6-7　暗号プロトコルの安全性評価技術の研究開発

吉田真紀

暗号プロトコルはインターネット上で安全な通信を実現するため広く利用されている。しかし、仕様上の問題から生じる深刻な脆弱性がしばしば指摘されており、安全性を評価する技術を確立することが極めて重要である。本稿では、暗号プロトコルの安全性評価に関するセキュリティアーキテクチャ研究室の主要な成果を紹介する。

1　まえがき

暗号プロトコルと呼ばれる暗号を用いた通信プロトコルの目的は、情報通信における盗聴やなりすましの防止といった安全性確立である。近年、暗号プロトコルに仕様上の問題から生じる深刻な脆弱性が頻繁に指摘されている [1]–[3]。例えば、2014 年 10 月 14 日（日本時間 15 日）、Web 上の安全な通信のために広く普及した SSLv3（Secure Socket Layer version 3.0）に対して、POODLE と呼ばれる仕様上の脆弱性をついた攻撃が発見され、通信内容が漏洩する可能性が指摘された [1]。このような脆弱性を早期に発見するためにも、暗号プロトコルの安全性評価技術の研究開発が喫緊の課題となっている。

一般に、暗号プロトコルの安全性評価では使用している暗号は安全という前提の下で通信内容の漏洩やなりすましが可能となる仕様上の欠陥がないかを確認する。安全性評価の方法として、人手による数学的証明と形式手法による自動検証がある。人手による数学的証明の場合、今までに蓄積された知見により多様な暗号プロトコルの評価が可能となるが、手間がかかり単純な誤解から攻撃を見逃す可能性がある。一方、形式手法の場合、暗号プロトコルを形式化して記述することで計算機による自動化が可能となり、手間が削減され単純な誤解や論理の飛躍を防ぐことができる。しかし、あらゆる暗号プロトコルを扱う自動化方法は存在しないことや、形式化（抽象化）の不備による攻撃見逃しが生じ得る。すなわち、人手による数学的証明と形式手法による自動検証には一長一短あり、安全性評価技術の研究開発では多角的な観点から様々な方法を考案することが重要となる。

本稿では、安全性評価技術に関する、人手による数学的証明と形式手法による自動証明の両方の成果 [4][5] を紹介する。人手による数学的証明の成果 [4] では、暗号プロトコルの性能限界を解明した。これにより達成目標が明らかとなるだけでなく、性能が限界を超していれば何らかの欠陥をもつことがわかり、安全性評価の基準として役立つ。形式手法による自動検証の成果 [5] では、対象とする暗号プロトコルは限られるが、攻撃を網羅できることを保証する形式化を提案した。これにより、安全性評価における攻撃見逃しの防止と手間の削減に寄与する。

以降の章構成を示す。まず、**2** で、人手による数学的証明の成果 [4] である、暗号プロトコルの性能限界の解明について紹介する。次に、**3** で、形式手法による自動検証の成果 [5] である、自動検証のための暗号プロトコルの形式化について紹介する。最後に **4** でまとめと将来の展望を述べる。

2　暗号プロトコルの性能限界

本研究では、秘匿計算（Secure Multi-party Computation: MPC）を実現する任意のプロトコルを対象とする。秘匿計算とは、複数の参加者が用意したデータを元に協力して様々な“タスク”を実行することを目的としており、Yao [6] によって導入された。ここで、各参加者の用意したデータは秘匿される。秘匿計算は暗号分野おいて極めて重要な技術であり、様々なモデルが提案されている [7]–[9]。本研究では特に、参加者間でのやりとりが一切不要（非対話型）であり、参加者の用意したデータだけでなく、どのようなタスクを実行しているかも秘匿する MPC（Non-interactive MPC: NIMPC）[9] を対象とする。本章では、まず非対話型のモデルの定義を示し、次に本研究の成果である性能の限界解明について紹介する。

2.1　非対話型モデル

文献 [9] で導入された NIMPC では、参加者間のやりとりを完全に排除した上で、可能な限り強い安全性を実現する。安全性は、プロトコルに従うが不正に情

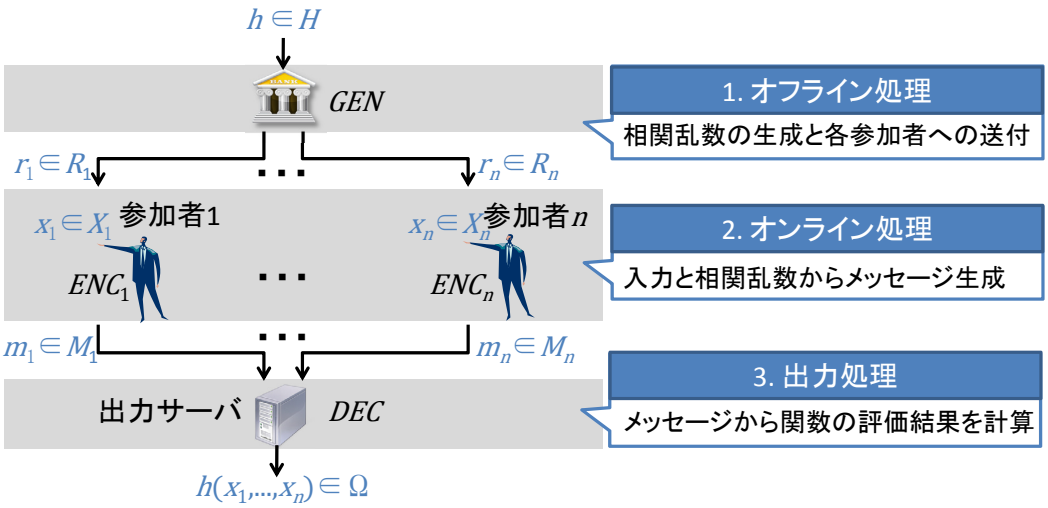

図 1　NIMPC プロトコルの処理 $\boldsymbol{P}(\boldsymbol{\Pi})$

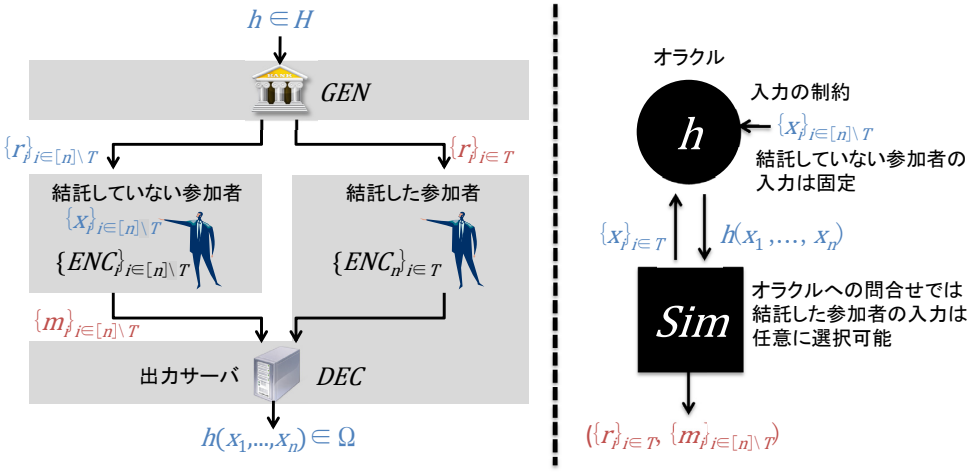

図 2　NIMPC プロトコルの安全性

報を入手しようとする（honest-but-curious）攻撃者に対して情報理論的に定義される。参加者数を n とし、$[n] = \{1, \ldots, n\}$ と定義する。参加者が実行できるタスクの全候補は関数族 H で定義され、参加者間が用意したデータを元に協力してタスクを実行することは各参加者 $i \in [n]$ がもつ入力 $x_i \in X_i$ に対して、関数族の元である関数 $h: X_1 \times \cdots X_n \to \Omega$ を評価することで表現される。

関数族 H に対する NIMPC プロトコルは 3 種のアルゴリズム $\Pi = (GEN, \{ENC_i\}, DEC)$ からなる。

- 乱数生成 $GEN: H \longrightarrow R_1 \times \cdots \times R_n$ は、評価対象とする関数 $h \in H$ から各参加者 $i \in [n]$ 向けに、ある種の相関をもつ乱数（以降、相関乱数）$r_i \in R_i$ を出力する。
- 符号化 $ENC_i: X_i \times R_i \longrightarrow M_i$ は、参加者 $i \in [n]$ がもつ入力 $x_i \in X_i$ と相関乱数 $r_i \in R_i$ からメッセージ $m_i \in M_i$ を出力する。
- 復号 $DEC: M_1 \times \cdots \times M_n \longrightarrow \Omega$ は、参加者 $i \in [n]$ のメッセージ $m_i \in M_i$ から関数の評価結果 $h(x_1, \ldots, x_n) \in \Omega$ を復元する。

これらのアルゴリズム $\Pi = (GEN, \{ENC_i\}, DEC)$ を用いた NIMPC プロトコルの処理 $P(\Pi)$ は、外部の参加者（出力サーバ）も参加した以下の 3 つの処理からなる（図 1）。

- オフライン処理：各参加者 $i \in [n]$ は、事前に相関乱数 $r_i \in R_i$ を受信する。
- オンライン処理：各参加者 $i \in [n]$ は、入力 $x_i \in X_i$ から相関乱数 $r_i \in R_i$ を使ってメッセージ $m_i \in M_i$ を計算する。
- 出力処理：出力サーバは、各参加者のメッセージ $m_i \in M_i$ から関数の評価結果 $h(x_1, \ldots, x_n) \in \Omega$ を計算し出力する。

ここで、参加者の通信複雑度は相関乱数長 $\log_2|R_i|$ とメッセージ長 $\log_2|M_i|$ の最大値とする。なお、集合 X

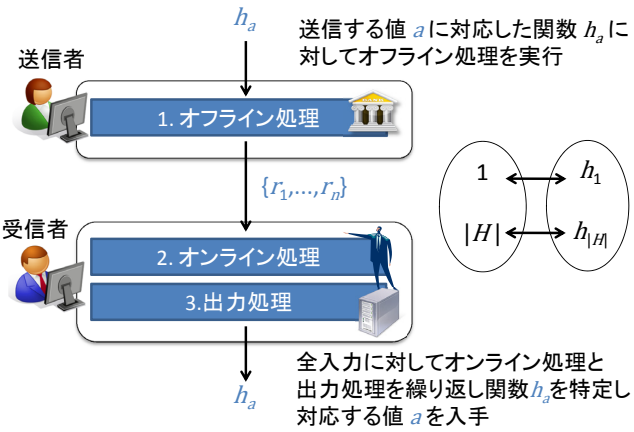

図 3　NIMPC プロトコルを用いたデータ送信

に対して $|X|$ はその要素数（cardinality）を表す。

関数族 H に対する NIMPC プロトコルの正しさは以下のように定義される。任意の関数 $h \in H$、各参加者 $i \in [n]$ の任意の入力 $x_i \in X_i$、任意の相関乱数 $(r_1, \ldots, r_n) \leftarrow GEN(h)$ に対して、$DEC(ENC_1(x_i, r_i), \ldots, ENC_n(x_n, r_n)) = h(x_1, \ldots, x_n)$ が成立する。すなわち、出力サーバによって正しい評価結果が計算される。

関数族 H に対する NIMPC プロトコル $P(\Pi)$ の安全性は、不正に情報を入手しようとする攻撃者と結託した参加者集合 $T \subseteq [n]$ への耐性（T-robustness）として定義される。定義では、評価対象の関数 h に関する情報と結託していない参加者の入力 x_i $(i \notin T)$ に関する情報が一切入手できないことが要求され、結託参加者と出力サーバの観測結果（結託参加者の相関乱数と結託していない参加者のメッセージ）をシミュレートできることと定義される（図 2）。

2.2　性能の限界解明

本研究では任意の参加者の結託 $T \subseteq [n]$ への耐性、すなわち完全耐性（full robustness）をもつ任意の NIMPC プロトコルに共通する性能の限界を初めて明らかとした。具体的には、通信量の削減の限界である通信複雑度の下限を導出した。これにより例えば、ある NIMPC プロトコル $P(\Pi)$ の通信量が導出した下限よりも少ない場合、$P(\Pi)$ は完全耐性ではなく何らかの情報（参加者間でどのようなデータを元にどのようなタスクをしているか）が漏れることが分かる。

導出した通信複雑度の下限は、対象とする関数族 H に対し、$\log_2 |H|$ となる。すなわち、評価できる関数が多いほど通信複雑度は大きくなる。ここでは、通信複雑度の下限導出の方針を説明する。まず、関数族 H に対する通信複雑度 C の NIMPC プロトコル $P(\Pi)$ が存在すると仮定する。そして、$P(\Pi)$ をブラックボックス的に利用することで乱数 $a \in \{1, \ldots, |H|\}$ を通信量 C で送信できることを示す。もし、通信複雑度が乱数の長さより短ければ、すなわち $C < \log_2 |H|$ であれば矛盾となるため、$C \geq \log_2 |H|$ が成立する。よって、通信複雑度の下限は $\log_2 |H|$ となり、評価対象とする関数の個数に依存する。なお、乱数 $a \in \{1, \ldots, |H|\}$ の送信は、$\{1, \ldots, |H|\}$ を H と一対一対応付けし、対応する関数の NIMPC プロトコル $P(\Pi)$ での評価で実現できる（図 3）。

表 1　NIMPC プロトコルの通信複雑度

	任意の関数（m ビット出力）	任意の指示関数（1 ビット出力）
導出した下限	$\|X\| \cdot m$	$\log_2 \|X\|$
従来プロトコル [9]	$\|X\| \cdot m \cdot d^2 \cdot n$	$d^2 \cdot n$
提案プロトコル	$\|X\| \cdot m \cdot \lceil \log_2 d \rceil^2 \cdot n$	$\lceil \log_2 d \rceil^2 \cdot n$

上述の通信複雑度の下限導出では、NIMPC プロトコルの正しさしか使っていないため、情報理論的な安全性だけでなく、計算量的な安全性をもつ NIMPC プロトコルにも共通する性能の限界である。

表 1 に主要な関数族の通信複雑度の下限を示す。指示関数（indicator function）とは、特定の入力の場合のみ出力が 1 となり、それ以外の場合は 0 となる関数であり、任意の関数を構成可能とする。指示関数の個数は入力の数 $|X|$ と等しいため、通信複雑度の下限は入力長 $\log_2 |X|$ となる。一方、任意の m ビット出力の関数の個数は $2^{|X| \cdot m}$ となり、通信複雑度の下限は入力長の指数 $|X| \cdot m$ となる。それに対して、従来研究 [9] で提案されている NIMPC プロトコルの通信複雑度は、入力長に対して指数的なギャップ（$d = \max_{i \in [n]} |X_i|$ に対し、$d^2 \cdot n$）があった。このギャップが意味することは、安全性に欠陥をもたせることなく、効率を向上できる可能性があることである。実際、本研究では、効率を大幅に向上させ、ギャップを入力長に対する二次多項式（$\lceil \log_2 d \rceil^2 \cdot n$）まで削減した NIMPC プロトコルを提案した。

3　自動検証のための暗号プロトコルの形式化

本研究では、公開鍵暗号基盤（Public-key Infrastracture: PKI）で共通の鍵を利用する暗号プロトコルを対象とし、Canetti と Herzog によって提案された自動検証のアプローチ [10] の下で形式化を提案した。本章では、まず [10] の自動検証のアプローチを紹介し、次に本研究の成果である暗号プロトコルの形式化の概要を示す。

3.1　自動検証のアプローチ

形式手法による自動検証における主要な課題は、形式化において値や操作を記号で適切に表現することである。ここで適切とは、攻撃を漏れなく網羅的に表現するための具体化と、攻撃の有無判定を効率化するための単純化を両立することである。この課題を解決するためにCanettiとHerzogが[10]で提案したアプローチは以下のとおりである。

- 汎用的結合可能性（Universally Composability: UC）の枠組みを利用：UC [11]では、暗号プロトコルが単体単一セッションの実行で安全ならば、任意のプロトコルの任意のセッションと並行して実行しても安全であること（結合可能性）が保証される。これにより、単体単一セッションの実行を表現できれば十分である。
- 記号的モデルの計算論的健全性の保証：記号的モデルが計算論的に健全であるとは、記号的に表現したモデルで攻撃が無ければ、UCの枠組みでも攻撃が無いことである。UCの枠組みにおいて攻撃の有無を判定するのは、“無視できない確率で起きる”実行に絞られる。これにより、“無視できない確率で起きる”実行を記号的に表現できれば十分となる。

上述の2つにより、単体単一セッション内で無視できない確率で起きる実行に絞って攻撃の有無を判定すれば十分となり、検証ツールによる自動化が可能となる。

UCの枠組みを利用した自動検証の研究はいくつか知られているが[10][12]、それらの研究はJUC（UC with Joint State）[13]と呼ばれる枠組みを利用している。ここで、JUCの枠組みとは、同一の暗号プロトコルのセッション間で状態（鍵）を共有しても結合可能性が保証される枠組みである（図4右）。言い換えると、暗号プロトコルごとに専用の鍵を使用する実行環境を対象としている。

一方、本研究では、インターネットで使用されているTLSなどの実プロトコルのように、PKIを用いて任意の暗号プロトコル、任意のセッション間で同じ鍵を使用する実行環境を対象とする。そのために、UCの枠組みとしてEUC（Externalized UC: EUC）[14]を利用する。ここでEUCの枠組みとは、新たにPKIをモデル化した理想的な機能（共有機能）を導入し、共有機能から入手した鍵を任意の暗号プロトコル、任意のセッション間で共有しても（使い回しても）結合可能性が保証される枠組みである（図4左）。

UCの枠組みとしてEUCを利用した本研究においても、従来研究と同様、記号的モデルの計算論的健全性の保証が、上述のアプローチにおける最重要課題となる。計算論的健全性の保証では、まず評価対象とする暗号プロトコルの構文を定義し、UCの枠組み（モデル）と記号的モデルにおける解釈を定義する（図5）。次に、それぞれ解釈の下で暗号プロトコルの正当な実行の履歴を定義する。そして、EUCモデルにおいて無視できない確率で起きる任意の正当な実行履歴に対して、対応する正当な記号的実行履歴が存在すること（マッピング補題）を証明する。これにより、記号的モデルで攻撃が無ければ、UCモデルでも攻撃が無いことが保証される。なお、マッピング補題の証明では、使用されている暗号の安全性を仮定し、対応する記号的実行履歴が正当でない（あるいは存在しない）ならば、暗号が安全でないでないことを示すことで矛盾を導出する。よって、暗号文や署名などの暗号的なデータの適切な記号化が不可欠となる。

3.2　記号的モデルの提案と計算論的健全性の証明方針

本研究では、EUCの枠組みに対して計算論的に健全な記号的モデルを提案した。以降では、まずEUCを対象とすることでマッピング補題の証明において、どのような課題が生じるかを述べ、その課題の解決方針を示す。

計算論的健全性の保証の最初のステップである記号的モデルでの解釈の定義では、暗号プロトコルで扱う

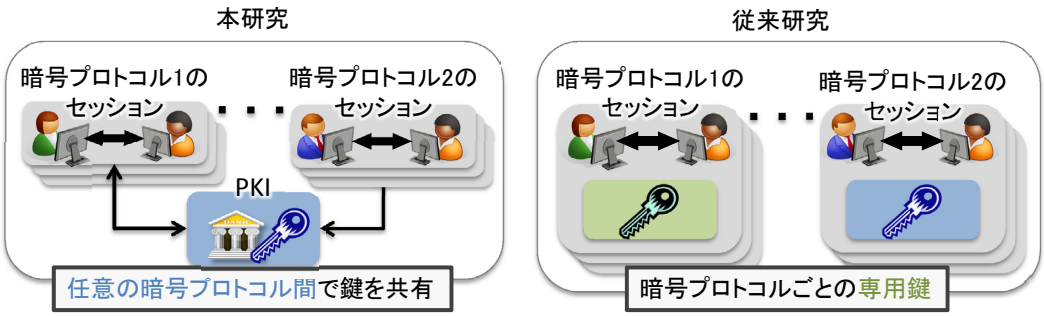

図4　本研究と従来研究で対象とする実行環境の違い

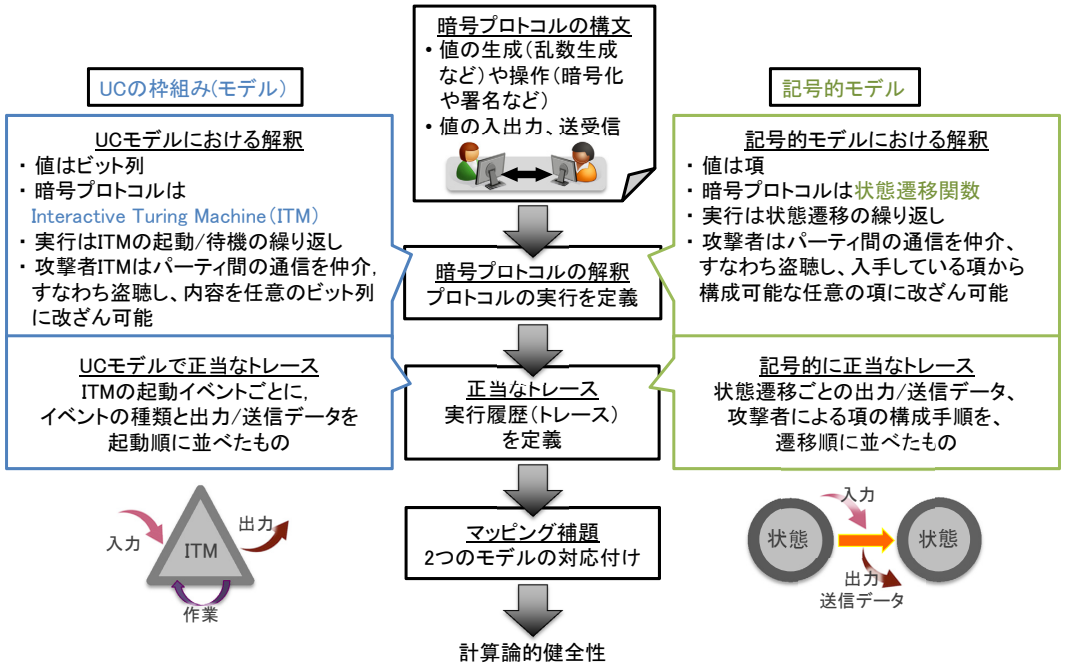

図 5　記号的モデルの計算論的健全性の保証の流れ

値や操作を表す記号を用意する。例えば、平文、暗号化・復号の操作と鍵、署名・検証の操作と鍵を表す記号として、$m, Enc, Dec, ek, dk, Sig, Ver, sk, vk$ を用意すれば、それらの記号を使って正当なトレースに含まれる暗号文や署名文を $Enc(ek, m), Sig(sk, m)$ の項で表すことができる。マッピング補題の証明における対応付けでは、暗号プロトコルに従って生成された値以外に、攻撃者によって不正に生成された値を考慮する必要がある。例えば、攻撃者が暗号文や署名文とタグ付けして送付した値であっても、不正に生成された可能性があり、$Enc(ek, m), Sig(sk, m)$ と対応付けて良いか否かの判断が問題となる。

JUC の枠組みの場合 [10][12]、使用されている暗号を JUC 安全なものとすることで、理想機能に置き換えることができる。理想機能で生成される暗号文と、それを復号した結果は正しく生成されたと分かるため、記号化結果の $Enc(ek, m)$ 及び m に対応付けして良い。一方、それ以外の「暗号文、復号結果とされる値」はまとめて“garbage data”（記号化不要なデータ）G に対応付けられる。

一方、本研究で扱う EUC の枠組みの場合、EUC 安全な暗号や署名は知られていないため、「暗号文、復号結果とされる値」が正しく生成されたとみなすか、garbage data とみなすかの新たな判断基準が必要となる。

本研究では新たな判断基準として、EUC の特徴である共有機能に着目する。共有機能は PKI をモデル化した理想機能であり、生成された鍵の正しさが保証される。よって、その鍵を用いて正しさを確認できる値を記号化結果と対応付けし、それ以外を garbage data G に対応付ける。これにより、使用されている暗号が EUC 安全でなくとも、マッピング補題の証明における対応付けができ、計算論的健全性が保証できる。

4　あとがき

本稿では、セキュリティアーキテクチャ研究室の研究成果のうち、暗号プロトコルの安全性評価技術に関する主要な成果を紹介した。成果の 1 つめ [4] は、非対話型の秘匿計算（NIMPC）を実現するあらゆる暗号プロトコルが共通してもつ性能の限界解明である。成果の 2 つめ [5] は、PKI で鍵を共有する暗号プロトコルを自動検証するための形式化である。

従来の安全性評価では、使用されている暗号は安全という前提が基本であった。近年、TLS（Transport Layer Security）に対して、Logjam 攻撃 [2]（2015 年 5 月）や SLOTH 攻撃 [3]（2016 年 1 月）など、脆弱な暗号を使わせる中間者攻撃が発見されている。よって、今後の展開として、暗号プロトコルで安全でない暗号も使用されることを想定した安全性評価技術の研究開発と社会実装が挙げられる。

【参考文献】

1 Möller B., Duong T., and Kotowicz K., “This POODLE bites: SSL 3.0 fallback (security advisory),” https://www.openssl.org/ bodo/ssl-poodle.pdf, 2014.
2 Adrian D., Bhargavan K., Durumeric Z., Gaudry P., Green M., Halderman J.A., Heninger N., Springall D., Thomè E., Valenta L., VanderSloot B., Wustrow E., Zanella-Bèguelin S., and Zimmermann P., “Imperfect forward secrecy: How Diffie-Hellman fails in practice,” In: The 22nd ACM Conference on Computer and Communications Security (ACM CCS 2015), pp.5–17, 2015.
3 Bhargavan K. and Leurent G., “Transcript collision attacks: Breaking authentication in TLS, IKE, and SSH,” In: The 23nd Annual Network and Distributed System Security Symposium 2016 (NDSS 2016).
4 Yoshida M. and Obana S., “On the (in) efficiency of non-interactive secure multiparty computation,” In: The 19th Annual International Conference on Information Security and Cryptology (ICISC 2015), LNCS, vol.9558, pp.185–193, Springer, Heidelberg, 2016.
5 吉田真紀 , 鈴木斎輝 , 藤原融 , “Externalized Universally Composable の枠組みに対する記号的モデル ,” 日本応用数理学会 2014 年春の連合発表会，数理的技法による情報セキュリティ , 2014 年 3 月 .
6 Yao A.C., “Protocols for secure computations,” In: The 23rd Annual Symposium on Foundations of Computer Science (FOCS '82) pp.160–164, 1982.
7 Chaum D., Crèpeau C., and Damgård I., “Multiparty unconditionally secure protocols,” In: The 20th Annual ACM Symposium on Theory of Computing (STOC '88), pp.11–19, 1988.
8 Hirt M. and Maurer U., “Player simulation and general adversary structures in perfect multiparty computation, ” In: Journal of Cryptology, 13(1), pp.31–60. Springer, Heidelberg, 2000.
9 Beimel A., Ishai Y., Kushilevitz E., Meldgaard S., and Paskin-Cherniavsky A., “Non-interactive secure multiparty computation,” In: The 34th Annual International Cryptology Conference (CRYPTO 2014). LNCS, vol.8617, pp.387–404, Springer, Heidelberg, 2014.
10 Canetti R. and Herzog J., “Universally composable symbolic analysis of mutual authentication and key-exchange protocols,” In: The Third Theory of Cryptology Conference (TCC 2006), LNCS, vol.3876, pp.380–403, Springer, Heidelberg, 2006.
11 Canetti R., “Universally composable security: A new paradigm for cryptographic protocols,” In: The 42nd Annual Symposium on Foundations of Computer Science (FOCS 2001). pp.136–145. IEEE Computer Society, 2001.
12 Dahl M. and Damgård I., “Universally composable symbolic analysis for two-party protocols based on homomorphic encryption,” In: The 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques (Eurocrypt 2014), LNCS, vol.8441, pp.695–712, Springer, Heidelberg, 2014.
13 Canetti R. and Rabin T., “Universal composition with joint state,” In: The 23rd Annual International Cryptology Conference (CRYPTO 2003). LNCS, vol.2729, pp.265–281, Springer, Heidelberg, 2003.
14 Canetti R., Dodis Y., Pass R., and Walfish S., “Universally composable security with global setup,” In: The Fourth Theory of Cryptology Conference (TCC 2007). LNCS, vol.4392, pp.61–85, Springer, Heidelberg, 2007.

吉田真紀　（よしだ　まき）

サイバーセキュリティ研究所
セキュリティ基盤研究室
主任研究員
博士（工学）
情報セキュリティ、暗号理論、
情報ハイディング