Peer to Peer 上の安全なコンテンツ流通システムの提案

油田 健太郎[†1]　岡崎 直宣[†2]　朴 美娘[†3]

P2P コンテンツ流通システムは接続ホスト数の拡張性の高さや，耐障害性の高いシステムの構築が可能であるため注目が集まっている．しかし，中央管理サーバを持たない Peer to Peer（P2P）ファイル共有システムにおける匿名性の高いファイルでは，流通するコンテンツの信頼性に問題が生じる．例えば，悪意のあるユーザが悪意あるプログラムを正規ファイルに似せて流通させることが可能である．本論文では，評判値を用いた (k, n) 閾値秘密分散法による鍵の供託手法を提案する．また，匿名性の高い P2P 環境下において，信頼ノードの選択基準を生成するために信頼評価手法を導入する．さらに，コンテンツへのアクセスに必要な鍵情報を秘密分散法を用いて分散化することで，コンテンツへのアクセス可用性を確保する．本手法により安全で実用的なコンテンツ流通システムを実現する．

Proposal of a secure contents distribution system on P2P

KENTARO ABURADA,[†1] NAONOBU OKAZAKI[†2] and MIRANG PARK[†3]

In recent years, with the improvement of the high speed communication infrastructure, P2P contents distribution system attracts attention. In P2P contents distribution system, nonexistence of a central management server brings the system its robustness. However, it also leads to the problem of contents reliability and accessibility. We propose a secure contents distribution system which improves its accessibility by introducing a secret sharing scheme.

1. はじめに

近年，広帯域な通信基盤の整備に伴い，情報源の統合管理技術の重要性が高まっている．なかでも，Peer to Peer（P2P）コンテンツ流通システムが注目を集めている．P2P とは，クライアント-サーバ型のような中央サーバを持たず，一般ユーザのクライアント同士が直接通信を行う通信モデルである．P2P の特徴として，耐故障性及び拡張性の高いネットワークが構築可能ということが挙げられる．例えば，クライアント-サーバ型では，端末数が非常に増加するとサーバの処理能力，あるいはサーバにつながっているネットワーク回線の能力が限界に達し，システムが停止してしまうのに対し，P2P 型では端末数が膨大になっても通信帯域などに余裕がある限りシステムの稼動が可能という特徴がある．しかし，中央管理サーバを持たない P2P ファイル共有システムにおける匿名性の高いファイルでは，流通するコンテンツの信頼性に問題が生じる．例えば，内容を改ざんしたコンテンツや，ウィルスプログラムを混入したコンテンツの流通を容易に行うことが可能である．

コンテンツの完全性を確保する技術として，公開鍵暗号を利用したディジタル署名技術がある．ディジタル署名とは，作成者や改ざんの有無が明確になりにくい電子文書の欠点を補い，誰が作成したものか，また，改ざんが行われていないかどうかの確認を可能にする技術である．しかし，ディジタル署名は公開鍵の正当性を保障する第３者機関の認証局を設置する必要があり，そのまま P2P に用いると分散環境の利点を損なう恐れがある．

それに対し，Palomar らは少数ノードの協力による多重署名を用いたコンテンツ認証手法を提案している[1)]．この手法では，コンテンツへのアクセス許可証と多重署名を用いたコンテンツ証明書を発行することにより，P2P の利点を損なうことなく，コンテンツの保全とアクセス制御を実現している．しかし，匿名性の高い P2P 環境下において，何の基準もなしに多重署名を行ってもらう信頼ノードを選別することは困難であり，またアクセス許可証を発行したコンテンツのオーナがシステムから離脱している間は，コンテンツへのアクセスは不可能といった問題がある．匿名性が高く，ノードの参加離脱が頻繁に発生する P2P の環境を考慮すると，実用的ではない．

†1 大分工業高等専門学校
Oita National College of Technology
†2 宮崎大学
University of Miyazaki
†3 神奈川工科大学
Kanagawa Institute of Technology

そこで本論文では，より実用的で安全なコンテンツの流通を目的とする信頼評価手法を導入し，秘密分散法を用いて複数ノードでの鍵の供託を実現した P2P コンテンツ流通システムを提案する．本手法では，匿名性の高い P2P 環境下において，コンテンツ証明書の多重署名を行う信頼ノードの選択の基準値を生成するために，信頼評価手法を導入する．また，ノードの参加離脱が頻繁に発生する P2P 環境においてコンテンツのアクセス可用性を確保するため，秘密分散法による鍵情報の分散化を行い，複数ノードによる鍵の供託を行う．

以下，第 2 章では既存研究である Palomar らの手法について述べる．第 3 章では，本手法の詳細について述べ，第 4 章では本手法のセキュリティに関する評価と考察を述べる．そして，第 5 章でまとめと今後の課題について述べる．

2. 関 連 研 究

ここではまず，従来の多重署名を用いたコンテンツ証明書発行を行う既存のアクセス制御手法とその課題について述べる．次に，この課題を解決するために，本論文で導入する (k, n) 閾値秘密分散法及び信頼評価手法について述べる．

2.1 多重署名に基づくアクセス制御手法

ここでは，既存の，多重署名を用いたコンテンツ証明書発行を行うアクセス制御手法として Palomar らの手法[1] を紹介し，その課題について述べる．Palomar らの手法は，少数のノードの協力によりコンテンツの保全とアクセス制御を実現しており，大きく分けて以下の 3 つのサブプロトコルに分かれている．

(1) コンテンツ証明書発行サブプロトコル
(2) 新規参入サブプロトコル
(3) コンテンツアクセスサブプロトコル

流通するコンテンツには，それぞれ非公開，一部公開といったようにセキュリティレベルを設け，各セキュリティレベルに対応した鍵によって暗号化されている．まず，(1) のコンテンツ証明書発行サブプロトコルでは，コンテンツを公開するノード（以下，オーナ）は，複数の信頼ノードを選択し，選択した信頼ノードによる多重署名によって，コンテンツ証明書を作成する．そして，暗号化した復号鍵の情報と共に公開する．次に，(2) の新規参入サブプロトコルでは，コンテンツへのアクセスを要求するノード（以下，要求ノード）に対して，要求を受けたコンテンツのオーナがアクセス承認証明書を発行する．この承認証明書には，暗号化された復号鍵を復号するための情報が含まれている．そして，(3) のコンテンツアクセスサブプロトコルにおいて，要求ノードはオーナにアクセスし，承認証明書を用いて復号鍵を取得し，コンテンツへアクセスすることが可能となる．承認証明書を持っていないノードは，ブルートフォースによるコンテンツの復号は可能であるが，多大な時間を要するため，現実的にはコンテンツへのアクセスは不可能である．しかしながら，この手法においては，匿名性の高い環境である P2P 環境下において，信頼ノードの選択基準をどのように設定するのかが問題となる．また，オーナがシステムから離脱している際には，復号鍵の取得は不可能であるため，コンテンツのアクセス可用性に問題が生ずる．そこで本論文では，信頼評価手法及び秘密分散法を導入することで，アクセス可用性を確保しつつ安全なコンテンツ流通システムを実現する方法について検討する．

2.2 (k，n) 閾値秘密分散法

(k，n) 閾値秘密分散法とは，Blakley[2] と Shamir[3] によって独立に提案された秘密分散法の一つである．秘密情報を n 個のシェアに分割し，閾値である k 個のシェアを収集することにより元の情報を取得することが可能となる．そして，k-1 以下のシェアを収集しても元の情報の復元はできず，一つ一つのシェアからは元の秘密情報の一部ですら取得することはできないといった特徴がある．また，シェアの情報サイズはすべて同じであるため，シェアの内容が推測され難いという特徴もある．

2.3 信頼評価手法

ウィルスの感染や，コンテンツ改ざんのリスク回避のために P2P コンテンツ流通システムには高い信頼性が必要であり，ノードの信頼性を評価する手法として，「プロファイル情報」による手法[4]，「推薦」に基づく手法[5]，「評判」に基づく手法[6)–9)] などが提案されている．このうち，評判に基づく信頼評価手法は，他の手法のように認証局を必要とせず，web 上の複数の評判を基に評価を行うため，P2P において信頼モデルを形成するのに最適と考える．そこで，本論文では信頼評価手法として，評判に基づく信頼評価手法のひとつである「過去の振る舞い」への評判値の平均化に基づく手法[6] を導入する．

3. 提 案 手 法

本章では，まず本提案手法の概要について述べた後，各プロトコルの処理について説明する．

3.1 概　　要

本提案手法では，P2P におけるアクセス可用性を確保した安全なコンテンツ流通システムを提案する．まず，信頼ノードの選択基準とする評判値を生成するために，信頼評価手法を導入する．評判値は，過去の振る舞いを基に増減する．

本手法は，大きく分けて以下の 4 つのサブプロトコルに分かれている．

(1)　コンテンツ証明書発行サブプロトコル

(2)　復号鍵分散サブプロトコル

(3)　新規参入サブプロトコル

(4)　コンテンツアクセスサブプロトコル

まず，(1) コンテンツを公開するオーナは，コンテンツ証明書発行に協力してもらう複数の信頼ノードを，評判値を基に選定する．選定した信頼ノードの多重署名によりコンテンツ証明書を発行後，(2) コンテンツを共通鍵暗号で暗号化し，復号鍵を秘密分散法により分散化，ネットワーク上に分散配置する．(3) コンテンツにアクセスしたい要求ノードは，オーナにアクセス許可証の発行を申請する．評判値などを基に申請を受諾したオーナは，復号鍵の情報を含んだアクセス許可証を発行する．(4) 要求ノードは，アクセス許可証を基に復号鍵を取得し，コンテンツへアクセスを行う．秘密分散法により鍵の供託を行うことで，オーナ不在時にも，アクセス許可証を保持している正規ユーザは，コンテンツへのアクセスが可能となる．

以下，まず本手法で用いる信頼評価手法について述べ，各サブプロトコルの詳細について説明する．

3.2　信頼評価手法

ここでは，本論文で提案する，「過去の振る舞い」への評判値の平均化に基づく手法[6] を導入した信頼評価手法について述べる．まず，本手法で定める評判値の更新基準と評判値生成に用いる信頼ベクトルの定義を示し，評判値評価プロセスについて説明する．

3.2.1　評判値の更新基準

評判値は，過去の振る舞いによって増減する．コンテンツを流通させる場合，まず，コンテンツの取引が正常に行われることが重要である．コンテンツの取引失敗が多ければ，コンテンツの流通に影響を及ぼす．そこで，コンテンツの取引成功時には，取引を行ったノード同士で互いのノードの評判値を増加させ，取引失敗時には互いに減少するように評判値の更新を行う．また，取引を行い入手したコンテンツが改ざんを受けているもの，ウィルスに感染しているものといった異常なコンテンツであれば，コンテンツを受け取った要求者は，自身の保持している，コンテンツを提供したノードの評判値を減少させる．また，コンテンツ異常の際，コンテンツ証明書を検証し，もし，証明書に異常がなかった場合，証明書を作成したコンテンツオーナと証明書の作成に協力した信頼ノード群すべてが異常コンテンツを流通させようとした疑いがあるため，これらすべての評判値を減少させる．以上のように評判値を更新することで，正常なコンテンツを流通させるための信頼関係を形成できると考えられる．評判値の更新基準を表 1 に示す．

	要求者	提供者
増加	コンテンツ取引成功時	コンテンツ取引成功時
減少	コンテンツ取引失敗時 コンテンツ異常時	–

表 1　評判値の更新基準
Table 1　Updating of a reputation value.

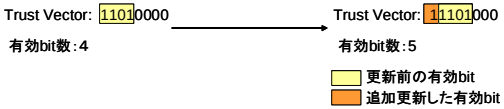

図 1　信頼ベクトルの更新処理
Fig. 1　Updating process of the reliance vector.

3.2.2　信頼ベクトル

信頼ベクトルとは，[6] にて用いられている，過去の振る舞いを示す $l$bit の 2 進数ベクトルである．ダウンロードや問い合わせの処理結果により更新を行い，有効 bit 数と共に保存する．値は，表 1 の判断基準を基に，増加の際には 1，減少の際には 0 と左端から更新し，1bit ずつ右にずらしていく．要求ノード N が提供ノード A と取引を行い，正常に取引が行われ，評判値を増加する場合の更新処理の例を図 1 に示す．各ノードは，信頼ベクトルをノード ID と有効 bit 数とともに各々で保持する．例えば，ノード ID が u_A，u_B，u_C であるノード A，B，C の信頼ベクトル保持しているノード N の場合，表 2 のような形式で保持されている．評判値として用いる信頼値と不信頼値は，信頼ベクトルを v，信頼ベクトルの 1 の補数を w，有効 bit 数を m として以下の式から導出される．

$$\begin{aligned} \text{信頼値} &= \frac{v}{2^m} \\ \text{不信頼値} &= \frac{w}{2^m} \end{aligned} \tag{1}$$

3.2.3　評判値収集プロセス

あるノード T の評価を知りたいノード A は，ノード T の評判値要求のクエリを自身の信頼する全ノードにブロードキャストする．クエリを受け取った受信ノードは，該当するノードの評判値を返答する．そして，ノード A は，受け取った値にノード A 自身の保持す

ノード ID	信頼ベクトル	有効 bit 数
u_A	11101000	5
u_B	10100000	3
u_C	11010111	8

表 2 信頼ベクトルテーブル
Table 2 Reliance vector table.

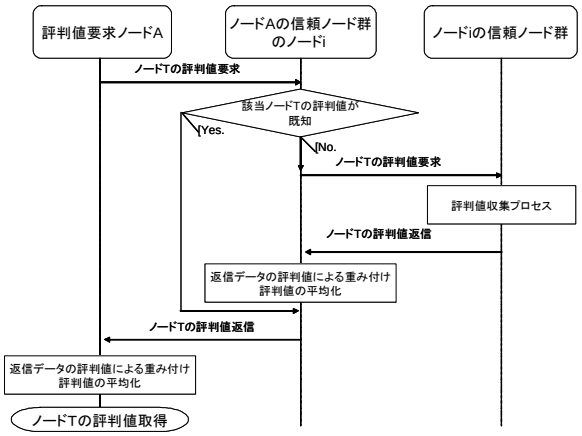

図 2 評判値収集プロセス
Fig. 2 Reputation value acquisition process.

る返信元ノード i の評判値を掛け合わせ，重み付けを行ったものを評判値 rep_1，rep_2,...，rep_n とする．もし該当ノードが未知であった場合，受信ノード自身の信頼する全ノードに対して，再帰的に評判値の問い合わせを行う．以上の処理を繰り返し行うことで，要求ノード A は，複数の評判値データを入手する．その後，入手した評判値を降順にソートし，評判値データの上位から任意の数 k のデータを採用し，採用したデータの単純平均化を行う．

$$\frac{\sum_{i=1}^{k} rep_i}{k} \tag{2}$$

平均化された評判値を基にノード T の評価を行う．処理の流れを図 2 に示す．本手法では，以上のプロセスより取得した評判値を基に，信頼ノードの選択と復号鍵受け渡しの判断を行う．

3.3 サブプロトコル

本節では，各サブプロトコルについて説明する．本手法では，ネットワークに参加しているすべてのノードは，公開鍵と対になる秘密鍵の組を二組所持しており，それぞれ秘匿通信とアクセス許可証発行，検証に用いる．以下に，本論文で用いる記号について示す．

- u_i: ノード i の ID
- m: コンテンツ
- x: 任意のデータ
- $H(x)$: 任意のデータ x のハッシュ値
- PK_i: ノード i の公開鍵
- SK_i: ノード i の秘密鍵
- VPK_i ノード i のアクセス許可証検証用の公開鍵
- VSK_i ノード i のアクセス許可証発行用の秘密鍵
- K_m: コンテンツの暗号化と復号に用いる共通鍵
- $enc_K(x)$: 鍵 K を用いて暗号化した任意のデータ x
- $S_i(x) = enc_{SK_i}(H(x))$: ピア i の任意のデータ x への署名
- C_B^A: A の B へ発行した証明書
- $Share_k$: IDk のシェア
- OLS: 信頼ノードリスト

3.3.1 コンテンツ証明書発行サブプロトコル

コンテンツ証明書発行サブプロトコルは，大きく分けて，信頼ノードの選択，分散署名，コンテンツの暗号化・コンテンツ証明書の生成の 3 つの処理に分かれる．コンテンツを公開する際には，これらの処理を順次行っていくことになる．

以下，コンテンツ証明書発行サブプロトコルの詳細を述べる．説明番号は図 3 に対応する．

コンテンツを公開したいオーナ A は，コンテンツ証明書を発行するために，まず（1）任意の数 n の信頼ノードを選択し，リスト $OLS =< u_A, u_1, u_2,..., u_n >$ を作成する．ここで，信頼ノードは，自身の保持している評価値テーブルを基に，評価の高いノードから選択していく．次に（2）コンテンツの証明書 $C =< u_A, ID_m, H(m), OLS >$ を作成し，多重署名に用いる初期署名 $C_0 =< C, S_A(C) >$ を生成する．そして,（3）分散署名プロセスを開始し，オーナ A は生成した C_0 をコンテンツ m と共に OLS の次のノード 1 に送信する．受け取ったノード 1 は,（4）コンテンツ m と署名 C を検証する．コンテンツ m の検証は，受け取ったコンテンツ m のハッシュ値を生成し，証明書 C に含まれているハッシュ値 $H(m)$ と比較することで行う．2 つのハッシュ値が同じであれば，署名申請を出しているコンテンツと，実際に検証を行おうとしているコンテンツが同一であると確認

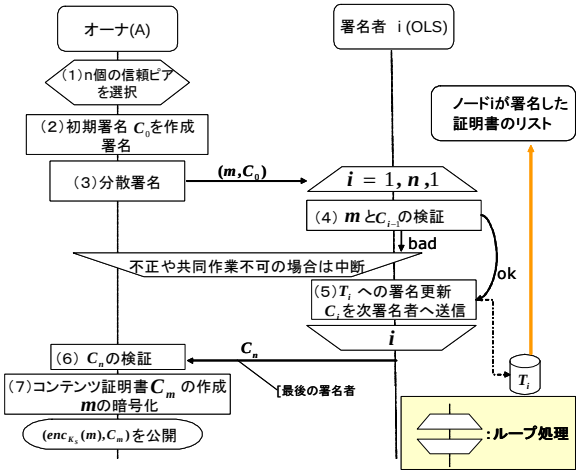

図 3 コンテンツ証明書発行処理
Fig. 3 Contents certification issue process.

できる．そのことによって，コンテンツ証明書の偽造を防ぐことが可能である．また，ここでコンテンツに不正プログラムが混入していないかをウィルス検知といった既存技術を用いて検証を行う．これら二つの問題がなければ（5）署名を行い，$C_1 =< C_0, S_1(C_0) >$ を作成し，OLS に記載されている次のノード 2 へコンテンツ m と C_1 を送信する．送信後，署名完了の通知メッセージをオーナ A へ送信し，署名した証明書のリストを更新し保持しておく．もし，検証の結果，問題があった場合には多重署名のプロセスを終了し，オーナに対して異常終了のメッセージを送信する．以上の処理を OLS に記載されている最後のノードまで繰り返し行う．つまり，署名とコンテンツを受け取ったノード i は，コンテンツの検証後，異常がなければ署名 $C_i =< C_{i-1}, S_i(C_{i-1}) >$ を生成し，OLS に記載されている自身の次のノード $i+1$ へコンテンツ m と署名 C_i を送信するといった処理を，OLS 最後のノードまで繰り返し行う．OLS 最後のノード n まで署名処理が終了した場合，オーナに多重署名 C_n を送信する．（6）受け取ったオーナは，多重署名 C_n に異常がないか検証を行う．異常がなければ，（7）コンテンツ証明書を $C_m =< C, C_n >$ として，コンテンツを共通鍵暗号を用いて暗号化した $enc_{K_m}(m)$ と共に公開する．処理の流れを図 3 に示す．

3.3.2 復号鍵分散化サブプロトコル

コンテンツの復号鍵である K_m を秘密分散法により分散化し，生成したシェアをネットワーク上に分散配置することで鍵情報の供託を行う．まず，(k，n) 閾値秘密分散法を用いてコンテンツの暗号化・復号に用いる共通鍵 K_m を分散化し，n 個のシェア $Share_1, \ldots, Share_n$ を作成する．次に n 個のシェアのそれぞれの保存先ノード ID $regID$ を乱数 R を用いて以下の式で導出する．

$$regID_j = H(ID_m \oplus R \oplus j)\ (j = 1, ..., n) \tag{3}$$

ここで，$\oplus$ は文字の連結を表す．

導出されたノード j に対して，オーナ O はシェアと乱数 R，分散数 n，秘密情報復元に必要なシェア数の閾値 k，アクセス許可証検証用の公開鍵 VPK_O，シェア受け渡しの可否基準とする評判値の閾値 λ から成るメッセージ $message(j)$ を，公開鍵暗号を用いた秘匿通信で送信する．

$$message(j) =< Share_j, R, n, k, VPK_A, \lambda > \tag{4}$$

導出で用いる乱数 R をアクセス許可証に含ませ，正規ユーザ以外からシェアの保存先を隠蔽することで，複数シェア保存先への DoS 攻撃によるコンテンツのアクセス障害を抑制できると考えられる．

3.3.3 新規参入サブプロトコル

新規参入サブプロトコルの処理について述べる．なお説明番号は，図 4 に対応する．まず，新規に参入するノード i は，自身の ID を以下の式で導出する．

$$u_i = H(IPaddress \oplus port) \tag{5}$$

次に，（1）ノード i は，任意のコンテンツにアクセスするために，そのコンテンツのオーナに対してアクセス許可証の発行申請を行う．なりすましによる不正なアクセス許可証の発行を防ぐため，公開鍵暗号方式を利用した相互認証を用いて処理を行う．まず要求ノード i は，オーナ O に発行申請を行うため，送信元 ID と送信先 ID，アクセス許可証の発行要求を含んだリクエスト情報 $req =< u_A, u_i, RF >$ を作成する．作成した req を自身の秘密鍵 SK_i を用いて暗号化し，リクエスト情報の平文と暗号文を共にオーナへ送信する．アクセス許可証の発行申請を受けたオーナは，（2）送信元である要求ノード i の公開鍵を取得し，受け取ったリクエストに改ざんなどの異常はないか検証を行う．異常がなければ，受信したリクエスト情報が確かに記載されている要求ノード i からのものであると確認することができる．検証後，要求ノード i の評判値を取得し，評判値の評価を基に発行の可否を決定する．評価の結果，許可証を発行する場合，（3）証明書 $C_i =< u_O, u_i, t >$ を作成する．ここで，t は許可証の有効期限である．オーナ O は，この証明書を自身の秘密鍵 SK_O で暗

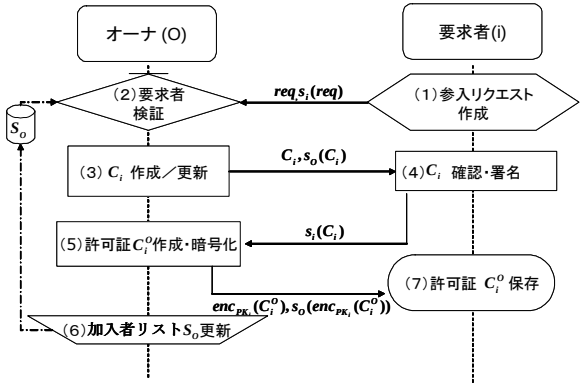

図 4 アクセス許可証発行処理
Fig. 4 Permission certificate issue process.

号化し，平文と暗号文を共にリクエストの返信として，要求ノード i に送信する．受け取った要求ノード i は,（4）オーナ O の公開鍵 PK_O を取得し，アクセス許可証を検証する．この検証を行うことで，確かに自身が要求申請したオーナ O からアクセス許可証の発行処理を行ってもらっていることが確認可能である．検証の結果，異常がなければ，C_i を自身の秘密鍵 SK_i で暗号化，C_i の平文と暗号文を共にオーナ O に確認メッセージとして送信する.（5）確認メッセージを受け取ったオーナ O は，受信メッセージを検証し，異常が無ければ，C_i を自身の保持している証明書発行の秘密鍵 VSK_O で暗号化する．この暗号化された $enc_{VSK_O}(C_i)$ をアクセス許可証の証明書 $CERT_i^O$ とする．$CERT_i^O$ に復号鍵の分散配置に用いた乱数 R と再度，有効期限 t をそれぞれ追加し，アクセス許可証 $C_i^O =< CERT_i^O, R, t >$ を作成する．そして，要求ノードの公開鍵 PK_i で暗号化し，暗号文に署名を行い許可証発行メッセージ $mes =< enc_{PK_i}(C_i^O), S_O(enc_{PK_i}(C_i^O)) >$ を要求ノードへ送信する．送信後,（6）オーナは許可証発行者リストを更新する.（7）許可証発行メッセージを受け取った要求ノードは，署名を検証し，異常が無ければ正規のアクセス許可証として C_i^O を保存する．処理の流れを図 4 に示す.

3.3.4 コンテンツアクセスサブプロトコル

コンテンツアクセスサブプロトコルは，大きく分けてアクセス許可証の確認，シェア保有者の探索，シェア受け渡しの可否の検証，有効数のシェア収集，コンテンツ復号鍵の取得の 5 つの処理を順次行っていく．

以下，コンテンツアクセスサブプロトコルの処理の詳細について説明する．なお説明番号は，図 5 に対応している．コンテンツにアクセスするには，目的のコンテンツへのアクセス許可証が必要である．そのため,（1）まずアクセス許可証の確認を行う．アクセス許可証を保持していない場合には，コンテンツのオーナに対して発行申請を行う．アクセス許可証を保持していた場合，有効な許可証であるか確認する．もし，失効していた場合には，アクセス許可証を保持していない場合と同様に，コンテンツのオーナ O に発行申請を行う．要求ノード i が有効なアクセス許可証を保持していた場合,（2）許可証に記載されている乱数 R を用いて，式 3 から，シェア保存先ノード ID である $regID_j$ を導出する．シェア保存先ノード ID 導出を，許可証に記載されている分散数である n の回数行えば，シェア保有者リスト LS(List of Sharer) を取得できる．そして，導出したシェア保有ノード群に対してアクセス許可証に含まれている証明書 $CERT_i^O = enc_{VSK_O}(C_i)$ をシェア要求メッセージと共に送信する．ここで，送信データは，送信先ノードの公開鍵を用いて送信データの暗号化を行い，送信する.（3）受信したシェア保存ノード j は，まず，証明書 $CERT_i^O$ を，保持している鍵 VPK_O を用いて検証する．検証は，保持している鍵 VPK_O を用いて受信した証明書 $CERT_i^O$ の復号を行い，C_i を確認する．異常なく復号でき，復号した C_i に記載されている情報に異常がなければ，受信した証明書は正規のものである.（4）次に要求ノード i の評判値を評価する．閾値 λ の条件を満たす評判値であれば，シェアの受け渡しを行う．ここで，シェアの受け渡しの際にも，シェア要求のときと同様に，公開鍵暗号を利用して，秘匿通信で行われる．処理の途中で異常があった場合，受け渡しの処理を終了し，終了メッセージを要求ノードに送信する．アクセス許可証を保持しており，かつ評判値が良好であれば,（5）復号に必要な十分な数 k 個のシェアを収集し，コンテンツの復号鍵を取得でき，コンテンツへのアクセスが可能となる．処理の流れを図 5 に示す.

4. 評価と考察

本章では，提案手法の性能評価を行う．評価項目は，暗号・復号処理のみを対象とした計算量である.

4.1 計 算 量

各サブプロトコルについて，処理の開始から終了までの暗号処理のステップ数と計算量を求め，評価を行う．各サブプロトコルにおける処理のステップ数と計算量を表 3 に示す．ここで，表中の記号は，H はハッシュ計算，S は署名処理，V は署名検証処理，PE は公開鍵暗号の暗号化処理，PD は公開鍵暗号の復号処理，CE は共通鍵暗号の暗号処理，CD

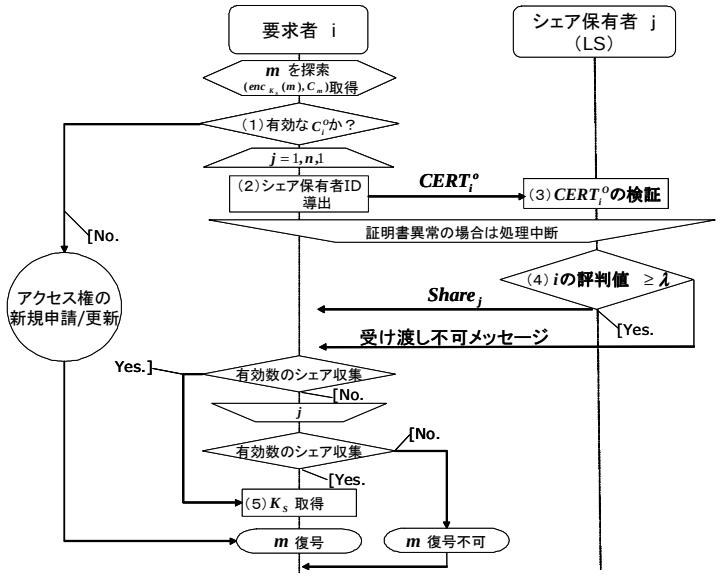

図 5 コンテンツアクセス処理
Fig. 5 Contents accessing process.

は共通鍵暗号の復号処理，SPL は秘密分散法による情報の分散化，COM は分散情報からの秘密情報の復元処理，p は署名者数，k は秘密情報の復元に必要なシェア数を表す．

コンテンツ証明書発行サブプロトコルの計算量は，署名者数 p に，復号鍵分散化サブプロトコルとコンテンツアクセスサブプロトコルの計算量は，シェアの閾値 k に，それぞれ依存する．この二つのサブプロトコルについて，表 3 のステップ数と，表 4 に示すベンチマーク[10] の値から求めたそれぞれの処理時間を図 6，図 7 に示す．なお，秘密分散法の処理時間は省いて求めている．shamir の秘密分散法[11] の鍵情報の分散化と復号の処理がそれぞれ 1 回ずつであり，処理時間は鍵長 256bit で数秒以内であることから，影響は少ないと考える．

図 6 より，コンテンツ証明書発行サブプロトコルは，署名者数 p が増加するほど，単調に処理時間が増加していき，また，コンテンツのサイズが増加するほど，同様に処理時間は増加する．例えば，署名者数が 25 人であり，公開したいコンテンツのサイズが 10MB であった場合，10 秒以内に処理は終了するが，署名者数が同じでありコンテンツのサイズが 500MB であった場合，処理が終了するまで約 2 分間かかってしまう．しかし，コンテンツ

サブプロトコル名	処理段階	暗号処理数	計算量
コンテンツ証明書発行	オーナのリクエスト	$1H + 1S$	
	分散署名	$pS + pH + \frac{p(p+1)}{2}V$	
	コンテンツの公開	$1CE$	
	合計:	$(p+1)(H + S + \frac{p}{2}V) + CE$	$\mathbf{O}(\|m\|p^2)$
復号鍵分散化	シェア生成	$1H + 1SPL$	
	シェアの分散配置	$kPE + kPD$	
	合計:	$H + SPL + k(PE + PD)$	$\mathbf{O}(k)$
新規参入	要求者のリクエスト	$1S$	
	オーナのリクエスト検証	$1V$	
	アクセス許可証発行	$3S + 2V + 1PE$	
	要求者の許可証の検証	$1V + 1PD$	
	合計:	$4S + 4V + PE + PD$	$\mathbf{O}(1)$
コンテンツアクセス	シェア保有者へのシェア要求	$kH + kPE$	
	アクセス許可証の検証	$kPD + kV$	
	シェアの受け渡し	$kPE + kPD$	
	シェアから復号鍵の復元	$1COM$	
	コンテンツの復号	$1CD$	
	合計:	$k(H + V + 2PE + 2PD) + COM + CD$	$\mathbf{O}(\|m\|k)$

表 3 計算量
Table 3 Computational complexity.

アルゴリズム	ms/operation
RSA 2048 Encription	0. 6
RSA 2048 Decryption	6. 08
RSA 2048 Signature	6. 05
RSA 2048 Verification	0. 16

アルゴリズム	MB/second
AES-256	96
SHA-256	111

表 4 各アルゴリズムの処理速度[10]
Table 4 Processing speed of each algorithm.

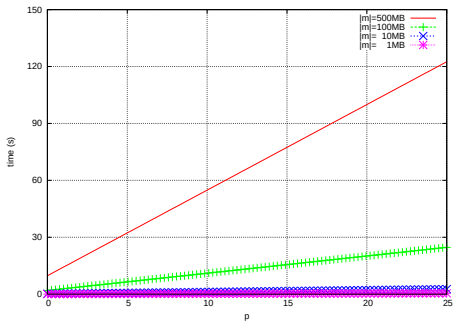

図 6 コンテンツ証明書発行サブプロトコルの処理時間
Fig. 6 Processing time of contents certification issuing subprotocol.

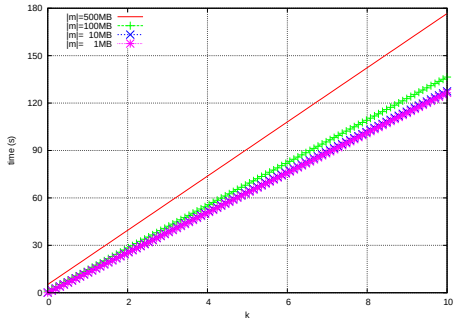

図 7 コンテンツアクセスサブプロトコルの処理時間
Fig. 7 Processing time of contents accessing subprotocol.

証明書の発行は，コンテンツを公開するときに 1 度だけ行う処理であるため，提案手法のこの処理時間は十分に現実的な速さであると考える．

図 7 より，コンテンツアクセスサブプロトコルの処理時間は，シェアの閾値とコンテンツサイズにに比例して増加していく．シェアの閾値 k が 6 であり，コンテンツサイズが 500MB だった場合，コンテンツにアクセスするまでに約 2 分間の時間を要する．コンテンツアクセスは，サブプロトコルの中で最も頻繁に発生する処理であるが，この処理時間はユーザの利便性から考えて許容範囲であると考える．

5. ま と め

本論文では，匿名性の高い環境下である P2P ネットワークにおいて，安全なコンテンツの流通を目的とする P2P コンテンツ流通システムを提案した．まず，コンテンツを公開するオーナは，コンテンツの完全性を保障するコンテンツ証明書の発行処理を行う．コンテンツ証明書の発行は，複数ノードによる多重署名によって作成する．協力してもらう複数の信頼ノードは，各ノードの過去の振る舞いを基に生成される評判値を判断基準に用いて選定する．選定した信頼ノードの多重署名によりコンテンツ証明書を発行後，コンテンツを共通鍵暗号で暗号化し，復号鍵を秘密分散法により分散化を行い，生成されたシェアをネットワーク上に分散配置する．コンテンツにアクセスしたい要求ノードは，オーナにアクセス許可証の発行申請を行う．評判値などを基に申請を受諾したオーナは，オーナの署名と復号鍵の情報を含んだアクセス許可証を発行する．要求ノードは，アクセス許可証を基に復号鍵を取得し，コンテンツへのアクセスが可能となる．

提案手法の各サブプロトコルについて，計算量による性能評価を行った．今後の課題として，計算機上での実装を行い，動作の確認を行うとともに，各サブプロトコルにおいての通信オーバーヘッド，全体で発生したデータ量に対する各ノードのデータ保持量を確認するといった，提案方式の実用性の検証が挙げられる．

参 考 文 献

1) Esther Palomar, Juan M.E. Tapiador, Julio C. Hernandez-Castro, Arturo Ribagorda,“Secure content access and replication in P2P networks”, Computer Communications31,pp.266-279(2008).
2) G.R.Blakley, “Safeguarding cryptographic keys“,Proceedings of the National Computer Conference 48,pp.313–317(1979).
3) A.Samir, “How to Share a Secret”, communication of the ACM, Vol.22, No.11, pp.612-613(1979).
4) StakhanovaN., BasuS., WongJ., StakhanovO., “Trust Framework for P2P Networks using Peer-Profile based Anomaly Technique”, Proseedings of the 2nd International Workshop on Security in Distributed Computing Systems(2004).
5) ChopraK, WallaceW, “Trust in Electronic Enviroments”,Proceedings of 36th Annual Hawaii International Conference on System Sciences,pp.331-340(2003).
6) A.A.Selcuk, ErsinUzun, M.R.Parriente, “A Reputation-Based Trust Management System for P2P Networks”, International Workshop on Global and Peer-to-Peer Computing, p.1(2004).
7) WithbyA., JosangA., IndulskaJ, “Filtering Out Unfair Ratings in Bayesian reputation Systems”, Proseedings of the 3rd International Joint Conference on Autonomous Agents and Multi Agent Systems(2004).
8) GuhaR., KumarR., RaghavanP., TomkinsA., “Propagation of Trust and Distrust”,Proceedings of the 13th International Conference on World Wide Web, pp.403-412(2004).
9) KamvarS., SchlosserM., Garcia-MolinaH., “The EigenTrust Algorithm for Reputation Manegement in P2P Networks”, Proceeding of the 12th International Conference on World Wide Web,pp.640-651(2003).
10) available from :http://www.cryptopp.com/benchmarks.html.
11) available from :http://point-at-infinity.org/ssss/.