次世代Web認証「パスキー」

Monicle ZATSUDAN LT

Yukiya Nakagawa a.k.a Nkzn / 2024.4.5

注意

- この資料は、社内のLTイベントで技術職以外のメンバー向けにお話ししたものです
- ざっくり雰囲気を感じてもらうために抽象度高めに作っています
- 抽象化する過程で正しくない表現になってしまっている部分も多々あるので、エンジニアが他人にパスキーを説明する時の根拠にこの資料を使うべきではありません

エンジニアの人はこっち読んでね

- https://goo.gle/passkeys
- https://fidoalliance.org/specifications/
- https://fidoalliance.org/fido2-2/fido2-web-authentication-webauthn/
- https://webauthn.io/
- https://blog.agektmr.com/2019/03/fido-webauthn
- https://blog.agektmr.com/2022/12/passkey
- https://blog.agektmr.com/2023/12/passkey-mythbusting
- https://firebase.uservoice.com/forums/948424-general/suggestions/46647016-support-authentication-with-passkeys
- https://moneyforward-dev.jp/entry/2023/04/05/134721

パスワード認証はなぜ動くのか

アプリ側

①パスワードは
AbCdEfG です

ユーザーID: taro

②ログインしたいです

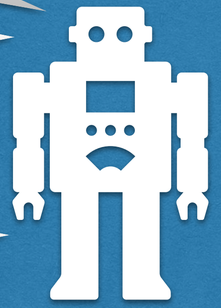
⑥よし、通れ

サーバー側

③ データベースを
チェックするね

④ 送ってきたパスワードと
データベースのパスワードが
一致したよ

⑤ なるほど
君はtaroくんだ



データベース

ユーザーID	パスワード	閲覧権限
taro	AbCdEfG	あり

パスワード認証の問題点

Case 1: 傍受

http:// なWebサイトや
パスワードなしWi-Fiなどによる
通信の盗聴
（httpsやパスワード付きWi-Fiで対策可能）

ブラウザ側

②ログインしたいです

①パスワードは
AbCdEfG です

ウィルスによる
キーボード入力の盗聴
（パスワードマネージャーで軽減可能）

パスワードを付箋で貼ったまま
スタバで仕事した
（どうにもならない）

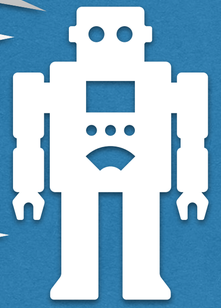
⑥よし、通れ

サーバー側
https://kaneiro.jp

③ データベースを
チェックするね

④ 送ってきたパスワードと
データベースのパスワードが
一致したよ

⑤ なるほど
君はtaroくんだ



データベース

ユーザーID	パスワード	閲覧権限
taro	AbCdEfG	あり

Case 2: 漏洩

ブラウザ側

①パスワードは
AbCdEfG です

ユーザーID: taro

②ログインしたいです

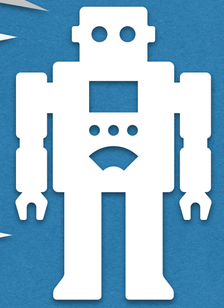
⑥よし、通れ

セキュリティ事故による
パスワードの流出
（ハッシュ化という難読化手法によって一定の対処は可能）
（でも難読化が甘いと80%くらいは復元できちゃうことも）

サーバー側
https://kaneiro.jp

③ データベースを
チェックするね

④ 送ってきたパスワードと
データベースのパスワードが
一致したよ



⑤ なるほど
君はtaroくんだ

データベース

ユーザーID	パスワード	閲覧権限
taro	AbCdEfG	あり

Case 3: 詐称

ブラウザ側

①パスワードは
AbCdEfG です

ユーザーID: taro

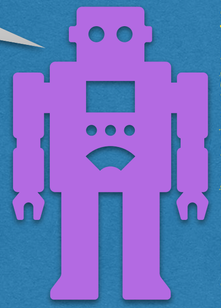
②ログインしたいです

ネタバラシとか
適当にYahooに飛ばすとか

サーバー側
https://kaneir0.jp

似てるけど
ちょっと違う

③バカめ！
パスワードはいただいた！



よく似た偽サイトによる
フィッシング詐欺
（SMSで届くスパムとかで誘導される）

理想のパスワード

理想のパスワード

- 傍受されても大丈夫（傍受されないことは期待できない）
- 漏洩されても大丈夫（どんなに注意していても漏洩はある）
- 偽サイトには入力できない（物理的に不可能であってほしい）

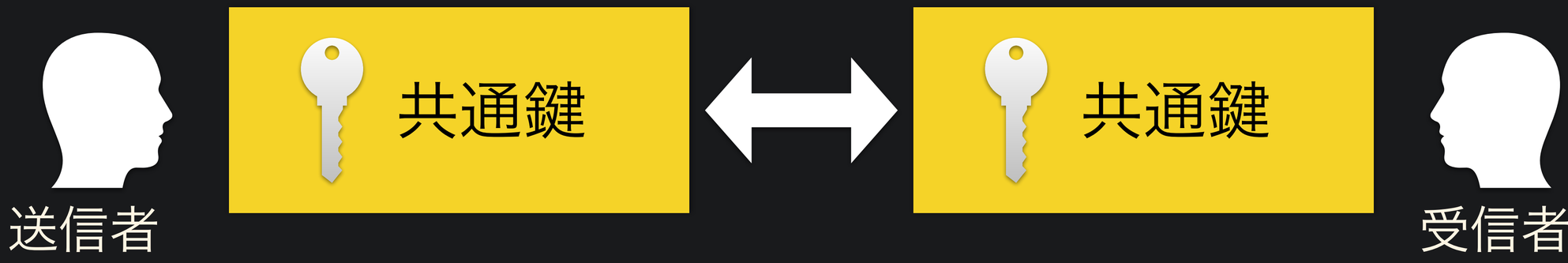
暗号の歴史から生まれたもの

【2000年以上続く戦争】暗号解読の歴史【ゆっくり解説】

https://youtu.be/Bt6wcDtANgw?t=1401

暗号の世界でも「鍵バレ」は長らく課題だった

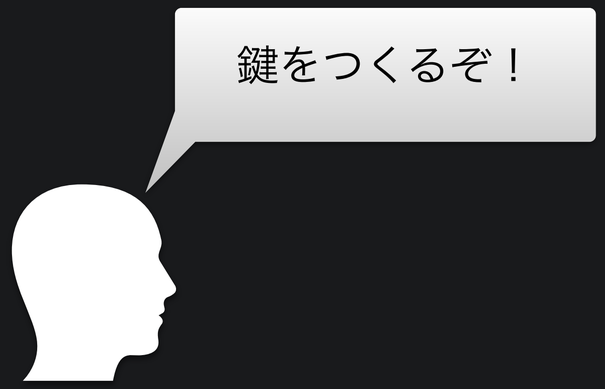
- 暗号は開封できないと意味がない
- 暗号化する時に使ったキーワードを復号する時にも使う
  - あるキーワードを使って変換表を作り出すイメージ
- 暗号化と復号に同じキーワードを使うので、**共通鍵方式**という



共通鍵方式の課題

- その性質上、一度は「送信者から受信者にキーワードを送信する」という通信が行われるため、これを傍受・強奪されると復号し放題になる

公開鍵方式の暗号はペアで作る

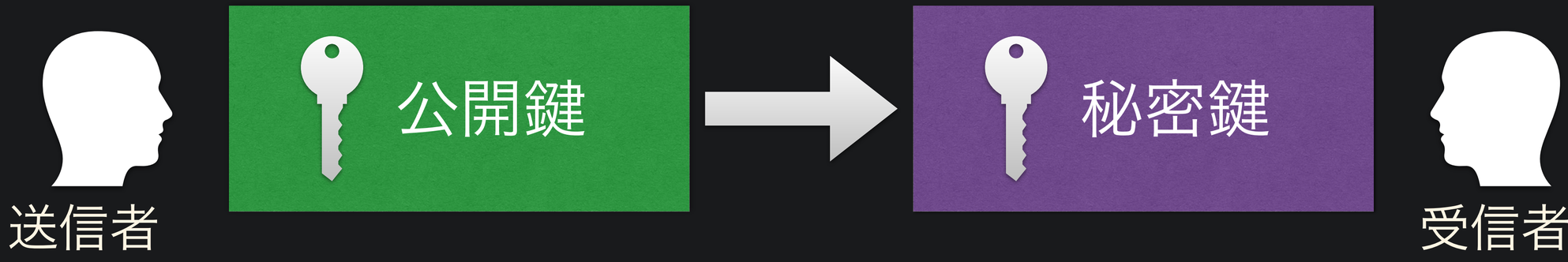


相手に渡す鍵
鍵を閉めるための鍵

作成者が持ち続ける鍵
鍵を開けるための鍵
誰にも渡さない

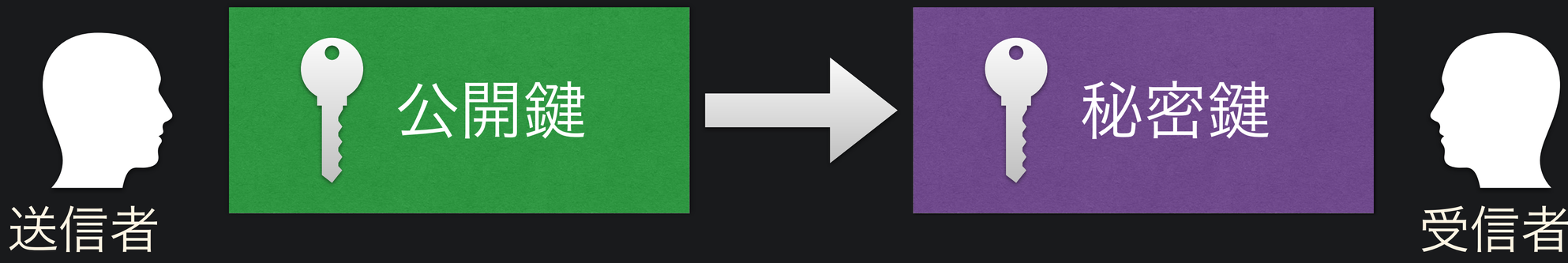
受信者しか開けることができない鍵

- 事前に送信者に公開鍵だけを送っておく
- 送信者は公開鍵で暗号化して受信者に送る
- 受信者は秘密鍵で復号する



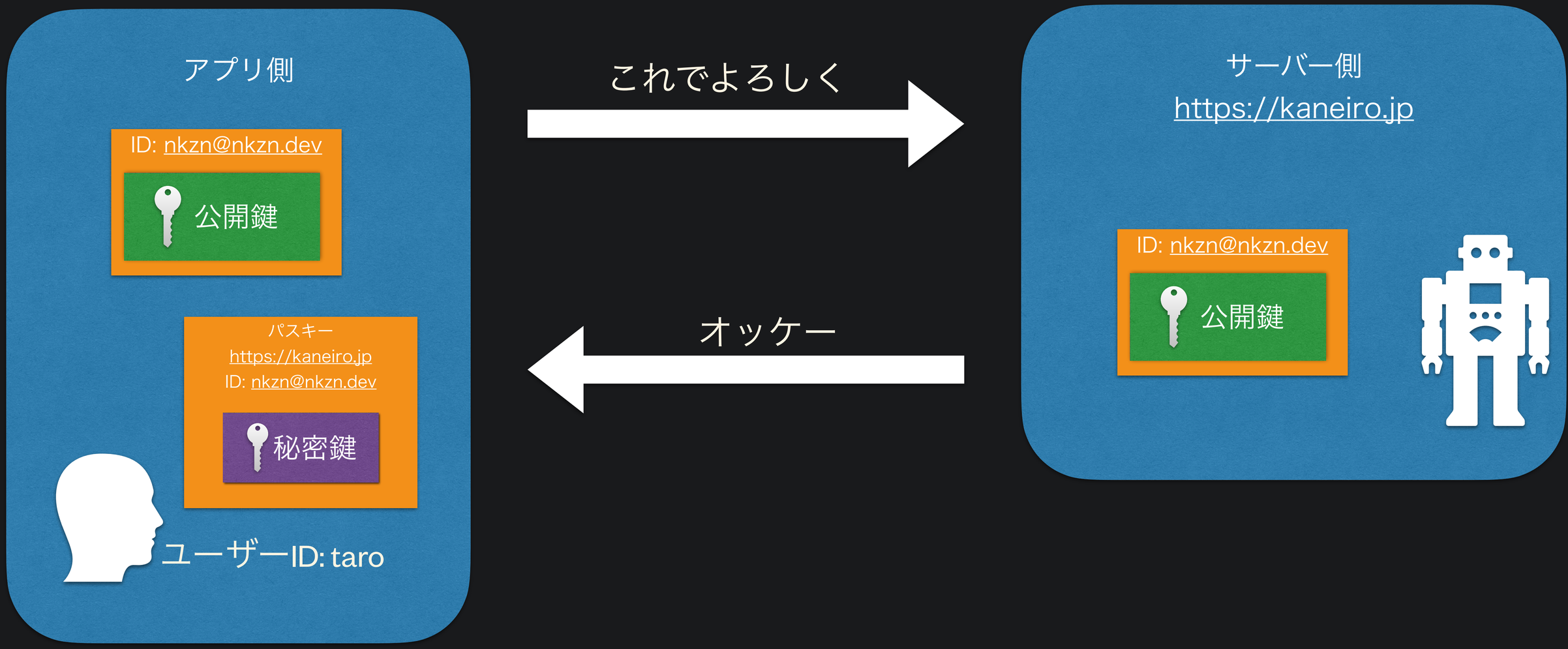
公開鍵方式のメリット

- 公開鍵は傍受される前提で作られている
- 公開鍵で暗号化したデータは公開鍵では開けられない
- 秘密鍵は一度も受信者の手を離れたことがないので傍受されない
- 安全にデータを送信できる



公開鍵暗号で認証する
パスキー

サインアップ


アプリ側
ID: nkzn@nkzn.dev
公開鍵
これでよろしく
サーバー側
https://kaneiro.jp
ID: nkzn@nkzn.dev
公開鍵
オッケー
パスキー
https://kaneiro.jp
ID: nkzn@nkzn.dev
秘密鍵
ユーザーID: taro

サインイン

アプリ側

パスキー
https://kaneiro.jp
ID: nkzn@nkzn.dev

秘密鍵

ユーザーID: taro

① ログインしたいです

② 暗号化データ送るから秘密鍵で
いい感じに再計算して送り返して

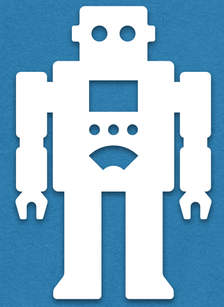
③ 計算結果を送る

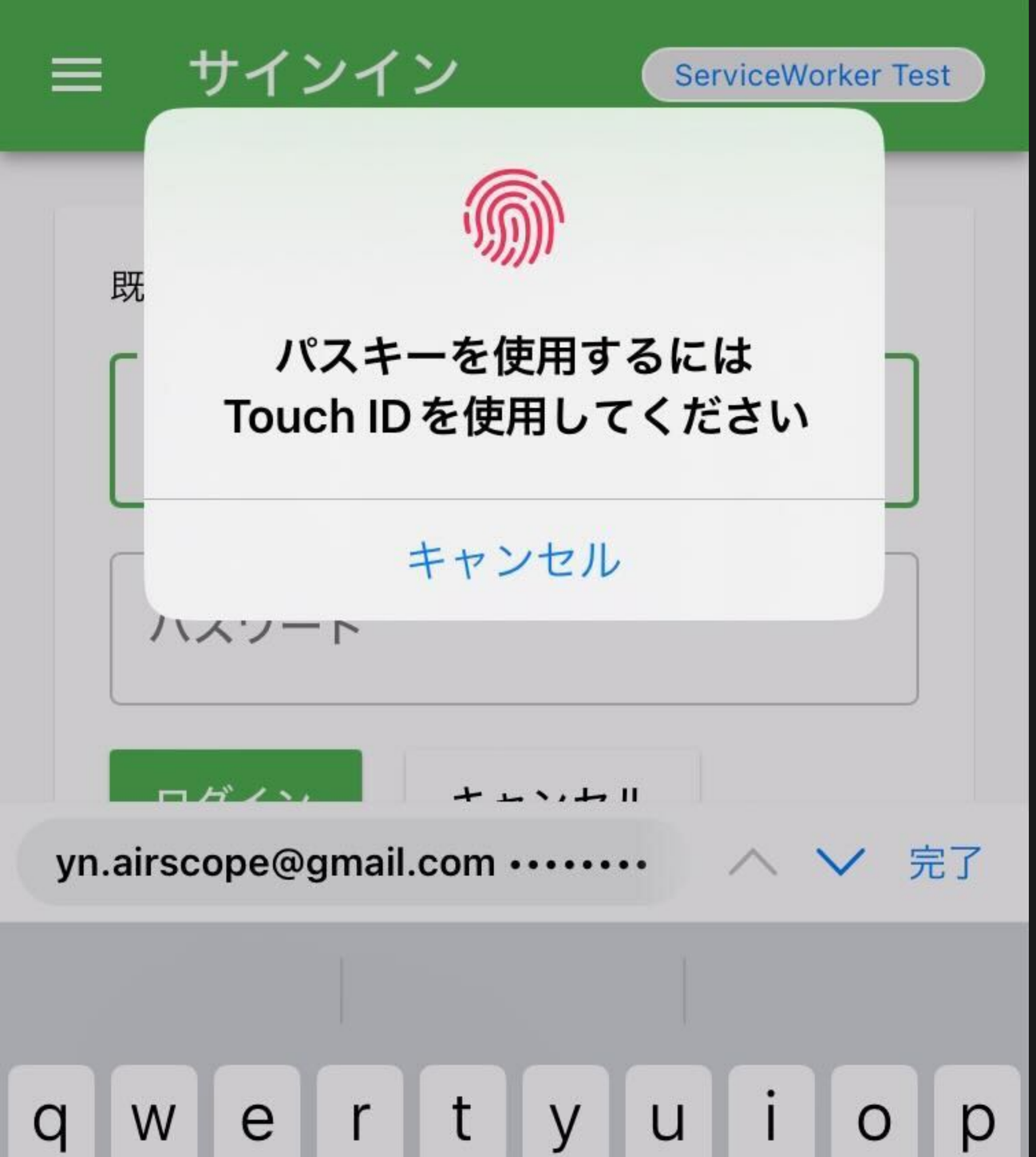
④ 検算したら②のデータを導けたので
君はtaroくんだね！

サーバー側
https://kaneiro.jp

ID: nkzn@nkzn.dev

公開鍵



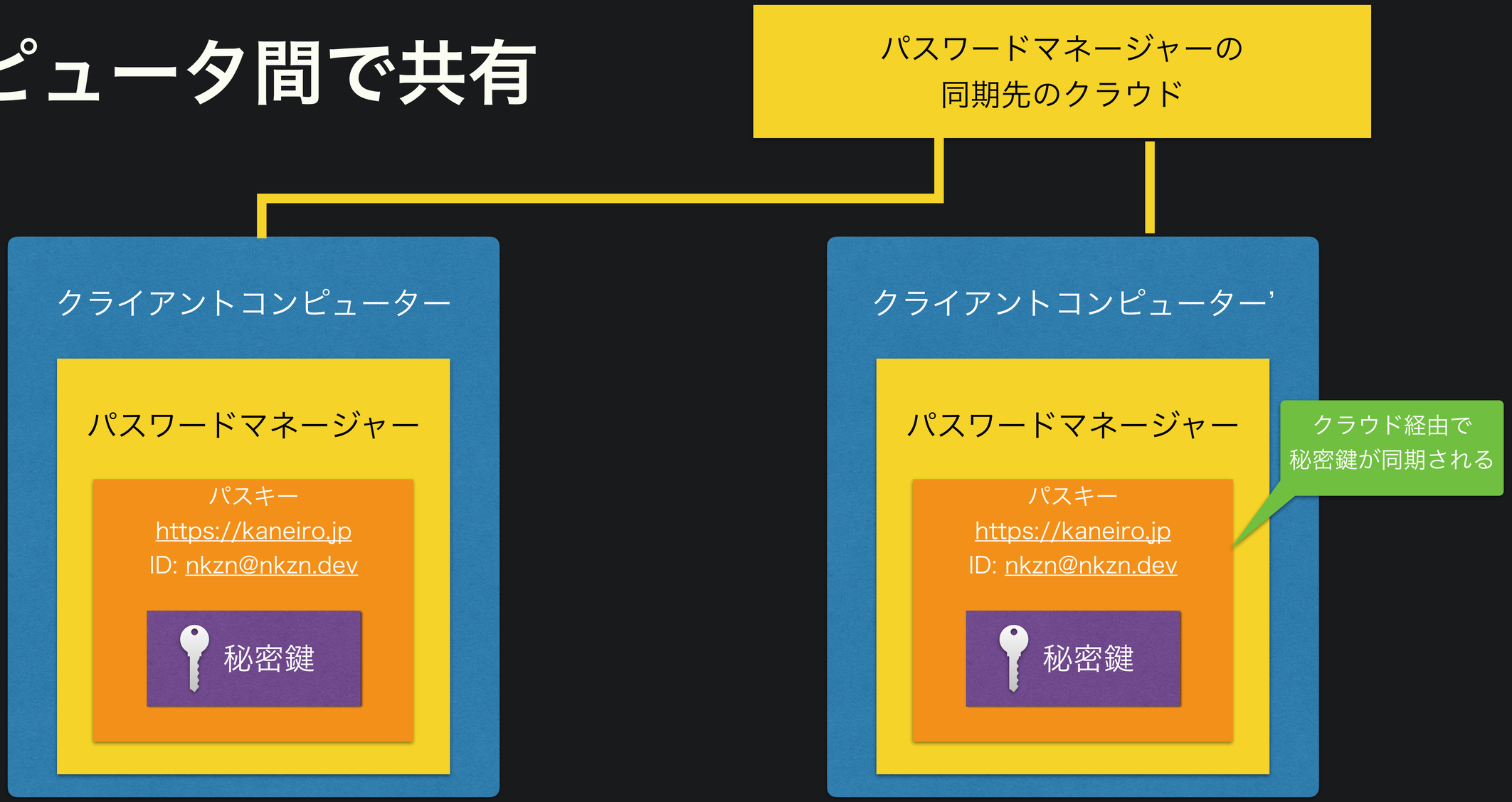


サインインの様子

スマホやPCが対応している場合は

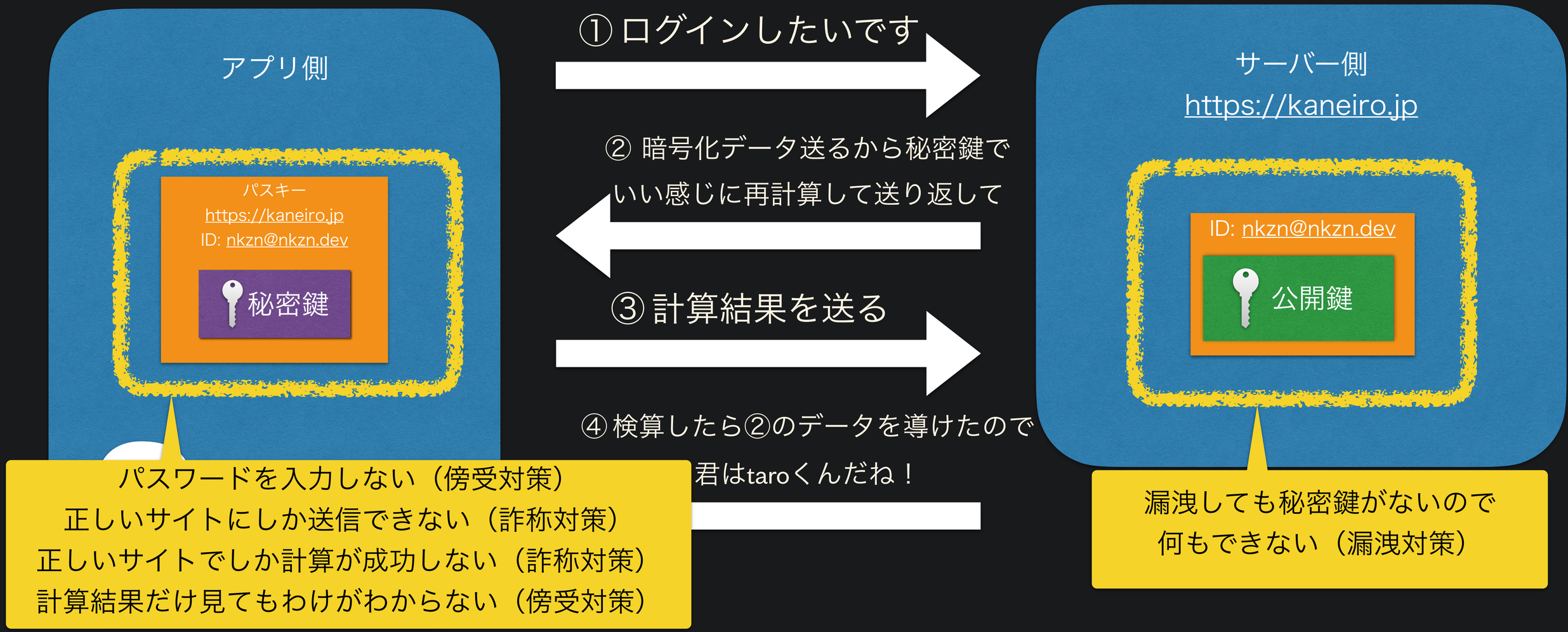
指紋認証や顔認証でログインする

コンピュータ間で共有



デバイス間で秘密鍵が共有されるので、所有している他のデバイスでもログインが容易

セキュリティ上のメリット



パスキーの活用事例

結構身近になってきました

パスキー使っていこうな