RSA 暗号について　　加本

＜１＞ RSA における「公開鍵」と「秘密鍵」の関係

1. まず前提

RSA では、メッセージのやり取りを安全にするために、

暗号化には公開鍵を使う　復号には秘密鍵を使う　という「役割分担」があります。

ただし、この 2 つの鍵は全くの別物ではなく、数学的に対になっています。

2. 例で考える（概念）

たとえば、ある人（B さん）が RSA 鍵を作ると、次のような鍵のペアができます：

公開鍵（B の）　例：（n = 3233, e = 17）　→ この「公開鍵」は誰でも使ってよい（ネットに公開しても OK）

秘密鍵（B の）　例：（n = 3233, d = 2753）　→ この「秘密鍵」は B さんだけが知っている（絶対に他人には見せない）

※ここで n は 2 つの素数の積、e は公開用指数、d は秘密の指数です。これらの具体的な計算は省略します。

3. 公開鍵と秘密鍵の関係性のポイント

公開鍵（n, e）で暗号化したものは、秘密鍵（n, d）でしか復号できません。

秘密鍵（n, d）で署名したものは、公開鍵（n, e）で検証できます。

つまり：公開鍵と秘密鍵は「逆向きに作用する、ぴったり対応したペア」なのです。

4. 通信の流れ（暗号化の例）

登場人物：A さん（送り手）、B さん（受け手）

手順：

B さんは、自分用の RSA 公開鍵と秘密鍵のペアを作る。

B さんは、公開鍵だけを A さんに送る。

A さんは、その公開鍵を使ってメッセージを暗号化する。

A さんは、暗号化されたメッセージを B さんに送る。

B さんは、自分の秘密鍵を使ってメッセージを復号する。

➡ この仕組みのおかげで、第三者は内容を読めません。

◆ なぜ安全なのか？

公開鍵から秘密鍵を「逆算」するのは非常に困難（現在の計算機では数千年かかる）。

公開鍵を使って暗号化しても、秘密鍵なしには元に戻せない。

＜２＞　RSA 暗号の具体例：公開鍵・秘密鍵による暗号化と復号

1. 鍵の設定

以下は RSA の簡易な例です。

公開鍵（誰でも使ってよい）: (n = 3233, e = 17)

秘密鍵（本人だけが知る）: (n = 3233, d = 2753)

2. 平文の設定

ここでは、文字 'A' の ASCII コードである 65 を平文として使います。

3. 暗号化

平文 65 を公開鍵（n = 3233, e = 17）を使って暗号化します。

計算式: 暗号文 = 65^17 mod 3233

結果: 暗号文 = 2790

4. 復号

暗号文 2790 を秘密鍵（n = 3233, d = 2753）を使って復号します。

計算式: 復号結果 = 2790^2753 mod 3233

結果: 復号結果 = 65（元の 'A' に戻る）

5. まとめ

このように、RSA では公開鍵で暗号化したメッセージは対応する秘密鍵でしか復号できません。逆に、秘密鍵で署名したものは公開鍵で検証できます。このような性質により、安全な通信や電子署名が実現されています。

＜３＞　RSA 暗号における鍵と安全性の関係：例（n=3233, e=17, d=2753）

1. はじめに

RSA 暗号は、大きな数を素因数分解することが困難であるという性質に基づいています。公開鍵（n, e）と秘密鍵（n, d）は数学的に対応しており、一方で暗号化したものは、他方でしか復号できません。

2. この例の鍵情報

公開鍵: (n = 3233, e = 17)

秘密鍵: (n = 3233, d = 2753)

3. n の正体：素因数分解

n = 3233 は以下の素数の積でできています：

p = 61, q = 53

したがって、n = p × q = 61 × 53 = 3233

4. ϕ(n) の計算と鍵の関係

RSA では次にオイラー関数 ϕ(n) を計算します。

ϕ(n) = (p - 1)(q - 1) = 60 × 52 = 3120

この ϕ(n) に対して、e = 17 は互いに素な数として選ばれました。

次に、以下を満たす d を求めます：

d × e ≡ 1 mod ϕ(n) → d × 17 ≡ 1 mod 3120

この条件を満たす d = 2753 です。

5. なぜ安全なのか

このように、n の素因数（p, q）を知らない限り ϕ(n) を計算することは困難です。したがって、秘密鍵 d を導くのは非常に難しいという点が RSA の安全性の根本にあります。

6. まとめ

・公開鍵と秘密鍵は、n を共有しつつ異なる指数（e, d）を持つ。

・n は 2 つの素数の積として成り立ち、その分解が困難なため、第三者が秘密鍵を割り出すのは困難。

・今回の例では、17 と 2753 は n を構成する素数とは関係がなくても、鍵として機能します。

・ただし、これらの値は ϕ(n) との関係で正しく構築されたペアでなければなりません。

n = 55 におけるオイラー関数 φ(n) の計算と RSA での活用

1. 素因数分解とオイラー関数 φ(n)

n = 55 を素因数分解すると、n = 5 × 11 となり、どちらも素数です。
このとき、オイラー関数 φ(n) は次のように計算されます：
φ(n) = φ(5 × 11) = φ(5) × φ(11) = (5 - 1) × (11 - 1) = 4 × 10 = 40
1 以上 55 未満の整数のうち、55 と互いに素な整数は 40 個あることを意味します。

2. 互いに素な整数の確認

55 と互いに素な 1〜54 の整数は以下の通り：
1, 2, 3, 4, 6, 7, 8, 9, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27, 28, 29, 31, 32, 33, 34, 36, 37, 38, 39, 41, 42, 43, 46, 47, 48, 51, 52, 54 （5 の倍数や 11 の倍数を除外）

3. RSA における φ(n) の利用

RSA 暗号では、この φ(55) = 40 を用いて以下の手順を踏みます：
1. 公開鍵 e を選ぶ（φ(n) = 40 と互いに素な整数）
   例：e = 3, 7, 9, 13 など（gcd(e, 40) = 1）
2. 秘密鍵 d を求める（e × d ≡ 1 mod 40 となる d を探す）
   例：e = 3 のとき、d = 27（3×27 = 81 ≡ 1 mod 40）

4. まとめ

n = 55 のとき：- 素因数：5, 11 - φ(55) = 40
- 1〜54 の中で 55 と互いに素な整数：40 個
- 使用可能な公開鍵 e：3, 7, 9, 13 など
- 秘密鍵 d：e の逆元 mod 40（例：e = 3 → d = 27）
このように、φ(n) は RSA 暗号において、公開鍵と秘密鍵の関係を数学的に決定する中心的な役割を果たしています。

🔍 **補足解説**

- **互いに素**とは：2 つの整数の最大公約数が 1 の関係。
- **除外される整数**：
  - 2 の倍数（偶数）：2, 4, 6, 8, 10, ..., 38
  - 5 の倍数：5, 10, 15, 20, 25, 30, 35, 40

これを除いたものが上記 16 個になります。
この 16 という数は、**オイラー関数 φ(40) の値**でもあります。

オイラー関数 φ(n) の公式と例（n = 40）

1. オイラー関数 φ(n) とは？

φ(n) は、「1 から n の手前までの整数の中で、n と互いに素な数の個数」を表す関数です。「互いに素」とは、最大公約数が 1 の関係です（gcd(a, n) = 1）。例：φ(9) = 6 （1, 2, 4, 5, 7, 8 が該当）

2. φ(n) の公式（素因数を使った式）

n を素因数分解したときの公式：

$\varphi(n) = n \times (1 - 1/p_1) \times (1 - 1/p_2) \times \dots$

ここで $p_1, p_2, \dots$ は n の素因数です。

3. n = 40 の場合で計算してみよう

【ステップ①】40 を素因数分解： $40 = 2^3 \times 5$ → 素因数は 2 と 5

【ステップ②】公式に代入：

$\varphi(40) = 40 \times (1 - 1/2) \times (1 - 1/5) = 40 \times 1/2 \times 4/5 = 40 \times 4/10 = 16$

4. 結果：φ(40) = 16

つまり、1〜39 の中で 40 と互いに素な数は 16 個ある。

【確認】互いに素な数（例）：1, 3, 7, 9, 11, 13, 17, 19, 21, 23, 27, 29, 31, 33, 37, 39

5. 応用：RSA 暗号での活用

RSA 暗号では次のように φ(n) を使います：

1. 素数 p, q を選び、n = p × q を計算
2. φ(n) = (p - 1)(q - 1)
3. φ(n) と互いに素な整数 e を公開鍵として選ぶ
4. $e \times d \equiv 1 \bmod \varphi(n)$ を満たす d を秘密鍵として求める

→ φ(n) は公開鍵と秘密鍵の関係を決める重要な値

6. まとめ

φ(n)：n 以下で n と互いに素な数の個数

計算方法：素因数分解を使った公式

RSA 暗号：鍵の生成に φ(n) を使用