情報漏洩や改竄に耐性のある暗号技術に関する研究

研究代表者	安永　憲司	金沢大学　理工研究域　助教
共同研究者	田中　圭介	東京工業大学　情報理工学研究科　准教授
共同研究者	吉田　真紀	大阪大学　情報科学研究科　助教

1 研究の背景

インターネットをはじめとする情報通信システムの発達により、多くの人々が複雑に入り組んだシステムを利用している。その中で、個人情報の秘匿や認証を行うための基礎技術として暗号技術は欠かせないものである。システムの複雑化により、実装および運用上の不備、予期せぬ動作、または悪意のある動作によってもたらされる情報漏洩は、避けられない問題である。特に、サイドチャネル攻撃と呼ばれる攻撃は、計算処理のタイミングや消費電力等の情報を利用して秘密情報を抜き出すものであり、完全に防ぐことが難しい攻撃である。既存の多くの暗号技術は、秘密情報が漏洩することを想定しておらず、ほんのわずかな漏洩によってシステム全体が破綻する恐れがある。

また、暗号理論は、情報セキュリティ技術の基礎理論として発展してきている。特に、現代的な暗号理論では、安全性は証明するものという考えが広く浸透してきている。つまり、安全性の定義や必要とする数学的仮定を正確に述べ、その数学的仮定のもとで安全性を達成することを証明することを目指している。このような枠組みでは、数学的仮定が正しい限りその暗号技術は必ず安全であることを保証できる。

一方で、この証明可能安全性という枠組みは万能というわけではなく、定義した安全性およびその安全性モデルの妥当性については議論を続ける必要がある。例えば、秘密鍵に関する情報は第三者には知られないということを通常仮定しているが、もしその秘密鍵に関する情報が敵対者に漏洩してしまった場合にどのような安全性が達成されるかについては、通常の安全性モデルでは何も示すことができない。さらに、このような点に関して、暗号理論研究の立場として、漏洩が起きてしまうのはそのようなシステムを実装する側の問題であり､その基盤理論である暗号理論はその点を考慮する必要がないとの考えがあったように思われる。

そのような状況に対して、情報漏洩を安全性モデルに取り込み、適切なモデルのもとで暗号技術が情報漏洩に耐性があることを証明するという研究が近年行われるようになってきた。本研究においても、適切なモデル化のもと、暗号技術が情報漏洩に耐性があることを証明することを考えていく。

2 情報漏洩に耐性のある公開鍵暗号

2-1 情報漏洩を考慮した安全性モデル

公開鍵暗号において、受信者の秘匿情報である秘密鍵が敵対者に漏洩した場合の安全性に関する研究は Akavia ら[1]によって始められた。彼女らは、漏洩を考慮した公開鍵暗号の安全性モデルを提案し、いくつかの既存方式がそのモデルにおいて安全であることを考察した。より具体的には、公開鍵暗号における安全性ゲームを考える際に、敵対者が漏洩オラクルにアクセスすることを許すモデルを考えた。敵対者は、任意の関数 f の記述をオラクルに提出すれば、秘密情報 s に関する部分情報として f(s)をオラクルから手に入れることができる。特に、秘密情報 s として受信者がもつ秘密鍵 sk を考えた。このような状況のもとで安全性の達成可能性に関する議論を行った。安全性ゲームは以下のとおりである。

暗号方式 $\Pi=(\mathrm{Gen},\mathrm{Enc},\mathrm{Dec})$ に対する敵対者 A との安全性ゲーム

1. $(\mathrm{pk},\mathrm{sk}) \leftarrow \mathrm{Gen}(1^k)$
2. $(m_0,m_1) \leftarrow A^{\mathrm{PreLeak}(\mathrm{sk})}(\mathrm{pk})$
3. $c \leftarrow \mathrm{Enc}(\mathrm{pk},m_b)$ ただし $b\in\{0,1\}$ はランダムに選択
4. $b' \leftarrow A^{\mathrm{PostLeak}(\mathrm{sk})}(c)$
5. $b = b'$ なら 1 を出力．それ以外は 0 を出力．

漏洩関数 PreLeak(・)および PostLeak(・)の記述は、敵対者がその関数を計算する回路を提出することで実現すると考えるため、敵対者の計算時間が多項式時間計算に制限されることから、回路のサイズも多項式サイズに制限され、結果として漏洩関数も多項式時間で計算可能な関数に制限されている。

敵対者 A が PreLeak(sk)および PostLeak(sk)というオラクルアクセスをしない場合が、通常の安全性ゲームである。暗号方式 Π は、任意の多項式時間敵対者 A に対して上記のゲームの出力の期待値が 1/2 程度であるときに安全であると言う。

上記の安全性において、漏洩関数 f に制限がない場合、安全性を達成することは不可能である。なぜならば、f として恒等関数を指定すれば、敵対者は秘密鍵 sk を入手することができ、上記のゲームで b を簡単に当てることができるからである。

そこで、漏洩関数 f に関して何らかの制限が必要となるが、Akavia らは漏洩する情報量の制限を考えた。つまり、f(sk)の出力長が予め制限されているモデルである。その長さを例えば m ビットとしよう。すると、出力長が制限されているとはいえ、任意の m ビット出力の計算が可能なモデルを表しているのである。このモデルは非常に強力な漏洩モデルであるといえる。また、このように、秘密鍵という記憶媒体に保存してある情報が漏洩するという状況は、Halderman ら[2]によって行われた研究を理論的にモデル化したものと言える。

上記の安全性ゲームでは、PreLeak(・)と PostLeak(・)という２つの漏洩関数が登場するが、敵対者が PostLeak(・)を利用できる場合、安全性は達成されないことがわかる。なぜならば、PostLeak(・)として、

PostLeak(sk) := { Dec(sk,c)の i ビット目. ただし i は m_0 と m_1 で値が異なる位置. }

という関数を指定すれば、c の平文が m_0 なのか m_1 なのが明らかになるため、敵対者は b を正しく推測できるのである。

以上の考察より、Akavia らは、秘密鍵の漏洩に耐性のある公開鍵暗号方式の安全性として、1.漏洩する情報量の上限が決まっており、2.情報の漏洩は PreLeak(sk)だけで起きる、というモデルを提案した。

2-2 漏洩耐性のある既存の公開鍵暗号

Akavia ら[1]は、いくつかの既存の暗号方式が、秘密鍵の漏洩に耐性があることを示した。

Naor と Segev[3]は、漏洩関数 PreLeak(・)が鍵生成の前に予め定まるような場合に、安全な方式の一般的構成法を示した。つまり、PreLeak(・)として漏洩オラクルにクエリする関数として、公開鍵 pk の情報に依存して計算されるものではなく、公開鍵の情報を使わずに計算するものに制限した場合である。この場合、強乱数抽出器 Ext を使い、鍵生成アルゴリズムでは、乱数として Ext(x,s)を利用することを考える。ここで、s は Ext の種であり、x は入力乱数であり秘密鍵として保持しておく。そして種 s を公開鍵とするのである。すると、秘密鍵 x に関する任意の漏洩があったとしても、x のエントロピーが残されている限り、Ext(x,s)は一様ランダムな乱数である。種 s を公開しているため、公開鍵 s に依存した漏洩が起きると、Ext(x,s)が一様ランダムになるとは限らないため、安全性が証明できない。

さらに、Naor と Segev は、Hash Proof System と呼ばれるシステムから一般的に、秘密鍵漏洩に対して安全である方式を提案した。この方式では漏洩関数 PreLeak(・)は公開鍵 pk に依存していたとしても安全性が保証される。

3 漏洩に耐性のある KEM/DEM 方式

漏洩に耐性のある公開鍵暗号方式についての研究としては、受信者の秘密鍵の漏洩に耐性のある方式に関する研究が盛んに行われていた。しかし、公開鍵暗号における秘密情報は、受信者の秘密鍵だけではない。送信者が暗号化時に生成する乱数も秘密情報といえる。公開鍵暗号の基本的な安全性である CPA 安全性を達成するには乱数の生成が不可欠であることが知られており、その乱数は生成した後は外部に漏洩しないことを通常は仮定しているからである。その乱数が敵対者に漏洩した場合に安全性がどのようになるかは一般的に不明である。

そこで本研究では、送信者の乱数情報の漏洩を考慮した公開鍵暗号方式に着目した。漏洩のモデルは、先行研究である Akavia ら[1]および Naor と Segev[3]と同じ、漏洩するビット長の上限だけが定まっているモデルを考えた。つまり、既存のモデルにおいて秘密鍵 sk が漏洩していた部分を暗号化用の乱数 r が漏洩する

形に変更した安全性ゲームを考える。

3-1 乱数漏洩に関する考察

乱数漏洩モデルにおける最初の考察として、PreLeak(r)が公開鍵 pk に依存して作られる場合に、安全な方式が存在しないことがわかった。具体的には、以下の漏洩関数を考える。

PreLeak(r) := { Enc(pk,m_0;r)のランダムに選んだ i ビット目 }

ここで、m_0 と m_1 は異なる平文であり、暗号方式が正当性を満たしているとすると、Enc(pk,m_0;r)と Enc(pk,m_1;r)は少なくとも 1 ビットは異なる値をとることがわかる。もしそのようなビット位置の値を PreLeak(r)により入手することが出来たとすると、その漏洩情報から与えられた暗号文 c の平文が m_0 であるか m_1 であるかを無視できない確率で区別することができる。したがって、どのような暗号方式に対しても攻撃が存在してしまうのである。

上記の考察より、秘密情報漏洩に関して、公開鍵情報に依存した漏洩については、秘密鍵の漏洩よりも乱数情報の漏洩の方が脅威であることが分かった。漏洩のタイミングによる安全性の達成可能性に関する比較をまとめると以下のようになる。

漏洩関数生成のタイミング	秘密鍵漏洩	乱数漏洩
公開鍵生成前	CPA 安全な方式 [3]	CPA 安全な方式 (本研究)
公開鍵生成後	Hash Proof System [3]	不可能 (本研究)
暗号文受信後	不可能 [1]	不可能 (本研究)

3-2 KEM/DEM 方式における乱数漏洩

KEM/DEM 方式は、Key Encapsulation Mechanism(KEM)および Data Encapsulation Mechanism(DEM)の略であり、公開鍵暗号方式を実現するためのフレームワークの一つである。KEM はランダムな鍵を共有するための方式であり、DEM はその鍵を使って秘密鍵暗号を実現する方式である。多くの効率的な公開鍵暗号方式が KEM/DEM フレームワークによって実現されている。

3-1 節で行った考察から、乱数漏洩に関しては公開鍵生成後の漏洩に耐性をもたせることができないことが分かった。しかし、漏洩関数として任意の計算を許しているため、攻撃者にはやや有利なモデルにおける不可能性であるため、漏洩関数をある程度制限して安全な方式を提案することには重要であると考える。

そこで、漏洩のモデルとして、KEM/DEM 方式に対して自然な形で制限を加えた漏洩モデルを考える。具体的には、KEM 方式の乱数漏洩と DEM 方式の乱数漏洩が独立に行われるというモデルを考える。さらに、DEM 方式の漏洩に関しては、敵対者が平文を選択した後に行われると仮定する。形式的な安全性の定義は以下のとおりである。

KEM/DEM 方式 Π=(K.Gen,K.Enc,K.Dec,D.Enc,D.Dec)に対する敵対者 A との安全性ゲーム

1. (pk,sk) ← K.Gen(1^k)
2. (c,K) ← K.Enc(pk;r)
3. (m_0,m_1) ← $A^{KemLeak(r)}$(pk)
4. d ← D.Enc(m_b,K;r') ただし b∈{0,1} はランダムに選択
5. b' ← $A^{DemLeak(r')}$(c,d)
6. b = b' なら 1 を出力. それ以外は 0 を出力.

任意の多項式時間敵対者 A に対して上記の安全性ゲームを行ったとしても出力の平均値が 1/2 程度であるとき、方式Πは安全であるという。

上記の安全性ゲームにおいて、KEM 方式の乱数漏洩は敵対者が平文を生成する際に起きるが、DEM 方式の乱数漏洩は平文を生成した後にしか起きないことに注意したい。KEM/DEM 方式では、KEM 暗号文には共有する秘密鍵が暗号化され、DEM 暗号文には平文が暗号化されている。そのため、送信すべき平文が決まる前に KEM 暗号文は事前に生成されている可能性がある。一方で DEM 暗号文は平文が決まる前に生成されることはない。

上記の安全性ゲームにおける漏洩の違いは、このような状況を捉えたものである。

3-3 乱数漏洩に耐性のある KEM/DEM 方式の構成

3-2 節で定義した安全性を満たす KEM/DEM 方式の一般的な構成法を与える。具体的には、エントロピー安全性をもつ KEM 方式から一般的に構成できることを示す。

エントロピー安全な KEM 方式とは、KEM 方式において乱数情報が漏洩したとしても、共有秘密鍵に計算量的なエントロピーがあることが保証できる方式のことである。より具体的には、ある公開鍵分布 PK^*が存在し、1. PK^*の分布は実際の公開鍵分布と計算量的に識別できず、2.PK^*から公開鍵 pk^*が選ばれたとき、敵対者が公開鍵 pk^*,暗号文 c^*,漏洩情報 $f(pk^*,r)$を手に入れたとしても、共有鍵 K'にエントロピーが残っているときにエントロピー安全であるという。形式的な定義は以下のとおりである。

KEM 方式 $\Pi=(K.Gen,K.Enc,K.Dec)$がλ-乱数漏洩に対して(κ, ε)-エントロピー安全であるとは、
1.効率的にサンプル可能な分布 PK^*が存在し、分布$\{pk:(pk,sk)\leftarrow Gen(1^n)\}$と計算量的に識別不可能
2.ある分布 K'が存在し、

$$H^{\infty}(K'|pk^*,c^*,f(pk^*,r)) \geq \kappa \quad \text{かつ}$$

$$\Delta((pk^*,c^*,K^*,f(pk^*,r)),\ (pk^*,c^*,K',f(pk^*,r))) \leq \varepsilon$$

を満たすときである。ここで、$pk^*\leftarrow PK^*$, $(c^*,K^*)\leftarrow K.Enc(pk^*;r)$であり、f は出力長$\lambda$以下の効率的に計算可能な任意の関数である。確率変数 A,B に対し、$H^{\infty}(A)$は A の最小エントロピー、$\Delta(A,B)$は A と B の統計的距離を表している。

エントロピー安全な KEM 方式をもとにして、乱数漏洩に耐性のある KEM/DEM 方式を構成する。エントロピー安全な KEM 方式$\Pi=(K.Gen,K.Enc,K.Dec)$と強乱数抽出器 Ext を使って構成する。具体的な構成法は以下のとおりである。

鍵生成: K.Gen と同様
KEM 暗号化: K.Enc と同様
DEM 暗号化: 秘密鍵 K と平文 m に対して、乱数 r'を選択し、d=(Ext(K,r')+m,r')を出力
KEM 復号: K.Dec と同様
DEM 復号: 秘密鍵 K と暗号文 d=(c,r')に対して、m'=Ext(K,r')+c を出力

適切なパラメータの設定下において、上記の方式は乱数漏洩に耐性のある KEM/DEM 方式である。安全であることの直観的な説明を行う。まず KEM 暗号文の乱数漏洩に関して、KEM 方式がエントロピー安全であることから、安全性ゲームの公開鍵を PK^*からのサンプルに変えても識別ができない。このとき、秘密鍵 K^*は鍵 K'と統計的距離が近く、K'はエントロピーをもつことが保証されている。そのため、Ext(K^*,r')は敵対者にとっては一様乱数のように見える。Ext(K^*,r')は平文 m を隠すためのマスクとして使われているため、平文 m は安全に暗号化されていることがわかる。また、DEM 暗号文の乱数漏洩に関して、漏洩するのは Ext への入力である r'であり、Ext は強乱数抽出器であることから、r'は K^*と独立に選ばれた一様乱数である限り、r'については全ビットが漏洩したとしても問題がない。DEM の乱数漏洩は暗号文が生成されてから行われるため、この段階では r'の前ビットが漏洩したとしても安全性には影響しない。以上より、KEM および DEM の乱数が漏洩したとしても、上記の方式は安全であることが分かる。

3-4 エントロピー安全性をもつ KEM 方式の構成

3-3 節において定義したエントロピー安全性をもつ KEM 方式の一般的な構成方法を与える。具体的には、Hash Proof System から一般的に構成できることを示す。Naor と Segev[3]は Hash Proof System から秘密鍵漏洩に対して安全な公開鍵暗号方式の一般的な構成法を示した。本節では、Naor と Segev の構成法をもとに、エントロピー安全な KEM 方式の構成法を与える。

構成のアイディアとしては、Naor と Segev の構成法における「秘密鍵」と「乱数」の役割を交換することで、秘密鍵漏洩に対して安全な方式を乱数漏洩に対して安全な方式にすることを考える。より具体的に述べると、Naor-Segev 方式における KEM 暗号化アルゴリズムを鍵生成アルゴリズムとして用い、Naor-Segev 方式

の鍵生成アルゴリズムを KEM 暗号化アルゴリズムとして用いる。つまり、Naor-Segev 方式の KEM 暗号文を公開鍵とし、Naor-Segev 方式の公開鍵を KEM 暗号文として利用するのである。このような役割の交換は、一般的な KEM 方式に対しては適用できない。その理由は、(1)暗号文は公開鍵に依存して生成される可能性があるため、KEM 暗号化を鍵生成の前に行うことができない可能性があり、(2)暗号文である公開鍵と KEM 暗号文生成時の秘密情報から共有鍵を生成する必要があるが、その鍵は一般的に、秘密鍵と KEM 暗号文から復号される共有鍵に一致するとは限らないからである。しかし、Hash Proof System ベースの KEM 方式は、この２つの問題点を回避できることが分かる。

まず、Hash Proof System を簡単に説明する。Hash Proof System を KEM 方式として見た場合、暗号文の生成に２つのモードがある。一つ目のモードでは正当な暗号文が生成され、その暗号文からは秘密鍵を利用して共有鍵を手に入れることができる。また、正当な暗号文を生成する際に選択する乱数を暗号文が正当であることの証拠と考える。もう一つのモードでは、不当な暗号文が生成され、この暗号文は共有鍵に関する情報を実質的に含んでいない。そして、この２つのモードは計算量的に識別ができないのである。暗号文から共有鍵を計算するのに Hash 関数を利用する。この Hash 関数 Λ に対しては、ある射影 μ が存在し、射影値が同じ秘密鍵に対しては、正当な暗号文に対する Λ によるハッシュ値が同じになる。また、不当な暗号文に対する Λ によるハッシュ値は、情報理論的に共有鍵 K に関する情報を含んでいない。

形式的には以下のように定義される。

Hash Proof System は HPS=(Param,Pub,Priv)から構成される。
Param: 入力 1^n に対して、(SK,PK,C,V,K, Λ, μ)の記述を出力する。
SK,PK,C,V,K は秘密鍵空間、公開鍵空間、暗号文空間、正当な暗号文空間、鍵空間を表す。
Pub: 入力 pk=μ(sk)と正当な暗号文 c∈V およびその証拠 w に対して、共有鍵 K=Λ_{sk}(c)を出力する。
Priv: 入力 sk と暗号文 c に対して、共有鍵 K=Λ_{sk}(c)を出力する。

計算量的な仮定として、部分集合所属問題が困難であることを仮定する。これは、任意の多項式時間敵対者 A に対して、正当な暗号文と不当な暗号文を識別することができないことを保証することである。

それでは、Hash Proof System を KEM 方式として見た場合に、上記で説明した「秘密鍵」と「乱数」の役割の交換が可能なことを確認する。まず、(1)に関して、暗号文の生成は正当な暗号文空間からのサンプルだけであり、公開鍵に依存しない。そのため(1)の問題は回避される。次に、(2)に関して、変換後の方式において、暗号文である公開鍵 pk=μ(sk)と秘密鍵である暗号文 c の証拠 w から共有鍵を取り出すには、Pub(pk,c,w)を使えばいいことが分かる。この共有鍵は、変換前の方式における共有鍵 Priv(sk,c)と一致するため、変換後の共有鍵は KEM 方式の共有鍵として問題なく利用できる。

実際の、エントロピー安全な方式の構成法は以下のとおりである。

鍵生成: 入力 1^n に対して、F=(SK,PK,C,V,K, Λ, μ)←Param(1^n)を生成し、c∈V および
その証拠 w をランダムに選び、K.PK=(F,c),K.SK=w を出力
KEM 暗号化: 入力 K.PK=(F,c)に対して、sk∈SK をランダムに選び、
pk=μ(sk),K=Λ_{sk}(c)を計算し、pk を暗号文として、K を共有鍵として出力
KEM 復号: 入力 K.PK=(F,c),秘密鍵 K.SK=w,暗号文 pk に対し、Pub(pk,c,w)を出力

Hash Proof System が正当性を満たしているとき、任意の(F,c),pk,w に対して Pub(pk,c,w)=Λ_{sk}(c)が成り立つため、上記の KEM 方式も正当性を満たしている。

上記の方式の具体的な実現例として、決定性 Diffie-Hellman 仮定にもとづく構成法を示す。G を素数位数 p の群とする。

鍵生成: 入力 1^n に対して、$x \in Z_p$ および $g_1, g_2 \in G$ をランダムに選び、$pk=(g_1, g_2, g_1^x, g_2^x)$, sk=x を出力
KEM 暗号化:入力 $pk=(g_1, g_2, pk_1, pk_2)$ に対して、$r,s \in Zp$ をランダムに選び、$c=g_1^r g_2^s, K=(pk_1)^r (pk_2)^s$ を出力
KEM 復号: 入力 sk および c に対して、$K=c^{sk}$ を出力

この方式の正当性については、$c=g_1^r g_2^s, pk_1=g_1^x, pk_2=g_2^x$ であることから、$K=(pk_1)^r (pk_2)^s=(g_1^x)^r (g_2^x)^s=(g_1^r g_2^s)^x=c^x$ であることから確認できる。

4 分割漏洩モデルにおける安全な方式

2-1 および 3-1 節における考察より、秘密鍵漏洩および乱数漏洩ともに、暗号文に依存した漏洩が起きると安全性が達成できないことが分かった。3 章では、それを踏まえ、KEM/DEM 方式に適した乱数漏洩モデルを考え、漏洩があっても安全な方式を提案した。

Halevi と Lin[4]は、分割漏洩モデルという緩和した漏洩モデルを考え、秘密鍵漏洩に対して、暗号文に依存した漏洩に対しても安全な方式の構成法を与えた。分割漏洩モデルでは、秘密鍵が２つの部分に分けられ、漏洩関数はそれぞれの部分毎に計算する漏洩はできても、２つの部分を同時に計算するような漏洩はできないモデルである。

そこで本研究では、分割漏洩モデルにおいて達成可能な安全性について考察を行った。結果として、分割漏洩モデルにおいては、比較的シンプルな方法で強い安全性を達成可能なことがわかった。

4-1 構成のアイディア

二情報源乱数抽出器 $\mathrm{Ext2}:\{0,1\}^t\times\{0,1\}^t\rightarrow\{0,1\}^m$ を考える。Borugain[5]は、Ext2 の各入力のエントロピーがある $\gamma<1/2$ に対して、γt 以上あれば、Ext2 の出力と一様乱数の統計的距離が $2^{-\Omega(m)}$ であるような Ext2 の構成法を与えた。この乱数抽出器のような、一様乱数との統計的距離が指数関数的に小さい乱数抽出器は、強力な漏洩耐性をもつことがわかった。それは、そのような Ext2 に対しては、Ext2 の出力に依存した漏洩が起きたとしても、２つの入力への漏洩が独立に行われている限り、出力の値は一様乱数に近いという性質である。このような性質があるため、公開鍵暗号における秘密鍵生成の乱数および暗号化の乱数を、Ext2 の出力に置き換え、Ext2 への入力を分割して漏洩が起きる２つの部分に対応させれば、既存の安全な方式が、そのまま分割漏洩モデルにおいても安全な方式になるのである。

4-2 秘密鍵・乱数同時漏洩に対して安全な方式の構成

公開鍵暗号方式 $\Pi=(\mathrm{Gen},\mathrm{Enc},\mathrm{Dec})$ が与えられたとき、それに漏洩耐性を持たせる具体的な構成法は以下のとおりである。

鍵生成: $s_1,s_2\in\{0,1\}^t$ をランダムに選び、$w=\mathrm{Ext2}(s_1,s_2)$ を計算し、$(pk,sk)=\mathrm{Gen}(1^n;w)$ を出力
暗号化: 入力 pk, m に対して、$r_1,r_2\in\{0,1\}^t$ をランダムに選び、$w'=\mathrm{Ext2}(r_1,r_2)$ を計算し、
$\mathrm{Enc}(pk,m;w')$ を出力
復号: 入力 sk, c に対して、$w=\mathrm{Ext2}(s_1,s_2)$ および $(pk,sk)=\mathrm{Gen}(1^n;w)$ を計算し、$\mathrm{Dec}(sk,c)$ を出力

上記の方式は、秘密鍵と乱数が同時に漏洩するような場合でも、Π と同じ安全性を達成することができる。つまり、Π が CCA 安全性をもつならば、得られた方式も CCA 安全性を満たし、ID ベース暗号であれば、得られた方式も漏洩に耐性のある ID ベース暗号となる。

上記の方法は、非常にシンプルなアイディアにもとづいているが、強力な漏洩耐性を持たせる技術である。この事実から、分割漏洩モデルは、非常に高い漏洩耐性を持たせやすいモデルであると言うこともできる。

5 改竄に耐性のある暗号技術

最後に、改竄に耐性のある暗号技術として、頑健符号を紹介する。頑健符号は、誤り訂正符号および誤り検出符号を弱めた概念として捉えることができる。誤り訂正符号は、誤りが起きた場合、それを訂正し、元の情報を復元する技術である。誤り検出符号は、情報の復元は行わず、誤りを検出する技術である。頑健符号は、誤りの検出さえ行わず、元の情報と独立した情報に復元する技術である。元の情報と独立した情報に復元するため、(悪意のある) 意図的な改竄ができない。誤りの訂正や検出よりも弱めた概念であるため、幅広い改竄 (誤り) に対して頑健にできる可能性がある。

Dziembowski ら[6]は、頑健符号の概念を導入し、誤りの検出が不可能であるような改竄クラスに対して、頑健符号が効率的に構成できることを示した。また、かなり広いクラスの改竄に対して頑健符号の存在性を示している。さらに、分割漏洩モデルを改竄攻撃に拡張した、分割改竄攻撃に対する頑健符号をランダムオ

ラクルモデルにおいて示した。

Liu と Lysyanskaya[7]は、分割漏洩・改竄攻撃に対する頑健符号を CRS モデルにおいて構成した。ただし、安全性は計算量的なものであり、構成においても、非対話ゼロ知識証明などのやや効率の悪い暗号技術を利用している。

Aggarwal ら[8]は、分割改竄攻撃における頑健符号の構成に取り組み、加法的組合せ論の技術を用いることで、情報理論的に安全で非常にシンプルな構成法の頑健符号を提案している。

【参考文献】

[1] Adi Akavia, Shafi Goldwasser, and Vinod Vaikuntanathan. Simultaneous hardcore bits and cryptography against memory attacks. In Omer Reingold, editor, Theory of Cryptography, 6th Theory of Cryptography Conference, TCC 2009, volume 5444 of Lecture Notes in Computer Science, pages 474–495, Springer 2009.

[2] J. Alex Halderman, Seth D. Schoen, Nadia Heninger, William Clarkson, William Paul, Joseph A. Ca- landrino, Ariel J. Feldman, Jacob Appelbaum, and Edward W. Felten. Lest we remember: cold-boot attacks on encryption keys. Communications of the ACM, volume 52, number 5, 2009.

[3] Moni Naor and Gil Segev. Public-key cryptosystems resilient to key leakage. In Shai Halevi, editor, CRYPTO, volume 5677 of Lecture Notes in Computer Science, pages 18–35, Springer, 2009.

[4] Shai Halevi and Huijia Lin. After-the-fact leakage in public-key encryption. In Yuval Ishai, editor, TCC, volume 6597 of Lecture Notes in Computer Science, pages 107–124, Springer, 2011.

[5] Jean Bourgain. More on the sum-product phenomenon in prime fields and its applications. International Journal of Number Theory, volume 1, number 1, pages 1–32, 2005.

[6] S. Dziembowski, K. Pietrzak, and D. Wichs. Non-malleable codes. In A.C.-C. Yao, editor, ICS, pages 434–452, Tsinghua University Press, 2010.

[7] F.-H. Liu and A. Lysyanskaya. Tamper and leakage resilience in the split-state model. In R. Safavi-Naini and Ran Canetti editors, CRYPTO 2012, pages 517–532, Springer, 2012.

[8] Divesh Aggarwal, Yevgeniy Dodis and Shachar Lovett. Non-malleable Codes from Additive Combinatorics. ECCC Report, TR13-081, 2013.

〈発　表　資　料〉

題　　名	掲載誌・学会名等	発表年月
Randomness Leakage in the KEM/DEM Framework	論文誌投稿中	未定
Public-Key Cryptography Resilient to Post-Challenge Key and Randomness Leakage in Split-State Model	国際会議投稿中	未定
Leakage-Resilience of Stateless/Stateful Public-Key Encryption from Hash Proofs	IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences	2013 年 6 月