

Nostr勉強会 #3

Nostrのしくみを理解するための 暗号技術入門



かすてらふい
@jiftechnify

自己紹介

- NIP-05: jiftechnify@c-stellar.net
- Nostr上での主な活動:
 - 作ったもの
 - Nosaray(のさらい): 過去のTLを見るツール
 - nosdump: 複数リレーからイベントを一気に取得できるCLIツール
 - nostr-fetch: 過去のイベント取得を楽にするライブラリ
 - 詳しい話は勉強会#1をみてね
 - リレー運営
 - wss://nrelay.c-stellar.net: フォロワーのみ書き込み可能
 - wss://nrelay-jp.c-stellar.net: 日本国内からのみアクセス可能
 - NIP日本語訳、たまにプルリク投げたり

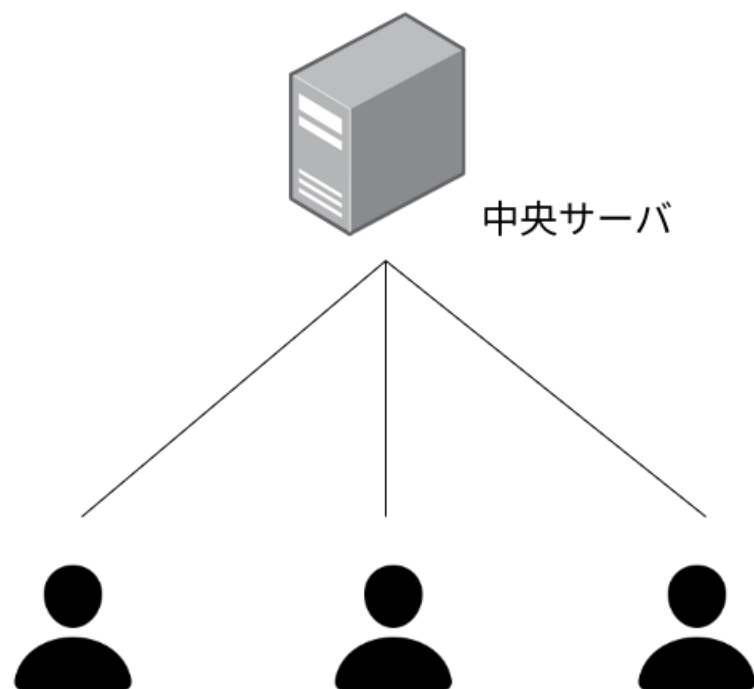
おしながき

- Nostrの基礎となる暗号技術について解説します
 - デジタル署名
 - 共通鍵暗号
- 次のような疑問を解消します
 - Nostrではどうやって投稿者の本人確認・投稿の改ざん防止を実現しているの？
 - どうして秘密鍵を漏らしてはいけないの？ 秘密鍵を漏らすと何が起こるの？
 - NostrのDMはどんなしくみになっているの？

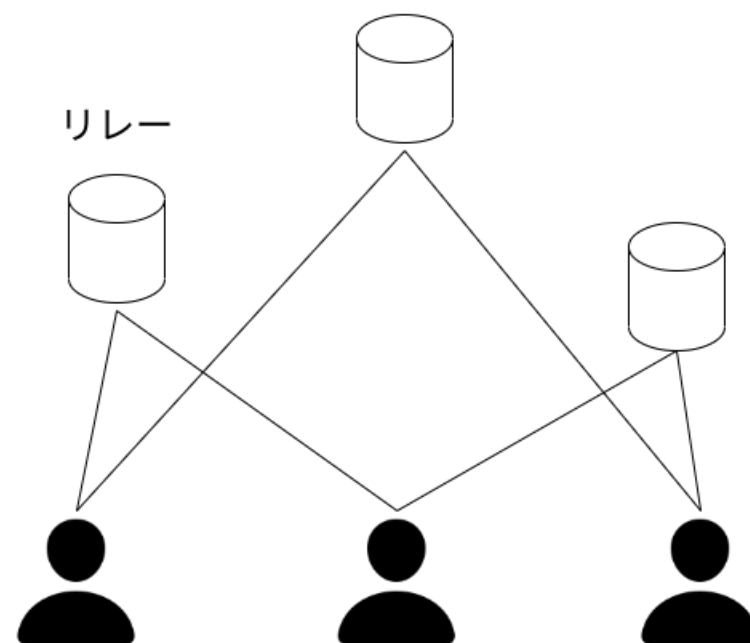
序. 通常のWebサービスとNostrの違い

いつもの図

通常のWebサービス

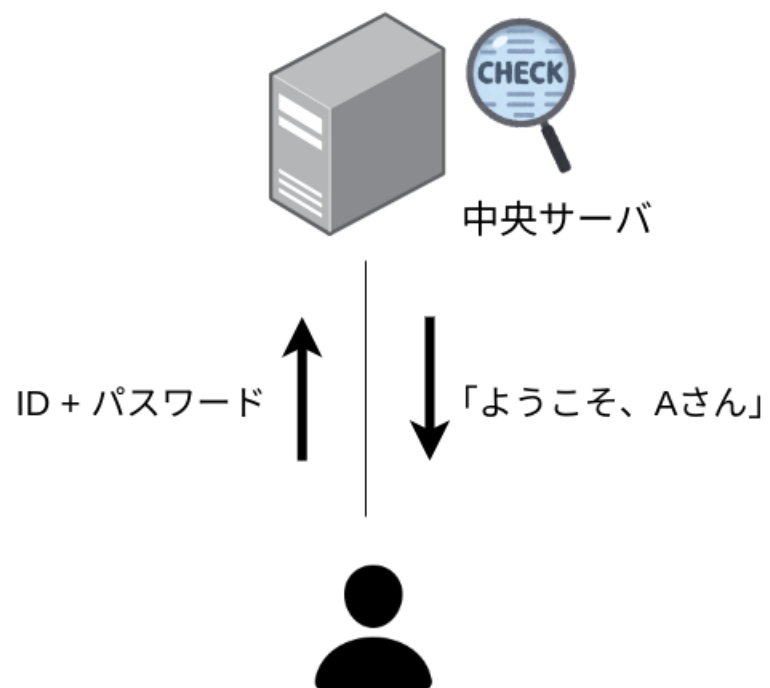


Nostr

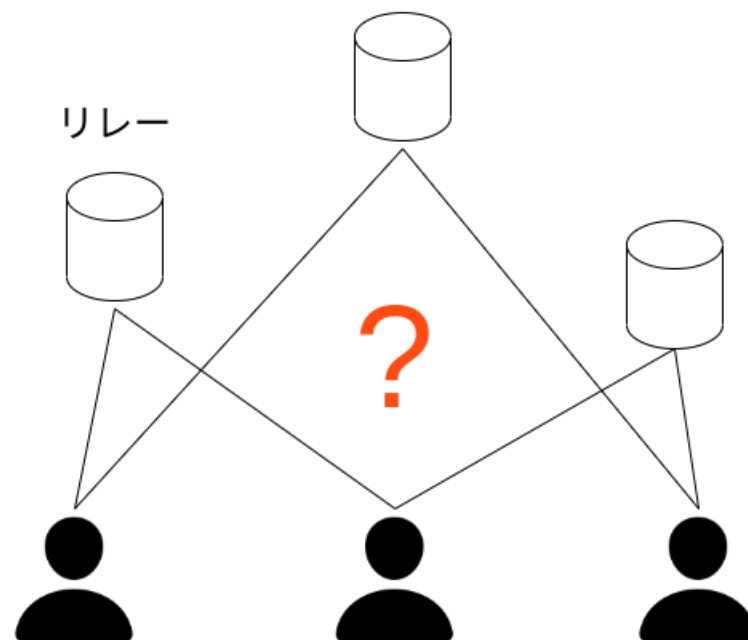


本人確認

通常のWebサービス



Nostr



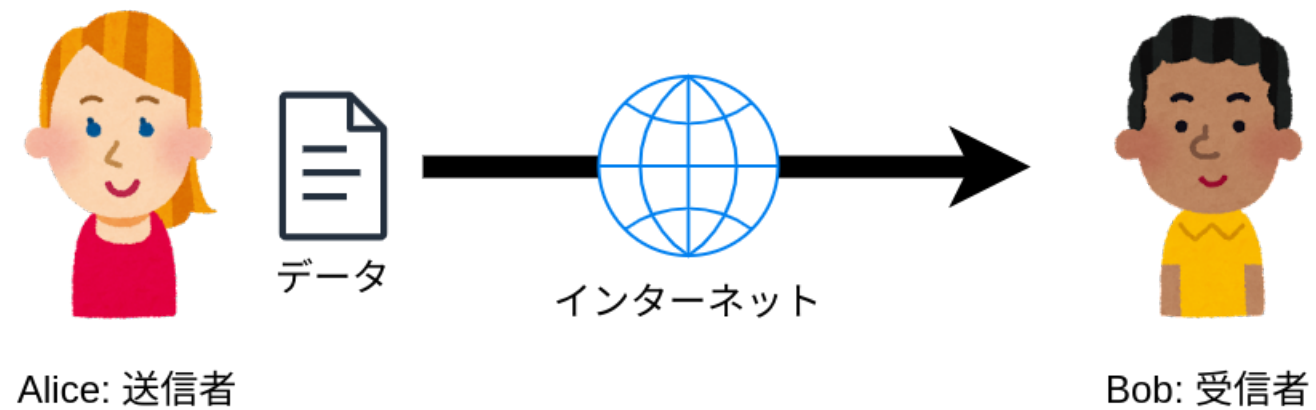
1. デジタル署名の話

アナログ世界の署名

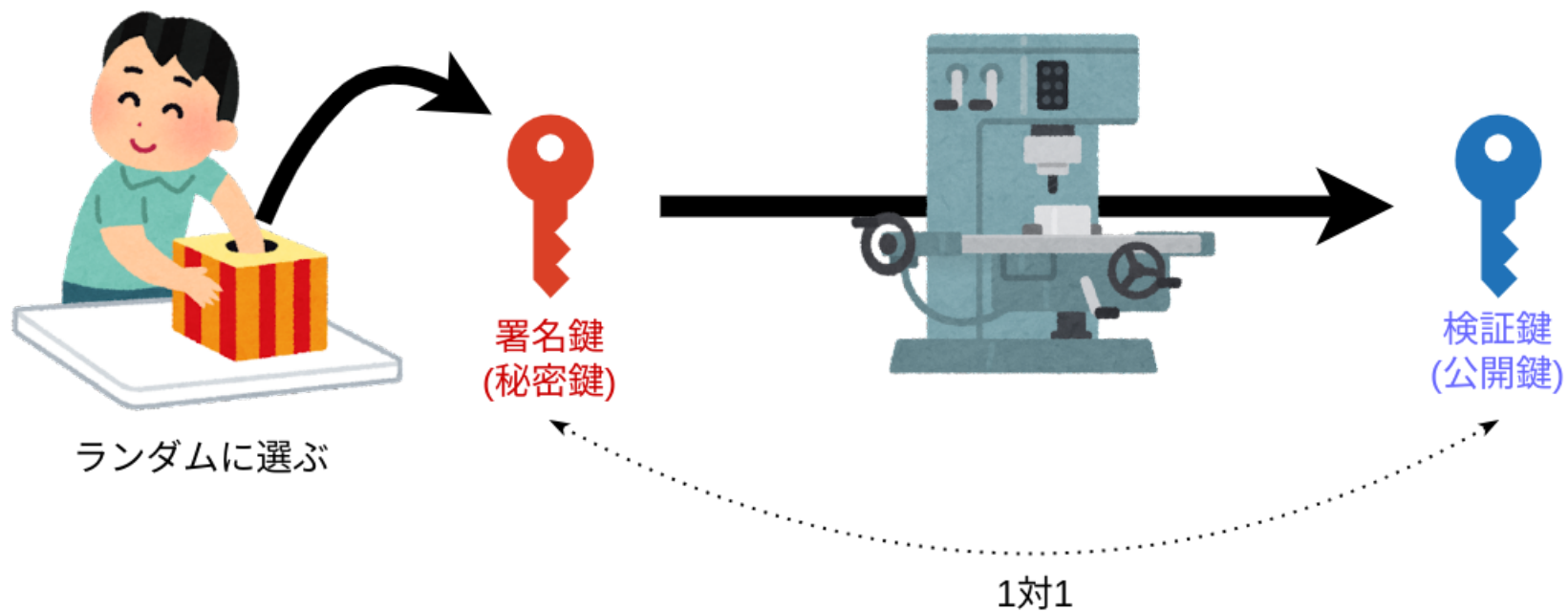


**デジタル署名 =
デジタル世界で
署名と同じしくみを実現するもの**

舞台設定

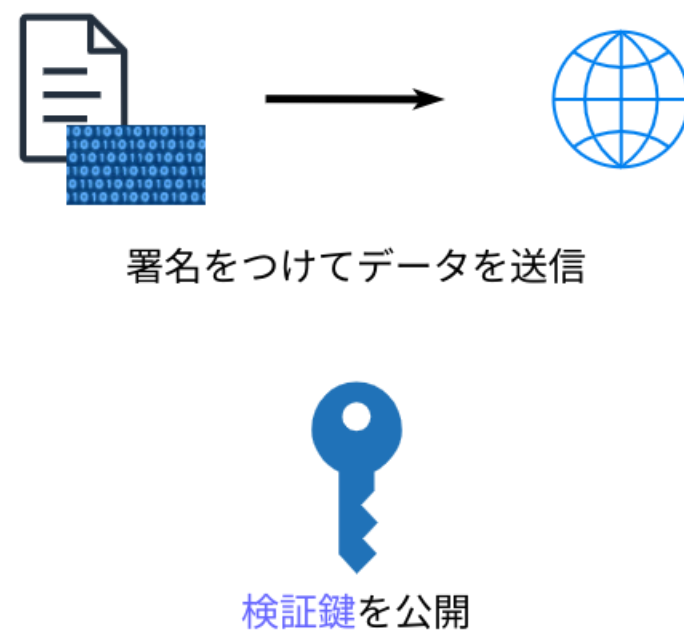
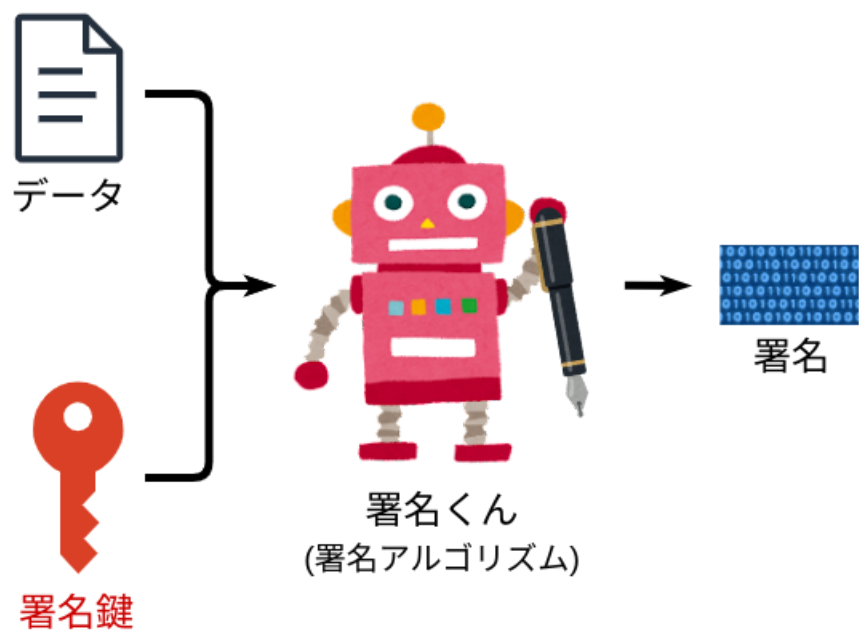


準備: 鍵生成



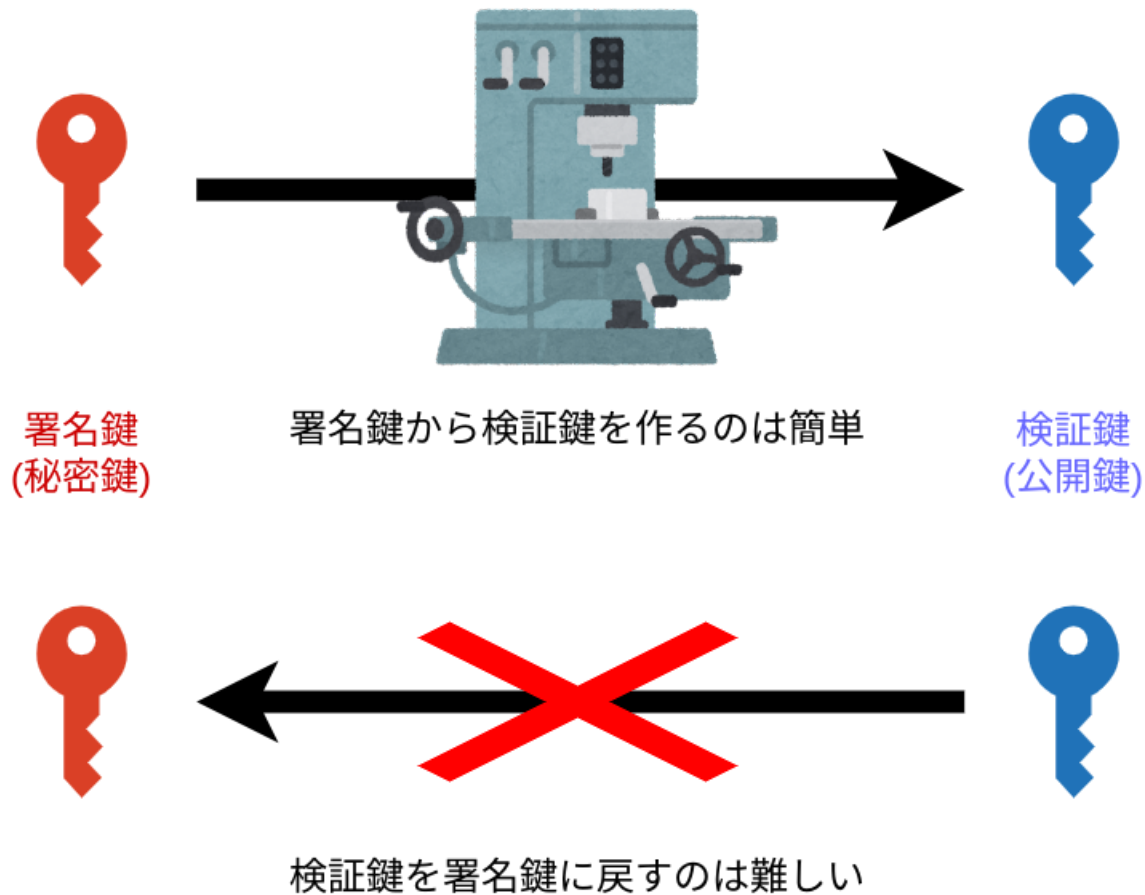
Alice: 送信者

署名

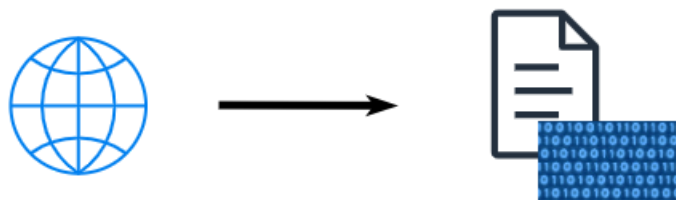


Alice: 送信者

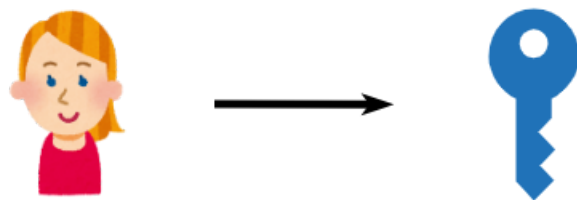
検証鍵を公開しても大丈夫？



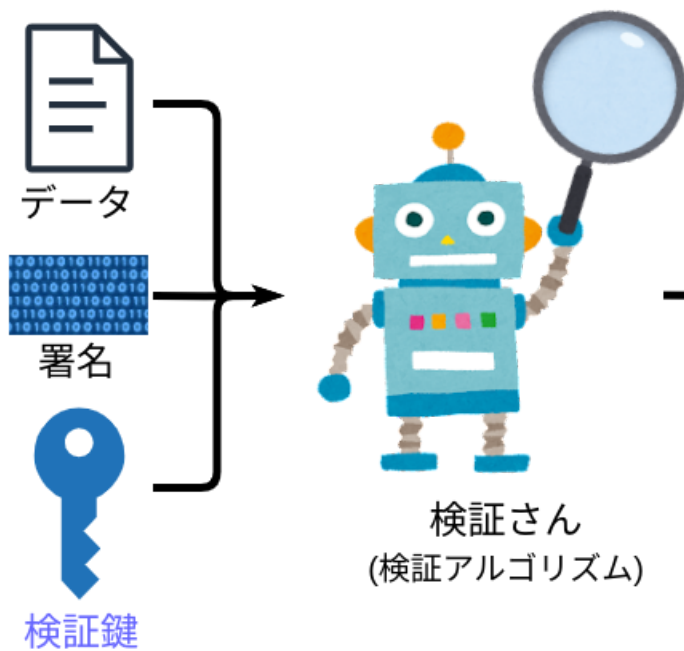
署名検証



署名がついたデータを受信

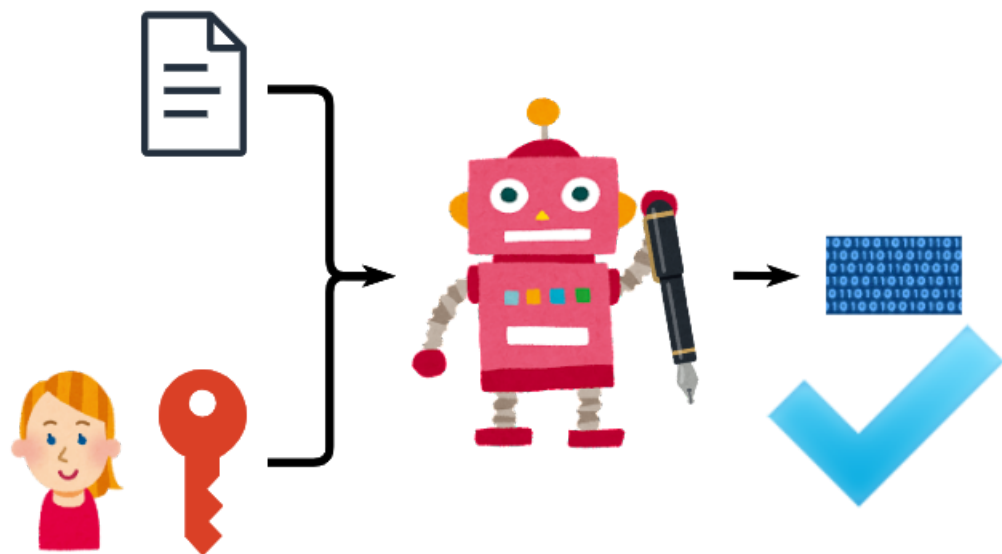


送信者の検証鍵を取得



Bob: 受信者

署名が正しいと何が証明できる？



検証鍵に対応する署名鍵によって
署名されたという事実



データが改ざんされていないこと

閑話休題: ハッシュのお話

元データ

ハッシュ値

Hello, Nostr!

526129966c2517ba9015ac2835cda4e02f1054aec4fb57dfae6ff894b0aae69a

ぽわ

5024d176a91cec3c0b9cd373e909ca7b8b2d50058d7d3968d625c9e43ee1c08c

ぽわ～

3e37bd9e1ad91868e706cd426544f43cec62b123f94e43e1f56687399f14a3cc

ぽわー

7f1de2eb3a207a1186fe3abe9f3539a39b54d92149fe6c90f4b1aa46d7f15f96

寿限無、寿限無、五劫の擦り切れ、...
グーリンダイのポンポコピーの
ポンポコナーの長久命の長助

8a7ea291ace12c21f06d3938d0161b8fe757cca49a045fea523676e95517de78

ハッシュマン
(ハッシュ関数)

- 元データの大きさによらず固定サイズ
- 元データが同じなら、必ず同じ値
- 元データが少しでも違えば、(事実上)必ず違う値

2. Nostrにおけるデジタル署名の利用

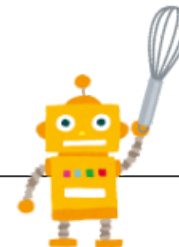
Nostrのイベントの中身


```
{
  "id": "651c4c03b7c34fd4f5d79e464b9488f41bfa35df3b520d81c9e034fd1a35116d",
  "pubkey": "d1d1747115d16751a97c239f46ec1703292c3b7e9988b9ebdd4ec4705b15ed44",
  "created_at": 1691124796,
  "kind": 1,
  "tags": [],
  "content": "ぽわ～",
  "sig": "295fce5b5bf2a893b2b61814bc4264c0ba17fab3f4f7f3d6231f96ca5ac21339886d6075d3a907ce8636fee0dd8c5fdf5f76e2b6193b66496c2ab6bc36ed0ec"
}
```

解剖: Nostrイベント

```
{  
  "id": "651c4c03b7c34fd4f...",  
  "pubkey": "d1d1747115d16...",  
  "created_at": 1691124796,  
  "kind": 1,  
  "content": "ぽわ～",  
  "tags": [],  
  "sig": "295fce5b5bf2a893..."  
}
```

=

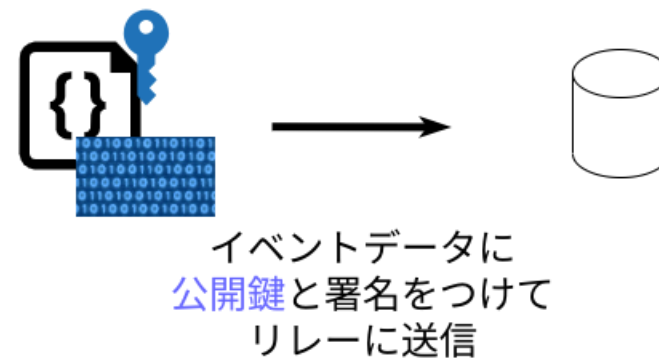
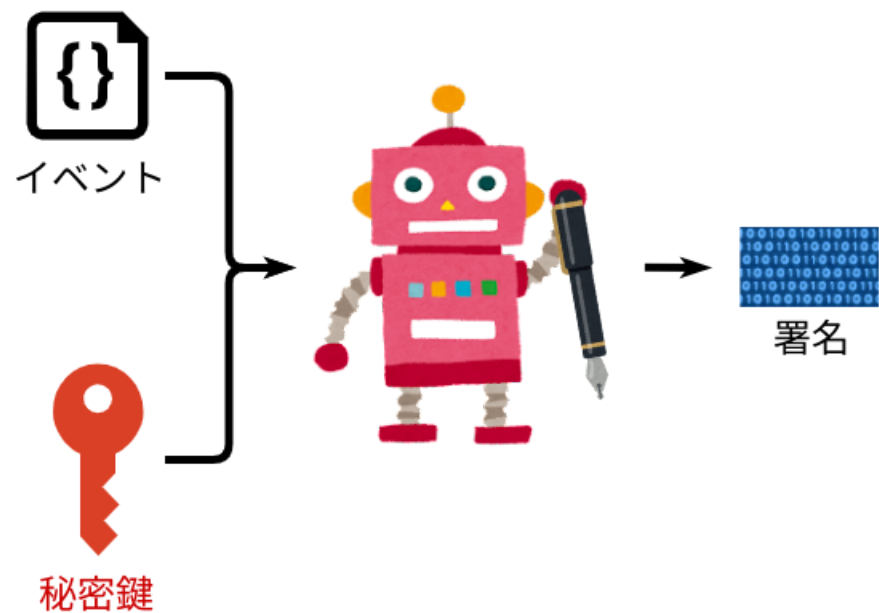


ID: (内容のハッシュ値)
発行者の公開鍵(検証鍵): 
発行時刻: 2023/8/4 13:53:16
種類: 通常の投稿
内容: 「ぽわ～」
タグ: (なし)

署名: 

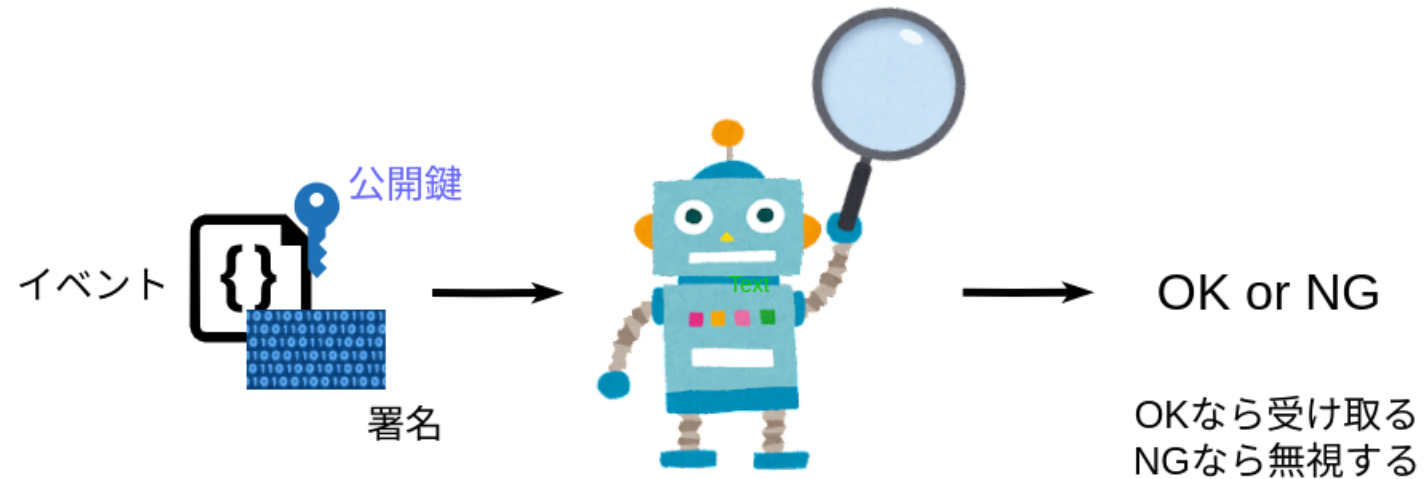
pubkey = public key = 公開鍵
sig = signature = 署名

イベントに署名する



Alice: 送信者

イベントの署名を検証する

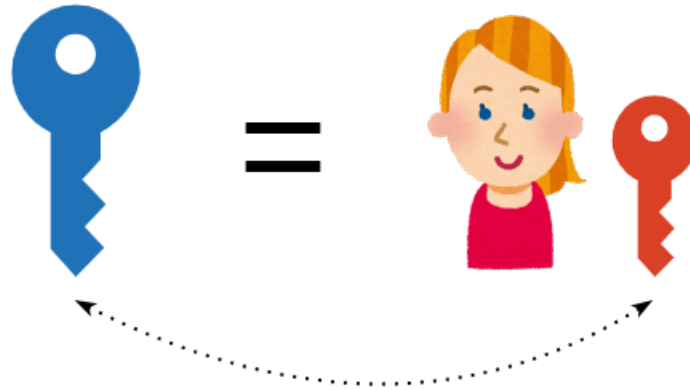


公開鍵 = アカウント

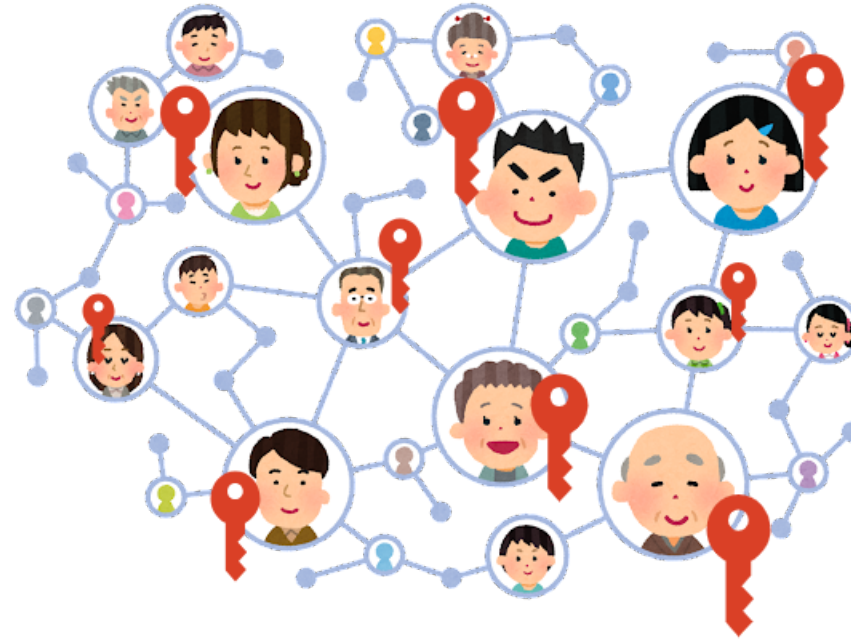
(秘密鍵が漏れていない限り、)

特定の公開鍵のもとで正しい署名を持つイベントを作れるのは、1人だけのはず

→ 公開鍵を、その人の「アカウントID」とみなせる



秘密鍵を漏らすと…



全員がAliceの名前で投稿できてしまう！

本当に本人証明になってるの？

以下のことを証明する、究極的な方法はない！

- ある公開鍵が本当に「〇〇さん」のものなのか？
- ある公開鍵が信用に足る人間のものなのか？

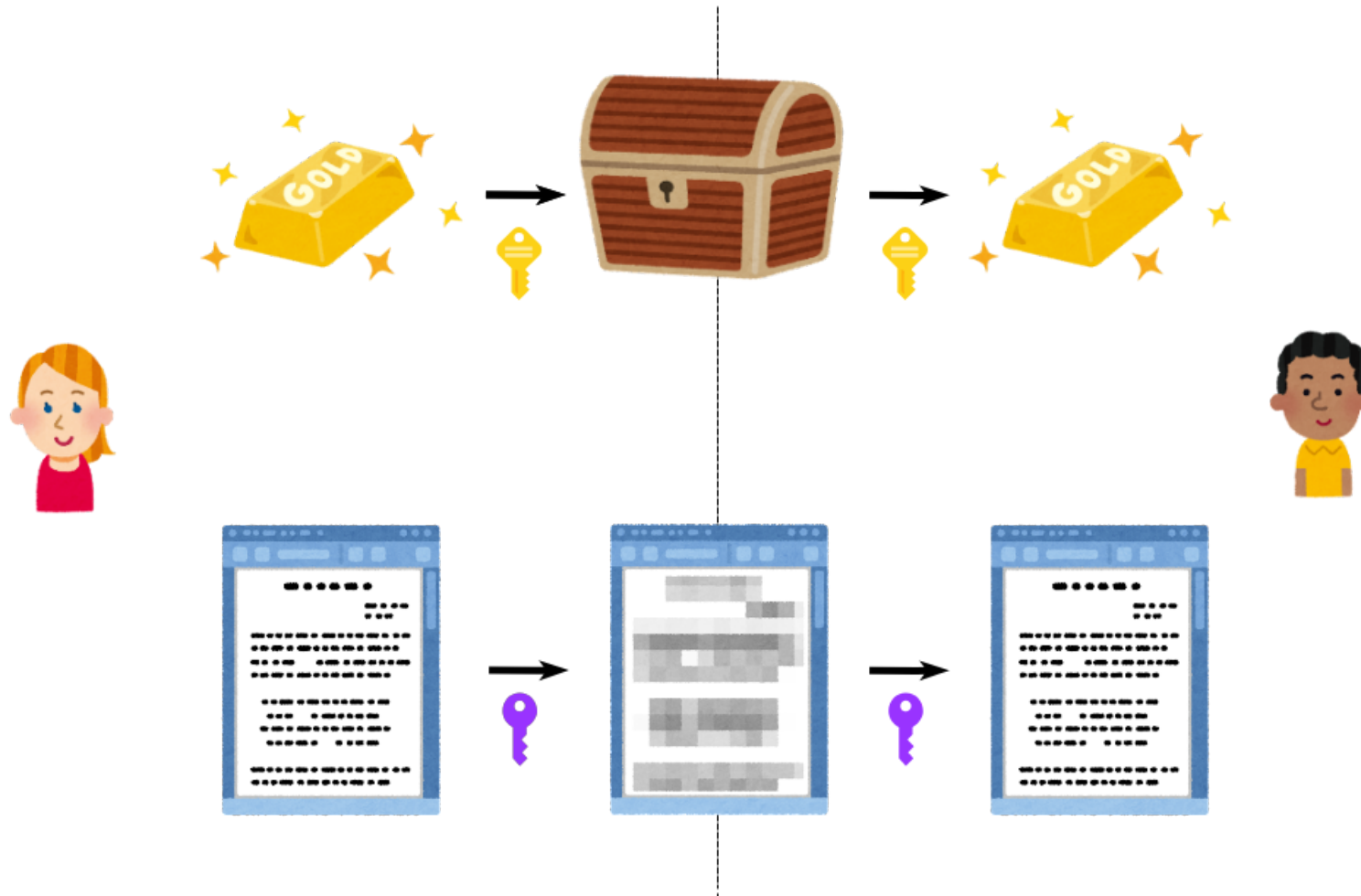
部分的に解決するしくみはあるが…

- NIP-05: ドメイン(例: c-stellar.net)に公開鍵を紐付ける
- NIP-39: 他のSNSアカウントに公開鍵を紐付ける

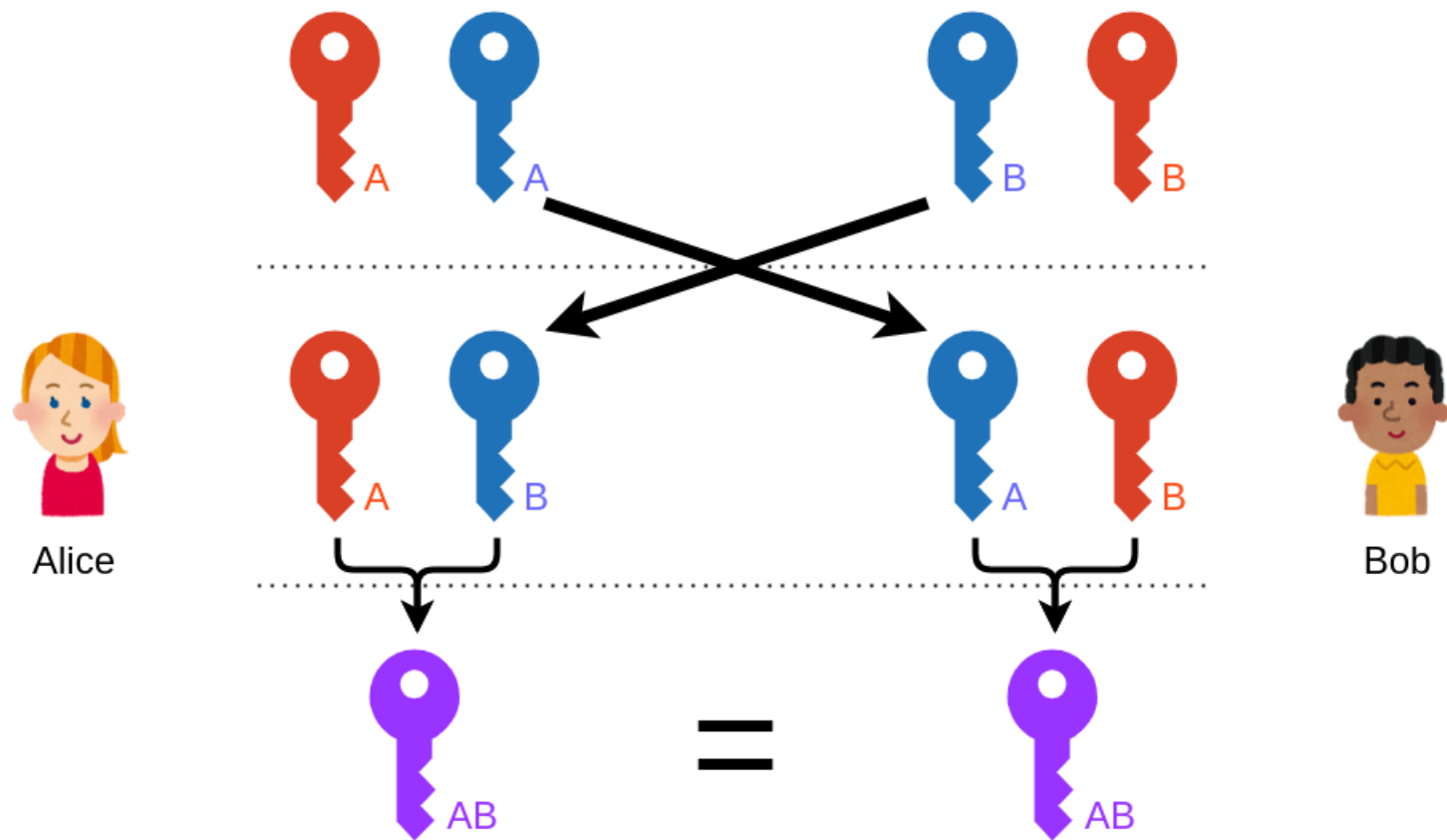
結局、最後に信用を生むのは「日々の行動の積み重ね」

3. NostrのDMのしくみ

いかにして秘密の通信を成立させるか

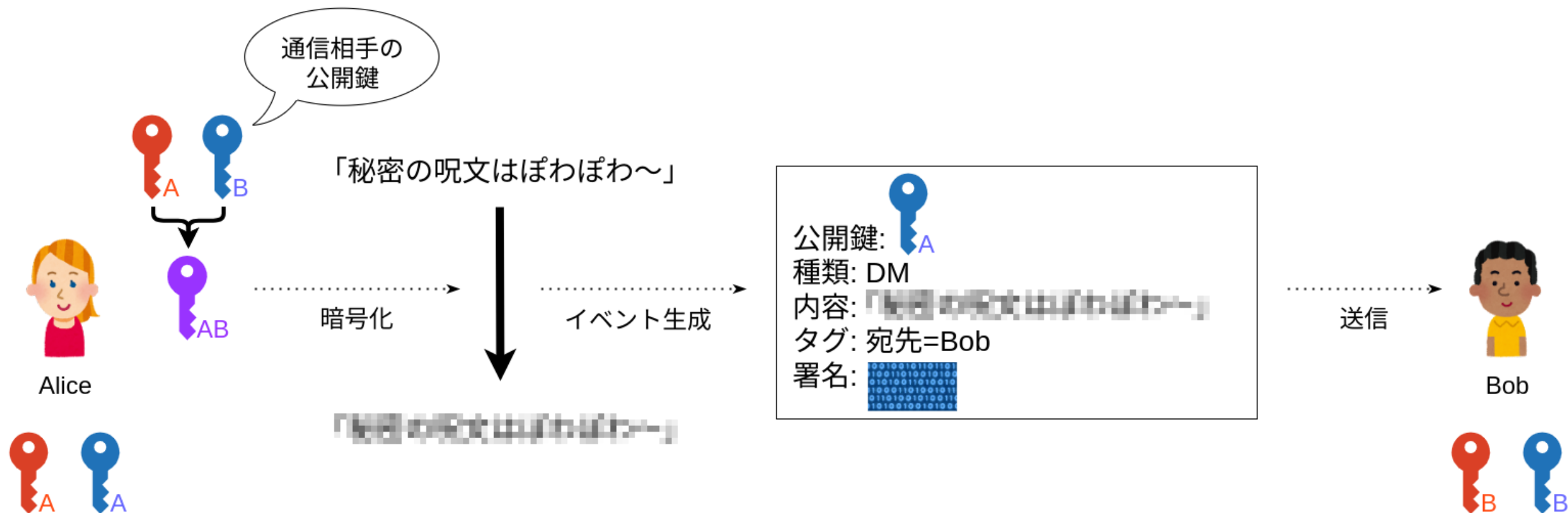


秘密を漏らさず鍵を共有する、たった一つの冴えた方法

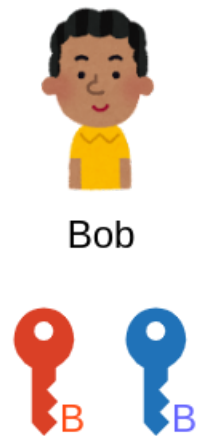



お互いが相手の公開鍵をもらって、自分の秘密鍵と合体させると...
両者の手元で同じ鍵(共通鍵)ができあがる！


DMを送信する: 暗号化

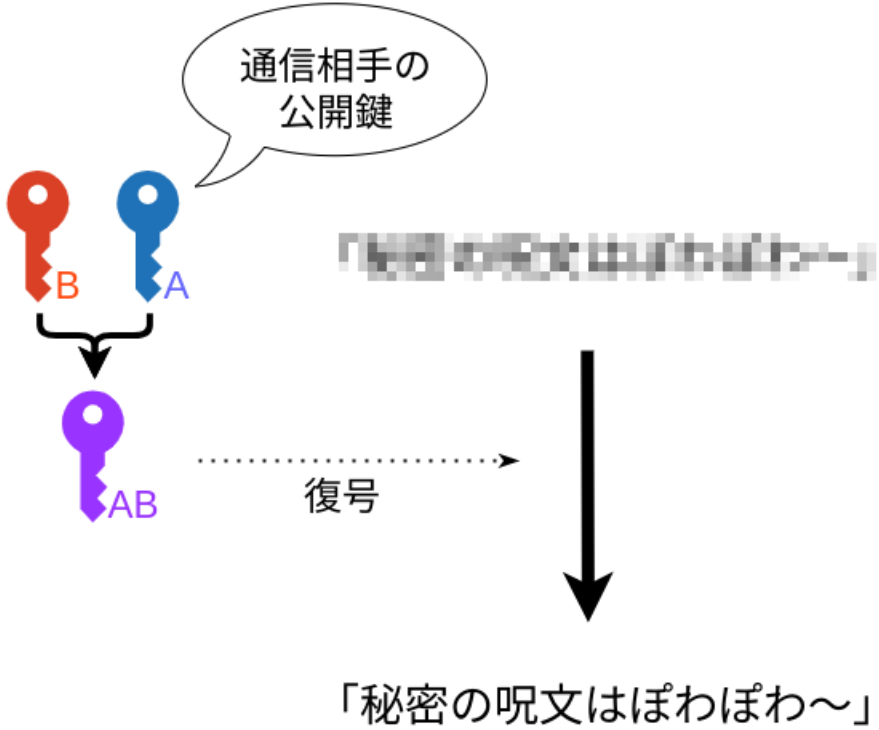
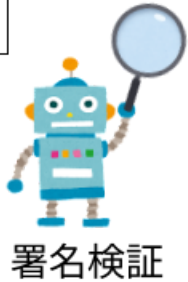


受信したDMを解読する: 復号





公開鍵: A
種類: DM
内容: 「秘密の呪文はぽわぽわ〜」
タグ: 宛先=Bob
署名: 



現在のDMの仕様(NIP-04)の問題点

- 誰が誰にDMを送ったか・いつ送られたかなど、本文以外の情報が外部に筒抜け
- 暗号化方式が固定
 - 暗号化方式が古くなるにつれ、攻撃者に解読される危険性は高まっていく
 - 時代の変化に合わせ、新方式に乗り換えられるしくみにしておく必要があるが…
- 改善のため、さまざまな提案がなされている

参考文献

- 暗号技術のすべて / IPUSIRON 著
- プログラミング・ビットコイン / Jimmy Song 著

