

# Bitcoinの技術的課題の整理と議論 2014.6.2

セコム株式会社 IS研究所  
暗号・認証基盤グループ  
佐藤 雅史

# Bitcoinの技術的課題の整理と議論 2014.6.2

日本ネットワークセキュリティ協会  
電子署名WG サブリーダー  
佐藤 雅史

# 自己紹介

- 佐藤 雅史 (さとう まさし)
- 普段は主に電子署名屋さんです。
- 標準化とかやっています。
  - JIS X 5092:2008 CMS利用電子署名(CAdES)の長期署名プロファイル
  - JIS X 5093:2008 XML署名利用電子署名(XAdES)の長期署名プロファイル
  - ISO 14533-1:2012, Long term signature profiles - Part 1: Long term signature profiles for CMS Advanced Electronic Signatures (CAdES)
  - ISO 14533-2:2012, Long term signature profiles - Part 2: Long term signature profiles for XML Advanced Electronic Signatures (XAdES)
  - ISO 14533-3 (計画中)
- 主な活動の場
  - JNSA, TBF (Time Business Forum), JIPDEC

# ビットコインの特徴 (簡単なおさらい)

- コイン所有権の移動をトランザクションの履歴で管理する。
  - コインの使用を表明するのに電子署名を用いる。
- トランザクションの存在を証明するためにブロック（台帳のようなもの）を生成しネットワーク参加者で共有する。
  - ブロック生成者は新しいコインや発生した取引手数料を獲得できる（マイニング）。
- コイン発行や使用に関してコントロールする中央の機関は存在しない。

いきなり余談ですが…  
(電子署名屋さんぽいことを…)

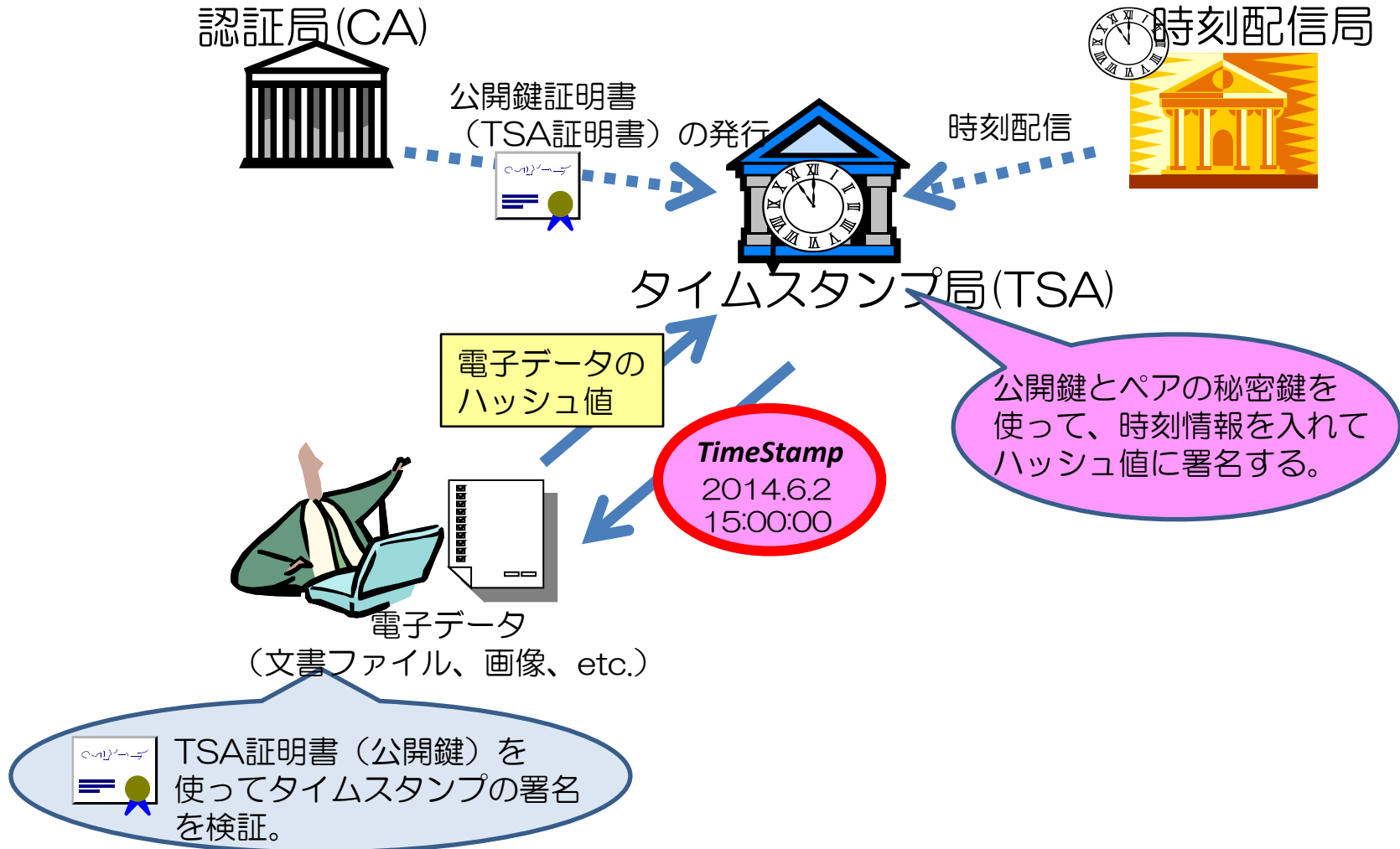
# 電子署名屋さんから見たビットコイン

- タイムスタンプ
  - 第三者的に証明可能な電子データの存在証明。  
（ある時間にそのデータが存在した証明）
  - よく電子署名と組み合わせて使われる。
- ビットコインはP2P型のタイムスタンプ

# タイムスタンプ技術 (1)

## ～ PKI (公開鍵基盤) 方式 ～

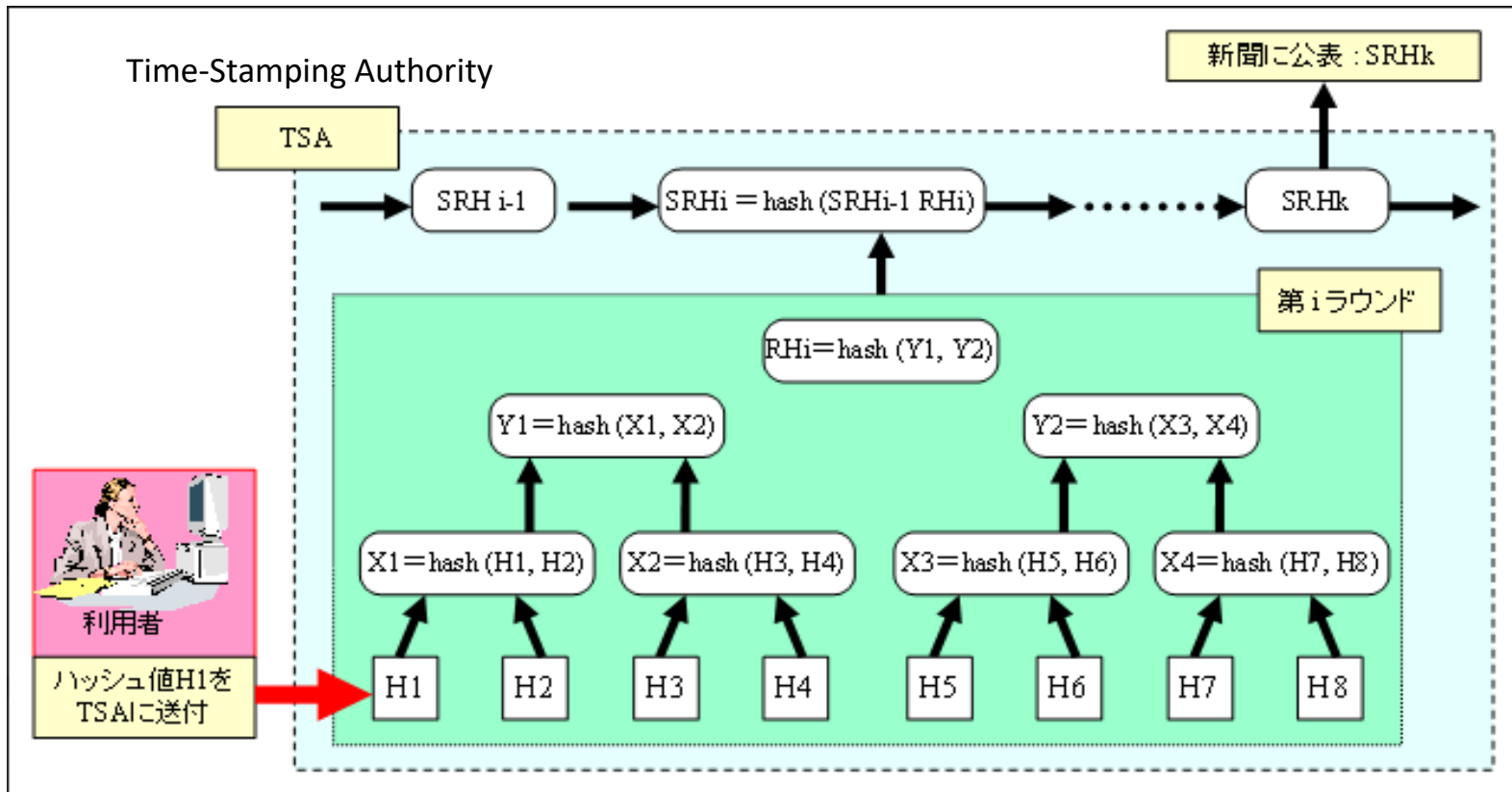
標準規格：RFC 3161, ISO/IEC 18014-2



# タイムスタンプ技術 (2)

## ～ リンキング方式 ～

標準規格：ISO/IEC 18014-3



引用元：<https://www.ipa.go.jp/security/pki/O93.html>

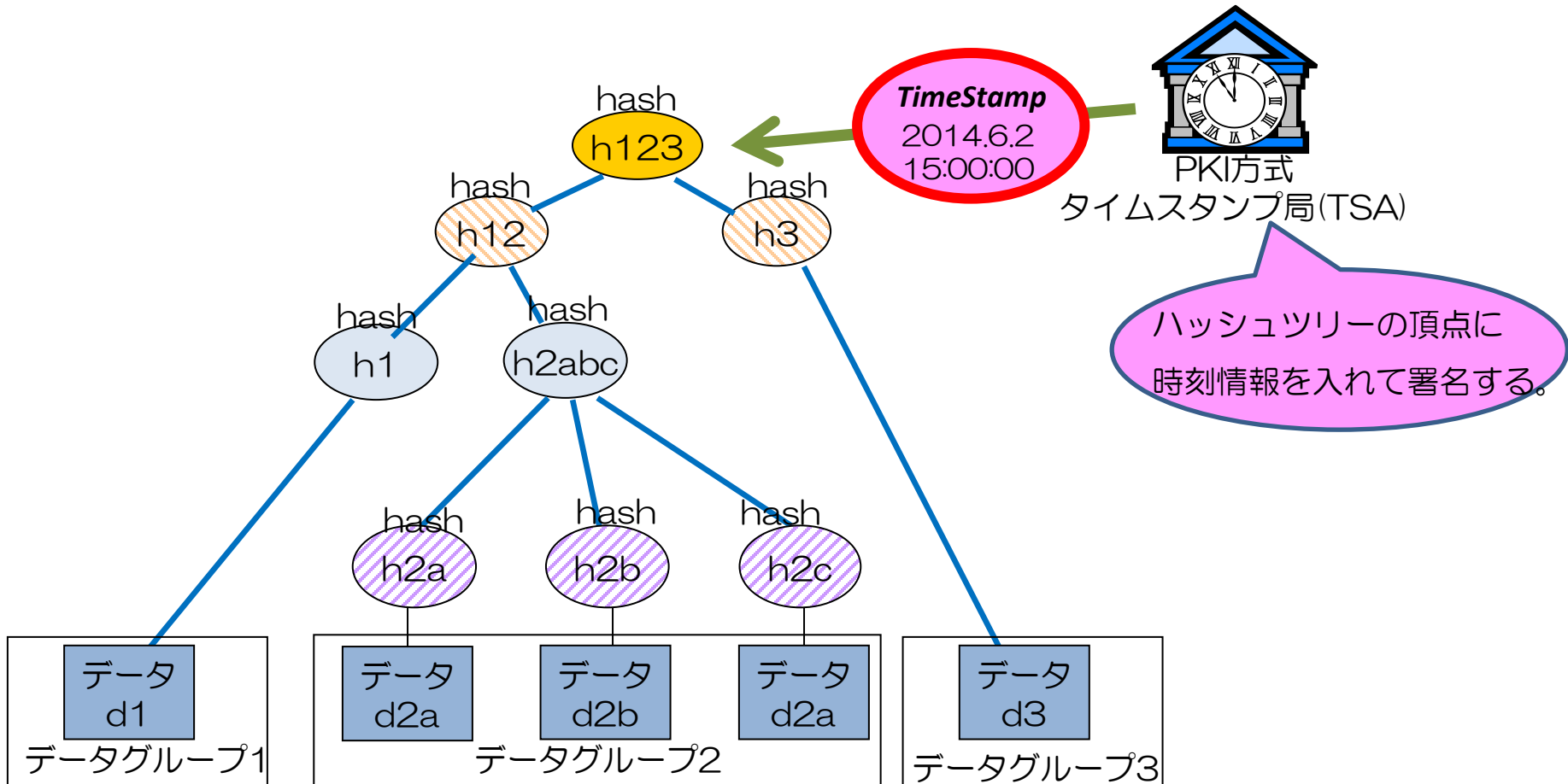
ハッシュアルゴリズムが脆弱化しない限り有効。  
 逆に言えば、ハッシュアルゴリズムの移行が困難。  
 (ビットコインも同様の問題がある?)



# タイムスタンプ技術 (3)

## ～ ERS(Evidence Record Syntax)方式 ～

標準規格 : RFC 4998, RFC 6283



PKI方式タイムスタンプを用いたリンクングのような仕組み。  
ハッシュアルゴリズムの移行ができる。

# ビットコインのタイムスタンプ

	従来のタイムスタンプ	ビットコインのタイムスタンプ
信頼のモデル	信頼できる第三者機関が存在する。	中央の機関は存在しない。 多数決による合意。
対象とするデータ	任意の電子データ。 (文書、画像データ、ログ等)	ビットコインのトランザクション (基本的には…)
ビジネスモデル	タイムスタンプ付与者 または検証者への課金。	利用者は無料 (minerが獲得する発掘コインと Transaction Fee)

# ビットコインを用いた存在証明①

## ～Proof of Existence～

Select a document and have it certified in the Bitcoin blockchain [What?](#)

Click or drag and drop your document here.  
Your document will NOT be uploaded. The cryptographic proof is calculated client-side.

Last documents registered:

Document Digest	Timestamp
<a href="#">af4d081feed4771fcb9374e56f7d8fc211aae26045331aa5d1886b282637e0c0</a>	2014-05-29 20:21:41
<a href="#">954f7d96502b5c5fe2e98a5045bca7f5e9ba11e3dbf92a5c0214a6aa4c7f2208</a>	2014-05-29 10:53:52
<a href="#">9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08</a>	2014-05-29 10:43:52
✓ <a href="#">7ecda76861bbdd15e50cf9915113e3939b6301ae4187a9b3c66f0cd03162e5ff</a>	2014-05-29 07:52:54
✓ <a href="#">1b560ca759d3a99f9fb7d66a1ba2e21a465ab14eac29a3381c0298523f11d64d</a>	2014-05-29 04:23:56

Last documents confirmed in the blockchain:

Document Digest	Timestamp
✓ <a href="#">7ecda76861bbdd15e50cf9915113e3939b6301ae4187a9b3c66f0cd03162e5ff</a>	2014-05-29 07:52:54
✓ <a href="#">1b560ca759d3a99f9fb7d66a1ba2e21a465ab14eac29a3381c0298523f11d64d</a>	2014-05-29 04:23:56

<http://www.proofofexistence.com/>

- 存在証明したい文書のハッシュ値(SHA256)を作る。
- 文書のハッシュ値とマーカをビットコイントランザクションのOutput Scriptに格納する。
  - Output Scriptの格納にはOP\_RETURNを用いる。
  - OP\_RETURNの後ろには任意のデータ(40bytesまで)を入れられる。

# ビットコインを用いた存在証明①

## ～Proof of Existence～

The screenshot shows the 'Proof of Existence' website interface. At the top, there is a navigation bar with links for 'Proof of Existence', 'Prove', 'About', 'API', 'Contact', and 'Source Code'. Below the navigation bar, the page title is 'Document information' followed by a document digest hash: `7ecda76861bddd15e50cf9915113e3939b6301ae4187a9b3c66f0cd03162e5ff`. This hash is highlighted with a red box. Below the hash, it states 'Registered in our servers since: 2014-05-29 07:52:54' and 'Transaction broadcast timestamp: 2014-05-29 12:33:07'. There is an icon of a magnifying glass with a green checkmark. A green box contains the text 'Document proof embedded in the Bitcoin blockchain!'. Below this, a grey box contains the heading 'Congratulations!' and a paragraph: 'This document's digest was successfully embedded in the Bitcoin blockchain. It is now permanently certified and proven to exist since the transaction was confirmed.' It also includes a blue link for 'Transaction 1cd5dce3f130f3d8d0fbbb904fbd9a6f83020a824ccb8ce1f9e26ee1c88986d8' and instructions on how to verify the document in the future.

### Last documents confirmed in the blockchain:

Document Digest	Timestamp
<code>7ecda76861bddd15e50cf9915113e3939b6301ae4187a9b3c66f0cd03162e5ff</code>	2014-05-29 07:52:54
<code>1b560ca759d3a99f9fb7d66a1ba2e21a465ab14eac29a3381c0298523f11d64d</code>	2014-05-29 07:52:54

# ビットコインを用いた存在証明①

## ～Proof of Existence～

取引 ビットコイン取引の詳細情報を閲覧する

1cd5dce3f130f3d8d0fbb904fd9a6f83020a824ccb8ce1f9e26ee1c88986d8

1BtScVjtFzflDe62v34uor6JZvMTBxWt1M (0.0001 BTC - Output) Unable to decode output address - (Unspent) 0 BTC

Summary		インプットおよびアウトプット	
Size	241 (bytes)	合計インプット	0.0001 BTC
受け取り時刻	2014-05-29 12:58:04	合計アウトプット	0 BTC
ブロックに含まれています	303167 (2014-05-29 12:58:04 +0 minutes)	Fees	0.0001 BTC
認証済み	139 認証済み	推定取引完了BTC	0 BTC
IPIによる中継	207.12.89.16 (whois)	Scripts	スクリプトおよびコインベースを隠す

### Input Scripts

3045022100e264efcc164d41b7eeb33d4d836eb440daa607f129790f0e387dcfb157675c1a022022fe3c81c401d06dbe50a4ba7dcd79458c24e4af1165240452c17461d0ef91660104bd184b34e4e20698a7670854e16f68c4ca2f9326572342998bdf1b1c4685644c2374e40c19ca20eeb3439e3255d468d3e92aa32f577df99bdb409c8f064462f7

### Output Scripts

OP\_RETURN 444f4350524f467ecda76861bbdd15e50cf9915113e3939b6301ae4187a9b3c66f0cd03162e5ff (デコード済み) j(DOCPROOF~ha...Q功...A...o...1b...)

奇妙な取引



# ビットコインを用いた存在証明②

## ～Virtual Notary～

### VIRTUAL-NOTARY *Notarize the intangible...*

Home Examine Certificate Get Certificate - FAQ About

Welcome to Virtual-Notary - a free and secure electronic notary service.













How does it work?

You select a factoid that you would like notarized. We check that factoid, create a record of it that you can refer to later, and issue you a cryptographically-signed certificate that attests to that factoid.

Use Cases

This [blog post](#) describes some of the use cases for Virtual Notary.

What would you like notarized?

 <p><b>Document</b> Notarize a document</p>	 <p><b>Web Page</b> Notarize the HTML content of a web page</p>
 <p><b>Twitter Feed</b> Notarize a twitter post anybody made</p>	 <p><b>Stock Prices</b> Notarize stock price information for public companies</p>
 <p><b>Weather Conditions</b> Notarize current weather conditions</p>	 <p><b>Exchange Rates</b> Notarize exchange rates for currencies worldwide</p>
 <p><b>DNS Entry</b> Notarize DNS information for a website</p>	 <p><b>Email Address Verification</b> Notarize the ownership of an email address</p>
 <p><b>University Affiliation</b> Validate and notarize affiliation with institutions</p>	 <p><b>Statement</b> Notarize a free-form statement</p>
 <p><b>Real Estate</b> Notarize the current real estate value of a specified property</p>	 <p><b>Random Drawing</b> Pick a random number</p>

<http://virtual-notary.org/>

Cornell University  
Computer Science Department

- 存在証明したい文書やWebサイトに対してX.509属性証明書を発行する。
- X.509属性証明書のハッシュチェーンを作り、順序関係を担保する。
- ハッシュチェーンの最新状態をTwitterとビットコインネットワークに流すことで第三者証明を可能にする。
- 24時間ごとにビットコイントランザクションのOutput Script(たぶんOP\_RETURN)にその時の最新ハッシュ値を埋め込んで流す。

本題に入ります

# ビットコインの気になるところ

- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- ブロックチェーンの肥大化
- 暗号アルゴリズムの脆弱化
- etc.

今日触れるのは  
このあたりのお話



# ビットコインを不正に得るには？

- 人様のビットコインを盗む
- 自分のビットコインを不正に増やす  
（二重使用）
- 他者から二重取りする  
（コイン受領の否認）

# ビットコインの気になるところ

- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- ブロックチェーンの肥大化
- 暗号アルゴリズムの脆弱化

# ウォレット盗難の脅威

ウォレット（トランザクションの署名に使う秘密鍵）の管理はとっても大事！

- PC/デバイスでの管理
  - マルウェア感染によるウォレット流出の脅威
  - 暗号（パスワード）はかけられるが、キーロガーを仕掛けられたり、ウォレット取得後にブルートフォースを掛けられたら…
  - ウォレットのバックアップについても注意が必要
    - 盗難ではないが、古いバックアップをリストアしてしない新しい鍵が消失してしまう事故も…
- オンラインサービスでのウォレット管理
  - そのサービスに脆弱性があった場合の秘密鍵流出や不正使用など
  - そもそも、そのサービスは信用できるのか？

# オフライン管理による保護①

- オンラインPC（ネットワーク接続したトランザクション受信専用PC）とオフラインPC（ネットワークに接続しない署名専用PC）の2台に分割して管理する。
- 送金する場合には、オンラインPCで生成したトランザクションデータ（署名なし）に対してオフラインPCで署名を行う。トランザクションデータはUSBキーを用いて移動させる。
- 専用ウォレット管理ソフト ARMORY  
<https://bitcoinarmory.com/>

**ARMORY**  
About Download Developers Support

**ARMORY  
BITCOIN  
WALLET**  
Taking Bitcoin Security and Usability to the Next Level

Cold Storage for your Bitcoins

Available for **DOWNLOAD NOW**

**SIMPLE SOLID SECURE**

An open source Bitcoin wallet management platform.

Designed from the ground up to provide the highest level of security for heavily-invested Bitcoin users, while still maintaining a high degree of usability and convenience. Its ease of use and advanced features make it one of the most popular alternative Bitcoin clients, and is featured on the main [Bitcoin website](#).

# オフライン管理による保護②

**Pi-Wallet**

Your cart  
0 Items

BTC EUR USD



Shop Pi Wallet? How to Build your own Free News Contact us Create account Login

**Offline wallet for savings**

„An offline wallet, also known as cold storage, provides the highest level of security for savings.“

Source: „http://bitcoin.org/en/secure-your-wallet“

**Featured**

 Pi Wallet* 124.95 EUR	 2 Pi Wallet SDHC cards* 84.95 EUR	 1 Pi Wallet replacement SDHC card* 54.95 EUR
--	--	---

\*The specified price is a final price excluding shipping costs. According to §19 UStG we do not display or charge taxes.

- <http://www.pi-wallet.com/>
- 前述のARMORYをRaspberry Piに組み込んだもの
- オフラインウォレットのために別のPCを用意する必要がなくなる。

# ビットコインの気になるところ

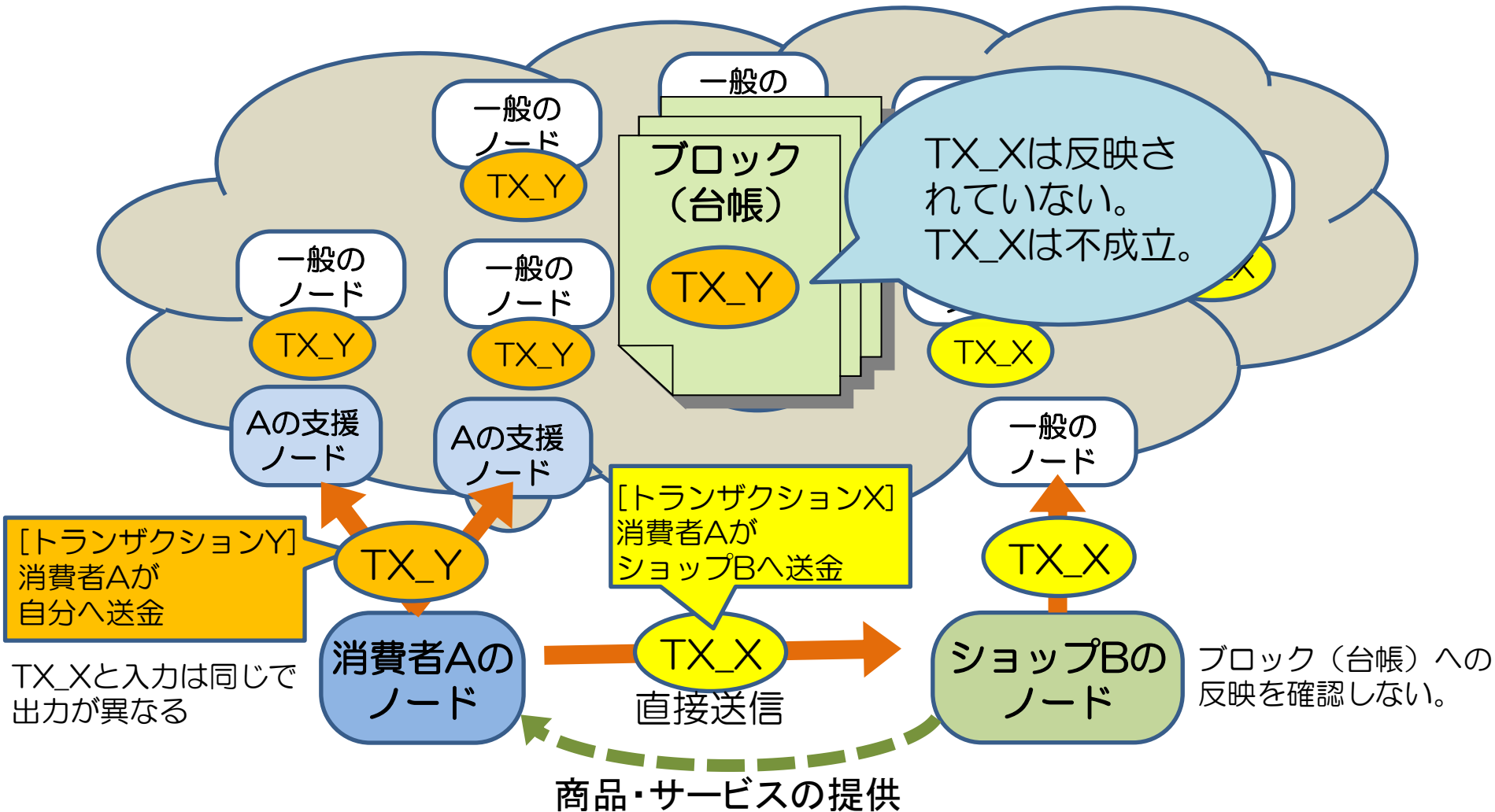
- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- ブロックチェーンの肥大化
- 暗号アルゴリズムの脆弱化

# コインの二重使用

ホントは自分のコインを使っちゃったんだけど、使ったことをみんなに内緒にできれば…

無限に使えるコインができるじゃないか！

# O-confirmation (Fast) payment におけるコイン二重使用



(参考) "Two Bitcoins at the Price of One? Double-Spending Attacks on Fast Payments in Bitcoin",  
Ghassan O. Karame, Elli Androulaki, Srdjan Capkun  
© 2013 SECUM CO., LTD.



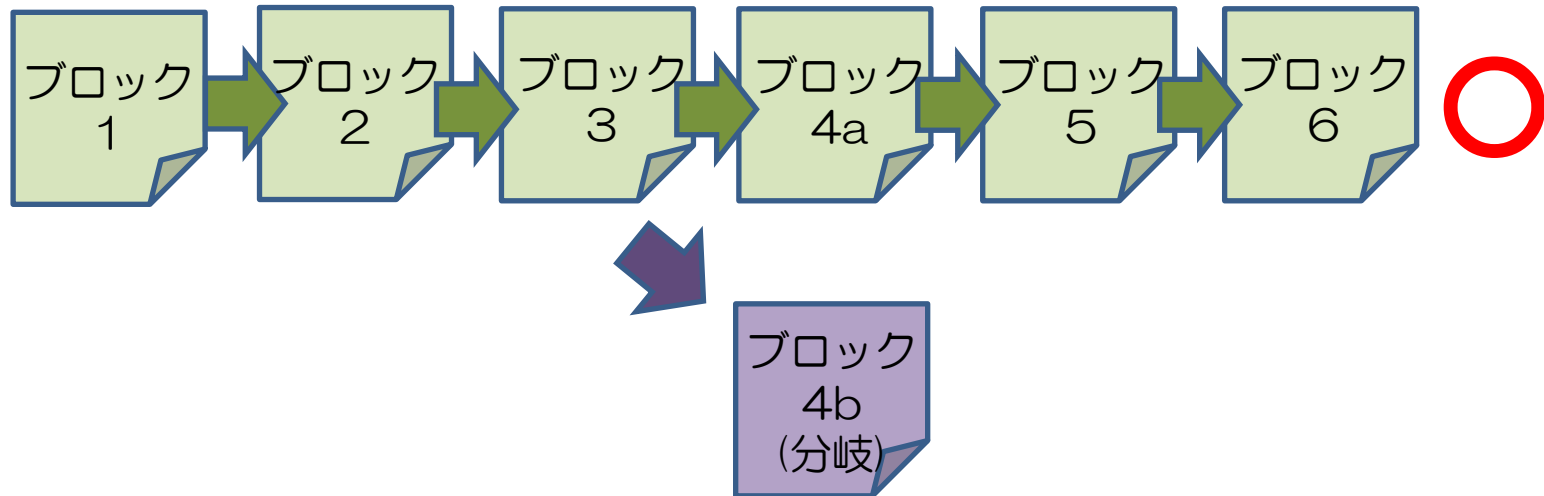


# N-Confirmations

- 0-Confirmationによる取引はリスクを考慮する必要がある。高額取引には推奨されない。
- トランザクション発生後、N個のブロック生成を待った後にトランザクションの確認を行ったほうがよい (N-Confirmations)。
  - 例えば、bitcoinオリジナルクライアントでは6 confirmationsが設定されている。
- N-Confirmationsに関する考察
  - Analysis of hashrate-based double-spending, Meni Rosenfeld, Dec 2012

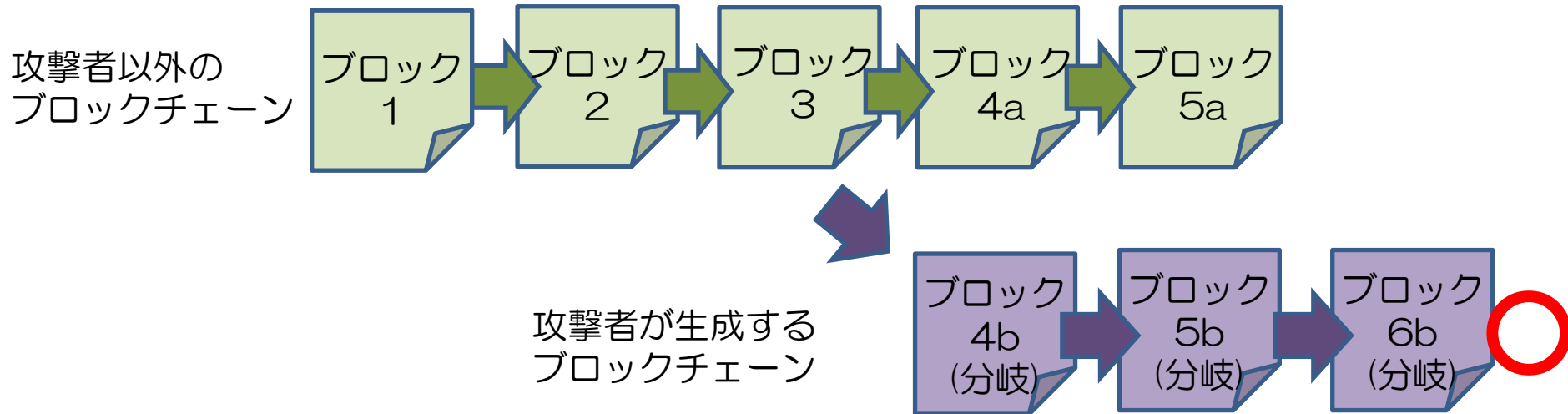
# ブロックチェーンのブランチ

## ブロックチェーン



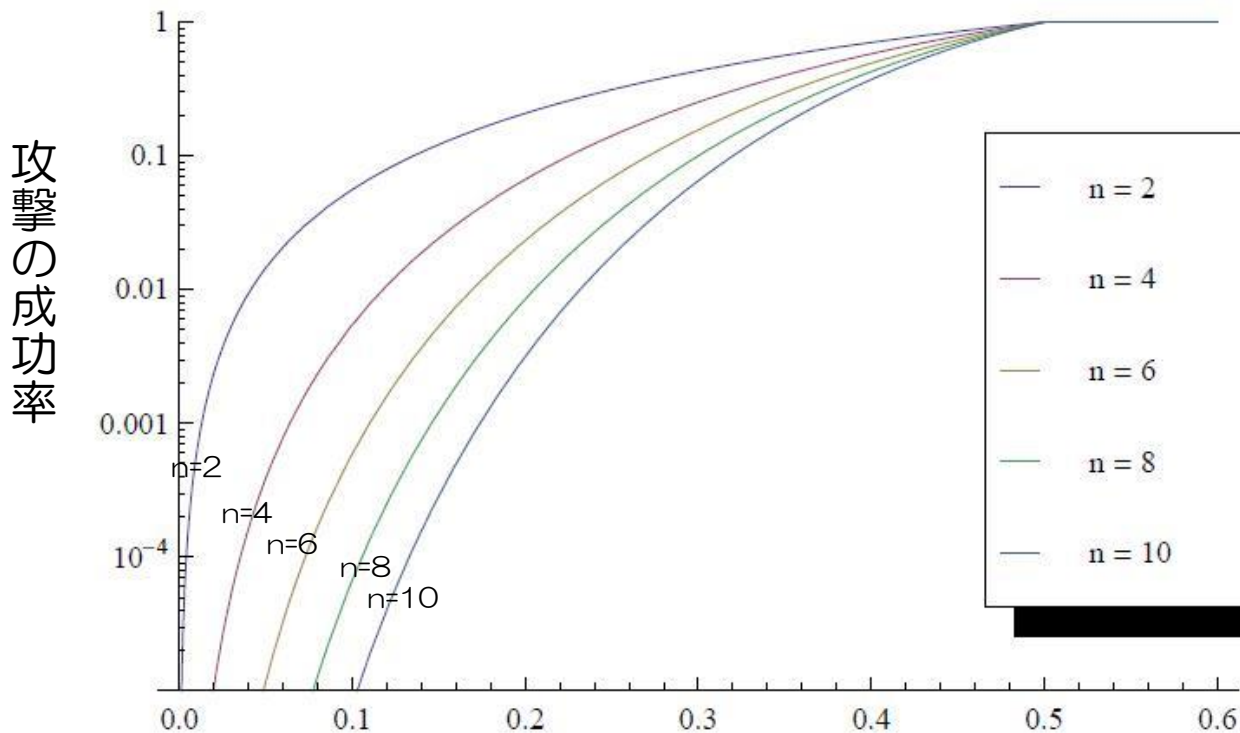
ブロックチェーンが分岐しても、ビットコインネットワークは最長のブロックチェーン（緑のブロック）を受け入れる。

# ブロックチェーンのキャッチアップ



攻撃者が他のノードよりも先にブロック生成を行い、  
最長のブロックチェーンを作り出すことができれば、  
攻撃者にとって都合の良いブロックを認めさせることができる。  
(例えば、自分のコインの2重使用を可能にするトランザクションを含める等)

# ブロックチェーン乗っ取りによる 攻撃の成功率

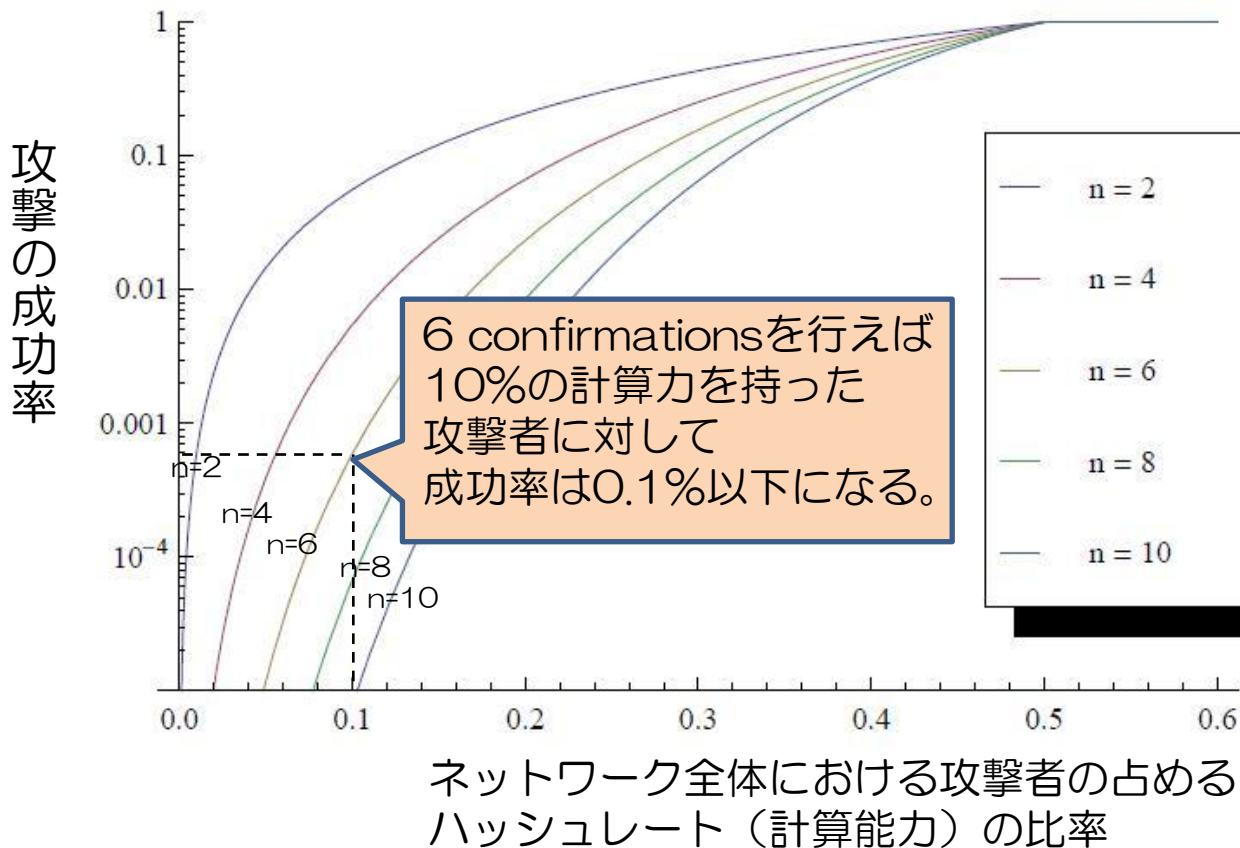


nはコイン取得者が  
実施するconfirmation  
(確認するブロック)の数

ネットワーク全体における攻撃者の占める  
ハッシュレート（計算能力）の比率

(引用元) “Analysis of hashrate-based double-spending”, Meni Rosenfeld

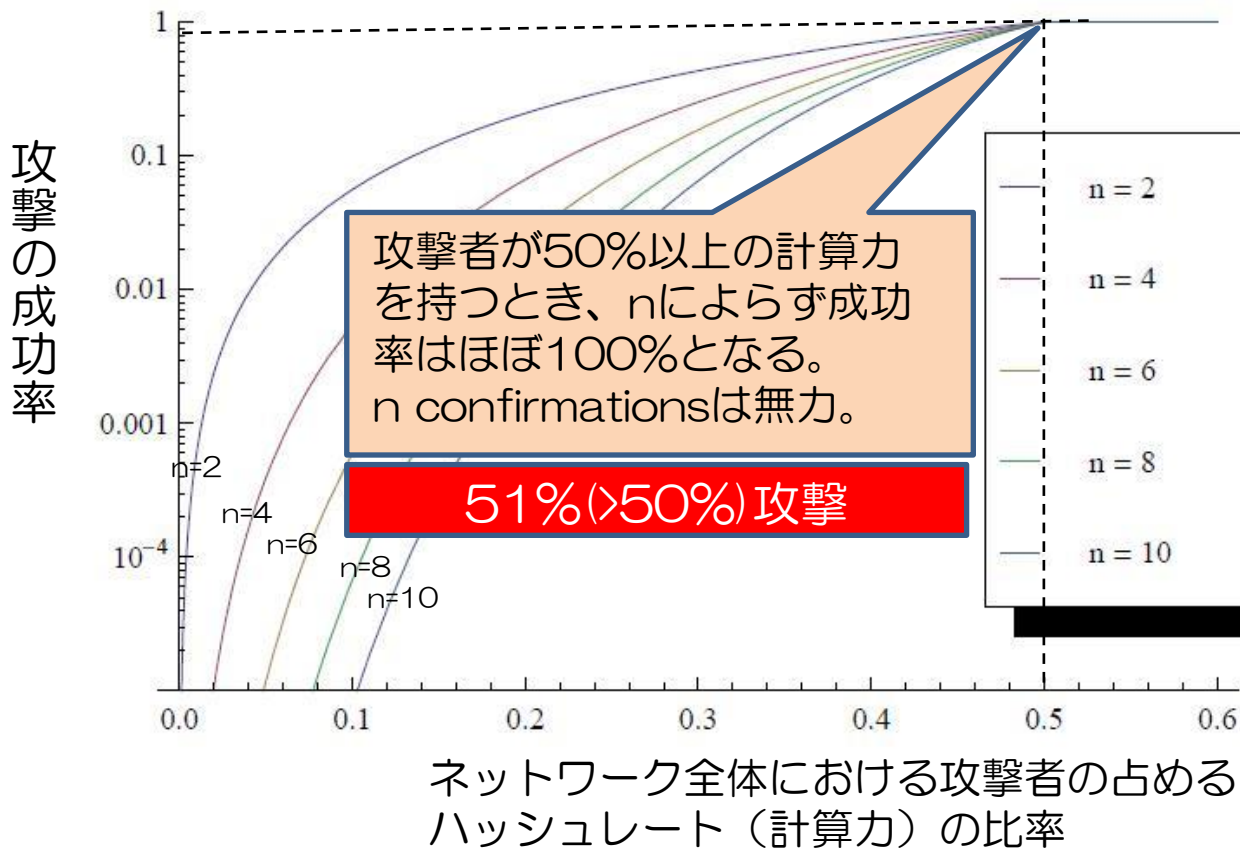
# ブロックチェーン乗っ取りによる コイン二重使用の成功率



nはコイン取得者が  
実施するconfirmation  
(確認するブロック)の数

(引用元) “Analysis of hashrate-based double-spending”, Meni Rosenfeld

# ブロックチェーン乗っ取りによる コイン二重使用の成功率



nはコイン取得者が  
実施するconfirmation  
(確認するブロック)の数

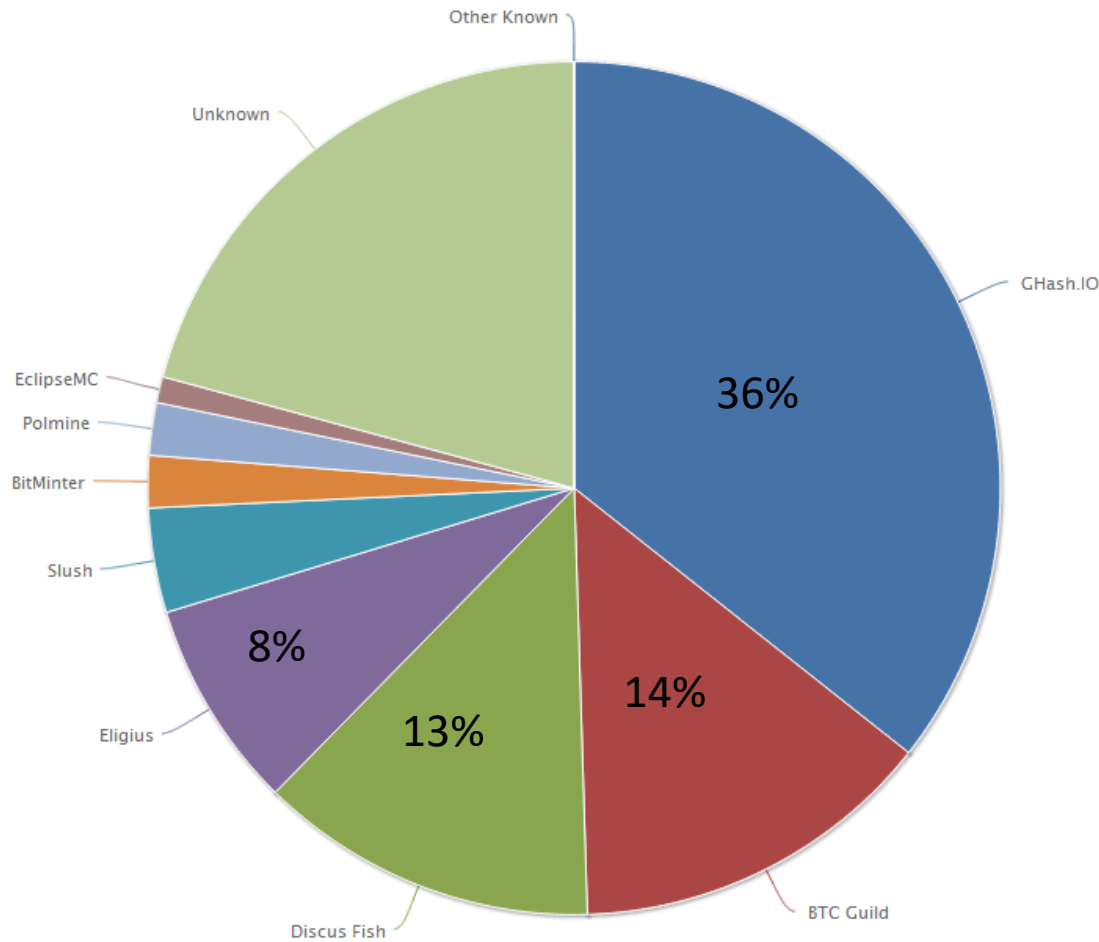
(引用元) “Analysis of hashrate-based double-spending”, Meni Rosenfeld

# N-Confirmations

- 0-Confirmationによる取引はリスクを考慮する必要がある。高額取引には推奨されない。
- トランザクション発生後、N個のブロック生成を待った後にトランザクションの確認を行ったほうがよい（N-Confirmations）。
  - 例えば、bitcoinオリジナルクライアントでは6 confirmationsが設定されている。
  - 51%攻撃が実行されないという前提が必要か
- N-Confirmationsに関する考察
  - Analysis of hashrate-based double-spending, Meni Rosenfeld, Dec 2012



# 現在のハッシュレート (2014.6.1現在)



# 51% (>50%) 攻撃による効果

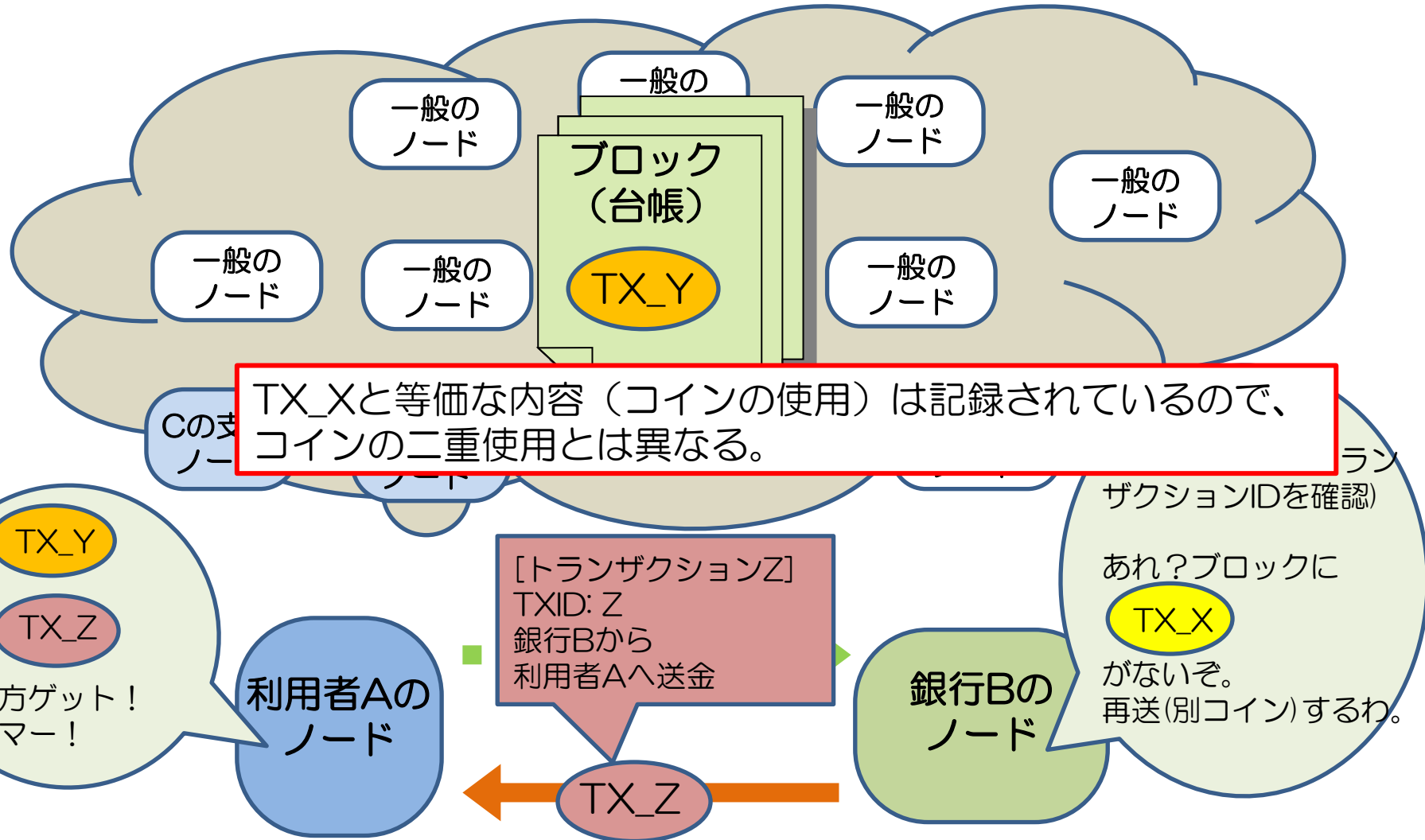
- 二重使用以外にも攻撃者に都合よく仕向けることができる。
  - 他者のトランザクションのconfirmationを妨害できる。
  - 他者のマイニングを妨害できる。

# ビットコインの気になるところ

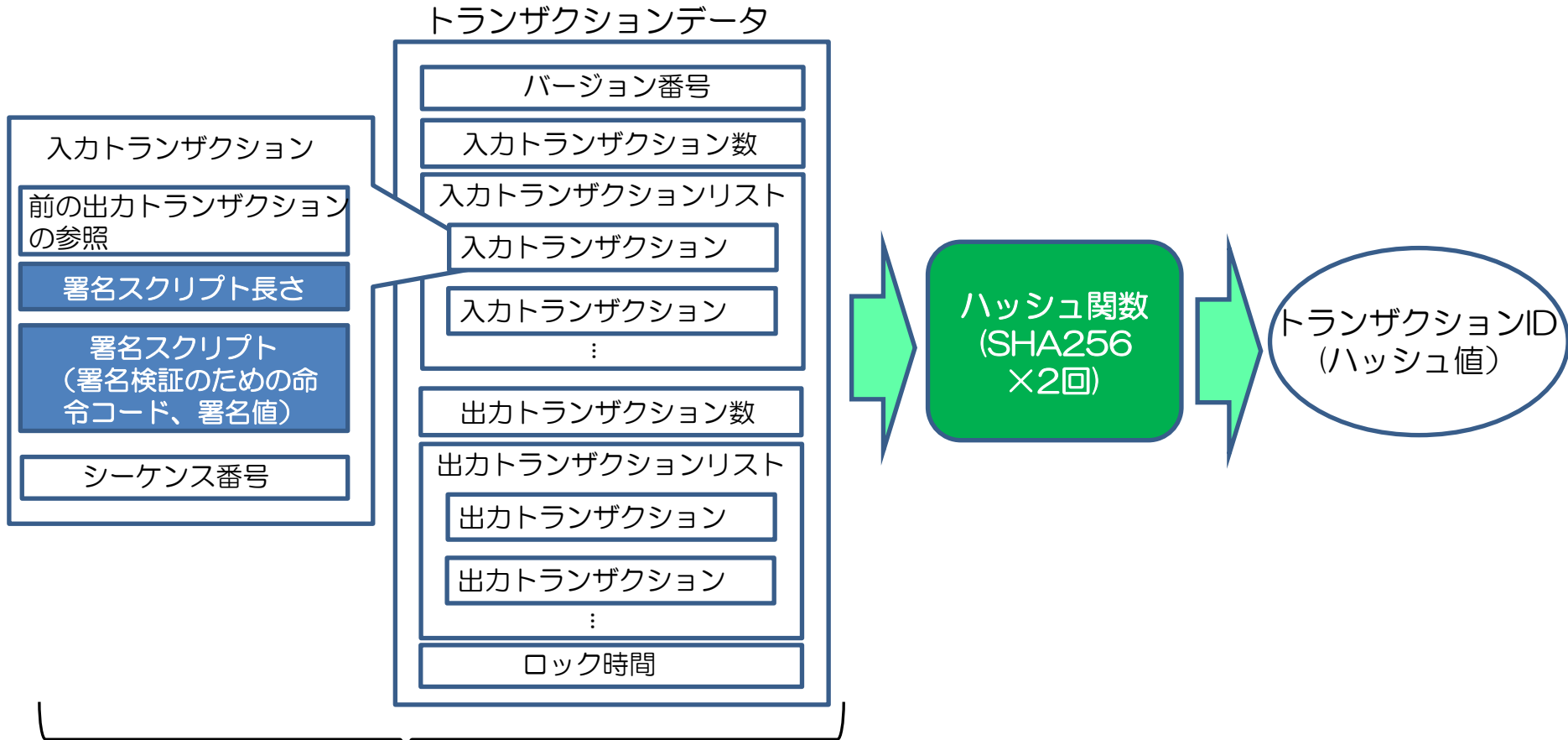
- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- ブロックチェーンの肥大化
- 暗号アルゴリズムの脆弱化



# コインの二重取り



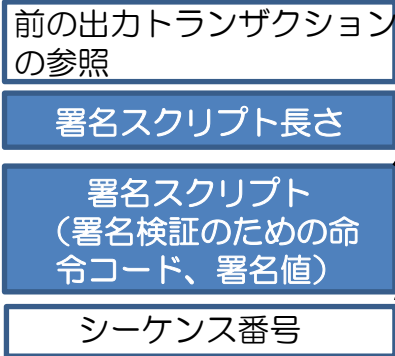
# トランザクションIDに対する不正な操作 ～トランザクション展性(Transaction Malleability)～



■以外の箇所を不正に修正すると署名値が不一致するため、データ改ざんとして検知される。署名値を変えても検知される。署名値の内容を変えずに■の箇所を微調整できれば、全く異なるハッシュ値 (ID) を生成させることができる (ハッシュ関数の性質)。

# トランザクションIDの改変方法例① (署名値の操作)

入カトランザクション



署名スクリプト

操作命令

署名値

操作命令をいじるケース

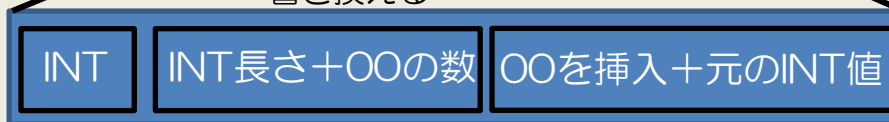
操作命令の結果に影響を与えない命令を挿入する。  
例えば、意味のないDROP命令の挿入など。

署名値をいじるケース

署名値の構造：DER(Distinguished Encoding Rules)による符号化



データの意味を変えずに書き換える

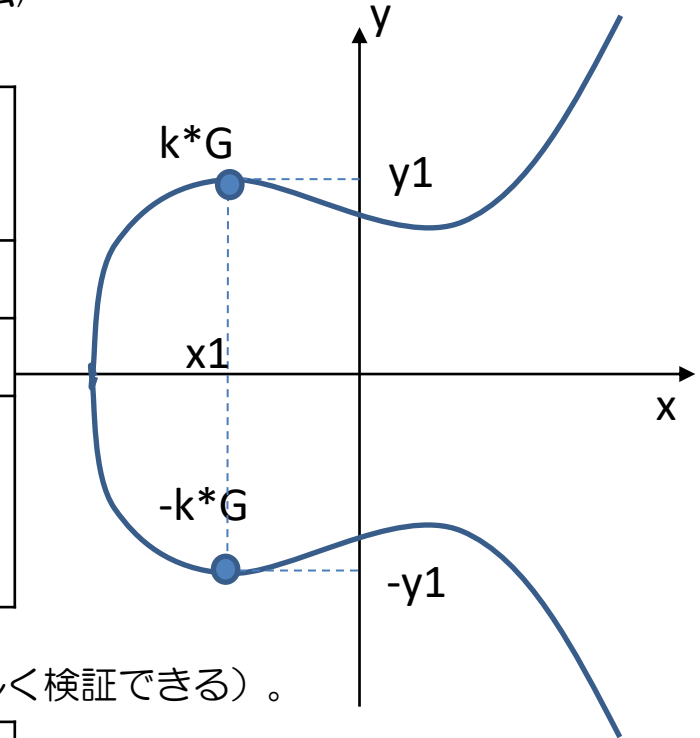


DERでは最短の整数値を用いる。つまり、先頭に00が続くのはNG。  
Bitcoinクライアントが使用していたOpenSSLのバグで00を挿入したデータも受け入れてしまっていたようだ。

# トランザクションIDの改変方法例② (異なる署名値への置換)

ECDSA (楕円曲線暗号を用いた署名アルゴリズム)  
による署名生成

共通パラメータ	<ul style="list-style-type: none"> <li>楕円曲線の関数</li> <li>楕円曲線上の点G</li> <li>Gの位数n</li> </ul>
署名に使う秘密鍵	d
検証に使う公開鍵	$Q=d*G$
署名値 (署名対象データ:M)	点 $k*G=(x_1, y_1)$ ( $k \in [1, n-1]$ ) を決めて以下を計算。 $r = x_1 \bmod n$ $s = (\text{hash}(M) + dr) / k \bmod n$



$-k*G=(x_1, -y_1)$  を使って  
署名値だけ変えられる (これも正しく検証できる)。

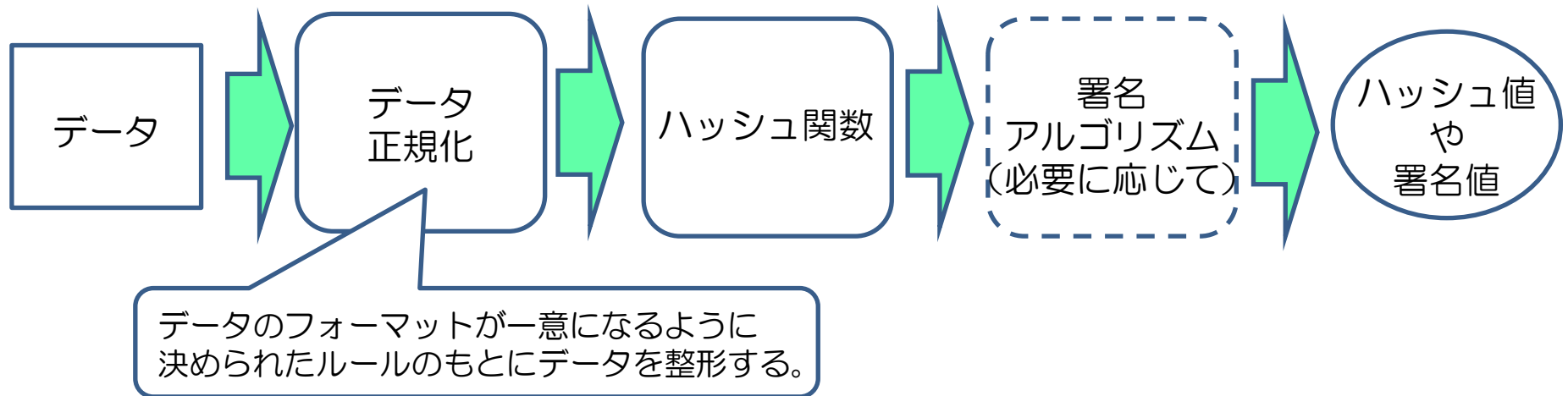
署名値 (署名対象データ:M)	$k' = -k \bmod n$ に置換すれば $r$ はそのまま $s' = -s \bmod n$
--------------------	--

トランザクション中の署名値を  $S \rightarrow S'$  に置換すれば、署名を無効化せずに、トランザクションID(ハッシュ値)を変えることができる。



# データ正規化の必要性

一般的にデータに対してハッシュ関数や署名演算をかけるときには、同じデータに対して出力される値が一意になるようにデータの正規化を行う。  
(電子署名屋さんにとっては常識?)



ビットコインはデータ正規化が甘かったのかも（当初は複数の実装がなかったせい?）。一部のデータ正規化については、オリジナルクライアントは対応済(v0.8, 2013年頃)。その他の正規化は？ オリジナルクライアント以外の実装は???

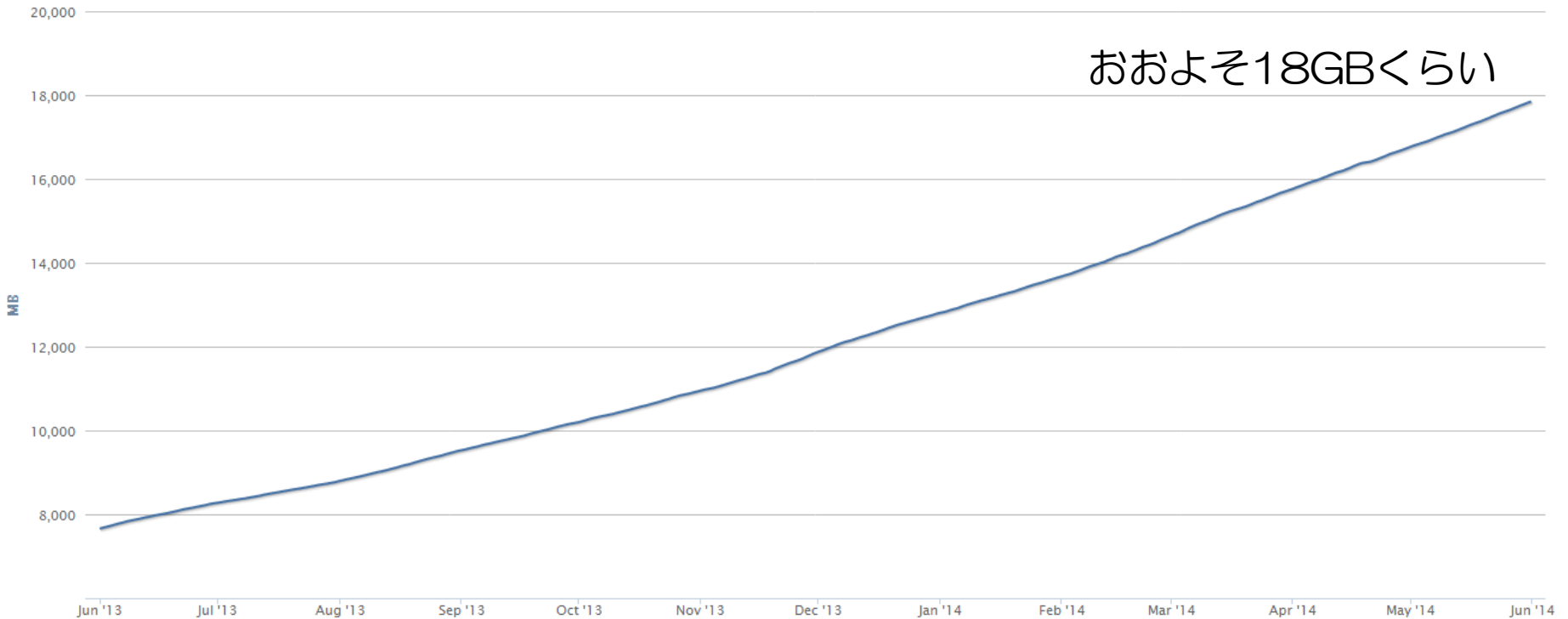
# ビットコインの気になるところ

- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- **ブロックチェーンの肥大化**
- 暗号アルゴリズムの脆弱化

# ブロックチェーンの肥大化

1年間（2013年6月～）

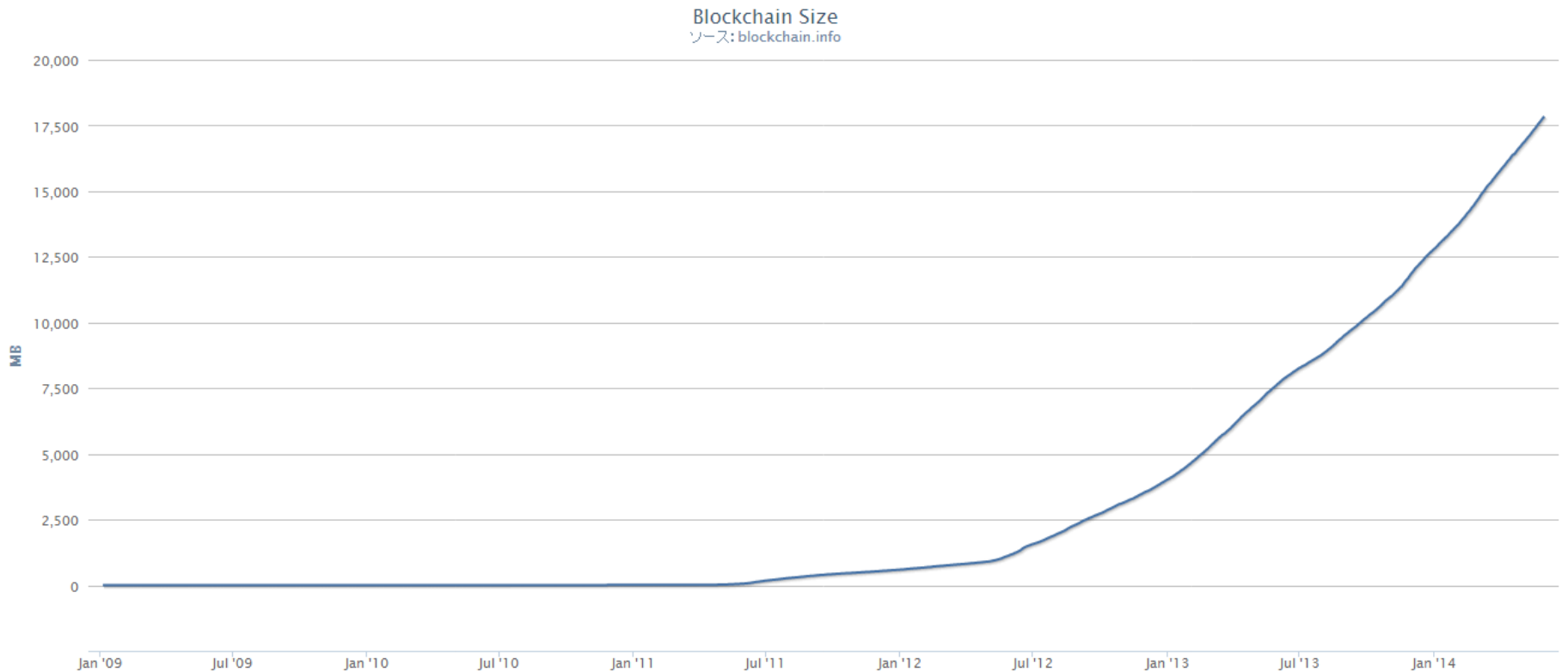
Blockchain Size  
ソース: blockchain.info



現在、全ブロックチェーンをダウンロードしようとすると  
結構な時間がかかります（数日とか…）

# ブロックチェーンの肥大化

全期間（2009年～）



ここ1年くらいの伸びがすごい！

# ブロックチェーンの肥大化への対策

- ビットコインネットワーク上でのブロックチェーンの圧縮が必要？
  - すでに検討されている？
- アプリケーションによる必要なトランザクションデータの抽出
  - 軽量クライアント用に既に拡張されている。
    - Bloom filterを使って自分に関係しそうな（プライバシーへ配慮）トランザクションデータのみをダウンロードする。
    - トランザクションデータを提供するノードへの依存性が高くなる。
    - Multibit, BitCoin Walletでサポート
- 将来、ブロックの蓄積や検証についても分業化が進む可能性がある？

# ビットコインの気になるところ

- ビットコインに対する攻撃
  - ビットコインの盗難
  - ビットコインの二重使用
  - ビットコインの二重取り  
(トランザクション展性)
- ブロックチェーンの肥大化
- 暗号アルゴリズムの脆弱化

# ビットコインの暗号やハッシュ

- 暗号が使われている箇所
  - トランザクションへの署名[公開鍵暗号]
  - 署名鍵（ウォレット）の保護[共通鍵暗号]
- ハッシュ関数が使われている箇所
  - ビットコインアドレス（公開鍵のハッシュ値）
  - トランザクションのハッシュ（ハッシュツリー、トランザクションID）
  - ブロックチェーン生成（マイニング）

# ビットコインの暗号やハッシュ

- 暗号が使われている箇所
  - トランザクションへの署名[公開鍵暗号]
  - 署名鍵（ウォレット）の保護[共通鍵暗号]
- ハッシュ関数が使われている箇所
  - ビットコインアドレス（公開鍵のハッシュ値）
  - トランザクションのハッシュ（ハッシュツリー、トランザクションID）
  - ブロックチェーン生成（マイニング）



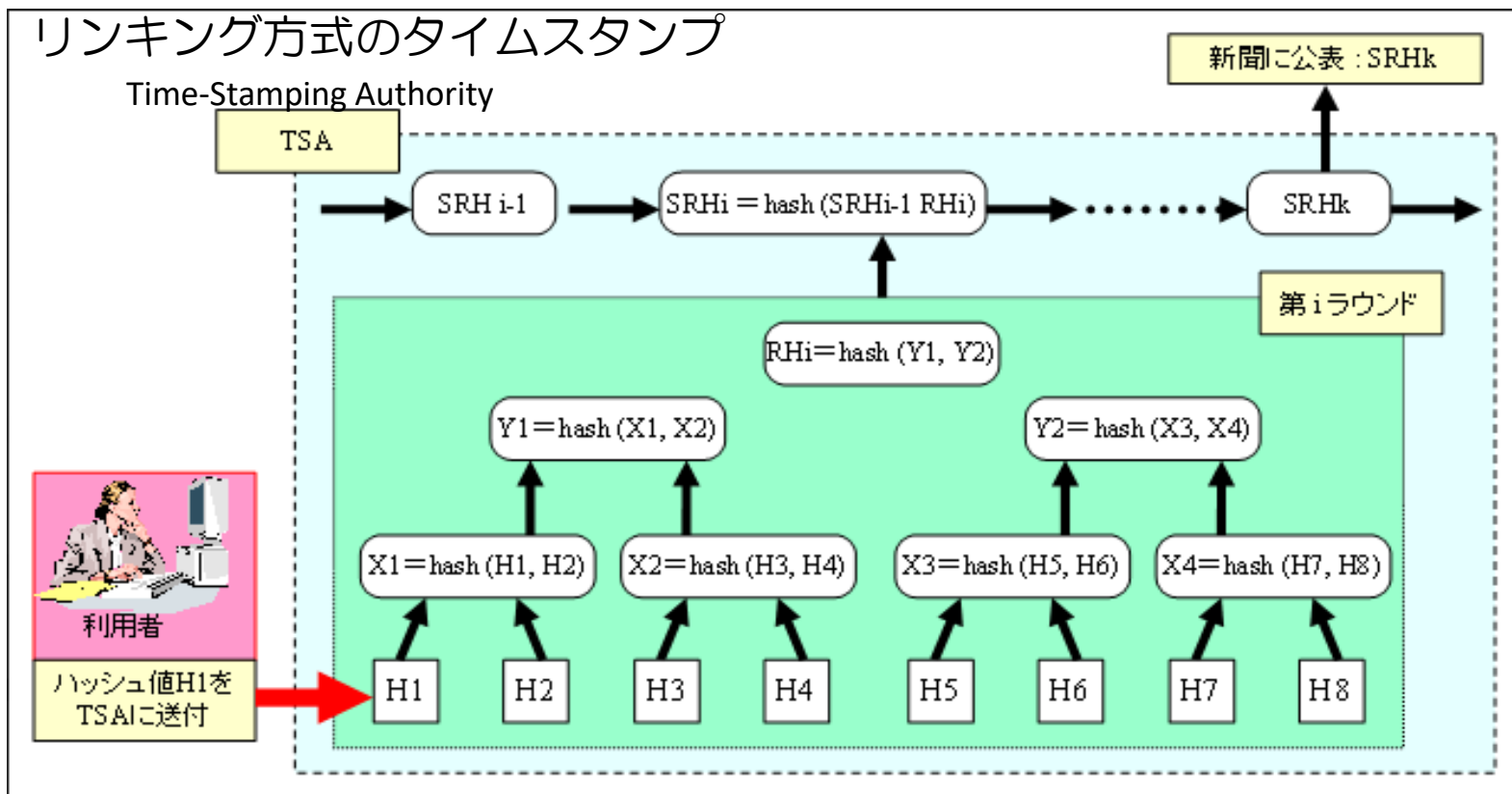
# 暗号やハッシュが脆弱化すると？

- 将来、暗号解読方法の解明や、計算能力の向上、実装ミスの発覚などによって、使用されている暗号やハッシュが弱体化するかも？
- そうなると、次のようなことが可能になるかも？（あくまで可能性の話）。
  - 秘密鍵の所有者以外の者がウォレットを盗むことなく同じ秘密鍵を得る（例えば、公開鍵から秘密鍵を作り出すとか etc.）
  - ブロックチェーン上のあるブロックや、一部のトランザクションを後から不正に入れ替える
- 暗号/ハッシュの弱体化がすぐにビットコインネットワーク崩壊につながるというわけではない。暗号/ハッシュの弱体化の内容とその影響度、他のビットコインネットワークのメカニズムと照らし合わせて、実際にどのようなリスクがあるのか考える必要がある。

## 暗号やハッシュの移行は可能か？

- ビットコインのプロトコルとソフトウェアを更新し、単純に新しい暗号/ハッシュアルゴリズムに入れ替えることで、その更新以降から生成されるトランザクションやブロックはより強力な暗号/ハッシュを利用できるようになるだろう。
- ただし、その場合、過去に作られたトランザクションやブロックは脆弱なままと言える。
  - 例えば、過去に放置したままにしているコインの所有者はどうなるか？

# ハッシュアルゴリズム移行を 考えるときのヒント？



引用元 : <https://www.ipa.go.jp/security/pki/093.html>

この形式のタイムスタンプは過去に作られたハッシュ値を新しいハッシュアルゴリズムで再生成するのは事実上困難。ビットコインのケースにおいても、過去のものをすべて作り直すことは難しいだろう。どこまで対策すべきなのか、どのようなリスクがあるか検討が必要。

# おわりに

- ビットコインは中央の機関に依存しない仕組みであるため、それゆえの難しい問題も生じている。
- ビットコインで生じた数々の問題は、開発コミュニティによって解決が試みられてきている。
  - それらの問題の中には、ビットコインだけでなく他のシステム等を考えるうえで参考になるものもあるかも。
- ビットコインのプロトコルも拡張されているので、決済以外の用途にも使えてくると面白そう。
  - 一方で、他の既存のビジネスへの脅威になる恐れも？
- 電子署名屋さんとしては、秘密鍵の管理や暗号アルゴリズム移行などに興味があるので、今後もう少し深掘りして考えてみたい。

ご清聴ありがとうございました。