令和 6 年度

修士学位論文

SAS-L を用いた動画像暗号通信方式の研究

A Study on Secure Video Communication Methods Using SAS-L

1275105　　高原 成功

指導教員　　清水 明宏

2025 年 2 月 28 日

高知工科大学大学院 工学研究科 基盤工学専攻

情報学コース

要 旨

SAS-L を用いた動画像暗号通信方式の研究

高原 成功

近年，IoT 機器の普及により，あらゆる情報がインターネット上で取り扱われている．IoT 機器の中でも，身近な IoT 機器として Web カメラが挙げられ，Web カメラは防犯用監視カメラとして用いられている．Web カメラは，インターネットを介して操作することができるため利便性は高いが，その一方で，映像に映る利用者の行動や個人情報の保護が重要な課題となっている．これを実現するためには，暗号通信が必要不可欠であるが，処理能力の低い IoT 機器では，十分なセキュリティ対策が施せない場合がある．

本研究では，処理能力の低い Web カメラでも，安全に動画像暗号通信を行えるような方式を提案する．加えて，プライバシ保護の観点から，動画像の表示方法についても検討を行う．

キーワード　IoT 機器，ワンタイムパスワード，SAS-L，暗号化，JPEG

Abstract

A Study on Secure Video Communication Methods Using SAS-L

TAKAHARA Nao

In recent years, with the spread of IoT devices, all kinds of information are handled over the Internet. Webcams are used as security surveillance cameras. Webcams are highly convenient because they can be operated via the Internet, but on the other hand, the protection of the user's behavior and personal information in the video images is a concern. On the other hand, the protection of users' behavior and personal information in the images is an important issue. To achieve this, cryptographic communication is essential. However, IoT devices with low processing power may not be able to implement sufficient security measures.

In this study, we propose a method for secure encrypted video communication even with a low-capacity webcam. In addition, from the viewpoint of privacy protection, we also consider how to display video images.

key words IoT Devices, one-time password, SAS-L, encryption, JPEG

目次

目次

図目次

表目次

第 1 章

はじめに

1.1 背景

近年，IoT（Internet of Things）時代の到来により，インターネット上に保存される個人情報が増大しており，様々なモノがインターネット上でやり取りされるようになっている [1]．IoT 機器のひとつとして，身近なモノでは Web カメラが挙げられ，Web カメラは公共施設の防犯用監視カメラや医療・産業分野などで活用されている．監視カメラの映像は，事故や事件の記録，または証拠として欠かせないものとなっており，事故原因の特定や，事件の解決に役立っている [2]．さらに，このような監視カメラの普及に加えて，IoT 機器である Web カメラの導入が進むことで，監視システムの可能性が広がっている．従来の監視カメラに比べて Web カメラはインターネットに接続されているため，遠隔からリアルタイムで映像を確認できるという利便性がある．しかし，その一方で，監視カメラは，事故や事件に無関係な人物の記録や撮影画像の流出により，プライバシを侵害してしまう問題も抱えている [3]．そのため，人物のプライバシ保護についても検討が必要である．加えて，個人情報の保護やインターネット上での通信の暗号化といったセキュリティ対策も重要な課題となっている．

IoT 機器は低価格化や小型化が進み，省電力化を目指す研究が行われており，処理能力が低くても動作する IoT 機器の需要が高まっている [4]．しかし，このような IoT 機器では，処理能力やメモリの制約により，十分なセキュリティ対策が施せない可能性がある．特に，カメラから撮影される動画像はデータサイズも大きいため，通信時に暗号処理に時間がかかるといった問題が懸念される．また，暗号鍵の管理の安全性についても検討が必要である．

したがって，IoT 機器の処理能力に関わらず，暗号化といったセキュリティ対策を実現できる仕組みが必要である．

本研究では，処理能力の低い Web カメラでも，安全な動画像暗号通信を行えるよう，極めて低負荷なワンタイムパスワード認証方式である SAS-L を暗号鍵の更新に用いた方式を提案し，評価を行う．また，プライバシ保護のため，動画像の特定領域の暗号化や表示方法についても検討を行う．

第 2 章

従来方式

2.1　事前共有鍵による暗号化

事前共有鍵方式（Pre-shared Key, PSK）は，通信を行う双方が事前に共有した対称鍵を用いて暗号化および復号を行う方式である．概要を図 2.1 に示す．本方式では，通信開始前に安全な経路を用いて暗号鍵を共有し，以降の通信を AES（Advanced Encryption Standard）等の安全な暗号方式を用いて動画像を保護する [5][6]．しかし，事前共有鍵方式では，暗号鍵の更新が困難であるため，一度鍵が漏洩すると全ての通信が解読される危険性がある．

関連研究として，暗号アルゴリズムに AES を用いて，動画像をリアルタイムに処理するものが提案されている [7]．この方式では，送信している動画像の全フレームを一つの鍵で暗号化し，通信を行っている．全フレームを同一鍵で暗号化することで，処理負荷を減らすことができるため，処理能力が限られた機器にも実装可能であるが，その一方で，同一の鍵を継続的に使用することによるセキュリティ上の問題が指摘されている．そのため，鍵更新が必要となるが．多数の IoT 機器に対する更新作業は煩雑となる．

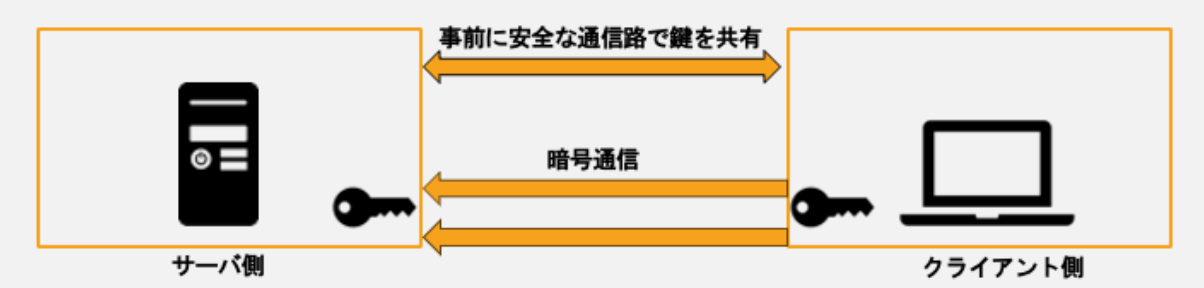

図 2.1　事前共有鍵方式

2.2　セッション鍵による暗号化

事前共有鍵方式の課題を解決する手法として，鍵交換プロトコルを利用したセッション鍵方式がある．セッション毎に異なる暗号鍵を生成し，その鍵を用いて通信データを暗号化および復号する方式である．概要を図 2.2 に示す．

RSA や DH 法といった公開鍵暗号方式を応用した鍵交換プロトコルを利用して，セッション鍵を安全に交換した後，AES などの共通鍵暗号方式で暗号通信を行う．安全性が向上した一方で，処理負荷の大きい鍵交換プロトコルが利用されるため，リソースの限られた IoT 機器では，実装が困難であったり，リアルタイム性が損なわれるといった課題が発生する場合がある．現在，鍵の交換方式で主流となっている DH 法について，下記に記す．

図 2.2　セッション鍵方式

2.2.1　Diffie-Hellman 鍵交換 (DH 法)

Diffie-Hellman（DH）鍵交換は，盗聴の恐れのある通信路上においても，安全に共通鍵を生成するためのプロトコルである．概要を図 2.3 に示す．DH 法は「離散対数問題の難しさ」に基づいており，非常に大きな素数を用いた場合，攻撃者が共通鍵を算出することは，現実的な時間では極めて困難であることを根拠として，安全性が担保されている [8]．DH 法の鍵交換手順について，以下に示す．

図 2.3　DH 方

2.2.2　DH 法の鍵交換手順

サーバとクライアント間で鍵交換をする状況を考える．まず，サーバとクライアントは共に，十分に大きな素数 p と，その原始根 g を事前に決定し，これらの値を公開する．

次に，サーバとクライアントは個別に秘密鍵を選び，それに基づいて公開鍵を生成する．サーバは秘密鍵 a をランダムに選び，公開鍵 A を式 (2.1) により計算する．

$$A = g^a \bmod p \tag{2.1}$$

同様に，クライアントは秘密鍵 b をランダムに選び，公開鍵 B を式 (2.2) により計算する．

$$B = g^b \bmod p \tag{2.2}$$

サーバは自分の公開鍵 A を クライアントに送り，クライアントは自分の公開鍵 B をサーバに送る．この交換後，サーバと クライアントは，それぞれ受け取った公開鍵を用いて共通鍵を計算する．サーバは式 (2.3)，クライアントは式 (2.4) により共通鍵を求める．

$$K = B^a \bmod p \tag{2.3}$$

$$K = A^b \bmod p \tag{2.4}$$

サーバ，クライアントが生成した共通鍵は，同一の値を取るため，暗号化通信に利用できる．しかし，大きな素数の生成等，処理負荷が大きいため，リソースの限られた IoT 機器では実装が困難である．

したがって，処理能力の低い IoT 機器でも，高速かつ安全な鍵交換を行える方式が求められる．

第 3 章

提案方式

従来方式の問題点を解決する方式として，SAS-L2 を鍵交換に用いた動画像暗号化システムを提案する．提案システムの概要を図 3.1 に示す．

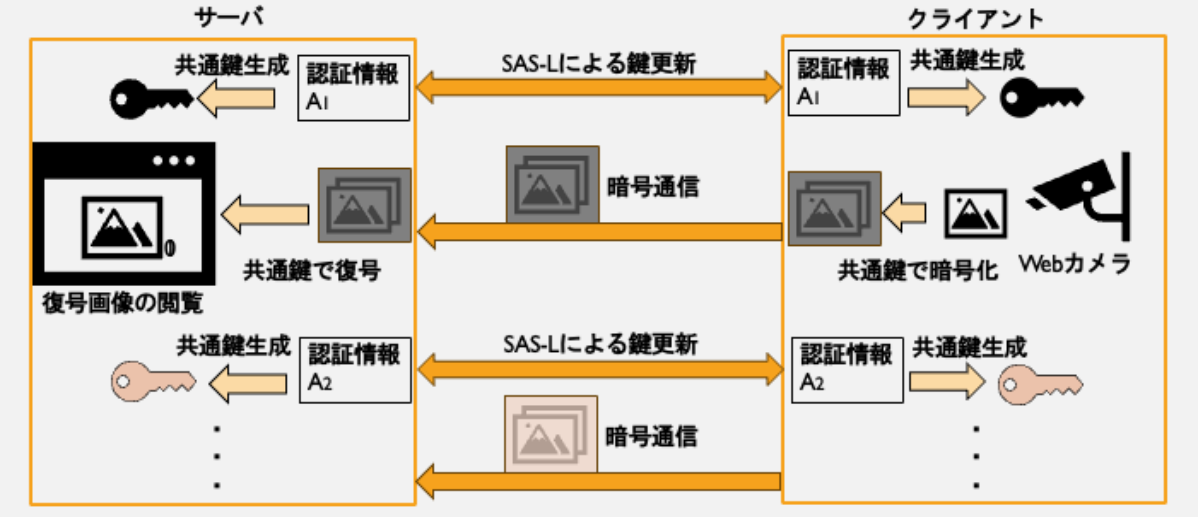

図 3.1　SAS-L カメラシステム

3.1　SAS-L ワンタイムパスワード認証方式

SAS-L(Simple And Secure password authentication protocol, Light-processing version) はサーバ・クライアントのいずれかでハッシュ関数の適用を必要としない，処理負荷が限りなく 0 に近いワンタイムパスワード認証方式である [9][10][11]．SAS-L には，上位レイヤの認証に適した SAS-L1 と，特に下位レイヤにおいて処理能力の小さいデバイスへ暗号通信機能のための鍵配送の機能を付加できる SAS-L2 のふたつのバージョンがある [9]．提

案方式では，処理能力の小さな IoT デバイスを想定してることから，SAS-L2 を採用した．以下に，その処理フローを示す．また，SAS-L は初回に認証情報を共有する登録フェーズと，初回以降行われる認証フェーズに分かれており，各フェーズ毎の処理フローを示す．

3.1.1　SAS-L2

この方式は，被認証者であるクライアントが，秘密に保持しているパスワード S を認証者であるサーバに開示しない従来の方式と異なり，認証者であるサーバが秘密情報である S をクライアント識別子として管理する．以下の説明において用いる記号および演算を表 3.1 に示す．

表 3.1　記号説明

記号	説明
S	被認証側の識別子
A_i	認証が i 回目の時の認証情報
H	ハッシュ関数
i	認証回数を表す 1 以上の整数
N_i	i 回目に生成された乱数
$=$	代入を表す
A = B?	A と B が一致するか比較
#	任意の演算を表す
$\oplus$	ビット単位の排他的論理和演算を表す

まず，サーバは，認証したいクライアントにあらかじめ初回認証情報と初回秘匿情報の登録を行っておく．クライアントの初期登録処理を，図 3.2 に示す．

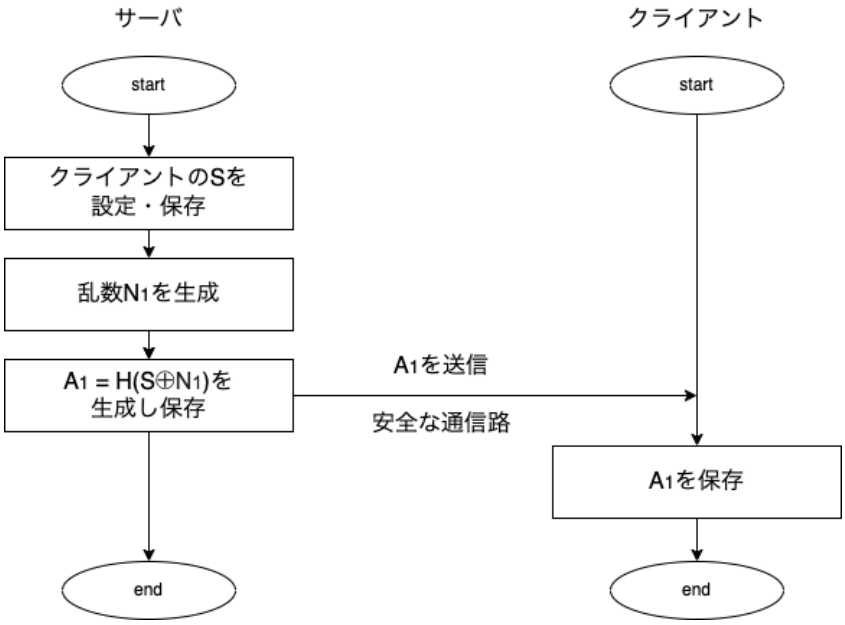

図 3.2　SAS-L2 初期登録処理フロー

登録フェーズ

サーバは，あらかじめクライアント識別子 S を設定・保持している．はじめに，サーバは，乱数 N_1 を生成し，保存する．その後，乱数 N_1 と設定し保持しているクライアント識別子 S を用いて，初回認証情報である A_1 を式 (3.1) により生成する．さらに，安全な手段でクライアントに送付する．初回認証情報 A_1 は，初回 (i=1) の認証に用いる認証情報である．

$$A_1 = H(S \oplus N_1) \tag{3.1}$$

認証フェーズ

初回 (i=1) 以降，i 回目の認証時の認証処理について，図 3.3 により説明する．このとき，サーバは S を保存している．また，クライアントは A_i を保存している．

まず，新しい乱数 N_{i+1} を生成する．その後，保存しているクライアント識別子 S と N_{i+1} を用いて，式 3.2 に示す A_{i+1} を生成する．

3.1 SAS-L ワンタイムパスワード認証方式

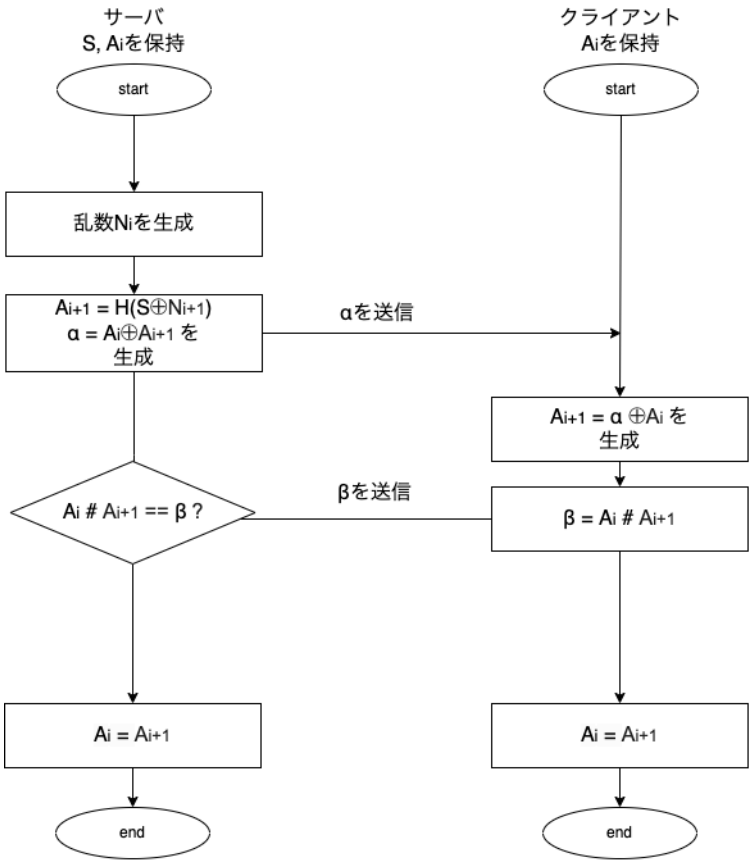

図 3.3　SAS-L2 の i 回目認証時の認証フロー

$$A_{i+1} = H(S \oplus N_{i+1}) \tag{3.2}$$

そして，A_i と A_{i+1} を用いて式 3.3 に示す α を算出する．

$$\alpha = A_i \oplus A_{i+1} \tag{3.3}$$

算出した α をクライアントへ送付する．このとき，A_i は現在の認証情報，A_{i+1} は次回用の認証情報である．クライアントは，サーバから受け取った α を，式 3.4 に示すように，保存している今回認証情報 A_i を用いて，A_{i+1} を算出する．

$$A_{i+1} = \alpha \oplus A_i \tag{3.4}$$

続いて，算出した A_{i+1} と保存している現在の認証情報 A_i を用いて検証子である β を式 3.5 によって算出し，その算出結果をサーバへ返送する．

$$\beta = A_i \# \ A_{i+1} \tag{3.5}$$

サーバは，クライアントから検証子 β を受け取ると，保持している A_i と A_{i+1} から算出できる $A_i \ \# \ A_{i+1}$ が β と一致するかどうかを比較する．両者が一致する場合，クライアントの認証が成立し，次回 $(i+1$ 回目$)$ の認証に用いる認証情報として A_{i+1} を保存する．両者が一致しない場合は，認証不成立となり，その旨を伝えるメッセージを送付する等を行い，処理を終了する．

3.2　動画像の暗号化

3.2.1　JPEG 圧縮

JPEG（Joint Photographic Experts Group）は，画像データを効率的に圧縮するための標準的な圧縮方式である．主に離散コサイン変換（DCT），量子化，ハフマン符号化を組み合わせて，データ量を大幅に削減する [12][13]．JPEG 圧縮は，画像の視覚的品質を保ちながらデータ量を削減できるため，動画像の効率的な伝送に適している．特に，IoT 機器やネットワーク帯域が限られた環境では，JPEG 圧縮によるデータ量削減が通信負荷の軽減に寄与する．提案方式では，まず動画像をフレーム単位で JPEG 圧縮し，暗号化処理の対象とすることで効率的な通信を実現する．

3.2.2　バーナム暗号

バーナム暗号（以下，XOR 暗号）は，一度限りのパッド（OTP: One-Time Pad）を用いた暗号化方式であり，シャノンにより理論的には完全な安全性を持つと証明されている [14]．処理の概要図を図 3.4 に示す．暗号化の際，平文と同じ長さ，もしくはそれ以上であるランダムな鍵をビットごとに排他的論理和（XOR）演算で組み合わせることで暗号文を生成す

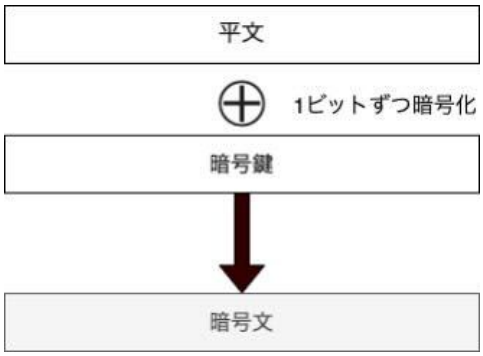

図 3.4　XOR 暗号の処理手順

る．復号時には，同じ鍵を再度使用して XOR 演算を行うことで平文を復元する．XOR 暗号の特徴として，鍵が真の乱数で一度しか使用されず，平文と同じ長さである場合，暗号文は完全に解読不可能となる．この条件を満たしていれば，暗号の強度は情報理論的安全性を持つ．提案方式では，動画像を暗号化する際に，フレーム単位で SAS-L2 で鍵を交換し，XOR 暗号を適応することで，安全かつ高速な暗号通信を実現する．

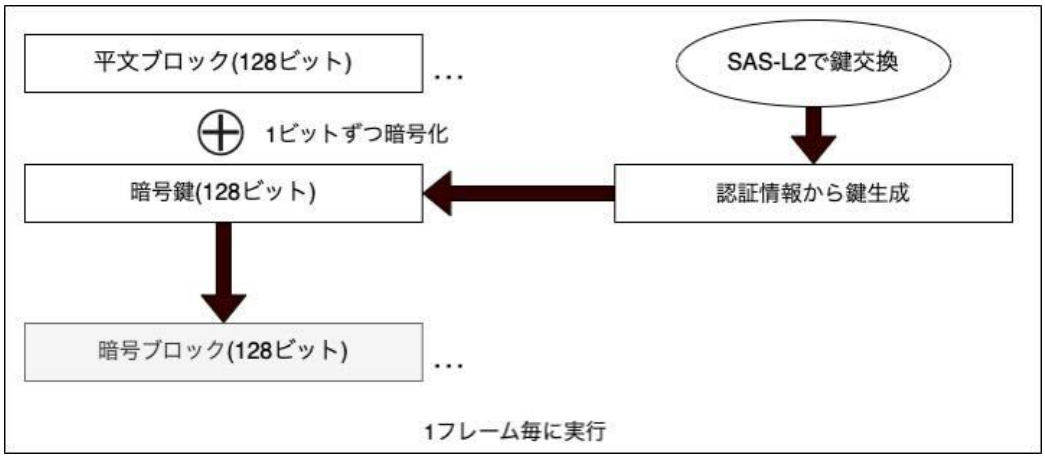

図 3.5　提案方式におけるフレーム暗号化

提案方式における，フレームの暗号化手順について，概要を図 3.5 に示す．提案方式では，サーバとクライアント間で SAS-L2 を利用して鍵交換を行う．その際，認証情報である A_i は外部から秘匿され，推測ができない情報である．この情報を用いて，ハッシュ関数や，SAS-L2 で共有する秘匿情報 (マスク値) を追加し，XOR 演算等を行い 128 ビットの暗号鍵を生成する．暗号鍵とフレームデータを 128 ビット毎に分割し，それぞれ XOR 暗号を適用

することで，フレームを暗号化する．このように，フレーム毎に異なる鍵で暗号化を行うことができる．

第 4 章

評価

4.1 開発環境

ここでは提案方式の評価を行う．評価に際しては，学内の無線 LAN 環境下で実験を行った．実験の概要を図 4.1 に示す．開発環境を表 4.1 に示す．

サーバには MacBook Air を，クライアントには，プログラミングが可能かつ，カメラ機能を内蔵している IoT 機器である M5Stack UnitV2 を採用した．

これら端末を用いて，下記の組み合わせを用意し，提案方式の速度評価を行った．

- AES 暗号化 (事前共有鍵方式) 鍵交換なし
- 提案方式 (SAS-L2 & XOR 暗号化) 1 フレーム毎に鍵更新
- AES 暗号化 (SAS-L2) 1 フレーム毎に鍵更新

表 4.1　開発環境

	サーバ	クライアント
利用端末	MacBook Air	M5Stack UnitV2
CPU	Apple M1 チップ	Arm Cortex-A7 デュアルコア 1.2 GHz
メモリ	8GB	128MB DDR3
OS	Sonoma14.3.1	Buildroot 2020.02.8
開発言語	Python3.13.0	Python3.8.6

図 4.1　速度評価実験概要

4.2　速度比較

表 4.2 は，AES の同一鍵を用いる方式と，AES，XOR に対してそれぞれ鍵交換方式に SAS-L2 を用いて，鍵の更新を行った際の fps 計測結果（ 画像を 10000 フレーム送信した際のもの）である．同一鍵を使い続ける方式では 15.32fps，SAS-L2 で鍵の更新を行った AES 暗号化では 11.03fps となった．100 枚の画像を送信している場合，同一鍵では 1 つの鍵が特定できれば 100 枚全てが解読できるのに対し，SAS-L2 を用いた方式では 100 枚の画像を解読するには 100 個の鍵を特定しなければならない．同一鍵を用いる方式よりも鍵の更新を行っている分，低速になったが，速度減衰を 28 %に抑え，鍵交換を行う動画像暗号通信が可能である．

表 4.2　fps 比較

方式	[fps]
AES 暗号化（同一鍵）	15.32
AES 暗号化と SAS-L2 鍵更新	11.03
XOR 暗号化と SAS-L2 鍵更新	13.42

4.2.1　AES 暗号ライブラリを使用しない実装

前述した 4.2 節の実装では，AES 暗号化を使用する際，ライブラリを呼び出して暗号化を行っている．暗号ライブラリを使用した AES の暗号化では，処理のほとんどを OpenSSL に委託しているため，暗号ライブラリを使用していない XOR と同条件で比較することがで

きない．そのため，AES 暗号ライブラリを使用しない実装を行った．実験結果は表 4.3 の通りである．AES 暗号ライブラリを使用しない実装では，0.080fps となった．

表 4.3　AES 暗号ライブラリ無し

方式	[fps]
AES 暗号化と SAS-L2 鍵更新	0.080
XOR 暗号化と SAS-L2 鍵更新	13.42

AES 暗号化で記述した内部処理は以下の通りであり，それぞれを個別の関数としてコード上に記述することで，ライブラリに依存しない形で実装した．

- **shift_rows()**

  行シフトを行う．行シフトを行うことで，列単位の依存関係を壊し，拡散性を高める．
- **sub_bytes()**

  データを S-BOX（置換表）を使用して変換することで暗号強度を高める．
- **mix_columns()**

  各列を線形変換し，拡散性を高める．最後のラウンドでは省略される．
- **add_round_key()**

  各ラウンドの鍵を XOR で演算し，データの秘匿性を確保する．
- **key_schedule()**

  暗号化・復号に用いるラウンド鍵を生成する．

AES 暗号化では，暗号ライブラリを用いることで，より高速な実装を行うことができるが，ライブラリを使用するためにはモジュールをインポートする必要がある．暗号ライブラリを使用すると，Web カメラの処理能力が低い場合や，ライブラリに対応していない機器の場合は，実装が難しくなるのに対し，XOR 暗号化では，単純な論理演算のみで行っているため，処理能力の低い機器にも実装可能であると言える．

4.3　CPU 使用率の比較

表 4.4 は，AES 暗号化と XOR 暗号化を行った際の CPU の平均使用率である．それぞれ SAS-L2 を用いて鍵交換を行っている．AES 暗号化の CPU 平均使用率 48.83%に比べて XOR 暗号化の CPU 平均使用率は 43.17%となっており，約 5%削減されていることが確認できる．

表 4.4　CPU 使用率比較

方式	CPU 平均使用率 [%]
AES 暗号化と SAS-L2 鍵更新	48.83
XOR 暗号化と SAS-L2 鍵更新	43.17

4.4　鍵交換方式の比較

表 4.5 は，動画像を XOR 演算で暗号化し，鍵交換方式に SAS-L2 と DH 法を使用したときの fps と，鍵交換 1 回にかかる平均時間を計測したものである．DH 法は 1024 ビットの素数を生成し鍵交換を行っている．SAS-L2 では，鍵交換の時間が 0.014s なのに対し，DH 法では 5.16s となっており，約 368 倍の差が出ていることが確認できる．また，DH 法を使用した時の fps は，0.02fps という結果になった．DH 法を用いる場合には，1 フレームごとの鍵交換は行わず，セッションごとでの暗号化通信に適していると言える．日本防犯設備協会の防犯カメラ撮影機能では，最低でも 2〜3fps で記録できる性能が条件とされており，リアルタイム性の観点から見ても，SAS-L2 を用いた動画像通信は有用であると言える [15].

表 4.5　鍵交換方式の性能比較

方式	[fps]	鍵交換の平均時間 [s]
XOR 暗号化と SAS-L2 鍵更新	13.42	0.014
XOR 暗号化と DH 鍵更新	0.02	5.16

4.5　部分的暗号化

M5Stack UnitV2 のカメラから撮影されたフレームを図 4.2 に示す．提案方式を応用し，図 4.2 の特定のピクセル領域に対して部分的暗号化を行ったものが図 4.3 である．

4.2 節，4.4 節と同様に，SAS-L2 を用いて鍵交換を行っており，JPEG 圧縮されたフレームデータ内のピクセル範囲を指定し，XOR で暗号化を行った．XOR 暗号化と SAS-L 鍵更新を適用することで，暗号化にかかる負荷を減らすことができ，加工処理を実現することが可能である．暗号化されたピクセル値は，元画像にランダムなノイズを与え，さらに対象領域のピクセルブロックを平均化することで，視覚的な加工を行っている．これにより，カメラに映っている対象人物の特定が困難になり，対象の動きや行動については今まで通り監視することが可能である．したがって，提案方式では，行動を監視すると同時に，プライバシの保護を行うことが可能であると言える．

図 4.2　標準出力

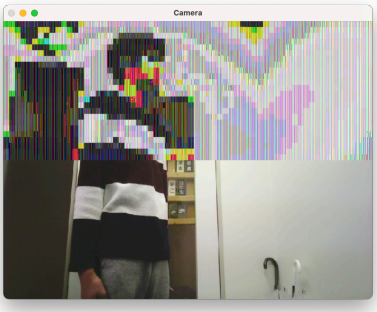

図 4.3　特定領域の暗号化

第 5 章

むすび

本研究では，処理能力の低い IoT 機器でも適切なセキュリティ対策が行えるよう，被認証側である Web カメラで一方向性変換の適用を必要としない SAS-L を鍵交換に用いた方式を提案した．従来の方式では，リソースの制約により同一鍵を用いて暗号通信が行われていた．しかし，SAS-L を用いることで，極めて低負荷な処理で鍵交換を行うことができるため，処理能力の低い機器でも，1 フレームごとに鍵の更新を行う動画像通信方式を実現できた．提案方式では，従来方式と比較して通信速度の低下は見られるものの，速度減衰を約 28 %に抑え，動画像の暗号通信を行うことが可能である．プログラム実行時の CPU 使用率においても，従来方式よりも提案方式の方が約 5%低いため，低負荷な実装が可能と言える．また，プライバシ保護の観点から，画像フレームの特定領域を部分的に暗号化することで，監視の有効性を維持しながらプライバシ保護を両立できる可能性を示した．

今後の課題として，複数端末に実装した場合の通信量増加による遅延の影響や，SAS-L における同期ずれ対策，また，移動物体の暗号化を動的に行う手法の検討が挙げられる．本研究の実装では，同一の LAN 内でそれぞれの方式を実行し，fps 比較を行っている．そのため，ネットワーク環境の影響が限定的な条件下での性能評価が中心となっている．しかし，実際の運用を想定した場合，インターネットを介した通信が必要となり，その際のネットワーク遅延の影響やパケットロス等が動画像通信のパフォーマンスに与える影響を無視することはできない．特に，通信距離の増加やネットワークの混雑状況によっては，fps の低下や，リアルタイム性を損なってしまう可能性がある．また，ネットワーク環境や通信手段の違いも考慮する必要があり，その影響を評価するためには異なるネットワーク下での実験が不可欠である．さらに，本研究の実装では開発言語に Python を用いているが，実用化を

見据えた際には，処理速度の向上を目的として C 言語等による実装で高速化を図る必要がある．

したがって，今後はより実際の実装環境に近い条件下で評価を行い，ネットワーク遅延が fps に与える影響の分析や，安定したパフォーマンスを実現するための検討が必要である．

謝辞

本研究の遂行及び論文の執筆にあたり，多くのご指導を頂きました高知工科大学情報学群 清水明宏教授に心より感謝し厚く御礼申し上げます．また，本研究の副査を担当していただいた高知工科大学情報学群 敷田幹文教授，横山和俊教授に深く御礼申し上げます．

最後に，多くの御助言を頂きました高知工科大学情報学群 セキュリティシステム研究室の関係者各位に深く感謝いたします．

参考文献

[1] M.ofInternal Affairs and J.Communications, ”White paper informa-tion and communications in japan 2021”, https://www.soumu.go.jp/johotsusintokei/whitepaper/r03.html, 2021. Accessed:2025-01-14.

[2] 日本建設情報総合センター,“監視カメラ等の普及と発展,”https://www.jacic.or.jp/books/jacicnews/pdf/jn357.pdf, 2025. Accessed: 2025-01-30.

[3] 藪田顕一，北澤仁志，”真正性証明とプライバシ保護を両立する監視カメラシステム”, 画像電子学会誌 第 38 巻 第 5 号（2009）.

[4] 大西篤史，黒田久泰,“省電力化を目的とした防犯カメラシステムの構築,”情報処理学会 , 第 73 回全国大会講演論文集 , pp523 - 524,2011.

[5] Federal Information Processing Standard Publication 197: Announcing the ADVANCED ENCRYPTION STANDARD(AES), (2001.11.26).

[6] CRYPTREC 軽量暗号ワーキンググループ,“CRYPTREC 暗号技術ガイドライン,”https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016jp.pdf, 2017. Accessed: 2025-01-14.

[7] 楊淵, 上竹嘉紀, 小林航也, 日下卓也,& 野上保之，Raspberry Pi を用いたカメラ動画像の AES 暗号化に対するリアルタイム処理性能評価，コンピュータセキュリティシンポジウム 2019 論文集,2019,812-816.

[8] Diffie, W. and Hellman, M.E.,”New directions in cryptography”, IEEE Trans. On Information Theory, Vol.IT-22, No.6, pp.644-654, 1976.

[9] K. MIZOGUCHI and A. SHIMIZU, ”A One-Time Password Authentication Scheme for IoT Environments, ” IEICE TECHNICAL Report, vol. 124, no. 257, LOIS2024-43, pp. 112–117, 2024.

[10] T. Hinata and S. Akihiro,“A VPN Method for Security Usage of Pub- lic Wireless

LANs, ” IEICE Technical Report, LIS2023-41, pp.51–56, 2024.

[11] W. Stallings, “ Secure hash algorithm, ” in Cryptography and NetworkSecurity: Principles and Practice 2nd ed., pp.193–197,Prentice-Hall,1999.

[12] 半谷精一郎，杉山賢二 共著，”JPEG・MPEG 完全理解”，コロナ社，2005 年 9 月 22 日初版発行.

[13] Wallace, Gregory K. ”The JPEG Still Picture Compression Standard.” *Communications of the ACM*, Vol. 34, No. 4, pp. 30–44, April 1991.

[14] Claude E. Shannon, ”Communication Theory of Secrecy Systems,” *Bell System Technical Journal*, vol. 28, no. 4, pp. 656–715, 1949.

[15] 公益社団法人日本防犯設備協会，”金融機関等防犯カメラシステムの設計基準・解説”，`https://www.ssaj.or.jp/bouhan_kiki/kinyu.html`，2025/1/14.