M-051

管理主体の異なるサービス群を安全に連携可能にする機構
A Secure Extension for Multi-Institutional Ubiquitous Service Systems

榎堀 優 † Yu Enokibori
谷川 善紀 † Yoshinori Tanigawa
西尾 信彦 ‡ Nobuhiko Nishio

滋賀県草津市野路東 1-1-1
Tel: +81 77 561 2741
vori@ubi.is.ritsumei.ac.jp nori@ubi.is.ritsumei.ac.jp nishio@cs.ritsumei.ac.jp

概 要

現在, 多くの組織や個人が様々なサービスを構築している. しかし, たとえネットワーク経由でそれらが接続されていたとしても, 双方を連携して動作させた例は少ない. その原因の一つとして, 正規利用者の管理主体が異なる複数のシステムを適正な権限にて安全に連携可能にする機構が存在しないことが考えられる. また, 個人が持ち歩いているモバイル機器を訪問先などのサービスと連携させて利用したい場合も同様の問題が発生する. 本稿では, Kerberos をベースにサービス提供機構を拡張し, 管理主体の異なるサービスシステムを空間という単位に捉え, 空間間のセキュリティと関係を定義することにより, サービス群と正規利用者情報を各組織・個人単位で管理しながら, それらを安全に連携可能にする機構を提案する.

Abstract

Today, many organizations and individuals are creating a large variety of services. However, even if these services are connected through a network, there are very few examples where such services operate in an interconnected way. One reason for this is the lack of systems that are able to coordinate multiple systems, with different schemes for user management, in a safe way with adequate authorization. Furthermore, the same problem arises when users carrying mobile terminals wishes to connect to and use services at location that they are visiting. In this paper, we are proposing an extended framework for service provision based on Kerberos, allowing groups of services and information about ordinary users that are managed on an organizational or personal level to be combined, handling service systems with different management bases as units of “ Space ”, while defining the security relations between different spaces.

1. はじめに

現在，多くの組織や個人が様々なサービスシステムを構築している．ネットワーク経由でそれらが接続されているならば，それらを連携させることにより，より利便性のあるサービスシステムを構築することができるはずである．例えば自身のモバイル端末を訪問先のサービスシステムと連携させ，モバイル端末中のファイルを出先のプレゼンテーションサービスで利用するといったことや，会社の部署ごとにサービスシステムを構築し相互連携させることで，管理責任を分散させながら会社全体のサービスシステムとして構築すること等が考えられる．しかし，実際に複数のサービスシステムを連携して動作させた例は少ない．

その原因のひとつとして利用者を管理している団体（個人）が異なる組織であるので，相互をセキュアに利用する必要があるためと考えている．その代表的な例としては，他の組織に訪問し，そこのサービスシステムに自身が持ち歩いているモバイル端末を接続して利用したい時の認証問題などがある．

分散ファイルシステムの世界でこれらを解決したものの一例として AFS[1] が存在する．われわれは本稿にて，ユビキタスサービスシステムの世界において同様の問題を解決し，各自が所持するサービスシステムやモバイル端末などを相互にセキュアに接続することを可能にする機構を提案する．

以降，第二章では既存技術と本稿の内容との関係について述べ、第三章にてセキュアなシステムを実現するために利用したアルゴリズムと機構，**空間**という単位とその運用方法について．第四章にてユーザ管理機構，第五章にてサービス管理機構，第六章にて空間全体の管理と空間間の動作である**結合**と**融合**の二つについて述べる．最後に第七章にて結論を述べ，まとめる．

†立命館大学理工学部情報学科
Department of Computer Science, Ritsumeikan University
‡立命館大学情報理工学部
Department of Computer Science, Ritsumeikan University

2. 既存技術について

ユビキタスサービスを提供する空間を実現するためのシステムとしては University of Illinois at Urbana-Champaign の GAIA プロジェクト [2]，慶應義塾大学の徳田らによる Smart Space Laboratory[3]，名古屋大学の宮尾・河口らによる Cogma Project[4]，東京大学の青山・森川らによる STONE ルーム [5] などがある．また，複数のユビキタス空間を連携させて動作させる実験としても前述の STONE ルーム－ Smart Space Laboratory 間ですでに実施されている．しかし，これらは利用者の権限や通信経路などのセキュリティを確保したものではなく，また，簡単に連携することが出来るものでもない．本稿が提案する技術はこれらを補完することを目的としている．

3. 構成全体について

一つの組織（個人）が管理するサービスシステムの集合を”空間”という一つの単位であるとみなし，それらの間で通信経路の安全性の確保し，相互の利用権限を定義することにより，その相互の連携をセキュアに行うことを可能にする．この”空間”は論理的単位であり，実際の部屋や建物などのような物理的制約には囚われない．

本システムの基本的なサービス利用モデルは主体がサービスを利用したい場合は始めに空間管理機構にて認証処理を行い，利用可能なサービス一覧を取得する．その後，一覧から利用するサービスを選択して空間管理機構にサービスへの接続を要求し，対象サービスとの通信を確立する．

空間とは，一主体が管理するサービスシステムの範囲を指す，その中にはユーザ情報管理とサービス情報管理が必要になる．しかし，既存のものが適用できる場合は少なく，既存のユーザ/サービス情報ディレクトリと空間の間で橋渡しをする Adapter Interface（*User Directory Adapter*（*UDA*）と *Service Directory Adapter*（*SDA*））を構築する．また，一つの組織が複数のディレクトリを併用していることも考えられるため，Adapter Interface は複数になりうる．それらを統合するための統合ディレクトリ（*Integrated User Directory*（*IUD*）と *Integrated Service Directory*（*ISD*））を構築する．最後に IUD と ISD を包含し，空間同士の関係を管理する空間管理機構（*Space Server*（*SS*））を構築する．われわれが設計した一つの管理主体による空間の構成例を図 1 に示す．

3.1 Master Key の取り扱い

空間のリソースを利用する主体としては図 1 の全要素以外に実際にサービスを利用するクライアントとサービスを提供する Service Provider が存在する．

空間を構成する全要素と Service Provider には設置時に Kerberos[6, 7, 8] の *Master Key* が設定され，通信経路はすべて Kerberos を利用して暗号化され，身分の証明と通信経路上の安全が確保されている．

もう一つの主体であるクライアントには，既存のユーザ情報ディレクトリ（NIS や LDAP）を利用する場合を考慮し，基本的には Master Key を割り振らない．

IUD にログイン認証判定をゆだねる機構を採用し，

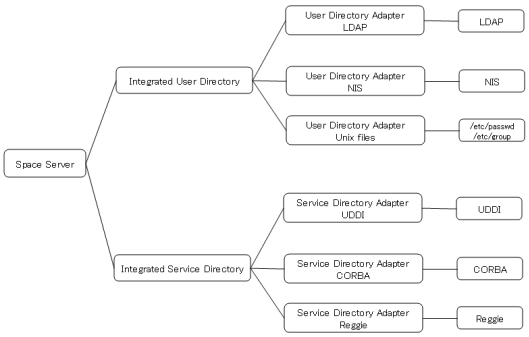

図 1: 空間構成例

Master Key の代わりとしている．

3.2 リソース利用手順

実際に空間内のリソースを利用する手順としては二種類存在する．一つは”クライアントが空間内のリソースを操作する”ことであり，もう一つは”サービスがサービスを操作（利用）する”ことが考えられる．特に後者については，基本となるサービスを利用しているクライアントの権限内にて他方のサービスを利用する場合，サービスが自身に割り振られている権限内にて他方のサービスを利用する場合の二種類が存在する．ユーザ権限で利用する場合は，実現に Kerberos に実装されている権限委譲を用いる．これにより，本来意図しない権限でのサービス利用事故が避けられる．クライアントが空間内のリソースを利用する時の基本的な動きと，サービスがサービスを利用する場合の基本的なシナリオを以下に示す．

- クライアントが空間内のリソースを利用する場合
  1. SS に認証要求を送り，認証処理を受ける．
  2. 認証処理が終了した場合，Kerberos で利用するセッション鍵を受け取る．このとき，このセッションで利用可能な権限が確定する．
  3. SS にサービス一覧取得要求を出し，SS がこれを ISD に問い合わせ返答する．
  4. クライアントは利用するサービスを特定し，サービス接続要求を SS に出し，Kerberos のセッション確立工程を経て，サービスとのセッションを確立する
- サービスがサービスを利用する場合
  1. サービスを利用する側のサービスがクライアントから利用されている場合（サービスの自律的動作ではない場合），クライアントに権限委譲処理を要求する
  2. 委譲処理が行われた場合は，委譲された権限を使用し，されなかった場合はサービス自身の権限を使用して，対象サービスへの接続要求を SS に申請する．

3. Kerberos のセッション確立工程を経て，サービスとのセッションを確立する

3.3 空間間の通信

連携したい空間はそれぞれのユーザ情報管理部に相手の空間が利用するログイン情報と権限を作成する．空間管理サーバがそのログイン権限で対象空間にログインすることにより，事前に許可されている空間同士の連携であることの証明と通信経路の暗号化を実現する．

空間間連携には対等の立場で利用する**結合**と，一方の空間がもう一方の空間に融合され，融合される空間の外部から見た場合に空間が一つ消滅するように見える**融合**がある．詳細については第六章にて詳述する．

4. ユーザ管理機構

ユーザ管理機構には各種既存ユーザ情報ディレクトリを空間に取り入れるための *UDA* と，複数の UDA から提供されるユーザ情報を統合し，一元的に利用・管理する *IUD* が存在する．

4.1 Role

空間内におけるサービスの利用権限は *Role* という単位を用いて管理する．UDA による認証が成功した場合，UDA は成功者に割り振られる Role を決定する．

4.2 Key

UDA が要求する認証情報は特定の形に限定することは出来ない．また，どの認証情報がどの認証機構（この場合は UDA）で利用されるかを明記しておく必要がある．よって，最低限 *Key* を構成する要素として下記の二点がある．

1. 使用する対象となる UDA の空間内名
2. 対象 UDA で必要となる認証情報（ユーザ名+パスワード，指紋，etc）

4.3 Key-Ring

利用者が複数のユーザ情報ディレクトリに自身の権限を保持している場合，ユーザはそれら複数の権限を同時に行使することが必要になる場合がある．逆に，必要のない権限は普段行使する権限内から除外しておきたい場合もある．

よって，Key を複数まとめ，ユーザが任意に追加・取り外しが可能な *Key-Ring* という形にして認証時に利用する．IUD は受け取った Key-Ring 内に存在する全 Key についてログインを試み，ログインに成功した UDA から取得した全 Role 情報を合計したものを以後のセッションの権限とする．

5. サービス管理機構

既存のサービスディレクトリを空間に取り入れるための *SDA* と，複数の SDA から提供される情報とサービス利用手順に統一的アプローチをもたらす *ISD* が存在する．

Service Access Control List	
Serivce	Required Role
WebDAV	Bob@UDA_name.IUD.SS_name, …
Light Control	admin@UDA_name.IUD.SS_name, …
⋮	⋮

図 2: SACL in SDA

5.1 Role Mapping Table

SDA は既存の UDDI や Jini の LookUp サーバ実装である Reggie などと実装のプロトコル差異を吸収する．また，セッション情報を保持できないサービス（HTTP を利用するものなど）にセッション管理機能を付加し Kerberos の暗号化機構が利用できないノードに対してその機能を補填する．さらに，各 SDA は図 2 のように各種サービスの実行権限と Role 間のマッピングを管理する．

6. 空間管理について

IUD と *ISD* を統合管理する *SS* が存在する．

6.1 Space Server

SS は Kerberos の KDC[§] の機能を兼ねており，ユーザとの認証受付処理，ノード間でのセッション確立，空間同士の結合・融合の管理などを行う．

6.1.1 詐称防止と認証前の安全性確保

Kerberos の仕様では現在ログインしようとしている KDC そのものが詐称されている場合に対応できない．また，ログインするまでの経路の安全性も確保することが出来ない．よって，ユーザと SS は，第三者機関による電子証明書と公開鍵暗号方式を利用して，サーバの正当性とログイン処理中の安全性を確保する．

6.2 結合

結合を目的として，空間 A が空間 B にログインした場合，空間 A のユーザ/リソースは空間 B のリソースを，ログインした空間ユーザに割り振られている Role が利用できる範囲内で利用することができる．

空間 A が空間 B にログインを試み，空間 B 内にて空間 A に Bob Role を割り振られた場合，空間 A にログインしているユーザは，空間 B を利用する権限をユーザ自身が保持していない場合でも，空間 B を Bob Role の権限で（もちろん認証なしで）利用することができる．これは，空間 A を利用中のユーザが空間 B ではすべて空間 A の Role にマッピングされることを意味する．これにより，空間 B の管理者は空間 A の権限のみを注視するだけでよいことになり，空間 A 内からの利用者すべてに対して適切な権限を割り振る必要がなくなる．また，空間 A 内からの利用者すべての認証処理を行う必要がなくなり，かつ，空間 A 以外から空間 A を利用中と詐

[§] Key Distribution center
認証とセッション暗号化のための秘密鍵を発行する．

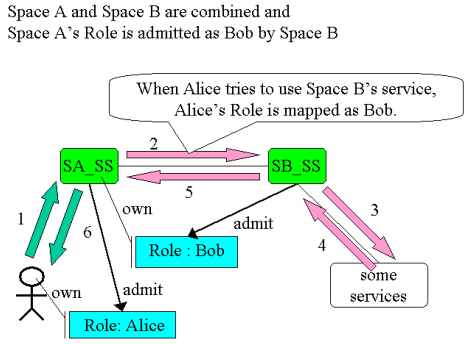

図 3: 結合時の動作例

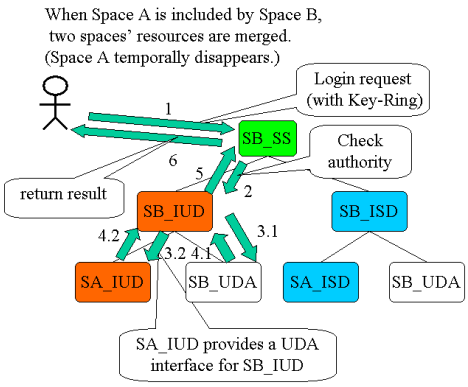

図 4: 融合時の動作例

称する要求を判別することが可能になる．このときの動作例を図 3 に示す．

しかし，このように管理者同士が事前に結合に合意し，空間ユーザが存在している場合ばかりとは限らない．このような場合，IUD では Guest を受けつける．その Guest 権限に対して，どのようなサービス実行を許すかどうかは運用ポリシー次第である．

6.3 融合

結合が対等な関係での連携を可能にするのに対し， 融合には力関係が存在する．融合を希望する空間は融合先の空間に対して自身が融合を許可されていることを証明するために空間ログイン処理を行う．ログイン処理が成功した場合， 融合される空間の ISD， IUD は融合先の空間の SDA， UDA として融合される．動きを図 4 に示す．

結合と融合の最大の違いは，自分が直接ログインしていない空間のサービスを利用する場合に，結合時は権限が空間の権限にマッピングされるのに対して，融合時は各ユーザが保持している権限のまま利用することができることにある．また，融合された空間を知らない利用者にとっては，空間が増強されたように見え，融合された空間が存在していた（している）ことがわからないことも特徴である．これにより，融合された空間のことを特に意識することなく，融合された空間のリソースを利用することができる．

7. 結論

本稿では管理主体の異なるサービスシステムを，セキュアに連携可能にする機構を提案した．複数のユビキタス環境を連携させて動作させることは既存のシステムの可能性を大幅に広げるものであり，現在もさまざまな実験や研究が進められている．この中で，複数の管理主体の異なるユビキタス空間が実現され，それらの相互接続の安全性と簡便性の向上が期待できる．

今後はネットワークセグメントやファイアウォールを越えた空間の結合/融合実験などを行うと共に，当システムの性能評価を行う予定である．また，ノート PC などの小さな空間が複数の空間内を渡り歩く（そして結合/融合する）という状況で，その移動する空間を全空間から検索し通信を確立することは，サービスをローミングする場合などに必要になる．たとえ，空間が移動しない場合においても空間の数が増加し遍在した場合の適切な空間の検索のために P2P 的アプローチによる分散ディレクトリの開発を行なう予定である．

8. 謝辞

本稿のシステムの実現・検証に必要な実験設備，人員を割いていただけた内田洋行次世代ソリューション開発センターの方々，ならびに，立命館大学ユビキタス環境研究室のメンバーに感謝する．

参考文献

[1] Satyanarayanan, M.: ”Scalable, Secure, and Highly Available Distributed File Access” IEEE Computer May 1990, Vol. 23, No. 5.

[2] GAIA Project
Official Page:http://choices.cs.uiuc.edu/gaia/

[3] Smart Space Laboratory Project
Official Page: http://www.ht.sfc.keio.ac.jp/SSLab/

[4] Cogma Project
Official Page:http://www.cogma.org/

[5] Tokyo University Aoyama Morikawa Laboratory
Official Page:http://www.mlab.t.u-tokyo.ac.jp/

[6] MIT Kerberos
Official Page:http://web.mit.edu/kerberos/www/

[7] KTH Heimdal
Official Page:http://www.pdc.kth.se/heimdal/

[8] Charlie Kaufman, Radia Perlman, Mike Speciner: Network Security, Prentice Hall, 2002/04/15.

[9] Steven E. Czerwinski, Ben Y. Zhao, Todd D. Hodes, Anthony D. Joseph, and Randy H. Katz: “ An Architecture for a Secure Service Discovery Service ” Mobicom ‘ 99 Seattle Washington USA. 1999.