社団法人 情報処理学会 研究報告
IPSJ SIG Technical Reports

2007－ＣＳＥＣ－38（３）
2007／7／19

分散属性認証方式のセキュリティと計算量について

上山 真貴子† 四方 順司‡ 松本 勉‡

†‡横浜国立大学大学院環境情報学府/研究院 〒240-8501 神奈川県横浜市保土ヶ谷区常盤台 79－7
E-mail: †ueyama@mlab.jks.ynu.ac.jp ‡{shikata, tsutomu}@ynu.ac.jp

あらまし 電子媒体を利用したサービスの増加・多様化に伴い，ユーザの属性を確認するための属性認証技術や，ユーザのプライバシ保護の需要が高まっている．筆者らはユーザが全ての個人情報を手元に集めるのではなく，秘密分散方式を利用して個人情報を分散管理する分散属性認証方式を提案している．これまで，サービス提供者がサービス利用条件を満たすユーザだけにサービスを提供できる不正利用防止機能とユーザが必要以上の個人情報を開示しないですむプライバシ保護機能を要件とする方式が，ディジタル署名技術や暗号化技術などの一般的なセキュリティ技術を利用して構成できることを示した．本稿では，この研究を発展させ，分散された属性情報を管理する属性認証機関へのアクセスの手間を軽減し，セキュリティ・プライバシについてもさらに手厚く保護する利便性の高い方式が構成できることを新たに示す．

キーワード 属性認証，秘密分散，プライバシ保護

On Security and Complexity of Shared Attribute Certification Systems

Makiko UEYAMA† Junji SHIKATA‡ and Tsutomu MATSUMOTO‡

†‡Graduate School of Environment and Information Sciences Yokohama National University
79-7 Tokiwadai, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-8501 Japan
E-mail: †ueyama@mlab.jks.ynu.ac.jp, ‡{shikata, tsutomu}@ynu.ac.jp

Abstract We are considering and proposing a shared attribute certification scheme with the aim of achieving user's privacy protection, in which authorities distribute and manage the user's privacy information by secret sharing scheme. In previous studies, we defined the notions of, as functions of the scheme, preventing illegal use of service and protecting the user's privacy . And we showed that can construct the scheme which has the two functions by combining some generic security techniques such as encryption schemes and digital signature schemes. In this paper, we propose the scheme which is more convenient and secure for user than the previous one.

Keyword Attribute Certification, Secret Sharing, User's privacy protection

1. はじめに

インターネットの普及により，電子媒体を利用したサービスが増加・多様化している．このようなサービスにおいては，しばしばサービス提供者がユーザの属性を確認する必要が生じるため，ユーザの属性情報を確認するための属性認証技術の需要が高まっている．その一方で，できるだけ属性などの個人に係わる情報の流出を防ぐようなプライバシの保護も求められている．そこで，我々はプライバシ保護を考慮した属性認証方式として，分散属性認証方式を提案し，考察している[1-3]．[1-3]で提案した分散属性認証方式は，従来の属性認証方式のモデルとは異なり，秘密分散を利用してユーザの属性情報を分散した分散属性情報と，ユーザの ID を組にして複数の機関に登録しておき，ユーザが選択した属性の種類に対応する分散属性情報を示す分散属性証明書(SAC : Shared Attribute Authority)を，一定数の機関が仮 ID をもとに発行し，ユーザが一定の数だけ SAC を集めて匿名属性証明書(AAC : Anonymous Attribute Certificate)としてサービス提供者に示す．必要以上にユーザのプライバシ情報を流出させないプライバシ保護機能と，従来の属性認証方式では考察されていなかった，サービスを利用するユーザと属性証明書を提示しているユーザが同じであることまでを要求するサービス不正利用防止機能を持つことが特徴である．また，分散属性認証方式がプライバシ保護機能と不正利用防止機能を持つことを一般的なセキュリティ技術であるディジタル署名技術などの要素技術の安全性に帰着させて証明した[3]．

本稿では，この研究を発展させ，分散された属性情報を管理する属性認証機関へのアクセスの手間を軽減し，セキュリティ・プライバシについてもさらに手厚く保護する利便性の高い方式が構成できることを新た

に示す．

本稿の構成は次の通りである．2 章でこれまでの研究を発展させた分散属性認証方式の概要を述べ，3 章でその構成例を示す．4 章で，安全性が離散対数問題と要素技術の安全性に帰着する分散属性認証方式を構成できることを証明し，5 章でこれまでの研究との差分を述べる．最後に，6 章でまとめる．

2. 分散属性認証方式の概要

本章では，新たに発展させた内容を踏まえた分散属性認証方式の概要と，持つべき機能として定めた機能要件及び機能要件を持つための条件として定めたセキュリティ要件を示す．

2.1. エンティティ

分散属性認証方式は，属性に応じたサービスを受けるにあたって，ユーザのプライバシを保護しつつ属性を示すような属性認証方式である．User, Service Point, Dealer, Shared Attribute Authority, 及び User Assistant のエンティティで構成される．

User：自身の属性を匿名属性証明書(AAC：Anonymous Attribute Certificate)を用いて示そうとするエンティティ．ID が X である User を User X と呼ぶ．

Service Point：User の属性を検証するエンティティ．属性に応じて User にサービスを提供する．ID が Y である Service Point を Service Point Y と呼び，Y は公開されている．例えば，User の属性に応じた Web サービスを提供するサーバが相当する．また，Shared Attribute Authority を信頼している．

Dealer：User の属性を秘密分散方式によって分散し，User に対応する User Assistant と，Shared Attribute Authority に配布するエンティティ．いかなる不正もしない．例えば，市役所などの信頼できる機関が相当する．

Shared Attribute Authority：User の分散属性情報と ID を組にして管理し，User の要求に応じて分散属性証明書(SAC：Shared Attribute Certificate)を生成するエンティティ．Dealer を信頼している．また，定められたプロトコルから逸脱する行為をしない．ID が SAA_i である Shared Attribute Authority を Shared Attribute Authority SAA_i と呼び，これが生成した分散属性証明書を SAC_i とする(i=2,3,…,n，n は正整数)．SAA_i は公開されている．ここで，Shared Attribute Authority SAA_i が管理する User X に関する情報を User X の分散属性情報を $a(X, i)$と表す．例えば，データベースサーバが相当する．

User Assistant：User と 1 対 1 に対応し，User に代わって計算等を行うエンティティ．SAC_i 生成を要求するほか，自らも分散属性証明書 SAC_X を生成し，SAC_i と合わせて Service Point に示す AAC を生成する．User X に対応する User Assistant を UA(X)と呼び，対応する User X 以外は使用できない．例えば，User X だけが使用する端末が相当する．UA(X)が管理する User X の分散属性情報を $a(X,X)$と表す．

2.2. 概要

分散属性認証方式では，User が Service Point Y に AAC を提示し，合格した場合にサービスを利用する．User はサービスを利用しようとするたびに新たに AAC を生成して提示する．User X が Service Point Y のサービスを利用する場合を例に，分散属性認証方式の概要を 4 つのフェーズに分けて説明する．また，図 1 に概要を示す．

登録フェーズ：User X が Dealer に自身の属性情報 $a(X)$ を示し，Dealer が $a(X)$を確認した後，秘密分散方式によって分散属性情報 $a(X, i)$および $a(X, X)$に分散して，それぞれ各 SAA_i および UA(X)に登録する．

AAC 提示要求フェーズ：User X が UA(X)を通じて，Service Point Yにサービス利用要求を出し，Service Point Yが AAC 提示要求を UA(X)に送り返す．次に，UA(X)が Shared Attribute Authority SAA_i に SAC_i 生成要求を出す．Shared Attribute Authority は SAC_i を生成して UA(X)に送り返す．

AAC 生成フェーズ：UA(X)が SAC_X を生成し，SAC_i と合わせて AAC を生成して Service Point Yに AAC 検証要求を送る．

サービス利用フェーズ：Service Point Y が AAC を検証し，合格した場合に User X に UA(X)を通してサービスを提供する．

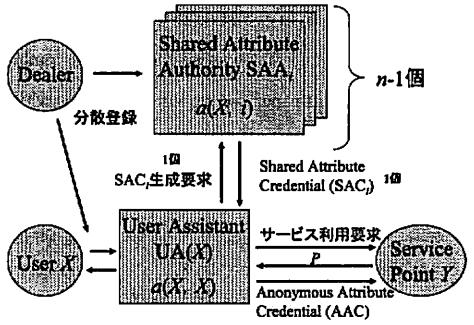

図 1　分散属性認証方式の概要

2.3. 機能と安全性

分散属性認証方式が持つべき機能を機能要件と呼び，分散属性認証方式が機能要件を持つために満たすべき条件をセキュリティ要件と呼ぶ．以降では，Service Point にサービス利用要求をするエンティティをサービス利用要求者，Service Point に AAC を示すエンティティを AAC 提示者，AAC を検証するエンティティを AAC 検証者，Service Point がサービスを提供するエンティティをサービス利用者と呼ぶ．

2.3.1. 機能要件

分散属性認証方式における機能要件は，以下の2つである．

・サービス不正利用防止機能

・プライバシ保護機能

以下に各機能要件の定義を示す．

定義 2. 1. [サービス不正利用防止機能]：「AAC 提示者の属性=AAC が示す属性＝サービス利用者の属性」となるときだけ，サービスを利用できる機能をサービス不正利用防止機能と呼ぶ．

定義 2. 2. [プライバシ保護機能]：User の個人情報を他のエンティティに必要以上に知られることが難しいような機能をプライバシ保護機能と呼ぶ．具体的に，次のような機能を定める．

(α)Shared Attribute Authority が管理する情報から属性情報を知ることが難しい．

(β)AAC 検証者が AAC に示された以上の属性情報を知ることが難しい．

(γ)SAC_i や AAC から Service Point に関する情報を得ることが難しい．

より厳密には，定義 2.5. の関連付け困難性を満たすことがプライバシ保護機能に相当する．

2.3.2. セキュリティ要件

攻撃者の能力を次のように定める．

・各エンティティ間の通信を盗聴できる．

・使用済みの AAC を入手できる．ただし，内容を指定して入手することはできない．

・Shared Attribute Authority が管理する属性情報を入手できる．ただし，Shared Attribute Authority の秘密鍵を入手することはできないものとする．

・自身が持つ情報に加えて，上記の攻撃で得た情報を使用して各エンティティと通信できる．

攻撃者として，Dealer と Shared Attribute Authority 以外のエンティティおよび分散属性認証方式に定めていない外部のエンティティを考える．ただし，自身の不利にあたる行為は行わないものとする．すなわち，次の行為である．

・攻撃者が User または User Assistant である場合，自らのプライバシを正当なサービス利用に必要な範囲以上に漏洩しない．

・攻撃者が Service Point である場合，不正なサービス利用につながる情報を漏洩しない．

また，登録フェーズおよび User と User Assistant 間のアクセスは正しく行われ，攻撃は考えない．

以上の攻撃者を想定したとき，分散属性認証方式が定義 2.1. と定義 2.2. に定めた機能要件を持つためのセキュリティ要件は，次の3つである．

・偽造困難性

・なりすまし困難性

・関連付け困難性

以下にセキュリティ要件の定義を示す．

定義 2. 3. [偽造困難性]：あるエンティティが，Service Point Y にサービス利用要求をした上で，自身と異なる属性を示す AAC'を提示して認証させることを偽造と呼び，偽造を試みる攻撃者を Fa と呼ぶ．Fa が偽造に成功することが難しいことを，偽造困難性と呼ぶ．

定義 2. 4[なりすまし困難性]：あるエンティティが Service Point Y にサービス利用要求をせずに，他の User になりすまして AAC を提示して認証させることをなりすましと呼び，なりすましを試みる攻撃者を Fb と呼ぶ．Fb がなりすましに成功することが難しいことを，なりすまし困難性と呼ぶ．

定義 2. 5. [関連付け困難性]：あるエンティティが，次のいずれか1つを行うことによって自身とは異なる User のプライバシ情報を得ることを関連付けと呼び，関連付けを行う攻撃者を Fc と呼ぶ．Fc が関連付けに成功することが難しいことを関連付け困難性と呼ぶ．

属性と User の関連付け：任意の Shared Attribute Authority SAA_i が管理する任意の User X に関する任意のデータから，その属性を知ること．

証明書と User の関連付け:2つの匿名属性証明書 AAC と AAC'があるとき，それぞれに対応する User が同じであるか否かを判定すること．

証明書とサービスの関連付け：AAC と AAC'が利用された Service Point が同じであるか否かを判定すること．

3. 分散属性認証方式の構成例

本章では，User Assistant が Shared Attribute Authority にアクセスする手間を軽減し，セキュリティ・プライバシについてもさらに手厚く保護する利便性の高い分散属性認証方式を，一般的なセキュリティ技術を要素技術として利用して構成した例を示す．

3.1. 準備

準備として，次の作業を行う．

属性領域の設定：L 個の属性種があるとき，属性名を $1, \ldots, L$ とし，属性名 1 に対応する属性値集合を Set-1，属性名 2 に対応する属性値集合を Set-2,...，属性名 L に対応する属性値集合を Set-L とする．全ての属性値を含む属性領域 AD は，次の直積で表すことができる．(L は正整数)

$$AD(1, \ldots, L) = \prod_{j=1}^{L} \{(j, \xi) \mid \xi \in Set\text{-}j\}$$

選択関数の設定：L 個の属性名に対応する属性領域

$AD(1, \ldots, L)$から，k個の属性名に対応する属性領域 $AD(P1, \ldots, Pk)$を選び出す選択関数を，$P_{p1,p2,\ldots,pk}(AD(1, \ldots, L)) = AD(1, \ldots, Pk)$で表す．

$$P_{p1,p2,\ldots,pk}(\prod_{j=1}^{L}\{(j,\xi)\,|\,\xi \in Set\text{-}j\})$$

$$= \prod_{j=P1,P2,\ldots,Pk}\{(j,\xi)\,|\,\xi \in Set\text{-}j\}$$

ただし，kは1よりも大きい正整数で，$P1, \ldots, Pk$は次の関係を満たす整数である．

$0 < P1 < \cdots < Pk$

3.2. 利用するセキュリティ技術

以下のセキュリティ技術を用いて構成する．

ディジタル署名方式：EUF-CMA 安全性(選択文書攻撃のもとで存在的偽造不可)を満たすディジタル署名方式を利用する[7]．つまり，正当な署名生成鍵を持つ署名者以外がディジタル署名を偽造することを許さない．本構成例では，Service Point から UA(X)に向けた通信の認証に Service Point Yのディジタル署名を利用し，UA(X)と各 Shared Attribute Authority SAA_i 間の認証に互いのディジタル署名を利用する．また，SAC_iを Shared Attribute Authority SAA_iのディジタル署名付きデータとする．

暗号化方式：IND-CCA 安全性(選択暗号文攻撃のもとで識別不可)を満たす公開鍵暗号化方式を利用する[8]．つまり，復号鍵を持たずに暗号文から何の情報も得られない．本構成例では，UA(X)と各 Shared Attribute Authority SAA_i間の通信を互いの公開鍵で暗号化する．また，UA(X)から Service Point Yへの通信は Service Point Yの公開鍵で暗号化し，Service Point Yから UA(X)への通信は，pk_Xで暗号化する．pk_Xは，User Xが Service Point Yにサービス利用要求をするときに一時的に生成して渡した暗号化鍵であるので，Service Point はpk_Xから User Xを特定できない．さらに，User Assistant と Service Point 間にプロキシサーバを置くなどして Service Point から User を匿名化するものとする．

秘密分散方式：2-out-of-n 分散方式を利用する．本構成例では，Dealer が属性種ごとに Shamir の(2, n)秘密分散方式[4]を利用して秘密分散し，分散後の Shared Attribute Authority SAA_iおよび User Assistant による分散属性値の変換に Proactive Secret Sharing の概念を利用する[5]．また，User Assistant が生成する SAC_Xを検証するため検証可能秘密分散方式の概念も利用する[6]．Attribute Authority SAA_iが記録するある属性値 S の分散値 S_i，$UA(X)$が記録する分散値 S_X，及び乱数 r で変換した分散値 S'_i，S_X'を次のように定める．

$u, v : u\,|\,(v-1)$である素数．公開されている．$(u > n)$

$\mathbf{Z}_v^*$: 位数がuである乗法群

$g : \mathbf{Z}_v^*$の要素．公開されている．

$S, r, m_x \in \mathbf{Z}_u$

$f(x) = S + m_X x \bmod u, \quad \delta(x) = f(x) + rx \bmod u$

$S_i = f(i), \quad S_X = f(1)$

$S_i' = \delta(i) = S_i + ri \bmod u, \quad S_X' = \delta(1) = S_X + r \bmod u$

ただし，u は十分大きく，m_Xは Dealer が User Xの属性情報を秘密分散のための計算をするたびにランダムに選ぶ値とする．このとき，次のようにして2個の分散属性情報 S'_iおよびS'_XからSを復元できる．

$$S = \delta(i)\frac{(0-1)}{(i-1)} + \delta(1)\frac{(0-i)}{(1-i)} \mod u$$

ハッシュ関数：出力するハッシュ値の長さが d ビットである衝突困難性を持つハッシュ関数 $H[\cdot]$を利用する．dは十分大きいものとする(dは正整数)．

3.3. 構成例

以下に，User Xが Service Point Yのサービスを利用するまでを4つのフェーズに分割して説明する．各フェーズの最後の( )内に，図2と対応するプロセスの番号を示す．

登録フェーズ：Dealer は User の属性情報を属性種ごとに秘密分散する．L 種類の属性に関する User Xの属性情報 $a(X)$及び Shared Attribute Authority SAA_iが管理する User Xの分散属性情報 $a(X,i)$，UA(X)が管理する分散属性情報 $a(X, X)$を次に示す．

$$a(X) = AD_X(1,\ldots,L) = \prod_{j=1}^{L}\{(j, a(X)\_j)\}$$

$$a(X,i) = AD_{X,i}(1,\ldots,L) = \prod_{j=1}^{L}\{(j, a(X,i)\_j)\}$$

$$a(X,X) = AD_{X,X}(1,\ldots,L) = \prod_{j=1}^{L}\{(j, a(X,X)\_j)\}$$

$$a(X,i)\_t = a(X)\_t + m_X i \bmod u \quad (t = 1,2,\cdots L)$$

$$a(X,X)\_t = a(X)\_t + m_X \bmod u \quad (t = 1,2,\cdots L)$$

$$a(x)\_t \in \mathbf{Z}_u \quad (t = 1,2,\cdots L)$$

Dealer は，X，$G_X = g^{m_X} \bmod v$および属性種ごとの有効期限と a(X, i)を各 SAA_iに登録し，UA(X)には属性種ごとの有効期限と a(X, X)を渡す(①②)．その後，Dealer は User Xに関するすべての情報を消去する．Shared Attribute Authority SAA_iが記録する User Xに関する情報は，$(X, a(X,i), G_X$,属性ごとの有効期限)の組である．ここで，$a(X)\_t$ は，属性名 t に対応する User Xの属性値であり，$a(X, i)\_t$ は，Shared Attribute Authority SAA_iが管理する，属性名 t に対応する User Xの分散属性値である．

AAC 提示要求フェーズ：UA(X)はサービス利用要求ご

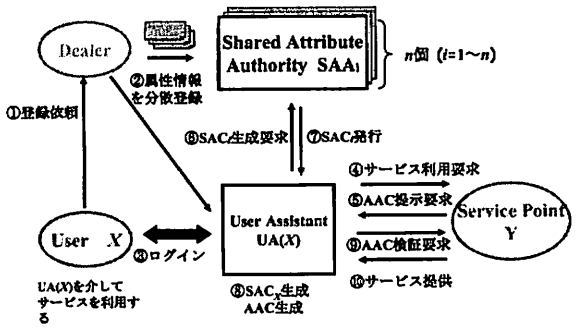

図 2　分散属性認証方式

とに一時的に利用する暗号化鍵と復号鍵のペア(pk_X, sk_X)を生成して記録した後，Service Point Yにサービス利用要求とpk_Xを送る．Service Point Yは，サービス利用要求ごとに仮 ID として乱数である q，変換係数として乱数である r_1 を新たに生成し，必要な属性種を選択する関数 P とともに(q, r_1, P)の組を AAC 提示要求として UA(X)に送る．また，Service Point Y は Q=H[q]⊕Y と R_1=H[r_1]，W=g^{r1} mod v を計算し，短い有効期間だけ(Q, R_1, W, pk_X)を記録し，q, r_1 をすぐに消去する．ここで，r_1 の長さを z ビットとし，z は十分大きいものとする(③~⑤)．

AAC 生成フェーズ：UA(X)が，AAC 提示要求から得た q, r_1 を用いて Q=H[q]⊕Y と R_1=H[r_1]を計算する．次に，UA(X)が必要な属性種情報とともに(Q, r_1)と自身の ID である X を SAC$_i$ 生成要求として，任意の Shared Attribute Authority SAA$_i$ に送る．ここで，UA(X)は (Q, R_1)を SAC$_i$ 検証をするまで記録しておき，q, r_1 をすぐに消去する(⑥)．Shared Attribute Authority SAA$_i$ は，管理している情報から SAC$_i$ 生成要求に対応する属性種の分散属性情報を選択して P(a(X, i))を得る．次に，Shared Attribute Authority SAA$_i$ は P(a(X, i))を構成する分散属性情報が全て有効期限内であった場合に，新たに乱数 r_2 と，R_2=$G_X g^{r2}$ mod v と r=r_1+r_2 mod u を生成する($r_2 \in Z_u$)．また，r を用いて P(a_r (X, i))を生成し，R_2 と，SAC$_i$ 生成要求の Q，Shared Attribute Authority SAA$_i$ が計算した R_1=H[r_1] とあわせてディジタル署名を添付した分散属性証明書 SAC$_i$ を生成し，r_2 とともに(SAC$_i$, r_2)の組にして UA(X)に送る．a_r(X, i)は，a(X, i)の各属性名に対応する分散値を r を用いて変換した情報を示す(⑦)．

$$a_r(X,i) = AD_{X,i,r}(1,\ldots,L) = \prod_{j=1}^{L}\{(j, a_r(X,i)\_j)\}$$

$$a_r(X,i)\_t = a(X,i)\_t + ri \bmod u \quad (t = 1,2,\cdots,L)$$

UA(X)は，送られてきた SAC$_i$ を検証する．以下に SAC$_i$ の構成を示す．

$$SAC_i = \{Q, P(a_r(X,i)), R_1, R_2\}$$

上記の SAC$_i$ の検証では，次の条件を全て満たすときに合格とする．

(Ⅰ)SAC$_i$ が，Shared Attribute Authority SAA$_i$ の公開鍵によってディジタル署名の検証に合格する

(Ⅱ)SAC$_i$ が含む Q が記録している Q と等しい

(Ⅲ)SAC$_i$ が含む R_1 が記録している R_1 と等しい

SAC$_i$ が合格した場合，UA(X)は r=r_1+r_2 mod u および P(a$_r$(X, X))を生成し，SAC$_X$ を生成する．SAC$_X$ の構成を次に示す．a_r(X, X)は，a(X, X)の各属性名に対応する分散値を r を用いて変換した情報を示す．

$$SAC_X = \{Q, P(a_r(X,X))\}$$

$$a_r(X,X) = AD_{X,X,r}(1,\ldots,L) = \prod_{j=1}^{L}\{(j, a_r(X,X)\_j)\}$$

$$a_r(X,X)\_t = a(X,i)\_t + r \bmod u \quad (t = 1,2,\cdots,L)$$

次に，UA(X)は SAC$_i$，SAC$_X$ をまとめて匿名属性証明書 AAC を次のように生成し，AAC 検証要求として Service Point Y に送る (⑧～⑨)．

$$AAC = \{Q, SAC_i, SAC_X\}$$

サービス利用フェーズ：Service Point Y は AAC を検証し，合格した場合にだけ，AAC に対応する Q と一緒に記録していた pk_X でサービスを暗号化して UA(X)に送り返す．また，Service Point Y は AAC が合格した場合と，Q の有効期限が過ぎたときに(Q,R_1,W, pk_X)を記録から消去する(⑩)．AAC 検証では，SAC$_i$ 検証に加えて，次を満たすときに合格とする．

(Ⅱ')SAC$_X$ が含む Q が記録している Q と等しい

(Ⅳ)AAC から得る P(a(X))が Service Point Y のサービス利用条件を満たす．

(Ⅴ)各属性名 t について，AAC から得た a(X)_t と a_r(X, i)_t が次の関係を満たす．

$$g^{a_r(X,i)\_t - a(X)\_t} \equiv (R_2 W)^r \pmod v$$

ここで，Shared Attribute Authority SAA$_i$ が SAC$_i$ 生成時に属性情報の有効期限を判定し，AAC が十分短い有効期限を持つ一度しか使えない証明書であるので，AAC 検証時に失効情報などの確認をする必要がない．UA(X)はサービス受け取り後(pk_X, sk_X)を消去する．

以上の Shared Attribute Authority SAA$_i$ と UA(X)，Y 間の情報のやりとりを図 3 に示す．

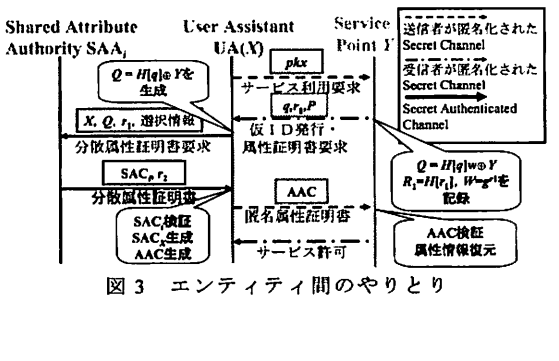

図 3　エンティティ間のやりとり

4. 安全性の検討

3 章で示した分散認証方式の構成例が，先に定めた機能要件を持つことを示す．

4.1. サービス不正利用防止機能

まず，認証機能を定義し，3 章で示した分散属性認証方式の構成例が認証機能を持つことを証明してから，サービス不正利用防止機能を持つことを証明する．

以降では，攻撃者 F が Service Point Y に提示する匿名属性証明書 AAC'を次の 2 つの分散属性証明書で構成されているとする．

・$SAC_i'=\{ Q',\ P(a_{r'}(X',\ i)),\ R_1',\ R_2'\}$

・$SAC_F'=\{Q_F, P(a_{rF}(F,\ F))\}\}$

$(Q'=H[q']\oplus Y',\ R_1'=H[r_1'],\ R_2'=G_{X'}g^{r2'} \bmod v)$

SAC_i'は，次の 3 種に分類できる．

(1)F が SAC 生成要求によって Shared attribute Authority SAA_i から入手した SAC_i (Q'，r_1'は任意，$r_2'=r_2$ は Shared Attribute Authority SAA_i が生成した値)

(2)F ではない User X が SAC_i 生成要求によって Shared Attribute Authority SAA_i から入手した SAC_i' (Q'，r_1'は User X が選択した値，$r_2'=r_2$ は Shared Attribute Authority SAA_i が生成した値，$X'=X$)

(3)F が Shared Attribute Authority SAA_i によらず偽造した SAC_i' (Q'，$P(a_{r'}(X',\ i))$，R_1'，R_2'は任意)

定義 4. 1. [認証機能]：「サービス利用要求者の属性＝AAC が示す属性」であることを AAC 検証者が正しく判定できる機能を認証機能と呼ぶ．

定理 4. 1. [認証機能]：分散属性認証方式は，偽造困難性となりすまし困難性を満たせば認証機能を持つ．

(証明) 省略する．

定理 4. 2. [偽造困難性]：3 章で示した分散属性認証方式の構成例は偽造困難性を持つ．

(証明)攻撃者 Fa が Service Point Y に AAC'を提示して偽造に成功したと仮定する．つまり，Service Point Y が仮 ID である q を Fa に発行したとき，Fa は，$Q=H[q]\oplus Y$ を含むが $P(a(Fa))$ と異なる属性を示す AAC'を提示する．

(a-1) SAC_i'が(1)に分類される場合

Fa が操作可能な値は，r_1'と SAC_{Fa}'である．q，Yはその値を知っているので，そのまま使用する．AAC'が SAC_i'検証の(Ⅰ)に合格するならば，SAC_i 生成要求に使用するディジタル署名の EUF-CMA 安全性により，$X'=Fa$ である．また，(Ⅴ)の検証に合格するならば，次の式を満たす($t\in\{1,\cdots,L\}$)．

$a_{Fa}(Fa,Fa)\_t\equiv a_r(Fa,Fa)\_t+(r_1'-r_1)i \pmod u$

ここで，(Ⅲ)の検証に合格するならば，ハッシュ関数の衝突困難性により $r_1'=r_1$ となるので，$r'=r$，$a_{Fa}(Fa,Fa)_t\equiv a_r(Fa,Fa)_t \pmod u$となり，偽造にならない．

したがって，Fa は偽造ができない．

(a-2) SAC_i'が(2)に分類される場合

User X は生成した SAC_i'を Service Point Y'に送る．$Fa\neq Y'$ならば，暗号化方式の IND-CCA 安全性により，User X が使用した q'，Y'，r_1'，使用前の AAC を入手できないため，Fa は使用済みの SAC_i'だけ入手できる．しかし，ハッシュ値の長さ d が十分大きいことにより，AAC'が SAC_i'検証の(Ⅱ)(Ⅲ)に合格するために必要な $H[q']\oplus Y'=H[q]\oplus Y$ と $H[r_1']=H[r_1]$を同時に満たす SAC_i'を入手できない．

$Fa=Y'$ならば，Fa は適当な q'，r_1'を選択して，User X を介して SAC_i'を入手できる．しかし，ハッシュ関数の衝突困難性により，SAC_i'が SAC_i'検証の(Ⅱ)に合格するために必要な $H[q']\oplus Fa=H[q]\oplus Y$ を満たす q'を見つけることができない．したがって，Fa は SAC_i'を入手できず，偽造できない．

(a-3)SAC_i'が(3)に分類される場合

ディジタル署名方式の EUF-CMA 安全性により，Fa は SAC_i'検証(Ⅰ)に合格するような SAC_i'を生成できないので，偽造できない．

以上のことから，3 章で示した分散属性認証方式の構成例は偽造困難性を持つ．

定理 4. 3. [なりすまし困難性]：3 章で示した分散属性認証方式の構成例はなりすまし困難性を持つ．

(証明)Service Point Y に対してなりすましに成功する攻撃者 Fb を仮定する．Service Point Y が仮 ID である q を Fb ではない User X に発行したとき，Fb は $Q'=Q$ ($Q=H[q]\oplus Y$)を含むが $P(a(X))$と異なる属性を示す AAC'を提示する．

(b-1)SAC_i'が(1)に分類される場合

User X は，Q に対応する AAC を Service Point Y に送る．Fb は暗号化方式の IND-CCA 安全性により q，r_1 を知ることができない．また，ハッシュ関数の衝突困難性とハッシュ値の長さ d が十分大きいことにより，Fb は適当な SAC_i'を生成してもらうために必要な Q'，r_1'を知ることができない．したがって，Fb は SAC_i'を入手することができず，なりすましできない．

(b-2)SAC_i'が(2)に分類される場合

(a-2)と同様に，暗号化方式の IND-CCA 安全性とハッシュ関数衝突困難性及びハッシュ値の長さ d が十分大きいことにより，Fb は SAC_i'を入手できない．したがって，なりすましできない．

(b-3) SAC_i'が(3)に分類される場合

(a-3)と同様にディジタル署名方式の EUF-CMA 安全性により，SAC_i'検証(Ⅰ)に合格するような SAC_i'を生成できないので，なりすましできない．

以上のことから，3 章で示した分散属性認証方式の構成例はなりすまし困難性を持つ．

定理 4. 4. [サービス不正利用防止機能]：3 章で示した分散属性認証方式が偽造困難性となりすまし困難性を満たせば，サービス不正利用防止機能を持つ．

(証明) 省略する．

以上から，3 章で示した分散属性認証方式の構成例は不正利用困難性を持つ．したがって，定理 4. 1，定理 4. 2，定理 4. 3 より，3 章で示した分散属性認証方式の構成例は認証機能を持つ．

4.2. プライバシ保護機能

3 章で示した構成例における分散属性認証方式がプライバシ保護機能を持つことを示す．

定理 4. 5. [関連付け困難性]：3 章で示した構成例における分散属性認証方式は関連付け困難性を持つ．

(証明) 3 章で示した分散属性認証方式において，*Fc* が属性 User の関連付け，AAC と User の関連付け，証明書とサービスの関連付けのそれぞれを行うことが難しいことを示す．

(c-1)属性と User の関連付け

Fc が Shared Attribute Authority SAA_i から入手した l 人分の管理情報を $\{(X_j, a(X_j, i), G_j=g^{m_j} \bmod v) | j=1, 2, \cdots, l\}$ とし(l は正整数)，使用済み AAC を AAC'$=\{Q', P(a_{r'}(X', i)), R_1'=H[r_1'], R_2'=G_{X'}g^{r_2'} \bmod v\}$とする．(2, n)秘密分散方式の性質と離散対数問題により，*Fc* は Shared Attribute Authority SAA_i の管理情報だけから$(X_j, a(X_j)_t)$の組を知ることができない ($t \in \{1,\cdots,L\}$)．*Fc* が Shared Attribute Authority SAA_i の管理情報と AAC'を併用して$(X', a(X')_t)$を知ったならば，離散対数問題とハッシュ関数の衝突困難性により r_1', r_2'を知ることができないので，次のアルゴリズムを実行することによって X'を知る．

Step1　まだ試していない値を r_2に入力する．(ただし，$r_2 \in Z_u$)

Step2　$R_2=G_jg^{r^2} \bmod v$ が R_2'と異なるとき Step1 に戻る．

Step3　$j=1$

Step4　$r=\{a_{r'}(X', i)-a(X_j, i)\}/i \bmod u$

Step5　まだ試していない値を r_1に入力する．長さが z ビット以下の値を全て調べ終わったら $j=j+1$ として Step4 に戻る．(ただし，$r_1+r_2 \equiv r(\bmod u)$)

Step6　$R_1=H[r_1]$が R_1'と異なるとき Step5 に戻る．

Step7　$r'=r$, $X'=X_j$を出力する．

u, z が十分大きいことにより，*Fc* は現実的な時間内に X'を見つけることができない．したがって，*Fc* が属性と User の関連付けを行うことはできない．

(c-2)AAC と User の関連付け

Fc が生成していない 2 つの匿名属性証明書 AAC と AAC'がそれぞれ$(Q, P(a_r(X)), R_1, R_2)$, $(Q', P(a_{r'}(X')), R_1', R_2')$に対応するとき，$X=X'$であるか否かを判定できることに相当する$(Q \neq Q', R_1 \neq R_1', R_2 \neq R_2', r \neq r')$．異なる匿名属性証明書 AAC, AAC'が同じ User X の属性名 t に対応する属性を示すならば，AAC を構成する SAC_iは r を用いて $a(X)\_t$ を変換した値 $a_r(X, i)\_t$ を，AAC'を構成する SAC_i'は r' を用いて $a(X)\_t$ を変換した値 $a_{r'}(X, i)\_t$ を含む($t \in \{1, \ldots, L\}$)．*Fc* が AAC と User の関連付けに成功するならば，次の式が成り立つことを確認することによって，$a_r(X, i)\_t$ と $a_{r'}(X, i)\_t$ が共通する値 $a(X, i)\_t$ から得られることが分かる．

$$a_r(X,i)\_t - a_{r'}(X,i)\_t \equiv ri - r'i \,(\bmod u)$$
$$\equiv (r-r')i \,(\bmod u)$$

しかし r, r'は Service Point Y と Shared Attribute Authority SAA_i が生成した乱数から生成する値であり，Shared Attribute Authority SAA_i は乱数を漏えいしないので，*Fc* は式が成り立つか否かを判定できない．したがって，*Fc* は AAC と User の関連付けができない．

(c-3)証明書とサービスの関連付け

Fc が生成していない 2 つの使用済み匿名属性証明書 AAC と AAC'について，AAC が Service Point Y に対応し，AAC'が Service Point Y'に対応するとき，*Fc* が $Y=Y'$であるか否かを判定できることに相当する．ただし，$Q=H[q] \oplus Y$, $Q'=H[q'] \oplus Y'$であり，q, q'はそれぞれ User X が Service Point Y, Service Point Y' から受け取った仮 ID である．異なる匿名属性証明書 AAC, AAC'にそれぞれ含まれる SAC_i, SAC_i'が同じ Service Point に対応するならば，それぞれが含む Q, Q'は次の条件を満たす．

$$Y = Y' \ (Q=H[q] \oplus Y, \quad Q'=H[q'] \oplus Y')$$

しかし，*Fc* は q, q'を知らないため $Y \neq Y'$を判定できない．

以上から，3 章で示した構成例における分散属性認証方式は関連付け困難性を持つ．

定理 4. 6 [プライバシ保護機能]：3 章で示した構成例における分散属性認証方式が関連付け困難性を満たせば，プライバシ保護機能を持つ．

(証明) 定義 2. 2. より，関連付け困難性を持つ分散属性認証方式は，プライバシ保護機能を持つ．したがって，定理 4. 5 より，3 章で示した構成例における分散属性認証方式は，プライバシ保護機能を持つ．

5. 比較

これまでの研究として，[3]で提案した分散属性認証方式とその構成例を旧方式と呼び，本稿で示した分散属性認証方式とその構成例を新方式と呼ぶことにする．本章では，論点である User の利便性を重視したセキュリティと計算量について，新旧両方式を比較する．

5.1. 旧方式の概要

文献[3]で提案した分散属性認証方式は，(k,n)秘密分散方式を用いており，User Assistant が分散属性情報を管理しないため，User Assistant が k 個の Shared Attribute Authority に生成してもらった分散属性証明書をもとに AAC を生成する．SAC_i と AAC の構成を次に示す．旧方式の r は k-1 個の乱数列であり，$a_r(X,i)$ は $a(X,i)$を r を用いて変換した値である．また，"||" は連結を表す．旧方式の概要を図 4 に示す．

$SAC_i = \{Q, a_r(X,i), R\}$

$AAC = \{Q, (SAC_{\alpha_j} \mid \alpha_j \in \{1,2,\ldots,k\}, j = 1,2,\ldots,k)\}$

(ただし，$j \neq j'$のとき$\alpha_j \neq \alpha_{j'}$)

$Q = H[q \| Y]$

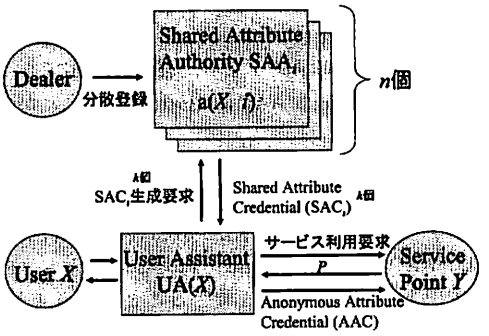

図 4　旧方式の概要

5.2. セキュリティについて

旧方式では，使用済み AAC'と Shared Attribute Authority SAA_i の管理情報の双方を用いた攻撃を考慮していなかった．そのため，Shared Attribute Authority SAA_iの管理情報から一つずつrを仮定しRを確かめていけば，AAC'を生成した User を特定することができた．

新方式では，rを仮定しても User を特定されないように，rをr_1とr_2と分割し，それぞれR_1，R_2を用いて検証することにした．したがって，4 章で示したとおり，旧方式と比較してより多くの情報を持った攻撃者でも属性と User の関連付けができない．

また，旧方式ではハッシュ関数の性質として，Yがyビットのとき，2 つのハッシュ値からそれぞれの入力値の下位yビットが等しいか否かを判定できないという性質が必要であったが，新方式ではこの性質を必要としない(yは正整数)．さらに，新方式では User の利便性を考慮した結果，UA(X)が SAC_Xを生成するが，検証可能秘密分散方式の概念を用いることによって，旧方式と同様に UA(X)の匿名を保ちつつ SAC_Xの内容を UA(X)が偽造できないようにした．

以上から，新方式は旧方式に比べ，より User のセキュリティ・プライバシを手厚く保護しているといえる．

5.3. 計算量について

旧方式では，(k, n)秘密分散方式を利用していたため，User Assistant は SAC_i生成要求フェーズと AAC 生成フェーズにおいて，暗号化，復号，署名検証回数がk回ずつ必要であり，また，乱数列 r を生成する必要であった．新方式では，SAC_iが 1 つであるので，User Assistant の同様の計算が 1 回ずつで済み，乱数を生成する必要もない．一方で，SAC_Xを生成するための計算が増えるが，署名生成する必要がないので，計算量が少なくて済む．したがって，旧方式と比較して User Assistant が負担する計算量を軽減することができる．このことは，携帯端末など計算能力に限界のある端末を User Assistant として使用することができることを示している．

6. まとめ

ユーザのプライバシ保護を考慮して提案した分散属性認証方式について，ユーザ側のセキュリティ・プライバシと計算量について再考し，ユーザにとってより利便性の高い分散属性認証方式を提案した．また，離散対数問題のもと，一般的なセキュリティ技術を用いて構成できることを示した．

文　献

[1] 松本勉，四方順司，滑藤武暢，古江岳大，上山真貴子，"分散属性認証方式に対する基本検討，" 電子情報通信学会技術研究報告(ISEC)，Vol. 105, No. 194, pp. 73-80, 2005.

[2] 松本勉，四方順司，滑藤武暢，古江岳大，上山真貴子，"分散属性認証方式に対する基本検討(2)，"Proc of SCIS2006, 3E3-3, Jan. 2006.

[3] 松本勉，四方順司，滑藤武暢，上山真貴子，"分散属性認証方式に対する基本検討(3)，" Proc of CSS2006, pp. 477-482, Oct. 2006.

[4] Adi Shamir, "How to Share a Secret," Commun. ACM 22(11), pp. 612-613, 1979.

[5] Amir Herzberg, Stanislaw Jarecki, Hugo Krawczyk, Moti Yung, "Proactive Secret Sharing Or: How to Cope With Perpetual Leakage," Proc. of CRYPTO '95, pp. 339-352, 1995.

[6] Paul Feldman, " A Practical Scheme for Non-Interactive Verifiable Secret Sharing," Proc. of FOCS'87, pp.427-437, 1987.

[7] Shafi Goldwasser, Silvio Micali, Ronald L. Rivest, "A Digital Signature Scheme Secure Against Adaptive Chosen-Message Attacks, "SIAM J. Comput 17(2), 281-308, 1988.

[8] Mihir Bellare, Anand Desai, David Pointcheval, Phillip Rogaway, "Relations Among Notions of Security for Public-Key Encryption Schemes, "Proc. of CRYPTO1998, LNCS1462、pp.26-45, 1998.