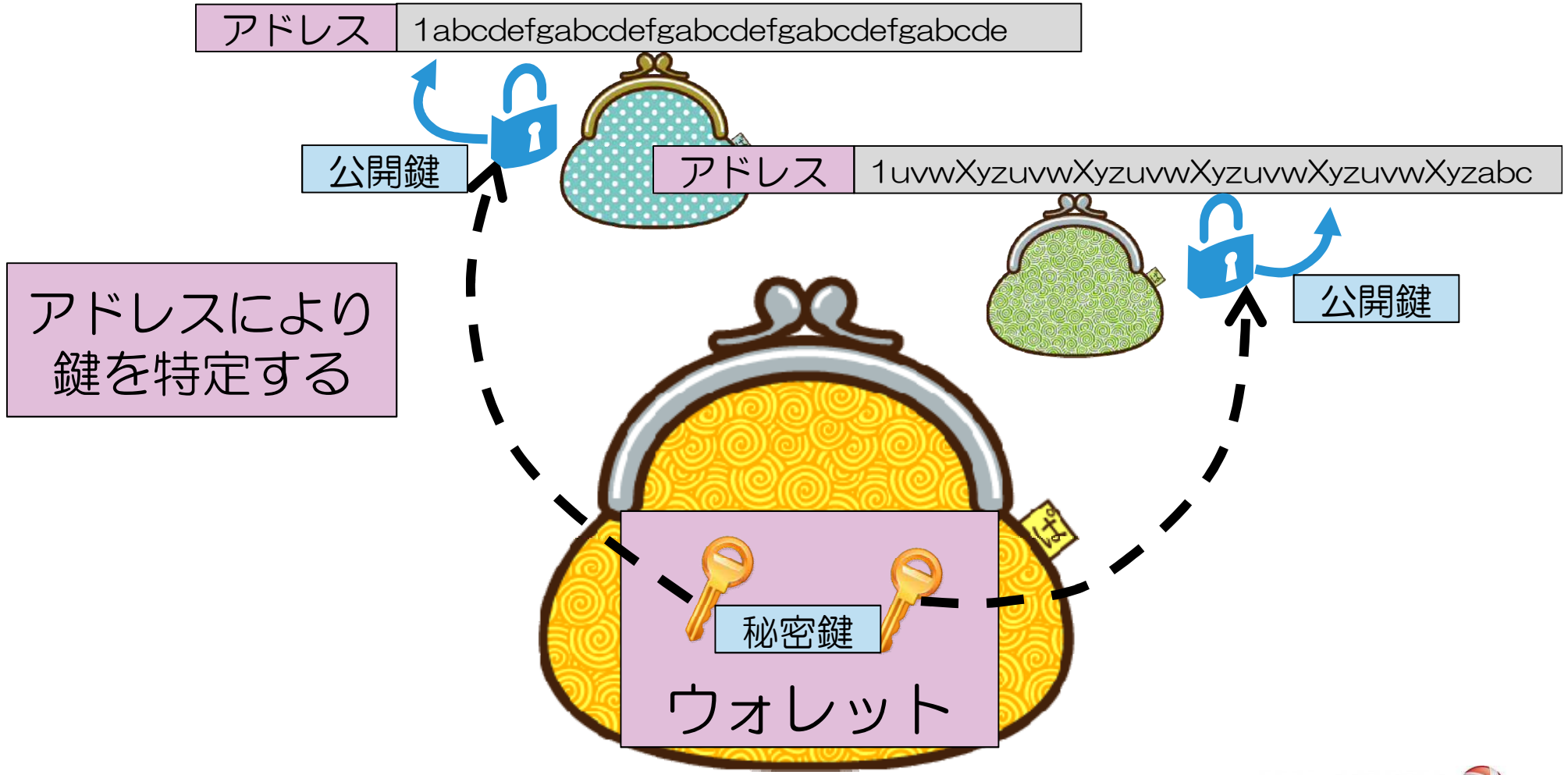
FUJI Xerox

Bitcoinの取引(出入金)のイメージ



FUJI Xerox

Bitcoinアドレスと秘密鍵/公開鍵



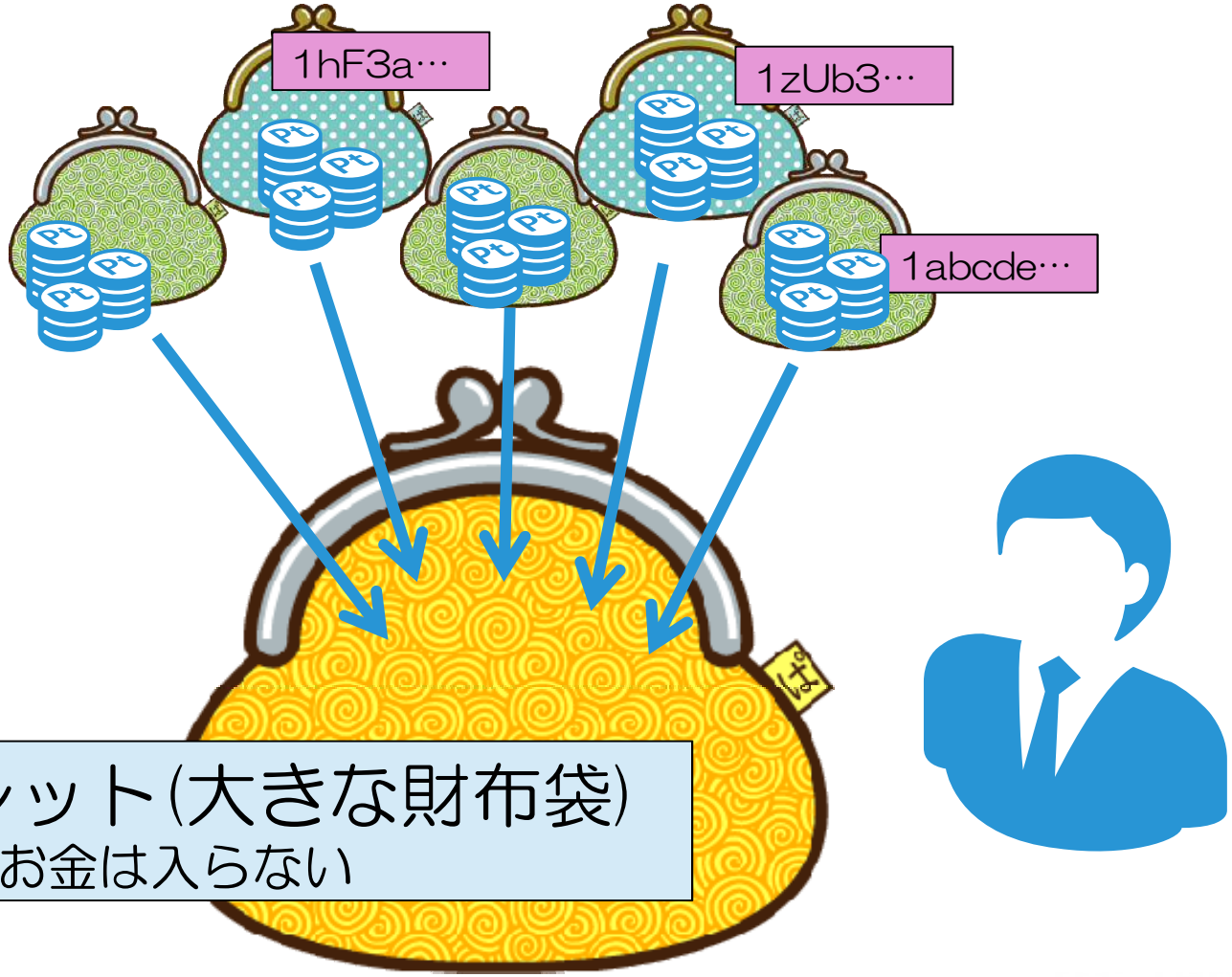
FUJI Xerox

BitCoinアドレス

FUJI Xerox

Bitcoinウォレット(財布)と Bitcoinアドレスのイメージ(再掲)

Bitcoinアドレス(小さな財布)
個々にお金が入る・口座番号(アドレス)を持つ



Bitcoinウォレット(大きな財布袋)
直接はお金は入らない

FUJI Xerox

Bitcoinのアドレスの特徴

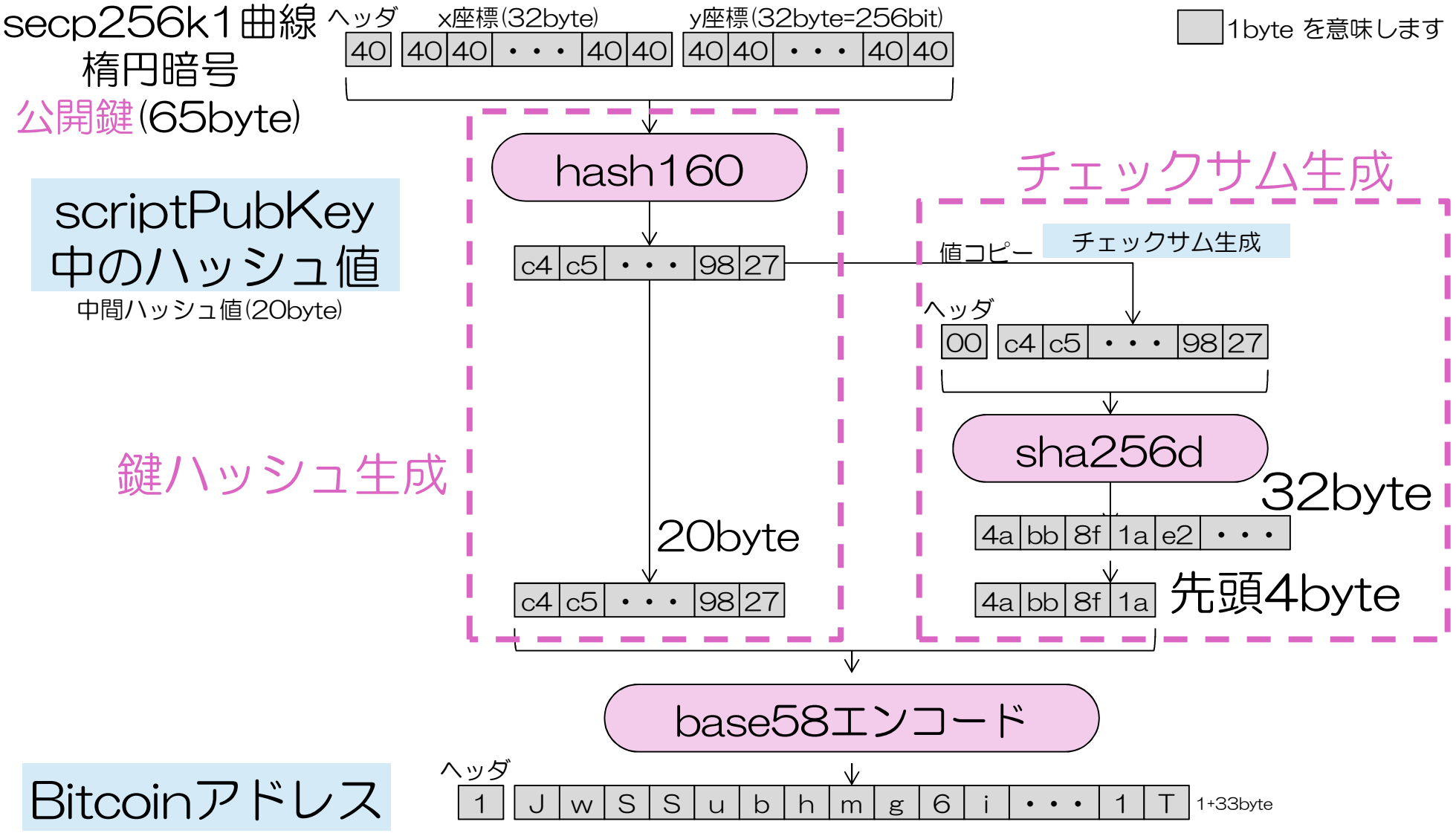
- アドレス毎の公開鍵から生成することができ、公開鍵を特定することができる。すなわち出金元、送金先を特定するID。
- 文字 “1” と33字の英数字でできた文字列(例：1abF3…)
- base58checkエンコーディングを使用している
- base58checkエンコーディングとは公開鍵ハッシュとチェックサムをbase58エンコーディングしたもの

- base58とはバイナリを58種の英数文字で表現したもの
  (数字10種＋大小文字26×2種－紛らわしい文字4種
  エルl,アイI,オーO,ゼロ0を除く=10+26x2-4=58)
  相手口座の入力ミスを防ぐことができる。

- base58はFlickrの画像のIDとしても使われている。

FUJI Xerox

図説：base58check encodingの公開鍵からアドレスの生成



※：ハッシュ関数を使っているので、一方向性の変換
※：ウォレットの秘密鍵エクスポートでも使われる(WIF形式)
※：X.509 公開鍵証明書の鍵識別子(SKID/AKID)の複雑版と言えなくもない

FUJI Xerox

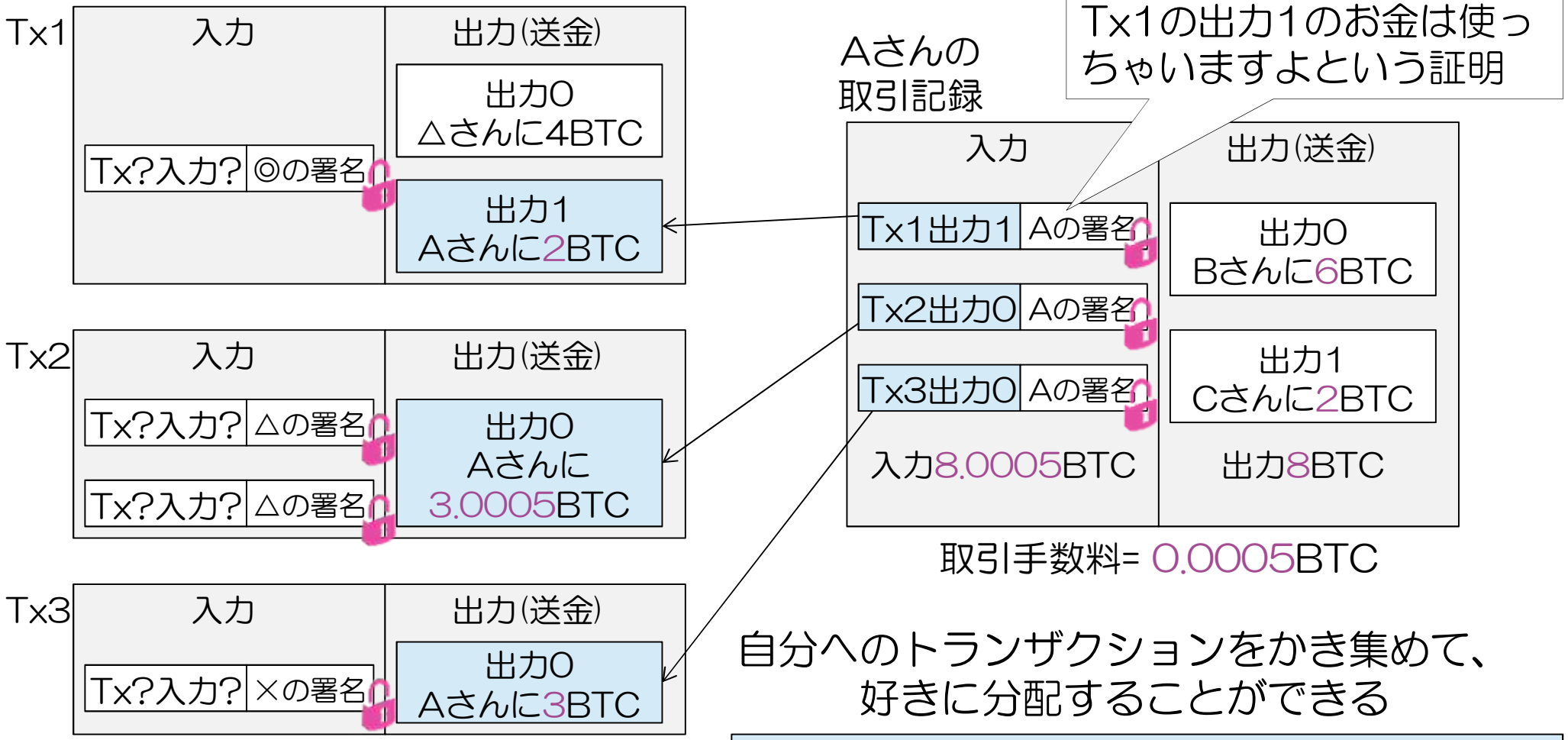
トランザクション(取引履歴)

あるアドレス(小財布)からアドレスへお金が移った
という証拠データ(署名データ)です

FUJI Xerox

トランザクションとは

1つの小財布(アドレス)の小銭をかき集めて
複数の小財布へお金の移動させる電子署名つき証拠データ



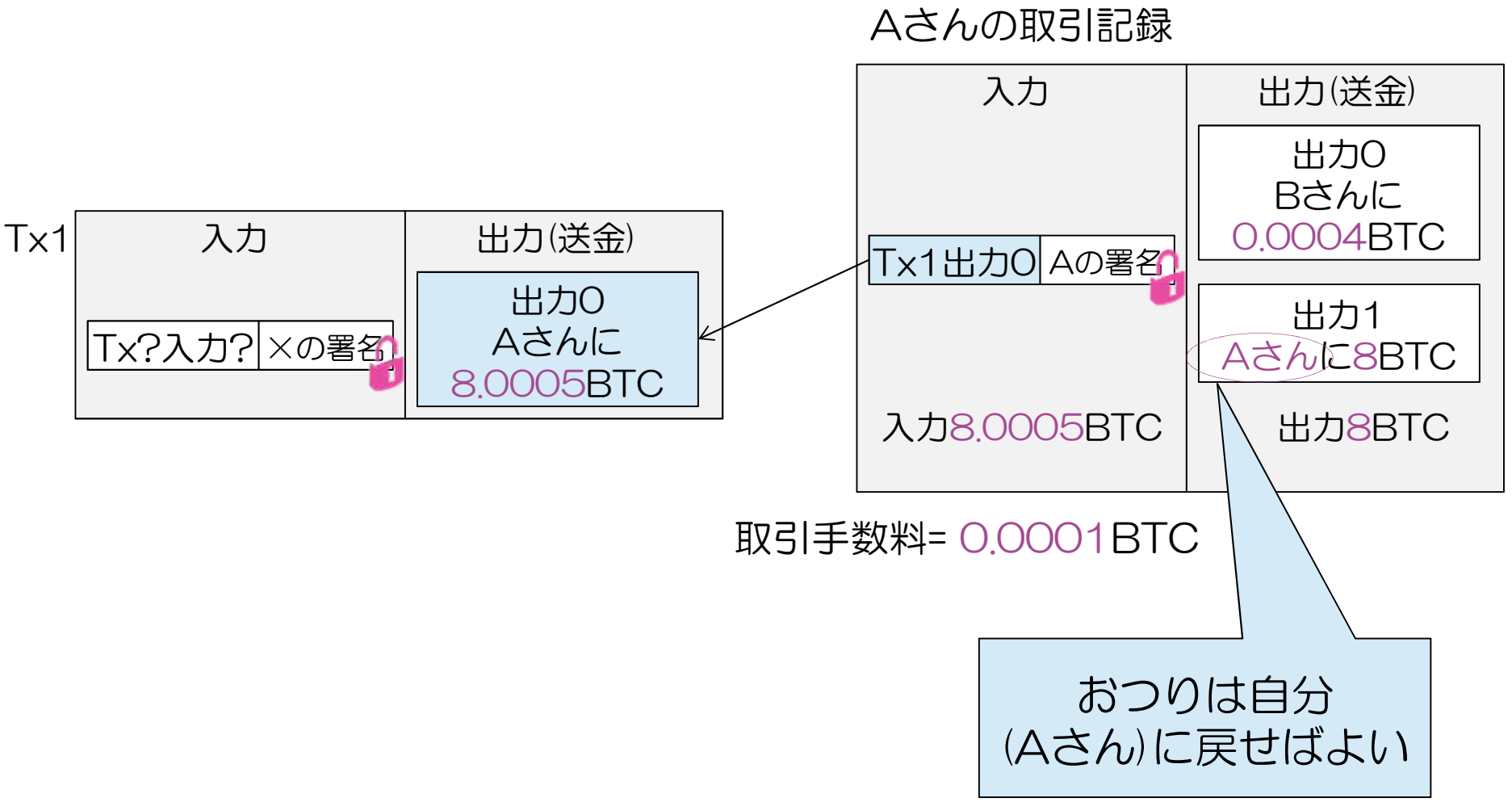
自分へのトランザクションをかき集めて、
好きに分配することができる

入力合算＝出力合算＋取引手数料

FUJI Xerox

大金の一部を使いたいとき

残り(おつり)を自分に送金すればよい



FUJI Xerox

トランザクションデータの構造

トランザクションのデータ構造(1)

トランザクションはバイト列

```
0100000001aa02ce4965…
```

バージョン		01000000
入力数		01
入力①	前の出力ハッシュ	484d40…
	前の出力インデックス	00000000
	スクリプトバイト長	8a
	署名スクリプト	473044…
	シーケンス終端記号	ffffffff
出力数		01
出力①	出金額	626401…
	スクリプトバイト長	8a
	公開鍵検証スクリプト	8a
ブロックロック時間		00000000

scriptSig スクリプト

PUSHDATA_47
ASN.1形式のECDSA署名値(R,S)
SIGHASH_ALL (全てを署名対象に)
PUSHDATA_41
EC公開鍵(x04 + X + Y)

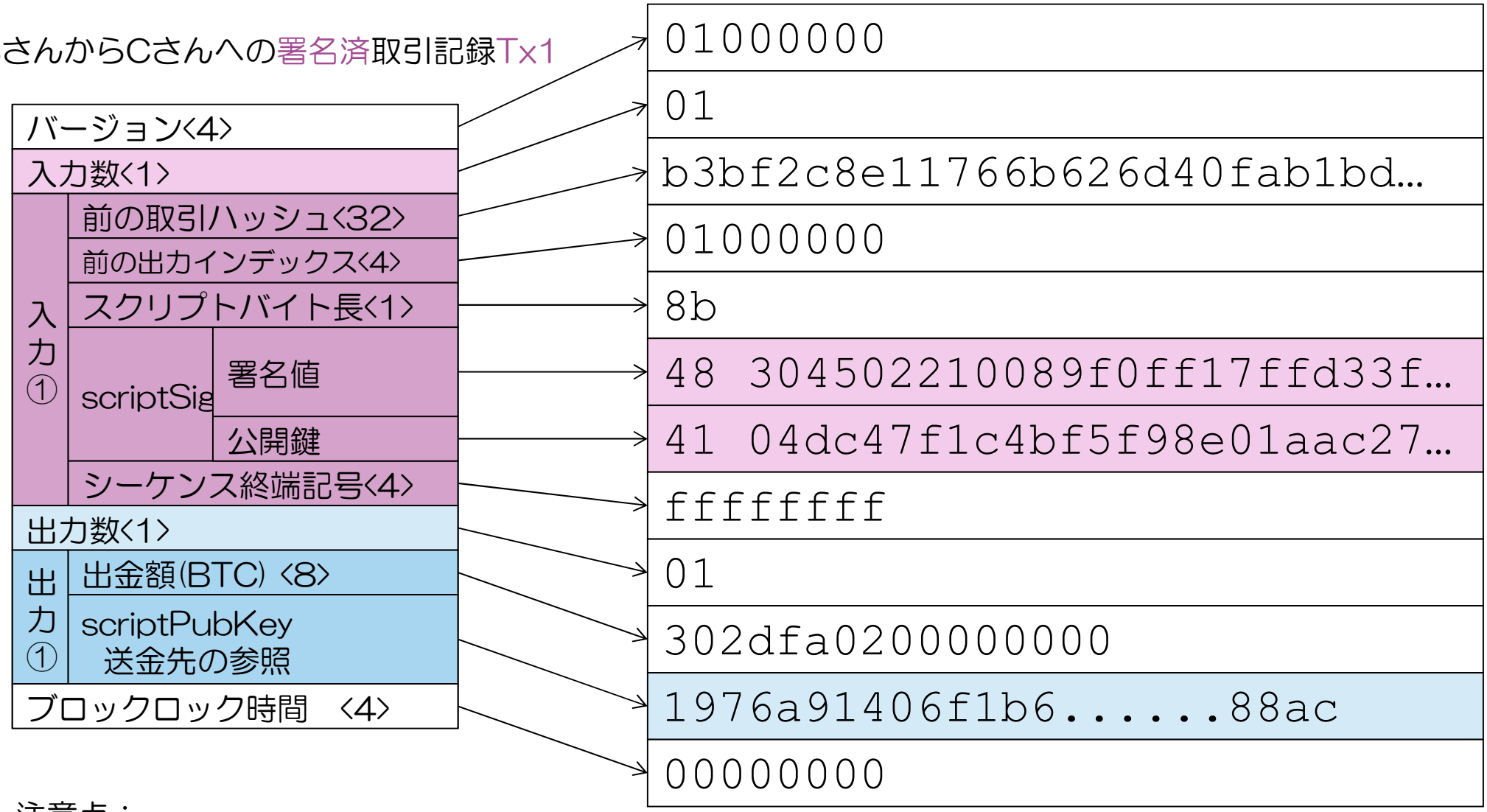
scriptPubKey スクリプト

OP_DUP
OP_HASH160
PUSHDATA 14
公開鍵ハッシュ値
OP_EQUALVERIFY
OP_CHECKSIG

FUJI Xerox

トランザクションと生データ

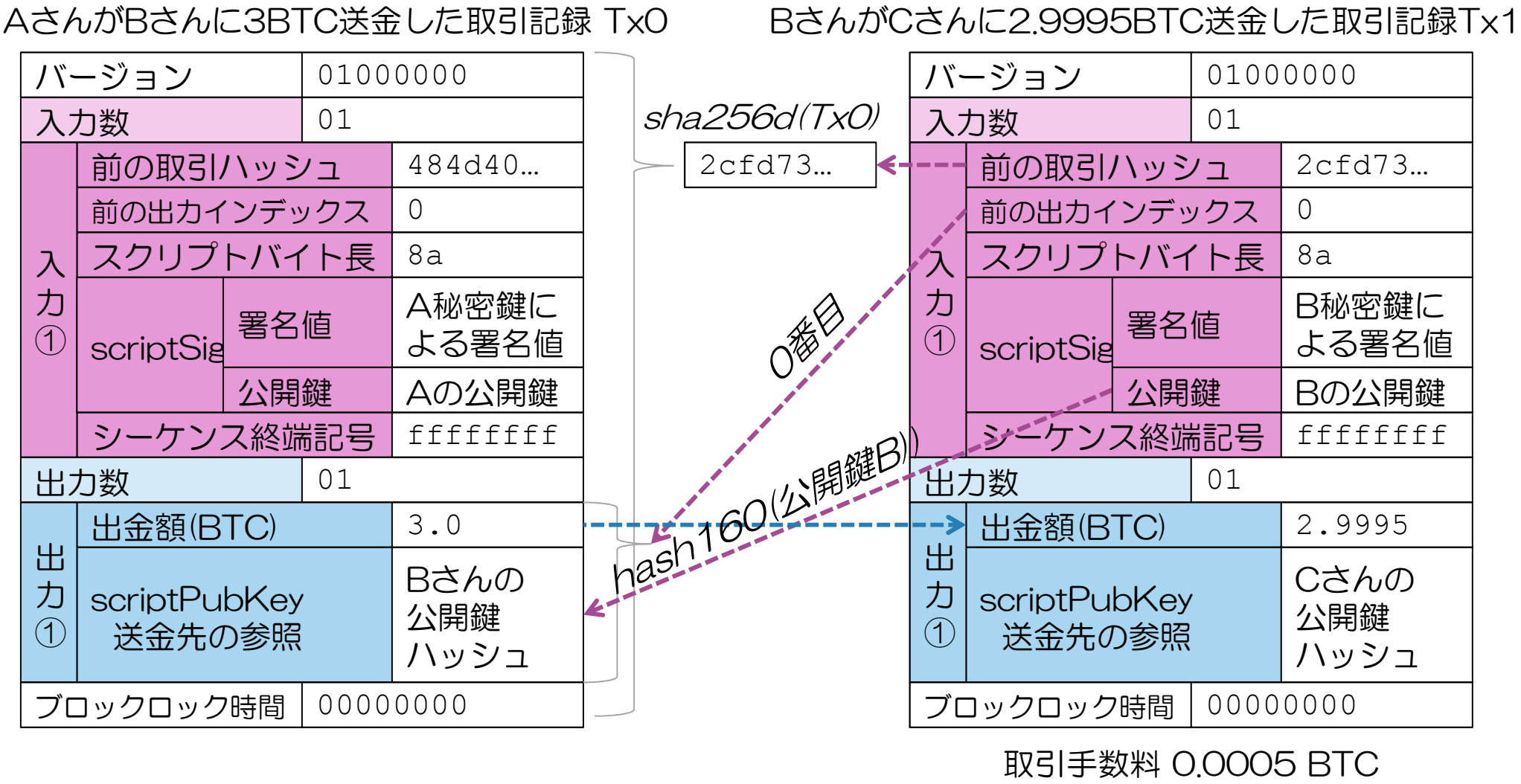
BさんからCさんへの署名済取引記録Tx1



注意点：
数値、ハッシュ値、TXIDなどはリトルエンディアン(=バイト逆順)、
スクリプトの中のデータ(ASN.1署名値、公開鍵など)はビッグエンディアンで表現される。

FUJI Xerox

BさんがCさんに送金する トランザクションの参照関係



FUJI Xerox

BさんがCさんに送金する署名済トランザクションの生成 (1/5)

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン		01000000
入力数		01
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	シーケンス終端記号	ffffffff
出力数		01
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間		00000000

とりあえず、
BからCに送金する
署名されていない枠は作る

BさんのECC secp256k1

秘密鍵	6ab3…
公開鍵	045b3…

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの生成 (2/5)

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン		01000000
入力数		01
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	スクリプトバイト長	19
	scriptPubKey 送金元の参照	Bさんの公開鍵ハッシュ
	シーケンス終端記号	ffffffff
出力数		01
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間		00000000
Hash Code Type		01000000

署名値はまだ決められないのでBさんアドの公開鍵ハッシュで仮置き(=複数入力時の場所識別)

hash160

BさんのECC secp256k1

秘密鍵	6ab3…
公開鍵	045b3…

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの生成 (3/5)

BさんからCさんへの署名前の取引記録Tx1 tbs

バージョン		01000000
入力数		01
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	スクリプトバイト長	19
	scriptPubKey 送金元の参照	Bさんの公開鍵ハッシュ
	シーケンス終端記号	ffffffff
出力数		01
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間		00000000
Hash Code Type		01000000

SHA256^2(Tx1 tbs)
二重ハッシュ計算

sha256d

3fde9e…

署名入力

ECDSA 署名器

仮トランザクション全体のダブルハッシュ

BさんのECC secp256k1

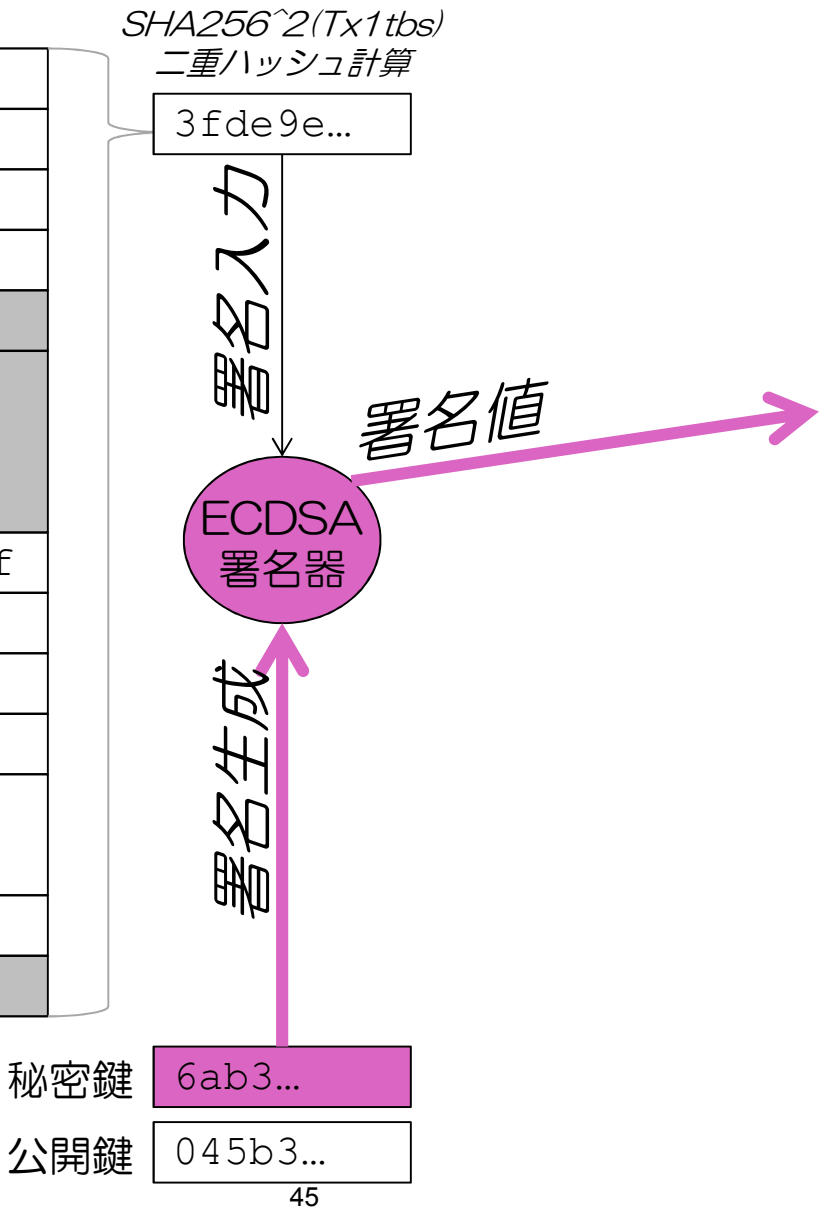
秘密鍵	6ab3…
公開鍵	045b3…

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの生成 (4/5)

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン		01000000
入力数		01
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	スクリプトバイト長	19
	scriptPubKey 送金元の参照	Bさんの公開鍵ハッシュ
	シーケンス終端記号	ffffffff
出力数		01
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間		00000000
Hash Code Type		01000000



BさんのECC secp256k1

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの生成 (5/5)

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン	01000000	
入力数	01	
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	スクリプトバイト長	19
	scriptPubKey 送金元の参照	Bさんの公開鍵ハッシュ
	シーケンス終端記号	ffffffff
出力数	01	
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間	00000000	
Hash Code Type	01000000	

SHA256^2(Tx1tbs)
二重ハッシュ計算

3fde9e…

署名入力

ECDSA 署名器

署名生成

秘密鍵 6ab3…

公開鍵 045b3…

BさんのECC secp256k1

署名値

設定

※ハッシュでなく鍵そのもの

BさんからCさんへの署名済取引記録Tx1

バージョン	01000000	
入力数	01	
入力①	前の取引ハッシュ	2cfd73…
	前の出力インデックス	0
	スクリプトバイト長	8a
	scriptSig 署名値	B秘密鍵による署名値
	scriptSig 公開鍵	Bの公開鍵
	シーケンス終端記号	ffffffff
出力数	01	
出力①	出金額(BTC)	2.9995
	スクリプトバイト長	19
	scriptPubKey 送金先の参照	cさんの公開鍵ハッシュ
ブロックロック時間	00000000	

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの検証(1/4)

BさんからCさんへの署名済取引記録Tx1

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		8a
	scriptSig	署名値	B秘密鍵による署名値
		公開鍵	Bの公開鍵
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000

hash160

7dbc…

①公開鍵を取り出し保存
②公開鍵のダブルハッシュ

BさんのECC secp256k1公開鍵 045b3…

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの検証 (2/4)

BさんからCさんへの署名済取引記録Tx1

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		8a
	scriptSig	署名値	B秘密鍵による署名値
		公開鍵	Bの公開鍵
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000

RIPEMD160(SHA256(公開鍵B))

7dbc…

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		19
	scriptPubKey 送金元の参照		Bさんの公開鍵ハッシュ
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000
Hash Code Type			01000000

署名前テンプレの生成

BさんのECC secp256k1 公開鍵 045b3…

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの検証 (3/4)

SHA256^2(Tx1tbs)
二重ハッシュ計算

BさんからCさんへの署名済取引記録Tx1

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		8a
	scriptSig	署名値	B秘密鍵による署名値
		公開鍵	Bの公開鍵
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000

RIPEMD160(SHA256(公開鍵B))

7dbc…

BさんからCさんへの署名前の取引記録Tx1tb

sha256d

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		19
	scriptPubKey 送金元の参照		Bさんの公開鍵ハッシュ
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000
Hash Code Type			01000000

3fde9e…

BさんのECC secp256k1 公開鍵 045b3…

仮テンプレ全体をsha256d

FUJI Xerox

BさんがCさんに送金する署名済トランザクションの検証 (4/4)

BさんからCさんへの署名済取引記録Tx1

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		8a
	scriptSig	署名値	B秘密鍵による署名値
		公開鍵	Bの公開鍵
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000

RIPEMD160(SHA256(公開鍵B))

7dbc…

BさんからCさんへの署名前の取引記録Tx1tbs

バージョン			01000000
入力数			01
入力①	前の取引ハッシュ		2cfd73…
	前の出力インデックス		0
	スクリプトバイト長		19
	scriptPubKey 送金元の参照		Bさんの公開鍵ハッシュ
	シーケンス終端記号		ffffffff
出力数			01
出力①	出金額(BTC)		2.9995
	スクリプトバイト長		19
	scriptPubKey 送金先の参照		cさんの公開鍵ハッシュ
ブロックロック時間			00000000
Hash Code Type			01000000

SHA256^2(Tx1tbs)
二重ハッシュ計算

3fde9e…

署名入力

ECDSA 検証器

署名検証

BさんのECC secp256k1　公開鍵

045b3…

FUJI Xerox

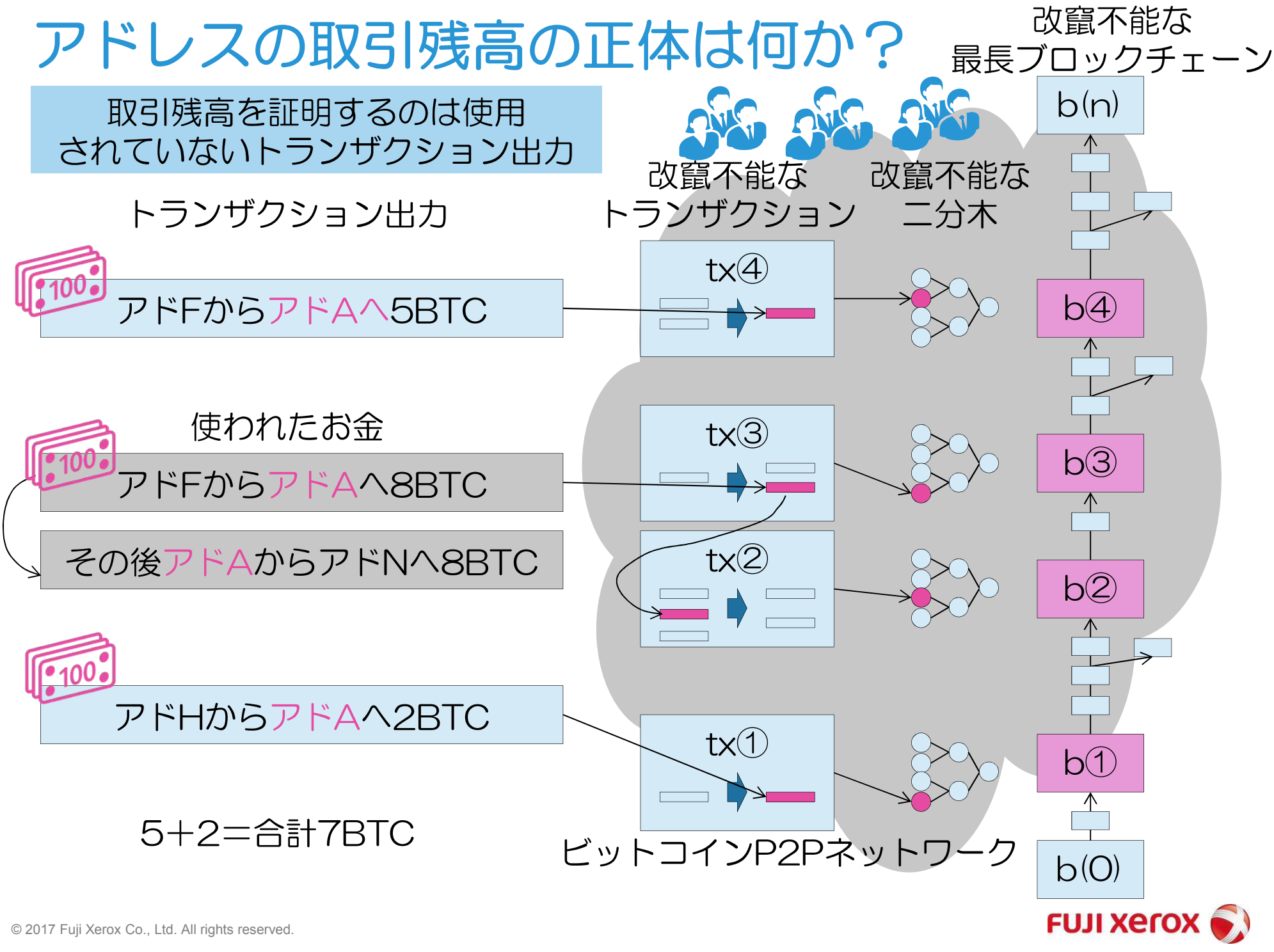
ビットコインの
お金の正体って何なの？

硬貨みたいな物じゃないらしい！

FUJI Xerox

アドレスの取引残高の正体は何か？

取引残高を証明するのは使用
されていないトランザクション出力

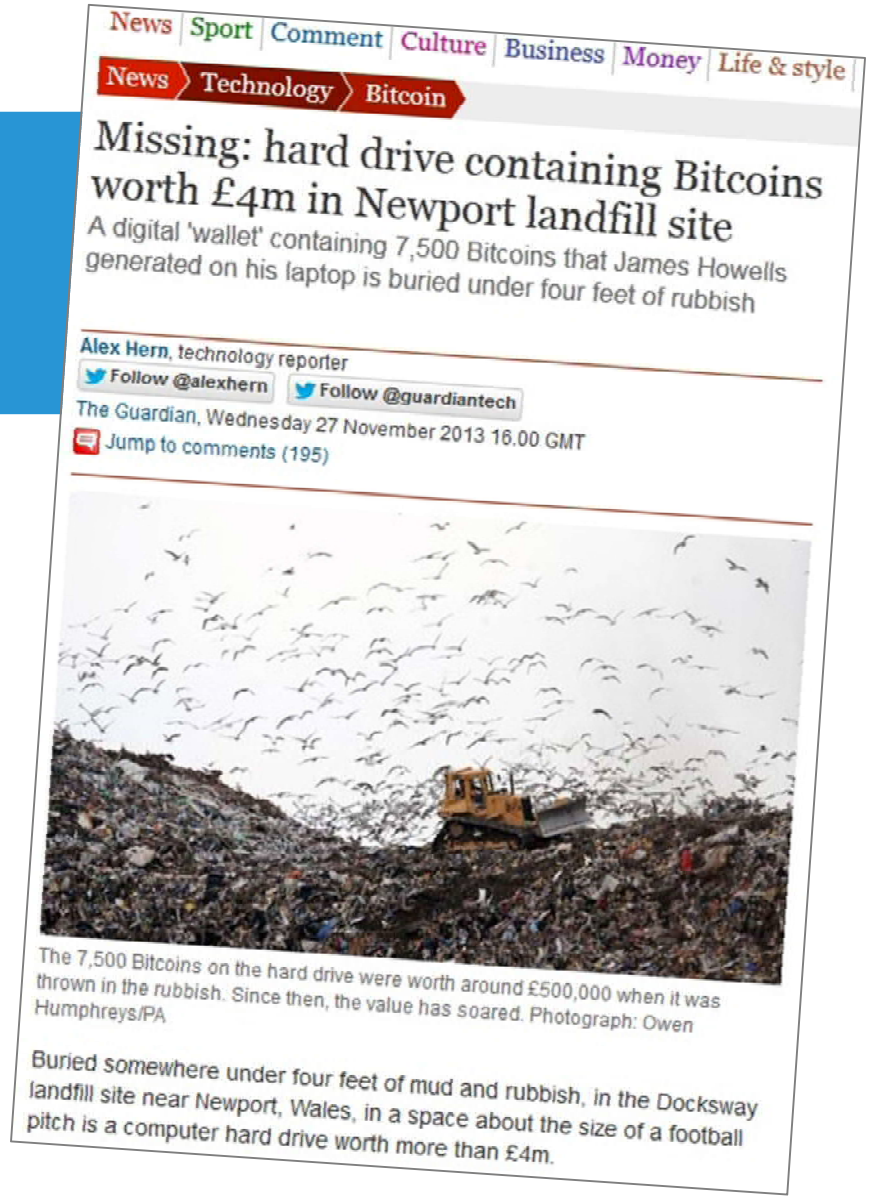


FUJI Xerox

ここでちょっと質問

7500BTC(約4億円)のビットコイン(の秘密鍵)の入ったハードディスクを間違って捨てちゃったという事件

P2Pネットワークで預金残高は保証されているのに取り戻せないの？



News | Sport | Comment | Culture | Business | Money | Life & style

News › Technology › Bitcoin

Missing: hard drive containing Bitcoins worth £4m in Newport landfill site

A digital 'wallet' containing 7,500 Bitcoins that James Howells generated on his laptop is buried under four feet of rubbish

Alex Hern, technology reporter

Follow @alexhern Follow @guardiantech

The Guardian, Wednesday 27 November 2013 16.00 GMT

Jump to comments (195)

The 7,500 Bitcoins on the hard drive were worth around £500,000 when it was thrown in the rubbish. Since then, the value has soared. Photograph: Owen Humphreys/PA

Buried somewhere under four feet of mud and rubbish, in the Docksway landfill site near Newport, Wales, in a space about the size of a football pitch is a computer hard drive worth more than £4m.

出典： http://www.theguardian.com/technology/2013/nov/27/hard-drive-bitcoin-landfill-site

FUJI Xerox

アドレスの秘密鍵をなくすとどうなる？

秘密鍵が無いと、もう
誰にもお金をあげられない

秘密鍵が無いと、
もう換金もできない

いわゆるお金が塩漬けの状態
(秘密鍵のバックアップが大事!)

FUJI Xerox

ビットコインの用語の説明(2)

① ブロックチェイン
② コインベーストランザクション
③ マイニング(発掘)

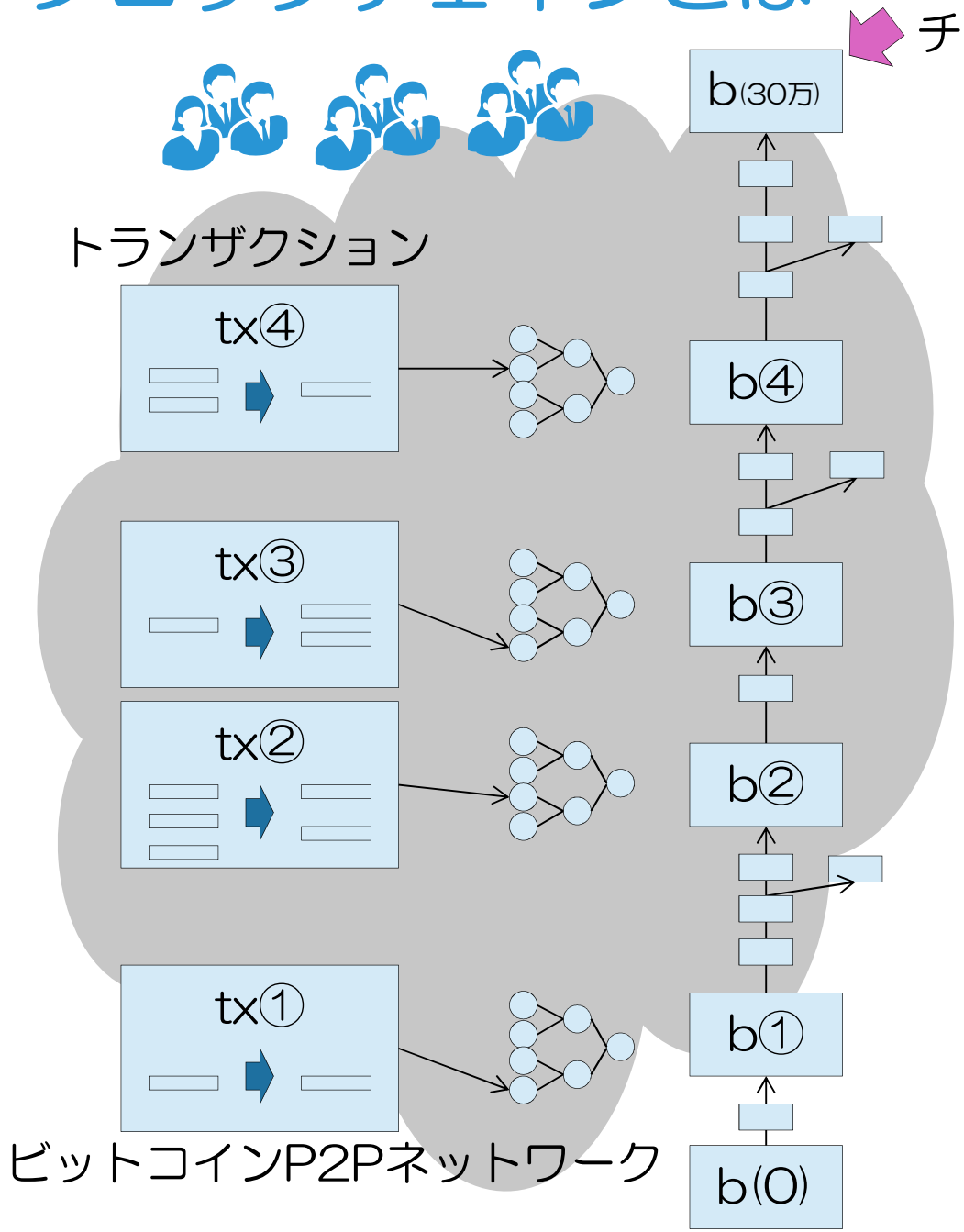
FUJI Xerox

ビットコインの
ブロックチェインについて

FUJI Xerox

ブロックチェインとは

(2017年7月現在)



- P2Pネットでトランザクションを改竄されないようにするための仕組み
- 約10分に1つブロックが増える
- 1ブロックで2千のトランザクションを含む
- 今47万ブロック
- 年5万ブロック増える
- ブロックを作った人には賞金(今は25BTC)

FUJI Xerox

なぜブロックチェインが必要なのか？

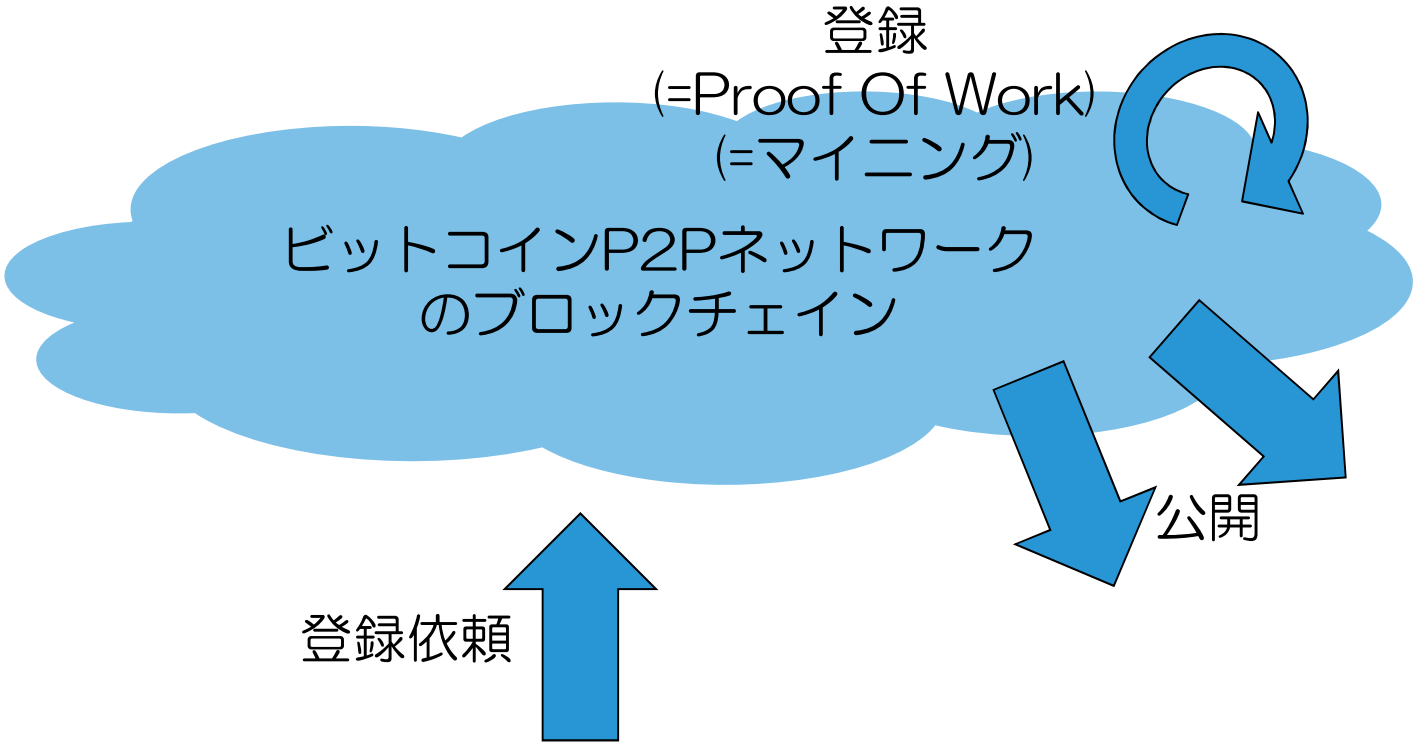
ビットコインP2Pネットワーク

Bさんの署名つきトランザクション	
入力	出力
Bさんには ７BTCある	Cさんに ７BTC渡す

このトランザクションは、ただ署名しただけであって、
ビットコインネットワークではこの取引を認めていない

FUJI Xerox

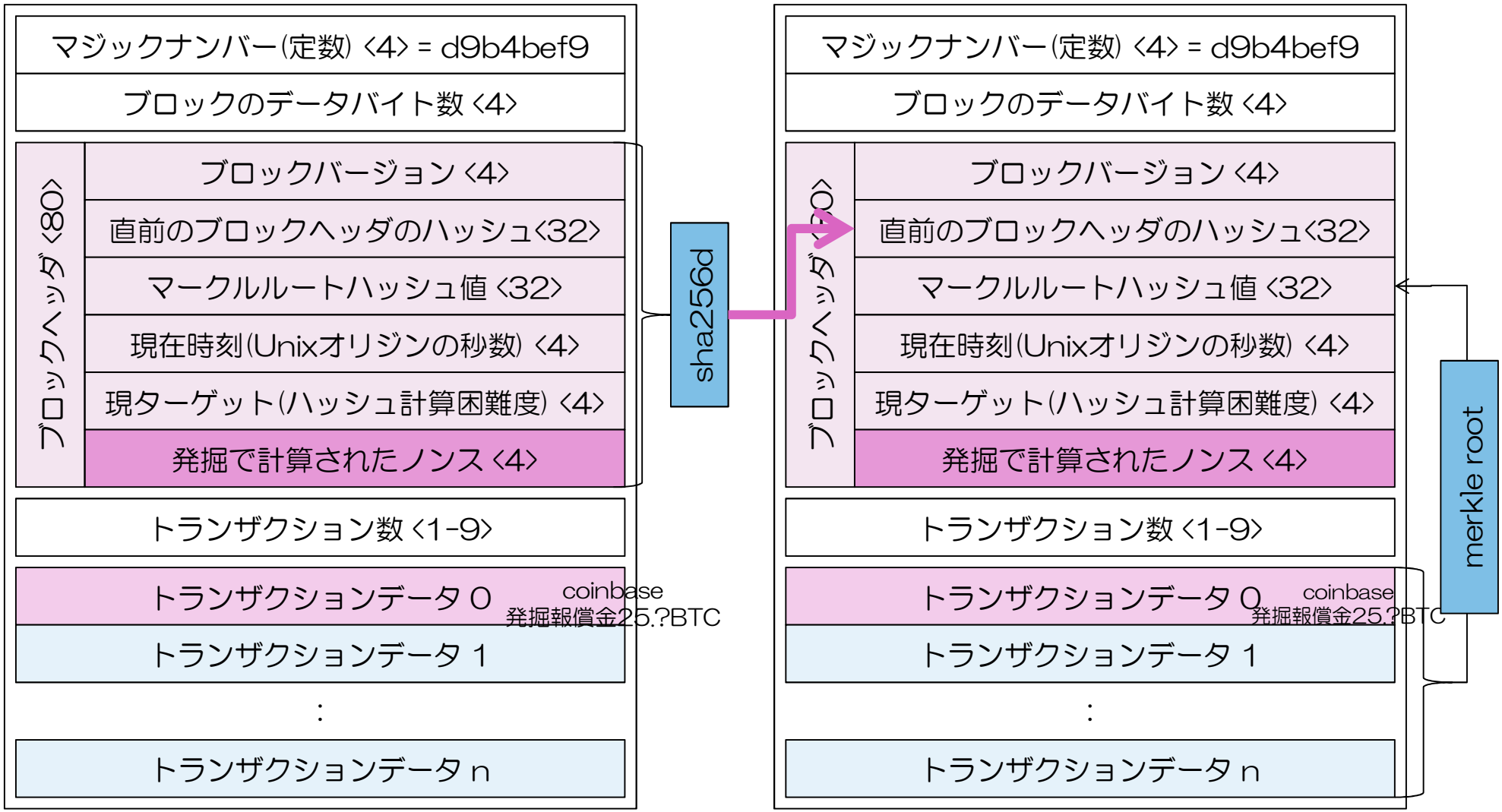
なぜブロックチェインが必要なのか？



P2Pネットワークに受け入れてもらうことで
取引が全ビットコイン利用者に受理される

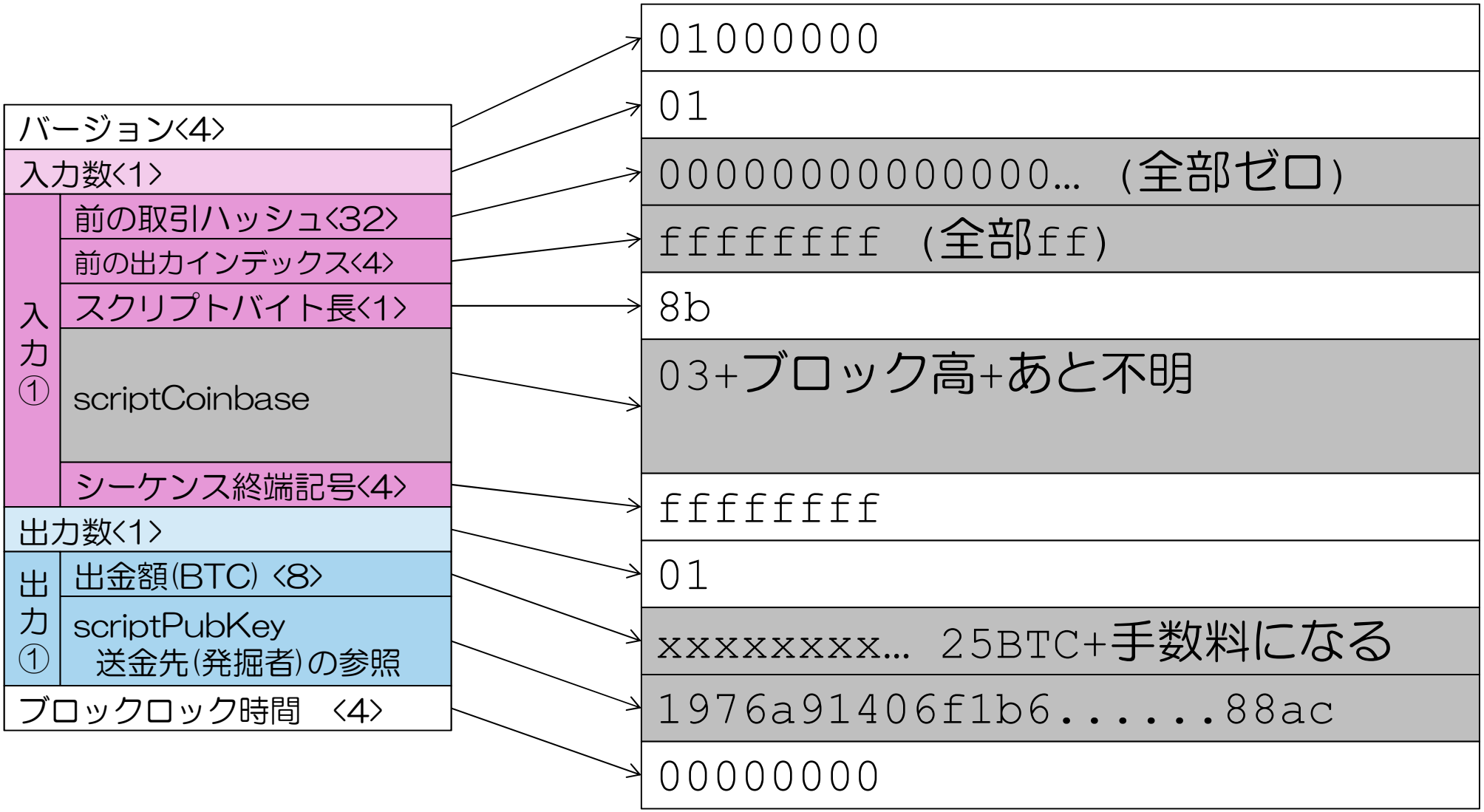
FUJI Xerox

ブロックのデータ構造



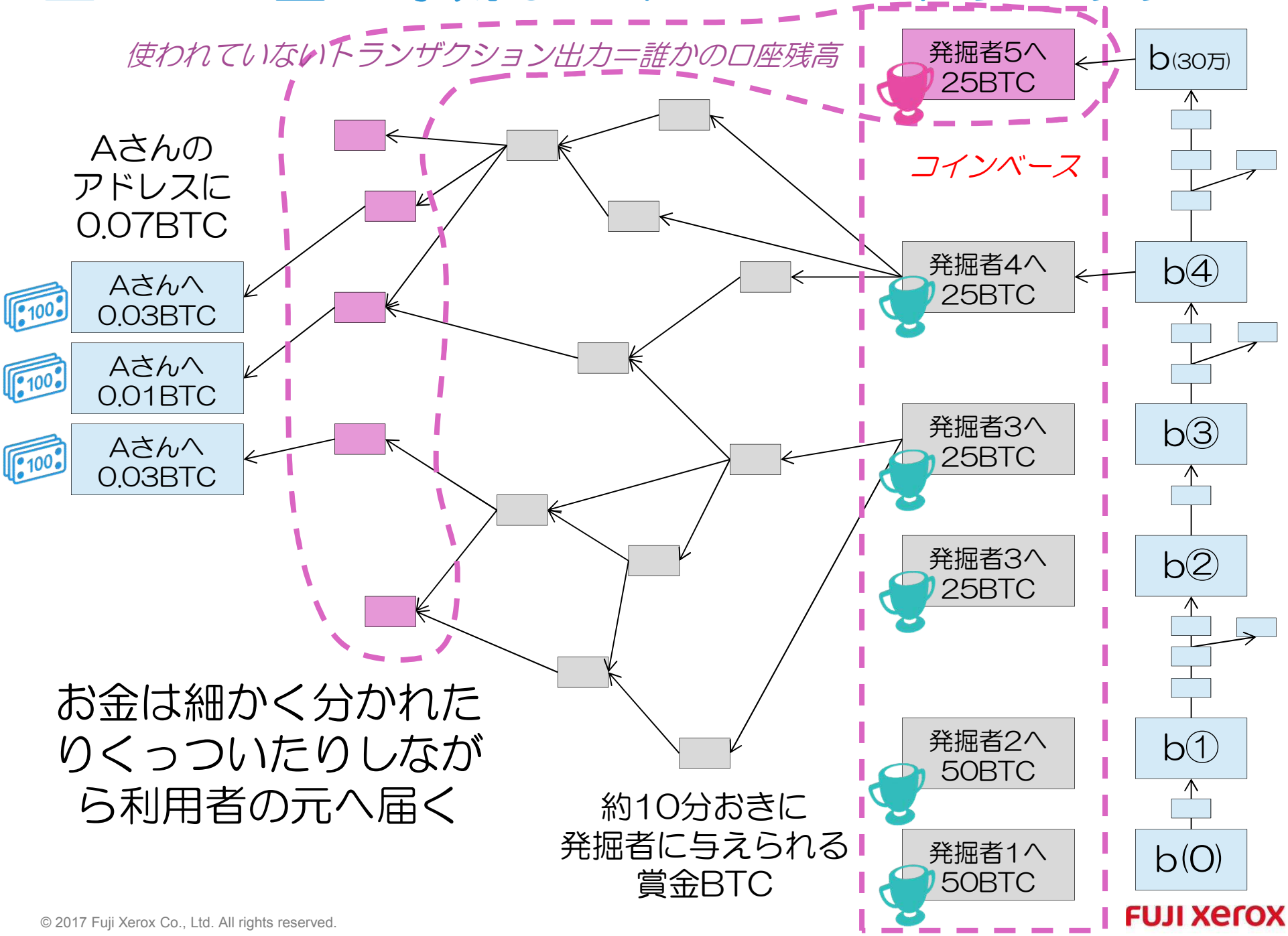
FUJI Xerox

コインベーストランザクション(発掘者への報酬25BTC)



FUJI Xerox

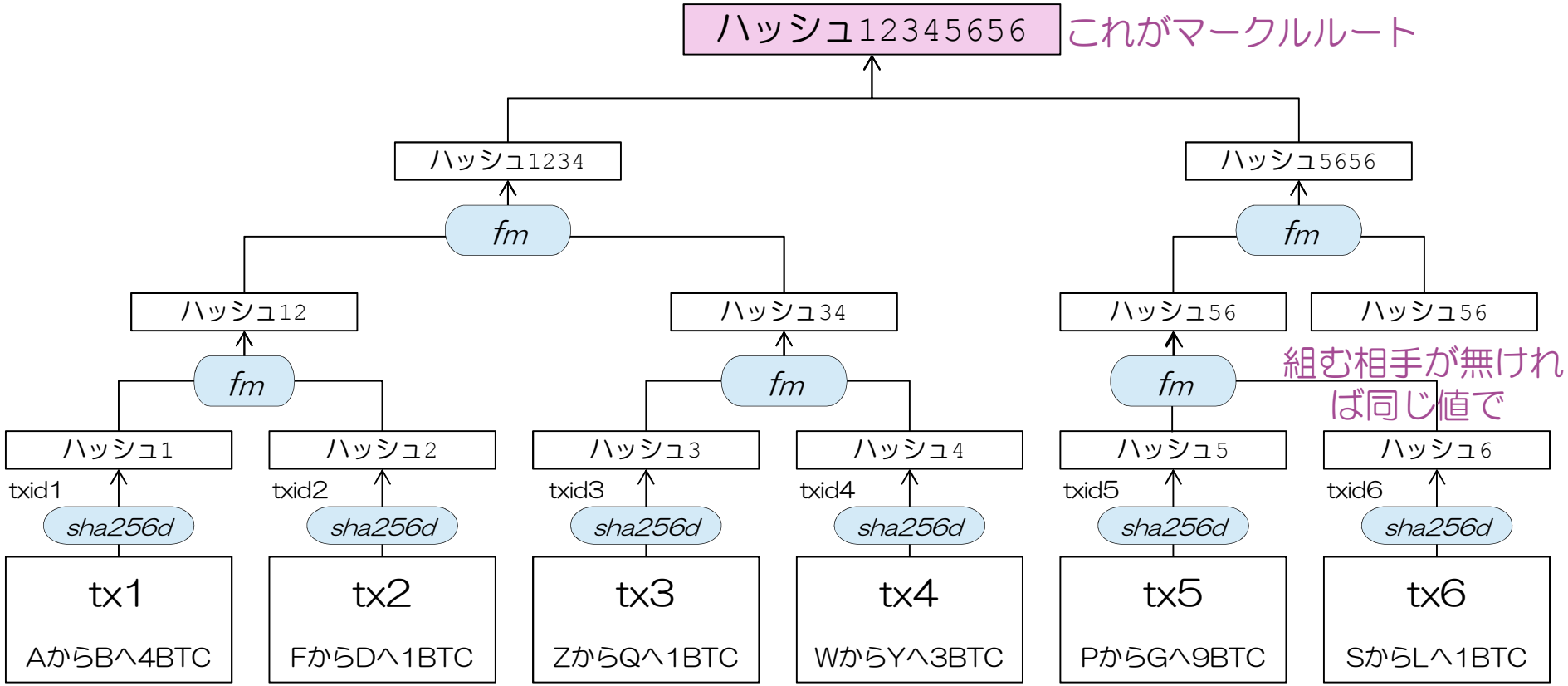
全てのお金の源流はコインベーストランザクション



マークルルートハッシュの計算

ブロックに含まれる全トランザクションデータの
ハッシュ値を木構造で組み合わせてそれをルート値として管理する。
→ 後続ブロックを持った時点でトランザクションを改竄できなくなる

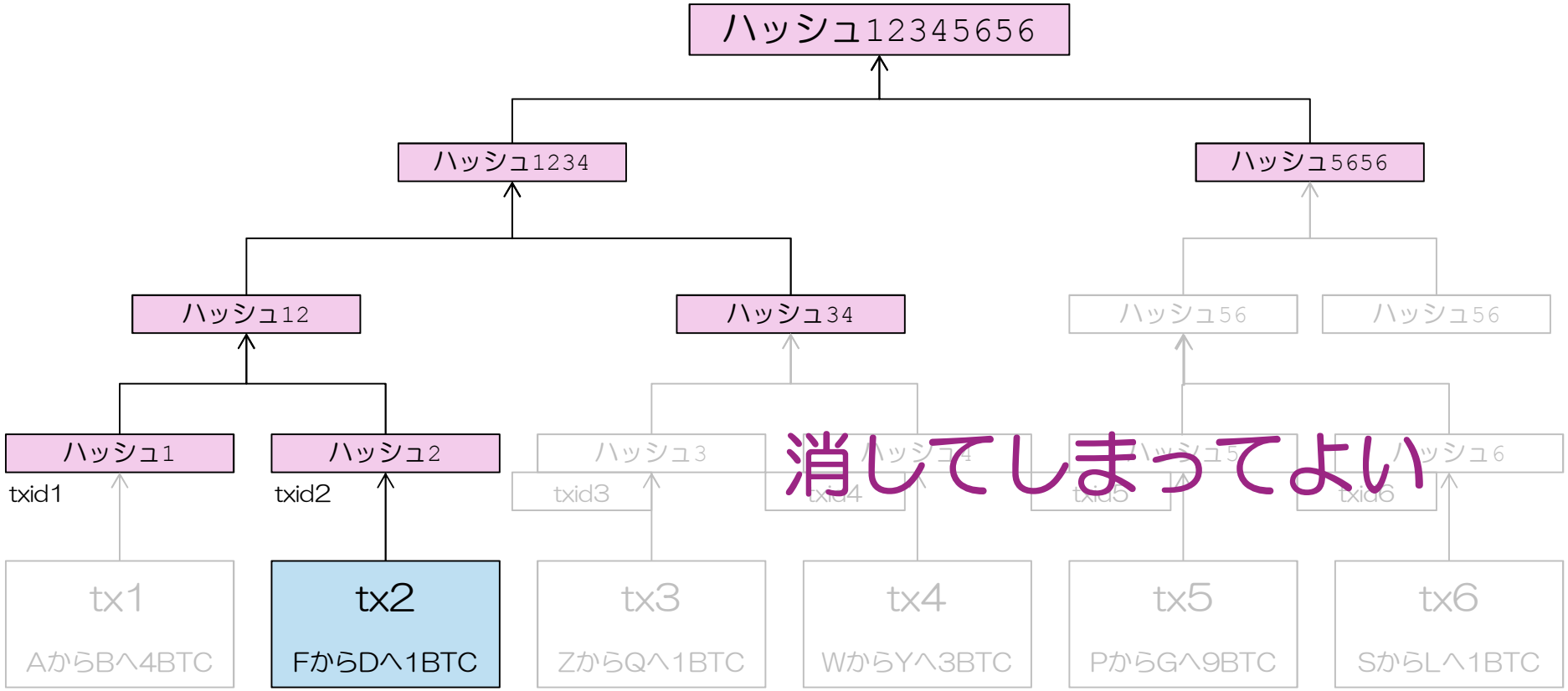
結合関数 *fm* =sha256d(rev(txida)+rev(txid$_b$))　※ rev: バイトの逆順



FUJI Xerox

マークルルートハッシュのメリット
いらない部分を保管しなくて済む

例えば、tx2が自分の関連するトランザクションだとして、
関係するものだけを残して後で検証したい場合、
下の薄いのは消してもいつでも再検証ができる



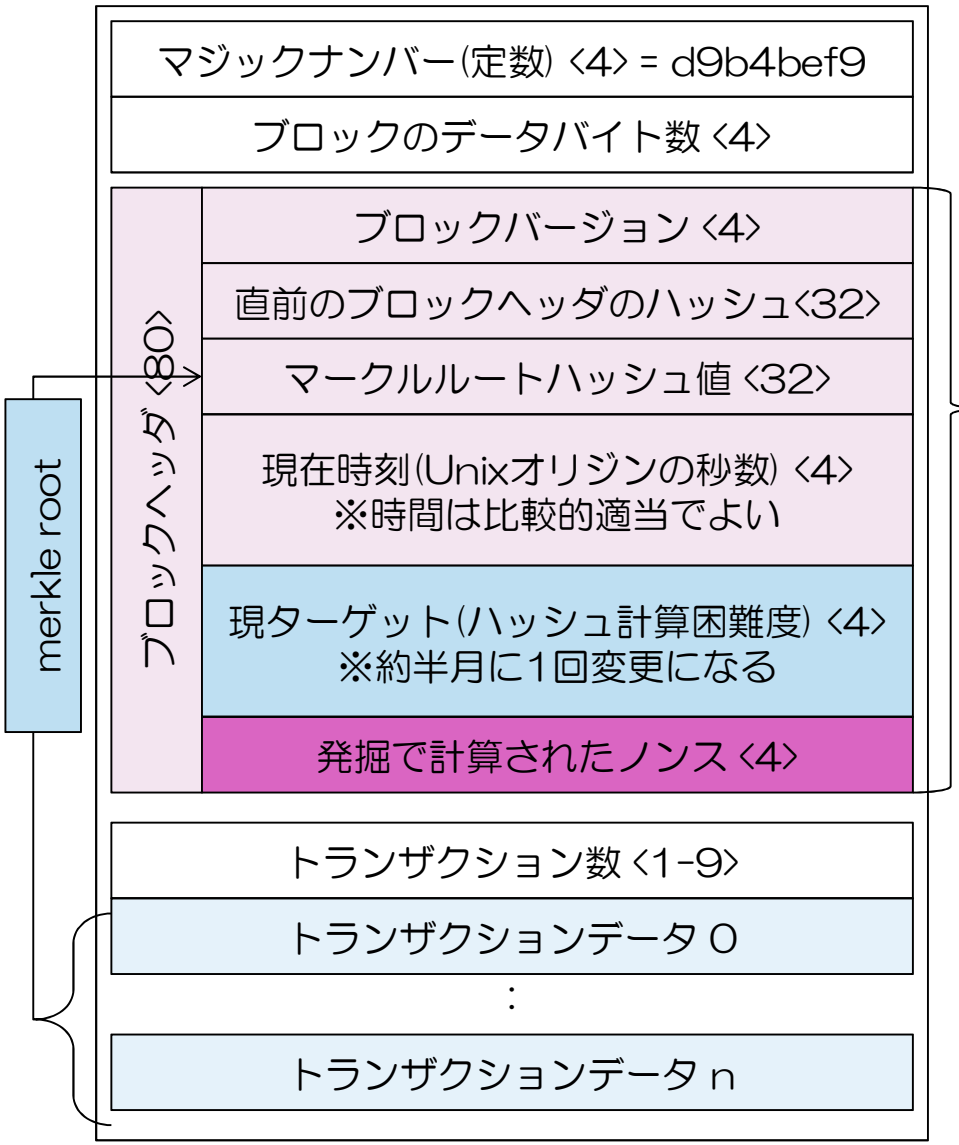
FUJI Xerox

ブロックの発掘

FUJI Xerox

ブロックの発掘とは？

ターゲットの値によりブロックヘッダ
ハッシュ値の許容される最大値が得られる
(全P2Pネットワーク上で
発掘が10分程度で終わるように調整)



nonce	hash
0	5c56c2883435b38aeba0e69fb2e0e3db3b22448d3e17b903d774dd5650796f76
1	28902a23a194dee94141d1b70102accd85fc2c1ead0901ba0e41ade90d38a08e
2	729577af82250aaf9e44f70a72814cf56c16d430a878bf52fdaceeb7b4bd37f4
3	8491452381016cf80562ff489e492e00331de3553178c73c5169574000f1ed1c
39	03fd5ff1048668cd3cde4f3fb5bde1ff306d26a4630f420c78df1e504e24f3c7
990	0001e3a4583f4c6d81251e8d9901dbe0df74d7144300d7c03cab15eca04bd4bb
52117	0000642411733cd63264d3bedc046a5364ff3c77d2b37ca298ad8f1b5a9f05ba
1813152	00000c94a85b5c06c9b06ace1ba7c7f759e795715f399c9c1b1b7f5d387a319f
19745650	000000cdccf49f13f5c3f14a2c12a56ae60e900c5e65bfe1cc24f038f0668a6c
243989801	0000000ce99e2a00633ca958a16e17f30085a54f04667a5492db49bcae15d190
856192328	0000000000000000e067a478024addfecdc93628978aa52d91fabd4292982a50

先頭が0が16個つくために
85万回ハッシュ計算をした

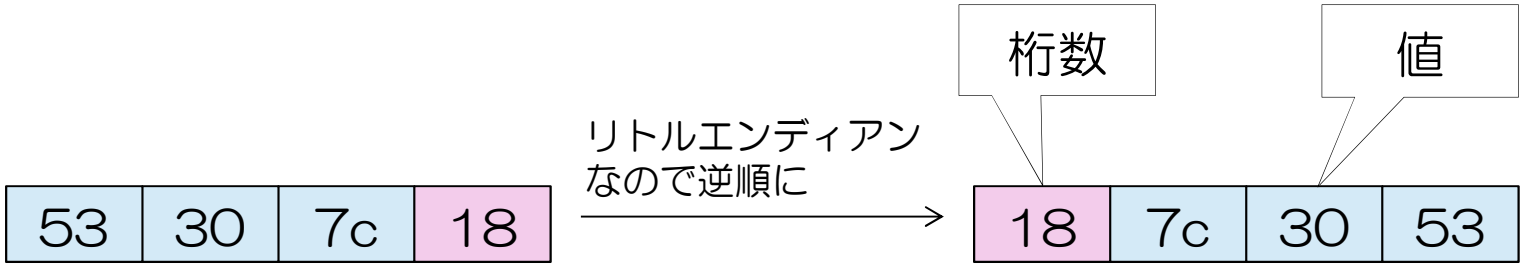
表の出典：Ken Shirriff's blog
Bitcoin mining the hard way: the algorithms, protocols, and bytes
http://www.righto.com/2014/02/bitcoin-mining-hard-way-algorithms.html

FUJI Xerox

ブロックのターゲット値の計算は？（v1当時）

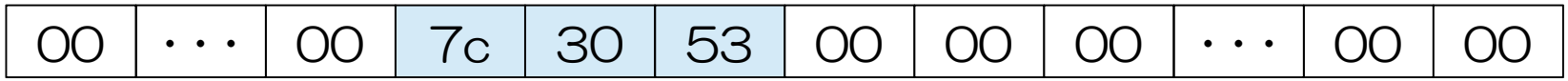
現ターゲット(ハッシュ計算困難度) <4>
※約半月に1回変更になる

ブロックヘッダ中の4バイト圧縮表現ターゲット



16進数32バイト表現のターゲット値(=hash256dの結果長と同じ)

先頭32-24=8バイトがゼロ　　全体でx18=24バイトになるように

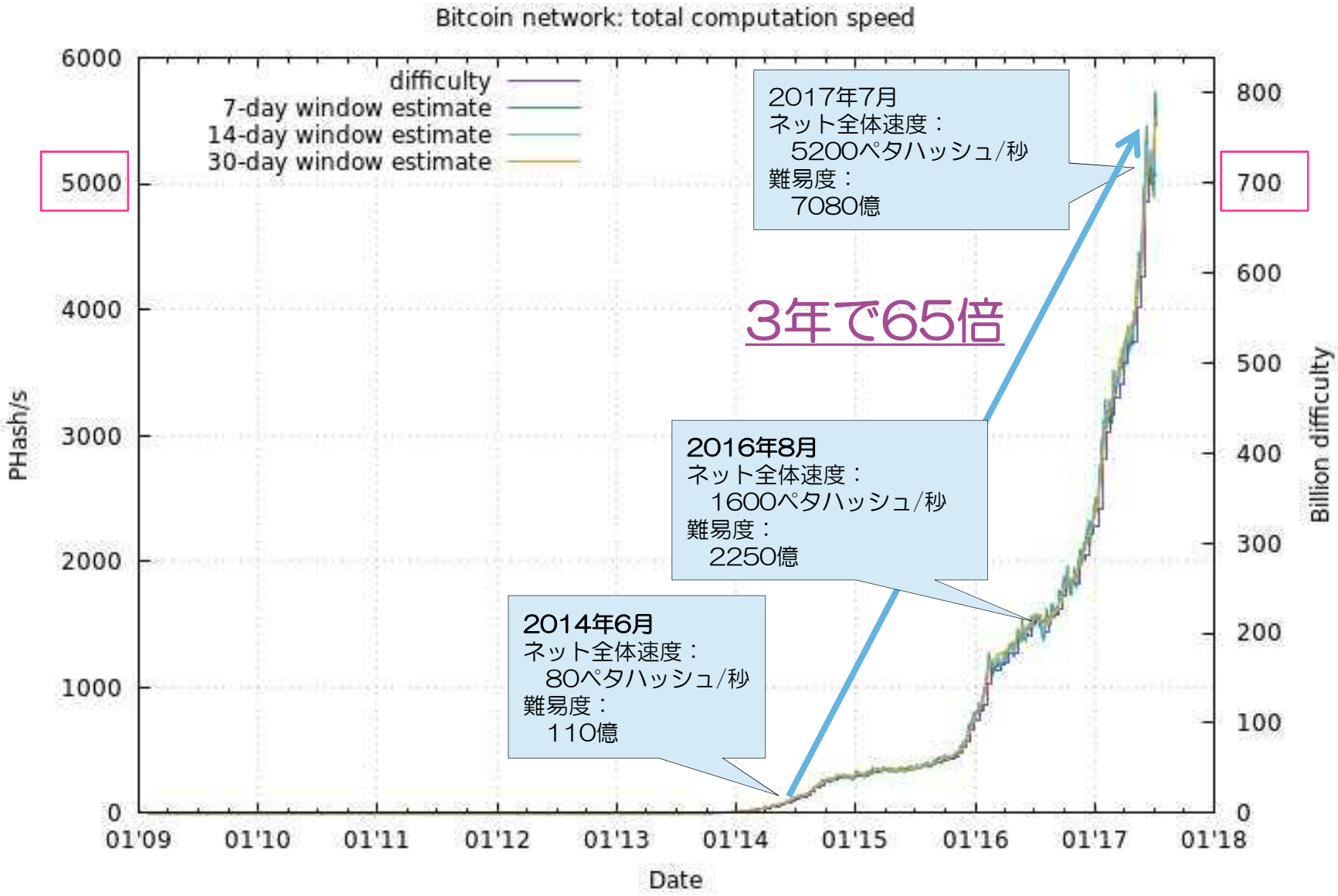


この値「以下」になるように発掘(=ハッシュ計算)を繰り返す

FUJI Xerox

難易度(Difficulty)の推移 (2017年7月現在)

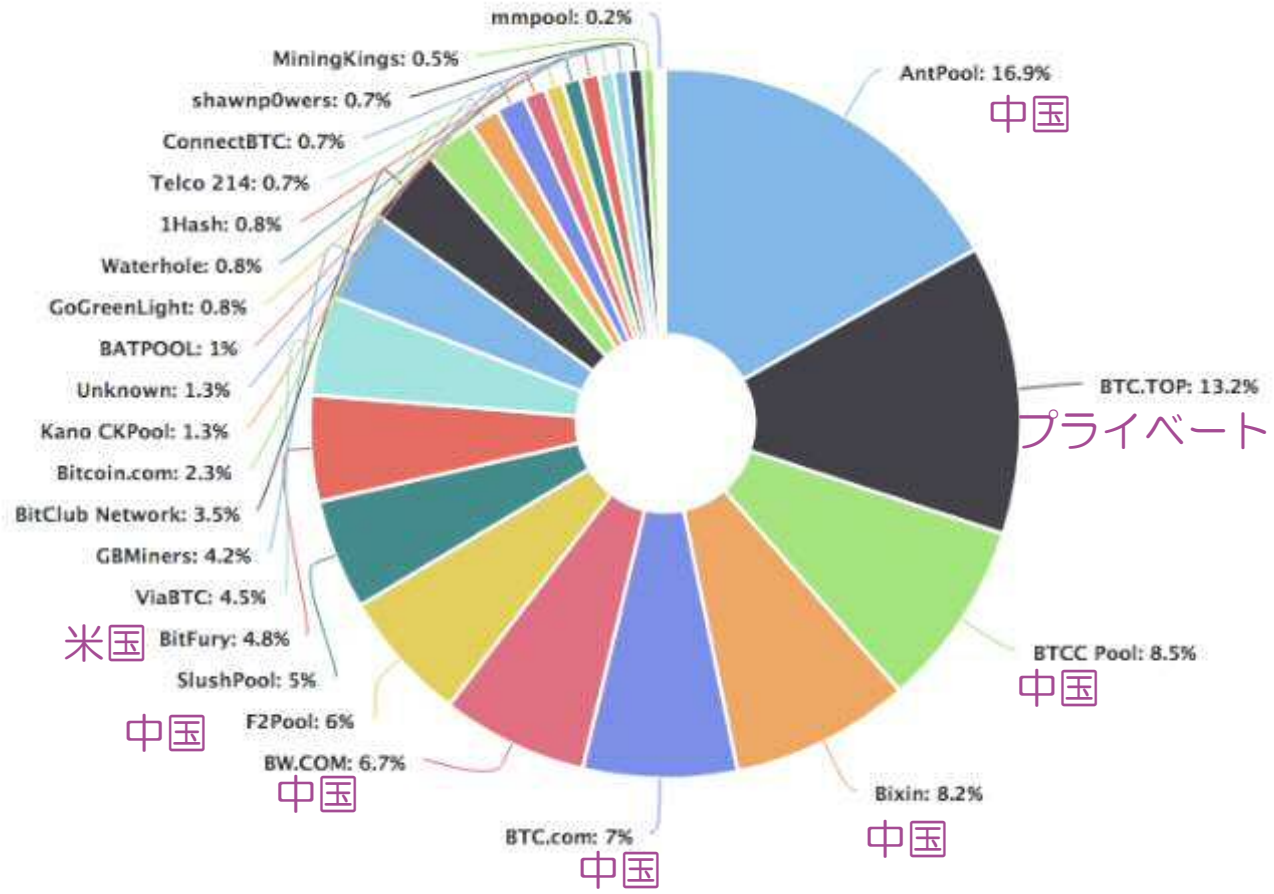
当然、ネット全体でのハッシュ計算速度と難易度は相関がある



FUJI Xerox

2016年、発掘プール市場における中国の台頭、2017年も続く

発掘プールの中国資本は2014年40%が2016年70%になった
50%越えでビザンチン将軍問題で不正取引が可能では？



2017年7月の発掘プール比率
出典：www.blocktrail.com

中国の水冷による発掘施設、月2億稼ぐ施設も
出典：http://gigazine.net/news/20131126-bitcoin-mining-facility/

FUJI Xerox

ビットコインに対する不安

FUJI Xerox

ビットコインに対する(最近の)不安

①取引き可能量が限界に近づいている
(ハードフォーク、ビットコインアンリミテッド)
②手数料高騰で当初期待していた少額決済には向かない
③とある国家によるビザンチン将軍問題による不正取引の可能性
④使われている暗号技術が危殆化したらどうするのか

- 不安があれば、利用者は(イーサリアムなど)他の通貨に移動すればよいのでは？ 簡単に移動できる
- 仕組みについては、ビットコインを維持しなければならない人だけが苦労すればよいのでは？

FUJI Xerox

まとめ

- 本日の内容
  - 最新動向をまじえた、ビットコインとは何かについて
  - ビットコインで使われている暗号技術
  - ビットコインで使われるデータ構造、仕組み
  - ビットコインで最近起きている不安

- 古いビットコインの仕組みを学ぶことで、今後の新しいブロックチェーン技術の理解の助けになれば幸いです。

FUJI Xerox