階層型アクセス制限機能による公開型セキュアブックマーク機能の提案と実装

中里 純二[†‡] 森 亮憲[‡] 勝本 道哲[‡] 菊池 浩明[†]

[†] 東海大学 工学研究科

[‡] 独立行政法人 情報通信研究機構

ユビキタスネットワーク技術の発達によりユーザは何処でもWebページを閲覧することが可能となり，増加するWebページに適切にアクセスするためのブックマーク機能が重要な役割を示すようになってきている．そこで，アクセス環境に応じたブックマークを表示する階層型アクセス制限機能を用いた公開型セキュアブックマークを提案する．このブックマークは，特定の閲覧ソフトに依存せず，利用者自身及び他の利用者のアクセス制限を利用可能でブックマーク情報の管理および公開ができ，利用場所によるアクセス制限を利用することでブックマーク情報の利便性を向上させ，暗号化技術を用いることにより安心して情報を利用できるものである．本論文では，Javaを用いてブロードキャスト暗号技術とアクセス制限機能のプロトタイプを実装し，その有効性を検証した結果を報告する．

Design and Implementation of the Secure Bookmark System

Junji Nakazato[†‡] Takanori Mori[‡] Michiaki Katsumoto[‡] and Hiroaki Kikuchi[†]

[†]Graduate School of Engineering, Tokai University

[‡]National Institute of Information and Communications Technology

According to the progress of ubiquitous technologies, users can access to web pages anytime and anywhere. In the environment, bookmark mechanisms for finding suitable web pages are important. Thus, we propose a secure bookmark system which uses a hierarchical access control mechanism to manage and share the bookmarks and employs cipher technologies to provide a safe and secure system. The system does not depend on particular viewers, and the access control mechanism of the sytem checks the system user and a place where the user is. We have implemented a prototype of our bookmark system using Java, which includes broadcast encryption technique to provide the access control mechanism. We have alse evaluated the effectiveness and usefulness of our bookmark system.

1 はじめに

近年，インターネットおよびユビキタスネットワーク技術の発達により，ユーザは何処でもWebページを閲覧することが可能となり，PCだけではなくPDA，携帯電話等複数の端末においてもWebページを閲覧できるようになった．またWebページも増加し，内容も多枝に渡り情報も充実してきている．このような環境において，適切にWebページへアクセスするためのブックマーク機能が重要な役割を示すようになってきている．

しかし，従来のブックマーク機能は，個々の閲覧ソフト固有であったり，端末に格納されていたりするものがほとんどであり，特定の環境においてしか利用できない，といった使いにくさが問題となってくる．そこで，アクセス環境に応じたブックマークを表示する階層型アクセス制限機能を用いた公開型セキュアブックマークを提案する．

これまでのブックマークの共有に関する研究ではPowerBookmark [1], Blink [2], Gaia [3], del.icio.us [4] などのようにオンラインで共有するための機能が実現されている．しかし，これらはブックマークをまとめたポータルサイト的な機能をはたしており，個人の知的ツールというよりは，ブックマークを検索する機能となってしまっている．本論文で提案する方式では，閲覧したいと思っているWebページに最初のアクセスで到達する機能を提供することを考える．さらに追加の関連情報を閲覧したい場合は，上記の関連研究の方式，あるいは他の検索機能を利用すればよいと考えている．

以降，2章では，本論文で提案するセキュアブックマークについて説明する．また，3章では階層アクセス制限機能について述べ，4章では階層アクセス制限機能を実現するためのブロードキャスト暗号について述べる．5章でJavaを用いたプロトタイプの実装と評価について説明し，6章でまとめを述べる．

2 公開型セキュアブックマーク

本論文で提案するブックマーク機能は，個人が後で閲覧したい，あるいは特定の情報を覚えておきたいと思っているWebページに素早くアクセスするためのブックマーク機能として，特定の閲覧ソフトに依存せず，利用者自身および他の利用者のアクセス制限を利用することでブックマーク情報の管理および公開ができる機能，さらに利用場所によるアク

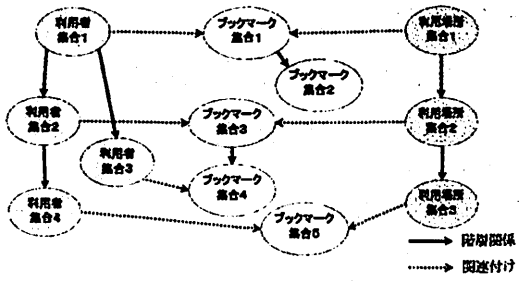

図 1: 分類の例とアクセス制御

セス制限を利用することで Web ページアクセスの利便性を向上し，暗号化技術を用いることにより安心してブックマークの管理，公開ができる方式を提案する．

これにより，ユーザは職場や自宅等のアクセス環境にお応じたブックマークの自動表示および複数台の PC 等を気にせず適切に Web ページへの閲覧が可能となり，利便性が向上すると考えている．

3 階層アクセス制御機能

本論文で提案する公開型セキュアブックマークでは，利用者および利用場所に応じて公開するブックマークおよび隠蔽するブックマークを決定する．以下では，まず，利用者，利用場所およびブックマークを分類して管理する方式について述べ，その後，分類した利用者，利用場所，ブックマークに対するアクセス制御方式について述べる．

3.1 分類管理方式

従来研究においてもブックマークを管理するさまざまな方式が提案されている [1, 2, 3, 5, 6, 7, 8, 9, 10]．本論文では，提案方式の機能を評価し，有効性を検証するため，1:一人の利用者が閲覧を許可する他の利用者を含めた集合，2：利用者のアクセス場所に応じた利用場所，および利便性を向上させるためのブックマークカテゴリを単純に分類する方式を採用した．提案方式の機能および有効性を確認することができれば，すでに提案されている管理方法を取り入れて機能を拡張すればよいと考える．

3.2 アクセス制御方式

ここでは，前節で述べた分類に基づいてアクセス制御を行う方法について述べる．アクセス制御では，利用者集合，利用場所集合およびブックマーク集合のそれぞれにおいて階層関係を定義し，利用者集合とブックマーク集合，および利用場所集合とブックマーク集合の間に関連付けを行うことでアクセス制御を行う．

例として図 1 のような場合を考える．この例では利用者集合が 4 個，利用場所集合が 3 個，ブックマーク集合が 5 個定義されている．同一種類の集合間の実線矢印は，階層関係を表している．矢印の根元は階層関係の上位を表し，矢印の先は階層関係の下位を表す．また，異種類の集合間の破線矢印は関連付けを表している．例えば，利用者集合 1 は利用者集合 2, 3 および 4 の上位に位置する．また，利用者集合 1 はブックマーク集合 1 に関連付けられている．

以上のような，集合および階層関係，関連付けを利用してアクセス制御を行う．ある利用者に対しては，その利用者が含まれている利用者集合に関連付けられているブックマークと，利用者が現在いる場所が含まれている利用場所集合に関連付けられているブックマークの和集合が表示される．このとき階層関係の下位の集合が関連付けられているブックマークも表示されるものとする．

例えば，ある利用者が利用者集合 1 に含まれているとする．この人が，利用場所 1 に含まれている場所にいると，すべてのブックマークが表示される．そして，この利用者が場所集合 3 に含まれている場所に移動すると，ブックマーク集合 5 のみが表示される．また，利用者集合 2 に含まれる人が，場所集合 3 に含まれる場所にいるときにも，ブックマーク集合 5 が表示される．しかし，利用者集合 3 に含まれる人が，場所集合 3 に含まれる場所にいるときには，ブックマークは一つも表示されない．

以上のようなアクセス制御を行うことにより，利用者およびその利用者がいる場所に応じて，表示するブックマークを制御することができる．

4 暗号化方式

本論文では，効率の良いアクセス制限機能実現のために，ブロードキャスト暗号を用いた．ブロードキャスト暗号とは，SKY PerfecTV! [11] などの有料コンテンツ配信に用いられている．複数の異なる鍵を所有するユーザに対し，正規のユーザのみ復号できるような暗号文を，効率よく同報通信するための技術である．そこで，ブロードキャスト暗号の不正ユーザの排除機能をアクセス制限機能に応用する．

効率の良いブロードキャスト暗号に関する研究は Naor らによって提案された [12] (Complete Subtree: CS 法) がある．また，その後，多くの研究者により効率の改善が行われている [13, 14, 15, 16]．本論文で提案する手法では，最も基本的な CS 法を導入する．

4.1 Complete Sub-tree 法 (CS 法)

ここでは，CS 法の概要を説明する．CS 法は，完全 2 分木の木構造を利用し，葉 (リーフ) に各利用者を配置する．そのため，最大利用者数 m に対して，深さ $\lceil \log_2 m \rceil$ の鍵管理木を構成する．各ノードには共通鍵アルゴリズムの鍵が配置され，各利用者は，根 (ルート) に至るまでのノードに配置された鍵を配布される．つまり，各利用者には，$\lceil \log_2 m \rceil + 1$ 個の鍵が配布される．

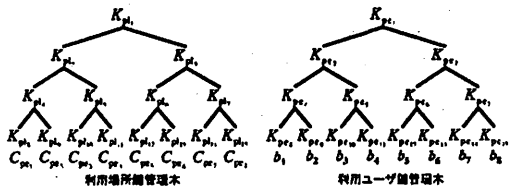

図 2: 鍵管理木

不正利用者が存在しないとき，全ての利用者が共通に所有しているルートの鍵を用いて，コンテンツの暗号化に用いたセッション鍵の暗号化を行う．

一方，不正利用者が存在するときは，その不正利用者の所有する鍵を無効とし，それ以外の鍵を用いてセッション鍵の暗号化を行う．このとき，利用者が共有している鍵をできるだけ用いることで，暗号化回数を少なくする．

4.2 CS 法を用いてアクセス制御

提案方式では，ブロードキャスト暗号の不正利用者排除機能を用いてアクセス制限機能を実現する．配布する鍵の数を極力少なくするために，一方向性ハッシュ関数を用いる．一方向性ハッシュ関数を組み合わせることにより，親ノードの鍵からそれ以降の子ノードの鍵を生成することができる．また，場所によるアクセス制限と，人によるアクセス制限を実現するために，利用者鍵管理木と利用場所管理木の 2 つの鍵管理木を構成する．暗号化は，ブックマークの分類ごとに施し，利用者用の鍵で暗号化した後に，利用場所用の鍵で暗号化を行う．二重に暗号化をすることで，両方の鍵が一致したときのみブックマークを復号することができる．

以下に提案方式を示す．

Step 0: ブックマークの分類　ブックマーク公開者 P は，ブックマーク B を n 個に分類する $(b_1, b_2, \ldots, b_n)$.

Step 1: 鍵生成　P は，図 2 のように共通鍵暗号アルゴリズムの鍵 K_{pe_1} と K_{pl_1} を生成する．生成した鍵をルートノードの鍵とし，一方向性ハッシュ関数を用いて各ノードの鍵を以下のように生成する．

$$\begin{cases} K_{pe_i} &= \mathbf{H}\left(K_{pe_{\lfloor i/2 \rfloor}} \mid i\right) \\ K_{pl_i} &= \mathbf{H}\left(K_{pl_{\lfloor i/2 \rfloor}} \mid i\right) \end{cases}$$

ここで，$i = 2, 3, \ldots, 2 \cdot 2^{\lceil \log_2 n \rceil} - 1$ とする．

Step 2: 暗号化　P は，利用者用の鍵管理木の各リーフに配置した鍵 K_{pe_i} を利用して分類ごとに分割したブックマークを以下のように暗号化する．

$$C_{pe_j} = \mathbf{E}_{K_{pe_j}}[b_j]$$

次に，利用場所用の鍵管理木の各リーフに配置した鍵 K_{pl_i} を利用して，以下のように C_{pe_i} を暗号化する．

$$C_{pl_i} = \mathbf{E}_{K_{pl_i}}[C_{pe_j}]$$

ここで，$i = 2^{\lceil \log_2 n \rceil} + j - 1$ $(j = 1, \ldots, n)$ とする．

Step 3: 鍵配布　P は，閲覧を許可する利用者 U に対し利用者用の鍵管理木から，閲覧を許可する利用場所 V に対し利用場所用の鍵管理木から，それぞれアクセス制限に応じた鍵を安全に配布する．ここで，すべての閲覧を許可する場合は，ルートの鍵を配布し，一定の制限を与える場合は，閲覧を許可するリーフの共通親ノードの鍵を配布する．例えば図 2 において，U に b_7, b_8 の閲覧を許可する場合は，その共通親ノードに設定されている鍵 K_{pe_7} を配布する．U は，K_{pe_7} より $K_{pe_{14}}$, $K_{pe_{15}}$ を生成可能なため，配布する鍵数を削減する事ができる．

Step 4: ブックマーク公開　P は暗号化されたブックマーク C_{pl_i} を公開用サーバ S で公開する．

Step 5: 閲覧　U は，S により公開された暗号化ブックマーク C_{pl_i} を取得する．まず，V に設定された鍵を用いて復号し，U に配布された鍵を用いて復号する．この時，利用場所 V に設定されている鍵と，利用者 U が所有している鍵が共に閲覧可能に設定されていないと，ブックマーク b_j を閲覧することができない．

5 実装と評価

提案方式の有用性を示すために，試験実装を行った．表 1 に開発環境を示す．

5.1 システム構成

本論文では，Java を用いて提案方式を実装した．Java で実装することにより，実行プラットフォームに依存しないシステムを提供することができる．また，ブックマーク閲覧のための試験実装であるため，ブックマークエントリの追加，削除の機能は省略している．実装システムでは，以下の 4 つの機能を提供する．

鍵生成　ブックマークの最大分類数 n を入力とし，$2 \cdot 2^{\lceil \log_2 n \rceil} - 1$ 個の 128bit の AES 鍵を XML フォーマットで出力する．

配布鍵生成　鍵生成で生成した鍵と，閲覧を許可するブックマークのリーフ番号を入力とし，配布用の鍵束を XML フォーマットで出力する．

既存ブックマークの暗号化　鍵生成で生成した鍵と，暗号化するブックマークを入力とし，XML フォーマットで暗号ブックマークを出力する．

表 1: 開発・実行環境

	Windows	Mac
CPU	Pentium4 3.2GHz	PowerMac G5
メモリ	2GB	4GB
開発言語	J2SDK 1.4.2_03	
暗号化アルゴリズム	AES 128bit	
ファイルフォーマット	XML	

(a) Safari実行画面

(b) FireFox実行画面　(c) IE実行画面

図 3: 閲覧画面

暗号ブックマークの閲覧機能　配布された鍵と，暗号ブックマークを入力とし，閲覧許可されたブックマークを Java Applet 上に表示する．

まず，鍵生成プログラム KeyGen により，利用者用鍵および利用場所用鍵を生成する．つぎに，ブックマーク暗号化プログラム HTMLParser により，ブックマークの暗号化を行う．ここで，ブックマークのどの分類が，どの鍵で暗号化されたかが決定するため，その鍵番号を利用して，配布用鍵生成プログラム KeySelector により，配布用鍵を生成する．

暗号ブックマーク閲覧プログラムは，利用機種や利用ブラウザに依存しないようにするため，Java Applet により実装した．ブックマーク公開用 URL にアクセスすることで，鍵を選択した後に図 3 のようなインタフェースを提供する（ここでは，それぞれ異なるアクセス場所からを想定している）．本システムでは，2 つのフレームを利用し，左側のフレームに Java Applet を表示する．ここで，Java Applet 内で暗号ブックマークが復号され，表示される．右フレームには，Java Applet 内に表示されているブックマークのリンクをクリックしたときに，ページの内容が表示される．

5.2 ファイルフォーマット

実装システムでは，XML Encryption をフォーマットに採用した．XML Encryption は 2002 年 12 月 10 日に W3C の勧告を受けた [17, 18]．しかし，XML Encryption では，ブロードキャスト暗号のような特殊な暗号方式に対応していない．そこで，属性値を利用することで，ブロードキャスト暗号に対応できるようにした．XML Encryption では，暗号化データを格納していることを示す ⟨EncryptedData⟩ タグ内に，任意の ID を付加することができる Id 属性をオプションで設定できる．実装システムでは，この ID を暗号化に用いた鍵番号とした．また，鍵情報

```
1: ⟨?xml version="1.0" encoding="UTF-8"?⟩
2: ⟨BOOKMARKS⟩
3:     ⟨EncrptedData Id="15"
4:     xmlns=http://www.w3.org/2001/04/xmlenc#
5:     MimeType="text/xml"⟩
6:         ⟨CipherData⟩
7:             ⟨CipherValue⟩ Qu...KO ⟨/CipherValue⟩
8:         ⟨CipherData⟩
9:         ⟨ds:KeyInfo xmlns:ds=
10:        "http://www.w3.org/2000/09/xmldsig#"⟩
11:            ⟨ds:KeyName⟩ Place Key ⟨/ds:KeyName⟩
12:            ⟨KeyIV⟩ BB...va ⟨/KeyIV⟩
13:        ⟨/ds:KeyInfo⟩
14:    ⟨/EncrptedData⟩
15: ⟨/BOOKMARKS⟩
```

図 4: 暗号化ブックマークの例

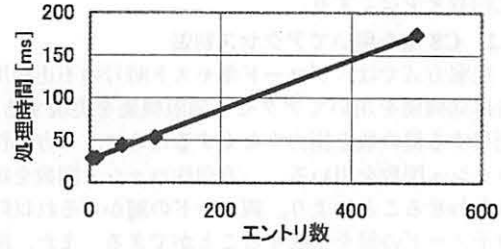

図 5: エントリに対する処理時間

を格納する ⟨KeyInfo⟩ タグ内，⟨KeyName⟩ タグで，利用者用の鍵による暗号化または，利用場所用の鍵による暗号化であるかを示し，⟨KeyValue⟩ タグに共通鍵暗号で用いた初期値（IV）を格納した．暗号化データや，鍵データは Base64 により符号化された値を用いた．図 4 は，ブックマーク全体をノード番号 15 番の鍵を用いて暗号化したものである．

5.3 評価

ここでは，実装システムの評価を行う．図 5 にブックマーク内のエントリ数に対する処理時間を，図 6 に階層（分類）の深さに対する処理時間を示す．処理時間とは，ブックマークを復号するためにかかる時間を表す．図 5 より，処理時間はブックマークのエントリ数に比例することがわかる．これは，提案方式では，ブックマークの暗号化を分類ごとに行うため，1 つの分類の情報に比例し，復号時間が増えるためである．また，図 6 より，階層が深くなるにつれて，指数関数的に処理時間がかかることがわかった．

実装システムの動作検証を行うため，すべてのブックマークを閲覧可能な利用者が以下の 3 箇所からアクセスした場合を想定して実験を行った．

- 仕事場からのアクセス
- 自宅からのアクセス
- 公共の場からのアクセス

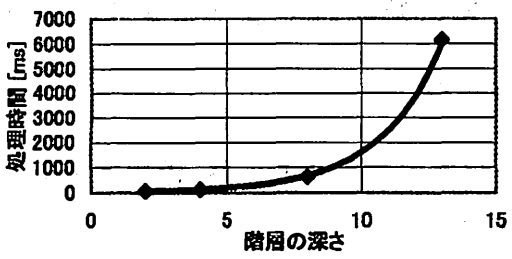

図 6: 階層の深さに対する処理時間

表 2: 実験結果

	利用場所鍵数	表示有無
仕事場	2	○
自宅	2	○
その他	1	○

本実験では，位置情報はあらかじめ設定しているものとして行った．また，ブックマークを，“仕事”，“趣味”，“その他”に分類し，同一階層にあるものとする．仕事場からは，“仕事”のみ，自宅からは“趣味”および“その他”，公共の場からのアクセスには“その他”のみ閲覧可能に設定し，正しい表示結果を得られるかを調べた．表 2 に実験結果を示す．利用場所鍵数とは，利用場所に設定する鍵である．例えば，図 2 の b_1 と b_3 を閲覧する場合は，共通親ノードを持たないため，$K_{pc_8}, K_{pc_{10}}$ の 2 つの鍵が必要となる．この結果，利用者および利用場所に応じたアクセス制御が行えることを確認した．これらの機能，特に利用場所に応じたアクセス制御は，1 章で挙げたような既存のブックマーク共有ツールでは提供されていないことから，提案方式はこれらと比較して有効性および利便性が高いと考えられる．

6 まとめ

本論文では、ブロードキャスト暗号を用いた階層アクセス制御機能を備える公開型セキュアブックマークの提案を行い，Java および XML Encryption フォーマットを用いた実装に関して述べ，簡単なプロトタイプによりその有効性および利便性が高いことを確認した．

本稿では簡易データで実験を行ったが，今後は GPS データおよび RFID タグと連動させた広域実験を行いより実用性の向上を行う予定である．さらに，文献 [3] でも指摘されているシステムを利用するためのログインが面倒であるという問題点を解決するために，指紋認証などの安全性を備えたユーザインタフェースを考案しブックマークが格納されているサーバへのアクセスを自動化することを考えている．これにより利用者の利便性が向上すると考えられる．

参考文献

[1] Li, W., Vu, Q., Agrawal, D., Hara, Y. and Takano, H.: PowerBookmarks: A System for Personalizable Web Information Organization, Sharing, and Management, *Proc. of 8th International World Wide Web Conference* (1999).

[2] Blink: http://www.blink.com. (2005 年 8 月現在)．

[3] 近藤秀和, 村岡洋一: Web サービスを利用した次世代型オンラインブックマークシステム, 信学技法 DE2004-4, pp. 19–24 (2004).

[4] del.icio.us: http://del.icio.us/. (2005 年 8 月現在)．

[5] Maarek, Y. S. and Ben-Shaul, I.: Automatically Organizing Bookmarks per Contents, *Proc. of 5th International World Wide Web Conference* (1996).

[6] Schmiedel, A. and Volle, P.: Using Structured Topics for Managing Generalized Bookmarks, *Proc. of 5th International World Wide Web Conference* (1996).

[7] Keller, R. M., Wolfe, S. R., Chen, J. R., Rabinowitz, J. L. and Mathe, N.: A Bookmarking Service for Organizing and Sharing URLs, *Proc. of 6th International World Wide Web Conference* (1997).

[8] Takano, H. and Winograd, T.: Dynamic Bookmarks for the WWW, *Proc. of 9th ACM Conference on Hypertext and Hypermedia* (1998).

[9] 中島伸介, 黒田慎介, 田中克己: 閲覧履歴を反映したコンテクスト依存型 Web ブックマーク, 情報処理学会論文誌（データベース）, Vol. 43, No. SIG 5 (TOD 14), pp. 23–36 (2002).

[10] Bookmarks Synchronizer: http://www.mozilla-japan.org/products /firefox/のブックマーク拡張機能. (2005 年 8 月現在)．

[11] SKY PerfecTV!: http://www.skyperfectv.co.jp/. (2005 年 8 月現在)．

[12] Naor, D., Naor, M. and J.Lotspiech: Revocation and Tracing Schemes for Stateless Receivers, *Proc. of Advances in Cryptology CRYPTO'2001 (LNCS2139)*, pp. 41–62 (2001).

[13] Halevy, D. and Shamir, A.: The LSD Broadcast Encryption Scheme, *Proc. of Advances in Cryptology CRYPTO'2002 (LNCS2442)*, pp. 47–60 (2002).

[14] Attrapadung, N., Kobara, K. and Imai, H.: Sequential Key Derivation Patterns for Broadcast Encryption and Key Predistribution Schemes, *Proc. of Advances in Cryptology ASIACRYPT'2003 (LNCS284)*, pp. 374–391 (2003).

[15] Asano, T. and Kamio, K.: Broadcast Encryption based on Rabin Tree Revisited, *Proc. of Symposium on Cryptography and Information Security (SCIS2005)*, pp. 427–432 (2005).

[16] Attrapadung, N., Kobara, K. and Imai, H.: Shor Encrypted Broadcast with Shot Key, *Proc. of Symposium on Cryptography and Information Security (SCIS2005)*, pp. 1129–1134 (2005).

[17] XML Syntax and Processing: http://www.w3.org/TR/xmlenc-core/. (2005 年 8 月現在)．

[18] Decryption Transform for XML Signature: http://www.w3.org/TR/xmlenc-decrypt. (2005 年 8 月現在)．