定数ラウンド (k, n) 秘匿モジュロ変換プロトコルの提案

加藤 遼 †　　桐淵 直人 †　　西脇雄高 †　　吉浦裕 †

† 電気通信大学
〒 182-8585　東京都調布市調布ヶ丘 1-5-1

あらまし 個人情報や秘密情報を保護しながら活用することがますます重要になっている. この課題を解決する手法として, 秘密分散法を用いた秘匿計算が期待されている. 秘匿計算の多くのアプリケーションでは, 異なる素数で分散されたシェア同士の秘匿計算が必要になる. その実現の手法として, モジュロ変換プロトコルがある. しかし, このプロトコルには定数の通信回数ではなく, くわえて計算の途中で排他的論理和の秘密分散法を用いているため, 通信のロバスト性が (k, n) から (k, k) に低下するという問題がある. 本研究では, 定数の通信回数で (k, n) のロバスト性を持つモジュロ変換プロトコルを提案した. そのプロトコルの通信回数は 26, 通信量は $40ln + 93l + 10n + 9$ である.

Improving Efficiency of Secure Multi-Party Computation

Ryo Kato†　　Naoto Kiribuchi†　　Yutaka Nishiwaki†　　Hiroshi Yoshiura†

†The University of Electro-Communications
1-5-1 Chouhugaoka, Chouhu, Tokyo 182-8585, JAPAN

Abstract It becomes more and more important to balance information usability and confidentiality. Secure multi-party computation based on secret sharing is expected to meet this challenge. It has serious problems, however, that it requires large amount of communications and storage volume. In this paper, we describe a method to reduce the communication and storage costs of secure multi-party computation based on secret sharing.

1　はじめに

個人の購買履歴や位置情報, 診療履歴などをデータベースに集積し, 直接あるいはマイニングを通じて利用する事により利用者に対して高度なサービスを提供する事が可能になる. しかし, それらの個人情報がデータベースから漏洩すれば, 様々な致命的な問題が引き起こされる. またクラウドコンピューティングでは利用者の機密情報を守ると同時に, 利用者のユーザービリティも高水準に保つ必要がある. したがって, 情報の利用と保護を両立する事が重要となっている.

情報の利用と保護を両立する有望な技術として, 秘密分散法に基づく秘匿計算が期待されている. 秘匿計算では, 秘密情報を n 人の参加者に分散して, 参加者間の通信により秘密情報を秘匿したままの状態でその関数値を求める事が出来る. 分散した秘密情報をシェアと呼ぶ. 秘密分散法と秘匿計算を用いる事により, 例えば個人の購買履歴を秘匿したまま, データ分析が可能である.

秘匿計算の多くのアプリケーションでは, 異なる素数で分散されたシェア同士の秘匿計算が必要である. そのような秘匿計算を可能にする手法として, モジュロ変換プロトコルがある [11].

モジュロ変換プロトコルの重要性を二つの例を挙げて説明する.

第一の例では, 一日の労働時間 ($0 \leq hour \leq 24$) と時給 ($0 \leq pay \leq 10000$) のような, 値域が大きく異なる二つの属性を持った秘密分散データベースを考える. このデータベースでは同じ素数で分散されたシェア同士の秘匿計算しか出来ないとする. 日給 ($pay \times hour$) を秘匿計算で求めたいならば, あらかじめ時給の属性に加え労働時間の属性まで, 時給の最大値 (10000) 以上の素数で分散しておく必要がある. しかし, そのような分散の方法を採ると, データ量が増大する. くわえて, ボーナス ($0 \leq bonus \leq 1000000$) のような, 時給の最大値 (10000) に収まりきらない属性を新たに加えたい時, シェアを一度復元してから, ボーナスの最大値以上の素数で再度分散し直す必要があるため, 利便性および安全性の低下の問題が生じる.

第二の例では, 組織 A と組織 B が存在し, 組織 A はデータ A' を, 組織 B はデータ B' を秘密分散して安全に管理している状況を考える. この時, 組織 A と B は異なるポリシーに基づくので, データ A' と B' は同じ素数で分散されているとは限らない. もしも異なる素数で分散されている場合, 組織 A と組織 B 間では, データ分析などの秘匿計算がモジュロ変換プロトコルなしには出来ない.

従来のモジュロ変換プロトコルには, 定数の通信回数ではなく, くわえて計算の途中で排他的論理和の秘密分散法を用いているため, 通信のロバスト性が (k, n) から (k, k) に低下するという問題がある. 本研究では, 定数の通信回数で (k, n) のロバスト性を持つモジュロ変換プロトコルを提案した. そのプロトコルの通信回数は 26, 通信量は $40ln + 93l + 10n + 9$ である.

2 先行研究

2.1 $Shamir(k, n)$ 閾値秘密分散法

分散の際は, ディーラは先ず秘密情報 $S \in Z_p$ となる $GF(p)$ 上の $k-1$ 次の多項式

$$f(x) = S + r_1 x + \cdots + r_{k-1} x^{k-1} \pmod{p}$$

を選ぶ. ただし $S < p$ かつ, $r_i \in Z_p$ は乱数である. 次に参加者 P_d に対し, ディーラは $f(d)$ を配布する ($1 \leq d \leq n$). 復元の際は, k 個以上の $f(d)$ から S が得られる [9]. 以降, $f(d)$ をシェアと呼ぶ.

2.2 $Shamir(k, n)$ 閾値秘密分散法を用いた秘匿計算

秘密情報を秘匿したまま, 秘密情報同士の, 和・積・大小判定・等号判定等が可能である [3][5][7][8][10] .

2.3 モジュロ変換プロトコル

モジュロ変換プロトコルを用いれば, 素数 p で分散されたシェアを, 素数 q で分散されたシェアに変換する事が可能である. しかし, 従来のモジュロ変換プロトコルは定数の通信回数ではなく, くわえて計算の途中で排他的論理和の秘密分散法を用いているため, 通信のロバスト性が (k, n) から (k, k) に低下するという問題点がある [11].

2.4 Bitwise Less-Than

Bitwise Less-Than プロトコルでは, ビットのシェア同士の比較, ビットのシェアと公開情報の比較が可能である. 本研究では Bitwise Less-Than プロトコルをビットのシェアと公開情報の比較のみに用いる. この時の通信回数は 7 、通信回数は $17l$ である [5].

2.5 Unbounded Fan-In Xor

2.5.1 プロトコルの機能

Unbounded Fan-In Xor では $[S_1]_p \oplus [S_2]_p \oplus \cdots \oplus [S_i]_p \oplus \cdots \oplus [S_m]_p$ を定数の通信回数で求める事が可能である ($S_i \in \{0, 1\}$)($1 \leq i \leq j$)[7].

2.5.2 プロトコルの手順

Step1::ビットの総和に 1 を加えたものを求める.

$[Sum]_p \leftarrow 1 + \sum [S_i]_p$

Step2::ラグランジェ補間公式を用いて, 以下のような関数 f を求める $(1 \leq x \leq m+1)$.

$f(x) = x + 1 \pmod 2$

Step3::$f([Sum]_p)$ を秘匿計算する.

2.5.3 通信回数・通信量

通信回数は 3, 通信量は $5m$ である.

3 表記規則

3.1 記号

素数: p, q

素数 p のビット数: l

秘密情報: S

3.2 プロトコル

和:

$[a+b]_p \leftarrow [a]_p + [b]_p$

排他的論理和:

$[a \oplus b]_p \leftarrow [a]_p \oplus [b]_p$

秘密公開:

$S \leftarrow reveal([S]_p)$

Bitwise Less-Than:

$[a > b]_p \leftarrow BLT([a_l]_p, \ldots, [a_1]_p,\ b)$

4 定数ラウンドモジュロ変換プロトコルの提案

4.1 乱数生成

4.1.1 プロトコルの機能

このプロトコルでは, 1 ビットの乱数のシェア $[R]_p$,$[R]_q$ を定数の通信回数で生成する事が可能である $(R \in \{0,1\})$.

4.1.2 プロトコルの手順

Step1:: $Player_i$ は 1 ビットの乱数 R^i を生成し $[R^i]_p$,$[R^i]_q$ を配布する $(R^i \in \{0,1\})(1 \leq i \leq n)$.

Step2:: Unbounded Fan-In Xor を用いて, 以下を秘匿計算する.

$[R]_p \leftarrow [R^1]_p \oplus [R^2]_p \oplus \cdots \oplus [R^n]_p$

$[R]_q \leftarrow [R^1]_q \oplus [R^2]_q \oplus \cdots \oplus [R^n]_q$

4.1.3 通信回数・通信量

Step1 に通信回数 1, 通信量 2 を必要とする. Step2 に通信回数 3, 通信量 $10n$ を必要とする. 合計で通信回数は 4, 通信量は $10n+2$ である.

4.1.4 表記規則

以降, このプロトコルを次のように表記する.
$[R]_p, [R]_q \leftarrow genRBDP$

4.2 ビットモジュロ変換

4.2.1 プロトコルの機能

このプロトコルでは, p で分散されたビットのシェア $[S]_p$ から, $[S]_q$ を定数の通信回数で求める事が可能である $(S \in \{0,1\})$.

4.2.2 プロトコルの手順

Step1::$[R]_p, [R]_q$ を生成する ($R \in \{0,1\}$).

$[R]_p, [R]_q \leftarrow genRBDP$

Step2::以下を秘匿計算する.

$[T]_p \leftarrow [S]_p \oplus [R]_p$

Step3::T を公開する.

$T \leftarrow reveal([T]_p)$

Step4::以下を計算する.

$[S]_q \leftarrow T \oplus [R]_q$

4.2.3 通信回数・通信量

Step1 に通信回数 4, 通信量 $10n+2$ を必要とする. Step2 に通信回数 1, 通信量 1 を必要とする. Step3 に通信回数 1, 通信量 1 を必要とする. 合計で通信回数は 6, 通信量は $10n+4$ である.

4.2.4 表記規則

以降, このプロトコルを次のように表記する.
$[S]_q \leftarrow moduloConvert([S]_p)$

4.3 モジュロ変換

4.3.1 プロトコルの機能

このプロトコルでは, p で分散されたシェア $[S]_p$ から, $[S]_q$ を定数の通信回数で求める事が可能である ($S \in Z_p$).

4.3.2 プロトコルの手順

Step1::$genRBDP$ プロトコルを l 回用いる ($R_i \in \{0,1\}$).

$[R_l]_p, \ldots, [R_1]_p, [R_l]_q, \ldots, [R_1]_q \leftarrow genRBDP$

Step2::R を秘匿した状態で R と p の大小比較をする. $p \leq R$ の時は Step1 に戻る.

$[B]_p \leftarrow BLT([R_l]_p, \ldots, [R_1]_p,\ p)$

$B \leftarrow reveal([B]_p)$

Step3::以下を秘匿計算する.

$[R]_p \leftarrow 2^0[R_1]_p + 2^1[R_2]_p + \cdots + 2^{l-1}[R_l]_p$

$[R]_q \leftarrow 2^0[R_1]_q + 2^1[R_2]_q + \cdots + 2^{l-1}[R_l]_q$

$[T]_p \leftarrow [S]_p + [R]_p$

$T \leftarrow reveal([T]_p)$

Step4::R と T の大小比較をする.

$[R > T]_p \leftarrow BLT([R_l]_p, \ldots, [R_1]_p,\ T)$

Step5::以下を秘匿計算する.

$[R > T]_q \leftarrow moduloConvert([R > T]_p)$

$[S]_q \leftarrow T + p \times [R > T]_q - [R]_q$

4.3.3 通信回数・通信量

Step1 に通信回数 4, 通信量 $10ln + 2l$ を必要とする. Step2 において Bitwise Less-Than プロトコルに通信回数 7 と通信回数 $17l$, 秘密公開に通信回数 1 と通信回数 1 を必要とする. したがって Step2 の合計は, 通信回数は 8, 通信量は $17l+1$ である. 文献 [5] に述べられているように $p > R$ が 4 回に 1 回成立するとして, Step1 と Step2 の合計は通信回数は 12, 通信量は $40ln + 76l + 4$ である. Step3 に通信回数 1, 通信量 1 を必要とする. Step4 に通信回数 7, 通信量 $17l$ を必要とする. Step5 に通信回数 6, 通信量 $10n+4$ を必要とする. 合計で通信回数は 26, 通信量は $40ln + 93l + 10n + 9$ である.

5 まとめ

異なる素数で分散されたシェア間での秘匿計算を可能にする手法として, モジュロ変換プロトコルがある. しかし, このプロトコルには定数の通信回数ではなく, くわえて計算の途中で排他的論理和の秘密分散法を用いているため, 通信のロバスト性が (k,n) から (k,k) に低下するという問題がある. 本研究では, 定数の通信回数で (k,n) のロバスト性を持つモジュロ変換プロトコルを提案した. そのプロトコルの通信回数は 26, 通信量は $40ln + 93l + 10n + 9$ である.

参考文献

[1] J. Bar-Ilan and D. Beaver, "Non-cryptographic fault-tolerant computing in constant number of rounds of interaction," Proc. of the 8th annual ACM Symposium on Principles of distributed computing, pp201–209, 1989.

[2] M. Ben-Or, S. Goldwasser and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," Proc. of the 20th annual ACM Symposium on Theory of computing, pp1–10, 1988.

[3] R. Cramer and I. Damgård, "Secure Distributed Linear Algebra in a Constant Number of Rounds," Proc. Crypto 2001, LNCS 2139, pp119–136, 2001.

[4] R. Cramer, I. Damgård, and Y. Ishai, "Share conversion, pseudorandom secret sharing and applications to secure computation," Proc. 2nd Theory of Cryptography Conference, LNCS 3378, pp342–362, 2005.

[5] I.Damgård, M. Fitzi, E. Kiltz, J. B. Nielsen and T. Toft, "Unconditionally Secure Constant-Rounds Multi-party Computation for Equality, Comparison, Bits and Exponentiation," Proc. 3rd Theory of Cryptography Conference (TCC 2007), LNCS 3876, pp285–304, 2006.

[6] R. Gennaro, M. O. Rabin and T. Rabin, "Simplified VSS and fast-track multiparty computations with applications to threshold cryptography," Proc. of the 17th annual ACM symposium on Principles of distributed computing, pp101–111, 1998.

[7] T. Nishide and K. Ohta, "Multiparty Computation for Interval, Equality, and Comparison Without Bit-Decomposition Protocol," Proc. 10th International Conference on Theory and Practice of Public-Key Cryptography(PKC), LNCS 4450, pp343–360, 2007.

[8] C. Ning and Q. Xu, "Multiparty Computation for Modulo Reduction without Bit-Decomposition and A Generalization to Bit-Decomposition," Proc. AsiaCrypt 2010, LNCS 6477, pp483–500, 2010.

[9] A. Shamir, "How to Share a Secret," Commun. ACM, vol.22, No.11, pp612–613, 1979.

[10] T. Toft, "Constant-Rounds, Almost-Linear Bit-Decomposition of Secret Shared Values," Proc. CT-RSA 2009, LNCS 5473, pp357–371, 2009.

[11] 加藤遼、桐淵直人、西脇雄高、吉浦裕 "GF(p) から GF(q) への秘匿モジュロ変換プロトコルの提案," 情報処理学会研究報告 Vol.2011-SPT-1 No.12, pp1-7, 2011.7