量子計算機にも安全な次世代暗号の研究

大阪大学 SEEDS プログラム 2020 実感科学研究受講生　河本 聖己（19030）
工学研究科 電気電子情報工学専攻 サイバーセキュリティ工学領域　宮地研究室

《 要 旨 》

近年，量子計算機の開発が加速しており，その完成により現在広く社会で用いられている RSA 暗号などの安全性の危殆化が指摘されている．そのため，量子計算機でも解読困難な暗号化技術（耐量子暗号）の研究が活発におこなわれている．その一例に，ディオファントス問題を用いた鍵交換技術が挙げられる．しかしながら，これに関する研究事例はまだ少なく，安全性や計算性能について未知の部分があった．本研究では，RSA 暗号の安全性や性能を検証した上で，ディオファントス問題の先行研究をもとに改良を加え，Magma と Python で実装し，計算性能を検証した．

１．研究の動機・目的

現在，日本では Society4.0（情報社会）に続く新たな社会として Society5.0 が提唱されている． Society 5.0 で実現する社会では，IoT（Internet of Things）で全ての人とモノがつながり，今までにない新たな価値を生み出すことを目的としている．また，人工知能（AI）やビッグデータの活用により，必要な情報が必要な時に提供されるようになり，日本や世界が抱える問題解決に寄与することが期待されている．これらを実現するにあたり，情報セキュリティの技術は重要な役割を果たす．

現在使われている「公開鍵暗号」では，離散対数問題（代表例に ElGamal 暗号）や整数の因数分解問題（代表例に RSA 暗号）を用いたものが挙げられる．しかしながら，これらは「量子計算機」の完成により今後 10 年程度で解読される可能性が研究者の間で指摘されており，量子計算機でも解読困難な暗号化技術（耐量子暗号）の研究が活発に行われている．本研究では， 因数分解ベース方式の RSA 暗号のアルゴリズムや特殊な状況下における解読可能性・攻撃について実装をもって確認した．その上で NP 困難問題である量子計算機に対しても安全な次世代暗号システムとして，ディオファントス問題を用いた暗号プロトコルの実装・計算性能の検証を Magma と Python の両方で行った．

２．研究の方法と過程

１．RSA 暗号の基盤技術

RSA 暗号は，1977 年に発明された公開鍵暗号のひとつであり，Ronald Linn Rivest, Adi Shamir, Leonard Max Adleman の３人の研究者によって提案された。この暗号方式は，世界で広く利用されており，大きな整数の素因数分解の困難さを安全性の根拠とする。

RSA 暗号では，以下のようなプロトコルを用いて暗号化・復号を行っている．

【鍵生成】

ユーザーは，素数 p,q（$|p|$と$|q|$はほぼ同じ）を生成し，$n=pq$，$\lambda(n)=LCM(p-1,q-1)$を計算する．
適当な，$e \in Z\_\lambda\,(n)(GCD(e,\lambda(n))=1)$ を定め，
$d \equiv e^{-1}\left(\mathrm{mod}\lambda(n)\right)$を計算し，以下のプロトコルで，暗号化・復号を行う．

《秘密鍵》d　　《公開鍵》e,n

【暗号化】$c \equiv m^e(\mathrm{mod}n)$　【復号】$m \equiv c^d(\mathrm{mod}n)$

ここで，m は暗号化の対象となるデジタル化された情報（平文），c は暗号された情報（暗号文）であり，同時にその２進表現としての整数値を意味する．このとき，$0<m<n$ とする．RSA 暗号の公開鍵の鍵サイズは，2020 年では 2048 ビットが推奨されている．これは，素因数分解のアルゴリズムの進歩と計算機の能力向上に依存するため，将来的にはさらに長い鍵長に引き上げられる可能性が高い．

RSA 暗号に限らず，多くの暗号では，大きな素数を生成する必要がある．今回の実感科学研究では，コンピュータがランダムに生成した整数が，素数かどうか判定する方法を実装して確認した．素数判定にあたっては，確実に素数かどうか判定する方法（例：試し割法，AKS 素数判定法）とそうでない方法（例：Fermat テスト，Miller-Rabin テスト，楕円曲線素数判定法）に分けられるが，一般的には後者の確率的素数判定法の方が高速である．ただし，確率的素数判定法の Fermat テストでは，無限に存在する絶対着素数（カーマイケル数）を素数として判定してしまうため，実際には Miller-Rabin テストが利用されることが多い．

２．RSA 暗号への攻撃

特殊な状況下では，RSA 暗号に対する有効な攻撃が存在する．RSA 暗号の安全性を理解する目的で，攻撃方法について習得した．

特殊な状況下では，

１．公開鍵の一つ n を$(p-1)$法による整数の因数分解で秘密鍵 d を入手可能

２．秘密鍵 d が十分小さい場合は，連分数を応用した方法で秘密鍵を公開鍵から入手可能

３．複数のユーザーが共通の n を利用している場合，秘密鍵を用いずに，暗号文と公開鍵から平文を入手可能

である．

これら３つの攻撃方法を実装し，実際に模擬攻撃を行った場合，３番目の方法が最も高速だった．ただし，これらの攻撃が成功するのは特殊なパラメータを用いていたからである．以下にその実験結果を示す．

【パラメータ】

公開鍵 (e,n) の鍵長：2047 ビット

秘密鍵 d の鍵長：2047 ビット

（連分数のみ，200 ビット）

暗号文 c の大きさ：2047 ビット

	$(p-1)$法	連分数	共通法
計算時間	60.999	0.126	0.059

計算時間(second)：40 回計算を行った平均値

使用した計算機環境（NIIS-C6）：

Processor 3.8GHz 8-Core Intel Core i7 / Memory 16 GB 2667 MHz DDR4 / Program Language Python 3.8.3 / Software Visual Studio Code 1.54.2 /

３．ディオファントス問題を用いた耐量子暗号

前述の RSA 暗号は，整数の因数分解問題を利用した暗号アルゴリズムであるが，これは量子計算機によって解読される可能性が 1994 年に発表されたショアの量子因数分解のアルゴリズムによって指摘されている．根本的な解決策として，量子計算機でも解くことが困難な数学的な計算問題を利用した耐量子暗号を用いることが挙げられる．本研究ではディオファントス問題をベースとした耐量子暗号の研究に取り組んだ．

【ディオファントス問題の定義】

整数が係数の多変数方程式

$$\sum a_{e_1 e_2 \dots e_m} x_1^{e_1} x_2^{e_2} \dots x_m^{e_m} = 0 \ \left(a_{e_1 e_2 \dots e_m} \in \mathbb{Z}\right)$$

の整数解を見つける問題をディオファントス問題という．特殊な場合を除いて，一般的にすべての方程式について整数の範囲での一般解法（アルゴリズム）が存在しないことが証明されている．２変数の一般解法も未知であり，更に，有理数の範囲で一般解法が存在するかどうかも未知である．２変数２次方程式 $ax^2 + by + c = 0$ の整数解の存在判定問題は，NP 完全問題であることが証明されている．

本研究では，多項式写像を用いたアルゴリズム[1]を参考に実装を行った．

４．多項式写像を用いた鍵交換プロトコル

q： 有限体の元の個数

n： 変数の個数

m：Bob の多項式写像の次数

d：Alice の多項式写像の次数

(1) Alice は，多変数方程式 $f(\underline{x}) = 0$ を Bob に送信し，Bob は解 $\underline{\sigma} \in \mathbb{F}_q^n$ を秘密扱いとする．（ $f(\underline{x})$ とは，$f(x_1, x_2, \dots, x_n)$ を略記したものである．）

(a) 解 $\underline{\sigma}(\in \mathbb{F}_q^n)$ をランダムに選択し，$\underline{f}(\underline{\sigma}) = 0$ を満たす $\underline{f}(\underline{x})$ を生成する．

(b) 解 $\underline{\sigma}$ は秘密とする．

(c) $\underline{f}(\underline{x})$ を公開情報として，Bob に送信する．

(2) Bob は多項式写像 $\underline{g}(\underline{x})$ と $\underline{c}(\underline{x})$ を Alice に送る

(a) 全単射アフィン写像 $\underline{g}(\underline{x}) = \left(g_1(\underline{x}), \dots, g_n(\underline{x})\right)$ をランダムに生成する．

(b) 単射となる多項式写像 $\underline{\psi}(\underline{x}) = \left(\psi_1(\underline{x}), \dots, \psi_n(\underline{x})\right) \in \mathbb{F}_q[\underline{x}]^n$ を $\deg \psi_j = m (1 \le j \le n)$ を満たすように生成する．

(c) 合成関数 $\underline{\psi}\left(\underline{g}(\underline{x})\right)$ を計算する．

(d) 多項式写像 $\underline{r}(\underline{x}) = \left(r_1(\underline{x}), \dots, r_n(\underline{x})\right) \in \mathbb{F}_q[\underline{x}]^n$ を $\deg r_i = m - d$ となるようにランダムに生成する．

(e) 多項式写像 $\underline{c}(\underline{x}) = \underline{\psi}\left(\underline{g}(\underline{x})\right) + \underline{f}(\underline{x})\underline{r}(\underline{x})$ を計算する．（$\underline{f}(\underline{x})\underline{r}(\underline{x})$ を加算することにより，$\underline{\psi}\left(\underline{g}(\underline{x})\right)$ を直接求められないようにしている）

(f) 多項式写像 $\underline{g}(\underline{x})$ と $\underline{c}(\underline{x})$ を Alice に送る．

(3) Alice が共通鍵 $\underline{s} \in \mathbb{F}_q^n$ を生成し，$\underline{s}$ を暗号化した $\underline{u} \in \mathbb{F}_q^n$ を Bob に送る．

(a) $\underline{g}(\underline{\sigma}) = \underline{s}$ を計算し共通鍵とする．

(b) $\underline{c}(\underline{\sigma}) = \underline{u}$ を計算して，これを Bob に送る．

(4) Bob が共通鍵 $\underline{s}$ を計算する．

$\underline{f}(\underline{\sigma}) = 0$ より，$\underline{c}(\underline{\sigma}) = \underline{\psi}\left(\underline{g}(\underline{\sigma})\right) = \underline{u}$ が成り立つ．

Bob は，$\underline{\psi}^{-1}(\underline{u}) = \underline{g}(\underline{\sigma}) = \underline{s}$ によって，共通鍵 $\underline{s}$ を入手する．

【参考図】鍵交換の流れ

Alice	Communication channel	Bob
$\underline{f}(\underline{x}) \xleftarrow{r} \Lambda_{n,d}$ $\underline{f}(\underline{\sigma}) = 0$ $(\underline{\sigma} \in \mathbb{F}_p^n)$	$\underline{f}(\underline{x})$ $\rightarrow$	
	$\underline{g}(\underline{x}), \underline{c}(\underline{x})$ $\leftarrow$	$\underline{\psi}(\underline{x}) \xleftarrow{r} (\Lambda_{n,m}^n)^*$ $\underline{g}(\underline{x}) \xleftarrow{r} \Lambda_{n,1}^n$ $\underline{r}(\underline{x}) \xleftarrow{r} \Lambda_{n,m-d}^n$ $\underline{c}(\underline{x}) = \underline{\psi}(\underline{g}(\underline{x})) + \underline{f}(\underline{x})\underline{r}(\underline{x})$
$\underline{g}(\underline{\sigma}) = \underline{s}$ $\underline{c}(\underline{\sigma}) = \underline{u}$	$\underline{u}$ $\rightarrow$	$\underline{s} = \underline{\psi}^{-1}(\underline{u})$

注釈： $\Lambda_{n,d} := \{f \in \mathbb{R}[x_1, \dots, x_n] | \deg f = d\}$

$\left(\Lambda_{n,d}^n\right)^* := \{\psi \in \Lambda_{n,d} | \psi \text{ is injective}\}$

5．プロトコルの検証

第 4 節で説明した多項式写像を用いた鍵交換プロトコルを Magma 及び Python で実装し，プロトコル実行に要した全体の時間を計測し，計算性能の検証を行った．その結果を次章で示す．なお，実装したプログラムは紙幅の制限により省略した．

3．研究の結果と考察

この暗号プロトコルにおける安全性が担保されるためには，以下の条件を満たす必要があるとされている．

条件 1

$N(n,d) := n\binom{n+d}{d}$ を定義する．

kをセキュリティパラメータとしたとき，

$q^{N(n,m-d)} > 2^k$ を満たす必要がある．

条件 2

$$\begin{cases} f(x_1, \dots, x_n) = 0 \\ c_1(x_1, \dots, x_n) = u_1 \\ \vdots \\ c_n(x_1, \dots, x_n) = u_n \end{cases}$$

上記の連立方程式を現実的な時間内に解くことが困難である必要がある．

【パラメータ】

$k = 128$ として，条件を満たすように，以下のパラメータを定めた．n は変数の個数を表し，多項式の複雑さを決める値である．このため，n の値を変化させて実験を行った．

$(q, n, m, d) = (17, n, 3, 2)$

n	Magma	Python 3.7
5	0.003	0.025
6	0.006	0.034
7	0.014	0.068
8	0.023	0.225
9	0.049	0.913
10	0.098	4.662
15	1.722	8.112
20	19.178	N/A

25	131.134	N/A

計算時間(second)：40 回計算を行った平均値

Magma を実行するに当たっては，宮地研究室の計算サーバーで実行した.

使用した計算機環境（MYJ-Crypto）：

Processor 2. 00GHz Intel Xeon E7-4830 v4 / Memory 3TB / OS Ubuntu 16. 04. 7 LTS / Software Magma V2.24-1, Visual Studio Code 1.54.2

実験を行った結果, 前章第 4 項における(4)の逆関数計算に膨大な時間を要した. そのため，実装時にプログラム上で改良を行い, 逆関数を直接計算せず, $\underline{u}$ と $\underline{\psi}$ を用いて，$\underline{\psi}^{-1}(\underline{u})$ を計算できるように工夫し，計算速度の向上を図った. しかしながら，(2)(c)の合成関数の計算は，n の値が大きくなることで計算量が多くなることは避けられず結果として，$n \geq 20$ のパラメータでは相当な計算時間を要することとなった. Python においては，$n \geq 20$ の場合，計算リソースを使い切り，計算することが出来なかった. また，$n = 10$ であれば単項式は 8,008 個，$n = 11$ であれば 12,376 個と爆発的に増加することが分かった.

セキュリティ要件を満たすためには，$n \geq 32$ である必要があるが, 実際にこれを満たすパラメータで計算することは高性能な計算機を以てしても Magma, Python ともに困難であった.

4．得られた結果

前述のように, 本プロトコルで行った計算では，合成関数の計算がボトルネックとなり，一般的な計算機で実行するには，非現実的な計算量を要する. このため，現段階でのプロトコルでは耐量子暗号として用いることは現実的に難しいと言える.

現在広く普及している RSA 暗号の暗号化･復号に要する時間は，私たちがウェブサイトを閲覧していても暗号化･復号による遅延を感じない程度に高速で, 中国式余剰定理など高速に計算する方法が確立されている.

ディオファントス問題ベースの暗号が，ポスト量子暗号技術として使えるようになるためには，高速に計算するための何らかの方法やプロトコルの大幅な改良が必要だと言える.

5．今後の発展と展開

今回の研究では，ディオファントス問題を用いた暗号プロトコルの検証の結果，セキュリティパラメータを満たして計算することが難しいことがわかった. 今後の課題として，プログラムの改良（特に，合成関数や逆関数の計算）や, その他の数学的性質を取り入れた新たなプロトコル改良に取り組みたい.

6．参考文献

[1] Koichiro Akiyama, Shuhei Nakamura, Masaru Ito & Noriko Hirata-Kohno (2019), *A key exchange protcol relying on polynomial maps*. International Journal of mathematics for Industry

[2] Shinya Okumura (2015), *A public key cryptosystem based on Diophantine equations of degree increasing types,* Pacific Journal of Mathematics for Industry

[3] Attila Berczes, Lajos Hajdu, Noriko Hirata-Kohno, Tunde Kovacs & Attila petho (2014), *A key exchange protocol based on Diophantine equations and S-integers*, JSIAM Letters

[4] Noriko Hirata-Kohno & Attila Petho (2013), *On a key exchange protocol based on Diophantine equations*, Infocommunications Journal

[5] 宮地充子，菊池浩明（2003），「ITtext 情報セキュリティ」，オーム社