Oblivious Verifiable Encryption: 復号が確率論的に制御可能な暗号化方式を目指して

高橋 大成[1,a)]　大塚 玲[2]　面 和成[1]

概要：本研究では，復号を確率的に制御可能な暗号方式を提案する．例として，復号できる確率を $1/100$ と設定した場合，暗号文は $1/100$ の確率で復号が可能となる．本提案は，機密性を保ちつつ一定の条件下で情報を公開できる性質をもち，かつ三者間プロトコルを可能にするため，データ共有など様々な分野への応用が可能である．例として，中央銀行が発行するデジタル通貨 CBDC において，犯罪対策のための監査機能が挙げられる．先行研究では，通常の送金は匿名であるが，一定期間内に既定値以上を送金すると，監査機関が送金内容を確認することが可能となる．しかし，マネーロンダリングなどの犯罪を目的とする者は，時間をかけて徐々に送金するか，既定値を１円でも超えなければ，監査を回避することが可能である．本提案を利用することで，(送金額)/(既定値) の確率で送金を監査対象とすることができるため，柔軟な監査を可能にし，犯罪の抜け道を塞ぐことに貢献できる．

キーワード：暗号, 秘密分散, Oblivious Transfer

Oblivious Verifiable Encryption: Toward an encryption scheme where decryption can be probabilistically controllable

Taisei Takahashi[1,a)]　Akira Otsuka[2]　Kazumasa Omote[1]

Abstract: This study proposes an encryption scheme that allows for probabilistic decryption control. For example, if the probability of successful decryption is set to $1/100$, the ciphertext can be decrypted with a probability of $1/100$. This proposal enables information to be disclosed under specific conditions while maintaining confidentiality. It also supports a three-party protocol, which makes it applicable in various fields, such as data sharing. An example is the Central Bank Digital Currencies (CBDCs) audit function, which central banks issue to prevent crime. In previous research, while ordinary transactions are anonymous, the auditing authority can review the transaction details if the transaction amount exceeds a specified value within a certain period. However, criminals engaged in money laundering can avoid detection by transferring small amounts over time or by keeping transactions just below the threshold. Using our proposed method, transactions can be made subject to audit with a probability of (transaction amount)/(specified value), allowing for more flexible auditing and helping to close loopholes that criminals might exploit.

Keywords: Cryptography, Secret Sharing, Oblivious Transfer

1. はじめに

従来の現代暗号の要件として，平文が暗号化されると，復号する際には適切な鍵を利用することで，復号できる確率は 1 であることが必須となっている．従来の現代暗号は

[1] 筑波大学
University of Tsukuba
[2] 情報セキュリティ大学院大学
Institute of Information Security
[a)] taisei.takahashi@risk.tsukuba.ac.jp

「平文のメッセージを隠す」ことに焦点が与えられてきたため，復号させても良い相手，つまり適切な鍵を持つ人間には，隠していた平文を復号させることが必要であったためである．

高い安全性を提供する反面，利便性に関しては制約が生じる．従来の暗号方式では，特定の利用者や状況に応じた，柔軟な利便性の提供が困難である．例として，機密情報を含むデータに対して，特定の時間帯や特定のネットワーク環境下でのみ復号が成功するようにすることや，異なるアクセス権限を持つ利用者に対して異なる復号確率を割り当てるような設定することでセキュリティレベルを細かく調整する，といった，柔軟な利便性の提供が困難である．

本研究では，復号確率が制御可能な暗号方式を提案する．

先行研究として Oblivious Transfer (OT) という暗号技術が存在する．送信者から受信者に，メッセージを確率的に送信可能としている．送信者が複数のメッセージを持ち，受信者がその中から特定のメッセージを選んで受け取るが，送信者はどのメッセージが選ばれたかを知らないというプロトコルである．OT は選択のプライバシーを保護することに特化しており，送信者が受信者の選択を知らないままデータを送信することができる．

一方，本研究は，暗号化されたデータに対して復号が成功する確率を調整できるというものである．例えば，復号の成功確率を特定の時間帯や特定のユーザに応じて設定することが可能である．本研究は，安全性と利便性のバランスを取ることを目的としており，重要な情報を必要なタイミングで必要な人にのみ提供し，その他の状況ではデータのセキュリティを確保することを目的としている点に，独自性が存在する．

2. 準備

構成として， Kate らの Polynomial Commitments (PC) [1] を利用する． ある多項式 $f(x)$ を明かさずとも，多項式の特定の値 $f(i)$ が正しいことを検証可能にする暗号技術である．また，PC では，秘密鍵を知らない第三者でも，公開鍵のみで多項式の特定の値 $f(i)$ が正しいことを検証可能にするウィットネスを作成することが可能となる．

PC では，公開鍵内のパラメータによって，検証可能な多項式の次数に上限を設けることが可能となるため，定められた確率で復号ができることを強制的に設定可能となる．

2.1 Polynomial Commitments [1]

Definition 1. Polynomial Commitments は，次の 6 つのアルゴリズムから構成される: $\mathsf{Setup}, \mathsf{Commit}, \mathsf{VerifyPoly}, \mathsf{CreateWitness}, \mathsf{VerifyEval}$.

- $\langle \mathsf{pk}, \mathsf{sk} \rangle \leftarrow \mathsf{Setup}(1^\kappa, t)$
  - t: 次数 t 以下の多項式にコミットする．
- $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{pk}, f(x))$
- $1/0 \leftarrow \mathsf{VerifyPoly}(\mathsf{pk}, c, f(x))$
- $\langle i, f(i), \omega_i \rangle \leftarrow \mathsf{CreateWitness}(\mathsf{pk}, f(x), i)$
- $1/0 \leftarrow \mathsf{VerifyEval}(\mathsf{pk}, c, i, f(i), \omega_i)$

2.1.1 $\mathsf{PolyCommit}_{\mathbf{DL}}$ の構成

Polynomial Commitments の構成として，離散対数 (DL) 問題仮定下の構成 $\mathsf{PolyCommit}_{\mathbf{DL}}$ を採用する．

- $\langle \mathsf{pk}, \mathsf{sk} \rangle \leftarrow \mathsf{Setup}(1^\kappa, t)$

$$\begin{aligned} \mathsf{sk} &= \alpha \\ \mathsf{pk} &= (\mathcal{G}, g, g^\alpha, \cdots, g^{\alpha^t}) \\ \mathcal{G} &= (e, \mathbb{G}, \mathbb{G}_T) \\ e &: \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T \end{aligned} \tag{1}$$

- $\mathsf{com} \leftarrow \mathsf{Commit}(\mathsf{pk}, f(x))$

$$\begin{aligned} &\mathsf{com} = g^{f(\alpha)} \in \mathbb{G} \\ &f(x) \in \mathbb{Z}_p[x], \quad \text{degree} \leq t \\ &\text{For } f(x) = \sum_{j=0}^{deg(f)} f_j x^j,\ \mathsf{com} = \prod_{j=0}^{deg(f)} (g^{\alpha^j})^{f_j} \end{aligned} \tag{2}$$

- $1/0 \leftarrow \mathsf{VerifyPoly}(\mathsf{pk}, c, f(x))$

$$c \stackrel{?}{=} g^{f(\alpha)} \tag{3}$$

output 1 if $c = \prod_{j=0}^{deg(f)} (g^{\alpha^j})^{f_j}$ for $f(x) = \sum_{j=0}^{deg(f)} f_j x^j$.

- $\langle i, f(i), \omega_i \rangle \leftarrow \mathsf{CreateWitness}(\mathsf{pk}, f(x), i)$

$$\begin{aligned} \omega_i &= g^{\psi_i(\alpha)} \\ \psi_i(x) &= \frac{f(x) - f(i)}{(x - i)} \end{aligned} \tag{4}$$

- $1/0 \leftarrow \mathsf{VerifyEval}(\mathsf{pk}, c, i, f(i), \omega_i)$

$$e(c, g) \stackrel{?}{=} e(\omega_i, g^\alpha / g^i) \cdot e(g, g)^{f(i)} \tag{5}$$

3. Oblivious Encryption Scheme

本セクションでは, 我々の提案する Oblivious Encryption Scheme の定義を記載する

Definition 2. (Oblivious Encryption Scheme). Oblivious encryption scheme OE は 3 つの効率的なアルゴリズムで構成される：$\mathsf{OE} = (\mathsf{OE.KGen}, \mathsf{OE.Enc}, \mathsf{OE.Dec}, \mathsf{OE.VerifyCT})$

- $\mathsf{KGen}(1^\lambda) \to (\mathsf{pk}, \{\mathsf{sk}_1, \mathsf{sk}_2, \cdots, \mathsf{sk}_n\})$: 確率的鍵生成アルゴリズムは，セキュリティパラメータ 1^λ を入力として受け取り，公開鍵と秘密鍵のペア $(\mathsf{pk}, \mathsf{sk})$ を出力する．
- $\mathsf{Enc}(\mathsf{pk}, m) \to (c, \omega)$: 確率的暗号化アルゴリズムは，

公開鍵 pk とメッセージ $m \in \mathcal{M}\lambda$ を入力として受け取り，暗号文 $c \in \mathcal{C}\lambda$ を出力する．

- $\mathsf{VerifyCT}(\mathsf{pk}, c, \omega) \to 1/0$: 決定論的な検証アルゴリズムは，公開鍵 pk と暗号文 c を入力として受け取り，1 または 0 を出力する．
- $\mathsf{Dec}(\mathsf{sk}_i, c, \omega) \to m$ or $\perp$: 決定論的な復号アルゴリズムは秘密鍵 sk と暗号文 c を入力として受け取り，メッセージ m または $\perp$ を出力する．

3.1 Oblivious Verifiable Encryption の安全性定義

本スキームは下記の安全性をもつ：

Definition 3. (Correctness). すべての $\lambda \in \mathbb{N}$, すべての $m \in \mathcal{M}$, すべての $\mathsf{pk} \in \{0,1\}^*$, およびすべての $\mathsf{sk} \in \{0,1\}^*$ に対して以下の条件が成り立つとき，OE を *correct* と定義する：

$$\Pr\left[\begin{array}{l|l} \mathsf{OE.Dec}(\mathsf{sk}, c) & (\mathsf{pk}, \{\mathsf{sk}_1, \cdots, \mathsf{sk}_n\}) \\ = m & \leftarrow \mathsf{OE.KGen}\left(1^\lambda\right) \\ & \mathsf{sk} \leftarrow\!\!\$\ \{\mathsf{sk}_1, \cdots, \mathsf{sk}_n\} \\ & (c, \pi) \leftarrow \mathsf{OE.Enc}(\mathsf{pk}, m, \omega) \\ & \mathsf{OE.VerifyCT}(\mathsf{pk}, c, \pi) = 1 \end{array}\right] = \frac{1}{n}$$

ここで，確率は KGen と Enc によって生成されたランダムなコインに基づく．

Definition 4. (Completeness). すべての $\lambda \in \mathbb{N}$, すべての $m \in \mathcal{M}$, およびすべての $\mathsf{pk} \in \{0,1\}^*$ に対して以下の条件が成り立つとき，OE を *complete* と定義する：

$$\Pr\left[\begin{array}{l} \mathsf{OE.VerifyCT}(\mathsf{pk}, c, \pi) = 1 \\ \qquad \left| \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{OE.KGen}\left(1^\lambda\right) \\ (c, \pi) \leftarrow \mathsf{OE.Enc}(\mathsf{pk}, m, \omega) \end{array} \right. \end{array}\right] = 1.$$

Definition 5. (Soundness) すべての $\lambda \in \mathbb{N}$, すべての $m \in \mathcal{M}$, およびすべての $\mathsf{pk} \in \{0,1\}^*$ に対して以下の条件が成り立つとき，OE を *sound* と定義する：

$$\Pr\left[\begin{array}{l} \mathsf{OE.VerifyCT}(\mathsf{pk}, c, \pi') = 1 \\ \qquad \left| \begin{array}{l} (\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{OE.KGen}\left(1^\lambda\right) \\ (c, \pi') \leftarrow \mathsf{OE.Enc}(\mathsf{pk}, m) \end{array} \right. \end{array}\right] < \mathsf{negl}(1^\lambda).$$

Definition 6. (Authenticated IND-CCA Security for OE). すべての効率的な攻撃者 $\mathcal{A} = (\mathcal{A}_1, \mathcal{A}2)$ に対して，アドバンテージ $\mathsf{Adv}^{\text{a-ind-cca}}\mathsf{OE}, \mathcal{A}(\lambda)$ が無視できる場合，

$$\mathsf{OE} := (\mathsf{OE.KGen}, \mathsf{OE.Enc}, \mathsf{OE.Dec}, \mathsf{OE.VerifyCT})$$

を (認証付き) IND-CCA セキュアと定義する．ここで、アドバンテージは次のように定義する：

$$\mathsf{Adv}^{\text{a-ind-cca}}_{\mathsf{OE},\mathcal{A}}(\lambda) := 2 \cdot \Pr\left[\mathsf{Exp}^{\text{a-ind-cca}}_{\mathsf{OE},\mathcal{A}}(\lambda)\right] - 1.$$

4. Construction

Receiver(R), Encryptor(E), Verifier(V)

(1) R は，次数がそれぞれ $k-1$ である $f(x), h(x)$ の 2 つの多項式，コミットメント，およびウィットネスを用意する．ここで， $f(0)$ にメッセージを設定する．

(2) R は V に対し， $i = 1, \cdots, n$ で，それぞれ $f(i), h(i)$ のどちらかの値を，1-out-of-2 Oblivious Transfer で送信する． また，以下の値を送信する：

$$(\mathsf{com}'_f, \mathsf{com}'_h), \omega_i^f, g' \tag{6}$$

ここで， $g' = g^{r_i}, \mathsf{com}'_f = g^{r_i f(\alpha)}$, for $i = 1, \cdots, n$.

(3) V は，受け取った値が確かに $f(x), h(x)$ の 2 つの多項式のいずれかの値であることを，以下の通り検証する:

$$\begin{cases} e(\mathsf{com}'_f, g') = e(\omega_i'^f, g'^\alpha / g'^i) \cdot e(g', g')^{f(i)} \\ e(\mathsf{com}'_g, g') = e(\omega_i'^f, g'^\alpha / g'^i) \cdot e(g', g')^{h(i)}, \end{cases} \tag{7}$$

ただし，どちらの多項式の値であるかを知ることはできない．

(4) V は，検証にパスすれば，受け取った値を R に送信する

(5) R は, 秘密鍵 α を利用して, 受け取った値が $f(x), g(x)$ のどちらの値であるかを確認する

(6) R は，k 個以上の $f(x)$ の値を得ることができれば，再構成し，$f(0)$ を得ることができる

5. Application

本研究を利用することで，デジタル通貨の柔軟な監査方式の実現などに応用が見込まれる．各国の中央銀行が発行する法定通貨建てのデジタル通貨 CBDC (Central Bank Digital Currency) では，Bank と呼ばれる機関が担保金を基にコイン (デジタル通貨) を発行し，利用者間で送金が行われる．最終的に，コインを所持する利用者がコインを Bank に戻すと，代わりに担保金を受け取ることができる仕組みである．

先行研究として，Wüst ら [2] は，原則的に送受信者は匿名であるが，一定期間内に既定値以上の送金を行うと，監査機関が身元を開示できる仕組みを設けた．しかし，マネーロンダリング等の犯罪を目的とする者は，時間をかけて少額を大量に送金するか，あるいは，送金額が比較的高額であっても既定値よりも一円でも小さければ，この監査を回避することが可能である. 仮に 100 万円以上の送金を監査対象としていても，99 万円の送金は対象とすることはできない．本研究を利用することで，送金額に対して確率的（例：$^{99}/_{100}$）に身元が開示される可能性をもつ，柔軟な監査方式を実現することが可能である．

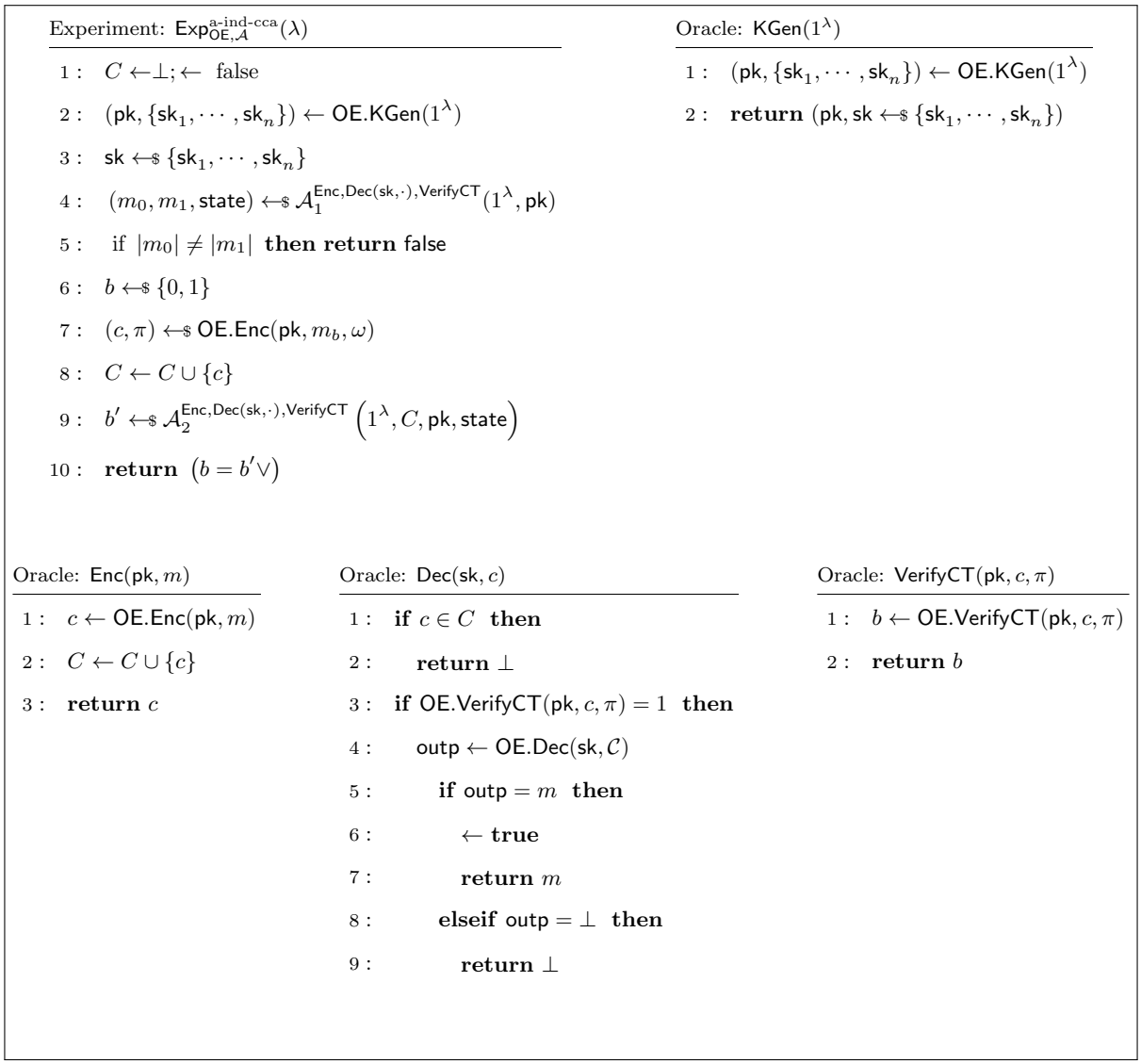

図 1 IND-CCA game of Oblivious Encryption Scheme

6. Conclusion

本研究で提案している復号確率が制御可能な暗号方式は，把握している限り，過去に提案されたことはない．

暗号文が確かに公開鍵で暗号化されており，秘密鍵で復号できるかを第三者で検証できるようにするか等，三者間プロトコルを実現している．それにより，ネットワークや時間帯等に応じて，データの共有を動的に制御でき，データ共有の利便性向上につながると考えている．

参考文献

[1] Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg. Constant-size commitments to polynomials and their applications. In Advances in Cryptology - ASIACRYPT 2010, pages 177–194. Springer Berlin Heidelberg.

[2] Karl Wüst, Kari Kostiainen, Noah Delius, and Srdjan Capkun. Platypus: A Central Bank Digital Currency with Unlinkable Transactions and Privacy-Preserving Regulation. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security, CCS '22, pages 2947–2960, New York, NY, USA, November 2022. Association for Computing Machinery.