特集論文

グラフデータベースを用いたセキュリティープロトコル向け形式検証

Formal Verification Using Graph－Database for Security Protocols

毛利寿志*
Hisashi Mori
米持一樹†
Kazuki Yonemochi
三澤　学‡
Manabu Misawa

*三菱電機㈱ 情報技術総合研究所(博士(工学))
†同社 伊丹製作所
‡三菱電機デジタルイノベーション㈱

要　旨

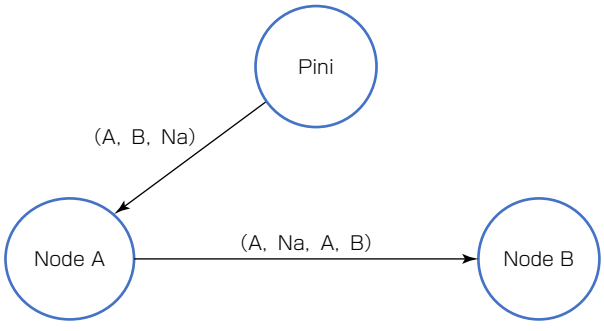
システム設計上，情報漏洩(ろうえい)や不正アクセス等の脆弱(ぜいじゃく)性が含まれないことを保証する方法として，セキュリティー検証がある。従来の検証方法では，形式言語理論やツール固有の言語といった高度な専門知識が必要，かつシステムが複雑になると検証が現実的な時間で完了しない，という課題があった。そこで，専門的で記述内容をすぐに理解しづらい言語表現ではなく，データ間の関係性を視覚的に把握できる一般的なグラフデータベース(以下“グラフDB”という。)を用いて，データ挿入と探索のデータベース操作でセキュリティー検証を可能にする方法を開発した。特定の暗号プロトコルを検証した結果，グラフ表示によって直感的な検証で従来と同様に脆弱性を検出した。さらに，検証時間が削減できることを理論的に示した。

1. ま え が き

システムの安全性やセキュリティーの確保に対する要求が高まっている。それらの確保には，設計段階でシステムが安全かつセキュアであると保証する方法が必要である。システムの一例としてセキュリティープロトコルがあるが，あるプロトコルが攻撃に対して正常に動作することを手動で解析することは困難である。この問題に対して，形式手法を用いた検証方法による解析の自動化は，有効な仕組みの一つと言える。セキュリティープロトコルに対する検証では，プロトコルやプロトコルに対する攻撃者の振る舞いを形式言語でモデル化し，プロトコルが満たすべきセキュリティー要求を検証項目として別の形式言語でモデル化し，ツールを用いて検証を実行する。このツールには，セキュリティープロトコル検証専用ツールと，汎用モデル検査器がある。このうちSPIN[(1)]等の汎用モデル検査器では，セキュリティーを含む様々な検証項目を記述できて，かつセキュリティーに関する検証事例も複数存在する。そのため，今回の研究ではセキュリティー検証の中でも汎用モデル検査器を用いた検証に焦点を当てる。様々なセキュリティープロトコルがSPINで検証されており，SPINを用いたセキュリティー検証の有用性は示されている。例えばMaggiらは，NSPK(Needham－Schroeder Public Key)プロトコルに対して検証を行い，脆弱性を導出した[(2)]。またHendaは，セキュリティープロトコル全般に対して盗聴や盗聴したメッセージの混入・再送を行う攻撃者を論理的に定義した。この定義を基に，NSPKプロトコルを含む３種類のセキュリティープロトコルに対して検証を行い，既知の脆弱性を導出した[(3)]。しかし，これらの従来技術は専用言語を習得する必要があるため，導入が比較的困難という課題があった。さらに，対象システムが複雑にな

```
mtype＝{A, B, Na};
Chan ca＝[0] of {mtype, mtype, mtype, mtype};
proctype Plni (mtype self; mtype party; mtype nonce)
{
        mtype g1;
        ca ! self, nonce, self, party;
}
proctype PRes (mtype self; mtype nonce)
{
mtype g2, g3;
        ca ? eval (self), g2, g3, eval(self);
}
init
{
        run Plni(A, B, Na)
}
```

(a)　従来技術SPINのプログラム記述例



(b)　グラフDBによる表示例

図１－従来技術SPIN及び提案手法グラフDBによる検証対象の表示例

ると，プロトコルや攻撃者の振る舞いを示す内部状態数が指数関数的に増大し検証が現実時間で完了しないという課題があった。

今回の研究では，最も広く利用されているグラフDBの一種であるNeo4j[4](注1)を採用し，グラフDBを用いたモデル検査を実装した。**図1**に，従来技術SPINによる対象システムの記述例と，同じ対象システムをグラフで表現したイメージ図を示す。**図1**は，送信者A(図中の$Node\ A$)が受信者B(図中の$Node\ B$)へ，メッセージNa及びAを受信者Bの公開鍵で暗号化し送信する通信(図中の$Node\ A$から$Node\ B$への矢印)を表す。

従来技術では，検証対象とするシステムの振る舞いを状態遷移に変換した後，その状態遷移をテキストでプログラムする(**図1**(a))。検証時にはプログラムが正しく検証対象を示していることを解読する必要がある。一方，提案手法によるグラフ表示ではシステムの振る舞いを視覚的に分かりやすく把握できる(**図1**(b))。ここで，$Pini$はプロトコルの開始を表す。また，矢印上の文字は，左から“送信者A，メッセージNa及びA，受信者Bの公開鍵で暗号化”を表す。対象システムを視覚的に分かりやすく把握できることによって，セキュリティー検証を行う際の負担を軽減できる。

(注1)　Neo4jは，Neo4j, Inc.の登録商標である。

2. 提案手法：グラフDBを用いたセキュリティー検証方法

直感的に検証できて，検証時間を削減できる，新たなセキュリティー検証方法を提案する。**2. 1節**では，セキュリティー検証方法として採用するモデル検査に必要な入力情報である対象システム，攻撃モデル，検証項目について，提案手法でどのように定義するかを述べる。**2. 2節**では，グラフDBを用いた実装方法及び実際に検証を実行した結果を述べる。

2. 1　提案手法の形式的表現の定義

提案手法では，検証したいセキュリティープロトコルや攻撃者の振る舞いを有向グラフで表して，検証したい性質をグラフDBのクエリーで表現することで，直感的な検証を実現している。この節では，グラフDB上で対象システム，攻撃モデル，検証項目をどのように定義するのかと，それらに対する検証結果が何を示すのかについて述べる。

(1)　対象システム

提案手法の対象システムは，既存モデル検査器の状態遷移と同等の有向グラフである。対象システムを有向グラフ(V, E)で表現する。ここで，Vは頂点の集合，Eは有向辺の集合である。次に，頂点と有向辺を定義する。

頂点は，r(送信者，受信者を示す項目)及びs(プロトコル中のどの状態かを示す項目)のラベルを含むと定義する。送信者(又は受信者)がプロトコルに沿ったメッセージを受信することを一つの頂点で表す。メッセージの内容が異なれば，それぞれ異なる頂点で表す。それらとは別に，プロトコル開始を表す$Init$及びプロトコル終了を表すEndの２種類の頂点を追加する。

有向辺は，t(暗号文又は平文を示す項目)，m(送信されるメッセージ内容を示す項目)，k(メッセージを暗号化する鍵を示す項目)のラベルを含むと定義する。プロトコル送信元から受信先までを一つの有向辺とする。

(2)　攻撃モデル

セキュリティープロトコルの脆弱性を網羅的に検査できる攻撃モデルとしてDolev-Yaoモデルが著名であり，**1章**で述べたHendaの研究[3]でも採用されている。提案手法でも同様に，Dolev-Yaoモデルに基づくHendaの攻撃モデルを用いる。すなわち，Hendaの攻撃モデルを追加した対象システムに対して，セキュリティー要件を満たすか否かを検証する。紙面の都合上，Hendaの攻撃モデルの厳密な定義については原著論文[3]を参照されたい。

攻撃モデルを対象システムに追加する手順を示す。まず，事前準備として，攻撃者が知る情報の集合$\mathbb{K}$を$ACTOR$(送信者／受信者／攻撃者)，$PLAIN$(平文)，$CIPHER$(暗号文)，$PKEY$(公開鍵)，$SKEY$(秘密鍵)の部分集合に分割する。ある状態を送信元とした場合の攻撃を表す頂点及び有向辺の作成手順は次のとおりである。

①$ACTOR$，$PLAIN$，$PKEY$の中から各要素を選択し，全ての組合せについて暗号文を作成

②送信する先の頂点(送信者又は受信者)が存在しない場合，送信先の頂点を作成

③作成した各暗号文又は$CIPHER$に格納済みの各暗号文をラベルとして，送信元-送信先の有向辺を作成

(3)　検証項目

提案手法では，従来技術SPINと同様に，検証項目として形式言語の一種であるLTL(Linear Temporal Logic：線形時相論理)相当の記述[5]を想定する。検証項目の一例として，ユーザー認証を取り上げる。従来技術[2][3]によると，モデ

ル検査でのユーザー認証(送信者Aが確かに受信者Bを認証している)の定義は,“送信者Aがプロトコルを実行完了したときはいつでも,他者Bはプロトコルを実行している”である。

⑷　検証結果

対象システム及び攻撃モデルを表す有向グラフ上で検証項目を表すクエリーを実行し,クエリーを満たさない経路が出力された場合,その検証項目が満たされなかったことになる。先に述べたとおり,提案手法の有向グラフとクエリーは,それぞれ従来技術(モデル検査器)の状態遷移と検査項目と同等である。したがって,クエリーの出力は従来技術の検証結果と同等である。

2.2　グラフDBを用いたモデル検査の作業フロー

この節では,提案手法であるグラフDBを用いたモデル検査の作業フローについて述べる(**図2**)。

⒜　従来技術　　⒝　提案手法

図2-従来技術及び提案手法を用いたセキュリティー検証の作業フロー

従来技術(**図2**(a))では,検証したいシステム仕様や攻撃モデルを選択した後,形式言語へ手動変換し,さらにモデル検査器のプログラムに変換してからモデル検査器で検証実行する必要があった。一方,提案手法(**図2**(b))では,検証したいシステム仕様と攻撃モデルを選択し,検証を実行するだけで検証結果が得られる。提案手法は,任意のセキュリティープロトコルについて,プロトコルや攻撃モデルをグラフDBに変換するアルゴリズムである。アルゴリズムを確立すれば,変換の自動化は容易であるため,今後プロトコルや攻撃モデルのグラフDBへの変換を自動化することを予定している。モデル検査手法にグラフDBを用いることで,設計者は一般のDB操作である頂点や有向辺の追加及び経路探索の知識でモデル検査の実施が可能になる。また,グラフDBを用いたことや,攻撃者の状態数,プロトコルで送受信される情報を表す変数,及び偽メッセージを適切に削減したことによって,提案手法を用いた検証は,従来技術に比べて,検証に必要な内部状態数を削減できる。

グラフDB上での実装の例題として,**1章**で挙げたNSPKプロトコルを対象に,**2.1節**で述べた検証項目である認証について検証した。具体的には,NSPKプロトコルや攻撃モデル,認証を表す論理式を**2.1節**で示した手順に沿ってグラフDB上で表現し,認証を満たさない経路のうち最短経路を求める方法と,全経路を求める方法を実行した。ここで,検証対象及び検証項目について検証を実行した際,その検証項目を満たさない経路が出力される。一方,検証対象が検証項目を満たす場合は,何も出力されない。実行した結果,両方の方法でそれぞれ反例が出力された。2種類の経路について検証項目を満たさないことを目視で確認したことによって,検証結果が正しいことを確認した。なお,紙面の都合上,NSPKプロトコルを表すグラフや検証結果のグラフは省略する。

3.　評　　価

従来技術であるSPINを用いた検証方法も,提案手法によるグラフDBを用いた検証方法も,対象システムの振る舞いを状態遷移として表して,検証項目を満たさない状態に到達した場合は反例として出力する。すなわち,検証時間は,作成される状態数に依存する。状態が多すぎる場合は検証が現実的な時間で完了しない危険性がある。そこでこの章では,

実行時間に関する論理的な評価として，従来技術及び提案手法で作成される状態数を比較する。状態数を比較した結果，提案手法は従来技術に比べて常に少ない状態数であることを確認した。例えば，NSPKプロトコル（メッセージ数$m=3$）の場合，従来技術の状態数は15,275である一方，提案手法は状態数20であり，約763分の１である（**図３**）。従来技術SPINでの状態数は，対象システムの状態数と検証項目の状態数の積で表される。一方で提案手法の状態数は，対象システムの状態数だけである。これは，グラフDBの探索機能によって，対象システムの状態だけを探索することで検証できるためである。従来技術及び提案手法の状態数をそれぞれ算出し，比較した結果を次に示す。

従来技術であるHendaの方式では，状態数は次の式で表される。

$$\{m+3+\sum_{i=0}^{m}\{(|Knows|-|PKEY|)^{(t-t_{|PKEY|}+(i-1))} \times\{(|Knows|-|PKEY|)^{(t-t_{PKEY})}\times|PKEY|+1\}\}\} \times(2^{|TL|}+1)$$

ここで，mはメッセージ数，$|Knows|$は攻撃者が知る情報の要素数，$|PKEY|$は攻撃者が知る情報のうち公開鍵の要素数，tは１メッセージ当たりの項数，$t_{|PKEY|}$は１メッセージ当たりの公開鍵を表す項数，$|TL|$は検証項目の各部分式の一番外側の演算子が様相演算子である式の集合をTLとしたときのTLの要素数である。

一方，提案手法では，状態数は次の式で表される。

$$(m+3)+|Init|+\sum_{i=0}^{m}((|PLAIN|^{t_{PLAIN}}+i+1))$$

ここで，$|Init|$は最初の送信者の状態数，$|PLAIN|$は平文全体の要素数，t_{PLAIN}は１メッセージ当たりの平文を表す項数である。

このような式に基づいて，Hendaの方式と提案手法についてメッセージ数mに伴う状態数の変化を比較したグラフを**図３**に示す。全体を通して，提案手法の方が従来技術よりも少ない状態数になることが分かる。一般のパソコンでは状態数500,000程度が検証の限界と言われている[6]。従来技術ではメッセージ数７のとき状態数が500,000を超えるが，提案手法では状態数が62と抑えられたままである。

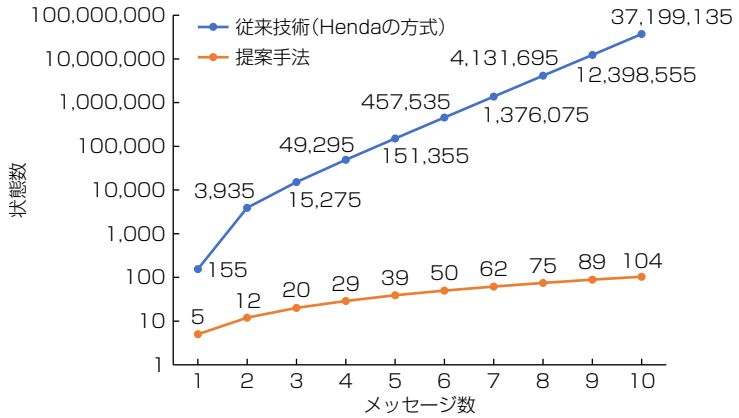


図３−メッセージ数に伴う従来技術及び提案手法の状態数の変化

4. む　す　び

セキュリティープロトコルを例題に，対象システムをグラフDBの有向グラフで記述し，検証項目をグラフDBのクエリーで記述するモデル検査手法を提案した。グラフDBという一般的な知識でグラフを作成できて，グラフを可視化する機能によって直感的な理解が可能になる。また，グラフDBの全パス探索機能を用いて，全反例を出力するモデル検査の実行も可能になる。さらに，有向グラフに対して検証項目を直接クエリーで記述し探索することで，モデル検査実行に際して状態数を削減できて，自動での検証が現実的な時間で可能になる。

今後，ほかのセキュリティープロトコルを対象にした場合に，従来技術SPINと提案手法で実行結果や状態数にどの程

度違いがあるかを検証する予定である。さらに，提案手法が認証以外の性質も定義・検証できることを示す予定である。また，システムの安全性も含めた検証を行えるよう，対象システムや検証項目の形式化を拡張する予定である。今後，システムはますます複雑になり，セキュリティー機能が正しく搭載されていることを手動で判断することが困難になる。提案手法を用いてシステムのセキュリティーを自動で保証することによって，設計段階の工数を大幅に削減できる。

参 考 文 献

(1) Holzmann, G. J.：The SPIN Model Checker，Addison－Wesley (2004)
(2) Maggi, P., et al.：Using SPIN to Verify Security Properties of Cryptographic Protocols，Lecture Notes in Computer Science，**2318**，187～204 (2002)
(3) Henda, B. N.：Generic and Efficient Attacker Models in SPIN，Proceedings of the 2014 International SPIN Symposium on Model Checking of Software，77～86 (2014)
(4) Robinson, I，ほか：グラフデータベース－Neo4jによるグラフデータモデルとグラフデータベース入門，オライリー・ジャパン (2015)
(5) 久野和敏，ほか：グラフデータベースを用いたモデル検査手法の提案，第198回SE研究発表会 (2018)
(6) Joesang, A：Security Protocol Verification using SPIN，the First SPIN Workshop(SPIN’95) (1995)