機密性を考慮した分散証明システム

南 和宏

†統計数理研究所
190-8562 東京都立川市緑町 10-3
kminami@ism.ac.jp

あらまし 分散証明システムは分散環境の各ホストに存在する様々な情報から有益な情報を導出するための有効な方法である。これまで特にルールベースのアクセスコントロールの実現方法として活発に研究されてきた。しかし一般には各ホストが保持する情報は機密性の要件を満足する必要があり、従来研究ではこの問題が十分に考慮されてこなかった. 本論文では、分散証明システムの機密性の要件を厳密に定義し,幾つかの代表的な証明システムの安全性の解析を行う.

Confidentiality in distributed proving

Kazuhiro Minami†

†Institute of Statistical Mathematics
10-3 Midori-cho, Tachikawa, Tokyo 190-8562, JAPAN
kminami@ism.ac.jp

Abstract In this paper, we explore the design space of sound and safe confidentiality-preserving distributed proof systems. Specifically, we develop a framework to analyze the theoretical best-case proving power of these types of systems by analyzing confidentiality-preserving proof theories for Datalog-like languages within the context of a trusted third party evaluation model. Our notion of safety, which is based on the concept of non-deducibility, ensures that malicious and colluding parties do not obtain unauthorized information about other principals' confidential data during the proof construction process either directly or through inference.

1 はじめに

分散証明システムは分散環境に点在する情報を組み合わせて新たな情報を同術するのに有効である. 特に分散アクセスコントロールシステムやトラスト管理システムのベースとして分散証明システムは多用されている[1, 2, 3, 4, 5, 6, 7, 8]。しかし分散証明システムは異なる管理ドメイン間で情報を流通させるため、機密情報の保護が重要になる.

既存の分散証明システムでは、システムに属する各サーバーが自身のアクセスコントロール･ポリシーに基づき、知識ベース内の情報を公開するかどうかを決める. しかしこのような直接的なアクセスコントロールの実行手段では情報間の論理関係に基づく推論攻撃の問題が適切に考慮させているかどうかは明らかではない.

例えば3人のユーザー, p_1, p_2, p_3 がそれぞれ下記の知識ベースを持つ Datalog による分散証明システムを考えてみる.

$KB_0 = \{f_0\}$
$KB_1 = \{f_1 \leftarrow p_0 \text{ says } f_0\}$
$KB_2 = \{f_2 \leftarrow p_1 \text{ says } f_1\}$

もしユーザー p_0 が事実 f_0 を p_1 に公開することを許可するが、 p_2 に対しては認めないとする. ユーザー p1 が p0 から f0 を受け取り、事実 f_1を導出し、それを p_2 に渡せば p_2 は事実f_2を導出する. この場合, p_2は直接的に f_0 に関する情報を得てはいないが間接的な情報漏洩が存在するかも知れない.

逆にユーザー p_0 が事実 f_0 を p_2 に公開することを許可するが、 p_1 に対しては認めないとする. この場合、通常の直接的なアクセスコントロールの実効手段を用いた場合、p_2 が事実 f_2 を導出することはない. しかし Minami らが考案した暗号化したデータに対して推論を行う手法では、p_2 が事実 f_2を導出することが可能である. この場合も機密情報の間接的な漏洩が存在しないかは明らかではない.

この疑問に答えるため、我々は分散証明システムの機密性を厳密に定式化した. 我々の機密性の定義は, 非類推性 (non-deducibility) の概念 [ ] に基づき間接的な推論に基づく情報漏洩を考慮する. さらにこの定義に基づき、複数の代表的分散証明システムの安全性を分析した.

以下、第2章で分散証明システムのシステムモデルを定義し、第3章で機密性の定義を行う. 第4章では具体的な分散証明システムの安全性の分析を行い、第5章で関連研究をまとめる. 最後に第6章で結論を述べる.

2 システムモデル

我々の考慮する分散証明システムは Datalogを拡張したBANロジックに似た言語をサポートし, システムに関するサーバー間のプロトコルは推論ルールとして抽象化される. 証明の構築は Trusted Third Party (TTP)上で行われると仮定する.

2.1 証明用論理言語

証明に用いる論理言語は Database を拡張し、誰が事実を発信したか明示的に示すものである. 例えば、

grant(U, db) ← role(U, doctor), ls says location(U, hospital)

というルールはもしユーザーU が doctor の役割をもち、且つ ls というサーバーが U は hospital にいると述べた場合にデータベース db へのアクセスを許可するというルールである. 我々の証明システムでは, サーバー間で交換されるのは、上の例の ls says location(U, hospital) のような引用事実 (quoted facts) のみで、ルールは決して公開されることはないと仮定する.

2.2 証明用推論ルール (proof theory)

我々は証明システムが新しい事実を証明する動作を証明用推論ルールのセットとして記述する. 例えば、機密性を考慮しない分散証明システムは以下の2つの推論ルールで記述できる.

$$\text{(COND)}\ \frac{(f \leftarrow q_1, \ldots, q_n) \in KB_i \quad q_k \in KB_i \text{ for all } k}{f \in KB_i}$$

$$\text{(SAYS)}\ \frac{f \in KB_i}{(p_i \text{ says } fj) \in KB_j}$$

ルール (COND)はユーザーp_i が自分の知識ベース KB_iの Datalog ルールを用いて新しい事実 f を導出するときに使われる. ルール (SAYS) はユーザーp_i が持つ事実 f が別のユーザーp_j の知識ベースに引用事実として公開されるときに用いられる.

2.3 Trusted Third Party (TTP)での証明構築

我々の証明システムの計算モデルでは、図1に示すように、各サーバーが TTP に対して知識ベースの全てのデータを渡し、そのデータに対して証明システムの証明用推論ルールを適用し、これ以上新しい事実が導出できない最終状態（fixpoint）を計算し、その状態での各サーバーの知識ベースを各サーバーに返す.

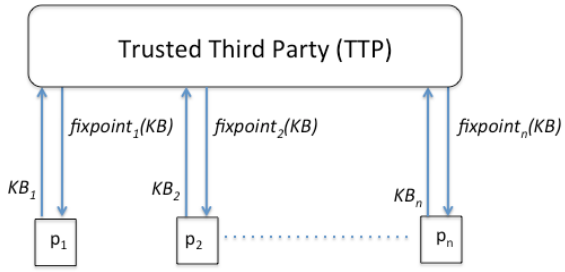

図1 TTPによる証明計算モデル

したがってこの計算モデルでは、分散証明システムは各知識ベースの状態を入力とし、最終状態を出力する関数とみなすことができる.

3 安全性

この章では、分散証明の安全性を機密データ保護の観点から定義する.

3.1 攻撃モデル

分散証明システムは有限個のユーザーの管理するサーバーから構成され、そのユーザーセット P のうちのある部分集合 A に属するユーザーが攻撃者と想定する. この攻撃者は受け身であり、分散証明システムのプロトコルを変更することはできない. これは 2.3 章で述べた TTP ベースの計算モデルから明らかである. しかし集合 A に属するユーザーは TTP から取得する計算結果を自由に共有し、集合 A 以外のユーザーの機密の情報を得ようとする.

各ユーザーp_i は自身が保持する可能性のある事実 f の真偽の機密性に関してアクセスコントロールポリシーを release(p_j, f) のように記述する. もし p_i が上記のポリシーを知識ベース KB_i に定義しているなら、ユーザーp_j はユーザーp_i が事実 f を保持しているかどうか知ることが許される.

なお各証明システムの証明用推論ルールは攻撃者を含む全てのユーザーに公知の事実とする. したがって攻撃者は任意の初期状態からの TTP の計算をシミュレートすることができる.

3.2 安全性の定義

我々の定義する安全性 は Sutherland により提案された非類推性（nondeducibility）の概念 [ ] に基づく. Sutherland は、システムの状態に関する可能世界の集合 W を考慮し、システムのビューを定義する情報関数（Information function）を導入することで推論というものを正式に定義する. すなわち可能世界の要素である w ∈ W に関する情報は w を引数といる情報関数の出力として定義される.

Sutherland は、2 つの情報関数 v1 : W → X と v2 : W → Y を以下のように定義する. ここで, W は可能世界の集合, X と Y はそれぞれ関数 v1 と v2 の値域である. ある x = v1(w) に対し、世界 w が属する集合をもとの W から下記の条件を満たす集合 S に狭めることが可能である.

$$S = \{ w' \mid w' \in W, v(w') = x \}.$$

ここでも集合 S に属する世界 w′ で、ある y ∈ Y に対し、v2(w′) = y を満足するものが存在しなければ、もともと考えていた世界 w において、v2(w) = y となる可能性はないと結論できる. つまり、このことは関数 v1 の出力から、関数 v2 の出力に関する情報（出力は y でありえない）という情報が漏洩したと見なすことができる. 図2に示す Sutherland の非類推性の概念は、このような関数 v1 から v2 に情報が漏洩することを禁止する.

定義 1(非類推性). 2 つの情報関数 v1 :W→X

とv2 :W→Y に関し、もし全ての世界 w ∈ W、全ての関数 v2 の出力しうる値 y ∈ Y に対して別の世界 w′ ∈ W が存在し、v1(w) = v1(w′) and y = v2(w′) を満足するならば関数 v1 から関数 v2 への情報の漏洩は存在しない.

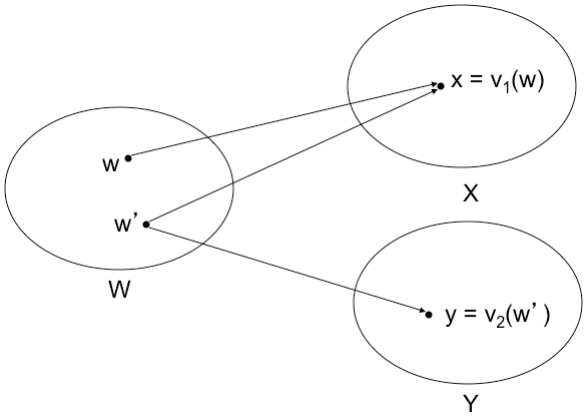

図2. 非類推性の概念

この非類推性の概念に基づき、分散証明システムの安全性を定義するため、可能世界の集合 W と情報関する v1 と v2を下記のように定義すればよい. 可能世界の集合 W は分散証明システムの全ての可能な初期状態とする. もしシステムに属するユーザーの集合が P なら、集合 W は $KB^{|P|}$ である. ここで KB は単一の知識ベースが取り得る全ての状態の集合である.

情報関数 v1は攻撃者であるユーザーの集合 A が分散証明システムをある初期状態 KB∈ $KB^{|P|}$ から実行したときに得られる知識、つまり集合 A に属する攻撃者の知識ベースの初期状態と最終状態である. ユーザーp_i の最終状態の知識ベースをKB^*_iと表す. そのとき関数v1 は以下のように定義できる.

定義2(関数 v1).関数 v1: $KB^{|P|} \times 2^P \rightarrow KB^{|A|} \times KB^{|A|}$は下記を満たす.

$$V1(KB, A) = \{(KB_i, KB^*_i) \mid p_i \in A\}.$$

関数 v2 を定義する前に攻撃者 A から機密性が守られなければいけない機密の事実の集合を定義する.

定義3 (機密の事実の集合 CF(KB)). KB を分散証明システムに属するユーザーの知識ベースの集合とすると

$$CF(KB) = \{p_i \text{ says } f \mid f \in F \wedge \text{for every } p_j \in A: \text{release}(p_j, f) \text{ not in } KB_i\}$$

関数 v2 は攻撃者 A から守られるべき機密の事実の集合として下記のように定義する.

定義4(関数 v2). 関数 v2: $KB^{|P|} \rightarrow 2^Q$ は下記を満たす.

$$v2(KB) = \{p_i \text{ says } f \mid (p_i \text{ says } f) \in CF(KB) \wedge f \in KB_j\}$$

最後に非類推性に基づく分散証明システムの安全性を定義する.

定義4(安全性). 証明用推論ルールの集合 I で定義された分散証明システムは下記の条件を満足すれば安全である. 全ての初期状態 KB ∈$KB^{|P|}$, 全ての攻撃者の集合 A ⊂P, 全ての機密の事実の部分集合 Q⊆CF(KB)に対して, 下記の条件を満足する別の初期状態 KB'が存在する.

1. v1(KB) = v1(KB')
2. Q = v2(KB'*)

最初の条件は攻撃者から見て、初期状態 KB と KB'が識別不可能であることを意味する. 2つめの条件は, もう一つの可能と考えられる初期状態 KB'で機密の事実の真偽がどのような場合も可能であることを意味する.

4 安全性分析

4.1 DAC システム

DAC システムは通常の直接的なアクセスコ

ントロールの実現手段を用いる. つまりユーザーpi は事実fをアクセス権限がある他のユーザーにのみ公開する. DAC システムは以下の2つの証明用推論ルールを持つ.

$$\text{(COND)}\ \frac{(f \leftarrow q_1, \ldots, q_n) \in KB_i \quad q_k \in KB_i \text{ for all } k}{f \in KB_i}$$

$$\text{(DAC-SAYS)}\ \frac{f \in KB_i \quad release(p_j, f) \in KB_i}{(p_i \text{ says } fj) \in KB_j}$$

ルール (COND)は 2.2 章の機密性を考慮しない証明システムの一つめのルールと同じである. ルール (DAC-SAYS) はユーザーpj が p_i から事実 f を受け取るためには、p_i が p_jにその権限を与える release(p_j, f) ∈ KB_j というポリシーを定義している必要がある.

次に DAC システムが定義4の安全性を満たすことを示す. 紙面の都合上、安全性の証明の基本的アイデアを以下に説明する. 証明のポイントは、攻撃者であるユーザーp_n は直接機密の事実, (p_i says f') ∈ CF(KB)を受け取ることがないという点にある. しかしその機密の事実から導出された別の事実 (p_i says f) が機密でないとすると、図3(a)に示すように事実 f を受け取ることは可能である. ユーザーp_nは pi がどのようなルールを持っているか全く知らないので受け取った事実 (p_i says f) がどのように導出されたか分からない. したがって図3(a)に示すようなユーザーp_i が最初から事実 f を持っていたという可能性を排除できない.

$$\text{(COND)}\ \frac{(f \leftarrow f') \in KB_i \quad f' \in KB_i}{f \in KB_i}$$

$$\text{(DAC-SAYS)}\ \frac{f \in KB_i \quad release(p_n, f) \in KB_i}{(p_i \text{ says } f) \in KB_n}$$

(a) Original proof

$$\text{(DAC-SAYS)}\ \frac{f \in KB_i \quad release(p_n, f) \in KB_i}{(p_i \text{ says } f) \in KB_n}$$

(b) Alternate proof

図3. 2つの識別不能な証明の例

4.2 Nested Encryption (NE) システム

次に Minami らの考案した分散証明の暗号プロトコル[ ]を抽象化した NE システムの安全性を示す. NE システムでは、公開鍵暗号方式で再帰的に暗号化された論理事実をその証明用推論ルールで取り扱う. 暗号化された引用事実は (q, e) のペアとして記述される. ここで q は引用事実, e は暗号化された真偽の2値の値である. ここで (p_i says f) が真であるとは、p_i の知識ベース KB_i に事実 f が存在することを意味する. 暗号化された値 e は下記の文法でその構造が規定される.

e ::= True | False | E_i(e) | e ∧ e

E_i(e) は値 e がユーザーp_iの公開鍵で暗号化されたことを意味する. 例えば, 値 True が最初ユーザーp_iの鍵で暗号化され、その後p_jの鍵で暗号化されたとするとその暗号化された値は $E_j(E_i(\text{True}))$ と表現される. NEシステムの証明用推論ルールは以下の4つである.

$$\text{(ECOND)}\ \frac{(f \leftarrow q_1, \ldots, q_n) \in KB_i \quad (q_k, e_k) \in KB_i \text{ for all } k}{\left(f, \bigwedge_{k=1}^{n} e_k\right) \in KB_i}$$

$$\text{(DEC1)}\ \frac{(q, E_i(e) \wedge e') \in KB_i}{(q, e \wedge e') \in KB_i} \qquad \text{(DEC2)}\ \frac{(q, \text{True}) \in KB_i}{q \in KB_i}$$

$$\text{(ENC-SAYS)}\ \frac{(f, e) \in KB_i \quad release(p_j, f) \in KB_i}{(p_i \text{ says } f, E_j(e)) \in KB_k}$$

ルール（ECOND)は DAC システムのルール(COND）に似ているが，通常の事実の代わりに暗号化された（q_i, e_i）の形式の事実に対して論理ルールを適用し、新しい暗号化された事実(f, $\wedge^k e_k$)を導出する. もし全ての e_k が True の値を含むなら事実 f は同じく真である. ルール(DEC1）はユーザーp_i の秘密鍵による暗号解読の操作に相当する. ルール（DEC2）は暗号を完全に取り除き True の値を得たときに通常の事実を導出する操作である. ルール(ENC-SAYS）はユーザー pi が暗号化された値（f, e）を他のユーザーpk に渡す場合，事実 f の真偽を知る権限をもつ第3のユーザーpj の公開鍵で暗号化することを表現している.

NE システムも定義4の安全性の要件を満たす. 証明は紙面の都合上割愛するが、安全である理由は、NE システムにおいて、ある真の値が再帰的に暗号化された場合、そのちょうど逆の順番に解読されないと最初の真の値は得られない点にある. よって、もし攻撃者がある証明のツリーから事実を導出した場合、その証明に含まれる機密の事実に対する暗号は、攻撃者ではない別のユーザーに解読される必要がある. そのプロセスは攻撃者からは見えないので機密の事実に関する識別不可能性が保証される.

4.3 Commutative Encryption (CE) システム

CE システムは NE システムを拡張し、交換可能な暗号をその証明用推論ルールの中で用いている. つまり真偽の値が複数のユーザーの鍵で暗号化された場合にそれとは全く逆の順序でなくても対応する秘密鍵で解読できることを意味する. CE システムでは暗号化した事実は（q, S）の対で表され、ここで S は公開鍵に相当するユーザーの集合である. 以下に CE システムの証明用推論ルールを示す.

$$\text{(CECOND)}\ \frac{(f \leftarrow q_1, \ldots, q_n) \in KB_i \quad (q_k, S_k) \in KB_k \text{ for all } k}{(f, \cup_{k=1}^{n} S_k) \in KB_i}$$

$$\text{(CEDEC)}\ \frac{(q, S) \in KB_i}{(q, (S \setminus \{p_i\}) \in KB_i}$$

$$\text{(CE-SAYS)}\ \frac{(f, S) \in KB_i \qquad release(p_j, f) \in KB_i}{(p_i \text{ says } f, (S \cup \{p_j\}) \in KB_k}$$

ルール（CECOND)は NE システムのルール(ECOND）に相当するが，違いは導出された事実に関連付けられる公開鍵の集合は前提に現れる各集合 S_k の和になる点である. ルール(CEDEC）はユーザーp_i による暗号解読の操作を表現しているが、この場合は集合 S から p_i を取り除くことに相当する. ルール（CE-SAYS)は NE システムの（ENC-SAYS）と同様に、事実を他のユーザーに渡す前に暗号化しているが、ここでの p_j の公開鍵による暗号化は集合 S に p_j を追加することになる.

CE システムの交換可能な暗号方式は NE システムのものよりも制限が少ないため，NE システムよりも多くの事実が導出できるはずである. しかし残念ながら CE システムは定義4の安全性を満足しない. 安全でない理由は以下の点にある. 攻撃者の一人がある事実を導出する際に攻撃者でないユーザーの機密の事実が証明で使われていたとする. NE システムの場合、暗号化された機密データは必ず別の非攻撃者のユーザーによって解読されてから渡されるのに対し、CE システムでは、攻撃者が最後に暗号を解読することがありえる。これと識別不可能な機密の事実を含まない別の状況は非攻撃者には作成することはできない.

5 関連研究

分散証明システムは主に分散アクセスコントロールの実現手段として研究されてきた. この分野の研究者の主な関心はロールやで権限委譲を表現する新しいセキュリティポリシーを定義するための言語を開発することであった. それに対し、本論文では証明システムの言語としては

一般的な Datalog をベースにしたものを想定しながらも，証明システムにおける機密情報の保護について考察を行った．少数の機密性に関する研究は存在するが，本研究のように間接的な情報漏洩を考慮して厳密に機密性の要件を定義した研究は存在しない．

その中で論理型のアクセスコントロールシステムの情報フロー解析に関する Becker の研究が本研究にもっとも近いものといえる．しかし Becker のセキュリティモデルの安全性の概念は本研究のものよりも弱く、機密の事実それぞれについて真偽両方が可能であればよいというものである．我々の定義では、複数の機密の事実が存在する場合、全ての真偽の組み合わせが可能であることが必要になる．また Becker の研究における研究者は証明システムの外部に存在する．

6 結び

本論文では、複数のユーザーが管理するサーバーから構成される分散証明システムにおいて、各知識ベースに含まれる機密情報の保護について考察した．我々は分散証明システムを証明用推論ルールのセットとして抽象化し、サーバー間の情報の交換はそれら推論ルールの中で表現する手法を採用した．我々の安全性の定義は非類推性の概念に基づくもので、証明システムの構成メンバーとして存在する攻撃者から機密の事実の真偽を保護することを保証する．

安全性に関する分析の結果、公開鍵暗号方式を用いて暗号化した事実に対する推論を行う NE システムが安全であることが明らかになった．これにより、通常の直接的なアクセスコントロールの実現方法を用いる場合よりもより多くの事実が安全に導出できることが分かった．しかし暗号公式を交換可能な場合に拡張すると安全性が損なわれることも明らかとなった．今後は機密性を保証する分散証明システムの中で最大限の事実の導出を行うシステムがどのシステムであるか、その上限を明らかにすることが今後の課題である．

参考文献

[1] Andrew W. Appel and Edward W. Felten. Proof-carrying Authentication. In Proceedings of the 6th ACM Conference on Computer and Communications Security, pages 52–62. ACM Press,
1999.

[2] Lujo Bauer, Scott Garriss, and Michael K. Reiter. Distributed Proving in Access-Control Systems. In Proceedings of the 2005 IEEE Symposium on Security and Privacy, pages 81–95, Washington, DC, USA, 2005. IEEE Computer Society.

[3] Moritz Y. Becker, Cedric Fournet, and Andrew D. Gordon. Design and semantics of a decentralized authorization language. Proceedings of the 20th IEEE Computer Security Foundations Symposium, pages 3–15, July 2007.

[4] John DeTreville. Binder, a logic-based security language. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, page 105, Washington, DC, USA, 2002. IEEE Computer Society.

[5] Yuri Gurevich and Itay Neeman. Dkal: Distributed-knowledge authorization language. Proceedings of the 21st IEEE Computer Security Foundations Symposium, pages 149–162, June 2008.

[6] Trevor Jim. SD3: A trust management system with certified evaluation. In Proceedings of the 2001 IEEE Symposium on Security and Privacy, pages 106–115. IEEE Computer Society, 2001.

[7] Ninghui Li, Joan Feigenbaum, and Benjamin N. Grosof. A logic-based knowledge representation for authorization with delegation. In Proceedings of the 1999 IEEE Computer Security Foundations Workshop, page 162, Washington, DC,

USA, 1999. IEEE Computer Society.
[8] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust-management framework. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, page 114, Washington, DC, USA, 2002. IEEE Computer Society.
[9]