汎用的結合可能性による暗号システムの安全性証明

岡本　龍明[†a)]　　真鍋　義文[††b)]

Security Proof of Cryptographic Systems Using Universal Composability Theory

Tatsuaki OKAMOTO[†a)] and Yoshifumi MANABE[††b)]

あらまし　暗号及び暗号応用システムの安全性を検証する方法として，従来は攻撃ベースの定式化が考えられてきた．近年，暗号プロトコルを組み合わせた複雑なシステムが設計される場合が増えてきている．従来の安全性定義はこのような場合を考慮していないために，プロトコルの組合せが安全性を損なう場合がある．この問題点を解決するため，汎用的結合可能性（UC）の理論が提唱された．UC 安全と証明されたプロトコルは，他のどのようなプロトコルと組み合わせて使用されてもその安全性が保証される．本論文ではこの理論の概要を述べるとともに，関連した安全性証明に関する数理的技法の研究を紹介する．

キーワード　暗号，数理的技法，汎用的結合可能性，安全性証明

1. ま え が き

暗号及び暗号応用システムの安全性を検証する方法としては，計算量理論に基づく安全性証明手法（計算論的証明手法）が標準的手法として暗号理論において確立している．一方，それら安全性を数理的技法（記号論理的証明手法）の立場からとらえた研究も活発に行われてきた．しかし，従来これら二つの手法の相互関係についてはほとんど研究されていなかった．

2000 年ごろより，標準的安全性証明（計算論的証明）に対する数理的技法の有効性（健全性/完全性）などが研究されるようになり，数理的技法が，計算論的証明手法の簡明化・機械化に有効であることが明らかになりつつある．

本論文では，最近急速に進展しつつある新しい計算論的証明手法，特に汎用的結合可能性の立場から，これら数理的技法との関係について述べる．

† 日本電信電話株式会社　NTT 情報流通プラットフォーム研究所，武蔵野市
NTT Information Sharing Platform Laboratories, NTT Corporation, 3–9–11 Midori-cho, Musashino-shi, 180–8585 Japan
†† 日本電信電話株式会社　NTT コミュニケーション科学基礎研究所，厚木市
NTT Communication Science Laboratories, NTT Corporation, 3–1 Morinosato-Wakamiya, Atsugi-shi, 243–0198 Japan
a) E-mail: okamoto.tatsuaki@lab.ntt.co.jp
b) E-mail: manabe.yoshifumi@lab.ntt.co.jp

2. 現代暗号の考える暗号機能とその安全性

まず現代暗号の考える安全性について説明しよう．現代暗号のもつ大きな特徴が，非常に強い意味の安全性を考えるということである．例えば，現代暗号では，暗号の安全性は単に暗号文からもとのメッセージ全体が露呈しないという最も素朴な安全性のみならず，暗号文からもとのメッセージの「いかなる部分情報」も露呈しないといった，より強い意味の安全性をもつことが求められるようになってきている．また，暗号文を解読しようとする攻撃のモデルも，攻撃者が単に通信文を受動的に盗聴するだけではなく，攻撃者が選んだ暗号文の復号結果を得られるような状況を考慮するようになってきている．

このような強い意味の安全性を考えることには，以下のような現実的な意味がある．現代暗号は，単に秘密情報を隠して送るという最も基本的な機能を超えて，秘密情報に関する様々な機能を実現しようとする総合的な科学として発展している．

例えば，現代暗号で盛んに研究されているテーマの一つが，暗号プロトコルであり，それは，複数の参加者が自分の秘密情報を隠したまま，全員が協調してある計算を安全に行う通信手順/アルゴリズムのことである．一例として，選挙のプロトコルでは，各参加者（投票者）の投票内容を秘密にしたまま，それぞれの候補者に何人の正当な投票者が投票したかを安全に計

算する．ここでは，秘匿機能は単に一基本機能にすぎず，正当性の認証機能や（ゼロ知識）証明機能などと複合的に組み合わせることで，選挙プロトコルのような高度なセキュリティ機能を実現する．

このとき，暗号機能を単独で利用する場合には問題が生じなくても利用状況においては安全でない場合がしばしば起こるのである．また，暗号機能は他の暗号（認証）機能と併用されることも多い．この場合，複数の暗号（認証）機能が相互に干渉しあって，安全が損なわれることもあるのである．

つまり，暗号機能がインターネットなどにおける多くのアプリケーションで必須機能と認識され，広範に使われるようになればなるほど，「どのような環境で利用しても安全である」ことが要求されるようになってきたのである．

暗号理論分野において（つまり，計算論的手法で），このような安全性を定式化する方法として，以下の二つのアプローチがある．

［攻撃ベース定式化］ 攻撃者とチャレンジャー間のゲームとして定式化するアプローチである．例えば，公開鍵暗号，ディジタル署名など多くの暗号機能の安全性がこの方法で定式化されている．このように定式化された安全性を証明する方法として，最近，ゲームを徐々に変換して作るゲーム列を解析することで，安全性を証明するような手法が発展しつつある．

［シミュレーションベース定式化］ 実際の方式（現実モデル）と，理想的な機能を使って現実をシミュレーションしたもの（理想モデル）との間のギャップが無視できるような場合に，この実際の方式が理想的な機能と同等の性質をもつととらえることにより，安全性を定式化する．このようなアプローチにより，暗号におけるすべての対象に対する統一的な安全性の定式化が可能となる．このようなアプローチの典型が，Canetti により提唱された汎用的結合可能性（Universal Composability：以下 UC と略すこともある）である [3], [4].

本論文では，様々な暗号プロトコルの基礎となる問題の一つである鍵交換問題を例として UC 理論の説明を行う．そのため，まず鍵交換機能について，従来の攻撃ベースの定式化を示したのち，UC における鍵交換機能の定式化を示す．

3. 攻撃ベース定式化の具体例：鍵認証交換の定式化

鍵交換は，安全でない（盗聴されているかもしれない）通信路を用いて，二人の参加者 Alice と Bob がある鍵の値を安全に共有する問題である．有名な Diffie-Hellman-Merkle（DHM）鍵交換方式(注1) [9] は盗聴者のみが存在する受動的な攻撃に対しては安全であるが，以下のような能動的な攻撃に対しては安全ではない．

今，Alice と Bob の通信路の中間にいて通信に能動的に関与する敵 Matt を考える．Matt は，Alice との間では，K_A という鍵を共有し，Bob との間では，K_B という鍵を共有する．Alice と Bob は，互いに正しい相手と鍵交換したと信じているため，Alice は鍵 K_A を用いて Bob に対して暗号通信を行い，Bob は鍵 K_B を用いて Alice に対して暗号通信を行う．一方，Matt は，Alice の Bob あての暗号文は，鍵 K_A を用いて復号し，その復号した平文を鍵 K_B を用いて暗号化して，その暗号文を Bob に送る．Bob の Alice あての暗号文も同様に処理する．このようにすると，Alice と Bob は，何の支障もなく通信ができるが，その通信内容はすべて Matt に盗聴されていることになる．

このような能動的攻撃は，「中間者（Man-In-the-Middle：MIM）攻撃」と呼ばれている．上記の例で分かるように，DHM 鍵交換方式は中間者攻撃に対しては安全でない．

上記の中間者攻撃が可能となった原因は，Alice が，通信している相手が Bob なのか Matt なのかを確認しないまま，Bob と決めつけて暗号通信を行ったことにあるので，通信相手の認証を行うことで上記の問題は回避される．このように，何らかの手段で通信相手の認証を行う鍵交換を「認証鍵交換（Authenticated Key Exchange：AKE）」と呼ぶ．そのうち，認証として公開鍵インフラストラクチャ（PKI）を利用する鍵交換を PKI ベース AKE と呼ぶ．PKI は，現在，電子署名法などの法的制度も整備され，社会的インフラストラクチャとして定着しつつある．PKI では，認証機関（CA）と呼ばれる（認定された）機関が，各利用者の本人性・正当性を何らかの方法で確認した後，その公開鍵に対して（電子的な）証明書を発行する．これは，各利用者の印鑑に対して役所が印鑑証明を発行することに対応している．本論文で言及する鍵交換方式はすべて PKI ベース AKE である．

Canetti と Krawczyk により，AKE の攻撃ベース

(注1)：一般にはこの方式は Diffie-Hellman 鍵交換方式と呼ばれているが，Hellman はこの問題を最初に考察した Merkle に敬意を表してこのように呼んでいる．

の定式化が行われている [6]．ここでは，その定式化を CK 安全性と呼ぶ．

この定式化では，各参加者（Alice，Bob など）と敵（Matt）が対象となり，すべての参加者間の通信は Matt により支配されていると考える（つまり，Matt は，盗聴，通信文の差し替え，廃棄などを自由にできる）．ここで，ある参加者が別の参加者と鍵交換のプロトコルを実行したとき，そのプロトコルの各参加者における入出力を「セッション」と呼ぶ（つまり，セッションは，各参加者ごとに定まる）．Alice と Bob が鍵交換のプロトコルを実行したときは，Alice のセッション sid_A と Bob のセッション sid_B が生じる．このとき，互いに交信したセッション sid_A と sid_B を「マッチングセッション」と呼ぶ．Alice と誰かとが鍵交換プロトコルを実行し，Alice がその相手と交換できたと考えた鍵を Alice のセッション sid_A の「セッション鍵」K_A と呼ぶ．

さて，Matt は，単に通信を支配するだけでなく，次の攻撃を許される．

セッション状態攻撃：ある参加者のセッションの状態（セッションをまたがる長期的な情報は含まれず，あくまでこのセッションだけに固有の情報）を Matt は知ることができる．

セッション鍵攻撃：あるセッションのセッション鍵を Matt は知ることができる．

参加者攻撃：ある参加者のもっているすべての情報を知ることができ，またその参加者の行動を支配できる．

次に，あるセッション sid が「攻撃されていない」ことを以下の条件（1）〜（3）をすべて満足することと定義する．

（1） セッション sid に関与するどちらの参加者も「参加者攻撃」を受けていない．

（2） セッション sid が「セッション状態攻撃」と「セッション鍵攻撃」のいずれも受けていない．

（3） セッション sid のマッチングセッションが「セッション状態攻撃」と「セッション鍵攻撃」のいずれも受けていない．

次に，敵 Matt は，ある「攻撃されていない」セッション sid を指定して，そこで次の二つのケースを区別できるかどうかのゲームを実行する．

（1） そのセッション sid のセッション鍵 K_{sid} を Matt に与える．

（2） K_{sid} と同じサイズのランダムな鍵を Matt に与える．

どのような敵 Matt も上の二つのケースを十分に区別できないとき，CK 安全であるという．

この CK 安全性が「中間者攻撃」に対処していることに注意されたい．中間者攻撃をする敵は，Alice と Bob それぞれに，別のセッションで通信しているため，それらはマッチングセッションにはならない．したがって，中間者攻撃で Bob との間のセッションでどのような攻撃をしても（しなくても），Alice との間で鍵交換ができると，敵は上記の（1）と（2）のケースを区別できるので，上の CK 安全性に反することになる．

それでは，CK 安全性を満足する鍵交換方式はあるのだろうか．Krawczyk は，ハッシュ関数を理想化したランダムオラクルモデルにおいて，整数論的に妥当な仮定の下で，HMQV 鍵交換方式 [10] がこの安全性を満足することを示している．

4. シミュレーションベース定式化：汎用的結合可能性（UC）

4.1 UC の定式化

汎用的結合可能性（UC）では，ある暗号機能を実行するプロトコルが理想的状況（理想的な機能を利用して実行）を模倣することができるならば安全と考える．例えば，ある公開鍵暗号方式が，理想的な公開鍵暗号機能をうまく模倣できることを示せば，その公開鍵暗号方式は理想的な公開鍵暗号と同様の安全性をもつと考える．

まず，実現したい理想機能（functionality）$\mathcal{F}$ を記述する．ここで，$\mathcal{F}$ とはある暗号機能を実現するための理想的な信頼できるサービスである．例えば，暗号機能は，誰かに届けたい通信文を安全な通信路を使って正しく仲介するようなサービスとして記述され，署名機能は，ある利用者の文書を預かっておき，問合わせがあれば，その利用者の文書であることを証明するようなサービスとして記述される．一例として，図 1 に，参加者 P_i と P_j が，理想的に鍵配送を行う鍵配送機能 $\mathcal{F}_{\mathrm{KE}}$ を示す [7]．ここで，“`Establish-session`” は，理想機能に対する命令（コマンド）を意味し，sid は，各理想機能 $\mathcal{F}_{\mathrm{KE}}$ の識別子である．UC においては，各 $\mathcal{F}_{\mathrm{KE}}$ は，一つのサブルーチンコールに相当し，同一の機能でも，複数回（サブルーチンとして）利用されると，それぞれが別の識別子により認識される（つまり，複数の $\mathcal{F}_{\mathrm{KE}}$ が生成される）．

理想機能　$\mathcal{F}_{\mathrm{KE}}$

$\mathcal{F}_{\mathrm{KE}}$ は，セキュリティパラメータ k, 参加者 $P_1, \ldots, P_n$ および敵 S に対して以下の動作を行う．

（1） ある参加者 P_i から (Establish-session, sid, P_i, P_j) を受信すると，(sid, P_i, P_j) を保存するとともにこの対を S に送る．さらに，(sid, P_j, P_i) がすでに保存されている場合には以下の手続きを実行する．

（a） P_i, P_j がともに corrupt されていない場合には $\kappa \xleftarrow{R} \{0,1\}^k$ を求め，(key, sid, κ) を P_i および P_j に送り，(key, sid, P_i, P_j) を S に送って停止する．
（b） P_i もしくは P_j が corrupt されている場合には，(Choose-value, sid, P_i, P_j) を S に送る．そして S から値 κ を受け取ると，(key, sid, κ) を P_i と P_j に送って停止する．

（2） 敵 S から (corrupt, P_i) もしくは (corrupt, P_j) を受信すると以下の動作を行う．key が未送付（参加者へ通じる通信路にまだ書かれていない）の場合には S に key の値を送る．そうでない場合には S に何も送らない．

図 1　鍵交換の理想機能 $\mathcal{F}_{\mathrm{KE}}$
Fig. 1　The key exchange functionality $\mathcal{F}_{\mathrm{KE}}$.

参加者が corrupt 攻撃（後述）をされていない場合に，この理想機能を用いた世界では，敵 S は，P_i と P_j が共有した鍵 κ に関する情報を一切得ることができないことに注目しよう．

理想機能の定義において重要な点の一つとして，敵 S との情報のやり取りがある．理想機能 $\mathcal{F}$ が S と情報のやりとりをすることには以下の三つの目的がある．

（1）許される影響：敵が各参加者の実行結果に影響を与えたとしても安全である場合に，その状況を理想機能で実現できるようにするためである．例えば公開鍵暗号の理想機能においては，暗号化のアルゴリズムは敵 S が生成して理想機能に渡して理想機能はそのアルゴリズムを利用するように定義される [3]．これは，真に理想的な公開鍵暗号機能は暗号化のアルゴリズム（生成する暗号文）がどのようなものであってもよいために，敵が暗号化アルゴリズムを生成することを許している．

（2）許される情報漏えい：例えば現実のプロトコルで，暗号化された通信路を用いて P_i から P_j に通信を行えば通信内容を敵が知ることはできないが，通信が行われた事実やメッセージのサイズに関する情報を現実世界の敵は得ることができる．このような，避けられない情報漏えいを理想世界でも許すために，理想機能から敵 S への情報伝達が行われる．具体的には，現実世界のプロトコルにおける P_i と P_j 間での通信に相当する処理を，理想世界で $\mathcal{F}$ 経由で行う場合には，メッセージ送信の事実に関する情報（内容などは含まれないもの）を S に対して送信する．図 1 の $\mathcal{F}_{\mathrm{KE}}$ においては，Establish-session というコマンドは現実世界での P_i と P_j の間での交換を意味するので，この事実は敵 S に通知される．

（3）許される遅れ：UC の現実世界においては，通信路はすべて敵の制御のもとで動作する．すなわち，P_i から P_j に対してメッセージ送信が行われた場合に，その通信の遅れを敵が制御できる．このような避けられない遅れを理想世界でも許すために，理想機能の実行を遅らせることを目的とした敵からの入力を許す．具体的には，理想機能がある参加者に対して出力を行う場合に，敵 S からメッセージを受け取った後に出力を行うように理想機能の定義を行う．S はこのメッセージの送信を遅らせることにより，S が望むだけ出力を遅らせることができる．

ここで技術的に重要なことは，ある参加者が敵に攻撃されたときに理想機能の動作をどのように規定するかである．この規定により理想機能の保証する安全性のレベルが左右される．UC においては，敵は参加者を corrupt という形で攻撃することができる．これは CK 安全性の定義における参加者攻撃に相当する．理想世界においては，敵 S が理想機能 $\mathcal{F}$ に対して (corrupt, P) というコマンドを送信することができる．このコマンドを受信すると $\mathcal{F}$ は P が corrupted という状態であることを内部で保存するとともに，S に対してある種の情報を送る．

ここでどのような情報を S に送るかは，達成すべき安全性のレベルによって異なる．標準的 corruption モデルにおいては，$\mathcal{F}$ は corrupt された P との間で送受信したメッセージの系列をすべて S に与える．更に，corrupt 以降は，corrupt された参加者から $\mathcal{F}$ へ送信されたメッセージは S に転送され，S によって変更されたメッセージが $\mathcal{F}$ に届く．$\mathcal{F}$ から corrupt された

参加者へのメッセージについても，S に送られて変更されたものが参加者に送られる．これにより，corrupt された参加者の内部状態を得ること，並びにそれ以降の処理を S がコントロールすることを可能とする．

それに対して，forward-security モデルの場合には $\mathcal{F}$ はすべての送信メッセージではなく，参加者が削除できない情報のみを S に送る．

$\mathcal{F}_{\mathrm{KE}}$ の定義においては corrupt 時の処理は以下のように定義されている．鍵を共有する前にいずれかの参加者が corrupt された場合には，鍵の値は敵 S によって決定される，すなわち得られる鍵のランダム性は保障されず，しかも鍵の値は，参加者を corrupt している敵に漏えいすることを意味している．

また，$\mathcal{F}_{\mathrm{KE}}$ は forward-security モデルの定義であるため，鍵の値が決定した後に参加者を corrupt しても鍵の情報は S に送信しない．これは，鍵が決まった後直ちに，鍵共有のために使用した乱数などの内部状態を参加者がすべて削除する場合に，参加者を corrupt した敵が鍵に関する情報を得ることができない状況を表現している．

次に，理想機能 $\mathcal{F}$ を使って暗号機能を理想的に実現した世界（理想世界）と現実の暗号方式/プロトコル π を実行する世界（現実世界）を区別する識別器（「環境」と呼ぶ）を導入し，どのような（確率的多項式時間 Turing 機械である）環境 Z にとっても（もっと詳しく述べると，どのような現実世界の敵 A に対しても理想世界の敵 S が存在して，どのような環境 Z にとっても），理想世界と現実世界を区別できないとき，π は，理想機能 $\mathcal{F}$ を模倣できていると考える．つまり，暗号プロトコル π は $\mathcal{F}$ として記述された暗号機能を理想的な形で実現していると考えてよい．このことを，暗号プロトコル π は，理想機能 $\mathcal{F}$ を UC 実現すると呼ぶ．ここで，環境は，二つの世界を区別するために，各参加者と敵の入出力を対話的にコントロールすることができる．このことを図 2 に示そう．

この UC のフレームワークで最も重要な性質は名前の由来にもなっている，UC 定理（汎用的結合可能性定理）である．

この定理を説明する前に，ハイブリッドプロトコルの概念を説明しよう．まず，暗号プロトコル π は，理想機能 $\mathcal{F}$ をサブルーチンとして利用する現実のプロトコルであるとき，プロトコルを $\mathcal{F}$-ハイブリッドプロトコルと呼ぶ（ここで，π は，$\mathcal{F}$ を複数回同時に用いてもよい）．また，π が $\mathcal{F}$ をサブルーチンとして使う代わりに現実のプロトコル ρ をサブルーチンとして使うプロトコルを $\pi^{\rho/\mathcal{F}}$ と表記する．

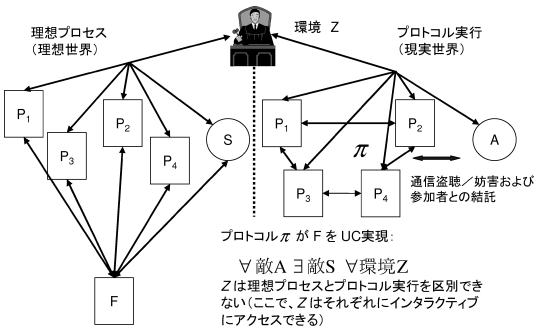

図 2　UC の定義
Fig. 2　Definition of UC.

［UC 定理］　暗号プロトコル π が，$\mathcal{F}$-ハイブリッドプロトコルであり，暗号プロトコル ρ が理想機能 $\mathcal{F}$ を UC 実現するならば，$\pi^{\rho/\mathcal{F}}$ は，$\mathcal{F}$-ハイブリッドプロトコル π を UC 実現する．

つまり，UC 定理が保証していることは，もし暗号プロトコル ρ が理想機能 $\mathcal{F}$ を UC 実現していることを示すことができれば，ρ をどのようなプロトコルのサブルーチンとして利用しても，ρ が $\mathcal{F}$ の機能を理想的に実現しているという性質が不変であるということである．

従来の安全性の定義ではこのような強い安全性は保証されていなかった．つまり，単独で使う場合には，目的とする機能（例えば，ゼロ知識証明）を満足していても，他のプロトコルと組み合わせて（若しくは，他のプロトコルの中で）使うと，本来の機能/安全性が保証されないことが起こり得たのである．例えば，二つのゼロ知識証明のプロトコルを並列に用いると，もはやゼロ知識証明ではない（知識を漏らす）ことが起こり得る．

このように UC 定理が成立する UC という方法論の意義は，以下の点にある．

（1）　プロトコル設計をモジュラー化することにより，複雑なプロトコル設計及び安全性証明を大幅に簡易化できる．

つまり，まず，実現したい理想機能 $\mathcal{F}$ を定め，よりシンプルなサブ機能 $\mathcal{F}_1, \ldots, \mathcal{F}_k$ に分割する．次に，各 $\mathcal{F}_1, \ldots, \mathcal{F}_k$ を UC 実現するプロトコル $\rho_1, \ldots, \rho_k$ を構成する．更に，$\mathcal{F}$ を UC 実現する $(\mathcal{F}_1, \ldots, \mathcal{F}_k)$-ハイブリッドプロトコル π を構成する．最後に，UC 定理に基づき $\mathcal{F}$ を UC 実現する現実プロトコル $\pi^{\rho_1/\mathcal{F}_1,\ldots,\rho_k/\mathcal{F}_k}$

SIG-DH プロトコル

初期情報：素数 $p, q(\ q|p-1$ を満足)，$g \in Z_p^*$（次数 q）．

（1） 参加者 P_i が (P_i, P_j, sid) を入力されると，メッセージ $(\texttt{signer},\ 0||sid)$ を $\mathcal{F}_{\rm SIG}$ に送る．参加者 P_j が (P_j, P_i, sid) を入力されると $(\texttt{signer},\ 1||sid)$ を $\mathcal{F}_{\rm SIG}$ に送る．

（2） P_i が $x \stackrel{R}{\leftarrow} Z_q$ を選び，$\alpha \leftarrow g^x$ を求め，(P_i, sid, α) を P_j に送る．

（3） (P_i, sid, α) を受信した P_j は $y \stackrel{R}{\leftarrow} Z_q$ を選び，$\beta \leftarrow g^y$ を求め，$(\texttt{sign},\ 1||sid, (sid, \beta, \alpha, P_i))$ を $\mathcal{F}_{\rm SIG}$ に送り，署名 σ_j を得る．P_j は (sid, β, σ_j) を P_i に送り，セッション鍵 $\gamma \leftarrow \alpha^y$ を求めて y を消去する．

（4） (sid, β, σ_j) を受信した P_i は $(\texttt{verify},\ 1||sid, P_j, (sid, \beta, \alpha, P_i), \sigma_j)$ を $\mathcal{F}_{\rm SIG}$ に送る．署名検証に成功した（$\mathcal{F}_{\rm SIG}$ が 1 を返してきた）場合には P_i は $(\texttt{sign},\ 0||sid, (sid, \alpha, \beta, P_j))$ を $\mathcal{F}_{\rm SIG}$ に送り，署名 σ_i を得る．P_i は (sid, σ_i) を P_j に送り，セッション鍵 $\gamma' \leftarrow \beta^x$ を求めて x を消去し，(sid, P_i, P_j, γ') を出力する．

（5） (sid, σ_i) を受信した P_j は $(\texttt{verify},\ 0||sid, P_j, (sid, \alpha, \beta, P_j), \sigma_i)$ を $\mathcal{F}_{\rm SIG}$ に送る．署名検証に成功すると，P_j は (sid, P_j, P_i, γ) を出力する．

図 3 鍵交換機能の $\mathcal{F}_{\rm SIG}$-ハイブリッドプロトコル
Fig. 3 A key exchange protocol by $\mathcal{F}_{\rm SIG}$-hybrid model.

を構成する．

（2） プロトコル ρ が理想機能 $\mathcal{F}$ を UC 実現すると仮定すると，ρ をどのような形で複数回組み合わせて利用しても，その安全性が保証される．

さて，ここではその詳細は省略するが，文献 [3] で規定された理想機能による UC 安全性は，ゲームベースの CK 安全性よりも高い安全性を保証することが示されている．

なお，ここで注意する必要がある点は，一つの理想機能 $\mathcal{F}_{\rm KE}$ は一つの sid にのみ対応しており，複数の sid にまたがった安全性は UC 定理で保証しているという点である．つまり，CK 安全性が複数のセッションを対象とした定式化であったことに対して，UC 安全性がそれとは顕著に異なる方法論で定式化していること，つまり「単一 sid の UC 安全性＋ UC 定理」により複数セッションの UC 安全性を定式化していることに注意されたい．

さて，前に HMQV 鍵交換方式が CK 安全性を満足することを述べたが，この UC 安全性を満足するだろうか．答は，否である．CK 安全性と UC 安全性の間にギャップがあるように具体的な鍵交換方式においても HMQV 鍵交換方式は CK 安全であるが UC 安全ではない．それでは，UC 安全性を満足する方式はどのような方式であろうか．それは，IEEE などで標準化されている ISO 9798-3 方式である．署名の理想機能 $\mathcal{F}_{\rm SIG}$ を利用する形式で記述したこの方式の $\mathcal{F}_{\rm SIG}$-ハイブリッドプロトコルを図 3 に示す [7]．

ここで，$\mathcal{F}_{\rm SIG}$ に対するコマンドの概要は以下のとおりである．

- $(\texttt{signer},\ sid)$：このメッセージの送信者 P_i が識別子 sid の署名機能の署名者であることを登録（新たな $\mathcal{F}_{\rm SIG}$ のインスタンスの生成）．
- $(\texttt{sign},\ sid, m)$：メッセージ m に対する署名の要求．ただし，ここで sign コマンドが受け付けられるのは signer コマンドで登録した署名者に限る．$\mathcal{F}_{\rm SIG}$ は署名 σ_i を返す．
- $(\texttt{verify},\ sid, P_i, m, \sigma_i)$：署名検証の要求．$\sigma_i$ が m に対する P_i の正しい署名であれば $\mathcal{F}_{\rm SIG}$ は 1 を返し，そうでなければ 0 を返す．

参加者が corrupt された場合も考慮した $\mathcal{F}_{\rm SIG}$ の完全な定義は [3] を参照のこと．なお，上記プロトコルで識別子 $0||sid$，$1||sid$ の意味は以下のとおりである．$\mathcal{F}_{\rm SIG}$ のインスタンスを二つ（P_i，P_j がそれぞれ署名者となるもの）生成する必要があるが，それらが他の $\mathcal{F}_{\rm SIG}$ のインスタンスと異なる識別子をもつようにするため，親のルーチンの識別子に 0，1 をそれぞれ結合したものを識別子として用いている．

4.2 UC に関する結果

上で述べたように，UC は，非常に強い意味の安全性を保証する．それでは，そのような UC 安全性をもつ方式をどのように構成すればよいであろうか．また，そのように強い安全性をもつ方式は現実に存在するのであろうか．その答は，ある意味で否定的であるが，別の意味では肯定的である．

（1） UC 公開鍵暗号の条件は，従来の標準的な安全性（選択暗号文攻撃に対して強秘匿/頑健）と同じである．また，UC 署名の条件も，従来の標準的な安全性（選択文書攻撃に対する存在的偽造不可）と同じ

である.（肯定的結果）

（2） 正しく動作する者が過半数の場合は，どのような多者暗号プロトコル機能も UC 実現可能（従来の結果がそのまま使える）.（肯定的結果）

（3） 正しく動作する者が過半数でない場合は，既存のいかなる多者暗号プロトコルも，特に，多くの応用のある「2 者暗号プロトコル」も UC 実現できない.例えば，多くの興味深い機能（コミットメント，ゼロ知識証明，コイン投げ等）が標準的モデル（プロトコルを実行する前に，いかなる事前処理も想定しないモデル）で UC 実現できない.（否定的結果）

（4） 共通参照情報（CRS：Common Reference String）モデル（プロトコルを実行する前に，ある仕様に基づく公開情報が第三者によって正しく作られ公開されることを想定するモデル）では任意の 2 者暗号プロトコル及び一般の多者暗号プロトコルが UC 実現可能である．つまり，共通参照情報モデルでは，UC コミットメントや UC ゼロ知識証明，UC 多者暗号プロトコルが実現可能である.（条件付肯定的結果）

5. 数理的技法により計算論的安全性を証明する可能性

上で示したように，計算論的アプローチでは，確率的多項式時間 Turing 機械（PPT）を敵のモデルとしてとらえ，いかなる PPT に対しても安全であることを示す．暗号（プロトコル）の安全性定義として，暗号分野では広く受け入れられているが，一般にその安全性証明は複雑で，間違った証明も多く見られる．例えば，上述の HMQV 鍵交換方式が CK 安全であることの証明も，ISO 9798-3 方式が UC 安全であることの証明も，決して簡単なものではない．暗号プロトコルによっては，間違った証明も多く見られる．そこで，それらの証明を簡明化/（部分）自動化する手段として，数理的アプローチを用いることは可能であろうか.

数理的（Formal methods）アプローチでは，対象とする暗号（プロトコル）を記号列で表現し，その記号列に対する論理的推論/書換規則などにより，安全性を示す．その証明は明確で,（部分的）自動化も可能である．しかし，従来は，その証明が保証する安全性の意義が必ずしも明確でなかった.

この問題を解決するには，数理的技法による証明が計算論的安全性に対して「健全性」を満足することが必要となる.（ここでの「健全性」とは，数理的技法による証明が計算論的安全性を保証することを意味する.）

さて，UC における安全性証明は，大きく二つに分類される．一つは，計算論的な仮定のみ（例えば，素因数分解が難しい）を用いてある方式が UC の意味で安全であること（例えば，公開鍵暗号やディジタル署名が UC 安全であること）を証明することである（下位レベルの証明若しくは直接的証明）．もう一つは，UC のある理想機能の存在を前提として，それらを組み合わせたプロトコル（UC におけるハイブリッドプロトコル）が UC 安全であることを証明することである（上位レベルの証明若しくは間接的証明）.

したがって，数理的技法を UC のような計算論的安全性の証明に適用する場合，上位レベルの安全性証明（UC のハイブリッドモデルの安全性証明）に数理的アプローチを適用する場合と，下位レベルの安全性証明（計算論的な仮定のみから，ある方式の安全性を証明）に数理的アプローチを適用する場合が考えられる．この下位レベルの安全性証明に適用する場合においては，UC のようなシミュレーションベースの定式化に対して数理的アプローチを適用する場合と，伝統的な攻撃ベースの定式化に対して数理的アプローチを適用する場合がある.

5.1 上位レベルの安全性証明に数理的アプローチを適用

5.1.1 初の健全性結果：Abadi-Rogaway

Abadi と Rogaway は，共通鍵暗号を用いた暗号プロトコルに対して，数理的技法による安全性が計算論的安全性を保証することを示した（つまり，数理的技法の計算的安全性への健全性を示した）[1] .

彼らは,（理想的）暗号機能を意味する記号 $\{M\}_K$ を導入するという Dolev-Yao 流の数理的技法を用いることで，計算論的安全性における強秘匿（IND）（と同様な）安全性をもつ暗号機能を利用した暗号プロトコルの計算論的安全性を数理的技法により証明できることを示した.

この強秘匿と同様な暗号機能を理想機能ととらえて UC の枠組みの中で考えれば，先程述べた上位レベル（ハイブリッドプロトコル）で数理的技法を適用した例とも考えられ，彼らの結果は，以下で述べる Canetti と Herzog の結果と極めて類似する.

5.1.2 数理的技法による UC 安全性（上位）：Canetti-Herzog

Canetti と Herzog は，Dolev-Yao 流の数理的アプローチが（ハイブリッドプロトコルでの）UC 安全性を

保証することを示した．また，既存の検査ツールを使って具体的鍵交換プロトコルの UC 安全性を示した [8]．

彼らは，（理想的）公開鍵暗号機能を意味する記号 $\{M\}_{PK}$ を導入するという Dolev-Yao 流の数理的技法を用いることで，公開鍵暗号を用いた鍵交換システムの計算論的安全性（ハイブリッドプロトコルでの UC 安全性）を数理的技法により証明できることを示した．

ここで，UC における公開鍵暗号の理想機能 $\mathcal{F}_{\mathrm{PKE}}$ を，数理的アプローチにおける記号 $\{M\}_{PK}$ に対応づけることで，上位レベル（ハイブリッドプロトコル）における計算論的安全性（UC 安全性）と数理的技法が自然に関係づけられる．

5.2 下位レベルの安全性証明に数理的アプローチを適用

5.2.1 数理的技法による UC 安全性（下位）：Canetti-Cheung-Kaynar-Liskov-Lynch-Pereira-Segala

Canetti らは，数理的技法において，記号列に対する書換え規則などの推論の代わりに確率的 I/O オートマトン（PIOA）のモデルを導入し，PIOA により表現した安全性（数理的技法による安全性）が（弱いモデルでの）UC 安全性（計算論的安全性）を保証することを（Oblivious Transfer という暗号プロトコルの一実現例について）示した [5]．

彼らの証明では，（ハードコア属性という）計算論的な仮定に基づき Oblivious Transfer の UC 安全性を示しており，上述した下位レベルの証明の例となっている．彼らが用いた証明手法では，シミュレーションベース定式化（UC の枠組み）における（ある種の）ゲーム列的な手法が採用されており，この後に述べる Blanchet と Pointcheval の手法とも通じるものがある．

5.2.2 ゲーム列による証明の形式化/自動化：Blanchet-Pointcheval

Blanchet と Pointcheval は，計算論的アプローチにおける攻撃ベース定式化で，最近流行しているゲーム列で安全性を証明する手法を数理的アプローチに適用している [2]．彼らは，ゲーム列におけるゲーム変換の法則を抽出し，それをプロセス計算に適用することでゲーム列を自動生成する手法を開発し，（攻撃ベース定式化による）計算論的安全性を数理的技法で（自動）証明することに成功した．

彼らの手法も，基本的な理想機能を仮定することなく，計算量的な仮定だけに基づき安全性を証明するという意味で，上述した下位レベルの証明の例となっている．

6. むすび

この分野の研究は始まったばかりで，今後の課題は山積している．上で述べたように，UC やゲーム列による証明手法といった最近進展の著しい計算論的手法の観点でとらえると，数理的技法との関係がより明確になると思われる．両アプローチが融合することで，暗号安全性の証明が簡明化し，証明手法に対する研究が進展することを期待したい．なお，この分野の研究では，暗号と数理的証明技法という異なる研究分野の研究者の交流が重要となる（文献 [1], [2], [5], [8] のいずれもそのような共同研究の結果である）．この分野の研究が発表される主な国際会議として Crypto，Eurocrypt，TCC（Theory of Cryptography Conference），IEEE CSF（Computer Security Foundation），またこの分野に特化したワークショップとして FCC（Formal and Computational Cryptography）があり，更に IEEE S&P（Symp. on Security and Privacy），ACM CCS（Conf. on Computer and Communications Security）のような会議でも重要成果がしばしば発表される．

国内でもこの研究分野の活性化を目的とした研究集会が開かれており，両分野の研究者の交流，研究協力の場になることが期待されている [11]．この分野の研究に関する解説論文は [12] を参照されたい．

謝辞 本論文に対して多くの有用なコメントを下さった編集委員の皆様に感謝致します．

文　献

[1] M. Abadi and P. Rogaway, “Reconciling two views of cryptography (the computational soundness of formal encryption),” J. Cryptol., vol.15, no.2, pp.103–127, 2002.

[2] B. Blanchet and D. Pointcheval, “Automated security proofs with sequences of games,” CRYPTO 2006, LNCS vol.4117, pp.537–554, 2006.

[3] R. Canetti, “Universal composable security: A new paradigm for cryptographic protocols,” FOCS’01, pp.136–145, 2001. The full paper version, IACR Cryptology ePrint Archive, http://www.iacr.org/2000/067

[4] R. Canetti, “Security and composition of cryptographic protocols: A tutorial,” SIGACT News, vol.37, nos. 3 & 4, 2006, http://eprint.iacr.org/2006/465

[5] R. Canetti, L. Cheung, D. Kaynar, M. Liskov, N. Lynch, O. Pereira, and R. Segala, “Using task-structured probabilistic I/O automata to analyze cryptographic protocols,” FCC, pp.34–39, 2006.
[6] R. Canetti and H. Krawczyk, “Analysis of key-exchange protocols and their use for building secure channels,” Eurocrypt 2001, LNCS vol.2045, http://eprint.iacr.org/2001/040
[7] R. Canetti and H. Krawczyk, “Universally composable notions of key exchange and secure channels,” Eurocrypt 2002, LNCS vol.2332, pp.337–351, http://eprint.iacr.org/2002/059
[8] R. Canetti and J. Herzog, “Universally composable symbolic analysis of mutual authentication and key-exchange protocols,” TCC 2006, LNCS, vol.3876, pp.380–403, 2006. the full paper version, http://eprint.iacr.org/2004/334
[9] W. Diffie and M. Hellman, “New directions in cryptography,” IEEE Trans. Inf. Theory, vol.IT-22, no.6, pp.644–654, Nov. 1976.
[10] H. Krawczyk, “HMQV: A high-performance secure Diffie-Hellman protocol,” CRYPTO 2005, LNCS vol.3621, pp.546–566, http://eprint.iacr.org/2005/176/
[11] 日本応用数理学会「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/
[12] 特集「数理的技法による情報セキュリティ」，応用数理，vol.17, no.4, pp.6–58, 2007.

（平成 20 年 8 月 25 日受付，11 月 20 日再受付）

岡本　龍明　（正員：フェロー）

1976 東大・工・計数卒．1978 同大大学院修士課程了．同年，日本電信電話公社に入社．1989〜1990 カナダ Waterloo 大客員助教授，1994〜1995 米国 AT&T Bell Laboratories 客員研究員．本会業績賞，小林記念特別賞，電気通信普及財団賞，科学技術庁長官賞，日経 BP 賞各受賞．著書：「暗号・ゼロ知識証明・数論」（共立出版），「現代暗号」（産業図書），「暗号と情報セキュリティ」（日経 BP 社）など．現在，NTT 情報流通プラットフォーム研究所岡本特別研究室長，NTT フェロー，京大大学院情報学研究科客員教授．2007 年度日本応用数理学会会長．工博．現在，暗号理論の研究に従事．

真鍋　義文　（正員）

1983 阪大・基礎工・情報卒．1985 同大大学院修士課程了．同年，日本電信電話（株）入社．1994〜1995 米国 Johns Hopkins 大客員研究員．現在，NTT コミュニケーション科学基礎研究所協創情報研究部情報基礎理論研究グループリーダ，京大大学院情報学研究科客員准教授．博士（工学）．分散アルゴリズム，暗号理論，グラフ理論に興味をもつ．