アドレスリストを秘匿し交わりの大きさを求める方式とその定点観測への応用

磯崎 邦隆　　菊池 浩明

東海大学大学院工学研究科
259-1292 平塚市北金目 1117
hakoten, kikn@tokai.ac.jp

あらまし ネットワーク空間を縦断した不正アクセスが定常的に試みられている. しかしながら, 各組織のセキュリティポリシー等の理由から, これらの情報を外部に提供することができない. そこで, 本論文では, n 個のアドレスリストを秘匿したまま, その交わりの大きさを算出する方法を考える. Kissner らは, 閾値以上に重複した全ての値を求めるプロトコルを提案しているが, 重複した値そのものが露見してしまう. 本稿では, 秘匿多項式評価を用いてこれを解く方式を提案し, 実装に基づいてその性能を評価する.

Protocols for Identifying Size of Intersections without revealing Address sets and its application to network port-scanning packets

Kunitaka Isozaki　　Hiroaki Kikuchi

Gradulate School of Engineering, Tokai University
1117 Kitakaname, Hiratsuka, 259-1292

Abstract Internet address space is widely scanning by malicious parties. These malicious attempts are, however, not allowed to be shared among several organizations because of the security policies. Our study aims to develop a method to identify the size of intersection of some subsets of malicious addresses without revealing the subsets. Kissner et al. proposed a protocol for computing the intersection, which reveals the addresses in the intersection and hence it is not appropriate for our purpose. In this paper, we then improve the privacy of the protocol and clarify the performance of the proposed protocol based on trial implementation.

1 はじめに

ウィルスやワーム等に感染したホストは, 新たな侵入先を求め, 定期的なポートスキャン等を実行している. しかし, これらの振る舞いは一様ではなく, 特定のアドレスブロックを集中的に探索するものや, 全IPアドレス空間を探索するものなど, 多種多様である. そこで [1] では, 各アドレスを観測したセンサの台数から, 不正ホストがアドレス空間を渡り歩く「渡り」の度合いを定量化することで不正ホストの振る舞いを観測している. 例えば, 図1は12台の異なるセンサで観測された不正な発信元アドレスの分布を示している. このような不正ホストの振る舞いを観測する際には, より多くのセンサを用いることが望ましい. しかし, 観測したIP

アドレス等の情報は組織内の秘密に関することも多くむやみに公開することができない.

このような問題に対して Freedman らは [2] で, 互いの持つ集合を多項式で表現することで, 互いの集合を秘匿したままその交わりの大きさだけを算出する秘匿多項式評価プロトコルを提案している. しかし, この方法では 2 者間に限られる. また, Kissner らは [3] で, マルチパーティーで任意の集合演算を秘匿したまま求めるプロトコルを提案している. しかし, 重複した値そのものが露見してしまうことになり, その計算コストも大きく実用的かどうかが不明であった.

そこで, 本研究では, [3] の秘匿多項式評価を改良して, n 個のアドレスリストを秘匿したまま, 閾値ごとにその交わりの大きさのみを算出する方法を提案する. 提案方式が現実的かどうかを明らかにするため, 実装に基づいてその性能を見積る.

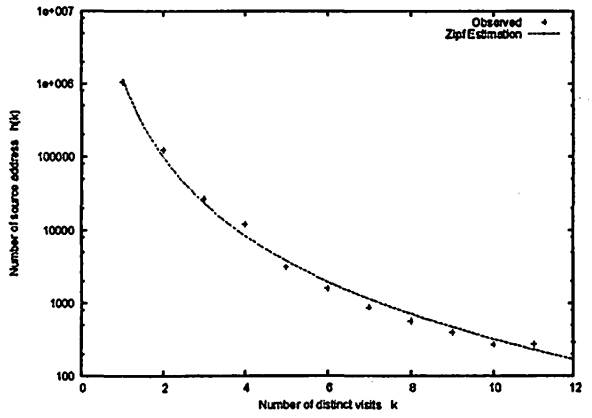

図 1: 観測されたビジット数 k についてのユニーク発信アドレス数とその近似

2 準備

2.1 Paillier 暗号 [4]

本方式では準同型性暗号として [4] で提案されている Paillier 暗号を用いる. Paillier 暗号では $N = pq$(p と q は素数) と $g \in Z^*_{N^2}$ が公開鍵, $\lambda(N)$ =LCM$(p-1, q-1)$ が秘密鍵である. 平文 M の暗号化は次のようになる.

$$C = E(M) = g^M r^n \bmod N^2$$

ここで, $r \in Z^*_N$ を乱数とする. 暗号文 C の復号は

$$M = D(C) = \frac{L(C^\lambda \bmod N^2)}{L(g^\lambda \bmod N^2)} \bmod N$$

で与えられる. ここで, L は

$$L(a) = \frac{a-1}{N} \bmod N^2.$$

秘密鍵は複数人の信頼できる鍵管理者で分散し, 閾値復号可能とすると定義される. 暗号関数 E は加法に関する次の準同型性を満たす.

$$\begin{aligned} E(a) * E(b) &= E(a+b), \\ E(a)^b &= E(a*b). \end{aligned}$$

2.2 2 者間秘匿積集合プロトコル [2]

Freedman らはクライアントとサーバが互いの持つ集合を秘匿したまま交わりの大きさを算出するプロトコルを提案している [2]. クライアントの持つ集合を $C = \{c_1, c_2, \ldots, c_k\}$, サーバの持つ集合を $S = \{s_1, s_2, \ldots, s_k\}$ とする. クライアントは多項式

$$\begin{aligned} P(x) &= (x-c_1)(x-c_2)\cdots(x-c_k) \\ &= \ell_k x^k + \cdots + \ell_0 \end{aligned}$$

を作成し, 係数 $\ell_0, \ldots, \ell_k$ を準同型性暗号により暗号化してサーバに送る. サーバは E の準同型性を利用し暗号化したままで, S の全ての s に対して $rP(s)$ を計算することで k 個の暗号文を作成し, クライアントへ送る. ここで r は乱数である. クライアントは送られた暗号文を全て復号し, 0 となる復号文の個数を得る. 0 の個数が 2 者間の持つ集合の交わり $C \cap S$ の要素数になる.

この方式では互いの持つ集合を秘匿したまま交わりの大きさを求めることは可能であるが, 2 者間に限定される. 通信コストを暗号文の数, 計算コストを暗号文をべき乗する回数とすると通信コストは $(2k+1)$ となる. 多項式の $k+1$ 個の係数の暗号化に $2k+2$ 回, 多項式評価に $k+1$ 回のべき乗がかかる. これを k について行うので k^2+k, 最後に k 個の暗号文の復号に $2k$ 回で, 計 k^2+5k+2 回のべき乗演算を要する.

2.3 マルチパーティ秘匿集合計算 [3]

Kissnerらはn人のプレーヤが互いの持つ集合を秘匿したままt人以上のプレーヤ間で和，積集合を算出する方法を提案している [3].

プレーヤiの持つ集合を$S_i = \{a_1, \ldots, a_k\}$，$S_i$の要素を根に持つ多項式を$f_i(x) = (x-a_1)(x-a_2)\cdots(x-a_k)$とする．$f_i(x)$の積$P(x) = f_1 f_2 \cdots f_n$を$t-1$回微分した$P^{(t-1)}(x)$を暗号化したまま求め，$t$人以上のプレーヤ間で共通な要素を同定する．各プレーヤは準同型性暗号を用いてf_iの係数を暗号化したまま$\phi(x) = P(x)s(x) + F(x)P^{t-1}(x)r(x)$を算出し復号する．ここで$F$は次数$t-1$で各自が取りうる要素を根に持たない多項式である．$r$，$s$は乱数を根に持つ次数$k$の多項式であり，プレーヤが分散して作成する．これにより各係数を一様な値にすることができる．各プレーヤは独自に多項式評価$b\phi(a_i)+a_i$を算出し，値が誰のものかわからないようにシャッフルしてから共有する．ここで，bは各プレーヤが自らの各要素毎に作成する乱数である．

この方式では要素そのものが算出され，交わりの大きさのみを求めることができない．また，$\phi(x)$から集合S_iの部分情報が漏れないようにランダム多項式r，sを必要とする．

2.4 ブラインド署名

ブラインド署名は署名者にメッセージを知られずに署名を行う方式である．

$N = pq$を法，eをRSA公開鍵とし，eの乗法逆元dを秘密鍵とする．ユーザはMの署名を得るために，乱数Rを選び

$$X = R^e M \bmod N$$

を署名者に送る．署名者は，$X^d \bmod N$を送り返す．ユーザは

$$M^d = RM^d/R \bmod N$$

によりアンブラインドして署名M^dを得る．

3 提案方式

3.1 概要

提案方式ではn人のプレーヤがそれぞれサイズk，要素$a \in \{1, \ldots, m\}$の集合$S_i = \{a_1, a_2, \ldots, a_k\}$を持つ．$t$人のプレーヤ間で共通な集合の要素の個数$B(t)$を得るプロトコルを3種類提案する．

「リーグ戦方式」では2者間秘匿積集合プロトコルを全プレーヤとの組み合わせについて実行する．次の「ブラインド署名方式」では，各要素は他者にわからないが，同じ値を持つ要素だけは一致するように一方向性関数にかける．ただし，要素を推測されるのを防ぐために，一方向性関数として信頼できる署名者によるブラインド署名を用いる．そして，第3の方式では，マルチパーティ秘匿集合計算を基にしてマルチパーティで秘匿したまま閾値以上の集合を求める．ただし，値そのものが漏れないように改良して行う．

3.2 提案1：リーグ戦方式

1. 互いに異なるプレーヤiとjの間で，相互に2者間秘匿積集合プロトコルを実行して$B_{i,j}(2) = |S_i \cap S_j|$を求める
2. すべての$i, j \in \{1, \ldots, n\}$の2組について1を実行する．

例えば，$S_1 = \{1,2\}$，$S_2 = \{1,3\}$，$S_3 = \{1,3\}$の場合，S_1とS_2で共通な要素の数$B_{1,2}(s) = 1$となり，数を算出することはできるが，$t > 2$ではt人で共通な要素数を得ることができない．

3.3 提案2：ブラインド署名方式

1. プレーヤiは，$a_{i,1}, \ldots, a_{i,k}$のk個を信頼できる署名者にブラインド署名してもらい，$a^d_{i,j}$を匿名通信路を経由してだれの値かわからないようにして公開する．$i = 1, \ldots, n$について繰り返す．
2. 公開された値のうち，t個出現している$a^d_{i_1,j_1} = \cdots = a^d_{i_t,j_t}$となる値の数を$B(t)$とする．

しかし，本方式ではプレーヤはブラインド署名された要素が自分の S_i の要素であるか否かがわかってしまう．部分情報が漏れることになる．

3.4 提案3：秘匿多項式評価

全プレーヤは公開鍵 pk を持ち，秘密鍵は複数人の信頼できる鍵管理者が分散して持つ．暗号方式は Paillier 暗号を用い，準同型性を有する．

1. プレーヤ $i(i=1,\ldots,n)$ は $S_i=\{a_1,\ldots,a_k\}$ について多項式

$$\begin{aligned} f_i &= (x-a_1)(x-a_2)\cdots(x-a_k) \\ &= \ell_k x^k+\cdots+\ell_0 \end{aligned}$$

を作成する．プレーヤ1は係数を pk により暗号化した $\lambda_1=(E(\ell_0),\ldots,E(\ell_k))$ をプレーヤ2へ送る．

2. プレーヤ $i=2,\ldots,n$ は $i-1$ 番目のプレーヤから受け取った λ_{i-1} と自らが持つ f_i から，準同型性を利用して $\lambda_i=\lambda_{i-1}\cdot f_i$ を作成し，プレーヤ $i+1$ に送る．$i+1$ 番目のプレーヤは，i 番目からもらった多項式 $\lambda_i(x)=\ell_0+\ell_1x+\cdots+\ell_{ki}x^{ki}$ の暗号文と自身のリストについての多項式 $f_{i+1}(x)=c_0+c_1x+\cdots+c_kx^k$ の積の暗号文を次のように求める．

$$\begin{aligned} \lambda_{i+1}(x) &= \lambda_i(x)f_{i+1}(x) \\ &= \sum_{j=0}^{ki+k+1}\Big(\sum_{\alpha+\beta=j}\ell_\alpha c_\beta\Big)x^j \\ &= \ell'_0+\ell'_1x+\cdots+\ell'_{ki+k+1}x^{ki+k+1} \end{aligned}$$

となる．よって，その係数 ℓ'_i は，E の準同型性により，

$$\begin{aligned} E(\ell'_j) &= \prod_{\alpha+\beta=j}E(\ell_\alpha)^{c_\beta} \\ &= E\Big(\sum_{\alpha+\beta=i}\ell_\alpha c_\beta\Big) \end{aligned}$$

により，暗号化したまま求められる．以上の処理を，$i=2,\ldots,n$ まで順に繰り返し，得られた λ_n の暗号文を公開する．

3. プレーヤ1は代表して $\lambda_n(x)$ を $t-1$ 回微分した $\lambda_n^{(t-1)}(x)$ を作成する．$\lambda_n^{(t-1)}(x)$ の暗号文 $(E(\ell_0),\ldots,E(\ell_{nk}))$ から，要素 a についての多項式の値 $\lambda_n^{(t-1)}(a)$ を，

$$\begin{aligned} \mu_a^t &= E(\lambda_n^{(t-1)}(a)) \\ &= \prod_{j=0}^{nk}E(\ell_j)^{a^j}=E(\sum_{j=0}^{nk}\ell_ja^j) \end{aligned}$$

により暗号化したまま求める．この多項式評価を全ての $a\in\{1,\ldots,m\}$ について行い，$(n-1)m$ 個の暗号文 μ_a^t を鍵管理者に送る．

4. 鍵管理者は閾値以上で協力して乱数 r を作り，暗号文 μ_a^t との積 $r\mu_a^t$ を復号する．a が t 人以上のプレーヤのリストにあるかどうかは，

$$D(r\mu_a^t)=\begin{cases} 0 & \text{if } a\in S_{i_1}\cap\cdots\cap S_{i_t}, \\ NZ & \text{otherwise} \end{cases}$$

により判定できる．ここで NZ は $r\lambda_n^{(t-1)}(a)\neq 0$ であり，r がわからないので0以外の乱数であることしか言えない．したがって，t 以上の交わりの数は，

$$B(t)=\Big|\{a|D(r\mu_a^t)=0,a\in\{1,\ldots,m\}\}\Big|$$

により与えられる．

$B(t)$ が t 人以上で共通の要素数である．t 人での共通要素の数を得るには $B(t)-B(t+1)$ を用いる．提案プロトコルで $t=1$，すなわち微分を行わなかった場合，全プレーヤが持つ要素のユニーク数を得る．

提案方式で復号されるのは乱数 r が掛けられた多項式の評価値であり，係数を一様にする必要はなく，a が t 以上の交わりに属するか否か以外の情報は漏れない．よって Kissner らの方式のようにランダムな多項式によって次数が増えることがない．

4 評価

3つの提案方式の通信コストと計算コストおよび安全性を評価する．

表 1: 提案方式の比較

方式	1. リーグ戦	2. ブラインド署名	3. 秘匿多項式評価
技術	[2]	ブラインド署名	[3]
3者以上の通信	×	○	○
秘匿性	○	×	○
通信コスト	$(2k+1)(n^2-n)$	nk	$O(n^2k+mn)$
計算コスト	$(k^2+5k+2)(n^2-n)$	$2nk$	$O(n^2k^2+nkm)$

4.1 実験方法

提案方式3をJavaで試験実装してその性能を計測した.ただし, プレーヤは定義域$\{1,\cdots,50\}$(すなわち$m=50$)の上のサイズ$k=10$の集合S_iを持ち, $t=1,\cdots,n$人が共通して持つ要素の個数を算出する. なお, $t=1$の場合は全プレーヤ間でのユニーク数である. 実験はWindows Vista SP1, 2.66GHz,Core 2 Quadの環境で行った.

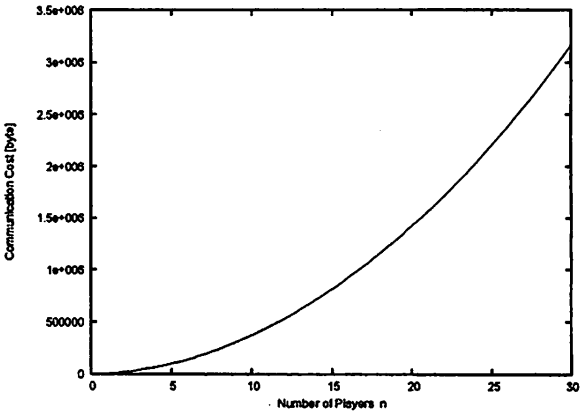

図 2: プレーヤ数nに対する総通信量$L(n)$

4.2 通信コスト

リーグ戦方式では, n人のプレーヤ間のすべての2組数は, $(n^2-n)/2$あり, 相互に行うので, 計n^2-n回分の2者間秘匿積集合プロトコル[1]が実行される.

ブラインド署名方式では, k個の署名をn人が公開するのでnk個分の送信コストがかかる.

秘匿多項式評価における通信コストは暗号化した多項式の係数の数$nk+1$に依存する. n人のプレーヤが送信する総通信コスト$L(n)$は次のように表わされる.

$$L(n)=\sum_{i=1}^{n} ik+1 \quad = \quad \frac{k}{2}n^2+\frac{k+1}{2}n$$

暗号の鍵の長さを1024bit, 暗号文のサイズを682byteとした時の通信量を図2に示す.

4.3 計算コスト

計算コストをべき乗計算の数とする. リーグ戦方式では, 1回にk^2+5k+2のコストがかかる2者間秘匿積集合プロトコルを組み合わせの数n^2-n回繰り返す. 一方, ブラインド署名方式は, n人が各々k回署名するだけなので$2nk$のべき乗しかかからない.

秘匿多項式評価の計算コストは, 主に3.4節のステップ1(暗号化), 2(積演算), 3a(微分), 3b(多項式評価)にかかるので, これを各々, W,V,Yで表す. まずPaillier暗号による多項式の暗号化と復号には各々$2kn$とmnの計算コストがかかる. この実測値は5961[ms], 27,085[ms]であった.

n番目のプレーヤが$n-1$番目の$(n-1)k+1$次の多項式λ_{n-1}と自分のf_nの積にかかる計算コスト$W(n,k)$は

$$W(n,k)=((n-1)k+1)(k+1)$$

である. 従って, n人全てについては, $\sum_{i=1}^{n} W(i,k)=(\frac{n^2-n}{2}k+n-1)(k+1)$かかる.

ステップ3における閾値tによる$\lambda(x)$の$t-1$回微分$\lambda_n(x)^{(t-1)}$の計算コスト$V(t,n,k)$は次のように表わされる.

$$V(t,n,k)=nk-t+2$$

[1] $t>2$の場合は求めることができないので, 比較の対象にはならない.

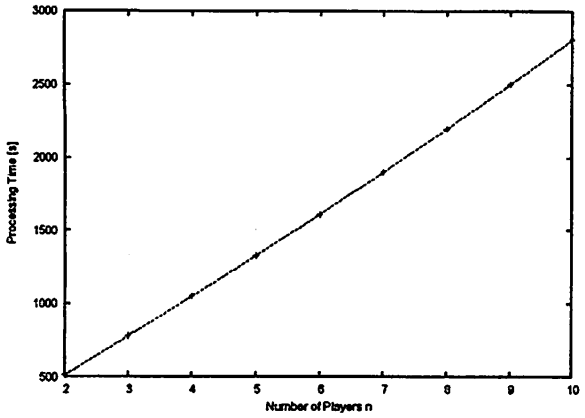

図 3: プレーヤ数 n についての総計算コスト

実際には，$t = 1, \ldots, n$ について繰り返すので $\sum_{t=1}^{n} V(t,n,k) = -\frac{1}{2}t^2 + (nk + \frac{1}{2})t - \frac{1}{2}$ のコストがかかる．

ステップ 3(後半) の代表プレーヤによる多項式評価における計算コストは次のように表わされる．

$$Y(t,n,k) = nk - t + 1$$

全ての t について行うと，計

$$\sum_{t=1}^{n} Y(t,n,k) = \frac{2k-1}{2}n^2 + \frac{1}{2}n$$

のコストが生じる．閾値が増えるに従って暗号文の係数が減少するため，計算コストも下がる．

提案方式 3 の一連の処理には，閾値 $t = 1, \ldots, n$ について，W, V, Y を行う必要があり，更に Y は全ての m 値について行うので，結局

$$\begin{aligned} \sum_{t=1}^{n} & W(t,k) + V(t,k,n) + Y(t,k,n) \cdot m \\ = \; & 3.02437n^2k^2 + 250.662nkm \\ = \; & O(n^2k^2 + nkm) \end{aligned}$$

の計算時間がかかる．以上より，プレーヤ数 n についての総計算コストは図 3 の様になる．

4.4 考察

表 1 に 3 方式の性能と安全性を整理する．リーグ戦方式は 3 者以上での通信において共通な要素の数を求めることができない．一方ブラインド署名方式では 3 者以上に対応できるものの，自分の有するリストに関しては，$B(t)$ に加えて共通な要素の値そのものも分かってしまうため秘匿性は不完全である．秘匿多項式評価は 3 者以上で秘匿性を保ちつつ共通な要素の数を求めることができる反面，大きなコストがかかる．

秘匿多項式評価の計算コストでは，積にかかる $W(n,k)$ が最大の計算オーダーをもつが，実測値では多項式評価 $Y(t,n,k)$ が圧倒的に大きい．これは $Y(t,n,k)$ の多項式の係数の値が大きいので暗号文のべき乗の指数が大きくなるためである．

5 結論

互いが持つ集合を秘匿したまま t 人が共通して持つ要素の数を求める秘匿多項式評価を提案し，通信コスト，計算コストを計測した．その結果，多項式評価に最も処理量がかかり，$n = 10$ の場合 2810[s] かかることが分かった．今後の課題として要素の総当たりによる多項式評価を改良し，コストを下げることが挙げられる．

参考文献

[1] 福野，小堀，菊池，寺田，土井，“インターネットの分散観測による不正侵入者の探索活動のマクロ・ミクロ解析”，情報処理学会研究報告 コンピュータセキュリティ (CSEC34)，pp. 299–304，2006.

[2] M. Freedman, K. Nissim, and B. Pinkas, “Efficient Private Matching and Set Intersection”, EuroCrypto'04, pp. 1–19, 2004.

[3] L. Kissner and D. Song, “Privacy-Preserving Set Operations”, Crypto'05, pp. 200–212, 2005.

[4] D. Paillier, “Public-key Cryptosystems Based on Composite Degree Residuosity Classes”, EuroCrypto'99, LNCS 1592, pp. 223–238, 1999.