修士論文

分散ネットワークにおける
コンテンツ管理システムの研究

2012年2月8日

指導教員　浅見　徹　教授

東京大学大学院情報理工学系研究科
電子情報学専攻　48-106409

楠　慶

内容梗概

本研究では，企業では無く一般の人々が草の根でコンテンツ共有する際に課題となるトラヒックとコンテンツ管理の問題を解決するため，サーバを有しない分散ネットワークにおけるコンテンツ管理を提案した．まず，現在分散ネットワークとして使われている P2P において，悪意のある不正ノードなどがコンテンツ削除命令に従わなくても，Multiple Secret Sharing を用いてコンテンツ管理が可能な手法を提案し，評価した．しかし P2P は既存 IP ネットワークのオーバレイネットワークであるため，下位層を考慮した配信が行えない問題がある．そのため，ISP によるサポートで下位層も考慮したルーティングが可能な次世代ネットワークアーキテクチャである Named Data Networking(NDN) において，コンテンツ管理のためのアプリケーションを設計した．さらに，NDN において，P2P と同様に悪意のある不正ノードが存在しても，Multiple Secret Sharing を用いてコンテンツ管理可能な手法を提案し，定性的な評価を行った．

目 次

第 1 章　序論 **1**

1.1　本研究の背景 2
1.2　本研究の目的 2
1.3　本研究の構成 3

第 2 章　P2P におけるコンテンツ管理 **4**

2.1　まえがき 5
2.2　Digital Right Management(DRM) 6
2.2.1　現在の商用動画配信で用いられている DRM 6
2.2.2　DRM の課題 7
2.3　動画配信に用いる P2P ネットワーク 7
2.3.1　P2P ネットワークの種類 7
2.3.2　Distributed Hash Table(DHT) 8
2.4　P2P における CGM サービスの要求条件 8
2.4.1　動画共有機能 8
2.4.2　著作権保護機能 9
2.5　基本手法:(k, n) 閾値秘密分散法を用いた手法 10
2.5.1　基本手法によるコンテンツ管理手順 10
2.5.2　基本手法によるコンテンツ管理性能と評価 11
2.6　修正手法:Multiple Secret Sharing による拡張 13
2.6.1　Multiple Secret Sharing 13
2.6.2　投稿者の初期設定手続き 14
2.6.3　コンテンツ投稿手続き 16
2.6.4　利用停止時の動作 18
2.7　計算量と配信コストの評価 19
2.7.1　計算量 19
2.7.2　配信の際に必要なコスト 19
2.8　シミュレーション 20
2.8.1　シミュレーション環境 20
2.8.2　シミュレーションシナリオ 22
2.8.3　結果と考察 23
2.9　おわりに 26

第 3 章　NDN におけるコンテンツ管理　27
3.1　はじめに 28
3.2　Named Data Networking 28
3.2.1　NDN の概要 28
3.2.2　NDN のセキュリティについて 31
3.3　コンテンツ管理システムの設計例 32
3.3.1　前提要件 32
3.3.2　Content Firewall を用いた設計例 32
3.4　コンテンツ管理システムの運用 35
3.4.1　コンテンツ公開 36
3.4.2　コンテンツ削除 38
3.5　提案システムのセキュリティに関する考察 38
3.5.1　D_{cmd} の拡散におけるセキュリティ 38
3.5.2　シャドウの送信におけるセキュリティ 38
3.6　おわりに 39

第 4 章　結論　40
4.1　本研究の主たる貢献 41
4.2　今後の課題 41

謝辞　43

参考文献　44

図 目 次

2.1 基本手法の手続き 10
2.2 違法コンテンツの分散鍵取得成功率 12
2.3 修正手法の初期設定手続き 15
2.4 修正手法のコンテンツ投稿時手続き 16
2.5 修正手法のコンテンツ利用停止時の手続き 18
2.6 オンライン率 20%のシミュレーション結果 23
2.7 オンライン率 40%のシミュレーション結果 24
2.8 オンライン率 60%のシミュレーション結果 24

3.1 Interest packet と Data packet のモデル 29
3.2 NDN router のモデル 30
3.3 提案ネットワーク構成 33
3.4 削除のリクエスト 33
3.5 削除の実行 34
3.6 削除の拡散 34
3.7 シャドウの分散 37
3.8 疑似シャドウの取得 37

表 目 次

2.1 シミュレーション条件 12
2.2 シミュレーションパラメータ 21

第1章

序論

1.1　本研究の背景

我々の身の回りのデジタルコンテンツは，基本的にはコピーが容易なため，インターネットで共有する機会が増えている．このようなコンテンツ共有において，コピーが誰にでもできてしまうと著作権，肖像権などに様々な問題があるため，コンテンツの利用を制御するコンテンツ管理は社会的インフラとして必要不可欠である．

単一のサーバからクライアントであるユーザにコンテンツを配信する場合にはこのようなコンテンツ管理は容易である．問題のあるコンテンツはサーバから削除すれば直ちに不正利用を阻止することができる．しかし近年のコンテンツは，スマートフォンの普及に伴い動画コンテンツが増えており，それによるトラヒックは年々増加している．

そのような大容量コンテンツ共有を低コストで行うための手段として以前からあるのが P2P によるコンテンツ共有アプリケーションである．しかし P2P アプリケーションは草の根アプリケーションとして発展したため，標準規格などがなかったこともあり，オーバレイネットワークとして下位層を考慮しない ISP にとってコストがかかるトラヒックを招いてしまった．さらに草の根であったことから，共有されるコンテンツは著作権など，法に触れるコンテンツが氾濫し，ユーザの間でコンテンツが分散してキャッシュされてしまうためコンテンツ管理も難しく，ASP に敬遠されてしまい社会的に認められるインフラにはなり得なかった．

そこで現在では，こうした草の根 P2P の分散ネットワーク技術を取り入れた，サーバによる分散ネットワークである Content Delivery Network(CDN) がコンテンツ配信技術として主に用いられている．しかし，CDN でも標準規格などはなく，配信技術は特許化しているものもある．さらに配信のためのサーバはユーザの近くに設置する必要があるため，世界中に企業が単独で設置しなければならない．そのため CDN によるコンテンツ配信コストは高く，こうした CDN による大規模なコンテンツ共有システムは Google などの大企業にしか提供できない．このような状況では，利用者は大企業にコンテンツの生殺与奪を全て握られてしまう．例えば Google の Youtube にアップロードした動画は，たとえ社会的意義の高いコンテンツであったとしても，政府などの圧力によって Google の判断によって”コンテンツ管理”されてしまい，利用できなくなってしまう．故に ISP や ASP から認められ，社会的なインフラとして標準になり得る別の選択肢として，個人に低コストでコンテンツ共有を可能にし，かつ継続性のある分散ネットワークにおけるコンテンツ共有の実現が望まれている．

1.2　本研究の目的

分散ネットワークにおけるコンテンツ共有の実現といってもその範囲は広い．その中で本研究の扱う範囲としては，ASP にとって非常に重要で，社会的なインフラとして認められるために必要不可欠なコンテンツ管理の実現とし，これを目的とする．

分散ネットワークとしては，まず現在唯一実用されている P2P を用いる．しかし P2P ではユーザの端末に専用のアプリケーションをインストールしてもらい，既存の IP ネットワークの上にオーバレイネットワークとしてネットワークを構築するため，各ノードは必ずしも信用できず，悪意のあるノードが管理者からの命令を無視する可能性や，オフラインになってしまいキャッシュにそもそもアクセスできない可能性がある．そのため各ノードにキャッシュとして残ってしまうコンテンツの管理は困難である．そこで本研究では，Multiple Secret Sharing を用いることで，設定された閾値の範囲内ならばネットワークにキャッシュがある程度残ってしまっても，コンテンツの不正利用を防ぐことができる手法を提案し，設定できる範囲の閾値が存在することをシミュレーションによって示し，評価する．

しかしながら，P2P では ISP に制御が困難なトラヒックを招いてしまうため，他の分散ネットワークでもコンテンツ管理を行う必要がある．そのため本研究ではさらに，実用ではないが将来の次世代インターネットアーキテクチャである Named Data Networking(NDN) を分散ネットワークとして用いるコンテンツ管理を提案する．この NDN では，P2P と異なり，コンテンツ管理のための機能が提案されている．しかし，それを用いたコンテンツ管理の枠組みは未だ無いため，Content Firewall と呼ばれるこの機能を用いたコンテンツ管理システムの設計例を示し，加えてネットワークのノードが信用できない場合に，P2P と同様に Multiple Secret Sharing を適用することでコンテンツ管理を行うシステムを提案する．

1.3　本研究の構成

本稿は 2 章と 3 章がそれぞれほぼ独立しており，2 章で P2P におけるコンテンツ管理の提案と評価を示し，第 3 章で NDN におけるコンテンツ管理を提案する．

そのうち，第 2 章ではまずコンテンツ管理手法として Secret Sharing を用いる従来手法を説明し，その応用として Multiple Secret Sharing を用いる提案手法を示し，シミュレーションによる評価を示す．

第 3 章ではまず分散ネットワークとして用いる NDN を説明し，NDN で考えられているコンテンツ管理のための機能である Content Firewall を用いた設計例を提案する．そして P2P のように信用できないノードが存在する場合に，設計例に P2P で用いた Multiple Secret Sharing を適用することでコンテンツ管理を可能にするシステムを提案し，それらのセキュリティについて考察を行う．

第2章

P2Pにおけるコンテンツ管理

2.1　まえがき

大容量コンテンツ配信，特に動画配信サービスのインターネットトラヒックは，年々大幅に増加している．2010 年末までには動画配信サービスのトラヒックは全世界のトラヒックのうち 40%占め，P2P ファイル交換を上回ると予測されている [1] [2]．その中でも，Youtube [3] やニコニコ動画 [4] などの一般利用者が制作したコンテンツを不特定多数に向けて共有する CGM(Consumer Generated Media) 型の動画共有サービスは国内の ISP 別のダウンロードサイトの割合において 4 分の 1 以上を占めている [5]．このような CGM 型動画共有サービスのトラヒックは，iPhone などの動画カメラ付きスマートフォンの普及や，3.9G の LTE サービスの開始によるモバイルインターネットの高速化で，今後さらに増加していくと考えられる．

このように増大するトラヒックに対して，従来のクライアントサーバ型配信よりも P2P による配信の方が，トラヒックを分散することでより低コストでコンテンツ配信することができる．そのため，P2P でコンテンツ配信するための様々な手法が研究され [6]，最近では Sharecast [7] や UG Live [8] など一部商用サービスで P2P による配信が行われている．しかしそのような P2P 配信は一般に広く普及しているとは言い難く，商用サービスで行われているのは通常のストリーミング動画配信などであり，前述の CGM 型動画共有サービスはほとんど全て通常のクライアントサーバ型で運営されている．さらに，今後も P2P によるトラヒックの相対量は縮小し，Contents Delivery Network(CDN) などのクライアントサーバ型配信が増大していくと予測されている [9]．

このように，P2P によるコンテンツ配信が普及しない原因の一つは，従来のクライアントサーバ型の動画配信サービスでは，違法コンテンツはサーバから削除して閲覧を制御することができたのに対して，P2P では全ての一般利用者に中継してしまった違法コンテンツを削除してもらうことが一般にはできないためであると考えられる．そのため，P2P による CGM 型動画共有サービスが普及するためには，CGM 型動画共有サービス特有の問題である一般利用者による違法コンテンツのアップロードに対応するためのコンテンツ閲覧制御機能が求められる．

従来のコンテンツ閲覧制御へのアプローチとして，P2P による商用コンテンツ配信サービスである Einy OnDemand と Skeedcast では，Windows Media Digital Rights Management(WMDRM) [10] を用いて動画を暗号化して配信し，その復号鍵を認証されたユーザのみに特定の集中サーバから別途配信し閲覧制御している．しかし，動画の配信主体が一般企業に限られている商用コンテンツ配信サービスと異なり，CGM 型動画共有サービスでは，動画の配信主体が一般利用者であるため動画の投稿者と動画の数が多く，動画の再生時間も短いため，復号鍵の取得が頻繁に行われる．そのため復号鍵配信サーバへの負荷が大きく，そのような負荷に対応できる設備投資が可能な企業でなければスケーラビリティを保ってサービスを提供することができない．また，既存の暗号化による DRM では，コンテンツの利用の可否がその復号鍵を配信するサーバの存在に依存してしまう．CGM 型動画共有サービスでは，コンテンツが利用者自身によって作成されるため，自らの作成

したコンテンツがサービス運営企業の都合によって利用できなくなることは受け入れがたいと思われる．そのため Veoh という P2P における CGM 型動画共有サービスや，P2P オンデマンド動画配信サービスである PPLive では暗号化による DRM がない [11] [12].

そこで我々は，コンテンツだけでなく，コンテンツの復号鍵も P2P で秘密分散法を用いて配信することで，DRM を低コストで実現し，設備投資が不可能な企業だけでなく個人でも運営することができる草の根的な，コンテンツ閲覧制御機能を有する CGM 型動画共有サービスを提案してきた [13, 14]．しかしこの手法は，動画投稿者が動画を投稿するたびに，対応する相異なる分散復号鍵を全て配り直す必要があった．不正なノードへの耐性のため分散復号鍵の数は，P2P ネットワークの規模に応じた適切な数必要なので，一般に少数の投稿者が多数のコンテンツを多数の利用者と共有する CGM 型動画共有サービスでは投稿者への負担を無視できない．そこで本稿では復号鍵の分散に Multiple Secret Sharing を用いる手法を提案する．修正手法では，最初に動画を投稿する際には基本手法と同様，対応する相異なる分散復号鍵を配信しなければいけないが，次に動画を投稿する際からは，特有公開情報と呼ぶコンテンツ特有の情報ひとつのみを毎回配信すればよい．

以下に本稿の構成を述べる．まず 2.2 節で DRM について，2.3 節で，本稿で使用する前提となる P2P ネットワークについて説明する．次に，2.4 節で本稿で目指す P2P 動画共有システムの要求条件について説明する．そして 2.5 節と 2.6 節で基本手法とその拡張を説明し，2.7 節で理論的な評価を，2.8 節でシミュレーションを行い，最後に 2.9 節で今後の課題を述べる．

2.2　Digital Right Management(DRM)

2.2.1　現在の商用動画配信で用いられている DRM

DRM とは，無制限にコピー，利用が可能なデジタルコンテンツに何らかの制限を設けて，コンテンツの著作権が侵害されることを防ぐ機能のことである．DRM の中，代表的なものは暗号化によるアプローチを用いている．現在使われているインターネット配信用 DRM の WMDRM や Fairplay はいずれも暗号化による DRM である．この暗号化による DRM に共通している点を，コンテンツを配信する側の配信者と，コンテンツを利用する利用者の二者を用いて示す．

1. 配信者はコンテンツを暗号化して，何らかの方法で利用者に配信する．
2. 配信者は何らかの方法で利用者を認証し，許可された利用者のみにコンテンツの復号鍵を何らかの方法で配信する．
3. 許可された利用者は，コンテンツ復号鍵を用いて暗号化コンテンツを復号して利用する．その際，コンテンツを利用するソフトウェアまたはハードウェアは配信者が用意した特別なもので，利用者の手元に復号されたコンテンツは保持されない．

なお，(3) の保持されないという状態は，ソフトウェアの不正な改造が無ければ DTCP により利用者が視聴するディスプレイへの出力まで保たれるが，その先のアナログの経路では保持されてしまう可能性がある．例えばディスプレイに映る動画をビデオカメラで録画されてしまうことなどが考えられる．このような場合でもコンテンツの DRM を保つための研究は，電子透かしなど様々に行われているが，このようなコピーは一般に品質の劣化が著しいため，本稿の範囲外とする．

2.2.2　DRM の課題

DRM には様々な課題があるが，CGM サービスでもっとも問題となるのは利用者の認証と復号鍵の配信である．利用者を認証するには，認証を行う配信者と利用者は何らかの手段で通信する必要がある．この通信を確実に実現するためには，サーバが必要だが，このサーバがもし配信者の都合で停止してしまった場合，利用者は認証できず，コンテンツを利用することができない．また，復号鍵の配信も，現在用いられているインターネット配信用 DRM ではサーバを用いているが，利用者の認証と同様にサーバが停止してしまうと復号鍵の配信も停止し，利用者はコンテンツの復号鍵を得ることができず，コンテンツを利用することができない．

このように従来の DRM では，本来制限無くコンテンツを利用できるはずの利用者の権利が，配信者によってのみ担保されてしまうという問題があった．CGM サービスでは，配信されるコンテンツの所有者は利用者ら自身であるため，サービス運営者のサーバによって認証と配信をする場合には自らの所有するコンテンツを利用する権利の生殺与奪をサービス運営者に握られてしまうことになってしまう．

2.3　動画配信に用いる P2P ネットワーク

2.3.1　P2P ネットワークの種類

P2P ネットワークの構成方式は，どのノードがどのファイルを持っているかの対応 (インデックス) を誰に持たせるかで Hybrid P2P，Super node P2P と Pure P2P の 3 種類に分類できる．Hybrid P2P では，インデックスを従来のクライアントサーバ型と同じく運営側のサーバで持つ．この方式では，結局運営側にサーバ負担が発生するため，本稿では採用しない．また Super node P2P は，長時間ネットワークに存在する Node や，端末の環境が高性能な Node を Super node として信頼してインデックスを持たせる．しかし CGM 型の動画配信サービスではノードの参加と離脱は頻繁に起こるため，Super node としての負担をユーザにさせるのは不適当であると考えられる．そのため本稿では Pure P2P を前提とする．Pure P2P では，インデックスを各ノードに分散させて持つため，各ノードに負担が分散しスケーラビリティが高い．

2.3.2　Distributed Hash Table(DHT)

Pure P2P の場合，インデックスの所在をどのように判断するかが問題となる．一番単純な方法は過去に通信したことのあるノードにインデックスの所在を問い合わせることだが，ノード数が多くなると問い合わせがネットワークにあふれてしまいスケーラブルではない．そのため，ノード数が多くても一定数の問い合わせでインデックスを持ったノードを発見するためのアルゴリズムとしてよく使われているのが Distributed Hash Table(DHT) である．

DHT には Pastry [15]，Chord [16]，Kademlia [17] などの様々なアルゴリズムがあるが，例えば Pastry の場合，10 万ノードが構成する P2P ネットワークでもたかだか 4 台のノードに問い合わせることで目的のノードに到達できる．本稿ではこれらの DHT アルゴリズムを採用し，大きなネットワークでも一定の通信回数で取得したいファイルを持つノードにたどり着けることを前提とする．

また，DHT に通常のハッシュ関数を用いた場合，問い合わせ (検索) は，ハッシュ関数の性質上完全一致検索しかできない．そのため部分一致検索を可能にするなど検索機能を拡張するために様々な手法が研究されている．本稿ではこれらの DHT の実装を用いて，利用者は利用したいコンテンツを確実に検索できることを前提とする．

2.4　P2P における CGM サービスの要求条件

P2P における CGM サービスは大きく分けて管理者と利用者で構成され，利用者は 2.3 節で説明した P2P ネットワークの構成ノードとなる．さらに利用者は，動画を投稿する投稿者，動画を見る閲覧者で構成され，利用者なら誰でも投稿者と閲覧者になることができる．以下に，本稿で想定している CGM サービスを実現するためのアプリケーションに要求される条件を動画共有と著作権保護に分けて説明する．

2.4.1　動画共有機能

CGM サービスでは，コンテンツは投稿者によって投稿され，オンデマンドで閲覧者のリクエストに応じて再生される．このため，動画を共有するには，① 利用者 ID の発行，② コンテンツの投稿，③ コンテンツ ID の発行，④ コンテンツの管理，⑤ コンテンツの閲覧といった基本機能が必要となる．まず 2.3 節で説明した DHT による P2P ネットワークを構成するために，全利用者に利用者識別子 (ID) を発行する必要がある．この ID は利用者の端末固有の情報と IP アドレスから DHT に基づいて発行しなければならないが，Pure P2P のため管理者などが発行主体になることはできない．そのため各利用者のアプリケーション側で ID を発行する機能が必要である．さらにコンテンツは投稿者によって投稿されるため，P2P ネットワークへの投稿機能がアプリケーションに必要である．また，投稿者が投稿したコンテンツを管理するため，投稿したコンテンツに一意なコンテンツ ID を

割り当てる必要がある．このコンテンツ ID も利用者 ID と同様に，利用者のアプリケーション側でコンテンツの内容から DHT に基づいて ID を発行する機能が必要である．この ID を利用して，投稿者と管理者は投稿したコンテンツを管理する．そして，閲覧者は投稿者のコンテンツの閲覧権限を持ち，コンテンツを自由に閲覧することができる．

2.4.2　著作権保護機能

動画共有システムを構築しようとすると，そのコンテンツに関する著作権を管理する機能 (DRM) が必要不可欠となる．特に CGM 型の動画共有システム (CGM サービス) では，コンテンツの投稿者が自分が著作権を持っていないコンテンツを不正に配信してしまう可能性がある．既存のサーバ・クライアント型 CGM サービスでは，管理者が著作権者から違法コンテンツの削除要求を受け，権利侵害を知っている，あるいは知りうる状態になった後に，直ちに削除あるいは利用停止の措置を取る必要がある．したがって，P2P における CGM サービスでも同等の著作権管理機能が必須である．

ここで，P2P における CGM サービスを運営する際に著作権管理機能を実現する最低限の機能要件を以下に示す．

1. コンテンツ削除は管理者と投稿者に認める．管理者は任意のコンテンツに，投稿者は自分のコンテンツのみに削除要求を出せる
2. 削除を要求された P2P ネットワークのノードは当該コンテンツの削除義務がある
3. 悪意のある不正ノードが削除要求を無視した場合でも，以降の当該コンテンツの視聴を高い確率で防止できる

CGM サービスの管理者は，著作権違反コンテンツの削除を行う責任があるため，コンテンツの著作権違反の状況を確認し，確認でき次第 P2P ネットワークに削除要求を出さなければいけない．そして P2P ネットワークにあるコンテンツキャッシュを完全に消すことができれば，コンテンツの利用はできなくなる．

しかし P2P ネットワークを構成しているノードは自由に参加と離脱ができるため，存在しているノードは必ずしもすべて信用できず，キャッシュの削除命令に応じないこともあるため完全削除は事実上不可能である．それでも，既存のサーバ・クライアント型と同様に，P2P による CGM サービスでも削除要求以降はコンテンツの利用を停止しなければならないため，キャッシュが完全に削除できない場合でも非常に高い確率で視聴を阻止できることが求められる．そのため，P2P ネットワークにコンテンツのキャッシュの一部が残ったとしても，コンテンツの利用はできないシステムにする必要がある．また，管理者や投稿者以外には削除要求の発行を許可しないため，公開鍵基盤 (PKI) を用いて削除要求を受け付ける際に管理者と投稿者の認証をする必要がある．ここで，実際には各ノードに不正コンテンツのキャッシュ削除を手動でしてもらうことは困難なので，不正コンテンツの削除要求にはアプリケーションで自動的に応じるよう実装する必要がある．よって本稿

では，削除要請に応じないノードとして，そもそもオフラインで削除要請が届かないノードと，削除要請を無視するような改造を施したアプリケーションを使用している悪質な不正ノードを想定する．

なお，削除要求以前に違法コンテンツが利用されてしまう可能性については，既存のサーバ・クライアント型 CGM サービスでも排除できていないことから，本稿の範囲外とする．

2.5　基本手法:(k, n) 閾値秘密分散法を用いた手法

2.4 章の要求条件を満足させるため，我々は，Shamir [18] と Blakley [19] の (k, n) 閾値秘密分散法 (Secret Sharing Scheme) を用いた手法を提案した [13, 14]．(k, n) 閾値秘密分散法とは，秘密を n 個に分散して保存し，そのうち k 個以上集めることができなければ元の秘密を復元できないという手法である．図 2.1 を用いて，この手続きを管理者，投稿者，閲覧者によるコンテンツの配信から利用，削除という流れを説明する．

2.5.1　基本手法によるコンテンツ管理手順

基本手法の概要を図 2.1 に示す．

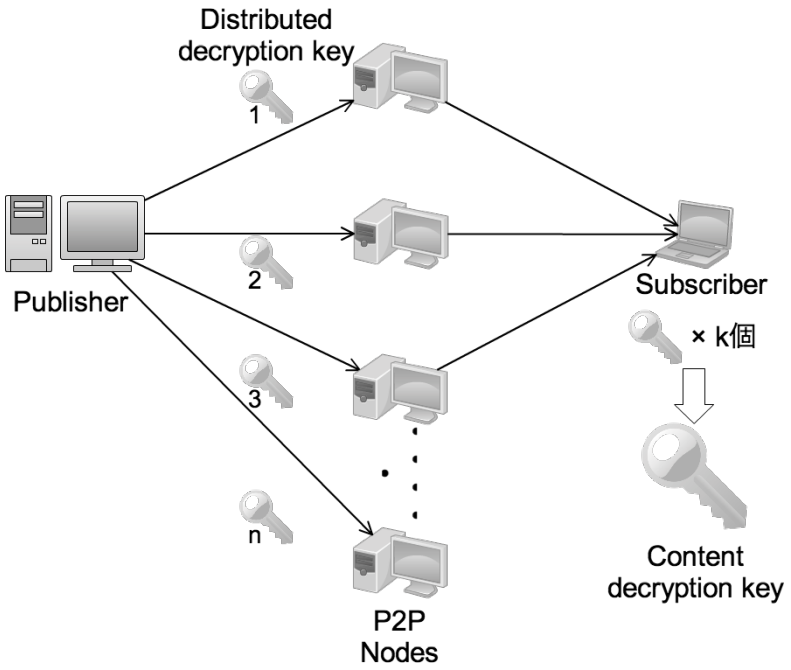

図 2.1: 基本手法の手続き

1. 投稿者は，投稿したいコンテンツを暗号化して P2P 担当ノードへ送信する．なお，P2P 担当ノードは 2.3.2 節で前提とした DHT に基づいて決定する．

2. 投稿者は，コンテンツ復号鍵を (k, n) 閾値秘密分散法で n 個の分散復号鍵として分散し，コンテンツと同様にそれぞれを各 P2P 担当ノードへ送信する．

3. 投稿者は暗号化分割投稿コンテンツとその分散復号鍵それぞれの ID のリストを投稿コンテンツのインデックス情報として P2P ネットワークに公開する．ここで，インデックス情報の ID は閲覧者がコンテンツを検索するときに用いるコンテンツ名などの文字列のハッシュである．

4. 閲覧者は，閲覧したいコンテンツをコンテンツ名などの文字列から得たハッシュで検索し，インデックス情報をダウンロードする．なお，検索には 2.3.2 節で前提としたコンテンツ名などの文字列の部分一致検索も可能な DHT を用いる．

5. 閲覧者は，インデックス情報を用いて各 P2P 担当ノードから全ての暗号化分割コンテンツと，k 個以上の分散復号鍵をダウンロードして，完全な暗号化コンテンツを結合する．

6. 閲覧者は，秘密分散法によってコンテンツ復号鍵を復元し，暗号化コンテンツを復号して利用する．

7. 不正なコンテンツの場合は，管理者が P2P ネットワークに分散復号鍵の削除要求を出し，P2P ネットワークの各ノードは当該コンテンツのインデックス情報，暗号化分割コンテンツと分散復号鍵を削除する．

ここで，削除要求時に要求に応じないノードがいたとしても，そのノード数が $k-1$ 人以下ならば，削除されずに残った分散復号鍵ではコンテンツ復号鍵を復元できなくなるため，コンテンツの不正利用を防ぐことができる．その際にそれらの削除されずに残った分散復号鍵の情報が一部 P2P ネットワークに残ることが考えられるが，利用できない情報は誰からもダウンロードされないため，そのような情報は長期的に自然消滅していくよう DHT アーキテクチャを設計すれば良い．

2.5.2　基本手法によるコンテンツ管理性能と評価

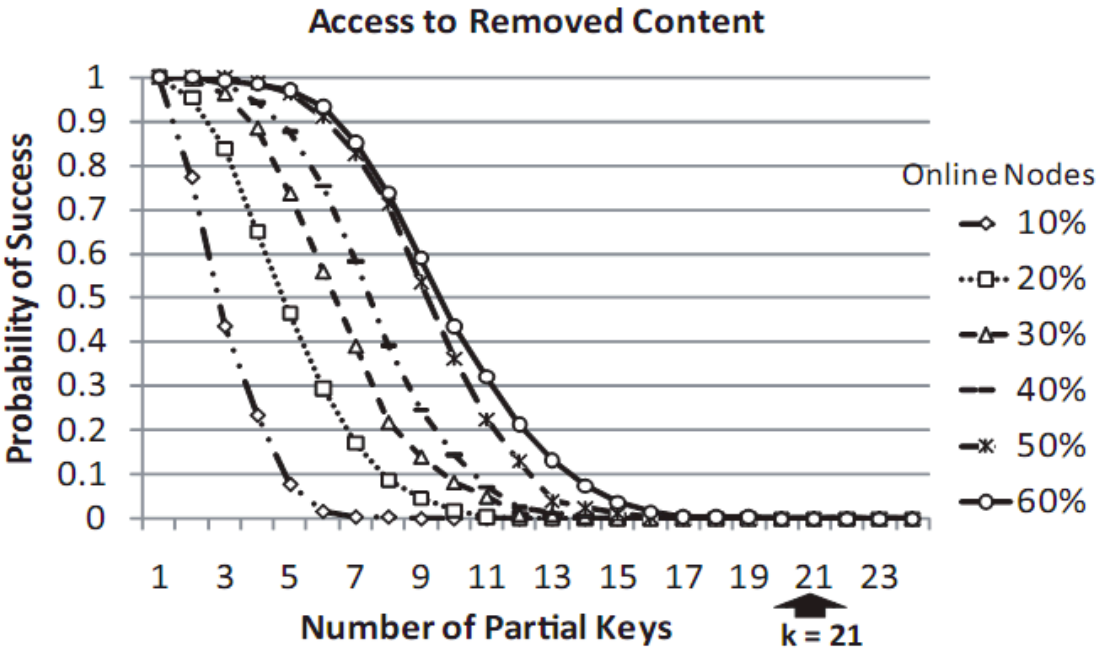

図 2.2: 違法コンテンツの分散鍵取得成功率

表 2.1: シミュレーション条件

ノード数	1000
不正ノードの割合	10%
オンライン率	10% - 60% (10% 間隔)
分散復号鍵数 n	100
タイムアウト時間	60 分
DHT	Pastry(キャッシュ無し)

シミュレーション条件 (表 2.1) に基づいて，管理者による削除要求の後に要求に応じない不正ノードによるコンテンツ不正利用を Overlay Weaver [20] を用いてシミュレーションした．ここで，オンライン率とは，ある瞬間に全ノードのうち P2P ネットワークのノードとして振る舞うノードの割合であり，例えば 10%の場合，ある瞬間に 1000 台中 100 台のノードが P2P ネットワークで稼働していることを示す．なお，ノードの参加と離脱のパターンはトラヒックモデル M/M/S(0) [21] に従うものとした．またタイムアウト時間とは削除要求後に不正ノードがコンテンツ不正利用を試みる時間である．PPLive におけるノードのオンライン時間は平均が 60 分の指数分布に従うため，60 分とした [22]．このシミュレーションで，不正ノードは全分散復号鍵にランダムでアクセスして分散復号鍵の取得を試みる．不正ノードがタイムアウトまでに取得できる分散復号鍵の数は，オンライン率が高ければ高いほど多くなるが，21 を超えることはない (図 2.2)．そのため秘密分散法における閾値 k を 22 以上に設定しておけば，不正ノードによるコンテンツ不正利用を防

ぐことができる．

基本手法のコンテンツ不正利用に対する耐性は，分散復号鍵数 n と閾値 k の割合で決まるが，P2P ネットワークのノード数と比較して n が小さすぎると，k も小さいため，不正ノードに取得されてしまう分散復号鍵にばらつきがでてしまい，不正ノードに k 以上の分散復号鍵を取得されてしまう確率を無視できなくなる．そのため，十分な確率でコンテンツ不正利用を防ぐために n は実際の P2P ネットワークのノード数に従って適当に多く設定する必要がある．しかし n が多くなると，投稿者はそれぞれ異なる分散復号鍵を投稿時に毎回 P2P ネットワークに送信しなければならないため，投稿者の負担が大きくなる．

2.6　修正手法:Multiple Secret Sharing による拡張

2.5 節の基本手法では，投稿者がコンテンツを投稿するたびに，P2P ネットワークの規模に応じた数の相異なる n 個の分散復号鍵を P2P ネットワークに送信する必要があり，投稿者の負担が大きいという問題が残っている．この投稿者の負担を軽減するため，秘密分散法の代わりに Multiple Secret Sharing を使うことで，たかだか 1 つの情報を P2P ネットワークに送信するだけで投稿可能にできる

修正手法では，利用者には投稿者と閲覧者の他に，拡散者という役割も必要になる．拡散者としては，用いる DHT のアルゴリズムに依って，送信するシャドウのハッシュに対応する ID を持っている利用者が P2P ネットワークから選ばれる．なお，シャドウとは Multiple Secret Sharing で用いられる分散情報のことであり，2.6.2 節で後述する．ここではまず，修正手法で用いる Multiple Secret Sharing について説明する．そして修正手法を投稿者の初期設定手続き，コンテンツ投稿手続き，問題のあるコンテンツを削除するときの順に説明する．

2.6.1　Multiple Secret Sharing

Multiple Secret Sharing Scheme とは，He と Dawson [23] によって提案された秘密分散法の一種である．(k, n) 閾値秘密分散法では，一度秘密情報を分散したあとに秘密情報を追加するためには，全ての分散情報を分散しなおさなければならなかった．しかし Multiple Secret Sharing では，1 つの分散情報のみを追加すれば秘密情報の追加を行うことができる．Multiple Secret Sharing は様々な手法が提案されているが，修正手法では Duo Liu らが提案した楕円曲線を用いた Multiple Secret Sharing [24] を用いる．なお (k, n) 閾値秘密分散法では秘密は情報理論的に安全だったが，Duo Liu らの Multiple Secret Sharing では，秘密は楕円曲線状の離散対数問題に基づく計算量的な安全が保証されているだけである．しかし，修正手法によってコンテンツの不正利用を防げるかどうかは拡散者の数などの運用パラメータに依存し確率的に決まるため，計算量的安全性で十分と考えられる．ここでは Duo Liu らが用いた楕円曲線と Bilinear self-pairing を引用して説明する．

楕円曲線

ある位数 q のガロア体 $GF(q)$ の楕円曲線を

$$E : y^2 = x^3 + ax + b \ \left(a, b \in GF(q),\ 4a^3 + 27b^2 \neq 0\right)$$

で定義する [24]．この曲線上の点は有限アーベル群であり，

- $P = (x_1, y_1) \in E$ とした時，$-P = (x_1, -y_1) \in E$
- もし $Q = (x_2, y_2) \in E$ かつ $Q \neq -P$ なら $P + Q = (x_3, y_3)$ で，

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

$$\lambda = \begin{cases} \frac{(y_2 - y_1)}{(x_2 - x_1),} & P \neq Q; \\ \frac{(3x_1^2 + a)}{(2y_1),} & P = Q. \end{cases}$$

である．

Bilinear self-pairing

ここでは，Lee によって提案された Bilinear self-pairing [25] について説明する．

K を標数が 0 または素数 p である体とし，$E = E(\overline{K})$ を $\overline{K}$ 上で定義された楕円曲線とする．ここで $\overline{K}$ は K の代数閉包である．もし，それぞれの点 $P \in E$ で $lP = 0$ を満たすような正整数 l が存在するなら，E はねじれ群である．そのような整数で最も小さいものを P の階数と呼ぶ．l-ねじれ点は $lP = 0$ を満たす点 $P \in E(\overline{K})$ である．$E(K)[l]$ を $E(K)$ の l-ねじれ点の部分群 $(l \neq 0)$ を表すものとし，$E(\overline{K})[l]$ を簡単に $E[l]$ と書くことにする．ここで，l は素数であり，K の標数 $char(K)$ は，0 または p であると仮定する (p と l は互いに素)．このとき，$E[l]$ は，$E[l] \cong \mathbb{Z}_l \otimes \mathbb{Z}_l$ のように二つの巡回群の直和に分解することができる．

G と H を $E[l]$ に対するある生成組であるとする．すると $E[l]$ 上の点は全て生成組 G と H を使って表すことができる．そこで，$E[l]$ 上の点 $P = a_1G + b_1H$，点 $Q = a_2G + b_2H$ を考える．ここで a_1, a_2, b_1, b_2 は $[0, l-1]$ の整数であるとする．ある固定された整数 $\alpha, \beta \in [0, l-1]$ に対して，$E[l] \times E[l] \to E[l]$ の関数 $\mathcal{L}_{\alpha,\beta}(P, Q) = (a_1b_2 - a_2b_1)(\alpha G + \beta H)$ を定義する．ここで，$\alpha, \beta = 0$ の時は自明として除く

2.6.2　投稿者の初期設定手続き

投稿者は，初めてコンテンツを配信する時，または拡散者の割り当てを更新する時のみ初期設定を行う．投稿者の初期設定手続きの概要を図 2.3 に示す．ここで，投稿者を D，n 人の拡散者を $\{U_1, U_2, \ldots, U_n\}$ とする．

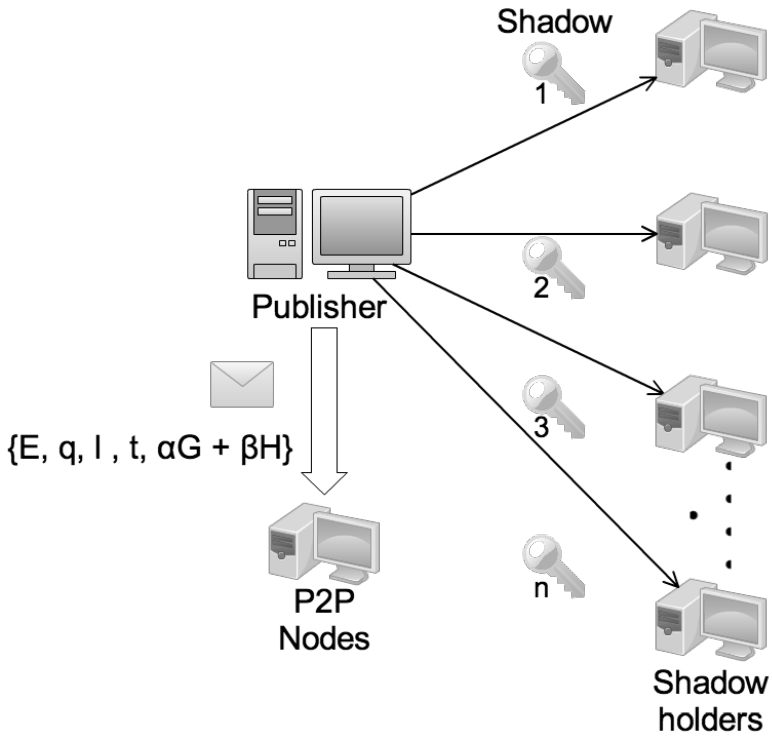

図 2.3: 修正手法の初期設定手続き

1. D は $GF(q)$ 上の楕円曲線 $E(GF(q))$ を選ぶ．ここで，$q = p^r$ で，p は $GF(q)$ での離散対数問題と，$E(GF(q))$ での離散対数問題が難しくなるように，十分に大きな素数とする．

2. D は大きな素数 l を選ぶ．ここで l はある小さな整数 t に対して $E[l] \subseteq E(GF(q^t))$ になるような数とする．

3. D は生成する対 $\{G, H\} \in E[l]$ と $\mathcal{L}_{\alpha,\beta}$ を決定する整数 $\alpha, \beta \in [1, l-1]$ を選ぶ．

4. D は公開情報 $\{E, q, l, t, \alpha G + \beta H\}$ を P2P で公開する．

5. D は以下のファンデルモンド行列 A を計算する:

$$A = \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & 2 & 2^2 & \cdots & 2^{k-1} \\ 1 & 3 & 3^3 & \cdots & 3^{k-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & (n-1) & (n-1)^2 & \cdots & (n-1)^{k-1} \end{pmatrix}_{n \times k}$$

6. D はランダムに $2k$ 個の数 $\{\tilde{a_j}, \tilde{b_j}\}_{1 \le j \le k}$ を選ぶ．ここで，それぞれ $\tilde{a_j}, \tilde{b_j} \in [0, l-1]$ $(1 \le j \le k)$ である．

7. D は

$$(a_1, a_2, \cdots, a_n)^T = A \cdot (\tilde{a_1}, \tilde{a_2}, \cdots, \tilde{a_k})^T$$

$$(b_1, b_2, \cdots, b_n)^T = A \cdot (\tilde{b_1}, \tilde{b_2}, \cdots, \tilde{b_k})^T$$

を計算する．

8. D は $1 \leq j \leq n$ のすべての拡散者 U_j に秘密が保たれる通信路でシャドウ $\{a_j, b_j\}$ を送信する．

2.6.3　コンテンツ投稿手続き

投稿者がコンテンツを投稿するたびに行われるのが，図 2.4 に概要を示すコンテンツ投稿手続きである．

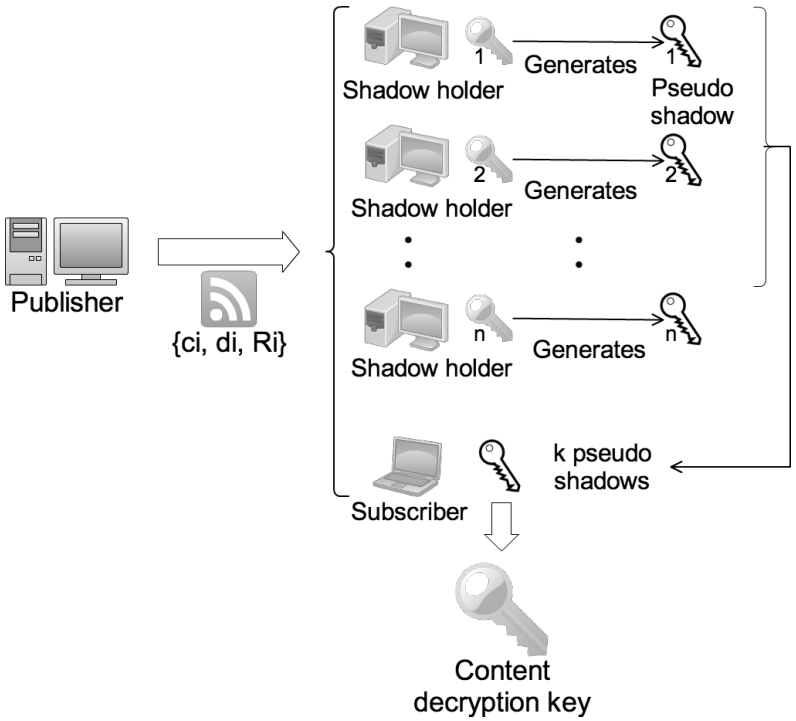

図 2.4: 修正手法のコンテンツ投稿時手続き

以下に投稿者，拡散者，閲覧者の順に各利用者の動作の流れを説明する．

投稿者の動作

ここでは，暗号化したコンテンツの復号鍵を m 個の異なる点 $\{M_1, M_2, \cdots, M_m\}$ とする．この $\{M_1, M_2, \cdots, M_m\}$ を秘密として分散する．以下で i は $1 \leq i \leq m$ のそれぞれでの手続きである．

1. D は $\tilde{P_k} = \tilde{a_k}G + \tilde{b_k}H$ を計算する．
2. D はランダムに $c_i, d_i \in [0, l-1]$ を選び，$Q_i = c_iG + d_iH$ を計算する．
3. D は $R_i = \mathcal{L}_{\alpha,\beta}(Q_i, \tilde{P_k}) + M_i$ を計算して，秘密ごとに異なる特有公開情報 $\{c_i, d_i, R_i\}$ を P2P ネットワークに公開する．

拡散者の動作

拡散者を $\{U_{u_1}, U_{u_2}, \cdots, U_{u_k}\}$ とする．以下で，i は $1 \leq i \leq m$ の，j は $1 \leq j \leq k$ のそれぞれでの手続きである．

1. U_{u_j} は，投稿者 D から受信したシャドウ $\{a_j, b_j\}$ を秘密に保持する．
2. U_{u_j} は，D が公開している公開情報 $\{E, q, l, t, \alpha G + \beta H\}$ と，秘密 M_i に対応する特有公開情報 $\{c_i, d_i, R_i\}$ をダウンロードする．
3. U_{u_j} は，擬似シャドウ $Q_{i,j} = \mathcal{L}_{\alpha,\beta}(Q_i, P_{u_j})$ を計算する．ここで，P_{u_j} と Q_i は，

   $$P_{u_j} = a_{u_j}G + b_{u_j}H$$

   $$Q_i = c_iG + d_iH$$

   である．
4. U_{u_j} は，閲覧者に擬似シャドウ $Q_{i,j}$ を送信する．

閲覧者の動作

閲覧者が秘密 $\{M_i\}_{1 \leq i \leq m}$ を復元する手続きを説明する．i は $1 \leq i \leq m$ の，j は $1 \leq i \leq k$ の間それぞれ繰り返す手続きである．

1. 閲覧者は U_{u_j} から擬似シャドウを $Q_{i,j}$ を受信する．
2. 閲覧者は

   $$T_i = \sum_{j=1}^{k} y_j Q_{i,j}$$

   $$y_j = \left(\prod_{J=1, J \neq j}^{k} (u_j - u_J) \right)^{-1}$$

   を計算する．
3. 閲覧者は D が公開している秘密ごとに異なる特有公開情報 $\{c_i, d_i, R_i\}$ をダウンロードし，秘密 $M_i = R_i - T_i$，即ちコンテンツの復号鍵を得る．

2.6.4　利用停止時の動作

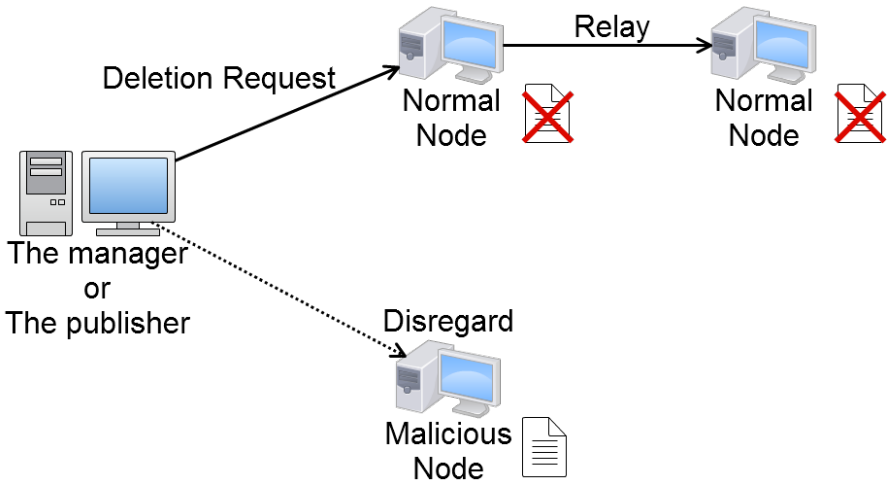

図 2.5: 修正手法のコンテンツ利用停止時の手続き

1. 何らかの理由でコンテンツの利用を停止する際には，管理者，または投稿者は当該コンテンツの擬似シャドウ配布の中止やキャッシュの削除を P2P ネットワークに要請する．

2. P2P ネットワークのノードは，要請を受信すると疑似シャドウのキャッシュがあれば削除し，別のノードにアクセスした時に要請を転送する (図 2.5)．そして，拡散者は要請を受信すると疑似シャドウの配布を中止する．

3. この時に要請に応じず擬似シャドウを配り続ける拡散者がいたとしても，その拡散者が $k-1$ 人以下ならば，P2P ネットワークに残る擬似シャドウではコンテンツ復号鍵を復元できなくなるため，コンテンツの利用を停止できる．

4. さらに，ある投稿者の全コンテンツの利用を停止したい場合には，管理者またはその投稿者は，当該投稿者のシャドウの削除を拡散者に要請する．

5. この時に要請に応じずシャドウを持ち続ける拡散者がいたとしても，その拡散者が $k-1$ 人以下ならば，$k-1$ 個以下の擬似シャドウしか計算できないため，やはりコンテンツ復号鍵を復元できなくなる．よって当該投稿者の全コンテンツの利用を停止できる．

2.7　計算量と配信コストの評価

2.7.1　計算量

ここでは，修正手法で必要とされる各利用者の計算量を投稿者，拡散者，閲覧者の順に示す．

投稿者の計算量

投稿者の初期設定手続き (2.6.2 節) では，大きい適当な素数を選択しているので，計算量は $O(1)$ である．また，コンテンツ投稿手続き (2.6.3) では適当な素数選択に加えて，秘密ごとに 4 回の乗算と 1 回の加算を行う．秘密の数はコンテンツ復号鍵のサイズに依存するが，コンテンツ復号鍵のサイズは一定で非常に小さい．よって計算量は $O(1)$ である．

拡散者の計算量

疑似シャドウの生成に 6 回の乗算を行っているので，1 疑似シャドウあたりの計算量は $O(1)$ である．

閲覧者の計算量

秘密 1 点の復元ごとに k^2 回の乗算と k 回の加算を行っている．よって計算量は $O(k^2)$ である．

2.7.2　配信の際に必要なコスト

動画コンテンツの暗号化に使われる復号鍵ファイルのサイズは，例えば現在広く使われている Windows media DRM(WMDRM) ではヘッダや ID などを含めて 2KB 以下しかない [10]．そのためコンテンツの復号鍵配信のみのスケーラビリティを考えたとき，配信コストとして問題となるのはその送信データ量ではなく送信のために必要な P2P ネットワーク構成ノードとの通信回数である．

基本手法では，2.5 節で説明した分散復号鍵の配信手続きにおいて，投稿者は n 個のそれぞれ異なる分散復号鍵を P2P ネットワークで拡散させる必要がある．n 台のノードにそれぞれ異なる分散復号鍵を DHT に基づいて送信するため，投稿者は DHT に基づいて分散復号鍵を配信すべき宛先ノードの所在を他ノードへ問い合わせる．この問い合わせ回数が 1 回の場合でも最低 n 台のノードと通信する必要があるが，実際には DHT において 1 回の問い合わせで宛先ノードを発見できるとは限らず，DHT のアルゴリズム性能に依って複数回他ノードへ宛先ノードの所在を問い合わせなければ宛先ノードのアドレスを知る

ことはできない．このプロセスが 1 コンテンツ投稿ごとに必要となるため，1 投稿者が投稿するコンテンツ数を m とすると，基本手法での投稿者が行う通信回数は $O(mn)$ である．

一方修正手法では，投稿者は最初にアクセス構造を設定する際の初期設定として，公開情報の P2P への公開と拡散者へのシャドウの配信が必要である．そのため初回投稿時には基本手法と同様に最低でも n 台のノード (拡散者) にそれぞれ異なるシャドウを配信する必要があるが，以降のコンテンツを投稿する際には P2P ネットワークにコンテンツごとに異なる特有公開情報 1 つを DHT に基づいて担当ノードに配信するだけでよい．そのため 1 投稿者が m 個のコンテンツを投稿する際に行う他ノードとの通信回数は $O((n+1)+m) = O(n+m)$ である．

なお，基本手法では投稿者から受け取った分散復号鍵を配信するだけだった担当ノードは，修正手法では拡散者という役割となり，配信されたシャドウをコンテンツ特有情報と組み合わせて計算した疑似シャドウを配信する必要がある．しかし，2.7.1 節で考察したように，そうした計算に際して一拡散者あたりに増える負担は疑似シャドウを最初に生成する時の 6 回の乗算に過ぎず，以降はその疑似シャドウを基本手法と同様に配信するだけである．そのため 1 拡散者あたりの負担はほとんど無視してもよい．

このように投稿者と拡散者の配信に関わるコストを，基本手法と修正手法とで比較すると，1 投稿者が 1 コンテンツしか投稿しない場合ほとんど投稿時のコストに違いはない．しかし 1 投稿者が投稿するコンテンツ数 m が増えると大きく通信回数に違いが出る．動画をアップロードする投稿者が一部の人に偏る CGM 型動画共有サービスにおいては，m は多くなると考えられるため，非常に重要であると考えられる．また，初回投稿以降の通信回数が 1 回で済むため，通信速度が限定されるモバイル端末での利用も容易になる．

2.8　シミュレーション

基本手法と修正手法で，管理者または投稿者が削除命令を出した後，当該コンテンツを不正利用できるかをシミュレーションすることで，2.4.2 節の著作権保護要件を満たすことを示す．

2.8.1　シミュレーション環境

2.3.2 節で前述した DHT のうち，広く利用されている Kademlia によるシミュレータを Python2.5.2 で実装した．シミュレータはトラヒックモデル M/M/S(0) [21] に従ってノードのオンライン化，オフライン化ができ，それぞれのノードは，最も人気のある上位 20%のコンテンツに 80%のアクセスが集中するような偏りのあるアクセスパターンを持つ．また，削除要求は，Kademlia に基づいて対象コンテンツを持つ可能性のある全てのノードに送ることができ，シミュレーションの実行中には全ノードが持っている key と value を各ステップごとに観察することができる．

Kademlia Network における各パラメータと修正手法のパラメータを，表 2.2 に示した．

表 2.2: シミュレーションパラメータ

	Parameter
k-bucket size	20
α	3
republish interval	60 min.
expire interval	120 min.
Number of nodes	1000
Number of malicious nodes	100
Number of contents	10
Number of shadows per content	100
Average online rate	20%，40%，60%
Percentage of heavy users	20%
Percentage of popular contents	20%
Average online period	60 min.
Usage period of each accessed content	10 min.
Timeout of relaying delete requests	30 min.
Number of contents chached on a node	4

なお，シミュレーション開始時に各ノードが初期状態として知っている k-bucket はランダムに 10 ノード分とし，シミュレーション中に最初に設定されたノード以外に新たなノードは接続してこないものとした．

Kademlia Network パラメータ

表 2.2 の Kademlia Network のパラメータである k-bucket のサイズと Kademlia Network のワイドパラメータである α は Kademlia を提案した Petar らの元論文 [17] にあるデフォルト値に従った．さらに，投稿者による再投稿を受けないと key-value を expire させる expire 間隔はデフォルト値では 24 時間だったが，シミュレーションを速く収束するために 120 分とした．

ノードのアクセスパターン

表 2.2 のノードのオンライン時間については，基本手法と同様に平均 60 分の指数分布に従うものとし，ノードの接続 (到着) 間隔も簡単のため指数分布に従うものとした．そして，各ノードがコンテンツにアクセスする頻度とコンテンツ人気の偏りもシミュレーションするため，頻繁にアクセスするノードと頻繁にアクセスされるコンテンツをそれぞれ 20% に設定した．頻繁なアクセスを実現するため，頻繁にアクセスするノードはコンテンツ利用終了 1 分後に次のコンテンツへアクセスし，それ以外のノードは 166 分後にアクセスするものとし，頻繁にアクセスされるコンテンツに 80% のアクセスが集中するよう実装した．なお，各ノードがコンテンツにアクセスする際には，コンテンツを視聴する間オンラインであることが期待できる．そのため所持シャドウの republish を，各ノードが他のノードにコンテンツにアクセスする際に行うよう実装した．

2.8.2　シミュレーションシナリオ

- 投稿者はコンテンツを投稿する．
- 各ノードは 90 分の間，前述のアクセスパターンに基づいて振る舞う．
- 投稿者は当該コンテンツのシャドウを持つノードに対して一度のみ削除を要請する．
- 各ノードは 210 分の間，前述のアクセスパターンに基づいて振る舞う．
- 各ノードは削除を要請されると該当シャドウを持っていた場合は削除し，以降は他の当該コンテンツのシャドウを持つノードに接続した際に，削除要請を拡散 (リレー) する
- 不正ノードは削除を要請されても要請を無視し，一度削除を要請されたコンテンツは expire 間隔を過ぎても expire させない．

削除要請は Kademlia における Publish とまったく同様に，lookup 手続きによって得られる k-bucket 全てに対して行われる．なお，削除要請は，通信相手が不正ノードか否か知ることはできないため不正ノードに対しても要請する．不正ノードは，削除要請を受けた段階で当該コンテンツが非合法コンテンツであると知り，削除を無視して expire させないようにして非合法コンテンツを保持する．そして，削除要請を効率的に P2P ネットワークに拡散させるため，一度削除要請を受けたノードは，設定された時間の間 (30 分間)，他のコンテンツアクセスなどの際に接続したノードに対して削除要請をリレーする．

削除要請のタイミングとしては，republish が始まる 60 分より前のコンテンツ投稿直後と，republish が始まって expire 間隔より前の republish 後と，expire 間隔の後の 3 つが考えられる．しかしコンテンツ投稿直後はシャドウを最初に投稿者が送信したノード以外は持っていないため，lookup 手続きでほぼ確実にそれら全てのノードに削除要請をすることができる．また，expire 間隔後は expire が発生してそもそもネットワークに残存シャドウが少なくなっている．そのため，今回のシミュレーションでは republish 後の 90 分に削除要請を行った．なお，今回のシミュレーションでは expire 間隔を短く実装したが，90 分後の削除要請より後の 120 分後に expire が始まるため，削除要請の効果に影響を及ぼすことはない．

2.8.3　結果と考察

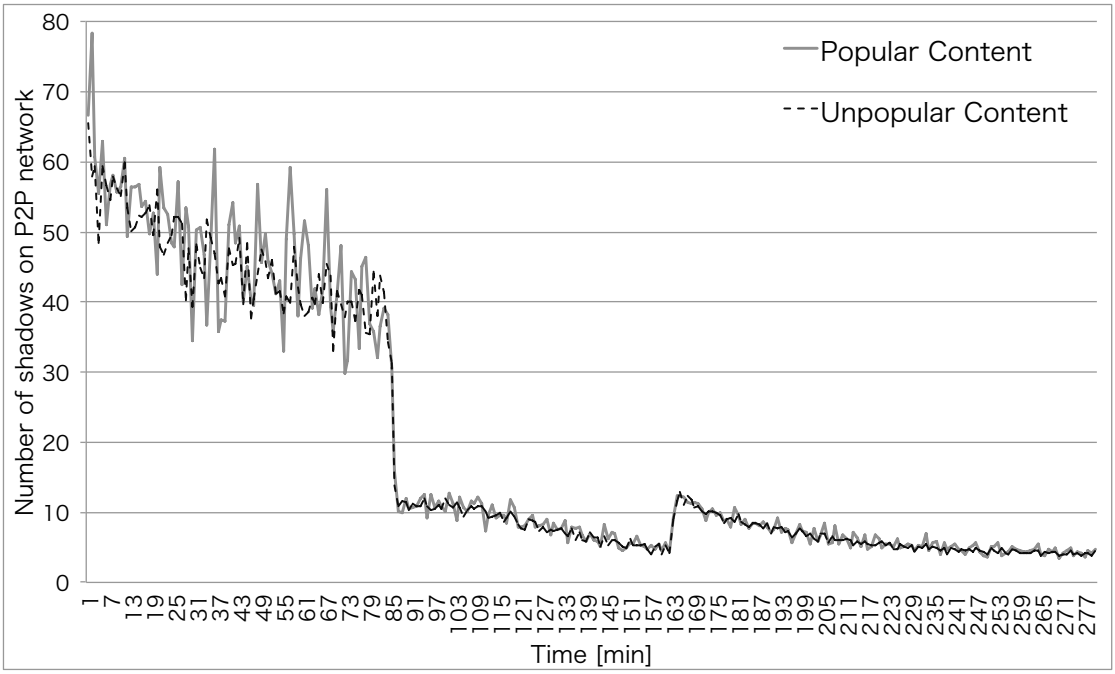

図 2.6: オンライン率 20%のシミュレーション結果

オンライン率 20%，40%，60%の場合，P2P ネットワーク上に残るシャドウ数の推移を

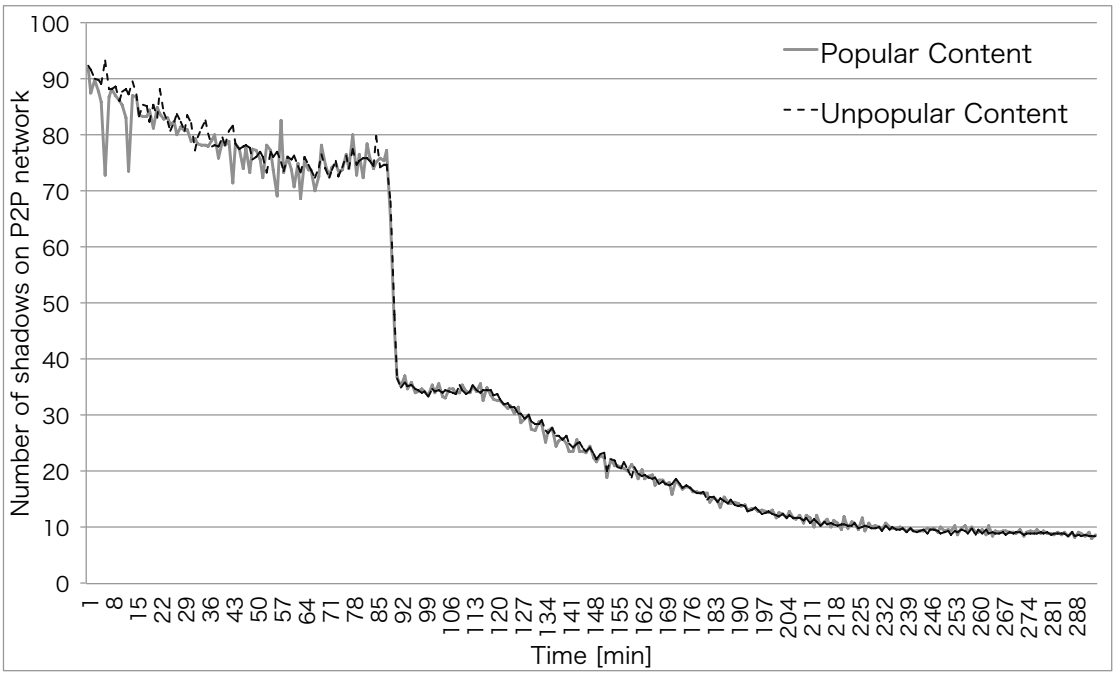

図 2.7: オンライン率 40%のシミュレーション結果

図 2.8: オンライン率 60%のシミュレーション結果

オンライン率の順に図 2.6，図 2.7，図 2.8 に示す．シミュレーションは 104 回繰り返し，その平均を取った．グラフの横軸は投稿者がコンテンツを投稿した時を 0 とするシミュレータ内部時間で，縦軸は 1 コンテンツあたり 100 個あるシャドウのうち，オンラインのノードが持っているシャドウ数，即ちオンラインに存在するシャドウ数で，実線は人気のあるコンテンツのシャドウ数を，破線は人気のないコンテンツのシャドウ数を示す．プロットしたのはその時刻にオンラインで取得可能な全シャドウ数であり，各ノードは lookup 処理で取得できた k-bucket の範囲でしか当該コンテンツを持つノードにアクセスできないため，実際にはこれより少ないシャドウしか得ることはできない．

最初に図 2.6 を考察する．オンライン率 20%では，ノードのオンライン-オフラインの切り替わりが激しいので，投稿直後から激しくシャドウ数が増減している．さらに今回のシミュレーションでは，全ノードのうち上位 20%のノードしか頻繁にアクセスしないこともあり，republish の頻度がシャドウを持ったノードがオフラインになる頻度より相対的に少ないため，シャドウ数が徐々に減少していっている．その後，163 分のあたりで急激にシャドウ数が回復しているが，これはアクセス頻度の低い残り 80%のノードが一斉にアクセスを行ったため，それらのノードが保持していたシャドウが一気に republish されたためである．削除要請がこなければこれらのサイクルが繰り返され，シャドウ数は長期的に緩やかな減少傾向となる．Kademlia ではオリジナルコンテンツを持つ投稿者が 24 時間に一度再投稿するので，この再投稿がある限りコンテンツの利用性は保たれる．図 2.6 では 90 分に投入された削除要請の後，一気にシャドウ数が 10 程度まで減少していることが観測できる．163 分あたりにアクセス頻度の低いノードが行う republish によって一時的にシャドウ数が回復しているが，それらのノードのアクセス先のノードには削除命令が行き渡っているので，削除命令直後の値を大きく上回ることはない．そのため閾値 k を 15 程度にしておけば，削除要請後は不正利用を防ぐことができることがわかる．

次に図 2.7 を見ると，最初の減少傾向はオンライン率 20%の時よりも遅くなっている．これは 20%の時よりもシャドウを持ったノードがオフラインになる頻度が少ないためである．この減少傾向は republish が始まる 60 分以降はさらに緩やかになっている．そして 90 分に削除要請がなされ，一気にシャドウ数が 40 以下まで減少している．そのため閾値 k を 40 程度にしておけば，削除要請後は即座に利用を防ぐことができることがわかる．なお，オンライン率 40%ではある程度 republish が行われているため，削除要請がすぐには行き渡らないが，最終的には 9 程度まで減少している．これはオンライン率 20%の最終的なシャドウ数 5 程度と比べて高いが，オンライン率の上昇と共に不正ノードにシャドウを獲得されてしまう確率が上がるためである．

最後に，図 2.8 では，オンライン率が高いため最初の減少傾向はさらに緩やかになっている．そして 90 分の削除要請の後，一気にシャドウ数が 60 程度まで減少している．そのため，この場合は閾値 k を 65 程度にしておけば良いことがわかる．なお，オンライン率 60%の場合では，republish がさらに活発に行われるため，なかなか削除要請が行き渡らない．しかし 169 分のあたりでアクセス頻度の低いノードが一斉にアクセスすることで，一気に削除要請が拡散されて最終的には 15 程度まで減少している．

なお，どのオンライン率でも，コンテンツの人気による削除への影響は少ない．このことから，不人気コンテンツでも適当な閾値 k を設定すれば，コンテンツの不正利用を防ぐことができることがわかる．

このように，不正ノード数を固定した今回のシミュレーションでは適切な閾値を設定すれば不正利用を防げることを示せたが，長期的なシステム運用においては，不正ノードが徐々に増加していく可能性が考えられる．そのためシステム運用当初はコンテンツの不正利用を防止できる閾値を設定できたとしても，不正ノードをネットワークから排除する手段がなければ，いつかは不正ノードに閾値を上回る数のシャドウを獲得されてしまい，不正利用されてしまう可能性が高くなってしまう．よって，不正ノードを正規ノードのネットワークから排除する機能が必要である．具体的には，不正ノードは削除要請がなされたシャドウを公開し続けているノードなので，各ノードが何らかの手段でそういった不正ノードの公開しているシャドウを察知できれば，k-bucket から排除したり，lookup 手続きで無視したりすることでネットワークから排除することができると考えられる．

また，このような機能をアプリケーションで実装すれば，削除要請を無視する不正なアプリケーションを用いようとする利用者に対して一定の抑止力となり，正規アプリケーションを使用するインセンティブにもなると考えられる．

2.9　おわりに

低コストで，DRM による非合法コンテンツへの対応が可能な P2P 配信による CGM 型動画共有サービスを目指して，コンテンツを暗号化して P2P 配信し，その復号鍵も (k, n) 閾値秘密分散法で P2P 配信する基本手法を紹介した．そしてその拡張として，(k, n) 閾値秘密分散法の代わりに Multiple Secret Sharing を用いる修正手法を提案し，(k, n) 閾値秘密分散法を用いた基本手法よりも，投稿者の 2 つ目以降のコンテンツを投稿する際の P2P ネットワーク構成ノードとの通信回数を抑えることができ，シミュレーションにより不正ノード存在時にも効果的にコンテンツの不正利用を阻止できることを示した．

第3章

NDNにおけるコンテンツ管理

3.1　はじめに

Named Data Networking(NDN) は，大規模コンテンツ配信を指向した次世代ネットワークアーキテクチャの一つである．NDN は従来の P2P や TCP/IP と比較して三つの優れた点がある．まず第一に P2P ネットワークのように分散したコンテンツ配送が可能で，かつ CDN のように物理トポロジを考慮したコンテンツ配送も可能なため，伝送効率が良いという点である．第二に通信されるすべてのデータは署名が義務づけられ，さらにコンテンツ本体は暗号化も可能なため Content Pollution に対応でき，さらに各ノード間でそれぞれフロー制御が可能な仕組みなどがあるため，DDoS 攻撃が非常に起こりにくいことなど，セキュリティ面に強い利点がある．そして第三に，ホストではなくコンテンツに名前付けを行うことで，Location independent なアーキテクチャを実現している点である．本稿では，不正コンテンツ削除などのコンテンツ管理が NDN では難しいという問題に対して，法的責任を持ってコンテンツ管理を行う Supervisor Application を用意し，ユーザからのコンテンツ削除リクエストをネットワーク全体に拡散させる手順を NDN の枠内で提案し評価する．

3.2　Named Data Networking

3.2.1　NDN の概要

NDN における通信は，データを消費する利用者とコンテンツ提供者が NDN router を介して Interest packet と Data packet(図 3.1 [26]) という二種類の Packet をやりとりすることによって成り立つ．NDN router は固定回線網，無線 LAN，3G/4G 網やローカルアプリケーションなどと通信可能な，複数のインタフェース (face) を持つことが可能なモデルとして想定されている (図 3.2)．ここではまず，NDN で利用者がコンテンツ提供者からデータを受信するまでの，一連の手順を Jacobson らの論文など [26,27] を参考に説明しつつ，NDN について説明する．なお，利用者やコンテンツ提供者は NDN router と 3.3 節でも用いる User Application を用いて通信するものとする．

利用者のリクエスト

利用者は望むデータに対応する Interest packet を上流の NDN router に送信する．Interest packet には対象となるデータのユニークな名前と，複数マッチ時の優先順位などのルールが含まれる．名前は URL のように可読性があり，階層化されているため，利用者は例えば/u-tokyo.ac.jp/videos/hoge.mp4 のように望むデータを指定することができる．さらに，このように人間が設定する部分に加えて，アプリケーションによって自動的に付加される末尾部分もある．この末尾部分は，タイムスタンプや，データを断片化する場合の管理ナンバーなどが含まれる．Data packet のサイズは，VoIP などの場合はレイテンシを短くするために小さくするなど，アプリケーションが任意に設定することが可能なの

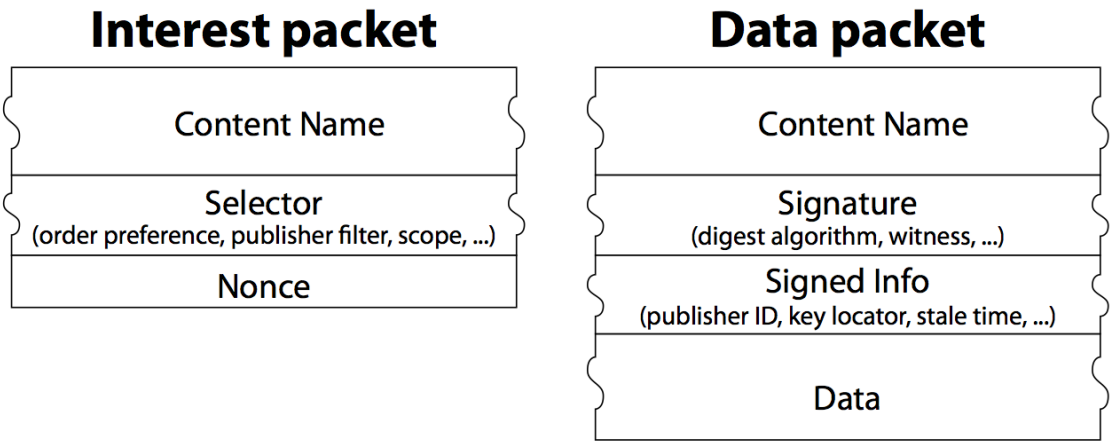

図 3.1: Interest packet と Data packet のモデル

で，そのような場合に管理ナンバーを付加する．これらの末尾部分を付加して，Interest packet の名前は例として/u-tokyo.ac.jp/voip/alice_to_bob/_v20120301/_s2 のように構成される．なお，このような名前は，URL のように利用者が覚えているものを直接指定する，名前のレポジトリを参照して指定するなど様々な方法が利用できる．名前のレポジトリとは，NDN で公開されているデータの名前のリストである．

NDN router の Interest packet 転送

NDN router は Interest packet を受信したインタフェースを Pending Interest Table(PIT) に記憶してから，Forwarding Information Base(FIB) に基づいて転送する．例えば 3.2.1 節で例としてあげた/u-tokyo.ac.jp/videos/hoge.mp4 のような Name の場合，/u-tokyo.ac.jp/ -> faces 1,2 のような Name に一部マッチする FIB エントリがあれば，その face に Interest packet を転送する．転送先でさらにマッチする/u-tokyo.ac.jp/video/ -> faces 1 のような FIB エントリを持つ NDN router がいれば徐々にコンテンツ提供者に近づいていくルーティングが実現される．このルーティングの途中で，FIB に受信した Interest packet の Name にマッチするエントリが全く存在しない場合は，Data packet の所在の手がかりを一切知らないということなので，Intrest packet を破棄する．なお，FIB は IP の代わりに名前を対象としたルーティングテーブルであり，これ以降同じデータに対する Interest packet を別のインタフェースから再び受信したときには，PIT エントリに対象インタフェースを追記するだけで転送は行わない．この FIB を構成するための適当なルーティングプロトコルとしては，OSPF [28] や BGP [29] の拡張が初期実装として提案されているが，将来的には IP forwarding table への手法 [30] を適用してローカルな FIB をアグリゲーションすることや，名前空間などを用いてよりスケーラブルなルーティングを行うことを目指して現在も研究が続けられている．

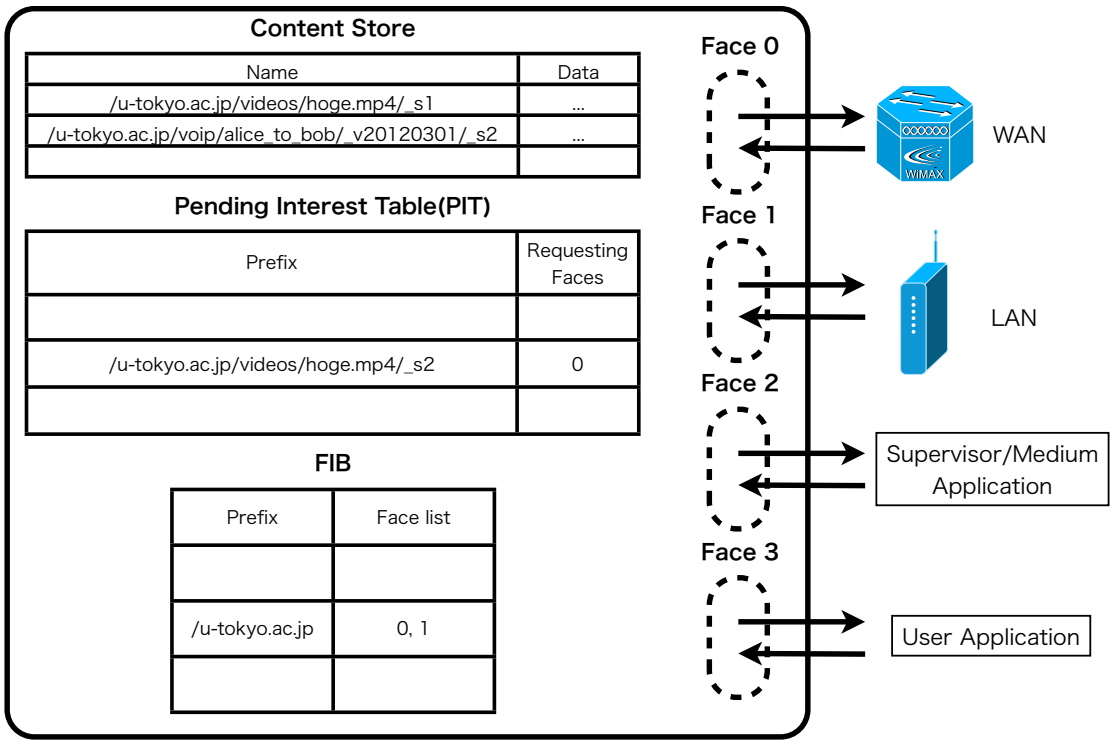

図 3.2: NDN router のモデル

コンテンツ提供者の応答

コンテンツ提供者は NDN router から Interest packet を受信し，マッチする Data packet を返信する．Data packet はコンテンツ提供者の持つ公開鍵で署名された名前とデータ本体で構成される．なお，コンテンツ提供者は自らの持つ Data packet を公開したい場合には事前に何らかの方法で Data packet の名前を告知する必要がある．この方法は FIB の構成や実装にもよるが，例えば名前のリポジトリを使う場合はリポジトリに対して登録作業を行えばよい．

NDN router の Data packet 転送

NDN router は Interest packet を転送したインタフェースから Data packet を受信した場合，PIT に記録されている過去に Interest packet を受信した全てのインタフェースに Data packet を転送し，PIT から対象エントリを削除する．同時に，Content Store に適当なキャッシュポリシーに従って Data packet をキャッシュし，以降はキャッシュした Data packet に対する Interest packet を受信した場合，コンテンツ提供者の代わりに Data packet を返信する．これを繰り返すことで Interest packet が転送されてきたパスを Data packet が逆順に辿っていくことになるので，各ホップで Interest packet と Data packet が

1 対 1 で存在することになる．そのため Interest packet の転送量をコントロールすることで，各ホップでフローコントロールが可能である．

利用者の受信

利用者は NDN router から望むデータの Data packet を受信する．なお，Interest packet 転送途中で FIB エントリが全くマッチしなければ Intrest packet は破棄される (3.2.1 節) ので，Data packet を得られず，依然としてデータ取得を望む場合，利用者は適当に設定されたタイムアウト時間の後 Interest packet を再送する．

3.2.2　NDN のセキュリティについて

NDN で実現されるセキュリティ

2.1 節の通り，NDN において Data packet は署名が義務付けられているので，ユーザはコンテンツ提供者から直接ではなく，近くの NDN router から Content Store にキャッシュされた Data packet を受信した場合でも，署名の検証を行うことで改ざんの検知が可能である．この署名については，全 Data packet について利用者が検証する必要があるので，検証を高速に実行可能な実装の検討が進んでいる．また，Data packet のデータ本体を暗号化することで，VoIP や IM などのアプリケーションを秘密に運用することも可能である．

これらのセキュリティを実現するためには，署名や暗号化に用いる公開鍵や証明書をどう運用するかが問題となる．NDN では暗号化することもできる署名された Data packet を，そのまま公開鍵や証明書として安全に通信することが可能なため，アプリケーションから様々な方法で公開鍵などを運用することができる．運用する方法としては，階層的 PKI [31]，PGP による Web of Trust [32] や SDSI/SPKI [33–35] が検討されている．また，鍵や証明書の失効については，CRL や OCSP [36] を用いることができるが，NDN では公開鍵なども Data packet として扱い，Content Store にキャッシュされるため，既存 IP ネットワークで現在用いられているものより非効率になる．この点についても研究が続いている．

コンテンツ管理

NDN では P2P ネットワークと同様に，それぞれの NDN router が Content Store に過去に通信したコンテンツをキャッシュするので，ネットワークにコンテンツが分散する．そのため，法的に問題のあるコンテンツを削除することなど，コンテンツ管理が難しいという問題がある．こうした問題に対して，NDN では Content Firewall と呼ばれる，各 NDN router で特定の名前の Packet を捨てるなどの Policy を強制する仕組みが検討されている [26]．

NDN に対する直接的攻撃

現在考えられている NDN に対する直接的な攻撃は，Interest Flooding Attacks と Content Pollution Attacks である．Interest Flooding Attacks は，既存 IP ネットワークで DDoS 攻撃に相当する攻撃で，Interest packet を過剰に送信する攻撃である．送信される Interest packet が実在するデータの Name を含んでいれば，前述した各ホップでのフローコントロールにより攻撃にはならないが，実在せず，しかも階層化によるアグリケーションが困難な Name の場合は，特定の NDN router に負担を集中させることが可能である．しかしそのような特定の Name に対する Interest packet は決して Data packet の応答を得ることができないため，そうした不審な Interest packet は中継する NDN router が Content Firewall で排除可能である．なお，NDN では Data packet は Interest packet への返信としてしか送信されないため，Data Flooding Attacks は成立しない．Content Pollution Attacks は Interest packet に対応する Data packet であると偽って不正な Data packet を返信する攻撃である．これについては基本的に利用者や NDN router が署名の検証を行えば防ぐことができるが，フィルタリングによってこうしたコンテンツを利用者側で排除する場合の負荷についても研究が続いている．

3.3　コンテンツ管理システムの設計例

3.3.1　前提要件

NDN router はプログラマブルな FIB が理想的に機能し，そうした FIB を Strategy Layer から用いることで，該当する NDN router への削除リクエストの拡散や，コンテンツを持つユーザが新たにネットワークに参加してきた場合の FIB 更新は適切に行われると仮定する．提案システムは，コンテンツの所有者や権利保有者などが使う User Application，全 NDN router で動作する Medium Application，Supervisor が用いる Supervisor Application の 3 種類から構成され (図 3.3)，各 NDN router の face と接続されて動作する (図 3.2)．Supervisor Application はコンテンツ管理の可否を判断するため，信頼できる NDN router に接続する必要があり，削除許可 D_{per} 発行専用秘密鍵 *key.sec.super* を持つ．そして Medium Application と User Application は，削除リクエスト D_{per} 検証専用公開鍵 *key.pub.super* を持つ．なお，以降，*Alice* の公開鍵と秘密鍵をそれぞれ *key.sec.alice*，*key.pub.alice* と表し，適当な公開鍵暗号 E とハッシュ関数 H を用いて，*key* を用いたコンテンツ C の暗号化を $E_{key}(C)$，コンテンツハッシュを $H(C)$，コンテンツの名前文字列を C_{name} のように記述する．

3.3.2　Content Firewall を用いた設計例

以下に，NDN router に備わる Content Firewall(3.2.2 節) を用いて，コンテンツ管理として最も一般的な削除を行う場合の実装について，削除のリクエストとその実行の順にそ

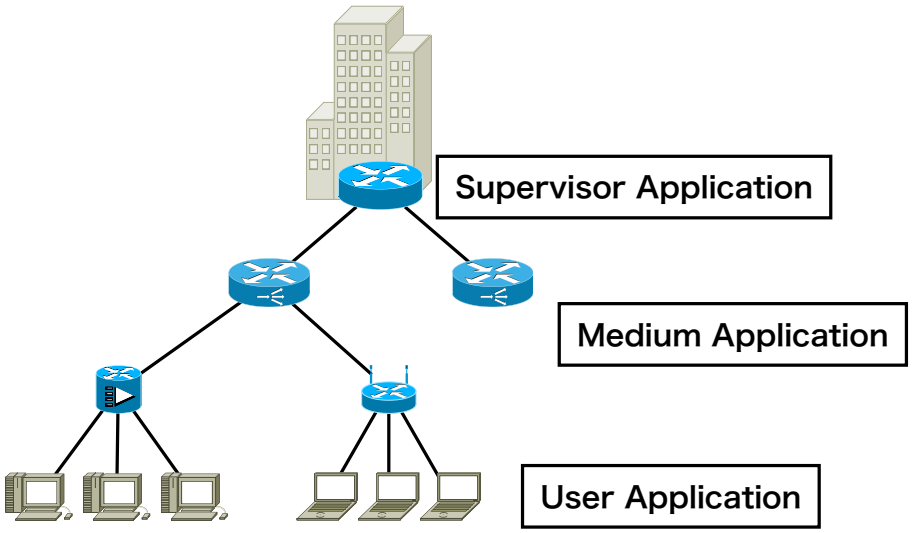

図 3.3: 提案ネットワーク構成

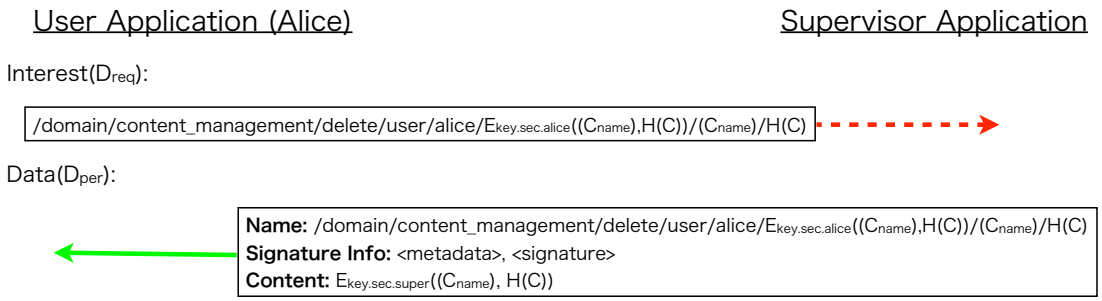

図 3.4: 削除のリクエスト

の動作を示す.

削除のリクエスト

あるコンテンツ C の削除を望む C の所有ユーザ $Alice$ は，User Application を用いて C の削除許可 D_{per} を得るために，$Alice$ の秘密鍵 $key.sec.alice$ を用いた署名 $E_{key.sec.alice}((C_{name}), H(C))$ を含む Interest packet D_{req}(図 3.4) を発行し，D_{req} を適当に更新されている FIB に基づいた face に送信する．なお，C が存在しない場合は，適当な NDN router で D_{req} は破棄されるので，D_{req} を用いた Interest Flooding Attacks は成立しない.

次に，下流 NDN ネットワークの face から D_{req} を受信した Supervisor Application は，$Alice$ が所有ユーザであるか検証するために，D_{req} の署名が C の署名と一致するか $Alice$ の公開鍵 $key.pub.alice$ を用いて検証する．この検証の結果，D_{req} の署名が C の署名と一

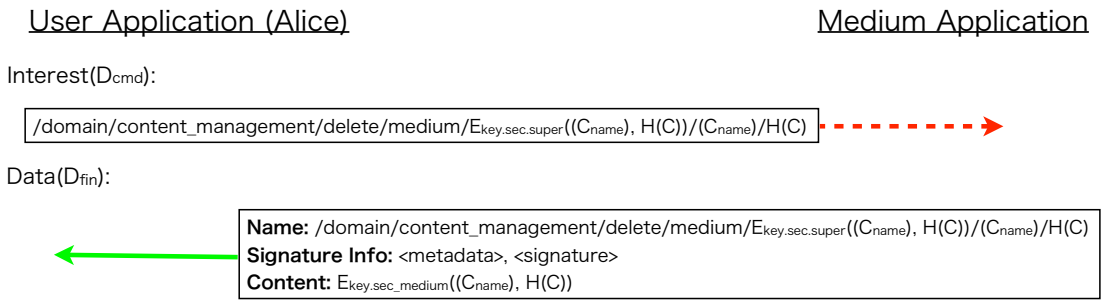

図 3.5: 削除の実行

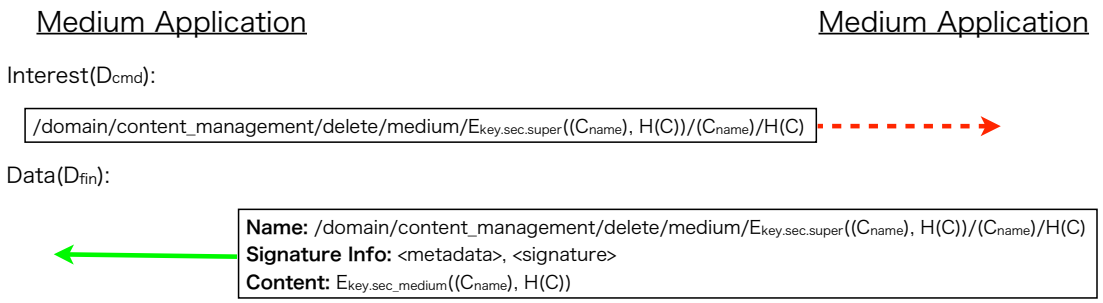

図 3.6: 削除の拡散

致すれば確かに *Alice* は C の所有ユーザなので，削除許可 D_{per} を Data packet として返信 (図 3.4) し，一致しなければ所有者以外の不正なリクエストなので D_{req} を破棄する．なお，D_{per} の Content は Supervisor Application のみが持つ D_{per} 発行専用秘密鍵 *key.sec.super* で C の名前や $H(C)$ を暗号化した署名である．

削除の実行

D_{per} を Data packet として受信した *Alice* は，User Application があらかじめ持つ D_{per} 検証専用公開鍵 *key.pub.super* で D_{per} の Content を復号し C の Name 文字列やハッシュが得られるかどうかを確認することでその正当性を検証する．正しいことが確認できたならば D_{per} の Content を Content Name に含む Interest packet D_{cmd} を FIB に基づいて Medium Application にブロードキャストする (図 3.5)．

Medium Application は，D_{cmd} を受信すると，まず D_{cmd} に含まれる D_{per} を D_{per} 検証専用公開鍵 *key.pub.super* で User Application と同様に検証する．正しいことが確認できたなら C を削除対象として Content Firewall の Policy(3.2.2 節) に追加し，Content Store に存在する C の Data packet をすべて削除する．そして Medium Application に接続された NDN router の FIB テーブルに存在する face のうち，自らの Medium Application が

接続された face と D_{cmd} を受信した face 以外に D_{cmd} を送信することで D_{cmd} を拡散する (図 3.6)．そして D_{cmd} を送信した全ての face から削除完了を示す Data packet D_{fin} が返信されるか，Interest packet が途中で破棄された場合のタイムアウト時間を過ぎたら，自らも署名を付加した D_{fin} を最初に D_{cmd} を受信した face に返信する (図 3.5, 3.6)．そして，これ以降は Content Firewall の Policy に従って C に関連する Interest packet や FIB の更新を無視する．なお，一度受信した Interest packet を，Medium Application で処理した後に他 face に転送するという，チェインメール的な動作をする機能については，原著の論文 [26] では言及されていない．そのためセキュリティ上問題がないか慎重に検討する必要がある．この点については 3.5.1 節で考察する．

Alice は D_{fin} を受信できれば，D_{cmd} がネットワーク全体で適当に拡散したということなので，コンテンツの削除状況を確認することができる．そして D_{cmd} の受信によりネットワーク全体の Medium Application で Content Firewall の Policy が更新されることから，新たに C を持つ他ユーザが出現した場合でも，各 NDN routet の Content Firewall で C の流通を防ぐことができる．なお，*Alice* が所有ユーザではないが，Stakeholder ではあった場合には，裁判や証明書類の郵送など何らかの外部手段を用いて自らが Stakeholder であることを Supervisor に証明し，Supervisor に直接 D_{cmd} の発行を依頼すれば同様に削除が可能である．そのために Supervisor Application は D_{req} を受信せずとも D_{cmd} を発行する機能も持たなければならない．

3.4　コンテンツ管理システムの運用

Supervisor Application と Medium Application が信頼できれば 3.3 節の設計例でコンテンツ管理が実現できる．しかし既存 IP ネットワークの置き換えとして NDN が広く普及する前の過渡期においては，利用者による必ずしも信頼できない NDN router が数多くあると考えられるため，ISP や ASP が信頼できる NDN router に Medium Application を十分な数設置することは難しいと考えられる．こうした場合，少数の Supervisor Application を持つ NDN router に多数の利用者が接続することになり，既存 IP ネットワークのサーバによる配信と変わらなくなってしまうため，信頼できない NDN router にも Medium Application を設置する必要がある．しかし，Medium Application が信頼できない場合は 3.3 節の設計例では D_{cmd} を無視されたらコンテンツ管理は不可能である．そのため，我々が過去に提案したそのような信頼できないネットワークでコンテンツ管理を行う手法 [37] を用いたシステムを提案する．以下に，Secret Sharing を用いることで，*Alice* が C を公開し，*Bob* が C を利用する時の手順と，*Alice* が C を削除する際の手順を示す．なお，以下に登場する公開情報，シャドウ，特有公開情報，疑似シャドウやネットワークの規模や悪意のあるノードの数に応じて決まる閾値 k や n について，詳細や生成方法については紙面の都合上割愛する．

3.4.1　コンテンツ公開

1. 何らかのコンテンツを NDN で公開する前に，*Alice* は自らの環境から生成した乱数情報などを用いて，公開情報と n 個のシャドウ $S_j(1 \leq j \leq n)$ を生成する．

2. *Alice* は公開情報とシャドウ $S_j(1 \leq j \leq n)$ を *Alice* の秘密鍵 $key.sec.alice$ で暗号化して NDN に公開する．ここで，公開情報は一般的な NDN と同じくルーティングアナウンスするだけ (3.2.1 節) でよいが，シャドウはネットワークに分散して保持されなければならない．そのため，一方的に Data packet を特定の host に送りつけることができない (そもそも特定の host が存在しない) 通常の NDN では，送信先である Medium Application によるサポートが必要である．これについては後述する．

3. *Alice* は公開したいコンテンツ C を *Alice* の持つ NDN で用いる $key.sec.alice$ とは異なる C 専用暗号鍵 $key.sec.alice.c$ で暗号化 $E_{key.sec.alice.c}(C)$ して NDN で公開

4. *Alice* は C 専用公開鍵 $key.pub.alice.c$ の特有公開情報 $SPI_{key.pub.alice.c}$ を生成して NDN で公開

5. C を利用したい *Bob* は，NDN で公開されている $E_{key.sec.alice.c}(C)$，*Aliice* の公開情報と $SPI_{key.pub.alice.c}$ を取得する．

6. *Bob* は $SPI_{key.pub.alice.c}$ を含み，シャドウ $S_j(1 \leq j \leq n)$ に対応する Interest packet $I_{PS.j}$ を送信する (図 3.8)．

7. シャドウ S_j を持つ Medium Application は $I_{PS.j}$ を受信し，含まれる $key.pub.alice.c$ の特有公開情報と S_j から疑似シャドウ PS_j を生成し，対応する Data packet $D_{PS.j}$ として暗号化して返信する (図 3.8)．

8. *Bob* は設定された閾値 k 個以上の $D_{PS.j}$ を受信できれば，Multiple Secret Sharing を用いて *Alice* の公開情報，$key.pub.alice.c$ の特有公開情報と k 個以上の疑似シャドウ PS_j から $key.pub.alice.c$ を復元し，$E_{key.sec.alice.c}(C)$ を復号して利用することができる．

シャドウをネットワークに分散して保持する方法として検討している手法のひとつを以下に説明する．まず，Medium Application は一定間隔でランダムに変更する ID を持つ．そして利用者はシャドウを送信するとき，シャドウを分散させたい ISP の名前 ($ISPname$) とランダムな ID($random\_ID$) を含む Interest packetSS_{req} を送信する (図 3.7)．利用者がランダムに選んだ ID と，Medium Application がその時設定している ID が一致すれば，Interest packet はその Medium Application に届き，シャドウの送信が可能であるという応答として Data packetSS_{res} を返信する．そして，Medium Application は続けてシャドウに対する Interest packetRS_{req} を送信し，利用者はこの Interest packet に対する Data packetRS_{res} を送信することでシャドウの送信を完了する (図 3.7)．このランダム

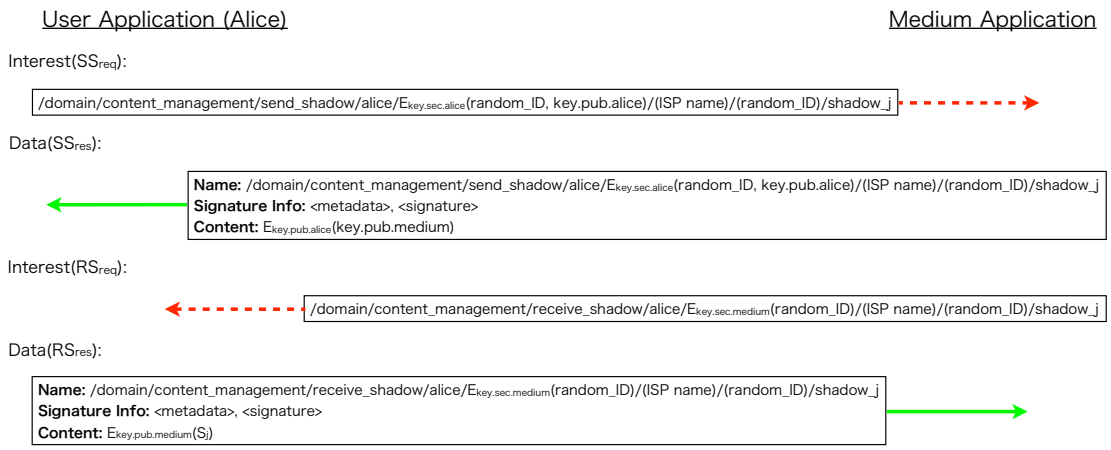

図 3.7: シャドウの分散

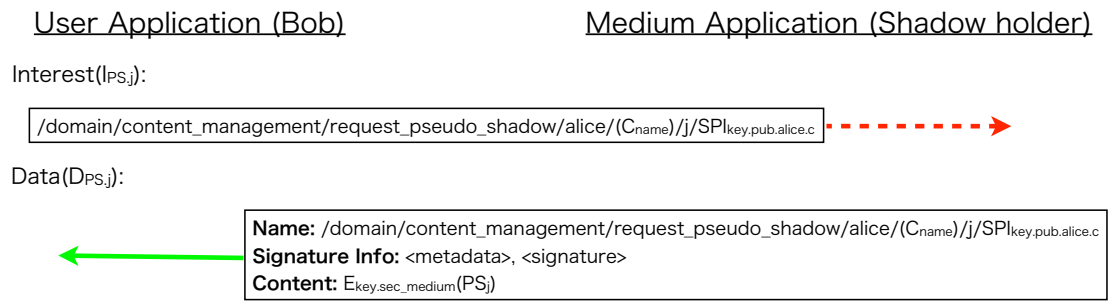

図 3.8: 疑似シャドウの取得

ID を割り当てる名前空間をどの程度の大きさ N にするかは，各 ISP に存在する Medium Application の数 l によってあらかじめ決めておかなければならない．理想的には $N = l$ で，この場合確実にシャドウの送信が可能であり，かつ各シャドウを所持する Medium Application は平均ひとつとなる．$N < l$ だった場合は，同様にシャドウの送信は確実に可能だが，各シャドウを所持する Medium Application が複数になり，3.4.2 節で後述する閾値 k を大きくしなければならない．$N > l$ だった場合には，各シャドウを所持する Medium Application はただ一つだが，選択したランダム ID が Medium Application と一致せず，送信に失敗する可能性がある．送信に失敗した場合は，Interest packet が途中で破棄されるので，通常の NDN と同様に再送信をせねばならない．なお，D_{cmd} の拡散と同じくこのような一方的データ送信はセキュリティ上慎重に検討する必要がある．この点については 3.5.2 節で後述する．

3.4.2　コンテンツ削除

1. *Alice* は *key.pub.alice.c* の特有公開情報を設計例を用いて削除する．
2. *Alice* は，疑似シャドウ PS_j に対して図 3.5 の設計例と同様に D_{cmd} を送信する．D_{cmd} を受信した Medium Application は，Content Store にある PS_j を削除し，Content Firewall Policy を更新することで以降の PS_j 生成を行わない．ここで無視する Medium Application がいても，$k-1$ 台の Medium Application が応じてくれれば疑似シャドウからの *key.pub.alice.c* 復元は計算量的に困難になるため，$E_{key.sec.alice.c}(C)$ の復号，つまり C の利用を阻止できる．
3. *Alice* が自らの公開しているコンテンツ全てを削除したい場合，*Alice* の公開情報とシャドウを同様に削除する．

3.5　提案システムのセキュリティに関する考察

3.5.1　D_{cmd} の拡散におけるセキュリティ

まず懸念されるのは，Interest packet である D_{cmd} を Flooding させることによる NDN ネットワークへの負荷である．しかし，Medium Application が D_{cmd} を転送した後に同じコンテンツ C に対する D_{cmd} を受信しても PIT に追記するだけなので，D_{cmd} はひとつの face に一度しか転送されない．さらに Data packet である D_{fin} も C の名前やハッシュを *key.sec.medium* を用いて暗号化しただけなので Interest packet と同程度に軽量であると考えられる．また，Flooding の範囲も D_{cmd} の Name に含まれる domain に限定される．そのため，D_{cmd} の Flooding 一度当たりの NDN ネットワークへの負荷は少ない．次に懸念されるのは，D_{cmd} は誰でも送信可能であることを Interest Flooding Attacks に悪用される可能性である．しかし，D_{cmd} を悪意のある攻撃者が大量に送信しても，D_{fin} の返信を待っている Medium Application が受信した場合は PIT に追記し，本来返信する必要のある face に D_{fin} を返信する時に D_{fin} を一度のみ攻撃者に対して返信するだけである．また，一度 D_{fin} を返信した Medium Application が攻撃者からの D_{cmd} を受信した場合も，Content Store にキャッシュされた D_{fin} を一度のみ攻撃者に対して返信するだけである．そのため通常の NDN の枠内のセキュリティが保たれると考えられる．

3.5.2　シャドウの送信におけるセキュリティ

存在しない Name に対する Interest packet による Interest Flooding Attacks は成立しない (3.2.2 節) が，異なる種類のシャドウに対する大量の SS_{req} を，任意のひとつの $(random\_ID)$ に送信すれば，SS_{res} の応答を得られてしまうため，Medium Application が次の $(random\_ID)$ を選択するまでの間は Interest Flooding Attacks が成立してしまう．そのため $(random\_ID)$ の更新間隔を短くすることや，Content Firewall で SS_{req} の Name

に含まれる /*domain*/*content_management*/*send_shadow*/ の転送を一定数に制限することが必要と考えられる.

3.6　おわりに

本稿では次世代インターネットアーキテクチャとして注目されている NDN において，コンテンツ管理アプリケーションの設計例を提案し，必ずしも信用できない NDN router が多く存在する状況で，Multiple Secret Sharing を用いたコンテンツ管理法を提案し，そのセキュリティについて定性的な考察を行った.

第4章

結論

4.1　本研究の主たる貢献

本研究では，一般の人々，ISP，ASP の三者に利益のあるコンテンツ共有システムを実現するために，特にコンテンツ配信において問題となるコンテンツ管理を行うための手法を提案した．まず，一般の人々と ASP に低コストでコンテンツ共有をもたらすため，現在において実用的である P2P アプリケーションによるオーバレイネットワークでコンテンツ管理を行う手法を提案した．コンテンツ復号鍵を Multiple Secret Sharing を用いて分散して共有することで，一定の閾値内のノードが何らかの理由でコンテンツ管理命令に従わなかった場合でも，コンテンツ復号鍵の復元を計算量的に不可能にできる点に新規性があり，そのようなシステムの提案を行ったという点に貢献がある．

そうしたシステムが，ノードの振る舞いが複雑な P2P ネットワークで動作するかを評価するために，ノードのアクセスパターンやコンテンツの人気の偏りも考慮したモデルを立て，それに基づいたシミュレータを設計し，シミュレーションを行い，適当な閾値が存在しシステムが動作することを明らかにした．

しかし P2P ネットワークはオーバーレイネットワークであるため，下の層である ISP の物理トポロジを考慮したコンテンツ配信は難しい．そこでさらに，一般の人々と ASP に加えて，ISP にも利益になるコンテンツ共有システムの実現のために，現在も研究が続けられている次世代インターネットアーキテクチャである Named Data Networking(NDN) においてコンテンツ管理を行う手法を提案した．NDN ではコンテンツ管理を行うための手段として Content Firewall という機能は考案されているが，コンテンツ管理システムは存在しなかった．そのため，Content Firewall を用いて実際にコンテンツ管理アプリケーションを設計した点と，さらに P2P と同様に Multiple Secret Sharing を用いることで，一定の閾値内のノードが信用できない状況でもコンテンツの不正利用を防ぐことができる点に新規性があり，そのようなシステムの提案を行ったという点で貢献がある．

4.2　今後の課題

本研究において解決されていない課題としては以下のものがある．

P2P における実装

本論文では，サーバレスな P2P として一般的な DHT である Kademlia による P2P において提案手法が実際に不正利用を防ぐことができるかどうか，シミュレーションによる検討しか行っていない．そのため，提案手法に加えて，実際に不正なノードをブラックリストなどで排除する仕組みや，提案手法の閾値などのパラメータを決定する仕組みなどを実装する必要がある．

NDN における実装

本論文では，Content Firewall という NDN の機能を用いたコンテンツ管理アプリケーションの設計のみしか行っておらず，その実装は現在もオープンソースで開発が続いている NDN 本体の実装に合わせて，ローカルアプリケーションの形で実現する必要がある．

謝辞

本研究および修士論文の執筆にあたり常に親身にご指導くださり，学会や研究会での発表の機会を多く与えてくださいました浅見徹教授に心から感謝いたします．さらに本研究へのご指導をはじめ，その他研究生活全般においてさまざまな場面での助言をくださいました川原圭博講師にお礼申し上げます．

また，研究生活のサポートのみならず，日々の研究室生活でお世話になりました浅見・川原研究室の秘書の安藤さんならびに学生のみなさまに感謝いたします．最後に，日々の生活を支えてくれた家族に心から感謝いたします．

参考文献

[1] Cisco Systems, “Cisco visual networking index: Forecast and methodology, 2009-2014,” 2010. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/
ns537/ns705/ns827/white_paper_c11-481360.pdf

[2] Cisco Systems, “Hyperconnectivity and the approaching zettabyte era,” 2010. http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/VNI_Hyperconnectivity_WP.pdf

[3] Google, “Youtube,” 2010. http://www.youtube.com

[4] ニワンゴ, “ニコニコ動画 (9),” 2010. http://www.nicovideo.jp/

[5] 山下達也, “インターネット・トラヒック最新状況,” 2010. http://www.ntt.com/interop/seminarpdf/08.pdf

[6] S. Androutsellis-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” ACM Computing Surveys, vol.36, no.4, pp.335–371, Dec. 2004. http://portal.acm.org/citation.cfm?doid=1041680.1041681

[7] Bitmedia and ANCL, “Sharecast,” 2010. http://scast.tv/sc2plus/index.html

[8] ウタゴエ株式会社, “UG Live,” 2010. http://www.utagoe.com/jp/index.html

[9] C. Labovitz, D. McPherson, and S. Iekel-Johnson, “Internet Observatory 2009 Annual Report,” The North American Network Operators, pp.1–32, 2009. http://www.nanog.org/meetings/nanog47/presentations/Monday/Labovitz_ObserveReport_N47_Mon.pdf

[10] Microsoft, “[MS-DRM]: Digital Rights Management License Protocol Specification,” 2010. http://download.microsoft.com/download/9/5/E/95EF66AF-9026
-4BB0-A41D-A4F81802D92C/[MS-DRM].pdf

[11] V. Networks, “Veoh video network,” 2010. http://www.veoh.com/

[12] PPLive Inc., “Pptv,” 2010. http://www.pptv.com/

[13] 王　亮，川原圭博，浅見　徹，“P2P 動画共有システムにおけるコンテンツ不正利用の低減手法とその評価,” 電子情報通信学会技術研究報告. IN, 情報ネットワーク, vol.108，no.458，pp.103–108，2009-02-24. http://ci.nii.ac.jp/naid/110007324722/

[14] Y. Kawahara, L. Wang, and T. Asami, “Resilient Suppressor Mechanism against Illegal Content Redistribution on Peer-to-Peer Video Sharing Networks,” Communications (ICC), 2010 IEEE International Conference onIEEE, pp.1–6 2010.

[15] A. Rowstron and P. Druschel, “Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems,” Middleware 2001Springer, pp.329–350 2001.

[16] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, and H. Balakrishnan, “Chord: A scalable peer-to-peer lookup service for internet applications,” Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communicationsACM, pp.149–160 2001.

[17] P. Maymounkov and D. Mazieres, “Kademlia: A peer-to-peer information system based on the xor metric,” Peer-to-Peer Systems, pp.53–65, 2002.

[18] A. Shamir, “How to share a secret,” Communications of the ACM, vol.22, no.11, pp.612–613, 1979.

[19] G. BLAKLEY, “Safeguarding cryptographic keys,” National Computer ConferenceAFIPS Press., p.313 1979.

[20] 首藤一幸，田中良夫，関口智嗣，“オーバレイ構築ツールキット Overlay Weaver,” 情報処理学会論文誌: コンピューティングシステム，vol.47，no.12，pp.358–367，2006.

[21] 秋丸春夫，川島幸之助，情報通信トラヒック―基礎と応用,（社）電気通信協会，東京，1990.

[22] Y. Huang, T.Z. Fu, D.-M. Chiu, J.C. Lui, and C. Huang, “Challenges, design and analysis of a large-scale p2p-vod system,” SIGCOMM Comput. Commun. Rev., vol.38, pp.375–388, Aug. 2008. http://doi.acm.org/10.1145/1402946.1403001

[23] J. He and E. Dawson, “Multistage secret sharing based on one-way function,” Electronics Letters, vol.30, no.19, pp.1591–1592, 2002.

[24] D. Liu, D. Huang, P. Luo, and Y. Dai, “New schemes for sharing points on an elliptic curve,” Computers & Mathematics with Applications, vol.56, no.6, pp.1556–1561, 2008.

[25] H.S. Lee, “A self-pairing map and its applications to cryptography* 1,” Applied Mathematics and Computation, vol.151, no.3, pp.671–678, 2004.

[26] V. Jacobson, D.K. Smetters, J.D. Thornton, M.F. Plass, N.H. Briggs, and R.L. Braynard, “Networking named content,” Proceedings of the 5th international conference on Emerging networking experiments and technologies - CoNEXT ’09, p.1, ACM Press, New York, New York, USA, 2009.

[27] L. Zhang, D. Estrin, J. Burke, and V. Jacobson, “Named data networking (ndn) project,” 2010. http://www.named-data.org/ndn-proj.pdf

[28] J. Moy, “OSPF Version 2,” Technical report, RFC2328, April 1998.

[29] Y. Rekhter, T. Li, and S. Hares, “A Border Gateway Protocol 4(BGP-4),” Technical report, RFC4271, Jan. 2006.

[30] Xin Zhao and Yaoqing Liu and Lan Wang and Beichuan Zhang, “On the Aggregatability of Router Forwarding Tables,” Proc. of the 29th conference on INFOCOM, pp.1–9, INFOCOM’10, March 2010.

[31] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, “Internet X. 509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile,” Technical report, RFC5280, May 2008.

[32] P.R. Zimmermann, The Official PGP User’s Guide, MIT Press, 1995.

[33] M. Abadi, “On SDSI’s Linked Local Name Spaces,” Journal of Computer Security, vol.6(1-2), pp.3–21, 1998.

[34] R. Rivest, B. Lampson, C.M. Ellison, B. Frantz, and S. Bell, “SPKI Certificate Theory,” Technical report, RFC2693, Sept. 1999.

[35] R.L. Rivest and B. Lampson, “Sdsi-a simple distributed security infrastructure,” Technical report, MIT Technical Report, 1996.

[36] M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol-OCSP,” Technical report, RFC2560, June 1999.

[37] 楠　慶，川原圭博，浅見　徹，山口和彦，“P2P 動画共有システムにおける Multiple Secret Sharing を用いたコンテンツ不正利用の低減手法,” 電子情報通信学会論文誌. B, 通信，vol.94，no.10，pp.1270–1282，2011-10-01.

発表文献

論文誌

[P1] 楠慶，川原圭博，浅見徹，山口和彦, “P2P 動画共有システムにおける Multiple Secret Sharing を用いたコンテンツ不正利用の低減手法, ” 信学論 B，Vol.J94-B,No.10, pp. 1270-1282, Oct. 2011

研究会

[P2] 楠慶, 川原圭博, 浅見徹, 山口和彦, “P2P 動画共有システムにおける Multiple Secret Sharing を用いたコンテンツ不正利用の低減手法, ” 信学技報，Vol.110, No.128, MoMuC2010-23, pp.57-62, July 2010

[P3] 加藤拓也，楠慶，川原圭博，浅見徹，“通信量削減によるスペクトラム情報可視化システムの性能改善，” 信学技報，Vol.110, No.449, IN2010-158, pp.85-90, March 2011.

[P4] 水谷昌彦，楠慶，川原圭博，浅見徹，“分断されたアクセス網における自律分散型認証技術の検討，” 信学技報，Vol.111, No.409, IN2012-123, pp.17-22, Jan 2012.

全国大会

[P5] 楠慶，川原圭博，浅見徹，山口和彦，“P2P 動画共有システムにおける Multiple Secret Sharing を用いたコンテンツ不正利 用の低減手法の評価，” 信学ソ大，B-7-30, Sept. 2010

[P6] 加藤拓也, T. T. Quang, 楠慶, 川原圭博, 浅見徹，”スペクトル情報可視化システムの設計と実装，” 信学ソ大，B-7-39, Sept. 2010.

[P7] 加藤拓也，楠慶，川原圭博，浅見徹，”スペクトラム情報可視化システムの性能改善手法と評価，” 信学総大，B-7-13, March 2011.