Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

CFB モードにおける IV の誤使用と計算量の低下現象について

沖津直樹[1]

情報セキュリティ大学院大学[1]

概要：CFB モードを使用する際、IV は予測不可能とするよう NIST SP800-38A から要求が出されている。この要求に違反すると、確定的性質(Deterministic Nature）が原因で予想した暗号文に到達するまでの計算量が急減する。そのため、予想した暗号文の出力が現実的な時間内で可能となる。違反した状態のテストケースをいくつか想定し、その計算量などを算出した。そして、テストケースに基づき実験を実施したので、その結果を報告する。また、ストリーム内に現れる特定のバイト列（各バイトが全て同じ整数で構成されている）の連続出現回数に着目すれば計算量の定式化が可能であったため、その定式を報告する。防御方法として、NIST SP800-38A の要求事項や実験結果に基づいて、計算量が減らないための方法を提案する。

キーワード：AES CFB モード，初期化ベクトル、確定的性質，決定論的性質

Misuse of IV in CFB mode and Computational complexity reduction phenomenon

NAOKI OKITSU[†1]

Institute of information security[†1]

Abstract: When using CFB mode, NIST SP800-38A has requested that iv be unpredictable. Violation of this requirement will drastically reduce the amount of computation required to reach the expected ciphertext due to its Deterministic Nature. Therefore, the expected ciphertext can be output within a realistic time. We assumed some test cases in a violated state and calculated the amount of calculation. Then, since the experiment was carried out based on the test case, the result is reported. In addition, since it was possible to formulate the amount of calculation by paying attention to the number of consecutive occurrences of a specific byte string (each byte is composed of the same integer) appearing in the stream, the formula is reported. As a defense method, we propose a method to prevent the amount of calculation from being reduced based on the requirements of NIST SP800-38A and experimental results.

Keywords: AES CFB mode, Initialization Vectors, Deterministic Nature

1. はじめに

NIST SP800-38A[1]では IV について次の要求事項が記載されている。

① IV は秘密である必要は無い。② CBC モード、CFB モードでは予測不可能でなければいけない。③ OFB モードでは、予測不可能である必要はないが、IV は暗号化操作の度に一意なノンス（カウンター、メッセージ番号）でなければいけない。④ CBC モード、CFB モード、OFB モードの実装がこの推奨事項に準拠しているか検証が必要である。⑤ IV が予測不可能であるか、一意であるか保証する必要がある。⑥どのような手順で保証されているか確認が必要である。

2. 動機

次の理由から IV の誤使用についてもっと研究が必要と考えた。① NIST SP800-38A に IV を使用する際の注意事項が記載されているが、記載量が少なく、その記載に違反して実装した場合にどのような危険性があるか、理由などが書かれていない。もっと具体例を示し、危険性をアピールすべきである。② NIST SP800-38A の要求事項に違反して実装してしまい、脆弱性(CVE-2020-1472)として報告されている件が実際に起きている。原因は、実装誤りであるが、IV の誤使用について注意喚起をする文献がもっと多くあれば、防げた可能性もある。アプリケーションの開発者は暗号の専門家ではないため、開発者向けに危険性を具体的に示すべきである。

3. 関連研究

3.1 確定的性質についての関連研究

同じ平文に対して同じキーを使って暗号化をすると同じ暗号文ができるが、このことは確定的性質や決定論的性質などと呼ばれている。

ECB モード[1]では、IV は使用されておらず、同じ平文を入力すると同じ暗号文ができる。この特徴が望ましくないアプリケーションでは ECB モードは使用しないよう注意事項が記載されている。

確定的暗号は、秘匿検索などの分野で研究が行われている。クラウド上で暗号化したキーワードを暗号化したままで検索できることを目指した研究で確定的暗号は使われている。[2][3][4]

確定的暗号の高速性と確率的暗号の安全性をきちんと区別して、その利点を組み合わせたハイブリッド方式なども提案されている[2]。特定の目的、用途で確定的暗号は使われ

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

ている。

3.2 IV の誤使用についての関連研究

RFC5297[5]ではノンスに基づいて認証を行う仕組みの場合、ノンスの再利用や誤用は機密性を保証できないとの記載がある。また、SIV(Synthetic Initialization Vector)はノンスの再利用や誤用に一定の耐性があるとの記述がある。

[6]では、C、C++、Java、Python、Go のライブラリを調査し脆弱性を作り込んでしまう原因を調査している。ライブラリ自体には問題はないが、開発者の誤使用が原因との指摘がされている。誤使用を減らすためにライブラリの設計者向けに推奨事項が示されている。

3.3 IV の誤使用が原因で発生した脆弱性についての報告

[7]は研究論文という形式ではなくホワイトペーパーである。CVE-2020-1472 に関しての報告である。本稿のテストケース 26 に該当するテストケースについての記載がある。攻撃が成功する他のテストケースについての記述が無く、防御方法が書かれていない。ストリーム中で出力される暗号文の 1 バイト目が 0 以上 255 以下の整数をとり、出る確率が 256 分の 1 になるとの記述がある。しかし、計算量の低下が起きてしまう原因までは突き止められていない。

3.4 シミュレーションを行う上で参考にしたプログラム

[8]を参考にシミュレーションプログラムを開発した。[8]は計算量まで書かれていないが、バイトをずらして攻撃が成功するかを試す発想がコード内に記載されている。

3.5 IV を誤使用している実例

[9]は、公開されている技術文書である。この文書内に IV をオールゼロで設定している記載がある。p107 3.3.1.4.4.1 AES Credential の箇所に SET IV=0 の記載がある。

4. 本稿のオリジナル部分

以下の点がオリジナル部分である。
①計算量が減少する原因が確定的性質であることを突き止めたこと、②実験計画の作成、③実験による実測、④机上計算や実測から得られた結果をもとに定式化した定式
⑤ストリーム内の処理を詳細化し計算量を机上計算したこと、⑥実験に使用したシミュレーションプログラム

5. 実験

5.1 実験概要

次の動きを模したシミュレーションプログラムを作成し、攻撃が成功するかを確認する。
①攻撃者から送られてきた平文を使って、攻撃対象側で暗号文を作成する。
②その作成した暗号文が攻撃者から送られてきた暗号と一致するかを確認する。
③一致したら攻撃成功とする。

5.2 シミュレーションプログラムの制約事項

python の pycryptodome のパッケージ、モジュールを使用してシミュレーションプログラムを作成した。シミュレーションプログラムには pycryptodome のモジュールの制約で以下の制約事項がある。

制約事項
① IV は 16 バイトである。そのため AES128 のみが確認できている。AES192 や AES256 は確認できていない。
②モード指定は無し。キーストリームを作る際、AES.new(key, AES.MODE_CFB, iv)の指定しかできない。CFB1、CFB8、CFB64、CFB128 のような指定はできない。
③入力平文は 8 バイトとする。
④出力される暗号文も 8 バイトとする。
⑤ 1 バイト、8 ビットなので、0 から 255 までの整数が表現可能

5.3 実験シナリオ

攻撃者と攻撃対象を次の通り想定する。この実験シナリオは、[7][9]を参考に作成した。

攻撃者	①ある正当な対象になりすまそうとしているが、正当な対象と攻撃対象との間の共有秘密情報（パスワードなど）は知らないものとする。 ②正当な対象の一部の情報（IP アドレスなど）は知っていて、それをなりすましに使用する。 ③ IV の値を知っている
攻撃対象	① SP800-38A の IV に関する要求（予測不可能）に違反して IV を設定している。 ② IV の値を公開している。

攻撃が成功した場合、パスワードを知らなくても任意の対象になりすますことができたことになり、次の仮定が崩れることとなる。

攻撃成功で崩れる仮定	①キーを知らない攻撃者は認証が通る暗号文を計算、推測できない。 ②キーを知っていれば共有秘密情報（パスワードなど）も知っている、だから正当な対象と考えて良い。

5.4 実験計画

SP800-38A の要求事項（予測不可能）に違反した状態の IV を想定する。以下の 4 パターンでテストケースを作成する。

	IV	攻撃者が送信する平文
ケース 1	1 バイトづつ変更する	固定
ケース 2	固定	1 バイトづつ変更する
ケース 3	固定	固定
ケース 4	可変	可変

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

それぞれのケースについて、以下の項目を机上で算出する
① 必要な試行回数（計算量）
② 総当たり攻撃に必要な計算時間
③ 出力予想できる暗号文

それぞれのテストケースについて、シミレーションプログラムを使って以下の項目を実測する
① 実際に出力された暗号文
② 攻撃の成否
③ 実際の試行回数
④ 暗号文出力までの PC での実行時間（秒）
⑤ ストリーム中で使用されたキー

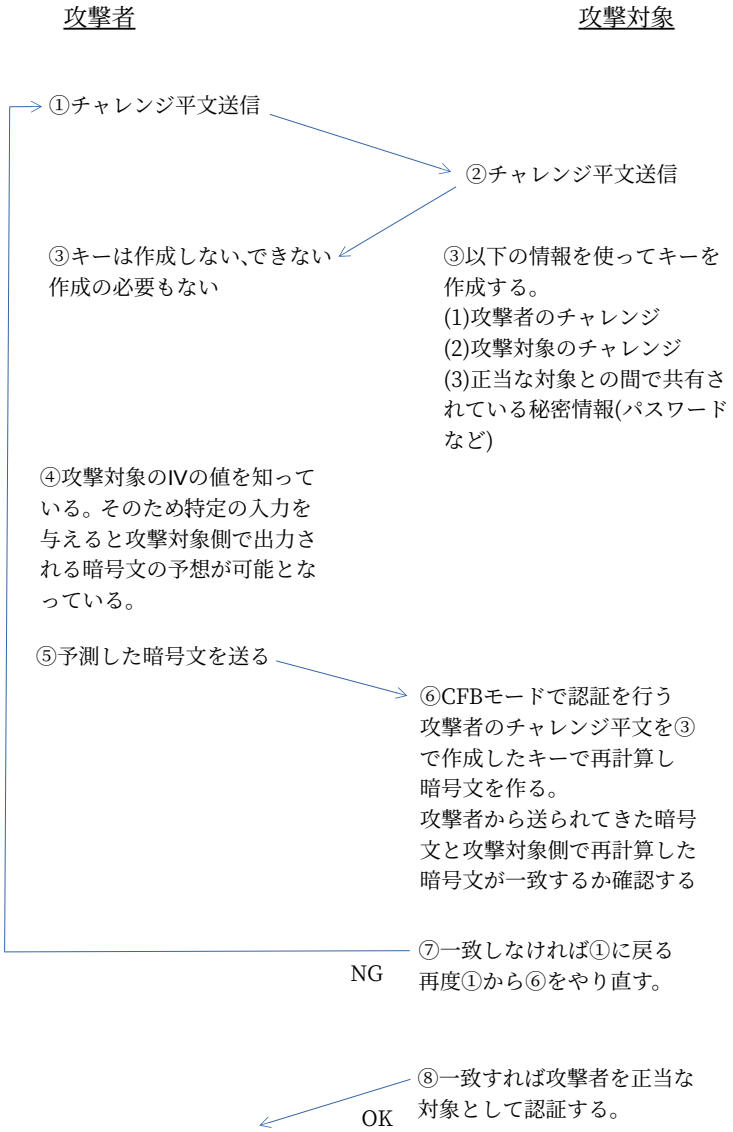

図 1　実験シナリオ

5.5 実験計画時の計算量の算出

計算量を算出するにあたり、ストリーム内での処理を詳細化した（図 2、図 3）。

5.6 実験結果

表 1-2、表 2-2、表 3-2、表 4-2 に示したように計算量が 256^4 以下のケースについては全て攻撃が成功した。出力予想暗号文と実際に出力された暗号文も一致した。

試行回数と攻撃時間については、実験計画と実際の実験では差があるが、実験は 1 回のみしか行っていないためであると思われる。実験回数を増やせば実験計画の値に収束すると思われる。今回の実験では実験回数を増やして収束するかまでは確認していない。

6. 計算量の定式

ストリーム内に現れる特定のバイト列の連続出現回数に着目すれば、計算量の定式化は可能である。定式は以下のとおりである。

y=256^(9-x)
y:計算量
x:ストリーム内に現れる特定のバイト列の連続出現回数

特定のバイト列とは、AES への入力に使われ、各バイトが全て同じ整数で構成されているバイト列のことである。本実験では AES への入力は 16 バイトであるため、以下のようなバイト列のことを指す。

例）特定のバイト列
0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
・・
7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7
・・
a,a,a,a,a,a,a,a,a,a,a,a,a,a,a,a
0<=a<=255 までの整数
・・
99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99
・・
255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255

x	y	AES への入力
1	256^8	全て異なるバイト列が入力される
2	256^7	2 回連続して同じバイト列が入力される
3	256^6	3 回連続して同じバイト列が入力される
4	256^5	4 回連続して同じバイト列が入力される
5	256^4	5 回連続して同じバイト列が入力される
6	256^3	6 回連続して同じバイト列が入力される
7	256^2	7 回連続して同じバイト列が入力される
8	256^1	8 回連続して同じバイト列が入力される

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

7. 防御方法

攻撃を失敗させるには計算量を確保することが必要である。256^6 の計算量（総当り攻撃で 223 年かかる）があれば十分防御できるはずである。もちろん、256^8 が望ましく可能であれば 256^8 の計算量を確保すべきである。

計算量を確保するにあたり、次の場合について計算量を確保する方法を提案する。

①誤設定した IV を変更できる場合
②誤設定した IV を変更できない場合

7.1 誤設定した IV を変更できる場合

SP800-38A の要求通り、ランダム関数などを使って IV を予測不可能に修正すべきである。そうすれば、256^8 の計算量が確保できる。

ストリームの処理中はキーが同じになってしまうので 、IV をランダムにすることで出力がランダムになっている 。AES の出力がランダムになるかは IV がランダムであるか否かに依存している。

AES には確定的性質と確率的性質の両面があることを知っておき、誤って IV を予測可能に設定して意図せず確定的暗号を作り込まないよう注意すべきである。

7.2 誤設定した IV を変更できない場合

誤設定した製品が出荷済で改修が実質的に不可能であるなど、なんらかの事情で誤設定した IV を変更できない場合は、攻撃者から送られてくる平文に対して引数の構成チェックを行うことを提案する。

攻撃者が送信する平文	計算量	総当り攻撃必要時間
8,5,5,5,5,5,5,5	256^7	57123.28294 年
5,8,5,5,5,5,5,5	256^7	57123.28294 年
5,5,8,5,5,5,5,5	256^6	223.13782 年
5,5,5,8,5,5,5,5	256^5	318.14572 日

表 2-1、表 2-2 の No17,18,19,20 に示したように、256^6 以上の計算量を確保すれば現実的に攻撃は不可能である。256^5 の計算量だと 318 日程度要するが、攻撃は可能である。

そのため、引数チェックの方法は、上位 3 バイトをみて全て同じ数であったら、引数チェックエラーとして処理しないことである。例えば、5,5,8,5,5,5,5,5 は引数チェックは OK とするが、5,5,5,8,5,5,5,5 は引数チェックエラーとする。

8. 参考文献

[1]NIST Special Publication 800-38A.2001 Edition. https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38a.pdf
[2]平野貴人,川合豊,小関義博.確定値を部分的に使った高速な共通鍵暗号ベース秘匿検索.Computer Security Symposium 2017
[3]山岡裕司,牛田芽生恵,伊藤孝一.頻度分布を k-匿名性で緩和する検索可能共通鍵暗号.Computer Security Symposium 2015
[4]菅孝徳,西出隆志,櫻井幸一.検索可能暗号の安全性再考.Computer Security Symposium 2011
[5]RFC5297.https://tools.ietf.org/pdf/rfc5297.pdf
[6]SomakDas,VineetGopal,KevinKing,AmruthVenkatraman.IV=0 Security Cryptographic Misuse of Libraries. https://courses.csail.mit.edu/6.857/2014/files/18-das-gopal-king-venkatraman-IV-equals-zero-security.pdf.May 14, 2014
[7]TomTervoort.Zerologon:Unauthenticated domain controller compromise by subverting Netlogon cryptography (CVE-2020-1472)September 2020
[8]TalBe'ery.https://github.com/talbeerysec/Zerologon/blob/master/zerologon.py
[9][MS-NRPC]:Netlogon Remote Protocol. https://winprotocoldoc.blob.core.windows.net/productionwindowsarchives/MS-NRPC/%5bMS-NRPC%5d.pdf

9. シミュレーションプログラム

攻撃対象内での動作を模したもので攻撃用コードではない。自由に使って頂いて構わない。

```
#!/usr/bin/env python
from Crypto.Cipher import AES
from Crypto.Random import get_random_bytes
import time

def test():
  Challenge = bytearray([5,5,5,5,5,5,5,5])
  Credentials = bytearray([7,7,7,7,7,7,7,7])
  print("test25\n");
  simulate_deterministic_nature(Challenge,Credentials)

def simulate_deterministic_nature(Challenge,Credentials):
  iv = bytearray([7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7])
  key = bytearray(16)
  enc = bytearray(8)
  enc[0]=1
  stream = bytearray(16)
  j=0
  start_time=time.time()
  while(enc != Credentials):
   key = get_random_bytes(16)
   cipher = AES.new(key,AES.MODE_CFB,iv)
   enc=cipher.encrypt(Challenge)
   stream = bytes(Challenge^enc for (Challenge,enc) in zip(Challenge,enc))
   j=j+1
  finish_time=time.time()
  Exection_time=finish_time-start_time
    print("Number of times: "+str(j)+"[PlainText] "+Challenge.hex()+"  [key] "+key.hex()+" [CipherText] "+enc.hex()+" [stream]"+stream.hex() +"\n")
  print("Attack_Success_Exection_time(sec): "+str(Exection_time)+"\n")

test()
```

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

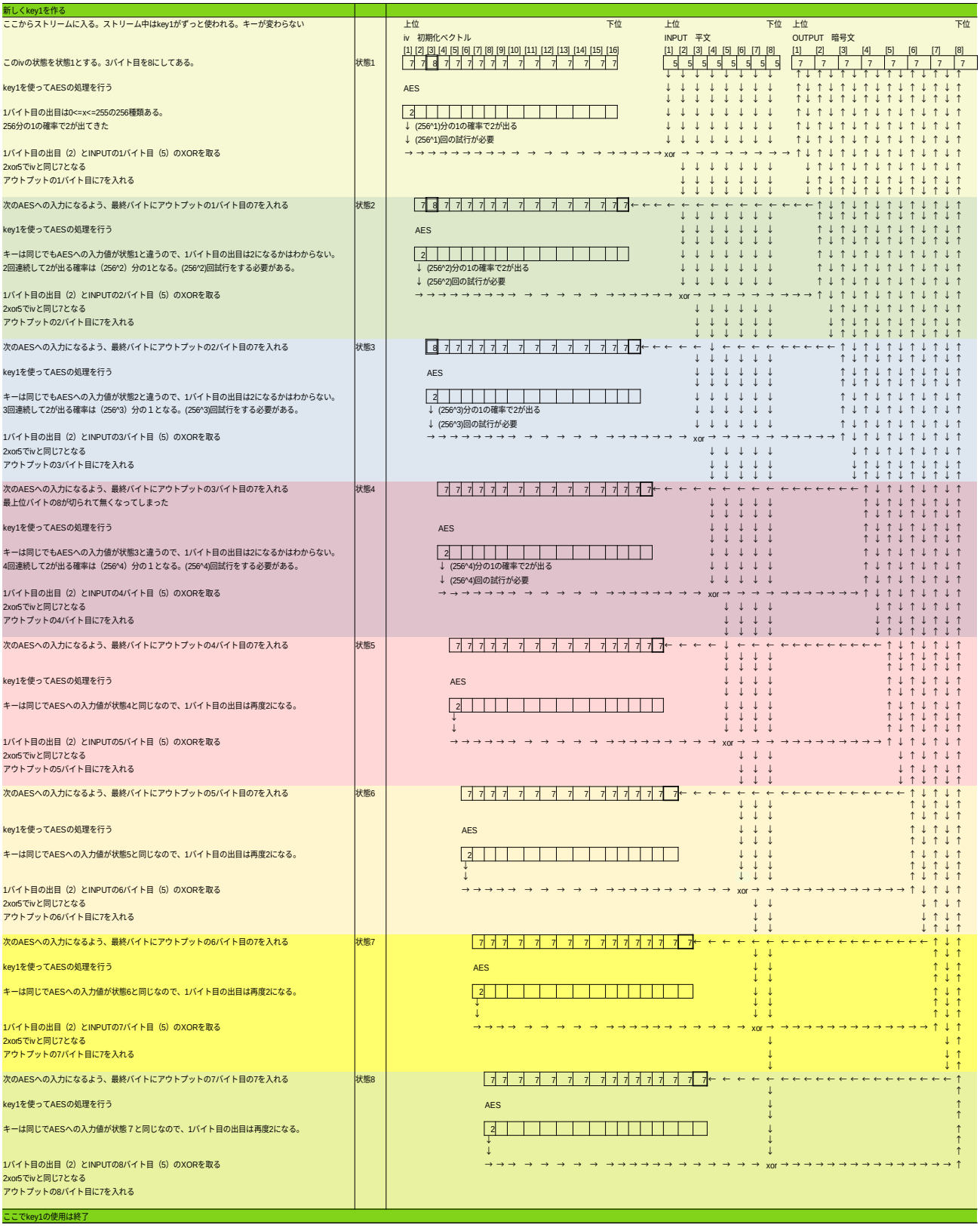

図２　ケース 1-14　ストリーム内での処理を詳細化

IV 7,7,8,7,7,7,7,7,7,7,7,7,7,7,7,7
攻撃者が送信する平文 5,5,5,5,5,5,5,5
出力予想した暗号文 7,7,7,7,7,7,7,7

バイト列 7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7 が
5 回 AES へ入力されている。
そのため、計算量は 256^(9-5)=256^4 となる。

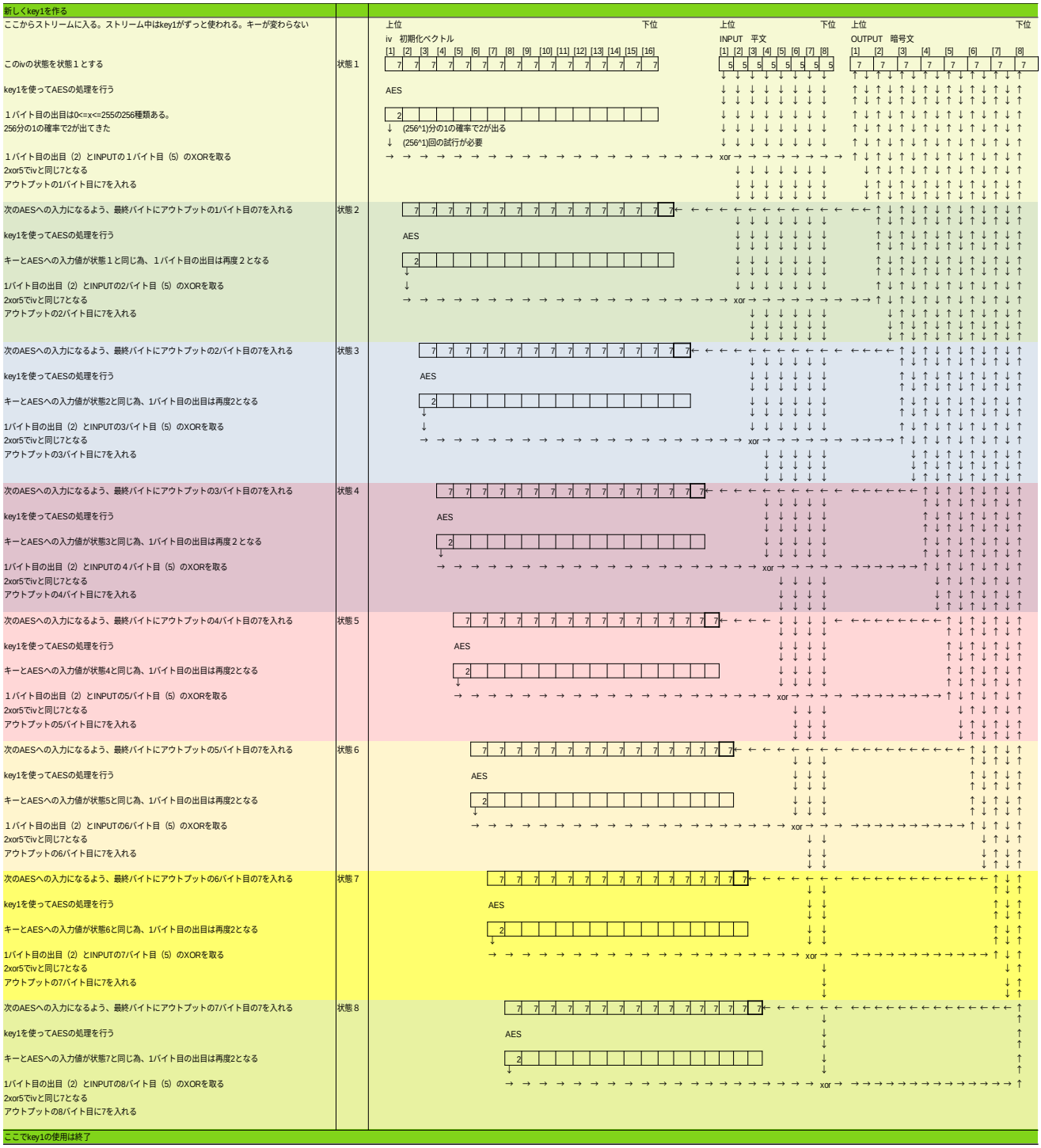

図３　ケース 2-25　ストリーム内での処理を詳細化

IV 7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7
攻撃者が送信する平文 5,5,5,5,5,5,5,5
出力予想した暗号文 7,7,7,7,7,7,7,7

バイト列 7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7 が
8 回 AES へ入力されている。
そのため、計算量は 256^(9-8)=256^1 となる。

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

	IV	入力平文	必要試行回数（計算量）	必要攻撃時間 ※1	出力予想暗号文	実際に出力された暗号文	攻撃成否	実際の試行回数	暗号文出力までの PC での実行時間（秒）	ストリーム中で使用されたキー
1	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,8	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
2	7,7,7,7,7,7,7,7,7,7,7,7,7,7,8,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
3	7,7,7,7,7,7,7,7,7,7,7,7,7,8,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
4	7,7,7,7,7,7,7,7,7,7,7,8,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
5	7,7,7,7,7,7,7,7,7,7,8,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
6	7,7,7,7,7,7,7,7,7,8,7,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
7	7,7,7,7,7,7,7,7,8,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
8	7,7,7,7,7,7,7,8,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
9	7,7,7,7,7,7,8,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
10	7,7,7,7,7,8,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^8	461168601842738 秒 (14623560.43387 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
11	7,7,7,7,8,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^7	1801439850948 秒 (57123.28294 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
12	7,7,7,7,8,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^6	7036874417 秒 (223.13782 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
13	7,7,7,8,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^5	27487790 秒 (318.14572 日)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
14	7,7,8,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^4	107374 秒 (29.82616 時間)	7,7,7,7,7,7,7,7	0707070707070707	成功	6286620645	158882.943431854 秒 (44.13415 時間)	fc9c8602047270888a9da4dcdf53a226
15	7,8,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^3	419 秒 (6.99050 分)	7,7,7,7,7,7,7,7	0707070707070707	成功	19189263	490.681556940079 秒 (8.17802 分)	5a4759c4c86755a6ae3dc98d018a4654
16	8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^2	1.6384 秒	7,7,7,7,7,7,7,7	0707070707070707	成功	14244	0.369987726211548 秒	d0b0c9f67771cf33179bec3a5bd7a530

表 1-1　ケース 1 の実験計画

表 1-2　ケース 1 の実験結果

※1　総当たり攻撃に必要な計算時間。1 ループあたりの必要時間 2.5/(10^5)sec で計算。

※2　実験実施せず。必要攻撃時間から判断し、256^4（107374 秒、 29.82616 時間 ）までの実験を行った。256^5 以上になると現実的な時間に実験が終了しないため。

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

	IV	入力平文	必要試行回数（計算量）	必要攻撃時間 ※1	出力予想暗号文	実際に出力された暗号文	攻撃成否	実際の試行回数	暗号文出力までの PC での実行時間（秒）	ストリーム中で使用されたキー
17	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	8,5,5,5,5,5,5,5	256^7	1801439850948 秒 (57123.28294 年)	7,10,7,7,7,7,7,7	※2	※2	※2	※2	※2
18	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,8,5,5,5,5,5,5	256^7	1801439850948 秒 (57123.28294 年)	7,10,7,7,7,7,7,7	※2	※2	※2	※2	※2
19	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,8,5,5,5,5,5	256^6	7036874417 秒 (223.13782 年)	7,7,10,7,7,7,7,7	※2	※2	※2	※2	※2
20	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,8,5,5,5,5	256^5	27487790 秒(318.14572 日)	7,7,7,10,7,7,7,7	※2	※2	※2	※2	※2
21	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,8,5,5,5	256^4	107374 秒(29.82616 時間)	7,7,7,7,10,7,7,7	'070707070a070707	成功	2124140252	54189.3640060425 秒 (15.05260 時間)	bed194e1e03f06e6548635d515be26fb
22	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,8,5,5	256^3	419 秒(6.99050 分)	7,7,7,7,7,10,7,7	'07070707070a0707	成功	6089656	157.309051036835 秒 (2.62181 分)	ae36fa38b8e166351ae9e5ca93d8c61b
23	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,8,5	256^2	1.6384 秒	7,7,7,7,7,7,10,7	'0707070707070a07	成功	3645	0.23145318031311 秒	480ef84eace9dc1576f211812e0e8291
24	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,8	256^1	0.0064 秒	7,7,7,7,7,7,7,10	'070707070707070a	成功	242	0.0165338516235352 秒	abc900a1ee39d5df401a89cf8ca0ce96
25	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	5,5,5,5,5,5,5,5	256^1	0.0064 秒	7,7,7,7,7,7,7,7	0707070707070707	成功	238	0.0258557796478272 秒	5490d3e530c786c6fb6a5f349e799a1d

表 2-1　ケース 2 の実験計画

表 2-2　ケース 2 の実験結果

	IV	入力平文	必要試行回数（計算量）	必要攻撃時間 ※1	出力予想暗号文	実際に出力された暗号文	攻撃成否	実際の試行回数	暗号文出力までの PC での実行時間（秒）	ストリーム中で使用されたキー
26	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0	0,0,0,0,0,0,0,0	256^1	0.0064 秒	0,0,0,0,0,0,0,0	0000000000000000	成功	174	0.0108475685119629 秒	d6f58f29ea2151d7e58b4f8575717610
27	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	7,7,7,7,7,7,7,7	256^1	0.0064 秒	7,7,7,7,7,7,7,7	0707070707070707	成功	343	0.0230238437652588 秒	2f69ce495fdc210e60355c17ed611446
28	99,99,99,99,99,99,99,99,99,99,99,99,99,99,99,99	99,99,99,99,99,99,99,99	256^1	0.0064 秒	99,99,99,99,99,99,99,99	6363636363636363	成功	97	0.00709891319274902 秒	2086dc66d9682f241eb346f061a541e5
29	'255,255,255,255,255,255,255,255,255,255,255,255,255,255,255,255	255,255,255,255,255,255,255,255	256^1	0.0064 秒	255,255,255,255,255,255,255,255	ffffffffffffffff	成功	774	0.0368108749389648 秒	8e90054aaf188f57e9b9018a26513362

表 3-1　ケース 3 の実験計画

表 3-2　ケース 3 の実験結果

Vol.2021-DPS-186 No.66
Vol.2021-CSEC-92 No.66
2021/3/16

	IV	入力平文	必要試行回数（計算量）	必要攻撃時間 ※1	出力予想暗号文	実際に出力された暗号文	攻撃成否	実際の試行回数	暗号文出力までの PC での実行時間（秒）	ストリーム中で使用されたキー
30	1,2,3,4,5,6,7,8,8,8,8,8,8,8,8,8	5,6,7,8,9,10,11,0	256^8	461168601842738 秒 (14623560.43387 年)	8,8,8,8,8,8,8,8	※2	※2	※2	※2	※2
31	1,2,3,4,5,6,7,7,7,7,7,7,7,7,7	5,6,7,8,9,10,0,0	256^7	1801439850948 秒 (57123.28294 年)	7,7,7,7,7,7,7,7	※2	※2	※2	※2	※2
32	1,2,3,4,5,6,6,6,6,6,6,6,6,6,6,6	5,6,7,8,9,0,0,0	256^6	7036874417 秒 (223.13782 年)	6,6,6,6,6,6,6,6	※2	※2	※2	※2	※2
33	1,2,3,4,5,5,5,5,5,5,5,5,5,5,5,5	5,6,7,8,0,0,0,0	256^5	27487790 秒(318.14572 日)	5,5,5,5,5,5,5,5	※2	※2	※2	※2	※2
34	1,2,3,4,4,4,4,4,4,4,4,4,4,4,4,4	5,6,7,0,0,0,0,0	256^4	107374 秒(29.82616 時間)	4,4,4,4,4,4,4,4	0404040404040404	成功	23525415427	586084.764806032 秒 (162.80132 時間)	dc41349201d3e12e46630036ca047133
35	1,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3	5,6,0,0,0,0,0,0	256^3	419 秒(6.99050 分)	3,3,3,3,3,3,3,3	0303030303030303	成功	36722825	953.409956932068 秒 (15.89016 分)	84ae702a84115abda5e775cc82aa9916
36	1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	5,0,0,0,0,0,0,0	256^2	1.6384 秒	2,2,2,2,2,2,2,2	0202020202020202	成功	266	0.0088193416595459 秒	dd09c41a90f7141f4fe65631d8fbe03b

表 4-1　ケース 4 の実験計画

表 4-2　ケース 4 の実験結果

表 5　ストリーム内ではＥＣＢモードが使われていることを示す表

test case	キー	AES への入力平文	ECB モードでの暗号文出力	出力したと予想していたＣＦＢモード のストリーム 1 バイト目の数
No14	fc9c8602047270888a9da4dcdf53a226	7,7,8,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**a4f67682fd0e7ea18ba5a6e0a1ae6c	2
		7,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**bac0898d9243dd052de66aac55140a	2
		8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**9a352f10d800dcc0f38d37bf7f66f6	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**ff5babd7776cd6fb5f909244c17bd6	2
No15	5a4759c4c86755a6ae3dc98d018a4654	7,8,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**421f641bfd8d9472a4029451a3c047	2
		8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**40f84220a42a697eabe5dfdbf8bb32	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**addec5c80d3030f315b5830b7e587b	2
No16	d0b0c9f67771cf33179bec3a5bd7a530	8,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**6a2f5566e8a1fbed095a52f0721981	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**f9345fa589ca75ccfa765889467a68	2
No21	bed194e1e03f06e6548635d515be26fb	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**e74bea55fa00f4e7477a808a73c972	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,10	**02**33a521445aaa2243db90d38bf43bb7	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,10,7	**02**71f8533e9e8e32bda1d77dc6e636aa	2
		7,7,7,7,7,7,7,7,7,7,7,7,7,10,7,7	**02**28671dff397ccf06ccbba8f13a732f	2

test case	キー	AESへの入力平文	ECBモードの暗号文出力	出力したと予想していたＣＦＢモード のストリーム１バイト目の数
No22	ae36fa38b8e166351ae9e5ca93d8c61b	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**3c3720ebb6ae8ef575d9b89bbe88ca	**2**
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,10	**02**856f64dd519726cd11cac3f7ea6eee	**2**
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,10,7	**02**ce299a602f35c0dc84ef5619d1018d	**2**
No23	480ef84eace9dc1576f211812e0e8291	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**4a8790ffaf08acc92163bf9f532585	**2**
		7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,10	**02**697f741ab7c92fdb4750df6208bd53	**2**
No24	abc900a1ee39d5df401a89cf8ca0ce96	7,7,7,7,7,7,7,7,7,7,7,7,7,7,7,7	**02**a92101625f024a7a84ef46b194e1d8	**2**
No35	84ae702a84115abda5e775cc82aa9916	1,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3	**06**ca08e9f0f197202fe69034f458b099	**6**
		2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3	**05**fed4836f54e3cf2bd54bcd97b7b957	**5**
		3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3	**03**d2126c44d2e2bddb3e0d772c52cbe5	**3**
No36	dd09c41a90f7141f4fe65631d8fbe03b	1,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	**07**af5b88a13ac6ca499016c92af415a5	**7**
		2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2	**02**fd62310644b7bbc8af2be7d5a8668b	**2**

表1-2、表2-2、表3-2、表4-2の結果より観測されたキーを使いＥＣＢモードでの暗号文出力処理を行った。
ＥＣＢモードで出力された暗号文の１バイト目の数と出力したと予想していたＣＦＢモード のストリーム１バイト目の数が確かに一致することが確認できた。