社団法人　情報処理学会　研究報告
IPSJ SIG Technical Reports

2005－ＣＳＥＣ－30（55）
2005／7／22

DOM ナップザック暗号の低密度攻撃に対する安全性の計算機実験による評価

名迫　健†　村上　恭通†

† 大阪電気通信大学通信工学科
〒572-8530 大阪府寝屋川市初町 18-8
E-mail: †nasako@m.ieice.org, ††yasuyuki@isc.osakac.ac.jp

あらまし　密度の低いナップザック暗号に有効な攻撃法に低密度攻撃がある．2005 年筆者らは，高密度なナップザック暗号である DOM ナップザック暗号を提案した．DOM ナップザック暗号は，密度が 1 以上に設定することが可能であり，低密度攻撃に高い耐性を有すると考えているが，実際に計算機実験による安全性の評価はまだ試みていなかった．本稿では，計算機により DOM ナップザック暗号に対し低密度攻撃により解読実験を試み，その結果，DOM ナップザック暗号は低密度攻撃に対して高い耐性を有することを確認する．

キーワード　ナップザック暗号，DOM ナップザック暗号，プリコーディング，高密度，低密度攻撃

Security of DOM Knapsack PKC against Low-Density Attack by Computer Experiment

Takeshi NASAKO† and Yasuyuki MURAKAMI†

† Department of Telecommunications and Competur Networks, Osaka Electro-Communication University
18-8, Hatsu-Cho, Neyagawa, Osaka, 572-8530 Japan
E-mail: †nasako@m.ieice.org, ††yasuyuki@isc.osakac.ac.jp

Abstract　The low-density attack(LDA) is an effective attack to the knapsack cryptosystems when the density is low. We proposed the DOM knapsack cryptosystem(DOM PKC) as a high-density knapsack cryptosystem in SCIS 2005. We think DOM PKC is secure against the low-density attack because the density of DOM PKC can be made above 1. In this paper, we confirm that DOM PKC is secure against the low-density attack with computer experiments.

Key words　knapsack PKC, DOM PKC, pre-coding, high density, low-density attack

1. はじめに

Merkle と Hellman は，暗号化処理が加算のみにより実行し得る公開鍵暗号として超増加数列をトラップドアに用いたナップザック暗号 (MH 暗号) を提案した [1]．しかしながら，MH 暗号は Shamir の攻撃 [2] により容易に秘密鍵が露呈するとされ，また低密度攻撃 [3]～[5] によって容易に平文が求められることが知られている．このことや，それ以降に提案されたナップザック暗号のほとんどが解読されていることから，MH 暗号をはじめとしてナップザック暗号の安全性は疑問視されていた．しかしながら，これだけを根拠にすべてのナップザック暗号が安全ではないと結論付けることはできない．また，量子コンピュータが実現すると，素因数分解問題，離散対数問題，楕円離散対数問題が解かれてしまうことが示され，多くの公開鍵暗号は解読されることが知られている．したがって，安全なナップザック暗号を探求することは非常に重要な意味を持つ．

実際，服部，村上，笠原らは低密度攻撃（LDA）に対して高い耐性を有するナップザック暗号方式として，プリコーディングを用いた高密度なナップザック暗号である SHP-III, IV, VII 暗号などを提案した [6], [7]．また，最近，筆者らも法縮小という高密度化手法を提案し，異なる観点から低密度攻撃に高い耐性を有する方式として CHK 暗号を提案している [8]．さらに，筆者らは SHP-III 暗号に用いられているプリコーディングという手法に着目し，MH 暗号同様に超増加性をトラップドアに用いた方式に応用した DOM ナップザック暗号を提案した [9]．

本稿では，計算機により DOM ナップザック暗号に対し，ナップザック暗号に有効であるとされている低密度攻撃を用いた解読実験を試みる．その結果，DOM ナップザック暗号は低密度

攻撃に対して高い耐性を有することを確認する.

2. DOM ナップザック暗号

本節では，MH 暗号を基礎とし，プリコーディングを使用した方式である DOM 暗号について述べる.

2.1 プリコーディング

暗号化に先立って，平文ベクトルを符号化平文ベクトルに変換する操作をプリコーディングと呼び，平文を拡大することによりナップザック暗号の高密度化を実現している手法である [6], [7].

本稿で扱うプリコーディングは，平文ベクトルおよび符号化平文ベクトルのいずれも n 次元 2 進ベクトルとし，それぞれ，$\boldsymbol{m} = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ および $\boldsymbol{m}^* = (m_1^*, m_2^*, \ldots, m_n^*) \in \{0,1\}^n$ と表記する．これら n 次元のプリコーディング全体の集合を $\mathcal{PC}_n$ とする.

2.1.1 逐次的プリコーディング

逐次的に復号を保証していくためには，符号化平文ベクトルの第 i 成分を平文ベクトルの第 1 成分から第 i 成分により決定する必要がある．すなわち，適当な写像 $\mathcal{F}_i : \{0,1\}^i \to \{0,1\}$ により，

$$m_i^* = \mathcal{F}_i(m_1, m_2, \ldots, m_i) \ (i = 1, 2, \ldots, n)$$

と表される必要がある．このような性質を持つプリコーディングを逐次的プリコーディングと呼び，n 次元の逐次的プリコーディング全体の集合を $\mathcal{SPC}_n$ とする.

逐次的プリコーディングは $\mathcal{F}_i(m_1, m_2, \ldots, m_i)$ の集合であるが，以後，単に $\mathcal{F}$ と書き，$\boldsymbol{m}^* = \mathcal{F}(\boldsymbol{m})$ と表す.

2.1.2 可逆プリコーディング

逐次的プリコーディング $\mathcal{H} \in \mathcal{SPC}_n$ のうち，

$i = 1$ のとき，

$$m_i^* = \mathcal{H}_i(m_i) = j \oplus m_i$$

ただし，j は 0, 1 のいずれかである．$i = 2, 3, \ldots, n$ については，任意の写像 $\mathcal{F}_{i-1} : \{0,1\}^{i-1} \to \{0,1\}$ とし，

$$\mathcal{H}_i(m_1, m_2, \ldots, m_{i-1}, m_i) = \mathcal{F}_{i-1}(m_1, m_2, \ldots, m_{i-1}) \oplus m_i$$

となるプリコーディングを定義する．このとき，

$$\begin{cases} m_i^* = \mathcal{H}_i(m_1, m_2, \ldots, m_{i-1}, m_i) \\ m_i = \mathcal{H}_i(m_1, m_2, \ldots, m_{i-1}, m_i^*) \end{cases}$$

が成立するため，可逆プリコーディングと定義する．なお，この可逆プリコーディング全体の集合を $\mathcal{IPC}_n$ とする.

喜安-Gray 逆変換に基づく二通りのプリコーディング $\mathcal{G}_0$ および $\mathcal{G}_1$ をそれぞれ以下のように定義する [6]．これらは可逆プリコーディングである.

$$\mathcal{G}_{0i}(m_1, m_2, \ldots, m_i) = \bigoplus_{k=1}^{i} m_k$$

$$\mathcal{G}_{1i}(m_1, m_2, \ldots, m_i) = 1 \oplus \bigoplus_{k=1}^{i} m_k$$

可逆プリコーディングを合成して生成される任意のプリコーディングは可逆プリコーディングである．DOM 暗号は可逆プリコーディングを用いた暗号方式である.

2.2 鍵 生 成

DOM 暗号の鍵を以下に示す.

秘密鍵：$P, \boldsymbol{v}, \boldsymbol{u}, w, \boldsymbol{t}$
公開鍵：$\boldsymbol{a}, \boldsymbol{b}, \mathcal{H}_i, n$

Bob は，可逆プリコーディング $\mathcal{H} \in \mathcal{IPC}_n$ を定め，それを公開し，以下に述べる手順に従って鍵を生成する.

まず，r-bit の正整数乱数ベクトルを成分とする $\boldsymbol{u} = (u_1, u_2, \ldots, u_n)$ を生成する.

次に，超増加ベクトル $\boldsymbol{v} = (v_1, v_2, \ldots, v_n)$ を生成する．ただし，v_i を

$$v_i > u_i + s_{i-1} \quad (i = 1, 2, \ldots, n)$$

を満たす正整数乱数とする．ただし，

$$s_i = \sum_{k=1}^{i} (u_k + v_k)$$

とする.

さらに，素数 P を

$$P > s_n$$

を満たすように生成する.

また，秘密鍵 $w \in \mathbb{Z}_P^*$ をランダムに生成する．さらに，次式に従ってモジュラ変換することにより $\boldsymbol{e} = (e_1, e_2, \ldots, e_n)$ および $\boldsymbol{f} = (f_1, f_2, \ldots, f_n)$ を得る.

$$\boldsymbol{e} = w\boldsymbol{v} \bmod P$$

$$\boldsymbol{f} = w\boldsymbol{u} \bmod P$$

最後に，ランダムに $\boldsymbol{t} = (t_1, t_2, \ldots, t_n) \in \{0,1\}^n$ を生成し，次式に従って公開鍵 $\boldsymbol{a} = (a_1, a_2, \ldots, a_n)$ および $\boldsymbol{b} = (b_1, b_2, \ldots, b_n)$ を得る.

$$a_i = \begin{cases} e_i & (\text{when} \quad t_i = 0) \\ f_i & (\text{when} \quad t_i = 1) \end{cases}$$

$$b_i = \begin{cases} f_i & (\text{when} \quad t_i = 0) \\ e_i & (\text{when} \quad t_i = 1) \end{cases}$$

公開鍵は $\boldsymbol{t}$ の値により，シャフルされていることに注意されたい.

なお，u_i および v_i が全数探索により露呈しない程度に設定するのが望ましい

2.3 暗 号 化

Alice は，平文 $\boldsymbol{m} \in \{0,1\}^n$ とプリコーディング $\mathcal{H}$ により，$\boldsymbol{m}$ から符号化平文ベクトル $\boldsymbol{m}^* = \mathcal{H}(\boldsymbol{m})$ を求め，公開鍵ベクトル $\boldsymbol{a}$ および $\boldsymbol{b}$ を用いて，次式により暗号化を行い，暗号文

C を得る.

暗号化アルゴリズム

```
for i = n downto 1 {
    m_i^* = H_{n-i+1}(m_n, m_{n-1}, ..., m_{i+1}, m_i)
}
C = ma + m^* b
```

2.4 復　　号

Bob は，まず，中間平文 M を

$$M = w^{-1}C \bmod P$$

として求め，以下に示す復号アルゴリズムによって復号を行う．$\widehat{\boldsymbol{m}} = (\widehat{m}_1, \widehat{m}_2, \ldots, \widehat{m}_n)$ および $\widehat{\boldsymbol{m}}^* = (\widehat{m}_1^*, \widehat{m}_2^*, \ldots, \widehat{m}_n^*)$ を

$$\widehat{m}_i = \begin{cases} m_i & (\text{when} \quad t_i = 0) \\ m_i^* & (\text{when} \quad t_i = 1) \end{cases}$$

$$\widehat{m}_i^* = \begin{cases} m_i^* & (\text{when} \quad t_i = 0) \\ m_i & (\text{when} \quad t_i = 1) \end{cases}$$

とすると，暗号文 C は $\widehat{\boldsymbol{m}}$ および $\widehat{\boldsymbol{m}}^*$ により次のように表すことができる．

$$C = \widehat{\boldsymbol{m}}\boldsymbol{e} + \widehat{\boldsymbol{m}}^*\boldsymbol{f}$$

したがって，中間平文 M は

$$M = \widehat{\boldsymbol{m}}\boldsymbol{v} + \widehat{\boldsymbol{m}}^*\boldsymbol{u}$$

となっている．

復号アルゴリズム

```
for i = n downto 1 {
    if (M ≧ v_i) {
        m̂_i = 1
    } else {
        m̂_i = 0
    }
    m̂_i^* = H_{n-i+1}(m̂_n, m̂_{n-1}, ..., m̂_{i+1}, m̂_i)
    M ← M - m̂_i v_i - m̂_i^* u_i
    if (t_i = 0) {
        m_i = m̂_i
    } else {
        m_i = m̂_i^*
    }
}
if (M = 0) {
    output m
}
```

3. 低密度攻撃

本章ではナップザック問題とその密度および，低密度攻撃について説明する．低密度攻撃は，トラップドアの如何にかかわらず低密度なナップザック暗号の解読に有効な攻撃法である．代表的なものに Lagarias, Odlyzko による方法 [4] や，Coster らによる方法 [5] などがある．

3.1 0-1 ナップザック問題とその密度

0-1 ナップザック問題とは，正整数の集合 $A = \{a_1, a_2, \ldots, a_n\}$（ナップザックと呼ばれる）と，その部分集合の和 C が与えられたときに，その部分集合を見出す問題である．換言すると，一次不定方程式

$$a_1x_1 + a_2x_2 + \cdots + a_nx_n = C \tag{1}$$

の解 $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ を見出す問題である．

ナップザック A の密度 d は，次式で定義される [4].

$$d = \frac{n}{\log_2(\max a_i)} \tag{2}$$

一般に 0-1 ナップザック問題は NP 完全問題であるが，超増加数列をナップザックとする場合は，簡単なナップザック問題と呼ばれ，容易に解くことができる．

ナップザック暗号はナップザックを公開鍵とし，0-1 ナップザック問題の難しさに安全性の根拠を置く．

3.2 高密度なナップザック問題

密度 d が 1 を越えるナップザック問題は，一つの C 対して，無数の解が存在し得る．平文空間と暗号文空間の比は，$2^n : 2^{dn}$ と表すことができる．したがって，密度が 1 を越えるときの一つの C に対応する解はおよそ $2^{(d-1)\log_2(\max a_i)}$ 個あると考えられる．すなわち，密度が 1 を越えるナップザック暗号においては，ある暗号文 C に対する平文は，無数に存在する解の中のひとつであり，無数に存在する解の中から真の平文を特定することは非常に困難である．したがって，密度が 1 を超えることは，安全性に非常に重要な意味をもつ．

3.3 LO 法

Lagarias, Odlyzko は，正整数の集合 $A = \{a_1, a_2, \ldots, a_n\}$ と，それらの部分集合の和 C を用いて，行列

$$B = \begin{pmatrix} 1 & 0 & \ldots & 0 & -a_1 \\ 0 & 1 & \ldots & 0 & -a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -a_n \\ 0 & 0 & \ldots & 0 & C \end{pmatrix}$$

を用意し，この各行ベクトルが張る格子 $L(B)$ に対し格子基底縮小を行うことにより，式 (1) の解 $(x_1, x_2, \ldots, x_n)$ を発見する解法を提案した（LO 法）．

この格子 $L(B)$ には $\boldsymbol{x} = (x_1, x_2, \ldots, x_n, 0)$ が含まれており，0-1 ナップザック問題においては $x_i \in \{0,1\}$ であるので，$\boldsymbol{x}$ のユークリッドノルムはごく小さい．したがって，格子基底縮小アルゴリズムを適用することによって，$L(B)$ に含まれる最短ベクトルを求めることができると，それが $\boldsymbol{x}$ である可能性が高く，密度 $d < 0.6463\ldots$ のとき $\boldsymbol{x}$ が最短ベクトルである確率が 1 であることが示されている [4].

3.4 CLOS 法

Coster, LaMacchia, Odlyzko, Schnorr は，LO 法より密度の高いナップザック問題に対しても有効である改良手法を提案した（CLOS 法）．

CLOS 法は，$\lambda > \sqrt{n}$ なる正整数 λ を定め，正整数の集合 $A = \{a_1, a_2, \ldots, a_n\}$ と，それらの部分集合の和 C と λ を用いて，行列

$$B' = \begin{pmatrix} 1 & 0 & \ldots & 0 & -\lambda a_1 \\ 0 & 1 & \ldots & 0 & -\lambda a_2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & -\lambda a_n \\ -1/2 & -1/2 & \ldots & -1/2 & \lambda C \end{pmatrix}$$

を用意し，この各行ベクトルが張る格子 $L(B')$ に対し格子基底縮小を行うことにより，式 (1) の解 x_i を発見する解法である．この格子 $L(B')$ には $\boldsymbol{x} = (x_1 - 1/2, x_2 - 1/2, \ldots, x_n - 1/2, 0)$ が含まれており，このユークリッドノルムはごく小さい．したがって，格子基底縮小アルゴリズムを適用することによって，$L(B')$ に含まれる最短ベクトルを求めることができると，それが $\boldsymbol{x}$ である可能性が高く，密度 $d < 0.9408\ldots$ のとき $\boldsymbol{x}$ が最短ベクトルである確率が 1 であることが示されている [5]．

4. DOM 暗号の低密度攻撃に対する安全性

本章では，低密度攻撃に対する DOM ナップザック暗号の安全性について考察する．

4.1 低密度攻撃

プリコーディングを用いた暗号方式に対して適用することができる低密度攻撃に連接鍵攻撃がある．本稿では，DOM 暗号に対して CLOS 法による連接鍵攻撃(注1) を計算機実験により実際に適用し，安全であることを確認する．なお，計算機実験においては，格子基底縮小アルゴリズムに LLL アルゴリズム [10] を使用した．

4.2 連接鍵攻撃（CLOS 法）

DOM 暗号の解読問題は，連接平文 $\widetilde{\boldsymbol{m}} = (m_1, m_2, \ldots, m_n, m_1^*, m_2^*, \ldots, m_n^*)$ および，連接鍵ベクトル $\boldsymbol{c} = (c_1, c_2, \ldots, c_{2n}) = (a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n)$ により，

$$\begin{aligned} C &= \widetilde{\boldsymbol{m}}\boldsymbol{c} \\ &= \boldsymbol{m}\boldsymbol{a} + \boldsymbol{m}^*\boldsymbol{b} \end{aligned}$$

なる $C, \boldsymbol{c}$ より $\widetilde{\boldsymbol{m}}$ を求める $2n$ 次元のナップザック問題に変換することができる．

したがって，DOM 暗号を連接鍵攻撃により解読する際の密度 d' は式 (2) より，

$$d' = \frac{2n}{\log_2(\max c_i)}$$

となる．

(注1)：一般に CLOS 法は LO 法よりも高密度のナップザック問題を解くことができるので，本稿では CLOS 法を用いることにする．

連接鍵攻撃は，$2n$ 次元のナップザック問題に対して低密度攻撃を適用することにより，この暗号を解読しようとする攻撃である．

DOM 暗号に対して，CLOS 法を用いた連接鍵攻撃を適用するには，$\lambda > \sqrt{2n}$ なる正整数 λ を定め，行列

$$B'' = \begin{pmatrix} 2 & \ldots & 0 & 0 & \ldots & 0 & -2\lambda a_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 2 & 0 & \ldots & 0 & -2\lambda a_n \\ 0 & \ldots & 0 & 2 & \ldots & 0 & -2\lambda b_1 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \ldots & 0 & 0 & \ldots & 2 & -2\lambda b_n \\ -1 & \ldots & -1 & -1 & \ldots & -1 & 2\lambda C \end{pmatrix}$$

を用意し，この各行ベクトルが張る格子 $L(B'')$ に格子基底縮小アルゴリズムを適用することによって，$L(B'')$ に含まれる最小ベクトルを求める．

5. 解読実験結果

DOM 暗号に対して $\lambda = 101$ (注2) とした行列 B'' を用いた CLOS 法により連接鍵攻撃を適用して，以下の手順で解読実験を行った(注3)．

- 次元 $n = 8, 9, \ldots, 35$ および $n = 40, 48, 56, 64$ について行った．
- 各次元について 100 個の公開鍵を作成し，各鍵について 100 個の平文をランダムに発生させ，それらを暗号化した暗号文に対して攻撃を行った．
- u_i のビット長を 32-bit と 64-bit の二つ場合について行った．

実験の解読率を図 1 および図 2 にそれぞれ，横軸を次元として，および横軸を密度として表す．

図 1 より，パラメータ設定 $r = 32$，および $r = 64$ のときにそれぞれ，$n > 25$ となる範囲，および $n > 30$ となる範囲ではまったく解読できないことが確認できた．

また，図 2 より，パラメータ設定 $r = 32$，および $r = 64$ のときにそれぞれ，$\rho > 0.7$ となる範囲，および $\rho > 0.9$ となる範囲ではまったく解読できないことが確認できた．

6. 考　　察

本研究では，各次元に対して 10000 回の解読実験を行ったが，一般に，この回数は必ずしも十分ではない．しかしながら，図 1 および図 2 において，急激に解読率が減少している事実から，これ以上多くの実験を行っても解読されることはないと考えられる．とはいえ，安全性に対してより信頼を得るためには，今後，より多くの実験を試みる必要があると考えている．

(注2)：正整数 λ に関しては，あらかじめ種々の数値で実験を行った結果，λ の値が小さ過ぎると解読率が悪くなり，大き過ぎると解読にかかる時間が長くなる．このことを考慮し，今回の実験においては $\lambda = 101$ を採用した．

(注3)：OS: VineLinux 3.1，コンパイラ: gcc 4.0.5，ライブラリ: NTL ver.5.4 [11]．

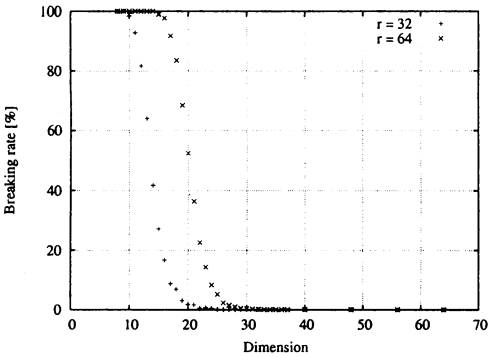

図 1　次元と解読率

Fig. 1　Breaking rate on n.

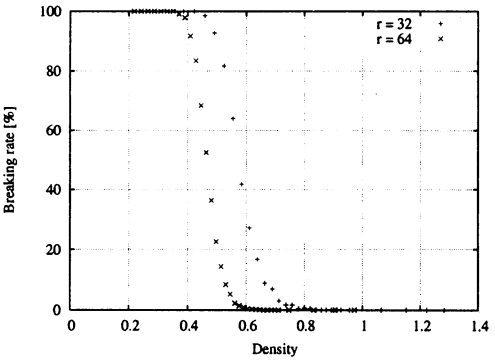

図 2　密度と解読率

Fig. 2　Breaking rate on density.

実際，密度が 1 を越えるナップザック暗号方式の解読実験において無数にある解のひとつを見付けたときがあったが，その解は，平文よりもノルムの小さい解であり，平文ではなかった．

なお，DOM 暗号の実用的なパラメータ設定において，例えば，$n = 128$, $r = 64$ と設定すると，$2n = 256$, $\log_2 \max c_i = 196$ (注4) と設計することが可能である．したがって，この設計において，暗号文 C に対応する解の個数はおよそ $2^{(d-1)\log_2(\max a_i)}$ 個あると考えられので，およそ 2^{60} 個程度存在することがわかる．この中から最小ノルムとは限らない唯一の平文を発見することは非常に困難であると結論付けられる．

7. むすび

DOM 暗号は 1 を越える高い密度を実現することが可能な暗号方式である．実際に計算機を用いて解読実験を行った結果，CLOS 法を用いた低密度攻撃を応用した連接鍵攻撃に対して DOM 暗号は安全であることが確認できた．

また，実験結果に対して詳細に考察を行った結果，DOM 暗号は実用的なパラメータにおいて，十分安全に使用することができることが明らかとなった．

DOM 暗号では，二種の秘密鍵の和は超増加数列となるため，Shamir の攻撃により解読される恐れがあることが指摘された．筆者らは，次元が大きい場合における Shamir の攻撃の有効性を疑問視しているが，DOM 暗号の Shamir の攻撃に対する安全性を詳細には検討していない．本件については，今後より詳細に検討を行う所存である．

(注4)：今回の実装において，上記の設定でデータを取ったところ，$\log_2 \max c_i = 196$, $d' = 1.302$ となった．

文　献

[1] R. C. Merkle, M. E. Hellman: "Hiding information and signatures in trapdoor knapsacks," IEEE Trans. Inf. Theory, IT-24(5), pp.525–530, Sept. 1978.

[2] A. Shamir : "A polynomial time algorithm for breaking the basic Merkle-Hellman cryptosystems," Proc. Crypto'82, LNCS, pp.279–288, Springer-Verlag, Berlin, 1982.

[3] E. F. Brickell : "Solving low density knapsacks," Proc. Crypto'83, LNCS, pp.25–37, Springer-Verlag, Berlin, 1984.

[4] J. C. Lagarias and A. M. Odlyzko : "Solving Low Density Subset Sum Problems," J. Assoc. Comp. Math., vol.32, pp.229–246, Preliminary version in Proc. 24th IEEE, 1985.

[5] M. J. Coster, B. A. LaMacchia, A. M. Odlyzko and C. P. Schnorr : "An Improved Low-Density Subset Sum Algorithm," In Advances in Cryptology Proc. EUROCRYPTO'91, LNCS, pp.54–67. Springer-Verlag, Berlin, 1991.

[6] 服部 保, 村上 恭通, 笠原 正雄 : "プリコーディングを用いる二, 三の加算暗号の提案", 2001 年 情報理論とその応用シンポジウム予稿集, pp.351–354, Dec. 2001.

[7] 服部 保, 村上 恭通, 笠原 正雄 : "SHP 暗号の安全性に関する二, 三の考察", 2002 年 暗号と情報セキュリティシンポジウム予稿集, pp.131–136, Jan. 2002.

[8] 名迫 健, 横山 晃子, 村上 恭通 : "ナップザック暗号の法縮小による高密度化手法および探索復号法の提案", 2004 年 情報理論とその応用シンポジウム予稿集, pp.123–126, Dec. 2004.

[9] 名迫 健, 横山 晃子, 村上 恭通 : "プリコーディングを用いた高密度 MH 型ナップザック暗号の提案", 2005 年 暗号と情報セキュリティシンポジウム予稿集, pp.949–954, Jan. 2005.

[10] A. K. Lenstra, H. W. Lenstra and L. Lovàsz : "Factoring polynomials with integer coefficients," Mathematische Annalen 261, pp.515–534, 1982.

[11] Victor Shoup : "NTL : A library for doing number theory", http://www.shoup.net/ntl/

[12] C. P. Schnorr, M. Euchner: "Lattice basis reduction: Improved practical algorithms and solving subset sum problems," Mathematical Programming, Vol.66, pp.181–191,1994.