様　式　Ｃ－１９、Ｆ－１９、Ｚ－１９　（共通）

科学研究費助成事業　　研究成果報告書

平成 ２８ 年　 ６ 月 １８ 日現在

機関番号：３４４１６
研究種目：基盤研究(C)（一般）
研究期間：2013～2015
課題番号：２５３３０１５２

研究課題名（和文）格子問題に基づく暗号プリミティブの構成法に関する研究

研究課題名（英文）Cryptographic primitive based on lattice problems

研究代表者
桑門　秀典（Kuwakado, Hidenori）

関西大学・総合情報学部・教授

研究者番号：３０２８３９１４

交付決定額（研究期間全体）：（直接経費）　　3,800,000円

研究成果の概要（和文）：Ajtai-GGH関数とSWIFFT関数は，原像困難性と衝突困難性が格子問題の困難性に帰着されるという点が既存のハッシュ関数よりも優れている．本研究では，まず，これらの関数の帰着技法やSWIFFT関数で利用されている高速化の工夫の調査を行った．その結果に基づき，Ajtai-GGH関数をメッセージの前処理として利用するハッシュ関数の構成法を考案し，安全性の理論的検証と計算機実験による性能評価を行った．格子問題と関連するnearest-neighbor問題を任意の有限体上で解くアルゴリズムを開発した．

研究成果の概要（英文）：The Ajtai-GGH function and the SWIFFT function are better than usual hash functions because the preimage resistance and the collision resistance can be reduced to the difficulty of lattice problems. We first investigated the reduction technique used in proofs of these functions and the fast implementation technique of the SWIFFT function. We next proposed a Merkle-Damgaard hash function with a message preprocessing that is based on the Ajtai-GGH function. We studied its collision resistance and the indifferentiability from a random oracle and performed computer experiments for measuring the performance.

研究分野：暗号理論

キーワード：暗号　格子　安全性　最短ベクトル問題　復号問題

１．研究開始当初の背景

安全なネットワークを実現する各種の暗号学的プロトコル(例: SSL)は，ユーザがそれを意識しなくてよいほど広く普及している．暗号学的プロトコルの安全性は，使用されている暗号プリミティブに依存しているので，安全な暗号プリミティブの研究開発は必要不可欠である．

公開鍵暗号やデジタル署名等の暗号プリミティブは，素因数分解，離散対数問題，格子問題等の計算量理論の観点からも深く研究された問題の困難性に基づいている．

一方，共通鍵暗号やハッシュ関数等の暗号プリミティブは，ヒューリスティックな攻撃に対する困難性に基づいている．さらに，安全性の議論で特定の攻撃に関する議論が占める割合が大きいだけでなく，セキュリティパラメータの変化に対する安全性の変化が議論できないので，暗号プリミティブをスケーラブルに設計することができないという問題がある．

本研究の目標は，計算量理論の分野で深く研究されている格子問題の困難性に基づく共通鍵暗号やハッシュ関数等の暗号プリミティブを開発することである．これが実現できれば，計算量理論の観点から深く研究されている問題の困難さに基づく暗号プロトコルが構成できるので，長期にわたり安全を保証できる暗号プロトコルを設計することができる．

２．研究の目的

格子問題の困難性に基づくハッシュ関数の先行研究として，Ajtai-GGH と SWIFFTX がある．本研究の目標の一つは，最新のハッシュ関数の構成技法と安全性解析手法を用いて，これらよりも実装性と安全性の点で優れているハッシュ関数を構成することである．

格子問題の困難性に基づくハッシュ関数が構成できれば，それを利用して，共通鍵暗号を構成することができる．安全なハッシュ関数と共通鍵暗号が実現できれば，それらを使って擬似乱数生成器を構成する手法は既に知られている．したがって，ハッシュ関数を構成することが本質的に重要である．

３．研究の方法

本研究では，まず，格子問題の困難性に基づくハッシュ関数の設計に取り組む．既存研究の構成法を変更すること，最近のプロセッサの先進的な命令を積極的に活用すること等により，設計したハッシュ関数が実用的な処理速度を達成する見込みがある．そして，設計したハッシュ関数の安全性が格子問題の困難性に帰着できることを証明する．

４．研究成果

（１）帰着技術の調査検討

帰着技術に関する最新の文献を調査し，その内容を精査した．従来の帰着手法と比較して，最近考案された確率分布を用いた帰着手法は，証明の見通しがよくなる．ただし，確率分布を用いた帰着手法を用いても，帰着効率の改善度は大きくない．

（２）格子問題を解くアルゴリズムの調査検討

格子問題，特に最短ベクトル問題を解くアルゴリズムは，古くから研究されているが，暗号技術との関連により，最近特に注目されている．格子問題を解く有力なアルゴリズムである PKZ とその関連アルゴリズムを調査した．これらのアルゴリズムは，理論的な改良によりもヒューリスティックな改良の積み重ねにより，効率が改善されている．

（３）格子問題に基づくハッシュ関数の調査検討

格子問題の基づく方式は，一般に基底のサイズが大きく，それが実装時の問題になりやすい．これを解決するために，基底に代数構造の一つであるイデアルを導入することにより，パラメータのサイズを削減することが行われる．これにより，サイズは大幅に小さくできるが，他のハッシュ関数と比較すると，まだ大きい．

イデアル格子問題の困難さに基づくハッシュ関数 Ajtai-GGH と SWIFFT は，その原像困難性と衝突困難性が，格子問題の困難性に帰着する証明が知られている暗号プリミティブである．しかし，これらは線形関数なので，そのままでは暗号学的な用途には適さない．Ajtai-GGH 関数が暗号方式に利用されたことはなく，SWIFFT は，それと非線形関数を組み合わせた圧縮関数が暗号学的なハッシュ関数 SWIFFTX に利用されている．

既存方式であるハッシュ関数 SWIFFTX のアルゴリズムを精査し，安全性の根拠となっている格子の基底ベクトルを導出した．イデアルの典型的な実現方法である巡回基底とランダムな基底を識別する既存アルゴリズムを調査し，SWIFFTX の基底への適用を検討したが，複数の巡回基底を使用する SWIFFTX には適用が難しいことが判明した．

SWIFFTX が利用している高速化手法の効果について理論的に考察し，さらに計算機実験によりそれを確認した．SWIFFT は，高速フーリエ変換(FFT)と skew-circulant 行列を利用して，Ajtai-GGH ハッシュ関数を高速化した関数である．その高速化の度合いを理論的に見積もり，さらに計算機実験で調べ，8～15 倍の計算速度の向上を確認した．この向上の度合いは，信号処理の分野で FFT を利用するときに期待される高速化と比較すると低いが，実用的な効果があることが判明した．

また，SWIFFT は，理論的には，パラメータを変えることで，対応する格子問題の困難性を向上することができる．しかしながら，前述の高速化手法を利用するためには，パラメ

ータの選択に関する制約が厳しく，格子問題の困難性を柔軟には変更できないことが判明した．つまり，SWIFFT は，非常に上手にパラメータを選び，高速化と安全性を両立していることが確認できた．

高速化手法の効果について理論的に考察と計算機実験に基づき，これらの関数を既存の暗号学的ハッシュ関数のメッセージの前処理として利用する新しい構成法（後述）を考案し，安全性検証と計算機実験による性能評価をおこなった．これらの関数を暗号学的ハッシュ関数の圧縮関数ではなく，メッセージの前処理に利用した点が既存の研究とは異なる点である．これらの関数を前処理に用いた場合，新構成法の衝突困難性は，既存の衝突困難性またはこれらの関数の衝突困難性に帰着できることが判明した．計算機実験による評価では，SIMD 命令を有する CPU や援用 GPU を有する CPU の場合，Ajtai-GGH 関数は SIMD 命令や GPU を活用すると効率良く計算できるため，処理速度の改善が可能である．

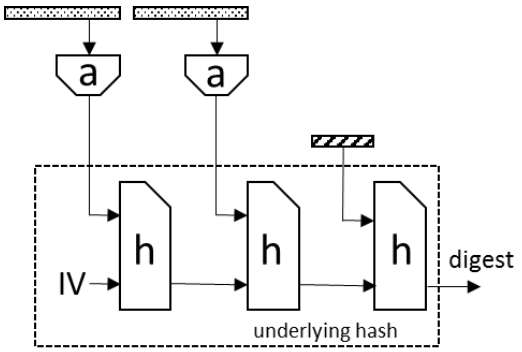

図 1: 前処理付きハッシュ関数

新しく提案した，メッセージの前処理付きのハッシュ関数の詳細を述べる．この構成法は任意の Merkle-Damgaard 型ハッシュ関数に適用できるが，SHA-512 を例として具体的なパラメータをあげて説明する．図 1 の点線で囲んだ部分が SHA-512 である．

prefix-free 符号化したメッセージを 4 [kiB]毎のブロックに分割する（図 1 の網掛けの部分）．それらを前処理関数 a に入力し，先頭ビットが常に 0 の 128 [B]の出力を得る．その出力を SHA-512 の圧縮関数 h へ順次入力する．SHA-512 にとっての入力長は圧縮関数の入力長の倍数となるため，最後のブロック（図 1 の斜線部）は先頭ビットが常に 1 であり，最後の 128 [b]が SHA-512 への入力ビット数（図 1 の場合，2048 [b]）である．

図 1 の場合，計算時間は，2 回の前処理関数 a と 3 回の圧縮関数 h の計算時間である．前処理関数がない通常の SHA-512 を使用した場合，65 回の圧縮関数 h の計算が必要である．したがって，1 回の前処理関数 a の計算時間が 31 回の圧縮関数 h の計算時間より短ければ，前処理付き SHA-512 の方が元の SHA-512 よりも計算時間が短くなる．

前処理関数 a について述べる．前処理関数 a には，Ajtai-GGH 関数を用いる．Ajtai-GGH 関数を用いる利点は，衝突困難性と原像困難性が格子問題の困難さに帰着できることが知られており，その格子問題に対する計算機実験も盛んに行われているので，安全性の高いパラメータが選択しやすい．SWIFFT と比べて，パラメータの自由度が高いため，既存のハッシュ関数の圧縮関数とパラメータの値を容易に合わせることができる．Ajtai-GGH 関数は以下のように定義される線形関数である．

$$y = A\,x \bmod q$$

ここで，q は 1024，A は 342 個の 96×96 の skew-circulant 行列を連接した 96×32832 の行列，x は長さ 32832 (=342×96) のバイナリベクトル，y は要素数 96，各要素が 10 [b] のベクトルとなる．上述したように，前処理関数 a への入力は 4 [kiB] (=32768 [b]) なので，長さ 32832 になるように，入力時に 0 でパディングをする．前処理関数 a への出力は 960 [b]なので，圧縮関数の入力長 1024 [b] になるように 64 個の 0 を先頭につける．

前処理関数 a の実装に必要な ROM 量は行列 A の ROM 量が支配的である．A は skew-circulant なので，約 40 [kiB] (=96×342×10 [b]) の ROM 量が必要である．SHA-512 の ROM 量が 640 [B]なので，前処理関数 a の ROM 量がかなり大きいことがわかる．これは，4.3 節の冒頭で述べたように，格子問題を利用する場合の問題点である．一方，前処理関数 a の計算時間は，Ajtai-GGH 関数の計算時間が支配的である．Ajtai-GGH 関数の計算は，行列とバイナリベクトルとの積なので，加算のみで実現できる．もし理想的な状態，つまり，完全に並列動作可能な加算器が十分にある場合，14 回の加算時間で Ajtai-GGH 関数を計算できる．これは，1 回の圧縮関数 h の計算時間より大幅に短い．しかしながら，CPU の SIMD 命令や援用 GPU を利用しても，この理想的な状態にはならないので，前処理関数 a の並列計算可能性による計算時間の短縮を実現できなかった．この点は今後の課題としたい．

前処理付きハッシュ関数の衝突困難性とランダムオラクルとの強識別不可能性について述べる．前処理付きハッシュ関数の衝突困難性は，前処理関数 a または圧縮関数 h の衝突困難性に帰着できる．前処理付きハッシュ関数の出力 (digest) が同じになる二つのメッセージ m1，m2 があったとする．前処理関数 a と圧縮関数 h の計算は確定的であるため，図 1 から，異なる入力に対して h の出力が同じになる事象，または異なる入力に対して a の出力が同じになる事象のいずれか (あるいは両方) が生じていることがわかる．したがって，前処理付きハッシュ関数の衝突困難性は，前処理関数 a の衝突困難性と圧縮関数 h の衝突困難性のうち，容易な方に帰着されるので，前処理関数 a の衝突困難性が高くなるようにパラメータを選べば，前処理付

きハッシュ関数の衝突困難性が元のハッシュ関数の衝突困難性を下回ることはない．

ランダムオラクルとの強識別困難性は，Cron らがハッシュ関数の安全性の一つとして提案した．しかし，Ristenpart らは，ハッシュ関数が，この安全性を満たしていたとしても，当初期待されていたような幅広いクラスの暗号プロトコルにそれを用いることはできず，single stage と呼ばれるクラスに属する暗号プロトコルであればそれを用いることができることを示した．その後，どのような暗号プロトコルであれば，ランダムオラクルとの強識別困難性を満たすハッシュ関数を用いることができるか，を明らかにする研究が進められているおり，実用的な暗号プロトコルがその中に含まれていることが判明している．よって，ハッシュ関数のランダムオラクルと強識別困難性を示すことは重要である．

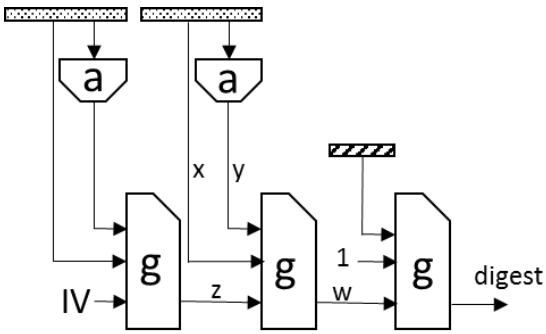

図２：前処理付きハッシュ関数の変形版 G

図１ではなく，図２のような前処理付きハッシュ関数 G を考える．ここで，圧縮関数 g の出力 w は以下のように定義される．

$$w = \begin{cases} h(y) & \text{if } (y = a(x) \wedge y = 0^*) \vee y = 1^* \\ \perp & \text{otherwise} \end{cases}$$

ここで，0*とは 0 で始まるビット列，1*は 1 で始まるビット列を意味する．図２に従って計算をする限り，圧縮関数 g は圧縮関数 h と等価である．変形版 G に対して，ランダムオラクルとの識別困難性を考える（図３）．

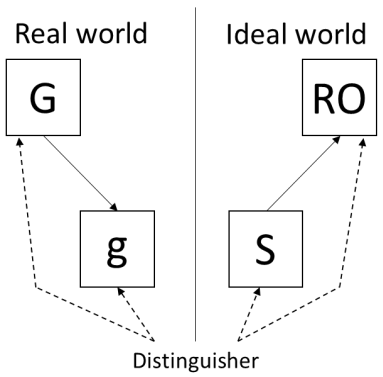

図３：攻撃モデル

図３の左側が前処理付きハッシュ関数の変形版 G と圧縮関数 g である．g で用いられる関数 h は入出力が固定長のランダム関数である．図３の右側がランダムオラクルと圧縮関数 g をシミュレートするシミュレータ S である．このとき，S のアルゴリズムを適切に定めれば，Distinguisher（識別者）が左側と右側を区別することが難しいことを示すことがハッシュ関数Gがランダムオラクルと強識別困難であることの証明となる．

この証明はハッシュ関数 G に対するものであるが，圧縮関数 g に対して，最初の条件を満たす入力以外しないことを前提にすれば，図１のように実装することができる．

（４）格子問題と関連する問題を利用した暗号プリミティブ

格子問題と関連する nearest-neighbor 問題を有限体上で解くアルゴリズムを研究した．二元有限体上で解くアルゴリズムのみが知られていたが，本研究により，素体上で解くアルゴリズムが明らかになった．このアルゴリズムは，nearest-neighbor 問題に基づく暗号方式の安全性評価に利用できるので，セキュリティパラメータの選定に役立つ．

５．主な発表論文等

（研究代表者、研究分担者及び連携研究者には下線）

〔雑誌論文〕（計　３件）

① Hidenori Kuwakado, Message Preprocessing for MD Hash Functions, IEICE Technical Report, Vol.115, 2016, pp.89-93, 査読無

② Shoichi Hirose, A Note on the May-Ozerov Algorithm for the Nearest Neighbor Problem over Any Finite Field, Proc. of 2016 Cryptography and Information Security, 1D2-5, 2016 pp.1-7, 査読無

③ Hidenori Kuwakado, Arithmetic in a Prime Field of SWIFFT/SWIFFTX, IEICE Technical Report, Vol. 114, 2015, pp.149-152, 査読無

６．研究組織

(1)研究代表者
桑門　秀典（KUWAKADO, Hidenori）
関西大学・総合情報学部・教授
研究者番号：30283914

(2)研究分担者
廣瀬　勝一（HIROSE, Shoichi）
福井大学・大学院工学研究科・教授
研究者番号： 20228836