準同型暗号を用いたスケーラブルかつE2EEな音声重ね合わせの実装

上野 真奈[1,a)]　光成 滋生[2]　小林 鉄太郎[1]　村上 啓造[1]

概要：本研究はウェブ会議などのオンラインコミュニケーションツールにおけるスケーラビリティとE2EEの両立を目的として, 楕円 Lifted ElGamal 暗号を用いた暗号化済み音声データ重ね合わせを実装する. 楕円 Lifted ElGamal 暗号を用いることで従来の E2EE オンラインコミュニケーションではできなかった, サービス提供サーバにおける演算処理が可能となり, スケーラビリティの大幅な改善を見込むことができる. 一方で, 演算コストの大幅な増加が問題となる. そこで, 本研究では暗号化済み音声データ重ね合わせを実装し, 各演算処理の速度を測定, リアルタイム通信が可能な処理速度であるかを検証する. 測定は複数の楕円曲線に対して行い, 最適な楕円曲線の検討も行った. この評価実験の結果, 256 ビットの楕円曲線を用い, かつ移動端末の上り通信速度で音声通話のみの実施を仮定した場合, 最大 1024 人まで同時接続した状態で, 暗号化または復号の処理をリアルタイムで実施できることが明らかになった.

キーワード：加法準同型暗号, 実装, 楕円曲線暗号, 音声

The Scalable E2EE Sound Addition System by Additive Homomorphic Cryptosystem

Mana Ueno[1,a)]　Shigeo Mitsunari[2]　Tetsutaro Kobayashi[1]　Keizo Murakami[1]

Abstract: For scalable E2EE online communication systems, such as web conferencing, we implemented the encrypted sound data addition system with additive homomorphic cryptosystem, Elliptic Curve Lifted ElGamal (EC Lifted ElGamal) cryptosystem. In the implementation with EC Lifted ElGamal cryptosystem, the biggest problem is an increase of the computational cost for encryption, addition and decryption processes. In this study, we measured the speed of each processing time with several elliptic curves, and verified whether the speed is enough for real-time communication. We confirmed that our system enables encryption or decryption on time in voice communication with more than 1024 person's connection.

Keywords: Additive Homomorphic Encryption, Implementation, Elliptic Curve Encryption, Sound data

1. はじめに

1.1 研究背景

厚生労働省が主導する働き方改革や, 2019 年に感染が拡大した Covid-19 の影響を受け, ニューノーマルな生活様式が構築されつつある. リモートワークの普及もその一例である. リモートワークの普及に伴い, これまでの直接の会話や対面での会議の代替手段として, Zoom ミーティング [1] や Cisco Webex[2] などのウェブ会議ツールの利用が広まっている. ウェブ会議などのインターネットを介したコミュニケーションにおいて, 第三者への情報漏えいを防ぐことは重要な課題である. 元 NSA 局員である Snowden による NSA の広範な盗聴活動の暴露 [3] 以来, インターネットコミュニケーションツールではエンドツー エンド暗号 (End-to-end Encryption, E2EE) 技術が実装されてい

[1] NTT 社会情報研究所
NTT Social Informatics Laboratories
[2] サイボウズ・ラボ
Cybozu Labs
[a)] mana.ueno.sw@hco.ntt.co.jp

る．ウェブ会議ツールにおいても, 多くで E2EE 技術の導入が進められており, たとえサービスを提供するサーバであってもコミュニケーションの内容を知ることができないようになりつつある [4].

E2EE 技術は第三者への情報漏えい防止に有効な一方で, デメリットもある．その一つが, 会議機能への制限で, 例えば同時接続人数の上限の制限である．ウェブ会議システムにおいて E2EE を実現するためにはリアルタイム性が重要であることから, メッセージの暗号化には, 公開鍵暗号方式と比較して演算コストが小さく, 通信におけるタイムラグの小さな共通鍵暗号が用いられている [4]．現状のウェブ会議システムで利用されている共通鍵暗号方式は暗号化された状態での演算処理を行えないという性質をもち, サービスを提供するサーバは音声重ね合わせなどの処理を行うことができない．結果として, 同時接続ユーザ数に比例して通信量とユーザ端末で行う処理が大きくなるという問題が生じ, 同時接続ユーザ数の最大値に制限がかかる．以上のように現在の E2EE 技術実装方式では, 同時接続ユーザ数の増加が難しい．

1.2 本研究の目標

本研究では, 従来両立の難しい E2EE と多人数接続の両方を実現することを目標とする．この目標を達成するため, 我々はメッセージの暗号化に準同型暗号方式を用いる．準同型暗号は, 平文と秘密鍵を持たないユーザであっても加算または乗算, あるいはその両方の演算を行うことができる性質を持つ暗号である．この性質を用いることで, ウェブ会議のサービスサーバは秘密鍵を持つことなく, ある程度の演算処理を行うことが可能になる．一方で, 準同型暗号は共通鍵暗号と比較して演算コストが大きく, リアルタイムの通信に用いるにはタイムラグが問題になる．そこで本研究では, 加法準同型性をもつ楕円 Lifted ElGamal 暗号方式 [5] を用いて, 音声通話の暗号化, 加算, 復号のそれぞれの処理を行い, その演算処理にかかる時間を測定する．また, 暗号方式に用いる楕円曲線を変更し, 最適な曲線について検討を行う．なお, 本研究では何らかの方法で事前に通信者間の鍵共有が完了しているものとして議論を進める．

1.3 本論文の構成

本論文の構成を以下に示す．2 章では準備として, WebRTC, 音声データおよび準同型暗号方式について述べる．3 章では, 従来技術の比較を行い, その問題点を明らかにする．4 章では, 従来技術の問題点の解決方法として, MCU 方式への準同型暗号の利用について述べる．5 章では, ウェブ会議システムへの実装要件と実装時の評価基準, 本研究における実装環境を示す．また, 楕円 Lifted ElGamal 暗号を用いた場合の, 非圧縮音声データの暗号化加算を実装し, その評価を行なう．最後に 6 章で実験結果を示し, 7 章でまとめとする．

2. 準備

2.1 WebRTC

現在使用されているウェブ会議ツールの多くは WebRTC(Web Real-Time Communication)[6] を基本としている． WebRTC とは Application Programming Interface(API) を経由して, ウェブブラウザやモバイルアプリでリアルタイム通信を実現することを目的として Apple, Google, Microsoft, Mozilla などが立ち上げたオープンソースのプロジェクト及びそのサポートする技術である．ウェブページ内で直接 Peer to Peer(P2P) 通信を行うことで, プラグインのインストールやネイティブアプリケーションのダウンロードを行うことなく, ウェブブラウザ間のボイスチャット, ビデオチャット, ファイル共有などを実装できるようになる．ウェブ会議における利用では, ユーザのみで P2P 通信を行うと各端末の負荷が重くなり通信品質が劣化するため, クライアント・サーバを介してデータの送受信を行うことで回線の問題を解決できるクライアント・サーバ方式が利用される．クライアント・サーバ方式には Selective Forwarding Unit(SFU) 方式と Multipoint Control Unit(MCU) 方式がある．

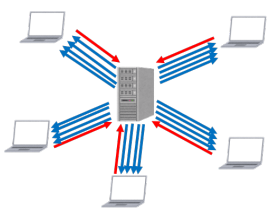

図 1 SFU 方式.

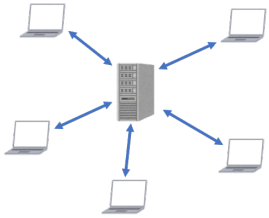

図 2 MCU 方式.

2.1.1 SFU 方式

SFU 方式はサーバがクライアントから集めたデータを別のクライアントに受け流す中継点となる方式である (図 1)．サーバ側は平文を必要とする処理を必要としないため, 共通鍵暗号方式を用いた E2EE はできる．一方で, クライアント端末は同時接続ユーザ数に比例した量の音声・動画を同時再生することとなるため, スケーラビリティを保つことが難しい．ハードウェア的なアプローチによって同時接続ユーザ数上限の改善が図られているが, 現在の最大同時接続可能人数は 30 人程度になっている．なお, Zoom は SFU 方式を基本とした方式で 200 人の最大同時接続可能人数を達成しているが, WebRTC とは異なる独自プロトコルを使用しており, 正確には SFU 方式ではない．

2.1.2 MCU 方式

MCU 方式はクライアントから集めた音声や映像などのデータをサーバ側で処理してクライアント側に流す方式で

ある (図 2). クライアントが取り扱うデータや通信に乗るデータのサイズが同時接続人数によらず常に一定になることから, 最大同時接続可能人数は問題ないが, サーバでデータの重ね合わせを行う際に平文が必要であり, E2EE とは相性が悪い.

2.2 音声データ

音声は符号化され音声ファイル形式に保存される. 音声ファイル形式には符号化した音声をそのまま保存する非圧縮形式と, 圧縮して保存する圧縮形式がある. WAV 形式 (RIFF waveform Audio Format)[7] は通常非圧縮, リニア PCM のサンプリングデータ用のフォーマットである. リニア PCM 方式は符号化方式の一つで, サンプリングされた音声データは時間に対して線形に保存される. WAV 形式は, その本体である音声データ部分が 16 ビットを 1 単位としたブロックになっている. WAV 形式の音声データ本体はブロックごとに加算することで 2 つ以上の WAV ファイルを 1 つに重ね合わせることができる.

2.3 準同型暗号

準同型暗号 (Homomorphic Encryption, HE) とは準同型性を持つ暗号方式であり, 基本的には公開鍵暗号方式である. 公開鍵暗号方式は 1976 年に Diffie と Hellman により提案された概念で [8], 共通鍵暗号とは異なり, あらかじめ送信者と受信者が鍵を共有しているという前提を用いない暗号方式である. そのかわりに, メッセージの受信者はあらかじめメッセージを暗号化するための公開鍵 pk と, それとペアになる復号するための秘密鍵 sk を生成し, 暗号化の鍵を公開する. メッセージの送信者は公開された鍵を用いて暗号文を生成し, 受信者はそれに対応する秘密鍵 sk で暗号文を復号する.

公開鍵暗号方式は基本的には以下の 3 つの関数で構成される.

Gen : 鍵生成関数. 事前準備にあたる関数で, 実行するとそのユーザの公開鍵 pk および秘密鍵 sk のペアを出力する.

Enc : 暗号化関数. 送信したい平文メッセージ m と公開鍵 pk を入力として, 暗号文 C を出力する.

Dec : 復号関数. 公開鍵 pk, 秘密鍵 sk と暗号文 C を入力とし, 平文メッセージ m あるいは復号不可を示す特別な記号 $\perp$ を出力する.

公開鍵暗号方式の中には準同型性を持つものがある. 準同型性とは, 二つの暗号文が与えられた際に, 平文や秘密鍵なしで演算を行うことができ, かつ, その演算結果が正常に復号できる性質である. 準同型暗号においては, 鍵生成 Gen, 暗号化 Enc, 復号 Dec の 3 つの関数に加えて下記の関数を持つ.

Eval : 準同型演算関数. 準同型演算関数は同一の公開鍵で暗号化された複数のメッセージ間に任意の演算を行う関数である. 平文メッセージを m_1, m_2, それぞれに対応する暗号化メッセージを C_1, C_2 とする. 暗号化メッセージを C_1, C_2 を入力として, 準同型演算結果 C または準同型演算不可を意味する記号 $\perp$ を出力する. また, 演算結果 C を入力として Dec を行うことで, $m_1 \circ m_2$ を得られる.

準同型暗号は平文や秘密鍵なしでできる演算の種類により, 完全準同型暗号と加法準同型暗号, 乗法準同型暗号に分類できる. 表 1 にそれぞれの準同型暗号が準同型性を持つ演算についてまとめた. 完全準同型暗号は加法・乗法の両方の演算について準同型性を持つが, 演算コストが大きな方式が多い. 加法準同型暗号と乗法準同型暗号はまとめて Partially homomorphic cryptosystems とも呼ばれる. それぞれが加算, 乗算のどちらかのみの準同型性をもつ方式で, 完全準同型暗号方式と比較して演算コストが小さい. なお, 表中には記載しないが, Somewhat partially homomorphic cryptosystems[9] と呼ばれる, 演算回数などの条件付きで両方の演算を行うことができる方式も存在する.

表 1 準同型暗号の分類

	加法準同型性	乗法準同型性
完全準同型暗号	○	○
加法準同型暗号	○	×
乗法準同型暗号	×	○

2.4 楕円 LiftedElGamal 暗号

Lifted-ElGamal 暗号 [5] は, ElGamal 暗号 [10] を基本とした乗法準同型暗号である. 楕円 Lifted-ElGamal 暗号は乗法準同型暗号である ElGamal 暗号方式に, 楕円曲線を組み合わせることで, 加法準同型暗号にしている. セキュリティパラメータを k, 平文空間を $\mathcal{M}$ とする. このとき, 楕円 Lifted-ElGamal 暗号は以下の 4 つのアルゴリズムにより定められる.

Gen: 素数位数 p の楕円曲線上の巡回群を $\mathbf{G} = \langle P \rangle$ とする. s を $0 \leq s < p$ の整数とし, これを秘密鍵とする. また, sP を公開鍵とする.

Enc: 平文を m とする. 乱数 r を $0 \leq r < p$ からとり, $Enc(m) = (mP + rsP, rP)$ とする.

Dec: $c = (S, T)$ に対して, $S - sT = (mP + rsP) - s(rP) = mP$ を計算する. 最後に離散対数問題 (Discrete logarithm Problem, DLP) を解いて m を得る.

Eval: 2 個の平文 m_1, m_2 に対応する暗号文 $C_1 = Enc(m_1)$, $C_2 = Enc(m_2)$ の各成分を加算し, 加算結果として C を出力する.
$Enc(m_1) + Enc(m_2)$

$= (m_1P + r_1sP, r_1P) + (m_2P + r_2sP, r_2P)$
$= ((m_1 + m_2)P + (r_1 + r_2)sP, (r_1 + r_2)P)$
$= Enc(m_1 + m_2)$
以上より, 加法準同型性が確認できる.

3. 従来方式の問題

3.1 MCU 方式と SFU 方式の比較

MCU 方式と SFU 方式の特徴をまとめると表 2 のようになる. 表中に示す×を改善すること , すなわち MCU 方式に E2EE を実装するか, SFU 方式の同時接続ユーザ数の上限を撤廃することでスケーラブルかつ E2EE されたウェブ会議方式を実現することができる. MCU 方式は使用する暗号方式が共通鍵暗号方式である限り, クライアントサーバに鍵を渡す必要がある. すなわち, E2EE を実現することはできない. また, SFU 方式の同時接続ユーザ数の上限はユーザ端末の性能がボトルネックとなる.

表 2 MCU 方式と SFU 方式の比較

方式	MCU 方式	SFU 方式
最大同時接続ユーザ数	○	×
データの E2EE	×	○

3.2 E2EE と同時接続可能人数の両立の困難性

従来のクライアント・サーバ方式で E2EE と最大同時接続可能人数の増加の両立を図る一般的な方法は SFU 方式のデータ通信量の効率化が主であるが, この方法についても最終的にはユーザ端末の性能がボトルネックとなる. また, データ通信量の効率化を行うにあたってはユーザ端末における処理が複雑になるため, 独自アプリケーションを導入する必要があり, MCU 方式と比較して利用環境の柔軟性が劣る. 200 人を超える大規模なウェブ会議の E2EE を実現するためには, 別の手段を用いる必要がある.

4. 提案手法

本研究ではウェブ会議システムにおける E2EE と最大同時接続可能人数の両立の困難性という問題に対して, MCU 方式にサーバにデータを秘匿したまま演算を行う方法を適用することによる解決を検討する. MCU 方式は 2.1.2 項で示したように, 同時接続ユーザ数が増加してもユーザ端末の負担が変化しないという特徴をもつ. また, 従来のウェブ会議における暗号化は, リアルタイム性が求められることから, 公開鍵暗号方式と比較して圧倒的に演算速度の早い共通鍵暗号方式で行われていた. しかし, 一般的な共通鍵暗号方式では, サーバは平文もしくは暗号鍵を知らない状態で演算処理を行うことができない以上のことから, 共通鍵暗号方式に変わる, サーバにデータを秘匿したまま演算を行う方法を検討する必要がある. サーバに情報を秘匿したまま演算を行う方法には複数あるが, ここでは準同型暗号を用いる.

MCU 方式と準同型暗号方式を組み合わせたウェブ会議システムを検討するにあたり問題となるのは処理速度と安全性である. 図 3 に MCU 方式に準同型暗号を適用した場合のデータの流れを示す. 単純な実装において演算処理の速度の考慮が必要になるのは暗号化, 加算, 復号の 3 点である. このうち暗号化, 復号はウェブ会議の参加ユーザが行う. また, 加算はウェブ会議サービスの提供サーバが行う. 暗号方式として加法準同型性を持つ楕円 Lifted ElGamal 暗号を用いる場合, 演算のコストが最も大きいのは復号であり, ついで暗号化である. 特に復号処理においては離散対数問題を解く必要があり, 演算速度だけでなく演算時の使用メモリについても考慮する必要がある. また, 多人数接続を想定した場合, 加算の演算コストは接続人数に比例して増加することから, 演算コストが問題になる. 安全性については, 鍵を持たない第三者が自由に加算できるという問題がある. 本研究では演算処理の速度に着目し, ウェブ会議システムにおける実用に耐えうる速度を実現できるかの検討を行う.

図 3 MCU 方式に準同型暗号を適用した場合のデータの流れ.

5. 実装要件と目標性能

5.1 前提条件

本研究は以下の条件を前提とし, ウェブ会議システムサーバにセッション鍵を共有せずに暗号化された音声データの重ね合わせを実装する.

(1) 複数人でのリアルタイム音声通信を想定.

(2) 音声データは非圧縮 WAV ファイル形式を使用する. サンプリング周波数は 16kHz で, 音声データブロック数は 1500000 である.

(3) 加法準同型性を持つ暗号方式 楕円 Lifted ElGamal 暗号を使用する.

(4) 複数スレッドによる並列処理を用いる. 使用可能スレッド数はユーザサイドの処理の場合は最大 4 スレッド, サーバサイドの処理の場合は最大 112 スレッドとする.

音声通話はウェブ会議システムにおいて最も重要な機能である. 一般的にウェブ会議における音声データは,

opus[15] などの圧縮形式を用いて取り扱う. 圧縮形式を用いることで, 音声データのサイズを小さくすることができるが, 圧縮の過程でデータの時間線形性が崩れる. 圧縮された音声データを暗号化したまま加算することは現在ある準同型暗号方式では不可能である. 以上の理由から, 本検討では圧縮音声を使用することができないため, 入力は非圧縮形式である WAV ファイル形式を用いることとする.

WAV ファイル形式の音声データの重ね合わせは, 各時間点における振幅値の加算のみで行うことができる. 本研究では加法準同型暗号である楕円 Lifted ElGamal 暗号を用いる. 加法準同型暗号方式は加算しか準同型性を持たないが, 完全準同型暗号方式と比較して演算コストが小さい. 楕円 Lifted ElGamal 暗号の復号では離散対数問題を解く必要があり, 演算コストが大きくなることが懸念されるが, 本実装における WAV ファイルの加算では, 一つの平文 m の範囲は符号付き 16 ビット ($-32768 <= m < 32767$) であり, 事前計算でテーブルを作って置くことで, 十分高速に解くことができる. 本検討において準同型性が必要な演算は加法のみであること, またリアルタイム音声通信のためには高速な処理が必要になることから, 楕円 Lifted ElGamal 暗号を用いる.

同様に, 処理の高速化のため, 本研究では複数スレッドによる並列処理を行う. 処理の振り分けには後述の OpenMP[14] を用いる. 使用可能なスレッド数の最大値は, ユーザ端末で演算を行う暗号化および復号については 4 スレッド, サービス提供サーバで演算を行う加算については 112 スレッドとする.

5.2 目標性能

本稿ではスループット, レイテンシ, データ通信量の 3 つの条件が全て満たされているとき, リアルタイム通信が可能であるとする. これらを目標性能として表 5.2 にまとめた.

5.2.1 スループット

スループットは単位時間あたりに与えられた処理を完遂できる量を示すパラメータである. MCU 方式に部分準同型暗号を適用するにあたっては暗号化, 加算, 復号の 3 つの処理それぞれについてスループットを達成することが求められる. 例えば暗号化のスループットの下限値は, サンプリング周波数 16kHz の WAV ファイル形式音声データを平文入力としたときに, 1 秒以内に 16000 回の暗号化処理を行うことができる必要があるという要請から決定される. この値から逆算して, 1 回の暗号化処理にかけても良い時間の最大値は 62.5×10^{-6} 秒である. これは動作周波数 1.4GHz の CPU を用いた場合は 87500 クロック, 動作周波数 2.7GHz の CPU を用いた場合は 168750 クロックに該当する. 本実験では使用する CPU に応じてこれらのクロック数の上限値と比較, 評価を行う.

5.2.2 レイテンシ

レイテンシは転送要求から実際にデータが送られてくるまでに生じる通信の遅延時間で, 本研究においては暗号化, 加算, 復号の 3 つの処理及びクライアント–端末間の通信にかかる時間を合計した値が目標性能を下回ることが求められる. レイテンシの目標性能は ITU-T Rec. G. 114[11] で規定される, 「ほとんどのユーザアプリケーションにとって許容範囲である」値を引用し, 150ms 以下にすることを目標とする. なお, 通信による遅延時間については参考値として東京–大阪間を 500km としたとき, その区間で 3ms 程度であると概算できる [2]. WebRTC における 1 パケットあたりの送信量が 20ms, 音声データのサンプリング周波数を 16kHz であると仮定すると, 1 パケットあたりに含まれるデータブロック数は 320 ブロックである. レイテンシを考えた時, 1 パケットあたりの暗号化, 加算, 復号の演算時間と通信遅延が表に示した目標性能を下回る必要がある. 上記の条件をもとに, 暗号化, 加算, 復号の演算をそれぞれ 1 回ずつ行った場合の和の上限値は 6.4×10^5 クロックになる.

5.2.3 データ通信量

データ通信量はユーザ–サーバ間で転送するデータのサイズである. 目標性能としては NTT ドコモの公開する移動通信端末の実行速度計測結果の上りの中央値である 35Mbps を設定した [13]. また, 本実験では, アフィン座標系と比較して演算が高速なヤコビ座標系で一貫してデータを取り扱う. 256 ビットのヤコビ座標系の楕円曲線を仮定した場合, 1 ブロックあたりの暗号文のサイズは 1536 ビットであり, データ通信量は 24.58Mbps となる. 同様にして, 他の楕円曲線のデータ通信量を計算すると, 160 ビット曲線で 15.36Mbps, 192 ビット曲線で 18.43Mbps, 224 ビット曲線で 21.50Mbps, 384 ビット曲線で 36.86Mbps であることがわかる.

表 3 目標性能

評価項目		目標性能
スループット [a]	1.4GHz CPU	87500 clock 以下
	2.7GHz CPU	168750 clock 以下
レイテンシ [b]		6.4×10^5 clock 以下
データ通信量 [c]		35Mbps 以下

[a] 5.2.1, [b]5.2.2, [c]5.2.3 をそれぞれ参照.

5.3 環境

本実装では 2 つの環境を用いた. これ以降, ユーザ端末を想定した環境を環境 1, ウェブ会議サービスサーバを想定した環境を環境 2 とし, それぞれの性能を以下に示す.

環境 1 :

- CPU : Intel Core i5

- 動作周波数：1.4 GHz
- コア数：4 (スレッド数：8)
- メモリ：16GB

環境 2：

- CPU：Xeon Platinum 8280 ×2
- 動作周波数：2.7GHz
- コア数：28 ×2 (スレッド数：56 ×2)
- メモリ：192GB

5.4 実装

5.4.1 暗号の適用方法

楕円 Lifted ElGamal 暗号ライブラリ [5] を用いて非圧縮音声ファイルを, 暗号化, 加算, 復号するプログラムを作成し, 入力データ 16 ビットの各演算処理にかかる時間を測定した. 非圧縮音声ファイルとしては WAV ファイル形式を使用した. WAV ファイル形式の部分準同型暗号による暗号化を図 4 に示す. 図 4 の上段は平文を表し, 下段は暗号文を示す. また, 図中には WAV ファイル形式のヘッダは記載しない. WAV ファイル形式の音声データは 16 ビットのブロックごとに暗号化される. ブロック暗号の ECB モードに近い適用方法であるが, 楕円 Lifted ElGamal 暗号の暗号化に乱数が含まれること, また, 楕円 Lifted ElGamal 暗号が DDH 安全であることから, 同一の平文を暗号化した結果が同じになることや, 平文が 0 である時に判別できるという問題はない.

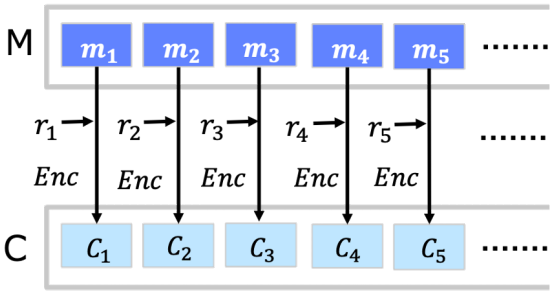

図 4 部分準同型暗号による音声データの暗号化.

5.4.2 OpenMP

本研究では処理の高速化のため OpenMP[14] を使用する. OpenMP は非営利団体 OpenMP Architecture Review Board(ARB) によって規定されている業界標準規格である. 共有メモリ型並列計算機用のプログラムの並列化を記述するための指示文, ライブラリ関数, 環境変数などを規格化したもので, これを用いることでユーザはマルチスレッドプログラムを平易なコードで記述することができる. 例としてアルゴリズム 1 に OpenMP で for 文によるループ計算を並列化する際の記述を示す. なお, OpenMP は C 言語及び Fortran 言語での実装が可能だが, ここでは C 言語で実装する場合を示す. なお, 使用する並列処理で使用するスレッド数は環境変数 OMP_NUM_THREADS で指定する.

アルゴリズム 1 OpenMP 利用の例

```
#pragma omp parallel for
for(i = 0; i < N; i++){
    (ループさせたい演算)
}
```

6. 実験結果と評価

6.1 実験 1：2 者間通信時の演算処理時間の測定

5.3 項で示した環境 1 において楕円 Lifted ElGmal 暗号方式を用いた音声データの暗号化, 加算, 復号を実装し, 各処理の演算にかかるクロック数を測定した. ここで, 入力となる音声データの数は 2 つであり, 本項における加算演算は 2 つの暗号文の加算であることに留意されたい. また, 音声データはそれぞれ 1500000 個のデータブロックで構成され, 1500000 回のループの平均値をベンチマークとする. 本項では OpenMP による 4 スレッド並列処理を行う. 各処理間の値の受け渡しはヤコビ座標系で行い, 暗号文は 3 つのパラメータを持つ楕円点 2 つでで表現される. 楕円曲線としては, Certicom Research のまとめる SEC 2: Recommended Elliptic Curve Domain Parameters の version 1.0[16] および version 2.0[17] で示される secp160k1, secp192k1, secp224k1, secp256k1, secp384r1 および, IEEE1363[18] で示される p160_1, NIST[19] が示す NIST_P192, NIST_P224, NIST_P256, NIST_P384 の 10 種類を用いた. 結果を表 4 にまとめた.

表 4 楕円曲線と演算速度

楕円曲線	暗号化	加算	復号
secp160k	4314	362	20511
p160_1	4283	372	26481
secp192k1	6443	404	33344
secp224k1	9674	533	41615
secp256k1	12557	582	39404
secp384r1	44396	1205	189133
NIST_P192	4733	332	27395
NIST_P224	9809	524	49450
NIST_P256	11650	552	59717
NIST_P384	43558	1362	197349

[clock]

結果より, 同一楕円曲線で暗号化, 加算, 復号の処理の演算時間を比較すると, 復号に最も処理時間が長く, 時点で暗号化, 最も処理時間が短いのは加算である. 2.4 項で示したように, 楕円 Lifted ElGamal 暗号の復号では $mP \to m$ を求める処理が含まれており, 離散対数問題を解く必要がある. また, 加算はヤコビ座標系の楕円加算 1 回のみで構成されるため, 暗号化, 復号と比較して大幅に処理時間が短い.

以上の結果を 5.2 項に示した, スループット, レイテンシ, データ通信量の 3 つの評価基準値と比較すると表 5 のようになる. 表に示す値のうち, 目標性能を満たさない値のセルをグレーにしている. 表より, 384bit 曲線は secp 曲線, NIST 曲線ともにスループットとデータ通信量の点で条件を満たしていないことがわかった. また, 3 つの基準を全て満たし, 音声リアルタイム通信における利用が可能なのは, 256 ビット以下の楕円曲線であることがわかった. なお, データ通信量については本実装ではより演算が高速なヤコビ座標系を用いて楕円点を表現しているが, ヤコビ座標系–アフィン座標系間の座標変換を効率的に行うことができれば, 表に示す値の $\frac{1}{3}$ 程度に削減することができる.

6.2 実験 2：多人数接続時のサーバ加算速度と並列処理

二つ目の実験として, ウェブ会議サービスのサーバ加算を想定して, 多人数の暗号化済み音声データの加算時間の測定を行った. 本実験では 5.3 項に示すサーバを想定した環境 2 を使用する, 楕円曲線は secp256k1 を使用し. 加算処理は 1500000 回繰り返した. 加算する暗号化済み音声データ数および並列処理に用いるスレッド数を変化させ, それぞれの演算にかかるクロック数を測定した, 暗号化済みデータ数は 1 から 1024 の範囲で変化させた. また, 並列処理に用いるスレッドの数は 1 から 112 の範囲で変化させ, ジョブの振り分けには OpenMP を使用した. 以上の結果を表 6 に示す. 5.2 項で示した加算処理 1 回あたりにかけることのできる最大クロック数は 168750 クロックであった. 表中では評価基準値を超えるセルをグレーにしている.

これらの測定値を用いて, 加算する暗号化済み音声データ数の増加と処理時間の関係, および並列処理に用いるスレッド数と処理時間の関係についてグラフにまとめた.

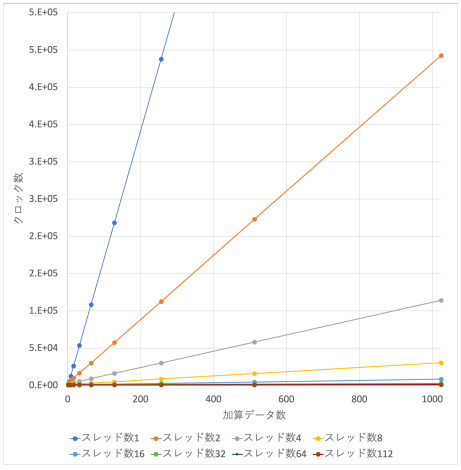

図 5 加算データ数と加算時間

6.2.1 暗号化済み音声データ数と演算時間

図 5 に暗号化済み音声データ数とサーバ加算時間の関係をまとめた. 図 5 の横軸は加算する暗号化済み音声データ数, 縦軸はサーバ加算にかかるクロック数を表す. 図中のプロットは並列処理に使用するスレッド数ごとに色分けしている. 図 5 より, 加算する暗号化済み音声データ数が増加すると, 比例して加算処理にかかる演算時間が増加することがわかった. また, 並列処理に用いるスレッドの数を増加させることで, 加算処理時間を削減することができる.

6.2.2 並列スレッド数と加算時間

図 6 に並列処理に使用するスレッド数とサーバ加算時間の関係をまとめた. 図 6 の横軸は並列処理に用いるスレッド数を表し, その 1 から 112 までの範囲で変化させた. また, 縦軸は加算処理にかかるクロック数を表す. 横軸は基数 2 の対数スケールとなっている. また, 図中のプロットは加算する暗号化済み音声データ数ごとに色分けしている. 図 6 より, スレッド数を増加に伴い, 加算処理が高速化できることがわかる, また, 加算する暗号化済み音声データの数が大きいほど傾きが大きく, 処理の並列化による高速化の効果が大きい. 並列処理に用いるスレッド数が小さい時はその高速化の効果は指数関数的であるが, 30 スレッドを超えた付近で高速化の効果が飽和することがわかった.

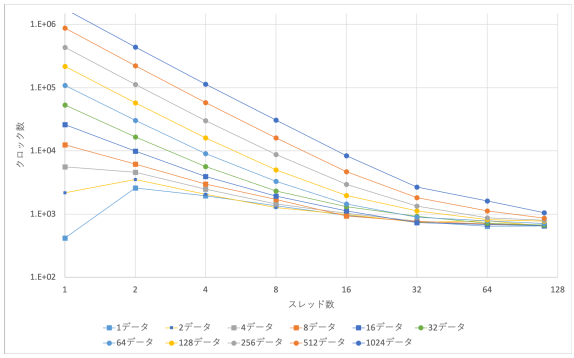

図 6 並列処理のスレッド数と加算時間

7. さいごに

本研究ではウェブ会議などのオンラインコミュニケーションのスケーラブルな E2EE を目的として, 楕円 Lifted ElGamal 暗号を用いた音声重ね合わせを実装した. 楕円 Lifted ElGamal 暗号を用いることで従来の E2EE オンラインコミュニケーションではできなかった, サービス提供サーバにおける演算処理が可能となる一方, 演算コストの大幅な増加が問題となることから, 実装については暗号化・加算・復号の各処理の速度および, 接続ユーザ数増加によるサーバ処理負担への影響を評価した. 本実装実験により, 256 ビット以下の楕円曲線を用いた場合, 楕円 Lifted ElGamal 暗号によるリアルタイム E2EE 通信が可能であることがわかった. また, オンラインコミュニケーションツールのサービス提供サーバにおける加算処理は, 接続ユーザ数の増加に対して指数関数的に処理時間が増加するが, 並列化処理を用いることで, 処理の高速化ができることがわ

表 5　測定値と目標性能の比較

		スループット			レイテンシ	データ通信量
		暗号化	加算	復号		
目標性能		87500 [clock]			643125 [clock]	35 [Mbps]
楕円曲線	secp160k	4314	362	20511	25186	15.36
	p160_1	4283	372	26481	31136	15.36
	secp192k1	6443	404	33344	40191	18.43
	secp224k1	9674	533	41615	51821	21.50
	secp256k1	12557	582	39404	52544	24.58
	secp384r1	44396	1205	189133	234734	36.86
	NIST_P192	4733	332	27395	32460	18.43
	NIST_P224	9809	524	49450	59784	21.50
	NIST_P256	11650	552	59717	71919	24.58
	NIST_P384	43558	1362	197349	242269	36.86

表 6　複数ユーザのデータ加算の演算速度

		加算データ数										
		1	2	4	8	16	32	64	128	256	512	1024
並列スレッド数	1	418	2168	5606	12479	26247	53646	189133	218372	437780	875454	1751803
	2	2591	3517	4615	6206	9922	16697	30323	57646	112616	222914	442309
	4	1945	2062	2486	2986	3935	5661	9104	16165	30193	58121	114401
	8	1358	1266	1452	1700	1938	2326	3318	5013	8783	16033	30667
	16	951	990	1033	937	1121	1316	1456	1976	2962	4675	8422
	32	757	743	778	765	735	921	898	1132	1337	1831	2698
	64	644	785	676	705	686	715	779	817	876	1122	1617
	112	646	651	666	657	655	671	707	777	801	865	1054

* 楕円曲線は secp256k1 を使用.　　[clock]

かった. 本実験では, secp256k1 曲線を用いた場合 1024 人まで同時接続が可能であることが実証された.

謝辞　インテル株式会社堀越将司氏による第二世代 Xeon SP 評価環境の協力を感謝する.

参考文献

[1] Zoom HP, "https://zoom.us/jp-jp/meetings.html".
[2] Cisco Webex HP, "https://www.webex.com/ja/index.html".
[3] G. Greenwald. "No Place to Hide: Edward Snowden, the NSA, and the U.S. Surveillance State", Metropolitan Books, USA, 2014.
[4] Josh Blum, et al., "E2E Encryption for Zoom Meetings", https://github.com/zoom/zoom-e2e-whitepaper, 2020.
[5] 光成 滋生, 「クラウドを支えるこれからの暗号技術」, 秀和システム, 2015.
[6] WebRTC Org., https://webrtc.org/
[7] Microsoft, "Extensible Wave-Format Descriptors", https://docs.microsoft.com/en-us/windows-hardware/drivers/audio/extensible-wave-format-descriptors
[8] W. Diffie and M. E. Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory, vol.IT-22, No.6, pp.644-654, Nov, 1976.
[9] C. Gentry, "A Fully Homomorphic Encryption Scheme", Ph.D. Dissertation, Stanford University, 2009.
[10] T. ElGamal, A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms, IEEE Transactions on Information Theory, Vol.31, No.4, pp.469-472, 1985, doi:10.1109/TIT.1985.1057074.
[11] ITU-T, Rec. G.114, https://www.itu.int/rec/T-REC-G.114
[12] Cisco, 「パケット音声ネットワークでの遅延について」, https://www.cisco.com/
[13] NTT ドコモ, "実行速度計測結果", https://www.nttdocomo.co.jp
[14] OpenMP HP, "https://www.openmp.org/"
[15] JM. Valin, K. Vos and T. Terriberry, "Definition of the Opus Audio Codec", IETF RFC 6716, IETF, 2012.
[16] Standards for Efficient Cryptography Group, "SEC 2: Recommended Elliptic Curve Domain Parameters", Certicom Research, 2000, Version 1.0.
[17] Standards for Efficient Cryptography Group, "SEC 2: Recommended Elliptic Curve Domain Parameters", Certicom Research, 2010, Version 2.0.
[18] "IEEE Standards: 1363-2000, 1363a-2004, 1363.1-2008, 1363.2-2008, 1363.3-2013"
[19] "Digital Signature Standard (DSS)", FEDERAL INFORMATION PROCESSING STANDARDS PUBLICATION, 2013.