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

2007－DPS－130 (6)
2007－CSEC－ 36 (6)
2007／3／1

送信者に認証機能を付加したブロードキャスト暗号の安全性に関する一考察

大川 直人† 土井 洋†

† 情報セキュリティ大学院大学 221–0835 神奈川県横浜市神奈川区鶴屋町 2–14–1
E-mail: †{mgs053103,doi}@iisec.ac.jp

あらまし ブロードキャスト暗号を用いると，送信者が指定した複数のユーザは復号できるが，それ以外のユーザは復号できないという機能を実現できる．2005 年に，Boneh らにより暗号文サイズなどを小さくできる方式が提案され，同年，金沢らにより送信者に認証機能を付加したブロードキャスト暗号が提案された．本論文では，送信者に認証機能を付加したブロードキャスト暗号に関する安全性の検討結果，具体的には幾つかの問題点について報告する．

Notes on the Broadcast Encryption with Sender Authentication

Naoto OHKAWA† and Hiroshi DOI†

† Institute of Information Security
2–14–1, Tsuruya-cho, Kanagawa-ku, Yokohama Kanagawa, 221–0835 Japan
E-mail: †{mgs053103,doi}@iisec.ac.jp

Abstract Broadcast Encryption allows a sender to distribute digital data securely and efficiently, though a broadcast channel to selected users. We evaluate the Broadcast Encryption with Sender Authentication, and point out some attacks for this scheme.

1. はじめに

ブロードキャスト暗号とは，Berkovits[1] や Fiat[4] らによって提案された暗号方式である．この暗号方式では，送信者が指定したユーザは暗号文を復号することができるが，それ以外のユーザは暗号文を復号することができない．また通常の暗号方式は送信者と受信者が一対一の関係であるが，ブロードキャスト暗号方式では一対多の関係となる．

2005 年に Boneh らによって，秘密鍵の更新機能を持たない受信者 (stateless receiver) に適したブロードキャスト暗号[3] が提案された（以降，BGW 方式と称す）．BGW 方式は，他のブロードキャスト暗号と比較して暗号文サイズと秘密鍵サイズが小さいという特徴を持つ．特に暗号文サイズは，ユーザ数に依存せず固定長である．

しかしながら，BGW 方式は送信者の認証機能を有しておらず，任意の者が任意のユーザに対しての暗号文を，公開情報のみを用いて生成することができる．これは，送信者と送信データの正当性が保障されないということを意味する．解決策として，単純に BGW 方式に既存の署名方式を組み合わせることが考えられる．しかしながら BGW 方式の鍵は特別な形をしているため，RSA 署名[7] や Schnorr[8] 署名と組み合わせるためには秘密鍵や公開鍵を別に用意しなければない．送信者に認証機能を付加したブロードキャスト暗号を考える場合，秘密鍵と公開鍵を別途用意することなく，ブロードキャスト暗号に組み込む形で実現できることが望ましい．2005 年に金沢らによって，送信者に認証機能を付加したブロードキャスト暗号[5] が提案された（以降，KOIO 方式と称す）．

本論文では，KOIO 方式の安全性を検討した結果を，具体的にはいくつかの問題点について報告する．

2. 準 備

2.1 ブロードキャスト暗号

ブロードキャスト暗号とは，送信者が任意に選んだユーザにのみ，安全にメッセージを送信することを目的とした暗号方式である．

ブロードキャスト暗号は，以下のようにモデル化することができる．

初期化・鍵生成フェーズ

システム管理者はセキュリティパラメータから各ユーザの

秘密鍵と公開鍵を生成する．ユーザ秘密鍵を各ユーザにそれぞれ秘密に配布し，公開情報を各ユーザが常に取得できる状態にする．

暗号化フェーズ

送信者はセッション鍵を生成し，その鍵でメッセージを暗号化する．ここで復号を許可するユーザを選択し，選択したユーザのみが各自の秘密鍵で復号できるように，セッション鍵を暗号化する．暗号化されたセッション鍵などをまとめたものはヘッダと呼ばれる．送信する暗号文は，ヘッダと暗号化メッセージで構成される．

復号フェーズ

復号を許可されたユーザは各自の秘密鍵を用いて，受信したヘッダからセッション鍵を復号する．復号して得られたセッション鍵を用いて暗号化メッセージを復号する．

2.2 ペアリング

定義 1. (双線形写像 e) [2]

p は大きな素数とし，$\mathbb{G}_1$ を位数 p の巡回加法群，$\mathbb{G}_2$ を位数 p の巡回乗法群とする． $\forall Q, R \in \mathbb{G}_1$ と $\forall a, b \in \mathbb{Z}$ が与えられた時，常に $e(aQ, bR) = e(Q, R)^{ab}$ を満たす写像 $e : \mathbb{G}_1 \times \mathbb{G}_1 \to \mathbb{G}_2$ を双線形写像 (Bilinear Map) という．ただし，$\mathbb{G}_1$ の生成元を P とするとき $e(P, P) \neq 1$ である．

次に，ペアリングに基づく問題と仮定を述べる．

定義 2. (Bilinear Diffie-Hellman 問題 (BDHP)) [2]

BDHP とは，$\mathbb{G}_1, \mathbb{G}_2, e, P \in \mathbb{G}_1$ および $aP, bP, cP (a, b, c \in \mathbb{Z})$ が与えられたとき，$e(P, P)^{abc}$ を求める問題である．

定義 3. (Bilinear Diffie-Hellman Expornent 問題 (BDHEP)) [3]

$\alpha \in \mathbb{Z}_q, n \in \mathbb{N}, T \in \mathbb{G}_1$ とする．BDHEP とは，任意のベクトル $(T, P, \alpha P, \alpha^2 P, \cdots, \alpha^n P, \alpha^{n+2} P, \cdots, \alpha^{2n} P)$ が与えられたとき，$e(P, T)^{\alpha^{n+1}}$ を求める問題である(注1)．

定義 4. (拡張 Bilinear Diffie-Hellman Expornent 問題 (拡張 BDHEP)) [5]

$\alpha \in \mathbb{Z}_q, n \in \mathbb{N}, T \in \mathbb{G}_1$ とする．拡張 BDHEP とは，任意のベクトル $(T, P, \alpha^{-(n-1)} P, \cdots, \alpha^{-1} P, \alpha P, \alpha^2 P, \cdots, \alpha^n P, \alpha^{n+2} P, \cdots, \alpha^{2n} P)$ が与えられたとき，$e(P, T)^{\alpha^{n+1}}$ を求める問題である (注 1)．

仮定 1. BDHP，BDHEP，拡張 BDEHP 問題を解く確率的多項式時間アルゴリズムは存在しない．

3. BGW 方式

BGW 方式は，2005 年に Boneh らによって提案された，受信者の結託に強い公開鍵型ブロードキャスト暗号である．それまでのブロードキャスト暗号と比べて秘密鍵サイズとヘッダサイズが小さいという特徴を持ち，特にヘッダサイズは固定長である．安全性は，仮定 1 の BDHP の困難性に基づく．

3.1 節に，具体的なプロトコルを示す．なお，セットアップと鍵生成は鍵生成センターが行うものとする．

3.1 プロトコル

セットアップ・鍵生成

(1) 乱数 $\alpha \in \mathbb{Z}_p$ を生成する．

(2) $P_i = \alpha^i P$ を計算する．$(i = 1, 2, \cdots, n, n+2, \cdots, 2n)$

(3) 乱数 $\gamma \in \mathbb{Z}_q$ を生成し，$Q = \gamma P$ を計算する．

(4) ユーザ $i \in \{1, \cdots, n\}$ の秘密鍵 D_i を計算し，秘密に配布する．なお，$D_i = \alpha^i Q$ である．

(5) $g = e(P, P_{n+1})$ とする．公開鍵 $PK = (P, Q, P_1, \cdots, P_n, P_{n+2}, \cdots, P_{2n}, g)$ を出力する．

暗　号　化

(1) 乱数 $t \in \mathbb{Z}_p$ を生成する．

(2) セッション鍵 $K = g^t$ を計算する．

(3) 復号を許可するユーザ集合 S を選び，$C_0 = t(Q + \sum_{j \in S} P_{n+1-j})$，
$C_1 = tP$ を計算する．
ヘッダ $Hdr = (C_0, C_1)$ とする．

(4) メッセージ m を鍵 K で暗号化し，暗号化メッセージ C_m を生成する．なお，ここでは任意の共通鍵暗号方式を利用できる．

(5) Hdr と C_m を出力する．

復　　号

(1) ユーザ $i \in S$ は，秘密鍵 D_i を用いて
$$K = \frac{e(P_i, C_0)}{e(C_1, D_i + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1+i-j})}$$
を計算をし，復号鍵 K を導出する．

(2) 鍵 K で C_m を復号し，メッセージ m を出力する．

BGW 方式は，暗号文を生成する際に一切の秘密情報を必要としていない．つまり，誰でもが任意の復号を許可するユーザに対して暗号文を送れるが，復号を許可されたユーザは送信者が誰なのかを識別することができない．安全な通信を行うためには，公開鍵型ブロードキャスト暗号方式に，効率よく送信者の認証機能を付けることができれば望ましい．

(注1)：与えられたベクトルに $\alpha^{n+1} P$ が抜けていることに注意されたい．

4. KOIO 方式

KOIO 方式は，2005 年に金沢らによって提案された，BGW 方式をベースに送信者認証を付け加えたブロードキャスト暗号である．安全性は，拡張 BDHEP を解くことが困難であることに基づいている．

4.1 プロトコル

セットアップと鍵生成は，鍵生成センターが行うものとする．

セットアップ・鍵生成

（1） 乱数 $\alpha \in \mathbb{Z}_p$ を生成する．

（2） $P_i = \alpha^i P$ を計算する．

$(i = -(n-1), \cdots, -1, 1, \cdots, n, n+2, \cdots, 2n)$

（3） 乱数 $\gamma \in \mathbb{Z}_p$ を生成し，$Q = \gamma P$ を計算する．

（4） ユーザ $i \in \{1, \cdots, n\}$ の秘密鍵 D_i を計算し，秘密に配布する．なお，$D_i = \alpha^i Q$ である．

（5） $g = e(P, P_{n+1})$ とする．公開鍵 $PK = (P, Q, P_{-(n-1)}, \cdots, P_{-1}, P_1, \cdots, P_n, P_{n+2}, \cdots, P_{2n}, g)$ を出力する．

暗 号 化

送信者 a（秘密鍵 D_a を所持）は，以下のようにメッセージ m を暗号化する．

（1） 乱数 $t \in \mathbb{Z}_p$ を生成する．

（2） セッション鍵 $K = g^t$ を計算する．

（3） メッセージ m を鍵 K で暗号化し，暗号化メッセージ C_m を生成する．なお，ここでは任意の共通鍵暗号方式を利用できる．

（4） 乱数 $r \in \mathbb{Z}_p$ を選び，$s = \mathcal{H}(g^r)$ を計算する．

（5） $y = r - st \bmod p$ を生成する．

（6） 復号を許可するユーザ集合 S を選び，
$C_0 = t(D_a + \sum_{j \in S} P_{n+1+a-j})$，
$C_1 = tP$ を計算する．

（7） ヘッダ $Hdr = (C_0, C_1, s, y)$ とする．

（8） Hdr と C_m を出力する．

復号・送信者認証

ユーザ $i \in S$ は，秘密鍵 D_i を用いて以下の計算をし，暗号文の復号と送信者の認証を行う．

（1） $K = \dfrac{e(P_{i-a}, C_0)}{e(C_1, D_i + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1+i-j})}$
を計算し，復号鍵 K を導出する．

（2） $s = \mathcal{H}(g^y K^s)$ が成り立つか検証する．成立しなければ，暗号文を破棄する．

（3） 上の検証式が成立するならば，鍵 K で C_m を復号し，メッセージ m を出力する．

ここで，復号および送信者認証が正しく行われることを示す．まず，鍵 K を正しく復号できていることを示す．

$$
\begin{aligned}
&\frac{e(P_{i-a}, C_0)}{e(C_1, D_i + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1+i-j})} \\
&= \frac{e(P_{i-a}, t(D_a + \sum_{j \in S} P_{n+1+a-j}))}{e(tP, D_i + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1+i-j})} \\
&= \frac{e(tP_i, (Q + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1-j}) + P_{n+1-i})}{e(tP_i, Q + \sum_{\substack{j \in S \\ j \neq i}} P_{n+1-j})} \\
&= e(tP_i, P_{n+1-i}) \\
&= e(P, P_{n+1})^t \\
&= g^t
\end{aligned}
$$

次に，検証式が成り立つことを示す．

$$
\begin{aligned}
\mathcal{H}(g^y K^s) &= \mathcal{H}(g^{r-st} g^{ts}) \\
&= \mathcal{H}(g^r) \\
&= s
\end{aligned}
$$

以上のことから，復号および送信者認証が正しく行われたことがわかる．

5. 問 題 点

KOIO 方式の安全性に関して検証し，

（1） 復号鍵 K の流用による暗号文の作成が可能である，

（2） 任意の送信者になりすまし，一人のユーザ $i \in S$ が受理する暗号文の作成が可能である，

（3） 任意の送信者になりすまし，復号を許可するユーザ全員が受理する暗号文の作成可能である，

という三つの問題点があることがわかったので報告する．このため KOIO 方式は，送信者の認証と送信データの正当性確認が達成されていないこととなる．

5.1 復号鍵 K の流用

攻撃者 $A \in S$ は，以前に送信者 a が送信したヘッダ $Hdr = (C_0, C_1, s, y)$ を用いて，メッセージ m を復号できたとする．すると Hdr を用いることで，別のメッセージ m' に対して，S を復号を許可するユーザ集合とする暗号文を生成

できる．a が送信した Hdr をそのまま流用するため，攻撃者 A は復号を許可するユーザ集合 S を変更することはできない．この暗号文は，送信者 a が生成したものとして復号・受理される．

以下に暗号化と復号・送信者認証を示す．

暗　号　化

攻撃者 $A \in S$ は，a から既に送られているヘッダ $Hdr = (C_0, C_1, s, y)$ を用いて以下のようにメッセージ m' を暗号化する．

（1） セッション鍵は，a から既に送られている K をそのまま使用する．

（2） メッセージ m' を鍵 K で暗号化し，暗号化メッセージ $C_{m'}$ を生成する．

（3） $Hdr = (C_0, C_1, s, y)$ と $C_{m'}$ を出力する．このときの Hdr は a から送られたものをそのまま利用する．

復号・送信者認証

復号は 4.1 節で示したものと同様である．鍵 K で $C_{m'}$ を復号し，メッセージ m' を出力する．検証式も同様である．

以上のように，セッション鍵 K を導出できるユーザ $i \in S$ は，セッション鍵 K を流用することで a になりすまして暗号文を生成することが可能となる．

5.2　一人のユーザを標的とする攻撃

任意の攻撃者 A は，一人のユーザ i_T が受理する暗号文を生成することができる．i_T は，A の生成した暗号文に対し復号・送信者検証を行うことにより，a が復号許可ユーザとして i_T を選んで生成した暗号文として受理する．

以下に暗号化と復号・送信者認証を示す．

暗　号　化

攻撃者 A は，攻撃対象者 i_T を選び，任意の集合 $S'(i_T \in S')$ を選ぶ．以下のようにメッセージ m' を暗号化する．

（1） 乱数 $t', k \in \mathbb{Z}_p$ を生成する．

（2） セッション鍵 $K' = g^k$ を計算する．

（3） メッセージ m' を鍵 K' で暗号化し，暗号化メッセージ $C_{m'}$ を生成する．

（4） 乱数 $r' \in \mathbb{Z}_p$ を選び，$s' = \mathcal{H}(g^{r'})$ を計算する．

（5） $y' = r' - s'k \bmod p$ を計算する．

（6） $C_0' = t'(Q + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1-j}) + k(P_{n+1+a-i_T})$，$C_1' = t'P_{-a}$ を計算する．

（7） ヘッダ $Hdr = (C_0', C_1', s', y')$ とする．

（8） Hdr と $C_{m'}$ を出力する．

復号・送信者検証

ユーザ $i_T \in S'$ は，4.1 節で示したものと同様に秘密鍵 D_{i_T} を用いて，暗号文の復号と送信者の認証を行い，受理する．

このとき，復号および送信者認証が正しく行われることを示す．まず，鍵 K' を正しく復号できていることを示す．

$$
\begin{aligned}
&\frac{e(P_{i_T-a}, C_0')}{e(C_1', D_{i_T} + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1+i_T-j})} \\
&= \frac{e(P_{i_T-a}, t'(Q + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1-j}) + k(P_{n+1+a-i_T}))}{e(t'P_{-a}, D_{i_T} + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1+i_T-j})} \\
&= \frac{e(P_{i_T-a}, t'(Q + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1-j}) + k(P_{n+1+a-i_T}))}{e(P_{i_T-a}, t'(Q + \sum_{\substack{j \in S' \\ j \neq i_T}} P_{n+1-j}))} \\
&= e(P_{i_T-a}, kP_{n+1+a-i_T}) \\
&= e(P, P_{n+1})^k \\
&= g^k
\end{aligned}
$$

次に，検証が成り立つことを示す．

$$
\begin{aligned}
\mathcal{H}(g^{y'}K^{s'}) &= \mathcal{H}(g^{r'-s'k}g^{s'k}) \\
&= \mathcal{H}(g^{r'}) \\
&= s'
\end{aligned}
$$

以上のように，攻撃者 A は，一人ではあるが任意の攻撃対象者 i_T に対して，a になりすました暗号文を生成することが可能である．このとき A は，i_T，a，S' を任意に選ぶことができる．ただし，$j \in S' \setminus i_T$ は暗号文を棄却できる．

本節で示した一人のユーザを標的とする攻撃より更に強い攻撃を 5.3 節に示す．

5.3　復号許可ユーザ全員に対する攻撃

任意の攻撃者 A は，任意の復号を許可するユーザ集合 S' に含まれる全ユーザが受理する暗号文を生成することができる．この暗号文は，送信者 a が生成したものとして復号・受理される．

以下に，暗号化と復号・送信者認証を示す．

暗　号　化

（1） 乱数 $r', y' \in \mathbb{Z}_p$ を選ぶ．

（2） $s' = \mathcal{H}(g^{y'}e(P, P_{n+1-a})^{r'})$ を計算する．

（3） セッション鍵 $K' = e(P, P_{n+1-a})^{s'^{-1}r'}$ を計算する．

（4） メッセージ m' を鍵 K' で暗号化し，暗号化メッセージ $C_{m'}$ を生成する．なお，ここでは任意の共通鍵暗号方式を利用できる．

(5) $C'_0 = s'^{-1}r'(Q + \sum_{j\in S'} P_{n+1-j})$, $C'_1 = s'^{-1}r'P_{-a}$ を計算する.

(6) ヘッダ $Hdr = (C'_0, C'_1, s', y')$ とする.

(7) Hdr と $C_{m'}$ を出力する.

復　　号

ユーザ $i \in S'$ は，4.1 節で示したものと同様に秘密鍵 D_i を用いて，暗号文の復号と送信者の認証を行う.

このとき，復号および送信者認証が正しく行われることを示す．まず，鍵 K を正しく復号できていることを示す.

$$
\begin{aligned}
&\frac{e(P_{i-a}, C'_0)}{e(C'_1, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j})} \\
&= \frac{e(P_{i-a}, s'^{-1}r'(Q + \sum_{j\in S'} P_{n+1-j}))}{e(s'^{-1}r'P_{-a}, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j})} \\
&= \frac{e(s'^{-1}r'P_{i-a}, Q + \sum_{j\in S'} P_{n+1-j})}{e(s'^{-1}r'P_{-a}, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j})} \\
&= \frac{e(\alpha^i s'^{-1}r'P_{-a}, Q + \sum_{j\in S'} P_{n+1-j})}{e(s'^{-1}r'P_{-a}, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j})} \\
&= \frac{e(s'^{-1}r'P_{-a}, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j} + P_{n+1})}{e(s'^{-1}r'P_{-a}, D_i + \sum_{\substack{j\in S' \\ j\neq i}} P_{n+1+i-j})} \\
&= e(s'^{-1}r'P_{-a}, P_{n+1}) \\
&= e(P_{-a}, P_{n+1})^{s'^{-1}r'} \\
&= K'
\end{aligned}
$$

次に，検証が成り立つことを示す.

$$
\begin{aligned}
\mathcal{H}(y'^{y'}K'^{s'}) &= \mathcal{H}(y'^{y'}(e(P, P_{n+1-a})^{s'^{-1}r'})^{s'}) \\
&= \mathcal{H}(y'^{y'}e(P, P_{n+1-a})^{r'}) \\
&= s'
\end{aligned}
$$

以上のように，攻撃者 A は任意の復号許可集合 S' に対して，a になりすましての暗号文を生成することが可能である．このとき A は，送信者 a，復号を許可するユーザ集合 S を任意に選ぶことができる.

6. ま と め

本論文では，送信者に認証機能を付加したブロードキャスト暗号に関する問題点を幾つか報告した．これらの問題点は，暗号文を生成する際に C_m と $K = g^t$ を用い，$s = \mathcal{H}(C_m, g^t, g^r)$ と s を計算し，また検証処理を若干増やすことで解決できると予想している．現在，フォーマルな安全性証明が可能であるか検討中である.

謝　　辞

本研究を進めるにあたり，筑波大学の岡本健講師，金沢史明氏から多くの有益な助言をいただきました．この場を借りて感謝を申し上げます.

参 考 文 献

1) S.Berkovits, "How to Broadcst a Seacret", Proc. of Asiacrypt'02, LNCS.vol.543, Springer-Verlag.pp-535-541.(1991)
2) D.Boneh, M.Franklin, "Identity-based encryption from the Weil pairing", Proc. of Crypt'01, LNCS,vol.2139,Springer-Verlag,pp.213-229.(2001)
3) D.Boneh, C.Gentry, B.Waters, "Collusion Resistant Broadcast Encryption with Short Ciphertexts and Private Keys", Proc. of Crypto'05, LNCS,vol.3624,Springer-Verlag,pp.258-275.(2005)
4) A.Fiat, M.Naor, "Broadcast Encryption", Proc. of Crypto'93, LNCS.vol.773,Springer-Verlag.pp.480-491.(1994)
5) 金沢史明, 岡本健, 猪俣敦夫, 岡本栄司, "送信者に認証機能を付加したブロードキャスト暗号", CSS'05 予稿集, (2005)
6) 金沢史明, 岡本健, 猪俣敦夫, 岡本栄司, "送信者に認証機能を付加したブロードキャスト暗号とその応用", 情報処理学会論文誌.Nov.2006.vol.47,No.11,pp.2992-3003.(2006)
7) R.Rivest, A.Shamir, L.Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, vol.21, no.2, pp.120-126(1978)
8) C.P.Schnorr, "Efficient Identification and Signatures for Smart cards", Proc. of Crypto'89, LNCS,vol435, pp.239-252, (1990)