CybozuLabs

楕円曲線と暗号

2020/12/21

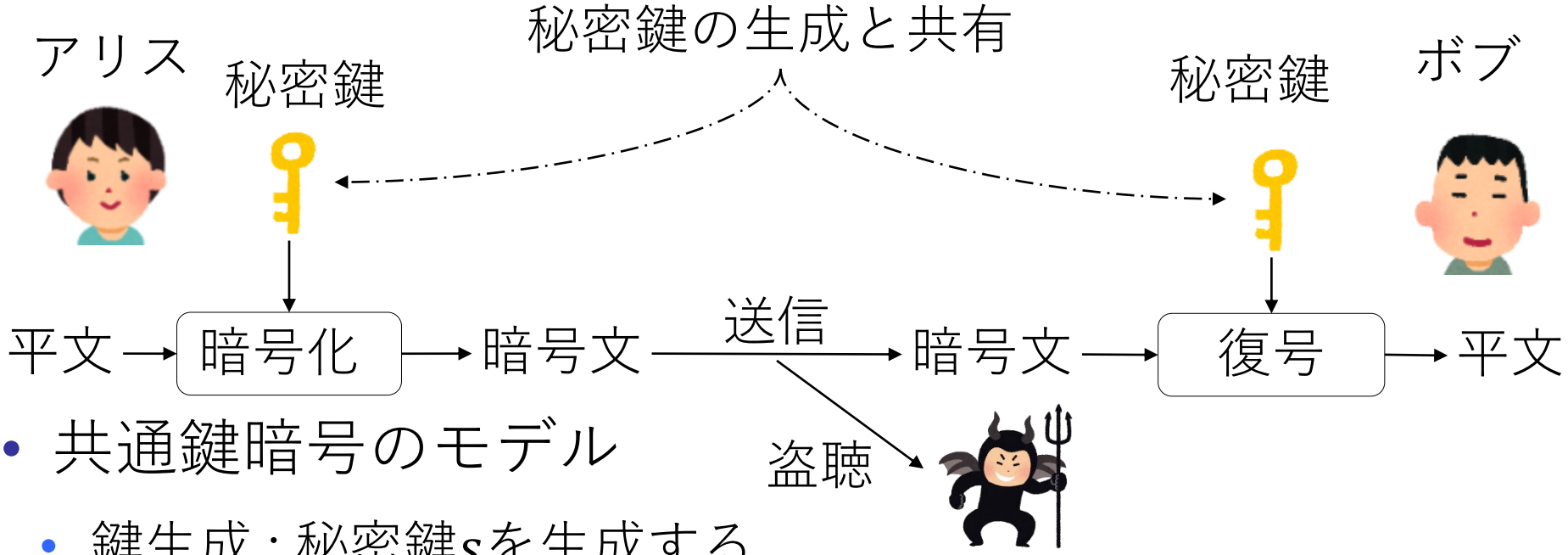
東京理科大学理工学部数学科談話会

光成滋生

目次

共通鍵暗号

- 暗号化と復号に同じ鍵を使う暗号



- 共通鍵暗号のモデル
  - 鍵生成：秘密鍵sを生成する
  - 暗号化：平文mを秘密鍵sで暗号化する $c = Enc(s, m)$
  - 復号：暗号文cを秘密鍵sで復号する $m = Dec(s, c)$
    - sを省略して$Enc(m)$や$Dec(c)$と書くこともある

イラストは(C)いらすとや https://www.irasutoya.com/

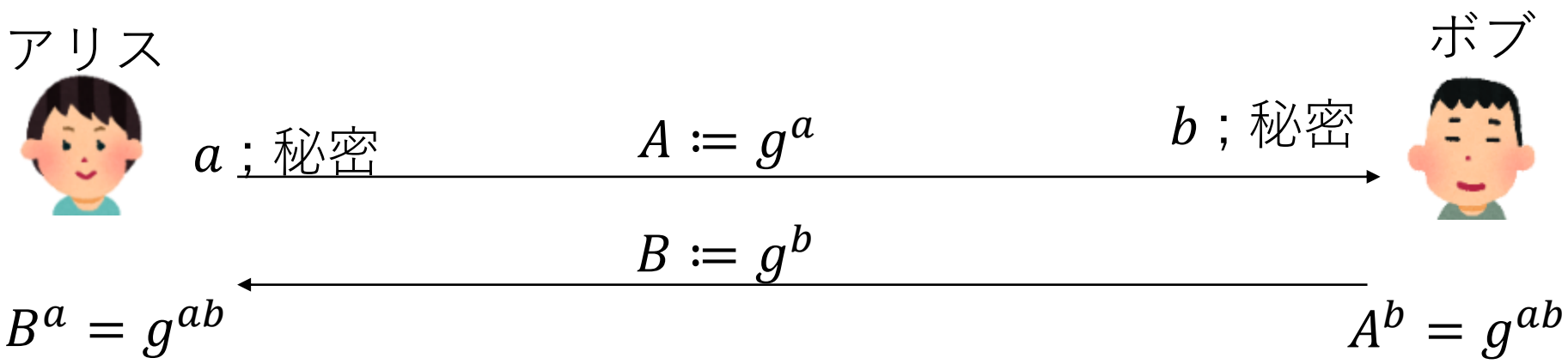
共通鍵暗号の性質

- 平文mを暗号化して復号すると元に戻る(正当性)
  - 任意のmに対して$Dec\big(s, Enc(s,m)\big) = m$
- 暗号文cから平文mの情報は得られない
  - 盗聴者がいても通信内容は漏洩しない
  - 「情報が得られない」とはどういう意味か
  - 後でもう少し詳しく考える

- 秘密鍵を事前に秘密に共有しておく必要がある
  - どうやって?

DH鍵共有

- DiffieとHellman(1976)
  - 1970年頃イギリスGCHQのEllisたちが発見していたらしい
  - p ; 素数, $G := \mathbb{F}_p^\times := \{1,2,\dots,p-1\}$, Gの生成元をgとする
- 方法
  - アリスはランダムに$a \in [1, p-1]$を選び$A := g^a$をボブに送る
  - ボブはランダムに$b \in [1, p-1]$を選び$B := g^b$をアリスに送る
  - アリスは$B^a = g^{ab}$を得る / ボブは$A^b = g^{ab}$を得る
  - $s := g^{ab}$を共有した(これを共通鍵暗号の秘密鍵に利用する)



DH鍵共有の安全性

- 盗聴者は次の情報を得る
  - $p, g, A = g^a, B = g^b$
  - ここから$s = g^{ab}$を得られるか?
- 原理的にはg, g^aからaを求められる
  - 離散対数問題DLP(Discrete Logarithm Problem)という
  - 現在知られている最速のアルゴリズムは数体篩法
    - 計算量$L_p(1/3, (^{32}/_{9})^{1/3})$
    - $L_p(s, c) = \exp\left((c + o(1))(\log p)^s(\log\log p)^{1-s}\right)$
  - p~2^{2048}程度のとき概ね2^{128}以上でDLPは解けない

DH問題

- g, g^a, g^bが与えられたとき
  - DLPは解けないかもしれないがg^{ab}は計算できるかもしれない
  - DHP(DH Problem)という
- DLPとDHPの関係
  - DLPが解ければDHPは解ける
  - 逆は不明(同等の計算量が必要なのか真に小さいのか)
  - 今のところ同等に難しいと思われている
  - 「DHPが困難ならDH鍵共有は安全」(トートロジー)
- 注意
  - 攻撃者は盗聴のみを想定
  - 途中経路で改竄を想定すると安全ではない(中間者攻撃)
    - その場合は署名(後述)などを組み合わせる

楕円曲線

- 有限体$\mathbb{F}_p(p > 3)$上定義された楕円曲線

$$E(\mathbb{F}_p) \coloneqq \left\{(x, y) \in (\mathbb{F}_p)^2 \middle| y^2 = x^3 + ax + b\right\} \cup \{0\}$$

  - Weierstrass方程式
  - 0は無限遠点, $a, b \in \mathbb{F}_p$, $4a^3 + 27b^2 \neq 0$
  - 楕円曲線に加法群の構造(点同士の加減算が定義される)が入る
- $G \coloneqq E[r] \coloneqq \{P \in E(\mathbb{F}_p) | rP = 0\}$とする
  - r倍して0となる点の集合(r等分点の集合)
  - PをGの生成元とする$G = \{0, P, 2P, \ldots (r-1)P\}$

ECDH鍵共有

CybozuLabs

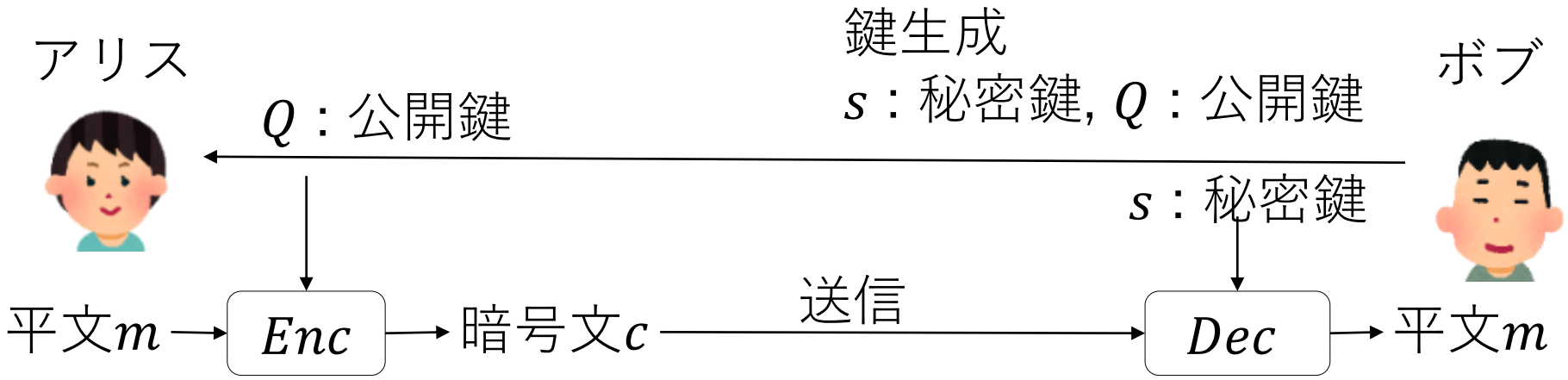
- DH鍵共有の楕円曲線版
  - アリスはaを選んでaPをボブに
  - ボブはbを選んでbPをアリスに
    - 二人だけがabPを計算できる
      - アリス : $a(bP) = abP$
      - ボブ : $b(aP) = abP$
- TLS1.3で必須の方法
  - 「https://~」でアクセスするとき利用される

ECDLPとECDHP

- 有限体のDLPとDHPの楕円曲線(Elliptic Curve)版
- ECDLP
  - P, aPが与えられたとき(aは未知), aを求めよ
- ECDHP
  - P, aP, bPが与えられたとき(a, bは未知), abPを求めよ
- 現在知られている最速のアルゴリズムは$O(\sqrt{r})$
  - r~2^{256}な素数なら2^{128}の計算コスト(解けない)
  - 有限体に比べて小さい群サイズで同等の安全性(2048 vs 256)
- 注意
  - ${}^{\mathbb{Z}}/_{p\mathbb{Z}} = \{0,1,2,\dots,p-1\}$, $g, h \in \left({}^{\mathbb{Z}}/_{p\mathbb{Z}}\right)^{\times}$が与えられたとき
    - $h = ag$となるaを求めるのは容易(拡張Euclid互除法)

公開鍵暗号

- 暗号化の鍵と復号する鍵が別
  - 暗号化する鍵は誰に知られてもよい



- 公開鍵暗号のモデル
  - 鍵生成：公開鍵(暗号化鍵)Qと秘密鍵(復号鍵)sの生成
  - 暗号化：平文mを公開鍵Qで暗号化 $c = Enc(Q, m)$
  - 復号：暗号文cを秘密鍵sで復号 $m = Dec(s, c)$
- 正当性：$Dec\big(s, Enc(Q, m)\big) = m$

Lifted-ElGamal暗号

- $G = \langle P \rangle$; DLPが困難な素数位数pの巡回加法群
  - $\{P, aP\}$からaを求めるDLPを$DLP_P(aP) \coloneqq a$と書くことにする
- 鍵生成
  - 秘密鍵$s \in \mathbb{F}_p$をランダムに選ぶ(以降$s \leftarrow \mathbb{F}_p$書く)
  - $Q \coloneqq sP$が公開鍵(DLPの困難性からsは分からない)
- 暗号化
  - 平文mに対して$r \leftarrow \mathbb{F}_p$をとり$Enc(Q, m) \coloneqq (mP + rQ, rP)$
  - $Enc(m; r)$と乱数rを明示することもある
- 復号
  - 暗号文$c \coloneqq (S, T)$に対して$dec(s, c) \coloneqq S - sT$
  - $Dec(s, c) \coloneqq DLP_P(dec(s, c))$; DLPが解ける範囲(e.g., $\sim 2^{32}$)のm
- 正当性 : $dec\big(s, Enc(Q, m)\big) = (mP + rQ) - s(rP) = mP$

Lifted-ElGamal暗号の安全性

- 「はい(1)」か「いいえ(0)」と分かっている暗号文
  - アンケート結果など
  - 攻撃者は自分で選んだ平文の暗号文を作れる(平文選択攻撃)
  - 同じ平文mでも毎回異なる暗号文になる必要がある
- Encが乱数を含まない決定的アルゴリズムなら
  - 平文がある小さい範囲にあると分かっているとする

暗号文$c = Enc(?)$が与えれた

一致するまで繰り返す

攻撃者は自分で暗号文を作れる

$Enc(1)$ 違う

$Enc(2)$ 違う

$Enc(3)$

…

違う…

$Enc(100)$

c

一致! $Dec(c) =? = 100$だ

- cf. 教科書的RSA暗号は決定的アルゴリズムなので安全でない

DDH(Decisional DH)問題

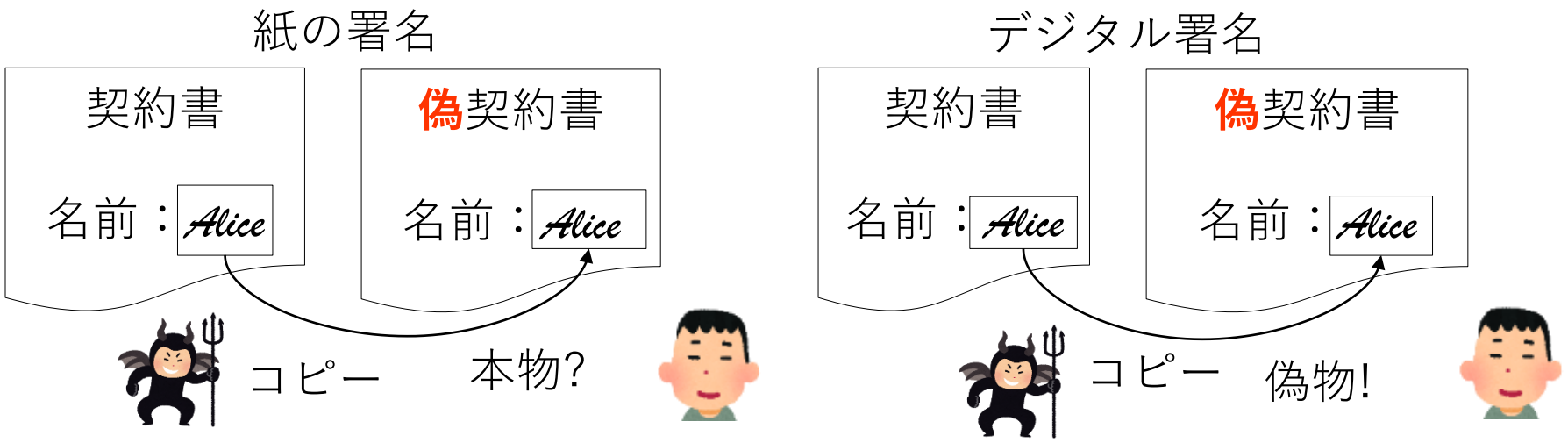
- 0の暗号文$c = Enc(0; r) = (rQ, rP)$
  - これが0の暗号文かそうでないかが分かると安全でない
- 攻撃者の視点
  - $P, Q = sP$を持っている, $c = (C_0, C_1) \stackrel{?}{=} (rsP, rP)$
  - もし(P, sP, rP)に対してDHPが解けるならrsPが求まる
  - C_0と比べることで$Dec(c) = 0$かそうでないかが分かる
- DDH問題
  - 「P, aP, bP, Q」が与えられたときに$Q = abP$か否か判定せよ
  - DDH問題が困難ならDHPが困難
  - DDH～DHPかどうかは未解決
- DDH問題が困難
  - ⇒ Lifted-ElGamal暗号は暗号文の区別ができない(ので安全)

安全性の定義

- 基本的に暗号は秘密鍵, 平文の全パターンを調べればどれかは当たる
  - ランダムに選んでも当たる確率は0ではない
    - これをもって「暗号はいつか破れる」というのはナンセンス
- 計算量的な仮定を入れる
  - セキュリティパラメータλ: 安全性を評価する指標
  - 攻撃者の能力はλに関して確率的多項式時間(PPT)
  - 多項式時間では解けないと思われている問題X
  - ある暗号があるPPTなアルゴリズムで破れるならXを解ける
    - Xが破れないならその暗号は安全
  - 攻撃者の攻撃手段もモデル化する
    - 時間があれば終わりのスライドで

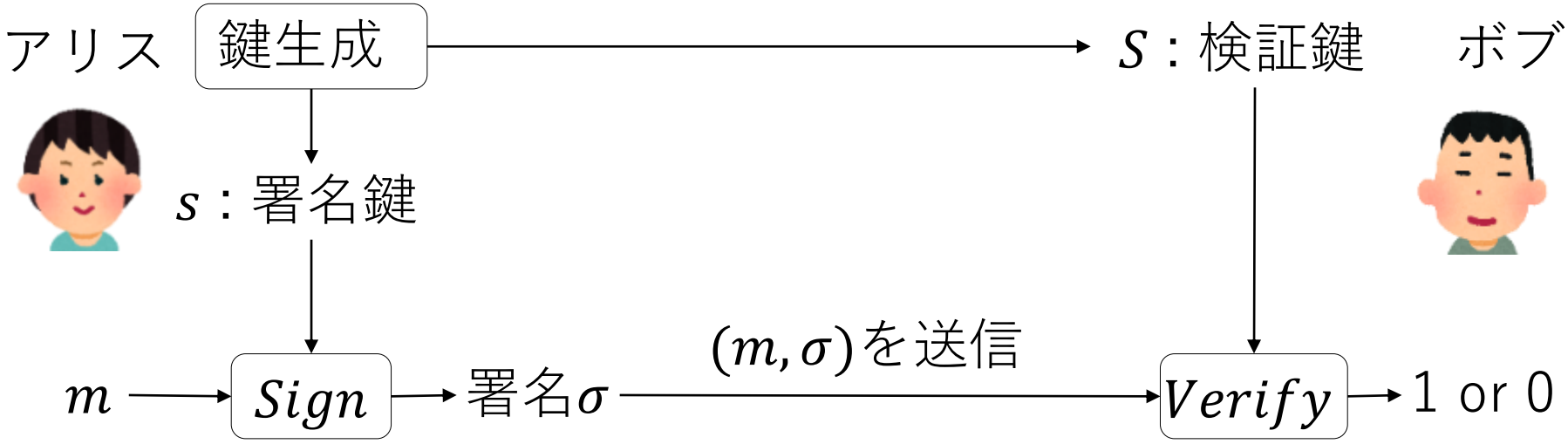
署名の概念

- ある人がある情報mに同意したことを保証する仕組み
  - 紙の書類に対するサインや判子
    - 偽造や書類の書き換えが可能
  - デジタルの場合はコピーが容易
    - サインや判子に相当するものを情報mと不可分な形にする
- 暗号化=機密性(秘匿性), 署名=完全性(正確さ)
  - 両方を組み合わせて通信の秘匿性と正しさを保証する



署名

- 鍵生成：署名鍵sと検証鍵Sの生成
  - 署名鍵は秘密, 検証鍵は公開
- 署名：メッセージmに対して署名鍵sで署名する
  - $\sigma := Sign(s, m)$
- 検証：メッセージmと署名σと検証鍵Sに対して
  - $Verify(S, m, \sigma)$ =1(受理) or 0(拒否)



署名の安全性

- 攻撃者は受理できる偽の署名を生成できてはいけない
- 攻撃者の持ち駒
  - 検証鍵Sとアリスが署名した多数の$\{(m_i, \sigma_i := Sign(s, m_i)\}$
  - 攻撃者が選んだm_iにアリスが署名してくれる状況を想定

S：検証鍵 ← 鍵生成

s：署名鍵

繰り返す

$m_i \neq m$ →

← $\sigma_i := Sign(s, m_i)$

攻撃対象

(m, σ) → $Verify$ → 0（どんなにがんばっても受理されない）

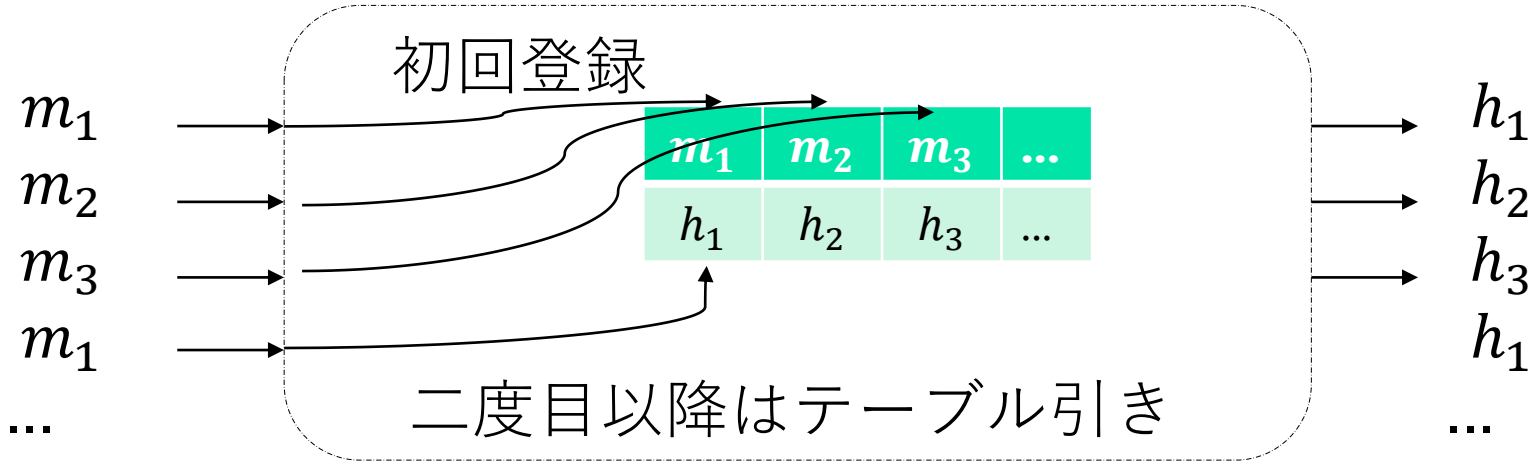
- 適応的選択メッセージ攻撃に対して存在的偽造不可

ハッシュ関数

- 署名σは固定長だが実際の応用で使われるメッセージmは任意長さであることが多い
  - メッセージを小さくする仕組みが必要
- (暗号学的)nビット出力ハッシュ関数$H: X \to Y := [0,2^n)$
  - Hは効率よく計算できるもの
- 一方向性
  - $x \in X, y := H(x)$が与えられたときにxを見つけられない
- 衝突困難性
  - $(x, x') \in X^2$で$x \neq x', H(x) = H(x')$なものを見つけられない
- 「見つけられない」の意味
  - 一方向性は$O(2^n)$で, 衝突困難性は$O(\sqrt{2^n})$で計算可能
  - nについての多項式時間でできない

ランダムオラクルモデル

- nビットハッシュ関数を理想化したもの
- 内部動作
  - 巨大なテーブルTをみなで共有する. 入力mに対して
    - 初めてのmなら乱数$h \in Y$を選び$T[m] = h$を登録しhを出力
    - Tに登録されたものなら$T[m]$を出力
  - 巨大なテーブルが必要なので効率的でない



- 実際のハッシュ関数(SHA-256やSHA-3)を理想のハッシュ関数とみなして安全性証明をすること

楕円曲線を用いた署名

- ECDSA(Elliptic Curve Digital Signature Algorithm)
  - 鍵生成 : $s \leftarrow \mathbb{F}_p$ 署名鍵, $Q := sP$ 検証鍵
  - 署名 : $k \leftarrow \mathbb{F}_p$をとり$kP$の$x$座標を$r$とする. $\sigma := \left(r, \frac{H(m)+sr}{k}\right)$
  - 検証 : 署名$\sigma := (r, t)$に対して$R := \frac{H(m)P+rQ}{t}$として
    Rのx座標がrに等しければ受理
    - 正当性 : $R = \frac{(H(m)P+rsP)k}{H(m)+sr} = kP$. 「秘密鍵で暗号化」ではない
- EdDSA(Edwards curve DSA)
  - Weierstrass曲線ではなくEdwards曲線(Montgomery型)を利用
  - ECDSAに比べて高速(TLS1.3~)
- Schnorr署名
  - DLPが困難でランダムオラクルモデルのもとで安全性証明あり

準同型暗号HE(homomorphic encyrption)

- 暗号文のまま操作できる暗号
- Lifted-ElGamal暗号は加法準同型暗号
  - $c_1 \coloneqq (S_1, T_1) \coloneqq Enc(m_1; r_1)$, $c_2 \coloneqq (S_2, T_2) \coloneqq Enc(m_2; r_2)$
  - $c_1 + c_2 \coloneqq (S_1 + S_2, T_1 + T_2)$; 暗号文同士の足し算の定義
  - $$\begin{aligned} Enc(m_1; r_1) + Enc(m_2; r_2) &= (m_1 P + r_1 Q, r_1 P) + (m_2 P + r_2 Q, r_2 P) \\ &= \big((m_1 + m_2)P + (r_1 + r_2)Q, (r_1 + r_2)P\big) \\ &= Enc(m_1 + m_2; r_1 + r_2) \end{aligned}$$
    - $Enc(m) + Enc(m) = Enc(2m)$
    - $Enc(2m) + Enc(m) = Enc(3m)$
  - 暗号文の整数n倍も計算できる
  - $n \times Enc(m; r) = Enc(m; r) + \cdots + Enc(m; r) = Enc(nm; nr)$

完全準同型暗号(Gentry 2009)

- "任意"の関数はandとxorで記述できる

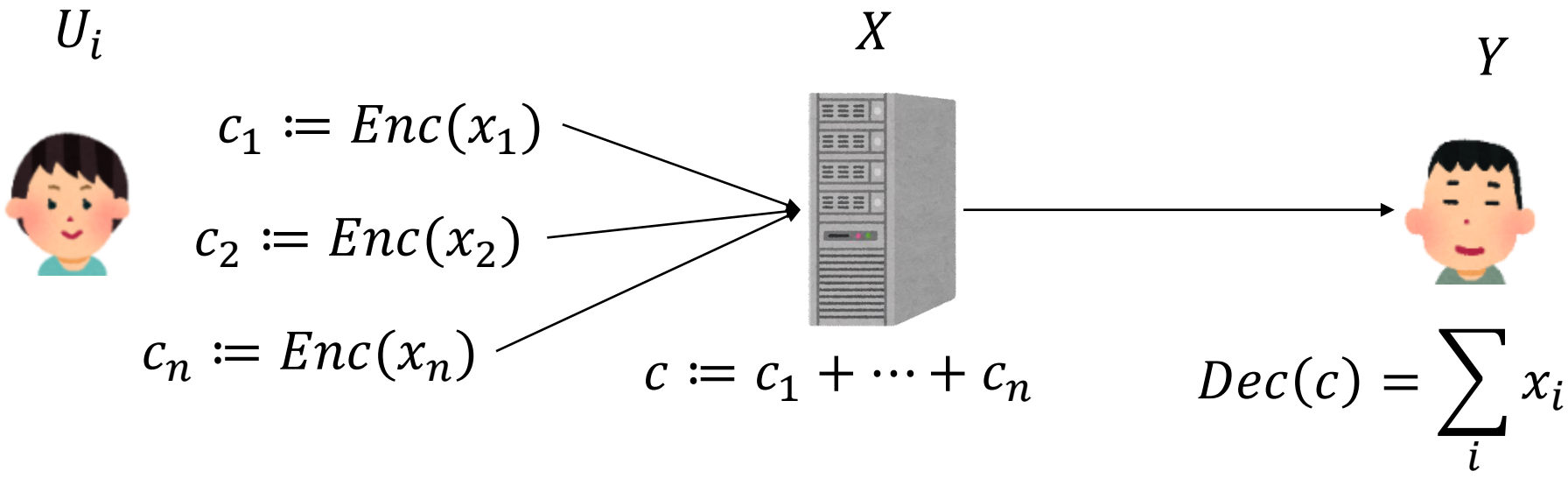
and	0	1
0	0	0
1	0	1

xor	0	1
0	0	1
1	1	0

- andは$\mathbb{F}_2$の掛け算, xorは$\mathbb{F}_2$の足し算
- $\mathbb{F}_2$における加算と乗算の暗号文の計算ができれば
  任意の関数の計算ができる
- $Enc\big(f(x)\big) = f\big(Enc(x)\big)$
- 多次元の格子を用いた暗号(格子暗号)ベースが主流
  - 暗号文や計算時間が比較的大きい

簡単な投票

- U_i ; 投票者, X ; 集計サーバ, Y : 結果を知る人
  - Yが公開鍵を公開する
    - $x_i \in \{0,1\}$

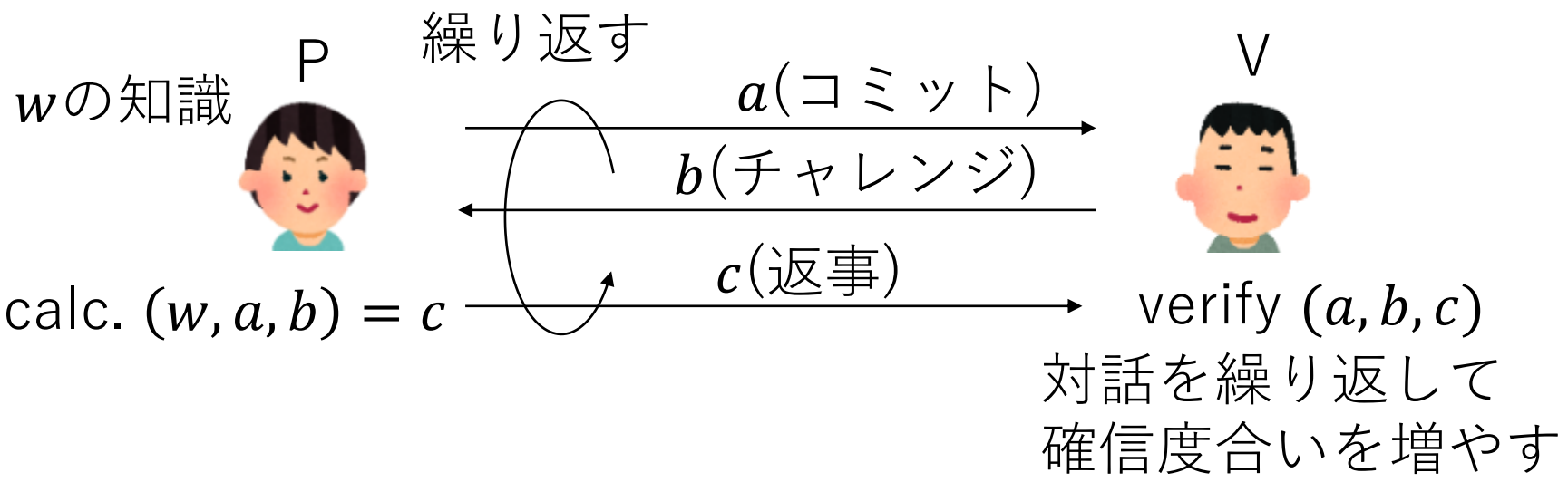


- もし悪意あるU_{i_0}が$x_{i_0} = 10$を暗号化して送ったら?
  - 正しい暗号文と不正な暗号文の区別をつけられない

知識のゼロ知識証明

CybozuLabs

- Goldwasser, Micali, Rackoffが定式化(1985)
  - ZKP(Zero knowledge Proof)
- 証明者Proverと検証者Verifierがいる
  - 対話的に情報をやりとりすることで
    Pがある命題Xを成立させるwを知っていることをVに示す
  - ただしwがXを満たすこと以外の情報はVに与えない
- Σプロトコル

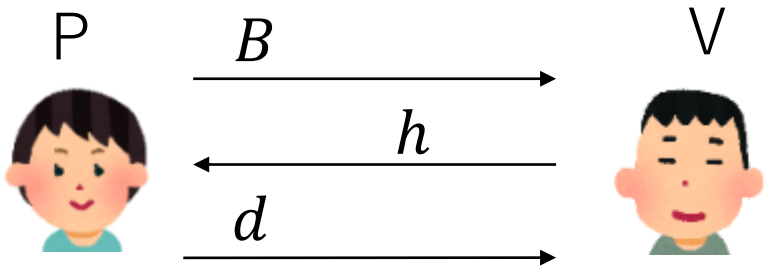


ZKPの性質

- 命令Xを成り立たせるwについて
- 完全性
  - 証明者Pがwを知っていれば検証者Vは必ず納得する
- 健全性
  - Vが納得すれば無視できる確率を除いてPはwを知っている
- ゼロ知識性
  - VはPがwを知っていること以上のwの情報は得られない
- 対話的
  - PとVのやりとりは対話的に行われる
- 非対話ゼロ知識証明
  - ランダムオラクルモデルのもとで対話的なZKPを非対話にする一般的な構成方法がある(Fiat-Shamir変換)

DLPのZKP

- $P, Q = wP$, Pはwを知っていることをVに示す
  - P, Vで(P, Q)を共有
  - P→V ; $b \in \mathbb{F}_p$を選んで$B \coloneqq bP$を送る
  - V→P ; $h \in \mathbb{F}_p$を選んで送る
  - P→V ; $d \coloneqq b + wh$を送る
  - V ; $dP - hQ \underset{?}{=} B$を確認



- 完全性
  - $dP - hQ = (b + wh)P - h(wP) = bP = B$
- 健全性
  - 同じBに対し異なるh, h'で受理すれば

$$B = dP - hQ = d'P - h'Q$$

    - $Q = \frac{d'-d}{h'-h}P$; DLPが解けている

$Dec(c) \in \{0,1\}$の非対話ZKPの構成例

CybozuLabs

- $X \coloneqq (Q, P)$, $A \coloneqq (A_0, A_1) = Enc(m; r)$
  - $m = 0$なら$A = r(Q, P)$となるrがある
  - $m = 1$なら$A' \coloneqq (A_0 - P, Q) = r(Q, P)$となる$r$がある
- 証明者 ($H: \{0,1\}^* \to \mathbb{F}_p$; ハッシュ関数)
  - $m = 0$のとき$d', h', b \leftarrow \mathbb{F}_p$をとる
    - $B' \coloneqq d'X - h'A'$, $B \coloneqq bX$, $h \coloneqq H(X, A, B, B') - h'$, $d \coloneqq b + hr$
  - $m = 1$のとき$d, h, b' \leftarrow \mathbb{F}_p$をとる
    - $B \coloneqq dX - hA$, $B' \coloneqq b'X$, $h' \coloneqq H(X, A, B, B') - h$, $d' \coloneqq b' + h'r$
  - 証明$\pi \coloneqq (d, h, d', h')$を検証者に送る
- 検証者
  - 与えられたX, A, πに対して$B \coloneqq dX - hA, B' \coloneqq d'X - h'A'$
  - $H(X, A, B, B') = h + h'$が成り立てば受理

証明の概要

- 完全性
  - $m = 0$なら$B = bX = (d - hr)X = dX - h(rX) = dX - hA$
  - $m = 1$なら$B' = b'X = (d' - h'r)X = d'X - h'(rX) = d'X - h'A'$
- ゼロ知識性
  - $m = 0$なら$d', h', b \leftarrow \mathbb{F}_p$より$d = b + hr$もランダム
- 健全性
  - $A = (a_0 Q, a_1 P)$, $B = (b_0 Q, b_1 P), B' = (b_0' Q, b_1' P)$とすると
    $B = dX + hA$より$b_0 = d + ha_0, b_1 = d + ha_1$
    よって$b_1 = b_0 + h(a_0 - a_1)$
  - $B' = d'X + h'A'$より$b_0{}' = d' + h'(a_0 - 1), b_1{}' = d' + h'a_1$
    よって$b_1' = b_0' + h'(a_0 - 1 - a_1)$
  - $H\big(Q, P, a_0 Q, a_1 P, b_0 Q, \big(b_0 + h(a_0 - a_1)\big)P, b_0' Q, \big(b_0' + h'(a_0 - 1 - a_1)\big)P\big) = h + h'$; $a_0 - a_1 \notin \{0,1\}$なら見つけるのは困難

ZKP証明系の比較

- zk-SNARK ; ペアリング(後述)ベース. 証明の検証が高速
- STARK ; 信頼のおける第三者機関が不要
- Bulletproof ; ECDLPに基づいた方式

Proof system	Σ-protocol	zk-SNARK	STARK	Bulletproof
対話・非対話	yes	no	no	no
証明サイズ	long	short	shortish	short
証明者の時間	$O(n)$	quasilinear	quasilinear big memory	$O(n)$
検証時間	$O(n)$	efficient	poly-log	$O(n)$
第三者機関	no	required	no	no
実用的か?	yes	yes	not quite	yes

- ブロックチェーンの秘匿系プロジェクトで注目度が高い
- 例 : 範囲証明, $a \in [0,2^n)$であることを示す
  - aの2進数表記$\sum a_i 2^i (a_i \in \{0,1\})$で$a_i \in \{0,1\}$をZKPで示す

Weilペアリング

- $E/\mathbb{F}_p$; 楕円曲線, $\gcd(r,p)=1$となるr
- $G \coloneqq E[r] = \{P \in E(\overline{\mathbb{F}_p}) | rP = 0\}$
  - $G \subseteq E(\mathbb{F}_{p^k})$となる$k$を選ぶ
  - $\mu_r \coloneqq \{x \in \mathbb{F}_{p^k} | x^r = 1\}$
- $e: G \times G \to \mu_r$; Weilペアリング(双線形写像)
  - $e(P_1 + P_2, Q) = e(P_1, Q)e(P_2, Q)$
  - $e(P, Q_1 + Q_2) = e(P, Q_1)e(P, Q_2)$
  - $e(aP, bQ) = e(P, Q)^{ab}$ for $a, b, \in \mathbb{Z}$
- GのDLPとμ_rのDLP
  - P, aP ; given, take $Q \in G$. $g \coloneqq e(P,Q)$, $h \coloneqq e(aP,Q) = g^a$
  - もし(g,h)のDLPが解けるならGのDLPが解ける(MOV還元)
  - 暗号ではどちらも解けないものが使われる(安全性仮定は多種)

三人で鍵共有

- DH鍵共有は二人の間で秘密の値を共有
- 三人で同じことができるか?
  - Weilペアリングを使うと可能(2000年Joux)
- アリス、ボブ、クリス
  - アリスはaを選んでaPをボブとクリスに送る
  - ボブはbを選んでbPをアリスとクリスに送る
  - クリスはcを選んでcPをアリスとボブに送る
- 共有
  - アリス ; $e(bP, cP)^a = g^{abc}$
  - ボブ ; $e(aP, cP)^b = g^{abc}$
  - クリス ; $e(aP, bP)^c = g^{abc}$

n人での鍵共有は?

- DHP相当問題が困難な群G, G_Tと多重線形写像

  $$e: G \times \cdots \times G \to G_T$$

  $$e(a_1P, a_2P, a_3P, \ldots, a_nP) = e(P, \ldots, P)^{a_1 a_2 \ldots a_n}$$

  があれば可能

  - 未解決
    - 存在すれば面白い暗号プロトコルが沢山できる
  - 今のところそんな写像は構成できていない
  - 否定的な結果がいくつか知られている

ペアリングの改良

- 様々な計算効率のよいペアリングが提案されている
- BLS12 ; $e: G_1 \times G_2 \rightarrow G_T$
  - 128ビットセキュリティで最も効率のよいものの一つ
  - G_1 ; $\mathbb{F}_p$上定義された楕円曲線r等分点の集合
  - G_2 ; $\mathbb{F}_{p^2}$上定義された楕円曲線r等分点の集合
  - $G_T \coloneqq \left\{x \in \mathbb{F}_{p^{12}} \middle| x^r = 1\right\}$
  - $e(aP, bQ) = e(P, Q)^{ab}$

- Weilペアリングは対称ペアリング
- BLS12は非対称ペアリング

ペアリングを用いた2LHE(2018)

- 「2L」は2次式の計算ができる(level 2)という意味
- $e: G_1 \times G_2 \to G_T$; 非対称ペアリング
  - $G_i := \langle P_i \rangle$; 生成元$(i = 1,2)$, $|G_i| = |G_T| = r$
  - G_Tを加法群表記する, $R := e(P_1, P_2)$はG_Tの生成元
- Enc_i ; G_iに関するLifted-ElGamal暗号
  - $s_i \in \mathbb{F}_r$; 秘密鍵, $Q_i := s_i P_i$; 公開鍵
  - $c_i := Enc_i(m_i; r_i) = (m_i P_i + r_i Q_i, r_i P_i) = ((m_i + r_i s_i) P_i, r_i P_i)$
- 暗号文c_1とc_2を「掛ける」
  - $c_i := (S_i, T_i)$として
  - $c_1 \times c_2 := \big(e(S_1, S_2), e(S_1, T_2), e(T_1, S_2), e(T_1, T_2)\big)$
    $= \big((m_1 + r_1 s_1)(m_2 + r_2 s_2) R, (m_1 + r_1 s_1) r_2 R, (m_2 + r_2 s_2) r_1 R, r_1 r_2 R\big) \in R^4$

乗算後の暗号文の復号

- $c := (A_0, A_1, A_2, A_3)$に対して

$$dec(c) := A_0 - s_2 A_1 - s_1 A_2 + s_1 s_2 A_3$$
$$Dec(c) := DLP_R(dec(c))$$

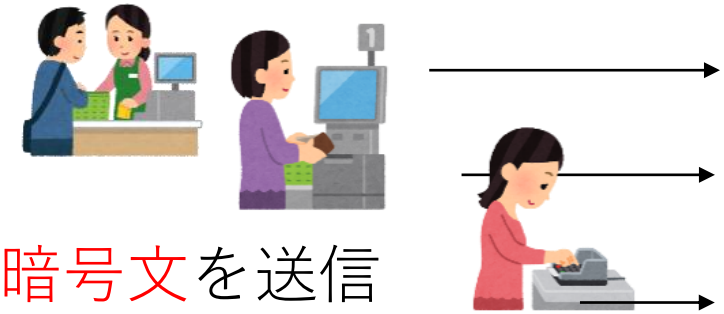
- $dec(c_1 \times c_2) = (m_1 + r_1 s_1)(m_2 + r_2 s_2)R - s_2(m_1 + r_1 s_1) r_2 R - s_1(m_2 + r_2 s_2) r_1 R + s_1 s_2 r_1 r_2 R = m_1 m_2 R$
  - $Dec(c_1 \times c_2) = m_1 m_2$
- 乗算後の暗号文は成分ごとの加算により「足せる」
- 内積演算
  - $x := (x_1, \ldots, x_n)$, $y := (y_1, \ldots, y_n)$に対して

$$Enc_1(x) := \left(Enc_1(x_1), \ldots, Enc_1(x_n)\right),$$
$$Enc_2(y) := \left(Enc_2(y_1), \ldots, Enc_2(y_n)\right)$$

  - $Enc_1(x) Enc_2(y) = \sum Enc_1(x_i) Enc_2(y_i) = Enc(\sum x_i y_i)$

応用例

- 暗号文のままデータの平均や分散を計算可能
- クロス集計の秘匿化
  - https://herumi.github.io/she-wasm/cross-demo-ja.html



暗号文を送信

	テレビ	マンガ	小説	カメラ
A	＊	＊	＊	＊
B	＊	＊	＊	＊
C	＊	＊	＊	＊
…	＊	＊	＊	＊

暗号文のままクロス集計

	マンガ購入	購入せず
カメラ購入	＊	＊
購入せず	＊	＊

→

	マンガ購入	購入せず
カメラ購入	80	20
購入せず	10	30

- 加法準同型暗号+αでがんばる / 完全準同型暗号でがんばる

暗号化したままテーブル引き

CybozuLabs

- T ; 区間$[0, n)$のテーブル$(T_a \in \mathbb{Z}\ ; a = 0, \ldots, n-1)$

a ; インデックス

$c = Enc(a)$ →

← $c' = Enc(T_a)$

$Dec(c') = T_a$

0	…	a	…	$n-1$
T_0	…	T_a	…	T_{n-1}

- 2LHEを使った方法($O(\sqrt{n})$の通信量)
  - $m := \lceil \sqrt{n} \rceil$, $a = mq + r (0 \le r < m)$, δ_{ij} ; クロネッカーのデルタ
  - $Enc(a) := \left(Enc_1(\delta_{qi}), Enc_2(\delta_{rj}) \right)_{i=0,\ldots,m-1, j=0,\ldots,m-1}$; $2m$個
  - $\sum_{ij} T_{im+j} \times Enc_1(\delta_{qi}) \times Enc_2(\delta_{rj})$

$$= Enc\left(\sum_{ij} T_{im+j} \delta_{qi} \delta_{rj} \right) = Enc(T_{qm+r}) = Enc(T_a)$$

  - https://ppdm.jp/ot/
  - $n = 100$万ならWebAssembly実装でも暗号化が1秒未満

BLS署名

CybozuLabs

- ペアリングを用いた署名
  - $e: G_1 \times G_2 \to G_T$; ペアリング $G_i = \langle P_i \rangle$; 生成元
  - $H_2: \{0,1\}^* \to G_2$; ハッシュ関数
- 鍵生成
  - $s \leftarrow \mathbb{F}_p$; 署名鍵(秘密鍵), $S \coloneqq sP_1$; 検証鍵(公開鍵)
- 署名
  - メッセージmに対して$\sigma \coloneqq Sign(s, m) \coloneqq sH_2(m)$
- 検証
  - メッセージmと署名σに対して
    $e(P_1, \sigma) = e(S, H_2(m))$が成立するなら$Verify(S, m, \sigma) = 1$
- 正当性 : validな署名なら
  - $e(P_1, \sigma) = e\big(P_1, sH_2(m)\big) = e\big(sP_1, H_2(m)\big) = e(S, H_2(m))$

BLS署名の特長

CybozuLabs

- 線形性
  - n個の署名鍵s_i, 検証鍵S_i, mに対するvalidな署名σ_i,
    任意の$a_1, \ldots, a_n \in \mathbb{F}_p$について
    $\sigma \coloneqq \sum_i a_i \sigma_i$は$S \coloneqq \sum_i a_i S_i$に対してvalidな署名
  - 何故なら$e(P_1, \sigma) = e(P_1, \sum a_i \sigma_i) = e\left(P_1, \sum a_i s_i H_2(m)\right)$
    $= e\left(\sum a_i s_i P_1, H_2(m)\right) = e\left(\sum a_i S_i, H_2(m)\right) = e(S, H_2(m))$

- 集約署名
  - n個の署名の検証コストを減らす
  - $\{\sigma_i, S_i\} \to (\sigma \coloneqq \sum \sigma_i, S \coloneqq \sum S_i)$
  - 注意 : Rogue-attackというのに気をつける
    - 攻撃者jが$\sigma' \coloneqq \sigma_0 - \sum_{i \neq j} \sigma_i$を作ると$\sigma = \sigma_0$で制御可能
    - 署名者が正しい署名鍵を持っていることを検証する必要あり

秘密分散

- 秘密情報$s \in \mathbb{F}_p$をn人でshareする$(y_1, \ldots, y_n)$
  - そのうちk人が集まって$Y = \{y_{i_1}, \ldots, y_{i_k}\}$を集めると$s$をrecover
  - e.g., 社長一人で秘密鍵を持つのではなく取締役で情報分散
- Shamirの秘密分散
  - 信頼のおける機関
    - $k-1$次多項式$f(x) = s_0 + s_1 x + \cdots + s_{k-1} x^{k-1}$
      - $s_1, \ldots, s_{k-1}$; random
  - share
    - 各ユーザ$U_i (i = 1, \ldots, n)$に$y_i \coloneqq f(i)$を秘密に配付する
  - recover
    - $\{y_1, \ldots, y_n\}$の任意のk個の部分集合$Y = \{y_{i_1}, \ldots, y_{i_k}\}$から$f(x)$は一意に確定するので$f(0) = s_0$が分かる
    - $f(0)$はYの線形和で計算可能(Lagrange補間)

秘密分散+BLS署名

- 信頼のおける機関A
  - $k-1$次多項式$f(x)$を作る. マスター検証鍵$S \coloneqq f(0)P_1$を公開
  - 各ユーザに署名鍵$s_i \coloneqq f(i)$を配付し検証鍵$S_i \coloneqq f(i)P_i$を公開
  - Aは生成した多項式を忘れる
- 各ユーザの署名
  - mに対して署名$\sigma_i \coloneqq Sign(s_i, m) = s_i H(m)$を生成
  - S_iで正しさを検証可能
- recover
  - 任意のk個の署名σ_iからマスター署名$\sigma = f(0)H(m)$を復元
  - マスター検証鍵Sで検証可能
- DKG(distributed key generation)
  - Aを仮定せずに各ユーザに署名鍵を配付する方法
- 署名を多数決に利用可能

量子計算機

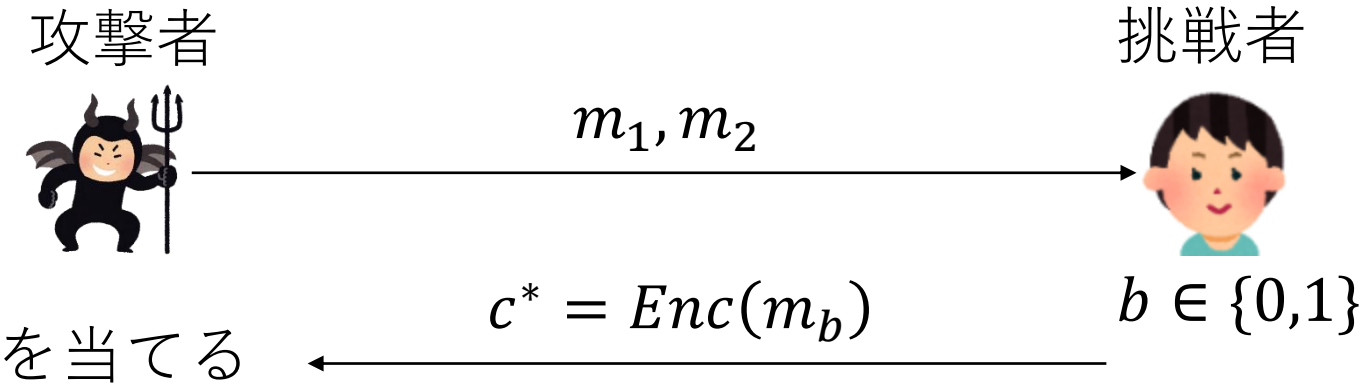
- 量子の性質を利用して計算する計算機
  - DLPを効率よく解くアルゴリズムが知られている
    - D-Wave(量子やきなまし型)では解けない(今のところ)
  - 実際の暗号を解くものが実用化されるのは数十年後?
- 耐量子計算機暗号
  - 量子計算機でも解けない暗号
  - 現在の計算機で実行可能
  - 格子暗号、楕円曲線の同種写像を用いた暗号などが候補
- cf. 量子暗号(=量子暗号通信, 量子鍵配送)
  - 量子の性質を利用して乱数を安全に共有する(盗聴不可能)
  - 現在の計算機で実現不可能
  - データを暗号化して保存したり計算したりは出来ない

参考文献

- 入門書
  - 『暗号技術のすべて』(IPUSIRON) 2017
    - 暗号全般の入門書, 暗号理論のはなし
  - 『クラウドを支えるこれからの暗号技術』2015
    - https://herumi.github.io/ango/ ; ペアリング暗号全般
- 専門書
  - 『公開鍵暗号の数理』(森山大輔, 西巻陵, 岡本龍明) 2011
    - 暗号理論のモデル, 安全性証明
  - 『Guide to Pairing-based cryptography』(Mrabet, Joye) 2017
  - 『現代暗号の誕生と発展』(岡本龍明) 2019
    - ブロックチェーン, zk-SNARK, 格子暗号, 完全準同型暗号
  - 『耐量子計算機暗号』(縫田光司) 2020
    - 楕円曲線の同種写像暗号, 格子暗号, 量子計算機

公開鍵暗号文の識別不可能性

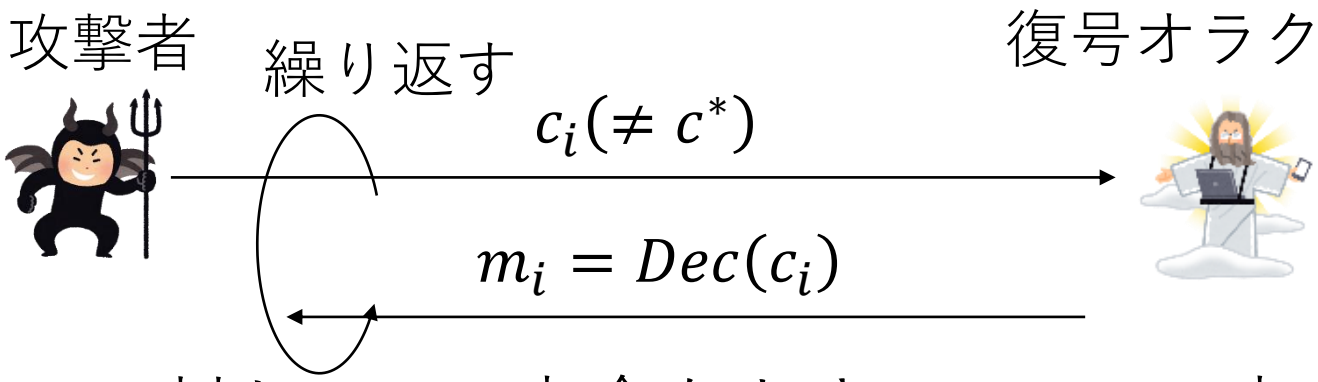
- INDゲーム(indistinguishability)
  - 攻撃者Aが2個の暗号文m_1, m_2を選び挑戦者Bに渡す
  - Bは$b = 0$ or 1を選び暗号文$c^* = Enc(m_{i_1})$をAに渡す
  - Aはcがどちらの暗号文を暗号化したものか当てれば勝ち



  - 当てられる確率Pがでたらめに当てる確率1/2とほぼ同じときIND安全という
    - セキュリティパラメータλに対して任意の$k \in \mathbb{N}$に対して
      「$n > n_k$なら$\left|P(n) - \frac{1}{2}\right| < n^{-k}$となる$n_k$が存在する」

IND-CCA2安全

- 攻撃者のレベル
  - 攻撃者は選んだ平文の暗号文を作れる
    - CPA(Chosen Plaintext Attack) ; 公開鍵暗号はいつでも
  - 攻撃者が選んだ暗号文の平文を知ることができる
    - CCA(Chosen Ciphertext Attack)
    - CCA2 ; ターゲット暗号文c^*をみてCCAができる
      - c^*はINDゲームの答えるべき暗号文
      - それ以外の暗号文の情報は全て教えてもらえるという状況



- CCA2に対してIND安全なときIND-CCA2安全という