コンピュータセキュリティ 11−14
（２０００．９．８）

鍵管理局の権限を分散した鍵寄託プロトコルの提案

崔 浩哲　　堀田 信司　　菊池 浩明　　中西 祥八郎

東海大学大学院工学研究科電気工学専攻
〒259-1292 神奈川県平塚市北金目 1117
Tel:0463-58-1211(内線 4164)
Fax:0463-59-4014
E-mail:{coo,hotta,kikn}@ep.u-tokai.ac.jp

あらまし: 現在，インターネットでは，クライアント同士で秘密通信を行なう際に用いるセッション鍵を寄託し，テロリストなどの捜査や，鍵の紛失に対応させたい要求がある．これに対して，宮崎らは Yaksha[Gan 95] をベースに，安全な鍵寄託システムを実現する試みを提案している [MISS 97]. 本稿では，鍵生成権限の分散と，捜査情報の漏洩を防止し，さらなるユーザのプライバシを保証するプロトコルを提案する．

Key Escrow Protocol with Distributed Power of Key Generation Authority

Kousetsu Sai　　Shinji Hotta　　Hiroaki Kikuchi
Shohachiro Nakanishi

Faculty of Engineering Tokai university
1117 Kitakaname, Hiratsuka, Kanagawa, 259-1292 Japan
Tel:0463-58-1211(Ext. 4164)
Fax:0463-59-4014
E-mail:{coo,hotta,kikn}@ep.u-tokai.ac.jp

Abstract: The importance of key escrow is increasing as many secret communications take placd over the Internet. It's required to provide an investigation of terrorists and to prepare for key lost. Miyazaki et. al[MISS 97] proposed a secure key escrow protocol based on the Yaksha[Gan 95]. In their protocol, the power of a key genarating authority is too strong and thus the secret key shared by client might be revealed through the KGA. In this paper, we propose a modified key escrow protocol which improves the privacy of shared key.

1 はじめに

現在，インターネットでは，セッション鍵を用いて通信路を暗号化し，第3者に情報を漏らさないことが可能である．しかし，秘密通信路はテロリストや犯罪に不正利用される恐れがある．そこで，鍵を寄託し，秘密通信路をテロリストの捜査や鍵の紛失に対応させる要求が生じている．しかし，鍵を寄託することは，不正を行っていないユーザのプライバシを脅かす大きな問題ともなりうる．

これに対して，暗号プロトコルを用いた様々な鍵寄託システムを実現する試みがされている [MISS 97][PI 00]. 宮崎らの方式 [MISS 97] は，Kerberos[NT 94] に公開鍵認証を導入した Yaksha[Gan 95] をベースとしたシステムで，各エンティティの認証を ElGamal 協力署名を用いて行い，なりすましや認証局の不正を防いでいた．また，セッション鍵とクライアント情報の管理を分離することでエンティティの権限の分散を行い，ユーザのプライバシ強化を図っていた．しかし，ユーザの最低限のプライバシを保証しつつ，有事に対応するためには，以下のような問題があった．

- 問題 (1)
  鍵生成機関がユーザの秘密鍵を知りうる可能性が残る．
- 問題 (2)
  捜査機関は認証機関に捜査対象者の ID を要求するため，認証機関に捜査情報が漏れ，ユーザの名誉を毀損する恐れがある．

そこで，本稿では，以上問題をすべて解決し，クライアントのプライバシを強化するプロトコルを提案する．

2 提案プロトコル

本節では，提案システムの基本となる ElGamal ブラインド復号と，ElGamal 多重暗号を説明し，その後，鍵共有と鍵寄託のプロトコルを説明する．

2.1 ElGamal ブラインド復号

「Alice は，Bob が自分の公開鍵で暗号化した文書を持っていて，Bob に復号してもらいたい．しかし，そのまま復号を依頼すると，暗号文から原文を求められてしまう.」という状況を考える．このような状況を解決するプロトコルにブラインド復号 [YS 96] がある．ここで，Mは平文，x_Bは Bob の秘密鍵，y_B は Bob の公開鍵で $y_B = g^{x_B}$，αと rは乱数である．計算は全て法 p のもとで行われる．

Step 0 Alice は以下の暗号文を持っている．

$$\begin{cases} c_1 = M \cdot y_B^r \\ c_2 = g^r \end{cases}$$

Step 1 Alice は c_2を乱数α，$c_2' = c_2 \cdot g^\alpha$の様にブラインドして，Bob に送る．

Step 2 Bob は c_2'を秘密鍵 x_Bで $c_2'' = c_2'^{x_B}$として，Alice に送る．

Step 3 Alice は，$c_2''' = c_2''/y_B^\alpha$ の様に Bob の公開鍵でアンブラインドをし，$m = c_1/c_2'''$ として M を受け取る．

2.2 ElGamal 多重暗号

本節では，ElGamal 多重暗号を示し，これが復号順序を換えても正しく復号可能な暗号系であることを示す．

暗号化処理

Step 1 ユーザ 0 はユーザ 1 の公開鍵 y_1と乱数 r_1 で以下のように暗号化する．

$$\begin{cases} c_0 = M \cdot y_1^{r_1} \\ c_1 = g^{r_1} \end{cases}$$

Step 2 ユーザ 0 はユーザ 2 の公開鍵 y_2と r_2で多重暗号化する．

$$\begin{cases} c_{0_{\{1,2\}}} = M \cdot y_1^{r_1} \cdot y_2^{r_2} \\ c_1 = g^{r_1} \\ c_2 = g^{r_2} \end{cases}$$

Step 3 すなわち多重暗号の一般形は以下のようになる．

$$\begin{cases} c_{0_\Lambda} = M \cdot y_1^{r_1} \cdot y_2^{r_2} \cdot \ldots \cdot y_n^{r_n} \\ c_1 = g^{r_1} \\ \vdots \\ c_n = g^{r_n} \end{cases}$$

ここで，$\Lambda, = \{1, \ldots, n\}$ である．

復号化処理

Step 1 ユーザ0は，ユーザiに部分復号を要求するため，c_iをユーザiに送る．

Step 2 ユーザiは，秘密鍵x_iで$c_i' = c_i^{x_i}$とし，ユーザ0に返信する．

Step 3 ユーザ0は，以下のようにして，全ての暗号化を解き，Mを得る．

$$M = \frac{c_{0_\Lambda}}{\prod_{i\in\Lambda} c_i'^{x_i}}$$

2.3 提案プロトコル記号定義

表記記号　定義

$\mathcal{A}$, $\mathcal{B}$	:通信を行うクライアント．
$\mathcal{S}$	:認証機関．
$\mathcal{G}$	:鍵生成機関．
$\mathcal{T}$	:捜査機関．
ID_i	:iに一意に定められた識別番号．
k_j	:jが生成するシェア鍵．
K_{ij}	:i,jが通信に用いるセッション鍵．
p	:大きな素数．
g	:乗法群Z_p^*で大きな位数を持つ元．
x_j	:jの秘密鍵．
y_j	:jの公開鍵．$y_j = g^{x_j}$．
$E_j()$	:公開鍵y_jによる暗号化処理．
$D_j()$	:秘密鍵x_jよる復号化処理．
r,u,t,α,γ	:乱数．

2.3.1 概要

本システムは，鍵生成機関$\mathcal{G}$と認証機関$\mathcal{S}$そしてクライアント$\mathcal{A}$,$\mathcal{B}$の3つのエンティティから構成にされている．クライアントは，予め生成され認証機関が管理を行っている鍵k_S,k_Gを利用する．これにより，不正なユーザによる偽の鍵の寄託を防止している．しかし，このままでは鍵生成機関がクライアントの鍵を漏らす可能性があり，クライアントは安心して利用できない．そこで，本システムは，鍵の生成を 鍵生成機関と認証機関が協力して行い，暗号化された鍵を認証機関が集中管理する．クライアントは認証機関によって認証された後，鍵生成機関にブラインド復号してもらう．

2.3.2 鍵共有プロトコル

Step 1 $\mathcal{G}$はm個のシェア鍵$k_{G_i}(i = 1,...,m)$を生成し，自分の公開鍵y_Gで

$$E_G(k_{G_i}) = \begin{cases} c_{1_i} = y_G^{r_i} \cdot k_{G_i} \\ c_{2_i} = g^{r_i} \end{cases}$$

と暗号化後，$\mathcal{S}$に送信する．また，計算は全て法pの上で行われる．

Step 2 $\mathcal{A}$は$\mathcal{B}$と通信する際，$\mathcal{S}$にID_Bを送り，セッション鍵を請求する．

Step 3 $\mathcal{S}$は，$\mathcal{G}$から受信したm個のシェア鍵からランダムに(c_{1_i},c_{2_i})を選び，自身のランダムに選んだシェア鍵k_{S_i}を掛け，$\mathcal{A}$の公開鍵にて

$$\begin{cases} c'_{1_i} = c_{1_i} \cdot k_{S_i} \cdot y_A^{u_i} \\ c_{2_i} = g^{r_i} \\ c_{3_i} = g^{u_i} \end{cases}$$

と多重暗号化し，$\mathcal{A}$に送信する．同様に$\mathcal{B}$についても行う．また，この時，鍵供出のために

$$L_i = (ID_A, ID_B, V_{AB}, E_S(k_{S_i}), E_G(k_{G_i}))$$

を保存しておく．ここで，V_{AB}はセッション開始時刻を保証するタイムスタンプ，$E_S(k_{S_i})$は

$$E_S(k_{S_i}) = \begin{cases} b_{1_i} = y_S^{t_i} \cdot k_{S_i} \\ b_{2_i} = g^{t_i}, \end{cases}$$

$E_G(k_{G_i})$はc_{1_i},c_{2_i}である．

Step 4 $\mathcal{A}$はc_{2_i}をg^αで$c'_{2_i} = c_{2_i} \cdot g^\alpha$とブラインドし，$\mathcal{G}$に送信する．

Step 5 $\mathcal{G}$はブラインドされたまま復号化処理$c''_{2_i} = c'^{x_G}_{2_i}$を行い，$\mathcal{A}$に送信する．

Step 6 $\mathcal{A}$は$\mathcal{G}$の公開鍵y_Gを用いて$c'''_{i_2} = c''_{i_2} \cdot y_G^\alpha$としてアンブラインド後，以下のように復号化して，セッション鍵

$$K_{AB} = \frac{c'_i}{c'''_{i_2} \cdot c_{3_i}^{x_A}} = k_{G_i} \cdot k_{S_i}$$

を得る．

2.3.3 検討

c_{2_i}をg^{α}でブラインドすることで，$\mathcal{G}$に対して，誰と誰が秘密通信しているかを秘密にしている．そして$\mathcal{A}$の公開鍵で暗号化することにより，第3者による盗聴を防いだ．宮崎方式との違いは，セッション鍵の生成を，分散した点である．鍵生成機関は，宮崎方式と同様に鍵を生成する．しかし，その鍵をシェア鍵とし，認証機関が生成したシェア鍵との積をセッション鍵とした．これにより，鍵生成の権限を$\mathcal{G}$の一点集中から分散させた．

2.4 鍵供出

2.4.1 概要

エンティティは捜査機関$\mathcal{T}$と認証機関$\mathcal{S}$の2つである．捜査対象(ここでは$\mathcal{A}$と$\mathcal{B}$)の漏洩は，特に大きなプライバシの侵害であると考え，認証機関に捜査対象を報せずに，捜査対象の鍵を入手するシステムを構築する．認証機関は，暗号化された鍵とIDの組をエントリーL_iとして$KeyList$を作り，有事の時，捜査機関へ提出する．捜査機関は，$KeyList$から必要な捜査対象の暗号化された鍵だけを取りだし，ブラインド復号を要求する．

2.4.2 プロトコル

Step 1 まず，$\mathcal{T}$は$\mathcal{S}$に要求文を送る．

Step 2 $\mathcal{S}$は，要求を認証し，

$$KeyList = (L_1, ..., L_n)$$

を$\mathcal{T}$に送る．

Step 3 $\mathcal{T}$は$KeyList$の中から，捜査対象者の情報を探し，g^{γ}を用いて，$b'_{2_i} = b_{2_i} \cdot g^{\gamma}$とブラインドし，$\mathcal{S}$に送る．

Step 4 $\mathcal{S}$は受信した情報をブラインドされたまま$b''_{2_i} = b'^{x_S}_{2_i}$と復元を行い，$\mathcal{T}$に返信する．

Step 5 $\mathcal{T}$は，$b'''_{2_i} = b''_{2_i} \cdot y_G^{-\tau}$とブラインドを外し，$k_{S_i} = b_{1_i}/b'''_{2_i}$とし，捜査対象者をSに知られること無く，Sのシェア鍵k_Sを得る．

Step 6 Step 3〜5と同様に$\mathcal{G}$と通信することで，シェア鍵k_Gを得て、セッション鍵Kを求め，捜査を開始する．

2.4.3 検討

捜査機関は受信した$KeyList$から捜査対象者のエントリーL_iを取りだし，認証機関と鍵生成機関にブラインド復号してもらうことにより，認証機関に対する捜査対象者情報の漏洩を防ぎ，プライバシの強化を図る．また，捜査機関の無制限な盗聴は，認証機関が鍵のブラインド復号の回数を限定することで防げると考える．欠点は，すべての鍵を渡す必要があるため，配送にかかるコストである．

3 終わりに

本稿では，クライアントのプライバシに重点をおき，クライアントが安心して利用できるための 最低限のクライアントのプライバシを保証しつつ，有事に対応できる鍵寄託プロトコルを提案した．今回は，クライアントのプライバシを守ることが最重要課題だったため，計算量，通信量 に対する考察を行わなかった．今後の課題としては，$keyList$の通信コストを削減するための考察や，各エンティティの不正を検出するようなプロトコルの構築等を行っていきたい．

4 謝辞

論文をまとめるにあたり，有益な意見を頂いた(株)高度移動通信セキュリティ技術研究所の井上 徹氏，朴 美娘氏，ならびに(株)東芝 SI技術開発センターの宮崎 信悟氏に感謝する．

参考文献

[MISS 97] 宮崎 他，“ElGamal 型協力署名の構成と鍵管理・鍵寄託システムへの応用”，情報処理学会論文誌, Vol.38, No10, pp.2074-2082, 1997.

[Gan 95] Ganesan, R. “Yaksha:Augmenting Kerberos with Public Key Cryptographt”, Proc. ISOC Symp. on Network and Distributed System Security, pp.132-143, 1995.

[PI 00] 朴 美娘，井上 徹，“グループ通信の鍵供託方式に関する一考察”，情報処理

学会研究報告-情処研報 (9), Vol.2000, No.36, pp.27-33, 2000.

[YS 96] 山根義則, 櫻井幸一, "無制限な盗聴を防ぐ鍵寄託方式", 暗号と情報セキュリティシンポジウム予稿集, pp.905-910, 1999.

[Cha 82] Devid L. Chaum, "Blind signature for untraceable payments", Advances in Cryptology Proceedings of CRYPT'82, pp.199-204, 1982.

[CPS 94] J. L. Camenisch, J. M. Piceteau and M. A. Stadler, "Blind signatuires Based on the Discrete Logarithm Problem" Advanses in Cryptology Proceedings of EUROCRYPT'94, 1994.

[ElGa 85] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms", IEEE Trans. on IT, 31, pp.644-654, 1985.

[NT 94] Neuman, B.C. and Ts'o, T. "Kerberos:An Authentication Service for Computer Networks", IEEE Communications, pp.31-38, 1994.