暗号プロトコルの安全性証明自動化技術

Automated Security Proofs for Cryptographic Protocols

花谷　嘉一　■ HANATANI Yoshikazu
村谷　博文　■ MURATANI Hirofumi
太田　和夫　■ OHTA Kazuo

安全性証明は，情報セキュリティ分野の基盤技術である暗号方式やデジタル署名方式などの安全性を保証する手段である。安全性証明により誰もが安全性を納得できるため，標準暗号の選定基準の一つとされている。安全性証明を与えるには高度な専門知識が要求されるため，安全性証明をソフトウェアで自動化する技術が注目を集めている。

東芝と国立大学法人 電気通信大学（以下，電気通信大学と略記）は共同研究により，暗号プロトコルの安全性証明を自動化するソフトウェアの一つである，CryptoVerifの証明能力の解析と改良を行った。改良前は安全性証明に失敗する例が存在したが，改良の結果，CryptoVerifの開発者とわれわれが与えた例の範囲では，すべての安全性証明が可能となった。

A security proof is a measure that ensures the security of fundamental cryptographic primitives such as public key encryption, digital signatures, and so on. As it convinces all parties concerned of the security of the primitives, it also provides a criterion for the standard cryptographic scheme. Advanced expertise is required to provide security proofs. There is consequently an increasing need for automated security proof technology.

Toshiba and the University of Electro-Communications have performed analysis and improvement of the proof capability of CryptoVerif security-proving software. Experiments with our improved CryptoVerif confirmed that it could successfully prove all examples given by both Bruno Blanchet and ourselves, whereas the original CryptoVerif might fail in a few examples.

1　まえがき

情報セキュリティ分野における基盤技術として，公開鍵暗号方式，デジタル署名方式，鍵共有方式などがある。従来，これら基盤技術の設計は経験的に行われ，提案と解読が繰り返されてきた。近年，これら基盤技術の安全性を保証する理論である，証明可能安全性が確立されつつある。証明可能安全であるとは，方式が定式化された安全性を満たすことを数学的に証明できることを指す。経験的な安全性とは異なり，証明可能安全性は，誰もが方式の安全性を納得することができるため，標準暗号の選定における評価項目の一つとして利用されている。

一般に，方式の安全性を証明するためには，高度な専門知識とともに，多くの労力を費やす必要がある。証明には，依存しあった確率的事象の評価が必要となるため，複雑になることも多い。そのため，証明の誤りが見落とされ，証明の発表後数年経ってから指摘された例もある。方式が複雑になるにつれて安全性の証明も複雑になるため，前述した問題がますます深刻になっている。

近年，暗号方式の安全性証明をソフトウェアで自動的に行う試みが盛んになっており，証明における人為的な誤りの防止や，暗号方式開発の効率化が期待されている。もちろん，ソフトウェアに実装ミスがあれば誤った証明が生成される場合があるため，ソフトウェアの正当性の検証も重要となる。

ここでは，証明可能安全性を証明する技術の一つであるゲーム列による安全性証明を紹介し，ゲーム列による安全性証明の一部を自動的に行うソフトウェアであるCryptoVerifについて述べる。更に，東芝と電気通信大学の共同研究による，CryptoVerifの証明能力の解析結果とその拡張の試みについて述べる。

2　証明可能安全性

暗号プロトコルの安全性を証明する方法の一つに，ある問題を解くことが難しいことが安全性の根拠であると証明する方法がある。この方法では，暗号プロトコルへの攻撃をモデル化し，モデルの下で安全性を破る攻撃者を利用することで，困難と信じられている問題（計算量的仮定）を効率的に解くアルゴリズムが構成可能であることを示す。言い換えると，暗号プロトコルへの攻撃は計算量的仮定を破るのと同じくらい難しいことを，数学的に証明している。つまり，計算量的仮定が成立していれば，攻撃モデルの範囲内では暗号プロトコルに対する有効な攻撃は存在しないことを意味している。この方法は帰着技法と呼ばれ，情報理論的な解析と並んで，現代暗号の安全性証明によく用いられている。

暗号プロトコルの自動証明のためには，評価対象である攻撃の状況を形式化する必要がある。形式化の方法としては，Dolev-Yaoモデル[1]が広く知られている。Dolev-Yaoモデルで

表1. 安全性証明方法のトレードオフ
Trade-offs of proof capability of frameworks for cryptographic protocols

証明方法	証明の容易さ	証明の仮定	証明の品質
人手による証明	×	○	○
Dolev-Yao モデル	○	×	×
B. Blanchetの枠組み	△	○	△

は，プロトコルでやり取りされるデータや性質を代数的な性質で表現し，攻撃者の能力を記号的な操作で表現することができる。しかし，形式化の際には絶対に破られない理想的な暗号系を仮定するため，Dolev-Yaoモデルで検証できる安全性と現実の安全性の間には隔たりがある。近年，この隔たりを埋めようとする研究が盛んに行われている。

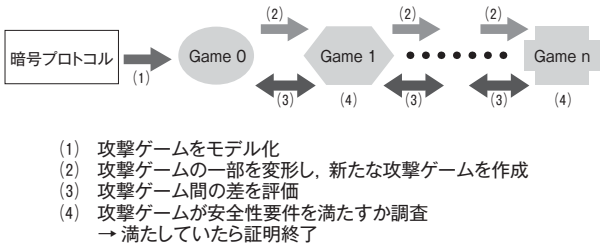
Dolev-Yaoモデルとは異なる形式化の方法として，B. Blanchetらが提案した枠組み[2]がある。この枠組みに基づき検証を行うソフトウェア CryptoVerifが公開されている。Dolev-Yaoモデルと比較すると，**表1**に示すように，より現実的な仮定に基づく安全性証明が可能であり，また，攻撃問題と計算量的仮定の違いを定量的に評価できるため注目を集めている[3]。

従来，暗号プロトコルの安全性を証明できるのは，高度な専門知識を習得した人に限られていたが，自動証明技術が実用化されると，専門知識がなくても評価対象の暗号方式を定められた方法に従って記述するだけで安全性証明が生成できるため，暗号プロトコルの設計・開発の効率化が期待される。また，ソフトウェアで安全性を検証できることが標準化の要件に採用される例も出てきており，自動証明技術は将来的に重要性を増していくと思われる。

3　ゲーム列による安全性証明

ゲーム列による安全性証明は，人手による証明でよく用いられ，暗号プロトコルへの攻撃の困難さの評価を簡約化する手法の一つである[4]。ゲーム列による安全性証明の処理手順を以下に示す（**図1**）。

(1)　攻撃者に許す戦略と攻撃の目標を明確に定義し，攻撃の状況をゲームとしてモデル化する。
(2)　攻撃ゲームの一部を変形した新たな攻撃ゲームを生成する。
(3)　変形によって生じる攻撃ゲーム間の差に注目して評価を行う。
(4)　偶然でしか攻撃が成功しないような攻撃ゲームに至るまで攻撃ゲームの変形と評価を繰り返すことで，評価対象の攻撃に対する安全性を評価する。評価対象の攻撃ゲームと，偶然でしか攻撃が成功しない攻撃ゲームの差が無視できるほど小さいとき，有効な攻撃が存在しないことが示せるため，安全性が証明できる。



(1)　攻撃ゲームをモデル化
(2)　攻撃ゲームの一部を変形し，新たな攻撃ゲームを作成
(3)　攻撃ゲーム間の差を評価
(4)　攻撃ゲームが安全性要件を満たすか調査
　→ 満たしていたら証明終了

図1. ゲーム列による安全性証明 ― ゲーム列による安全性証明では，評価対象が複雑であっても，変形規則の与え方によって評価対象を明確に絞り込むことで誤りが抑制できる。
Security proof with sequences of games

ゲーム列による検証では，評価対象が複雑であっても，変形規則の与え方により評価対象を明確に絞り込むことで誤りを抑制しようとする。また，評価対象が明確であるため可読性の高い証明となる。攻撃ゲーム間の確率分布の差を評価しやすいように攻撃ゲームを変形できるかが，ゲーム列による検証のポイントとなる。

4　CryptoVerifによる安全性証明

CryptoVerifは，ゲーム列による安全性証明に基づく検証を行うソフトウェアである。専用の文法で記述された入力に対して，安全性が成立するか否かを判定する。

CryptoVerifを用いて安全性証明を行うには，まず，評価対象の暗号方式とその攻撃モデルを記述する。更に，書換え規則を，観測等価性を満たした二つのプロセスで記述する。入力として与える評価対象の攻撃モデル及び書換え規則の記述と，書換え規則の観測等価性の評価は人間が行う必要がある。観測等価性とは，二つのプロセスのふるまいを，観測者の視点から得られる情報を基に識別できる確率は十分に低いという性質で，複雑な確率評価が必要となる場合があるが，一度評価してしまえばほかの証明にも流用できるという利点がある。

CryptoVerifは，攻撃モデルに対して書換え規則を適用することで新たな攻撃ゲームを生成し，ゲーム列を構成する。

CryptoVerifによる処理手順を**図2**に示す。CryptoVerifは，攻撃モデル及び書換え規則が記述されたファイルが入力されると，以下の(1)～(4)の処理を繰り返す。

(1)　攻撃モデルと書換え規則を比較する。
(2)　適用可能な書換え規則を見つけた場合，プロセスの置換えを行って新しい攻撃ゲームを生成する。
(3)　観測等価性を基に，攻撃ゲーム間の違いを識別できる確率を評価する。
(4)　安全性要件を満たすか否か判定する。

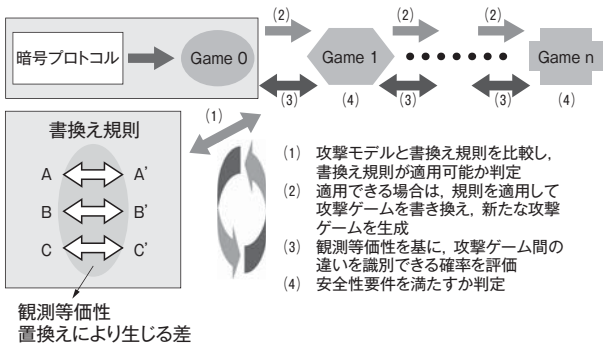


図2. CryptoVerifの処理手順 — CryptoVerifは，攻撃モデルと書換え規則が記述されたファイルが入力されると，(1)〜(4)の処理を繰り返し，安全性要件を満たす攻撃ゲームに変形できたとき，結果を出力し動作を停止する。

Security proof using CryptoVerif

(1)〜(4)の処理を繰り返し，安全性要件を満たす攻撃ゲームに変形できたとき，CryptoVerifは結果を出力し動作を停止する。書換え規則として与えられている観測等価性を評価し，評価結果を足し合わせることで，暗号プロトコルへの攻撃問題の難しさと計算量的仮定の差を定量的に評価する。

CryptoVerifは全自動モードの証明だけではなく，人間が適用対象の規則を指定する対話的なモードの証明もサポートしている。CryptoVerifによる全自動モードの証明は開発途上であり，全自動の証明に失敗したとしても，対話的なモードを用いることで証明可能となる方式が多く存在する。

B. Blanchetは，CryptoVerifで安全性証明を行うためのサンプルファイルも公開している。サンプルファイルには，落とし戸付き一方向性置換(注1)や，理想的な安全性を持ったハッシュ関数など，暗号プロトコルを構成するうえで基本的な部品が書換え規則として準備されており，これを用いて，公開鍵暗号，デジタル署名，及び鍵交換といった重要な基盤技術の安全性が証明できると言われている。

5　CryptoVerifの証明能力の解析と改良

CryptoVerifの証明能力を解析するために，われわれは新しい証明例の構成を行った。具体的には，新たに計算量的仮定を定式化し，それを用いて暗号プロトコルの証明を試みるという方法をとった。定式化の対象とした計算量的仮定は，安全性証明によく用いられるCDH（Computational Diffie-Hellman）仮定で定式化した記述を用いて，DH（Diffie-Hellman）鍵交換などの方式が自動証明できることを確認した[5]。定式化した書換え規則は，**図3**に示すLとRの二つのプロセスと，

(注1)　公開情報を用いる関数演算は容易であるが，秘密情報なしで逆関数演算を行うことが困難となる1対1関数のこと。

$$
\begin{aligned}
&L : \mathbf{foreach}\ i_k \le n_k \mathbf{do}\ O_{gr}() := r \xleftarrow{R} seed; \mathbf{return};\\
&\quad (O_G() := \mathbf{return}(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r));\\
&\quad |\mathbf{foreach}\ i_{fa} \le n_{f4a}\ \mathbf{do}\ O_{gA}() := a \xleftarrow{R} [1, |G_{\mathrm{p}}|]; \mathbf{return};\\
&\qquad (O_A() := \mathbf{return}(f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a))\\
&\qquad |\mathbf{foreach}\ i_{fb} \le n_{f4b}\ \mathbf{do}\ O_{gaB}() := b \xleftarrow{R} [1, |G_{\mathrm{p}}|]; \mathbf{return};\\
&\qquad\quad (O_{aB}() := \mathbf{return}(f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), b))\\
&\qquad\quad |\mathbf{foreach}\ i_1 < n_1\ \mathbf{do}\ O_{eq'}(z' : G_{\mathrm{p}}) :=\\
&\qquad\qquad \mathbf{return}(z' \overset{?}{=} f(\mathrm{Ggen1}(r), f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a), b))\\
&\qquad\quad |O_{ans}() := \mathbf{return}(f(\mathrm{Ggen1}(r), f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a), b)))))\\
&\approx_{p^{CDH4}=n_k n_{fa} n_{fb} \mathrm{Succ}^{CDH}_{\mathbb{G}}(t+(n_k-1)t_{\mathrm{Gen}}+(2n_k n_{f4a} n_{f4b}+n_k n_{f4a}-3;t_f)}\\
&R : \mathbf{foreach}\ i_k \le n_k \mathbf{do}\ O_{gr}() := r \xleftarrow{R} seed; \mathbf{return};\\
&\quad (O_G() := \mathbf{return}(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r))\\
&\quad |\mathbf{foreach}\ i_{f4a} \le n_{fa}\ \mathbf{do}\ O_{gA}() := a \xleftarrow{R} [1, |G_{\mathrm{p}}|]; \mathbf{return};\\
&\qquad (O_A() := \mathbf{return}(f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a))\\
&\qquad |\mathbf{foreach}\ i_{fab} \le n_{fab}\ \mathbf{do}\ O_{gaB}() := b \xleftarrow{R} [1, |G_{\mathrm{p}}|]; \mathbf{return};\\
&\qquad\quad (O_{aB}() := \mathbf{return}(f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), b))\\
&\qquad\quad |\mathbf{foreach}\ i_1 \le n_1 \mathbf{do}\ O_{eq'}(z' : G_{\mathrm{p}}) :=\\
&\qquad\qquad \mathbf{if\ defined}(k)\ \mathbf{then\ return}(z' \overset{?}{=} f(\mathrm{Ggen1}(r), f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a), b))\\
&\qquad\qquad \mathbf{else\ return}(\mathrm{false})\\
&\qquad\quad |O_{ans}() := k \leftarrow \mathrm{mark}; \mathbf{return}(f(\mathrm{Ggen1}(r), f(\mathrm{Ggen1}(r), \mathrm{Ggen2}(r), a), b)))))
\end{aligned}
$$

図3. CDH仮定の性質を表す書換え規則 — CryptoVerifによる安全性証明では，CDH仮定に基づき定式化した書換え規則は，LとRの二つのプロセスと，二つのプロセスのふるまいの違いを見分けられる確率で与えられる。

Rewriting rules for computational Diffie-Hellman (CDH) assumption

二つのプロセスのふるまいの違いを見分けられる確率で与えられる。

独自に定式化したCDH仮定を用いた安全性証明例の作成の過程で，計算量的仮定の定式化に採用するモデルが異なると，証明可能な暗号プロトコルも異なることがわかった。更に，できる限り強力な攻撃モデルを採用して定式化すると，様々な暗号プロトコルの証明に適用可能な書換え規則を定式化できることを明らかにした。

また，証明例の作成の過程で，CryptoVerifには次の二つの問題があることを明らかにした。

(1)　文法的に正しい入力を与えても，安全性証明の途中で異常終了し，証明結果が得られないことがある[6]。

(2)　安全性が証明できない状況であっても，安全であると誤った判定をすることがある[7]。

問題(1)の原因は，書換え規則が適用可能であるか判定する処理にあったことから，われわれは判定法を修正し，(1)が生じないように改良した。また，問題(2)の原因は，書換え規則の適用処理にあったことから，われわれは，①書換え規則の記述法を工夫する，②書換え規則の適用処理を修正する，という2通りの解決法を開発した。方法①は，同じ性質を表現した書換え規則であっても，証明対象の暗号方式に応じて書換え規則を変更する必要がある。一方，方法②は，証明対象が異なったとしても，同じ書換え規則で証明可能である。

オリジナルの書換え規則を用いた場合，オリジナルの書換え規則に工夫を加えた場合，及び改良したCryptoVerifを用いた場合の実験結果を**表2**に示す。従来のCryptoVerifでは，デジタル署名の一種であるGDH（Gap Diffie - Hellman）署名の安全性証明に失敗する。ある書換え規則を変更する

表2. CryptoVerifの改良結果

Results of improvement of CryptoVerif

CryptoVerif	GDH署名	[BR93] PKE	その他の証明例
オリジナル	×	○	○
オリジナル＋書換え規則変更	○	×	○
規則適用法を変更	○	○	○

〔BR93〕PKE：M. Bellareらが提案した公開鍵暗号方式

と，GDH署名の安全性証明は成功するものの，変更後の書換え規則を用いると，M. Bellareらが提案した公開鍵暗号方式[8]の証明に失敗する。CryptoVerifのゲーム変形法を改良したわれわれの方法では，書換え規則に変更を加えることなく，B. Blanchetとわれわれが与えたすべての例が正しく証明可能となった。

6　あとがき

ここでは，情報セキュリティ基盤技術の安全性を保証する概念である証明可能安全性を，自動的に検証する技術について述べた。

CryptoVerifでは，証明可能安全性の基となる安全性モデルと，ゲーム変形の基となる書換え規則を人間の手で定式化する必要がある。書換え規則の定式化には複雑な確率評価が必要となる場合があるが，定式化した規則は以降の安全性検証でも用いることができるという利点がある。

近年，安全性をソフトウェアで検証できることが標準化の要件になった例もある。今後，安全性証明の自動検証は，暗号プロトコルの設計・開発や標準化活動などに必須の技術となりうることから，証明能力の向上を目指し更なる研究・開発を進めていく。

文　献

(1) Dolev, D., et al. On the security of public key protocols. IEEE Transactions on Information Theory. **29**, 2, 1983, p.198－207.

(2) Blanchet, B., et al. "Automated security proofs with sequences of games", CRYPTO 2006. California, 2006-8, IACA. Heidelberg, Springer Berlin, 2006, p.537－554.

(3) 真野　建，ほか．ゲーム列による安全性証明の形式化と自動化．応用数理．**17**, 4, 2007. p.38－46.

(4) Shoup, V. "Sequences of games: a tool for taming complexity in security proofs". Cryptology ePrint Archive : Report 2004/332. 2004. ＜http://eprint.iacr.org/2004/332＞，（参照2009-05-11）.

(5) 花谷嘉一，ほか．“Blanchetフレームワークにおける CDH仮定の取り扱い”．暗号と情報セキュリティシンポジウム（SCIS）2008. 宮崎，2008-01，電子情報通信学会．4E2-4.

(6) 角野陽輔，ほか．“安全性検証ツールCryptoVerifの改良：異常終了に対する一対策”．SCIS2009. 大津，2009-01，電子情報通信学会．4C2-1.

(7) 花谷嘉一，ほか．“CryptoVerifの証明能力の改良：誤った判定の回避”．日本応用数理学会2009年春の研究部会連合発表会「数理的技法による情報セキュリティ」研究部会．＜http://nicosia.is.s.u-tokyo.ac.jp/jsiam-fais/2009-spring-jsiam.html＞，（参照 2009-05-11）.

(8) Bellare, M., et al. "Random oracles are practical: A paradigm for designing efficient protocols". Proceedings of the 1st ACM Conference on Computer and Communications Security. Virginia, 1993-11, SIGSIS. NY, ACM, 1993, p.62－73.

花谷　嘉一　HANATANI Yoshikazu

研究開発センター コンピュータ・ネットワークラボラトリー。暗号技術の研究・開発に従事。日本応用数理学会会員。
Computer & Network Systems Lab.

村谷　博文　MURATANI Hirofumi, D.Sc.

研究開発センター コンピュータ・ネットワークラボラトリー主任研究員，理博。暗号技術及び暗号応用システムの研究・開発に従事。電子情報通信学会，情報処理学会，IEEE会員。
Computer & Network Systems Lab.

太田　和夫　OHTA Kazuo, D.Sc.

電気通信大学 情報通信工学科教授，理博。情報セキュリティ及び暗号理論の研究に従事。国際暗号学会（IACR），電子情報通信学会会員。
The University of Electro-Communications