クラウド環境におけるデータ委託と外部操作に適した秘密分散方式の研究

2016年3月

須賀 祐治

クラウド環境におけるデータ委託と外部操作に適した秘密分散方式の研究

須賀 祐治

システム情報工学研究科

筑波大学

2016年3月

研究概要

クラウド環境におけるセキュリティ上の課題のひとつとして可用性の確保がある．2011 年に発生した東日本大震災以降，ディザスタリカバリや事業継続計画に対する関心がビジネスの観点からも高まっている．具体的には安定的な電力供給が見込めない，もしくは地震の余波によるネットワーク遮断の可能性からクラウドリソースを利用できない状況が鑑みられている．またデータセンターへの物理的な被害により，データ紛失という問題も考えられる．平成 27 年版情報通信白書において「クラウドサービスの導入理由」についてのアンケート結果が報告されており,「安定運用・可用性が高くなるから」がアウトソーシングのメリットに続けて上位にランクされていることからも，可用性はニーズの高い要件のひとつと考えることができる．

このような背景のもと，クラウド環境での秘匿性の確保も重要視されている．クラウドは「必要に応じた処理能力を低コストで確保でき，その能力を手軽に利用できる」サービスであり，様々な業界で利用されている．クラウドの利用形態としては，パブリッククラウドとプライベートクラウドに分類することが多い．パブリッククラウドはインターネットなどオープンに提供されているクラウドを指す．一方，プライベートクラウドは企業等の閉じたネットワークで利用されるなど，全ての権限をコントール配下に置く専用クラウドと捉えることができる．そのためパブリッククラウドとプライベートクラウドは用途，特に扱われる情報の重要度に応じて，また，事業者の信頼度に応じて使い分けがなされている．

一方で，クラウド利用に躊躇する顧客も存在する．その大きな理由は，個人向けクラウドだけでなく企業向け・官公庁向けクラウドにおいて「商用のクラウドサービス業者に安心してデータを預けられるか」という懐疑的な考え方を完全に払拭できないためである．この不安に対する解決策として様々な技術が暗号コミュニティから提案されている．分散データを秘密にしたままデータマイニングする Privacy-preserving Data Mining, 検索キーを秘匿したまま（暗号化された）検索結果を取得する Searchable Encryption のほか，暗号化データを（信頼していない）クラウドに計算作業を委託して計算結果を入手可能にする Gentry による Fully Homomorphic Encryption などがそれにあたる．また，岡本-高島方式は復号条件が AND，OR，NOT 演算，閾値ゲートにより構成される関係式で表現可能であり，より柔軟性の高い復号形態を持つためクラウド環境に適した暗号方式として注目されている．

これらの背景のもと，上記 2 つの課題をバランスよく持つ技術として秘密分散方式 (Secret Sharing Scheme; SSS) が提案されており Blakley と Shamir により独立に提案された (k, n)-しきい値秘密分散法がよく知られている．(k, n)-しきい値秘密分散法は，秘密情報 S を n 個の分散情報（シェア）に符号化し配布した状態で，任意の k 個の分散情報からは S を復号可能

であるが，任意の $k-1$ 個の分散情報からは S に関する情報は全く得られないという性質を持つ．本技術の導入効果として（１）漏洩リスクの分散（＝一部漏洩しても暴露されない），（２）紛失リスクの分散（＝一部紛失しても復元可能）の 2 つがあり，アプリケーション・ユースケースに応じて上記パラメータ k, n を選択できる自由度を持つ．

本論文では，まずはじめに，データに対する秘密分散方式として，分散・復元時に排他的論理和だけを用いて構成する理想的な秘密分散法 (XOR-SSS) の新しい構成方式を提案する．XOR-SSS は排他的論理和だけを用いるため従来の Shamir による構成法に比べてはるかに高速に分散・復元が可能であるという優位性を持つ．またオリジナルとシェアのサイズが不変な秘密分散法でもあり，ストレージを有効に使う意味でもクラウドへの適合性が高い．

ここで，クラウド事業者はデータをアーカイブする際に，秘匿性を高めるために，秘密分散方式の適用だけでなく，データの暗号化処理も併用することを考える．さらに，クラウド事業者は不測のデータクラッシュなどの障害に対処するなど可用性を高めるために，顧客から預かったデータを主体の異なる他の事業者にコピーもしくはデータの一部を暗号化・秘密分散処理を行った上で再配布を行うというケースも検討する．このとき，秘密分散処理とデータ暗号化がお互いに可換な処理であるとすると，どこに・どのように暗号化済み分散データが流通していたとしても，元データの復元が容易になることが分かる．データ暗号化処理としてストリーム暗号やブロック暗号の CTR モードの利用など，鍵データを平文に足しこむ形の演算で暗号化を行っているケースにおいて，排他的論理和 (XOR) 演算のみを用いて構成する秘密分散法を併用することにより「秘密分散処理とデータ暗号化の可換性」を保つことができる．

さらに，クラウド上で暗号化したまま演算を行う秘密計算・委託計算への拡張についても試みている．クラウドをデータのアーカイブ先として利用する静的な利用に対して，クラウドサービスに計算を委託し，暗号化状態のまま計算して結果を得るという動的な利用方法のひとつである．

次にクラウド環境における完全性の確保について検討する．2015 年度も，事件・事故などで漏えいしたユーザ ID とパスワードのリストを用いたとみられる不正ログインが後を絶たない状況が続いた．そのため ID・パスワードだけを用いた認証方式は危険であるという認識が拡がり，他の認証方式を併用するなど新しい ID 管理技術が注目されている．認証で必要となる秘密情報のうち”Something you have”に属するハードウェアトークンでは，物理的媒体に表記された定期的に表示が更新されるワンタイムパスワードをトークンとして提示し，多要素認証の 1 つとして利用されることが一般的である．ワンタイムパスワード方式は，従来の ID・パスワード方式の置き換えや併用により，近年のリスト型攻撃への対策案の 1 つとして期待されている．”Something you have”に属するパスワード（秘密情報）としてログイン時に利用される物理的媒体が存在するが，ネットワークに接続可能になった時点で，そのデバイス自身がハッキングされてしまい秘密情報が漏れてしまうという問題が生じてしまう．つまり，インターネットなどの広域網に繋がらないでも安全に管理できる物理的媒体で秘密情報を分散しておくニーズがあるとも言える．

この背景のもと，復号に計算リソースを利用しない視覚復号型秘密分散法 (VSSS) について

も取り扱う．VSSS は，機密画像を複数のシェア画像にあらかじめ分散し OHP シートのような透過性を持ち物理的に重ね合わせが可能な媒体に印刷して，シェア画像を重ね合わせることで目の錯覚を用いて機密画像を復元する方法である．ここでクラウド上のデータにアクセスするための権限をコントロールし，しかるべきエンティティがしかるべきデータにアクセスを許可するフェーズにおいて VSSS を適用することを検討する．

(k, n)-しきい値視覚復号型秘密分散法は, 復元するための権限としては平等に分散情報が分配されるため, さまざまなアクセス構造が想定される実利用においてはうまく適用できないことが多い．そこで本論文では, グラフで表現されたアクセス構造を持つ視覚復号型秘密分散法 GVSSS について追求する．はじめに，既存の概念を拡張して辺の有無ではなく 2 頂点間の距離に基づいたアクセス構造を持つ視覚復号型秘密分散法 G_dVSSS を提案する．これにより 2 枚のシェアを重ねあわせたときに, シェアの距離により復元される機密画像が異なるような VSSS を構成することを実現した．また，既存の GVSSS に対して，復元画像のコントラストの改善を検討し, より見やすい構成方法についても新たに提案する．これまでの構成方法であるスターグラフ分割法は，与えられたグラフ G をスターグラフ，つまり 1 頂点（中心）からしか辺が存在しないグラフに分割する方式である．スターグラフからは画像拡大率 2 の生成行列を持つ GVSSS が構成できることが知られており，この画像拡大率 2 の生成行列を利用して，分割されたスターグラフの生成行列を連結する（各分割スターグラフの生成行列を横に並べる）ことでグラフ G に対する生成行列を構成することができる．しかし，グラフ間のエッジの数が多い場合や頂点数が多い場合には，画像拡大率は肥大化する傾向にあり，見やすさの点で問題が生じる．また，スターグラフ分割法は必ずしも相対差が最大となる構成方法ではない．そこで本論文では，スターグラフ分割法を改善し，完全 n 部グラフ分割法，辺消去法を提案し，画像拡大率を下げる提案を行う．さらに，与えられたグラフに対し，それ以上画像拡大率を下げられない optimal な構成方法を導出する方法を検討した．グラフ G とその生成行列（つまり GVSSS-(G, m) の構成方法）が与えられたとき，optimal かどうかを判定することは非常に難しいため，画像拡大率を固定した上で，どのようなグラフで optimal な構成が可能かを分類するというアプローチを取り，画像拡大率 4 までの GVSSS の分類を行っている．さらに，画像拡大率を抑えるために，復元画像として白黒反転画像を許容する視覚復号型秘密分散法 $G^{\pm}$VSSS を提案する．その結果，例えば，頂点数 9 の Lattice graph $L_2(3)$ においては，スターグラフ分割法では画像拡大率が 14 必要であったが，本論文では画像拡大率を 3 にまで抑える生成行列の構成事例など大きな改善を行うことができた．

最後に，クラウド事業者が顧客からのデータをアーカイブし，顧客の要求に応じてデータを処理するユースケースにおいて XOR-SSS と GVSSS を用いた実際の構成例について検討する．秘匿性と可用性の要件を満たす秘密分散方式において，格納データのライフサイクルに着目する．また，秘密分散方式の 1 種である視覚復号型と呼ばれる方式をエンティティの認証・認可として併用することで，CIA に準えられる 3 つの要件をトータルでカバーできることを確認し，本論文で扱う提案方式の有用性を示す.

目 次

表 目 次

図 目 次

第1章　研究背景

本論文では，クラウド環境に対するセキュリティ機能の一つとして秘密分散（Secret Sharing Scheme）技術の適用を検討する．よく知られている (k,n)-しきい値秘密分散法は，秘密情報 S を n 個の分散情報（シェア）に符号化し配布した状態で，任意の k 個の分散情報からは S を復号可能であるが，任意の $k-1$ 個の分散情報からは S に関する情報は全く得られないという性質を持つ．本技術の導入効果として（1）漏洩リスクの分散（＝一部漏洩しても暴露されない），(2）紛失リスクの分散（＝一部紛失しても復元可能）の 2 つがある．両者ともにクラウドサービス導入を妨げる要因を解消すると考えられており，実際にサービス提供される事例もある．

本章では，秘密分散技術がクラウドサービスと親和性が高いことを示すために，まずクラウド環境におけるセキュリティ上の課題について整理する．

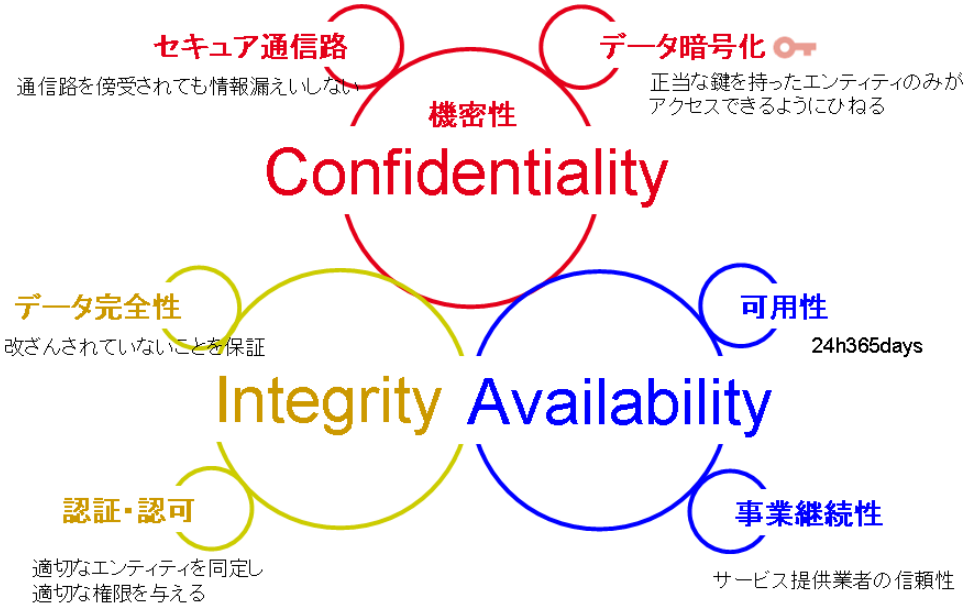

図 1.1: CIA モデル概要

1.1　クラウドにおけるセキュリティ要件

必要に応じた処理能力を低コストで確保でき，その能力を手軽に利用できるクラウドサービス [1, 2] の利用が拡大している．クラウドの用途として，パブリッククラウドとプライベートクラウドに分類することが多い．パブリッククラウドはインターネットなどオープンに提供されているクラウドを指す．一方，プライベートクラウドは企業等の閉じたネットワークで利用されるなど，全ての権限をコントール配下に置く専用クラウドと捉えることができる．そのためパブリッククラウドとプライベートクラウドは用途，特に扱われる情報の重要度に応じて，また，事業者の信頼度に応じて使い分けがなされている．

一般的なセキュリティ要件の考え方を論じる際に用いる CIA(Confidentiality, Integrity and Availability) モデルを用いてクラウド利用環境のセキュリティ要件について分類が検討されている [3]．また，各省庁からもクラウド環境における情報セキュリティに関して，利用者の視点と事業者の視点で取りまとめられた各種ガイドライン [4, 5, 6, 7] が発行されている．次節以降，図 1.1 に提示する CIA モデルにおいてクラウドで提供すべきセキュリティ要件ごとに整理する．

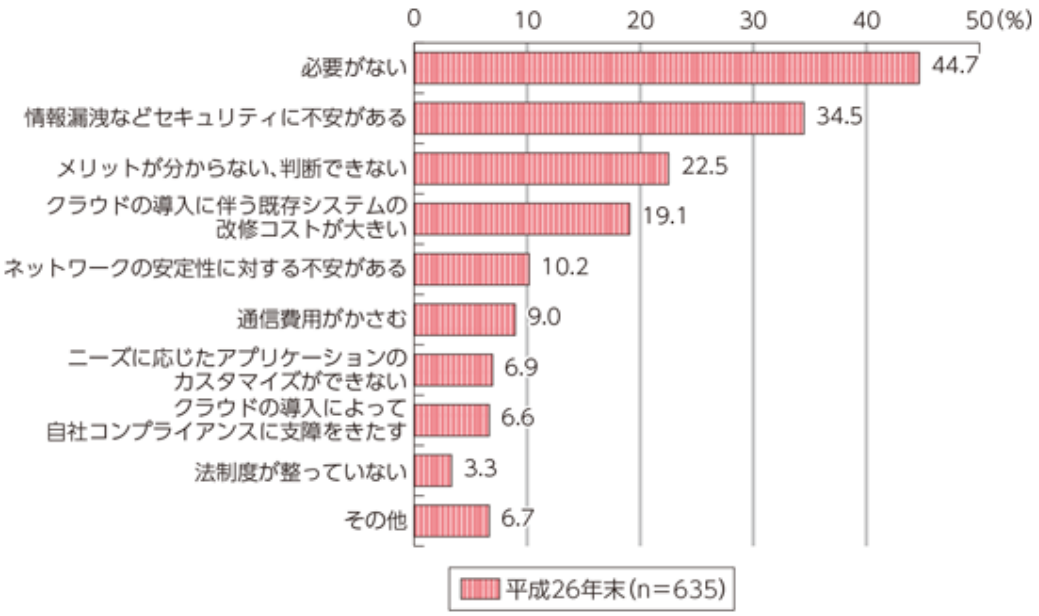

図 1.2: クラウドサービスを利用しない理由（出典：平成 27 年版情報通信白書 図表 7-2-1-23）

1.1.1　秘匿性

図 1.2 に示すように平成 27 年版情報通信白書 [8] にはクラウドサービスを利用しない理由（図表 7-2-1-23）が報告されており,「必要がない」が 44.7 %と最も高く，次いで「情報漏洩などセキュリティに不安がある」（34.5 %）というアンケート結果が報告されている．

これは，個人向けクラウド（個人情報：電子メール，画像，住所録，家計簿），企業向けクラウド（社内機密情報，顧客情報），官公庁向けクラウド（国家機密情報，住民情報）において「商用のクラウドサービス業者に安心してデータを預けられるか？」という疑問が払拭できないことに起因すると考えられる．

この不安に対する解決策として様々な技術が暗号コミュニティから提案されている．分散データを秘密にしたままデータマイニングする Privacy-preserving Data Mining [9], 検索キーを秘匿したまま（暗号化された）検索結果を取得する Searchable Encryption [10] のほか，暗号化データを（信頼していない）クラウドに計算作業を委託して計算結果を入手可能にする Gentry による Fully Homomorphic Encryption [11] などがそれにあたる．その後も継続的に改良 [12] が続けられている．また，岡本-高島方式 [13] は復号条件が AND，OR，NOT 演算，閾値ゲートにより構成される関係式で表現可能であり，より柔軟性の高い復号形態を持つためクラウド環境に適した暗号方式として注目されている．このように，クラウド環境においても実用的なツールとして近づきつつある．

1.1.2 完全性

データ完全性については通常のストレージサービスで利用されているような改ざんチェック機構を用いることができると考えられる．また通信データの完全性についても，通常利用されるような標準的なセキュアプロトコルを利用する限りは，これまでと同様に改ざん防止機構を利用可能である．

次にクラウドサービスにおける認証・認可機構について考察していく．クラウドサービス上で暗号化されていたとしても，当該データにアクセスするための認証・認可の仕組みは必要である．経済産業省より発行のクラウドセキュリティガイドライン [6] は 2013 年度版に改訂されており，事業者側で行う対策がまとめられている．その中で 11 章にてアクセス制御に関する記載があり「クラウド事業者は，提供するクラウドサービスにおいて，利用者のアクセス制御機能を提供することが望ましい」等の記載がされている．しかし認証時には ID・パスワードベースの認証しか想定されていない．さらに，複数のクラウドサービスをマッシュアップして利用する際には，ユーザ-クラウド間の認証・認可の仕組みだけでなく，クラウド-クラウド間の認証・認可の仕組みが必要となる．その際には，あるユーザのデータを本来委託した事業者以外のエンティティが取り扱うことも発生するため，権限委譲の仕組みも必要とされる．

これらを鑑みると，ユーザ認証時に ID・パスワードだけを利用するサービスでは不安視されてしまう．実際，2015 年度も事件・事故などで漏えいしたユーザ ID とパスワードのリストを用いたとみられる不正ログインが後を絶たない状況が続いた．そのため ID・パスワードだけを用いた認証方式は危険であるという認識が拡がり，他の認証方式を併用するなど新しい ID 管理技術が注目されている．認証で必要となる秘密情報のうち”Something you have”に属するハードウェアトークンでは，物理的媒体に表記された定期的に表示が更新されるワンタイムパスワードをトークンとして提示し，多要素認証の 1 つとして利用されることが一般

的である．ワンタイムパスワード方式は，従来の ID・パスワード方式の置き換えや併用により，近年のリスト型攻撃 [14] への対策案の 1 つとして期待されている．

”Something you have”に属するパスワード（秘密情報）としてログイン時に利用される物理的媒体が存在するが，ネットワークに接続可能になった時点で，そのデバイス自身がハッキングされてしまい秘密情報が漏れてしまうという問題が生じてしまう．つまり，インターネットなどの広域網に繋がらないでも安全に管理できる物理的媒体で秘密情報を分散しておくニーズがあるとも言える．

1.1.3 可用性

東日本大震災以降，ディザスタリカバリや事業継続計画に対する関心が高まっている．具体的には安定的な電力供給が見込めない，もしくは地震の余波によるネットワーク遮断の可能性からクラウドリソースを利用できない状況が鑑みられている．またデータセンターへの物理的な被害により，データ紛失という問題も考えられる．

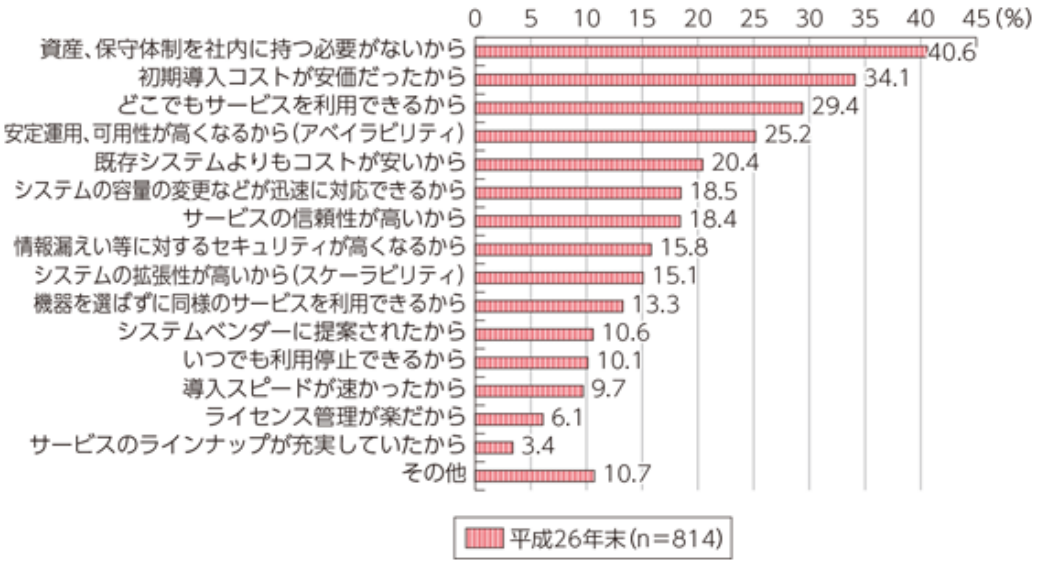

図 1.3: クラウドサービスの導入理由（出典：平成 27 年版情報通信白書 図表 7-2-1-22）

図 1.3 は平成 27 年版情報通信白書 [8] におけるクラウドサービスの導入理由（図表 7-2-1-22）についての調査結果であり,「資産・保守体制を社内に持つ必要がないから」(40.6%),「初期導入コストが安価だったから」(34.1%),「どこでもサービスを利用できるから」(29.4%) に続けて「安定運用・可用性が高くなるから」(25.2%) がアウトソーシングのメリットに続けて第 4 位に挙げられている．ここでも，セキュリティ 3 要件のうち可用性の確保に高いニーズがあることが分かる．

そのほか，SLA(Service Level Agreement) による利用者にサービスの品質を保証する制度の SaaS 版として経済産業省が取りまとめたガイドライン [4] には，サービス稼働率，平均復旧

時間，データ消去の要件，通信の暗号化レベルに関する取り決めの例が掲載されている．また「クラウド事業者の信頼性」に対するお墨付き制度として SAS 70 type II[15] や FMMC による 「ASP・SaaS 安全・信頼性情報開示認定制度」[16] などが存在する．

1.1.4　秘密分散技術適用におけるビジネス上の課題

実際に秘密分散共有技術をクラウド環境でサービスインする動きも見受けられる．しかし公開されている情報からはサービスを構成するための具体的な技術要素が不明瞭である．そのため，クラウドサービス事業者が不正する，構成方法自体が脆弱であるなどの事由により，顧客から預かった情報が復元できてしまう可能性がある．このビジネス上の課題に対して，技術適用とその技術の確からしさを顧客に提示することが必要である．顧客への説明責任を踏まえ，シェアの一部を利用者側でプライベートクラウドとして運用するというビジネスモデルを好む顧客が潜在的に存在するとも考えられる．可用性要件の節で前述したように「お墨付き制度」だけでクラウド事業者を選択するのではなく，透明性のあるセキュリティ技術を提供することで，顧客への安心感を提供するべきである．

平成 27 年版情報通信白書 [8] における「クラウドサービスの利用動向」内に，クラウドサービスの利用内訳（図表 7-2-1-21）についての調査結果が報告されている．図 1.4 で示したように，ファイル保管・データ共有が第 1 位の利用形態であり，クラウドユーザの半数近くが利用していることが報告されている．

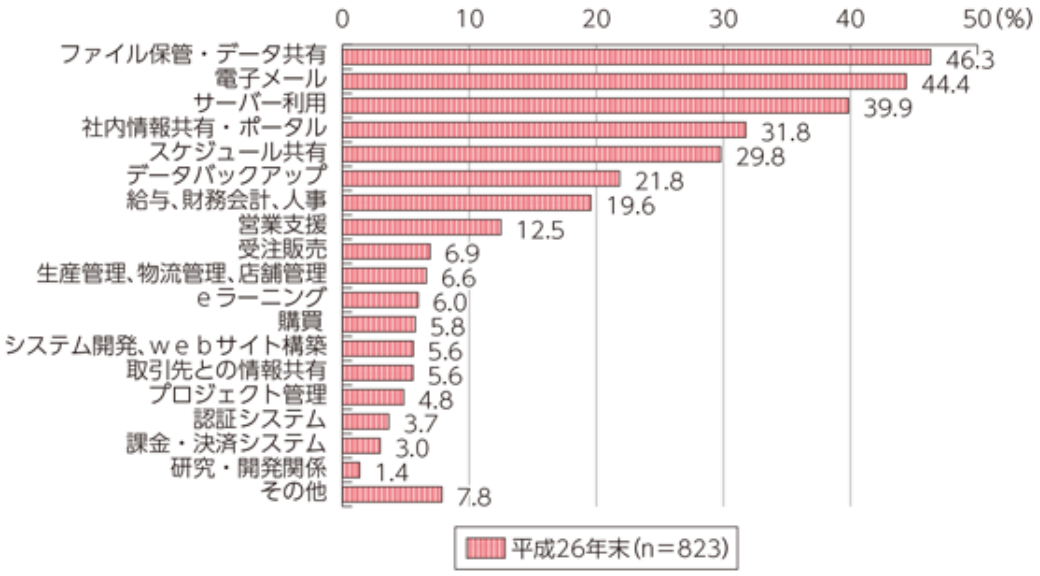

図 1.4: クラウドサービスの利用内訳（出典：平成 27 年版情報通信白書 図表 7-2-1-21）

そこで本論文では，クラウド事業者が顧客からのデータをアーカイブし，顧客の要求に応じてデータを処理するケースにおいて，クラウドでの利用に適した秘密分散方式の利用方法を検討していく．

1.2　データフローモデル

データをアーカイブする際にクラウド事業者は，前節で述べたように秘匿性と可用性を高めるために，１）データ暗号化，２）秘密分散法の両方を併用することが考えられる．また，クラウド事業者は不測のデータクラッシュなどの障害に対処するため，やはり可用性を高めるために，顧客から預かったデータを主体の異なる他の事業者にコピーもしくはデータの一部を暗号化・秘密分散処理を行った上で再配布を行うというケースを考える．

1.2.1　データフローの分類

１）データ暗号化と２）秘密分散法の両方を併用する場合には，図 1.5 に示すように E 型と F 型の 2 つのフローが考えられる．E 型は１）データ暗号化→２）秘密分散法の処理順（左回り），F 型はその逆で２）秘密分散法→１）データ暗号化の処理順（右回り）と考えればよい．

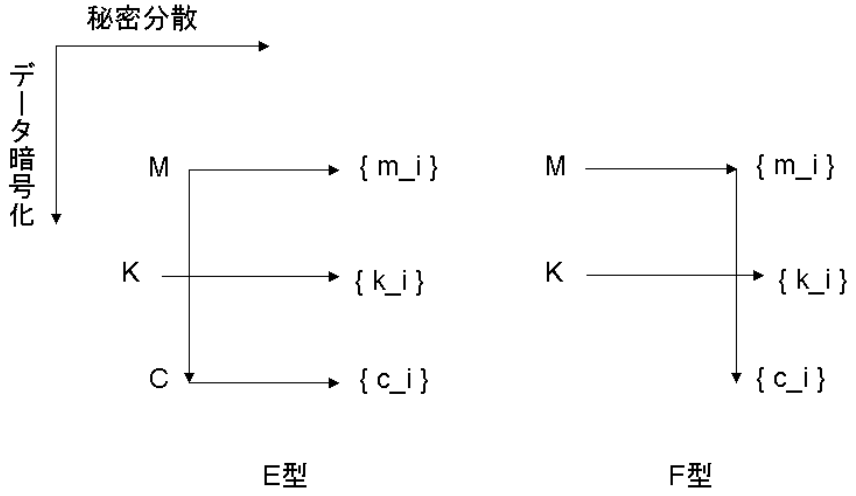

図 1.5: データフローモデル

暗号化処理においては，顧客データ M を鍵 K で暗号化したデータを C と表記する．また秘密分散処理においては，データ D（M, K, C のいずれか）をシェアの集合（分散データ）$\{d_i\}$ に分散すると表記する．E 型，F 型ともに，暗号処理に用いた鍵を秘密分散して委託するケースも考えられる点に留意する．この場合，鍵だけではなく，暗号化データとともに格納するケースでは，格納するクラウド事業者をそれぞれ異なる主体に委託するなどの考慮が必要となる．

1.2.2　データ処理要件

前述のデータフローモデルにおいては，以下のことに留意する必要がある．アーカイブ依頼されたクラウド事業者は，異なる主体の事業者にデータを複製する場合，図 1.6 のように同

一主体が「アクセス可能グループ」となる複数のシェアを意図せず得てしまうケースを避ける必要がある．復元権限を得たクラウド事業者は不正またはサーバ等をクラックされることにより，顧客データを復元できてしまうためである．

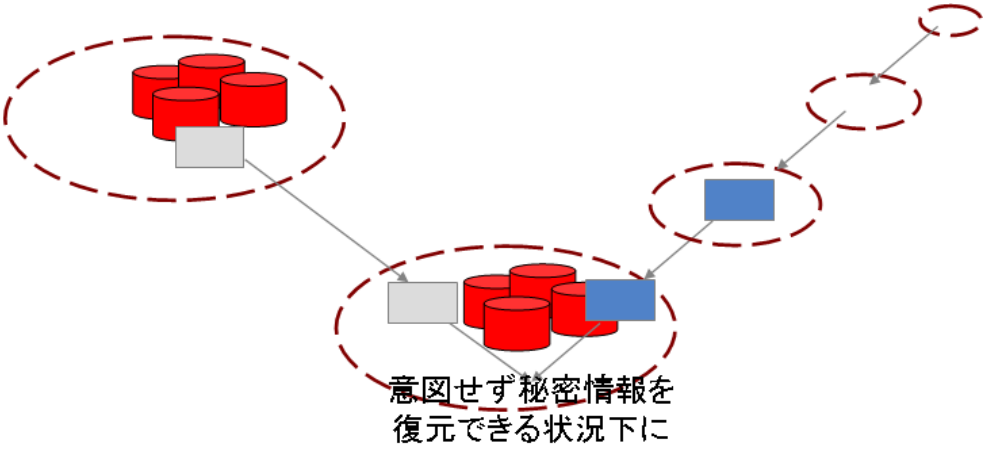

図 1.6: 意図せず復元可能な状態下になるケース

これを避けるためには，顧客からどのようにデータが伝播しているかについて知ることができるように配慮すべきである．これは，事業者間のデータフローだけでなく，同一事業者内においてデータを複製・分散しているケースでも同様である．このとき前節のデータフロー図などの表記法を用いて視覚化し，常時顧客に伝えることができると考えられる．

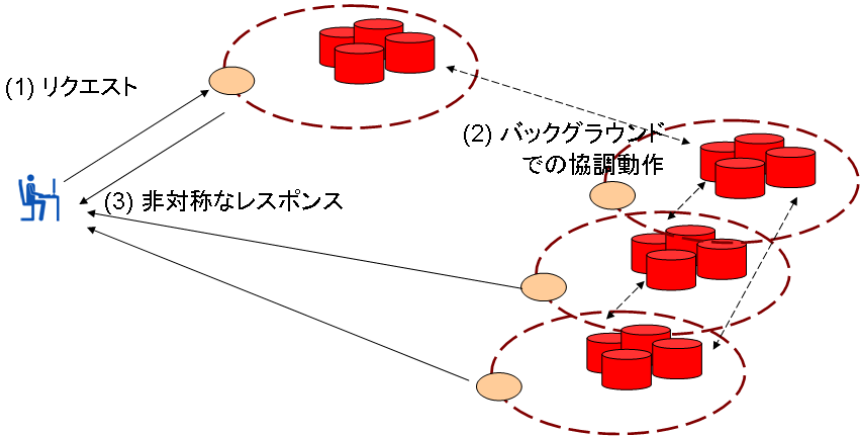

図 1.7: 非対称なクラウドサービス

1.3 本論文が解決すべき課題

複数のクラウドで複製，分散が繰り返されている顧客データの断片を復元・復号するフェーズを考える．その際には前述したデータフロー図を遡ることで復元・復号作業を行うことが一般的であると考えられる．しかしこのサービスは必ずしも対称である必要はない．つまり図 1.7 にように，リクエストに対するレスポンスを当該リスクエストサービスが返答する必要はない．例えば，ネットワークトポロジーを鑑み，最も転送に効率のよい経路を選択する，課金が最も低いクラウドサービスから優先的に選択する等，クラウドサービスの連携により，より良い選択肢をユーザに与えるビジネスモデルの提供が望まれる．

この連携サービスにおいては，各クラウド共通の標準的な API，フォーマットなどの準備が必要であるが，処理レベルに鑑みると図 1.8 のように新しい要件として「秘密分散処理とデータ暗号化の可換性」が必要であることがわかる．ここで，クラウド環境の処理速度を考慮すると高速処理が可能な方式の適用が望ましい．

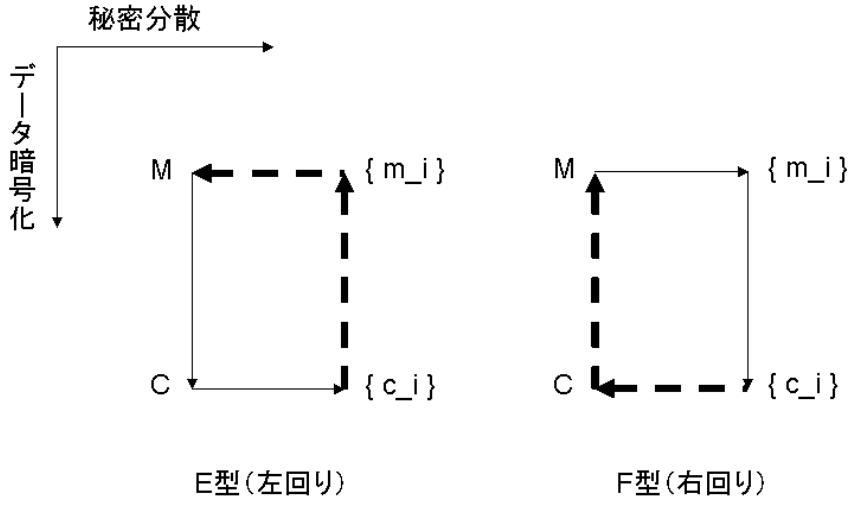

図 1.8: E 型, F 型における復元処理サイクル

具体的には，複数の暗号化・分散処理を経て，数ホップ先に格納されたデータ群から，復元に必要なデータに直接アクセスして転送・復元処理を行うことで，処理速度を軽減する技術が有益である．このとき，本モデルはクラウドサービスの信頼度から，クラウドサービスにデータ復元委託を行うケースも含まれていることに留意する．つまり，データを委託した後も，暗号化状態のまま分散復元する，もしくは分散状態で復号処理する等，外部からの操作可能性が要件の一つとして必要とされていることを示している．

さらに，秘密分散方式の適用により秘匿性と可用性の要件を満たすことから，秘密分散方式の 1 種である視覚復号型と呼ばれる方式をエンティティの認証・認可として併用することで CIA に準えられる 3 つの要件をトータルでカバーできる方式について検討していく．

第2章　秘密分散方式の一般的な構成法

本章では，秘密分散方式の一般的でよく知られた構成事例について取り上げる．秘密情報 S の所有者と，通常ディーラーと呼ばれる分散者は同じエンティティとしても，別々のエンティティとしても考えてもよい．ディーラーは秘密情報 S を n 個の分散情報（シェア）に符号化して配布する役割を持つ．

秘密分散法（Secret Sharing Scheme）の一例である (k,n)-しきい値秘密分散法は，秘密情報 S を n 個の分配情報（シェア）W_i $(1 \leq i \leq n)$ に符号化する方式であり，任意の k $(k \leq n)$ 個の分配情報 から秘密情報 S を復号することは可能であるが， $k-1$ 個以下の分配情報からは秘密情報 S に関する情報は全く得られないという性質を持つ．(k,n)-しきい値秘密分散法は 1979 年に Shamir と Blakley によって独立に提案されている [17] [18].

(k,n)-しきい値秘密分散法は復元のためのアクセス構造としては，もっとも単純な方式の一つであり，様々な構成方法がこれまで提案されている [19, 20, 21].

2.1　多項式補間を用いる (k,n)-しきい値秘密分散法

(k,n)-しきい値秘密分散法は，しきい値 k 個以上の分散情報から秘密情報 S を一意に復元することが可能である．また，k 個未満の分散情報からは S についての情報は一切漏えいしないことを特徴とする．

2.1.1　分散アルゴリズム

1. $S < p$ かつ分散数 $n < p$ である任意の素数 p を選ぶ.
2. $GF(p)$ の元から，異なる n 個の $x_i (i = 1, \ldots, n)$ をランダムに選択し，これらをシェア ID とする.
3. $GF(p)$ の元から，$k-1$ 個の係数 $a_l (l = 1, \ldots, k-1)$ をランダムに選択し，以下の曲線を生成する.
$$W = S + a_1 x + a_2 x^2 + \cdots + a_{k-1} x^{k-1}$$
4. 上記の x に各シェア ID を代入して得られる分散情報 W_i を計算し，各エンティティごとに (x_i, W_i) を配布する.

2.1.2 復元アルゴリズム

1. シェア ID x_{i_m} および シェア $W_{i_m}(m = 1, \ldots, k)$ を復元のために収集する．
2. 分散時に用いた曲線に x_{i_m} と W_{i_m} を代入し，k 個の連立方程式を解き，秘密情報 S を復元する．S の復元の際には，Lagrange の補間公式を用いることが一般的である．

例 2.1 (多項式補間を用いる $(2,3)$-SSS) 秘密情報 $S = 9$ に対して，素数 $p = 19$ として $GF(19)$ 上の演算を考える．またゼロ以外の $GF(19)$ の元から $a_1 = 14$ をランダムに選択する．$W = S + a_1 x = 9 + 14x$ と置くとシェア *ID* $1, 2, 3$ に対応して

- $(1, 9 + 14 \cdot 1 \mod 19) = (1, 4)$
- $(2, 9 + 14 \cdot 2 \mod 19) = (2, 18)$
- $(3, 9 + 14 \cdot 3 \mod 19) = (3, 13)$

をシェアとして配布する．

復元時には例えば，シェア $(1, 4)$ とシェア $(2, 18)$ から

$$\Delta = \frac{18 - 4}{2 - 1} = 14$$

を得る．分散時に利用した曲線は $f(x) = S + \Delta x$ であることから

$$S = 18 - 2\Delta = -10 \equiv 9 \mod 19$$

を復元することができる．

また，シェア $(1, 4)$ とシェア $(3, 13)$ から復元する場合を考えると，$GF(19)$ 上で 2 の逆数は 10 であることから

$$\Delta = \frac{13 - 4}{3 - 1} = 9 \cdot 10 \mod 19 = 14$$

を得る．この場合も

$$S = 13 - 3\Delta = -29 \equiv 9 \mod 19$$

と同様に復元することができる．

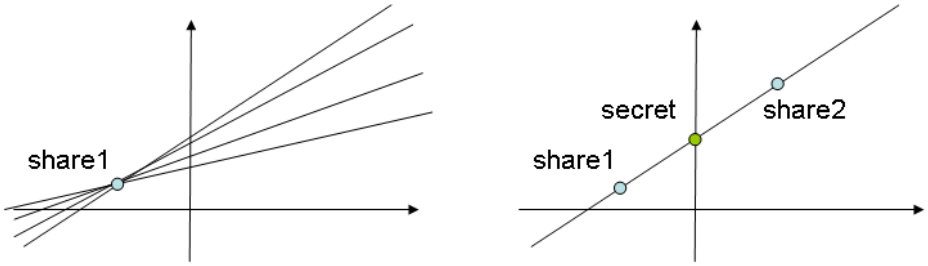

図 2.1: 多項式補間を用いる $(2, n)$-しきい値秘密分散法の原理

図 2.1 は $(2, n)$-しきい値秘密分散法の原理を表している．$k = 2$ であることから配布時に定める曲線は 1 次式であり直線となる．シェア $share1$ を持つエンティティは，手元にある情報からは直線を定めることができず，秘密情報を知りえることはない．一方で 2 つのシェア $share1, share2$ が分かると，直線が定まり W 軸切片にあたる秘密情報を知ることができる．

2.1.3 情報量的表現と理想的な秘密分散方式

秘密情報 S を n 個の分散情報 $W = \{W_1, \ldots, W_i, \ldots, W_n\}(i = 1, \ldots, n)$ に符号化するとき，分散情報 W の部分集合 $\overline{W} = \{W_{i_m}\}(i_m = 1, \ldots, k)$ から S が完全に復号できるとき，部分集合 $\overline{W}$ をアクセス可能グループと呼ぶ．部分集合の情報量については以下のように情報理論的な表現を用いることができる．W の部分集合 X が持つエントロピーを $H(X)$ とし，秘密情報 S のエントロピーを $H(S)$ とおく．このとき秘密分散方式の条件は，X がアクセス可能グループでならば $H(S|X) = 0$, X がアクセス可能グループでないならば $H(S|X) > 0$ と書くことができる．

また任意のユーザに配布されたシェア U は $H(U) \geq H(T)$ を満たし，特に任意の U に対して $H(U) = H(T)$ を満たすときその秘密分散方式は理想的 (ideal) であると呼ぶ．上記で説明した多項式補間に基づく (k, n)-しきい値秘密分散法は理想的な方式である．

このとき，任意の相異なる k 個の分散情報 $W_{i_1}, W_{i_2}, \ldots, W_{i_k}$ から S が復元できることは $H(S|W_{i_1}, W_{i_2}, \ldots, W_{i_k}) = 0$ が成立していることを意味する．一方で任意の相異なる $k-1$ 個の分散情報 $W_{i_1}, W_{i_2}, \ldots, W_{i_{k-1}}$ からは一切 S に関する情報が漏れないことは $H(S|W_{i_1}, W_{i_2}, \ldots, W_{i_{k-1}}) = H(S)$ が成立していることを意味する．

2.2 (k, L, n)-ランプ型秘密分散法

分散情報 W_i を生成する多項式を以下のように変更することで，分散情報の小型化を実現する [22].

秘密情報を L 分割して $s = (s_0, s_1, \ldots, s_{L-1})$ とし，式を以下の様に定めて，分散情報 W_i を計算する．

$W_i = s_0 + s_1 x_i + \cdots + s_{L-1} x_i^{L-1} + a_L x_i^L + \cdots + a_{k-1} x_i^{k-1}$ 復号の際は (k, n)-しきい値秘密分散法と同様の手順で連立方程式を解き $s = (s_0, s_1, \ldots, s_{L-1})$ を求める．これにより分散情報のサイズを $1/L$ にすることができるというメリットがある．一方で，本方式では，しきい値である k 個未満の分散情報から段階的に秘密情報に関するデータの漏えいが生じるというデメリットがあり [23]，一部のデータ漏洩を許容する必要が生じる．

このとき，任意の相異なる L 個の分散情報からは一切 S に関する情報が漏れないこと，つまり $H(S|W_{i_1}, W_{i_2}, \ldots, W_{i_L}) = H(S)$ が成立している．また $W_{i_1}, W_{i_2}, \ldots, Wi_L$ から S が復元できることは $H(S|W_{i_1}, W_{i_2}, \ldots, W_{i_k}) = 0$ が成立していることを意味する．ちょうど $L+1$ 個から $k-1$ 個の分散情報 $W_{i_1}, W_{i_2}, \ldots, W_{i_s}$ $(L < m < k)$ からは一部の情報が漏れ出していることから $H(S) > H(S|W_{i_1}, W_{i_2}, \ldots, W_{i_m}) > 0$ が成立していることを意味する．

2.3 排他的論理和演算のみを用いた高速な (k, n)-しきい値秘密分散法

分散時，復号時に排他的論理和演算のみを用いることで秘密情報の分散・復元処理を高速に実現できる (k, n)-しきい値秘密分散法が提案されている [24] [25]．これらの方式は，分散情報のデータ長と秘密情報のデータ長は等しくなるというメリットを持つ．以降において，W_i などの i にあたるインデックスは明示しない限り素数 n_p を法としたものとする．また，希望する分散数 n が素数ではない場合，n よりも大きな素数 n_p を設定して (k, n_p)-しきい値秘密分散法を構築し，その中の n 個を用いることで目的を達成するため，ここでは $n = n_p$ として説明を行う．

2.3.1 分散アルゴリズム

1. 秘密情報 $M \in \{0,1\}^{(n-1)d}$ を d ビットごとに等分割し $n-1$ 個の部分秘密情報を生成する．ここで $M_0 \in \{0\}^d$ とおく．

$$M = M_1 || M_2 || \cdots || M_{n-1} \ (M_i \in \{0,1\}^d)$$

2. d ビットの独立乱数 R_β^α を全て独立に $(k-1)n-1$ 個生成する．

$$R_0^0, R_1^0, \ldots, R_{n-2}^0, R_0^1, \ldots, R_{n-2}^1, R_{n-1}^1, \ldots, R_0^{k-2}, \ldots, R_{n-1}^{k-2} \in \{0,1\}^d$$

3. 以下の式を用いて部分分散情報 $W_{i,j}$ を $(0 \le i \le n-1, 0 \le j \le n-2)$ においてそれぞれ生成する．

$$W_{i,j} = M_{i-j} \oplus (\oplus_{h=0}^{k-2} R_{h \cdot i+j}^h) \in \{0,1\}^d \ (0 \le i \le n-1, 0 \le j \le n-2)$$

4. $0 \le i \le n-1$ において各部分分散情報 $W_{i,0}, W_{i,1}, \cdots, W_{i,n-2}$ を連結して $(n-1)d$ ビットの分散情報 W_i を生成する．

$$W_i = W_{i,0}||W_{i,1}||\cdots||W_{i,n-2}$$

ここで， $W_i \in \{0,1\}^{(n-1)d}$ より $M \in \{0,1\}^{(n-1)d}$ と同じデータ長であり，理想的な秘密分散方式であることに留意する．

5. 分散情報 W_i を各エンティティに配布する．

2.3.2 復元アルゴリズム

$\{W_i\}_{0\leq i\leq n-1}$ のうち任意の k 個の分散情報 $W_{t_0},\ldots,W_{t_{k-1}}$ $(0 \leq t_0 \leq \cdots \leq t_{k-1} \leq n-1)$ から秘密情報を復元する場合を示す．

1. k 個のシェアを部分分散情報に分割する．

   $W_{t_0} \to W_{t_0,0}, W_{t_0,1}, \ldots, W_{t_0,n-2}$

   $\vdots$

   $W_{t_{k-1}} \to W_{t_{k-1},0}, W_{t_{k-1},1}, \ldots, W_{t_{k-1},n-2}$

2. ここで集まった全ての各部分分散情報を以下のように表し， $kn-2$ 元の 2 進数ベクトル $V_{(t_i,j)}$ を生成する．部分分散情報 $W_{t_i,j}$ の場合，

   $$T_{(k,n)} := (S_1,\ldots,S_{n-2},R_0^0,\ldots,R_{n-2}^0,R_0^1,\ldots,R_{n-1}^1,\ldots,R_0^{k-2},\ldots,R_{n-1}^{k-2})^T$$

   に対して

   $$W_{t_i,j} = V_{(t_i,j)} \cdot T_{(k,n)}$$

   を満たすように $V_{(t_i,j)}$ を定める．

   例えば $k=4, n=5$ にて $W_{2,1}$ から以下のように定めることができる．

   $$W_{2,1} = M_4 \oplus R_1^0 \oplus R_3^1 \oplus R_0^2$$

   $$T_{(3,5)} = (M_1,\ldots,M_4,R_0^0,\ldots,R_3^0,R_0^1,\ldots,R_4^1,R_0^2,\ldots,R_4^2)^T$$

   $$V_{(2,1)} = (0001\ 0100\ 00010\ 10000)$$

3. 前ステップにおける $V_{(t_0,0)},\ldots,V_{(t_{k-1},n-2)}$ の $k(n-1)$ 個のベクトルから以下の 2 進数の $k(n-1)\times(k\cdot n-2)$ の行列 $G^{(k,n)}_{(t_0,\ldots,t_{k-1})}$ を生成する．

   $$G^{(k,n)}_{(t_0,\ldots,t_{k-1})} = (V_{(t_0,0)},\ldots,V_{(t_0,n-2)},\ldots,V_{(t_{k-1},0)},\ldots,V_{(t_{k-1},n-2)})^T$$

4. 集まったすべての部分分散情報を $k(n-1)$ 元のベクトル $W_{(t_0,\ldots,t_{k-1})}$ と表す．

$$W_{(t_0,\ldots,t_{k-1})} := (W_{(t_0,0)},\ldots,W_{(t_0,n-2)},\ldots,W_{(t_{k-1},0)},\ldots,W_{(t_{k-1},n-2)})^T$$
$$W_{(t_0,\ldots,t_{k-1})} = M^{(k,n)}_{(t_0,\ldots,t_{k-1})} \cdot R_{(k,n)}$$

ここで，行列 $G^{(k,n)}_{(t_0,\ldots,t_{k-1})}$ を Gauss-Jordan の消去法 (掃き出し法) を用いて対角化処理を行う．これによって，全ての部分秘密情報に該当する部分を求める．

5. 全ての部分秘密情報を連結して秘密情報を復元する．

$$M = M_1||M_2||\cdots||M_{n-1}$$

例 2.2 (排他的論理和演算のみを用いた XOR-$(4,5)$-SSS [24]) $n_p = n = 5$ とおき，秘密情報 M を $M = M_1||M_2\,||M_3||M_4(n^{'} = 4)$ のように *4* つの部分に分割し，$M_0 \in \{0\}^d$ とおく．ここで各シェア $W_i = W_{i,0}||W_{i,1}||\cdots\,||W_{i,n-2}(0 \le i \le n-1)$ を以下のようにおく．

	$j=0$	$j=1$	$j=2$	$j=3$
$W_{0,j}$	$M_0 \oplus R^0_0 \oplus R^1_0 \oplus R^2_0$	$M_1 \oplus R^0_1 \oplus R^1_1 \oplus R^2_1$	$M_2 \oplus R^0_2 \oplus R^1_2 \oplus R^2_2$	$M_3 \oplus R^0_3 \oplus R^1_3 \oplus R^2_3$
$W_{1,j}$	$M_4 \oplus R^0_0 \oplus R^1_1 \oplus R^2_2$	$M_0 \oplus R^0_1 \oplus R^1_2 \oplus R^2_3$	$M_1 \oplus R^0_2 \oplus R^1_3 \oplus R^2_4$	$M_2 \oplus R^0_3 \oplus R^1_4 \oplus R^2_0$
$W_{2,j}$	$M_3 \oplus R^0_0 \oplus R^1_2 \oplus R^2_4$	$M_4 \oplus R^0_1 \oplus R^1_3 \oplus R^2_0$	$M_0 \oplus R^0_2 \oplus R^1_4 \oplus R^2_1$	$M_1 \oplus R^0_3 \oplus R^1_0 \oplus R^2_2$
$W_{3,j}$	$M_2 \oplus R^0_0 \oplus R^1_3 \oplus R^2_1$	$M_3 \oplus R^0_1 \oplus R^1_4 \oplus R^2_2$	$M_4 \oplus R^0_2 \oplus R^1_0 \oplus R^2_3$	$M_0 \oplus R^0_3 \oplus R^1_1 \oplus R^2_4$
$W_{4,j}$	$M_1 \oplus R^0_0 \oplus R^1_4 \oplus R^2_3$	$M_2 \oplus R^0_1 \oplus R^1_0 \oplus R^2_4$	$M_3 \oplus R^0_2 \oplus R^1_1 \oplus R^2_0$	$M_4 \oplus R^0_3 \oplus R^1_2 \oplus R^2_1$

復元時には，例えば W_0, W_1, W_2, W_4 の *4* つのシェアが入手可能な場合，以下のように復元することが可能である．

- $M_0 = W_{0,0} \oplus W_{0,1} \oplus W_{0,2} \oplus W_{0,3} \oplus W_{1,3} \oplus W_{2,1} \oplus W_{2,3} \oplus W_{4,0} \oplus W_{4,2} \oplus W_{4,3}$
- $M_1 = W_{0,1} \oplus W_{0,2} \oplus W_{0,3} \oplus W_{1,0} \oplus W_{1,1} \oplus W_{1,2} \oplus W_{2,0} \oplus W_{4,3}$
- $M_2 = W_{0,2} \oplus W_{0,3} \oplus W_{1,1} \oplus W_{1,2} \oplus W_{2,3} \oplus W_{4,1}$
- $M_3 = W_{0,3} \oplus W_{1,2} \oplus W_{2,0} \oplus W_{2,2} \oplus W_{4,0} \oplus W_{4,3}$

次章以降の構成

以降の章では特に $k = 2$ の場合を取り扱っていく．$k = 2$ とした場合の $(2,n)$-しきい値秘密分散法は，分散処理を暗号化，復元処理を復号と捉えることができる．つまりシェアそのものが暗号文でもあり鍵データでもある，と考えることができる．

また $k = 2$ の場合でも，上記で触れたように別のシェアと突き合わせることによって，必ずしも秘密情報が復元されないアクセス構造を考え，$(2,n)$-しきい値秘密分散法の拡張についても検討していく．

第3章　排他的論理和演算のみを用いて構成する秘密分散法(XOR-SSS)

3.1　排他的論理和演算のみで構成される秘密分散法

排他的論理和演算のみで構成される (k,n)-しきい値秘密分散方式 (XOR-SSS) は藤井, 多田ら [28, 29, 30]，栗原ら [31, 24] によって独立に提案されている．シンプルな具体例として [28] に記載の XOR-$(2,3)$-SSS について説明する．

例 3.1 (XOR-$(2,3)$-SSS) 秘密情報 $M = M_1||M_2$（M_i のサイズは d ビット）に対して d ビットデータ R_0, R_1 を生成し，シェア $W_i(i = 0, 1, 2)$ を

W_0	$(M_0 \oplus R_0) \,\|\|\, (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \,\|\|\, (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \,\|\|\, (M_1 \oplus R_1)$

とおくことで構成可能である．ただし || はデータの連結を，$\oplus$ はビットごとの排他的論理和を示し M_0 は各ビットが全て *0* で構成されているものとする．

本方式は，各シェアサイズが分散対象のデータサイズと同じとなる理想的な秘密分散法式であることも分かる．

3.1.1　データ暗号化と秘密分散処理が可換な方式

XOR-SSS を用いる場合，M_i として暗号化データを代入，つまりキーストリーム K_i との XOR 演算結果 $K_i \oplus M_i$ を代入する（ストリーム暗号やブロック暗号にて CTR モードを利用する）ことにより，暗号化処理と秘密分散処理が可換である．例えば，図 1.8 における E 型（暗号化→秘密分散）の場合，XOR-$(2,3)$-SSS を構成する例 3.1 においては，暗号処理 $M_i \to K_i \oplus M_i$ を施したあと秘密分散処理を行う，つまり，

W_0	$((K_0 \oplus M_0) \oplus R_0) \,\|\|\, ((K_2 \oplus M_2) \oplus R_1)$
W_1	$((K_1 \oplus M_1) \oplus R_0) \,\|\|\, ((K_0 \oplus M_0) \oplus R_1)$
W_2	$((K_2 \oplus M_2) \oplus R_0) \,\|\|\, ((K_1 \oplus M_1) \oplus R_1)$

とシェアを構成すればよい．ここで，暗号化と秘密分散処理を行う主体が同一である場合には K_0 を各ビットが 0 の NULL データとしても一般性を失わない．E 型復元（復号→秘密復

元）処理においては XOR 演算が可換であることから $((K_i \oplus M_i) \oplus R_i) \oplus K_i = M_i \oplus R_i$ が成立することにより，本方式が正しく動作することがわかる．

また，F 型（秘密分散処理→暗号化）の場合，

W_0	$(M_0 \oplus R_0) \oplus K'_{0L} \,\|\|\, (M_2 \oplus R_1) \oplus K'_{0R}$
W_1	$(M_1 \oplus R_0) \oplus K'_{1L} \,\|\|\, (M_0 \oplus R_1) \oplus K'_{1R}$
W_2	$(M_2 \oplus R_0) \oplus K'_{2L} \,\|\|\, (M_1 \oplus R_1) \oplus K'_{2R}$

とシェアを構成し，F 型復元（秘密復元→復号）処理においては $(M_i \oplus K'_{**}) \oplus K'_{**} = M_i$ が成立することにより，E 型復元と同様に本方式が正しく動作することがわかる．

暗号化処理と復号処理の主体が同一かどうかによって鍵データ K'_{**} をどのように配備するか復元処理を顧客側もしくはトラストなクラウドサービスで行うケースにおいては M_i に関わる K'_{**} を同じように構成することで秘密復元と復号を同時に行うことができるメリットを有する．

上記は XOR-$(2,3)$-SSS のみについて具体的に扱ったが，任意の XOR-(k,n)-SSS [30, 24] においても，分散・復元フェーズにおいて XOR 演算のみを用いているため容易に拡張可能である．

3.2 新しい XOR-SSS の構成方式

3.2.1 提案方式 1（基本方式）

本節にて，従来手法の前提と同じく XOR 演算だけを用いた新しい構成方式について取り上げる．ここでターゲットデータ M の分割数を $n^{'}$ とおくと，従来手法では素数 n_p に対して $n^{'} = n_p - 1 = n - 1$ でしか構成できないという制約があった．まず，提案方式に基づいて構成した XOR-$(2,4)$-SSS with $n^{'} = 2 \neq n_p - 1$ について具体的構成方式について紹介する．

例 3.2 (トイケース XOR-$(2,4)$-SSS) $M = M_1 \,||\, M_2\ (n^{'} = 2), M_0 \in \{0\}^d, n_p = 3$

W_0	$(M_0 \oplus R_0)$	$\|\|$	$(M_1 \oplus M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0)$	$\|\|$	$(M_0 \oplus R_1)$
W_2	$(M_1 \oplus M_2 \oplus R_0)$	$\|\|$	$(M_1 \oplus R_1)$
W_3	$(M_2 \oplus R_0)$	$\|\|$	$(M_2 \oplus R_1)$

$F()$ を複数のシェアを入力すると，各パートで復元されるデータを出力する関数と考える．例えば $F(\{W_0, W_1\}) = \{M_1(\text{左パート}), M_1 \oplus M_2(\text{右パート})\}$ となり結果的に M_1, M_2 の両方を復元することが可能となる．以下，他のシェアについても列挙すると

- $F(\{W_0, W_2\}) = \{M_1 \oplus M_2, M_2\}$,
- $F(\{W_0, W_3\}) = \{M_2, M_1\}$,
- $F(\{W_1, W_2\}) = \{M_2, M_1\}$,

- $F(\{W_1, W_3\}) = \{M_1 \oplus M_2, M_2\}$,
- $F(\{W_2, W_3\}) = \{M_1, M_1 \oplus M_2\}$

となり，全ての場合において M_1, M_2 の両方を復元できることが分かる．

提案方式 1 の効率性

W_1 の左パート $M_1 \oplus M_2 \oplus R_0$ については XOR 演算が増加しているようにみえるが，実際には W_2 にて $M_1 \oplus R_0$ を算出していることからこれを利用することが可能である．分散時には（d ビットの）XOR 演算 6 回，復元時は高々XOR 演算 3 回で演算可能である．従来方式 [28] においては XOR-$(2, 5)$-SSS with $n^{'} = 4$ の一部を利用することになり，分散時には (d/2 ビットの）XOR 演算 16 回，復元時は高々XOR 演算 7 回が必要となることから，わずかながら提案方式が有利である．

以下，任意の n に対する XOR-$(2, n)$-SSS の構成方法について述べる．

構成方式 3.3 素数 n_p に対して $n = n_p + 1$ となる *XOR-$(2, n)$-SSS with $n^{'} = n_p - 1$* をシェア $W_i (i = 0, \ldots, n_p)$ は $n^{'}$ 個 $W_{ij} (j = 0, \ldots, n^{'} - 1)$ のパーツ の連結と考える．一般の n に対して $n = n_p + 1$ となる n_p が存在しない場合には，n_p より大きな素数を選択することで当該 n をカバーすることができる．

ターゲットデータ M は $M_1, \ldots, M_{n'}$ の連結で各データ長を *d* ビット，$M_0 \in \{0\}^d$ とする．このとき W_{ij} を以下のようにおく．

- $W_{i0} := M_1 \oplus M_{n'+2-i} \oplus R_0 (i = 1, \ldots, n^{'})$
  *(*ただし 添字は $\mod n_p$ とするため $W_{00} = R_0, W_{10} = M_1 \oplus R_0$*)*
- $W_{0j} := M_1 \oplus M_{j+1} \oplus R_j \ \ (j = 1, \ldots, n^{'} - 1)$
- $W_{1j} := W_{0,j-1} \oplus R_{j-1} \oplus R_j \ \ (j = 1, \ldots, n^{'} - 1)$
- $W_{ij} := W_{i-1,j-1} \oplus R_{j-1} \oplus R_j \ \ (i = 1, \ldots, n^{'}, j = 1, \ldots, n^{'} - 1)$
- $W_{n'+1,j} := M_2 \ \oplus, \ldots, \oplus \ M_{n'} \oplus R_j \ \ (j = 0, \ldots, n^{'} - 1)$

定理 3.4 上記構成方法により *XOR-$(2, n)$-SSS* を実現することができる．

Proof. まず $W_{n'+1}$ 以外の 2 つのシェアから復元するケースを考える．$k = 2$ であることから復元時には，各パートのランダムデータ R_* がキャンセルされて M_* の線形結合のデータが $n^{'}$ 個算出される．

特に $W_{\alpha,\alpha}$ $(0 \leq \alpha \leq n^{'} - 1)$ は $M_0 + R_\alpha$ であり，足し合わされるシェアは $M_1 + M_*$ がそのまま算出される．そのほか，特に $W_{\alpha,\alpha+1}$ $(0 \leq \alpha \leq n^{'} - 1)$ は $M_1 + R_\alpha$ であり，また，足し合わされるシェアは M_0 もしくは $M_1 + M_*$ であることから確実に一つのパラメータは値が確定する．これを M_β とおくこととする．それ以外の M_* の線形結合のデータは $n^{'} - 1$ 個あり，不明の値は M_β 以外の $n^{'} - 1$ 個あることから，それらが線形独立であれば算出可能であ

る．一方で添字の巡回は $\bmod n_p$ であることから互いに独立したデータが得られることが保証される．よって $\{M_i\}\ (1 \leq i \leq n^{'})$ を復元可能である．

次に復元に要するシェアとして $W_{n^{'}+1}$ を含む場合を考える．もう片方のシェアと足し合わせた場合，必ず $M_0 + R_*$ と $M_1 + R_*$ の両方を含むことから，$\sum_{1 \leq i \leq n^{'}} M_i$ と $\sum_{2 \leq i \leq n^{'}} M_i$ を得ることが分かる．故に，確実に M_1 が復元される．そして M_1 以外の M_* の線形結合のデータは $n^{'} - 1$ 個あり不明の値は M_1 以外の $n^{'} - 1$ 個あることから上記と同様の議論により，$\{M_i\}$ $(2 \leq i \leq n^{'})$ も復元可能である． ∎

3.2.2 提案方式 2(シェア復元)

ハッシュ関数による「シェア復元」を行う方式を提案する．基本的かつシンプルなアイデアは以下のとおりである：他のシェアから一方向性ハッシュ関数を用いて当該シェアを復元する．ここで一方向性ハッシュ関数は SHA-2,SHA-3 などの dedicated なハッシュ関数を用いることを想定する．従来の XOR-(n, k)-SSS の構成方式に着目すると，シェアとして元の秘密情報に依存せずランダムデータのみで構成されている箇所があることがわかる．このランダムデータを構成する際に他のシェアを入力値としてハッシュ関数を用いて構成することで当該シェアの配布を行わない，つまりシェアのサイズを秘密分散対象データのサイズよりも小さくすることを可能にする．

従来の XOR-$(2, 3)$-SSS の一例を考える．

W_0	$(M_0 \oplus R_0) \,\|\|\, (M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \,\|\|\, (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \,\|\|\, (M_1 \oplus R_1)$

このとき R_0, R_1 はランダムデータであることからそのものを配布するのではなく，他のシェアから生成することを想定する．つまり配布するシェアとしては

W_0	$\|\| \,(M_2 \oplus R_1)$
W_1	$(M_1 \oplus R_0) \,\|\|\, (M_0 \oplus R_1)$
W_2	$(M_2 \oplus R_0) \,\|\|\, (M_1 \oplus R_1)$

としておき，W_{00}（W_0 の左パート）は $W_{01} = M_2 \oplus R_1$（W_0 の右パート）からシェアホルダーが復元時に算出する方式である．このケースでは W_0 の配布データサイズを半分にすることを可能とする．特にターゲットデータサイズの大きい場合に有効であると考えられる．

具体的な構成方法としては R_1 は従来例と同じくランダムデータとして構成し，ハッシュ関数 $H()$ とすると，共通鍵を $H(W_{01})$ として AES の CTR モードを利用してランダムデータを d ビット算出して R_0 として利用する方式が考えられる．

3.2.3 提案方式 3(元データ復元)

前節ではハッシュ関数によるシェア復元を行ったが，この考え方を用いて「元データ復元」を行う方式について説明する．具体的にはダミーデータとして元データの暗号化データを用いる方式である．

$k = 2, n = 4$ において復元に失敗するケース

$n^{'}(= n_p - 1) = 3$ のトイケースを考える．以下は n_p が素数ではないため失敗事例のように見受けられる．

W_0	$(M_0 \oplus R_0) \,\|\|\, (M_3 \oplus R_1) \,\|\|\, (M_2 \oplus R_2)$
W_1	$(M_1 \oplus R_0) \,\|\|\, (M_0 \oplus R_1) \,\|\|\, (M_3 \oplus R_2)$
W_2	$(M_2 \oplus R_0) \,\|\|\, (M_1 \oplus R_1) \,\|\|\, (M_0 \oplus R_2)$
W_3	$(M_3 \oplus R_0) \,\|\|\, (M_2 \oplus R_1) \,\|\|\, (M_1 \oplus R_2)$

しかしこの状況で復元されるシェアを考えると以下のようになることがわかる．

- $F(\{W_0, W_1\}) = \{M_1, M_3, M_2 \oplus M_3\}$,
- $F(\{W_0, W_2\}) = \{M_2, M_1 \oplus M_3, M_2\}$,
- $F(\{W_0, W_3\}) = \{M_3, M_2 \oplus M_3, M_1 \oplus M_2\}$,
- $F(\{W_1, W_2\}) = \{M_1 \oplus M_2, M_1, M_3\}$,
- $F(\{W_1, W_3\}) = \{M_1 \oplus M_3, M_2, M_1 \oplus M_3\}$,
- $F(\{W_2, W_3\}) = \{M_2 \oplus M_3, M_1 \oplus M_2, M_1\}$

ここで $\{W_0, W_2\}$ と $\{W_1, W_3\}$ 以外の組は全ての M_i が復元可能であること，これら 2 例については M_2 および $M_1 \oplus M_3$ までは復元できていることがわかる．

そこで R_0 として $ENC(H(M_1 \oplus M_3); M_1)$ を R_1 として $ENC(H(M_1 \oplus M_3); M_3)$（ただし $ENC(key; data)$ は共通鍵 key を用いて data を AES などにより暗号化したデータ）とおく．$\{W_0, W_2\}$ を用いた復元時には $M_1 \oplus M_3$ から共通鍵を導出して M_1 を復元し，さらに M_3 を復元することが可能となる（$\{W_1, W_3\}$ においても同様）．実際に実装を行う際には，M_i のサイズをブロック長に揃えるか，暗号化データ長を丁度 d ビットにするためにパディング方式に工夫が必要となる．また本方式はターゲットデータが短い場合には共通鍵のエントロピーが十分取れないケースが発生するため安全に利用することができない点に留意する．

3.2.4 $(3, 2, 4)$-ランプ型秘密分散方式の構成

ターゲットデータに関する情報が部分的に漏洩することを許すことでシェアサイズをターゲットデータの 1/L に抑えることができるというメリットを持つ (k, L, n)-ランプ型秘密分散法 [22, 32, 33] がある．高荒ら [34] は XOR-SSS を一般的にランプ型に変換する方式を提案している．また松本ら [35] は XOR 演算だけを用いた効率的な $(3, 2, 4)$-ランプ型秘密分散方式をヒューリスティックに構成している．

本節では前節のアイデアを基にシンプルで効率的な $(3, 2, 4)$-ランプ型秘密分散方式の構成について提案する．ここでは $n' = 2$ とする．

W_0	R_0
W_1	$M_1 \oplus M_2 \oplus R_0$
W_2	R_1
W_3	$M_1 \oplus M_2 \oplus R_1$

ただし $R_0 := ENC(key; M_1)$, $R_1 := ENC(key; M_2)$, $M_{12} := M_1 \oplus M_2$, $R_{01} := R_0 \oplus R_1$, $key := H(M_{12}||R_{01}||M_{12} \oplus R_{01})$ とおく．このとき

- $F(\{W_0, W_1\}) = \{M_{12}\}$
- $F(\{W_0, W_2\}) = \{R_{01}\}$
- $F(\{W_0, W_3\}) = \{M_{12} \oplus R_{01}\}$
- $F(\{W_1, W_2\}) = \{M_{12} \oplus R_{01}\}$
- $F(\{W_1, W_3\}) = \{R_{01}\}$
- $F(\{W_2, W_3\}) = \{M_{12}\}$

となり 2 つのシェアからは一部の情報しか復元できない．

一方で 3 つのシェアからは $M_{12}, R_{01}, M_{12} \oplus R_{01}$ のデータが全て復元できるためこれらからハッシュ関数を用いて key を復元し，R_0 または R_1 から それぞれ M_1 または M_2 を復元することでターゲットデータの復元を行うことができる．つまり (3,2,4)-ランプ型秘密分散方式を構成したことを意味する．また各シェアのデータ長は，元の秘密情報の $1/L = 1/2$ で構成している点にも留意する．

3.3 XOR-SSS における同型性の導入と代数的解析

3.3.1 XOR-SSS のマトリクス表現

$k = 2$ を満たす XOR 演算だけを用いた秘密分散方式の構成方法を考察していく．ターゲットデータ M の分割数を $n^{'}$ とすると XOR-$(2, n)$-SSS のシェア $W_i(i = 0, \ldots, n)$ を次のマトリクスで表現することとする．ここで $n^{''} := n^{'} - 1$，$W_i = W_{i0} \,||\, \ldots \,||\, W_{in^{''}}$ とする．

W_0	W_{00}	$\ldots$	$W_{0n^{''}}$
W_1	W_{10}	$\ldots$	$W_{1n^{''}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
W_i	W_{i0}	$\ldots$	$W_{in^{''}}$
$\ldots$	$\ldots$	$\ldots$	$\ldots$
W_n	W_{n0}	$\ldots$	$W_{nn^{''}}$

栗原らの方式のマトリクス表現

まず巡回置換行列を用いた XOR-$(2, n)$-SSS の構成方式について説明する．素数 n_p に対して $n = n_p$ となる XOR-$(2, n)$-SSS with $n^{'} = n_p - 1$ をシェア $W_i(i = 0, \ldots, n_p - 1)$ は $n^{'}$ 個 $W_{ij}(j = 0, \ldots, n^{'} - 1)$ のパーツ の連結と考える．ターゲットデータ M は $M_1, \ldots, M_{n'}$ の連結で各データ長を d ビット，$M_0 \in \{0\}^d$ とする．また M_0 と同じサイズのダミーデータ $R_i(i = 0, \ldots, n_p)$ をランダムに選択する．このとき W_{ij} を $M_{(j-i) \mod n_p} \oplus R_j$ とおく．

上記構成方式に基づいて構成される XOR-$(2, 3)$-SSS with $n^{'} = 2$ は以下の通りである．

例 3.5 (XOR-$(2, 3)$-SSS [24]) $M = M_1 || M_2\ (n^{'} = 2), M_0 \in \{0\}^d$

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$
W_1	$M_2 \oplus R_0$	$M_0 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_2 \oplus R_1$

さらに $n_p = 5$ のときに構成される XOR-$(2, 5)$-SSS with $n^{'} = 4$ は以下の通りである．

例 3.6 (XOR-$(2, 5)$-SSS [24])

W_0	$M_0 \oplus R_0$	$M_1 \oplus R_1$	$M_2 \oplus R_2$	$M_3 \oplus R_3$
W_1	$M_4 \oplus R_0$	$M_0 \oplus R_1$	$M_1 \oplus R_2$	$M_2 \oplus R_3$
W_2	$M_3 \oplus R_0$	$M_4 \oplus R_1$	$M_0 \oplus R_2$	$M_1 \oplus R_3$
W_3	$M_2 \oplus R_0$	$M_3 \oplus R_1$	$M_4 \oplus R_2$	$M_0 \oplus R_3$
W_4	$M_1 \oplus R_0$	$M_2 \oplus R_1$	$M_3 \oplus R_2$	$M_4 \oplus R_3$

提案方式 1（3.2.1 節）のマトリクス表現

素数 n_p に対して $n = n_p + 1$ となる XOR-$(2, n)$-SSS with $n^{'} = n_p - 1$ をシェア $W_i(i = 0, \ldots, n_p)$ は $n^{'}$ 個 $W_{ij}(j = 0, \ldots, n^{'} - 1)$ のパーツ の連結と考える．ターゲットデータ M は $M_1, \ldots, M_{n'}$ の連結で各データ長を d ビット，$M_0 \in \{0\}^d$ とする．また M_0 と同じサイズのダミーデータ $R_i(i = 0, \ldots, n_p)$ をランダムに選択する．

提案方式 1（3.2.1 節）に基づいて構成される XOR-$(2, 4)$-SSS with $n^{'} = 2 \neq n - 1 = 3$ は以下の通りである．

例 3.7 (XOR-$(2, 4)$-SSS (3.2.1 節)) $n_p = 3, M = M_1 || M_2\ (n^{'} = 2), M_0 \in \{0\}^d$

W_0	$M_0 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_1	$M_1 \oplus R_0$	$M_0 \oplus R_1$
W_2	$M_1 \oplus M_2 \oplus R_0$	$M_1 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_2 \oplus R_1$

さらに $n_p = 5$ のときに構成される XOR-$(2, 6)$-SSS with $n^{'} = 4$ は以下の通りである．ただし $M_{234} := M_2 \oplus M_3 \oplus M_4$ とおく．M_{234} は $\bigoplus_{t=1}^{n'} M_t \oplus M_1$ であり M_1 の補集合と考えることもできる．

例 3.8 (XOR-$(2, 6)$-SSS (3.2.1 節))

W_0	$M_0 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$	$M_1 \oplus M_3 \oplus R_2$	$M_1 \oplus M_4 \oplus R_3$
W_1	$M_1 \oplus R_0$	$M_0 \oplus R_1$	$M_1 \oplus M_2 \oplus R_2$	$M_1 \oplus M_3 \oplus R_3$
W_2	$M_1 \oplus M_4 \oplus R_0$	$M_1 \oplus R_1$	$M_0 \oplus R_2$	$M_1 \oplus M_2 \oplus R_3$
W_3	$M_1 \oplus M_3 \oplus R_0$	$M_1 \oplus M_4 \oplus R_1$	$M_1 \oplus R_2$	$M_0 \oplus R_3$
W_4	$M_1 \oplus M_2 \oplus R_0$	$M_1 \oplus M_3 \oplus R_1$	$M_1 \oplus M_4 \oplus R_2$	$M_1 \oplus R_3$
W_5	$M_{234} \oplus R_0$	$M_{234} \oplus R_1$	$M_{234} \oplus R_2$	$M_{234} \oplus R_3$

3.3.2 XOR-$(2, n)$-SSS における同型性の導入

上記例のようにシェアを表現するマトリクス $\{W_{ij}\}$ に対して，次のように同型性を定義する．

定義 3.9 (XOR-$(2, n)$-SSS における同型性) *XOR-$(2, n)$-SSS* Ψ の W_{ij} 成分がマトリクス表現されているとき，以下の操作に基づいて変形されたマトリクスから生成される *XOR-$(2, n)$-SSS* は Ψ と同型であるとする．

1. ある行を他の行と入れ替える
2. ある列を他の列と入れ替える
3. ある列の全てのサブシェアのそれぞれに対して同じデータを *XOR* で足し込む

(1) については配布されるシェアのインデックス（添字）が変更したのみであり (2) についてはシェアの各パートの順番が変更されたに過ぎず配布されるデータとしては同一である．(3) については足し込むランダムデータ R_i に変化が生じたに過ぎず，選択されたランダムデータが異なるだけであると解釈できる．つまり，上記操作を行ったとしても安全性を確保しつつ，別の異なる XOR-$(2, n)$-SSS を構成することが可能である．ただしシェア生成時の効率性については増減する場合があることは自明である．

次に例 3.5 をもとに実際の変形例を見ることにする．

例 3.10 (例 3.5 の変形例)

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1^{'}$
W_1	$M_2 \oplus R_0$	$M_1 \oplus R_1^{'}$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1^{'}$

上記例はサブシェア $W_{t1}(t = 0, \ldots, 2)$ に M_1 を足しこんだデータと同一である．ランダムデータとして R_1 ではなく $R_1^{'}$ と記載しているがランダムデータの選択には依存しないため以降は R_i と記載しても問題ない．次に 例 3.7 を変形して例 3.5 との拡張性を見い出す．下記例は例 3.7 においてサブシェア $W_{t1}(t = 0, \ldots, 2)$ に $M_1 \oplus M_2$ を足しこんだデータである．

例 3.11 (例 3.7 の変形例)

W_0	$M_0 \oplus R_0$	$M_0 \oplus R_1$
W_1	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus R_1$
W_2	$M_1 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$
W_3	$M_2 \oplus R_0$	$M_1 \oplus R_1$

これを例 3.10 と比較すると例 3.11 では W_1 が新たに追加された拡張方式であることがわかる．

XOR-SSS をシェアを表現する方式として上記例のマトリクス表現ではなく，各サブシェア W_{ij} を次のように $\mathbb{Z}_2^{n^{'}}$ の元として表現することとする．$W_{ij} = \bigoplus_{t=1}^{n^{'}} \alpha_t M_t$ のとき $w_{ij} = (\alpha_1, \ldots, \alpha_{n'}) \in \mathbb{Z}_2^{n^{'}}$ とベクトル表現を行う．このとき各列に同じように出現するランダムデータのパート R_i およびゼロデータ M_0 は無視しても一般性を失わない．

以下は例 3.11 をベクトル表現に変換したものである．

例 3.12 (例 3.11 のベクトル表現)

W_0	$(0, 0)$	$(0, 0)$
W_1	$(1, 1)$	$(0, 1)$
W_2	$(1, 0)$	$(1, 1)$
W_3	$(0, 1)$	$(1, 0)$

XOR-$(2, 4)$-SSS のトイケースを例 3.12 のようにベクトル表現したとき，各列のベクトルが $n^{'}$ 次元ベクトル空間を構成していることが分かる．具体的には例 3.12 において $w_{10} = w_{20} + w_{30}$, $w_{11} = w_{21} + w_{31}$ を満たしている．ここで $+$ は $\mathbb{Z}_2^m$ 上の加算を意味する．

この例に見られるように m 次ベクトル空間から XOR-$(2, 2^m)$-SSS を構成する可能性について本節にて議論していく．その準備のためはじめにいくつかの定義を行う．

$\mathbb{Z}_2^m$ を張る基底 $b = b^{(1)}, b^{(2)}, \ldots, b^{(m)}$ を考える．ここで $b^{(i)}$ は m 次元ベクトルである．このとき $\mathbb{Z}_2^m$ の元は $\sum_{i=1}^{m} \alpha_i b^{(i)}$ for $\alpha_i \in \mathbb{Z}_2$ と表現できる．

定義 3.13 (基底の和) *2* つの基底 b_i, b_j に対して基底の和を次のように定義する．

$$b_i + b_j := \{b_i^{(1)} + b_j^{(1)}, b_i^{(2)} + b_j^{(2)}, \ldots, b_i^{(m)} + b_j^{(m)}\}.$$

ここで $+$ は $\mathbb{Z}_2^m$ 上の加算を意味する．

注意 3.14 (XOR 演算と $\mathbb{Z}_2^m$ 上の加算の関係) *2* つの *m* 次元ベクトル $b^{(i)}, b^{(j)} \in \mathbb{Z}_2^m$ の $\mathbb{Z}_2^m$ 上の加算はビットごとの排他的論理和と同一視できる．例: $(0,0,1,1)+(0,1,0,1) = (0,1,1,0) \in \mathbb{Z}_2^4$.

3.3.3 2-伝播基底集合の定義とその性質

定義 3.15 (2-伝播基底集合) $\mathbb{Z}_2^m$ を張る基底 $\{b_i\}(i = 1, \ldots, l)$ の集合が以下の条件を満たすとき $\{b_i\}$ を *2*-伝播基底集合という： b_1 は全てゼロベクトルで構成される（便宜上 b_1 も基底と同一視する）．任意の異なる *2* つの基底 b_i, b_j に対して $b_i + b_j$ が $\mathbb{Z}_2^m$ の基底となる．

定義 3.16 (k-伝播基底) *3* 以上の k に対して *2*-伝播基底と同様に以下のように *k*-伝播基底を定義することが可能である．$\mathbb{Z}_2^m$ を張る基底 $\{b_i\}$ の集合が *k*-伝播基底であるとは，任意の異なる k 個の基底 $b_{i_1}, \ldots, b_{i_k}$ に対して $\sum_{j=i_1,\ldots,i_k} b_j$ が $\mathbb{Z}_2^m$ の基底となることと定義する．本論文では利用しないが，この概念を用いることで *XOR-*$(k, 2^m)$*-SSS* もしくはランプ型 *XOR-*$(k, 2^m)$*-SSS* を構成することができると考えられる．

定理 3.17 (2-伝播基底集合の位数) $\mathbb{Z}_2^m$ 上の *2*-伝播基底集合 $\{b_i\}$ の位数は最大 2^m である．

Proof. 2-伝播基底集合 $\{b_i\}$ に $2^m + 1$ 以上の元が存在した場合，$b_i^{(0)} = b_j^{(0)}$ となるベクトルが存在する．このとき基底の和 $b_i + b_j$ を考えると $(b_i + b_j)^{(0)}$ は ゼロベクトルであり定義である「$(b_i + b_j)$ は $\mathbb{Z}_2^m$ の基底である」という事実に反する．ゆえに $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ の位数は高々 2^m である． ∎

定義 3.18 (基底 b と基底行列 B) $\mathbb{Z}_2^m$ 上のベクトル集合 $b = b^{(1)}, b^{(2)}, \ldots, b^{(m)}$ に対して行ベクトルで構成された $m \times m$ 行列 B について，実数上の行列と同様に行列の階数 *(rank)* を定義可能である．つまり線形独立な行ベクトルの個数を $rank(B)$ と記載する．

注意 3.19 *2*-伝播基底集合 $\{b_i\}$ の各基底に対する基底行列を $\{B_i\}$ とすると，$rank(B_1) = 0$（b_1 はゼロベクトルの集合のため），$rank(B_i) = m(i = 2, \ldots, l), rank(B_i + B_j) = m(i, j = 1, \ldots, l)$（ただし $i \neq j$）を満たす．

定理 3.20 *2*-伝播基底集合 $\{b_i\}$ が存在すれば，任意の $\alpha_i \in \mathbb{Z}_2$ に対する $\sum_{i=1}^{m} \alpha_i b_i$ も *2*-伝播基底集合 $\{b_i\}$ に含むことができる．

Proof. 2-伝播基底集合 $\{b_i\}$ の各基底に対する基底行列を $\{B_i\}$ とする．このとき，任意の $\alpha_i \in \mathbb{Z}_2$ に対して $rank(\sum_{i=1}^{m} \alpha_i Bi) = m$ であることを示せばよい．基底行列 B_i に対して 基底行列 B_j を足す操作は行列の $rank$ が変わらないことから行列への基本操作が行われてい

る，すなわち B_i に対して両側から行列が掛けられていることを意味する．つまり $B_i + B_j = L_j B_i R_j$ となる 2 行列 L_j, R_j が存在する．一方で $B_j + B_i = L_i B_j R_i$ を満たす． $B_i + B_j = B_j + B_i$ であることから式変形を行うと $B_i = (L_j^{-1} L_i) B_j (R_i R_j^{-1})$ となり $B_i = L^{'} B_j R^{'}$ となる 2 行列 $L^{'}, R^{'}$ が存在する．よって $\sum_{i=1}^{m} \alpha_i B_i = B_{i_1} + B_{i_2} + \ldots, B_{i_t} = L_{i_2} B_{i_1} R_{i_2} + B_{i_3} + \ldots + B_{i_t} = \ldots = L B_{i_1} R$ となる 2 行列 L, R が存在し $rank(B_{i_1}) = m$ であることが分かる．

∎

補題 3.21 $\mathbb{Z}_2^m$ 上の *2*-伝播基底集合 $\{b_i\}$ の位数は 2^t という形となる．*2*-伝播基底集合 $\{b_i\}$ の中に *t* 個の生成基底 $\{c_i\}(i = 1, \ldots, t)$ を持ち各基底 b_i はこれらで張られるベクトル空間の元，つまり $b_i = \sum_{j=1}^{t} \alpha_j c_j$ を満たす．

注意 3.22 $\mathbb{Z}_2^m$ 上の *2*-伝播基底集合 $\{b_i\}$ の位数が 2^m の *optimal* な場合，任意の j に対して $\{b_i^{(j)}\}$ $(i = 1, \ldots, 2^m)$ はすべて異なるベクトルで構成される．

次に $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ から XOR-$(2, 2^m)$-SSS を構成することを示す．

定理 3.23 (XOR-$(2, 2^m)$-SSS の構成) *optimal* な $\mathbb{Z}_2^m$ 上の *2*-伝播基底集合 $\{b_i\}$ $(i = 1, \ldots, 2^m)$ が存在するとき，ベクトル表現 $\{w_{ij} = b_i^j\}$ $(i = 1, \ldots, 2^m, i = 1, \ldots, m)$ を持つ *XOR-$(2, 2^m)$-SSS* が存在する．

Proof. 2-伝播基底集合の定義から任意の異なる u, v に対して $b_u + b_v$ もまた基底となるため $w_1^* = w_{u1} + w_{v1}, \ldots, w_m^* = w_{um} + w_{vm}$ は $\mathbb{Z}_2^m$ 上の基底となる．相異なる 2 つのサブシェアの XOR 和 $W_u \oplus W_v$ の 第 l 成分は $\bigoplus_{s=1}^{m} w_l^{*(s)} M_s$ となる．ここで M_s に対する独立した連立方程式が m 個存在することになり $M_s (s = 1, \ldots, m)$ の全てを復元できることを意味する．∎

3.3.4 2-伝播基底集合の存在性について

定理 3.23 に示したように optimal な $\mathbb{Z}_2^m$ 上の 2-伝播基底集合 $\{b_i\}$ $(i = 1, \ldots, 2^m)$ の存在性を示せば XOR-$(2, 2^m)$-SSS が存在することがわかる．本節では定理 3.17 に示されるように最大位数 2^m となる 2-伝播基底集合の存在性について扱う．

小さい m に対する存在性確認シミュレーション

以下のようなプログラムを用いて存在性について確認を行った．

プログラム.

1. set $m > 1$ （次元 m を一意に決定）
2. set $b_1 := \{(0, \ldots, 0), \ldots, (0, \ldots, 0)\}$ （m 次元ゼロベクトルを m 個並べたもの）
3. set $b_2 := \{(1, \ldots, 0), (0, 1, \ldots, 0), \ldots, (0, \ldots, 1)\}$ （m 次元正規ベクトルを m 個並べたもの）
4. set c:= 3

5. ${b_c}^R \in (\mathbb{Z}_2^m)^m$ をランダムに選択する

6. check $rank(B_c + B_i)$ =?=m for all $i = 1, \ldots, c-1$

7. if YES then add b_c into $\{b_i\}$, set $c := c + 1$; if (c == 2^m) then break;

8. if NO then return step 5

3.3.5 具体的構成例

以下小さい位数における具体的構成例について示す．W_0 は ゼロベクトル基底に呼応し，$W_1, \ldots, W_m$ は 補題 3.21 の生成基底 c_i に対応する．全シェアは定理 3.23 に記載の $\bigoplus_{s=1}^{m} w_l^{*(s)} M_s$ で構成される．

例 3.24 ($m = 3$: XOR-$(2, 2^3)$-SSS)

W_0	$(0,0,0)$	$(0,0,0)$	$(0,0,0)$
W_1	$(1,0,0)$	$(0,1,0)$	$(0,0,1)$
W_2	$(0,1,1)$	$(1,0,0)$	$(0,1,0)$
W_3	$(1,1,0)$	$(0,1,1)$	$(1,0,0)$

上記ベクトル表現をマトリクス表現は以下の通りである．$n' = 3$ として $M = M_1||M_2||M_3$ と分割する．W_0から W_3 はベクトル表現をそのまま利用し W_4 は W_1, W_2 のベクトル表現の和から，W_5 は W_1, W_3 から，W_6 は W_2, W_3 から，W_7 は W_1, W_2, W_3 から生成している．

W_0	R_0	R_1	R_2
W_1	$M_1 \oplus R_0$	$M_2 \oplus R_1$	$M_3 \oplus R_2$
W_2	$M_2 \oplus M_3 \oplus R_0$	$M_1 \oplus R_1$	$M_2 \oplus R_2$
W_3	$M_1 \oplus M_2 \oplus R_0$	$M_2 \oplus M_3 \oplus R_1$	$M_1 \oplus R_2$
W_4	$M_1 \oplus M_2 \oplus M_3 \oplus R_0$	$M_1 \oplus M_2 \oplus R_1$	$M_2 \oplus M_3 \oplus R_2$
W_5	$M_2 \oplus R_0$	$M_3 \oplus R_1$	$M_1 \oplus M_3 \oplus R_2$
W_6	$M_1 \oplus M_3 \oplus R_0$	$M_1 \oplus M_2 \oplus M_3 \oplus R_1$	$M_1 \oplus M_2 \oplus R_2$
W_7	$M_3 \oplus R_0$	$M_1 \oplus M_3 \oplus R_1$	$M_1 \oplus M_2 \oplus M_3 \oplus R_2$

例 3.25 ($m = 4$: XOR-$(2, 2^4)$-SSS)

W_0	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$	$(0,0,0,0)$
W_1	$(1,0,0,0)$	$(0,1,0,0)$	$(0,0,1,0)$	$(0,0,0,1)$
W_2	$(1,1,0,0)$	$(1,0,0,0)$	$(0,0,1,1)$	$(0,0,1,0)$
W_3	$(0,0,1,1)$	$(1,0,0,1)$	$(0,1,1,0)$	$(0,1,0,0)$
W_4	$(0,1,0,1)$	$(0,1,1,0)$	$(1,1,0,0)$	$(1,0,0,0)$

例 3.26 ($m = 5$: XOR-$(2, 2^5)$-SSS)

W_0	$(0,0,0,0,0)$	$(0,0,0,0,0)$	$(0,0,0,0,0)$	$(0,0,0,0,0)$	$(0,0,0,0,0)$
W_1	$(1,0,0,0,0)$	$(0,1,0,0,0)$	$(0,0,1,0,0)$	$(0,0,0,1,0)$	$(0,0,0,0,1)$
W_2	$(0,0,1,0,0)$	$(1,0,0,0,0)$	$(0,1,0,0,0)$	$(0,0,0,1,1)$	$(0,0,0,1,0)$
W_3	$(1,1,0,0,0)$	$(1,0,0,0,1)$	$(0,0,0,1,1)$	$(0,0,1,1,0)$	$(0,0,1,0,0)$
W_4	$(0,0,0,1,0)$	$(0,0,0,1,1)$	$(1,0,0,0,0)$	$(0,1,1,0,0)$	$(0,1,0,0,0)$
W_5	$(0,1,1,1,1)$	$(0,1,1,0,0)$	$(0,0,0,0,1)$	$(1,0,1,0,1)$	$(1,0,0,0,0)$

例 3.27 ($m = 6$: XOR-$(2, 2^6)$-SSS)

W_0	$(0,0,0,0,0,0)$	$(0,0,0,0,0,0)$	$(0,0,0,0,0,0)$	$(0,0,0,0,0,0)$	$(0,0,0,0,0,0)$	$(0,0,0,0,0,0)$
W_1	$(1,0,0,0,0,0)$	$(0,1,0,0,0,0)$	$(0,0,1,0,0,0)$	$(0,0,0,1,0,0)$	$(0,0,0,0,1,0)$	$(0,0,0,0,0,1)$
W_2	$(0,0,0,0,0,1)$	$(1,0,0,0,0,1)$	$(0,1,0,0,0,0)$	$(0,0,1,0,0,0)$	$(0,0,0,1,0,0)$	$(0,0,0,0,1,0)$
W_3	$(0,0,0,0,1,0)$	$(0,0,1,0,0,0)$	$(1,0,0,0,0,0)$	$(0,1,0,0,0,0)$	$(0,0,0,0,1,1)$	$(0,0,0,1,0,0)$
W_4	$(0,0,0,1,0,1)$	$(1,0,0,0,1,1)$	$(1,1,0,0,0,1)$	$(0,1,0,0,0,1)$	$(0,0,1,0,0,1)$	$(0,0,1,0,0,0)$
W_5	$(0,0,1,0,1,0)$	$(0,0,1,1,0,0)$	$(0,0,1,0,1,1)$	$(1,0,0,0,0,0)$	$(0,1,0,0,1,0)$	$(0,1,0,0,0,0)$
W_6	$(0,1,0,0,0,1)$	$(0,1,1,1,0,1)$	$(1,0,1,1,0,1)$	$(0,1,0,0,1,0)$	$(1,0,0,1,0,0)$	$(1,0,0,0,0,0)$

上記例のように XOR-$(2, 2^6)$-SSS の存在性が示されている一方で，リードソロモン符号などで用いられる $\mathbb{Z}_2$ 上の既約多項式との関連性について指摘 [37] がなされている．$\mathbb{Z}_2$ の元を係数とする多項式を $\mathbb{Z}_2$ 上の既約多項式で除算した余りの集合はガロア拡大体 $GF(2^m)$ を構成し，拡大体の元が $GF(2)$ 上のベクトル空間の基底（正規基底）は常に存在することが知られている．例えば $GF(2^4)$ の場合には $x^4 + x + 1$ が既約多項式に該当し前述の例もここから構成できる．

本提案方式の優位性

本提案方式では，例えば XOR-$(2, 2^6)$-SSS が構成できる．このとき秘密情報を 6 つのパートに分け，6 変数の連立方程式を解いて復元することができる．同じ XOR-$(2, 2^6)$-SSS を従来の栗原らによる方式で構成する場合との比較を行うと，2^6 を超える最も小さな素数は 37 であり，秘密情報を 36 個のパートに分ける必要がある．このとき 36 変数の連立方程式を解く必要があるため，本方式に優位性があることが分かる．

3.4 秘密計算法への適用

クラウドをデータのアーカイブ先として利用する静的な利用に対して，クラウドに計算を委託し，暗号化状態のまま計算して結果を得るという動的な利用方法にもニーズがシフトしつつある．近年多発する漏洩事件・事故に起因して，プライバシドリブンの考え方が定着しつつある．入力データを秘匿して秘密裏に演算を行って出力を行う秘密計算法は，CPU・クラウドのマシンパワーも向上していることから現実的な計算量で解決する方式が提案され続

けている．特に，前述した Fully Homomorphic Encryption よりも軽量で，制約はあるものの複数のエンティティ（例えばクライド，サーバ，場合によりユーザ自身の PC）間の協調計算により秘密計算を行うマルチパーティプロトコルが注目されている．医療データなどよりセンシティブなデータに対しても正しくかつ安全に活用できる実験結果も常に公表され続けている．

Shamir による多項式補間を用いる (k,n)-SSS をベースに秘密計算を行う方式が Ben-Or らによって提案されている [38]．その原理は下記のように非常に単純である．2 つの秘密情報 $S_f = f(0), S_g = g(0)$ がそれぞれ曲線 $y = f(x), y = g(x)$ を用いて分散されているとき，シェア ID x_i が割り振られているエンティティには $f(x_i), g(x_i)$ がシェアとして配布されている．このとき $S_f + S_g$ を秘密計算したい場合には，各エンティティが $f(x_i) + g(x_i)$ を計算することで $(f+g)(x_i)$ を計算していることとなり，各エンティティから事前計算された $(f+g)(x_i)$ を用いて復元処理を行うと，$(f+g)(0)$ つまり $f(0) + g(0) = S_f + S_g$ を最終的に得ることができる．本論文でも，これを基本アイデアとして用いる．

軽量な提案方式としては例えば，千田らによる $(2,3)$-SSS を用いた方式 [39] [40] [41] や XOR-SSS を用いる方式 [42]，実数演算上のナイーブな提案 [43] などが挙げられる．これらに共通するのは，数値を事前に数値を分解することによる秘密分散処理を行っており，計算代行者 U_∞ に各エンティティから出された結果を集約して，利用者が計算結果を得るというトポロジーモデルである．

このとき，秘密分散後のシェアデータを持つエンティティ U_* から 計算代行者 U_∞ に送信されるデータを第三者が収集することによって，もしくは計算代行者 U_∞ の計算結果を第三者が不正に取得，さらに計算代行者 U_∞ 自身の不正などによって安全に利用できない問題が生じる．本提案はこの問題に対する解決方法についても触れる．

提案方式は計算対象の入力に対して計算結果を得るために，分散エンティティ $U_i (0 \leq i \leq n-1)$ と，計算結果を集約する 計算代行者 U_∞ の協調計算を行う方式である．依頼者は入力に対して XOR-(k,n)-SSS で秘密分散を行い，各分散エンティティに事前に配布しておくことが前提となる．計算時には，各分散エンティティが持つ複数のシェアデータから演算を行い計算代行者 U_∞ と共有する．計算代行者 U_∞ は結果を依頼者に返却するものとする．

3.4.1 XOR 演算（ナイーブな方式）

XOR 演算は可換であることからほぼ自明な方式となる．依頼者は入力 A と B を事前に各エンティティに分散しておき，計算時には $A \oplus B$ を計算代行者から得る方式である．具体的な例として以下の XOR-$(3,4)$-SSS を用いて説明する．

例 3.28 (XOR-$(3,4)$-SSS)

W_0	$M_0 \oplus R_0 \oplus R_{01}$	$M_0 \oplus R_1 \oplus R_{11}$
W_1	$M_1 \oplus M_2 \oplus R_0 \oplus R_{11}$	$M_2 \oplus R_1 \oplus R_{21}$
W_2	$M_1 \oplus R_0 \oplus R_{21}$	$M_1 \oplus M_2 \oplus R_1 \oplus R_{31}$
W_3	$M_2 \oplus R_0 \oplus R_{31}$	$M_1 \oplus R_1 \oplus R_{01}$

エンティティU_i に対して上記の分散規則に基づいて A, B それぞれの W_i を配布する．例えば U_0 に対しては $A_0 \oplus R_0^A \oplus R_{01}^A$, $A_0 \oplus R_1^A \oplus R_{11}^A$ と $B_0 \oplus R_0^B \oplus R_{01}^B$, $B_0 \oplus R_1^B \oplus R_{11}^B$ が配布されることになる．

計算依頼が行われた際には，各エンティティにおいて A, B それぞれから生成されたシェアを XOR 演算を行い 計算代行者 U_∞ に送信する．例えば U_0 では，$(A_0 \oplus R_0^A \oplus R_{01}^A) \oplus (B_0 \oplus R_0^B \oplus R_{01}^B)$, $(A_0 \oplus R_1^A \oplus R_{11}^A) \oplus (B_0 \oplus R_1^B \oplus R_{11}^B)$ を計算することになる．これは整理すると以下のように書き換えることができる：$(A_0 \oplus B_0) \oplus R_0^{AB} \oplus R_{01}^{AB}$, $(A_0 \oplus B_0) \oplus R_1^{AB} \oplus R_{11}^{AB}$

これは $A_i \oplus B_i$ を秘密分散したときのシェア分配と同じ構造であることは明らかであり，計算代行者 U_∞ は，通常の XOR-SSS 復元処理ルーチンを用いることで $A_i \oplus B_i$ を復元することができる．

第三者および計算代行者の不正を防止する方法

同じ秘密データ部に A や B を直接利用せず，あらかじめ鍵ストリーム K を XOR 演算でマスクしておくことで本来のデータが復元できないようにすることができる．このとき A, B 両者にマスクする必要はなく，片方のみで十分である．例えばシェア生成時に $A \oplus K$, B に対するシェアを計算しておくと，計算代行者 U_∞ から返却されてくるデータは $A \oplus B \oplus K$ であり鍵ストリームで暗号化しておく効果が得られる．なお，全てのエンティティに A, B のデータに対する漏洩もなく，一様ランダムデータとして見えるため安全に処理可能である．

依頼者は最後に $A \oplus B \oplus K$ に対して K を足し合わせて結果を得る必要があるため K の鍵管理を行うデメリットが発生することになる．また，本方式は改ざんに対しては効果がないため計算処理は全てのエンティティが正しく行う semi-honest モデルでの議論と考えることができる．

3.4.2 加減算

入力 A, B に対して $A \pm B$ を出力することを考える．演算に用いられる元は適当な $L = 2^l$ に対して $\mathbb{Z}/L\mathbb{Z}$ 上で行うことを想定する．既存の XOR-VSSS を利用して，XOR 演算の代わりに $\mathbb{Z}/L\mathbb{Z}$ 上の加算・減算に演算に書き換えた numeric-version のマトリクスを利用する．

構成方式 3.29 ($\mathbb{Z}/L\mathbb{Z}$ 上の秘密分散（XOR-SSS の numeric-version）） *XOR-$(2, n)$-SSS* で分散時に用いられるマトリクスを用いて以下のように構成する．

- *XOR-SSS* での R_j は *numeric-version* でも同様に $\mathbb{Z}/L\mathbb{Z}$ 上からランダムに選択する
- *XOR-SSS* では *XOR* 演算で R_j を足しこんで秘匿化しているが，シェア W_i の添字の奇偶性に応じて加算・減算を選択する．
  - シェア W_i の添字が奇数の場合は加算．
  - シェア W_i の添字が偶数の場合は減算．

- *XOR-SSS* にて M_α がシェア W_{ij} に複数出現する場合には正負を交互にして演算する．
- *numeric-version* での復元時に W_i と W_j を足し合わせる際には，添字の和 $i+j$ が奇数の場合には加算，偶数の場合には減算して r_* を相殺する．

具体的には例 3.11 を用いて構成した例を紹介する．

例 3.30 (例 3.11 の numeric-version)

W_0	$m_0 + r_0$	$m_0 + r_1$
W_1	$m_1 - m_2 - r_0$	$m_2 - r_1$
W_2	$m_1 + r_0$	$m_1 - m_2 + r_1$
W_3	$m_2 - r_0$	$m_1 - r_1$

ここで m_1, m_2 と r_0, r_1 は $\mathbb{Z}/L\mathbb{Z}$ の元であり m_0:= 0 である．r_0, r_1 は XOR-VSSS におけるランダムデータにあたり，numeric-version においては乱数として認識される．XOR-SSS においてランダムデータをキャンセルすることで元データを取り出す作業が行われている方式と同じく，r_* をキャンセルする必要が出てくる．その際には，XOR 演算では正負をケアする必要が無かったが，numeric-version では算出される値を補正する場合が発生しており，上記ルールのように，添字の和が奇数の場合には加算，偶数の場合には減算して r_* を相殺するように設計している．

例 3.31 (例 3.30 の演算処理方法)

$F(W_0, W_1)$	$+$	$m_1 - m_2$	m_2
$F(W_0, W_2)$	$-$	$-m_1$	$-m_1 + m_2$
$F(W_0, W_3)$	$+$	m_2	m_1
$F(W_1, W_2)$	$+$	$2m_1 - m_2$	m_1
$F(W_1, W_3)$	$-$	$m_1 - 2m_2$	$-m_1 + m_2$
$F(W_2, W_3)$	$+$	$m_1 + m_2$	$2m1 - m_2$

例 3.30 において，各シェアの出力を見ていく．その際の乱数キャンセルのために，2 列目に示したように各シェアを加算するか減算するかを選択する必要がある．いずれのケースも，連立方程式を解くことで元データを復元することができることが分かる．これは XOR-SSS においてシェア同士を XOR 演算で加算する際に線形独立であることから $\mathbb{Z}/L\mathbb{Z}$ 上の演算でも線形独立であり，復元可能である．

本提案方式の優位性

Ben-Or らによる方式 [38] は Shamir による多項式補間を用いる (k, n)-SSS をベースにしているため，ある素数 p に対して $GF(p)$ 上の演算が必要である．一方で，本提案方式は $\mathbb{Z}/(2^l)\mathbb{Z}$ 上で演算を行う方式のため現在の CPU アーキテクチャに適しており，Ben-Or らによる方式に比べ，高速な演算を行うことが可能である．

第4章　視覚復号型秘密分散法（Visual Secrect Sharing Scheme, VSSS）

秘密分散法（Secret Sharing Scheme）の一例である (k,n)-しきい値秘密分散法は，秘密情報 S を n 個の分配情報（シェア）w_i $(1 \leq i \leq n)$ に符号化する方式であり，任意の k $(k \leq n)$ 個の分配情報 から秘密情報 S を復号することは可能であるが，$k-1$ 個以下の分配情報からは秘密情報 S に関する情報は全く得られないという性質を持つ.

新しいタイプの秘密分散方式として，秘密情報および分散情報に画像を用いる視覚復号型秘密分散法（Visual Secrect Sharing Scheme, VSSS）[61] が提案されている．機密画像を複数のシェア画像に予め分散し OHP スライドのように透過性を持ち物理的に重ねあわせが可能なものに印刷しておく．元の機密画像を復元する際にはそれぞれのシェア画像を重ね合わる操作により，特に計算機のリソースを使うことなく，目の錯覚を利用して機密画像を復元することができる.

VSSS は (k,n)-しきい値秘密分散法に基づく方式（以下 (k,n)-VSSS）が構成 [61] され，これまでに

・復元時のコントラスト（見やすさ）の改善 [56] [57]

・グレースケール・カラー画像への適用 [59] [62]

・一般的な復元のためのアクセス構造の検討 [44] [49]

などの切り口で研究の進展がなされており，本論文では最後のアクセス構造に着目して新しい方式を提案する.

(k,n)-VSSS は各シェア画像に対して平等に復元のための権限が与えられている．しかし秘密情報の分散を目的に実際に VSSS を利用する場合には，復元のための特権を持つメンバーが存在したり，特定のグループ内のメンバーだけでは復元が許可されない等の多様なアクセス構造が必要とされる．以上の背景のもとグラフで表現されたアクセス構造（graph-based access structure）を持つ VSSS [44] である グラフタイプ VSSS（以下 GVSSS） が提案されている.

GVSSS では，グラフの頂点集合をシェア画像の集合と対応づけ，２頂点間に辺が存在する場合には（２頂点に対応する２つのシェア画像から）機密画像が復元され，辺が無い場合には復元されないというアクセス構造を持つ．$(2,n)$-VSSS は，上記のグラフとして完全グラフを利用した場合に相当するため GVSSS は $(2,n)$-VSSS の拡張と考えることができる．本論文ではさらに GVSSS の概念を拡張して，辺の有無（つまり距離１かどうか）ではなく２頂点間の距離に基づいたアクセス構造を持つ G_dVSSS を提案する．既存の GVSSS では１つの機密画像のみがシェア画像に埋め込まれていたが，本論文で提案するスキームでは２枚のシェア画像を重ねあわせたときに，シェア間の距離に応じて復元される機密画像が異なるような

multi-G$_d$VSSS を構成することを実現した．このように複数の機密画像を埋め込むことができるため，既存の GVSSS に比べ 利用用途が拡がると考えられる．また，特に強正則グラフに対して機密画像のコントラストの改善を検討し，既存方式に比べより見やすい構成方法についても新たに提案する．

4.1 従来の視覚復号型秘密分散法

4.1.1 視覚復号型 (k, n)-しきい値秘密分散法

Naor, Shamir による (k, n)-VSSS [61] の構成方法について説明する．機密画像 SI は 2 値（白黒）画像とし 各画素成分 $SI(x, y)$ は画素が白（透明）の場合には 0，黒の場合には 1 と表現するものとする．機密画像 SI を構成する各画素は，シェア画像においては m 個の画素で表現される．つまり成分 $SI(x, y)$ は シェア画像において m 画素で構成される (x, y) ブロックに対応し，シェア画像は機密画像の m 倍に拡大される．この m を 画像拡大率 と呼ぶ．m が平方数であればシェア画像の縦横比を変更しないで構成できるが，非平方の場合には，比率を変える，画素形状を変える，平方数になるように冗長部分を付けるなどの処理が必要となる．

複数枚のシェア画像を重ね合わせた際に各ブロック（m 個の部分画素）における黒画素数の差（コントラスト）によって，白か黒かが視覚的に認識され，機密画像を得ることができる．シェア画像の構成には後述する 生成行列 を用いる．以下 (k, n)-VSSS における (1) 生成行列の定義，(2) 分散画像の構成方法，(3) 具体的な生成行列の例を示す．

4.1.2 生成行列の定義

(k, n)-VSSS におけるシェア画像の構成には 2 種類の生成行列（Basis Matrix）S_0 及び S_1 を用いる．これらの各生成行列は，共に $n \times m$ のバイナリ行列（成分が 0 か 1）であり，行列の各行はシェア画像の集合 $W = \{w_i | 1 \leq i \leq n\}$ により添字付けされる． 2 つの閾値 d_0, d $(1 \leq d_0 < d \leq m)$ に対して，生成行列 S_0, S_1 は次の 3 つの性質を持つ．

(i) S_0 の任意の異なる k 個の行ベクトルを選択し，OR（論理和）演算を施したベクトルの重みは d_0 以下である．

(ii) S_1 の任意の異なる k 個の行ベクトルを選択し，OR 演算を施したベクトルの重みは d 以上である．

(iii) $1 \leq q < k$ を満たす q 個の任意の部分集合 $W' = \{w_{i_1}, \ldots, w_{i_q}\} \subset W$ に対し，S_0, S_1 の行をそれぞれ W' に制限した 2 つの $q \times m$ 行列 は，列の入れ替えにより同じ行列となる．

相対差 α を $\alpha = (d - d_0)/m$ と定義する．相対差は復元する際に白か黒かを判別する際の重要なパラメータであり，できるだけ大きいことが望まれる．また視覚的に復元可能な α の値は 1/25 程度までとされ，それ以上小さくなると機密画像の復元が困難になる．

例 4.1 ($(3,3)$**-VSSS)** $(3,3)$-*VSSS* を構成する生成行列の例を挙げておく．この例では $m = 4, \alpha = 1/4$ である．

$$S_0 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \; S_1 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

4.1.3 シェア画像の構成方法

前節の性質を持つ生成行列 S_0, S_1 を用いて機密画像 SI からシェア画像を作成する方法を説明する．機密画像 SI の各成分 $SI(x, y)$ に対して次の処理を行う．

1. 生成行列として，$SI(x, y) = 0$ の場合は S_0，$SI(x, y) = 1$ の場合は S_1 を選択する．
2. m 次の置換群から 置換 ϕ を任意に選ぶ.
3. 各シェア画像 w_i $(1 \leq i \leq n)$ の (x, y) ブロックは，生成行列の w_i 行成分（m 変数のベクトル）に置換 ϕ を施したものとする．

4.1.4 グラフタイプ VSSS

D.R.Stinson らによる グラフタイプ VSSS（GVSSS）[44] について説明する．グラフ G とは，頂点集合 V と，辺（$V \times V$ の元）集合 E とのペアであり，$G = (V, E)$ と記述される．本論文では，多重辺やループを持たない無向 (undirected) グラフと仮定する．また任意の 2 頂点には必ずパスが存在する連結（connected）グラフのみを取り扱う．

GVSSS [44] は，頂点集合 V を VSSS におけるシェア画像の集合 $W = \{w_i | 1 \leq i \leq n\}$ と同一視し，２つのシェア画像（２頂点）間に辺が存在する場合には機密画像が復元され，辺が無い場合には復元されないというアクセス構造を持つ．

以上の定義のもと，一般のグラフに対して（グラフを固定したときの）画像拡大率の最小値 m^* に関するいくつかの定理と，頂点数が 4 までの全ての無向グラフに対して m^* と生成行列が分類されている [44].

4.1.5 スターグラフ分割による構成

一般のグラフに関してシステマティックに GVSSS を構成する方法 [44] を紹介する．グラフ G をスターグラフ，つまり 1 頂点（中心）からしか辺が存在しないグラフに分割する．スターグラフが与えられた場合，S_0 の各行を $\{1, 0\}$，S_1 の各行を中心頂点のみ $\{1, 0\}$ それ以外

を $\{0,1\}$ とした画像拡大率 2 の生成行列を持つ GVSSS が構成できる．この画像拡大率 2 の生成行列を利用し，分割されたスターグラフの生成行列を連結する（各分割スターグラフの生成行列を横に並べる）ことでグラフ G に対する生成行列を構成することができる．生成行列の連結方法については 4.2.1 節で構成方法を述べる．分割されたスターグラフの個数を β とすると，画像拡大率は $m = 2\beta$ であり，相対差は $\alpha = 1/m = 1/(2\beta)$ となる．

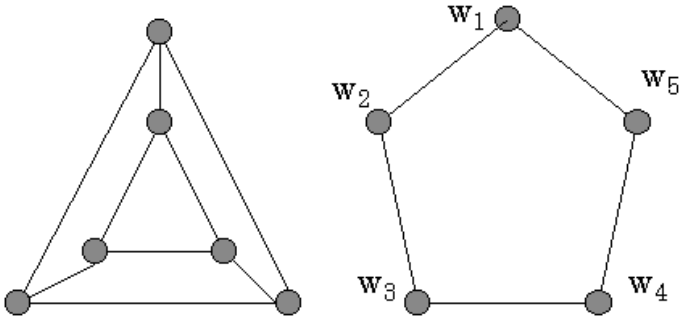

図 4.1: グラフ $\Delta_3, Paley(5)$

図 4.1（左）のグラフ Δ_3 をスターグラフ分割する場合には最低 4 つのコンポーネントを必要とする．そのため本構成法では最小で画像拡大率が 8（つまり相対差は最大 1/8）の生成行列を持つ GVSSS しか構成できない．しかし次の生成行列により画像拡大率 3（つまり相対差は 1/3）の GVSSS を構成することが知られている．

例 4.2 (GVSSS-$(\Delta_3, 3)$)

$$S_0 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, S_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

画像拡大率 m の生成行列を持つ GVSSS を GVSSS-(G, m) と表記すると定めると，スターグラフ分割法を用いて GVSSS-$(\Delta_3, 8)$ を生成する一方で GVSSS-$(\Delta_3, 3)$ が存在していることを示している．このようにスターグラフ分割法は必ずしも相対差が最大となる構成方法ではないため，コントラストを改善する研究の余地が残されていると言える．

次節では辺の有無による機密画像の復元の可否というスキームを拡張して，新しいアクセス構造を持つ multi-G$_d$VSSS を提案する．さらにスターグラフ分割法を利用した構成方法よりも画像拡大率を小さくする（相対差を大きくする）手法についても検討している．

4.2 GVSSS の拡張 multi-G$_d$VSSS

本節にて GVSSS の拡張を行うが，まずいくつかの記号の定義をしておく．n 頂点グラフ $G=(V,E)$ の最大距離を D としたとき，$V\times V$ は 2 点間の距離により $E_0, E_1(=E), \ldots, E_D$ に分割することができる．つまり E_i: $=\{(x,y)\in V\times V | x,y$ の距離が $i\}$ とすると $V\times V=\bigcup_{i=0}^{D} E_i$ となる．さらに E_i に対する $n\times n$ 隣接行列 A_i を次のように定義する．

$$(A_i)_{xy}\colon = \begin{cases} 1 & \text{if } (x,y)\in E_i, \\ 0 & \text{if } (x,y)\notin E_i. \end{cases}$$

$\Gamma := \{1,\ldots,D\}$ を $\Gamma=\bigcup_{i=1}^{l}\Gamma_i$ となるような互いに素な l 個の部分集合 $\Gamma_1,\ldots,\Gamma_l$ に分割する．特に Γ_0: $=0$ とおく．このとき E_{Γ_k}: $=\bigcup_{j\in\Gamma_k} E_j$ とすると，$V\times V=\bigcup_{k=0}^{l} E_{\Gamma_k}$ となる．

以上の定義のもと 2 頂点の距離に基づき次のような 2 つのスキームを導入することができる．

アクセス構造（1）

特に $l=2$ つまり $V\times V=E_0\cup E_{\Gamma_1}\cup E_{\Gamma_2}$ と仮定する．異なる 2 頂点 w_i, w_j を選択したとき $(w_i,w_j)\in E_{\Gamma_1}$ の場合は機密画像が復元されるが，$(w_i,w_j)\in E_{\Gamma_2}$ の場合は復元できないというアクセス構造を考える．

ここで $\Gamma_1=\{1\}$ すなわち $E_{\Gamma_1}=E$ を満たす場合は，従来型 GVSSS に相当する．つまり，異なる 2 頂点 w_i, w_j を選択したとき $(w_i,w_j)\in E_{\Gamma_1}$ の場合は機密画像が復元されることから，新しいグラフとして $G_1=(V,E_{\Gamma_1})$ を考えることにより，従来型 GVSSS に帰着することができる．

アクセス構造（2）

異なる 2 頂点 w_i, w_j を選択したとき $(w_i,w_j)\in E_{\Gamma_k}$ の場合は機密画像 SI_k が復元される．つまり l 個の機密画像 SI_k $(1\le k\le l)$ のうちいずれかが復元されるというアクセス構造を考える．

$l\ge 3$ の場合も 2^l 個の拡張生成行列を考えることにより同様に構成可能であるため，本節では特に $l=2$ の場合についてのみ取り上げる．2 つの機密画像を SI_1, SI_2 としそれぞれの画素成分を $SI_1(x,y), SI_2(x,y)$ とする．また SI_1, SI_2 の画素数は同一であるとする．

アクセス構造（1）は従来の GVSSS に帰着できるため，以降はアクセス構造（2）の構成を検討する．アクセス構造（2）を持つ拡張 GVSSS を multi-G$_d$VSSS と表記する．

拡張生成行列の定義

従来のように 2 つの生成行列を用いるのではなく，$S_{(0,0)}, S_{(1,0)}, S_{(0,1)}, S_{(1,1)}$ の 4 $(= 2^2)$ つの拡張生成行列を利用する．これらの拡張生成行列は次の 3 つの性質を持つように生成する．ただし $W = \{w_i | 1 \leq i \leq n\}$ で添字付けされたバイナリ行列 B に対して，(w_i, w_j) 成分を B の w_i 行目のベクトルと w_j 行目のベクトルの OR 演算を施したベクトルの重みと定義した対称行列を $R(B)$ とおく．

$(i)'$ $R(S_{(1,0)}) - R(S_{(0,0)})$ は $\sum_{i \in \Gamma_1} A_i$ の定数倍である．

$(ii)'$ $R(S_{(0,1)}) - R(S_{(0,0)})$ は $\sum_{i \in \Gamma_2} A_i$ の定数倍である．

$(iii)'$ $R(S_{(1,1)}) - R(S_{(0,0)})$ は $\sum_{i \in \Gamma_1 \cup \Gamma_2} A_i$ の定数倍である．

シェア画像の構成方法

機密画像 SI_1, SI_2 の各成分 $SI_1(x, y), SI_2(x, y)$ に対して，$SI_1(x, y) = a$ かつ $SI_2(x, y) = b$ の場合に生成行列として $S_{(a,b)}$ を選択する．以降の処理は 4.1.3 と同様である．

例 4.3 (生成行列の例) 図 *4.1*（右）の グラフ $Paley(5)$ に対して，$\Gamma_1 = \{1\}, \Gamma_2 = \{2\}$ となる *multi-G_dVSSS* を構成する $m = 7$ の生成行列は次の通りである．

$$S_{(0,0)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix},$$

$$S_{(1,0)} = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix},$$

$$S_{(0,1)} = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \end{bmatrix},$$

$$S_{(1,1)} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}.$$

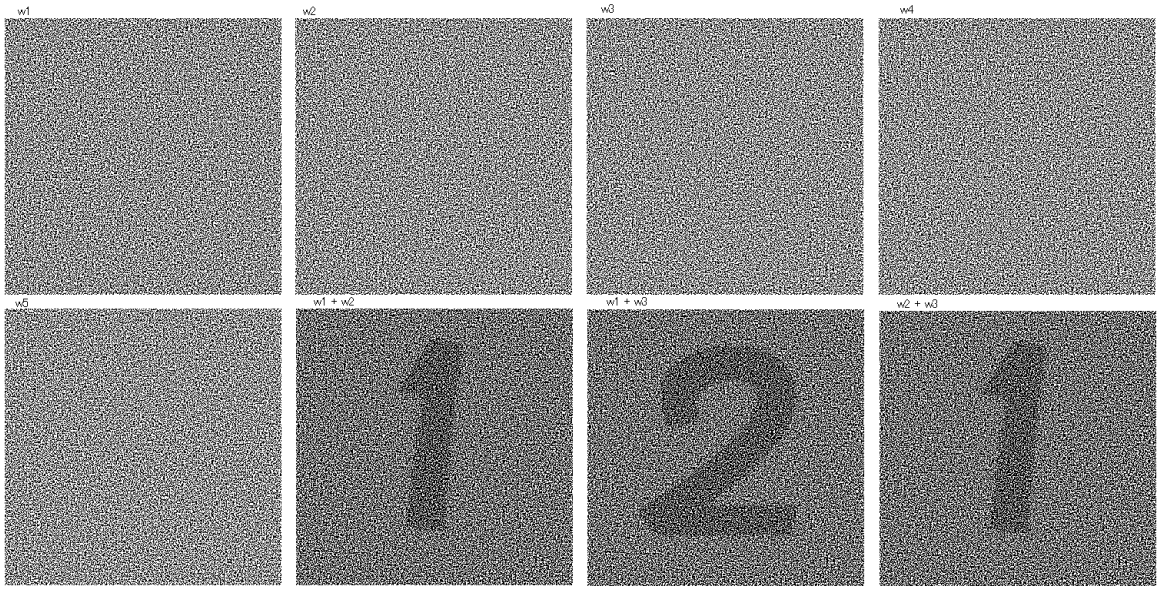

図 4.2: シェア画像と復元画像

図 4.2 はそれぞれ w_i $(1 \leq i \leq 5)$ に対応したシェア画像（ただし $m = 9$ に拡張）と，シェア画像を重ねあわせたときの復元画像として $w_1 + w_2, w_1 + w_3, w_2 + w_3$ の例を示した．距離 1 の場合には機密画像 SI_1「1」が，距離 2 の場合には機密画像 SI_2「2」が復元される．

既存方式との違い

アクセス構造（2）は複数画像を埋め込むことのできる従来の提案と類似しているように捉えられるが，提案方式は次を満たし，既存方式とは異なることに留意されたい．

- 複数の画像が埋め込まれている領域が異なる方式 [57] ではない．
- 重ねあわせる枚数により異なる復元画像を得る方式 [54] [56] ではない．
- シェア画像を 1 画素分ずらすことにより異なる復元画像を得る方式 [52] ではない．

4.2.1 生成行列連結による構成法

従来型 GVSSS から multi-G$_d$VSSS を構成する単純な方式を述べる．グラフ $G_1 = (V, E_{\Gamma_1})$ に基づく G$_d$VSSS の生成行列を $S_0(G_1), S_1(G_1)$，画像拡大率を $m(G_1)$ とし，グラフ $G_2 = (V, E_{\Gamma_2})$ においても同様の記号を用いる．

このとき $S_{(a,b)} := [S_a(G_1)|S_b(G_2)]$ とする 画像拡大率 $m(G_1) + m(G_2)$ の 4 つの生成行列を用いて multi-G$_d$VSSS が構成できる．しかし l が増大するに従い画像拡大率が大きくなる欠点がある．そこで，次節にて強正則グラフを用いて画像拡大率の増大を抑える方法について検討していく．

4.2.2 強正則グラフ利用による画像拡大率の削減

本節では以下の考察によりグラフに対して強正則という制約を設けた上で検討を行う．以降において，I は単位行列，J は 成分がすべて 1 の行列を表すものとする．一般に，バイナリ行列 B のすべての行ベクトルが同じ重み（ここでは k_B）を持つとき $R(B)$ は

$$\text{(r1) } R(B) = 2k_B J - BB^T$$

と書くことができる．(r1) 式から，$\Gamma' \subset \Gamma$ に対し

$$\text{(r2) } MM^T = kI + \sum_{d \in \Gamma'} A_d$$

となるようなバイナリ行列 M を見つけることにより，本論文で提案するアクセス構造（1）を持つ G$_d$VSSS を構成することが可能であることがわかる．ここで G$_d$VSSS はアクセス構造（1）の定義から，グラフの距離に応じて復元の可否を決定する GVSSS の拡張方式と定める．2 頂点間の距離 d が $d \in \Gamma'$ に含まれる場合には機密画像を復元し，それ以外の場合には復元せず，機密画像に関する一切の情報を漏らさない方式である．

ここで，最大距離が 2，頂点数 n のグラフが

$$\text{(r3) } A_1 {A_1}^T = kI + \lambda A_1 + \mu A_2 \text{（ただし } 0 < \mu < k < n-1 \text{ とする）}$$

を満たすとき，強正則（Strongly Regular）と呼ぶ．このとき各パラメータ n, k, λ, μ 間には次の関係が成立している [46] [60].

$$\text{(r4) } k(k-\lambda-1) = \mu(n-k-1)$$

例 4.4 (強正則グラフ $Paley(5)$) グラフ $Paley(5)$ は $(n, k, \lambda, \mu) = (5, 2, 0, 1)$ のパラメータを持つ強正則グラフである．

$Paley(5)$ の隣接行列 A_1 および A_2 は

$$A_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix},\ A_2 = \begin{bmatrix} 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \end{bmatrix}$$

であり，

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}^T = \begin{bmatrix} 2 & 0 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 & 1 \\ 1 & 0 & 2 & 0 & 1 \\ 1 & 1 & 0 & 2 & 0 \\ 0 & 1 & 1 & 0 & 2 \end{bmatrix} = 2I + A_2$$

を確認できる．

以降，(r2) 式と (r3) 式を比較し λ と μ の差を利用して従来型 GVSSS を構成する手法について詳細に述べる．

$Mat\{(A)_a(B)_b\}$ を行列 A を a 回, 行列 B を b 回横に並べた行列とする．このとき上式を満たす強正則グラフに対して次が成立する．

(m1) $R(Mat\{(I)_a(1_n)_b\}) = (a+b)J + a(A_1 + A_2)$.

(m2) $R(Mat\{(A_1)_1(1_n)_c\}) = (k+c)J + (k-\lambda)A_1 + (k-\mu)A_2$.

ただし 1_n は全ての成分が 1 の $n \times 1$ 行列とする．(m1)(m2) 式より $\lambda < \mu$ を満たすとき，J の項と A_2 の項の係数が一致するように方程式を解くことにより，$a = k - \mu, b = \mu, c = 0$ を選択することで従来型 GVSSS が構成できる．さらに，

(m3) $R(Mat\{(A_1 + I)_1(1_n)_d\}) = (k+1+d)J + (k-\lambda-1)A_1 + (k-\mu+1)A_2$

と，アダマール行列 [47][48] により構成可能な

(m4) $R(H) = (2p+1)J + (p+1)(A_1 + A_2)$

を満たす $4p+3$ 次正方行列 H を用いると，$\mu = \lambda + 1$ が成立する場合にのみ multi-G$_d$VSSS が構成できる．このとき $a = p = k - \mu, b = a + 1, d = k - 2\mu, c = d + 1$ を選択すればよい．また画像拡大率 を m とすると相対差は $1/m$ となる．

4.2.3 評価

以下 multi-G$_d$VSSS に関する 2 つの構成方法について画像拡大率の比較により有効性を評価する．

- （既存方式）4.1.5 節のスターグラフ分割法により構成された従来型 GVSSS を用いて 4.2.1 節の生成行列連結法で構成する手法
- （提案方式）4.2.2 節の (m3)(m4) 式を用いて構成する手法

両方式ともに，画像拡大率を m とすると相対差は $1/m$ で表され，相対差は画像拡大率にのみに依存する．つまり画像拡大率が小さいほど相対差は大きくなり，よりコントラストを改善することを意味する．

評価対象とするグラフは，頂点数 13 以下の全ての強正則グラフとした．4.2.2 節の手法に適用するためには $\mu = \lambda + 1$ を満たす必要があるが，頂点数が 13 以下の強正則グラフは全て $\mu = \lambda + 1$ を満たすことが知られている [60]．また，頂点数が 13 を超え，$\mu = \lambda + 1$ を満たす強正則グラフは $Paley(17)$ などが挙げられるが，相対差が現実的ではないため評価対象から除外した．

表 4.1 は上記の 2 方式を適用したときの画像拡大率を比較したものである．4.2.1 節のスターグラフ分割法に比べ 4.2.2 節の強正則グラフ利用方式は画像拡大率 m を削減していることがわかる．

表 4.1: multi-G$_d$VSSS の構成法の違いによる画像拡大率の比較

グラフ	n	k	λ	μ	スターグラフ分割法 4.2.1 節	強正則グラフ利用 4.2.2 節
$L_2(2)$	4	2	0	1	8	7
$Paley(5)$	5	2	0	1	12	7
$L_2(3)$	9	4	1	2	28	21
$Petersen$	10	3	0	1	32	23
$Paley(13)$	13	6	2	3	50	43

ここで，上記各グラフの定義を行う．Lattice graph $L_2(m)$ $(m \geq 2)$ は $V = S \times S$ (S は位数 m の集合）とし，相異なる直積の元が同じ座標成分を持つときに辺を持つグラフである．Paley graph $Paley(q)$ (q は $q \equiv 1 \pmod 4$ を満たす素数べき) は $V = GF(q)$ (位数 q の有限体) とし，相異なる元の差が平方数であるときに辺を持つグラフである．Petersen graph は V を位数 5 の集合における位数 2 の部分集合の集まりとし，相異なる部分集合が互いに素のときに辺を持つグラフである．

4.2.4 提案方式の利用例

VSSS は秘密分散機能を備えた印刷装置への適用ばかりでなく，ユーザの個人認証と同時に（信用できないかもしれない）端末に対する認証が可能 [52] [58] [55] であり，ヒューマンクリプト [51] において非常に有効なアプリケーションであると考えられている．

本論文で提案した multi-G$_d$VSSS を対面型属性証明に適用した例を示す．各シェア画像は各エンティティに配布され，直接対面を通してお互いのシェアを重ねあわせることで相手がどのような属性を保持するか，どのようなグループに属するかを示す用途に用いることができる．このとき従来は同じグループのメンバーかどうかを認識するという目的で利用される方式が想定されるが multi-G$_d$VSSS を用いることにより各エンティティごとにあるエンティティから見た他エンティティとの関係が異なるという性質を用いて，次のような用途に利用できる．各シェア画像を持つエンティティの年齢がどのくらい離れているかを，グラフ上の２エンティティ間の距離で表現しておく．これによりお互いの年齢を公開しないで相手の年齢が自分とどのくらい離れているかだけを知る手段に利用できる．

さらに，本論文ではグラフとして無向グラフのみを扱ったが，multi-G$_d$VSSS を 2 点間の距離を辺の重みに置きかえることで，重み付けグラフを用いる方法への拡張が可能であると考えられる．

4.3 復元画像として白黒反転画像を許容する視覚復号型秘密分散法

[61] では (k,n)-しきい値秘密分散法に基づく方式（以下 (k,n)-VSSS）が構成されているが，復元のためのアクセス構造は (k,n)-しきい値法に限らず，様々なアクセス構造を持った VSSS が提案されている．そのひとつに，グラフで表現されたアクセス構造（graph-based access structure）を持つ VSSS [44] である GVSSS が存在する．

GVSSS は復元のためのアクセス構造をグラフで表現する．グラフの頂点集合をシェア画像の集合と対応づけ，２頂点間に辺が存在する場合には（２頂点に対応する２つのシェア画像から）機密画像が復元され，辺が無い場合には復元されないというアクセス構造である．$(2,n)$-VSSS は，上記のグラフとして完全グラフを利用した場合に相当するため GVSSS は $(2,n)$-VSSS の拡張と考えることができる．

GVSSS はスターグラフ分割法 [44] などにより，任意のグラフに関してシステマティックに構成することができるが，画像拡大率（機密画像の１画素に対するシェア画像の画素数）は必ずしも最小となる構成方法ではないことを前節で見てきた．本節では，さらに復元画像のコントラストを改善するための新しい構成方法を複数提案する．また，提案方式により，いくつかのグラフにおいて画像拡大率を最小にする optimal な結果を得た．

4.3.1 グラフに関する定義・記法

グラフ G とは，頂点集合 V と，辺（$V \times V$ の元）集合 E とのペアであり，$G=(V,E)$ と記述される．本論文では，多重辺やループを持たない無向グラフと仮定する．各行，各列を

V（ただし $|V| = n$ ）により添字付けされた G に対する $n \times n$ 隣接行列 $Adj(G)$ を次のように定義する．

$$(Adj(G))_{xy} := \begin{cases} 1 & \text{if } (x,y) \in E, \\ 0 & \text{if } (x,y) \notin E. \end{cases}$$

グラフ $G = (V, E)$ に対して $V' \subset V$ かつ $E' \subset E$ となるようなグラフ $G' = (V', E')$ を部分グラフ (subgraph) と呼ぶ．さらに部分グラフ $G' = (V', E')$ が 任意の $u, v \in V'$ に対して、$\{u, v\} \in E \Leftrightarrow \{u, v\} \in E'$ が成り立つときに、誘導部分グラフ (induced subgraph) と呼ぶ．G に対する誘導部分グラフの集合を $Ind(G)$ と置く．$G \in Ind(G)$ とするが，空グラフ（辺を含まないグラフ）は $Ind(G)$ には含まれないとする．

完全グラフとは，異なる 2 頂点間に必ず辺を持つグラフであり，n 頂点を持つ完全グラフを K_n と表記する．クリーク (clique) は，グラフに含まれる完全部分グラフを指し，$\omega(G)$ をグラフ G に対する最大クリークの大きさとする．

完全 2 部グラフとは，頂点集合 V を V_1, V_2 に分割 $(V = V_1 \cup V_2, V_1 \cap V_2 = \emptyset)$ したとき，任意の $u \in V_1, v \in V_2$ の間に辺が必ず存在するが，他には辺がないグラフである．$|V_1| = a_1$, $|V_2| = a_2$ であるとき，K_{a_1,a_2} と表記する．特に $K_{1,a}$ を スターグラフと呼ぶ．

完全 n 部グラフとは，頂点集合 V を $V_1, V_2, \ldots, V_n$ に分割 $(V_i \cap V_j = \emptyset (i \neq j))$ したとき，任意の $u \in V_i, v \in V_j (i \neq j)$ の間に辺が必ず存在するが，他には辺がないグラフである．$|V_i| = a_i$ であるとき，$K_{a_1,a_2,\ldots,a_n}$ と表記する．

n 頂点グラフ $G = (V(G), E(G))$ に対する n 部グラフとは，頂点集合 V を $V_1, V_2, \ldots, V_n$ に分割したとき，G の各頂点 $V(G) = \{v_1, v_2, \ldots, v_n\}$ を $\{V_1, V_2, \ldots, V_n\}$ と対応付け，任意の $u \in V_i, v \in V_i((v_i, v_j) \in E(G))$ の間に辺が必ず存在するが，他には辺がないグラフである．$|V_i| = a_i$ であるとき，$K_{a_1,a_2,\ldots,a_n}(G)$ と表記する．G を完全グラフとした $K_{a_1,a_2,\ldots,a_n}(K_n)$ は完全 n 部グラフ $K_{a_1,a_2,\ldots,a_n}$ と一致する．

4.3.2 GVSSS の拡張 G±VSSS に関する代数的定義

$G = (V, E)$ に対する GVSSS [44] は，頂点集合 $V(|V| = n)$ をシェア画像の集合 $W = \{w_i | 1 \leq i \leq n\}$ と同一視し，２つのシェア画像（２頂点）間に辺が存在する場合には機密画像が復元され，辺が無い場合には復元されないというアクセス構造を持つものであった．

機密画像は２値（白黒）画像とし 各画素成分 $SI(x, y)$ は画素が白（透明）の場合には０，黒の場合には１と表現するものとする．またシェア画像も２値画像であり同様の表現方式を持つとする．機密画像 SI を構成する各画素は，シェア画像においては m 個の画素で表現される．つまり成分 $SI(x, y)$ は シェア画像において m 画素で構成される (x, y) ブロックに対応し，シェア画像は機密画像の m 倍に拡大される．この m を画像拡大率と呼ぶ．２枚のシェア画像を重ね合わせた際に各ブロック（m 個の部分画素）における黒画素数の差（コントラスト）によって，白か黒かが視覚的に認識され，機密画像を得ることができる．このとき復元画像として白黒反転画像を許容する拡張方式（文献 [65] においては VCS_2 として定義されている方式）を G±VSSS と表記することとする．

G$^{\pm}$VSSS における生成行列

G$^{\pm}$VSSS の構成方法を表現するために，2 種類の生成行列（Basis Matrix）S_0 及び S_1 を用いる．これらの各生成行列は，共に $n \times m$ のバイナリ行列（成分が 0 か 1）であり，行列の各行はシェア画像の集合 $W = \{w_i \mid 1 \leq i \leq n\}$ により添字付けされる．また m を画像拡大率とする．アクセス構造としてグラフ G で表現された画像拡大率 m の生成行列を持つ G$^{\pm}$VSSS を G$^{\pm}$VSSS-(G, m) と表記する．

$w(v)$ を ベクトル v の重み（ベクトル内の 1 の個数）とする．また，各行を $W = \{w_i \mid 1 \leq i \leq n\}$ で添字付けしたバイナリ行列 B に対して，$B(w_i)$ を B の w_i 行ベクトルとしたとき，(w_i, w_j) 成分を $w(B(w_i) + B(w_j))$ と定義した n 次対称行列を $R(B)$ とおく．行列 A に対し，A の各成分の絶対値を成分として持つ A と同サイズの行列を $abs(A)$ と置く．さらに $norm(A)$ を $(norm(A))_{xy} := 0$ if $A_{xy} = 0$, $(norm(A))_{xy} := 1$ if $A_{xy} \neq 0$ と定義する．このとき G$^{\pm}$VSSS-(G, m) の生成行列は次の定義を満たすように構成される．

定義 4.5 (G$^{\pm}$VSSS-(G, m) の生成行列) G$^{\pm}$VSSS-(G, m) の $|V(G)| \times m$ 生成行列 S_0, S_1 は $norm(\ R(S_1) - R(S_0)\ ) = Adj(G)$ を満たす.

例 4.6 (G$^{\pm}$VSSS-$(\Delta_3, 3)$)

$$Adj(\Delta_3) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}.$$

$$S_0 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ S_1 = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}.$$

$$R(S_0) = \begin{bmatrix} 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 2 & 2 & 2 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 \\ 2 & 2 & 2 & 1 & 1 & 1 \end{bmatrix}.$$

$$R(S_1) = \begin{bmatrix} 2 & 3 & 3 & 3 & 2 & 2 \\ 3 & 2 & 3 & 2 & 3 & 2 \\ 3 & 3 & 2 & 2 & 2 & 3 \\ 3 & 2 & 2 & 1 & 2 & 2 \\ 2 & 3 & 2 & 2 & 1 & 2 \\ 2 & 2 & 3 & 2 & 2 & 1 \end{bmatrix}.$$

よって

$$norm(\ R(S_1) - R(S_0)\ ) = R(S_1) - R(S_0) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} = Adj(\Delta_3)$$

を満たすことが確認できる.

4.3.3 GVSSS に対する最小画像拡大率に関する既存の結果

グラフ G を与えたとき，GVSSS-(G, m) が存在する m の最小値を $m^*(G)$ とおく．特に GVSSS-(G, m) が optimal であるとは，$m = m^*(G)$ を満たすときとする．$m^*(G)$ に関しては既に次のような結果が知られている.

定理 4.7 ([44] Theorem 7.3) $m^*(K_n) = min\ \{m|\ n \leq {}_mC_{\lfloor \frac{m}{2} \rfloor}\}$.

定理 4.8 ([44] Theorem 7.4) $m^*(G) \geq m^*(K_{\omega(G)})$.

定理 4.8 は $m^*(G)$ の下界を与えているが，与えられたグラフ G の最大クリークを求める問題は NP 完全問題である.

4.3.4 Independent の定義

与えられたグラフ G に対して GVSSS-(G, m) の生成行列によっては，次に挙げる例 4.9 のように，配布先は異なるが全く同じシェア画像が存在する場合がある.

例 4.9

$$Adj(G) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},\ S_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix},\ S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}.$$

定義 4.10 GVSSS-(G, m) の生成行列 S_0, S_1 が次を満たすとき，GVSSS-(G, m) を independent という．S_0 の w_i 行ベクトルと S_0 の w_j 行ベクトルが一致し，S_1 の w_i 行ベクトルと S_1 の w_j 行ベクトルが一致するような i, j $(i \neq j)$ は存在しない.

例 4.6 は independent である．independent という概念は GVSSS だけでなく，復元画像として白黒反転画像を許容する視覚復号型秘密分散法 G$^{\pm}$VSSS にも同様に適用することが可能であり，independent G$^{\pm}$VSSS に関して次が成り立つ．

定理 4.11 $^{\exists}$ *independent G$^{\pm}$VSSS-*$(G,2) \Rightarrow G \in Ind(K_{2,2})$.

証明. $m=2$ となる S_0, S_1 の取り得る場合を全て列挙すると次のようになる．

$$\tilde{S_0} = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \\ w_6) \end{matrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \tilde{S_1} = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \\ w_6) \end{matrix} \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

上記行列の $R(\tilde{S_1}) - R(\tilde{S_0})$ は次の通りである．

$$R(\tilde{S_1}) - R(\tilde{S_0}) = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

$norm(\ R(\tilde{S_1}) - R(\tilde{S_0})\ )$ の列成分または行成分がすべて 0 の行・列を削除した行列を隣接行列とするグラフは $K_{2,2}$ である． ∎

系 4.12 $^{\exists}$ *independent G$^{\pm}$VSSS-*$(G,3)$ *s.t.* $G \notin Ind(K_{2,2}) \Rightarrow$ *independent G$^{\pm}$VSSS-*$(G,3)$ は *optimal* である．

上記の定理 4.11，系 4.12 の independent という条件をはずすと，容易に次を得る．

定理 4.13 $^{\exists}$ *G$^{\pm}$VSSS-*$(G,2) \Rightarrow$ ある正整数 a_1, a_2, a_3, a_4 に対して $G \in Ind(K_{a_1,a_2,a_3,a_4}(K_{2,2}))$.

系 4.14 $^{\exists}$ *G$^{\pm}$VSSS-*$(G,3)$ *s.t.* いかなる正整数 a_1, a_2, a_3, a_4 に対しても $G \notin Ind(K_{a_1,a_2,a_3,a_4}(K_{2,2}))$ $\Rightarrow$ *G$^{\pm}$VSSS-*$(G,3)$ は *optimal* である．

4.3.5 グラフ分割による G$^{\pm}$VSSS の構成

スターグラフ分割

グラフ G をスターグラフ $K_{1,a}$，つまり 1 頂点（中心）からしか辺が存在しないグラフに分割する．スターグラフにおいては，S_0 の各行を $\{1,0\}$，S_1 の各行を中心頂点のみ $\{1,0\}$，それ以外を $\{0,1\}$ とした GVSSS-$(K_{1,a},2)$ が構成できる．この画像拡大率 2 の生成行列を利用し，分割されたスターグラフの生成行列を連結（各分割スターグラフの生成行列を横に並べる）することでグラフ G に対する生成行列を構成することができる．分割されたスターグラフの個数を $\beta(G)$ とすると GVSSS-$(G, 2\beta(G))$ が構成できることがわかる [44] .

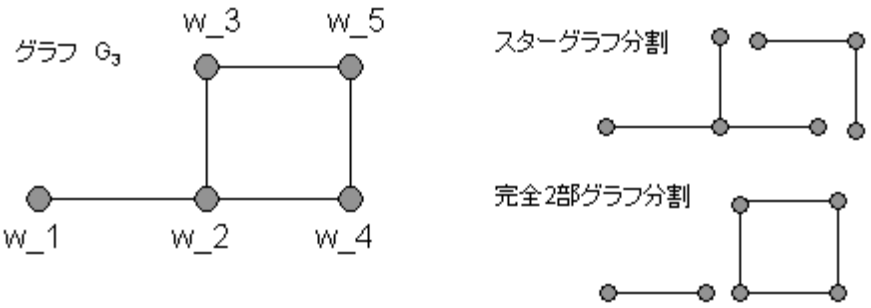

図 4.3: スターグラフ分割，完全 2 部グラフ分割

例 4.15 (スターグラフ分割) 図 *4.3*（左）のグラフ G_3 *[45]* に対するスターグラフ分割は図 *4.3*（右上）であり 下の生成行列を持つ *GVSSS*-$(G_3, 4)$ を構成することができる．

$$S_0 = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; S_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

完全 n 部グラフ $K_{a_1,a_2,\ldots,a_n}$ への適用

前節のスターグラフ分割を拡張させる．

構成方式 4.16 グラフ分割後の生成行列として *GVSSS*-$(K_{1,a}, 2)$ を利用する代わりに，*GVSSS*-$(K_{a_1,a_2,\ldots,a_n}, m^*(K_n))$ を利用する．つまり，*GVSSS*-$(K_n, m^*(K_n))$ の生成行列をそれぞれ a_i 回同じ行ベクトルを繰り返した生成行列を利用する．

例 4.17 (n 部グラフ分割) 図 *4.3*（左）のグラフ G_3 に対する完全 n 部グラフ分割は図 *4.3*（右下）であり 下の生成行列を持つ *GVSSS*-$(G_3, 4)$ を構成することができる．

$$S_0 = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \end{matrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \; S_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}.$$

n 部グラフ $K_{a_1,a_2,\ldots,a_n}(G)$ への適用

構成方式 4.18 グラフ分割後の生成行列として *GVSSS*-$(K_{a_1,a_2,\ldots,a_n}, m^*(G))$ を利用する．つまり，*GVSSS*-$(G, m^*(G))$ の生成行列をそれぞれ a_i 回同じ行ベクトルを繰り返した生成行列を利用する．

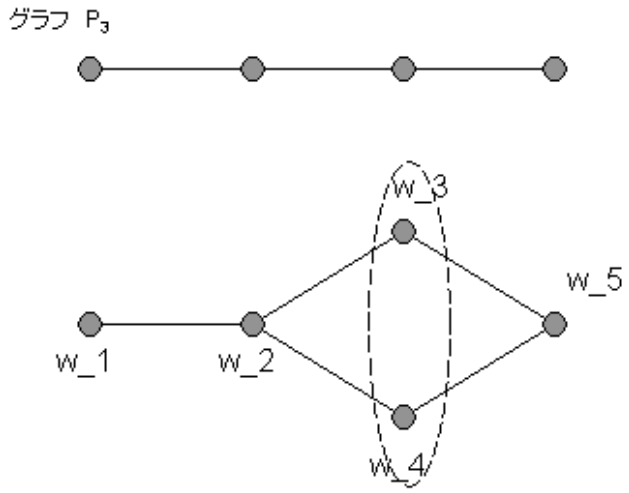

図 4.4: グラフ $P_3, K_{1,1,2,1}(P_3)$

例 4.19 (n 部グラフへの適用) G_3 に対する n 部グラフへの適用は図 *4.4* のとおりである．*GVSSS-*$(P_3, 3)$ から下の生成行列を持つ *GVSSS-*$(G_3, 3)$ を構成することができる．

$$S_0 = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \end{matrix} \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; S_1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

辺消去方式

構成方式 4.20 与えられたグラフの辺集合を，集合差も許して $K_{a_1,a_2,\ldots,a_n}$ 及び $K_{a_1,a_2,\ldots,a_n}(G)$ で表記し「差」を S_0, S_1 を入れ替えるなどして実現する．

例 4.21 (辺消去方式) 図 *4.5* のように $E(G_3) = E(K_{2,3}) - (w_1, w_5) = E(K_{2,3}) - E(K_{1,1})$ と表記することができることを利用する．それぞれ *GVSSS-*$(K_{2,3}, 2)$, *GVSSS-*$(K_{1,1}, 2)$ の生成行列を連結する際に辺 (w_1, w_5) が消去されるように *GVSSS-*$(K_{1,1}, 2)$ の S_0, S_1 を入れ替える処理を行い，下の生成行列を持つ *GVSSS-*$(G_3, 3)$ を構成することができる．

$$S_0 = \begin{matrix} w_1) \\ w_2) \\ w_3) \\ w_4) \\ w_5) \end{matrix} \left[\begin{array}{c|cc} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \end{array}\right], \; S_1 = \left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{array}\right].$$

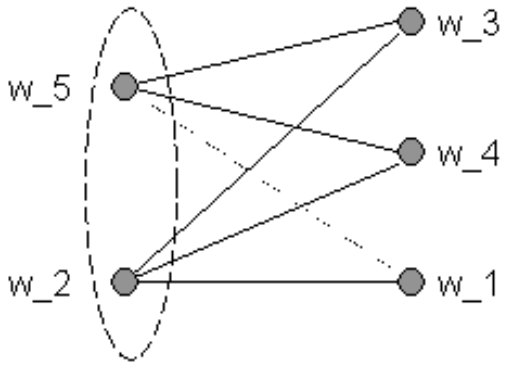

図 4.5: グラフ $K_{2,3} - K_{1,1}$

評価

図 4.3 記載の G_3 を例に挙げて提案方式の優位性（スターグラフ分割法では m=4 であった画像拡大率を，提案方式 2,3 では m=3 に削減した）が示された．一方，例 4.17,4.19 により $m^*(G_3) \leq 3$ が成立する．$G_3 \notin Ind(K_{a_1,a_2,a_3,a_4}(K_{2,2}))$ であり，系 4.14 により optimal な構成であることがわかる．

その他 C_6 (6 頂点のサイクルグラフ) 及びその誘導部分グラフ（P_5 など）も optimal な例である．GVSSS-$(C_6, 3)$ を構成する生成行列は次の通りである．GVSSS-$(P_5, 3)$ は GVSSS-$(C_6, 3)$ における生成行列の 5 行分だけを利用すればよい．

例 4.22 (GVSSS-$(C_6, 3)$)

$$S_0 = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \; S_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

4.3.6 あえて Independent GVSSS にしないことの意味

分散画像情報が漏洩した場合に，漏洩元を特定することができない問題を解決するために補正行列追加による non-independent VSSS への移行方法について触れておく．

S_0, S_1 ともに全く同じ行ベクトルとなっている列集合のうち最大の集合の位数を α とおく．$2^m \geq \alpha$ を満たす m に対して m 列の補正列ベクトルを加える．補正列ベクトルは $\{0, 0, \ldots, 0\}$, $\{1, 0, \ldots, 0\}, \{0, 1, \ldots, 0\}, \ldots, \{1, 1, \ldots, 1\}$ のように 2^m 通りのベクトルから異なるベクトルを選択し S_0, S_1 ともに同じ列ベクトルを追加する．

例 4.23 (補正行列追加)

$$Adj(G) = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, S_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 0 \end{bmatrix}, S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \\ 0 & 1 \end{bmatrix}$$

に対して下記のように補正行列を入れることができる．

$$S_0 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{array}\right], S_1 = \left[\begin{array}{cc|cc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{array}\right].$$

例 4.24 例 *4.23* は必ずしも最小の画像拡大率で構成したとは限らない点に注意する．例 *4.9* に対してうまく構成した事例が以下となる．この例は *independent* であり，かつ *optimal* な構成例である．

$$S_0 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, S_1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}.$$

4.4 画像拡大率が 3 の GVSSS,G$^{\pm}$VSSS の分類

GVSSS-$(C_6, 3)$（例 4.22）に見られるようにいくつかの optimal な GVSSS の構成事例を見てきた．以降 $m^*(G) \leq 4$ と画像拡大率を固定させた上で，どのようなグラフにおいて GVSSS-$(G, 3)$ または G$^{\pm}$VSSS-$(G, 3)$ および GVSSS-$(G, 4)$ または G$^{\pm}$VSSS-$(G, 4)$ が構成可能なのかについて分類を試みた（$m^*(G) = 4$ のケースは次節）．

視覚復号型暗号ではなく通常の秘密分散方式においては，いくつかの同様の研究がなされており，シェア数を限定した上でのスモールケースについて分類されている [45].

定理 4.11 は G$^{\pm}$VSSS-$(G, 2)$ の分類を完全に終えていることを意味している．そこで，次節にて $m^* = 3$ のケースについて定理 4.11 と同様の分類結果を示すこととする．

4.4.1 G$^{\pm}$VSSS-$(G,3)$ の分類

補題 4.25 $^{\exists}$ *independent G*$^{\pm}$*VSSS-*$(G,3) \Leftrightarrow G \in \Delta_3{}^{\pm}$．ただし 頂点数 *18* のグラフ $\Delta_3{}^{\pm}$ の隣接行列は以下のとおり：

$$\begin{bmatrix}
0&1&1&1&0&0&1&0&0&0&1&1&1&0&0&1&0&0\\
1&0&1&0&1&0&0&1&0&1&0&1&0&1&0&0&1&0\\
1&1&0&0&0&1&0&0&1&1&1&0&0&0&1&0&0&1\\
1&0&0&0&1&1&1&0&0&1&0&0&0&1&1&0&1&1\\
0&1&0&1&0&1&0&1&0&0&1&0&1&0&1&1&0&1\\
0&0&1&1&1&0&0&0&1&0&0&1&1&1&0&1&1&0\\
1&0&0&1&0&0&0&1&1&1&0&0&1&0&0&1&0&0\\
0&1&0&0&1&0&1&0&1&0&1&0&0&1&0&0&1&0\\
0&0&1&0&0&1&1&1&0&0&0&1&0&0&1&0&0&1\\
0&1&1&1&0&0&1&0&0&0&1&1&1&0&0&1&0&0\\
1&0&1&0&1&0&0&1&0&1&0&1&0&1&0&0&1&0\\
1&1&0&0&0&1&0&0&1&1&1&0&0&0&1&0&0&1\\
1&0&0&0&1&1&1&0&0&1&0&0&0&1&1&0&1&1\\
0&1&0&1&0&1&0&1&0&0&1&0&1&0&1&1&0&1\\
0&0&1&1&1&0&0&0&1&0&0&1&1&1&0&1&1&0\\
1&0&0&0&1&1&1&0&0&1&0&0&0&1&1&0&1&1\\
0&1&0&1&0&1&0&1&0&0&1&0&1&0&1&1&0&1\\
0&0&1&1&1&0&0&0&1&0&0&1&1&1&0&1&1&0
\end{bmatrix}.$$

上記補題から $\Delta_3{}^{\pm}$ の誘導部分グラフに例 4.22 に示した C_6（6 頂点のサイクルグラフ）や P_5（C_6 を 5 頂点に限定したグラフ）が得られることが分かる．しかも例 4.22 とは異なる生成行列を導出している．以下がその例である．

例 4.26 (例 4.6 とは異なる G$^{\pm}$VSSS-$(\Delta_3,3)$ の構成法)

$$S_0 = \begin{bmatrix}
0&0&1\\0&0&1\\0&0&1\\0&1&0\\0&1&0\\0&1&0
\end{bmatrix},\ S_1 = \begin{bmatrix}
0&0&1\\0&1&0\\1&0&0\\0&0&1\\0&1&0\\1&0&0
\end{bmatrix}.$$

4.2.2 節で示した強正則グラフを用いた構成方法は，白黒反転画像を許容しない GVSSS であった．この構成においては 9 頂点の Lattice graph $L_2(3)$（各頂点から 4 本のエッジを持つ regular グラフ）に対する GVSSS は画像拡大率が 21（スターグラフ分割法は 28）であったが，今回の G$^{\pm}$VSSS の構成では 3 にまで下げることができる．つまり G$^{\pm}$VSSS-$(L_2(3),3)$ を構成しており，しかもこれは optimal な例となっている．

次に $\Delta_3{}^{\pm}$ の誘導部分グラフの分類を行った．このとき，重複のカウントを避けるために independent な事例だけを抽出している．プログラムは GAP [67] と拡張パッケージの一つである Grape package [68] を用いた．また，付録 A に GVSSS-$(G, 3)$ の分類に利用した GAP コードを掲載した．

以下，パラメータ d は与えられたグラフの頂点数を意味する．

$d = 3$（**頂点数 3 のケース**）

$\Delta_3{}^{\pm}$ の頂点インデックス $(1, 2, 3)$: $\sim K_3$

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

注意 4.27 上記の K_3 が唯一 $m^* = 3$ を満たすグラフであり，これは，$m = 2$ で構成できないため *optimal* である．P_3 が数え上げられていない理由は P_3 は $K_{1,2}(P_2)$ と同型であり $G^{\pm}$*VSSS*-$(P_3, 2)$ が構成可能であるためである．

$d = 4$

$(1, 2, 3, 4)$:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$(1, 2, 4, 6)$: $\sim P_4$

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

注意 4.28 頂点数 *4* の完全グラフ K_4 はリストに存在しない．

$d = 5$

$(1, 2, 3, 4, 5)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,1 \\ 1,1,0,0,0 \\ 1,0,0,0,1 \\ 0,1,0,1,0 \end{bmatrix}$$

$(1,2,3,4,7)$:

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,0,0 \\ 1,1,0,0,0 \\ 1,0,0,0,1 \\ 1,0,0,1,0 \end{bmatrix}$$

$(1,2,3,4,8)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,1 \\ 1,1,0,0,0 \\ 1,0,0,0,0 \\ 0,1,0,0,0 \end{bmatrix}$$

$(1,2,4,6,8)$: $\sim P_5$

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,0,0,1 \\ 1,0,0,1,0 \\ 0,0,1,0,0 \\ 0,1,0,0,0 \end{bmatrix}$$

注意 4.29 頂点数 *5* のサイクルグラフ C_5 はリストに存在しない.

$d = 6$

$(1,2,3,4,5,6)$: $\sim \Delta_3$

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1,2,3,4,5,7)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,0 \\ 1,0,0,1,0,0 \end{bmatrix}$$

$(1,2,3,4,5,9)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,1,0,1,0,0 \\ 0,0,1,0,0,0 \end{bmatrix}$$

$(1,2,4,6,8,9)$: $\sim C_6$

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,1,0,0,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

注意 4.30 頂点数 *6* のサイクルグラフ C_6 はリストに存在しており，もちろんこれも *optimal* である.

$d = 7$

$(1,2,3,4,5,6,7)$:

$$\begin{bmatrix} 0,1,1,1,0,0,1 \\ 1,0,1,0,1,0,0 \\ 1,1,0,0,0,1,0 \\ 1,0,0,0,1,1,1 \\ 0,1,0,1,0,1,0 \\ 0,0,1,1,1,0,0 \\ 1,0,0,1,0,0,0 \end{bmatrix}$$

$(1,2,3,4,5,7,9)$:

$$\begin{bmatrix} 0,1,1,1,0,1,0 \\ 1,0,1,0,1,0,0 \\ 1,1,0,0,0,0,1 \\ 1,0,0,0,1,1,0 \\ 0,1,0,1,0,0,0 \\ 1,0,0,1,0,0,1 \\ 0,0,1,0,0,1,0 \end{bmatrix}$$

$d = 8$

$(1, 2, 3, 4, 5, 6, 7, 8)$:

$$\begin{bmatrix} 0,1,1,1,0,0,1,0 \\ 1,0,1,0,1,0,0,1 \\ 1,1,0,0,0,1,0,0 \\ 1,0,0,0,1,1,1,0 \\ 0,1,0,1,0,1,0,1 \\ 0,0,1,1,1,0,0,0 \\ 1,0,0,1,0,0,0,1 \\ 0,1,0,0,1,0,1,0 \end{bmatrix}$$

注意 4.31 頂点数 *8* のサイクルグラフ C_8 はリストに存在しない.

$d = 9$

$(1, 2, 3, 4, 5, 6, 7, 8, 9)$: $\sim L_2(3)$

$$\begin{bmatrix} 0,1,1,1,0,0,1,0,0 \\ 1,0,1,0,1,0,0,1,0 \\ 1,1,0,0,0,1,0,0,1 \\ 1,0,0,0,1,1,1,0,0 \\ 0,1,0,1,0,1,0,1,0 \\ 0,0,1,1,1,0,0,0,1 \\ 1,0,0,1,0,0,0,1,1 \\ 0,1,0,0,1,0,1,0,1 \\ 0,0,1,0,0,1,1,1,0 \end{bmatrix}$$

注意 4.32 $d \geq 10$ のときには *independent* な誘導部分グラフは存在しない．このリストにおいて $K_3, \Delta_3, C_6, L_2(3)$ のみが *regular* なグラフである.

これらの結果を踏まえ，最終的に以下の定理を得ることができる.

定理 4.33 $^\exists$ *independent* $G^{\pm}$*VSSS-*$(G, 3) \Leftrightarrow G \in Ind(L_2(3))$.

これは，$\Delta_3{}^{\pm}$ に含まれる independent な誘導部分グラフの集合は $L_2(3)$ に含まれる independent な誘導部分グラフの集合に包括されているためである．グラフのリストについては変化はなく $d \leq 9$ までで抑えられている点にも注意する.

4.4.2 GVSSS-$(G, 3)$ の分類

前節の G$^{\pm}$VSSS-$(G, 3)$ の分類を踏まえ，従来型 GVSSS の $m^* = 3$ のケースについて分類を試みる.$\Delta_3^{\pm}$ の構成方法は，定理 4.11 と同様の方式であり，以下のように構成している.

$$\tilde{S_0} = \begin{bmatrix} 0,0,1 \\ 0,0,1 \\ 0,0,1 \\ 0,1,0 \\ 0,1,0 \\ 0,1,0 \\ 1,0,0 \\ 1,0,0 \\ 1,0,0 \\ 1,1,0 \\ 1,1,0 \\ 1,1,0 \\ 1,0,1 \\ 1,0,1 \\ 1,0,1 \\ 0,1,1 \\ 0,1,1 \\ 0,1,1 \end{bmatrix}, \tilde{S_1} = \begin{bmatrix} 0,0,1 \\ 0,1,0 \\ 1,0,0 \\ 0,0,1 \\ 0,1,0 \\ 1,0,0 \\ 0,0,1 \\ 0,1,0 \\ 1,0,0 \\ 1,1,0 \\ 1,0,1 \\ 0,1,1 \\ 1,1,0 \\ 1,0,1 \\ 0,1,1 \\ 1,1,0 \\ 1,0,1 \\ 0,1,1 \end{bmatrix}.$$

上記行列の $R(\tilde{S_1}) - R(\tilde{S_0})$ は次の通りである.

$$R(\tilde{S_1}) - R(\tilde{S_0}) = \begin{bmatrix} 0 & + & + & - & 0 & 0 & - & 0 & 0 & 0 & - & - & + & 0 & 0 & + & 0 & 0 \\ + & 0 & + & 0 & - & 0 & 0 & - & 0 & - & 0 & - & 0 & + & 0 & 0 & + & 0 \\ + & + & 0 & 0 & 0 & - & 0 & 0 & - & - & - & 0 & 0 & 0 & + & 0 & 0 & + \\ - & 0 & 0 & 0 & + & + & - & 0 & 0 & + & 0 & 0 & 0 & - & - & + & 0 & 0 \\ 0 & - & 0 & + & 0 & + & 0 & - & 0 & 0 & + & 0 & - & 0 & - & 0 & + & 0 \\ 0 & 0 & - & + & + & 0 & 0 & 0 & - & 0 & 0 & + & - & - & 0 & 0 & 0 & + \\ - & 0 & 0 & - & 0 & 0 & 0 & + & + & + & 0 & 0 & + & 0 & 0 & 0 & - & - \\ 0 & - & 0 & 0 & - & 0 & + & 0 & + & 0 & + & 0 & 0 & + & 0 & - & 0 & - \\ 0 & 0 & - & 0 & 0 & - & + & + & 0 & 0 & 0 & + & 0 & 0 & + & - & - & 0 \\ 0 & - & - & + & 0 & 0 & + & 0 & 0 & 0 & + & + & - & 0 & 0 & - & 0 & 0 \\ - & 0 & - & 0 & + & 0 & 0 & + & 0 & + & 0 & + & 0 & - & 0 & 0 & - & 0 \\ - & - & 0 & 0 & 0 & + & 0 & 0 & + & + & + & 0 & 0 & 0 & - & 0 & 0 & - \\ + & 0 & 0 & 0 & - & - & + & 0 & 0 & - & 0 & 0 & 0 & + & + & - & 0 & 0 \\ 0 & + & 0 & - & 0 & - & 0 & + & 0 & 0 & - & 0 & + & 0 & + & 0 & - & 0 \\ 0 & 0 & + & - & - & 0 & 0 & 0 & + & 0 & 0 & - & + & + & 0 & 0 & 0 & - \\ + & 0 & 0 & + & 0 & 0 & 0 & - & - & - & 0 & 0 & - & 0 & 0 & 0 & + & + \\ 0 & + & 0 & 0 & + & 0 & - & 0 & - & 0 & - & 0 & 0 & - & 0 & + & 0 & + \\ 0 & 0 & + & 0 & 0 & + & - & - & 0 & 0 & 0 & - & 0 & 0 & - & + & + & 0 \end{bmatrix}.$$

ただし $+$ は 1 を $-$ は -1 を表している．このとき同じインデックスの行・列において $-$ が含まれているものを削除し，最大の部分行列を得ることを考える．具体的には,

$$R(\tilde{S_1}) - R(\tilde{S_0}) = \begin{bmatrix}
\mathbf{0} & + & + & - & 0 & 0 & - & 0 & 0 & 0 & - & - & + & 0 & 0 & + & \mathbf{0} & \mathbf{0} \\
+ & \mathbf{0} & + & 0 & - & 0 & 0 & - & 0 & - & 0 & - & 0 & + & 0 & \mathbf{0} & + & \mathbf{0} \\
+ & + & \mathbf{0} & 0 & 0 & - & 0 & 0 & - & - & - & 0 & 0 & 0 & + & \mathbf{0} & \mathbf{0} & + \\
- & 0 & 0 & 0 & + & + & - & 0 & 0 & + & 0 & 0 & 0 & - & - & + & 0 & 0 \\
0 & - & 0 & + & 0 & + & 0 & - & 0 & 0 & + & 0 & - & 0 & - & 0 & + & 0 \\
0 & 0 & - & + & + & 0 & 0 & 0 & - & 0 & 0 & + & - & - & 0 & 0 & 0 & + \\
- & 0 & 0 & - & 0 & 0 & 0 & + & + & + & 0 & 0 & + & 0 & 0 & 0 & - & - \\
0 & - & 0 & 0 & - & 0 & + & 0 & + & 0 & + & 0 & 0 & + & 0 & - & 0 & - \\
0 & 0 & - & 0 & 0 & - & + & + & 0 & 0 & 0 & + & 0 & 0 & + & - & - & 0 \\
0 & - & - & + & 0 & 0 & + & 0 & 0 & 0 & + & + & - & 0 & 0 & - & 0 & 0 \\
- & 0 & - & 0 & + & 0 & 0 & + & 0 & + & 0 & + & 0 & - & 0 & 0 & - & 0 \\
- & - & 0 & 0 & 0 & + & 0 & 0 & + & + & + & 0 & 0 & 0 & - & 0 & 0 & - \\
+ & 0 & 0 & 0 & - & - & + & 0 & 0 & - & 0 & 0 & 0 & + & + & - & 0 & 0 \\
0 & + & 0 & - & 0 & - & 0 & + & 0 & 0 & - & 0 & + & 0 & + & 0 & - & 0 \\
0 & 0 & + & - & - & 0 & 0 & 0 & + & 0 & 0 & - & + & + & 0 & 0 & 0 & - \\
+ & \mathbf{0} & \mathbf{0} & + & 0 & 0 & 0 & - & - & - & 0 & 0 & - & 0 & 0 & \mathbf{0} & + & + \\
\mathbf{0} & + & \mathbf{0} & 0 & + & 0 & - & 0 & - & 0 & - & 0 & 0 & - & 0 & + & \mathbf{0} & + \\
\mathbf{0} & \mathbf{0} & + & 0 & 0 & + & - & - & 0 & 0 & 0 & - & 0 & 0 & - & + & + & \mathbf{0}
\end{bmatrix}.$$

のうちインデックス 1,2,3,16,17,18 だけを残すと以下の行列を得る．

$$\begin{array}{r} 1) \\ 2) \\ 3) \\ 16) \\ 17) \\ 18) \end{array} \begin{bmatrix}
0 & + & + & + & 0 & 0 \\
+ & 0 & + & 0 & + & 0 \\
+ & + & 0 & 0 & 0 & + \\
+ & 0 & 0 & 0 & + & + \\
0 & + & 0 & + & 0 & + \\
0 & 0 & + & + & + & 0
\end{bmatrix}$$

この行列は全て + で構成されていることが分かる．さらに，この行列を隣接行列と持つグラフは independent であることが分かる．そのほかにもインデックス 1,2,3,13,14,15 や インデックス 7,8,9,10,11,12 などの選択肢があるが，いずれもこの行列はちょうど グラフ Δ_3 の隣接行列と一致し，次の定理を得る．

定理 4.34 $^{\exists}$ *independent GVSSS-*$(G, 3) \Leftrightarrow G \in Ind(\Delta_3)$.

Δ_3 に含まれる頂点数 3 以上の independent な誘導部分グラフは 5 つであり，以下，列挙する．これらの誘導部分グラフについては例 4.22 に見たように従来の GVSSS でも構成可能であることを示している．

- $d = 3$: $(1, 2, 3)$: $\sim K_3$
- $d = 4$: $(1, 2, 3, 4)$
- $d = 4$: $(1, 2, 4, 6)$: $\sim P_4$
- $d = 5$: $(1, 2, 3, 4, 5)$
- $d = 6$: $(1, 2, 3, 4, 5, 6)$: $\sim \Delta_3$

4.4.3 optimal GVSSS-$(G,3)$，G$^{\pm}$VSSS-$(G,3)$ で構成される Hasse diagram

誘導部分グラフの関係において，異なる 2 つのグラフの包含関係に半順序の関係を導入することで図 4.6 のように Hasse diagram を構成することができる．色付きの箇所は従来型 GVSSS でカバーできている範囲であることから，G$^{\pm}$VSSS で構成できるグラフが多く存在していることが分かる．

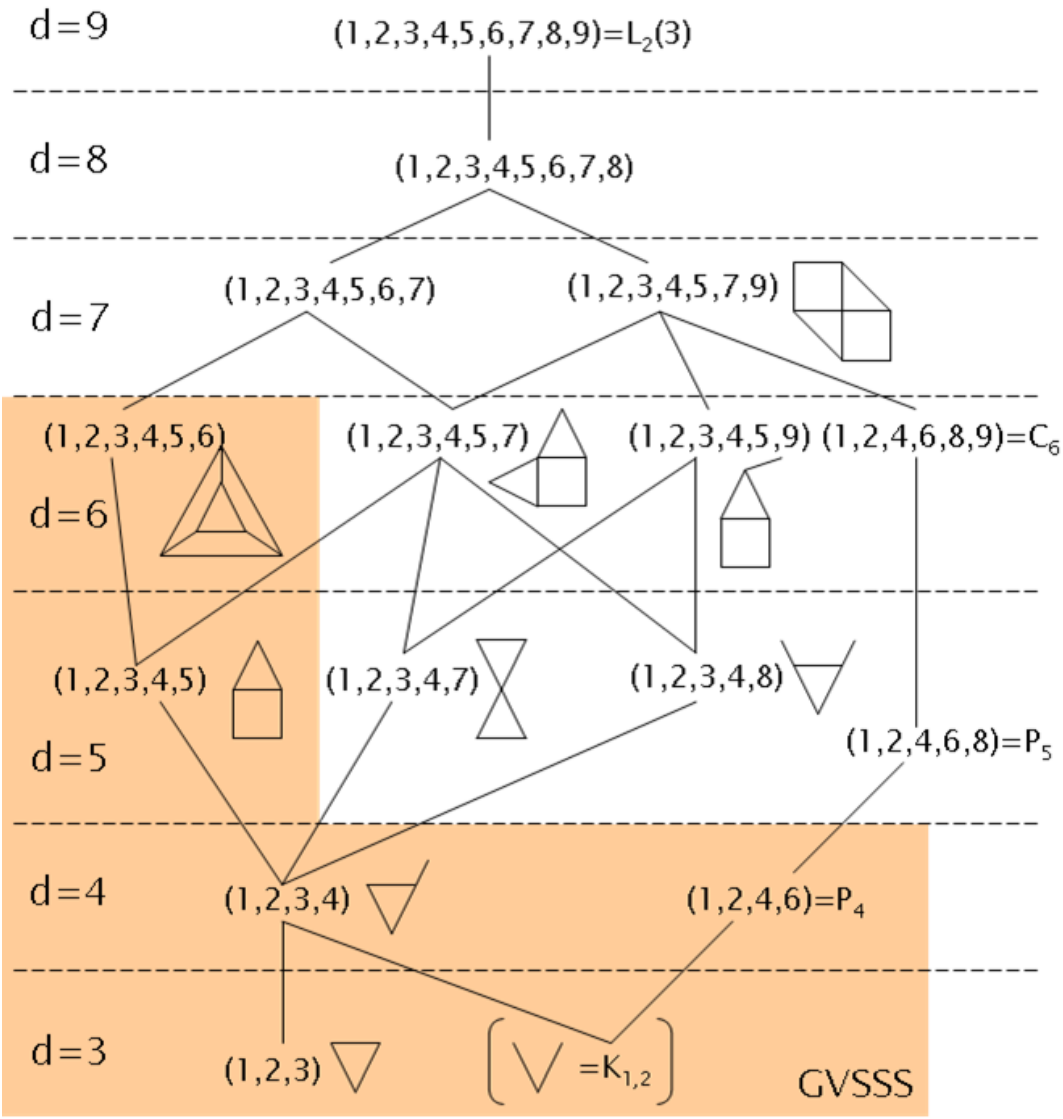

図 4.6: optimal G$^{\pm}$VSSS-$(G,3)$ で構成される Hasse diagram

optimal GVSSS-$(G,3)$，G$^{\pm}$VSSS-$(G,3)$ を満たすグラフの総数

optimal な GVSSS-$(G,3)$ および $^{\pm}$VSSS-$(G,3)$ を満たすグラフの総数は以下のとおりである．

表 4.2: optimal GVSSS-$(G,3)$，G$^{\pm}$VSSS-$(G,3)$ を満たすグラフの総数

d =	3	4	5	6	7	8	9	≥10
GVSSS-$(G,3)$	1	2	1	1	0	0	0	0
G$^{\pm}$VSSS-$(G,3)$	1	2	4	4	2	1	1	0

4.5 画像拡大率が 4 の GVSSS,G$^{\pm}$VSSS の分類

4.5.1 GVSSS-$(G,4)$ の分類

次に 4.4.2 節と同様に GVSSS-$(G,4)$ の分類を試みる．S_0, S_1 の行ベクトルとして可能なものを列挙すると以下のとおりである：$\tilde{S_0} = [$

```
[0,0,0,1],[0,0,0,1],[0,0,0,1],[0,0,0,1],
[0,0,1,0],[0,0,1,0],[0,0,1,0],[0,0,1,0],
[0,1,0,0],[0,1,0,0],[0,1,0,0],[0,1,0,0],
[1,0,0,0],[1,0,0,0],[1,0,0,0],[1,0,0,0],
[0,0,1,1],[0,0,1,1],[0,0,1,1],
[0,0,1,1],[0,0,1,1],[0,0,1,1],
[0,1,0,1],[0,1,0,1],[0,1,0,1],
[0,1,0,1],[0,1,0,1],[0,1,0,1],
[1,0,0,1],[1,0,0,1],[1,0,0,1],
[1,0,0,1],[1,0,0,1],[1,0,0,1],
[0,1,1,0],[0,1,1,0],[0,1,1,0],
[0,1,1,0],[0,1,1,0],[0,1,1,0],
[1,0,1,0],[1,0,1,0],[1,0,1,0],
[1,0,1,0],[1,0,1,0],[1,0,1,0],
[1,1,0,0],[1,1,0,0],[1,1,0,0],
[1,1,0,0],[1,1,0,0],[1,1,0,0],
[0,1,1,1],[0,1,1,1],[0,1,1,1],[0,1,1,1],
[1,0,1,1],[1,0,1,1],[1,0,1,1],[1,0,1,1],
[1,1,0,1],[1,1,0,1],[1,1,0,1],[1,1,0,1],
[1,1,1,0],[1,1,1,0],[1,1,1,0],[1,1,1,0]
```

],

$\tilde{S}_1 = [$

```
[0,0,0,1],[0,0,1,0],[0,1,0,0],[1,0,0,0],
[0,0,0,1],[0,0,1,0],[0,1,0,0],[1,0,0,0],
[0,0,0,1],[0,0,1,0],[0,1,0,0],[1,0,0,0],
[0,0,0,1],[0,0,1,0],[0,1,0,0],[1,0,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,0,1,1],[0,1,0,1],[1,0,0,1],
[0,1,1,0],[1,0,1,0],[1,1,0,0],
[0,1,1,1],[1,0,1,1],[1,1,0,1],[1,1,1,0],
[0,1,1,1],[1,0,1,1],[1,1,0,1],[1,1,1,0],
[0,1,1,1],[1,0,1,1],[1,1,0,1],[1,1,1,0],
[0,1,1,1],[1,0,1,1],[1,1,0,1],[1,1,1,0]
```

$].$

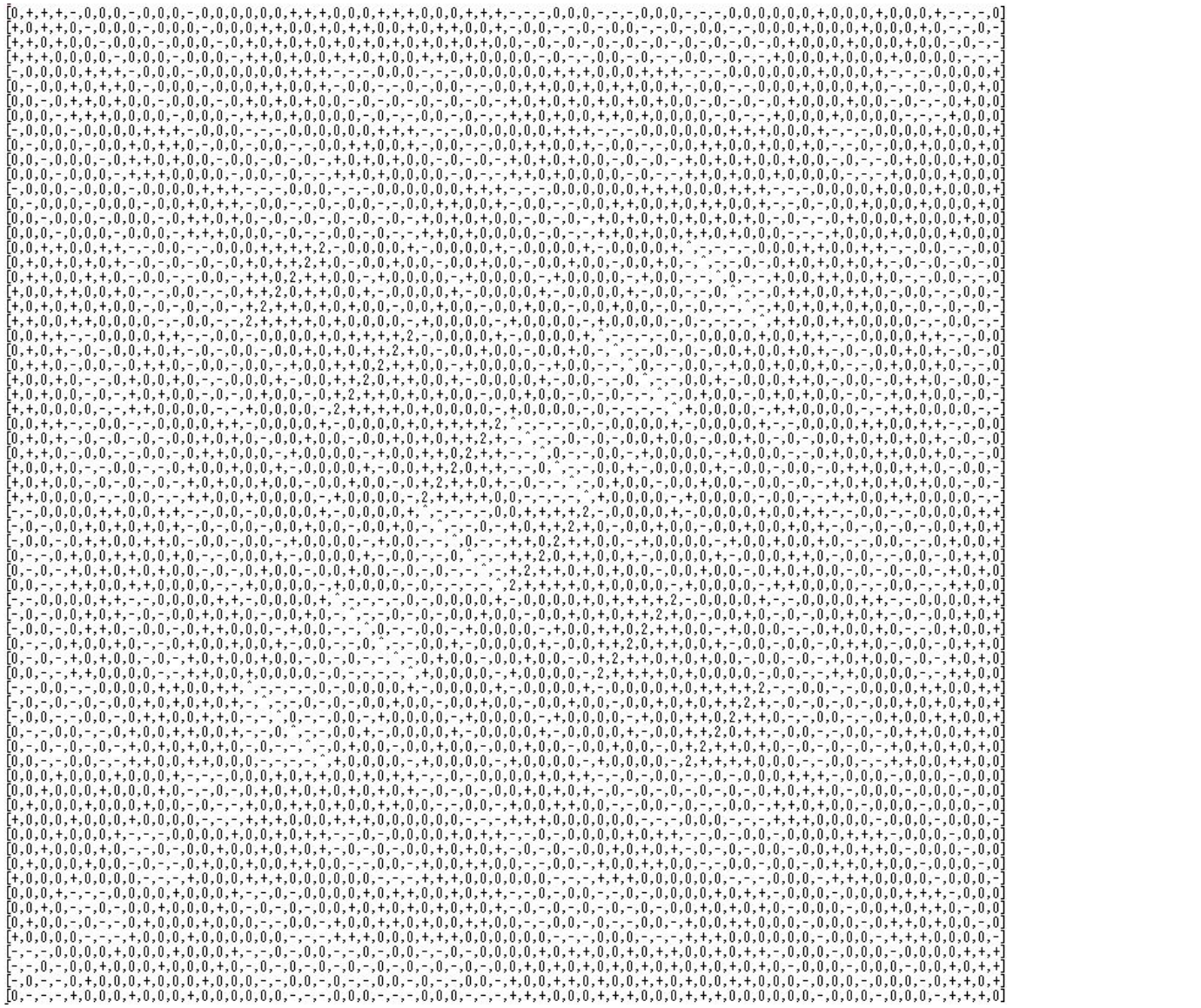

図 4.7: 行列 $R(\tilde{S}_1) - R(\tilde{S}_0)$

図 4.7 のように 68×68 正方行列 $R(\tilde{S}_1) - R(\tilde{S}_0)$ を構成し，この行列のうち非負の要素のみで構成でき，かつ independent な行列を構成していく．ここで，図 4.7 において $+$ は 1 を $-$ は -1 を ˆ は -2 を表現している．

具体的にはインデックス 1,2,3,4,17,18,19,20,21,22,53,54,55,56 を用いて得られたグラフは以下のとおりであり，これを Δ_4 とすると以下の定理を得る．

$$Adj(\Delta_4) = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

定理 4.35 $^{\exists}$ *independent GVSSS-*$(G, 4) \Leftrightarrow G \in Ind(\Delta_4)$.

Δ_4 に含まれる頂点数 3 以上の independent な誘導部分グラフを列挙しておく．これらは，復元画像として白黒反転画像を許容しない従来型 GVSSS を構成可能なグラフの集合である．紙面の都合上，グラフの隣接行列は掲載せず，インデックスのみ掲載しているケースもあることに留意する．

$d = 3$（**頂点数 3 のケース**）

Δ_4 の頂点インデックス $(1, 2, 3)$: $\sim K_3$

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

$d = 4$

$(1, 2, 3, 4)$: $\sim K_4$

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$(1,2,3,5)$:

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$(1,2,5,6)$: $\sim P_4$

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix}$$

$d = 5$

$(1,2,3,4,5)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,1,0 \\ 1,1,0,1,1 \\ 1,1,1,0,1 \\ 0,0,1,1,0 \end{bmatrix}$$

$(1,2,3,4,11)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,1,0 \\ 1,1,0,1,0 \\ 1,1,1,0,1 \\ 0,0,0,1,0 \end{bmatrix}$$

$(1,2,3,5,6)$: $\in Ind(\Delta_3)$

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,1,0,1 \\ 1,1,0,1,0 \\ 0,0,1,0,1 \\ 0,1,0,1,0 \end{bmatrix}$$

$(1,2,3,5,7)$:

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,1,0,1 \\ 1,1,0,1,1 \\ 0,0,1,0,1 \\ 0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 7, 11)$:

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,1,1,0 \\ 1,1,0,1,0 \\ 0,1,1,0,1 \\ 0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 11, 12)$:

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,1,0,0 \\ 1,1,0,0,1 \\ 0,0,0,0,1 \\ 0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 6, 7, 11)$:

$$\begin{bmatrix} 0,1,0,0,0 \\ 1,0,1,1,0 \\ 0,1,0,1,0 \\ 0,1,1,0,1 \\ 0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 6, 11, 12)$: $\sim P_5$

$$\begin{bmatrix} 0,1,0,0,0 \\ 1,0,1,0,0 \\ 0,1,0,0,1 \\ 0,0,0,0,1 \\ 0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 10, 11, 12)$:

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,1,0,0 \\ 1,1,0,1,1 \\ 0,0,1,0,1 \\ 0,0,1,1,0 \end{bmatrix}$$

$(5, 6, 7, 8, 9)$: $\sim K_5$

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,1,1 \\ 1,1,0,1,1 \\ 1,1,1,0,1 \\ 1,1,1,1,0 \end{bmatrix}$$

注意 4.36 *GVSSS-*$(G, 3)$ においては Δ_3 の *1* 点を削除した誘導部分グラフの *1* 種類（インデックス $(1, 2, 3, 5, 6)$）しか構成できなかったが，*GVSSS-*$(G, 4)$ においては *10* 種類のグラフで構成可能である．インデックス $(1, 2, 3, 5, 6)$ 以外のグラフはすべて $m^* = 4$ において *optimal* な例となる．

$d = 6$

$(1, 2, 3, 4, 5, 6)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,1 \\ 0,0,1,1,0,1 \\ 0,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 10)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,0 \\ 0,0,1,1,0,1 \\ 1,1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 11)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,1 \\ 0,0,1,1,0,0 \\ 0,0,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 13)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,0 \\ 0,0,1,1,0,1 \\ 0,1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 11, 12)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,1 \\ 1,1,1,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 7)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,1,1 \\ 1,1,0,1,0,1 \\ 0,0,1,0,1,1 \\ 0,1,0,1,0,1 \\ 0,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 8)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,1,0,1,0 \\ 1,1,0,1,0,0 \\ 0,0,1,0,1,1 \\ 0,1,0,1,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 9)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,1,0,1,0 \\ 1,1,0,1,0,1 \\ 0,0,1,0,1,1 \\ 0,1,0,1,0,1 \\ 1,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 7, 11)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,1,1,0 \\ 0,0,1,0,1,0 \\ 0,1,1,1,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 7, 12)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,1,1,1 \\ 0,0,1,0,1,0 \\ 0,1,1,1,0,0 \\ 0,0,1,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 7, 13)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,1,1 \\ 1,1,0,1,1,0 \\ 0,0,1,0,1,1 \\ 0,1,1,1,0,0 \\ 0,1,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 11)$:

$$\begin{bmatrix} 0,1,1,0,1,0 \\ 1,0,1,0,1,0 \\ 1,1,0,1,0,0 \\ 0,0,1,0,1,0 \\ 1,1,0,1,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 11, 12)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,0,0 \\ 1,1,0,1,0,1 \\ 0,0,1,0,0,0 \\ 0,0,0,0,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 11, 13)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,0,0,0,0,1 \\ 0,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 7, 11, 12)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,1 \\ 0,1,1,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 7, 11, 14)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,0 \\ 0,1,1,0,1,1 \\ 0,0,0,1,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 11, 12, 13)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,1,0 \\ 0,0,0,0,1,1 \\ 0,0,1,1,0,1 \\ 0,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 6, 7, 10)$:

$$\begin{bmatrix} 0,1,0,0,0,1 \\ 1,0,0,1,1,1 \\ 0,0,0,1,1,1 \\ 0,1,1,0,1,1 \\ 0,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 6, 7, 11)$:

$$\begin{bmatrix} 0,1,0,0,0,0 \\ 1,0,0,1,1,0 \\ 0,0,0,1,1,0 \\ 0,1,1,0,1,0 \\ 0,1,1,1,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 5, 6, 9, 11)$:

$$\begin{bmatrix} 0,1,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,0,1,1,0 \\ 0,1,1,0,1,0 \\ 1,0,1,1,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 5, 6, 11, 12)$:

$$\begin{bmatrix} 0,1,0,0,0,0 \\ 1,0,0,1,0,0 \\ 0,0,0,1,0,0 \\ 0,1,1,0,0,1 \\ 0,0,0,0,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 6, 11, 13)$:

$$\begin{bmatrix} 0,1,0,0,0,0 \\ 1,0,0,1,0,1 \\ 0,0,0,1,0,1 \\ 0,1,1,0,0,0 \\ 0,0,0,0,0,1 \\ 0,1,1,0,1,0 \end{bmatrix}$$

$(1, 2, 5, 10, 11, 12)$:

$$\begin{bmatrix} 0,1,0,1,0,0 \\ 1,0,0,1,0,0 \\ 0,0,0,1,0,0 \\ 1,1,1,0,1,1 \\ 0,0,0,1,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 6, 7, 10, 11)$:

$$\begin{bmatrix} 0,1,0,0,1,0 \\ 1,0,1,1,1,0 \\ 0,1,0,1,1,0 \\ 0,1,1,0,1,1 \\ 1,1,1,1,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 6, 7, 11, 12)$:

$$\begin{bmatrix} 0,1,0,0,0,0 \\ 1,0,1,1,0,0 \\ 0,1,0,1,0,1 \\ 0,1,1,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 6, 9, 11, 12)$:

$$\begin{bmatrix} 0,1,0,1,0,0 \\ 1,0,1,0,0,0 \\ 0,1,0,1,0,1 \\ 1,0,1,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 6, 10, 11, 12)$:

$$\begin{bmatrix} 0,1,0,1,0,0 \\ 1,0,1,1,0,0 \\ 0,1,0,1,0,1 \\ 1,1,1,0,1,1 \\ 0,0,0,1,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1, 5, 6, 7, 8, 9)$:

$$\begin{bmatrix} 0,0,0,0,1,1 \\ 0,0,1,1,1,1 \\ 0,1,0,1,1,1 \\ 0,1,1,0,1,1 \\ 1,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 5, 6, 7, 8, 11)$:

$$\begin{bmatrix} 0,0,0,0,1,0 \\ 0,0,1,1,1,0 \\ 0,1,0,1,1,0 \\ 0,1,1,0,1,1 \\ 1,1,1,1,0,0 \\ 0,0,0,1,0,0 \end{bmatrix}$$

$(1, 5, 6, 8, 9, 10)$:

$$\begin{bmatrix} 0,0,0,1,1,1 \\ 0,0,1,1,1,1 \\ 0,1,0,1,1,1 \\ 1,1,1,0,1,1 \\ 1,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 5, 8, 9, 10, 13)$:

$$\begin{bmatrix} 0,0,1,1,1,0 \\ 0,0,1,1,1,1 \\ 1,1,0,1,1,1 \\ 1,1,1,0,1,1 \\ 1,1,1,1,0,0 \\ 0,1,1,1,0,0 \end{bmatrix}$$

$(5, 6, 7, 8, 9, 10)$:

$$\begin{bmatrix} 0,1,1,1,1,1 \\ 1,0,1,1,1,1 \\ 1,1,0,1,1,1 \\ 1,1,1,0,1,1 \\ 1,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$d = 7$

$(1, 2, 3, 4, 5, 6, 7)$:
$(1, 2, 3, 4, 5, 6, 8)$:
$(1, 2, 3, 4, 5, 6, 9)$:
$(1, 2, 3, 4, 5, 6, 11)$:
$(1, 2, 3, 4, 5, 6, 12)$:
$(1, 2, 3, 4, 5, 6, 14)$:
$(1, 2, 3, 4, 5, 10, 11)$:
$(1, 2, 3, 4, 5, 11, 12)$:
$(1, 2, 3, 4, 5, 11, 13)$:
$(1, 2, 3, 4, 5, 13, 14)$:
$(1, 2, 3, 4, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 9)$:
$(1, 2, 3, 5, 6, 7, 11)$:
$(1, 2, 3, 5, 6, 7, 12)$:

$(1, 2, 3, 5, 6, 9, 11)$:
$(1, 2, 3, 5, 6, 9, 12)$:
$(1, 2, 3, 5, 6, 9, 14)$:
$(1, 2, 3, 5, 6, 11, 12)$:
$(1, 2, 3, 5, 6, 11, 14)$:
$(1, 2, 3, 5, 7, 9, 11)$:
$(1, 2, 3, 5, 7, 9, 13)$:
$(1, 2, 3, 5, 7, 10, 11)$:
$(1, 2, 3, 5, 7, 11, 12)$:
$(1, 2, 3, 5, 7, 11, 13)$:
$(1, 2, 3, 5, 7, 11, 14)$:
$(1, 2, 3, 5, 7, 12, 13)$:
$(1, 2, 3, 5, 10, 11, 12)$:
$(1, 2, 3, 5, 10, 11, 13)$:
$(1, 2, 3, 5, 11, 12, 13)$:
$(1, 2, 3, 5, 11, 13, 14)$:
$(1, 2, 3, 7, 9, 11, 12)$:
$(1, 2, 3, 7, 9, 11, 13)$:
$(1, 2, 3, 7, 11, 12, 14)$:
$(1, 2, 5, 6, 7, 8, 9)$:
$(1, 2, 5, 6, 7, 8, 10)$:
$(1, 2, 5, 6, 7, 8, 11)$:
$(1, 2, 5, 6, 7, 8, 12)$:
$(1, 2, 5, 6, 7, 10, 11)$:
$(1, 2, 5, 6, 7, 10, 13)$:
$(1, 2, 5, 6, 7, 10, 14)$:
$(1, 2, 5, 6, 7, 11, 12)$:
$(1, 2, 5, 6, 7, 11, 13)$:
$(1, 2, 5, 6, 9, 10, 11)$:
$(1, 2, 5, 6, 9, 11, 12)$:
$(1, 2, 5, 6, 10, 11, 12)$:
$(1, 2, 6, 7, 8, 9, 10)$:
$(1, 2, 6, 7, 8, 9, 11)$:
$(1, 2, 6, 7, 8, 10, 12)$:
$(1, 2, 6, 7, 8, 11, 12)$:
$(1, 2, 6, 7, 10, 11, 12)$:
$(1, 2, 6, 9, 10, 11, 12)$:
$(1, 5, 6, 7, 8, 9, 10)$:

$(1, 5, 6, 7, 8, 9, 11)$:
$(1, 5, 6, 7, 8, 9, 13)$:
$(1, 5, 6, 7, 8, 9, 14)$:
$(1, 5, 6, 8, 9, 10, 12)$:

$d = 8$

$(1, 2, 3, 4, 5, 6, 7, 8)$:
$(1, 2, 3, 4, 5, 6, 7, 11)$:
$(1, 2, 3, 4, 5, 6, 7, 14)$:
$(1, 2, 3, 4, 5, 6, 8, 11)$:
$(1, 2, 3, 4, 5, 6, 8, 12)$:
$(1, 2, 3, 4, 5, 6, 9, 10)$:
$(1, 2, 3, 4, 5, 6, 9, 11)$:
$(1, 2, 3, 4, 5, 6, 9, 13)$:
$(1, 2, 3, 4, 5, 6, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 11, 14)$:
$(1, 2, 3, 4, 5, 6, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 12, 14)$:
$(1, 2, 3, 4, 5, 10, 11, 12)$:
$(1, 2, 3, 4, 5, 10, 11, 13)$:
$(1, 2, 3, 4, 5, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 11, 13, 14)$:
$(1, 2, 3, 4, 11, 12, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 8, 9)$:
$(1, 2, 3, 5, 6, 7, 8, 11)$:
$(1, 2, 3, 5, 6, 7, 8, 12)$:
$(1, 2, 3, 5, 6, 7, 9, 10)$:
$(1, 2, 3, 5, 6, 7, 9, 11)$:
$(1, 2, 3, 5, 6, 7, 9, 12)$:
$(1, 2, 3, 5, 6, 7, 9, 13)$:
$(1, 2, 3, 5, 6, 7, 9, 14)$:
$(1, 2, 3, 5, 6, 7, 11, 12)$:
$(1, 2, 3, 5, 6, 7, 11, 14)$:
$(1, 2, 3, 5, 6, 7, 12, 13)$:
$(1, 2, 3, 5, 6, 8, 11, 12)$:
$(1, 2, 3, 5, 6, 9, 10, 11)$:
$(1, 2, 3, 5, 6, 9, 10, 12)$:
$(1, 2, 3, 5, 6, 9, 10, 14)$:

$(1, 2, 3, 5, 6, 9, 11, 12)$:
$(1, 2, 3, 5, 6, 9, 11, 13)$:
$(1, 2, 3, 5, 6, 9, 11, 14)$:
$(1, 2, 3, 5, 6, 9, 12, 14)$:
$(1, 2, 3, 5, 6, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 11, 12, 14)$:
$(1, 2, 3, 5, 7, 9, 10, 11)$:
$(1, 2, 3, 5, 7, 9, 11, 12)$:
$(1, 2, 3, 5, 7, 9, 11, 13)$:
$(1, 2, 3, 5, 7, 9, 12, 13)$:
$(1, 2, 3, 5, 7, 9, 13, 14)$:
$(1, 2, 3, 5, 7, 10, 11, 12)$:
$(1, 2, 3, 5, 7, 10, 11, 13)$:
$(1, 2, 3, 5, 7, 10, 11, 14)$:
$(1, 2, 3, 5, 7, 11, 12, 14)$:
$(1, 2, 3, 5, 7, 11, 13, 14)$:
$(1, 2, 3, 5, 10, 11, 12, 13)$:
$(1, 2, 3, 5, 10, 11, 13, 14)$:
$(1, 2, 3, 7, 9, 10, 11, 12)$:
$(1, 2, 3, 7, 9, 11, 13, 14)$:
$(1, 2, 5, 6, 7, 8, 9, 10)$:
$(1, 2, 5, 6, 7, 8, 9, 11)$:
$(1, 2, 5, 6, 7, 8, 9, 13)$:
$(1, 2, 5, 6, 7, 8, 10, 11)$:
$(1, 2, 5, 6, 7, 8, 10, 12)$:
$(1, 2, 5, 6, 7, 8, 10, 13)$:
$(1, 2, 5, 6, 7, 8, 10, 14)$:
$(1, 2, 5, 6, 7, 8, 11, 12)$:
$(1, 2, 5, 6, 7, 8, 11, 13)$:
$(1, 2, 5, 6, 7, 8, 11, 14)$:
$(1, 2, 5, 6, 7, 10, 11, 12)$:
$(1, 2, 5, 6, 7, 10, 11, 14)$:
$(1, 2, 5, 6, 7, 10, 13, 14)$:
$(1, 2, 5, 6, 9, 10, 11, 12)$:
$(1, 2, 5, 6, 9, 10, 11, 13)$:
$(1, 2, 5, 6, 9, 10, 11, 14)$:
$(1, 2, 6, 7, 8, 9, 10, 11)$:
$(1, 2, 6, 7, 8, 9, 10, 13)$:

$(1, 2, 6, 7, 8, 9, 11, 12)$:
$(1, 2, 6, 7, 8, 10, 11, 12)$:
$(1, 2, 6, 7, 8, 10, 12, 14)$:
$(1, 5, 6, 7, 8, 9, 10, 11)$:
$(1, 5, 6, 7, 8, 9, 10, 14)$:

$d = 9$

$(1, 2, 3, 4, 5, 6, 7, 8, 9)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 11)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 12)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 14)$:
$(1, 2, 3, 4, 5, 6, 7, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 7, 11, 14)$:
$(1, 2, 3, 4, 5, 6, 8, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 8, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 9, 10, 11)$:
$(1, 2, 3, 4, 5, 6, 9, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 9, 11, 13)$:
$(1, 2, 3, 4, 5, 6, 9, 11, 14)$:
$(1, 2, 3, 4, 5, 6, 9, 13, 14)$:
$(1, 2, 3, 4, 5, 6, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 11, 12, 14)$:
$(1, 2, 3, 4, 5, 6, 12, 13, 14)$:
$(1, 2, 3, 4, 5, 10, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 11, 12, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 10)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 11)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 12)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 13)$:
$(1, 2, 3, 5, 6, 7, 8, 11, 12)$:
$(1, 2, 3, 5, 6, 7, 8, 11, 14)$:
$(1, 2, 3, 5, 6, 7, 8, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 11)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 12)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 11, 12)$:
$(1, 2, 3, 5, 6, 7, 9, 11, 13)$:
$(1, 2, 3, 5, 6, 7, 9, 11, 14)$:

$(1, 2, 3, 5, 6, 7, 9, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 9, 12, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 11, 12, 14)$:
$(1, 2, 3, 5, 6, 9, 10, 11, 12)$:
$(1, 2, 3, 5, 6, 9, 10, 11, 14)$:
$(1, 2, 3, 5, 6, 9, 10, 12, 13)$:
$(1, 2, 3, 5, 6, 9, 10, 12, 14)$:
$(1, 2, 3, 5, 6, 9, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 9, 11, 12, 14)$:
$(1, 2, 3, 5, 6, 9, 11, 13, 14)$:
$(1, 2, 3, 5, 7, 9, 10, 11, 12)$:
$(1, 2, 3, 5, 7, 9, 10, 11, 13)$:
$(1, 2, 3, 5, 7, 9, 11, 13, 14)$:
$(1, 2, 3, 5, 7, 10, 11, 12, 14)$:
$(1, 2, 3, 5, 7, 10, 11, 13, 14)$:
$(1, 2, 5, 6, 7, 8, 9, 10, 11)$:
$(1, 2, 5, 6, 7, 8, 9, 10, 13)$:
$(1, 2, 5, 6, 7, 8, 9, 11, 12)$:
$(1, 2, 5, 6, 7, 8, 9, 11, 13)$:
$(1, 2, 5, 6, 7, 8, 9, 13, 14)$:
$(1, 2, 5, 6, 7, 8, 10, 11, 12)$:
$(1, 2, 5, 6, 7, 8, 10, 11, 14)$:
$(1, 2, 5, 6, 7, 8, 10, 12, 14)$:
$(1, 2, 5, 6, 7, 8, 10, 13, 14)$:
$(1, 2, 6, 7, 8, 9, 10, 11, 12)$:

$d = 10$

$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 9, 11)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 9, 13)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 11, 14)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 12, 14)$:
$(1, 2, 3, 4, 5, 6, 7, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 7, 11, 12, 14)$:

$(1, 2, 3, 4, 5, 6, 8, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 8, 12, 13, 14)$:
$(1, 2, 3, 4, 5, 6, 9, 10, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 9, 10, 11, 14)$:
$(1, 2, 3, 4, 5, 6, 9, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 9, 11, 13, 14)$:
$(1, 2, 3, 4, 5, 6, 11, 12, 13, 14)$:
$(1, 2, 3, 4, 5, 10, 11, 12, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 10, 11)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 10, 12)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 11, 12)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 11, 13)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 8, 9, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 8, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 8, 11, 12, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 11, 12)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 11, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 12, 13)$:
$(1, 2, 3, 5, 6, 7, 9, 10, 12, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 11, 12, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 11, 13, 14)$:
$(1, 2, 3, 5, 6, 7, 9, 12, 13, 14)$:
$(1, 2, 3, 5, 6, 9, 10, 11, 12, 13)$:
$(1, 2, 3, 5, 6, 9, 10, 11, 12, 14)$:
$(1, 2, 3, 5, 6, 9, 10, 12, 13, 14)$:
$(1, 2, 3, 5, 7, 9, 10, 11, 13, 14)$:
$(1, 2, 5, 6, 7, 8, 9, 10, 11, 12)$:
$(1, 2, 5, 6, 7, 8, 9, 10, 11, 13)$:
$(1, 2, 5, 6, 7, 8, 9, 10, 13, 14)$:

$d = 11$

$(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 12)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 9, 11, 13)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 9, 13, 14)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 13)$:
$(1, 2, 3, 4, 5, 6, 7, 8, 11, 12, 14)$:

$(1,2,3,4,5,6,7,8,12,13,14)$:
$(1,2,3,4,5,6,7,11,12,13,14)$:
$(1,2,3,4,5,6,9,10,11,12,13)$:
$(1,2,3,4,5,6,9,11,12,13,14)$:
$(1,2,3,5,6,7,8,9,10,11,12)$:
$(1,2,3,5,6,7,8,9,10,12,13)$:
$(1,2,3,5,6,7,8,9,11,12,13)$:
$(1,2,3,5,6,7,8,9,11,13,14)$:
$(1,2,3,5,6,7,8,9,12,13,14)$:
$(1,2,3,5,6,7,9,10,11,12,14)$:

$d = 12$

$(1,2,3,4,5,6,7,8,9,10,11,12)$:
$(1,2,3,4,5,6,7,8,9,11,12,13)$:
$(1,2,3,4,5,6,7,8,9,11,13,14)$:
$(1,2,3,4,5,6,7,8,11,12,13,14)$:
$(1,2,3,4,5,6,9,10,11,12,13,14)$:
$(1,2,3,5,6,7,8,9,10,11,12,13)$:
$(1,2,3,5,6,7,8,9,10,12,13,14)$:

$d = 13$

$(1,2,3,4,5,6,7,8,9,10,11,12,13)$:
$(1,2,3,4,5,6,7,8,9,11,12,13,14)$:

$d = 14$

$(1,2,3,4,5,6,7,8,9,10,11,12,13,14)$: $\sim \Delta_4$

4.5.2 G$^{\pm}$VSSS-$(G,4)$ の分類

前節で構成した 68×68 正方行列（図 4.7）を隣接行列とするグラフを $\Delta_4^{\pm}$ とする．このとき頂点数は $68 (= 4^2+6^2+4^2)$ である．一般の m においては，Δ_m は ${}_mC_1{}^2 + {}_mC_2{}^2 + \ldots {}_mC_{m-1}{}^2$ の頂点数を持つ．G$^{\pm}$VSSS-$(G,4)$ の存在性に関しては以下が成立する．

定理 4.37 $^{\exists}$ *independent G$^{\pm}$VSSS-*$(G,4) \Leftrightarrow G \in Ind(\Delta_4^{\pm})$.

$\Delta_4^{\pm}$ に含まれる頂点数 3 以上の independent な誘導部分グラフを列挙しておく．これらは，復元画像として白黒反転画像を許容する G$^{\pm}$VSSS を構成可能なグラフの集合である

$d = 3$

Δ_4 の頂点インデックス $(1, 2, 3)$: $\sim K_3$

$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

注意 4.38 K_3 は $m^* = 3$ のリストにも存在するため，ここでは新たなグラフは登場しない.

$d = 4$

$(1, 2, 3, 4)$: $\sim K_4$

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$(1, 2, 3, 5)$:

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$(1, 2, 5, 7)$: $\sim P_4$

$$\begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

注意 4.39 *GVSSS*-$(G, 4)$ のリストと同様であり，$G^{\pm}$*VSSS* の概念を導入する効果は見られない.

$d = 5$

$(1, 2, 3, 4, 5)$:

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,1,0 \\ 1,1,0,1,0 \\ 1,1,1,0,0 \\ 1,0,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 4, 17)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,1,0 \\ 1,1,0,1,1 \\ 1,1,1,0,1 \\ 0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,1 \\ 1,1,0,0,0 \\ 1,0,0,0,1 \\ 0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,0 \\ 1,1,0,0,0 \\ 1,0,0,0,1 \\ 0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 9)$:

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,0,0 \\ 1,1,0,0,0 \\ 1,0,0,0,1 \\ 1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 10)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,1 \\ 1,1,0,0,0 \\ 1,0,0,0,0 \\ 0,1,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 19)$:

$$\begin{bmatrix} 0,1,1,1,0 \\ 1,0,1,0,1 \\ 1,1,0,0,1 \\ 1,0,0,0,0 \\ 0,1,1,0,0 \end{bmatrix}$$

$(1,2,3,5,21)$:

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,0,0 \\ 1,1,0,0,1 \\ 1,0,0,0,1 \\ 1,0,1,1,0 \end{bmatrix}$$

$(1,2,5,7,10)$: $\sim P_5$

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,0,0,1 \\ 1,0,0,1,0 \\ 0,0,1,0,0 \\ 0,1,0,0,0 \end{bmatrix}$$

$(1,2,5,7,19)$: $\sim C_5$

$$\begin{bmatrix} 0,1,1,0,0 \\ 1,0,0,0,1 \\ 1,0,0,1,0 \\ 0,0,1,0,1 \\ 0,1,0,1,0 \end{bmatrix}$$

$(1,2,22,28,34)$: $\sim K_5$

$$\begin{bmatrix} 0,1,1,1,1 \\ 1,0,1,1,1 \\ 1,1,0,1,1 \\ 1,1,1,0,1 \\ 1,1,1,1,0 \end{bmatrix}$$

注意 4.40 *GVSSS-*$(G,4)$ においては *10* 種類のグラフで構成可能であり，本リストにおいては *G*$^{\pm}$*VSSS-*$(C_5,4)$ のみが初出となる．

例 4.41 (G$^{\pm}$VSSS-$(C_5,4)$)

$$S_0 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},\ S_1 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}.$$

$$R(S_0) = \begin{bmatrix} 1 & 1 & 2 & 2 & 2 \\ 1 & 1 & 2 & 2 & 2 \\ 2 & 2 & 1 & 1 & 2 \\ 2 & 2 & 1 & 1 & 2 \\ 2 & 2 & 2 & 2 & 2 \end{bmatrix}.$$

$$R(S_1) = \begin{bmatrix} 1 & 2 & 2 & 2 & 3 \\ 2 & 1 & 1 & 2 & 2 \\ 2 & 1 & 1 & 2 & 2 \\ 2 & 2 & 2 & 1 & 3 \\ 3 & 2 & 2 & 3 & 2 \end{bmatrix}.$$

よって

$$R(S_1) - R(S_0) = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix} \text{より } norm(\ R(S_1) - R(S_0)\ ) = Adj(C_5)$$

を満たすことが確認できる.

$d = 6$

$(1, 2, 3, 4, 5, 6)$:

$$\begin{bmatrix} 0,1,1,1,1,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,0,0 \\ 1,1,1,0,0,0 \\ 1,0,0,0,0,1 \\ 0,1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 9)$:

$$\begin{bmatrix} 0,1,1,1,1,1 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,0 \\ 1,1,1,0,0,0 \\ 1,0,0,0,0,1 \\ 1,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 10)$:

$$\begin{bmatrix} 0,1,1,1,1,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,0,0 \\ 1,1,1,0,0,0 \\ 1,0,0,0,0,0 \\ 0,1,0,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 17)$:

$$\begin{bmatrix} 0,1,1,1,1,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,1 \\ 1,1,1,0,0,1 \\ 1,0,0,0,0,0 \\ 0,0,1,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 20)$:

$$\begin{bmatrix} 0,1,1,1,1,1 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,0 \\ 1,1,1,0,0,1 \\ 1,0,0,0,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 23)$:

$$\begin{bmatrix} 0,1,1,1,1,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,1 \\ 1,1,1,0,0,1 \\ 1,0,0,0,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 5, 26)$:

$$\begin{bmatrix} 0,1,1,1,1,1 \\ 1,0,1,1,0,0 \\ 1,1,0,1,0,0 \\ 1,1,1,0,0,1 \\ 1,0,0,0,0,0 \\ 1,0,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 4, 17, 18)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,1 \\ 0,0,1,1,0,1 \\ 0,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 17, 22)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,0 \\ 0,0,1,1,0,1 \\ 1,1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 17, 23)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,0 \\ 1,1,0,1,1,1 \\ 1,1,1,0,1,1 \\ 0,0,1,1,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 4, 17, 24)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,1,0,1 \\ 1,1,0,1,1,0 \\ 1,1,1,0,1,1 \\ 0,0,1,1,0,0 \\ 0,1,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 7)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 8)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 9)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,0 \\ 1,0,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 11)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,1,0,1,0,0 \\ 0,0,1,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 19)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,1 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,1,0,1,0,1 \\ 0,1,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 6, 22)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,1,1 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,1,0,1,0,1 \\ 1,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 9)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,0,0,1,0,0 \\ 1,0,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 10)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,0 \\ 0,1,0,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 12)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 17)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,1 \\ 0,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 19)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,0 \\ 0,1,1,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 20)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,1 \\ 0,0,0,1,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 21)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,0,0,1,0,0 \\ 1,0,1,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 23)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,0,0,1,0,0 \\ 0,0,1,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 25)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,0,0,1,0,1 \\ 0,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 26)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,0 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,0 \\ 1,0,0,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 8, 27)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,1,0 \\ 0,0,0,1,0,1 \\ 1,0,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 9, 21)$:

$$\begin{bmatrix} 0, 1, 1, 1, 1, 1 \\ 1, 0, 1, 0, 0, 0 \\ 1, 1, 0, 0, 0, 1 \\ 1, 0, 0, 0, 1, 1 \\ 1, 0, 0, 1, 0, 0 \\ 1, 0, 1, 1, 0, 0 \end{bmatrix}$$

$(1, 2, 3, 5, 9, 32)$:

$$\begin{bmatrix} 0, 1, 1, 1, 1, 1 \\ 1, 0, 1, 0, 0, 0 \\ 1, 1, 0, 0, 0, 0 \\ 1, 0, 0, 0, 1, 0 \\ 1, 0, 0, 1, 0, 0 \\ 1, 0, 0, 0, 0, 0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 15)$:

$$\begin{bmatrix} 0, 1, 1, 1, 0, 0 \\ 1, 0, 1, 0, 1, 0 \\ 1, 1, 0, 0, 0, 1 \\ 1, 0, 0, 0, 0, 0 \\ 0, 1, 0, 0, 0, 0 \\ 0, 0, 1, 0, 0, 0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 19)$:

$$\begin{bmatrix} 0, 1, 1, 1, 0, 0 \\ 1, 0, 1, 0, 1, 1 \\ 1, 1, 0, 0, 0, 1 \\ 1, 0, 0, 0, 0, 0 \\ 0, 1, 0, 0, 0, 0 \\ 0, 1, 1, 0, 0, 0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 21)$:

$$\begin{bmatrix} 0, 1, 1, 1, 0, 1 \\ 1, 0, 1, 0, 1, 0 \\ 1, 1, 0, 0, 0, 1 \\ 1, 0, 0, 0, 0, 1 \\ 0, 1, 0, 0, 0, 1 \\ 1, 0, 1, 1, 1, 0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 22)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,1,1 \\ 1,1,0,0,0,0 \\ 1,0,0,0,0,1 \\ 0,1,0,0,0,0 \\ 1,1,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 29)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,0 \\ 1,1,0,0,0,1 \\ 1,0,0,0,0,1 \\ 0,1,0,0,0,1 \\ 0,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 10, 30)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,1,1 \\ 1,1,0,0,0,0 \\ 1,0,0,0,0,1 \\ 0,1,0,0,0,0 \\ 0,1,0,1,0,0 \end{bmatrix}$$

$(1, 2, 3, 5, 12, 19)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,1 \\ 1,0,0,0,0,0 \\ 0,0,0,0,0,1 \\ 0,1,1,0,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 12, 31)$:

$$\begin{bmatrix} 0,1,1,1,0,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,1 \\ 1,0,0,0,0,1 \\ 0,0,0,0,0,1 \\ 0,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 17, 21)$:

$$\begin{bmatrix} 0,1,1,1,0,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,1,1 \\ 1,0,0,0,0,1 \\ 0,0,1,0,0,1 \\ 1,0,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 21, 24)$:

$$\begin{bmatrix} 0,1,1,1,1,0 \\ 1,0,1,0,0,1 \\ 1,1,0,0,1,0 \\ 1,0,0,0,1,1 \\ 1,0,1,1,0,1 \\ 0,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 5, 21, 26)$:

$$\begin{bmatrix} 0,1,1,1,1,1 \\ 1,0,1,0,0,0 \\ 1,1,0,0,1,0 \\ 1,0,0,0,1,0 \\ 1,0,1,1,0,0 \\ 1,0,0,0,0,0 \end{bmatrix}$$

$(1, 2, 3, 8, 25, 31)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,0,1,1 \\ 1,1,0,0,1,1 \\ 0,0,0,0,1,1 \\ 0,1,1,1,0,1 \\ 0,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 3, 19, 25, 31)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,1,1,1,1 \\ 1,1,0,1,1,1 \\ 0,1,1,0,1,1 \\ 0,1,1,1,0,1 \\ 0,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 7, 10, 11)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,1,0,0,0,1 \\ 0,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 7, 10, 12)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,0 \\ 0,1,0,0,0,1 \\ 0,0,0,0,1,0 \end{bmatrix}$$

$(1, 2, 5, 7, 10, 19)$:

$$\begin{bmatrix} 0,1,1,0,0,0 \\ 1,0,0,0,1,1 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,1,0,0,0,0 \\ 0,1,0,1,0,0 \end{bmatrix}$$

$(1, 2, 5, 7, 10, 27)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,0 \\ 0,1,0,0,0,0 \\ 1,0,0,0,0,0 \end{bmatrix}$$

$(1, 2, 5, 7, 10, 32)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,0,0,1,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,1,0,0,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 5, 7, 12, 26)$:

$$\begin{bmatrix} 0,1,1,0,0,1 \\ 1,0,0,0,0,0 \\ 1,0,0,1,0,0 \\ 0,0,1,0,0,1 \\ 0,0,0,0,0,1 \\ 1,0,0,1,1,0 \end{bmatrix}$$

$(1, 2, 7, 8, 17, 22)$:

$$\begin{bmatrix} 0,1,0,0,0,1 \\ 1,0,0,0,0,1 \\ 0,0,0,1,1,0 \\ 0,0,1,0,1,0 \\ 0,0,1,1,0,1 \\ 1,1,0,0,1,0 \end{bmatrix}$$

$(1, 2, 7, 17, 19, 22)$:

$$\begin{bmatrix} 0,1,0,0,0,1 \\ 1,0,0,0,1,1 \\ 0,0,0,1,1,0 \\ 0,0,1,0,1,1 \\ 0,1,1,1,0,1 \\ 1,1,0,1,1,0 \end{bmatrix}$$

$(1, 2, 17, 18, 19, 22)$:

$$\begin{bmatrix} 0,1,0,0,0,1 \\ 1,0,0,1,1,1 \\ 0,0,0,1,1,1 \\ 0,1,1,0,1,1 \\ 0,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 2, 17, 22, 28, 34)$:

$$\begin{bmatrix} 0,1,0,1,1,1 \\ 1,0,0,1,1,1 \\ 0,0,0,1,1,1 \\ 1,1,1,0,1,1 \\ 1,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 6, 18, 21, 26, 27)$:

$$\begin{bmatrix} 0,0,0,1,1,1 \\ 0,0,1,0,1,1 \\ 0,1,0,1,0,1 \\ 1,0,1,0,0,1 \\ 1,1,0,0,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 6, 18, 21, 26, 32)$:

$$\begin{bmatrix} 0,0,0,1,1,1 \\ 0,0,1,0,1,1 \\ 0,1,0,1,0,0 \\ 1,0,1,0,0,0 \\ 1,1,0,0,0,1 \\ 1,1,0,0,1,0 \end{bmatrix}$$

$(1, 6, 18, 21, 27, 33)$:

$$\begin{bmatrix} 0,0,0,1,1,1 \\ 0,0,1,0,1,1 \\ 0,1,0,1,1,1 \\ 1,0,1,0,1,1 \\ 1,1,1,1,0,1 \\ 1,1,1,1,1,0 \end{bmatrix}$$

$(1, 17, 18, 21, 22, 26)$:

$$\begin{bmatrix} 0,0,0,1,1,1 \\ 0,0,1,1,1,0 \\ 0,1,0,1,1,0 \\ 1,1,1,0,1,0 \\ 1,1,1,1,0,0 \\ 1,0,0,0,0,0 \end{bmatrix}$$

注意 4.42 *GVSSS-*$(G, 4)$ においては *32* 種類のグラフで構成可能であり，本リストは *57* 種類あることから *25* 種類のグラフが初出となる．

$d \geq 7$

計算結果は省略する．$d = 7$ では 378 種類のグラフで構成可能であることが分かっている．$d \geq 8$ においては GAP [67] や計算機のリソース制限により枝切りできず，計算結果が得られないケースが見られた．

optimal GVSSS-$(G,4)$，G$^{\pm}$VSSS-$(G,4)$ を満たすグラフの総数

optimal な GVSSS-$(G,4)$ および G$^{\pm}$VSSS-$(G,4)$ を満たすグラフの総数は以下のとおりである．

表 4.3: optimal GVSSS-$(G,4)$，G$^{\pm}$VSSS-$(G,4)$ を満たすグラフの総数

d =	3	4	5	6	7	8	9	10	11	12	13	14	15	-	31	32	33
GVSSS-$(G,4)$	1	3	10	32	56	75	58	39	16	7	2	1	0		0	0	0
G$^{\pm}$VSSS-$(G,4)$	1	3	11	57	378	-	-	-	-	-	-	-	-		26	7	2

4.6 G$^{\pm}$VSSS における複数の画像の埋め込み multi-G$^{\pm}$VSSS

前節までで G$^{\pm}$VSSS の構成法と小さい画像拡大率に対するグラフ分類の事例を見てきた．比較的小さい画像拡大率（$m^* = 2, 3, 4$）で G$^{\pm}$VSSS を構成可能なグラフのリストがあることから，これらを組み合わせることで複数の画像の埋め込みを可能とする multi-G$^{\pm}$VSSS を実現することができる．

4.6.1 アクセス構造を表現する行列

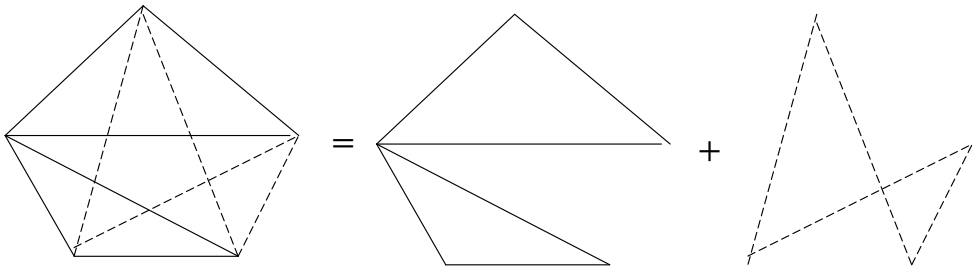

図 4.8: K_5 のグラフ分割例

4.2.1 節で述べたように，従来型 GVSSS から multi-G$_d$VSSS を構成する単純な方法がある．異なる VSSS に対する生成行列を連結する方式である．これは特に従来型 GVSSS に限った構成方法ではなく，同様の方法で multi-G$^{\pm}$VSSS を構成可能である．

ここでは例えば 5 つのシェアを配布して，2 つのシェアの組み合わせに応じて復元される画像が異なるように構成することを考える．例として以下のようなアクセス行列を考える．

例 4.43

$$\begin{bmatrix} 0,a,a,a,a \\ a,0,a,b,b \\ a,a,0,b,b \\ a,b,b,0,a \\ a,b,b,a,0 \end{bmatrix}$$

ここで行列成分の *"a"* および *"b"* は，*2* つのシェアを重ね合わせたときにどちらの画像を復元するかどうかを指し示したものである．図 *4.8* はこの行列をグラフ表現したものであり，完全グラフ K_5 が *2* つのグラフ G_a, G_b に分割されていることが分かる．

それぞれのグラフは

- $m^* = 3$ の *G*$^{\pm}$*GVSSS* リストに含まれるため *G*$^{\pm}$*GVSSS-*$(G_a, 3)$
- 定理 *4.11* より *G*$^{\pm}$*GVSSS-*$(G_b, 2)$

で構成可能なことが分かる．これらの生成行列を単純に連結することでトータルで画像拡大率 *5* で *multi-G*$^{\pm}$*VSSS* を構成できる．

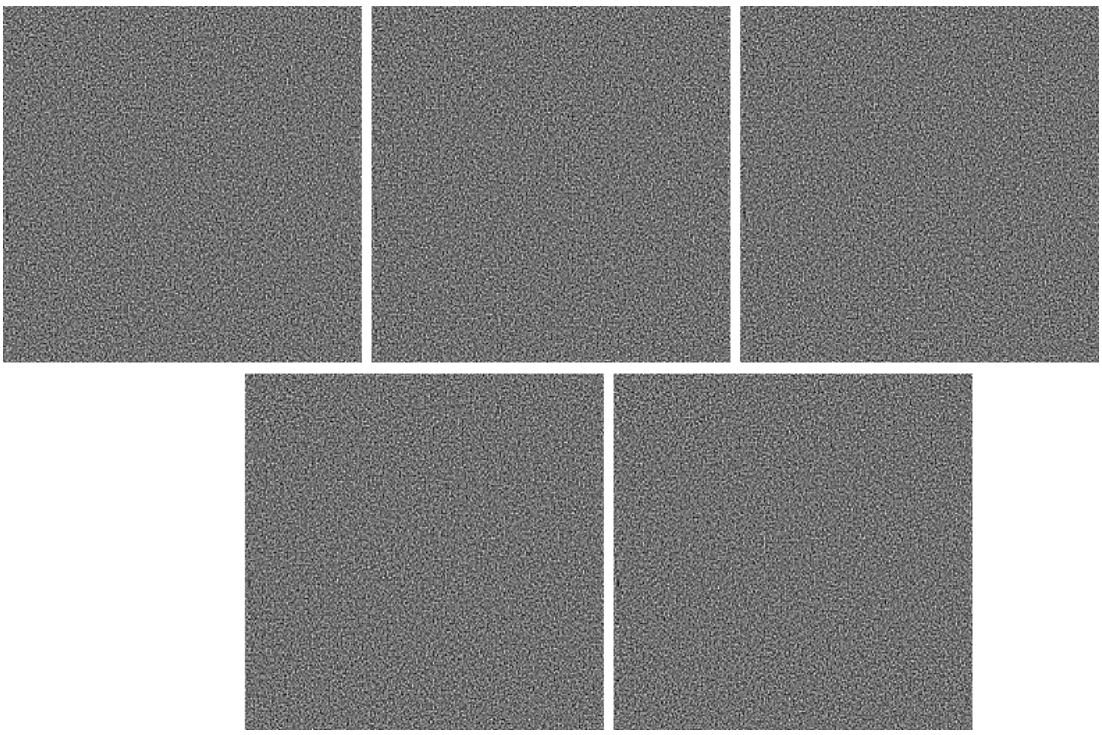

図 4.9: K_5 の頂点に紐付けされる 5 つのシェア画像

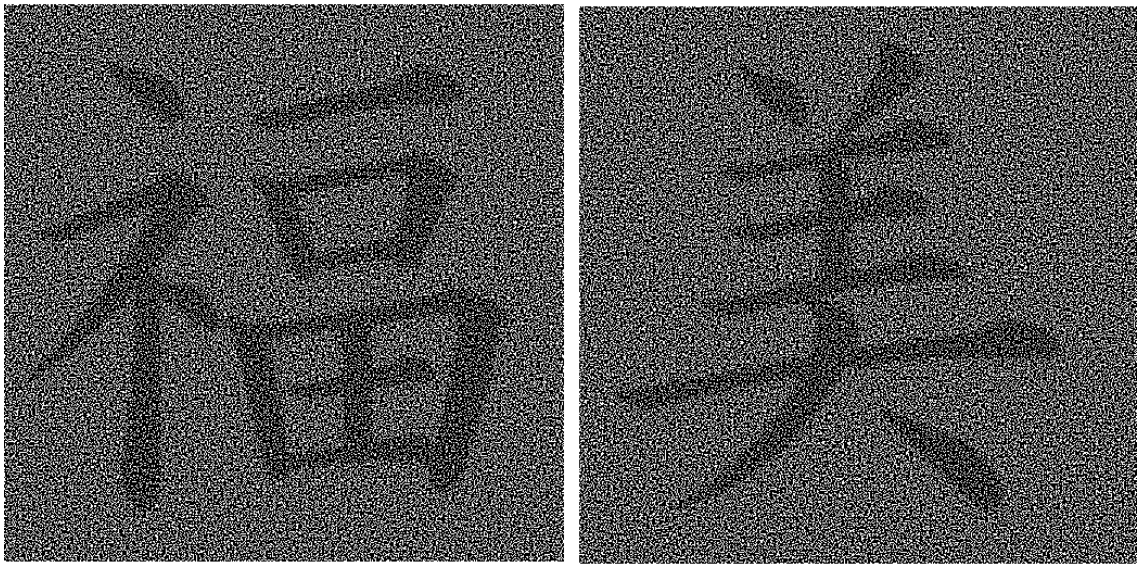

図 4.10: 2 パターンの復元画像

図 4.9，図 4.10 の例においては画像拡大率を 5 から $9(=3\times3)$ に拡大して画像を表現している．

第5章　XOR-SSSとVSSSを同時に利用する形態について

ここまで議論してきたXOR-SSSとVSSSについてCIAに準えられる3つの要件（機密性，完全性，可用性）をバランスよく，かつトータルでカバーできるユースケースについて議論していく．本来XOR-SSS等の秘密分散方式は，漏洩リスクの分散と紛失リスクの分散というメリットを持つことから，機密性と可用性をカバーする技術である．一方でVSSSもXOR-SSS同様の機能と持つが，この技術をCIAのうちの「完全性」でカバーすることを検討する．これにより，根本的には同じ秘密分散というひとつの技術でありながら，トータルですべての要件をカバーすることができる．

具体的には，クラウド事業者が顧客からのデータをアーカイブし，顧客の要求に応じてデータを処理するユースケース，特に更新や削除の頻度が比較的高いデータアーカイブとしての利用を想定する．これはすでに議論してきたようにクラウド環境においてはニーズが高く，かつ実用的な利用用途である．

5.1　データの取り扱いに関する考察

実際に秘密分散方式でデータを分散処理を行うユースケースにおいて考慮すべきいくつかの要件を列挙する．

5.1.1　格納データの暗号化

格納データを暗号化する場合の実際に暗号化を行う主体を考える．このとき，アーカイブサービスの利用者による暗号化と事業者が独自に暗号化する場合の2種類に分類できる．前者は事業者に情報を漏らさないため，後者はユーザ・事業者とは異なる第3者に（サービスへの攻撃などにより）情報を漏らさないための暗号化処理である．

ともに暗号化時に用いた鍵管理が重要であり，復元できなくなることへの対処が必要である．鍵データのバックアップを他のアーカイブサービスを利用したり，相互に預けるなどして冗長化を図る必要がある．一方で鍵データを外部に預けることが不安であるならば，他の対策を講じる必要が出てくる．

5.1.2 格納データの分散化

バックアップ目的でのデータ分散化だけを想定するのではなく，常に更新されるデータの分散化も考慮すべきである．このときデータの分散化を行う主体を考えると，やはりサービス利用者主導型と事業者主導型の 2 つのタイプに分けることができる．前者はユーザの管理ドメインにおいて分散処理を行った後に，複数のアーカイブサービスに格納する場合が想定される．一方で後者は，管理するクラウドのキャパシティが超えた場合など，一時的にデータを他の事業者に退避させるようなケースなどが考えられる．

いずれの場合も更新頻度の高いデータ扱う場合には，レプリケーションだけでも管理が難しいと考えられる．そのため秘密分散処理を行いながら運用するためには高速な分散・復元方式が望まれる．

5.1.3 格納データの更新

すでに複数のアーカイブサービスに格納しているケースにおいてデータ更新の機能について考察する．XOR-SSS を利用して n 箇所のアーカイブサービスにデータを保管している場合，1 箇所のデータを書き換えることで，全体のデータ更新を行うことが可能である．しかし，XOR で足しこむデータを鑑みると，更新する該当箇所とその差分が事業者に漏れてしまう問題がある．よって 2 箇所以上のデータを同時に書き換えることが望ましい．ただし n 箇所の全てのデータを同時に操作する必要はなく，最低 2 箇所のデータ更新で十分である．

5.1.4 格納データの削除

データ暗号化を行っていることから機密性を保っているとも考えられるが，可用性の観点から見るとデータの冗長化を行っており，削除が正しく処理されていない場合には，復元されてしまう恐れがある．これは秘密分散方式のメリットでもある「一部紛失しても復元可能」であることが要因となっている．サービスが停止される場合にも利用者のリクエストが正しく処理されるべきであり，事業継続性のある事業者の選択が望まれる．

5.1.5 格納データへのアクセス

格納データが漏れることを前提にして暗号化・分散化を行ってはいるが，インターネットを経由すれば誰でも当該データにアクセスできるような設計は避けるべきである．例えばアクセストークンに似たケースとして URL さえあれば認証無しでアクセスできるサービスも見受けられる．URL 自体に認証情報を含んだり，URL を長くすることで簡単に生成ルールが分からないように設計されているが，電子メールの傍受などで URL が漏れてしまうケースを考えると決して望ましい方式ではない．ワンタイム URL や，比較的短い時間のみアクセス可能な仕組みによる保護も考えられるが，根本的な対策として，何らかの認証および認可の仕組みを導入すべきである．

5.2　データ流通のバリエーション

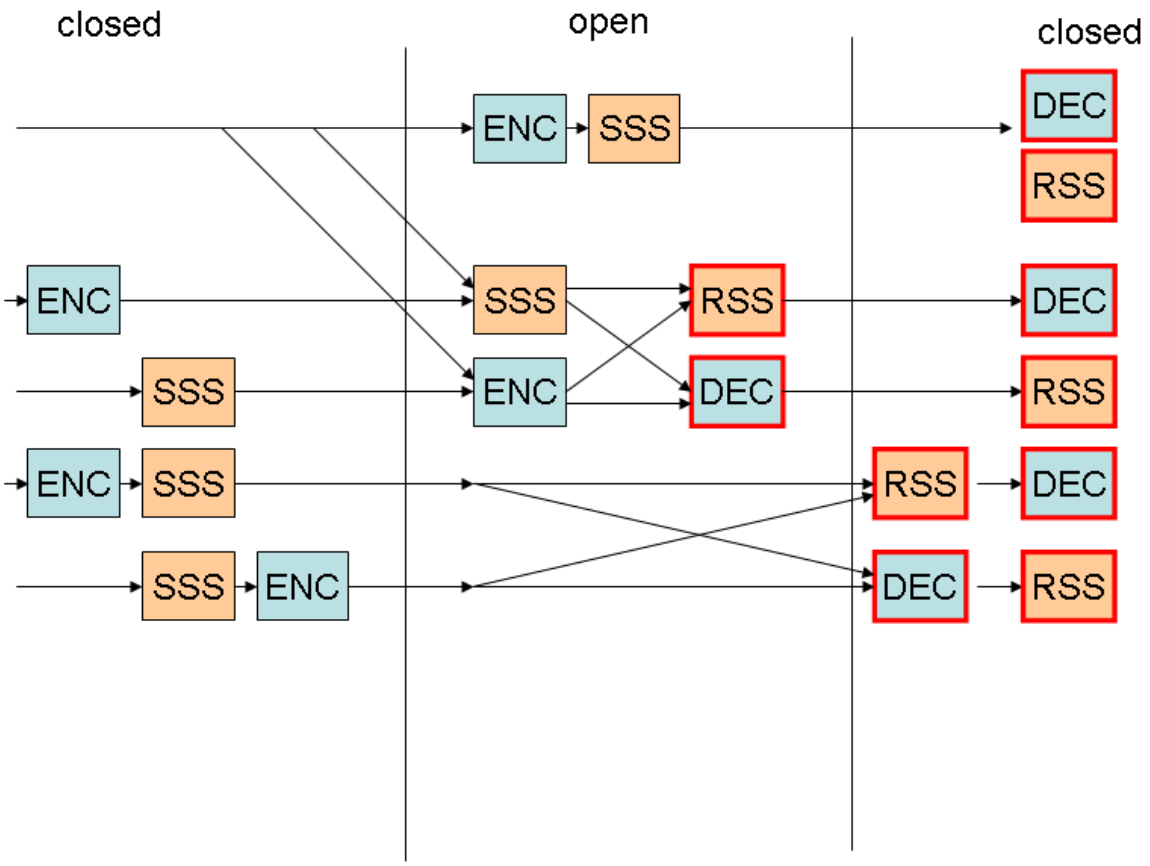

図 5.1: データ流通のバリエーション

上記の考察を踏まえると図 5.1 のようなデータ流通のバリエーションが考えられる．ENC は暗号化，DEC は復号処理，SSS は秘密分散，RSS は復元処理を表現している．また 3 パートのうち両サイドはクローズな環境で処理を行い，中央のエリアはオープンな環境で処理を行うことを意味している．これらのフローのバリエーションを鑑みると，本論文で提案したようにデータ暗号化と秘密分散が可換であることが望まれていることが分かる．

また，前節の考察にあるように，ある程度の処理速度が要求されていることから XOR-SSS でデータ暗号化を行うことが望ましい．さらに アクセス制御においては，VSSS が従来の ID・パスワード方式の置き換えや併用を推進すると考えられる．スマートフォンなどにおいてワンタイムパスワードアプリが推奨されているが，ネットワークに接続可能になった時点で，そのデバイス自身がハッキングされてしまい秘密情報が漏れてしまうという問題が生じてしまう．つまり，インターネットなどの広域網に繋がらないで安全に管理できる物理的媒体で秘密情報を分散しておくニーズがあるとも言える．

5.2.1　オンライン利用

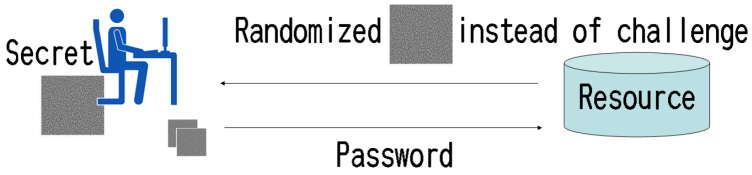

図 5.2: VSSS を認証に用いる基本アイデア

XOR-SSS でデータ暗号化をカバーする一方で，VSSS で認証を行う基本アイデアが図 5.2 である．まず，ユーザがあるリソースにアクセスするために自身の識別子 (identifier) を伝えてログインを試みる．サーバは，識別子情報からデータベースを参照することでユーザが持つ VSSS のシェア画像を得て，チャレンジ画像をその都度生成してユーザに返却する．このとき VSSS の仕組みを利用してシェアを重ね合わせることでパスワードを復元できるようなチャレンジ画像を生成することができる．この方式により，その都度ワンタイムパスワードをサーバとユーザが共有することで「VSSS ログイン」を可能としている．

例えばオンラインバンキング等のケースにおいて，トランザクションを依頼するようなケースを考える．このときインターネット網でバンキングサイトを閲覧し，シェア画像を獲得する．これを手元に物理的に存在するシェア画像と重ね合わせることでワンタイムパスワードを得て，携帯電話網などの別のチャネルでログイン（ユーザ認証）を行う．複数チャネルを用いた認証を行うため，既存方式に比べより安全な方式である．

現在も通常のパスワードベースのログインに加えて，残高や履歴閲覧などでは必要ないが，振込などの重要なトランザクションが発生した際に乱数表などを用いて第 2 段階のログインを必要とするケースがある．これと同様に，ログイン時にはこれまでの方法を用いるが，重要なトランザクションが発生した場合など，認可のために本方式を追加的に利用するケースが考えられる．

さらにこの仕組みを利用することで，手元にあるシェア画像を定期的に更新することも可能となる．シェア画像をサーバ側で印刷して物理的に郵送する方式もあるが，シェア画像をインターネット経由で配信して OHP シートなどにユーザが印刷してシェア画像を更新するケースも考えられる．

5.2.2　オフライン利用

前節のように認証・認可を行うシチュエーションにおいてワンタイムチケットが物理的に郵送されるケースを想定する．このとき multi-GVSSS を用いることにより 1 枚のチケットで複数人に対して異なる機密データを提示することが可能となる．具体的にはリソースにアクセスするためのパスワードの更新などの用途が考えられる．

さらに，手元で使っていた VSSS のシェアを更新する際にも利用できる．シェアそのものを郵送して受領する場合には，盗難される危険性が存在する．一方で，旧シェアを用いて秘密

情報（パスワード）を復元し，そのパスワードを用いて新シェアをサーバから SSL/TLS などの安全な通信路を用いてダウンロードする場合には，プリンタや当該ユーザが利用するローカルネットワーク環境が安全であるという条件のもとで，安全にシェア更新が可能となる．

5.2.3　秘密分散技術の普及を目指して

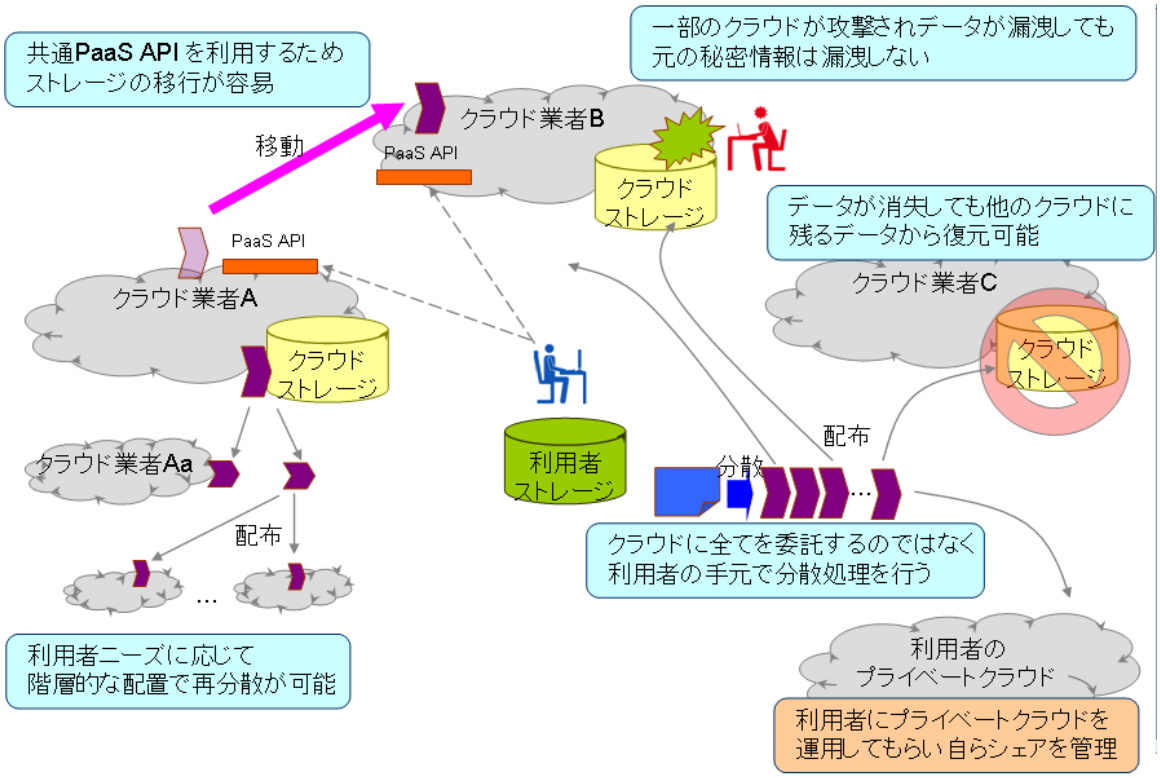

図 5.3: 今後のクラウド環境における秘密分散技術の利用

図 5.3 は今後想定されるクラウド環境における秘密分散技術の利用形態の予測を示している．まず，秘密分散技術における API やフォーマットなどの標準化が推進されることで利用者が増大し，利用コストが削減されると考えられる．特に，分散後データのフォーマットが標準化されれば，異なる事業者への移行が容易となり，業界全体が活性化される．また，各業者で共通の API が利用可能になることで利用のための障壁が削減されるため潜在的なユーザを発掘し，秘密分散技術の利用が促されることになる．

本論文で扱った技術は，これらの利用形態にもマッチしており，今後さらなる利用の促進が進み，利用者のメリットが増えることを切望する．

第6章　まとめと今後の課題

まず初めに一般的なセキュリティ要件の考え方を論じる際に用いる CIA(Confidentiality, Integrity and Availability) モデルを用いて，クラウド環境におけるセキュリティ要件を洗い出した．どのような技術が利用検討・実施されているのかについて分析した結果，本論文では秘匿性と可用性の要件を満たす秘密分散方式に着目した．また，秘密分散方式の 1 種である視覚復号型と呼ばれる方式をエンティティの認証・認可として併用することで CIA に準えられる 3 つの要件をトータルでカバーできる秘密分散方式を提案した．

データに対する秘密分散方式として，分散・復元時に排他的論理和だけを用いて構成する理想的な秘密分散法 (XOR-SSS) について新しい構成方式について提案した．XOR-SSS は排他的論理和だけを用いるため従来の Shamir による構成法に比べてはるかに高速に分散・復元が可能であるという優位性を持つ．またオリジナルとシェアのサイズが不変な秘密分散法でもあり，ストレージを有効に使う意味でもクラウドへの適合性が高い．

ここで，データをアーカイブする際にクラウド事業者は，秘匿性と可用性を高めるために，１）データ暗号化，２）秘密分散法の両方を併用することが考えられる．さらに，クラウド事業者は不測のデータクラッシュなどの障害に対処するなど可用性を高めるために，顧客から預かったデータを主体の異なる他の事業者にコピーもしくはデータの一部を暗号化・秘密分散処理を行った上で再配布を行うというケースも検討した．このとき，秘密分散処理とデータ暗号化がお互いに可換な処理であるとすると，どこに・どのように暗号化済み分散データが流通していたとしても，復元が容易になることが分かる．準同型性を有する関数を用いることで実現可能であるが，今回は排他的論理和 (XOR) 演算の可換性を用いた方法について検討を行った．つまり，暗号化としてストリーム暗号や，ブロック暗号の CTR モードの利用など，鍵データを平文に足しこむ形の演算で暗号化を行っているケースにおいては「秘密分散処理とデータ暗号化の可換性」を保つことができることを利用している．

さらに，クラウド上で暗号化したまま演算を行う秘密計算・委託計算への拡張についても試みている．クラウドをデータのアーカイブ先として利用する静的な利用に対して，クラウドに計算を委託し，暗号化状態のまま計算して結果を得るという動的な利用方法のひとつである．入力データを秘匿して秘密裏に演算を行って出力を行う秘密計算法は，クラウドのマシンパワーも向上していることから現実的な計算量で解決できるようになった．特に，前述の Fully Homomorphic Encryption よりも軽量で，制約はあるものの複数のエンティティ（例えばクライド，サーバ，場合によりユーザ自身の PC）間の協調計算により秘密計算を行うマルチパーティプロトコルが注目されている．これは，近年多発する漏洩事件・事故に起因して，プライバシドリブンの考え方が定着しつつあり，ユーザニーズの高い分野でもある．具

体的には，医療データやゲノムデータなど，よりセンシティブなデータに対しても正しくかつ安全に活用できる実験も公表され続けている．

次に，認証・認可に用いるトークンの提示フェーズにおいて，復号に計算リソースを利用しない視覚復号型秘密分散法 (VSSS) を適用することを検討した．VSSS は，機密画像を複数のシェア画像にあらかじめ分散し OHP シートのような透過性を持ち物理的に重ね合わせが可能な媒体に印刷して，シェア画像を重ね合わせることで目の錯覚を用いて機密画像を復元する方法である．白黒画像を入力として，各ピクセルごとにどのようにシェア画像を生成するか定めた 2 つの生成行列 S_0, S_1（S_0 は白画素，S_1 は黒画素）を用いて機密画像からシェア画像を作成することができる．このとき，秘密画像の 1 ピクセルがシェア画像の何ピクセルに拡大されるかを示す画像拡大率と相対差（白か黒かの判別しやすさに関するパラメータ）は生成行列の構成方法に大きく依存する．一般的には，画像拡大率を小さくすることで復元時のコントラストをよりよくする効果があるため，できるだけ画像拡大率を抑える必要がある．

従来例としてよく知られる (k, n)-しきい値視覚復号型秘密分散法は, 復元するための権限としては平等に分散情報が分配されるため, さまざまなアクセス構造が想定される実利用においてはうまく適用できないことが多い．そこで，本論文ではグラフで表現されたアクセス構造を持つ視覚復号型秘密分散法について追求した．既存のグラフタイプ視覚復号型秘密分散法（GVSSS）はグラフの頂点集合をシェア画像の集合と対応づけ，２頂点間に辺が存在する場合には（２頂点に対応する２つのシェア画像から）機密画像が復元され，辺が無い場合には復元されないというアクセス構造を持つ．(2,n)-VSSS は，グラフとして完全グラフを利用した場合に相当するため GVSSS は (2,n)-VSSS の拡張と考えることができる．本論文では GVSSS の概念を拡張して様々な亜種を提案している．

G_dVSSS は，辺の有無（つまり距離 1 かどうか）ではなく 2 頂点間の距離に基づいたアクセス構造を持つ VSSS である．また，既存の GVSSS では 1 つの機密画像のみがシェア画像に埋め込まれていたが，本論文で提案するスキームでは２枚のシェア画像を重ねあわせたときに，シェア間の距離に応じて復元される機密画像が異なるような multi-G_dVSSS を構成することも実現している．このように複数の機密画像を埋め込むことができるため，既存の GVSSS に比べ 利用用途が拡がると考えられる．

また，既存の GVSSS に対して，復元画像のコントラストの改善を検討し, より見やすい構成方法についても新たに提案している．これまでの構成方法であるスターグラフ分割法は，与えられたグラフ G をスターグラフ，つまり 1 頂点（中心）からしか辺が存在しないグラフに分割する方式である．スターグラフからは画像拡大率 2 の生成行列を持つ GVSSS が構成できることが知られており，この画像拡大率 2 の生成行列を利用して，分割されたスターグラフの生成行列を連結する（各分割スターグラフの生成行列を横に並べる）ことでグラフ G に対する生成行列を構成することができる．しかし，グラフ間のエッジの数が多い場合や頂点数が多い場合には，画像拡大率は肥大化する傾向にあり，見やすさの点で問題が生じる．また，スターグラフ分割法は必ずしも相対差が最大となる構成方法ではない．そこで本論文では，スターグラフ分割法を改善する，完全 n 部グラフ分割法，辺消去法を提案し，画像拡大率を下げている．

さらに，与えられたグラフに対し，それ以上画像拡大率を下げられない optimal な構成方法を導出する方法を検討した．グラフ G とその生成行列（つまり GVSSS-(G, m) の構成方法）に対して optimal かどうかを判定することは非常に難しいため，画像拡大率を固定した上で，どのようなグラフが optimal かどうかを分類するというアプローチを取り，画像拡大率 4 までの GVSSS の分類を行っている．

さらに，画像拡大率を抑えるために，復元画像として白黒反転画像を許容する視覚復号型秘密分散法 G$^{\pm}$VSSS の提案も行っている．その結果，例えば，頂点数 9 の Lattice graph $L_2(3)$ においては，スターグラフ分割法では画像拡大率が 14 必要であったが，本論文では画像拡大率を 3 にまで抑える生成行列の構成事例など大きな改善を行うことができた．

最後に，クラウド事業者が顧客からのデータをアーカイブし，顧客の要求に応じてデータを処理するユースケースにおいて XOR-SSS と GVSSS を用いた実際の構成例について検討した．秘匿性と可用性の要件を満たす秘密分散方式において，格納データのライフサイクルに着目し，秘密分散方式の 1 種である視覚復号型と呼ばれる方式をエンティティの認証・認可として併用することで，CIA に準えられる 3 つの要件をトータルでカバーできることを確認し，本論文で扱う提案方式の有用性を示した．

謝辞

本論文を執筆するにあたり，熱心にご指導頂きました 筑波大学大学院システム情報工学研究科リスク工学専攻岡本栄司教授に深く感謝いたします.

古賀弘樹教授，片岸一起准教授，西出隆志准教授，金山直樹助教には大変お忙しい中，論文の審査をお引き受け頂きました．本研究に関連して東京大学 國廣昇准教授，電気通信大学岩本貢准教授には多くの議論を交えて的確なアドバイスを頂いたことで，論文をよりよいものにすることができました．株式会社インターネットイニシアティブ山井美和様，齋藤衛様には社会人学生への通学をご許可頂きました．会社の同僚として，また，大学での先輩としてご支援いただいた加藤雅彦氏には，入学から論文提出までの様々なアドバイスを頂戴致しました．そのほか多くの皆様のご支援を賜り，本論文を書き終えることができました．心より感謝致します.

遠方より筆者の健康を気遣いしてくれた父と在学中急逝した母に，また，大学へ行くことを後押ししてくれた妻と，それを快諾してくれた子供達に心から感謝します.

参考文献

[1] Open Cloud Manifesto, http://www.opencloudmanifesto.org/

[2] NIST, Special Publication 800-145, ”The NIST Definition of Cloud Computing”, 2011.

[3] 須賀, ”クラウド時代のセキュリティ確保とプライバシ保護を実現する暗号プロトコル技術”, NICT 情報通信セキュリティシンポジウム, 2010.

[4] 経済産業省, SaaS 向け SLA ガイドライン, 2008 年 1 月, http://www.meti.go.jp/committee/materials/downloadfiles/g80207c05j.pdf

[5] 独立行政法人情報処理推進機構, クラウド事業者による情報開示の参照ガイド, 2011 年 4 月, https://www.ipa.go.jp/security/cloud/tebiki_guide.html

[6] 経済産業省, クラウドセキュリティガイドライン 2013 年度版, 2013 年 3 月, http://www.meti.go.jp/press/2013/03/20140314004/20140314004-2.pdf

[7] 総務省, クラウドサービス提供における情報セキュリティ対策ガイドライン, 2014 年 4 月, http://www.soumu.go.jp/menu_news/s-news/01ryutsu03_02000073.html

[8] 総務省, 平成 27 年版情報通信白書, 第 3 部「基本データと政策動向」第 2 節「ICT サービスの利用動向」(3) クラウドサービスの利用動向, http://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h27/html/nc372130.html

[9] C.C.Aggarwal and P.S.Yu ”Privacy-Preserving Data Mining: Models and Algorithms “ , Advances in Database Systems Series, Springer-Verlag, 2008.

[10] Hakan Hacigumus, Hakan Hacg Um Us, Bala Iyer, Chen Li,Sharad Mehrotra, ”Executing SQL over Encrypted Data in the Database-Service-Provider Model”, SIGMOD02, pp.216-227

[11] Craig Gentry, ”Fully homomorphic encryption using ideal lattices”, Annual ACM Symposium on Theory of Computing, 2009.

[12] Damien Stehle, Ron Steinfeld, ”Faster Fully Homomorphic Encryption”, ASIACRYPT2010.

[13] Okamoto, Takashima, ”Fully Secure Functional Encryption with General Relations from the Decisional Linear Assumption”, CRYPTO 2010, pp.191-208.

[14] 総務省, リスト型攻撃対策集について, http://www.soumu.go.jp/main_content/000265404.pdf

[15] SAS 70 type II, http://www.sas70.com/

[16] FMMC, ASP・SaaS 安全・信頼性情報開示認定制度, http://www.fmmc.or.jp/asp-nintei/

[17] A.Shamir, ”How to Share a Secret”, Communications of ACM, Vol.22, No.11, pp.612-613, 1979.

[18] G.Blakley, ”Safeguarding Cryptographic Keys”, Proceedings of the National Computer Conference, pp313-317, 1979.

[19] 上原, 西関, 岡本, 中村, マトロイド的アクセス構造を持つ秘密共有法, 電子情報通信学会論文誌 Vol. J69-A, No.9 pp.1124-1132, 1986.

[20] M.Ito, A.Saito, T.Nishizeki, ”Secret Sharing Scheme Realizing General Access Structure”, Proceedings of 1987 IEEE Global Telecommunications Conference (GLOBECOM 1987), 99-10, 1987.

[21] M. Iwamoto, General Construction Methods of Secret Sharing Schemes and Visual Secret Sharing Schemes, http://ohta-lab.jp/users/mitsugu/research/Thesis/Thesis-iwamoto.pdf

[22] G. R. Blakley, C. Meadows, Security of Ramp Schemes, CRYPTO ’84,vol.196 pp.242-268(1984).

[23] H. Yamamoto: On Secret Sharing Systems Using (k,L,n) Threshold scheme, The Journal of the Institute of Electronics, Information and Communication Engineers, vol.J68-A,no.9, pp.945-952(1985).

[24] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka, On a fast (k, n)-threshold secret sharing scheme, IEICE Trans. Fundamentals, vol.91-A, no.9, Sep. 2008.

[25] J. Kurihara, S. Kiyomoto, K. Fukushima, and T. Tanaka: A fast (k,L,n)-threshold ramp secret sharing scheme, IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, vol. E92-A, No. 8, pp. 1808-1821(2009).

[26] Y. Suga, New Constructions of (2, n)-Threshold Secret Sharing Schemes Using Exclusive-OR Operations, Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013, pp.837-842, 2013

[27] H. Krawczyk: Secret sharing made short, LNCS vol. 773, Springer, pp. 136—146(1993).

[28] 藤井, 多田, 保坂, 栃窪, 加藤, 高速な (2, n) 閾値法の構成法とシステムへの応用, 8C-2, CSS2005.

[29] 多田, 藤井, 保坂, 栃窪, 加藤, 閾値 3 の秘密分散法の構成法, 8C-3, CSS2005.

[30] 藤井, 栃窪, 保坂, 多田, 加藤, 排他的論理和を用いた (k,n) しきい値法の構成法, ISEC2007-05.

[31] 栗原, 清本, 福島, 田中, 排他的論理和を用いた高速な (4,n) 閾値秘密分散法と (k,n) 閾値法への拡張, ISEC2007-04.

[32] G. R. Blakley and C.Meadows, Security of Ramp Schemes, CRYPTO84, pp.242-268, 1984.

[33] 山本博資, (k; L; n) しきい値秘密分散システム, 電子通信学会論文誌, vol.J68-A, no.9, pp.945-952, 1985.

[34] 高荒, 岩村, XOR を用いた高速な (k,L,n) ランプ型秘密分散法に関する研究, コンピュータセキュリティシンポジウム 2009，B9-3, 2009.

[35] 松本, 清藤, 鴨志田, 新谷, 佐藤, “ セキュアデータ保管サービス向け高速秘密分散方式, 第 29 回暗号と情報セキュリティシンポジウム SCIS2012, 1E2-4，電子情報通信学会，2012.

[36] Shamir Adi, ”How to share a secret”, Communications of the ACM 22 (11): 612-613, 1979

[37] 國廣, $(2, 2^m)$ 秘密分散の構成法, unpublished, 2013-07-1

[38] M. Ben-Or, S. Goldwasser and A. Wigderson, ”Completeness Theorems for Non-cryptographic Fault-Tolerant Distributed Computation”, Proc. 20th Ann. ACM Symp. Theory of Computing (STOC 88),1988.

9.

[39] 千田, 五十嵐, 高橋, ”効率的な 3 パーティ秘匿関数計算の提案とその運用モデルの考察”, 第 48 回 CSEC 研究会, 2010.

[40] 千田, 五十嵐, 高橋, ”マルチパーティ計算による効率的なビット分解プロトコル”, 第 50 回 CSEC 研究会, 2010.

[41] 五十嵐, 千田, 高橋, ”高効率 3 パーティ秘匿関数計算の情報理論的安全性”, 第 50 回 CSEC 研究会, 2010.

[42] 橋, 須賀, 岩村, ”XOR を用いる秘密分散法の多値化とそれを用いた秘匿計算法”, 第 65 回 CSEC 研究会, 2014.

[43] 金岡, 宮西, 韓, 北上, 佐藤, 浦野, 白鳥, ”実数演算可能な軽量秘密計算法の一考察”, コンピュータセキュリティシンポジウム 2014, 2E3-4, 2014.

[44] G.Ateniese, C.Blundo, A.D. Santis, D.R. Stinson, Visual Cryptography for General Access Structures, Information and Computation 129, 86-106, 1996.

[45] C.Blundo, A.D.Santis, D.R.Stinson, U.Vaccaro, Graph decompositions and secret sharing schemes, EUROCRYPT'92, pp.1-24, 1992.

[46] A. E. Brouwer, A.M.Cohen and A.Neumaier, Distance -Regular Graphs, Springer-Verlag, 1989.

[47] E. Bannai and T. Ito, Algebraic Combinatorics I :Association schemes, Benjamin / Cummings, Menlo Park, California, 1984.

[48] C.Blundo, A.D.Santis, D.R.Stinson, On the Contrast in Visual Cryptography Schemes, Journal of Cryptology 12, 261-289, 1999.

[49] C.Choi, J.Park, R.Kohno, Contrast Analysis According to Hierarchical Access Structure on VCS, SITA97, Vol.20 No.1 pp.217-220, 1997.

[50] 岩本, 山本, 複数の画像を秘密画像とする視覚復号型秘密分散法, SITA2001, pp.565-568, 2001.

[51] 今井, 古原, 渡辺, ヒューマンクリプトとは, ISEC 2000-17, 2000.

[52] 加藤, 視覚復号型秘密分散法とその応用方式の検討, 博士論文, 1996.

[53] 加藤, 今井, 視覚復号型秘密分散法の拡張構成方法, 電子情報通信学会論文誌, A Vol. J79-A No.8, pp.1344-1351, 1996.

[54] 加藤, 今井, 複数の画像を隠すことのできる視覚復号型秘密分散法の個人認証方式への適用, SITA96, pp.661-664, 1996.

[55] 加藤, 今井, 覗き見攻撃を考慮した視覚復号型秘密分散法を用いた個人認証方式, SCIS97, 25C, 1997.

[56] M.Kim, J.Park, S.Park, K.Kim, A Study on Secret Sharing Scheme Using Visual Cryptography, SCIS97, 25B, 1997.

[57] M.Kim, S.Shin, J.Park, New Construction for Multiple Visual Secret Sharing, SCIS2000, B44, 2000.

[58] K.Kobara, H.Imai, Limiting the Visible Space Visual Secret Sharing Schemes and their Application to Human Identification, ASIACRYPT'96, pp.185-195, 1996.

[59] H.Koga, H.Yamamoto, Proposal of a Lattice-Based VSSS for Color and Glay-scale Images, IEICE Trans. on Fundmentals, vol. E81-A, no.6, pp.1262-1269, 1998.

[60] J.H.van Lint, R.M.Wilson, A Course in Combinatorics, Cambridge Univ. Press.

[61] M.Naor, A.Shamir, Visual Cryptography, EUROCRYPT'94, pp.1-12, 1994.

[62] M.Naor, A.Shamir, Visual Cryptography 2, Lecture Notes in Computer Science 1189, pp.179-202, 1997.

[63] 須賀, 岩村, 櫻井, 今井, 複数の機密画像を埋め込み可能なグラフタイプ視覚復号型秘密分散方式の拡張, 情報処理学会論文誌, Vol.42, No.8, pp.2106-2113, 2001.

[64] Y.Suga, "New Paradigm in Graph-Based Visual Secret Sharing Scheme by Accepting Reversal in Black-White Images", IFIP TC11 20th International Conference on Information Security (SEC2005), 525-536, 2005.

[65] W.G.Tzeng, C.M.Hu, "A New Approach for Visual Cryptography", Designs, Codes and Cryptography, Volume 27, Issue 3, 207-227, 2002.

[66] E.R.Verheul, H.C.A.van Tilborg., Constructions and properties of k out of n visual secret sharing scheme, Designs, Codes, and Cryptography, vol.1, no.2, pp.179-196, 1997.

[67] GAP: Groups, Algorithms, Programming - a System for Computational Discrete Algebra, `http://www.gap-system.org/`

[68] GAP package GRAPE: GRaph Algorithms using PErmutation groups, `http://www.gap-system.org/Packages/grape.html`

本論文の内容に関係の深い発表文献

学術論文

- 須賀 祐治, 岩村 恵市, 櫻井 幸一, 今井 秀樹, ”複数の機密画像を埋め込み可能なグラフタイプ視覚復号型秘密分散方式の拡張”, 情報処理学会論文誌 42 巻 8 号, pp.2106-2113, 2001

査読付国際会議論文

- Yuji Suga, ”New Paradigm in Graph-Based Visual Secret Sharing Scheme by Accepting Reversal in Black-White Images”, Security and Privacy in the Age of Ubiquitous Computing, IFIP TC11 20th International Conference on Information Security (SEC 2005), pp.525-536, 2005
- Yuji Suga, ”Optimal Constructions of Visual Secret Sharing Schemes for the Graph Access Structure”, Proceedings of the Fifth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2011, pp.634-638, 2011
- Yuji Suga, ”New Constructions of (2, n)-Threshold Secret Sharing Schemes Using Exclusive-OR Operations”, Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2013, pp.837-842, 2013
- Yuji Suga, ”A Fast $(2, 2^m)$-Threshold Secret Sharing Scheme Using m Linearly Independent Binary Vectors”, 16th International Conference on Network-Based Information Systems, NBiS 2013, pp.539-544, 2013
- Yuji Suga, ”Classification of Generalized Graph-type (2,n)-Visual Secret Sharing Schemes and Optimal Construction For Multiple Secrets”, International Computer Symposium 2014 Workshop on Cryptography and Information Security, 2014-12
- Yuji Suga, ”Consideration of the XOR-operation based Secure Multiparty Computation”, The Ninth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS-2015), 2015-07

本論文と異なる内容の発表文献

査読付国際会議論文

- Yuji Suga, ”Encryption Methods for Restricted Data Limited in Value Range”, Computational Science and Its Applications - ICCSA 2010 Proceedings, Part IV, pp.284-295, 2010
- Yuji Suga, ”A Fair Exchange Protocol for Attribute Certification”, 2011 Third International Conference on Intelligent Networking and Collaborative Systems (INCoS), pp.457-461, 2011

- Yuji Suga, ”SSL/TLS status survey in Japan - transitioning against the renegotiation vulnerability and short RSA key length problem”, The 7th Asia Joint Conference on Information Security (AsiaJCIS 2012), Best Paper Award, 2012
- Yuji Suga, ”Interactive Fingerprint”, 26th International Conference on Advanced Information Networking and Applications Workshops, WAINA 2012, pp.1023-1026, 2012
- Yuji Suga, ”Countermeasures and Tactics for Transitioning against the SSL/TLS Renegotiation Vulnerability”, Sixth International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing, IMIS 2012, pp. 656-659, 2012
- Yuji Suga, ”Access Control Methods Suitable for Stream Contents Delivery Cloud Services by Using Hash-Chain Based Key Derivation Schemes”, 15th International Conference on Network-Based Information Systems, NBiS 2012, pp.794-798, 2012
- Yuji Suga, ”SSL/TLS status survey in Asia region - Transitioning against the renegotiation vulnerability, CRIME attacks and untrusted X.509 certificate”, International Conference on Internet Technologies & Society 2013 (ITS 2013) , 2013-11
- Yuji Suga, ”SSL/TLS Servers Status Survey about Enabling Forward Secrecy”, The 17th International Conference on Network-Based Information Systems (NBiS 2014), 2014-09

査読なし国内学会

- 須賀祐治, 岩村恵市, 櫻井幸一, ”距離を考慮したグラフに基づくアクセス構造を持つ視覚復号型秘密分散方式”, コンピュータセキュリティシンポジウム 2000 (2000-10)
- 須賀祐治, 岩村恵市, 櫻井幸一, ”グラフ分割法による視覚復号型秘密分散方式の構成”, SCIS2002 (2002-01)
- Yuji Suga,”Classification of Generalized (2,n)-Visual Secret Sharing Schemes with Pixel Expansion is up to 4”, SCIS2012 (2012-01)
- 須賀祐治, ”排他的論理和を用いた (k,n) 閾値秘密分散法の新しい構成とその優位性について”, コンピュータセキュリティシンポジウム 2012 論文集, 2012(3), 185-192 (2012-10-23)
- Yuji Suga,”Optimal Construction of Graph-based Visual Secret Sharing Schemes for Multiple Secrets”, The 7th International Workshop on Security (IWSEC2012) Poster session - Best Poster Award (2012-11)
- 須賀祐治, ”巡回置換行列を用いず m 次元数ベクトル空間を用いて XOR 演算だけで構成可能な $(2, 2^m)$-閾値秘密分散法”, 研究報告コンピュータセキュリティ（CSEC）,2013-CSEC-62(16), 1-8 (2013-07-11)
- 須賀祐治, ”XOR 演算ベースの閾値秘密分散法から秘密計算法を構成する試み”, 研究報告コンピュータセキュリティ（CSEC）,2015-CSEC-69(11), 1-7 (2015-05-14)

付 録A　GAPコード

4.4.2 節で取り扱った GVSSS-$(G, 3)$ の分類に利用した GAP コードは以下のとおりである.

```
#########################################################
# in the case of m=3
#########################################################
LoadPackage("grape");

# define Matrices
S0:= [
[0,0,1],[0,0,1],[0,0,1], [0,1,0],[0,1,0],[0,1,0], [1,0,0],[1,0,0],[1,0,0],
[1,1,0],[1,1,0],[1,1,0], [1,0,1],[1,0,1],[1,0,1], [0,1,1],[0,1,1],[0,1,1]
];
S1:= [
[0,0,1],[0,1,0],[1,0,0], [0,0,1],[0,1,0],[1,0,0], [0,0,1],[0,1,0],[1,0,0],
[1,1,0],[1,0,1],[0,1,1], [1,1,0],[1,0,1],[0,1,1], [1,1,0],[1,0,1],[0,1,1]
];

local i, j, i1, j1;
N:= Length(S0);

# calculate A0(white OR-ed matrix), A1(black)
A0 := []; A1 := [];
for i in [ 1 .. N ] do
A0x := []; A1x := [];
for j in [ 1 .. N ] do
Add(A0x, S0[i][1]+S0[j][1]-S0[i][1]*S0[j][1]
       + S0[i][2]+S0[j][2]-S0[i][2]*S0[j][2]
       + S0[i][3]+S0[j][3]-S0[i][3]*S0[j][3]);
Add(A1x, S1[i][1]+S1[j][1]-S1[i][1]*S1[j][1]
       + S1[i][2]+S1[j][2]-S1[i][2]*S1[j][2]
       + S1[i][3]+S1[j][3]-S1[i][3]*S1[j][3]);
```

```
od;
Add(A0, A0x);
Add(A1, A1x);
od;

# A1-A0 is the difference matrix.
A01:=A1-A0;
for i in [ 1 .. N ] do
for j in [ 1 .. N ] do
if A01[i][j]= -1 then A01[i][j]:=1; fi;
# if A01[i][j]=  2 then A01[i][j]:=1; fi;
# if A01[i][j]= -2 then A01[i][j]:=1; fi;
od;
od;

local comb_index;
# order means number of veritices of induced subgraph
order := 5;
comb := Combinations([ 1 .. N ], order);;
#NrCombinations([ 1 .. N ], order);

###
# Induced sub graph from difference matrix (A1-A0)
InducedASet := [];
# connected graphs from InducedASet
InducedAConnectedSet := [];
# #of InducedASet = #of InducedAConnectedSet_comb_index
# a combination index related to a certain connected induced subgraph
InducedAConnectedSet_comb_index := [];

for comb_index in [1 .. NrCombinations([ 1 .. N ], order) ] do

InducedA := [];
for i in [ 1 .. order ] do
InducedAx := [];
for j in [ 1 .. order ] do
Add(InducedAx, A01[comb[comb_index][i]][comb[comb_index][j]]);
od;
```

```
Add(InducedA, InducedAx);
od;
Add(InducedASet, InducedA);

### Is inducedSubGraph connected ?
An := Sum([1..order-1], x->InducedA^x);
check := 1;
for i1 in [ 1 .. order ] do
for j1 in [ 1 .. order ] do
#### debug for connected verification codes
if An[i1][j1] = 0 then check := check *0; fi;
od;
od;


## Pruning non inpedendent graphs
### There exist same row === non inpedendent
for i1 in [ 1 .. order ] do
for j1 in [ i1+1 .. order ] do
if i1 < j1 then
  if InducedA[i1] = InducedA[j1] then check:=check*0; fi;
fi;
od;
od;

# sI
if check = 1 then
Add(InducedAConnectedSet, InducedA);
Add(InducedAConnectedSet_comb_index, comb[comb_index]);
fi;

od;

Length(InducedASet);
Length(InducedAConnectedSet);
#Length(InducedAConnectedSet_comb_index);

####################################
```

```
## Isomorphism check
G := Group( (1,2) ); II := Group( (18,19) );

# results of connected induced subgraph by cutting isomorphic ones
InducedAConnectedIsoSet := [];
# a combination index related to one of InducedAConnectedIsoSet
InducedAConnectedIsoSet_comb_index := [];
debug_index := 1;
for comb_index in [1 .. Length(InducedAConnectedSet)] do

K := Graph( II, [1.. order], OnPoints, function(x,y)
     return InducedAConnectedSet[comb_index][x][y]=1; end, true);

# accepting first one
if Length(InducedAConnectedIsoSet) = 0 then
Add(InducedAConnectedIsoSet, K);
Add(InducedAConnectedIsoSet_comb_index,
                   InducedAConnectedSet_comb_index[debug_index]);
Print(debug_index); Print("\n");
fi;

# checking whether given graph is isomorphic to one of list of graahs
bool1 := false;

for i in [1 .. Length(InducedAConnectedIsoSet)] do
bool1 := bool1 or IsIsomorphicGraph(K, InducedAConnectedIsoSet[i]);
od;

if bool1 = false then
Add(InducedAConnectedIsoSet, K);
Add(InducedAConnectedIsoSet_comb_index,
                     InducedAConnectedSet_comb_index[debug_index]);
Print(debug_index); Print("\n");
fi;

debug_index := debug_index +1;
od;
```

```
################################################
## generating adjancency matrix from InducedAConnectedIsoSet
InducedAConnectedIsoSetAdM := [];

for i in [1 .. Length(InducedAConnectedIsoSet)] do

# genarating all 0 matrix (allocation of null adjacency matrix)
Induced0 := [];
for i1 in [ 1 .. order ] do
Induced0x := [];
for j1 in [ 1 .. order ] do
Add(Induced0x, 0);
od;
Add(Induced0, Induced0x);
od;
InducedB := [];
InducedB := Induced0;

for j in [1 .. order] do
ad_indexes := Adjacency(InducedAConnectedIsoSet[i], j);
 for k in [1 .. Length(ad_indexes)] do
 InducedB[j][ad_indexes[k]] := 1;
 InducedB[ad_indexes[k]][j] := 1;
 od;
od;
Add(InducedAConnectedIsoSetAdM, InducedB);
od;

# output results
Print("order ="); order;
Length(InducedASet);
Length(InducedAConnectedSet);
InducedAConnectedIsoSet_comb_index;
InducedAConnectedIsoSetAdM;
```