Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

効率的な耐故障性を持つグループ鍵共有方式

波多野　哲也[†1]　　宮地　充子[†1]

グループ鍵共有 (GKE) は n プレーヤーが安全でないネットワーク上で鍵共有できる．しかし, 任意のプレーヤーの故障やバッテリーの不足, ネットワーク障害などでプロトコル実行に失敗すると鍵共有できない．良く知られている GKE として Buurmester Desmedt GKE(BD) がある．以前に BD に焦点を当てた論文では, 通信量が $O(n)$ 必要な BDI であるためプレーヤーの計算資源やバッテリーなどの環境が同じである必要がある．そのため, 耐故障性を実現するためには $O(n)$ より大きな通信量が必要である．我々は CH-GKE と呼ばれる効率的な耐故障性を実現した GKE を提案した．耐故障性を実現するために, 通信量が $O(\log_2 n)$ 必要な BDII を拡張した．BDII では各々のプレーヤーの計算資源やバッテリーなどの環境が同じである必要は無い．この論文では, 既存の耐故障性を実現した GKE と提案方式である CH-GKE の比較を行い, 各プレーヤーの故障率を変えたときの最適な GKE を示す．

Efficient Group Key Agreement Robust against Some Node Faults

Tetsuya Hatano[†1] and Atsuko Miyaji[†1]

Group key exchange (GKE) allows a large group of n parties to share a common secret key over insecure channels. We was is to extend the well-known Burmester-Desmedt GKE with communicational complexity $O(\log n)$ (BDII) to have robustness, i.e. with resistance to party failures resulting from party crashes, run-down batteries of parties, and network failures. BD II also satisfies scalability: each party does not need to have the same environment such as computational resources, batteries, etc. The previous schemes in this area focus on Burmester-Desmedt GKE with communicational complexity $O(n)$ (BDI), where BDI does not satisfy scalability. As a result, the previous robust GKEs need communicational complexity larger than $O(n)$ and do not satisfy scalability. We proposed robust GKE with scalability, called CH-GKE.

In this paper, When the failure rate of each player is changed, we compare CH-GKE to previous GKE schemes.

1. はじめに

グループ鍵共有 (GKE) は n プレーヤーが安全でないネットワーク上で鍵共有できる．このことにより, グループ内のプレーヤーはメッセージの暗号化や復号をすることができる．大きなグループ内で, 安全にコミュニケーションできることは完全性を満たす部分の応用として必要である．例えば家や学校，災害地など，多くの分野で使われているアドホックワイヤレスネットワークなどを利用し，大きなグループ間でコミュニケーションをすることは現在のアプリケーションではなくてはならない要素である．しかし，多量の価値のある情報がネットワーク上を流れるようになったことにより, それら情報の盗聴や改ざん, 破壊, 不正アクセスといった悪意を持った攻撃者から情報を守る必要がある．広く知られている GKE は DH 鍵共有に基づいており, BDI と BDII は一定ラウンドで鍵を共有することができる．効率性の観点での BDI と BDII の大きな違いは, BDI の通信量が $O(n)$ 必要であることに対し, BDII の通信量が $O(\log_2 n)$ な点である．また実用性の観点での BDI と BDII の大きな違いは, BDI ではプレーヤーの計算資源やバッテリーなどの環境が同じである必要があることに対し, BDII ではプレーヤーの計算資源やバッテリーなどの環境が同じである必要はない．例えば, あるプレーヤーは大きな計算資源を持っているが, もう一方のプレーヤーは限られた計算資源やバッテリーしか持っていなかった場合などが挙げられる．しかし, 任意のプレーヤーの故障やバッテリーの不足, ネットワーク障害などでプロトコル実行に失敗すると鍵共有できない．実際に, 1 人でもプレーヤーが故障してしまうと, 鍵共有に失敗してしまう．故障した場合に再びプロトコルを実行する必要があり, 失敗する回数分計算量やラウンド数が増加してしまう．従って, 任意のプロトコル故障に対して耐性を持った GKE が必要である.[1] は最初に耐故障性を実現した GKE を提案した．この方式では $O(n)$ ラウンド必要である．同様に,[6] は一定ラウンドで耐故障性を実現した GKE を提案した．この方式では $O(n^2)$ の通信量が必要である.[8] は効率的な GKE を提案した．この論文では JKT と呼んでいる．JKT は $O(nt)$ の通信量が必要であり, t プレーヤーの故障に対して耐性を持っている．JKT は BDI に耐故障性を実現した構成になっており, 上記の BDI の性質を持っている．しかし通信量が $O(\log_2 nt)$ であることと拡張性を達成することができていない．我々は

†1 北陸先端科学技術大学院大学
Japan Advanced Institite Science and Technology

⋆1 本研究の一部は科学研究費・基盤 A（21240001）の助成を受けています．

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

効率的な一定ラウンドの耐故障性を実現した GKE を提案した．この論文では CH-GKE と呼んでいる．CH-GKE は通信量が $O(\log_2 n)$ で $\frac{1}{2}$-プレーヤーの故障に耐性を持った一定ラウンドの GKE を行う．CH-GKE は Square Decision Diffie Hellman 仮定の下で安全である．本研究の内容は以下の通りである．2 章では準備として GKE のモデルと安全性の仮定，評価についての説明をする．3 章では既存研究である耐故障性を実現した JKT について説明をする．4 章では[11] で提案した CH-GKE についての説明をする．5 章では評価について説明をする．そして最後に 6 章で結論を述べる．

2. 準　　備

2.1 GKE のモデルと安全性の仮定

以下では，$\mathbb{G}$ を素数位数 p の巡回群とし，k をセキュリティーパラメーターとする．

Definition1 (Decisional Diffie-Hellman (DDH) 問題) g, y, g^a, $g^b \in \mathbb{G}$ が与えられたとき，$\{g^a, g^b, g^{ab}\}$ と $\{g^a, g^b, y\}$ を区別する問題を DDH 問題という．

Definition2 (Square-Computational Diffie-Hellman (Square-CDH) 問題) g, $g^a \in \mathbb{G}$ が与えられたときに $\{g^{a^2}\}$ を出力する．

Definition3 (Square-Decisional Diffie-Hellman (Square-DDH) 問題) g, y, $g^a \in \mathbb{G}$ が与えられたとき，$\{g^a, g^{a^2}\}$ と $\{g^a, y\}$ を区別する問題を Square-DDH 問題という．

Definition4 P_1, P_2,$\cdots$, P_n をプロトコル Π に参加している多項式時間で計算するチューリングマシーンとする．プロトコル Π とはグループ鍵共有 (GKE) プロトコルであり，全てのグループプレーヤーがあるプロトコルを実行したとき各々のプレーヤー P_i が同じ鍵 $K = K_i$ を計算することができることである．このとき P_i をプレーヤーと呼び，グループを n プレーヤーとする．

Definition5 プロトコル Π を n プレーヤーに対する GKE プロトコルとするとき，k をセキュリティパラメータとする．またネットワーク上で全てのコミュニケーションを盗聴できるような外部の受動攻撃者 $\mathcal{A}$ を仮定する．このとき $\mathcal{A}$ は Execute と Reveal, Test のアクセスを許される．Execute は何もアクセスされていないインスタンスによって実行された複写を出力する．Reveal は処理を終了したインスタンスのセッション鍵を出力する．Test は $\mathcal{A}$ がセッション鍵かランダムな数を与えられたとき $b = 0$ か $b = 1$ を出力する．このとき，Π を攻撃する $\mathcal{A}$ のアドバンテージ $|\mathrm{Prob}[Succ] - 1/2|$ がパラメータ k で無視できるほど小さければ受動攻撃に対して安全であるとする．$Succ$ は Test を使い，ビット b を推測したときに成功する事象である．

2.2 GKE の表記と仮定

GKE プロトコルはベキ演算 (EM) と乗算 (M), 逆元演算 (I) を使い，共有鍵を構成する．また我々の論文では通信量と計算量を保ちつつ，任意のプレーヤーの数に対して故障できる GKE プロトコルに焦点を当てる．最初に GKE プロトコルにおけるいくつかの評価方法を述べる．この論文では計算量を評価するとき，1 人に対しての最大に受信するメッセージ数 (最大受信量) と送信するメッセージ数 (最大送信量) に分ける．メッセージは決められた受信者に送付する point to point **p** と全員にメッセージを送付する Broadcast **b** に分けて考えるが，マルチキャストとブロードキャストは区別しない．このとき計算量を EM, M, I の数で比較し，評価を行う．

次に我々の論文では補助元 (auxiliary elements) という表現を利用する．任意の GKE プロトコルでは，任意のプレーヤーが自身の秘密鍵と公開情報を使い，共有鍵を計算することができる．しかし，あるプレーヤーが他のプレーヤーによって送信された公開情報を使い，鍵計算を行うので，共有鍵を計算するために必要な公開情報である補助元の送信に失敗した場合に共有鍵を計算することが出来ない．プレーヤーの故障に対して耐性をもった GKE プロトコルを達成するためにあるプレーヤーの故障に対して補助できるような補助元の与えかたについての議論を行う．

3. 既 存 研 究

この章では以前の GKE である BDI[4] で耐故障性を実現した JKT[8] を説明する．BDI では図 3.1 からわかるようにパーティーをリング構造に配置する．(プレーヤーのインデックスは n を法とするので，P_0 は P_n であり，P_{n+1} は P_1 である.) BDI はプレーヤーの端末故障に対して耐性を持っていない．もし，2nd ラウンドで共有鍵に必要な補助元のブロードキャストが失敗した場合，鍵を共有することができない．そこで JKT と呼ばれるプレーヤーの端末故障に耐性をもったプロトコルが提案された[8]．JKT は信頼できるブロードキャストチャネルを仮定し，2 つ以上の連続した t 人の故障がない場合を除き，任意のプレーヤーの端末故障に対して耐性を持っている．このとき以下のやりとりを行い，共有鍵 K を生成する．

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

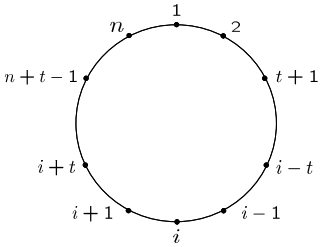

図 3.1 BDI

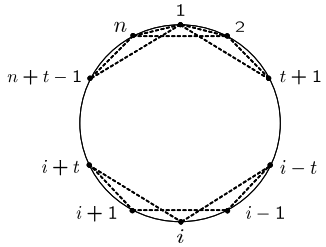

図 3.2 JKT

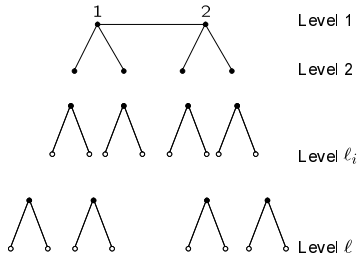

図 3.3 BDII

Protocol1 (t-JKT[8])

(1) プレーヤー P_i は秘密に選んだ $r_i \in \mathbb{Z}_p^*$ に対して, $z_i = g^{r_i}$ を計算し, ブロードキャストする.

(2) `A-List` は 1st ラウンドで送られてきたプレーヤーのインデックスリストとする. $|k-i| \leq t$ を満たすプレーヤーのインデックス (プレーヤー P_i は最も近い左側の t 人と右側の t 人) に対して, $x_{[k,\,i]} = \left(\frac{z_i}{z_k}\right)^{r_i} = g^{r_i^2 - r_k r_i}$ を計算する.

(3) `A-List` は 2st ラウンドで送られてきたプレーヤーのインデックスリストに更新する. そして `A-List` に載っているプレーヤーを順番に $\{P_{a_1}, P_{a_2}, \cdots, P_{a_m}\}$ 並べる. このとき $m \leq n$ である. プレーヤー P_i は共有鍵 $K = z_{a_{i-1}}^{m \cdot r_{a_i}} \cdot X_{a_i}^{m-1} \cdot X_{a_{i+1}}^{m-2} \cdots X_{a_{i-2}} = g^{r_{a_1} r_{a_2} + r_{a_2} r_{a_3} + \cdots + r_{a_m} r_{a_1}}$, を計算する. このとき, $X_{a_i} = x_{[a_{i-1},\,a_i]} \cdot (x_{[a_{i+1},\,a_i]})^{-1} = g^{r_{a_i} r_{a_{i+1}} - r_{a_{i-1}} r_{a_i}}$ である.

JKT ではプレーヤー P_i は $2t$ 個の補助元だけをブロードキャストする. すなわち $|k-i| \leq t$ に対しての $x_{[k,\,i]}$ だけをブロードキャストする (図 3.2). さらにハミルトンサイクルを使い, 共有鍵を計算する方法に改良することもできる.

Theorem1 (t-JKT[8]) $\mathbb{G}$ 上で square-DDH 問題が難しいと仮定すると, 受動攻撃に対して GKE プロトコルは安全である.

Protocol2 (t-JKT'[8]) JKT は耐故障性を実現している. しかし, 2 つ以上の連続した t 人の故障があった場合, 鍵を共有することができない. ここで任意の故障に対して耐性を持たすために[8] の手法を利用する. JKT の方式の 2nd ラウンドを鍵共有できるまで繰り返し行うことによって, 任意の故障に対して耐性を持たすことができる. 但し, ラウンド数と通信量は増加してしまう.

4. 効率的な耐故障性を実現した GKE 方式[11]

この章では, 以前に我々が提案した BDII[5] で耐故障性を実現した GKE プロトコルを説明する. BDII では図 3.3 からわかるようにパーティーを二分木構造に配置する. そしてプレーヤー P_i の親, 左の子, 右の子, 子孫をそれぞれ $\mathtt{par}(i)$, $\mathtt{l.child}(i)$, $\mathtt{r.child}(i)$, $\mathtt{ances}(i)$ と表記する. 但し, P_0 と P_1 はお互いに親であるようにする. ($P_1(P_2)$ は $P_2(P_1)$ の親の関係になる.) 従って, 木の葉であるプレーヤーを除いた全てのプレーヤーは 1 人の親と 2 人の子を持っている. BDII はプレーヤーの端末故障に対して耐性を持っていない. そこで CH-GKE と呼ばれるプレーヤーの端末故障に耐性をもったプロトコルを提案した[11]. CH-GKE は姉妹のどちらかが故障していない場合を除き, 任意のプレーヤーの端末故障に対して耐性を持っている. BDII では P_0 と P_1 は互いに親の関係であるが, それぞれの子孫とは関係を定義しない. これに対して, CH-GKE では, P_0 と P_1 は互いに親の関係だがそれぞれの子にとっては, おばの関係になる. (図 4.2 を参照) 図 4.1 では P_0 と P_1 の関係を表す. BDII と BDI を比べたときに, BDII ではプレーヤーの計算資源やバッテリーなどの環境が同じである必要はないので, CH-GKE は各々のプレーヤーの計算資源やバッテリーなどの環境が異なる場合に, JKT[8] より効果的に働く.

4.1 CH-GKE

この章では姉妹のどちらか一方の故障に耐性を持った CH-GKE を説明する. 図 4.3 は CH-GKE でのプレーヤーの関係を示している. 図中の線は補助元の流れを示しており, 例えば $\mathtt{aunt}(i)$ によって構成した補助元は i と $\mathtt{sis}(i)$ に送信する. このとき n プレーヤー P_1,

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

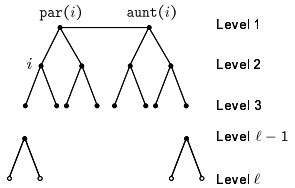

図 4.1 Party Tree in CH-GKE

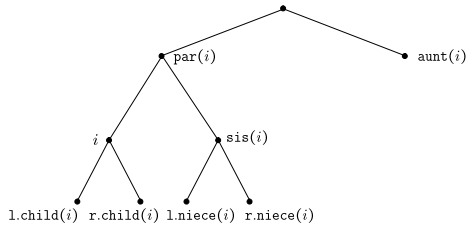

図 4.2 Neighbors in CH-GKE

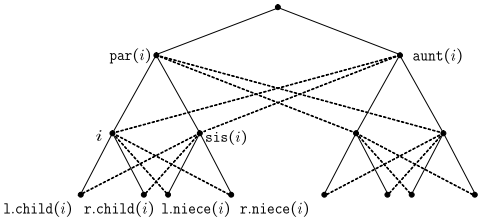

図 4.3 CH-GKE

P_2,$\cdots$,P_n は以下のようなやりとりを行い, 共有鍵 K を生成する.

Protocol3 (CH-GKE)

(1) プレーヤー P_i は秘密に選んだ $r_i \in \mathbb{Z}_q^*$ に対して, $z_i = g^{r_i}$ を計算し, 隣人に送信する.

(2) A-List は 1st ラウンドで送られてきたプレーヤーのインデックスリストする. このとき, 高さ 1 のプレーヤー P_i ($i \in \{1,2\}$) は 5 つの補助元である $x_{[\mathtt{par}(i),\, i]}$ と $x_{[i,\, \mathtt{l.child}(i)]}$, $x_{[i,\, \mathtt{r.child}(i)]}$, $x_{[i,\, \mathtt{l.niece}(i)]}$, $x_{[i,\, \mathtt{r.niece}(i)]}$ を以下のように計算し, 高さ ≥ 2 のプレーヤーにマルチキャストする.

$$x_{[\mathtt{par}(i),\, i]} = \left(\frac{z_{\mathtt{par}(i)}}{z_i}\right)^{r_i} = g^{r_{\mathtt{par}(i)} \cdot r_i - r_i^2};$$

$$x_{[i,\, \mathtt{l.child}(i)]} = \left(\frac{z_i}{z_{\mathtt{l.child}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{l.child}(i)} \cdot r_i};$$

$$x_{[i,\, \mathtt{r.child}(i)]} = \left(\frac{z_i}{z_{\mathtt{r.child}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{r.child}(i)} \cdot r_i}$$

$$x_{[i,\, \mathtt{l.niece}(i)]} = \left(\frac{z_i}{z_{\mathtt{l.niece}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{l.niece}(i)} \cdot r_i};$$

$$x_{[i,\, \mathtt{r.niece}(i)]} = \left(\frac{z_i}{z_{\mathtt{r.niece}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{r.niece}(i)} \cdot r_i}.$$

また, 高さ $\ell_i \geq 2$ のプレーヤー P_i ($i \in \{1,2\}$) は 6 つの補助元である $x_{[\mathtt{par}(i),\, i]}$ と $x_{[\mathtt{aunt}(i),\, i]}$, $x_{[i,\, \mathtt{l.child}(i)]}$, $x_{[i,\, \mathtt{r.child}(i)]}$, $x_{[i,\, \mathtt{l.niece}(i)]}$, $x_{[i,\, \mathtt{r.niece}(i)]}$, を以下のように計算し, 高さ $\geq \ell_{i+1}$ のプレーヤーにマルチキャストする.

$$x_{[\mathtt{par}(i),\, i]} = \left(\frac{z_{\mathtt{par}(i)}}{z_i}\right)^{r_i} = g^{r_{\mathtt{par}(i)} \cdot r_i - r_i^2};$$

$$x_{[\mathtt{aunt}(i),\, i]} = \left(\frac{z_{\mathtt{aunt}(i)}}{z_i}\right)^{r_i} = g^{r_{\mathtt{aunt}(i)} \cdot r_i - r_i^2};$$

$$x_{[i,\, \mathtt{l.child}(i)]} = \left(\frac{z_i}{z_{\mathtt{l.child}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{l.child}(i)} \cdot r_i};$$

$$x_{[i,\, \mathtt{r.child}(i)]} = \left(\frac{z_i}{z_{\mathtt{r.child}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{r.child}(i)} \cdot r_i}$$

$$x_{[i,\, \mathtt{l.niece}(i)]} = \left(\frac{z_i}{z_{\mathtt{l.niece}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{l.niece}(i)} \cdot r_i};$$

$$x_{[i,\, \mathtt{r.niece}(i)]} = \left(\frac{z_i}{z_{\mathtt{r.niece}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{r.niece}(i)} \cdot r_i}.$$

(3) A-List は 2nd ラウンドで送られてきたプレーヤーのインデックスリストに更新する. プレーヤー P_i は P_i から P_0 もしくは P_1 までの経路に含まれるプレーヤーのインデックスとし, この経路を $\underline{\mathtt{ances}}(i)$ とする. もし姉妹のどちらか一方の故障なら $\underline{\mathtt{ances}}(i)$ は存在する. このときプレーヤー P_i は以下のように共有鍵 K を計算する.

$$K = z_{\mathtt{par}(i)}^{r_i} \cdot \Pi_{j \in \underline{\mathtt{ances}}(i)} X_j = g^{r_1 r_2},$$

このとき, $X_j = x_{[\underline{\mathtt{par}}(j),\, j]} x_{[j,\, \underline{\mathtt{child}}(j)]} = g^{r_{\underline{\mathtt{par}}(j)} r_j - r_j^2} g^{r_j^2 - r_{\underline{\mathtt{child}}(j)} r_j} = g^{r_{\underline{\mathtt{par}}(j)} r_j - r_{\underline{\mathtt{child}}(j)} r_j}$ となり, そして $\underline{\mathtt{par}}(j)$ ($\underline{\mathtt{child}}(j)$) は経路上のプレーヤー P_j の親 (子) に対応する.

4.2 Security of CH-GKE

CH-GKE(Protocol 3) を破ることができる受動攻撃者を使い, Square-DDH Problem を解くことを示す.

Theorem2 $\mathbb{G}$ 上で Square-DDH が難しいと仮定し, CH-GKE(Protocol 3) を Π とす

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

るとき, Π は安全な GKE プロトコルである.

$$\mathtt{Adv}_{\Pi}^{\mathtt{GKE}}(t, q_{ex}) \leq \mathtt{Adv}_{\mathbb{G}}^{\mathtt{Square-DDH}}(t'),$$

ここで $\mathtt{Adv}_{\Pi}^{\mathtt{GKE}}(t, q_{ex})$ は Π に対する攻撃者であり, 時間 t で Execute にアクセスできる. $\mathtt{Adv}_{\mathbb{G}}^{\mathtt{Square-DDH}}(t')$ は $t' = t + q_{ex}(13n - 15)EM$ で Square-DDH を解こうとする攻撃者であり, n プレーヤーの数だけ増加する.

Proof: Π に対して, 時間 t 実行するアルゴリズム $\mathcal{A}$ を与えたとき, Square-DDH を解くことができる攻撃者 $\mathcal{B}$ を構成する.

$\mathcal{B}$ に組 $(g, y, h) \in \mathbb{G} \times \mathbb{G} \times \mathbb{G}$ を与える. ここで, $y = g^x$($\mathcal{B}$ には x の値を知らせない) とし, $h = g^{x^2}$ もしくは $\mathbb{G}$ 上からランダムに選んだ値とする. そのとき, $\mathcal{B}$ は $\mathcal{A}$ を実行し, $h = g^{x^2}$ かどうかを判定する. $\mathcal{B}$ は $z_1 = y$ とおく. 次に, ランダムに $c_2, \cdots, c_n \in \mathbb{Z}_p$ を選び, $i \geq 2$ に対して z_i を以下のようにおく.

$$z_i = z_{i-1} \cdot g^{-c_i} = g^{x - \sum_{j=2}^{i} c_j}.$$

z_i は以前に z_{i-1} を計算しているので, 計算することができる. ただし, $\mathcal{B}$ は $r_i = x - \sum_{j=2}^{i} c_j$ より, 指数部分である r_i は知らない. ここから, $\mathcal{B}$ は以下のように $x_{[\mathtt{par}(i),\, i]}$ と $x_{[\mathtt{aunt}(i),\, i]}$, $x_{[i,\, \mathtt{l.child}(i)]}$, $x_{[i,\, \mathtt{r.child}(i)]}$, $x_{[i,\, \mathtt{l.niece}(i)]}$, $x_{[i,\, \mathtt{r.niece}(i)]}$ を計算できる.

P_1 の場合では, $\mathcal{B}$ は c_2 を知っているので, 以下のように補助元を計算できる.

$$x_{[\mathtt{par}(1),\, 1]} = \left(\frac{z_{\mathtt{par}(1)}}{z_1}\right)^{r_1} = g^{r_2 r_1 - r_1^2} = g^{(x - c_2)x - x^2} = y^{-c_2}$$

残りの 4 つの補助元を考える. $\mathtt{l.child}(1)$ の数を $\mathtt{num.lc}(1) > 2$ とする. このとき, $\mathcal{B}$ は $\sum_{j=2}^{\mathtt{num.lc}(1)} c_j$ を知っているので, 以下のように補助元を計算できる.

$$x_{[1,\, \mathtt{l.child}(1)]} = \left(\frac{z_1}{z_{\mathtt{l.child}(1)}}\right)^{r_1} = g^{r_1^2 - r_{\mathtt{l.child}(1)} r_1} = g^{x^2 - (x - \sum_{j=2}^{\mathtt{num.lc}(1)} c_j)x} = g^{(\sum_{j=2}^{\mathtt{num.lc}(1)} c_j)x}$$
$$= y^{\sum_{j=2}^{\mathtt{num.lc}(1)} c_j},$$

残りの 3 つの補助元も上記のように計算できる.

P_2 の場合では, $\mathcal{B}$ は c_2 を知っているので, 以下のように計算できる.

$$x_{[\mathtt{par}(2),\, 2]} = \left(\frac{z_{\mathtt{par}(2)}}{z_2}\right)^{r_2} = g^{r_1 r_2 - r_2^2} = g^{x(x - c_2) - (x - c_2)^2} = y^{c_2} g^{-c_2^2},$$

残りの 4 つの補助元も同じ方法で計算することができる.

$P_i (i \geq 3)$ の場合では, $\mathcal{B}$ は 6 つの補助元を上記と同じ方法で計算できる. $x_{[\mathtt{par}(i),\, i]}$ と $x_{[\mathtt{aunt}(i),\, i]}$ の 2 つの補助元を考える. $\mathtt{par}(i)$ の数を $\mathtt{num.par}(i) < i$ とする. このとき, $\mathcal{B}$ は $\forall c_j$ を知っているので, 以下のように補助元を計算できる.

$$x_{[\mathtt{par}(i),\, i]} = \left(\frac{z_{\mathtt{par}(i)}}{z_i}\right)^{r_i} = g^{r_{\mathtt{par}(i)} r_i - r_i^2} = g^{(x - \sum_{j=2}^{\mathtt{num.par}(i)} c_j)(x - \sum_{j=2}^{i} c_j) - (x - \sum_{j=2}^{i} c_j)^2}$$
$$= y^{-\sum_{j=\mathtt{num.par}(i)+1}^{i} c_j \; (\sum_{j=\mathtt{num.par}(i)+1}^{i} c_j)(\sum_{j=2}^{i} c_j)}$$

$\mathcal{B}$ は $x_{[\mathtt{aunt}(i),\, i]}$ を上記と同じ方法で計算できる. 残りの 4 つの補助元を考える. $\mathtt{l.child}(i)$ の数を $\mathtt{num.lc}(i) > i$ とする. このとき, $\mathcal{B}$ は $\forall c_j$ を知っているので, 以下のように補助元を計算できる.

$$x_{[i,\, \mathtt{l.child}(i)]} = \left(\frac{z_i}{z_{\mathtt{l.child}(i)}}\right)^{r_i} = g^{r_i^2 - r_{\mathtt{l.child}(i)} r_i} = g^{(x - \sum_{j=2}^{i} c_j)^2 - (x - \sum_{j=2}^{\mathtt{num.lc}(i)} c_j)(x - \sum_{j=2}^{i} c_j)}$$
$$= y^{-\sum_{j=i+1}^{\mathtt{num.lc}(i)} c_j \; (\sum_{j=2}^{i} c_j)(\sum_{j=i+1}^{\mathtt{num.lc}(i)} c_j)}$$

残りの 3 つの補助元も上記のように計算できる.

$c_i (i \geq 2)$ がランダムな一様分布であるとき, z_i と $x_{[i,\, j]}$ は Π での理想的な状態となる. 複写は各々のプレーヤーに対して, 以下のように成る.

$$T = \{z_i, x_{[\mathtt{par}(i),\, i]}, x_{[\mathtt{aunt}(i),\, i]}, x_{[i,\, \mathtt{l.child}(i)]}, x_{[i,\, \mathtt{r.child}(i)]}, x_{[i,\, \mathtt{l.niece}(i)]}, x_{[i,\, \mathtt{r.niece}(i)]}\},$$

A-List は the 2nd round まで計算できた全てのプレーヤーのインデックスリストである. Test クエリーを要求した上で, $\mathcal{B}$ は以下のように共有鍵 K を出力する.

$$K = g^{r_1 r_2} = g^{x(x - c_2)} = g^{x^2} y^{-c_2} = h y^{-c_2}.$$

もし, K が正当な共有鍵であるなら, $h = g^{x^2}$ は正当な Square-DDH である. $\mathcal{B}$ は T を生成するために, 計算時間 $(13n - 15)EM$ を追加することによって, $\mathcal{A}$ と同様なアドバンテージで成功する. このとき, $\mathcal{B}$ は, 他の (g^r, y^r, h^{r^2}) を生成できるので, $\mathcal{A}$ はシングルの Execute クエリーを作る場合を考えることができる. ∎

The Katz-Yung のコンパイラー[9] とその改良[7] は能動攻撃に対して安全な認証 GKE プロトコルに変換する. このゲームでは Execute クエリーに加えて, Send クエリーを追加する.

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

Corollary1 任意の GKE プロトコル Π にコンパイラーを適用した認証 GKE プロトコル Π' は能動攻撃に対して安全である．このとき攻撃者は Send クエリーに q_s 回, Execute クエリーに q_{ex} 回アクセスできる．

$$\mathtt{Adv}^{\mathtt{AKE-fs}}_{\Pi'}(t, q_{ex}, q_s) \leq \frac{q_s}{2}\mathtt{Adv}^{\mathtt{KE}}_{\Pi}(t', q_{ex}) + \mathsf{Succ}_{\Sigma}(t') + \frac{q_s^2 + q_{ex}q_s}{2^k},$$

このとき，攻撃者が時間 t で Π' に対して，攻撃者が成功するアドバンテージを $\mathtt{Adv}^{\mathtt{AKE-fs}}_{\Pi'}(t, q_{ex}, q_s)$ とする．Succ_{σ} は時間 t で攻撃者が新しい平文/署名のペアを偽造できる最大のアドバンテージとする．

5. 評　　価

この章では, 章 3 で説明した既存研究の GKE と章 4 で提案した CH-GKE の比較を行う．CH-GKE と JKT, CS は耐故障性を実現している．特に CS は任意の故障に対して耐性を持っている．ここで故障に対して, 全てのプロトコルの条件を同じにするために Protocol 2 の手法を利用する．CH-GKE, JKT の方式の 2nd ラウンドを鍵共有できるまで繰り返し行うことによって, 任意の故障に対して耐性を持たすことができる．1 つの端末故障率を γ, 繰り返すラウンド数を δ, 鍵共有に成功する確率を $Succ$, 鍵共有に失敗する確率を f とする．

(1) CH-GKE は少なくとも姉妹の片方に active プレーヤーがいた場合, プロトコル実行に成功する．従って, 姉妹を 1 つの組とすると, 少なくとも姉妹の片方が active な確率は $1-\gamma^2$ なので, 各々のプレーヤーの故障がランダムに起こり, 独立に起きていることから CH-GKE の鍵共有に成功できる確率は $Succ = (1-\gamma^2)^{\frac{n}{2}}$ である．繰り返すラウンド数は $\delta = \frac{1}{Succ} = \frac{1}{(1-\gamma^2)^{\frac{n}{2}}}$ で計算することができる．

(2) t−JKT は 2 つ以上の連続した t 人の中で, 少なくとも 1 人に active プレーヤーがいた場合, プロトコル実行に成功する．このとき, t−JKT の鍵共有に成功できる確率は[8] より $Succ \geq 1 - \frac{n^2}{2}\cdot\gamma^{2t}$ である．繰り返すラウンド数は $\delta = \frac{1}{1-\frac{n^2}{2}\cdot\gamma^{2t}}$ で計算することができる．表 1 では t−JKT' の 1 つの端末故障 γ を変化させたときのラウンド数 $r(n = 500)$ を表したものである．この表から 4 つの故障率に対しての最適な t の値がわかる．例えば, $\gamma = 0.5$ では t が 11 から 13 のときが最適な値である．

以上のことを使い, 比較を行う．図 5.1 では 1 つの端末故障 γ を変化させたときの失敗確率 $f(n = 500)$ を表している．例えば提案方式である $CH-GKE$ では $\gamma = 0.006$ の場合, 99%の確率で鍵を共有できることがわかる．また 11-JKT では $\gamma = 0.475$ の場合, 99%の確率で鍵を共有できることがわかる．次に図 5.2 では 1 つの端末故障 γ を変化させたときのラウンド数 $r(n = 500)$ を表している．例えば, 提案方式である CH-GKE では故障率が 0.05 の場合, ラウンドが 3 であれば, 任意の故障に対して耐性を持つことがわかる．また 4-JKT では故障率が 0.125 の場合, ラウンドが 2 であれば, 任意の故障に対して耐性を持つことがわかる．

表 2(表 3) では CH-GKE と JKT, BDI, BDII, CS のプレーヤー 1 人に対しての通信量 (計算量) と故障数を表している．$\mathbf{p}$ と $\mathbf{b}$ の表記は章 2 で定義している．表から CH-GKA は $O(\log_2 n)$ の通信量と計算量で耐故障性を実現していることがわかる．

表 1 $\gamma = 0.5, 0.25, 0.1, 0.01$ に対する, t−JKT の t を変化させたときの成功確率 $Succ$ とラウンド数 δ, 受信メッセージ数の変化 ($n = 500$)

$\gamma = 0.5$

t	$Succ$	ラウンド数	送信メッセージ数
9	0.52	2.91	35
10	0.88	2.14	24
11	0.97	2.03	24
12	0.99	2.008	25
13	0.99	2.002	27

$\gamma = 0.25$

t	$Succ$	ラウンド数	送信メッセージ数
5	0.88	2.14	12
6	0.99	2.008	13

$\gamma = 0.1$

t	$Succ$	ラウンド数	送信メッセージ数
3	0.88	2.14	8
4	0.99	2.001	9

$\gamma = 0.01$

t	$Succ$	ラウンド数	送信メッセージ数
2	0.99	2.0014	5

6. 結　　論

JKT は BDI を拡張し, 2 つ以上の連続した t 人の故障が無い場合を除き, 任意の耐故障性を実現した．この方式では 1 プレーヤーにつき $O(nt)$ の通信量と計算量が必要である．他の

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

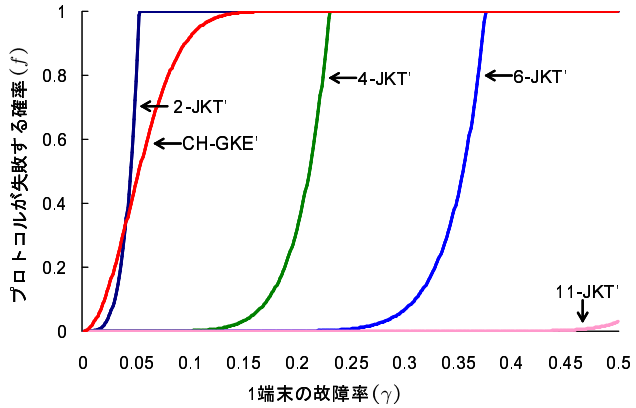

図 5.1 1 つの端末故障 γ を変化させたときの失敗確率 $f(n = 500)$

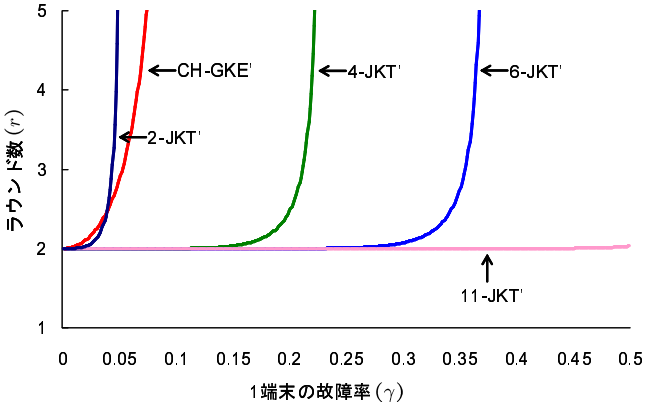

図 5.2 1 つの端末故障 γ を変化させたときのラウンド数 $r(n = 500)$

耐故障性を実現した GKE[3] は任意の故障に対して耐性を持つために $O(n^2)$ の通信量が必要である．我々が提案した CH-GKE は BDII を拡張し，少なくとも姉妹の一方が生きている場合に任意の耐故障性を実現した．この方式では 1 プレーヤーにつき $O(\log_2 n)$ の通信量と計算量が必要である．CH-GKE は Square-DDH 仮定の下で安全である．

表 2 n プレーヤー GKE の受信 (送信) メッセージ数の比較

party's type	large computational resources		low computational resources	
	sent message	received message	sent message	received message
BDI	$\mathbf{b}+2\mathbf{p}$	$(n-1)\mathbf{b}+2\mathbf{p}$	$\mathbf{b}+2\mathbf{p}$	$(n-1)\mathbf{b}+2\mathbf{p}$
BDII	$2\mathbf{b}+3\mathbf{p}$	$\log_2 n\mathbf{b}+3\mathbf{p}$	$\mathbf{p}$	$\log_2 n\mathbf{b}+\mathbf{p}$
JKT	$(2t+1)\mathbf{b}$	$(2t+1)n\mathbf{b}$	$(2t+1)\mathbf{b}$	$(2t+1)n\mathbf{b}$
CS	$2n\mathbf{b}$	$(n^2+n)\mathbf{b}$	$2n\mathbf{b}$	$(n^2+n)\mathbf{b}$
CH-GKE	$6\mathbf{b}+6\mathbf{p}$	$6\log_2 n\mathbf{b}+6\mathbf{p}$	$2\mathbf{p}$	$6\log_2 n\mathbf{b}+2\mathbf{p}$

表 3 n プレーヤー GKE の計算量と故障数の比較

party's type	large computational resources			low computational resources			robustness
	$\#EM$	$\#I$	$\#M$	$\#EM$	$\#I$	$\#M$	#max faults
BDI	3	1	$2(n-1)$	3	1	$2(n-1)$	0
BDII	4	2	$\log_2 n$	2	0	$\log_2 n$	0
JKT	$2t+1$	$2t$	$2(n-1)$	$2t+1$	$2t$	$2(n-1)$	$2t-1$
CH-GKE	8	6	$\log_2 n$	2	0	$\log_2 n$	$\frac{n}{2}$

参 考 文 献

1) Y. Amir, Y. Kim, C. Nita-Rotaru, J. Schultz, J. Stanton, and G. Tsudik, "Exploring robustness in group key agreement", *In Proceedings of ICDCS'01*, 399-409, IEEE CS, 2001.
2) A. Abdel-Hafez, A. Miri, and L. Orozco-Barbosa, "Authenticated group key agreement protocols for ad hoc wireless networks", Journal of Network Security, 2007.
3) T. Brecher, E. Bresson, and M. Manulis, "Fully Robust Tree-Diffie-Hellman Group Key Exchange", *In Proceedings of CANS'09*, LNCS **5888**(2009), 478-497, Springer-Verlag.
4) M. Burmester and Y. Desmedt, "A secure and efficient conference key distribution system", *In Proceedings of Eurocrypt'94*, LNCS **950**(1994), 275-286, Springer-Verlag.
5) M. Burmester and Y. Desmedt, "Efficient and secure conference key distribution", *In Security Protocols*, LNCS **1189**(1997), 119-130, Springer-Verlag.
6) C. Cachin and R. Strobl, "Asynchronous group key exchange with failures", *In Proceedings of PODC'04*, 357-366, ACM press, 2004.
7) Y. Desmedt, T. Lange and M. Burmester, "Scalable authenticated tree based group key exchange for ad-hoc groups", *In Proceedings of FC'07*, LNCS **4886**(2007), 104-118, Springer-Verlag.
8) S. Jarecki, J. Kim and G. Tsudik, "Robust Group Key Agreement Using Short

Vol.2010-DPS-142 No.28
Vol.2010-CSEC-48 No.28
2010/3/5

Broadcast", *In Proceedings of ACM CCS 2007*, 411-420, ACM 2007.
9) J. Katz and M. Yung, "Scalable Protocols for Authenticated Group Key Exchange", *In CRYPTO 2003*, LNCS **2729**(2003), 110-125, Springer-Verlag.
10) E. Konstantinou, "Cluster-based group key agreement for wireless ad hoc networks", ARES 2008, 2008.
11) 波多野 哲也, 宮地充子, "'効率的な耐故障性をもつ鍵共有方式", Computer Security Symposium, CSS2009-B8-3 (2009-10), 817-822.