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

2005－ＣＳＥＣ－30 (45)
2005／7／22

分散属性認証方式に対する基本検討

松本 勉† 四方 順司† 清藤 武暢† 古江 岳大† 上山 真貴子†

†横浜国立大学大学院環境情報研究院 〒240-8501 横浜市保土ヶ谷区常盤台 79-7

E-mail: †{tsutomu, shikata, seito, furue, ueyama}@mlab.jks.ynu.ac.jp

あらまし 本稿では，個人認証を行う主体である認証者の手元に個人認証を受けるもの(ユーザ)の個人情報をすべて集めずに,プライバシ保護を志向した分散認証技術について，課題の抽出，及び基本方式の検討を行う．すなわち，ユーザの属性情報を秘密分散方式により分散して管理する複数の分散属性認証機関と，ユーザの属性情報を認証者にどの程度示すかをユーザ自らが制御できるモジュールであるユーザアシスタントを導入したモデルを考案し，さらに，この分散認証技術の基本方式を提案する．

キーワード 属性認証，秘密分散，匿名認証，プライバシの保護

On Shared Attribute Certification

Tsutomu MATSUMOTO† Junji SHIKATA† Takenobu SEITO†
Takahiro FURUE† Makiko UEYAMA†

† Graduate School of Environment and Information Sciences, Yokohama National University,
79-7 Tokiwadai, Hodogaya-ku, Yokohama-shi, Kanagawa, 240-8501 Japan

E-mail: †{tsutomu, shikata, seito, furue, ueyama}@mlab.jks.ynu.ac.jp

Abstract In this paper, we extract and study the basic scheme about the shared attribute certification in consideration of user's privacy protection that the entity, which certificates user's identity, does not have all of the user's information. That is, we show a basic model of shared attribute certification, which introduce shared attribute authorities and the modules called user assistant. In this model, the shared authorities manage the user's attribute information by using secret sharing schemes, and the user assistants, which can select the range about the user's attribute information for generating the attribute credentials. In addition, we propose a basic protocol of shared attribute certification.

Keyword attribute certification, secret sharing, anonymous certification, user's privacy protection

1. はじめに

インターネットの普及に伴い，電子媒体を介した非対面方式のサービスが増加することが予想される．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サービス利用資格を非対面のまま確認する必要が生じる．そこで，ユーザのサービス利用資格を判断するために，非対面のままユーザの属性を示すことを可能にする技術として，属性認証技術の需要が高まっている．

属性認証の基本的な方法として，個人認証によって特定したあるユーザに対して，まとめて管理されている全ての属性情報について属性証明書を発行する方法がある．しかし，サービスの内容によって必ずしも個人認証によってユーザを特定する必要がない場合や，サービス利用資格の有無を判定するのに必要な情報が異なる場合が考えられる．例えば，利用者に年齢制限を課さなければならない非対面方式のサービスがあったとき，サービス提供者はサービスを利用しようとしているユーザの年齢を確認する必要があるが，実際に誰なのかまでは分からなくてもよい．このようなときに，あるユーザが自身の全ての属性を示す属性証明書を利用した場合，必要の無い属性情報までを開示してしまい，プライバシの保護という観点から望ましくない．そこで，ユーザを特定しない属性認証方式や([1-3])，必要な属性情報だけを示すことができる属性認証方式が研究されてきた([1, 2])．

文献[1]では，プライバシの保護という観点から，必要な属性情報を選択して属性証明書を生成すること，及び，どのようなサービスを利用したのか，ということも含めた情報を個人と結び付けられないように，属性証明書に匿名性を持たせることを目的として構成された属性認証方式が提案されている．一度生成した属性証明書を複数回利用することを想定しているため，属性証明書を提示した User が,属性証明書に示されて

いる属性情報を持っている User と同一であることを示す機関がモデルに組み込まれている．

本稿では，文献[1,2]と同じように必要な属性情報を選択して個人ＩＤを含めた必要以上の属性情報が属性証明書から流出しないようにする機能を持ち，同時に，属性情報を管理する機関からの情報漏洩に対しても耐性を持たせるために，秘密分散を利用して属性情報を分散して保管する分散属性認証方式を考案した．その一例として，既存技術を用いてできるだけ少ないエンティティで構成できるシンプルな基本モデルを提案する．

本稿の構成は,次の通りである．2 章で属性認証について課題を検討して本稿の方針を述べる．3 章で本稿の目標とする条件と考案分散属性認証モデルを説明し，4 章でプロトコルを示す．5 章でプロトコルが目標を達成したかどうか評価し,最後に 6 章で考察を述べる．

2. 属性認証に関する基本検討

本章では，属性認証の適用例として，サービスを利用しようとするユーザが，サービス提供者に対して属性証明書を提示することによってサービス利用資格を示すことを想定し，属性認証の課題を検討する．

2.1 節では想定環境としてエンティティや通信などの環境を設定する．次に，2.2 節では基本的な属性認証モデルを例に，属性認証に求められる要求や，解決すべき課題を整理する．最後に，2.3 節において本稿で注目した分散属性認証について基本方針を述べる．

2.1. 想定環境

2.1.1. エンティティ

検討対象の属性認証を構成するエンティティについて述べる．

User：自身の属性情報を証明する属性証明書を Service Point に提示して Service Point が提供するサービスを利用しようとするエンティティ．X というIDを持つ User を User X と呼ぶ．ただし,属性情報を自ら暴露するなど上記以外の行為をしないものとする．

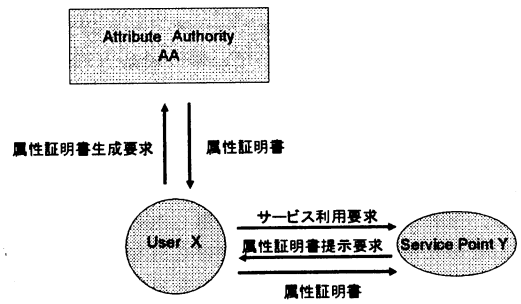

図 1. 属性認証の課題検討対象モデル

Service Point：サービスを提供するエンティティ．Y というIDを持つ Service Point を Service Point Y と呼ぶ．User の属性証明書を検証して，サービスの利用を許可できるかどうか判断する．

Attribute Authority：User の属性情報を管理し，User の属性証明書を発行するエンティティ．AA という ID を持つ Attribute Authority を Attribute Authority AA と呼ぶ．ただし,管理している属性情報を暴露したり，嘘の情報を用いて属性証明書を生成したりするなどの上記以外の行為はしないものとする．また,個人認証によって本人であると確認できた User に対してだけ属性証明書を発行するものとする．

User X はサービス利用時に Attribute Authority AA が発行した属性証明書を Service Point Y に示す．属性証明書は発行機関である Attribute Authority AA の署名を含み，Service Point Y はその署名が有効であるときに属性証明書で示された内容を信頼する．各エンティティの関係を図 1 に示す．

2.1.2. 前提条件

属性認証を検討するに当たって，次のことを前提とする．

1)**Secret Channel**：送受信者以外のエンティティが通信の内容を知ることができない．例えば，User X と Attribute Authority AA 間の通信が Secret Channel である場合，User X と Attribute Authority AA 以外のエンティティは通信の内容を知ることができない．

2)**Secret Authenticated Channel**：送受信者以外のエンティティが通信の内容を知ることができず，また別のエンティティの振りをして通信を行うことができない．例えば，User X と Attribute Authority AA 間の通信が Secret Authenticated Channel である場合，User X と Attribute Authority AA 以外のエンティティは通信の内容を知ることができず，また User X や Attribute Authority AA の振りをして通信を行うこともできない．

3)**ディジタル署名方式**：Secret Authenticated Channel での通信や属性証明書生成において，署名生成機能が盗まれたり，解読されたりするなどして危殆化しない限りは安全なディジタル署名方式を使用する．

4)**暗号化方式**：Secret Channel や Secret Authenticated Channel での通信において,暗号化機能が盗まれたり，解読されたりするなどして危殆化しない限りは安全な暗号化方式を使用する．

5)**属性情報の信頼性**：Attribute Authority は，信用できると判断した属性情報だけを登録する．したがって，Service Point は信頼する Attribute Authority から発行された属性証明書を信用することができる．

2.2. 属性認証の課題

本節では，前節で想定した属性認証モデルについて求められる要求を整理し，考慮しなければならない課題について検討する．ただし,User と Attribute Authority 間の通信を Secret Authenticated Channel に限定して考察する．

2.2.1. プライバシの保護

Service Point が User のサービス利用資格を検査するとき，必ずしもその User に関する全ての属性情報が必要というわけではない．例えば，年齢さえ分かれば利用できるサービスや，サービス利用資格がある会員である事実さえ分かれば利用を許可できるサービスがあると考えられる． User が誰であるのか，またどのような属性を持っているのかという情報は，その User のプライバシ情報であり，可能な限り User のプライバシを保護できることが望ましい．

そこで，User が Service Point に示す属性の種類を選択し，その選択結果に基づいた属性証明書を生成できる仕組みが求められる．ただし，User が誰であるのかを示す User ID も属性の一部と考えると，User ID を隠したまま属性を証明することも考えられる．以降,User ID を隠すことができる性質を匿名性と呼ぶことにする．Service Point が匿名性を持った属性証明書に示された属性を持つ User を識別するとき,User のディジタル署名で確認するなどの User ID を用いた識別方法を利用できないという課題がある．

この課題を解決するために,文献[1-3]では属性証明書に示された属性情報と属性証明書を提示したユーザの関連を保証するための証明書を発行する機関を新たに設けている．

2.2.2. 属性情報の安全な管理

User のプライバシ情報を可能な限り保護することを考えると,Attribute Authority からの属性情報の漏洩を防ぐ必要がある．ところが，事故や他者からの攻撃によって属性情報が流出した場合に User のプライバシを保護することができなくなってしまうという課題がある．したがって，事故や攻撃に対しても，Attribute Authority が情報漏洩に強い仕組みが求められる．この課題については,文献[1-3]では考慮されていない．

2.2.3. 有効期限

属性情報には,その種類によって異なる有効期限が存在する場合が考えられる．たとえば属性情報が所属する学校名であれば有効期限は在学期間であるし,会員資格であれば有効期限は会費を支払って会員資格が認められている期間である．そこで,属性証明書の有効期限として Attribute Authority が発行する証明書としての有効期限だけではなく,属性情報に基づいた有効期限を考慮した仕組みが求められる．

2.3. 基本アイディア

本稿では,事故や攻撃によって Attribute Authority から属性情報が漏洩した場合でもできるだけプライバシを保護するという課題に注目し,Attribute Authority を複数設けて,User ごとに秘密分散を用いて分散した属性情報をそれぞれに預けることを考えた．以降では複数ある Attribute Authority の一つを,Shared Attribute Authority と呼ぶことにする．各 Shared Attribute Authority は,少なくとも User ID と,その分散された属性情報である分散属性情報の組を管理し,それぞれ要求された値に対して分散属性証明書を生成する．

事故や攻撃によって Shared Attribute Authority から情報が漏洩したとしても,その属性情報は分散された値であるため,それだけでは内容を知られることができない．したがって,分散しないときに比べ,属性情報そのものが漏洩することを防ぐことができる．

Shared Attribute Authority が User ID を記録するのは，User を個人認証で確認することを可能にするためである．User と Shared Attribute Authority 間で個人認証ができれば，Shared Attribute Authority が User ID を確認できた User の要求に対してだけ分散属性証明書を生成することができる．すると,Service Point が Shared Attribute Authority を信頼していれば，User が各 Shared Attribute Authority から分散属性証明書を集めて Service Point に渡すと,Service Point が各分散属性証明書を検証することによって,分散しないときと同様に User のサービス利用資格を検査できる．

3. 基本モデル

本章では，提案する分散属性認証方式の基本的なモデルを説明する．前章で述べた User，Service Point に加えて新たに考える User Assistant，Shared Attribute Authority(SAA)，Service Point，Dividing Authority の 3 種類のエンティティについて説明する．次に,前章の検討をもとに目標とする事柄をまとめ,基本モデルを説明する．

3.1. 追加するエンティティ

新たに加えるエンティティは,以下の 3 種類である．

User Assistant : User の代わりに Shared Attribute Authority に User の分散された属性情報を証明する分散属性証明書(Shared Attribute Credential : SAC)の発行を要求し，SAC を用いて User の属性情報を証明する匿名属性証明書(Anonymous Attribute Credential : AAC) を生成して Service Point に示すエンティティ．User X の User Assistant を UA(X)と呼ぶ．User Assistant UA(X)は例えば端末のようなもので，User X の代わりに計算などの作業を行う．ただし，一度

も検証されていない AAC を横流しするなど上記以外の行為は行わないものとする．また,User X と UA(X)は認証等を経て 1 対 1 に対応しているものする．

Dealer：User の属性情報が信頼できるかどうかを確認し，その分散値を属性ごとの有効期限情報とともに Shared Attribute Authority に配布するエンティティ．ただし,User の属性情報を分配した後消去するものとし,属性情報を暴露するなど上記以外の行為をしないものとする．

Shared Attribute Authority：User の分散された属性情報を管理し，User Assistant の請求に従って分散属性証明書を発行するエンティティ．Attribute Authority の一種である．SAAi という ID を持つ Shared Attribute Authority を，Shared Attribute Authority SAAi と呼ぶ．Shared Attribute Authority が n 個ある場合は,i は 1~n を動く．ただし，管理している属性情報を暴露したり，嘘の情報を用いて属性証明書を生成したりするなど本来の目的に沿わない行為はしないものとする．また,個人認証によって本人であると確認できた User に対してだけ分散属性証明書を発行するものとする．

3.2. 目標

前章で行った属性認証に関する検討を踏まえて，次の条件を満たす分散属性認証方式を考える．以降では，匿名属性証明書 SAC を属性証明書と呼ぶことにする．

1)再利用困難性：生成された属性証明書は,一度しか利用できない．

2)関連付け困難性：User が属性同一の User が持つ属性情報に対して生成された複数の匿名性を持った属性証明書を入手しても,属性証明書から得られる情報を利用して,同一の User が持つ属性情報に対して生成されたという事実が分からない．

例えば,ある User X が持つ属性情報で,(年齢,性別)についての属性証明書と,(性別,住所)についての属性証明書があるとき,二つを同じ User が持つ属性であると関連付けて,(年齢,性別,住所)という属性の組を持った User がいることは分からない．

3)属性情報漏洩困難性：事故や攻撃が原因で Shared Attribute Authority が管理する情報が漏洩しても,Attribute Authority が一つだけの方式で Attribute Authority から情報ガ漏洩する場合に比べて属性情報が漏洩する危険が少ない．

3.3. 概要

User X が Service Point Y のサービスを受けるときについて,準備段階,User X がサービス利用要求をしてから Service Point Y が属性証明書を要求するまで,User Assistant UA(X)が属性証明書の要求を受けてから属性証明書を生成するまで,そして User Assistant UA(X)が属性証明書を生成してから Service Point Y が属性証明書を検証するまでの段階別に説明する．

3.3.1. 準備

分散属性認証で使用する,次の方式を選択する．

- 署名方式
- 暗号化方式
- 秘密分散方式

Dealer が User X の属性情報を確認した後,秘密分散方式を用いて分散した値を属性種ごとの有効期限とともに Secret Authenticated Channel で Shared Attribute Authority SAAi(i=1~n)に分配する．

3.3.2. 認証手順

本節ではモデルの認証手順を示す．各エンティティの関係を図 2 に示す．

1)サービス利用要求：User X が User Assistant UA(X)を通して Service Point Y に Secret Channel でサービス利用を要求する．

2)仮 ID 発行：Service Point Y は,User Assistant UA(X)から受け取ったサービス利用要求に対して仮 ID を発行し,その有効期限とともに記録する．有効期限が過ぎた仮 ID は消去する．

3)属性証明提示要求：Service Point Y が,User Assistant UA(X)に対して Service Point Y が必要とする属性の種類に基づく含む属性証明書の提示を Secret Channel で要求する．

4)分散属性証明書生成要求：User Assistant UA(X)が,Service Point Y からの属性証明提示要求に従い,Shared Attribute Authority SAAi(i=1~n)に分散属性証明書の生成を Secret Authenticated Channel で要求する．

5)分散属性証明書生成：Shared Attribute Authority SAAi(i=1~n)が UA(X)の分散属性証明書生成要求に従い,分散属性証明書 SACi を生成して,User Assistant UA(X)に Secret Authenticated Channel で送る．ただし，有効期限が切れている属性についての分散属性証明書は生成しない．

6)分散属性証明書検証：User Assistant UA(X) が，Shared Attribute Authority SAAi から受け取った分散属性証明書 SACi について次のことを確かめる．

- 各 SACi が Shared Attribute Authority SAAi によって生成され，また生成されてから改ざんされていないこと．
- SACi が User Assistant UA(X) が生成した分散属性証明書生成要求に対応すること．

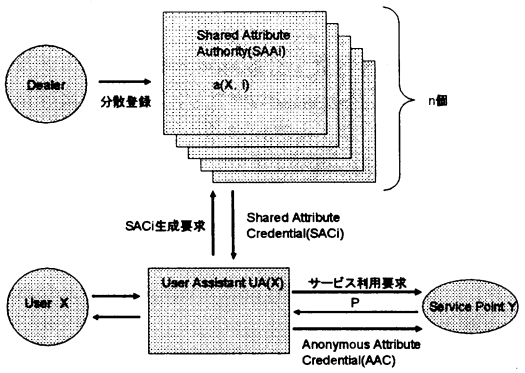

図 2. 提案モデル

また,必要な SACi を集めて次のことを確かめる.

・復元して得られる属性情報が妥当な意味を持つこと.

7)匿名属性証明書生成：分散属性証明書検証の結果が合格であれば，User Assistant UA(X) が SACi から匿名属性証明書 AAC を生成し,Service Point Y に Secret Channel で送る.

8)匿名属性証明書検証：Service Point Y が，User Assistant UA(X)から受け取った AAC について次のことを確かめる.

・AAC に含まれる SACi が Shared Attribute Authority SAAi によって生成され,また生成されてから改ざんされていないこと.

・AAC が有効期限内の仮 ID に対応していること. AA が対応する仮 ID が Service Point Y に記録されていない場合は不合格とする.

・復元して得られる情報が Service Point Y がサービス利用を許可するに値する意味を持つこと.

上記の 3 つ全てを確認できれば合格とする. 合格した場合は User Assistant UA(X)に Secret Channel でサービス利用許可を伝え,記録から仮 ID を消去する. つまり,同じ仮 ID に対応した属性証明書を検証した場合,早く検証した方のみが合格する. この規則を早い者勝ち規則と呼ぶ.

4. プロトコル

前章で説明したモデルのプロトコルを述べる.

4.1. 準備

準備として,次の作業を行う.

使用方式の選択：署名方式,暗号化方式,秘密分散方式を選択し,公開するべき情報は公開する.

属性領域の設定:1～L の L 個の属性種があるとき,1, …L は属性名を示し，属性名 1 に対応する属性値集合を Set-1，…，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は，次の直積で表すことができる. (L は正整数)

$AD(1, \cdots, L) = \{(1, \xi) \mid \xi \in \text{Set-1}\} \times \cdots \times \{(L, \xi) \mid \xi \in \text{Set-L}\}$

選択関数の設定：L 個の属性名に対応する属性領域 AD(1，…，L)から, k 個の属性名に対応する属性領域 AD(P1，…，Pk)を選び出す関数を，射影 P =(P1，…，Pk)で表す.

$P((1, \xi\_1), \cdots, (L, \xi\_L)) = ((P1, \xi\_P1), \cdots, (Pk, \xi\_Pk))$

ただし，k は 1 よりも小さい正整数で，P1，…，Pk は次の関係を満たす整数である.

$0 < P1 < \cdots < Pk$

User X の属性値の設定：Dealer が User X の属性値が信頼できることを確認する. User X の属性値を a(X)とすると，次のように表すことができる.

$a(X) = ((1, a(X)\_1), \cdots, (L, a(X)\_L)) \in AD(1, \cdots, L)$

SAAi に登録する User X の属性値の設定：Dealer が秘密分散方式を用いて User X の属性値を分散する. ある a(X)の分散された属性値を a(X, i)とすると，次のように表すことができる.

$a(X, i) = ((1, a(X, i)\_1), \cdots, (L, a(X, i)\_L)) \in AD(1, \cdots, L) \quad (i=1,\cdots,n)$

復元関数の設定：Dealer が User X の属性名 b に対する分散値を，属性名 b に対する属性値 a(X)_b に復元する関数を F_b とすると，次のように表すことができる.

$a(X)\_b = F\_b(a(X, 1)\_b, \cdots, a(X, n)\_b)$

属性値の有効期限の設定：Dealer が User X の各属性種に応じて有効期限を定め,a(X, i)とともに Shared Attribute Aouthority に登録する. User X の属性名 b に対応する属性値 a(X)_b の有効期限を t(X)_b とすると，Shared Attribute Authority SAAi が記録する User X に対する値は，(X, a(X, i), t(X)_1, …, t(X)_L)の組で表される. (i=1,…,n)

分散属性証明書に反映する属性値の選択：選択関数 P を使って, User X の属性値 a(X)から必要な属性値を選択することを次のように表すことができる.

$P(a(X)) = ((P1, a(X)\_P1), \cdots, (Pk, a(X)\_Pk))$

$= ((P1, F\text{-}P1(a(X, 1)\_P1, \cdots, a(X, n)\_P1), \cdots, (Pk, F\text{-}b(a(X, 1)\_Pk, \cdots, a(X, n)\_Pk))$

4.2. 認証手順

User X が Service Point Y のサービスの利用を要求したときから，Service Point Y がサービス利用の可否を判断するまでの流れを示す.

1)サービス利用要求：User Assistant UA(X) が，Service Point Y に Service Request を送る.

2)仮 ID 発行: Service Point Y が,仮 ID となる乱数 q と,その有効期限 T,サービス利用可否を判定するのに必要な属性情報を示す選択関数 P を生成する. ただし, P はサービス利用要求を受信する前にあらかじめ生成しておいても良い.

3)属性証明書要求: Service Point Y が, 属性証明書要求として(q, P)の組を User Assistant UA(X) に送り返す.

4)分散属性証明書生成要求: User Assistant UA(X) が(q, P)を確認し, パラメータを生成して分散属性証明書生成要求として RQi=(X, q, P, r)を生成する. 次に,User Assistant UA(X) が, 各 Shared Attribute Authority SAAi に(UA(X), SAAi, RQi)を送る.

5)分散属性証明書生成: 各 Shared Attribute Authority SAAi が UA(X)から受信した RQi を確認し, r を使って P(a(X, i))を加工した値 $P_r(a(X, i))$を生成する. 次に,Shared Attribute Authority が分散属性証明書として(q, $P_r(a(X, i))$)に対する Shared Attribute Authority SAAi の署名つき文書 SACi=SignedBySAAi(q, $P_r(a(X, i))$)を生成する. ただし, a(X, i)_b を含む分散属性証明書を生成しようとした時刻が有効期限 t(X)_b を過ぎている場合は, SACi を生成しない. 最後に, (SAAi, UA(X), SACi)を User Assistant UA(X) に送る.

6)分散属性証明書検証: User Assistant UA(X) が, 次の 3 つを確かめ,全て合格なら分散属性証明書検証の結果を合格とする.

- i=1,…,n について,VerifyForSAAi(SACi)=1 となれば合格とする. ただし,VerifyForSAAi(SACi)=1 は, SACi に含まれる署名が SAAi によって生成され,また改ざんされていないことが確認できたことを示す.
- i=1,…,n について, FirstComponent(MessageRecoveryForSAAi(SACi))=q となれば合格とする. ただし,FirstComponent(M)とはデータ M の第 1 成分のことを指し, MessageRecoveryForSAAi(SACi)とは SACi の署名を除くデータ(q, $P_r(a(X, i))$)を示す.
- 次の式が成り立てば合格とする.

F(SecondComponent(MessageRecoveryForSAA1(SAC1)), …,SecondComponent(MessageRecoveryForSAAn(SACn))) =P(a(X)) is reasonable.

ただし,SecondComponent(M)とはデータ M の第 2 成分のことを指し,上記の式は SACi の署名対象データの第 2 成分を復元させた値の集合が P(a(X))であり, その値が reasonable であることを示す. ここでいう reasonable とは,属性情報として意味を持つ内容であることを示す. もし意味を持たない情報が復元されたら, いずれかの Shared Attribute Authority SAAi が記録している分散属性値が壊れている可能性がある.

7)匿名属性証明書生成: 分散属性証明書検証の結果が合格であれば, User Assistant UA(X) が匿名属性証明書 AAC を次のように生成する.

AAC=(q, (SAA1, SAC1), …, (SAAn, SACn))

8)匿名属性証明書検証: Service Point Y が, 次の 3 つのことを確かめ,全て合格すれば匿名属性証明書検証の結果を合格とする.

- i=1,…,n について,VerifyForSAAi(SACi)=1 となれば合格とする.
- AAC の第 1 成分が Service Point Y が記録している q と一致すれば合格とする.
- i=1,…,n について, FirstComponent(MessageRecoveryForSAAi(SACi))=q であれば合格とする.
- 次の式が成り立てば合格とする.

F(SecondComponent(MessageRecoveryForSAA1(SAC1)), …,SecondComponent(MessageRecoveryForSAAn(SACn))) =P(a(X)) is reasonable.

ただし,ここでいう reasonable とは,Service Point Y がサービス利用を許可できる内容であることを示す.

5. 評価

本章では,次の攻撃を想定したとき, 本稿で提案したプロトコルが 3.2.節で述べた目標を達成しているかどうかを評価する.

攻撃 1: ある User Z が User X の振りをすることを目的とした攻撃.

攻撃 2: 属性証明書から得られる情報を元に属性証明書に示されていること以上の情報を得ることを目的とした攻撃.

攻撃 3: 何者かが Shared Attribute Authority を攻撃して記録してある秘密情報を奪い, User X の属性情報を入手することを目的とした攻撃.

5.1. 再利用困難性

Service Point が一度受理した属性証明書に関しては, 対応する仮 ID を消去してしまうので,たまたま同じ仮 ID が記録されていた場合を除き合格することは無い. したがって,上記 3 つの攻撃に対しても再利用困難性の条件を満たしていると判断できる.

5.2. 関連付け困難性

User X のために生成された属性証明書には, 個人を特定する ID である X が含まれていない. したがって, 属性証明書を見ただけではその持ち主である User X を特定することができない.

攻撃 1 によって User Z が User X の属性証明書を複数入手した場合でも,User Z はその内容を知ることが

できないので,User X の属性情報を入手することができない.

一方,攻撃2として内容が見える形で同じUser X の複数の属性証明書を入手した場合を考えると,その仮ID は全て異なる.分散属性証明書を生成するたびに,パラメータ r を作用させて分散された属性情報の値を異なるものにすれば,どの User の,どの分散属性証明書の分散された属性情報も,全て毎回異なるので,関連付ける情報を得られない.

また,Service Point Y は通信相手として User Assistant UA(X) を認識しているが,User Assistant UA(X) が固定 IP を利用していない限り,Service Point Y は複数回属性認証を行った同一の User を結びつけることはできない.

攻撃 3 対しては,属性証明書が生成できなくなっても,関連付けの根拠にはならない.

以上のことから,上記 3 つの攻撃に対して,関連付け困難性の条件を満たしていると判断できる.

5.3. 属性情報漏洩困難性

攻撃 3 に対する条件である.

属性情報は秘密分散を用いて Shared Attribute Authority に記録してあるので,User Z によってある 1 つの Shared Attribute Authority SAAi が攻撃されたとしても,そこから得られる情報からは属性情報の内容を知ることはできない.したがって,どこか 1 箇所の Attribute Authority を攻撃するだけで属性情報の内容が分かってしまうような属性認証方式に比べ,属性情報の漏洩に強いと言える.このことから,属性情報漏洩困難性の条件を満たしていると判断できる.

なお,Shared Attribute Authority に対しても属性情報の内容が分からない形になっているので,User にとって属性情報を知られずに済む.また,秘密分散の利用方法によっては事故などによって Shared Attribute Authority の記録するデータを紛失しても,属性証明書を生成し続けることが可能である.

6. 考察.

6.1. Dealerについて

3 章で述べたモデルでは,秘密分散を利用して属性情報を分配管理するので,属性情報の分散を担当する機関である Dealer を必要とする.

Shared Attribute Authority が信頼する Dealer からだけ分散された属性情報の登録を受け付けることにすれば,Service Point が Shared Attribute Authority を通して Dealer も信用したことになるので,2.1.節で述べた前提である Service Point が Attribute Authority が保証する情報を信頼することの根拠になる.

Dealer の例として,クレジット会社が考えられる.クレジット会社が顧客情報を分散して Shared Attribute Authority に配布する.配布後は自ら記録していた顧客情報を消去する.すると,Shared Attribute Authority はクレジット会社がそれまで保有していた顧客情報を属性情報として管理することができ,クレジット会社は顧客情報の管理業務を分散属性認証に委託したことになる.つまり,既存の機関に Dealer としての役割を与えることができる.

6.2. User の識別方法

属性証明では,属性証明書を提示した User が,属性証明書に示されている属性情報を持っている User と同一であることが重要である.そこで,特に匿名性を持った属性証明書について,Service Point の User 識別方法について考察する.

例えば文献[1]では,生成した属性証明書を何度も利用できるが,匿名性を持った属性証明書とその属性情報を持っている User を関連付けるために必要な登録証明書を発行する機関を設けている.一方,本稿で示したモデルでは,属性証明書を一度だけ使用する使い捨て型ではあるが,User の識別のために新たな機関を設けていない.

3 章で示したモデルでは,Shared Attribute Authority が User を知っている.したがって,User の分散属性証明書生成要求に対して個人認証を行い,確認できた User に対してだけ分散属性証明書を発行することが可能である.一方,Service Point は仮 ID を確認することによって属性証明書を提示した User がサービス利用要求者と同一であることを知ることができる.

このように,新たな機関を設けるのではなく,再利用困難性という条件と仮 ID を用いることによって Service Point が User を識別し,サービス利用を要求している User が提示された属性証明書に示された属性を持っていることを可能にしている.

6.3. 有効期限の管理方法

4 章で示したプロトコルでは,Attribute Authority が管理する属性情報に対して属性の種類ごとに適当な有効期限を記録している.属性証明書の有効期限であるパラメータ T をごく短時間に設定すれば,Shared Attribute Authority が有効期限切れの属性情報を含む属性証明書を生成しないようにすることによって,Service Point が有効と認める属性証明書は有効期間内の属性情報しか含まないようにできる.したがって,このプロトコルでは Service Point が属性情報の有効期限を確認するための行為を行う必要は無く,新たな機関を設ける必要もなくなる.

また,属性情報ごとの有効期限は Shared Attribute Authority 内にだけ記録されていて,分散属性証明書には反映されない.したがって,有効期限の値が原因と

なって関連付けが行われることは無い.

新たな機関を設けることなく属性情報の有効期限を確認する方法として,Dealer が有効期限を付加した属性情報を分散して配布することも考えられる. この場合は,分散属性証明書検証及び匿名属性証明書検証において各属性の有効期限を確認する必要がある.

6.4. 属性の選択方法

4 章で示したプロトコルでは Service Point Y が必要な属性情報を選択関数 P によって示しているが, 実際に分散属性証明書生成要求を出すのは User X の管理下にある User Assistant UA(X)である. したがって, User X は気に入らない属性情報を要求されたら拒否することも可能であり, また選択関数 P の示し方によっては User X が条件を絞り込んでから分散属性証明書生成要求を出すことができるので, User X が属性証明書で示す属性情報を選択することも可能である.

6.5. 属性証明書奪取による影響

5 章で述べた攻撃 1 について考察する.

前提より, User Assistant UA(X) と Shared Attribute Authority SAAi 間の通信から User Z が情報を入手したり, どちらかの振りをして通信したりすることはできない. また, User Z は User Assistant UA(X)にアクセスできない. したがって, User Z が User X の振りをして Shared Attribute Authority SAAi から User X の属性情報に対応する匿名属性証明書 AAC を入手することは不可能である. つまり, User Z は User Assistant UA(X) と Service Point Y 間の通信から AAC を入手しない限り, User X の属性情報に対応した AAC によってサービスを利用する攻撃 1 を成功させることはできない.

Uzer Z が AAC を入手したとき,User Assistant UA(X) と Service Point Y 間の通信が Secret Channel であるために AAC の内容を見ることができないが,AAC の内容をそのままにして送信元情報を書き換えて Service Point Y に対して User X の振りをすることは可能である. AAC を入手するだけで User Assistant UA(X)と User X 間の通信を遮断しない場合は,User Z が送信元情報を自身のものに書き換えている間にも User Assistant UA(X) から送られた AAC が Service Point Y に近づくので, Service Point Y は User Z が送信元情報を書き換えて送付した AAC を受信するよりも先に, User Assistant UA(X) が送付した AAC を受信していると考えられる. したがって, 同一の仮 ID を有する AAC を受信した場合に後から受信した方を拒否する早い者勝ち規則から, 攻撃 1 は成功しないと考えられる.

一方, User Z が User Assistant UA(X)から Service Point Y への通信を遮断しつつ AAC を入手した場合は,User Z が User X の AAC を奪取したことになり, User Z による攻撃 1 が成功すると考えられる. ただし, このような攻撃は殆どの匿名性を持った認証に有効であり, また成功させることが難しいと考えられるので, 本稿で示したプロトコルの重大な欠陥とまではならないと判断できる. また,User Assistant UA(X)と Service Point Y に AAC を 2 度に分けて送るなどの工夫をすれば,User Z が適当な通信を 2 度奪取しなければならなくなるので,プロトコルの工夫次第で攻撃 1 の成功率をさらに低くすることも可能である.

6.6. 秘密分散方式について

5.2.節で述べたように,関連付け困難性を満たすためには同じ User の同じ属性情報についての分散属性証明書であっても毎回異なる内容である必要がある. つまり,パラメータ r を作用させて分散情報の値を変化させても,復元して得られる値が一定であるような秘密分散方式を選択する必要がある.

もし,同じ User の同じ属性情報についての分散属性証明書の内容が固定されていれば,複数の属性証明書について同じ Shared Attribute Authority が生成した分散属性証明書を比較し,同じ内容の分散属性証明書があれば,同じ User が持つ属性情報についての属性証明書を関連付けることが可能になる.

一方,本稿の 4 章で述べたプロトコルでは n-out-of-n 秘密分散方式を利用しているが,そうでなければならない理由はない. また,属性の種類ごとに異なる秘密分散方式を用いることも可能である. このように,プロトコルの工夫次第で属性情報の追加やリスク分散の要素を組み込むことが可能である.

文　献

[1] 楊井 裕之, 吉田真紀, 藤原 融, "プライバシー保護を考慮した属性認証プロトコル," Proc. of SCIS2003, 2003.

[2] E.R.Verheul,"Self-Blindable Credential Certificates from the Weil Pairing," Proc. of ASIACRYPT2001, LNCS2248, pp.533-551, Springer-Verlag, 2001.

[3] Jan Camenisch, Anna Lysyanskaya, "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation." EUROCRYPT 2001: 93-118