追跡可能暗号とその周辺の話題に関して

山田翔太
(産業技術総合研究所)

この発表の内容

- 暗号理論の枠組みの簡単な紹介
- Trace and Revoke Scheme
- 以下の論文の説明

  Shweta Agrawal, Sanjay Bhattacherjee, Duong Hieu Phan, Damien Stehlé, *Shota Yamada:*
  Efficient Public Trace and Revoke from Standard Assumptions.
  ACM-CCS 2017

この発表の内容

- 暗号理論の枠組みの簡単な紹介

- Trace and Revoke Scheme

- 以下の論文の説明

  Shweta Agrawal, Sanjay Bhattacherjee, Duong Hieu Phan, Damien Stehlé, *Shota Yamada:*
  Efficient Public Trace and Revoke from Standard Assumptions.
  ACM-CCS 2017

暗号技術とは

- 悪い人がいる状況下で、行いたいことを安全に行うための技術全般（公開鍵暗号、共通鍵暗号、電子署名、認証、ハッシュ関数、Traitor Tracing…）
  - 暗号化の場合:
    やりたいこと = 情報の送受信
    悪い人がやること = 盗聴
  - 認証の場合:
    やりたいこと = 本人確認
    悪い人がやること = なりすまし

暗号理論の枠組み

暗号技術に関して、達成したい機能性及び安全性要件を定義（＝暗号方式が破られる/破られないとはどういうことかを定義）

（候補となる）方式を設計

設計方式に安全性の証明を与える

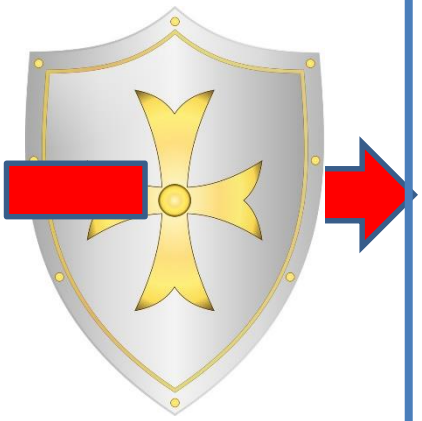
安全性証明の枠組み

- 暗号方式の安全性を無条件で証明することは極めて難しいことが多い（例えば公開鍵暗号の安全性を無条件で示せたらP≠NPの証明に成功したことになる）

- そこで、数学的に難しい問題（確率的多項式時間で解けなさそうな問題）を用意して、その問題の困難性に安全性を帰着する

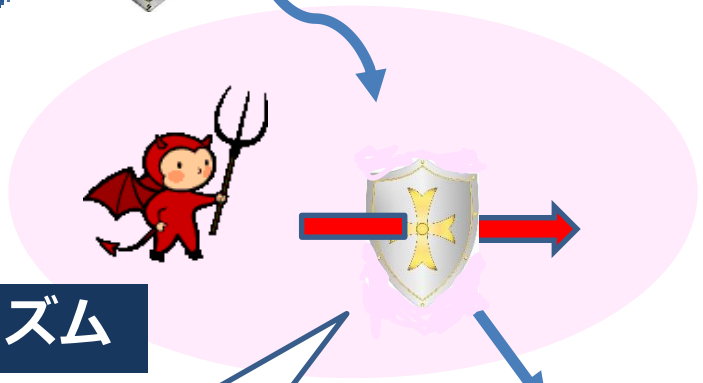
安全性証明の枠組み

攻撃者



暗号方式

数学的問題



アルゴリズム

数学的問題を暗号方式の中に”埋め込む”

暗号方式  を破る攻撃者  を仮定
⇒困難な数学的問題  を解けるアルゴリズム  が存在
⇒仮定が誤り．暗号方式は安全

利用する数学的な問題

- XX問題が難しいという仮定をXX仮定という
- より難しい問題の困難性の仮定のもと、安全性を証明したい

LWE仮定（格子系仮定）

素因数分解仮定

離散対数仮定

DCR問題

DDH仮定

利用する数学的な問題

- それぞれの問題の正確な難易度はわかっていない
- 色々な仮定に基づいて暗号方式を設計しておくことが備えになる

素因数分解仮定

離散対数仮定

DDH仮定

DCR問題

LWE仮定

この発表の内容

- 暗号理論の枠組みの簡単な紹介
- Trace and Revoke Scheme
- 以下の論文の説明

  Shweta Agrawal, Sanjay Bhattacherjee, Duong Hieu Phan, Damien Stehlé, *Shota Yamada:*
  Efficient Public Trace and Revoke from Standard Assumptions.
  ACM-CCS 2017

暗号化コンテンツ配信システム

有料番組配信等

:1000円

視聴可能

視聴不可

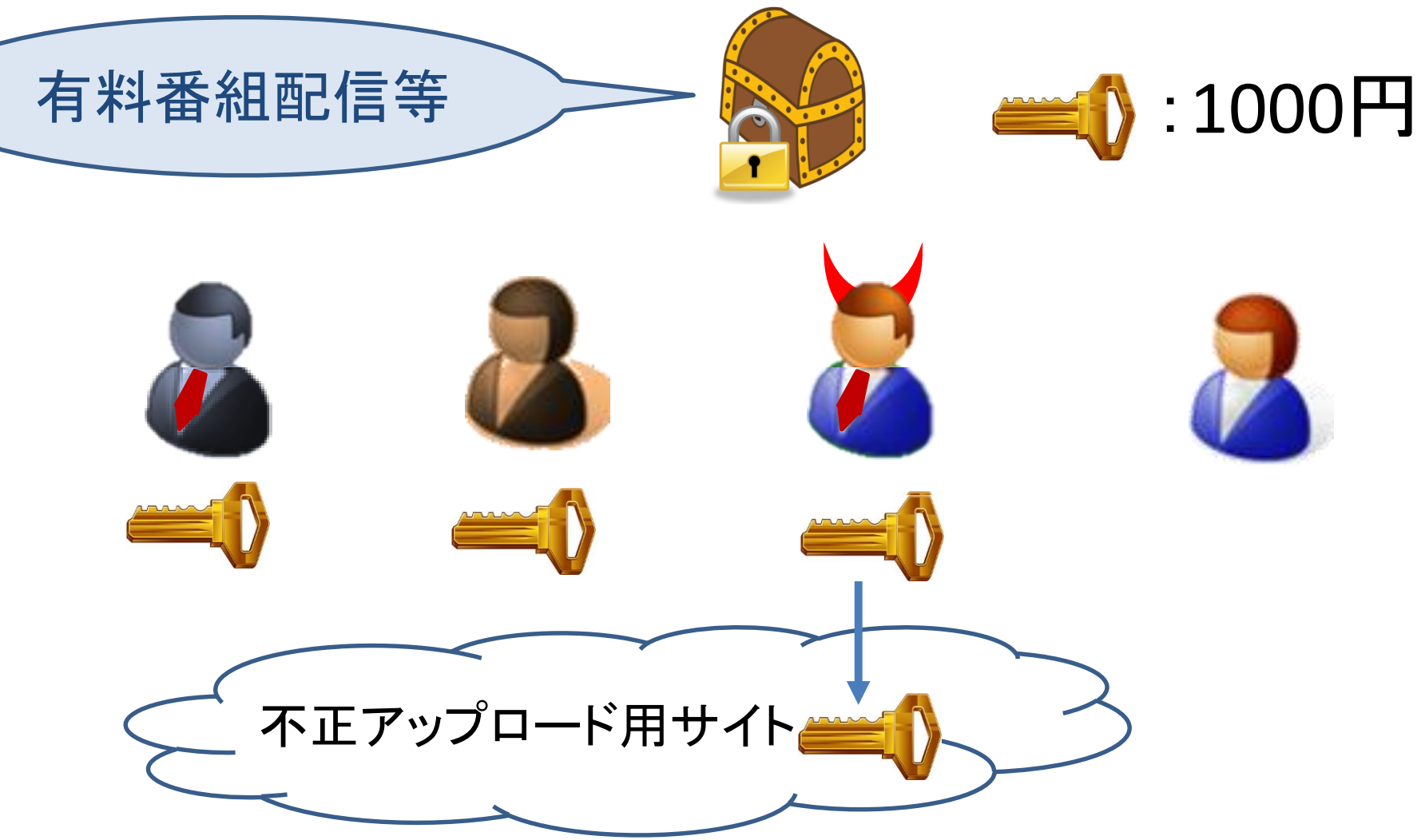
ナイーブな方法の問題点

有料番組配信等

:1000円

不正アップロード用サイト

ナイーブな方法の問題点



ナイーブな方法の問題点

有料番組配信等

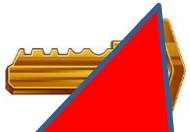
:1000円

不正アップロード用サイト

ナイーブな方法の問題点

有料番組配信等

:1000円



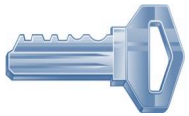
不正アップロード用サイト

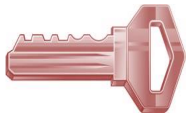
お金を払わずに
コンテンツにアクセス

解決策

鍵をユーザごとに
違うものにする

:1000円



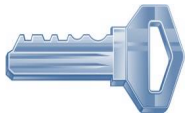


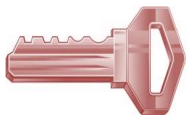
不正アップロード用サイト

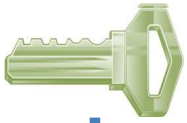
解決策

鍵をユーザごとに
違うものにする

:1000円



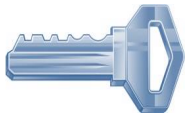


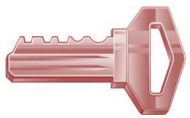


解決策

鍵をユーザごとに
違うものにする

:1000円



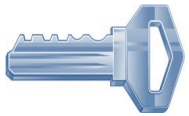


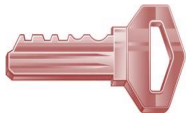
不正アップロード用サイト

解決策

鍵をユーザごとに
違うものにする

:1000円



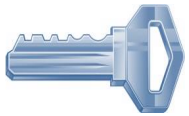


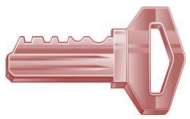
不正アップロード用サイト

解決策

鍵をユーザごとに
違うものにする

:1000円

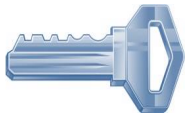


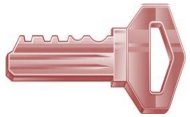


“Traceability”
⇒不正アップロードに対する抑止力

Traceability

:1000円

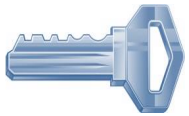


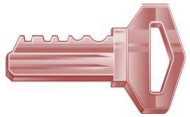


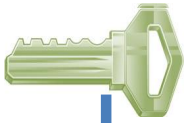
不正アップロード用サイト

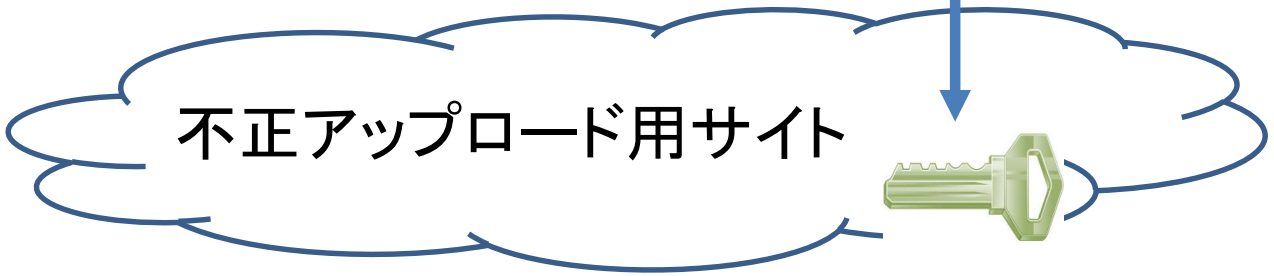
Traceability

:1000円





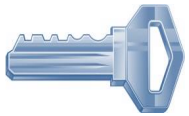


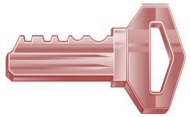


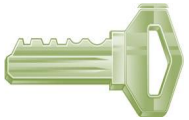
Traceability

Traceability

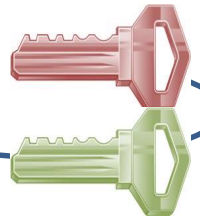
:1000円







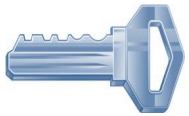
不正アップロード用サイト

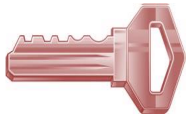


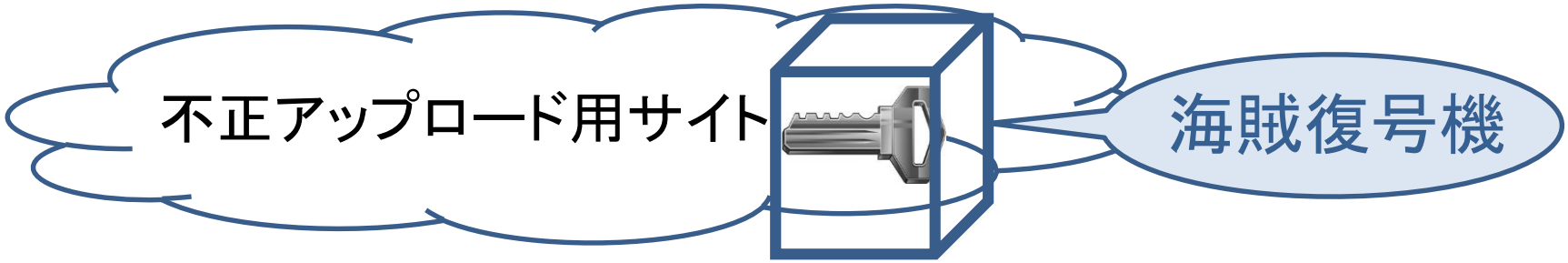
鍵を混ぜて
復号機を作る

Traceability

:1000円

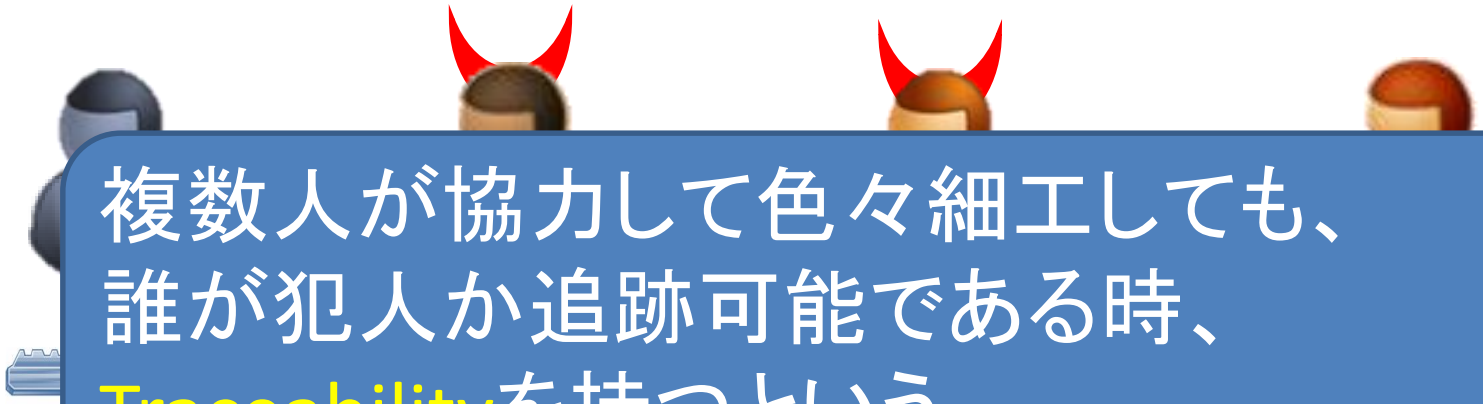






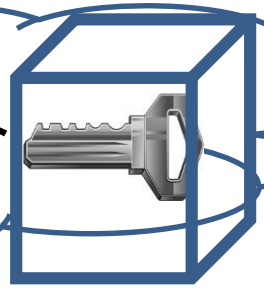
Traceability

:1000円



複数人が協力して色々細工しても、
誰が犯人か追跡可能である時、
Traceabilityを持つという

不正アップロード用サイト



海賊復号機

Public Traceability

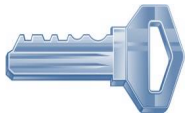
- 海賊復号機から不正者を追跡する計算処理が、秘密情報なしで実行可能な時Public Traceabilityをもつという。

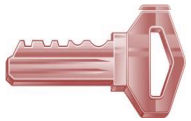
- 以下の利点
  - Trace algorithmの第三者への依頼が可能
  - 不正者が誰であるか、誰でも検証可能

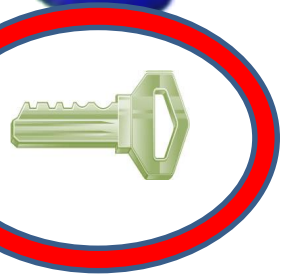
Revocationの必要性

: **月額**1000円

契約やめる
（もう料金は
支払わない）

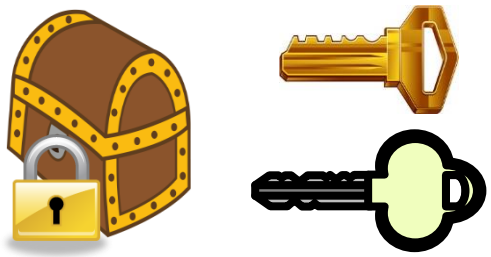




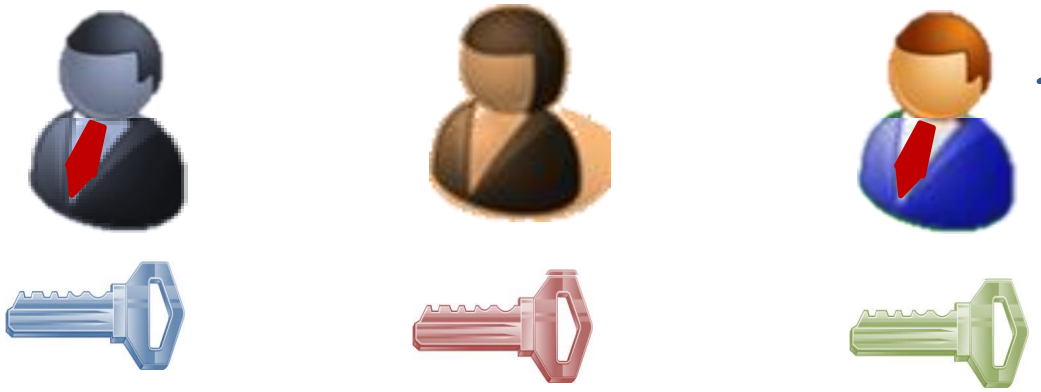


があれば契約終了後も
暗号文を復号してコンテンツを受信可能

ナイーブな解決策:システム再構築

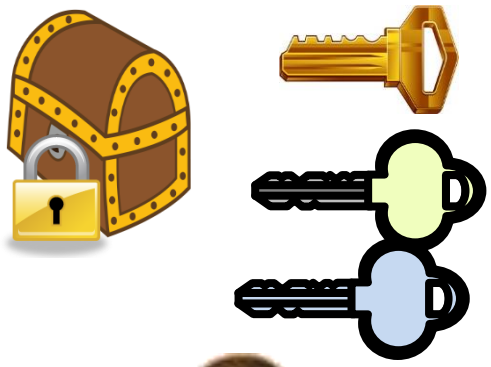


: **月額**1000円

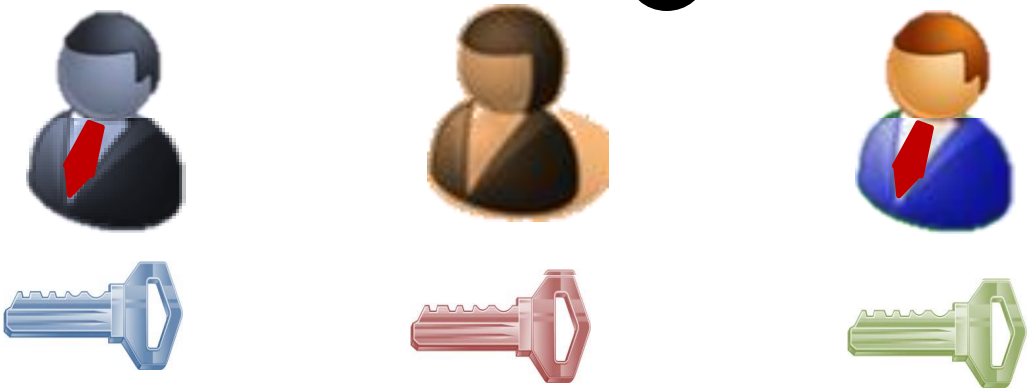


契約やめる
（もう料金は
支払わない）

ナイーブな解決策:システム再構築

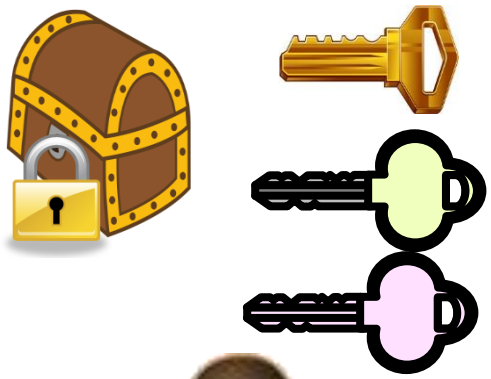


: **月額**1000円

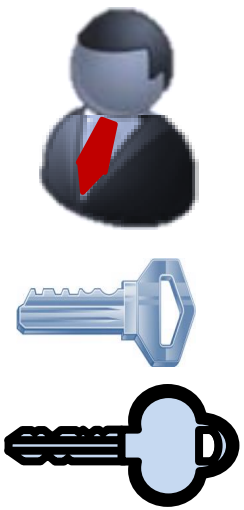


契約やめる
（もう料金は
支払わない）

ナイーブな解決策:システム再構築

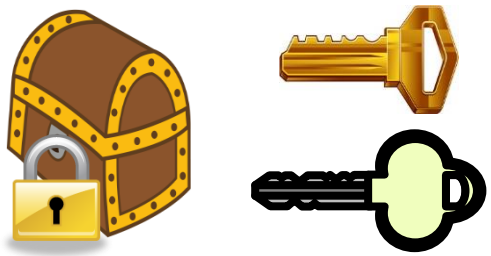


: **月額**1000円

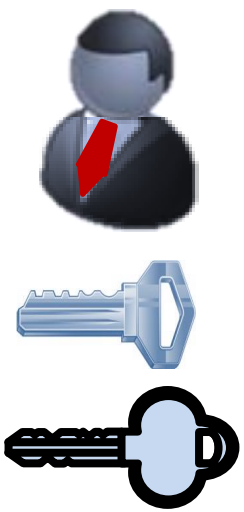


契約やめる
（もう料金は
支払わない）

ナイーブな解決策:システム再構築



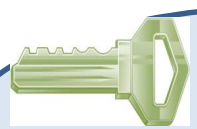
: **月額**1000円



契約やめる
(もう料金は
支払わない)

効率が悪い！

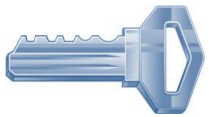
Revocation

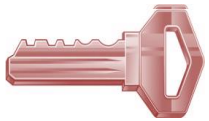

以外の鍵
で復号可能

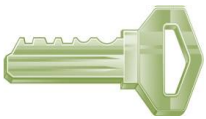

：月額1000円

暗号化の際に、
Revoked userを指定可能

契約をやめたユーザ
は復号できない







この発表の内容

- 暗号理論の枠組みの簡単な紹介

- Trace and Revoke Scheme

- 以下の論文の説明

  Shweta Agrawal, Sanjay Bhattacherjee, Duong Hieu Phan, Damien Stehlé, *Shota Yamada:*
  Efficient Public Trace and Revoke from Standard Assumptions.
  ACM-CCS 2017

紹介論文の内容

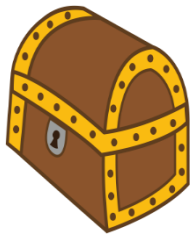
- Public TraceabilityをもつTrace and Revoke Schemeの提案
  - 内積暗号を部品として用いて構成

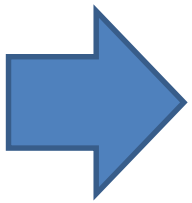
- 既存の内積暗号を使って、様々な数学的仮定から、(比較的)効率的な方式が得られる

内積(関数型)暗号

- $\mathrm{Setup}(1^{\lambda}) \rightarrow (\mathrm{mpk}, \mathrm{msk})$
- $\mathrm{Encrypt}(\mathrm{mpk}, \vec{x}) \rightarrow C_{\vec{x}}$
- $\mathrm{KeyGen}(\mathrm{msk}, \vec{y}) \rightarrow sk_{\vec{y}}$
- $\mathrm{Dec}\left(C_{\vec{x}}, sk_{\vec{y}}\right) \rightarrow \langle \vec{x}, \vec{y} \rangle$

$C_{\vec{x}}$ $sk_{\vec{y}}$

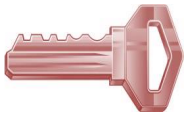


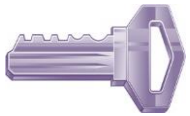


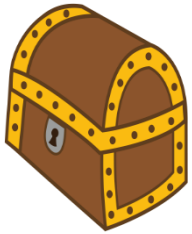
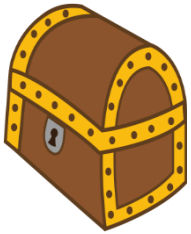
$\langle \vec{x}, \vec{y} \rangle$

内積暗号の満たす(べき)安全性

Given $sk_{\overrightarrow{y_1}}$ $sk_{\overrightarrow{y_2}}$ … $sk_{\overrightarrow{y_Q}}$



…


$C_{\overrightarrow{x_0}}$  $\approx$ $C_{\overrightarrow{x_1}}$ 

if $\langle x_0, y_j \rangle = \langle x_1, y_j \rangle$ $\forall j \in [Q]$

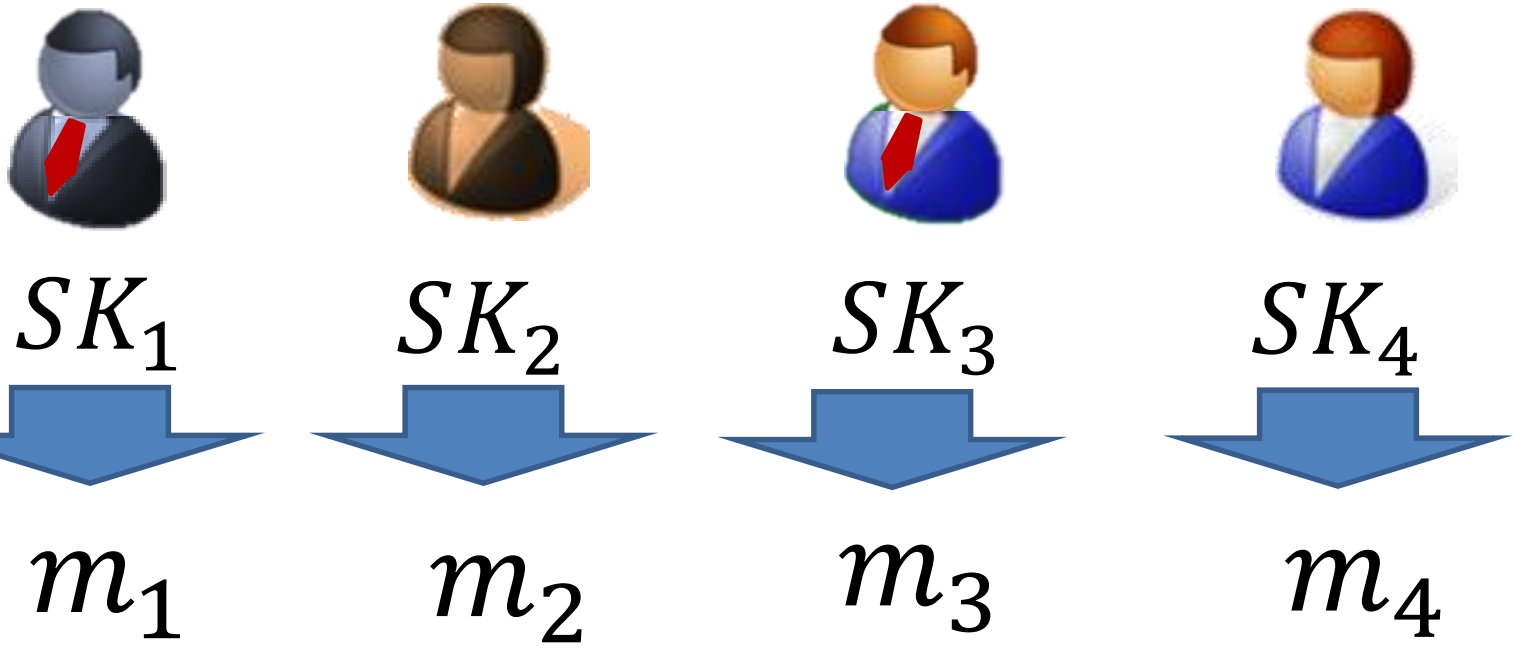
（直観）復号して得られる以上の情報は得られない

Tracing Algorithmのアイデア(1)

“自明な構成”をまず説明

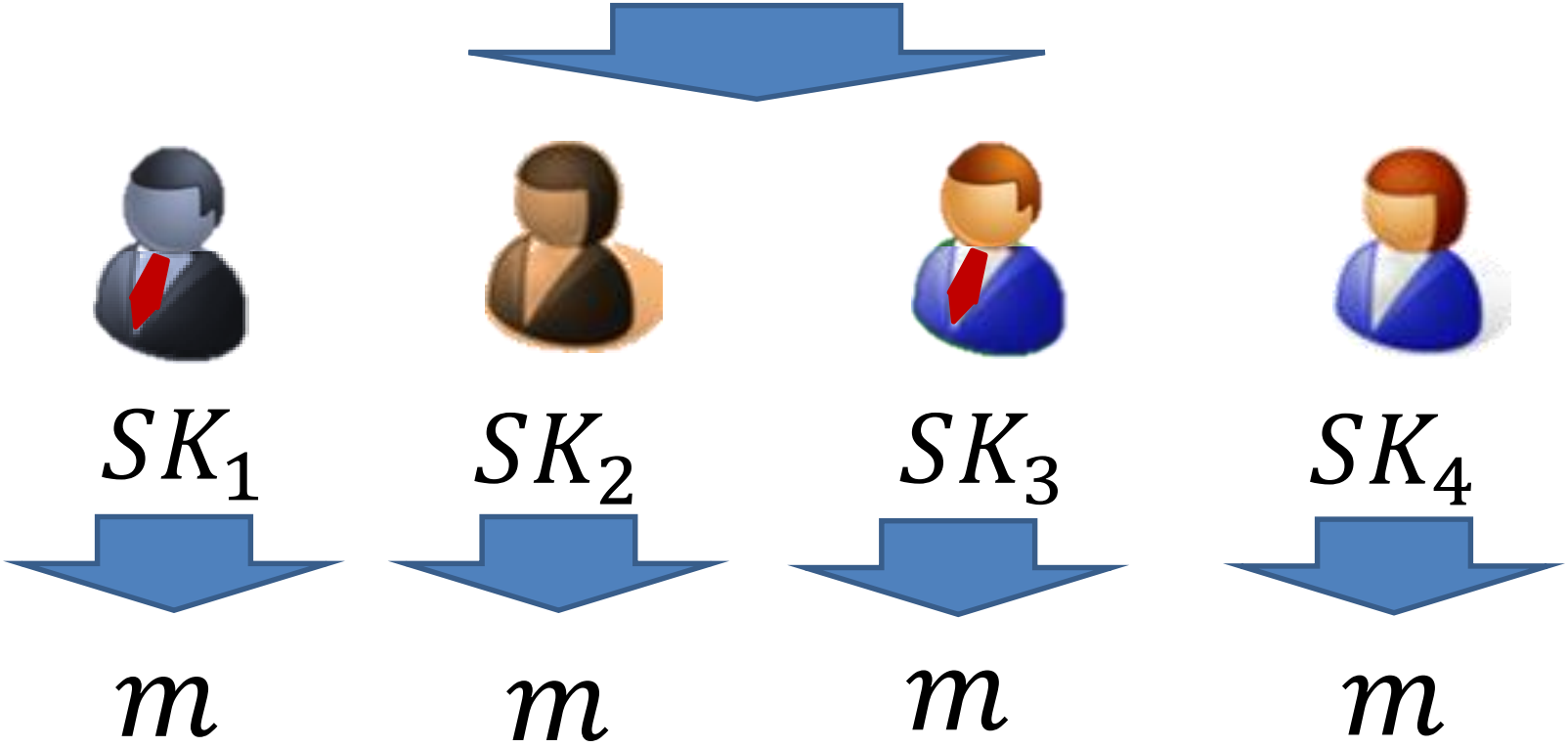
表記: $\mathrm{Enc}(m_1, m_2, m_3, m_4)$

$= (\mathrm{Enc}_{PK_1}(m_1), \mathrm{Enc}_{PK_2}(m_2), \mathrm{Enc}_{PK_3}(m_3), \mathrm{Enc}_{PK_4}(m_4))$

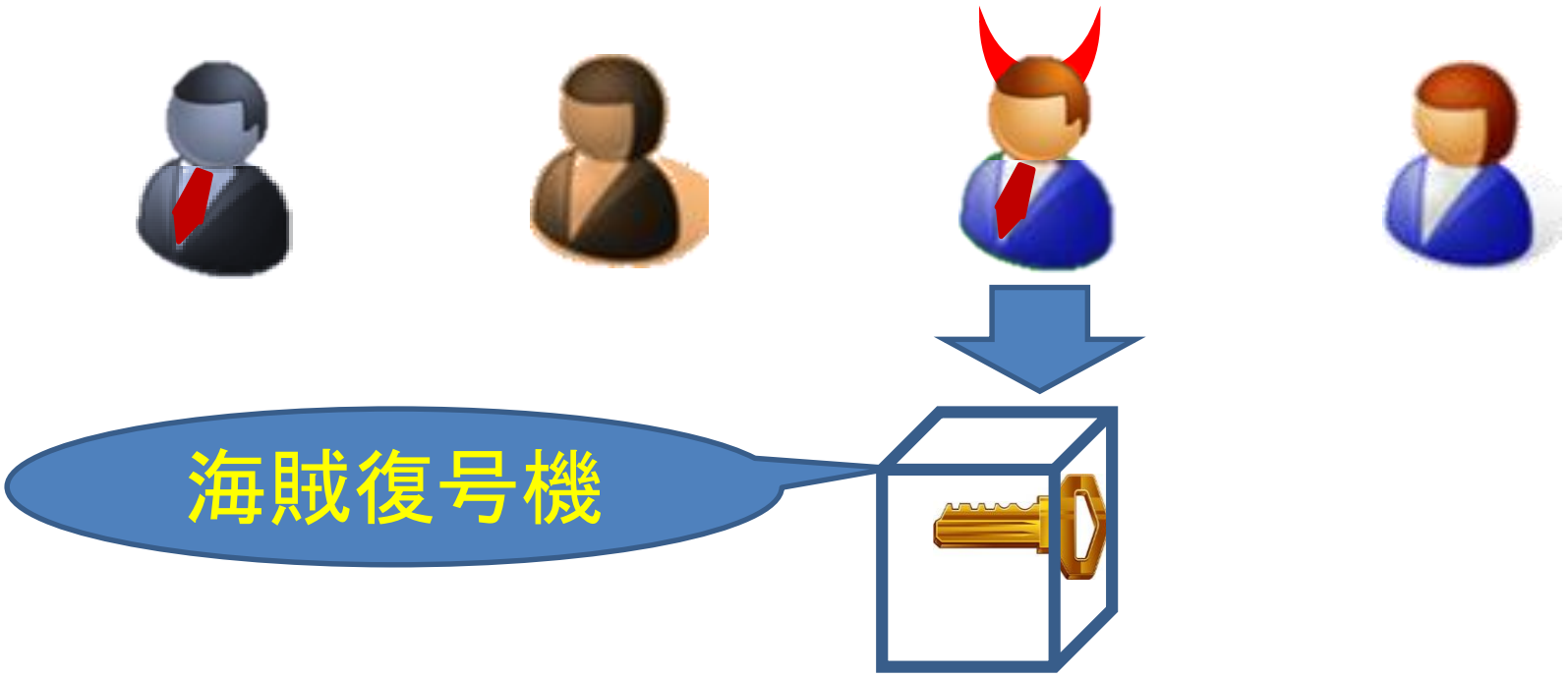


Tracing Algorithmのアイデア(2)

通常の暗号文：CT=Enc(m, m, m, m)

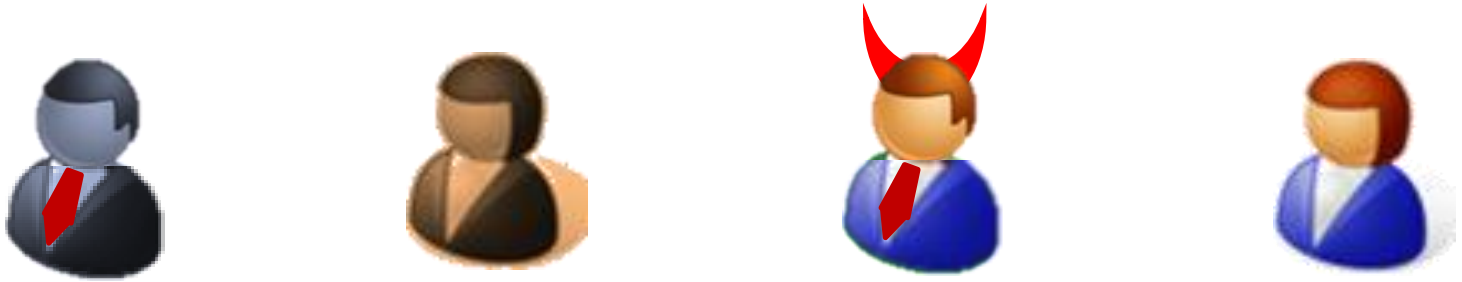


Tracing Algorithmのアイデア(3)



海賊復号機に、通常の形でない暗号文を入力して出力を調べることにより、犯人(海賊復号機作成者)を特定

Tracing Algorithmのアイデア(4)

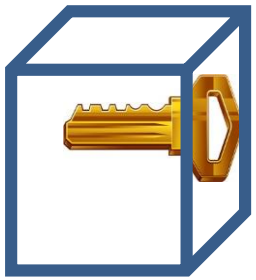


$\mathrm{Enc}(m,m,m,m)$ → m

$\mathrm{Enc}(m',m',m',m')$ → m'

$\mathrm{Enc}(m,m,m,m')$, $\mathrm{Enc}(m,m,m',m')$, $\mathrm{Enc}(m,m',m',m')$,
のどこかで出力が切り替わる

Tracing Algorithmのアイデア(5)



の入出力

$\mathrm{Enc}(m,m,m,m) \Rightarrow m$

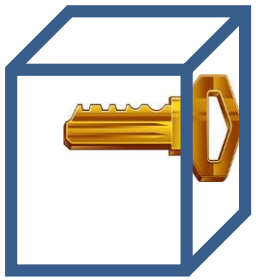
$\mathrm{Enc}(m,m,m,m') \Rightarrow ??$

$\mathrm{Enc}(m,m,m',m') \Rightarrow ??$

$\mathrm{Enc}(m,m',m',m') \Rightarrow ??$

$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$

Tracing Algorithmのアイデア(5)



の入出力

$\mathrm{Enc}(m,m,m,m) \Rightarrow m$

$\mathrm{Enc}(m,m,m,m') \Rightarrow ??$

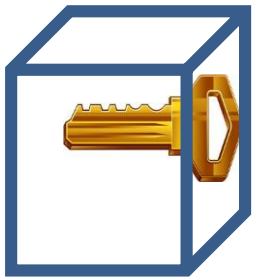
$\mathrm{Enc}(m,m,m',m') \Rightarrow ??$

$\mathrm{Enc}(m,m',m',m') \Rightarrow ??$

$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$

ポイント：  は3番目の要素の違いしかわからない

Tracing Algorithmのアイデア(5)



の入出力

$\mathrm{Enc}(m,m,m,m) \Rightarrow m$

$\mathrm{Enc}(m,m,m,m') \Rightarrow m$

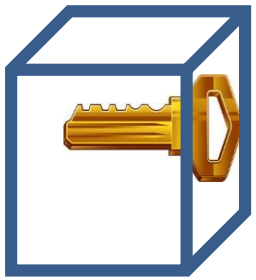
$\mathrm{Enc}(m,m,m',m') \Rightarrow ??$

$\mathrm{Enc}(m,m',m',m') \Rightarrow ??$

$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$

ポイント:  は3番目の要素の違いしかわからない

Tracing Algorithmのアイデア(5)



の入出力

$\mathrm{Enc}(m,m,m,m) \Rightarrow m$

$\mathrm{Enc}(m,m,m,m') \Rightarrow m$

$\mathrm{Enc}(m,m,m',m') \Rightarrow ??$

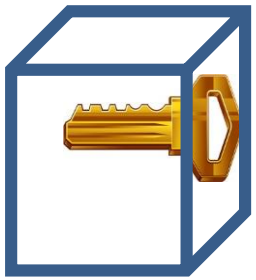
$\mathrm{Enc}(m,m',m',m') \Rightarrow m'$

$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$

ポイント：

は3番目の要素の違いしかわからない

Tracing Algorithmのアイデア(5)

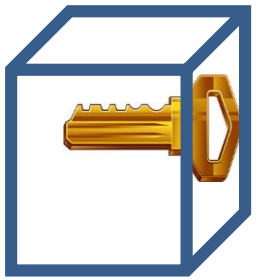


の入出力

$$\mathrm{Enc}(m,m,m,m) \Rightarrow m$$
$$\mathrm{Enc}(m,m,m,m') \Rightarrow m$$
$$\mathrm{Enc}(m,m,m',m') \Rightarrow m'$$
$$\mathrm{Enc}(m,m',m',m') \Rightarrow m'$$
$$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$$

ポイント:  は3番目の要素の違いしかわからない

Tracing Algorithmのアイデア(5)



の入出力

$\mathrm{Enc}(m,m,m,m) \Rightarrow m$

$\mathrm{Enc}(m,m,m,m') \Rightarrow m$

$\mathrm{Enc}(m,m,m',m') \Rightarrow m'$

$\mathrm{Enc}(m,m',m',m') \Rightarrow m'$

$\mathrm{Enc}(m',m',m',m') \Rightarrow m'$

ポイント:  は3番目の要素の違いしかわからない

不正ユーザに対応するスロットでしか、
出力は切り替わらない
⇒出力が切り替わるスロットが不正ユーザに対応

Tracing Algorithmのアイデア(6)

“自明な構成”では、ユーザごとに違う復号結果
が得られることが本質

Enc(m_1, m_2, m_3, m_4): ユーザiの復号結果はm_i

内積暗号で同じ状況を実現するには？

- ユーザiにはベクトル$\overrightarrow{v_i}$と，対応する秘密鍵$sk_{\overrightarrow{v_i}}$が対応
- ユーザiの復号結果がm_iになるように暗号化するには、

$$\langle \vec{x}, \overrightarrow{v_i} \rangle = m_i$$

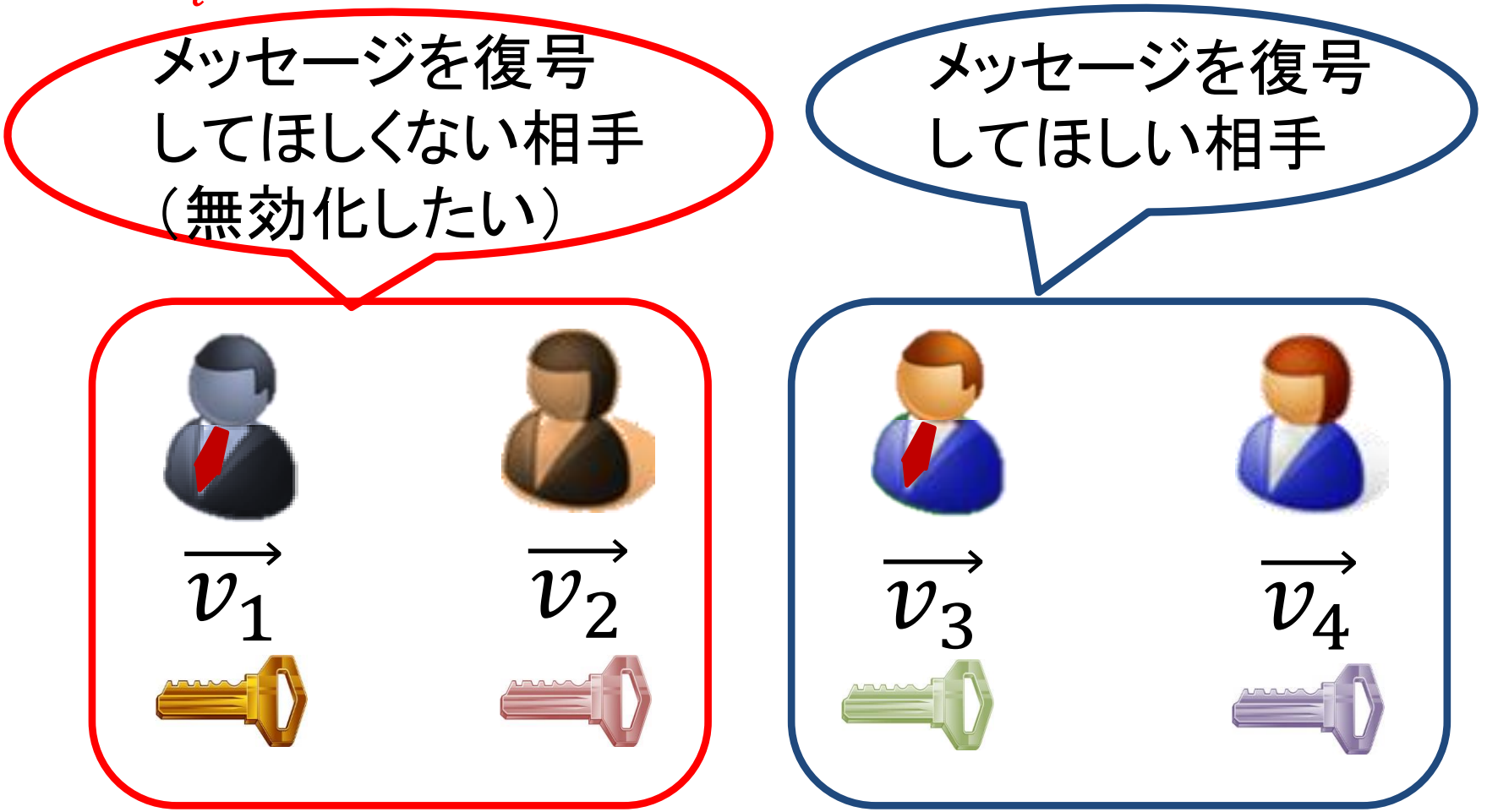
であるような$\vec{x}$を内積暗号で暗号化すればよい
（$\vec{x}$は連立方程式を解くことで得られる）

Public Traceabilityに関して

- ここまでのスライドで説明したとおり、Traceabilityは持っている
- 公開鍵内積暗号からスタートするので自然にPublic Traceabilityを持つ
  - 追跡処理で必要なのは暗号化処理と海賊復号機の実行処理のみだが、どちらも公開パラメータのみで実行可能

Revocationのアイデア（1）

- ユーザiにはベクトル$\overrightarrow{v_i}$と，対応する秘密鍵$sk_{\overrightarrow{v_i}}$が対応

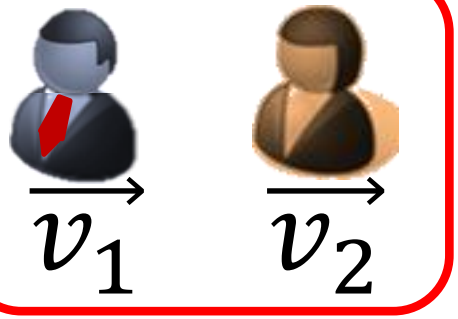


Revocationのアイデア（2）

Find $\vec{x}$ s.t. $\langle \vec{x}, \overrightarrow{v_1} \rangle = \langle \vec{x}, \overrightarrow{v_2} \rangle = 0$

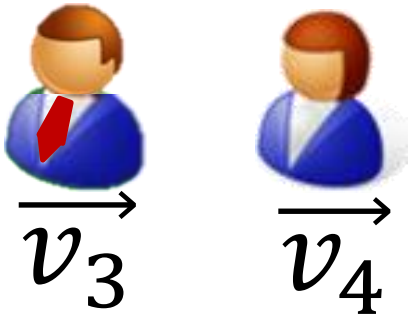
$\langle \vec{x}, \overrightarrow{v_3} \rangle \neq 0, \langle \vec{x}, \overrightarrow{v_4} \rangle \neq 0$

暗号文: $\left(\vec{x}, \mathrm{Enc}(\mathrm{mpk}, m \cdot \vec{x}) = C_{m \cdot \vec{x}}\right)$



$C_{m \cdot \vec{x}}$を復号:

$\langle m \cdot \vec{x}, \overrightarrow{v_1} \rangle = 0$ ⇒ mの取り出し不可 ✓



$C_{m \cdot \vec{x}}$を復号:

$\langle m \cdot \vec{x}, \overrightarrow{v_3} \rangle = m \cdot \overbrace{\langle \vec{x}, \overrightarrow{v_3} \rangle}^{\neq 0}$

⇒ mの取り出し可 ✓

TracingとRevocationの組み合わせ

- どちらの方法も、ユーザにベクトル$\vec{v_i}$と，対応する秘密鍵$sk_{\vec{v_i}}$を対応させるので組み合わせるのは容易
- 通常暗号文　$(\vec{x}, Enc(m \cdot \vec{x}))$の形

無効化ユーザと直交

- 犯人を捜すときの暗号文(probe ciphertext)は$(\vec{x}, Enc(m \cdot \vec{x} + \vec{y}))$という形

ここを変えて各ユーザの復号結果をコントロール

得られた結果のまとめ

鍵クエリ回数がBounded、
鍵クエリはランダムベクトルに対してのみ

[定理]
（標準的安全性よりも弱い安全性を満たす）
内積暗号から、Public Trace&Revoke方式が設計可能

既存の内積暗号[ALS16]を利用することで、
DDH仮定、DCR仮定、LWE仮定からの構成が得られる

LWEの構成については、弱い安全性でよいことを利用して、
[ALS16]に比べてパラメータの改善が可能
(Sub-exponential v.s. slightly super-polynomial approximation factor)