

会議や少人数授業のための TV 会議システムの導入

高嶋 隆一

Introduction of Video Conference Systems for Meetings or Lectures with a Small Number of Students

Ryuichi TAKASHIMA

Accepted July 20, 2020

抄録：新型コロナウイルスの蔓延により、人的な接触を減らすための TV 会議システムを使った会議や少人数授業が注目を集めている。TV 会議システムはいままでは研究所の計算機を専門とするグループによって導入され、素粒子実験などのテーマを絞った目的に利用されてきた。今では直接の人的接触を行わない環境で会議や学生の指導を行うことが求められており、それを実現するセキュリティ的にも問題のない TV 会議システムでオープンソースである Jitsi の導入を試みたので、実際の運用も含めた現状について考察する。

検索語：TV 会議, オープンソース, Jitsi

Abstract: Due to the spread of new coronaviruses, conferences and small-group classes that use video conferencing systems to reduce human contact are attracting attention. Until now, video conference systems were introduced by groups specializing in computers at Research Laboratories, and have been used for focused themes such as elementary particle experiments. Nowadays, it is necessary to conduct meetings and instruct students in an environment where there is no direct human contact, so we aim to introduce the open source Jitsi as a video conferencing system that does not have any security problems to realize this. This study reports the current situation of Jitsi use, including actual operation.

Key words: video conference, open source, Jitsi

1. はじめに

素粒子物理学の実験分野で最初に World Wide Web¹⁾ が利用されるようになってから、世界各国に分散する研究グループ間の連絡方法が進化を続けてきている。その一つとして、映像や音声を計算機同士の通信機能を通じて行う TV 会議システムの利用が進められてきた。最初はエコーキャンセル機構を含んだ特殊な機材が必要とされた。その時導入された機材には H.323 プロトコルを使用しているものであると謳われており、Polycom 社製のものがよく利用されていた。最近ではパソコンのマイクや Web カメラ、スピーカーを組み合わせた簡便な TV 会議システムが利用されるようになってきた。調べて見ると Session Initiation Protocol というものに移行したようである。CERN や KEK といった素粒子実験の大規模な研究所ではその技術に基づいた Vidyo というシステムが導入されている。その際には会議室を設定するためのサーバーと呼ばれる計算機が必要となる。その計算機を運用するには特別なノウハウが必要とされたので、研究所の計算機グループが導入作業を行ってきた。ところが最近、コロナウィルスで自宅から業務を行うリモートワークが推奨されるなどして、小規模な事業所にもそのシステムの運用が行われるようになってきた。本学においても TV 会議システムを運用することにより、コロナウィルスの感染リスクを減らす試みが求められている。この問題について情報センターの運用担当者に相談したところ、セキュリティやネットワーク資源の利用が過大にならないような運用方法について提案をいただいた。

特に重要なのがセキュリティの問題である。この中核をなす Secure Socket Layer やサーバ証明書、鍵ファイルについて理解をすることが、TV 会議システムやメールサーバーを運用するために必要である。²⁾ 偶然に検索で見つけたオープンソースと呼ばれる無料で配布されている Jitsi を試行的に導入して見ることにした。Jitsi はブルガリヤ語でワイヤーの意味ということである。³⁾ 今回は大学の情報センターが運用している計算機システムの認証の形態に準拠する形でのセキュリティの保証形態をとることになった。この認証の鍵となる数学理論についても簡単に触れて、実際の運用の基礎についても解説する。また TV 会議のページを表示するには DocumentRoot と呼ばれる場所が通常使われてきた `/var/www/html` から jitsi が実際に入っている `/usr/share/jitsi` を指すように変更する必要がある。これは http の実際の動作プログラムである apache2 の VirtualHost という機能を使って実現する。またインストールのスクリプトがエラーなく実行されることが大事である。また jicofo, jitsi-videobridge2, prosody という 3 つのサービスが自動的に開始されるが、このサービスがエラーなく実行されていることも重要となる。問題が起きた時の対処法についても解説する。また、実際にビデオ会議を使ったときに留意すべきことについて説明し、その体験に基づく運用方法についても考察する。

2. 秘密鍵ファイルの生成とサーバ証明書の取得

まず現在の暗号化通信の仕組みについて理解することが必要である。通常ビデオ会議や Web のサービスを行う場合、サーバーにアクセスすることになる。その場合に利用者の個人情報がネット上を流れることになる。その状況は声に出して大声で叫んでいるのと同じなので、何を言っているか分からないように banana なら cbobob とひと文字ずつずらして叫ぶと暗号化ということになる。サーバーに html などの情報の送信を依頼することになるがそのような依頼のメッセージそのものを暗号化する

のが暗号化通信となる。

暗号化と復号化をメッセージについて行う方式として、現在もっとも多く使われているのが RSA 方式と呼ばれるもので、基本原理は二つの素数 p, q について $n = pq$ で掛けあわせた数 n を法とする計算を行う。このようなことは計算機の上では普通に行われている。計算機がやりとりする数は 2 進数で 32 ビットとなっているからである。この時、オイラー関数は $\phi(n) = (p-1)(q-1)$ で定義される。 n より小さく n と互いに素な自然数は $\phi(n)$ 個あり、それらは掛け算について群となる。 n と互いに素な数というわけで、 p, q の倍数を除いた数となるので、 $\phi(n)$ 個となるわけである。 $\phi(n)$ より小さい正の整数で、互いに素である e (encrypt の意味) を一つ選ぶ。その e に対して、 $\phi(n)$ を法とする計算で逆数となる d (decrypt の意味) を選ぶことができる。この時サーバに送りたいメッセージ m は n を法とする数のどれかとなる。暗号化されるメッセージを c とすると、 $c = m^e$ として暗号化される。オイラーの定理により、全ての n より小さく n と互いに素な m に対しては $m^{\phi(n)} = 1$ となる性質がある。そのほかの p, q の倍数も復号されることがわかっている。そのことから、復号化は $m = c^d$ により行われる。これにより、サーバー側に、 p, q, d を含んだ秘密鍵を含んだ非公開の鍵ファイルがあれば、サービスを受ける側はサーバーが提示するサーバー証明書にある n, e の公開鍵を使って、要求するメッセージを含んだ文字列を暗号化して通信することができる。

具体例として $p = 3, q = 5$ としてみたとすると $n = 15, \phi(15) = 8$ となる。そうすると $e = 3, d = 3$ を選ぶことができる。15 を法とする数 m を 3 回かけたもの数を暗号化した数 c は、それを 3 回かけると元の m に戻るというわけである。15 を法として、15 未満の 15 と互いに素な数を 8 回かけたものはすべて 1 となる。またそれ以外の、3 と 5 の倍数も 9 回掛けるとともに戻る。

通常はこれを最大 2048 ビットの数の中で行うのが以下の鍵対生成となる。公開鍵を含んだサーバー証明書はそのサーバーを承認している上位の機関の正当性を示している。ブラウザーは自動的にその正当性を確認するため、通信がブラウザーを通じて行うことができるようになっている。サーバ証明書は上位機関の証明なしに作ることも出来るが、その場合は警告が出るか、通信中のどれかの段階で拒否されることになる。

実際に Jitsi がインストールされるのは 22 行目の apt コマンドとなる。最初に素粒子実験のサーバーにテスト的にインストールしようとしたときは、必要なポートが開いていなかったためにインストールが途中で止まるということになった。ドメイン名と httpds のポートが開いていない場合、TV 会議のサーバーとしては動作しないのでインストールが実行されずエラーで終了する。そこで、テスト運用は諦めて、情報センターに正式にお願いすることになった。その時に Let's Encrypt のサーバ証明書をインストールして使用することは、情報センターとしては推奨しないということで、情報学研究所に正式な申請をすることになった経緯がある。22 行目のインストールの手順で正統性を保証されたサーバー証明書の場所を入力し、秘密鍵との整合性が確認されることにより、インストールが成功する。その結果、インストール後に prosody の status を見ることによる動作確認が出来る。またセキュリティの観点からは Uncomplicated Fire Wall (ufw) というツールを使って必要なポートのみを開けておく必要がある。画像などのビットストリームが流れるポートは https (443 番) ポートでこのポートが https のサービスが動作する番号である。今回は慣れている apache2 を https のサービスを行うプログラムとして利用している。またこの時、apache2 の実行主体は www-data であるので /usr/share/jitsi の所有権も変更した。

サーバ証明書の作成を依頼するときには、指示文書に従うと組織の部門名をきめて申請することが

ソースコード 1: サーバー証明書と秘密鍵の設定, テレビ会議のインストール

```

1 # 秘密鍵を作るため、200kbyte 程度の乱数を生成する
2 head -c 200k /dev/urandom > randfile1.txt
3 head -c 200k /dev/urandom > randfile2.txt
4 head -c 200k /dev/urandom > randfile3.txt
5 cd /etc/ssl/private
6 # まずパスフレーズを考えておき、情報学研究所の指示に従って秘密鍵server.key を生成する
7 openssl genrsa -des3 -rand randfile1.txt:randfile2.txt:randfile3.txt 2048 > server.key
8 # IPC に送るための Certificate Signing Request(CSR)を作成する。
9 # 秘密鍵をつかって、
    IPC の指示文書にに従い、パスフレーズ、国コード、地域、組織名等を入力する
10 openssl req -new -key server.key -sha256 -out server.csr
11 # CSR を IPC に送る。メールで、サーバ証明書の一部を server.cer として取得し
12 # 中間証明書intermediate.cer (情報学研究所と上位の組織の関係を記述) も取得して結合し
13 # サーバ証明書cert.pem を作成する。この中には上位組織 (NII 等)の署名が入っていて、公開鍵も入
    っている
14 cat server.cer intermediate.cer > cert.pem
15 # 実際に秘密鍵を使うときはパスフレーズが入っていると利用不能なため、外しておく
16 openssl rsa -in server.key -out server.key
17 # この後、http のサービスを行う nginx をいれたり apache2 に変えてみたり、jitsi-
    meet をいれたり
18 # 入れなおしたりした
19 apt purge jicofo jitsi-meet-prosody jitsi-meet-web jitsi-meet-web-config jitsi-videobridge2 \
20 lua-bitop lua-event lua-expat lua-filesystem lua-sec lua-socket lua5.1 prosody
21 # 再度入れなおすためには --
    reinstall スイッチを入れる。証明書、秘密鍵の場所が途中で聞かれる
22 apt --reinstall install jitsi-meet -y
23 # ここでサーバ証明書やがどんなものか見ておくのも良い。
    x509 が証明書用のスイッチ、まず中間証明書
24 openssl x509 -in intermediate.cer -text -noout
25 # 最終的なサーバ証明書cert.pem の中身を見る。この中には暗号化用の素数が含まれる
26 openssl x509 -in cert.pem -text -noout
27 # 秘密鍵server.key の中身を見る。復号化用の素数などが含まれる
28 openssl rsa -in server.key -text -noout
29 # テレビ会議に必要なサービスの状態を確認する
30 systemctl status jicofo
31 systemctl status jitsi-videobridge2
32 systemctl status prosody
33 # prosody の 5281 ポート関連、localhost 関連のエラーは無視して良い

```

必要となっている。セキュリティに責任を持つ部門名を指定することになるが、理学科ということになるので「Department of Science」で申請を行うこととした。理学科では教科教育と教科専門という二つ分野の教員の昇任申請を行う分野がある為である。この辺りも情報センターの担当者と相談しながら決めていった。

apache2 を通常のインストールで入れた場合、ドメイン名だけでアクセスしたとき、どのディレクトリを見に行くかが、/var/www/html となっている。この部分を変更する必要がある。

/etc/apache2/virtualhost-available に default の設定ファイルがあるが、これをどこか別のところに移すなりして、jitsi のバーチャルホストの設定のみが機能するようにすれば良い。

この段階ではホスト名にアクセスすると、誰でも TV 会議の部屋を開設出来る。これではトラフィックのコントロールが出来ないので、部屋の開設権を登録してあるユーザーのみに与えることが必要である。これはサーバーの管理者しかできないので、音声レベルの適切な管理やマイクロフォンの増幅度の適正化など、部屋の開設者がトラフィックについて管理できる方に開設権を与えることが必要である。

開設権を与えるコマンドは `prosodyctl register` と `prosodyctl adduser` である。会議室の名前そのものが一種のパスワードとなるので、会議の開催時刻、部屋の名前をあらかじめ会議の参加者にメールで告知しておく、会議が成立することになる。また会議の参加者も第 3 者が開催時刻、部屋名を知ることの無いようなセキュリティに関するリテラシーや情報倫理を有していることが必要である。

3. 実際の運用を行った結果に関して

サーバーの認証を得るときの組織の部門名を理学科としたことから、まずは理学科の教員に作成した会議室への入室の案内をメールで送った。数名の教員に入室していただいて、音声とビデオの画像の表示の状況を確認した。こちらの端末が `kuewifi` で接続した場合、`hepnet` という外部のネットワークに接続した場合なども音声、画像とも途切れることなく接続が可能であった。また龍谷大学の TA と接続試験を行った時には画面共有の機能を使い、会議時の資料提示が可能であることも確認した。この時は先方は長浜の自宅で、こちらは学内に引き込んだ `hepnet` の計算機からという条件であったが問題なく動作を確認出来た。また、情報センターの担当者には会議室の開設権限をとってもらい運用試験を行った。情報センターでも利用を検討している `Google クラスルーム`、既に利用者が居ると思われる `Skype` や `Zoom` と比べても、利用料金や制限が必要無い点が評価された。

また最近、大学院生と前期授業の打ち合わせを `Jitsi` を使って行ったところ、`Microsoft Teams` という TV 会議システムを情報センターで試験して見たという話を聞いた。学生の `Office365` のサイトライセンスが学生には与えられており、その中に `Teams` が含まれているらしい。教員も同様に個別に登録してライセンスを取得する仕組みとなっている。ただそちらは教員が通常使用している `Office 2019` とは別物ということになる。

リモートでの授業開始は 2020 年 4 月 20 日であり、履修登録も今年度の最終盤となっていた。このシステムの利用者を増やしていくことも課題となっている。また大学院の授業などでは、このシステムの設置の経験も含めて、サーバーシステムを理解するための演習も行いながら取り上げていく予定である。

また、いままでも素粒子実験のネットワークの運用時に `SSL` を使ったメールサーバーの運用を行ってきたが、その場合にメールクライアントの利用時に発生するエラーメッセージに悩まされてきた。今回正当な証明書の発行手順について、情報センターからの指示文書に従った設定を行うことにより、警告やエラーメッセージが発生することなく TV 会議システムを使うことができた。情報システムの運用の際やユーザとして暗号化通信を使う際にも、今回の経験が役に立つと思われる。

さらに、今回、鍵情報を取り扱った `Jitsi` のインストールを行ったが、インストールスクリプトを使う際、秘密鍵のある場所を指定したりしているが、動作するスクリプトに改ざんがないか不安になる。このようなことも、サーバーの管理ではどのように改ざんがないことが保証されているかということも管理する側が理解しておくことが必要である。`Ubuntu` などのパッケージのインストール先が正当なものであることが大事な部分である。とくに `Ubuntu` のシステムを運用するときは配布元が重要であり、それによって利用者の個人情報保護される仕組みが保証されている。今回の `jitsi` の配布元も、`Ubuntu` のシステムが保証するレポジトリではないため、まず `Ubuntu` のサーバーに信用できるサーバーであることを通知してからインストールが開始される。サーバーを運用するということはみ

だりに不明のソフトウェアをインストールすることは厳に慎む必要がある。以下にソフトウェアをダウンロードする元が正当な開発元であることをサーバに設定するための手順を記す。

ソースコード 2: Jitsi のソフトウェアのインストール先の健全性の保証

```
1 sudo wget -qO - 'https://download.jitsi.org/jitsi-key.gpg.key' | apt-key add -
2 sudo echo 'deb https://download.jitsi.org/stable/' >> /etc/apt/sources.list.d/jitsi-stable.list
```

インストール後、`top` コマンドで確認すると、サービスを行っている `jicofo` と `jitsi-videobridge2` は `java` で動作し、`prosody` は `lua` というプログラムが実際に動作している。

4. まとめと今後の課題

新型コロナウイルスの蔓延という事態で、大学への出勤も難しいという事態となり、リモートワーク、遠隔授業などの検討が進んでいる。このような事態の中で、素粒子実験で行われてきた TV 会議システムの利用がどのようにできるかを考察した。偶然検索でヒットした Jitsi について調べるうちに、インストール手順が非常に簡単であるようにみえた。実際にやってみると現状ではサーバーのドメイン名の取得に始まり、情報センターの協力なしにはできないものであることがわかった。通常なら難しい状況ではあったが、情報センターが丁寧に対応してくださり、経験も不足している筆者にもサービスを開始することができた。しかしながら利用によって他のサービスを圧迫するようなら、停止という事態もありうるということである。音声 が 100kbps 程度の帯域が必要であることを考えると、参加者が最初にビデオ信号を送った後は、ビデオを切り、発表者が画面共有を行うだけであれば一つの会議が帯域を使い切ること はなさそうである。会議がどの程度行われているかは `log` ファイルを見ていけばわかるわけであるが、帯域の使用状況をモニターするツールも必要であり、帯域使用状況、会議室利用におけるセキュリティ問題の監視などまだまだ管理技術を磨いていく必要があると思われる。

謝辞

年度当初の情報センターにとって繁忙期であるにも関わらず、五十嵐誠氏には情報学研究所へのサーバ証明書の必要手続きについて、指示文書の作成など面倒な作業をお願いした。筆者がこの手続きについても未経験であるにも関わらず、丁寧な対応をしていただいたことに感謝します。また、コロナウィルスの蔓延に大学が対応するために大変な時期に理学科の先生方には、運用のテストを行っていただきました。また情報センター長の多田知正先生にも、会議室のホスト登録の試験をしていただきました。皆様に感謝いたします。

参考文献

- 1) 富田眞治・藤井康雄編著『情報社会とコンピュータ』6 章情報の検索・探索, 2005, 61-73 頁
- 2) ヨハネス・ブーフマン著 (林芳樹訳)『暗号理論入門』シュプリンガー・フェアラー東京, 2001
- 3) Jitsi Meet - ‘ More secure, more flexible, and completely free video conferencing ’ <https://jitsi.org/jitsi-meet/>

