F-002

完全準同型暗号のデータマイニングへの利用に関する研究動向
A Survey on the use of Fully Homomorphic Encryption Scheme in Data Mining

佐藤 宏樹† 馬屋原 昂† 石巻 優† 今林 広樹† 山名 早人†
Hiroki Sato Akira Umayabara Yu Ishimaki Hiroki Imabayashi Hayato Yamana

1. はじめに

近年，国や企業などの組織が保有するデータ量が爆発的に増加するにつれ，こうした保有データの利活用，すなわちビッグデータ解析が脚光を浴びている．そして，ビッグデータの保管から解析までがクラウド上で行われるようになってきている[1]．また，組織が収集した膨大なデータを他の組織のデータと統合することで，より価値のある情報を発見できる可能性が高まっている．このようにデータの利活用が必要とされる一方，多くのデータはプライベートな情報や機密情報を含むため，クラウドや共同計算者を含む他者に情報を開示すべきではなく，秘密情報の保護と活用の両立が求められている．

本稿では，秘密を保持したままデータの解析を行う技術として，完全準同型暗号を用いたプライバシー保護データマイニングの研究調査の結果を報告する．プライバシー保護データマイニング（PPDM: Privacy Preserving Data Mining）とはプライバシーの保護とデータの利活用を両立するための技術であり，2000 年の Lindel ら[2]と Agrawal ら[3]の論文発表以降，盛んに研究が行われている．PPDM の研究はプライバシーを保護する対象から 2 つに分類できる．1 つはデータマイニングを行う前のデータを保護する技術である．もう 1 つは，公表するデータマイニングの結果から元のデータの特定を防ぐ技術である[4]．本稿の対象は前者のデータ保護が該当する．

PPDM の要素技術にはノイズを付加する手法[3]や個人を特定できる情報を削除する匿名化手法[5]，暗号技術を用いる手法などがある．ノイズ付加や匿名化はともに計算コストが小さいが，正確なデータマイニングの結果を得ることができない．正確な結果を得るためには，計算コストが大きくても暗号技術を用いる必要がある．

PPDM を実現する暗号技術として，特に準同型暗号が近年注目を集めている．準同型暗号（HE: Homomorphic Encryption）とは，加算や乗算の演算に対して準同型性を持つ（平文空間の演算に対応する暗号文空間の演算が存在する）暗号方式であり，暗号化状態のまま演算を行うことができる．準同型暗号としては，従来，元の平文に対する乗算が可能な RSA 暗号[6]や加算が可能な Paillier 暗号[7]などが知られていた．これらは暗号文上で元の平文に対して行える演算が限定されるが，2009 年に Gentry が加算と乗算のともに可能な完全準同型暗号（FHE: Fully HE）の構成法を提案[8]し，暗号文上で平文に対して任意の演算（四則演算）を行うことができるようになった．しかし，FHE は暗号文長が長く計算量も大きく実用的ではない点や，暗号化状態での計算が整数加乗算に限られる点などの解決すべき課題が多く，PPDM に適用する上で数々の問題が残されている．

本稿では，FHE を用いたデータマイニングの研究調査および，前述の課題に対する取り組みをまとめる．以下，2 節でサーベイ手法と全体動向を示す．3 節では FHE の研究を，4 節で PPDM の研究をまとめ，FHE と PPDM の関係を説明する．5 節では FHE をデータマイニングに適用した研究を紹介し，6 節で FHE を用いたデータマイニングの課題とそれに対するこれまでの手法を紹介する．最後に，7 節でまとめを行う．

†早稲田大学 Waseda University

2. サーベイ方法と全体動向

2.1 調査手法

本サーベイでは論文調査を 2 段階に分けて行った．第 1 段階では「完全準同型暗号（FHE）」と「プライバシー保護データマイニング（PPDM）」の各分野について近年の主要な動向を調査した．第 2 段階では，完全準同型暗号により暗号化したデータに対してデータマイニングを行う研究に絞り，詳細に動向を調査した．

第 1 段階では，暗号分野・データマイニング分野の各国際会議において 2010 年以降に発表された論文のうち，タイトルと概要から対象研究であると判断された論文を選んだ．

暗号分野では下記の会議から調査した．

1. IEEE Symposium on Security & Privacy
2. USENIX Security Conferences/Symposium
3. EUROCRYPT
4. CRYPTO
5. ACM Computer and Communications on Security

データマイニング分野は下記の会議を選んだ．

1. ACM SIGKDD
2. IEEE ICDM
3. ACM SIGMOD

加えてサーベイ論文[9][10]や，収集した論文が参照する論文や被参照数の多い論文も対象とした．

次に第 2 段階では，FHE を用いたデータマイニングの具体的な実装や評価を行った論文を収集するためにキーワード検索を行った．2010 年以降の論文で，2016 年 5 月末時点でインターネットから取得可能な英語または日本語の論文を対象とした．IEEE Xplore Digital Library[1]と ACM Digital Library[2]にて「”fully homomorphic encryption”」，Google Scholar[3]にて「”fully homomorphic encryption” “data mining”」 を検索キーワードとして検索した結果からタイトルと概要を元に論文を選択した．

2.2 論文数の動向

第 1 段階の結果として，論文数の推移を図 1 にまとめる．FHE に関する研究も PPDM の研究も常に一定数の論文が発表され続けている．

[1] http://ieeexplore.ieee.org
[2] http://dl.acm.org
[3] https://scholar.google.com

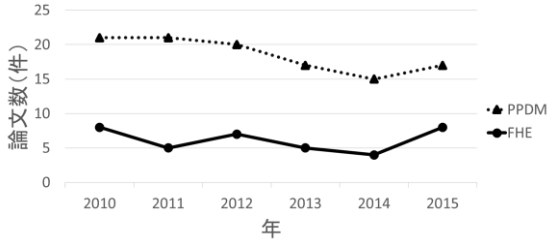

図 1. FHE と PPDM の論文数推移

2.3 本稿対象研究の動向

第 2 段階の結果として，FHE をデータマイニングなどに応用した研究の論文数の推移を図 2 にまとめる．論文数は増加し続けており，前項の結果と比較して，応用研究が急速に盛んになっている．

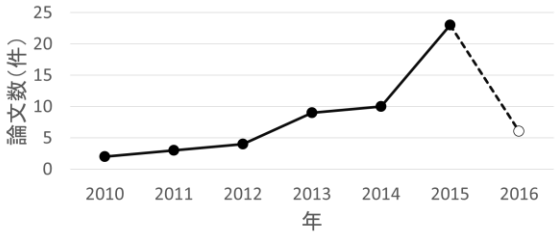

図 2. PPDM の FHE による実装報告論文数
（2016 年は 5 月末までの件数）

3. 完全準同型暗号

本節では，これまでの完全準同型暗号（FHE）の研究をまとめる．公開鍵暗号である FHE は，(秘密鍵, 公開鍵)からなる鍵ペア(sk, pk)，t個の平文m_iとこれに対応する暗号文$c_i \leftarrow Encrypt(pk, m_i)$，任意の回路$C$に対し，以下の式(1)が成り立つ暗号系である．なお，$Evaluate(pk, C, c)$は，公開鍵pkを用いて暗号文cに対して計算回路Cを適用することを表す．

$$c' \leftarrow Evaluate(pk, C, \{c_1, c_2, \dots, c_t\}) \Rightarrow Decrypt(sk, c') = C(m_1, m_2, \dots, m_t) \quad (1)$$

ビット演算を考えると 1bit の平文に対して AND と OR が成り立てば任意の計算を行えるため，加法と乗法からなるCについて式(1)が成り立てば FHE である．なお，式(1)が成り立つCが加法もしくは乗法のみから成るとき，それぞれの暗号系を加法準同型暗号，乗法準同型暗号と呼ぶ．

暗号化されたデータに対して任意の計算を行う手法は，RSA 暗号発表[4]の翌年 1978 年に Rivest ら[12]が初めて概念を示した．Rivest らは 4 節で述べる委託計算型のデータマイニングをタイムシェアリングシステム上で実行するため，完全準同型暗号の方針を示した．RSA 暗号，Paillier 暗号[7]や ElGamal 暗号[11]などはすべて暗号文上で平文に対する加算もしくは乗算の片方のみが可能である．2005 年には Boneh らにより任意回数の加算と 1 回の乗算が可能な BGN 暗号[13]が提案されるが，任意の回数の加乗算がともに可能な暗号の具体的な実現は 30 年間，未解決であった．

2009 年に Gentry が FHE の具体的な構成法を発表するが，そのままでは計算機上に実装できないほどの計算量であった[8]．以降，FHE の膨大な計算量の削減と暗号文長の短縮を目指して研究が進められ，大幅な高速化が図られてきた．

3.1 FHE の構成法

現在提案されているすべての FHE は，暗号文に毎回ランダムなノイズを付加している．このノイズは暗号の解読困難性を保つために必要不可欠である．直感的に考えれば「誤差がなければ，ある平文を暗号化したものは同一となり，総当たり方式で見破ることができる」ことになる．

平文を暗号化したものにはノイズが含まれるため，加算や乗算を繰り返すたびにノイズが増加する．特に，乗算では初期の手法では指数的に，新しい手法でもほぼ線形的に増加する．そして，ノイズ部分がある閾値（具体的には暗号文空間の法）を超えると本来の結果とノイズとを分離できなくなり，これは平文を正しく復号できないことを意味する．Gentry はこの問題を解決するために bootstrapping と呼ばれる蓄積されたノイズを削減する手法を提案したが，3.3 項で述べるように 30 秒～30 分もの計算が必要となる[14]．そのため，乗算時のノイズの増え方が緩やかになるようにして bootstrapping の実行回数を少なくする手法が提案されている[15][16].

FHE は，Somewhat HE (SHE), leveled FHE, FHE の 3 種類に分類できる[8][9]．これは，その方式が扱うことのできる計算回数（回路の深さと呼ばれ，鍵生成時に決定する）によって分類される．

SHE は計算可能な乗算の回数を限定し，leveled FHE や FHE よりも高速に動作する暗号である．乗算の上限回数を多く確保しようとすると，その分生成されるパラメータサイズが大きくなってしまい，乗算のたびに暗号文のサイズが大きくなる．実用上は少ない回数の計算（加算の制限はほぼ無いが，乗算は一般的に 2,3 回程度）が許される．

Leveled FHE は，bootstrapping を用いず，事前に決めた回数までの乗算を可能とする暗号である．生成されるパラメータは計算の上限回数の増加に対して緩やかに大きくなり，計算を繰り返しても暗号文のサイズが増加せず，復号回路の計算量がセキュリティパラメータの多項式で抑えられる．SHE よりも計算が遅いが，乗算回数の上限が小さくない．

FHE は，計算で可能な回数に上限のない方式を指す．既存手法では，SHE または leveled FHE に対して bootstrapping を用いることが唯一の FHE 構成の手段である．

FHE のアプリケーションを考える際，暗号文上での計算が 2,3 回程度であれば SHE を，2,3 回よりは多いが回数がある（大きくない）上限回数以下に抑えられる場合は leveled FHE を，事前に見積もることができない場合や大きな回数の場合は FHE を用いる．

3.2 FHE の方式分類[1]

FHE は 2009 年に Gentry により初めて提案された[8]．翌 2010 年には Smart らが Gentry の方式よりも空間・時間計算量を削減した SV 方式[17]を提案した．Gentry による方式と SV 方式はイデアル格子をベースにしているが，イデアル格子で保証されるある解読困難性が多項式時間の量子アルゴリズムで解読できるという指摘[18]もある．なお，FHE は量子コンピュータ（量子アルゴリズム）でも解読できないとされる困難性を暗号ベースにしている．

2010 年には，近似 GCD（Approximate greatest common divisor problem，誤差を含んだ数値から最大公約数を求め

[1] 暗号方式の名称は主に提案論文の著者の頭文字を並べている．

る）問題を暗号ベースとした DGHV 方式[19]が提案された．

2011 年に Smart らは，複数の整数をベクトルとして扱い 1 つの暗号文に暗号化した上で，ベクトルの要素同士の演算をする SIMD 計算（Single-Instruction Multiple-Data operations）を提案[20]した．2011 年に Brakerski ら[15]は modulus switching というノイズ削減手法を提案した．これを用いて，2012 年に RLWE（Ring Learning With Error）問題を暗号ベースとする BGV 方式[21]が提案された．BGV 方式は初めて平文空間を 1bit から複数ビットに拡張する手法も示した．また，2012 年には他に，NTRU（N-th Degree Truncated Polynomial Ring，多項式環上の格子の最短ベクトル問題に基づく暗号）をベースに FHE を構成した LTV 方式[22]も提案された．

2012 年には Brakerski により，modulus switching より計算量の抑えられるノイズ削減技術が提案[16]され，前述の各方式の後継として次の方式が提案された．

- BGV 方式→FV 方式[23]
- LTV 方式→YASHE 方式[24]
- DGHV 方式→ CLT 方式[25]

FV 方式と YASHE 方式は Lepoint らにより leveled FHE として実装され比較が行われた．FV はノイズ増加が小さく，YASHE は計算が高速であると報告されている[26]．

2013 年に Gentry らにより提案された GSW 方式[27]は，暗号文が行列で表現され，これまでの方式では乗算時に必要だったパラメータ（evaluation key と呼ばれる）が不要になった．時間計算量が増えるものの空間計算量を減らした新たな FHE の構成法を示した．また，GSW 方式を基に，ID ベース完全準同型暗号や属性ベース完全準同型暗号を構成できる．

3.3 主な実装

FHE の計算機上への実装は 2010 年の Smart らによる SV 方式が最初であるが，bootstrapping に対応していない Somewhat HE であった．Bootstrapping を含む完全な FHE の実装は 2011 年の Gentry らによる実装[28]が最初である．このとき，1 つの暗号文に 1bit の平文が対応し，暗号文 1 つあたりの bootstrapping に 30 秒から 30 分[1]（パラメータによる）を要した．

最初に公開された実装のソースコードは Perl ら[29]による SV 方式の libScarab[2]である．現在広く利用されている実装は，2012 年に IBM が BGV 方式を実装した HElib[3]である[30][31]．2014 年には HElib に bootstrapping が実装され[14]，完全な FHE の実装となった．このとき 1 つの暗号文に対応する平文は任意のビットサイズから選んだベクトル（例えば2^{16}を法とする2^{10}次ベクトル）を扱うことができ，1 回の bootstrapping に平文空間を 16bit とすると約 5 分[4]要した．

Ducas ら[32]は，2015 年に FHEW[5]と呼ばれる GWS 方式の FHE の実装を公開した．また，leveled FHE では，2014 年に Aslett ら[33]が FV 方式を C++言語で実装し R 言語向けにパッケージ化したライブラリ[6]，2015 年と 2016 年に MS Research が YASHE 方式と FV 方式をそれぞれ実装した SEAL[7]などが公開されている．

4. プライバシー保護データマイニング

プライバシー保護データマイニング（PPDM）は，2000 年に Lindel ら[2] と Agrawal ら[3]により発表された論文に端を発し，以降，盛んに研究が行われている．PPDM の研究は，対象とするデータの保持形態から「分散保持」と「集約保持」に分類ができる．クラウドへの委託計算の概念が注目される以前は主に，複数ノードに分散保持するデータに対して，各ノードが他ノードの情報を得ることなく，それらのデータセットの和集合に対するデータマイニングを実行して結果を共有することを目的としていた．しかし近年は，委託計算を想定した集約保持されたデータに対する研究も行われるようになっている．データが保持される形態が異なれば，これまでの PPDM とは異なるプロトコルが必要となる．以下， PPDM で想定されるノード間の関係モデルと要素技術をまとめる．

4.1 ノード間のモデル

以下では，PPDM において想定されるノード関係を「マルチパーティ」「委託計算」「マルチパーティ委託計算」の 3 種類に分類し説明する．なお，PPDM では多くの場合，あるノードが semi-honest であると仮定している．Semi-honest（passive, honest-but-curious）とは，決められたプロトコルには従うが，その過程で取得可能な情報を蓄積し情報を推測するモデルである[34]．Semi-honest の他に，malicious（active）モデルがあり，このモデルでは他のノードとの結託や通信内容の改ざんを行うが，PPDM で対象とする事例は少ない．これは，semi-honest を想定したほうが malicious モデルよりも単純で計算量の少ないプロトコルを構成できるためであるが，実世界を想定すると malicious モデルを考えるべきだという主張もある[35]．

4.1.1 マルチパーティモデル

マルチパーティモデルは，Lindel ら[2] と Agrawal ら[3]に続く PPDM の研究で主に想定されるモデルである．データを保持する複数のノードが，それらのデータを互いに公開することなく，統合したデータに対するデータマイニングの結果のみを共有する場面を想定する．データを保持するノード間で通信を繰り返すことにより，目的の結果を得る．

4.1.2 委託計算モデル

委託計算モデルではクライアントとサーバ（クラウド）の 2 ノードを想定し，クラウドを semi-honest とみなす．クライアントが秘匿状態のデータをサーバに送信し，サーバは秘匿状態のまま処理を行うことで，クライアントだけが結果を得ることを目的とするモデルである．

例えば，クライアントが所持するデータを秘匿してサーバに保管し，このデータに対する処理を必要に応じて行う場面や，秘匿されたクエリに対して，サーバ上のデータベースを参照して処理を行う場面が考えられる．

4.1.3 マルチパーティ委託計算モデル

マルチパーティの委託計算モデルでは，複数のクライアントと 1 つのサーバを想定する．複数のクライアントが別々に秘匿化したデータをサーバに保管し，それらを統合したデータからデータマイニング結果を得る．

[1] Intel Xeon E5450, 3GHz CPU, 24GB of RAM
[2] https://github.com/hcrypt-project/libScarab
[3] https://github.com/shaih/HElib
[4] Intel Xeon X5570, 2.93GHz CPU
[5] https://github.com/lducas/FHEW
[6] http://www.louisaslett.com/HomomorphicEncryption
[7] http://sealcrypto.codeplex.com

例えば，準同型暗号を用いると，異なる鍵で暗号化されたデータ間では準同型演算は行えない．そのため，一般に他の 2 つの場合よりも技術的にハードルが高い．信頼できる第三者の存在を仮定し，委託計算モデルと同じプロトコルが適用できるようにする場合もある．

4.2 PPDM における要素技術

PPDM 実現のためのアプローチには，大きく分けて，データの改変によるものと暗号技術によるものの 2 つがある．

4.2.1 データの改変

データの改変に分類される技術には，個人を特定できる情報を削除するため，データの匿名化を行う k-匿名性[5]や摂動（perturbation）技術に基づく手法[3]などがある．匿名化による手法は，1 つのデータで匿名性が守られていても，複数の匿名化データを統合したときに個人情報が明らかになる可能性がある．また，摂動技術に基づく手法はデータマイニングアルゴリズムの入力として与えられるデータに対して，事前にランダムなノイズを加えることでプライバシーを保護する．このとき，加えたノイズを知っている者はデータマイニングの出力からノイズをできるだけ削減できるように，データマイニングの種類に応じてノイズの加え方を工夫することができる．しかし，ノイズの付加されたデータから，元のデータの一部を推測できることがある．例えば，元のデータに特異な点がある場合や元のデータがほぼすべて同じ値である場合は推測が容易になる．また，ノイズ付加の程度が強いとデータマイニングの結果の正確さが低くなるため，プライバシー保護とデータマイニングの結果の正確さがトレードオフの関係にある．

4.2.2 暗号技術

暗号技術に基づく手法は，入力データにノイズを加えることなく，暗号化することによりプライバシーを保護する．摂動技術が持つ問題は生じないが，一般に計算量が増加する．ここでは代表的な秘匿回路，秘密分散，準同型暗号を用いた手法を取り上げる．

秘匿回路（Garbled Circuit）を用いた手法[36]はマルチパーティモデルを前提とし，Multi-Party Computation（MPC）とも呼ばれる．最初に目的のデータマイニングを AND と OR のゲートの組み合わせ回路で表現し，全ノード間で共有する．その各ゲートの入出力を，第三者の助けを得ずにノード間で秘匿しながら通信を行うことでデータマイニングの結果を得る技術である．FHE と比べて通信量と計算量ともに増加する報告がある[39][45]．また，計算はデータを保持するノードが行う必要があり，委託計算モデルでは使えない．

鍵の分散管理を目的としていた秘密分散（Secret Sharing）を PPDM に適用する手法[37][38]は，保持するデータを複数箇所に分散して保管し，そのうちの一定数以上が集まって初めて元のデータを取り戻せる性質を利用する．FHE と同様に和と積が計算できるため，任意の関数が計算可能である．一般に準同型暗号や Garbled Circuit よりも計算量が小さいが，データの委託先が複数必要であり，計算の際には関係する全ノード間での通信が必要になる．

準同型暗号を用いると，ノード間のモデルに関わり無く，第三者を必要とせずに PPDM を行うことができる．準同型暗号の中でも特に FHE は暗号化状態のまま任意の計算が行えるため，加法や乗法だけの部分準同型暗号よりもデータ利用の幅が広い．FHE を用いた PPDM の課題は 6 節で扱う．

5. FHE のデータマイニングへの実装報告

完全準同型暗号の具体的な実現法が示されて以降，PPDM を FHE により実現する手法が複数提案されている．その中から代表的なものを表 1 にまとめる．FHE の応用先にはゲノム情報や医療情報を想定する研究が多く見られた．コストが高くても個人情報を守りながらデータの活用が必要とされる領域であると考えられる．

また，多くの場合で bootstrapping を使用せず，leveled FHE を用いていた．これは，3.3 項で述べたように bootstrapping 処理は時間がかかるためだと考える．

今回収集した論文中の実装で使われた FHE の方式を図 3 にまとめた[1]．このうち，3.3 項で述べたライブラリを使用しているものが 22 件（うち 15 件が HElib），独自に実装しているものが 13 件であった．FHE の方式は，3.2 項で述べたように FV，YASHE，CLT が効率的と考えられるが，それらを用いたものは少ない．これらの方式を実装した高速なライブラリが公開されていないためと考える．

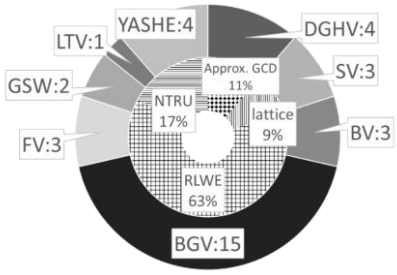

図 3. DM における FHE 実装の内訳
（内側は暗号ベースを，外側は FHE 方式を示す）

6. 研究課題と既存手法

本節では，FHE を用いたデータマイニングにおける課題を以下の 2 つに分類し，従来の解決手法を示す．

1. 暗号文上での計算効率
2. 大きな暗号文に起因する計算時間と通信量

6.1 暗号文上での計算効率

FHE では，データを暗号化する際にビット単位で暗号化する場合と整数のまま暗号化する場合で計算効率が異なる．ビット単位でデータを暗号化した場合，計算はビット演算で実現することになる．このため，加乗算だけでなく除算や指数対数演算など任意の計算が可能となる．しかし，CPU が持つ加算器や乗算器を用いることができずソフトウェアで実現する必要があるため，演算に時間がかかる．一方で，平文空間を大きくして整数のまま暗号化すると，CPU の演算器の機能を用いて効率的に計算ができるが，加乗算以外の計算を扱うのは困難になる．さらに，暗号化されたデータに対して比較演算を行って得られる結果も暗号化されているため，比較を用いた命令の分岐ができない．

本項では，これら計算手法の工夫を述べた後，比較結果が暗号化されていることに対する工夫を述べる．

[1] 同一著者による同一アプリケーションの実装は 1 つにまとめた．また，プロトコル提案にとどまる論文もあった．そのため，2.3 項での論文総数とは一致しない．

表 1. FHE を用いたデータマイニング関係の代表的な実装報告

	著者	研究拠点	モデル (4.1 節参照)	方式 実装	実装詳細，パフォーマンスなど
Apriori	Kaosar ら (2012) [39]	ビクトリア大学 オーストラリア	2 パーティ	SV libScarab	水平分割*2 されたデータセットに対する手法の提案．通信を繰り返すが，Garbled circuit よりも高速．著者らは FP Tree の構成法も別の論文[40]で提案．
	Liu ら (2015) [41]	浙江工商大学 中国	委託計算	DGH 記載なし	5000 トランザクション（10 アイテム/トランザクション）に対して 100 秒程度．（Intel Core i5, 2.5GHz CPU, 8GB RAM）
	高橋ら (2016) [42]	早稲田大学 日本	委託計算	BGV HElib	SIMD 演算を用いて Liu らの手法より 10 倍の高速化．128bit セキュリティ．
LDA	Graepel ら (2013) [43]	MS Research Cambridge イギリス	委託計算	FV 独自実装	FHE を用いた初めての機械学習．Linear Means Classifier と Fisher’s Linear Discriminant Classifier の実装．128bit セキュリティ（LM），80bit セキュリティ（FLD）特徴量 10bit，100 アイテムの学習に LM は 2.6 秒，FLD は 5.5 時間．（Intel Core i7, 2.8GHz CPU, 8GB RAM）
SVM	Liu ら (2015) [44]	南洋理工大学 シンガポール	2 パーティ 委託計算	BGV HElib	垂直分割*2 されたデータセットに対する手法の提案．16bit セキュリティ．アイテム数 400 で 30 分，1000 で 3 時間．（8x Intel Xeon E5-1620, 3.6GHz CPU, 8GB RAM）
決定木	Bost ら (2015) [45]	MIT CSAIL アメリカ	委託計算	BGV HElib	二分決定木をあらかじめ式で表し，FHE 上で実現．サーバは学習済みのモデルのパラメータを持ち，クライアントが識別したいデータを持つことを想定．80bit セキュリティ．深さ 4，ノード数 6 の木に対する識別 1 回に約 0.9 秒．（Intel Core i7, 2.66GHz, 8GB RAM）
	Khedr ら (2015) [46]	トロント大学 カナダ	委託計算	GSW 独自実装	GPU に最適化した FHE を独自に実装し，HElib と比べ暗号文長を 2/3 ほどに削減し乗算を 1000 倍以上の高速化．決定木以外に，キーワードマッチによるメールフィルターやワード検索も実装．80bit セキュリティ．（Intel Core i7, 3.5Hz CPU, 32GB RAM, NVIDIA GeForce GTX980）
ランダムフォレスト ロジスティック回帰	Aslett ら (2015) [47]	オックスフォード大学 アメリカ	委託計算	FV R-package	実際に Amazond EC2 インスタンス上にて実行し評価．ベイズ分類で 10 秒前後．深さ 3 のランダムフォレストに約 6 分．（36x Intel Xeon E5-2666v3, 2.9GHz CPU 相当の環境）
GWAS*1 関連研究	Bos ら (2014) [48]	MS Research Redmond アメリカ	委託計算	YASHE Lepoint らの実装[26]	ロジスティック回帰，COX の比例ハザードモデルを実装し，実際の委託計算のシナリオを想定して Windows Azure 上で実験．
	Lauter ら (2015) [49]	MS Research Redmond アメリカ	マルチパーティ 委託計算	YASHE 記載なし	遺伝子配列に対する Hardy-Weinberg Equilibrium (HWE), Linkage Disequilibrium (LD)，EM-algorithm，Cochran-Armitage Test for Trend (CATT)を実装．80bit セキュリティ．識別に，平文空間が小さいと 1 秒以内，大きくて 5 秒程度．（Intel Core i7-3770S, 3.10GHz CPU, 8GB RAM）
	Lu ら (2015) [50]	筑波大学 日本	マルチパーティ 委託計算	BGV HElib	カイ 2 乗テストと LD を実装し評価．前処理として度数分布表を作成することで大規模なデータの解析を実現．Lauter らの手法より約 10,000 倍の高速化．（Intel Xeon, 2.6GHz CPU, 32GB RAM）
	Wang ら (2015) [51]	カリフォルニア大学 アメリカ	マルチパーティ 委託計算	BGV HElib	初めて rare variant（まれな変異）に対する解析を行うため，FHE 上での精確ロジスティック回帰(exact logistic regression)における p 値の計算．これらの実現のため，棄却サンプリングや暗号文上での比較も実装．
ニューラル ネットワーク	Dowlin ら (2016) [52]	MS Research Redmond アメリカ	委託計算	YASHE SEAL	5 層ネットワーク（途中の層の線形計算部分をまとめることで実際の計算は 3 層分）で 28×28 ピクセル（1byte/ピクセル）の画像の入力を，5 万件/1 時間以上の速度で識別．（Intel Xeon E5-1620, 3.5GHz CPU, 16GB RAM）
ハミング距離による パターンマッチ 生体認証	Yasuda ら (2015) [53]	富士通研究所 日本	委託計算	BV 独自実装	C 言語とアセンブリで独自の SWHE の実装．特定の計算に向いた暗号文のパッキング手法を利用．適用先として DNA 配列や指紋認証などを挙げる．平文空間 14bit のとき，暗号化，内積計算，復号化にそれぞれ 0.1 秒程度．64bit セキュリティ．（Intel Xeon X3480, 3.07GHz CPU, 16 GB RAM）
Private Information Retrieval	Cheon ら (2015) [54]	ソウル大学校 韓国	委託計算	BGV HElib	暗号化 DB から指定した条件に合致するデータにのみ計算を行い，結果を返すシステムの提案．例えば，10,000 件のデータからある値以上のデータの個数や和を返すクエリに約 50 分．（Intel Xeon 2.3 GHz CPU, 192GB RAM）

*1 Genome Wide Association Study (GWAS)：遺伝子配列の中でも個人差がある部分に注目した疾病や形質との関連の統計的調査．
GWAS では，信頼できる第三者の存在を仮定したマルチパーティ委託計算モデルを想定することが多い．
*2 データの水平分割：データエントリで分割しエントリに対する属性を共有する．垂直分割：データエントリを共有し，属性を分割する．
※実験環境の説明の粒度が一致していないが，それぞれの論文中の記述に従い，表記のみ統一した．

6.1.1 計算手法の工夫

除算を避けるための工夫として，平文で扱うときとは異なる計算方法を用いる手法がある．Graepel ら[43]は，フィッシャーの線形判別器を実装し評価した．この学習段階は一般に固有値ベクトルを用いることで簡単に計算できるが，FHE 上で固有値を求める際に必要な除算を避けるため，最急降下法を用いて近似する手法を示した．さらに，繰り返しの収束判定ができないため，繰り返し回数をあらかじめ固定している．この論文の著者らの実験では，30bit の特徴量に対して学習と識別を行い，3 回以上繰り返すことで平文での実行時とほぼ同じ識別結果を得たと報告されている．

テイラー展開を用いて式を加算と乗算のみに近似する手法がある．Bos ら[48]はテイラー展開を用いて式(2)のようにロジスティック関数を多項式に近似し回帰計算を実装し評価した．加えて，このときの多項式の係数は分数になるため，すべて整数になるように式(3)のように正規化を行った．x^7の項まで展開することで，$f(x)$の値において小数点 2 桁までの精度を保持する結果を得たと報告されている．

$$f(x) = \frac{e^x}{e^x + 1} = \frac{1}{2} + \frac{1}{4}x - \frac{1}{48}x^3 + \frac{1}{480}x^5 - \frac{17}{80640}x^7 + O(x^9) \quad (2)$$

$$F(x) = 80640f(x) = -17\mathrm{x}^7 + 168x^5 - 1680x^3 + 20160x + 40320 \quad (3)$$

また，式(3)のように正規化を用いて小数を扱うことを避けるのではなく，FHE 上で直接，固定小数点や浮動小数点演算を扱う手法の提案[55][56]もある．

6.1.2 比較に対する工夫

委託計算モデルにおいて，比較結果に依存して次に行うべき計算が大きく異なるとき，一旦クラウドからクライアントにデータを送り返し，クライアントが復号して比較を行う手法がある．Liu ら[40]は，頻出パターンマイニングの Apriori を実装し評価した．Apriori では，数えた個数が最小サポート以上であるか繰り返し判定する必要があるため，確認が必要になるたびにサーバはクライアントにデータを送り返すプロトコルを提案した．

また，比較結果が暗号化された 0 か 1 で得られることを利用し，必要な計算を全て行う方法もある．Bost ら[45]は，二分決定木を用いた識別を実装し評価した．特徴量をビットで表記し，モデルを表す式を事前に用意する．例えば，図 4 の決定木は式(4)の形で表すことで，FHE を用いて計算を行うことができる．ここで，b_1, b_2は特徴量のビット，c_1, c_2, c_3は識別値を表す．

$$b_1(b_2c_3 + (1 - b_2)c_2) + (1 - b_1)c_1 \quad (4)$$

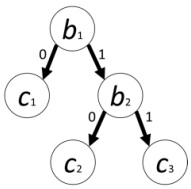

図 4. 二分決定木の例

2 つの暗号化された数値を比較する計算方法は，$m$bit の 2 つの数値に対する比較を$O(\log m)$の計算量で抑えられる Garay らの手法[57]が最速と考えられる．Togan ら[58]は，HElib を用いて FHE 上でこの比較関数を実装し，128bit セキュリティで 16bit の整数の比較に約 30 秒[1]要したと報告している．

データマイニングにおいてソートを行う場面は少なくない．FHE では条件分岐を行わないソーティングネットワークによりソートを構成すると効率が良い．Kim ら[59]はソートの途中で必要な bootstrapping の回数を減らし，後述の SIMD 演算を活用したソート手法を提案した．HElib を用いて実装し，98bit セキュリティで 128bit の整数 16 個のソートに約 14 時間[2]要したと報告されている．

6.2 大きな暗号文に起因する計算時間と通信量

FHE を用いると，平文に対して暗号文が長くなる．例えば，CLT 方式を用いて 4MB のデータを暗号化すると，暗号文長が約 280GB になる[26]．これに対して，暗号文を短くするため，複数の整数をベクトルとして 1 つの暗号文に暗号化（パッキング）した上で，ベクトルの要素同士の演算を実現する技術がある．これにより，ベクトルの大きさの分だけ暗号文の大きさを削減することが可能となる．Smart らの提案した SIMD 演算[20]では，中国剰余定理を用いてパッキングされた値に対して，1 つの計算（加算または乗算）を並列に行うことができる．しかし，それでも 1 つの暗号文が大きいため計算に時間がかかり，また PPDM の際には通信量も大きくなる．

6.2.1 長い計算時間に対する工夫

表 1 で示したように，実装と評価の報告された FHE のアプリケーションは，実用的な時間で計算が終了していない例か，小さなデータに対してのみ実験を行った例がほとんどである．例えば，HElib では 1 回の加算に 0.002 秒，乗算に 3.6 秒[3]要すると報告[46]されている．また，bootstrapping に時間がかかるため，多くの場合で bootstrapping を用いずに leveled FHE を実装している．

SIMD 演算とは異なる手法で並列計算を高速化する手法もある．Yasuda ら[53]は，Lauter ら[10]が最初に提案したパッキング手法を改良して，2 つのベクトルの内積計算に特化された効率的な計算手法を実装し評価した．SIMD 演算のバッチ処理により内積計算を行うには 1 回の乗算とデータの個数の回数の加算が必要だが，この手法では乗算 1 回のみで複数の内積値を得ることができる．この手法を用いてハミング距離の計算を行うことで，DNA 配列のパターンマッチングや生体認証に応用できるとしている．

機械学習では教師データに対して繰り返し学習を行う計算プロセスをとるため，暗号化状態での学習は重い処理となる．例えば，Graepel ら[43]は 10bit の特徴量ベクトル 100 本を教師データとして，6.1.1 項で示したフィッシャーの線形判別器の学習を行う際，55 時間を要したと報告している． 既存手法における機械学習の学習（回帰）段階は，このように bootstrapping が不要になるように学習の繰り返し回数を少なく設定する場合か，モデルが既に完成していると想定する場合に限られている．Dowlin ら[52]はニュー

[1] Intel Xeon E5-1620, 3.60GHz, 12GB of RAM
[2] Intel Core i7 4790K 4GHz, 32GB of RAM
[3] Intel Core i7 5930K, 3.5 GHz, 32 GB of RAM

ラルネットを FHE 上で実装し評価したが，学習は平文で行っている．この論文で著者らは，学習も暗号化状態で行うことは可能であると述べるが，学習済みや学習中のモデル内部のパラメータが分からないため，チューニングや解析ができないという利便性低下の可能性を指摘している．

GPU や FPGA[60]を用いた高速化は成果を出している．Khedr ら[46]は GPU を用いて GSW 方式の leveled FHE を実装した．1 回の乗算の所要時間を HElib と比較して 1000 倍以上の高速化を実現したと報告している．しかし， HElib は SIMD 演算によるバッチ処理を備えるが，著者らは実装していないため，単純な比較はできない．

6.2.2 通信量の大きさ

委託計算ではクラウド上に大量のデータを送信する必要があるため，できる限り暗号文長を短くすることが高速化につながる．Lauter ら[10]は，クライアントとサーバ（クラウド）間の通信において，共通鍵暗号と組み合わせる解決策を提案している．クライアントはデータをクラウドに送信する際に FHE で暗号化を行わない．かわりに，最初に共通鍵暗号である AES の共通鍵を FHE で暗号化したものをサーバへ送信し，その後は送りたいデータを AES で暗号化してサーバへ送信する．サーバは送られてきた AES で暗号化されたデータを FHE で暗号化し，FHE で暗号化されたまま AES 復号処理を行うことで，サーバは FHE で暗号化されたデータを得る．この流れを図 5 に示す．

この手法は，通信量の削減という利点だけでなく，クライアントよりもサーバの方が一般に計算機性能が高いことを生かすことができる．FHE による AES の暗号化・復号化は実際に HElib を用いた実装と評価[30][61]が報告されている．AES-128 の 180 ブロックを並列に処理し約 18 分[1]要した（128bit の 1 ブロックあたり 6 秒）という．他にも CLT 方式での報告[25]，LTV 方式での報告[62]が行われている．また， FV 方式と YASHE 方式の比較のため，ブロック暗号の SIMON を題材とした論文[26]もある．

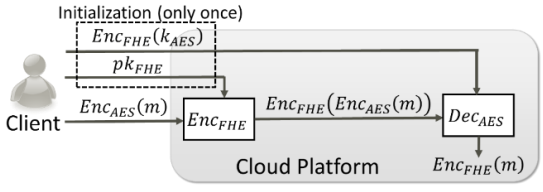

図 5．AES を用いた通信量削減手法
（[26]の Fig.1.を参考に作成）

Lauter らは， Brakerski ら[21]が提案した modulus switching を用いて，サーバからクライアントの通信量削減も提案した．Modulus switching は暗号文中のノイズ削減手法であるが，これを利用して暗号文を「これ以上暗号文上で計算すると正しく復号できなくなるが，サイズの小さい暗号文」に変換できる．

どちらの手法もサーバ側の計算量と 2 者間の通信量がトレードオフの関係にある．

7. おわりに

本稿では，完全準同型暗号（FHE）を用いたプライバシー保護データマイニングの研究動向を調査した．FHE を利用することで，これまでの分散データに対するプライバシー保護データマイニングだけでなく，クラウドコンピューティングへの委託計算においても，プライバシー保護データマイニングを行うことが可能となる．実用化に向けた課題として，暗号文上での効率的な計算手法が限られること，膨大な計算量と通信量があることを述べた．これらの課題に対し，データマイニング手法に工夫を加えるアプローチ，FHE の暗号方式に改良を加えるアプローチ，ハードウェアを活用するアプローチなどから研究が進められており，今後もより一層研究が進められると期待される．

我々は今後，今回のサーベイを基にして高速な汎用 FHE ライブラリの構築とこれを用いたデータマイニング手法の研究を進め，随時報告を行う所存である．

謝辞

本研究の一部は，JST CREST の支援によるものである．

参考文献

[1] “平成 26 年通信利用動向調査の結果”, 総務省, (2015).
[2] Yehuda Lindell, Benny Pinkas, "Privacy Preserving Data Mining", CRYPTO 2000, LNCS, vol. 1880, pp. 36–54, (2000).
[3] Rakesh Agrawal, Ramakrishnan Srikant, "Privacy-preserving data mining", Proc. of the ACM SIGMOD 2000, vol. 29, no. 2, pp. 439–450, (2000).
[4] Cynthia Dwork, "Differential Privacy", Automata, Languages and Programming, LNCS, vol. 4052, pp. 1–12, (2006).
[5] Latanya Sweeney, "k-anonymity: A model for protecting privacy", Int’l J. on Uncertainty, vol. 10, no. 5, pp. 557–570, (2002).
[6] Ronald L. Rivest, Adi Shamir, Len Adleman, "A method for obtaining digital signatures and public-key cryptosystems”, CACM, vol. 21, no. 2, pp. 120-126, (1978).
[7] Pascal Paillier, "Public-key cryptosystems based on composite degree residuosity classes", EUROCRYPT ’99, LNCS, vol. 1592, pp. 223–238, (1999).
[8] Craig Gentry, "A fully homomorphic encryption scheme", PhD Thesis, Stanford University, (2009).
[9] Frederik Armknecht, Colin Boyd, Christopher Carr, Kristian Gjosteen, Angela Jaschke, Christian A. Reuter, Martin Strand, "A Guide to Fully Homomorphic Encryption", Cryptology ePrint Archive 2015/1192, (2015).
[10] Kristin Lauter, Michael Naehrig, Vinod Vaikuntanathan, "Can homomorphic encryption be practical?", Proc. of the 3rd ACM workshop on CCSW ’11, pp. 113–124, (2011).
[11] Taher Elgamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms", IEEE Trans. on Information Theory, vol. 31, no. 4, pp. 469–472, (1985).
[12] R. L. Rivest, L. Adleman, M. L. Dertouzos, "On Data Banks and Privacy Homomorphisms", Foundations of Secure Computation, pp. 169–180, (1978).
[13] Dan Boneh, Eu-Jin Goh, Kobbi Nissim, "Evaluating 2-DNF formulas on ciphertexts", Theory of Cryptography, LNCS, vol. 3378, pp. 325–341, (2005).
[14] Shai Halevi, Victor Shoup, "Bootstrapping for HElib", EUROCRYPT 2015, LNCS, vol. 9056, pp. 641–670, (2015).
[15] Zvika Brakerski, Vinod Vaikuntanathan, "Efficient Fully Homomorphic Encryption from (Standard) LWE", 2011 IEEE 52nd Annual Symp. on FOCS, pp. 97–106, (2011).
[16] Zvika Brakerski, "Fully homomorphic encryption without modulus switching from classical GapSVP", CRYPTO 2012, LNCS, vol. 7417, pp. 868–886, (2012).
[17] Nigel P. Smart, Frederik Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes", PKC 2010, LNCS, vol. 6056, pp. 420–443, (2010).
[18] Ronald Cramer, Léo Ducas, Chris Peikert, Oded Regev, "Recovering Short Generators of Principal Ideals in Cyclotomic Rings", EUROCRYPT 2016, LNCS, vol. 9666, pp. 559–585, (2016).

[1] Intel Core i5-3320M, 2.6GHz, 4GB of RAM

[19] Marten Van Dijk, Craig Gentry, Shai Halevi, Vinod Vaikuntanathan, "Fully Homomorphic Encryption over the Integers", EUROCRYPT 2010, LNCS, pp. 24–43, (2010).
[20] Nigel P. Smart, Frederik Vercauteren, "Fully homomorphic SIMD operations", Designs, Codes, and Cryptography, vol. 71, issue. 1, pp. 57–81, (2014).
[21] Zvika Brakerski, Craig Gentry, Vinod Vaikuntanathan, "(Leveled) Fully Homomorphic Encryption without Bootstrapping", Proc. of the 3rd ITCS, pp. 309–325, (2012).
[22] Adriana López-Alt, Eran Tromer, Vinod Vaikuntanathan, "On-the-fly multiparty computation on the cloud via multikey fully homomorphic encryption", Proc. of the 44th STOC, pp. 1219–1234, (2012).
[23] Junfeng Fan, Frederik Vercauteren, "Somewhat Practical Fully Homomorphic Encryption", Cryptology ePrint Archive 2012/144, (2012).
[24] Joppe W. Bos, Kristin Lauter, Jake Loftus, Michael Naehrig, "Improved Security for a Ring-Based Fully Homomorphic Encryption Scheme", Cryptography and Coding, LNCS, vol. 8308, pp. 45–64, (2013).
[25] Jean-Sebastien Coron, Tancrède Lepoint, Mehdi Tibouchi, "Scale-invariant fully homomorphic encryption over the integers", PKC 2014, LNCS, vol. 8383, pp. 311–328, (2014).
[26] Tancrède Lepoint, Michael Naehrig, "A Comparison of the Homomorphic Encryption Schemes FV and YASHE", AFRICACRYPT 2014, LNCS, vol. 8469, pp. 318–335, (2014).
[27] Craig Gentry, Amit Sahai, Brent Waters, "Homomorphic Encryption from Learning with Errors: Conceptually-Simpler, Asymptotically-Faster, Attribute-Based", CRYPTO 2013, LNCS, vol.8042, pp. 75–92, (2013).
[28] Craig Gentry, Shai Halevi, "Implementing Gentry's fully-homomorphic encryption scheme", EUROCRYPT 2011, LNCS, vol. 6632, pp. 129–148, (2011).
[29] Henning Perl, Michael Brenner, Matthew Smith, "Poster: an implementation of the fully homomorphic smart-vercauteren crypto-system", Proc. of the 18th ACM Conf. on CCS, pp. 837–840, (2011).
[30] Craig Gentry, Shai Halevi, Nigel P. Smart, "Homomorphic Evaluation of the AES Circuit", CRYPTO 2012, LNCS, pp. 850–867, (7417).
[31] Shai Halevi, Victor Shoup, "Design and Implementation of a Homomorphic-Encryption Library", IBM Research, (2012), http://researcher.ibm.com/researcher/files/us-shaih/he-library.pdf, accessed 2016-05-06.
[32] Léo Ducas, Daniele Micciancio, "FHEW: Bootstrapping homomorphic encryption in less than a second", EUROCRYPT 2015, LNCS, pp. 617–640, (2015).
[33] Louis J. M. Aslett, Pedro M. Esperança, Chris C. Holmes, "A review of homomorphic encryption and software tools for encrypted statistical machine learning", arXiv:1508.06574, (2015).
[34] Oded Goldreich, "Foundations of Cryptography – A Primer", Foundations and TrendsTM in Theoretical Computer Science, (2005).
[35] Murat Kantarcioglu, Onur Kardes, "Privacy-Preserving Data Mining in the Malicious Model", Int'l J. of Information and Computer Security, vol. 2, no. 4, pp. 353–375, (2008).
[36] Andrew Chi-Chih Yao, "How to generate and exchange secrets", 27th Annual Symp. on Foundations of Computer Science, pp. 162–167, (1986).
[37] Jake Loftus, Nigel P. Smart, "Secure outsourced computation", AFRICACRYPT 2011, LNCS, vol. 6737, pp. 1–20, (2011).
[38] M. Ben-Or, Shafi Goldwasser, Avi Wigderson, "Completeness Theorems for Non-Cryptographic Fault Tolerant Distributed Computation", Proc. of the 20th STOC, pp. 1–10, (1988).
[39] Mohammed Golam Kaosar, Russell Paulet, Xun Yi, "Fully homomorphic encryption based two-party association rule mining", Data & Knowledge Engineering, vol. 76–78, (2012).
[40] Mohammed Golam Kaosar, Xun Yi, "Secure Two-Party Association Rule Mining Based on One-Pass FP-Tree", Int'l J. of Information Security and Privacy, vol. 5, no. 2, pp. 13-32, (2011).
[41] Junqiang Liu, Jiuyong Li, Shijian Xu, Benjamin C. M. Fung, "Secure Outsourced Frequent Pattern Mining by Fully Homomorphic Encryption", Big Data Analytics and Knowledge Discovery, LNCS, vol.9263, pp. 70–81, (2015).
[42] 高橋 卓巳, 石巻 優, 山名 早人, "SV パッキングによる完全準同型暗号を用いた 安全な委託 Apriori 高速化", DEIM Forum 2016, (2016).
[43] Thore Graepel, Kristin Lauter, Michael Naehrig, "ML Confidential Machine Learning on Encrypted Data", Information Security and Cryptology – ICISC 2012, LNCS, vol. 7839, (2013).
[44] Fang Liu, Wee Keong Ng, Wei Zhang, "Encrypted SVM for Outsourced Data Mining", 2015 IEEE 8th Int'l Conf. on Cloud Computing, pp. 1085–1092, (2015).
[45] Raphaël Bost, Raluca Ada Popa, Stephen Tu, Shafi Goldwasser, "Machine learning classification over encrypted data", Cryptology ePrint Archive 2014/331, (2014).
[46] Alhassan Khedr, Glenn Gulak, Senior Member, Vinod Vaikuntanathan, "SHIELD: Scalable Homomorphic Implementation of Encrypted Data-Classifiers", IEEE Trans. on Computers, (2015).
[47] Louis J. M. Aslett, Pedro M. Esperança, Chris C. Holmes, "Encrypted statistical machine learning: new privacy preserving methods", arXiv :1508.06845, (2015).
[48] Joppe W. Bos, Kristin Lauter, Michael Naehrig, "Private predictive analysis on encrypted medical data", J. of Biomedical Informatics, vol. 50, pp. 234–243, (2014).
[49] Kristin Lauter, Adriana López-Alt, Michael Naehrig, "Private Computation on Encrypted Genomic Data", LATINCRYPT 2014, LNCS, vol. 8895, pp. 3–27, (2015).
[50] Wen-Jie Lu, Yoshiji Yamada, Jun Sakuma, "Privacy-preserving Genome-wide Association Studies on Cloud Environment using Fully Homomorphic Encryption", BMC Medical Informatics and Decision Making, vol. 15, Suppl. 5, (2015).
[51] Shuang Wang, Yuchen Zhang, Wenrui Dai, Kristin Lauter, Miran Kim, Yuzhe Tang, Hongkai Xiong, Xiaoqian Jiang, "HEALER: Homomorphic computation of ExAct Logistic rEgRession for secure rare disease variants analysis in GWAS", Bioinformatics, vol. 32, no. 2, pp. 211–218, (2015).
[52] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, John Wernsing, "CryptoNets: Applying neural networks to Encrypted data with high throughput and accuracy - Microsoft research", Microsoft Research TechReport 2016-3, (2016).
[53] Masaya Yasuda, Takeshi Shimoyama, Jun Kogure, Kazuhiro Yokoyama, Takeshi Koshiba, "New packing method in somewhat homomorphic encryption and its applications", Security and Communication Networks, vol. 8, no. 13, pp. 2194–2213, (2015).
[54] Jung Hee Cheon, Miran Kim, Myungsun Kim, "Search-and-compute on encrypted data", Financial Cryptography and Data Security, LNCS, vol. 8976, pp. 142–159, (2015).
[55] Seiko Arita, Shota Nakasato, "Fully Homomorphic Encryption for Point Numbers", Cryptology ePrint Archive 2016/402, (2016).
[56] Jung Hee Cheon, Andrey Kim, Miran Kim, Yongsoo Song, "Floating-Point Homomorphic Encryption", Cryptology ePrint Archive 2013/421, (2016).
[57] Juan Garay, Berry Schoenmakers, José Villegas, "Practical and Secure Solutions for Integer Comparison", PKC 2007, LNCS, pp. 330–342, (2007).
[58] Mihai Togan, Cezar Pleşca, "Comparison-based computations over fully homomorphic encrypted data", IEEE Int'l Conf. on Communications, pp. 1–6, (2014).
[59] Pyung Kim, Younho Lee, Hyunsoo Yoon, "Sorting Method for Fully Homomorphic Encrypted Data Using the Cryptographic Single-Instruction Multiple-Data Operation", IEICE Trans. on Communications, vol. E99.B, no. 5, pp. 1070–1086, (2016).
[60] David Bruce Cousins, Kurt Rohloff, Chris Peikert, Rick Schantz, "An update on SIPHER (Scalable Implementation of Primitives for Homomorphic EncRyption) - FPGA implementation using Simulink", 2012 IEEE Conf. on High Performance Extreme Computing, (2012).
[61] Craig Gentry, Shai Halevi, Nigel P. Smart, "Homomorphic Evaluation of the AES Circuit (Updated Implementation)", Cryptology ePrint Archive 2012/099, (2015).
[62] Yarkın Doröz, Yin Hu, Berk Sunar, "Homomorphic AES Evaluation using NTRU.", Cryptology ePrint Archive 2014/039, (2014).