論文

NFC を用いた ElGamal 暗号しきい値復号システムの開発

Development of a Threshold Decoding System
for ElGamal Encryption Using NFC

大村 光徳，宮崎 真一郎，松嶋 智子，山嵜 彰一郎
Kotoku Omura, Shinichiro Miyazaki, Tomoko K. Matsushima and Shoichiro Yamasaki

A (*k*,*n*) threshold based secret sharing scheme has been used to improve both safety and reliability of highly confidential information. When this scheme is applied to preservation of secret key in public key cryptography, there is a risk of the reconstructed secret key being stolen. To overcome such a drawback, a (*k*,*n*) threshold based encoding scheme for ElGamal encryption system has been proposed. In this work, we developed a threshold decoding system for ElGamal encryption using NFC (Near Field Communication). The developed system distributes secret keys of ElGamal cryptographic system to NFC devices such as IC cards and Android smartphones using (*k*,*n*) threshold scheme. We confirmed that the developed system can decode cryptograms without reconstructing the secret key.
Keywords: ElGamal cryptographic system, (*k*,*n*) threshold scheme, Near Field Communication, Android

1. はじめに

近年，個人情報のみならず，組織の機密情報の漏えいに関する事件が大きな社会問題となっている．NPO ネットワークセキュリティ協会の調査[1]では，2017 年度に生じた情報漏えいインシデントのうち、件数において約 60%は、誤操作，紛失・置き忘れ，管理ミスなどの組織内部の人為的ミスが原因であると報告している．また，17.4%は外部からの不正アクセスが原因であると報告している．組織内部の人為的ミスおよび外部からの不正アクセスを完全に排除できない現状では，第三者に知られたくない秘密情報を，意図した相手のみが入手できるように保護する必要がある．暗号技術は，この要求に対する有効な解決策であり，情報化社会を支える基盤技術の一つとなっている．

情報を暗号化する方式には，大別して共通鍵暗号方式と公開鍵暗号方式の二つがある．共通鍵暗号方式は，暗号化鍵と復号鍵が同じ，もしくは暗号化鍵から復号鍵を容易に推測できる方式である．一方，公開鍵暗号方式は暗号化鍵と復号鍵が異なり，暗号化鍵から復号鍵を求めることが難しい方式である．現在では多くの実用的なシステムで公開鍵暗号が利用されている．公開鍵暗号において，暗号化鍵は公開鍵として公開し，復号鍵は秘密鍵として正当な所有者以外には漏れないように厳重に管理される．秘密鍵を何らかの方法により盗聴者が入手すれば，暗号文は盗聴者に容易に解読されてしまう．また，秘密鍵の所有者が秘密鍵を紛失した場合，正当な所有者であっても暗号文を復号することが不可能となる．そのため，公開鍵暗号では秘密鍵の管理が重要な課題となっている．

秘密鍵などの機密性の高い情報の管理において，安全性および信頼性を同時に高める方法として，秘密分散法が考案された．秘密分散法の代表的な方式として，Shamir の(k,n)しきい値法[2]がある．公開鍵暗号の秘密鍵を保存する際に，(k,n)しきい値法を適用すれば，信頼性を高めることが可能である．しかし，この方法を単純に適用した場合，復号の際に計算機上に秘密鍵が復元される．つまり，秘密鍵が盗難にあう危険性を完全には排除できない．そこで，公開鍵暗号の方式として拡張 ElGamal 暗号を用い，これに(k,n)しきい値法で分配された秘密鍵のシェアを持つk個の端末が協力して復号処理を行うことで，秘密鍵が復元されることなく，暗号文の復号を行う方式が提案された．これをしきい値復号と呼ぶ[3]．しきい値復号では，それぞれの端末が計算能力を有している必要がある．

ところで，Near Field Communication (NFC)が，チケット発行，料金決済などのサービスにおいて，口座番号などの秘密情報の交換を実現する近距離無線技術として注目されている．また，NFC の機能が実装された数多くのスマートフォンが普及し，それらのサービスに広く利用されている．

そこで，本研究において，暗号文の復号時に秘密鍵の盗難の危険性を排除した，「NFC を用いた ElGamal 暗号しきい値復号システム」の開発を行った．

本稿では，2 章で開発システムに関連する公開鍵暗号技術，3 章で秘密分散法，4 章で NFC 規格について，お

よび 5 章で開発したシステムを説明する．最後に，6 章にてまとめを述べる．

2. 公開鍵暗号

2.1. 公開鍵暗号の概要

公開鍵暗号は，暗号化鍵と復号鍵が異なり，暗号化鍵から復号鍵を求めることが現実的には困難であるように設計されている．例えば，Alice が鍵生成アルゴリズムにより秘密鍵S_Kおよび公開鍵P_Kのペアを作成し，公開鍵P_Kを公開しているとき，Bob が Alice にメッセージ(以降，平文)Mを暗号化して送る手順は以下の通りである．

(1) Bob は公開されている Alice の公開鍵P_Kを入手する．
(2) Bob は公開鍵P_Kを用いて，暗号化アルゴリズムにより平文 Mから暗号文 Cを生成する．
(3) Bob は暗号文 Cを Alice に送信する．
(4) Alice は自分だけが知っている秘密鍵S_Kを用いて，暗号文 Cから平文 Mを復号する．

暗号文 Cを復号し平文 Mを取得できるのは，平文 Mの暗号化に用いた公開鍵P_Kとペアとなる秘密鍵S_Kを持っている者のみである．つまり，通信路上で暗号文 Cが第三者に盗聴されることがあっても，秘密鍵S_Kが盗まれない限り，平文 Mの内容が漏えいしない，というのが公開鍵暗号の概念である．

上記のとおり，公開鍵暗号では秘密鍵の保管・管理が重要となる．Alice の秘密鍵S_Kを何らかの方法により盗聴者が入手していれば，通信路上で盗聴した暗号文 Cから平文 Mを復号されてしまう．また，Alice が自分の秘密鍵S_Kを紛失した場合，Bob からの暗号文 Cから平文 Mを復号することが不可能となる．

2.2. 離散対数問題

現在使用されている多くの公開鍵暗号方式は，桁数の大きな数の素因数分解が困難である問題，あるいは離散対数問題を安全性の根拠として設計されている．

pが素数のとき， pと互いに素であるものの集合を$\mathbb{Z}_p^* = \{1,2,\cdots,p-1\}$と表す．$g$を$\mathbb{Z}_p^*$の原始元とすると，任意の$h \in \mathbb{Z}_p^*$に対し，

$$h = g^x \pmod{p}, \tag{1}$$

となるような整数$x(0 \le x \le p-2)$が存在する．このxをhの離散対数という．つまり，離散対数問題とは，素数p ，$\mathbb{Z}_p^*$の原始元gおよび$h \in \mathbb{Z}_p^*$が与えられたときに，式(1)を満たす，$x \in \{0,1,\cdots,p-2\}$を求めよ，という問題である．素数pが小さければ解は容易に求まる．しかし，現在，離散対数問題を解く効率的なアルゴリズムは発見されておらず， xからhを求めるのは容易であるが，pに大きな素数を選択すれば，hからxを求めるのが困難である．この非対称性が整数の素因数分解と乗算の関係に類似しており，公開鍵暗号システムにおける計算量的安全性の根拠となっている．

2.3. ElGamal 暗号

ElGamal 暗号[4]は，1984 年に ElGamal により提案された，有限体上の乗法群における離散対数問題の困難性を安全性の根拠としている公開鍵暗号方式である．ElGamal 暗号における，鍵の生成，暗号化，および復号の手順は以下の通りである．

(1)　鍵の生成

素数pおよび整数gを定める．ただし，gは$\mathbb{Z}_p^*$の原始元である．次に，ランダムな整数$x \in \mathbb{Z}_{p-1}$を選択し，$y = g^x \mod p$を計算する．このとき，$P_K = (p,g,y)$が公開鍵となり，$S_K = x$が秘密鍵となる．

(2)　平文の暗号化

乱数$r \in \mathbb{Z}_{p-1}$，および公開鍵P_Kを用いて，平文$M \in \mathbb{Z}_p$から暗号文$C = (C_1, C_2)$を下式により計算する．

$$C = (C_1, C_2) = (g^r \mod p,\ My^r \mod p). \tag{2}$$

(3)　暗号文の復号

秘密鍵$S_K = x$と暗号文$C = (C_1, C_2)$から，下式により平文Mを復号する．

$$M = C_2 / C_1^x. \tag{3}$$

この復号結果が正しいことは，下式により明らかである．

$$\begin{aligned} C_2/C_1^x &= My^r/(g^r)^x, \\ &= M(g^x)^r/(g^r)^x, \\ &= M. \end{aligned} \tag{4}$$

公開鍵暗号のひとつである RSA 暗号[5]は，入力に対して出力が 1 通りに決まる，確定的暗号方式である．対して，ElGamal 暗号は，暗号化アルゴリズムにより出力される暗号文Cの値が，乱数rの影響により不確定となる，確率暗号の性質を持つ．ElGamal 暗号における暗号化関数をEncとすると，平文$M' = M''$，$r' \ne r''$であれば，

$$\mathrm{Enc}(P_K, M', r') \ne \mathrm{Enc}(P_K, M'', r''), \tag{5}$$

となる．つまり，ElGamal 暗号を使用する場合，通信の都度乱数rを生成することが重要である．もし，同一の乱数r'によって異なる 2 つの平文$M' \ne M''$を暗号化すると，暗号文$C' = (C_1', C_2')$と$C'' = (C_1'', C_2'')$の間に，$C_2'/C_2'' = M'/M''$となる関係が成立することは，式(2)より明らかである．1 組のMおよびC_2のペアが解ると，他のすべてのMはC_2から復号され，既知平文攻撃による暗号文の解読が可能となってしまう．

2.4. 拡張 ElGamal 暗号

ElGamal 暗号の暗号文は，平方剰余記号の性質を利用して，平文Mが平方剰余であるか否かの部分情報が漏えいする可能性が指摘されている[6]．これは，原始元gの位数$p-1$が合成数であることに起因する．そこで，拡張 ElGamal 暗号と呼ばれる方式では，原始元gを，位数が素数である巡回群Gの生成元αに置き換えることにより，上記の問題が回避される．$\mathbb{Z}_p^*$の部分群である$G=\langle\alpha\rangle$は以下の通り構成する．

下式により，素数pおよびqを生成する．

$$p = 2q + 1. \tag{6}$$

p，qおよびgから，下式の通りαを決定する．

$$\alpha = g^{(p-1)/q} \mod p. \tag{7}$$

このαをq乗すると，Fermat の定理から，

$$\alpha^q = \left(g^{(p-1)/q}\right)^q \mod p = g^{p-1} \mod p \equiv 1, \tag{8}$$

となる．すなわち，$\mathbb{Z}_p^*$の部分群であり，位数がqの巡回群，

$$\langle\alpha\rangle = \{1, \alpha, \alpha^2, \cdots, \alpha^{q-1}\} \mod p, \tag{9}$$

が生成できる．

拡張 ElGamal 暗号における，鍵の生成，暗号化，および復号の手順は以下の通りである．

(1) 鍵の生成

ランダムな整数$x \in \mathbb{Z}_{q-1}$を選択し，$y = \alpha^x \mod p$を計算する．このとき，$P_K = (p, q, \alpha, y)$が公開鍵となり，$S_K = x$が秘密鍵となる．

(2) 平文の暗号化

乱数$r \in \mathbb{Z}_{q-1}$，および公開鍵P_Kを用いて，平文$M \in \langle\alpha\rangle$から暗号文$C = (C_1, C_2)$を下式により計算する．

$$C = (C_1, C_2) = (\alpha^r \mod p,\ My^r \mod p). \tag{10}$$

(3) 暗号文の復号

ElGamal 暗号と同様に，秘密鍵$S_K = x$と暗号文$C = (C_1, C_2)$から，式(3)により平文Mを復号する．

3. 秘密分散法

3.1. 概要

公開鍵暗号方式では秘密鍵の管理が重要となる．秘密鍵を紛失してしまうと，ペアとなる公開鍵で暗号化された暗号文の復号が不可能となり，可用性の損失を生じる．対策として，秘密鍵の複製を作成することが考えられる．しかし，複製による秘密鍵の盗難により，第三者に暗号文を復号される危険が増し，機密性の損失を生じる．秘密鍵などの秘密情報の管理において，可用性および機密性の両立を実現する策として，秘密分散法が注目されている．

3.2. (k, n)しきい値法

秘密分散法の 1 つである(*k*,*n*)しきい値法は，1979 年に Shamir により提案された[2]．(*k*,*n*)しきい値法では，秘密情報 S を n 個のシェア$(p_1, p_2, \cdots, p_n)$に分散する．このシェアを k 個以上集めると，秘密情報 S の復元が可能であるが，k 個未満のシェアでは復元不可能となる．図 1 に(3,5)しきい値法による秘密情報の分散，および復元の例を示す．

2 次元座標において，k 個の点$(i_1, y_{i_1}), (i_2, y_{i_2}), \cdots, (i_k, y_{i_k})$が与えられたとき(ただし，$i_1$，$i_2$，…，$i_k$は互いに異なる)，その点を通る多項式

$$y = f(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_{k-1} x^{k-1}, \tag{11}$$

が一意に定まる．多項式$f(x)$を求めるには，次式の Lagrange の補間公式を用いる．

$$f(x) = \sum_{j=1}^{k} y_{i_j} \lambda_j(x). \tag{12}$$

ただし，式(12)において，

$$\lambda_j(x) = \prod_{m=1, m\neq j}^{k} \frac{x - i_m}{i_j - i_m}, \tag{13}$$

とする．

(*k*,*n*)しきい値法は，この多項式補間を用いて秘密分散および復元を行う．また，(*k*,*n*)しきい値法の安全性は，前述の多項式補間の理論に基づいている．

3.2.1. (k, n)しきい値法による秘密分散手順

秘密情報Sを分散するために，$k < n$ を満たす k および

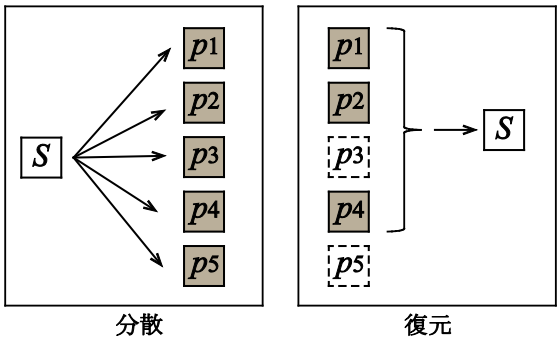

図 1 (3,5)しきい値法

n を選択する．次に，$q > \max(S, n)$ となる素数 q を選択する．式(11)において$a_0 = S$，他の係数を以下の通りランダ

ムに設定することにより，

$$0 \le a_i \le q-1 \ \ (i = 1,2,3,\cdots,k-1), \tag{14}$$

GF(q)上の k-1 次多項式$g(x)$を決定する．

$$g(x) = S + a_1 x + a_2 x^2 + \cdots a_{k-1} x^{k-1} \ \bmod \ q. \tag{15}$$

決定した多項式では，$g(0) = S$となることは明らかである．

次に，多項式$g(x)$を利用して，秘密情報 S から n 個のシェア

$$p_i = g(i) \qquad (i = 1,2,,\cdots,n), \tag{16}$$

を計算する．

3.2.2.　(k, n) しきい値法による秘密情報の復元

k-1 次多項式は，k 個のデータにより一意に定まることは述べた．すなわち，k 個のシェア$\{p_{i_1}, p_{i_2}, \cdots, p_{i_k}\}(1 \le i_j \le n)$から式(12)を用いて$g(x)$を定め，$g(0) = S$により秘密情報 S が復元できる．

3.3. 拡張 ElGamal 暗号のしきい値復号法

秘密情報の管理において可用性および機密性の両立を実現する方策として，(k,n)しきい値法を用いることを述べた．しかし，k 個以上のシェアを集め復元を行った計算機がセキュリティ上の問題を有する場合，秘密情報が盗難にあう可能性は排除できない．

そこで，拡張 ElGamal 暗号に(k,n)しきい値法を適用するにあたり，秘密鍵を復元することなく，k 個以上のシェアから暗号文を復号する方式が考えられた．

この方式では，秘密鍵S_Kを，秘密情報として式(15)における S に設定し，n 個のシェア$\{g(1), g(2), \cdots, g(n)\}$に分散し管理する．暗号文$C = (C_1, C_2)$ の復号手順は以下の通りである．

式(12)より，

$$S = f(0) = \sum_{j=1}^{k} y_{i_j} \lambda_j(0) \ \bmod \ q, \tag{17}$$

となり，

$$C_1^S = C_1^{\sum_{j=1}^{k} y_{i_j} \lambda_j(0) \ \bmod \ q} \ \bmod \ p, \tag{18}$$

が成り立つ．$y_i = g(i)$であることから，k 個のデータ$z_j = C_1^{y_{i_j}}$を集め，次式によりC_1^Sを求めることができる．

$$C_1^S = \prod_{j=1}^{k} z_j^{\lambda_j(0)} \ \bmod \ p. \tag{19}$$

最後に，式(3)より平文Mが復号できる．この復号過程において秘密鍵$S_K = S$は復元されておらず，盗難の危険が排除される．

4. NFC

4.1. RFID と NFC

近年，Pasmo，Suica などに代表される交通系カード，運転免許証，住民基本台帳カードなどの，RFID(Radio Frequency Identification)技術を用いた非接触 IC カードが普及している．非接触 IC カードは交信距離により，密着型(2 ㎜以内)，近接型(10 ㎝以内)，近傍型(70 ㎝以内)に分類される．近接型の国際標準規格である ISO/IEC 14443 では，通信方式の違いにより，Type-A，Type-B に分類されている．また，国際標準として規格化されなかったが，Sony が開発した FeliCa[7]が前述の交通系カードとして日本国内で普及している．各近接型 IC カード規格の比較を表 1 に示す．

NFC(Near Field Communication)は，2002 年に Philips と Sony により共同開発された，ISO/IEC 14443 など既存の RFID 規格を網羅した近距離無線通信規格である[8]．2003 年に ISO/IEC 18092 として NFC IP-1，2005 年に拡張規格である NFC IP-2 が ISO/IEC 21481 として，国際標準規格に制定されている．NFC IP-1 では，ISO/IEC 1443 Tye-A，JIS X 6319-4 FeliCa の通信プロトコル規格に対応している．NFC IP-2 では，NFC IP-1 の規格に加え，ISO/IEC 1443 Type-B などの通信プロトコルに対応している．

4.2.　NFC Forum 仕様

NFC の普及などを目的に設立された NFC Forum により，必須の実装仕様として以下の 3 項目が策定されている．

(1)　NDEF(NFC Data Exchange Format)
(2)　NFC Forum Tag
(3)　NFC の動作モード

4.2.1.　NDEF

NDEF[9]は，NFC 通信のデータ交換において，デバイス間で互換性を持たせる目的で策定されたデータフォーマットであり，1 つ以上のアプリケーションが定義するペイロードを，単一のメッセージ構造にカプセル化する

表 1　各近接型 IC カード規格の比較

	ISO/IEC 1443 Type-A	ISO/IEC 1443 Type-B	JIS X 6319-4 FeliCa
通信距離	～2 ㎝	～5 ㎝	～5 ㎝
通信速度	106KB/s	106KB/s	211KB/s
特　徴	通称 MIFARE NXP と呼ばれ，世界的に普及している	日本の省庁，地方自治体が主に使っている	日本では，交通系カードとして普及している

ように設計されている．図 2 に NDEF メッセージの構造を示す．NDEF メッセージは，1 つ以上の NDEF レコードから構成される．NDEF レコードは，ペイロード長，ペイロードタイプなどのオプションが記述されたヘッダとペイロードから構成される．

4.2.2. NFC Forum Tag

現在，以下に示す Type 1 から Type 4 まで 4 種類のタグが，NFC Forum 仕様のタグとして定められている[10]．

(1) Type 1 Tag

ISO/IEC 14443 A に基づいた，読み取り・書き換え可能なタグである．96Byte のメモリが利用でき，2KByte まで拡張可能である．通信速度は，106Kbit/s である．

(2) Type 2 Tag

ISO/IEC 14443 A に基づいた，読み取り・書き換え可能なタグである．48Byte のメモリが利用でき，2KByte まで拡張可能である．通信速度は，106Kbit/s である．

(3) Type 3 Tag

FeliCa として知られている，JIS X 6319-4 に基づいており，タグは製造時に読み書き可能，または読み取り専用に事前設定が可能である．メモリサイズは可変であり，1 サービスにつき最大 1MByte まで利用できる．通信速度は 212Kbit/s または 424Kbit/s である．

(4) Type 4 Tag

ISO/IEC 14443 A/B と完全に互換性があり，タグは製造時に読み書き可能，または読み取り専用に事前設定が可能である．メモリサイズは可変であり，1 サービスにつき最大 32KByte まで利用できる．通信速度は，最大 424Kbit/s である．

4.2.3. NFC の動作モード

NFC において，通信を開始するデバイスをイニシエータと呼び，イニシエータに応答するデバイスをターゲットと呼ぶ．また，NFC デバイスは，スマートフォンに代表される NFC モバイル端末，NFC タグ，NFC リーダの 3 種類に分類される．NFC モバイル端末，NFC リーダはそれぞれ自身の電源を使用するため，アクティブデバイスと呼ばれ，NFC タグは通信相手の電力を利用するため，パッシブデバイスと呼ばれる．図 3 に示す通り，どの NFC デバイスがペアとなり通信を行うかにより，3 種類の動作モードが NFC Forum により定義されている．

(1) Read/Write モード

Read/Write モードでは，アクティブデバイス(NFC リーダ)がイニシエータとなって通信を開始し，ターゲットである NFC タグ(パッシブデバイス)に格納されたデータの読み取り，変更が可能である．

(2) Peer-to-Peer モード

Peer-to-Peer モードでは，2 つの NFC モバイル端末が双方向で接続を確立し，2 つのアクティブデバイス間で双方向の「Request-Response」モデルに基づく通信が可能である．

(3) Card Emulation モード

Card Emulation モードにおいて，NFC モバイル端末は ISO/IEC 14443 A/B および JIS X 6319-4 FeliCa に基づく標準規格と完全に互換性を確保している．ユーザが NFC リーダに，スマートフォンなどの NFC モバイル端末をかざすと，上記規格の IC カードのように動作する．

4.3. NFC のセキュリティ

4.3.1. Secure Element

非接触によるチケットの発券，料金の支払いなどのアプリケーションの実装を可能とするためには，クレジットカード番号などの個人情報を，如何にして安全に NFC モバイル端末(または，IC カード)と交換するかが問題となる．この解決策として，NFC の関連アプリケーションを，保護されたメモリ環境で実行，およびデータを保存する，SE(Secure Element)と呼ばれる仕組みが用意されて

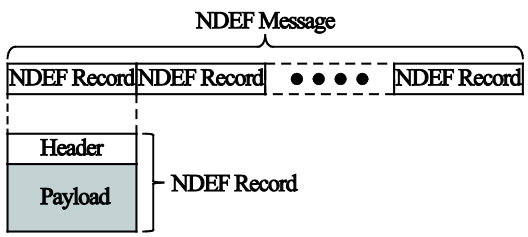

図 2 NDEF メッセージ

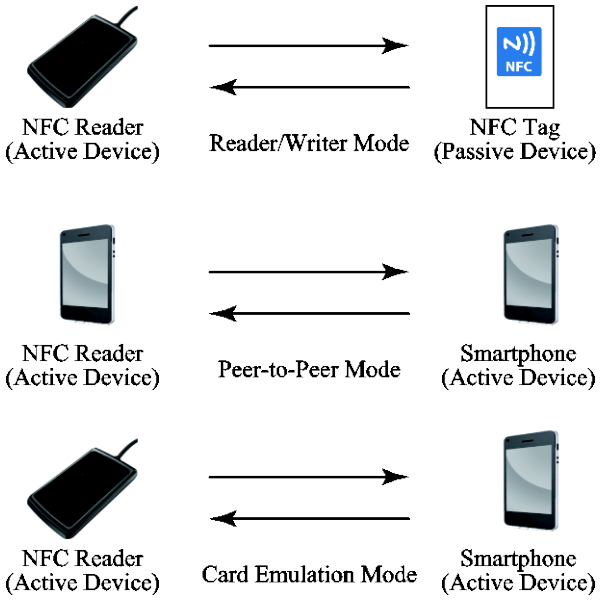

図 3 NFC の動作モード

いる．

4.3.2.　無線通信インタフェイスのセキュリティ

NFC デバイス間のデータ交換は，無線通信インタフェイスで行われるため，盗聴がセキュリティ上の問題となるのは明らかである．数㎝の近距離で行われる NFC アプリケーションによるデータ交換は，無線 LAN など他の無線方式と比較して，通信路のセキュリティ確保は優位となるが，盗聴などの脅威は完全に排除されるわけではない．

盗聴などの通信路上の脅威から NFC アプリケーションのデータを保護する唯一の解決策は，RSA に基づく Diffie-Hellman のような標準的な鍵合意プロトコル[11]を適用し，2 つの NFC デバイス間でセキュアなチャネルを確立することである[12]．

本研究では，(k,n)しきい値法で分散されたシェアを NFC デバイス間で交換する方式を検討するため，デバイス間のセキュアなチャネル確立によるデータ保護は開発対象外とした．

5.　開発システム

本研究では，以下のシナリオを想定し ElGamal 暗号しきい値復号システムの開発を行った．

(1) Alice は拡張 ElGamal 暗号の鍵ペアを作成し，公開鍵$P_K = (p, q, \alpha, y)$を公開する．

(2) 秘密鍵S_Kを(k,n)しきい値法により n 個のシェア$\{p_1, p_2, \cdots, p_n\}$に分割し、それぞれを IC カード$(N_1, N_2, \cdots, N_n)$に記録する．その後，秘密鍵$S_K$を破棄する．(1)および(2)の処理を図 4 に示す．

(3) Bob は平文Mを公開鍵P_Kで暗号化し，暗号文$C = (C_1, C_2)$を Alice へ送信する．この処理を図 5 に示す．

(4) Alice は k 個のシェア$\{p_{i_1}, p_{i_2}, \cdots, p_{i_k}\}(1 \le i_j \le n)$により，暗号文$C = (C_1, C_2)$を，式(19)および式(3)により復号し，平文$M$を得る．この処理を図 6 に示す．

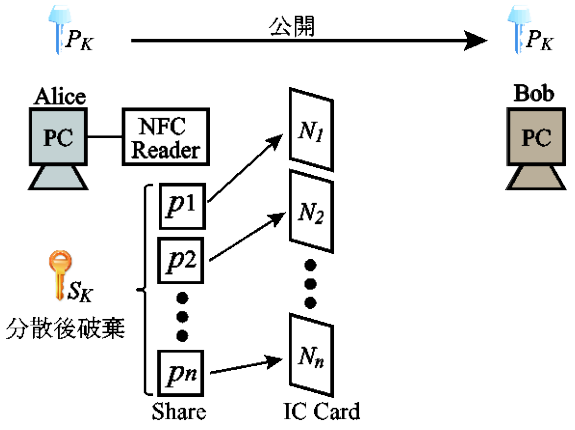

図 4　鍵ペアの作成および公開

また，本研究では，(k,n)しきい値法による分散情報を書き込む NFC デバイスとして，①IC カード，②Android 端末の 2 種類を用いた 2 システムを開発した．また，両システム共に NFC リーダデバイスとして，PaSori RC-S330[注 1](以降 PaSori と略す)を使用した．

5.1.　IC カードを用いたシステム

このシステムでは，分散情報を格納する NFC デバイスとして，PaSori と Reader/Writer モードでデータの読み書きが可能である ISO/IEC 14443 Type-A 規格の Mifare カード(以降 IC カード)を選択した．

このシステムには，①拡張 ElGamal 暗号の鍵生成(Gen)，②秘密鍵を(k,n)しきい値法により分散(Share)，③公開鍵で平文を暗号化(Enc)，④暗号文と k 個のシェアにより復号(Dec)，および⑤PaSori を用いて IC カードとデータの読み書き(Read/Write)，の機能が必要である．Alice の計算機に，Gen，Share，Dec，および Read/Write のモジュールを実装し，Bob の計算機には Enc モジュールを実装する．各モジュールの実装において，乱数列の生成には，Mersenne Twister[13]疑似乱数列生成器を用いた(初期シードに std::random_device で生成した乱数を与える)．また，Fermat テストおよび Miller テストのアルゴリズム[14]により，素数判定を行っている．

素数p，qなどのパラメータは 32bit 符号付整数型で実装した．演算時のオーバーフローを避けるため，乗算の

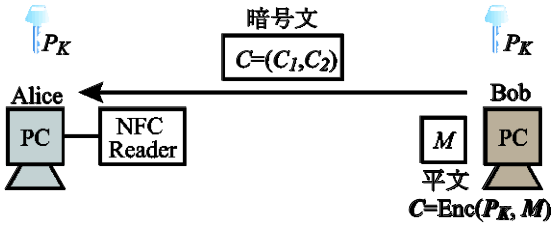

図 5　平文の暗号化および送信

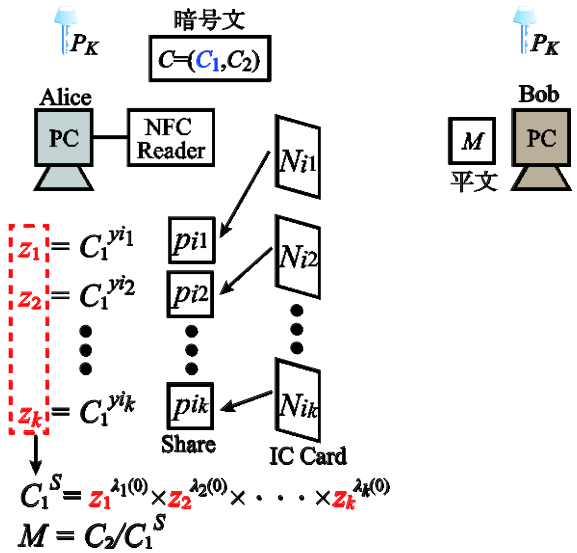

図 6　暗号文の復号

都度，mod による剰余演算を行うように実装した．しかし，素数qに$\sqrt{2^{32}-1} \fallingdotseq 65{,}536$より大きい値を設定すると演算にオーバーフローが発生し，正しい処理ができなかった. 今回の検証では, 素数qが 65,536 未満であれば, 正しい処理結果が確認できた.

5.1.1. Gen モジュール(Alice の計算機)

Gen モジュールは以下のように動作し，秘密鍵S_Kおよび公開鍵P_Kを生成する．

1: 乱数列生成器により，秘密鍵$S_K = x$を生成する．
2: 素数 q の候補 *num*1 を入力する．
3: *num*1 が素数であるか判定し，素数でない場合，*num*1 未満の素数を計算し，q に設定する．
4: 2q+1 が素数であるか判定し，素数であれば，p=2q+1 と設定する．素数でなければ，1:に戻る．
5: p，q を入力として，$\mathbb{Z}_p^*$の部分群であり，位数がqの巡回群$\langle\alpha\rangle$を生成する．
6: $y = \alpha^x \mod p$を計算し$P_K = (p, q, \alpha, y)$を公開鍵とする．

5.1.2. Share モジュール(Alice の計算機)

Share モジュールは以下のように動作し，秘密鍵S_Kを n 個のシェアに分割する．

1: k，n を入力する．ただし，$k < n$ である．
2: 式(14)，(15)，(16)を用いて，秘密鍵$S_K = x$を n 個のシェア$\{p_1, p_2, \cdots, p_n\}$に分散する．

5.1.3. Enc モジュール(Bob)の計算機)

Enc モジュールは以下のように動作し，入手済みの公開鍵P_Kから平文Mを暗号化し，暗号文$C = (C_1, C_2)$を出力する．

1: 乱数列生成器により，乱数 r を生成する．
2: 平文Mを入力する．ただし，$M < p$とする．
3: 式(10)を用いて，暗号文 $C = (C_1, C_2)$を出力する．

5.1.4. Dec モジュール(Alice の計算機)

Dec モジュールは以下のように動作し, k 個のシェアと暗号文$C = (C_1, C_2)$から平文Mを復号する．

1: $i_j\left(1 \le i_j \le n\right)$番目のシェア$p_{i_j}$を入力し，$z_j = C_1^{y_{i_j}}$を計算する．ただし，$y_{i_j} = p_{i_j}$である．これを異なるシェアに対して k 回繰り返す．
2: 式(13)，(19)を用いて，C_1^sを計算する．
3: 式(3)より，暗号文Cを復号し平文Mを得る．

5.1.5. Read/Write モジュール(Alice の計算機)

Read/Write モジュールは，PaSori の販売元である Sony が提供する SDK for NFC Starter Kit[15]を用いてプログラムの開発を行った．今回は，1 枚の IC カードに，シェア値$p_{i_j}\left(1 \le i_j \le n\right)$を 3Byte, およびシェア番号$i_j$を 1Byte の合計 4Byte のデータを書き込むこととした．

Share モジュール処理後に IC カードへの書き込み処理, および IC カードからのシェアの読み込み処理は, 以下の通り動作する．

1: felicalib_nfc_initialize 関数を呼び出して, NFC アクセスライブラリを初期化する．
2: felicalib_nfc_open 関数を呼び出して，FeliCa ポートをオープンし，使用可能状態にする．
3: pollingCard 関数を呼び出して，IC カードのポーリング処理を開始する．
4: felicalib_nfc_thru 関数を呼び出して，IC カードへコマンドを発行する(用意したデータを書き込む, またはデータを読み取る).
5: felicalib_nfc_stop_dev_access 関数を呼び出して, デバイスの使用権を開放する．
6: felicalib_nfc_stop_poll_mode 関数を呼び出して，IC カードのポーリング処理を停止する． 3:～6:の処理を，書き込み時は n 回，読み取り時は k 回繰り返す．
7: felicalib_nfc_close 関数を呼び出して，FeliCa ポートをクローズする．
8: felicalib_nfc_uninitialize 関数を呼び出して, NFC アクセスライブラリの処理を終了する．

5.2. Android 端末を用いたシステム

スマートフォンやタブレットの OS として普及している Android は Ver.2.3(API Level 9)から NFC に対応した．しかし, このバージョンでは NFC タグの読み取り機能のみが提供されており，NFC タグへの書き込みは Ver.2.3.3(API Level 10)以上から対応している．Ver.4.0(API Level 14)から上記機能に加え，タグのフィルタ機能，および Android Beam[16]と呼ばれる機能が追加された．本研究では，Ver.4.0(API Level 14)以上の NFC 搭載 Android 端末を対象に開発を行い，NFC 搭載端末である富士通製 arrows M04 および SAMSUNG 製 Galaxy S8+のスマートフォンを用いて動作検証を行った．前者が Android Ver.7.1(API Level 25)，後者が Ver.7.0(API Level 24)搭載モデルである．

IC カードを用いたシステムでは, IC カードが計算能力

を有さないため，$z_j = C_1^{y_{i_j}}$の計算処理は Alice の計算機が担った．そこで，IC カードの代わりに，計算能力を有する Android 端末を Card Emulation モードで用いると，Alice の計算機に直接秘密鍵のシェアを送ることなく，$z_j = C_1^{y_{i_j}}$の計算結果z_jを Alice の計算機へ送ることが可能となる．図 7 に Android 端末を用いた復号処理を示す．

このシステムにおいて，Alice の計算機に実装する Gen および Share モジュール，Bob の計算機に実装する Enc モジュールは，IC カードを用いるシステムから変更はない．Alice の計算機に実装する Dec モジュールの変更および Android 端末に NFC アプリケーションの開発，実装を行った．

Android 端末の NFC アプリケーションでは，①Alice の計算機からシェア$p_{i_j}(1 \le i_j \le n)$およびC_1を読み取る(Read)，②$z_j = C_1^{y_{i_j}}$を計算する(Calc)，③z_jを Alice の計算機へ送信する(Write)の 3 機能が中心となる．今回，①および②は Android Beam 機能を用いて実装した．NFC および Android Beam 機能は，android.nfc パッケージにあるクラス等を利用して開発を行った．

また，IC カードを用いたシステムと同様に，素数などのパラメータを 32bit 符号付整数型で実装したため，$q <$ 65,536の条件で検証を行い，正し処理結果が確認できた．

5.2.1.　Dec モジュール(Alice の計算機)

Dec モジュールは以下のように動作し，k 台の Android 端末から$\{z_1, z_2, \cdots, z_k\}(1 \le k \le n)$を読み取り，暗号文$C = (C_1, C_2)$から平文$M$を復号する．

1: $i_j(1 \le i_j \le n)$番目のアンドロイド端末からz_jを読み取る．これを異なる端末に対して k 回繰り返す．
2: 式(13)，(19)を用いて，C_1^Sを計算する．
3: 式(3)より，暗号文Cを復号し平文Mを得る．

5.2.2.　Read モジュール(Android 端末)

Android 端末が Android Beam で NDEF メッセージを受信すると，Intent と呼ばれる仕組みにより対象アプリケーションへ通知される．開発したアプリケーションでは，Intent クラスの getIntent メソッドを呼び出して，シェア$p_{i_j}(1 \le i_j \le n)$またはC_1が格納されている NDEF メッセージを処理する．

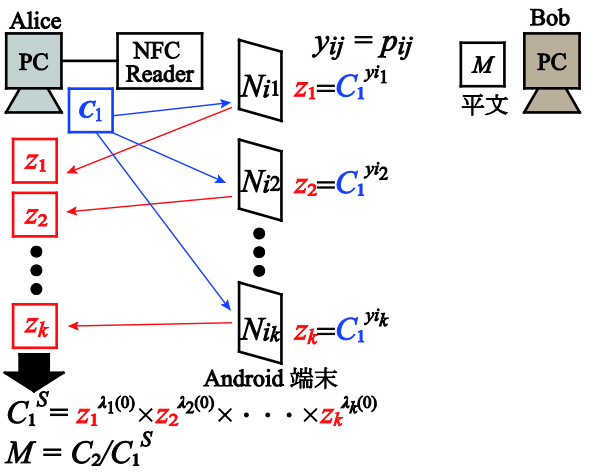

図 7　Android 端末を用いた復号処理

5.2.3.　Calc モジュール(Android 端末)

Read モジュールによりシェア$p_{i_j}(1 \le i_j \le n)$を取得後，暗号文$C = (C_1, C_2)$の復号時に，再度 Read モジュールによりC_1を読み込む．その後，Calc モジュールは$z_j = C_1^{y_{i_j}}$により，z_jを計算する．

5.2.4.　Write モジュール(Android 端末)

NdefMessage および NdefRecord クラスのメソッドを用いて，シェア$p_{i_j}(1 \le i_j \le n)$およびシェア番号i_jを格納する NDEF メッセージを作成する．その後，NfcAdapter クラスの setNdefPushMessage メソッドを呼び出すことにより，Android Beam で NDEF メッセージを送信する．

6.　まとめ

本研究では，暗号文の復号時に秘密鍵の盗難の危険性を排除した，「NFC を用いた ElGamal 暗号しきい値復号システム」として，①IC カードを用いたシステム，②Android 端末を用いたシステムの開発を行い，暗号文を正しく復号できることを確認した．①のシステムでは，IC カードが計算能力を有さないため，最終的に復号を行う計算機に秘密鍵のシェアが出現する問題があった．しかし，②のシステムでは，Android 端末内で暗号文復号の初期計算を行うことにより，秘密鍵シェアを出現させずに復号処理を行うことが可能となった．これにより，復号時に秘密鍵および，そのシェアが盗まれる危険性が排除された．このシステムは，ある組織において，n人の構成員に IC カードまたは Android 端末を用いてシェアを分散し，不特定のk人$(n > k)$が集まれば，秘密情報を復号できる，というアプリケーションへ応用が可能である．

今回開発した両システムでは，プログラムに使用した変数型の影響により，素数qの上限が 65,536 となった．多倍長演算による処理を実装することにより，より大きな桁数の素数に対応させることが今後の課題である．

NFC の Peer-to-Peer モードを利用すれば，Android 端末のみで ElGamal 暗号しきい値復号システムを実現可能である．また，Android Ver.4.1(API Level 16)以上であれば，Bluetooth によるデータ転送が Android Beam で可能である．NFC はデータ転送速度が数百 kbps であるが，Bluetooth であればデータ転送速度が数十 Mbps となり，大幅に転送時間が改善される．NFC の Peer-to-Peer モードを利用したシステムの開発も今後の課題である．

謝辞

本研究を進めるにあたりプログラム開発に協力いただいた，職業能力開発総合大学校電子情報専攻の学生諸氏に感謝いたします．また，本研究は JSPS 科研費 16K06375

の助成を受けたものです.

（原稿受付 2019/1/10，受理 2019/4/19）

参考文献

[1] 情報セキュリティインシデントに関する調査報告書, https://www.jnsa.org/result/incident/data/2017incident_survey_sokuhou_ver1.1.pdf, Accessed 19 Nov. 2018.

[2] A. Shamir: “How to share a secret”, Communications of the ACM Vol.22, pp.612-613, 1979.

[3] 黒澤馨，尾形わかは :「現代暗号の基礎数理」，コロナ社，東京，pp-119-121 (2014).

[4] T. ElGamal: “A public key cryptosystem and a signature scheme based on discrete logarithms”, IEEE transactions on information theory 31, no. 4, pp.469-472, 1985.

[5] R.L.Rivest, A. Shamir and L.M. Adleman: “A Method for Obtaining Digital Signatures and Public-Key Cryptosystems”, Communications of ACM Vol.21, pp.120-126, 1978.

[6] Y. Tsiounis and M. Yung: “On the security of ElGamal based encryption”, International Workshop on Public Key Cryptography, pp. 117-134, Springer, Berlin, Heidelberg, 1998.

[7] Felica, https://www.sony.co.jp/Products/felica/, Accessed 19 Nov. 2018.

[8] NFC Forum, https://nfc-forum.org/, Accessed 19 Nov. 2018.

[9] NFC Forum: “NFC Data Exchange Format (NDEF)”, NFC Forum, (2006).

[10] NFC Forum Issues Specifications For Four Tag Types, https://nfc-forum.org/newsroom/nfc-forum-issues-specifications-for-four-tag-types/, Accessd 18 Dec. 2018.

[11] W.Diffie and M.Hellman: “New Directions in Cryptography”, IEEE Trans. on Information Theory, IT-22 pp.472-492, 1976.

[12] E. Haselsteiner and K. Breitfuß: “Security in near field communication (NFC)”, In Workshop on RFID security, pp. 12-14. 2006.

[13] M.Matsumoto and T.Nishimura: “Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator”, ACM Trans. on Modeling and Computer Simulation, 8(1), pp.3-30, 1998.

[14] G.L.Miller: “Riemann's hypothesis and tests for primality”, Journal of computer and system sciences 13.3 1976.

[15] SDK for NFC Starter Kit, https://www.sony.co.jp/Products/felica/business/products/ICS-D010_cons.html, Accessd 30 Dec. 2018.

[16] Beam NDEF messages to other devices, https://developer.android.com/guide/topics/connectivity/nfc/nfc#p2p, Accessed 31 Dec. 2018.

注

［注 1］PaSoRi RC-S330 は 2018 年 11 月現在生産終了となっており，PaSoRi S380 が後継製品として販売されている(https://www.sony.co.jp/Products/felica/consumer/products/RC-S380.html).

*大村 光德，博士（情報科学）
職業能力開発総合大学校，能力開発院，〒187-0035 東京都小平市小川西町 2-32-1 email:oomura@uitec.ac.jp
Kotoku Omura, Faculty of Human Resources Development, Polytechnic University of Japan, 2-32-1 Ogawa-nishimachi, Kodaira-shi, Tokyo 187-0035.

*宮崎 真一郎，博士（工学）
職業能力開発総合大学校，能力開発院，〒187-0035 東京都小平市小川西町 2-32-1 email:miyazaki@uitec.ac.jp
Shinichiro Miyazaki, Faculty of Human Resources Development, Polytechnic University of Japan, 2-32-1 Ogawa-nishimachi, Kodaira-shi, Tokyo 187-0035.

*松嶋 智子，博士（工学）
職業能力開発総合大学校，能力開発院，〒187-0035 東京都小平市小川西町 2-32-1 email:tomoko@uitec.ac.jp
Tomoko K. Matsushima, Faculty of Human Resources Development, Polytechnic University of Japan, 2-32-1 Ogawa-nishimachi, Kodaira-shi, Tokyo 187-0035.

*山嵜 彰一郎，工学博士
職業能力開発総合大学校，能力開発院，〒187-0035 東京都小平市小川西町 2-32-1 email:syamasa@uitec.ac.jp
Shoichiro Yamasaki, Faculty of Human Resources Development, Polytechnic University of Japan, 2-32-1 Ogawa-nishimachi, Kodaira-shi, Tokyo 187-0035