Computer Security Symposium 2021
26 - 29 October 2021

レンジクエリに対するノイズ付きボリューム漏洩攻撃

小谷 俊輔[1,a)]　國廣 昇[2]

概要：暗号化データベースに対するレンジクエリの応答ボリュームサイズを観測することにより，データベース全体のヒストグラムを復元する攻撃が提案されている．この攻撃では，固定の窓幅 b に対して，幅 b 以下の全てのクエリ対する応答ボリュームサイズが得られるという仮定のもとで復元を行う．ボリュームサイズに偏りが大きい状況で，ボリュームサイズにノイズが全く無いときには，容易にデータベース全体のヒストグラムを復元することが可能である．一方，ノイズが存在する状況は，既存研究では考慮されておれず，どの程度のノイズが乗っても復元できるかは明らかではない．本発表では，b が 2 であり，ボリュームサイズに偏りが大きいという仮定のもとで，ノイズがある場合にも有効な攻撃を提案する．提案手法では，観測したボリュームサイズをもとに，グラフ理論的なアプローチにより，ヒストグラムの復元を行う．観測したボリュームサイズからグラフを構成し，頂点を少なくとも一度通る経路を探索することにより，データ全体を復元している．さらに，人工的なデータを用いて実験を行い，観測したボリュームサイズの偏りやノイズをパラメータとしたデータベース復元率の評価を行う．

キーワード：ボリューム漏洩攻撃，レンジクエリ，ノイズ

Noisy Volume Leakage Attack against Range Queries

Shunsuke Odani[1,a)]　Noboru Kunihiro[2]

Abstract: Many attacks have been proposed to recover the histogram of the encrypted database by using the observed response volume size of range queries. While it is easy to recover the histogram under the situation that the volume size is highly biased and includes no noise, the situation with noise has never been considered in existing works. It is not clear how much noise we can deal with. This paper proposes an effective attack under the assumption that there exists noise in the observed volume size for all queries of width one or two. In the proposed method, we employ a graph-theoretic approach. We construct a graph from the observed volume size and recover the histogram by finding paths passing through the vertices at least once. We also conduct experiments to evaluate the recovery rate from artificial data and show that our proposed algorithm is effective even if there exists noise.

Keywords: Volume Leakage Attack, Range Queries, Noise

1. はじめに

データを外部サーバにアウトソーシングする際，盗聴者からだけでなくサーバ自体からも情報を隠す必要があり，その安全性評価や安全な実装法は重要な研究課題である [1–6, 10, 13]．典型的な解決策は，暗号化されたデータをサーバに保存し，クライアントは暗号化されたクエリをサーバに送信し，サーバは，その暗号化されたクエリを処理し，クライアントのみが復元可能な応答を返信する方法である．

しかし一般に，機密性と利便性はトレードオフの関係にあるため，どのような方式でも，様々な種類の情報が漏洩し，その漏洩情報を攻撃者が悪用することにより，深刻な安全性の問題となる可能性がある．近年，どのレコードがアクセスされているかのアクセスパターンを学習すること

[1] 筑波大学情報理工学位プログラム, Degree Programs in Systems and Information Engineering, University of Tsukuba
[2] 筑波大学システム情報系，University of Tsukuba
[a)] s2020576@s.tsukuba.ac.jp

により，暗号化されたデータを全て復元する攻撃が提案されている [8,11,12]．更に最近では，データベースに対するクエリの応答ボリュームサイズを利用した新しい攻撃手法が提案されている [7,9,11]．

この攻撃者は，クエリに対する応答の内部情報自身は利用せず，そのサイズのみに注目し，暗号化されたクエリに対応するレコードの個数の復元を攻撃の目的とする．この攻撃は，ボリュームサイズに偏りが大きい状況で，ボリュームサイズにノイズが全く無いときには，容易にデータベース全体のヒストグラムを復元することが可能である．一方，ノイズが存在する状況は，既存研究では考慮されておれず，どの程度のノイズが乗っても，復元できるかは明らかではない．

本研究では，これまでの既存研究では行われていないノイズが存在する状況を想定する．また，多くの既存研究と同様に，暗号化データベースへの問い合わせに対する応答の際に，何個のレコードが返されたかを攻撃者が知ることができる状況を想定する．また，攻撃者が得られるボリュームサイズは窓幅 2 以下のクエリのみであると限定し，ボリュームサイズに偏りが大きいという仮定とする．それらの状況で，データベースカウントの復元に有効な新たなノイズ付きボリューム漏洩攻撃を提案する．さらに，人工的なデータを用いて，我々の提案攻撃を実験的に検証し，観測したボリュームサイズの偏りやノイズをパラメータとしたデータベース復元率の評価を行う．

この攻撃では，クエリを発行するクライアントとデータベースをホストするサーバーの間のネットワークにおけるトラフィックのみを観測する盗聴者をモデルとしている．実際，クライアントとサーバーの間のやり取りがすべて暗号化されている TLS のような安全である通信プロトコルであっても，送信されるデータの方向性およびサイズは隠されていないため，受動的な盗聴者であっても，トラフィック情報を観測することが可能である．そのため，トラフィック情報を用いるボリューム攻撃が可能である．

1.1 攻撃者モデル

本研究では，レンジクエリをサポートする暗号化されたデータベースのボリュームサイズの漏洩を利用したデータベースの復元を攻撃の目標とする．データベースのレコードは数値ラベル $i \in \{1, 2, \ldots, N\}$ であるとする．例えば，医療データベースにおいては，これらのラベルは，患者の年齢，入院した日数および病気の重症度などの記録に対応する．v_i が値 i に関連付けられたレコードの個数であるとすると，ベクトル

$$\boldsymbol{v} = (v_1, v_2, \ldots, v_N)$$

は個々のラベルのデータベースカウントと呼ぶ．

攻撃者は，レンジクエリに対する応答ボリュームサイズを取得できると仮定する．ここで，レンジクエリは，$1 \le i \le j \le N$ に対して，Query(i, j) という形式の問い合わせである．ここで，Query(i, j) に対する応答は，i 以上 j 以下のラベルを持つすべてのレコードで構成される．応答ボリュームサイズは，

$$v_i + \cdots + v_j$$

で与えられる．また，$j - i + 1$ をレンジクエリの窓幅と呼ぶことにする．本研究で考える攻撃者は，クエリによる応答ボリュームサイズを得ることができるが，クエリ自体に関する情報は得られないことに注意されたい．

実際，クライアントとサーバの通信において，クエリに対してサーバが応答するボリュームサイズには，幾らかのノイズが乗ることが想定される．しかし，既存研究では，ノイズが存在する状況は考慮されておらず，どの程度のノイズが乗った場合においても，復元可能かは明らかになっていない．本研究では，攻撃者がノイズ付きボリュームサイズを観測した状況下でも，データベースカウントを復元できる攻撃手法の提案を行う．

2. 関連研究

これまでに行われている暗号化データベースに対するボリューム漏洩攻撃を紹介する．

本研究で用いる記号を導入する．データベースカウント $\boldsymbol{v}$ およびレンジクエリ集合 Q に対して，関数 $\mathcal{L}(\boldsymbol{v}, Q)$ を，クエリ $q \in Q$ に対する返答の集合とする．攻撃者は，サーバとの通信により得られた $\mathcal{L}(\boldsymbol{v}, Q)$ を利用して，$\boldsymbol{v}$ を復元できた時，攻撃成功とみなす．

Grubbs ら [9] は，N 個のレコードがあるデータベースにおいて，全てのレンジクエリの集合 Q_N に対する応答ボリュームサイズの集合 $W = \mathcal{L}(\boldsymbol{v}, Q_N)$ が得られるという仮定のもとで動作する攻撃（以下, GLMP 攻撃とする）を提案している．ここで，全てのレンジクエリの集合 Q_N は以下の式 (1) で表せる．

$$Q_N = \{(x, y) \mid 1 \le x \le y \le N\} \tag{1}$$

応答ボリュームサイズは，$N(N+1)/2$ 個得られることに相当する．また，レンジクエリの窓幅 N 以下と設定した状況に相当する．攻撃者が学習する情報は，以下で与えられる．

$$\mathcal{L}(\mathbf{v}, Q_N) = \left\{ \sum_{i=x}^{y} v_i \mid 1 \le y - x + 1 \le N \right\} \tag{2}$$

この GLMP 攻撃では，データベースカウントの復元をグラフのクリーク発見問題に帰着させることにより行われる．グラフの生成法として，観測したボリュームサイズを頂点とし，頂点と頂点の差がグラフ中のある頂点の値として存在する場合は，その頂点間に辺をはる．このグラフか

ら，N 個の頂点からなるクリーク探索の問題に帰着させることにより，データベースカウントの復元を行っている．クリーク発見問題は NP 完全であり，多項式時間アルゴリズムは存在しない．そのため，GLMP 攻撃では，ヒューリスティックな手法を用いることにより，クリークの探索を行っている．

Gui ら [7] は窓幅 $b(\ll N)$ 以下の全てのクエリの集合 Q_b 対する応答ボリュームサイズの集合 $W = \mathcal{L}(\boldsymbol{v}, Q_b)$ が得られるという状況での攻撃を提案している（以下, GJW 攻撃とする）．ここで，窓幅 b 以下の全てのレンジクエリの集合 Q_b は以下の式 (3) で表せる．

$$Q_b = \{(x, y) \mid (1 \leq y - x + 1 \leq b) \& (1 \leq x, y \leq N)\} \tag{3}$$

この論文で考えている仮定は，攻撃者が観測できるレンジクエリの窓幅の制限があり，GLMP 攻撃での仮定より弱いため，より現実的な仮定である．攻撃者が学習する情報は以下で与えられる．

$$\mathcal{L}(\mathbf{v}, Q_b) = \left\{ \sum_{i=x}^{y} v_i \mid 1 \leq y - x + 1 \leq b \right\} \tag{4}$$

GJW 攻撃では, 最大ボリュームサイズを起点とした初期解をまず生成し，解を反復的に拡張することにより，データベースカウントの復元を行っている. 具体的には，まず，観測された各ボリュームサイズを頂点としたグラフを用いて，k 個の頂点からなる部分的なクリークの探索を行う事により，初期部分解を生成する．2 個から b 個の連続するボリュームサイズの和はボリュームサイズの集合 W に含まれているという制約を利用して，解の長さが N になるまで，一つずつ左右に拡張していく手法を用いている．

GLMP 攻撃や GJW 攻撃は，攻撃者が観測できるクエリの応答ボリュームサイズにノイズが乗ることが考慮されていない．本研究では，データベースカウントが重複していないという状況を仮定するものの，ノイズを考慮したアルゴリズムの提案を行う．

3. 提案手法

3.1 仮定

具体的な説明の前に，提案手法が前提とする仮定を列挙する．

- 攻撃者は，レンジクエリ Q_2 に対する応答ボリュームサイズを入手できると仮定する．つまり，攻撃者は，窓幅 2 内のクエリの応答のボリュームサイズを全て収集できると仮定する．
- すべての $i = 1, \ldots, N$ に対して，データベースカウント v_i は正の整数である．
- 最大ラベル N は攻撃者にとって既知である．
- データベースカウントは重複していない．つまり，任意の異なる i, j に対して，$v_i \neq v_j$ である．
- ノイズの下限値および上限値は攻撃者にとって既知である．ここで，ノイズの範囲は，ノイズパラメタを δ として，$[-\delta, \delta]$ であるとする．

本稿では，以上の仮定の下で，有効なアルゴリズムの提案を行う．具体的な提案の前に，仮定の妥当性を検証する．攻撃者が収集できるクエリの窓幅 $b = 1$ のみに限定する場合，復元するデータベースカウントの順序を定めることができない．そのため，順序を含めてレコードを復元できる最小の窓幅は $b = 2$ である．また，最大ラベルの仮定は合理的である．例えば，医療記録では，患者の年齢はいくつかのデータセット（年齢グループ）に属していている．そのうえ，先行研究でも仮定されている．

我々の攻撃は，GLMP 攻撃 [9] を参考に，グラフ理論的なアプローチを用いる．提案手法では，まず，観測したボリュームサイズからグラフを構成し，そのグラフ中の頂点を少なくとも一度通る経路を探索して列挙することにより，データベースカウントを復元する手法を用いる．

3.2 ノイズ付きボリュームサイズ

実際の通信でクエリに対するサーバ側からの応答として，ボリュームサイズを観測するときに，正しい値が得られるとは限らず，ノイズが乗った値を得ることが想定される．いま，$R_{x,y}$ を Query(x, y) に対する応答ボリュームサイズに乗るノイズ分布とする．ここで，ノイズ分布は，(x, y) のみに依存すると仮定する．レンジクエリにおける窓幅は $b = 2$ であると設定しているため，攻撃者が得ることができるノイズ付き漏洩関数は式 (4) を修正することにより，以下のように定める．

$$\mathcal{L}_R(\mathbf{v}, Q_2) = \left\{ \sum_{i=x}^{y} v_i + r_{x,y} \mid 0 \leq y - x \leq 1, r_{x,y} \leftarrow R_{x,y} \right\} \tag{5}$$

攻撃者はデータベースカウント $\boldsymbol{v} = (v_1, \ldots, v_N)$ について，クエリに対するサーバからの応答を観測し，エラー付きのボリュームサイズの集合 $W = \mathcal{L}_R(\boldsymbol{v}, Q_2)$ より，$\boldsymbol{v}$ を復元することを目的とする．

3.3 提案するボリューム漏洩攻撃の流れ

実際のアルゴリズムを説明する前に，提案するアルゴリズムの概略を具体例を用いて説明する．最大ラベル $N = 4$, データベースカウント $\boldsymbol{v} = (18, 19, 15, 10)$ とする．簡単のため，$\delta = 0$ の場合，つまり，ノイズが乗らず正しい応答ボリュームサイズが得られると仮定する．窓幅 2 の制約のもと，攻撃者が観測するボリュームサイズの集合 W は式 (5) の漏洩関数より，

$$W = \mathcal{L}_R(\boldsymbol{v}, Q_2) = \{10, 15, 18, 19, 25, 34, 37\}$$

となる．

次に，応答ボリュームサイズより，以下の手順に従い，

グラフを生成する．

- 頂点は，各ボリュームサイズとする．
- 頂点 v_i, v_j に対して，$v_i - v_j = v_k$ を満たす v_k が存在するとき，v_i と v_j の間に辺をはる．

生成されたグラフを図 1 に示す．

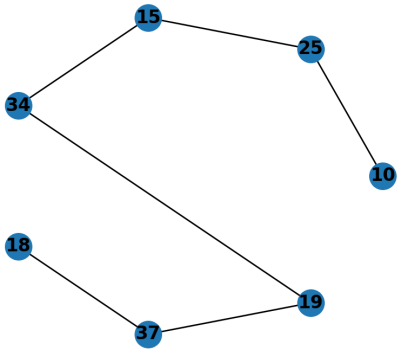

図 1　生成されたグラフ

次に，得られたグラフに対して，全ての頂点を少なくとも一度は通り，辺が全て異なる連結部分グラフを全て求める．図 1 で示したグラフの場合，グラフ自身が条件を満たす唯一の連結部分グラフになっている．通過する頂点は順に，$(10, 25, 15, 34, 19, 37, 18)$ およびその逆順となる．

最後に，通過した頂点の順列より，偶数番目に通過した頂点を取り除いた頂点列を求める．この例の場合，$(10, 15, 19, 18)$ のみとなり，これが求める解となる．

3.4　提案するノイズ付きボリューム漏洩攻撃

GLMP 攻撃 [9] と同様に，我々の攻撃では，解を求める際に，グラフ理論的なアプローチを用いる．観測されるボリュームサイズにノイズが乗る状況では，GLMP 攻撃と同じ方法で，辺をはることができない．そのため，我々の提案攻撃では，辺のはり方をノイズが含む場合にも対応できるように拡張している．

提案するノイズ付きボリューム漏洩攻撃は，4 ステップからなる．以下，詳細に 4 つのステップを説明する．

1　クエリに対する応答ボリュームサイズの観測

(i)　攻撃者は, 幅 2 以下の全てのクエリに対するノイズ付き応答ボリュームサイズの集合 $W = \mathcal{L}_R(\boldsymbol{v}, Q_2)$ を観測する．

2　グラフの生成

以下の 2 つのステップに従い，グラフを生成する．

(i)　観測した各ボリュームサイズをグラフの各頂点とする．

(ii)　頂点 v_i, v_j に対して，

$$|v_i - v_j - v_k| < 3\delta \tag{6}$$

を満たす v_k が存在するとき，v_i と v_j の間に辺をはる．

3　経路の探索

ステップ 2 により生成されたグラフ中において，全ての頂点を少なくとも一度は通る経路を，以下の 2 つのステップに従い，全て求める．

(i)　次数が 1 の頂点のうち，最小の頂点を初期解とする．

(ii)　初期解の頂点から通過する頂点の順序を

$$\check{\mathbf{v}} = (\check{v}_1, \check{v}_2, \ldots, \check{v}_k) \tag{7}$$

とし，$\check{\mathbf{v}}$ を部分解と定義する．部分解の長さ k が $k = 2N - 1$ となるまで，以下の条件式のもと，幅優先探索を行い，候補解の集合を求める．

$$\check{v}_{2i-1} \neq \check{v}_{2j-1} \quad (i \neq j, \quad i, j \in \mathbb{N}) \tag{8}$$

$$\check{v}_{2n-1} < \check{v}_{2n}, \quad \check{v}_{2n} > \check{v}_{2n+1} \quad (n \in \mathbb{N}) \tag{9}$$

4　解の復元

(i)　ステップ 3 により得られた長さ $2N - 1$ の候補解は，窓幅 1 と窓幅 2 のボリュームサイズの連結された要素の頂点列である．候補解の要素のうち，奇数番目を残した頂点列

$$\check{\mathbf{v}} = (\check{v}_1, \check{v}_3, \ldots, \check{v}_{2N-1}) \tag{10}$$

を求める．さらに，求めた頂点列の各要素を整数値に丸めたものを求める解として出力する．

ステップ 3 により得られる初期解が正しいならば，それは窓幅 1 のレンジクエリ $\mathrm{Query}(1, 1)$ または $\mathrm{Query}(N, N)$ に対する応答ボリュームサイズ v_1 あるいは v_N のいずれかである．経路探索で得られる N の部分解 (7) が正しいならば，窓幅 1 と窓幅 2 のレンジクエリに対する応答ボリュームサイズの順序の繰り返しとなるように生成されている．したがって，条件式 (8) はデータベースカウントが重複していない仮定を利用している．また，条件式 (9) は連続する窓幅 1 のボリュームサイズと窓幅 2 のボリュームサイズの関係を表している．我々の攻撃の疑似コードをアルゴリズム 1–4 に示す．

アルゴリズム 1 グラフの生成

Input: $W = \mathcal{L}_R(\mathbf{v}, Q_2), \alpha$
Output: $G(V, E)$

```
procedure CONSTRUCT GRAPH(W, α)
    G ← Graph()
    for all w ∈ W do
        G.add_node(w)
    for all w_i, w_j ∈ W | w_i < w_j do
        if |w_j − w_i − w| ≤ |α| | w ∈ W then
            G.add_edge(w_i, w_j)
    return G(V, E)
```

アルゴリズム 2 初期解の決定

Input: $G(V, E)$
Output: $\{(v) \mid v \in V\}$

procedure INITIAL SOLUTION($G(V, E)$)
  $S \leftarrow \{\}$
  for all $v \in V$ **do**
    if $G.degree(v) == 1$ **then**
      $S.append(v)$
  return $min(S)$

アルゴリズム 3 経路の探索

Input: $S, G(V, E), N$
Output: $\{(s_1, ..., s_{2N-1})\}$

procedure FIND PATH(S,G(V,E),N)
  $S \leftarrow \{\}$
  $Dup\_N \leftarrow (2N-1) - |V|$
  for $i \leftarrow 1, N-1$ **do**
    $S \leftarrow$ EXTEND UP$(S, G(V, E))$
    $S \leftarrow$ EXTEND DOWN$(S, G(V, E))$
  return S
procedure EXTEND UP($S, G(V, E)$)
  $S^{'} \leftarrow \{\}$
  $Next\_V \leftarrow \{\}$
  for all $s \in S$ **do**
    $Next\_V \leftarrow \{v\} \mid v \in G.neighbors(s[-1]), v > s[-1]$
    for all $next\_v \in Next\_V$ **do**
      if $next\_v \notin s$ **then**
        $S^{'} \leftarrow S^{'} \cup s.append(next\_v)$
      else if $Dup\_N \neq 0$ **then**
        $S^{'} \leftarrow S^{'} \cup \{s.append(next\_v)\}$
        $Dup\_N \leftarrow Dup\_N - 1$
  return $S^{'}$
procedure EXTEND DOWN($S, G(V, E)$)
  $S^{'} \leftarrow \{\}$
  $Next\_V \leftarrow \{\}$
  for all $s \in S$ **do**
    $Next\_V \leftarrow \{v\} \mid v \in G.neighbors(s[-1]), v < s[-1]$
    for all $next\_v \in Next\_V$ **do**
      if $next\_v \notin s$ **then**
        $S^{'} \leftarrow S^{'} \cup s.append(next\_v)$
      else if $Dup\_N \neq 0$
      and $next\_v \notin s[0::2]$ **then**
        $S^{'} \leftarrow S^{'} \cup \{s.append(next\_v)\}$
        $Dup\_N \leftarrow Dup\_N - 1$
  return $S^{'}$

3.5 ノイズ付きボリューム漏洩攻撃の流れ

実際にボリュームサイズにノイズが乗った場合の例を示す．例として，先程と同じ最大ラベル $N = 4$，データベースカウント $\boldsymbol{v} = (18, 19, 15, 10)$ を考える．攻撃者が，ノイズ付きボリュームの集合を観測した場合において，グラフの生成まで説明する．部分グラフの探索以降のステップは，ノイズがない場合と同一である．

アルゴリズム 4 解の復元

Input: $W \leftarrow \mathcal{L}_R(\mathbf{v}, Q_2), N, \alpha$
Output: $\{(s_1, \ldots, s_N) \mid s_i \in W\}$

procedure VOLUME ATTACK(W, N, α)
  $S \leftarrow \{\}$
  $S^{'} \leftarrow \{\}$
  $G(V, E) \leftarrow$ CONSTRUCT GRAPH(W, α)
  $S^{'} \leftarrow$ INITIAL SOLUTION$(G(V, E))$
  $S^{'} \leftarrow$ FIND PATH$(S, G(V, E), N)$
  for all $s^{'} \in S^{'}$ **do**
    $S.append(s^{'}[0::2])$
  return S

窓幅 2 の制約のもと，ステップ 1 より，攻撃者が観測するノイズ付きボリュームサイズの集合 W は式 (5) の漏洩関数より，$W = \mathcal{L}_R(\boldsymbol{v}, Q_2) = \{10.03, 15.06, 18.03, 19.13, 24.96, 34.09, 36.95\}$ が得られたとする．ステップ 1 より，グラフは図 2 に生成されたグラフを記載する．

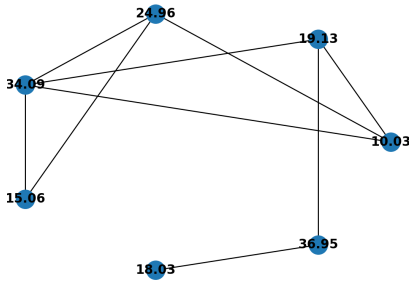

図 2 生成されたグラフ

この例の場合では，条件をみたす頂点列は，$[18.03, 36.95, 19.13, 34.09, 15.06, 24.96, 10.03]$ となる．また，求める解は整数値に丸められた $[18, 19, 15, 10]$ となる．

このように，ノイズが乗ることで，生成されるグラフの頂点間に余分な辺が追加される．ノイズ付きボリューム漏洩攻撃のステップ 3 の経路探索において，正解となる全ての頂点を少なくとも一度は通る経路を抽出することが困難になることがわかる．

4. 実験

本実験のデータ処理および提案攻撃を Python で実装した．提案攻撃は，プロセッサ： クアッドコア Intel Core i3 3.6GHz，メモリ：8GB 2400MHz DDR4 の計算機で行った．

4.1 ノイズの設定

本研究では，ノイズ分布 $R_{x,y}$ として x, y の値によらず，定義域が $[-\delta, \delta]$ で，平均 0，分散 1 の切断正規分布に従うものとする．平均 0, 分散 1 の正規分布の確率密度関数を $\phi(x)$，その累積分布を $\Phi(x)$ で表すとする．定義域が $[-\delta, \delta]$ の切断正規分布の密度関数は

$$f(x) = \frac{\phi(x)}{\Phi(\delta) - \Phi(-\delta)} \tag{11}$$

で定義される．以上より，攻撃者が得ることができるノイズ付き漏洩関数は，以下のように求められる．

$$\mathcal{L}_{2,R}(\mathbf{v}) = \left\{ \sum_{i=x}^{y} v_i + r \mid y - x \leq 1, r \leftarrow f \right\} \tag{12}$$

4.2 評価

前節で述べた提案攻撃の実験的評価を行う．復元するデータベースカウント $\boldsymbol{v}$ の長さ N を $N = 20, 30$ と設定した．復元するデータベースカウント $\boldsymbol{v}$ の偏りの設定として以下を考える．各レコードの個数 v_i の取り得る値の範囲として，$1 \leq v_i \leq 5000$ および $5001 \leq v_i \leq 10000$ の 2 つを考え，その範囲内で一様ランダムに生成した．データベースカウントの偏りとノイズによって，どのような結果が得られるのかを検証する．

ここで，解候補数が膨張して，解の探索が不能になる条件として，候補解数が N^3 を超えると計算を中止するとした．実験結果は以下の図 3 に基づき整理する．

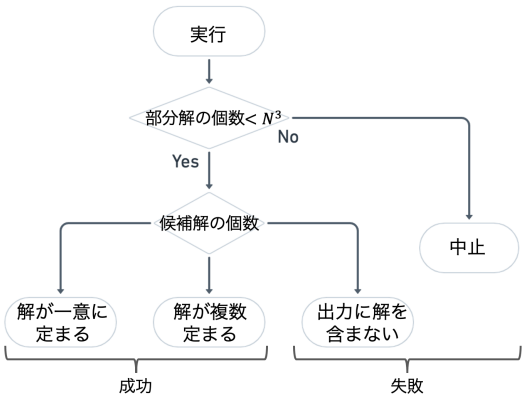

図 3 実験結果の内訳

復元成功の項目としては，一意の正しい解が得られた場合，正しい解を含む複数解が得られた場合の 2 つがある．復元失敗の項目には，出力に正しい解が含まれない場合および計算が中止された場合の 2 つがある．

各設定に対して，100 回試行した実験結果として，表 1–4 に示す．加えて，これらの表の結果より，ノイズパラメタ δ に対して，復元の成功確率の関係を図 4 に示す．

5. 考察

前節のグラフ図 4 の結果より，我々の提案手法は，ノイズパラメタ $\delta < 0.5$ のときには，80%以上の割合でデータベースカウントの復元に成功している．その一方で，ノイズパラメタ $\delta < 5.0$ のときは，平均して 20%以下の割合でデータベースカウントの復元に失敗している．データベースの偏りが同じで，データベースカウントの長さ N が異なる場合は，ノイズパラメタが大きくになるにつれ，$N = 30$ の方が $N = 20$ に比べて 30%程度復元成功確率が低くなっていることが確認できる．また，データベースの長さが等しく，データベースカウントの偏りが異なる $1 \leq v_i \leq 5000$ と $5001 \leq v_i \leq 10000$ の比較では，ノイズに影響されないような後者の大きい値の範囲のデータベースの方が，前者と比較して，最大で 40%程の高い確率で復元に成功していることが確認できる．

ノイズが大きい場合に，復元に失敗している原因は，提案アルゴリズムのステップ 3 の初期解を決定する際に，次数 1 の頂点が存在しなくなっているためだと考えられる．これは，ステップ 2 のグラフの生成において，頂点間に追加される辺の個数が多くなってしまうことが起因している．本研究では，式 (6) の頂点間に辺をはる許容差を 3δ とし，ノイズを全てカバーできるように設定した．しかし，ノイズ分布によっては，この許容差を小さく設定することによって，復元できる成功確率が高くなると考えられる．

6. まとめと今後の研究課題

本研究では，暗号化されたレンジクエリに対するノイズ付きボリューム漏洩攻撃の改良を提案した．我々の攻撃は，ノイズが小さく，データベースが密ではない場合において，高い確率データベースを復元することが可能である．

今後の研究課題として，辺をはる許容差を変動させた提案攻撃の実験評価を行う．ノイズ分布が正規分布のような値が平均値の付近に集積するような分布に従っている場合，許容差を小さく設定することで，提案アルゴリズムおよび計算量削減の有効性が高まると予想する．

謝辞 本研究の一部は，JPSP 科研費 JP19K22838 の支援の助成を受けて行われた．

参考文献

[1] J. Baek, R. Safavi-Naini, and W. Susilo,“On the integration of public key data encryption and public key encryption with keyword search,” in Proc. of ISC2006, LNCS4176, pp. 217–232, 2006.

[2] D. Boneh, G. D. Crescenzo, R. Ostrovsky, and G. Persiano, “Public key encryption with keyword search,” in Proc. of EUROCRYPT2004, LNCS3027, pp. 506–522, 2004.

[3] D. Cash, J. Jaeger, S. Jarecki, C. S. Jutla, H. Krawczyk, M. Rosu, and M. Steiner, “Dynamic searchable encryption in very-large databases: Data structures and implementation,” in Proc. of NDSS2014, pp. 1–16, 2014.

[4] C. Curino, E. P. C. Jones, R. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, and N. Zeldovich, “Relational cloud: A database-as-a-service for the cloud,” in Proc. of CIDR2011, pp. 235–240, 2011.

表 1　$N = 20, 1 \leq v_i \leq 5000$ のときの復元成功率

ノイズの範囲	成功確率			失敗確率		
	一意 (辺数)	複数 (辺数, 解の個数)	合計	解を含まず（辺数）	計算中止	合計
0	96% (41.28)	1% (52.00, 2.00)	97%	3% (46.00)	0%	3%
$-0.2 \sim 0.2$	93% (40.88)	4% (48.25, 2.00)	97%	3% (46.00)	0%	3%
$-0.4 \sim 0.4$	74% (45.45)	14% (49.21, 2.71)	88%	12% (52.83)	0%	12%
$-0.5 \sim 0.5$	70% (47.99)	19% (52.32, 2.63)	89%	11% (53.09)	0%	11%
$-0.6 \sim 0.6$	58% (49.88)	21% (54.71, 2.57)	79%	21% (53.81)	0%	21%
$-0.8 \sim 0.8$	45% (52.53)	24% (57.67, 2.67)	69%	31% (59.84)	0%	31%
$-1.0 \sim 1.0$	47% (58.23)	19% (59.79, 3.16)	66%	34% (65.09)	0%	34%
$-1.5 \sim 1.5$	35% (65.17)	18% (71.11, 5.11)	53%	47% (71.72)	0%	47%
$-2.0 \sim 2.0$	16% (74.25)	17% (80.65, 18.41)	33%	67% (83.39)	0%	67%
$-5.0 \sim 5.0$	0%	0%	0%	100% (141.25)	0%	100%
$-10 \sim 10$	0%	0%	0%	100% (228.77)	0%	100%

表 2　$N = 30, 1 \leq v_i \leq 5000$ のときの復元成功率

ノイズの範囲	成功確率			失敗確率		
	一意 (辺数)	複数 (辺数, 解の個数)	合計	解を含まず（辺数）	計算中止	合計
0	89% (69.64)	1% (74.00, 2.00)	90%	10% (74.7)	0%	10%
$-0.2 \sim 0.2$	77% (70.31)	12% (72.75, 2.17)	89%	11% (73.45)	0%	11%
$-0.4 \sim 0.4$	40% (88.42)	18% (93.61, 3.22)	58%	42% (93.71)	0%	42%
$-0.5 \sim 0.5$	48% (92.67)	20% (97.55, 3.55)	68%	32% (97.06)	0%	32%
$-0.6 \sim 0.6$	27% (100.26)	18% (103.06, 4.00)	45%	55% (110.15)	0%	55%
$-0.8 \sim 0.8$	9% (113.67)	24% (114.33, 4.17)	33%	67% (117.69)	0%	67%
$-1.0 \sim 1.0$	6% (122.33)	15% (133.33, 5.47)	21%	79% (135.03)	0%	79%
$-1.5 \sim 1.5$	2% (168.00)	7% (146.71, 10.71)	9%	89% (171.87)	2%	91%
$-2.0 \sim 2.0$	0%	0%	0%	97% (213.11)	3%	100 %
$-5.0 \sim 5.0$	0%	0%	0%	100% (408.12)	0%	100%
$-10 \sim 10$	0%	0%	0%	100% (671.86))	0%	100%

[5] R. Curtmola, J. A. Garay, S. Kamara, and R. Ostrovsky, “Searchable symmetric encryption: improved definitions and efficient constructions,” in Proc. of ACM CCS2006, pp. 79–88, 2006.

[6] L. Fang, W. Susilo, C. Ge, and J. Wang, “A secure channel free public key encryption with keyword search scheme without random oracle,” in Proc. of CANS2009, LNCS5888, pp. 248–258, 2009.

[7] Z. Gui, O. Johnson and B. Warinschi,“Encrypted databases: New volume attacks against range queries,” in Proc. of ACM CCS2019, pp. 361–378, 2019.

[8] P. Grubbs, M. Lacharité, B. Minaud, and K. G. Paterson, “Learning to reconstruct: Statistical learning theory and encrypted database attacks,” in Proc. of IEEE S&P2019, volume 00, pp. 480–496, 2019.

[9] P. Grubbs, M. Lacharité, B. Minaud, and K. G. Paterson,“Pump up the volume: Practical database reconstruction from volume leakage on range queries,” in Proc. of ACM CCS2018, pp. 315–331, 2018.

[10] H. Hacigumus, B. Iyer, and S. Mehrotra, “Providing database as a service,” in Proc. of ICDE2002, pp. 29–38, 2002.

[11] G. Kellaris, G. Kollios, K. Nissim, and A. O’Neill, “Generic attacks on secure outsourced databases,” in Proc. of ACM CCS2016, pp. 1329–1340, 2016.

[12] M. Lacharité, B. Minaud, and K. G. Paterson, “Improved reconstruction attacks on encrypted data using range query leakage,” in Proc. of IEEE S&P2018, pp. 297–314, 2018.

[13] E. Stefanov, C. Papamanthou, and E. Shi, “Practical dynamic searchable encryption with small leakage,” in Proc. of NDSS2014, pp. 23–26, 2014.

表 3　$N = 20, 5001 \leq v_i \leq 10000$ のときの復元成功率

ノイズの範囲	成功確率			失敗確率		
	一意 (辺数)	複数 (辺数, 解の個数)	合計	解を含まず（辺数）	計算中止	合計
0	100% (38.83)	0%	100%	0%	0%	0%
$-0.2 \sim 0.2$	98% (39.04)	1% (42.00, 2.00)	99%	1% (42.00)	0%	1%
$-0.4 \sim 0.4$	91% (40.47)	7% (42.57, 2.29)	98%	2% (42.00)	0%	2%
$-0.5 \sim 0.5$	82% (40.60)	14% (44.21, 3.07)	96%	4% (41.00)	0%	4%
$-0.6 \sim 0.6$	85% (40.96)	12% (44.83, 3.83)	97%	3% (45.67)	0%	3%
$-0.8 \sim 0.8$	73% (42.18)	22% (46.36, 3.73)	95%	5% (44.80)	0%	5%
$-1.0 \sim 1.0$	70% (43.04)	24% (46.75, 6.92)	94%	6% (46.50)	0%	6%
$-1.5 \sim 1.5$	55% (45.35)	31% (49.06, 4.90)	84%	14% (50.07)	0%	14%
$-2.0 \sim 2.0$	32% (47.5)	46% (51.93, 6.61)	78%	22% (53.64)	0%	22%
$-5.0 \sim 5.0$	10% (59.4)	34% (66.79, 14.00)	44%	55% (69.02)	1%	56%
$-10 \sim 10$	0%	4% (82.5, 49.5)	4%	90% (93.72)	6%	96%

表 4　$N = 30, 5001 \leq v_i \leq 10000$ のときの復元成功率

ノイズの範囲	成功確率			失敗確率		
	一意 (辺数)	複数 (辺数, 解の個数)	合計	解を含まず（辺数）	計算中止	合計
0	100% (61.05)	0%	100%	0%	0%	0%
$-0.2 \sim 0.2$	96% (61.06)	1% (64.00, 2.00)	97%	3% (67.33)	0%	3%
$-0.4 \sim 0.4$	76% (65.39)	19% (72.37, 4.63)	95%	5% (67.80)	0%	5%
$-0.5 \sim 0.5$	67% (67.48)	25% (71.48, 3.44)	92%	8% (71.12)	0%	8%
$-0.6 \sim 0.6$	66% (69.00)	26% (73.50, 3.77)	92%	8% (70.38)	0%	8%
$-0.8 \sim 0.8$	59% (72.85)	22% (77.27, 4.36)	81%	19% (76.84)	0%	19%
$-1.0 \sim 1.0$	38% (77.29)	40% (80.80, 4.85)	78%	22% (81.09)	0%	22%
$-1.5 \sim 1.5$	22% (83.50)	38% (89.13, 5.68)	60%	40% (93.47)	0%	40%
$-2.0 \sim 2.0$	5% (98.60)	37% (96.62, 11.41)	42%	56% (101.61)	0%	56%
$-5.0 \sim 5.0$	0%	1% (120.00, 32.00)	1%	92% (161.71)	7%	99%
$-10 \sim 10$	0%	0%	0%	100% (242.83)	0%	100%

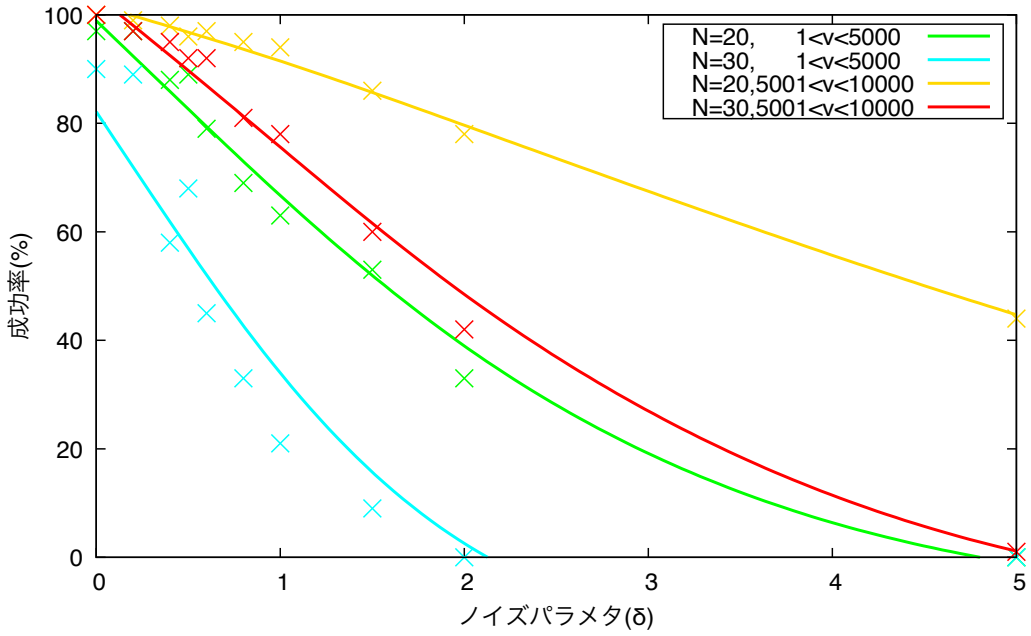

図 4　ノイズと復元成功率の関係