符号理論･暗号理論

- No.13 公開鍵暗号 -

渡辺　裕

Coding Theory / Cryptography

- No.13 Public Key Crypt System -

Hiroshi Watanabe

公開鍵暗号

- 公開鍵暗号の考案
  - ディフィー (Whitfield Diffie) と ヘルマン (Martin E. Hellman) (1976)
- RSA暗号
  - 素因数分解の困難さを利用
  - リベスト (Ron Rivest), シャミル (Adi Shamir), エイドルマン (Leonard Adleman) (1997)
- 共通鍵暗号の問題点
  - 鍵の受け渡し

Public Key Cryptography

- asymmetric-key cryptosystem published
  - Whitfield Diffie and Martin E. Hellman (1976)
- RSA Cryptography
  - Based on difficulty of factoring large prime numbers
  - Ron Rivest, Adi Shamir, Leonard Adleman (1997)
- Solve problem of common key crypt system
  - Security of key handout

暗号化システム

- 処理手順
  - 受信者は自分の公開鍵*P*を全世界に公開
  - 送信者は公開鍵*P*を使ってメッセージを暗号化し送信
  - 受信者は公開鍵*P*と対になる秘密鍵*S*を使って受信内容を復号し，送信者からのメッセージを読む

- 暗号強度
  - 暗号通信を不正に傍受しようとする者(傍受者)が，送信者が送信した暗号化されたメッセージを傍受したとする．傍受者は，公開鍵*P*は知っているが，秘密鍵*S*は分からない．*P*から*S*を割り出すことは計算時間的に極めて難しい．

Crypt System

- Procedure
  - Receiver exposes own public key *P* to the world
  - Sender encrypt a message by the public key P
  - Receiver decrypt a received data using private key *S*, which is a pair with public key *P*, and read the message

- Security strength
  - Let some intercept person can illegally receive cipher communication. The intercept person know public key *P*, but not private key *S*. It is difficult to know S from P in a practical time period.

暗号化アルゴリズム

- 鍵生成アルゴリズム
  - ユーザの公開鍵と秘密鍵を生成
  - 解読の困難さを制御するセキュリティ・パラメータと乱数を入力
  - ユーザごとに異なった乱数
- 暗号化アルゴリズム
  - メッセージと暗号文を送りたい相手(受信者)の公開鍵とを入力して暗号化アルゴリズムを実行
- 復号アルゴリズム
  - 受信者は復号アルゴリズムに自分の秘密鍵と暗号文を入力して，もとのメッセージを復元

Encryption Algorithm

- Key generation algorithm
  - Generate user's pair of public and private keys
  - Input security parameter to control difficulty of decryption
  - Different random number for each user
- Encryption algorithm
  - Input a message and public key of a receiver, to whom sender wishes to send a message
- Decryption algorithm
  - Receiver inputs own private key and received data, and recover a message

DH鍵共有 (1)

- ディフィー・ヘルマン鍵共有(Diffie-Hellman key exchange)
  - 事前の秘密の共有無しに，盗聴の可能性のある通信路を使って，暗号鍵の共有を可能にする暗号プロトコル
  - 通信を行いたい2者が各々公開鍵と秘密鍵 を用意し，公開鍵のみを公開
  - お互いに秘密の値から作成されるデータを相手に送信し，各自自分の秘密鍵と受信したデータから共通鍵を作成できる方法

- 暗号プロトコル
  - 公開されている数：素数p，$(Z/pZ)^*$の生成元g
  - 送受信者X, Yは秘密の鍵a, bを選択
  - Xは $A=g^a \pmod p$ を計算しYに送信
  - Yは $B=g^b \pmod p$ を計算しXに送信

DH Key Exchange (1)

- Diffie-Hellman key exchange
  - allows two parties that have no prior knowledge of each other to jointly establish a shared secret key over an insecure communications channel
  - Two parties prepare each private and public key, and open only public keys
  - Generate their common key by sending data generated by their private key
- Encryption protocol
  - Open number: prime number p, root of $(Z/pZ)^*$ g
  - Sender and receiver X, Y select secret key a, b
  - X calculates $A=g^a \pmod p$ and send it to Y
  - Y calculates $B=g^b \pmod p$ and send it to X

DH鍵共有 (2)

- 暗号プロトコル(続き)
  - Xは受信データBと秘密鍵aを用いて $K_A=B^a \pmod p$ を計算
  - Yは受信データAと秘密鍵bを用いて $K_B=A^b \pmod p$ を計算
  - $K_A=K_B= g^{ab} \pmod p$ となり共通の値となるから，これを共通鍵として以降の暗号化に使用

- 安全性
  - $A=g^a \pmod p$ と $B=g^b \pmod p$ を盗聴して入手しても $g^{ab} \pmod p$ を多項式時間で計算する方法は現在の所存在しない
  - 多項式時間：計算理論において多項式で表される計算時間．例：$O(n^2)$

DH Key Exchange (2)

- Encryption protocol (Contd.)
  - *X* calculates $K_A = B^a \pmod p$ by received data *B* and private key *a*
  - *Y* calculates $K_B = A^b \pmod p$ by received data *A* and private key *b*
  - $K_A = K_B = g^{ab} \pmod p$, thus, it can be used as a shared key from now on
- Security
  - Even if $A = g^a \pmod p$ and $B = g^b \pmod p$ can be obtained, there is no method to calculate $g^{ab} \pmod p$ in a polynomial-time
  - Polynomial time: calculation time is upper bounded by a polynomial expression in the size of the input for the algorithm. Example: $O(n^2)$

RSA暗号 (1)

- システムパラメータ
  - k:セキュリティ・パラメータ
  - p, q: 素数, k/2ビット
  - a: Z_n平文空間の元(メッセージ)
- 鍵生成
  - $n=pq$
  - $\Phi(n)=(p-1)(q-1)$：オイラー関数
  - e: $\Phi(n)$未満の正の整数で, $\Phi(n)$とお互いに素な数
  - $de=1 \pmod{\Phi(n)}$ を満たすdを求める
  - 公開鍵:(n, e)
  - 秘密:d

RSA Encryption (1)

- System parameter
  - k:Security parameter
  - p, q: prime numbers, k/2 bit
  - a: input message in space Z_n
- Key generation
  - $n=pq$
  - $\Phi(n)=(p-1)(q-1)$: Euler function
  - e: positive integer less than $\Phi(n)$, coprime to $\Phi(n)$
  - Obtain de which satisfies $de=1 \pmod{\Phi(n)}$
  - Public key :(n, e)
  - Private key:d

RSA暗号 (2)

- 暗号化
  - 暗号化する平文メッセージをaとする
  - 公開鍵:(n, e)
  - $b = a^e \pmod n$ を計算
  - 暗号化されたメッセージbが生成される
- 復号
  - 復号すべきメッセージはb
  - 秘密鍵:d
  - $a' = b^d \pmod n$ を計算
  - $a' = a$であることを確認

RSA Encryption (2)

- Encryption
  - Let a message for encryption be a
  - Public key:(n, e)
  - Calculate $b = a^e \pmod n$
  - Cipher b is generated
- Decryption
  - The message which should be decrypted is b
  - Private key :d
  - Calculate $a' = b^d \pmod n$
  - Confirm $a' = a$

RSA暗号 (3)

- 完全性の証明 ($a' = a$であることを確認)
  - $b = a^e \pmod n$ であるから
    $a' = b^d \pmod n = a^{ed} \pmod n$
  - $de=1 \pmod{\Phi(n)}$ であるから
    $a' = a^{(m\Phi(n)+1)} \pmod n$
  - ここで, nに対してaが素な関係にあるときには, オイラーの定理により, $a^{\Phi(n)} \equiv 1 \pmod n$ なる関係が成立するから
    $a' = a^m a^1 \pmod n = a \pmod n$

RSA Encryption (3)

- Proof of correctness (confirm $a' = a$)
  - $b = a^e \pmod n$, thus
    $a' = b^d \pmod n = a^{ed} \pmod n$
  - $de=1 \pmod{\Phi(n)}$, thus
    $a' = a^{(m\Phi(n)+1)} \pmod n$
  - Where, if n and a are in coprime relation, by Euler's theorem, $a^{\Phi(n)} \equiv 1 \pmod n$ holds, thus
    $a' = a^m a^1 \pmod n = a \pmod n$

RSA暗号 (4)

- 完全性の証明 (続き)
  - 次に, nに対してaが素な関係にないときには, 最大公約数pが存在し, $a=pi=aq+qj$ $(0<i<q,\ 0 \leqq j<p)$と書けるから

    $a' = 0 \pmod{p}$

    $a' = a = aq \pmod{q}$

  - 中国人の剰余定理, $p*xp=1+q*xq$ から

    $a' = 0*q*xq + aq*p*xp = (p*i-q*j)*p*xp$

    $= p*i*p*xp = p*i*(1+q*xq) = p*i = a \pmod{n}$

RSA Encryption (4)

- Proof of correctness (Contd.)
  - Nest, if n and a are not in a coprime relation, there is a GCD number p, thus we can write $a=pi=aq+qj$ $(0<i<q,\ 0 \leqq j<p)$

    $a' = 0 \pmod p$

    $a' = a = aq \pmod q$

  - From Chinese remainder theorem, $p*xp=1+q*xq$

    $a' = 0*q*xq + aq*p*xp = (p*i-q*j)*p*xp$

    $= p*i*p*xp = p*i*(1+q*xq) = p*i = a \pmod n$

オイラーの定理 (1)

- オイラーの定理
  - a を n と互いに素な整数とするときに,
    $a^{\Phi(n)} \equiv 1 \pmod{n}$
    が成り立つ
  - ここで, $\Phi(n)$はn未満のnに素な自然数の個数を表し, オイラー関数と呼ばれる

- オイラー関数の例
  - $n=6$のとき, $1, 2, 3, 4, 5, 6$のうち6と素な数は, 1と5の2個であるから, $\Phi(6)=2$
  - $n=7$のとき, $1, 2, 3, 4, 5, 6, 7$のうち7と素な数は, $1, 2, 3, 4, 5, 6$の6個であるから, $\Phi(7)=6$
  - $n=p$(素数)のとき, $\Phi(p)=p-1$

Euler's theorem (1)

- Euler's theorem
  - When a and n are coprime integers,

    $a^{\Phi(n)} \equiv 1 \pmod{n}$

    holds.
  - Where, $\Phi(n)$ is a number of coprime numbers to n, as well as less than n, it is called Euler's totient function
- Example of Euler's totient function
  - When $n=6$, coprime number to 6 among $1, 2, 3, 4,$ are $1$and $5,$ thus $\Phi(6)=2$
  - When $n=7$, coprime number to 7 among $1, 2, 3, 4, 5, 6, 7$ are $1, 2, 3, 4, 5, 6,$ thus, $\Phi(7)=6$
  - When $n=p$(prime number), $\Phi(p)=p-1$

オイラーの定理 (2)

- オイラーの定理(続き)
  - 二つの素数 $a=3$, $n=7$とすると, オイラー関数$\Phi(7)=6$
    $3^{\Phi(7)} \pmod 7 = 3^6 \pmod 7 = 729 \pmod 7 = 1$
    $(104*7=728)$
  - 二つの素数 $a=7$, $n=3$とすると, オイラー関数$\Phi(3)=2$
    $7^{\Phi(3)} \pmod 3 = 7^2 \pmod 3 = 49 \pmod 3 = 1$
    $(16*3=48)$

Euler's Theorem (2)

- Euler's theorem (Contd.)
  - Let two prime numbers be $a=3$, $n=7$, Euler's function is $\Phi(7)=6$

    $3^{\Phi(7)} \pmod 7 = 3^6 \pmod 7 = 729 \pmod 7 = 1$

    $(104*7=728)$

  - Let two prime numbers be $a=7$, $n=3$, Euler's function is $\Phi(3)=2$

    $7^{\Phi(3)} \pmod 3 = 7^2 \pmod 3 = 49 \pmod 3 = 1$

    $(16*3=48)$

オイラーの定理 (3)

- オイラーの定理
  - nが正の整数でaをnと互いに素な正の整数としたとき,
  - $a^{\Phi(n)}=1 \pmod n$

- オイラーの定理の証明
  - nと互いに素なn以下の正の整数の集合を $B=\{b_1, b_2, \dots, b_{\Phi(n)}\}$ とする
  - この集合の要素にそれぞれaを乗じた集合Aは, $A=\{ab_1, ab_2, \dots, ab_{\Phi(n)}\}$となる. aとnは互いに素であるから、集合A,Bは法をnとしたときに一致し, 当然その積も法nにおいて等しい
  - Bの要素の積をPとすれば, $P=Pa^{\Phi(n)} \pmod n$
  - nとpはお互いに素であるから, $a^{\Phi(n)}=1 \pmod n$

Euler's Theorem (3)

- Euler's therem
  - Let n be positive integer, and a and n are coptime numbers,
  - $a^{\Phi(n)}=1 \pmod n$
- Proof
  - Let a set of positive integers less than n, and coprime to eachother be $B=\{b_1, b_2, \dots , b_{\Phi(n)}\}$
  - A set multiplied a to the above is $A=\{ab_1, ab_2, \dots , ab_{\Phi(n)}\}$. Since a and n are coprime each other, elements of set A,B are equal under modulo n, thus, muliplied elements are equal.
  - Let multiplied elements of B *be* P, $P=Pa^{\Phi(n)} \pmod n$
  - n and p are coprime each other, $a^{\Phi(n)}=1 \pmod n$

RSA暗号の例 (1)

- RSA暗号の具体的な数値例
  - 素数p, qが $p=9010279$, $q=9623083$ の場合
  - $n=pq=9010279*9623083 =86706662670157$
  - $\Phi(n)=(p-1)(q-1)=9010278*9623082 =86706644036796$
  - $\Phi(n)=86706644036796$と素な関係にある数の例として $e=184436886841$
  - $ed=1 \pmod{\Phi(n)}$ となるdを互除法で求める
  - $d=70276475859277$
  - 公開鍵$(e,n)=(184436886841, 86706662670157)$
  - n未満の非負整数xに対して, $x^e \pmod n$ によって暗号化

Example of RSA (1)

- Concrete example of RSA
  - When prime numbers p, $q$are $p=9010279$, $q=9623083$
  - $n=pq=9010279*9623083 =86706662670157$
  - $\Phi(n)=(p-1)(q-1)=9010278*9623082 =86706644036796$
  - Example of number which is coprime to $\Phi(n)=86706644036796$ is $e=184436886841$
  - Obtain d which satisfies $ed=1 \pmod{\Phi(n)}$
  - $d=70276475859277$
  - Public key$(e,n)=(184436886841, 86706662670157)$
  - Encrypt a positive number x less than n by $x^e \pmod{n}$

RSA暗号の例 (2)

- RSA暗号の具体的な数値例
  - 素数p, qが $p=13$, $q=5$
  - $n=pq=13*5=65$
  - $\Phi(n)=12*4=48$と素である数の例$e=11$
  - $11d=1 \pmod{48}$ となるdを互除法で求める
  - $d=13$
  - 公開鍵$(e,n)=(11,65)$
  - 秘密鍵$d=13$
  - $x^e \pmod{n}=2^{11} \pmod{65} =2048 \pmod{65} = 33$
  - 復号は $b^d \pmod{n} = 33^{13} \pmod{65}$
    この時点で手計算不可能

Example of RSA (2)

- Concrete example of RSA
  - Prime numbers p, q are $p=13$, $q=5$
  - $n=pq=13*5=65$
  - A number e which is coprime to $\Phi(n)=12*4=48$, $e=11$
  - Obtain d which satifies $11d=1 \pmod{48}$
  - $d=13$
  - Public key $(e,n)=(11,65)$
  - Private key $d=13$
  - $x^e \pmod{n}=2^{11} \pmod{65} =2048 \pmod{65} = 33$
  - Decryption $b^d \pmod{n} = 33^{13} \pmod{65}$
    Difficult to calculate by hand

RSA暗号の例 (3)

- RSA暗号の具体的な数値例
  - 素数p, qが $p=11$, $q=3$
  - $n=pq=11*3=33$
  - $\Phi(n)=10*2=20$と素である数の例$e=7$
  - $7d=1 \pmod{20}$ となるdを互除法で求める
  - $d=3$
  - 公開鍵$(e,n)=(7, 33)$
  - 秘密鍵$d=3$
  - 暗号化は, $x^e \pmod{n} = 3^7 \pmod{33} = 2187 \pmod{33} = 9$
  - 復号は, $b^d \pmod{n} = 9^3 \pmod{33} = 729 \pmod{33} = 3$ (復号できた)

Example of RSA (3)

- Concrete example of RSA
  - Prime numbers p, q are $p=11, q=3$
  - $n=pq=11*3=33$
  - Coprime number $e=7$ to $\Phi(n)=10*2=20$
  - Obtain d which satisfies $7d=1 \pmod{20}$
  - $d=3$
  - Public key $(e,n)=(7, 33)$
  - Private key $d=3$
  - Encryption , $x^e \pmod{n} = 3^7 \pmod{33} = 2187 \pmod{33} = 9$
  - Decryption, $b^d \pmod{n} = 9^3 \pmod{33} = 729 \pmod{33} = 3$ (done)

RSA暗号の例 (4)

- 例(3)の結果の確認
  - $a' = b^d \pmod n = a^{ed} \pmod n$
  - $b^d \pmod n = 9^3 \pmod{33} = 729 \pmod{33} = 3$
  - $a^{ed} \pmod n = 3^{(7*3)} \pmod{33} = 3^{21} \pmod{33} = 10460353203 \pmod{33} = 3$

    $316980400*33=1046353200$

Example of RSA (4)

- Confirmation of example (3)
  - $a' = b^d \pmod n = a^{ed} \pmod n$
  - $b^d \pmod n = 9^3 \pmod{33} = 729 \pmod{33} = 3$
  - $a^{ed} \pmod n = 3^{(7*3)} \pmod{33} = 3^{21} \pmod{33} = 10460353203 \pmod{33} = 3$

    $316980400*33=1046353200$

ElGamal暗号 (1)

- ElGamal暗号(ElGamal Encryption)
  - 位数が大きな群の離散対数問題が困難であることを安全性の根拠とした公開鍵暗号(1984)
  - Diffie-Hellman鍵共有方式で共有した乱数を1回だけ使う暗号通信

- 暗号方式
  - 鍵生成
    - 巡回群Gで，位数qが素数かつqがkビットであるものを選ぶ
    - Gの生成元 g を選ぶ
    - xを$\{0,\ldots, q-1\}$からランダムに選ぶ
    - $h = g^x$とする
    - 公開鍵:(G,q,g,h)，秘密鍵:x

ElGamal Encryption (1)

- ElGamal Encryption
  - Public key encryption based on difficulty of discrete exponential problem of a group with large order (1984)
  - Cipher communication using random number which is used in Diffie-Hellman key exchange
- Encryption method
  - Key generation
    - Cyclic group G with order q which is prime number and k bit
    - Select element g *of* G, and x from $\{0,\ldots, q-1\}$
    - $h = g^x$
    - Public key :(G,q,g,h), Privete key:x

ElGamal暗号 (2)

- 暗号方式(続き)
  - 暗号化
    - Gの元mを平文として受け取る
    - rを$\{0,\ldots, q-1\}$からランダムに選ぶ
    - $c_1 = g^r$, $c_2 = mh^r$ を計算
    - 暗号文: (c_1, c_2)
  - 復号
    - $m = c_2 (c_1^x)^{-1}$ で平文を得る
- 完全性
  - $m' = c_2 (c_1^x)^{-1} = m (g^x)^r ((g^r)^x)^{-1} = m$

ElGamal Encryption (2)

- Encryption method (Condt.)
  - Encryption
    - Receive element m of G as a message
    - Chose r from $\{0,\ldots, q-1\}$ randomly
    - Calculate $c_1 = g^r$, $c_2 = mh^r$
    - Cipher message: (c_1, c_2)
  - Decryption
    - $m = c_2 (c_1^x)^{-1}$
- Correctness
  - $m' = c_2 (c_1^x)^{-1} = m (g^x)^r ((g^r)^x)^{-1} = m$

楕円曲線暗号

- 楕円曲線暗号(Elliptic Curve Cryptography: ECC)
  - 楕円曲線上の離散対数問題 (EC-DLP) を安全性の根拠とする暗号
  - ビクタ・ミラー (Victor Miller) とニール・コブリッツ (Neal Koblitz) が発明 (1985)
  - 楕円RSA, DSAを楕円曲線上で定義した楕円DSA (EC-DSA), DH鍵共有を楕円化した楕円DHなど
  - EC-DLPを解く準指数関数時間アルゴリズムは未発見
  - RSA暗号と比べて同レベルの安全性をより短い鍵で実現でき, 処理速度も速いことがメリット
  - 楕円曲線の方程式を $E=y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$ は加算・乗算に関して群をなすことを利用

Elliptic Curve Cryptography

- Elliptic Curve Cryptography: ECC
  - Based on safety of Elliptic Curve Discrete Logarithmic Problem 9EC-DLP)
  - Invented by Victor Miller and Neal Koblitz (1985)
  - EC-RSA, EC-DSA in which DSA is defined on elliptic curve, EC-DH in which DH key exchange is defined on elliptic curve
  - Semi-exponential functional time algorithm to solve EC-DLP is not found
  - Same level security with RSA with shorter key, faster processing
  - Utilize characteristics that elliptic curve function $E=y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$ makes a group

RSA暗号の応用

- RSA暗号の応用
  - ハイブリッド方式
  - Transport layer security
  - IPsec
  - PKI
- 楕円曲線暗号の応用
  - 標準化作業中
  - 実際のアプリケーションには未使用

Application of RSA

- Application of RSA
  - Hybrid method
  - Transport layer security
  - IPsec
  - PKI
- Application of elliptic curve encryption
  - Under standardization work
  - Not yet used for real applications