

データベース構築促進及び技術開発に関する報告書

認定サービスにおけるデータベースの 有効利用に関する調査研究

平成12年3月

財団法人 データベース振興センター

委託先 株式会社 シネジャーナルプロダクション



この事業は、競輪の補助金を受けて実施したものである。

データベース構築促進及び技術開発に関する報告書

認定サービスにおけるデータベースの 有効利用に関する調査研究

平成12年3月

財団法人 データベース振興センター

委託先 株式会社 シネジャーナルプロダクション

序

データベースは、わが国の情報化の進展上、重要な役割を果たすものと期待されている。今後、データベースの普及により、わが国において健全な高度情報化社会の形成が期待される。さらに海外に対して提供可能なデータベースの整備は、国際的な情報化への貢献および自由な情報流通の確保の観点からも必要である。現在わが国で流通しているデータベースの中でわが国独自のものは約半数であるが、わが国データベースサービスひいてはバランスある情報産業の健全な発展を図るためには、今後もわが国独自のデータベースの構築およびデータベース関連技術の研究開発を強力に促進し、データベースの拡充を図る必要がある。

このような要請に応えるため、(財)データベース振興センターでは日本自転車振興会から機械工業振興資金の交付を受けて、データベースの構築および技術開発について民間企業、団体等に対して委託事業を実施している。委託事業の内容は、社会的、経済的、国際的に重要で、また地域および産業の発展の促進に寄与すると考えられているデータベースの構築とデータベース作成の効率化、流通の促進、利用の円滑化・容易化などに関係したソフトウェア技術・ハードウェア技術である。

本事業の推進に当って、当財団に学識経験者の方々と構成されるデータベース構築・技術開発促進委員会（委員長 東海大学教授 上條史彦氏）を設置している。

この「認定サービスにおけるデータベースの有効利用に関する調査研究」は、平成11年度のデータベースの構築促進および技術開発促進事業として実施した課題の一つで、当財団が株式会社シネ・ジャーナルプロダクションに対して委託実施したものである。この成果が、データベースに興味をお持ちの方々や諸分野の皆様方のお役に立てば幸いである。

なお、平成11年度データベースの構築促進および技術開発促進事業で実施した課題は次表のとおりである。

平成12年 3 月

財団法人 データベース振興センター

平成11年度 データベース構築・技術開発促進事業委託課題一覧

区 分	No.	課 題 名	企 業 名
一 般	1	認定サービスにおけるデータベースの有効利用に関する調査研究	(株) シネジャーナルプロダクション
	2	ゲームソフトにおける知的財産権管理流通ビジネスに関する調査研究	京都リサーチパーク (株)
	3	地形データを効率よくデータベース化する技術開発	(株) セタ
	4	戦前期雑誌記事索引100万件の第一期データベース構築	(株) 皓星社
	5	POSデータに対するデータマイニング手法群の比較研究と実用化技術の開発	(株) 日経リサーチ
	6	北九州産業技術発展の歴史 記録の収集とデータベース化	(株) 日鉄技術情報センター
地域振興	7	写真データベース構築	(株) 琉球新報社
	8	バリアフリー施設データベースのプロトタイプ作成	(株) 札幌ネクシス
	9	新居浜市機械産業における加工技術ノウハウ継承データベース構築	(株) 四国インターネット
	10	インターネットによるラジオニュース公開型データベース構築	(株) 山梨放送

目次

はじめに	1
1. 課題性の研究	3
1-1 目的	3
1-2 実施内容	3
(1) 認定情報データベースの構築	3
(2) 認定情報データベースの更新	4
(3) 認定情報データベースの検証と認定認証	4
1-3 実施体制	4
2. 現状の認定サービスの現状	5
2.1 ウェブの用途の拡大	5
2.2 現状の認定サービスの試行例	5
2.3 マーク不正利用に関する対策の現状	11
2.4 なりすましに対する新規技術動向	17
3. 認定情報データベースの構築	19
3.1 認証情報の設計	19
3.2 検証の方法	22
4. 認定情報データベースの更新	25
4.1 サーバ-クライアント分散システム	25
4.2 サーバ-サーバ分散システム	27
5. 認定情報データベースの検証と認定認証	30
5.1 マークの認証	31
5.2 検証サーバの認証	34
6. なりすまし対策の検証機構	35

6.1 認証のチェーン	35
6.2 検証操作	36
6.3 画像フォーマット	37
7. まとめ	42
7.1 データベースの形態	42
7.2 データベースの更新	43
7.3 なりすましの防御対策	43
7.4 検証機構の提案	44
8. 今後の課題	45

はじめに

近年のインターネットの発展と低額化に伴い、ここ1, 2年にインターネットユーザが爆発的に伸びてきている。インターネットが発展するにつれ、専門家や研究者に限らず、初心者でもインターネットを使用するようになってきた。インターネットが一般化する以前から盗聴・改ざん・なりすまし等の懸念が持たれていて、様々な研究や PEM(Privacy Enhanced Mail)や SSL(Secure Socket Layer)などのプロトコルの制定、アプリケーションの開発は行われてきた。

近年インターネット上の事件や事故の報道が増えてきて、ここ数年特にセキュリティ、並びに認証の必要性が問われている。インターネットでは本人認証が難しいため、詐欺やなりすましによる誹謗中傷などが問題になってきている。

現在広く一般に使用されているウェブブラウザでは、通信上の盗聴を防ぐ対処として、SSLを使用することができる。最近ではSSLを使用しているオンラインショップも多いが、SSLはサーバホスト名が正しいことを認証し、サーバクライアント間で秘匿通信を行うものだけである。

オンラインショップなどを行う場合、SSLやSET(Secure Electronic Transaction)等の暗号・認証機能を使って、サーバ認証の後、暗号化してセキュアに通信することは現時点でも可能である。しかし、実際商品を購入する場合、本当にその店が信用できるか、個人情報を利用されないか等のユーザが不安な要因が購入を促進しない原因のひとつとも考えられる。すなわち、バーチャルな世界とリアルな実社会とを認証し、結びつける手立てがないため、このような不安は解消されないでいる。

対策として、存在確認、信用情報や個人情報の取り扱い等について、オフラインでオンラインショップを運営する業者を認定し、この業者とインターネット上のサーバをどのように結びつけ、実ユーザと結びつける「認証サービス」を如何に提供できるかにある。

そこで、今回は最近特に利用され始めてきている認定マークとその認証方法についてデータベースを使って効率的に行う方法について考察する。

1. 課題性の研究

1-1 目的

インターネットの発展に伴い、セキュリティ、並びに認証の必要性が問われている。盗聴・改ざん・なりすまし等は様々な方法で対策がとられている。しかしそれらはサーバやマシンなど機器を認証するものであり、それらの上で動いているアプリケーションやさらにそれらを動かしている人間や事業者について認証している訳ではない。

オンラインショップなどを行う場合、SSL(Secure Socket Layer)等の暗号・認証機能を使って、サーバ認証の後、暗号化してセキュアに通信することは現時点でも可能である。しかし、実際商品を購入する場合、本当にその店が信用できるか、クレジットカードを悪用されないか、商品が信用できるものであるか、また、個人情報を流用されないか等のユーザが不安な要因が購入を促進しない原因のひとつと考えられる。

対策としては、信用や個人情報の取り扱い等についてオンラインショップを運営する業者を認定し、この業者とインターネット上のサーバをどのように結びつけるかにある。

そこで、今回は最近特に利用され始めてきている認定マークとその認証方法についてデータベースを使って効率的に行う方法について考察する。

1-2 実施内容

本調査研究では、公開鍵証明書をターゲットとして以下の項目を主たる内容としている。

(1) 認定情報データベースの構築

データベース間での更新を行う上で最適な配列の考察。

(2) 認定情報データベースの更新

分散データベースを考慮した場合の最適な更新の仕方の考察。

(3) 認定情報データベースの検証と認定認証

検証の対象となるオブジェクトと検証データベースの認証についての考察。

1-3 実施体制

当社内に当該事業のための「特別調査、研究委員会」を編成し、事業の実施に当たる。

なお、必要に応じて専門家によるプレゼンテーションおよびヒアリングを実施する。

調査研究委員会の編成

市川 隆	(日本情報処理開発協会・常務理事)
山鳥 雄嗣	(日本情報処理開発協会・〃)
木村 耕	(電気通信大学 情報工学科 助教授)
久保悌二郎	(江戸川大学 社会学部マス・コミュニケーション学科 教授)
北野 博之	(日本情報処理開発協会・情報セキュリティ対策室)

2. 現状の認定サービスの現状

2.1 ウェブの用途の拡大

ウェブは元々ネットワークに接続されたアカデミックな機関の間で学术论文などの情報交換を行うために開発されたものであった。最近ではウェブ上のサービスも充実してきており、電子商取引に限らず様々な分野でインターネットとウェブ技術が使われるようになってきて、カタログを見たり、社章や認定マークをつけたカラフルページが増え、実社会に近づいてきている感がある。

主に広告の世界であるが、最近では様々なバナー広告を貼ってマークの画像データで宣伝しているウェブページをよく見かけるようになってきた。これはビジュアル的であり、インターネットユーザにもわかりやすい仕組みである。しかし、画像データはコピーが可能である。広告の世界では、わざわざコピーして使用することもあまり考えられないが、悪意を持ったものがイメージダウンのために、別のサーバへコピーしわざと広告主の意図しないウェブページへとリンクを張ることもありえる。

今後電子商取引に関わらず、様々な分野で会員や資格をウェブ上でアピールする手段として、まだまだ少ないが、バナー広告のようにある画像データを貼るケースが多くなると思われる。

2.2 現状の認定サービスの試行例

インターネット上でビジュアル的に表現できるようになったら、次に考えるのがホテルの丸適マークや飲食店の免許証のようにある「資格」をアピールすることである。現在まだまだ少ないが、認定証として認定マークを画像データ化し、その認定マークをウェブ上でアピールしている事例を見てみる。

1999 年 11 月現在、財団法人日本情報処理開発協会(JIPDEC)¹のプライバシーマーク事務局²では、認定を受けた事業者はプライバシー事務局から配布された画像データを自社のウェブページに貼り、プライバシーマーク事務局が設置する認定事業者リストのページを表示するようになっている。しかしこれだけでは、正当な認定を受けた事業者への強制だけに過ぎず、悪意のある事業者が画像データをコピーし、あたかも認定を受けているかのように振舞うことが可能である。また、正当な認定者でも認定範囲や期間を越えた使用を防げない。

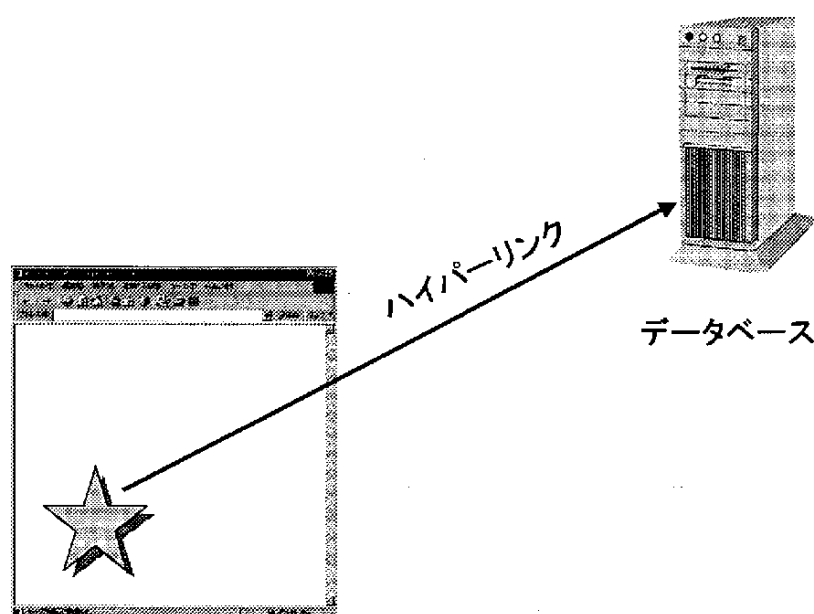


図 1 認定マークとハイパーリンク

また、プライバシーマーク事務局では平成 11 年度にマーク検証システムを開発中³の

¹ <http://www.jipdec.or.jp/>

² <http://www.jipdec.or.jp/security/privacy/>

³ <http://privacymark.gr.jp/> (1999.12 月現在開発中)

ようであるが、画像に刷り込まれた電子透かし⁴を解読する plug-in アプリケーションをウェブブラウザにインストールしなければならない。電子透かしは元のデータ特性をそのままに付加情報を刷り込むものであり、元の画像データの特性はそのまま、画像データとして表示される。すなわち、plug-in をインストールしないままでは何も気づかず画像を表示してしまう。この方式では

- ・ クライアント側である程度の検証が行える(オフライン検証)

である反面、

- ・ オフライン検証時は失効情報や詳細データがわからない
- ・ 検証の為に plug-in アプリケーションをわざわざインストールしなければならない
- ・ 固有のサービスの plug-in になりかねない

などの問題があるように思える。

⁴ 深層暗号とも呼ばれ、人間の視覚特性等を利用して元のデータに別の情報を刷り込む技術

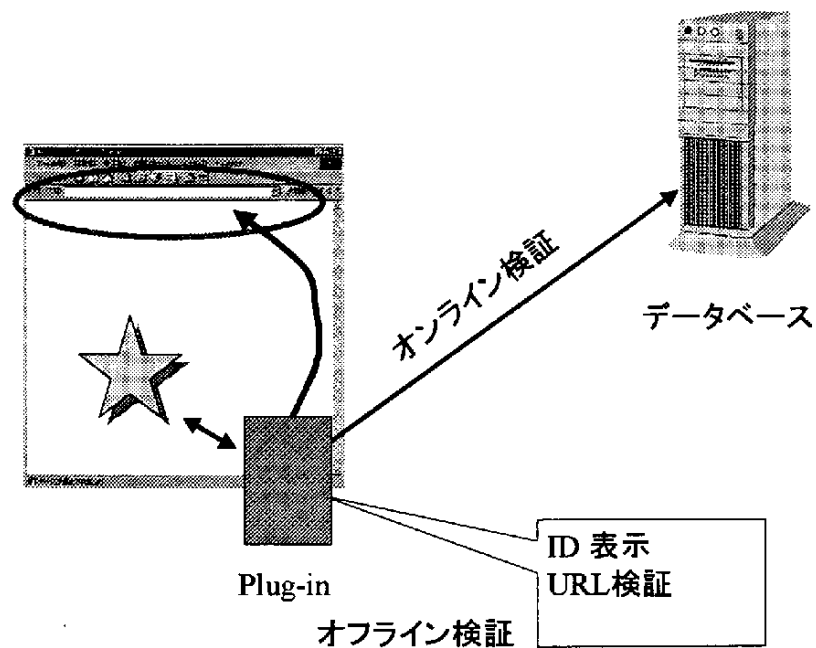


図 2 プライバシーマークの検証スキーム

社団法人日本通信販売協会(通販協)⁵が1999年8月から実証実験を行っているオンライン・トラスト・マーク(オンラインマーク)は、マスター検証サーバ上にある JAVA Applet を呼び出し、画像の表示前に HTTP のプロトコルのリンク元の情報⁶を元に有効性検証を行っているようである。この仕組みは

- ・ 絶えず検証時の情報(失効情報を含めた)を得ることができる

反面、

⁵ <http://www.jadma.org/>

⁶ ウェブブラウザの環境変数 HTTP_REFERER

- ・ ブラウザから呼び出される度に毎回マスター検証サーバへ問い合わせを行い、サーバへ極集中的に負荷がかかる
- ・ HTTP プロトコルのリンク元の情報を利用しているようであり、フレームに対応していない(要するにフレームを使えばなりすましが可能)

などの問題があると思われる。

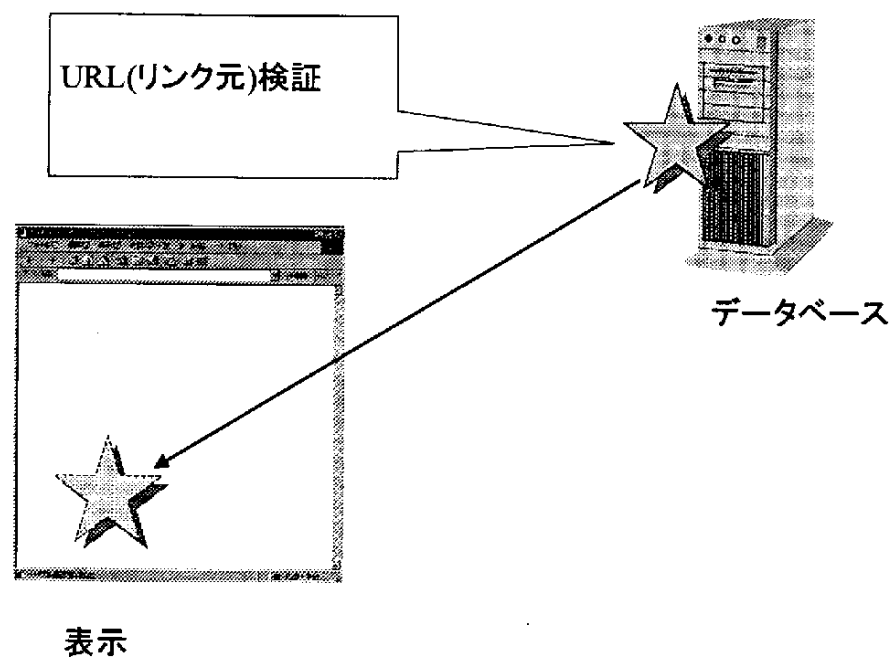


図 3 オンラインマークの検証スキーム

データベースへ問い合わせで検証を行う“オンライン検証”はどちらも事前に

- マークが貼られる(表示される)ページの URL

を登録し、

- マーク自体(プライバシーマークの場合)
- マークを表示するプログラムをリンクしているの元のページ URL(オンラインマークの場合)

を検証対象情報としてデータベースを参照し、有効であるか否かを判別しているようであり、検証の機能としてはほぼ同機能であると思われる。

以上のケースを考察してみると、マーク画像データと検証データベースの関係は「クライアント-サーバ分散モデル」として捉えることができ、

検証対象 (シンボル) ⇔ 検証レファレンス (データベース)

の設計が重要であるといえる。

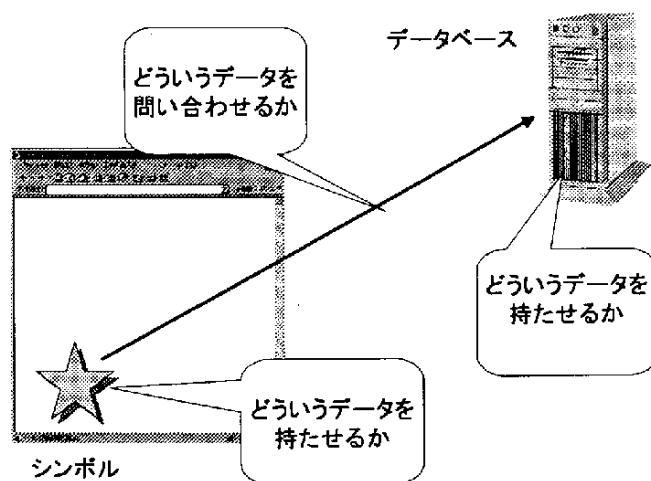


図 4 シンボルとデータベースの関係

2.3 マーク不正利用に関する対策の現状

インターネット上では、事業者の実体が見えなかったり、画像データが簡単にコピーされるなどの問題があり、簡単になりすましを行える危惧がある。これはいくらデータベースの設計を高めたところで、検証の対象となるデータを成りすまされたら、意味のないシステムとなる。

なりすましに関しては、大きく分けて以下の2つの大きな問題が考えられる。

1. 不正事業者のなりすまし

認定を受けていない事業者がマークをコピーして使用し、あたかも正当に認定を受けているかのようにユーザへ見せる。

2. 不正検証サーバのなりすまし

上記の不正事業者自身、または第三者が検証サーバを構築し、あたかも正当に検証を行ったかのように振る舞わせる。

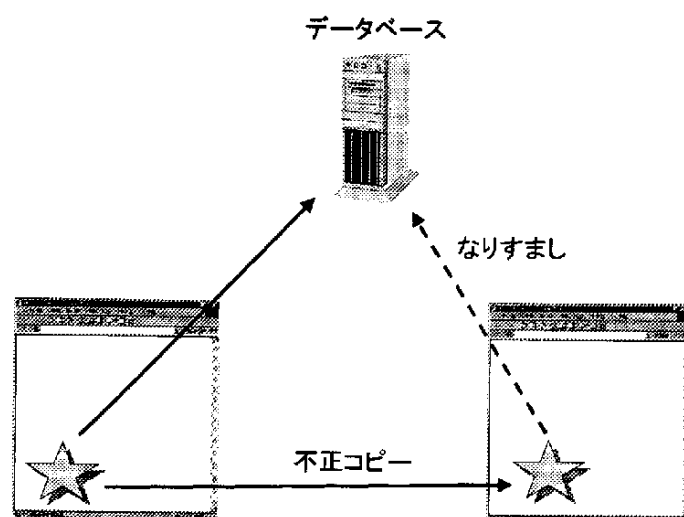


図 5 不正事業者のなりすまし

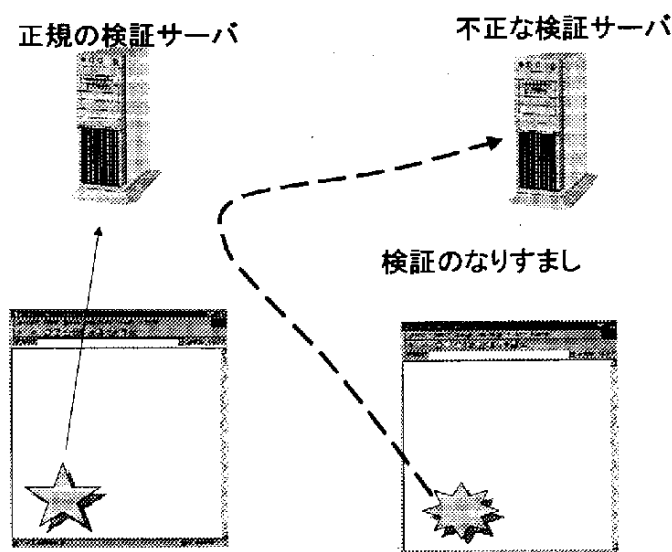


図 6 不正検証サーバのなりすまし

マークの画像自体がインターネット上で使用される以上、現状のプロトコルに従わざるを得ず、また、ブラウザに表示された画像はダウンロードやコピー&ペーストによって比較的簡単にコピーでき、再利用することができる。また、画像自体はコピーせず、画像をページにリンクとして表示させることも現在の HTML プロトコルでは造作もないことである。

そこで不正事業者のなりすまし対策としては、不正コピーがあることを前提に行わねばならない。そこで

- a. マーク自体が正式な認定マークであるか否か
- b. 認定マークが正規の配布先から配布されたものであるか否か
- c. 正規の認定マークであった場合、正規の利用者が使用しているか否か

の検証が必要となると思われる。

検証サーバがなりすまされた場合、いくら「不正事業者のなりすまし」対策を行っても全く無意味なシステムになってしまう。「不正検証サーバのなりすまし」についても対策をとらねばならない。

現状の HTML プロトコルでは、コンテンツ管理者が検証サーバへリンクさせるハイパーリンクを含め自由に書ける状態であり、これら検証先へのハイパーリンクを含めた HTML コンテンツ自身の検証などはなされていない状況である。

現在オンラインショップをはじめとするセキュア通信を行う為に利用されている公開鍵証明書を用いた SSL や TLS はサーバ自身の認証しか行うことができず、仕様上の問題やブラウザでの確認画面の実装の問題等で、現状ではこれらの問題には対処できてない。

「1. 不正事業者のなりすまし」を防ぐには、現時点では以下の対策が考えられる。

(1) 画像データへの工夫

不正コピーを防いだり、不正コピーされても大丈夫なように画像データ自体に識別情報を盛り込むなど。

(2) サーバの HTML 記述に依存しない検証方法

検証サーバの問い合わせ先を改変できないようにするなど。

(1) の処理は、

- プライバシーマーク

画像データに電子透かし処理を施し、画像データに識別子を刷り込み、コピーされてもその識別子が残るようにしている。

- オンラインマーク

JAVA を利用して、「画像のダウンロード」ができないように対処している。

(2) については、

- プライバシーマーク

画像データは通常の方法で貼ってもらい、検証時には専用の plug-in ソフトを使用して、画像データの URL とマークが貼られてある URL の情報を正当な検証サーバへ送る。

- オンラインマーク

JAVA Applet を起動するリンク元の情報を元に、正当な事業者の URL からプログラムが起動されているか否かを判断している。

「2. 不正検証サーバのなりすまし」についてはかなり難しい。すなわち検証作業事態をなりすまされたら、“不正に行われている”という事実を検出しにくいからである。プライバシーマークでは、

o 専用の plug-in ソフトを使用させることで、正当な検証サーバへ送る。

ということを「不正検証サーバのなりすまし」の防御の拠り所としているようである。可能性は低いと思われるが、不正な plug-in を配布してなりすましサーバによる擬似検証が行えなくもない。

オンラインマークの場合の「不正検証サーバのなりすまし」を考え、例えば、ここで「自己検証機能」を持った JAVA Applet を考えてみる。自己検証機能とは、自身で

正しく使われているか否かを検出し、一般ユーザに表示する機能とする。

以下の手順で作成する。

1. マークを貼るページの URL を事前に登録(通知)してもらう
2. 上記 URL を盛り込んだオンラインマークを参考に画像を表示する JAVA Applet を作成する。
3. この個別にカスタマイズした JAVA Applet を事業者へ配布し、タグの代わりに、<APPLET>...</APPLET>タグでウェブページに貼る。

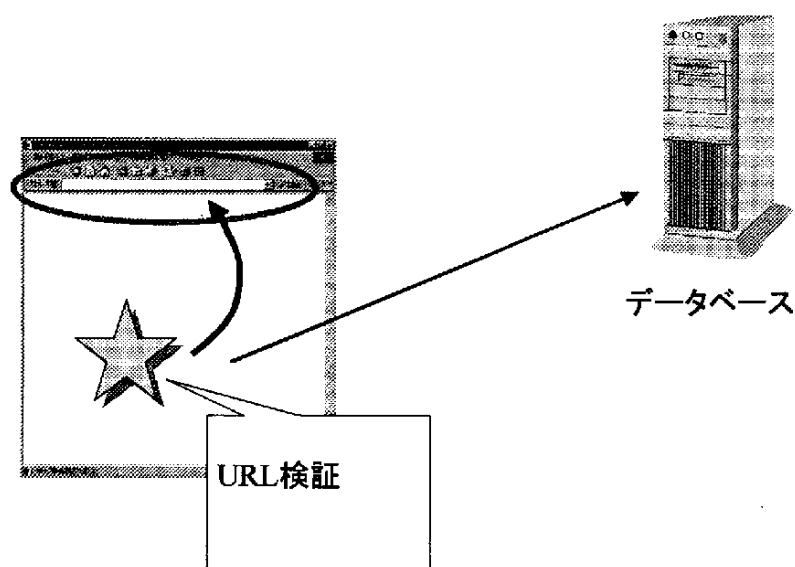


図 7 自己検証機能付きマーク

これによってユーザがこのウェブページをブラウザで閲覧した時、ブラウザの URL が Applet 内に盛り込んだ情報と一致すれば OK、一致しなければ NG とすることができる。

この時、OK の時は正しいマーク表示、NG の場合は「不正である」などの表記をするこ

とも可能である。

プライバシーマーク、オンラインマークの利点を生かして、第3番目の方法として、画像データ自身に識別子を持たせ、自己識別機能を持ったオブジェクト的な画像データを考える。

しかし自己検証機能付きマークのように plug-in などの補助的な検証アプリケーションを使わず、ウェブ上のアプリケーションで対処だけでは、簡単に JAVA Applet が偽造されることが安易に予想される。ウェブの技術だけでは簡単になりすましの問題は難しい。

ここまで考えると、JAVA Applet のように

- ・ 画像データ
- ・ 検証プログラム

を同一のファイルで行うのは難しく、画像ファイルを別に作成し、plug-in などの検証アプリケーションを用いる方法でも十分だと思われるが、「不正検証サーバのなりすまし」に対してはほぼ完全に行うには、データベースの仕組みやウェブ関連技術に頼らず、他の「第3者的なよりどころ」を使わざるを得ない。現状を考えると、

(3) 他の認証メカニズムの利用

現在 SSL や TLS 等を始め、さまざまな分野で普及が広まっている公開鍵証明書の機構を利用し、マークの画像と公開鍵証明書をバインドさせ、検証システムとしては CA(認証局)を利用した公開鍵証明書の検証機構を利用するなど。

という選択肢しかないように思えてくる。

2.4 なりすましに対する新規技術動向

一方、IETF⁷, W3C⁸ は 1999 年 6 月より共同で HTML、XML などのコンテンツの署名並びに署名検証を行うプロトコル XMLSignature⁹が提唱されている。この XMLSignature では、暗号・認証技術を使うことにより、データベースを使わず、コンテンツの有効性などを電子署名とその有効性確認などや失効情報を公開鍵証明書を利用した PKI(Public-key Infrastructure)の機構を用いるメリットがある。しかし、XMLSignature はまだ検討段階であり、更に実装されたアプリケーションはない状況である。また、CRL¹⁰(証明書破棄リスト)をデータベースと捉えると、XMLSignature もデータベースを利用したプロトコルであるといえる。

XMLSignature では、電子署名を施し、検証するためのフォーマットであり、上部ページ単位でのコンテンツ、オリジナリティ、作成日付の保証をサービスしようとするものである。署名の行為をコンテンツ作成者ではなく、信頼された第3者によって行われれば、ウェブページを“公証人的”役割で保証をすることができる。

⁷ Internet Engineering Task Force(<http://www.ietf.org>)

⁸ World Wide Web Consortium(<http://www.w3.org>)

⁹ <http://www.w3.org/Signature/>

¹⁰ CRL(Certificate Revocation List) 公開鍵証明書におけるいわゆるブラックリスト

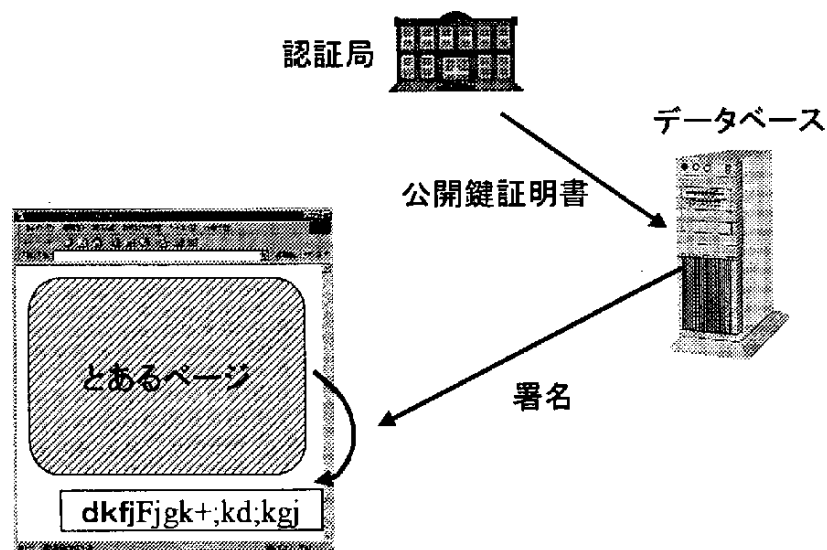


図 8 XMLSignature による書名

XMLSignature はその名の通りテキストベースであるが、一方、ビジュアル的なマークはバイナリデータである。しかし、一般的な電子署名という行為自体はバイナリデータについても行うこともできる。現時点では、XMLSignature ではバイナリデータであるマーク画像データをターゲットとしてないようであるが、利用の仕方ではマークの保証も行える可能性もある。

これらデータベースへ検証を行う機構については、3、4章のデータベースの設計の考察を踏まえ、再度5、6章で検討する。

3. 認定情報データベースの構築

3.1 認証情報の設計

ここでは、なりすましの問題を考えず、インターネット上での認証情報とそれを検証するためのデータベースの関係について効率的で効果的なデータベースを考える。

ここでは、ある資格認定を想定し、それを証明する事項について以下の項目を考えてみる。

- (1) 使用 URL(マークが表示されている URL)
- (2) 認証番号(ID)
- (3) 認証受諾者情報(企業名、氏名など)
- (4) 認証サービス名
- (5) 認証元
- (6) 認証の種類
- (7) 有効期限
- (8) コンタクト先
- (9) 有効・無効情報

などがある。先に述べた3つの事例でのマーク検証の方法について表1にまとめてみた。

	オフライン検証	オンライン検証
プライバシーマーク	(2)	(1),(3),(4),(5), (6),(7),(8),(9)
オンラインマーク	—	(1),(2),(3),(4), (5),(6),(7),(8),(9)
自己検証機能付きマーク	(1),(2),(3),(4),(5), (6),(7),(8)	(1),(2),(3),(4), (5),(6),(7)(8),(9)

表 1 検証の方法と内容

インターネットサービスを考えた場合、経路や回線の太さの問題にがある可能性があるため、問い合わせの頻度を減らした方がいいと思われる。また、現状の通信インフラを考えると、キャッシュを使わず、同じ情報を短い時間の間に何回も何回も不必要に問い合わせるのはあまり重要でないように思われる。先にも述べた通りウェブページに表示された時点で毎回マスターデータベースへ問い合わせるのは効率が悪いように思える。

検証対象 (シンボル) ⇔ 検証レファレンス (データベース)

を考えた場合、不変的なデータはシンボルに乗せ、あまり問い合わせる頻度の少ない変動データだけ、必要時に応じてデータベースに問い合わせるのが合理的なやり方であると思われる。

例えば、運転免許証を考えてみた場合、使用者の住所、氏名などの情報はシンボルである免許証に記述されている。住所変更した場合やその他の情報が変更になった場合、“裏書”として免許証の裏側に追記される。住所などの重要な情報の変更は再度再発行すべきところであるが、発行者側が“追記”という形で署名し、運営上簡略化し

で行っている処置であろう。

マークを事業者へ配布せず、検証データベースで管理する場合はデータベース側の変更で済む。しかし、マークを配布することを想定した場合、“追記”という形は難しい。技術的には可能であるが、处理的には再発行するのと同じであるため、記載事項の変更がある場合は再発行処理を行った方がいいだろう。認定を受けている事業者も記載情報が古いままであると検証に支障を来すので、再発行処理を怠ることはないと思われる。

このように変更の手間はあるものの、インターネット上では少なからず分散化を想定したネットワークであり、先にも述べたようにネットワーク・トラフィックの軽減、サーバの負荷、フォールト・トレランス、処理の時間などを考えた場合、必然的に分散化した方が望ましい。

オフライン、オンラインでの検証と分けた場合、オンラインにおける情報としては、

(9) 有効・無効情報

だけがデータベース情報として付加されるのみで十分である。すなわち、認定を受けた事業者全員が素直に無効の変更をするとはあまり思えないからである。

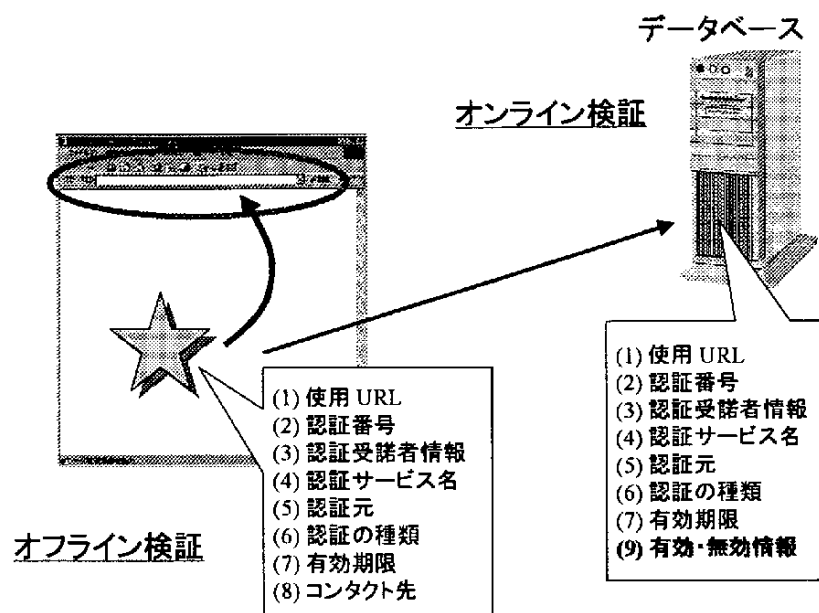


図 9 オフライン検証とオンライン検証

3.2 検証の方法

なりすましの問題は別として、検証対象であるマークからデータベースへ検証する方法は、よく使われている CGI などのコマンドを元にした URL をマークから取りだし、HTTP プロトコルによって検証対象のデータをデータベースへ渡し、結果を返すことになるだろう。

`http://v.gr.jp/cgi-bin/verify?license=A123456&url=http://a.b.co.jp/index.html`

ここでは、例として、マークが表示されている

・ ユーザ ID

- ・ マークが表示されているウェブページの URL

をデータベースに問い合わせ検証を行っていることを意味している。すなわち、

「<http://a.b.co.jp/index.html>」に表示されているユーザ ID「A123456」のマークは認証されたマークであり、かつ表示を許可された URL であるか？

を検証しているものである。なりすましを考えない場合、検証するために問い合わせる情報はこれだけで十分であると思われる。

失効情報だけを見るだけであれば、CRL のようなブラックリストを配布し、クライアントで検証することも可能である。サーバの負荷を軽減するためには望ましいスタイルである。しかし、ここでは述べないが、クライアント側でブラックリストの検証を行う場合、

- ・ 認証情報に変更がない
- ・ ブラックリストが偽造されない(正しい認証サーバから出されている)

ということが必須となる。

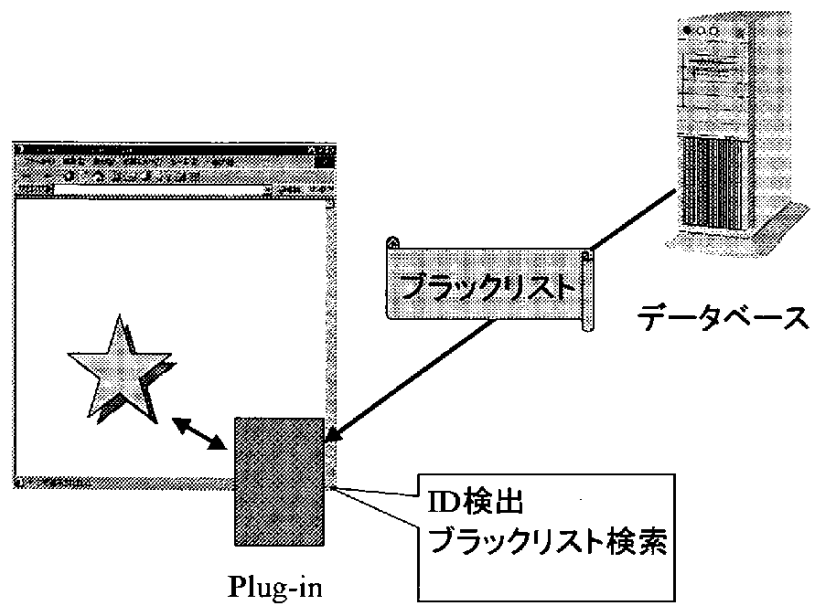


図 10 クライアントでの検証

4. 認定情報データベースの更新

4.1 サーバ-クライアント分散システム

先にも述べた通り、データベースの“更新”や“変更”の問題が必ず出てくる。一極集中的に管理を行う場合にはあまり考えなくてもいい事項であるが、サーバ-クライアント分散化を考えた場合、シンボルに情報を盛り込まなくてはならない。更新に関係するものは先の(1)～(9)の情報すべてが対象となる。

(1) 認証番号

(6) 有効期限

を考えてみる。資格を考える場合、いかなる場合でも「有効期限」が存在すると思われる。更新によって資格の「認証番号」が変わる場合と変わらない場合があると思うが、どちらにしても古い情報との整合性として「履歴情報」が必要になる。

認証番号が変わらない場合はそれをキーとして履歴情報を結びつければいいが、認証番号(ID)が変わる場合には別のユニークなシリアル番号を設ける必要がある。

また、マークのビジュアル的に変更点がない場合は、画像データ自体を変更する必要はないが、ユーザフレンドリの為にビジュアル的にも

(6) 有効期限

を合わせてマークとしてビジュアル化した方が望ましいので、必然的に画像データと合わせて更新した方がよい。

また、複数の認証サービスにも適用できるようにするには

(10) 検証先データベースの CGI などのコマンドの URL

をそれぞれの分散データベース先を明記した方が効率的である。

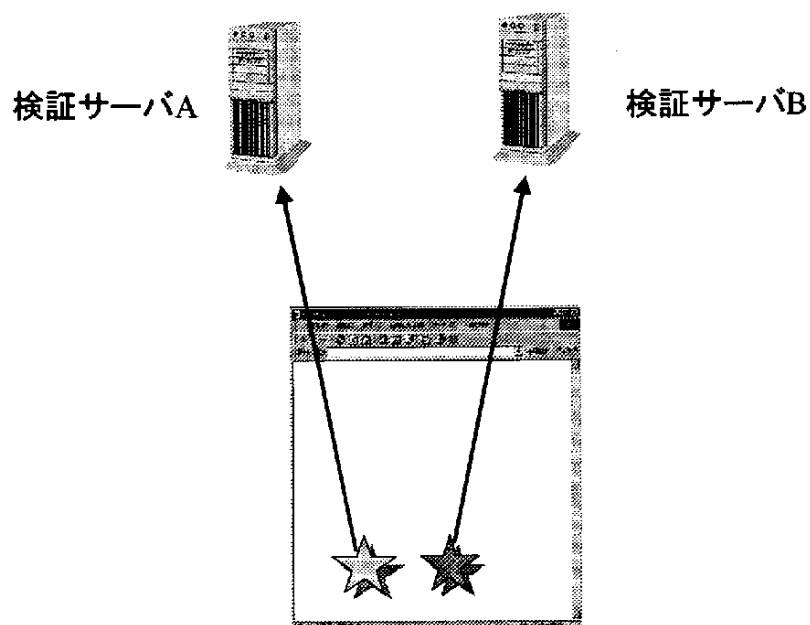


図 11 複数の認証マークの検証

まとめるとマークが持つ情報とデザインは図 12 のようになるのが望ましい。図 12 の例は極端であるが、認証マークは通常大きさの制約があるので、実際には表示項目も図 12 のようにはならないだろう。

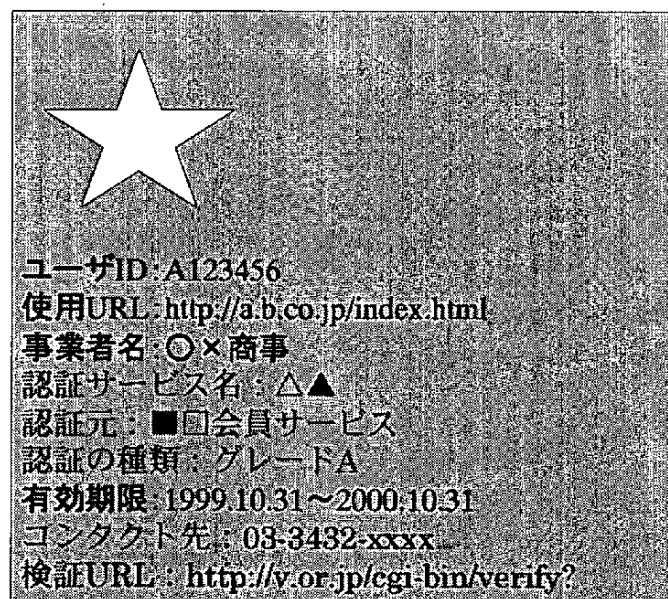


図 12 シンボルとしてのマークの例

あと、ユーザインターフェースを考えると、マークが持っているデータを用いて

- ・ 使用 URL のチェック
- ・ 有効期限の情報

はクライアントでも行えるにした方が望ましい。

4.2 サーバ-サーバ分散システム

1 台の検証サーバでサービスを行う場合は別として、認証サービス毎、同じ認証サービスでもグレード毎、地域毎で検証サーバの分散化を考えた場合、認証サーバ間、または認証サーバ-検証サーバ間でのデータの交換が必要となる。データ交換にセキュリ

ティが必要であれば、前年度の報告書¹¹で書かれたものを利用するとして、ここでは詳しく述べない。

ここでは認証データ自体の分散化を考える。分散化の例として、例えばプライバシーマーク事務局では、業界団体などの代理認証機関を設けている。認証を行う機関単位では、認定を受ける事業者やユーザ情報を蓄えており、認証データベースをも分散化して個別に管理した方が効率がよいと思われる。

また、先にも述べたように複数の認証サービスにも適用できるようにするには

(10) 検証先データベースの CGI などのコマンドの URL

をそれぞれの分散データベース先を明記した方が効率的である。分散化を行うのであれば、ユーザ(事業者)認証の他に

分散サーバの認証をどのように行うか？

が問題となる。分散サーバ自体もユーザ(事業者)同様に検証できる仕組みがあれば十分だと思われる。

¹¹ 複数認証局間における認証データベースの有効利用に関する調査研究

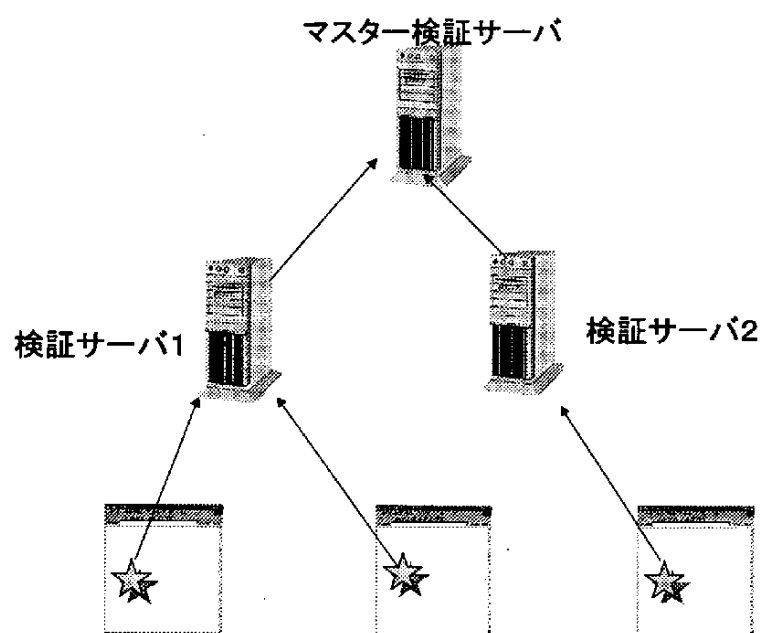


図 13 認証の階層構造

このように、検証サーバ自体も検証できる仕組みが必要となるが、これはクライアントがマークを認定するシステムをそのまま利用できるものと思われる。

5. 認定情報データベースの検証と認定認証

いくらデータベースの設計をうまくやっても検証サーバ自体のなりすましが行われたらどうしようもない。すなわち、

`http://v.gr.jp/cgi-bin/verify?license=A123456&url=http://a.b.co.jp/index.html`

の通りに、データ検証の手順をいくら正しく行ったとしても、その手順自体が不正であれば、正しい検証が行えないからである。ここで2章でのなりすましの問題を具体的に説明すると、

1. 検証情報データ 「A123456」や「`http://a.b.co.jp/index.html`」が偽造されている可能性がある
2. 検証サーバ「v.gr.jp」自体が偽造されている可能性がある

ということである。おさらいしてみると、プライバシーマークでは

- ・専用 plug-in による検証情報データと検証サーバ URL のなりすましを防止

オンラインマークでは

- ・HTTP_REFERER による認証と排除¹²

¹² 正しく言えば、HTTP_REFERER で「A123456」と「`http://a.b.co.jp/index.html`」を代用

を行っている。

5.1 マークの認証

認定を受けたユーザ(事業者)毎に **JAVA Applet** のようなプログラムを配布する方法は

プログラム自体のなりすましが容易に行える

危惧がある。しかし、プログラム自体の認証方法があれば大丈夫のように思われる。事実、**JAVA** には「**objectSign**¹³」と呼ばれる電子署名を施した **Applet** を作ることができ、**Netscape** では公開鍵証明書をインストールし、署名を行った公開鍵証明書をブラウザのユーザが意図的に使用許可することで使用可能となり、この行為自体で認証元を特定することができる。

¹³ <http://developer.iplanet.com/docs/manuals/java.html>

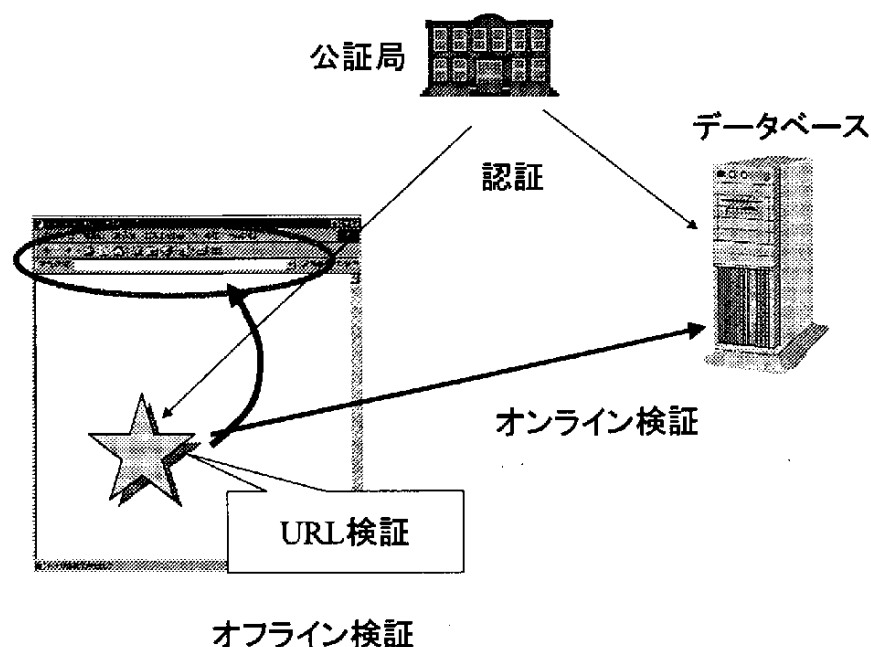


図 14 第3者認証付き自己検証マーク

しかし、悪意を持ったマークは検証先をも偽造する可能性があるので、「正当な場合」の設定では不正利用を検証できない可能性がある。プログラム自体を偽造されてしまえば、

- (1) plug-in のように検証プロセスをブラックボックス化する
- (2) 検証時に objectSign 自体を検証する
- (3) SSL などのクライアント認証を用いた通信を行う

などの検証を行っても、完全になりすまされてしまう。このような観点から分散管理を考える場合、画像データと検証プログラムは別にして、なおかつ検証プロセスを偽造されないようにする必要がある。

画像データの場合では、plug-in などの検証ツールで検証を行うためこのような成りすましは起きないものと思われる。

しかし、複数の認証サービスで発行しているマーク画像の検証を行うユニバーサルな plug-in では、検証サーバの URL をも認証データとしてマークに盛り込むので厄介である。なぜなら、

「マークを偽造すれば、正当な plug-in に対して偽造検証サーバへ問い合わせるよう仕組むことが可能」

だからである。これを防止するためには次に述べる「検証サーバの認証」をも行う必要がある。

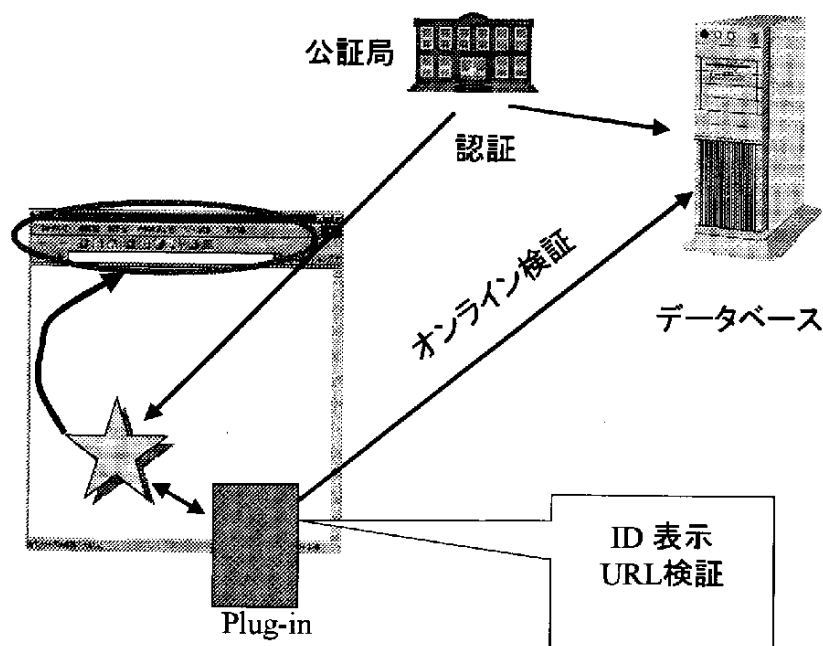


図 15 第 3 者認証付きマーク検証

5.2 検証サーバの認証

検証サーバもマーク検証と同様な処理で検証を行えると思われる。

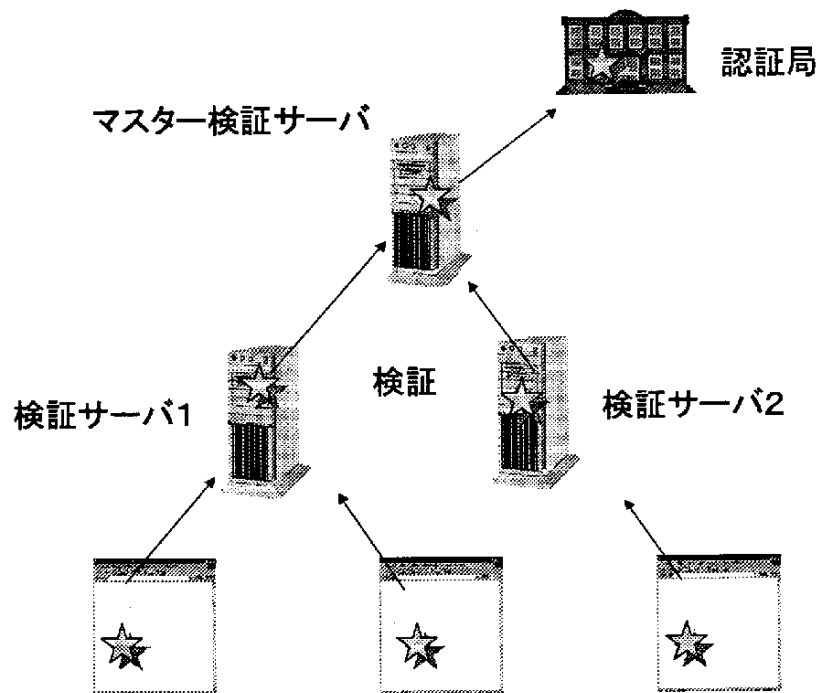


図 16 検証サーバの検証

6. なりすまし対策の検証機構

6.1 認証のチェーン

なりすましを防御するには、ブラウザは信用できるものとして(これを第1の拠り所として)

- ・マーク画像データ
- ・plug-in
- ・検証データベース

すべてについて、何かしらの認証が必要となる。公開鍵証明書の機構を用いれば、認証局をトップとした階層構造を作ることができ、これら認証のチェーンによってそれぞれを“正当なものである”と認証することが可能であり、なおかつ第2の拠り所として不正な利用を防ぐことが可能になる。

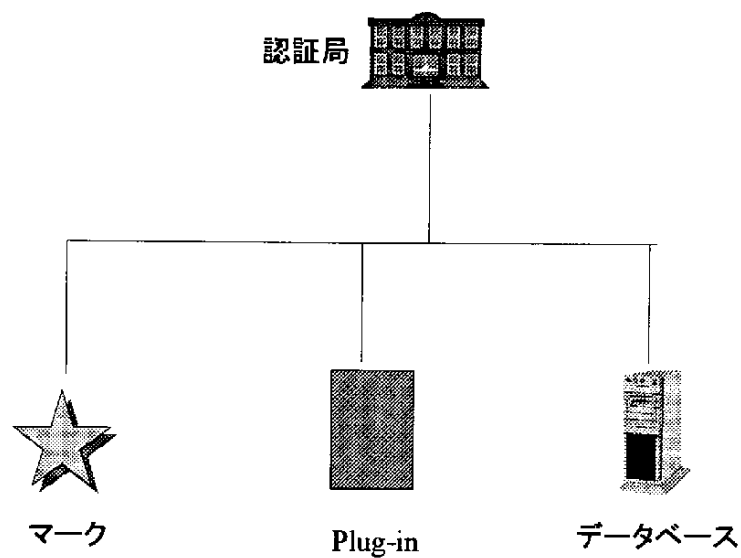


図 17 認証のチェーン

6.2 検証操作

プライバシーマークの検証では、**plug-in** のインストールと右クリックで操作して検証を行っており、オンラインマークでは通常の左クリックで検証を行っていた。自動検証を除き、このように簡単な操作で検証できるのが望ましい。しかし、**JAVA Applet** や **JAVA Script** でも可能な操作は偽造の恐れがある。

実際に検証する作業として、**plug-in** や別アプリケーションではなく、**SSL** や **S/MIME** などの署名検証のようにブラウザに標準で実装される方が望ましい。**SSL** や **S/MIME** では、ブラウザに専用ボタンがあり、これらをクリックすることで検証や検証情報の閲覧が行えるようになっている。

SSL や **S/MIME** のようにブラウザに標準になれば、ユーザフレンドリも高くなり、なおかつ、なりすましの問題が解決できると思われる。

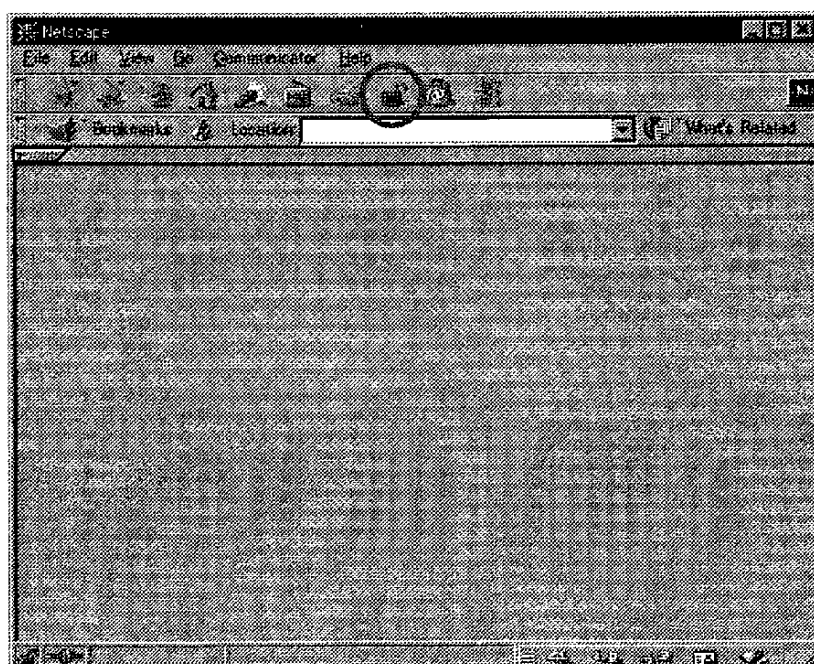


図 18 セキュリティボタン

6.3 画像フォーマット

画像データとして配布する形式を考える。

W3C が規格を定めた PNG¹⁴ という GIF に代わる画像フォーマットではかなりの文字列を収められるテキストエリアあり、ここに検証用のデータを入れることが可能である。しかし、ただテキストで入力すると、なりすまされる可能性があるので、

- ・ 認証元の証明書
- ・ データベース情報(認証データ)の署名
- ・ 画像データの署名

でテキストエリアに入れることとする。

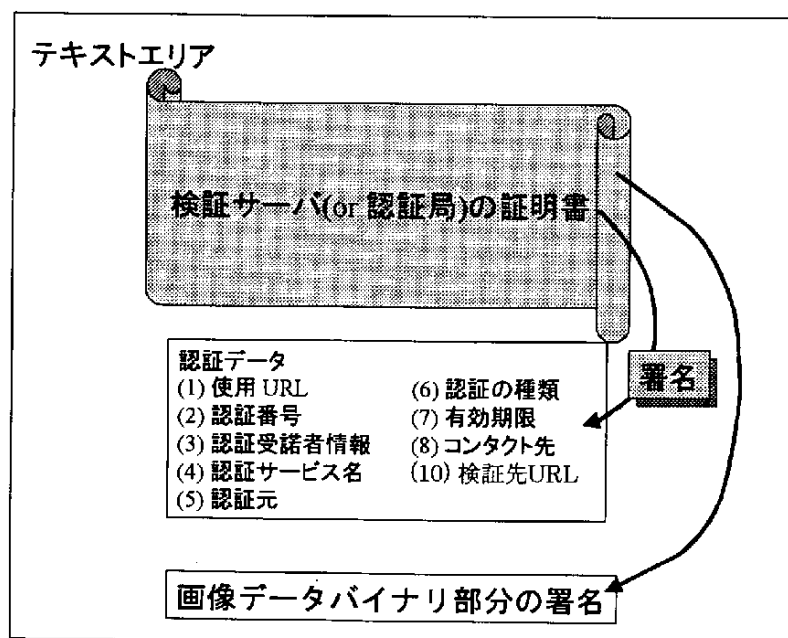


図 19 署名付き画像データのデザイン

¹⁴ Portable Network Graphic(<http://www.w3.org/Graphics/PNG/>)

認証情報を「証明書」として位置付けると、図 19 の「認証データ」部分は、検証サーバ(or 認証局)から発行された公開鍵証明書でもよい。認定事業者の公開鍵証明書を利用することで、

- ・公開鍵証明書伝播手段として画像データを使う

ということも可能になる。PEM のフォーマットに倣うと以下のようなフォーマットになる。

```
<cacert>
ASN.1 形式の認証局の公開鍵証明書
</cacert>
<cert>
ASN.1 形式の認定事業者の公開鍵証明書
</cert>
<sign>
ASN.1 形式の画像データのバイナリ部分の署名
</sign>
```

図 20 署名付き画像フォーマット

具体的には、次ページのようになる。

```
<cacert>
MIIDSTCCAawUCAQIwCwYHKoZlZjgEAwUAMIGKMQswCQYDVQQGEwJKUDE0MAwGA1UE
CBMFVG9reW8xKTAnBgNVBAoTIFByaXZhY3ItYXJrIE9mZmIjZSBNYXJrIENBIChe
U0EpMRowGAYDVQQDExFwcmI2YWN5bWFyay5nci5qcDEkMCIGCSqGS1b3DQEJARYV
ZGV2QHByaXZhY3ItYXJrLmdyLmpwMB4XDk5MDczMTAwMDAwMFoXDTAwMDMzMTAw
MDAwMFowYsxCzAJBgNVBAYTAkpQM4wDAYDVQQIEwVU2t5bzEqMCgGA1UEChMh
UHJpdmFjeW1hcmsgT2ZmaWN1IE1hcmsgUHVibGlzaGVyMRowGAYDVQQDExFwcmI2
YWN5bWFyay5nci5qcDEkMCIGCSqGS1b3DQEJARYVZGV2QHByaXZhY3ItYXJrLmdy
LmpwMIIBtjCCASsGBYqGSM44BAEwggEeAoGBAIA1pxp+oQQhp52qF21Eq6Sawi00
58ZglisUdacVhoubvAEiKb9aZRwWp7xxnkm0GnsVA6RFBAnzZxptctgqpG79nH+
a/vqXb7MrJhzrGadLmp/fjYpyRRN9/+1X0cCLS/SALWskIZ+dDnJpsnGJ5GU2fc0
lvW1zIJ2VRYxe/w3AhUAp0fMh64oY8LK4P6nBQPrzTXldkECgYAsyOMsgC6LsK+D
o0waz8J5uBJ6USGjb7WLXkEgBXJWxC3+0YUEJJ++ei5ONPXiq4xvxxvMKnl04k/M
B3L1DzAUYPDTwKW1m30m4hDzepW8Y7LmiecdhggB71Rcag2YAktnJO1DvN/5oqN
VJGm3D1LNAJ1C1a5tNvEBgF+J1C5FA0BhAACgYBDYoWZsKn4+1x3V1m3PW2JTjHB
J4P/TxfdvL5oy/wBd9vXq9sRqLu0YoXOMkHQ7psWf/E7Wnr/b90KAFajcJ18wH1N
14PTRK/i5nwz01rMfmj0ubpXTY+MoKznAhi/NmYAXMoK/Yg6qRPr62LbKt1v2lmz
+xtTxNLS36lklpjd2DALBgcqhkj00AQDBQADMQAawLgIWAJEeMw48Jvf50tQ56DyZ
aPkr6NazAhUAyZlZsZmap1nP2T4elszjD5YZHfU=
```

```
</cacert>
```

```
<cert>
```

```
MIIDITCCAoqgAwIBAgIBGzANBgkqhkiG9w0BAQQFADCBnTElMAkGA1UEBhMCSIAx
DjAMBGNVBAgTBVRva3lvMSUwIwYDVQQKExxQcmI2YWN5bWFyayBPZmZpY2UgQWNj
ZXNzIENBMRUwEwYDVQQLEwxFeHBlcmI2ZW50YWwGjAYBgNVBAMTEXByaXZhY3It
YXJrLmdyLmpwMSQwIlgYJKoZlhvcNAQkBFhVkJXZAcHJpdmFjeW1hcmsgT2ZmaWN1
HhcNOTkxMTAxMDAwMDAwWcNMDExMDMxMDAwMDAwWjCBYjELMAkGA1UEBhMCSIAx
EjAQBGNVBAgTCU1pbmF0by1rdTEgMB4GA1UEChMXUHJpdmFjeW1hcmsgQTEyMzQ1
Nig30CkxIDAeBgNVBAoTF1ByaXZhY3ItYXJrIFRlZmIjZSBNYXJrIE9mZmIjZSBNYXJr
EwxFeHBlcmI2ZW50YWwGjAYBgNVBAMTHWhOdHA6Ly9wcmI2YWN5bWFyay5nci5q
cC9kZXlYvMSQwIlgYJKoZlhvcNAQkBFhVkJXZAcHJpdmFjeW1hcmsgT2ZmaWN1
DQYJKoZlhvcNAQEBBQADgYOAMIGJAoGBALACYd+Z+/UVggaTlnZEDXi79uUVDv04
XWNknB/LdStlfiXeiMHVgKfOb//bEHJQ+OXE/huwmAOes6fyPfhtEs2JJJe/kfli
X81xZJwTGeVlspvH7CF7Rbi6m3+peb01Hk1P1Kk0CjoHQ5vFpubaqNaM/c8S0JR/
EBdXc+aMfJ9fAgMBAAGjQjBAMBOGA1UdDgQWBBrm3WJpKu/pl+wFGeKI//gNHAXI
wTAMBgNVHRMEBTADAQH/MBEGCWCsAGG+EIBAQQEAwIB9jANBgkqhkiG9w0BAQQF
AAOBgQCeNeUT9D7kUlkbc90+/mXNZKaAcWUkd5WwhsytuEic3xk8THgZEIliwPx
YnkrOdkoMsrGPt09Lb6emlwY0doEmXGLS28mXvJ6HMqPBuP6+q5Y7wxIudj+bBoa
enbLuK34YM29TtUyNDA3TbqhDBLxtQoEwMM1+zJt+MWY3rf1ug==
```

```
</cert>
```

```
<sign>
```

```
MCwCFHMBi7xrZLAW78dNQlpMbOG0hE1nAhQoe6Y7cZAgVbrxnZmMKz6LojpPUg
```

```
</sign>
```

認証データ(cert タグで囲まれた部分)の中身は

Certificate

issuer :/C=JP/ST=Tokyo/O=P Office Access CA/OU=Experimental/CN=v.gr.jp/ Email=d@v.gr.jp
subject:/C=JP/ST=Minato-ku/O=Privacymark A123456/O=P TEST Dev. 1/OU=Experimental/
CN=http://p.gr.jp/dev//Email=dev@p.gr.jp
serial :1B

Certificate:

Data:

Version: 3 (0x2)
Serial Number: 27 (0x1b)
Signature Algorithm: md5WithRSAEncryption
Issuer: C=JP, ST=Tokyo, O=P Office Access CA, OU=Experimental,
CN=p.gr.jp/Email=dev@p.gr.jp

Validity

Not Before: Nov 1 00:00:00 1999 GMT
Not After : Oct 31 00:00:00 2001 GMT
Subject: C=JP, ST=Minato-ku, O=P A123456(78), O= TEST. 1, OU= Experimental,
CN=http://p.gr.jp/dev//Email=d@p.gr.jp

Subject Public Key Info:

Public Key Algorithm: rsaEncryption

RSA Public Key: (1024 bit)

Modulus (1024 bit):

00:b0:02:61:df:99:fb:f5:15:83:26:93:96:76:44:
0d:78:bb:f6:e5:15:0e:f3:b8:5d:63:64:9c:1f:cb:
75:24:e5:7c:85:de:d4:c1:d5:80:a7:f4:6f:ff:db:
10:72:50:fb:45:c4:fe:1b:b0:98:0d:1e:b3:a7:f2:
3d:f8:6d:ac:4b:36:24:97:bf:91:f2:22:5f:cd:71:
64:9c:13:19:eb:e5:b2:9b:c7:ec:21:7b:44:18:ba:
9b:7f:a9:79:b3:a5:1e:4d:4f:d4:a9:0e:0a:3a:07:
43:9b:c5:a6:e6:da:a8:d6:8c:fd:cf:12:d0:94:7f:
10:17:57:73:e6:8c:7c:9f:5f

Exponent: 65537 (0x10001)

X509v3 extensions:

X509v3 Subject Key Identifier:

66:DD:62:69:2A:EF:E9:23:EC:05:19:E2:A5:FF:F8:0D:1C:0C:48:C1

X509v3 Basic Constraints:

CA:FALSE

Netscape Cert Type:

SSL Client, S/MIME, Object Signing

X509v3 CRL Distribution Points: ブラックリスト URI

URI:http://v.gr.jp/p.crl

Verification: 有効性検証 URI

URI:http://v.gr.jp/cgi-bin/verify.cgi

Signature Algorithm: md5WithRSAEncryption

9e:35:e5:13:f4:3e:e4:52:59:1b:71:bf:4e:fb:f9:97:35:92:
9a:01:c5:94:91:de:56:86:1b:32:b6:e1:08:73:7c:64:f1:31:
e0:64:49:65:8b:03:f1:62:79:2b:39:d9:28:32:ca:c6:3e:d3:
bd:2d:be:9e:9a:5c:18:39:da:04:99:71:8b:4b:6f:26:5e:f2:
7a:1c:ca:8f:06:e3:fa:fa:ae:58:ef:0c:48:b9:d8:fe:6c:1a:
1a:7a:76:cb:b8:ad:f8:60:cd:bd:4e:d5:32:34:30:37:4d:ba:
a1:0c:12:f1:b5:0a:04:c0:c3:35:fb:32:6d:f8:c5:98:de:b7:
f5:ba

という風になっている。

ここで問題となるのは上記データの取り出し方である。テキストデータであるので、エディターで開けるのであるが、一般にインターネットで利用できるようにするには画像データ内のテキスト情報を URI¹⁵として、上記データそれぞれを

`http://a.b.co.jp/pic/mark.png#cacert`

`http://a.b.co.jp/pic/mark.png#cert`

`http://a.b.co.jp/pic/mark.png#sign`

等のようにして、それぞれのデータをブラウザから取得できるようにする必要がある。

¹⁵ Uniform Resource Identifier

7. まとめ

7.1 データベースの形態

画像表示用のプログラムを配布し、そのプログラムで有効性検証をも行うことが可能であるが、

- ・ マーク画像の表示
- ・ 有効性検証

を同一のファイルで行うには、有効性検証行為すべてを偽造され、なりすますまされる危険がある。そこで、

- ・ マーク画像データ
- ・ 検証プログラム

を分離させ、それぞれになりすまし対策を施さなければならない。マーク画像データは広く伝播されるものであると考えると、

- ・ マーク画像データ
- ・ 検証データベース

との間のクライアント・サーバ間の分散データベースモデルと捉えることとなる。

また、異なる複数の認証サービスや検証サーバの分散化をも考えると、サーバ・サーバ間の分散データベースモデルも含まれることになる。

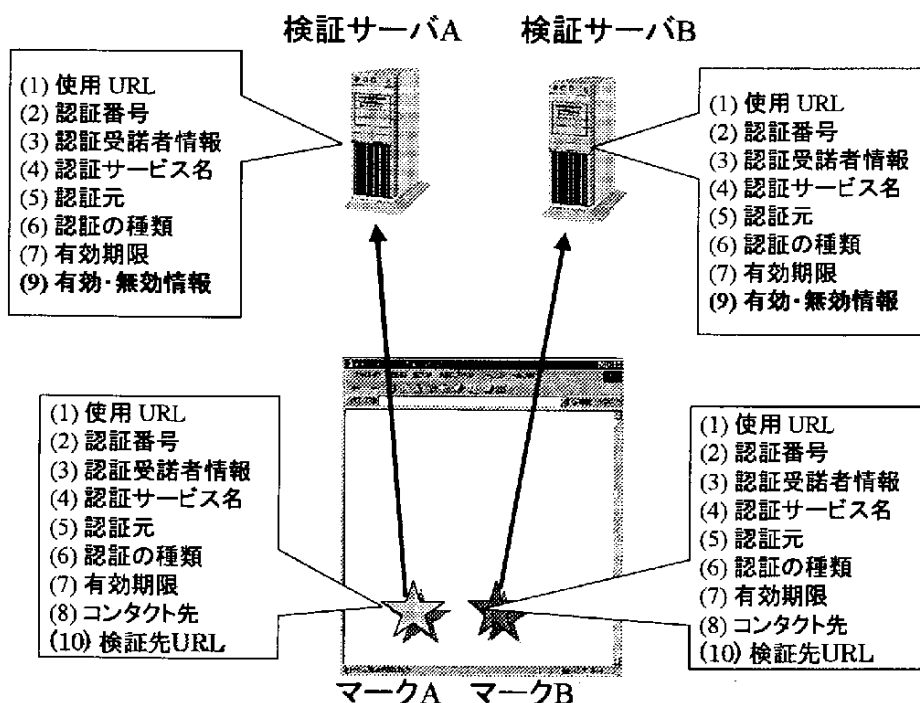


図 21 分散データベースモデル

7.2 データベースの更新

分散データベースでは必ずしも更新・変更作業が必要となる。認証データが変更になった場合、マーク画像データの再発行が望ましい。サーバだけ管理するデータは「破棄情報」だけサーバで行うことが望ましい。

サーバ・サーバ間は特にデータのやり取りを必要としない。必要である場合は本事業の10年度の下記の報告書¹⁶を参考に行うのが望ましい。

7.3 なりすましの防御対策

なりすましを防ぐには、以下のオブジェクトに対して何かしら“第3者的な”認証が必要となる。

¹⁶ 複数認証局間における認証データベースの有効利用に関する調査研究

- ・ 認定マーク画像データ
- ・ 検証プログラム
- ・ 検証データベース

公開鍵証明書機構を用いて上記のオブジェクトすべての認証を行うことを提案する。

7.4 検証機構の提案

画像フォーマットに識別子情報を盛り込む以下のようなフォーマットを提案した。

```
<cacert>
  ASN.1 形式の認証局の公開鍵証明書
</cacert>
<cert>
  ASN.1 形式の認定事業者の公開鍵証明書
</cert>
<sign>
  ASN.1 形式の画像データのバイナリ部分の署名
</sign>
```

と、これら認証情報を

<http://a.b.co.jp/pic/mark.png#cacert>

<http://a.b.co.jp/pic/mark.png#cert>

<http://a.b.co.jp/pic/mark.png#sign>

URI¹⁷として取り出せるように提案する。

8. 今後の課題

今まで検討してきたことは現在の技術で実装可能である。しかし、ユーザフレンドリを考えた場合、操作の容易性、わかりやすさ、汎用性を考えると

- ・ なりすまし防止の画像フォーマットの作成
- ・ 画像データ内の認証データの取り出すメカニズム
- ・ ブラウザへのマーク検証機能の組み込み

が必要となってくる。

¹⁷ Uniform Resource Identifier

—— 禁 無 断 転 載 ——

平成12年3月発行

発 行 財団法人データベース振興センター
東京都港区新橋二丁目13番8号
新橋東和ビル5階
TEL 03-3508-2430

委託先 株式会社シネジャーナルプロダクション
東京都新宿区四谷4丁目21番37号
TEL 03-3355-1594

印刷所 株式会社エイワ
東京都板橋区板橋2丁目14番8号
TEL 03-3579-6601

