Kobe University Repository : Kernel

PDF issue: 2026-11-23

ストリーム暗号の安全性に関する研究

大東, 俊博

(Degree)
博士（工学）

(Date of Degree)
2008-03-25

(Date of Publication)
2009-06-26

(Resource Type)
doctoral thesis

(Report Number)
甲4202

(URL)
https://hdl.handle.net/20.500.14094/D1004202

※ 当コンテンツは神戸大学の学術成果です。無断複製・不正使用等を禁じます。著作権法で認められている範囲内で、適切にご利用ください。

博 士 論 文

ストリーム暗号の安全性に関する研究

平成 20年 1月

神戸大学大学院自然科学研究科

大 東　俊 博

内容梗概

情報や通信の機密性を保護するための技術として，暗号技術が知られている．暗号技術の中でもストリーム暗号は特に高速処理を必要とする場合の暗号化処理で使われ，近年の通信の高速化やデータの大容量化に伴い注目されている．ストリーム暗号では鍵を用いて擬似乱数系列 (キーストリーム) を生成し，それを平文と排他的論理和することにより暗号文を生成する．ストリーム暗号の安全性はキーストリームの性質に依存することが知られており，特にキーストリームの一部から乱数の種である鍵や内部状態 (鍵と等価なデータ) を復元できる解読法が存在する場合に問題が生じる．本研究では代表的なストリーム暗号である RC4 の安全性を評価することを目的とし，RC4 の解読法に関して議論する．RC4 は多くの商用のアプリケーションや標準プロトコルで使われており，近年では類似した構造を持つストリーム暗号も多数設計されていることから，その安全性を解析することはストリーム暗号の安全性評価において重要と考えられる．RC4 の鍵は可変長であり，8 ビットから 2048 ビットの間で鍵長を選択可能である．128 ビットなどの実用的な鍵長では，無線 LAN 用プロトコルである WEP に対する攻撃などの特別な場合を除いて，RC4 に対する有効な解読法は未だ示されていない．

ストリーム暗号では，暗号化処理毎の鍵であるセッション鍵を用いて，KSA と呼ばれる処理により内部状態を初期化する．そして，初期化された内部状態 (初期状態) を入力とする PRGA と呼ばれる処理によりキーストリームを生成する．RC4 ではキーストリームから初期状態を復元する内部状態復元攻撃は活発に議論されているが，初期状態を復元した後にセッション鍵を復元する方法については議論されていない．そこで，RC4 の初期状態の一部からセッション鍵を復元する方法を提案する．まず，KSA の構造を解析することで，セッション鍵と初期状態間において有意な確率で成立する線形方程式を導出する．さらに，その線形方程式を効果的に用いることで，初期状態の一部 (セッション鍵長と同じサイズのデータ) からセッション鍵を復元する方法を与える．この方法では，セッション鍵の復元に要する計算量は KSA を一度実行する程度であり，初期状態の復元に成功するセッション鍵は暗号学的に無視できない割合で存在する．すなわち，初期状態が復元されたとき，提案手法を用いれば高確率でセッション鍵を容易に復元可能であることが示される．

次に，RC4 に対する内部状態復元攻撃の中で最も有効な Knudsen らの方法に注目し，その方法の効率化について議論する．Knudsen らの方法ではキーストリームと内部状態間で成立する関係式を用いる木探索によって初期状態を復元する．そのアルゴリズムでは，その関係式を用いる枝刈りによって計算量を削減しているため，キーストリームの情報を多く利用できる

ならば更に効率的に初期状態を探索することが可能となる．Knudsen らの方法は必要なキーストリームのサイズを小さくすることを重視したアルゴリズムである．そこで，Knudsen らの木探索アルゴリズムを解析し，キーストリームの情報をより多く利用できるようにアルゴリズムを改良することで，初期状態を効率的に探索する方法を提案する．

WEP に組み込まれた RC4 では，キーストリームから秘密鍵を復元する鍵復元攻撃が知られている．これはパケット毎に生成される WEP のセッション鍵が初期化ベクトル (IV) と秘密鍵を連結することによって作られることに注目した攻撃である．ここで，IV はパケット毎に変化する公開値である．WEP に対する鍵復元攻撃では，秘密鍵及びキーストリームとの相関が生じるような特定のパターンの IV を用いて秘密鍵を復元する．このような IV は weak IV と呼ばれている．また，weak IV をフィルタリングして取り除くことにより鍵復元攻撃を防ぐ WEPplus と呼ばれる技術も存在し，その技術は対策の手軽さから多くの無線 LAN 製品で採用されている．

従来の weak IV に関する研究では，秘密鍵との間に強い相関が生じる weak IV について考察し，それらを特定していた．このような weak IV では選択される IV に強い条件が与えられることが多く，IV 空間における weak IV の割合は小さくなる傾向にあるため，WEPplus による対策が有効となる．一方，秘密鍵との相関が小さな weak IV でも多数集めることができれば秘密鍵を復元する際に有効に働くと考えられる．そこで，秘密鍵との相関が小さい weak IV に注目した鍵復元攻撃を提案する．まず初めに，KSA における内部状態の遷移を解析することによって相関が小さい weak IV を特定する．特定する weak IV は IV の選択における条件が少なくなるため，IV 空間に大きな割合で存在する．具体的には，初期に提案された鍵復元攻撃である FMS 攻撃では約 0.02%の割合の IV が weak IV となるが，本研究で特定する weak IV は IV 全体の約 81%を占める．この weak IV をフィルタリングして取り除くとすると，IV 空間を極端に小さくなり実用上の問題が生じてしまうため，WEPplus による対策は困難と考えられる．さらに，この weak IV を効果的に利用可能な鍵復元攻撃を提案する．具体的には，weak IV が持っている秘密鍵との相関の大きさが秘密鍵に依存して変化することに注目し，相関の大きさによって weak IV を選別することで秘密鍵を復元する確率を向上させる方法を与える．提案する weak IV の選別法では，鍵復元攻撃の処理中に生じる情報を利用して weak IV の持っている秘密鍵との相関の大きさを判別する．また，計算機実験によって，この選別法を用いる鍵復元攻撃が現在知られている最も有効なフィルタリングパターンを用いる WEPplus に対して有効に働くことを確認する．

weak IV に関する議論が盛んに行われた結果，最近になって，WEP に対する既知 IV 攻撃が提案された．既知 IV 攻撃では任意の IV が利用可能，すなわち全ての IV が weak IV となるために WEPplus による対策は不可能である．既知 IV 攻撃で最も効果的な方法は Tews らによっ

て提案された攻撃 (PTW 攻撃) であり，その攻撃は任意の IV とそれに対応するキーストリームの先頭の 15 バイトを多数利用して秘密鍵を復元する．提案手法では，RC4 の KSA においてセッション鍵が繰り返し利用されることに注目し，キーストリームの 16 バイト目以降を鍵復元攻撃に利用する．具体的には，攻撃に利用できるキーストリームのサイズを 22 バイトまで拡張する方法を与え，観測できる暗号化パケット数が等しい条件において秘密鍵を復元できる確率を PTW 攻撃と比べて向上させる．特に 20,000 個の暗号化パケットを観測できる場合において，104 ビットの秘密鍵を復元できる確率は提案手法は PTW 攻撃の 2 倍程度まで向上する．

以上のように本論文では，RC4 に対する解読法に関して複数の視点から検討する．まず，RC4 の初期状態から高確率でセッション鍵を復元する方法を与え，さらに初期状態をキーストリームから得るための方法の効率化を行う．また，WEP に実装された RC4 に関する安全性についても議論し，秘密鍵との相関が比較的小さな weak IV を利用した効果的な鍵復元攻撃を提案する．最後に，最近議論されている WEP に対する既知 IV 攻撃についても改良し，観測できる暗号化パケット数が等しい条件において秘密鍵を復元できる確率を向上させる．

目　次

第 1 章　緒論 **1**

第 2 章　ストリーム暗号 **5**

2.1　緒言 5

2.2　ストリーム暗号の構造 5

2.3　ストリーム暗号の安全性 6

2.4　RC4 8

2.4.1　鍵スケジューリングアルゴリズム (KSA) 8

2.4.2　擬似乱数生成アルゴリズム (PRGA) 9

2.5　RC4 に対する従来の安全性評価 10

2.5.1　鍵復元攻撃 10

2.5.2　内部状態復元攻撃 11

2.5.3　出力予測攻撃 11

2.5.4　識別攻撃 11

2.6　結言 12

第 3 章　RC4 における初期状態からのセッション鍵復元手法 **13**

3.1　緒言 13

3.2　初期状態から容易に復元可能なセッション鍵 13

3.3　線形方程式 14

3.4　初期状態からのセッション鍵復元 18

3.5　計算機実験 22

3.6　結言 22

第 4 章　RC4 に対する内部状態復元攻撃の効率化 **25**

4.1　緒言 25

4.2 Knudsen らの内部状態復元攻撃 25
4.3 効率的な内部状態復元攻撃 28
4.3.1 深さ優先指向探索の有効性 28
4.3.2 アルゴリズム 29
4.4 計算機実験 31
4.5 結言 31

第 5 章 WEP における新しい weak IV と鍵復元攻撃への応用 33

5.1 緒言 33
5.2 WEP におけるセッション鍵の生成 34
5.3 鍵復元攻撃の原理 34
5.3.1 攻撃方程式 34
5.3.2 攻撃方程式を用いた統計的な鍵復元手法 35
5.4 秘密鍵との相関が小さな weak IV 36
5.4.1 攻撃方程式 37
5.4.2 weak IV の割合 41
5.5 推測値を利用する weak IV の選別法 41
5.5.1 従来研究 42
5.5.2 推測値選別法 42
5.5.3 提案手法 43
5.6 評価 45
5.7 結言 46

第 6 章 WEP に対する既知 IV 攻撃の改良 49

6.1 緒言 49
6.2 定義 49
6.3 PTW 攻撃 49
6.4 提案手法 50

6.5 評価 53
6.6 結言 54

第 7 章 結論 **55**

謝 辞 **57**

参考文献 **59**

関連発表 **65**

図 目 次

図 2.1　IV を用いる場合のストリーム暗号の暗号化/復号処理 6

図 3.1　$t' = 2, 3, \ldots, N$ の場合に式 (3.1) が成立する条件 16

図 3.2　式 (3.1) の成立確率 23

図 4.1　内部状態復元攻撃の成功率の比較 32

図 5.1　統計的な鍵復元手法 36

図 5.2　提案する攻撃方程式における fortunate event 38

図 5.3　式 (5.12) から生じる偏差の実験値と理論値 41

図 5.4　CGV approach を利用した鍵復元処理のアルゴリズム 44

図 6.1　σ_i が復元される確率 51

表 目 次

表 3.1　提案手法によって復元できるセッション鍵 K とその初期状態 S_0 $(= S^*_{256})$ およびポインタ j^*_t の例 23

表 4.1　計算機実験のパラメータ 28

表 4.2　BFOS と DFOS の比較 29

表 5.1　式 (5.12) のための fortunate event の時刻 t における条件とその成立確率 . . 37

表 5.2　subevent$(a+1)$ における $S^*_{t-1}[j^*_t] = j^*_t$ の成立確率と j^*_t の関係 40

表 5.3　鍵復元攻撃の評価 (Case 1) 47

表 5.4　鍵復元攻撃の評価 (Case 2) 47

表 6.1　提案手法と PTW 攻撃によって復元できた秘密鍵の個数 53

表 6.2　確率 P_{err} でキーストリームが誤る場合における提案手法と PTW 攻撃によって復元できた秘密鍵の個数 54

第1章　緒論

近年の電子商取引の発達や行政サービスの電子化によって，重要なデータを通信路上でやり取りする機会が増加してきている．また，個人情報の漏洩 [1] やデータの改ざん [2] などが多発しており，データの安全性を保証するための暗号技術が重要視されてきている．データの機密性を保護するための暗号化処理において，特に多量のデータを扱う場合には共通鍵暗号方式 [3] が用いられている．更に，近年の情報の大規模化や通信の高速化に伴い，共通鍵暗号方式の中でも高速処理に優れているストリーム暗号 [3, 4, 5] が注目されている．

ストリーム暗号は鍵スケジューリングアルゴリズム (KSA：Key Scheduling Algorithm) と擬似乱数生成アルゴリズム (PRGA：Pseudo-Random Generation Algorithm) から構成されている．KSA ではセッション鍵を用いて内部状態を初期化し，PRGA では初期化された内部状態 (初期状態) を入力として任意長の擬似乱数系列 (キーストリーム) を生成する．暗号文は平文とキーストリームの排他的論理和によって得られる．復号は同じセッション鍵を用いて生成されるキーストリームと暗号文の排他的論理和をとることで実行される．ストリーム暗号では異なる平文に対して同一のキーストリームを使用しないように，暗号化処理毎にセッション鍵を変更することが望まれる．セッション鍵は初期化ベクトル (IV：Initialization Vector) と秘密鍵を用いて生成されるか，もしくは任意に作られた秘密鍵が代入されて使われている．ここで，IV は暗号化処理毎に変化する公開値である．

一般に暗号の安全性はアルゴリズムを公開した上で専門家による解読法の議論を受けることによって評価される．長期間議論されて有効な解読法が発見されなかった暗号は，その時点において安全とみなされる．特に共通鍵暗号では，秘密鍵を網羅的に探索する方法と比べて効率的な方法が存在しないかについて議論されることになる．暗号の解読法に関する研究は，暗号設計者が気づかなかった欠点に関して議論することで，暗号の厳密な安全性を評価するという意味で必要不可欠な研究である．

ストリーム暗号では攻撃者がキーストリームの一部を得られるという条件において解読法が議論される．特に脅威となる解読法は，キーストリームの一部から秘密鍵を復元する攻撃 (鍵復元攻撃) 及び初期状態を復元する攻撃 (内部状態復元攻撃) である．内部状態復元攻撃が成功する場合，攻撃者は得られた初期状態から生成されるキーストリームを全て計算でき，復号と同様の処理によって暗号文全体を解読できる．鍵復元攻撃が成功する場合には更に深刻であり，攻撃者は得られた秘密鍵を用いることで全ての IV に対応するセッション鍵を計算できるため，それぞれのセッション鍵に対応する暗号文は全て解読されてしまう．また，秘密鍵自体がパスワードなどの意味のあるデータである場合，そのデータが漏洩することによって二次的な被害が生じることも考えられる．一方で，キーストリームの乱数性の評価を目的とした解読

法も存在する．識別攻撃はキーストリームと真性乱数系列を区別する攻撃であり，出力予測攻撃は一部のキーストリームからそれ以外の部分のキーストリームを高確率で予測する攻撃である．これらはストリーム暗号の潜在的な脆弱性を指摘し，今後の鍵復元攻撃や内部状態復元攻撃の発展に寄与するものであるが，暗号に対する大きな脅威とは言いがたい．

本研究では，代表的なストリーム暗号である RC4[3] の安全性，特に内部状態復元攻撃や鍵復元攻撃に注目して評価することを研究テーマとする．RC4 は多くの商用のアプリケーションや標準プロトコルで使われており，その安全性を評価する意義は大きい．さらに，RC4 と類似した構造を持つストリーム暗号が多数提案されており，RC4 の安全性評価がそれらに適用可能な場合も考えられるため，ストリーム暗号の安全性評価において RC4 の評価は特に重要だと考えられる．一般的な実装の RC4 では，8 ビットから 2048 ビットまでの可変長のセッション鍵を利用でき，2048 ビットの内部状態と 2 つのポインタを用いて任意長のキーストリームを 8 ビット単位で出力する．1987 年に開発されて以来，RC4 の安全性は多くの暗号学者によって議論されており，特に識別攻撃 [6, 7, 8, 9, 10, 11] 及び出力予測攻撃 [8, 11] に関して多くの研究成果が報告されている．直接の脅威となり得る攻撃では，内部状態復元攻撃が提案されているが [8, 11, 12, 13, 14, 15]，これらは 128 ビットなどの実用的な秘密鍵長においては有効ではない．以上のように，実用的な秘密鍵長において RC4 に対して直接の脅威となる解読法は未だ発見されていない．

RC4 では内部状態復元攻撃に関しては盛んに議論されているが，鍵復元攻撃への発展，すなわち得られた初期状態から秘密鍵を復元する方法に関しては議論されていない．秘密鍵を復元するためには少なくともセッション鍵を復元する必要がある．秘密鍵と IV からセッション鍵が生成されている場合，セッション鍵生成処理が可逆であれば得られたセッション鍵と IV から秘密鍵を復元できる．また，秘密鍵をセッション鍵として用いる場合には，セッション鍵から即座に秘密鍵が得られる．そこで，第一の研究テーマとして，RC4 の初期状態の一部からセッション鍵を復元する方法について検討する．まず初めに，RC4 の KSA の撹拌処理が不十分であることに注目し，初期状態の一部とセッション鍵の一部の間で高確率で成立する線形方程式を導出する．さらに，この線形方程式を複数用いることで，セッション鍵長と同程度のサイズの初期状態のデータからセッション鍵を復元する方法を提案する．この手法で復元可能なセッション鍵は暗号学的に無視できない割合で存在する．これは，初期状態の一部を効率的に復元する攻撃が発見された場合，セッション鍵を高確率で復元できることを意味している．

第二の研究テーマとして，内部状態復元攻撃の効率化について検討する．1998 年に Knudsen らによって内部状態復元攻撃 [12] が提案されて以来，様々な内部状態復元攻撃が提案されている [8, 11, 13, 14]．汎用的な内部状態復元攻撃の中では Knudsen らの方法が最も有効な攻撃として知られている．Knudsen らの方法はキーストリームと内部状態の一部の間で関係式を作

り，その関係式を用いる木探索アルゴリズムによって効率的に初期状態を復元する．Knudsen らの方法における木探索アルゴリズムは必要なキーストリームのサイズを小さくすることを重視したために，余分な計算が必要となってしまい，削減できる計算量が比較的少なくなっていた．Knudsen らの方法で必要なキーストリームサイズは他の解読法と比べても極めて小さく，それを数倍程度に増加させたとしても大きなデメリットとならない．そこで，攻撃に必要なキーストリームサイズを Knudsen らの方法の 2 倍程度に増加させ，その情報を利用することで余分な計算を排除できる木探索アルゴリズムを提案する．Knudsen らの方法では 2^{797} の計算量で復元可能な初期状態が存在したのに対し，提案手法を用いれば 2^{587} の計算量で復元可能な初期状態が存在している．すなわち，RC4 の内部状態復元攻撃への安全性は本手法によって低下することになる．

RC4 の安全性に関する特別な事例として，無線 LAN 用のプロトコルである WEP(Wired Equivalent Privacy)[16] に組み込まれた RC4 に対する解読法がある．2001 年に Fluhrer らによって WEP のセッション鍵生成処理の不備を突いた鍵復元攻撃が提案された [9, 17]．この攻撃は FMS(Fluhrer, Mantin, and Shamir) 攻撃として知られている．WEP ではパケット毎に使われるセッション鍵は IV と秘密鍵を連結することによって生成される．この場合，セッション鍵はパケット毎に一部しか変化せず，さらに変化する部分のセッション鍵のデータは攻撃者からは既知の値となる．Fluhrer らは，このような条件において特定のパターンの IV (weak IV) が使われた場合にキーストリームの先頭の 1 バイトと IV 及び秘密鍵の間に強い相関が生じることを示し，その相関を用いて秘密鍵を復元している．FMS 攻撃では $4,000,000$ 個から $6,000,000$ 個の暗号化パケットを観測することで 104 ビットの秘密鍵を復元できる．ただし，この脆弱性はプロトコルにおけるセッション鍵生成処理に問題があったため生じたものであり，RC4 自身が破られたわけではないことに注意する．FMS 攻撃はセッション鍵がパケット毎に独立に生成されるように実装することで防ぐことができる [9]．実際，インターネットで使われるプロトコルである SSL(Secure Sockets Layer)[18] や WEP の代替プロトコルである WPA(Wi-fi Protected Access)[19] では，そのような実装を採用しているため FMS 攻撃の脅威を受けない．一方で，セッション鍵生成処理を変更せず，IV の選択を工夫することで FMS 攻撃を防ぐ技術も存在する．この WEPplus[20] と呼ばれる技術では，フィルタリングによって IV の中から weak IV を取り除くことで FMS 攻撃を実行できないようにしている．WEPplus は軽微な変更を WEP に施すだけで実現できるため，多くの無線 LAN メーカでこの技術が採用されている．この方法で安全性を保つためには weak IV が新たに特定されるたびにフィルタリングパターンを更新する必要がある．

FMS 攻撃が提案されて以来，その改良法 [21, 22, 23, 24, 25] によって多くの weak IV が特定されている．それらの weak IV は FMS 攻撃と同程度の秘密鍵との相関を持っているもので

あり，比較的相関が小さくなるような weak IV に関しては議論されていなかった．そこで，第三の研究テーマでは秘密鍵との相関が小さな weak IV に注目し，そのような weak IV を用いた鍵復元攻撃を提案する．本研究で特定する weak IV は秘密鍵との相関が小さいが IV 空間内に大きな割合で存在するため，weak IV を多数集めることで秘密鍵の情報を効果的に得ることができる．この weak IV の割合は IV 全体の約 81%であり，フィルタリングによって取り除く場合には IV 空間を極端に狭めてしまい実用上の問題が生じてしまうため，WEPplus による対策は困難と考えられる．さらに，weak IV が持っている秘密鍵との相関の大きさが秘密鍵に依存して変化する場合があることに注目し，相関の大きさによって weak IV を選別することで秘密鍵を復元できる確率を向上させる方法を与える．提案する選別法では，鍵復元攻撃の処理中で生じる情報を利用し，秘密鍵との相関が比較的大きな weak IV を選別する．また，この選別法を用いる鍵復元攻撃が現在知られている最も有効なフィルタリングパターンを用いる WEPplus に対して有効であることを計算機実験により確認する．

第三のテーマに関する研究を行った後，最近になって，WEP に対する既知 IV 攻撃が Klein[26] によって提案された．既知 IV 攻撃では全ての IV が weak IV となるために WEPplus による対策は不可能である．さらに Tews らは Klein の攻撃を改良し，実装面における効率を改善した [27]．Tews らの攻撃は PTW 攻撃と呼ばれており，WEP に対する鍵復元攻撃の中で最も効果的な方法として知られている．第四のテーマでは PTW 攻撃を改良し，観測できた暗号化パケット数が等しい条件において秘密鍵を復元できる確率を向上させる方法を提案する．PTW 攻撃ではキーストリームの先頭の 15 バイトと IV を利用して秘密鍵を復元する．提案手法では，RC4 の KSA においてセッション鍵が繰り返し利用されることに注目し，キーストリームの 16 バイト目以降を鍵復元攻撃に利用する．具体的には，攻撃に利用できるキーストリームのサイズを 22 バイトまで拡張する方法を与え，1 個のパケットから得られる秘密鍵の情報を増加させる．本手法は PTW 攻撃よりも高確率で秘密鍵を復元でき，特に 20,000 個の暗号化パケットを観測できた場合において，104 ビットの秘密鍵を復元できる確率が PTW 攻撃の 2 倍程度まで向上する．

以下，本論文の構成について述べる．第 2 章では，ストリーム暗号の構成及びその安全性評価について述べる．さらに本研究で対象とする RC4 のアルゴリズム及び安全性に関する従来研究をまとめる．第 3 章では，第一の研究テーマである RC4 の初期状態の一部からセッション鍵を復元する方法を提案する．第 4 章では，第二の研究テーマとして Knudsen らによる RC4 に対する内部状態復元攻撃を効率化する方法を提案する．第 5 章では，第三の研究テーマとして秘密鍵との相関が小さな weak IV を用いて効果的に WEP の秘密鍵を復元する方法を提案する．第 6 章では，第四の研究テーマとして WEP に対する最新の鍵復元攻撃である PTW 攻撃を改良する手法を提案する．最後に，第 7 章では，結論として研究成果をまとめる．

第2章　ストリーム暗号

2.1　緒言

多量のデータを扱うような暗号化処理では，主に共通鍵暗号方式が用いられている．共通鍵暗号方式とは暗号化と復号で同一の鍵を用いる方式であり，事前に送信者と受信者で鍵を共有する必要がある．通常，鍵の共有には公開鍵暗号方式 [3] と呼ばれる別の技術が用いられる．ストリーム暗号は共通鍵暗号方式の一種であり，特に高速処理が要求される場面で用いられる．

本章では，まずストリーム暗号の構造について述べ，その暗号化/復号処理について説明する．次に，ストリーム暗号の安全性について述べ，解読法を分類する．さらに，本論文で対象とする RC4 のアルゴリズム及びその安全性の解析に関する従来研究を紹介する．

2.2　ストリーム暗号の構造

ストリーム暗号は平文をシンボルと呼ばれる小単位毎に逐次的に暗号化する方式であり，1ビットや1バイトまたは数バイト単位で暗号化するアルゴリズムが多い．ストリーム暗号ではセッション鍵を乱数の種として予測が困難な擬似乱数系列 (キーストリーム) を生成する．暗号文は，生成したキーストリームと平文を排他的論理和 (XOR : eXclusive-OR) することで生成される．復号では，同一のセッション鍵を用いてキーストリームを生成し，キーストリームと暗号文を排他的論理和することで暗号文を平文に戻す．シンボル単位に分割された平文を $M_1, M_2, \ldots$，暗号文を $C_1, C_2, \ldots$，キーストリームを $Z_1, Z_2, \ldots$ と定義すると，暗号化は式 (2.1)，復号は式 (2.2) のように表される．

$$C_t = M_t \oplus Z_t \quad (t = 1, 2, \ldots) \tag{2.1}$$

$$M_t = M_t \oplus Z_t \oplus Z_t = C_t \oplus Z_t \quad (t = 1, 2, \ldots) \tag{2.2}$$

ストリーム暗号の核となるのは擬似乱数生成器，すなわちセッション鍵からキーストリームを生成する処理である．この処理は鍵スケジューリングアルゴリズム (KSA : Key Scheduling Algorithm) と擬似乱数生成アルゴリズム (PRGA : Pseudo-Random Generation Algorithm) から構成されている．ストリーム暗号では大きなサイズの内部状態を持っており，KSA では小さなサイズのセッション鍵を用いて内部状態を初期化する．KSA によって初期化された内部状態のことを特に初期状態と呼ぶ．PRGA は初期状態を入力とし，内部状態を更新しながら任意長のキーストリームを生成する．

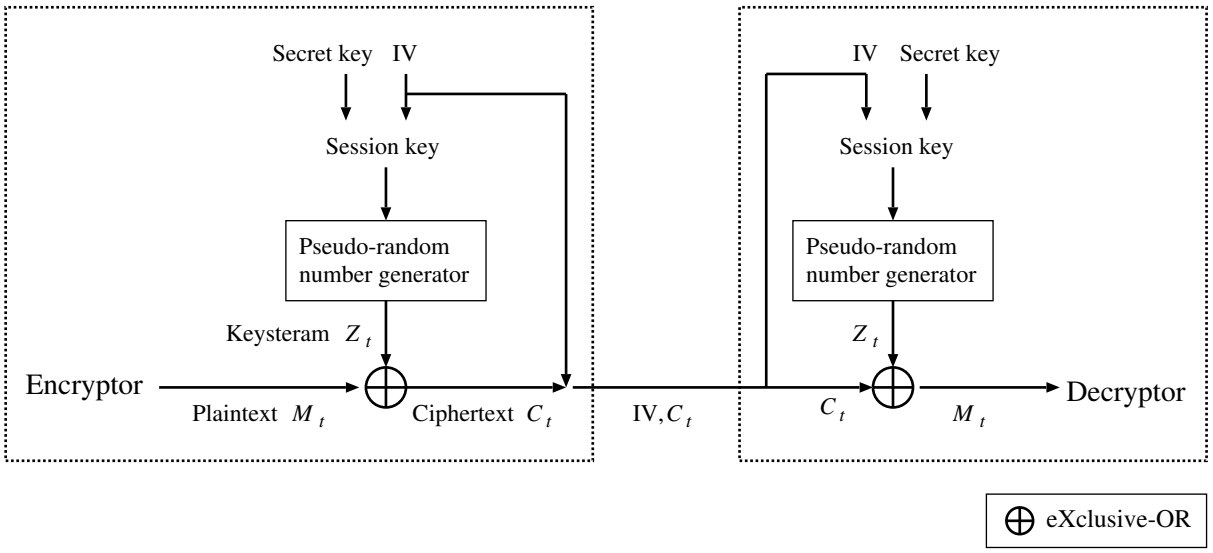


図 2.1　IV を用いる場合のストリーム暗号の暗号化/復号処理

ストリーム暗号では異なる暗号化処理において同一のキーストリームが使われることは弱点となる．なぜなら，同一のキーストリームによって平文 P，P' を暗号化されたとすると，それらの暗号文 C，C' を用いて

$$C_t \oplus C'_t = (P_t \oplus Z_t) \oplus (P'_t \oplus Z_t) = P_t \oplus P'_t \tag{2.3}$$

のように演算すれば平文間の差分情報が漏洩してしまうからである．そこで，ストリーム暗号では暗号化処理毎に異なるセッション鍵を用いることで同一のキーストリームが生成されることを防いでいる．セッション鍵は初期化ベクトル (IV：Initialization Vector) と事前に共有している秘密情報 (秘密鍵) を用いて生成されるか，もしくは任意に作られた秘密鍵が代入されて使われる．ここで，IV は暗号化処理毎に変化する公開値である．図 2.1 は IV を用いてセッション鍵を生成する場合のストリーム暗号の暗号化/復号処理を示したものである．また，図から IV の矢印を削除すれば，秘密鍵を単独でセッション鍵として用いる場合のストリーム暗号の暗号化/復号処理となる．

2.3　ストリーム暗号の安全性

一般に暗号はアルゴリズムを公開した上で各種解読法に対して耐性を持つことが望まれる．暗号は有限長の鍵を用いた変換によって暗号化/復号するため，秘密鍵を全数探索する攻撃 (鍵全数探索攻撃) を回避することは困難である．ℓ ビットのサイズの秘密鍵であれば，鍵全数探索攻撃に要する計算量は 2^ℓ となる．通常，鍵全数探索攻撃を計算量的に実行困難にするために，秘密鍵のサイズは 128 ビットなどの十分大きな値が用いられる．暗号設計者は鍵全数探索

攻撃に十分に耐性があるサイズの秘密鍵を使うようにした上で，鍵全数探索攻撃が最も有効な攻撃になるように設計する．もし，鍵全数探索攻撃より少ない計算量で実行できる解読法が発見されたならば，その暗号は暗号設計者の意図しない欠陥が存在することになり，その暗号は暗号学的には解読されたとみなされる．

ストリーム暗号の安全性は平文の一部を暗号解読者が推測可能という条件の下で評価される．式 (2.1) より，平文の一部が得られているならば，暗号文の一部を用いて

$$M_t \oplus C_t = M_t \oplus (M_t \oplus Z_t) = Z_t \tag{2.4}$$

のようにキーストリームの一部を得ることができる．もし，キーストリームの性質が悪く，キーストリームの一部からキーストリームの他の部分を得ることができるならば，暗号解読者は新たに得られたキーストリームのデータに対応する平文のデータを式 (2.2) から得ることができる．また，キーストリームの一部から秘密鍵や初期状態を復元できるなら，それらから生成されるキーストリームを全て計算可能であるため，対応する暗号文は全て解読されてしまう．以上のように，ストリーム暗号の安全性はキーストリームの性質に依存する．

ストリーム暗号に対する解読法は大きく以下の 4 つに分類される．

- 鍵復元攻撃
- 内部状態復元攻撃
- 出力予測攻撃
- 識別攻撃

ストリーム暗号に対する解読法の中で特に脅威となるのは，鍵復元攻撃と内部状態復元攻撃である．内部状態復元攻撃 [28, 29, 30, 31, 32, 33, 34] はキーストリームの一部から初期状態を復元する攻撃であり，得られた初期状態からキーストリーム全体を計算できる．したがって，その初期状態を用いて暗号化された暗号文は全て解読可能となる．鍵復元攻撃 [35, 36, 37, 38] はキーストリームの一部から秘密鍵を復元する攻撃である．鍵復元攻撃が成功する場合には更に深刻であり，得られた秘密鍵を用いることで全ての IV に対応するセッション鍵を計算できる．ゆえに，それぞれのセッション鍵を用いて暗号化された暗号文は全て解読可能となる．また，秘密鍵自体がパスワードなどの意味のあるデータである場合，そのデータを用いて不正アクセスが行われるなどの二次的な被害が生じる可能性もある．

出力予測攻撃 [11] はキーストリームの一部の情報を用いてキーストリームの他の部分をランダムに推測するより高確率で予測する攻撃である．出力予測攻撃が実現すれば一部の暗号文を平文に戻すことができる．また，識別攻撃 [39, 40, 41] はキーストリームと真性乱数系列を

区別する攻撃である．これらの攻撃はキーストリームの乱数性の評価という位置付けであり，ストリーム暗号に対する大きな脅威とは言いがたい．しかしながら，少なくともストリーム暗号の潜在的な脆弱性を指摘したことになるため，その脆弱性が将来的に鍵復元攻撃や内部状態復元攻撃に寄与する可能性はある．

2.4 RC4

RC4[3] は 1987 年に RSA Data Security 社の Rivest によって開発された可変長鍵のストリーム暗号である．RC4 はソフトウェアによる高速処理性能に優れており，Adobe Acrobat や Lotus Notes などの商用のアプリケーションやインターネット用のプロトコルである SSL(Secure Sockets Layer) 3.0/TLS(Transport Layer Security) 1.0[18]，無線 LAN 用のプロトコルである WEP(Wired Equivalent Privacy)[16] 及び WPA(Wi-Fi Protected Access) [19] などで広く使われている．さらに，近年では RC4A[10]，VMPC stream cipher[42]，ISAAC[43]，Py family (Py, Pypy, Py6, TPy, TPypy, TPy6)[44, 45, 46]，NGG[47]，MV3[48] など RC4 と類似した構造を持つストリーム暗号が多数提案されているなど，RC4 は注目されている暗号の一つである．

RC4 は開発されて以来，そのアルゴリズムは非公開であったが，1994 年にソースコードがインターネット上に匿名で流されて実質的に仕様が公開された形になった．しかしながら，RSA Data Security 社は公開されたコードを公式に RC4 と認めておらず，公開されたコードは ARC4(Alleged RC4) や arcfour と表記される RC4 と等価な動作をするアルゴリズムとしてインターネット上で公開されている．本研究では，ARC4 のアルゴリズムを RC4 とみなした上で安全性を評価している．

RC4 は可変長のセッション鍵 K を用いる．セッション鍵長 ℓ は 8 ビットから 2048 ビットの間で選択でき，実用的には $\ell = 128$ が使われる．RC4 の内部状態 S は $N(= 2^n)$ 個の要素からなり，それぞれの要素は n ビットの変数である．n は RC4 が処理単位 (ワード長) であり，実用的には $n = 8$，$N = 256$ が使われている．また，RC4 は内部状態の更新で 2 つのポインタ i, j を用いており，それぞれ n ビットの変数である．以下の節では RC4 の KSA と PRGA の処理を説明し，その性質について示す．以後の説明では，KSA と PRGA における内部状態とポインタを区別できるように，KSA の内部状態とポインタを $(S^*,\ i^*,\ j^*)$，PRGA の内部状態とポインタを $(S,\ i,\ j)$ のように異なる記号で表記している．

2.4.1 鍵スケジューリングアルゴリズム (KSA)

KSA の時刻 t の内部状態は $S_t^* = (S_t^*[0], S_t^*[1], \ldots, S_t^*[N-1])$ のように表される．$S_t^*[x],\ x \in \{0, 1, \ldots, N-1\}$ は内部状態の要素である．KSA の時刻 t の 2 つのポインタは i_t^*，j_t^* のように

表される．KSA では内部状態とポインタに以下のように初期値を代入する．

$$S_0^*[x] = x \text{ for } \forall x \in \{0, 1, \ldots, N-1\} \tag{2.5}$$

$$i_0^* = j_0^* = 0 \tag{2.6}$$

ℓ ビットのセッション鍵は $K = (K[0], K[1], \ldots, K[L-1])$ のように n ビット毎に L 個の要素に分割される．ここで，$K[y],\ y \in \{0, 1, \ldots, L-1\}$ はセッション鍵の要素であり，$L = \lceil \ell/n \rceil$ である．初期状態 $S_0 (= S_N^*)$ は時刻 $t = 1, 2, \ldots, N$ で式 (2.7)–(2.9) を実行することで得られる．

$$j_t^* = (j_{t-1}^* + S_{t-1}^*[i_{t-1}^*] + K[i_{t-1}^* \bmod L]) \bmod N \tag{2.7}$$

$$S_t^*[x] = \begin{cases} S_{t-1}^*[j_t^*] & \text{if } x = i_{t-1}^* \\ S_{t-1}^*[i_{t-1}^*] & \text{if } x = j_t^* \\ S_{t-1}^*[x] & \text{if } x \neq i_{t-1}^*, j_t^* \end{cases} \tag{2.8}$$

$$i_t^* = (i_{t-1}^* + 1) \bmod N \tag{2.9}$$

RC4 の KSA に関する性質を以下に示す．まず，ポインタ i^* は初期値が $i_0^* = 0$ であり，時刻 $t = 1, 2, \ldots, N$ で用いられるため，式 (2.9) は

$$i_t^* \ = \ t \tag{2.10}$$

と等価になる．次に，ポインタ j^* は時刻 $t = 1, 2, \ldots, N$ で一様でランダムな変数とみなされる．j_t^* は $S_{t-1}^*[i_{t-1}^*]$ と $K[i_{t-1}^* \bmod L]$ に依存して変化する変数である．時刻 t が小さいとき，$K[i_{t-1}^* \bmod L]$ の影響によってポインタ j_t^* はランダム化される．時刻 t が大きいとき，$S_{t-1}^*[i_{t-1}^*]$ にセットされている値は式 (2.8) によるスワップ操作でランダム化されているため，ポインタ j_t^* もランダム化される．さらに，内部状態の要素はスワップ操作でのみ変化することから，$S_t^*[x] \neq S_t^*[x'],\ x \neq x'$ のように全ての要素で値は異なるという性質も存在する．

2.4.2 擬似乱数生成アルゴリズム (PRGA)

PRGA の時刻 t の内部状態は $S_t = (S_t[0], S_t[1], \ldots, S_t[N-1])$ のように表される．PRGA の時刻 t の 2 つのポインタは i_t，j_t のように表される．内部状態とポインタは以下のように初期化される．

$$S_0[x] = S_N^*[x] \text{ for } \forall x \in \{0, 1, \ldots, N-1\} \tag{2.11}$$

$$i_0 = j_0 = 0 \tag{2.12}$$

PRGA の時刻 t における出力 (キーストリーム) は Z_t で表される．ここで，Z_t は n ビットの変数である．PRGA の時刻 t では，以下の処理によって内部状態を更新し，Z_t を出力する．

$$i_t = (i_{t-1} + 1) \bmod N \tag{2.13}$$

$$j_t = (j_{t-1} + S_{t-1}[i_t]) \bmod N \tag{2.14}$$

$$S_t[x] = \begin{cases} S_{t-1}[j_t] & \text{if } x = i_t \\ S_{t-1}[i_t] & \text{if } x = j_t \\ S_{t-1}[x] & \text{if } x \neq i_t, j_t \end{cases} \tag{2.15}$$

$$Z_t = S_t[(S_t[i_t] + S_t[j_t]) \bmod N] \tag{2.16}$$

PRGA においても KSA と同様にポインタ j_t は一様でランダムな変数とみなされ，内部状態の全ての要素で値は異なるという性質を持つ．

2.5 RC4 に対する従来の安全性評価

RC4 は使用実績を重ねるにつれて多くの暗号研究者の興味を引くことになった．その結果，RC4 の安全性に関する様々な研究結果が報告されている．しかしながら，WEP に対する鍵復元攻撃など特殊な場合を除いて，実用的な秘密鍵長 (128 ビットなど) における RC4 に対して直接の脅威を与える攻撃は発見されていない．以下の節では，RC4 の安全性に関する従来研究をまとめている．従来研究の説明では，特に断らない限り $n = 8$ の場合における評価結果を示しているものとする．

2.5.1 鍵復元攻撃

良く知られた鍵復元攻撃として，2001 年に Fluhrer，Mantin，Shamir によって提案された選択 IV 攻撃および関連鍵攻撃が存在する [9, 17]．これらの攻撃は特定のセッション鍵生成処理を採用している場合のみ有効な攻撃であり，RC4 自身に対する汎用的な攻撃法ではないことに注意する．SSL や WPA などの最新のプロトコルで使われている RC4 では，これらの攻撃を実行できないセッション鍵生成処理が採用されている．

選択 IV 攻撃は特に WEP に実装されている RC4 に対する方法が有名であり，初期に提案されたものは FMS 攻撃 (Fluhrer, Mantin, and Shamir attack) と呼ばれている．FMS 攻撃は秘密鍵長の線形オーダの計算量で実行できるため，現実的な時間で WEP の秘密鍵を復元できてしまう．また，104 ビットの秘密鍵を FMS 攻撃で復元するためには約 $4,000,000$ 個から $6,000,000$ 個の暗号化パケットを観測できればよい．FMS 攻撃は Stubblefield，Ioannidis，Rubin によって現実的に実現可能であることが確認され [49]，多くのオープンソースの WEP 解読ソフトウェアに実装されている [21, 50, 51, 52, 53]．現在，WEP に関する鍵復元攻撃は更に高度化しているが [21, 22, 24, 23, 26, 27]，その詳細は第 5 章と第 6 章で議論するためここでは省略する．

2.5.2 内部状態復元攻撃

1998 年に Knudsen，Meier，Preneel，Rijmen，Verdoolaege によって RC4 に対する内部状態復元攻撃が提案された [12]．Knudsen らの方法はキーストリームを利用する枝刈りを備えた木探索アルゴリズムであり，攻撃に要する計算量は初期状態の全数探索の平方根程度まで削減される．また，同時期に Mister と Tavares によって Knudsen らの方法と似た内部状態復元攻撃も提案されている [13]．2002 年には Golić によってキーストリームを入力とする事後確率の繰り返し計算に基づく内部状態復元攻撃 [14] が提案された．しかしながら，この手法が要する計算量は Knudsen らの方法よりも大きい．また，ある特別なパターンの内部状態を対象にする方法として，キーストリームから内部状態の一部の要素を効率的に復元できる方法も提案されている [8, 11, 15]．

どのような内部状態に対しても実行可能という意味で汎用的な内部状態復元攻撃では，初期に提案された Knudsen らの方法が最も有効な攻撃として知られている．しかしながら，Knudsen らの方法を用いたとしても初期状態の復元に 2^{797} の計算量が必要となり，798 ビット以上の大きなサイズの秘密鍵でしか有効な攻撃とならない．128 ビットなどの実用的な秘密鍵長において有効な内部状態復元攻撃は未だ発見されておらず，内部状態復元攻撃の効率化は重要な研究課題となっている．

2.5.3 出力予測攻撃

RC4 に対する出力予測攻撃は 2005 年に Mantin によって提案されている [11]．Mantin は Recyclable Fortuitous States と呼ばれる特別な内部状態に注目し，高確率でキーストリームの一部を予測する．この攻撃では，キーストリームの 2^{45} バイトを観測した後にそれ以外の 1 ビットを確率 0.85 で予測でき，2^{50} バイトを観測した後にそれ以外の 8 ビットを確率 0.82 で予測できる．

2.5.4 識別攻撃

1997 年に Golić は Z_t と Z_{t+2} の最下位ビット間に相関があることを示し，$2^{44.7}$ バイトのキーストリームの観測によって実行可能な識別攻撃を提案した [6]．2000 年には Fluhrer と McGrew によって Golić の方法が改良され，Z_t と Z_{t+1} 間の相関を利用することで必要なキーストリームのバイト数を $2^{30.6}$ まで減少させている [7]．2001 年に Fluhrer，Mantin，Shamir は invariance weakness に基づく weak key のクラスを発見した [9]．その weak key は，$n + q(L-1) + 1$ ビットに特定の条件が与えられたセッション鍵であり，その鍵を用いた場合には初期状態の特定の $O(qN)$ ビットを確率 0.5 で推測できる．ここで，q は攻撃で用いるパラメータであり $q \leq n, 2^q | L$

を満さなければならない．Fluhrer らはこの weak key を用いた識別攻撃を提案している．この識別攻撃は秘密鍵のサイズが小さい場合 (例えば，64 ビットの秘密鍵) においては文献 [7] の方法より効果的に動作する．2005 年に Mantin は 2 バイト単位の特徴的なキーストリームの出現分布における統計的な偏りを利用する識別攻撃を提案した．この攻撃は $2^{26.5}$ バイトのキーストリームを観測すれば確率 2/3 以上で成功し，汎用的な識別攻撃の中では最も強力な方法である [11].

2001 年に Mantin と Shamir は $Z_2 = 0$ がランダムな式と比べて 2 倍の確率 $(2/N)$ で成立することを示した [8]．さらに 2004 年には，Paul と Preneel によって $Z_1 \neq Z_2$ が成立する確率が高いことが示された [10]．このようなキーストリームの部分的な偏りは，複数の異なるセッション鍵から生じるキーストリームが観測可能であるという条件における識別攻撃で利用される．このような条件において，文献 [8] の方法は 2^8 バイト，文献 [10] の方法では 2^{25} バイトのキーストリームを観測すれば真性乱数系列とキーストリームを識別可能となる．

2.6 結言

本章では，ストリーム暗号の構造およびその安全性について述べた．また，本論文で対象とするストリーム暗号である RC4 のアルゴリズムを説明した．さらに，RC4 への安全性評価に関する従来研究を分類し紹介した．

第3章　RC4における初期状態からのセッション鍵復元手法

3.1　緒言

RC4では，内部状態復元攻撃 [8, 11, 12, 13, 14, 15] に関する議論は多数行われているが，鍵復元攻撃については議論が進んでいない．そこで，内部状態復元攻撃と組み合わせることで実現できる鍵復元攻撃について検討する．内部状態復元攻撃によって初期状態が復元できたとき，秘密鍵を復元するためには少なくともKSAの逆変換によって初期状態からセッション鍵を得る必要がある．秘密鍵とIVからセッション鍵が生成されたとき，セッション鍵生成処理が可逆であればセッション鍵とIVから秘密鍵が得られる．また，任意に作成した秘密鍵をセッション鍵として用いている場合は，セッション鍵から即座に秘密鍵が得られる．このようにして実行される鍵復元攻撃を防ぐためには，初期状態からセッション鍵を計算することが困難となるようにKSAを設計することが望まれる．

本章では，RC4のKSAの構造を解析し，少ない計算量で初期状態の一部からセッション鍵を復元する方法を与える．まず，内部状態の2個 (または1個) の要素からセッション鍵の1個の要素を推測する線形方程式を与える．この線形方程式はランダム探索より高確率でセッション鍵の要素を復元できる．次に，線形方程式を複数用いることで，内部状態の先頭の L 個の要素からセッション鍵の L 個全ての要素を復元する方法を提案する．この方法に要する計算量はKSAを1回実行する程度であり，容易に計算可能である．また，提案手法の成功確率を理論的に評価し，少ない計算量で復元可能なセッション鍵が暗号学的に無視できない割合で存在することを示す．

3.2　初期状態から容易に復元可能なセッション鍵

本節では全数探索より効率的に初期状態から復元できるセッション鍵について議論する．言いかえれば，そのような ℓ ビットのセッション鍵はKSAの処理を 2^{ℓ} 回実行する計算量より効率的に復元される．本研究では，特に，容易に復元可能なセッション鍵として以下のセッション鍵を定義する．

定義 1 初期状態から ℓ ビットのセッション鍵を復元するとき，復元に要する計算量がKSAの処理を高々 ℓ 回実行する計算量であるならば，そのようなセッション鍵はinvertible-keyと呼ばれる．

さらに議論の単純化のために，必要な初期状態のデータが小さいinvertible-keyとして以下のセッション鍵を定義する．

定義 2 ℓ ビットの invertible-key が初期状態の ℓ ビットから復元されるならば，そのような invertible-key は non-expanded invertible-key と呼ばれる．

non-expanded invertible-key が大きな割合で存在することは，少ない計算量で初期状態の一部からセッション鍵を復元できる確率が高くなるため，KSA にとって望ましくない．以後の議論では RC4 の KSA を解析し，non-expanded invertible-key となるセッション鍵の割合を求める．

3.3 線形方程式

本節では，RC4 の KSA における内部状態の撹拌が不十分であることに注目し，セッション鍵の要素と初期状態の要素間で高確率で成立する線形方程式を与える．

KSA の時刻 t における内部状態は 2 つの要素 $S^*_{t-1}[i^*_{t-1}]$，$S^*_{t-1}[j^*_t]$ をスワップすることで更新される．式 (2.10) より $S^*_{t-1}[i^*_{t-1}] = S^*_{t-1}[t-1]$ となる．このとき，$S^*_t[t-1]$ を除く全ての要素において，$x \neq j^*_t$ を満たしていれば $S^*_t[x] = S^*_{t-1}[x]$ が成立する．したがって，i^*_{t-1} に参照されない内部状態の要素は，ほとんどの要素で値は変化しない．この性質を一般化し，さらにその性質から作られた関係式の成立確率を以下の補題で与える．

補題 1 t_s と t_e を $t_s < t_e$ を満たす時刻とする．時刻 $t = t_s+1, t_s+2, \ldots, t_e$ において常に i^*_{t-1} が参照しない内部状態要素のインデックスの集合を $X = \{0, 1, \ldots, N-1\} \backslash \{t_s, t_s+1, \ldots, t_e-1\}$ とする．さらに，X から任意の r 個の要素を選択して作られる部分集合を B とする．時刻 $t = t_s+1, t_s+2, \ldots, t_e$ において j^*_t が一様でランダムな変数と仮定したとき，$S^*_{t_e}[b] = S^*_{t_s}[b]$ for $\forall b \in B$ は確率 $((N-r)/N)^{t_e-t_s}$ で成立する．

(証明) 時刻 t における内部状態の変化は $S^*_{t-1}[i^*_{t-1}]$ と $S^*_{t-1}[j^*_t]$ の値をスワップすることのみで生じる．式 (2.10) から $i^*_{t-1} = t-1$ が成立するため，KSA の時刻 $t = t_s+1, t_s+2, \ldots, t_e$ では常に $i^*_{t-1} \neq b$ for $\forall b \in B(\subset X)$ が成立する．もし KSA の時刻 $t = t_s+1, t_s+2, \ldots, t_e$ で常に $j^*_t \neq b$ for $\forall b \in B$ を満たすならば，$S^*_{t-1}[b]$ for $\forall b \in B$ はその期間で変化しないため，$S^*_{t_e}[b] = S^*_{t_s}[b]$ for $\forall b \in B$ が成立する．時刻 $t = t_s+1, t_s+2, \ldots, t_e$ で j^*_t が一様でランダムな変数と仮定すると，$S^*_{t_e}[b] = S^*_{t_s}[b]$ for $\forall b \in B$ は確率 $((N-r)/N)^{t_e-t_s}$ で成立する．(証明終)

補題 1 は任意の期間 (時刻 $t = t_s+1, t_s+2, \ldots, t_e$) において任意の個数 ($r$ 個) の内部状態の要素が変化しない確率を導出するときに用いることができる．本節で提案する線形方程式の成立確率や 3.4 節で提案する初期状態からセッション鍵を復元する方法の成功確率を計算する際にこの補題が使われる．

内部状態の要素が変化し難いという性質を考慮して，初期状態の 2 個 (または 1 個) の要素からセッション鍵の 1 個の要素を推測する以下の線形方程式を得る．

$$K[x \bmod L] = \begin{cases} S_0[x] & \text{if } x = 0 \\ (S_0[x] - S_0[x-1] - x) \bmod N & \text{if } x = 1, 2, \ldots, N-1 \end{cases} \tag{3.1}$$

式 (3.1) が成立する条件を以下の補題で与える．

補題 2 $t' \in \{1, 2, \ldots, N\}$ を $t' = x + 1$ を満たす時刻とする．$t' = 1$ の場合，式 (3.8) が成立するならば式 (3.1) が成立する．$t' = 2, 3, \ldots, N$ の場合，式 (3.2)–(3.8) が成立するならば式 (3.1) が成立する．

$$S^*_{t'-2}[i^*_{t'-1}] = i^*_{t'-1} \tag{3.2}$$

$$S^*_{t'-2}[j^*_{t'-1}] = j^*_{t'-1} \tag{3.3}$$

$$S^*_{t'-1}[j^*_{t'}] = j^*_{t'} \tag{3.4}$$

$$j^*_{t'-1} \neq i^*_{t'-1} \tag{3.5}$$

$$j^*_{t'} \neq i^*_{t'-2} \tag{3.6}$$

$$S^*_N[t'-2] = S^*_{t'}[t'-2] \tag{3.7}$$

$$S^*_N[t'-1] = S^*_{t'}[t'-1] \tag{3.8}$$

(証明) $t' = 2, 3, \ldots, N$ の場合，式 (3.2)–(3.8) を満たすならば式 (3.1) が成立することを示す．式 (3.2)，(3.5) が満たされるとき，式 (2.8) より $S^*_{t'-1}[i^*_{t'-1}] = i^*_{t'-1}$ が成立する．また，式 (2.10) より $i^*_{t'-1} = t' - 1$ が成立する．このとき，式 (2.7) は以下のように書き直すことができる．

$$K[(t'-1) \bmod L] = (j^*_{t'} - j^*_{t'-1} - (t'-1)) \bmod N \tag{3.9}$$

式 (2.8)，(2.10)，(3.3)，(3.4) より，$S^*_{t'-1}[t'-2] = j^*_{t'-1}$ と $S^*_{t'}[t'-1] = j^*_{t'}$ が成立する．さらに，式 (2.8)，(2.10)，(3.6) より $S^*_{t'}[t'-2] = S^*_{t'-1}[t'-2] = j^*_{t'-1}$ が成立する．ゆえに，$S_0 = S^*_N$ と式 (3.7)，(3.8) より 2 個の等式 $S_0[t'-2] = S^*_N[t'-2] = j^*_{t'-1}$ と $S_0[t'-1] = S^*_N[t'-1] = j^*_{t'}$ が導出される．$x = t' - 1$ と置き直すと，これらの等式と式 (3.9) より式 (3.1) が成立する．図 3.1 は $t' = 2, 3, \ldots, N$ の場合に式 (3.1) が成立するための条件を図示したものである．

次に $t' = 1$ の場合に式 (3.1) が成立する条件について考える．$t' = 1$ のとき，式 (2.5) より $S^*_{t'-1}[i^*_{t'-1}] = S^*_0[i^*_{t'-1}] = i^*_{t'-1}$ となるため，式 (3.2)，(3.5) の条件が必要ない．さらに，式 (2.6) から $j^*_{t'-1} = j^*_0 = 0$ が成立するため，式 (3.3)，(3.6)，(3.7) の条件も必要ない．また，式 (2.5) より $S^*_{t'-1}[j^*_{t'}] = S^*_0[j^*_{t'}] = j^*_{t'}$ となるため，式 (3.4) は必要ない．したがって，$t' = 1$ のときは式 (3.8) が成立するならば式 (3.1) が成立する． (証明終)

補題 2 の条件が成立する確率を補題 3 で与える．

補題 3 E_{S1} を補題 2 の条件が満たされるという事象とする．時刻 $t = 1, 2, \ldots, N$ において j^*_t が一様でランダムな変数と仮定したとき，E_{S1} の生起確率は

$$\Pr(E_{S1}) = P_1 \quad \text{if } x = 0 \tag{3.10}$$

$$\Pr(E_{S1}) = P_2 \cdot P_3 \cdot P_4 \cdot P_5 \quad \text{if } x = 1, 2, \ldots, N-1 \tag{3.11}$$

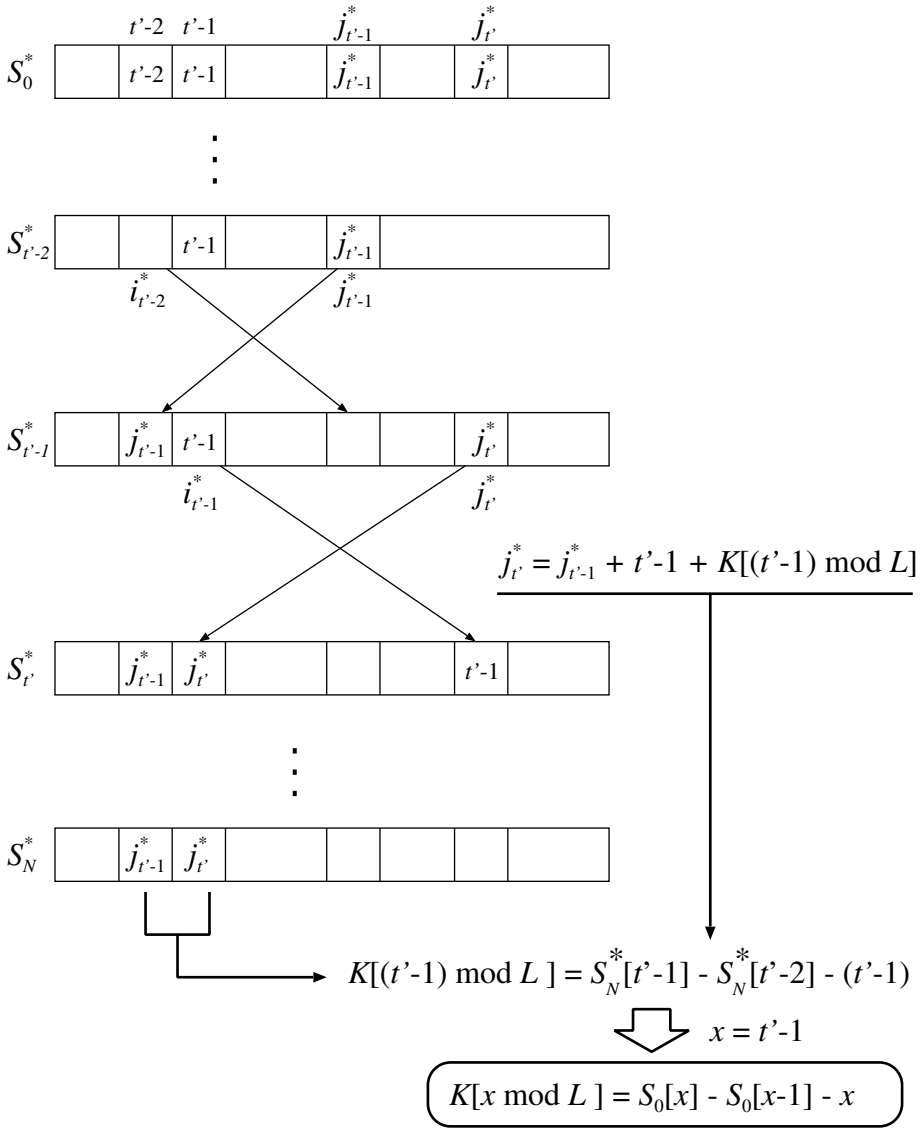


図 3.1　$t' = 2, 3, \ldots, N$ の場合に式 (3.1) が成立する条件

となる．ここで,

$$
\begin{aligned}
P_1 &= \left(\frac{N-1}{N}\right)^{N-1} \\
P_2 &= \left(\frac{N-1}{N}\right)^{x-1} \\
P_3 &= \frac{x}{N}\cdot\frac{1}{N} + \frac{N-x-1}{N}\cdot\left(\frac{N-1}{N}\right)^{x-1} \\
P_4 &= \frac{x-1}{N}\cdot\frac{1}{N} + \frac{N-x}{N}\cdot\left(\frac{N-1}{N}\right)^{x} \\
P_5 &= \left(\frac{N-2}{N}\right)^{N-x-1}
\end{aligned}
$$

である．

(証明) $t' = 2, 3, \ldots, N$ $(x = 1, 2, \ldots, N-1)$ の場合，$\Pr(E_{S1})$ は式 (3.2)–(3.8) の成立確率の積で与えられる．

第一に，式 (3.2) の成立確率を求める．式 (2.5) より $S_0^*[i_{t'-1}^*] = i_{t'-1}^*$ が満たされるため，式 (3.2) の成立確率は $S_{t'-2}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*]$ の成立確率と等しい．式 (2.10) より時刻 $t = 1, 2, \ldots, t'-2$ では常に $i_{t-1}^* \neq i_{t'-1}^* = t'-1$ となるため，$S_{t'-2}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*]$ が成立する確率は補題 1 によって以下のように計算できる．

$$P_2 = \left(\frac{N-1}{N}\right)^{t'-2} = \left(\frac{N-1}{N}\right)^{x-1} \tag{3.12}$$

第二に，式 (3.3)，(3.5) が共に満たされる確率を求める．式 (3.5) の条件は $j_{t'-1}^* \leq i_{t'-1}^* - 1$ と $j_{t'-1}^* \geq i_{t'-1}^* + 1$ の 2 つの場合に分けて考えることができる．それぞれの場合における式 (3.3) の成立確率から式 (3.3)，(3.5) が共に満たされる確率を得ることができる．$j_{t'-1}^*$ が一様でランダムな変数であるとき，$j_{t'-1}^* \leq i_{t'-1}^* - 1$ の生起確率は $(t'-1)/N$ である．$j_{t'-1}^* \leq i_{t'-1}^* - 1$ を満たしている場合，$S_{t'-2}^*[j_{t'-1}^*]$ は時刻 $t = j_{t'-1}^* + 1$ で既にスワップされており，その値はランダム化されている．ゆえに，$j_{t'-1}^* \leq i_{t'-1}^* - 1$ が満たされている場合に $S_{t'-2}^*[j_{t'-1}^*] = j_{t'-1}^*$ が成立する確率は $1/N$ とみなせる．また，$j_{t'-1}^* \geq i_{t'-1}^* + 1$ の生起確率は $(N-t')/N$ となる．$j_{t'-1}^* \geq i_{t'-1}^* + 1$ を満たしている場合，時刻 $t = 1, 2, \ldots, t'-2$ で常に $i_{t-1}^* \neq j_{t'-1}^*$ が成立する．ゆえに，$j_{t'-1}^* \geq i_{t'-1}^* + 1$ が満たされている場合に $S_{t'-2}^*[j_{t'-1}^*] = j_{t'-1}^* = S_0^*[j_{t'-1}^*]$ が成立する確率は補題 1 によって $((N-1)/N)^{t'-2}$ のように計算できる．よって，式 (3.3)，(3.5) が共に満たされる確率は以下のように与えられる．

$$\begin{aligned} P_3 &= \frac{t'-1}{N} \cdot \frac{1}{N} + \frac{N-t'}{N} \cdot \left(\frac{N-1}{N}\right)^{t'-2} \\ &= \frac{x}{N} \cdot \frac{1}{N} + \frac{N-x+1}{N} \cdot \left(\frac{N-1}{N}\right)^{x-1} \end{aligned} \tag{3.13}$$

同様にして，式 (3.4)，(3.6) が共に満たされる確率は以下のように与えられる．

$$\begin{aligned} P_4 &= \frac{t'-2}{N} \cdot \frac{1}{N} + \frac{N-t'+1}{N} \cdot \left(\frac{N-1}{N}\right)^{t'-1} \\ &= \frac{x-1}{N} \cdot \frac{1}{N} + \frac{N-x}{N} \cdot \left(\frac{N-1}{N}\right)^{x} \end{aligned} \tag{3.14}$$

第三に，式 (3.7)，(3.8) が共に満たされる確率を求める．時刻 $t = t'+1, t'+2, \ldots, N$ において常に $i_{t-1}^* \neq i_{t'-2}^*$ と $i_{t-1}^* \neq i_{t'-1}^*$ が成立するため，式 (3.7)，(3.8) が共に満たされる確率は補題 1 によって以下のように計算できる．

$$P_5 = \left(\frac{N-2}{N}\right)^{N-t'} = \left(\frac{N-2}{N}\right)^{N-x+1} \tag{3.15}$$

したがって，$x = 1, 2, \ldots, N-1$ のとき，$\Pr(E_{S1})$ は式 (3.12)–(3.15) の積によって式 (3.11) のように与えられる．

さらに $t' = 1$ $(x = 0)$ の場合の E_{S1} の生起確率について議論する．この確率は式 (3.8) の成立確率によって与えられる．時刻 $t = t' + 1, t' + 2, \ldots, N$ において常に $i^*_{t-1} \neq i^*_{t'-1}$ が成立するため，式 (3.8) の成立確率は補題 1 によって $((N-1)/N)^{N-1} = P_1$ のように計算できる．したがって，$x = 0$ のとき，$\Pr(E_{S1})$ は式 (3.10) によって与えられる．　(証明終)

提案した線形方程式の成立確率を定理 1 で与える．

定理 1 E_{T1} を式 (3.1) が成立しているという事象とする．E_{S1} が生じない場合の E_{T1} の生起確率を $1/N$ と仮定する．このとき，E_{T1} の生起確率は

$$\frac{1}{N} \cdot \Big((N-1) \cdot \Pr(E_{S1}) + 1 \Big) \tag{3.16}$$

となる．

(証明) 補題 2 より，E_{S1} が生じている場合の E_{T1} の生起確率は 1 である．式 (3.1) が偶然成立する確率は $1/2^n = 1/N$ であることから，E_{S1} が生じない場合の E_{T1} の生起確率を $1/N$ と仮定する．このとき，E_{T1} の生起確率は以下のように与えられる．

$$\begin{aligned} \Pr(E_{T1}) &= \Pr(E_{T1}|E_{S1}) \cdot \Pr(E_{S1}) + \Pr(E_{T1}|\overline{E_{S1}}) \cdot \Pr(\overline{E_{S1}}) \\ &= 1 \cdot \Pr(E_{S1}) + \frac{1}{N} \cdot (1 - \Pr(E_{S1})) \\ &= \frac{1}{N} \Big((N-1) \cdot \Pr(E_{S1}) + 1 \Big) \end{aligned}$$

(証明終)

式 (3.1) が成立するとき，セッション鍵の 1 個の要素 $K[x \bmod L]$ は初期状態の 2 個 (または 1 個) の要素 $S_0[x]$ と $S_0[x-1]$(または $S_0[x]$ のみ) から復元される．定理 1 より，式 (3.1) によって $K[x \bmod L]$ を復元できる確率は，ランダム探索で $K[x \bmod L]$ を得られる確率と比べて $(N-1) \cdot \Pr(E_{S1}) + 1$ 倍大きいことが分かる．さらに，補題 3 より初期状態の要素のインデックス x が小さいほど式 (3.1) によって $K[x \bmod L]$ を復元できる確率が大きくなることが確認できる．

3.4　初期状態からのセッション鍵復元

本節では，線形方程式を複数利用することで，ℓ ビットのセッション鍵を初期状態の先頭の ℓ ビットから復元する方法を与える．提案手法に要する計算量は KSA を 1 回実行する程度であるため，この方法によって復元できるセッション鍵は non-expanded invertible-key となる．しかしながら，他の方法によって復元可能な non-expanded invertible-key が存在をしないことを証明できないため，本研究で得られるのはセッション鍵全体に対する non-expanded invertible-key の割合の下界である．

まず初めに，提案手法において，線形方程式，すなわち式 (3.1) をどのようなパラメータで用いるかについて議論する．セッション鍵の全ての要素 $K[0], K[1], \ldots, K[L-1]$ を復元するためには，異なる x に関する線形方程式を少なくとも L 個用いる必要がある．多くの場合，セッション鍵の 1 個の要素を復元するために線形方程式は内部状態の 2 個の要素を必要とする．すなわち，セッション鍵を復元するためには内部状態の $2L$ 個の要素が必要となってしまう．しかしながら，ある特別な場合には必要な内部状態の要素数を減少させることが可能である．選択する線形方程式が $x = x'$ と $x = x' + 1$ のように x が隣り合ったものであるとき，必要な内部状態の要素は 1 個減少する．なぜなら，必要となる内部状態の要素が 2 つの線形方程式で 1 個分重複するからである．さらに，$x = 0$ における線形方程式では必要な内部状態の要素は 1 個だけで良い．ゆえに，$x = 0, 1, \ldots, L-1$ で線形方程式を用いることで必要な内部状態の要素を L 個まで削減することができる．このように線形方程式を選択した場合，ℓ ビットのセッション鍵は初期状態の $L \cdot n$ $(\sim \ell)$ ビットの初期状態から L 個の線形方程式を計算するだけで復元されることになる．

上記の議論をふまえて，以下に提案手法のアルゴリズムを与える．

アルゴリズム 1

入力: $S_0[0], S_0[1], \ldots, S_0[L-1]$.

出力: $K[0], K[1], \ldots, K[L-1]$.

Step 1: x に 0 をセットする．

Step 2: 式 (3.1) によって $K[x \bmod L]$ を計算する．

Step 3: x を $x = x + 1$ のように更新する．

Step 4: もし $x \neq L$ ならば Step 2 へ，そうでないなら $K[0], K[1], \ldots, K[L-1]$ を出力して終了する．

(アルゴリズム終)

この方法はセッション鍵を復元するために初期状態の先頭の L 個の要素を必要とするため，内部状態復元攻撃によってキーストリームからそれらの要素を得られたときに実行されることになる．また，復元したセッション鍵が正しいか否かは，そのセッション鍵から実際にキーストリームを生成し，それを観測した正しいキーストリームと比較することで確認できる．

ここで，提案手法の計算量について議論する．N を法としたときの加算と減算の計算量が等しいと仮定する．このとき，式 (2.7) と線形方程式の計算量は等しくなる．また，KSA では

式 (2.7)–(2.9) を N 回計算している．L と N は $L \leq N$ の関係を満たすため，提案手法で実行される L 回の線形方程式の演算は KSA を 1 回実行する際の計算量より小さくなる．

次に，提案手法によってセッション鍵を復元できる確率について議論する．その確率の正確な値は $x = 0, 1, \ldots, L-1$ における式 (3.1) の成立確率の積をとるだけでは得られない．なぜなら，それぞれの線形方程式で補題 2 の条件が重複する部分があるため，$x = 0, 1, \ldots, L-1$ の全てにおいて式 (3.1) が成立している確率は単純にそれぞれの成立確率の積をとった値より大きくなるからである．したがって，提案手法によってセッション鍵を復元できる確率を得るために，$x = 0, 1, \ldots, L-1$ の全てにおいて式 (3.1) が成立するための条件を再整理し，以下のように与える．

補題 4 時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17)，(3.18) が成立しており，なおかつ式 (3.19) が成立しているならば，$x = 0, 1, \ldots, L-1$ の全てにおいて式 (3.1) は成立する．

$$j_t^* = c \text{ for } \exists c \in \{i_{t-1}^*, L, L+1, \ldots, N-1\} \tag{3.17}$$

$$S_{t-1}^*[j_t^*] = j_t^* \tag{3.18}$$

$$S_N^*[t'-1] = S_L^*[t'-1] \text{ for } \forall t' \in \{1, 2, \ldots, L\} \tag{3.19}$$

(証明) 時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17)，(3.18) が成立しており，なおかつ式 (3.19) が成立しているときに $x = 0, 1, \ldots, L-1$ の全てにおいて式 (3.1) が成立することを示す．時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17) が成立しているとき，$S_{t'-1}^*[i_{t'-1}^*] = S_0^*[i_{t'-1}^*] = i_{t'-1}^* = t'-1$ for $\forall t' \in \{1, 2, \ldots, L\}$ が成立する．このとき，式 (3.9) は $t' = 1, 2, \ldots, L$ で常に満たされる．時刻 $t = 1, 2, \ldots, L$ で常に式 (3.18) が満たされているとき，$S_{t'}^*[i_{t'-1}^*] = S_{t'}^*[t'-1] = j_{t'}^*$ for $\forall t' \in \{1, 2, \ldots, L\}$ が成立する．さらに，時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17) が満たされているとき，$S_L^*[t'-1] = S_{t'}^*[t'-1] = j_{t'}^*$ for $\forall t' \in \{1, 2, \ldots, L\}$ が成立する．ゆえに，$S_0 = S_N^*$ と式 (3.19) から L 個の等式 $S_0[t'-1] = S_N^*[t'-1] = j_{t'}^*$ for $\forall t' \in \{1, 2, \ldots, L\}$ が導出される．$x = t'-1$ と置き直すと，これらの等式と式 (3.9) より $x = 0, 1, \ldots, L-1$ の全てにおいて式 (3.1) が成立する． (証明終)

補題 4 の条件が成立する確率を補題 5 で与える．

補題 5 E_{S2} を補題 4 の条件が満たされるという事象とする．時刻 $t = 1, 2, \ldots, N$ において j_t^* が一様でランダムな変数と仮定したとき，E_{S2} の生起確率は

$$\Pr(E_{S2}) = P_6 \cdot P_7 \tag{3.20}$$

となる．ここで，

$$P_6 = \left(\frac{N-L+1}{N}\right)^L \cdot \left(\frac{N-1}{N}\right)^{\frac{L(L-1)}{2}}$$

$$P_7 = \left(\frac{N-L}{N}\right)^{N-L}$$

である．

(証明) $\Pr(E_{S2})$ は時刻 $t = 1, 2, \ldots, L$ で式 (3.17)，(3.18) が常に成立する確率と式 (3.19) が成立する確率の積で与えられる．

まず初めに，前者について議論する．時刻 $t = 1, 2, \ldots, L$ において j^*_{t-1} が一様でランダムな変数であるとき，時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17) が成立する確率は $((N - L + 1)/N)^L$ となる．時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17) が成立しているという条件における式 (3.18) の成立確率について考える．このとき，時刻 $t'' = 1, 2, \ldots, t - 1$ で常に $i^*_{t''-1} \neq j^*_t$ を満たし，さらに式 (2.5) より $S^*_0[j^*_t] = j^*_t$ が満たされるため，時刻 t における式 (3.18) の成立確率は補題 1 によって $((N-1)/N)^{t-1}$ のように計算できる．よって，時刻 $t = 1, 2, \ldots, L$ で常に式 (3.18) が成立する確率は $\prod_{t=1}^{L}((N-1)/N)^{t-1}$ のように得られる．したがって，時刻 $t = 1, 2, \ldots, L$ で常に式 (3.17)，(3.18) が成立する確率は以下のように与えられる．

$$\begin{aligned} P_6 &= \prod_{t=1}^{L} \frac{N-L+1}{N} \cdot \prod_{t=1}^{L} \Big(\frac{N-1}{N}\Big)^{t-1} \\ &= \Big(\frac{N-L+1}{N}\Big)^{L} \cdot \Big(\frac{N-1}{N}\Big)^{\frac{L(L-1)}{2}} \end{aligned} \tag{3.21}$$

次に，後者について議論する．時刻 $t = L + 1, L + 2, \ldots, N$ で常に $i^*_{t-1} \neq 0, 1, 2, \ldots, L - 1$ が成立するため，式 (3.19) の成立確率は補題 1 によって以下のように計算できる．

$$P_7 = \Big(\frac{N-L}{N}\Big)^{N-L} \tag{3.22}$$

したがって，$\Pr(E_{S2})$ は式 (3.21)，(3.22) の積によって式 (3.20) のように与えられる．

(証明終)

セッション鍵が提案手法によって復元される確率，すなわち $x = 0, 1, \ldots, L - 1$ の全てにおいて式 (3.1) が成立している確率を定理 2 で与える．

定理 2 E_{T2} は $x = 0, 1, \ldots, L - 1$ の全てにおいて式 (3.1) が成立しているという事象とする．E_{S2} が生じない場合の E_{T2} の生起確率を $1/N^L$ と仮定する．このとき，E_{T2} の生起確率は

$$\frac{1}{N^L} \cdot \Big((N^L - 1) \cdot \Pr(E_{S2}) + 1\Big) \tag{3.23}$$

となる．

(証明) 補題 4 より，E_{S2} が生じている場合の E_{T2} の生起確率は 1 である．$x = 0, 1, \ldots, L - 1$ の全てにおいて式 (3.1) が偶然成立する確率は $(1/2^n)^L = 1/N^L$ であることから，E_{S2} が生じ

ない場合の E_{T2} の生起確率を $1/N^L$ と仮定する．このとき，E_{T2} の生起確率は以下のように与えられる．

$$\begin{aligned}\Pr(E_{T2}) &= \Pr(E_{T2}|E_{S2}) \cdot \Pr(E_{S2}) + \Pr(E_{T2}|\overline{E_{S2}}) \cdot \Pr(\overline{E_{S2}}) \\ &= 1 \cdot \Pr(E_{S2}) + \frac{1}{N^L} \cdot (1 - \Pr(E_{S2})) \\ &= \frac{1}{N^L} \cdot \Big((N^L - 1) \cdot \Pr(E_{S2}) + 1 \Big)\end{aligned}$$

(証明終)

式 (3.20)，(3.23) から提案手法の成功確率を計算することができる．もしセッション鍵がランダムに選択されるならば，式 (3.20)，(3.23) から提案手法によって復元可能なセッション鍵の割合を計算できる．実用的なパラメータとして $\ell = 128$ と $n = 8$ を選んだとき，提案手法で復元できるセッション鍵の割合は $1/2^{24.4}$ となる．したがって，このようなパラメータにおける non-expanded invertible-key の割合の下界は $1/2^{24.4}$ となる．この割合は暗号学的には無視できないほど大きい．

3.5 計算機実験

本章の解析の妥当性を検証するために計算機実験を行う．実験対象は 2^{40} 個のセッション鍵とその初期状態であり，セッション鍵のパラメータは $\ell = 128$，$n = 8$ とする．

第一の実験では，全ての x について線形方程式を用いて内部状態の要素からセッション鍵の要素の復元を試み，復元に成功した鍵の割合を得ることで線形方程式の成立確率の実験値を得る．図 3.2 は線形方程式の成立確率の実験値と理論値を x 毎に示したものである．図より，理論値は実験値を良く近似できていることが確認できる．

第二の実験では，提案手法を用いて実際に初期状態の先頭の ℓ ビットからセッション鍵の復元を試みる．実験の結果，2^{40} 個のセッション鍵から $47,286$ $(\approx 2^{15.5})$ 個の鍵が復元できた．復元できたセッション鍵の割合は $1/2^{24.5}$ $(= 2^{15.5}/2^{40})$ となり，それは理論値 $(1/2^{24.4})$ とほぼ同じ値であることが確認できる．また，実験によって得られた，提案手法によって復元できるセッション鍵とその初期状態及びポインタの例を表 3.1 に示す．

3.6 結言

本章では少ない計算量で RC4 の初期状態の一部から復元できるセッション鍵が無視できない割合で存在することを示した．まず，初期状態から容易に復元できるセッション鍵として non-expanded invertible-key を定義した．そして，初期状態の 2 個 (または 1 個) の要素からセッション鍵の 1 個の要素をランダム探索より高確率で推測する線形方程式を与えた．さら

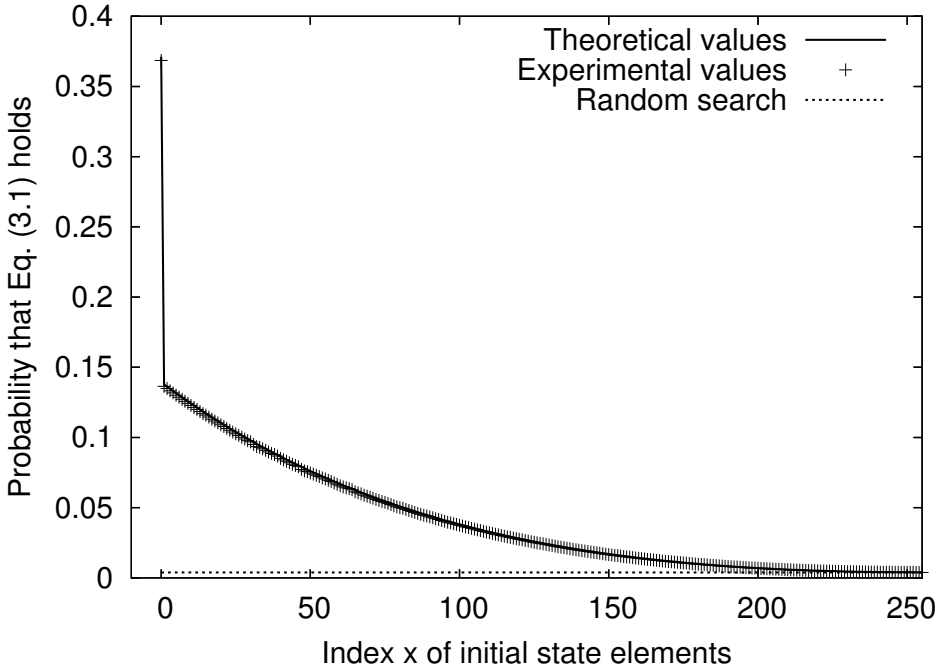


図 3.2　式 (3.1) の成立確率

表 3.1　提案手法によって復元できるセッション鍵 K とその初期状態 S_0 $(= S^*_{256})$ およびポインタ j^*_t の例

y	$K[y]$															
0–15	172	78	82	235	33	244	57	20	242	200	222	185	94	52	227	221

x	$S_0[x](= S^*_{256}[x])$															
0–15	172	251	79	61	98	91	154	181	175	128	104	44	150	215	200	180

t	j^*_t															
1–16	172	251	79	61	98	91	154	181	175	128	104	44	150	215	200	180
17–32	112	207	51	49	102	111	190	233	243	212	204	160	26	107	108	104
33–48	52	163	23	37	106	129	224	27	53	38	46	18	123	220	233	245
49–64	209	50	151	148	213	241	96	171	213	214	238	226	124	179	212	240
65–80	220	107	255	45	146	203	74	165	223	240	20	24	194	67	116	83
81–96	79	238	146	127	244	61	204	55	129	162	218	152	82	227	36	96
97–112	107	26	112	190	67	156	233	100	117	166	168	192	60	221	46	32
113–128	208	143	83	177	32	51	226	109	215	24	112	52	206	127	224	58
129–144	239	149	105	215	124	245	180	79	201	26	130	198	176	113	226	48
145–160	108	75	239	109	160	236	49	88	79	176	148	232	171	124	253	121
161–176	55	38	209	221	162	221	127	58	80	193	73	103	197	166	55	28
177–192	97	34	38	20	187	182	246	193	107	236	132	195	221	206	20	176
193–208	187	192	94	207	180	168	108	71	71	151	63	61	241	242	146	245
209–224	165	76	112	46	141	185	43	194	140	45	101	249	151	135	72	109
225–240	151	198	166	238	243	204	235	230	115	79	23	182	205	238	190	237
241–256	226	134	165	116	233	99	163	174	152	59	19	205	39	249	218	249

に，L 個の線形方程式を用いて初期状態の先頭の L 個の要素から ℓ ビットのセッション鍵 (セッション鍵の L 個の要素) を復元する方法を提案した．提案手法に要する計算量は KSA を 1 回実行するより小さく，提案手法で復元できるセッション鍵は non-expanded invertible-key とな

る．128 ビットのセッション鍵をランダムに選択するとき，提案手法は $1/2^{24.4}$ の割合の鍵を復元できることから，RC4 には少なくとも $1/2^{24.4}$ の割合の non-expanded invertible-key が存在することが示された．したがって，初期状態の先頭の 128 ビットを復元する攻撃が提案されたならば，128 ビットのセッション鍵は高確率で復元できることが明らかになった．

第4章　RC4に対する内部状態復元攻撃の効率化

4.1　緒言

内部状態復元攻撃はPRGAの構造を利用してキーストリームから初期状態を効率的に復元する攻撃である．内部状態復元攻撃はKSAの構造に依存しないため，秘密鍵やセッション鍵長に依存せずに攻撃に要する計算量が決定される．RC4のような可変長の鍵を用いるアルゴリズムの場合，解読法が有効な秘密鍵長の上限を評価できるという意味においても内部状態復元攻撃に関する研究は重要である．

RC4に対する内部状態復元攻撃は多数提案されているが [8, 11, 12, 13, 14, 15]，任意の初期状態に対して実行可能という意味で汎用的な攻撃では1998年にKnudsenらが提案した方法 [12] が最も効率的である．Knudsenらの方法はキーストリームと内部状態要素間で関係式を作り，その関係式を用いる木探索アルゴリズムによって効率的に初期状態を復元する．通常使われる $n = 8$ のRC4に対して，Knudsenらの方法では256個の初期状態の要素のうち100個が事前に分かっているという前提において，2^{30} の計算量で復元可能な初期状態の存在が示されている．

本章では，Knudsenらの方法を改良し，さらに効率的に初期状態を復元できる方法を提案する．Knudsenらの方法における木探索アルゴリズムは必要なキーストリームのサイズを小さくすることを重視したために余分な計算が必要となる．Knudsenらの方法で必要なキーストリームサイズは 2^8 バイト程度であり十分小さく，それを数倍程度に増加させたとしても大きなデメリットとならない．そこで，攻撃に必要なキーストリームサイズをKnudsenらの方法の2倍程度に増加させる代わりに，余分な計算を排除できるような木探索アルゴリズムを与える．また，初期状態の要素のうち73個が事前に分かっているという前提において，提案手法によって 2^{20} の計算量で復元可能な初期状態が存在することを計算機実験により示す．

4.2　Knudsenらの内部状態復元攻撃

初期状態を復元するための最も単純な方法は，初期状態に対する全数探索である．2^n 個ある内部状態の要素は全て異なり，それぞれに n ビットの値が割り当てられているため，初期状態を全数探索するのに必要な計算量は $2^n!$ になる．Knudsenらは時刻 t においてキーストリーム Z_t と内部状態の3個の要素間で関係式を作り，その関係式を用いる木探索アルゴリズムによって効率的に初期状態を復元する方法を提案した．Knudsenらの方法は必要な計算量を $\sqrt{2^n!}$ 程度まで削減できる．しかしながら，Knudsenの方法でもワード長 n が大きくなると計算量が膨大になってしまい，$n > 5$ の場合では現実的な時間では実行が困難となる．n が大きな場合

での評価では，初期状態の要素のうち k 個が事前に分かっているという仮定を置いた上で，残りの $2^n - k$ 個の要素を復元するための計算量を求める．文献 [12] では，$n = 8$ で初期状態の 256 個の要素のうち $k = 100$ 個が既知の場合，Knudsen らの方法と 2^{30} の計算量によって復元可能な初期状態が存在することが示されている．初期状態の 100 個の要素を最も単純な方法である総当たりで求めるとすると，そのパターン数は $256!/(256 - 100)! \approx 2^{767}$ となり，計算量は $2^{767} \times 2^{30} = 2^{797}$ となる．このような初期状態を 798 ビット以上の秘密鍵で作成した場合には内部状態復元攻撃が有効に働く．Knudsen らの方法に要する計算量は全数探索の計算量である $2^n! = 2^{1684}$ と比べて十分小さいが，通常使われる 128 ビットの秘密鍵などを考慮すると更に計算量を削減する必要がある．

Knudsen らの方法は時刻 t 毎の再帰的な関数を用いる木探索によって内部状態を探索する．探索は式 (2.13)–(2.15) を使って行われ，時刻 t で $S_{t-1}[i_t]$，$S_{t-1}[j_t]$ の要素に値が割り当てられていなければ，その要素に値をセットすることになる．また，キーストリーム Z_t と内部状態の要素の関係式を用いる枝刈りも実行される．関係式は式 (2.15)，(2.16) を変形して以下のように得られる．

$$S_t^{-1}[Z_t] = (S_{t-1}[i_t] + S_{t-1}[j_t]) \bmod N \tag{4.1}$$

ここで，$S_t^{-1}[Z_t]$ は Z_t が格納されている要素のインデックスである．式 (4.1) は 3 つの未知変数 $S_{t-1}[i_t]$，$S_{t-1}[j_t]$，$S_t^{-1}[Z_t]$ を含んでおり，未知変数のうち 2 つが既に割り当てられているならば式 (4.1) によって残り 1 つを計算できる．このとき，計算によって得られた値が既に他の内部状態の要素にセットされていたならば式 (4.1) に矛盾が生じる．式 (4.1) に矛盾が生じたときは，それまでにセットした値のいずれかが誤っていたと判断し，バックトラックして別の候補を探索する．また，未知変数の 3 つ全て割り当てられているときも，それらが式 (4.1) を満たすかをチェックすることで矛盾が生じていないかチェックできる．

上述の枝刈り処理を備えた Knudsen の方法のアルゴリズムを以下に示す．アルゴリズムは時刻 $t = 1$ の Step K1 から開始する．

アルゴリズム 2 (Knudsen らの方法)

Step K1 $S_{t-1}[i_t]$ に値が割り当てられているかチェックする．

(a) 割り当てられているならば，Step K2 へ．

(b) 割り当てられていないならば，まだ内部状態の要素にセットされていない値を $S_{t-1}[i_t]$ の候補として保持する．そして，候補を 1 つ選んで $S_{t-1}[i_t]$ にセットして Step K2 へ．もし探索に失敗してバックトラックしてきたならば，$S_{t-1}[i_t]$ に別の候補をセットして再度 Step K2 へ．すべての候補をセットしていた場合はバックトラックする．

Step K2 Z_t が内部状態の要素に割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_{t-1}[j_t]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t+1$ の Step K1 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，Step K3 へ．

Step K3 $S_{t-1}[j_t]$ に値が割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_t^{-1}[Z_t]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t+1$ の Step K1 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，まだ内部状態の要素にセットされていない値を $S_{t-1}[j_t]$ の候補として保持する．そして，候補を 1 つ選んで $S_{t-1}[j_t]$ にセットし，式 (4.1) によって $S_t^{-1}[Z_t]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t+1$ の Step K1 へ．もし矛盾が生じるか，探索に失敗してバックトラックしてきたならば，$S_{t-1}[j_t]$ に別の候補をセットして再度時刻 $t+1$ の Step K1 へ．すべての候補をセットしていた場合はバックトラックする．

(アルゴリズム終)

アルゴリズムは内部状態の全ての要素がキーストリームと矛盾せずにセットされたときに終了し，内部状態 S_t が得る．さらに，S_t に逆方向への PRGA の処理を実行して初期状態 S_0 を得る．逆方向への PRGA の処理に用いる式は，式 (2.13)–(2.15) を変形して以下のように与えられる．

$$i_{t-1} = (i_t - 1) \bmod N \tag{4.2}$$

$$j_{t-1} = (j_t - S_t[j_t]) \bmod N \tag{4.3}$$

$$S_{t-1}[x] = \begin{cases} S_t[j_t] & \text{if } x = i_t \\ S_t[i_t] & \text{if } x = j_t \\ S_t[x] & \text{if } x \neq i_t, j_t \end{cases} \tag{4.4}$$

Knudsen らの方法の有効性を検証するために計算機実験を行う．実験の条件を表 4.1 に示す．ワード長 $n = 8$ でランダムに $100,000$ 個の初期状態を選び，計算量の上限を 2^{20} としている．また，事前に初期状態の k 個の要素を知っている条件で実験を行っている．その結果，$100,000$ 個全ての初期状態を復元するためには $k = 207$，全体の半数の初期状態を復元するためには $k = 167$ にする必要があることが分かった．また，初期状態の復元に成功した最小の k は $k = 112$ となった．

表 4.1　計算機実験のパラメータ

ワード長	$n = 8$
解読対象	$100,000$ 個の初期状態
計算量の単位	内部状態の更新回数
解読成功の条件	2^{20} の計算量で内部状態を復元できた場合
事前に知っている初期状態の要素の数	先頭の k 個

Knudsen らの方法には二つの重要な処理がある．一つは式 (4.1) を用いる枝刈りの処理であり，計算量を減少させる効果がある．もう一つは，内部状態の要素に対して全数探索的な方法で値をセットする処理であり，それは Step K1-(b) と Step K3-(b) に含まれている．この処理によって計算量が増加し，$k < 112$ の条件では全ての初期状態で内部状態を復元する前に計算量の上限に達してしまった．Knudsen らの方法は時刻毎にポインタによって参照される内部状態の要素 $S_{t-1}[i_t]$，$S_{t-1}[j_t]$ の両方に対して全数探索的な方法で値をセットしていく．これは，それぞれの時刻においてキーストリームと関係のある内部状態の要素に可能な限り値をセットしていくことになるため，探索中に時刻 t が大きくなることを防ぐ効果がある．すなわち，時刻 t を深さとみなした場合において，Knudsen らの方法は幅優先指向探索 (BFOS : Breadth-First Oriented Search) と考えることができる．BFOS における探索では t が小さい際の Z_t のみ必要となるため，攻撃で用いるキーストリームのサイズを減少させる働きがある．計算機実験によって $n = 8$ で Knudsen らの方法に必要なキーストリームのサイズは 2^8 バイト程度であることを確認している．

4.3　効率的な内部状態復元攻撃

4.3.1　深さ優先指向探索の有効性

Knudsen らの方法は BFOS であり，時刻 t を進めるよりも内部状態の要素に値をセットしていくことを優先する．しかしながら，木探索アルゴリズムでは時刻 t 毎に得られるキーストリーム Z_t を用いる枝刈りによって計算量を削減しているため，計算量の面では時刻 t を進めることを優先する深さ優先指向探索 (DFOS : Depth-First Oriented Search) のほうが優れていると考えられる．そこで，各時刻において全数探索的な方法で値をセットする 2 つの要素 $S_{t-1}[i_t]$ と $S_{t-1}[j_t]$ のうち，$S_{t-1}[j_t]$ に関する処理を削除することで Knudsen らの方法を DFOS に変更することを提案する．具体的には Step K3-(b) において $S_{t-1}[j_t]$ に値をセットする処理を削除する．

$S_{t-1}[i_t]$ に値をセットする処理を削除できない理由を説明する．RC4 の PRGA の内部状態の更新はポインタ i_t，j_t のスワップによって行われている．i_t は既知であるので，j_t の値さえ

表 4.2　BFOS と DFOS の比較

復元できる初期状態の個数	BFOS の k の値 (Knudsen の方法)	DFOS の k の値 (4.3.1 節の方法)
1	112	82
$50,000$	167	112
$100,000$	207	154

把握できれば内部状態を更新していくことができる．式 (2.14) より，j_t を更新するためには $S_{t-1}[i_t]$ に値がセットされている必要があるため，$S_{t-1}[i_t]$ に値をセットする処理は削除することができない．

木探索アルゴリズムを DFOS に変更することの有効性を検証するために計算機実験を行う．実験の条件は Knudsen らの方法における計算機実験と同様に表 4.1 に従うとし，表 4.2 に実験結果を示す．DFOS による方法では $100,000$ 個全ての初期状態を復元するためには $k = 154$，全体の半数の初期状態を復元するためには $k = 112$ にする必要があることが分かった．また，初期状態の復元に成功した最小の k は $k = 82$ となった．したがって，DFOS による方法は BFOS より初期状態の既知の要素数が少なくても成功するため，効率的な方法であることが確認できる．また，木探索アルゴリズムを DFOS に変更したことで攻撃に要するキーストリームのサイズは増加したが，2^9 バイト程度までの増加におさまっていることを実験によって確認している．このサイズは Knudsen らの方法の 2 倍程度であるが，十分小さいために大きなデメリットとならないと考えられる．

4.3.2　アルゴリズム

前節では Knudsen らの方法の木探索アルゴリズムを DFOS に変更することで効率的な探索が可能となることについて議論した．本節では DFOS による探索に更に BCR(Backtracking for Candidate Reduction) という新たな効率的な処理を加えたアルゴリズムを与える．

アルゴリズム 3 (提案手法)

Step P1 $S_{t-1}[i_t]$ に値が割り当てられているかチェックする．

(a) 割り当てられているならば，Step P2 へ．

(b) 割り当てられていないならば，BCR の処理を実行する．BCR の処理で矛盾が生じた場合はバックトラックする．次に，まだ内部状態の要素にセットされていない値を $S_{t-1}[i_t]$ の候補として保持する．そして，候補を 1 つ選んで $S_{t-1}[i_t]$ にセットして Step P2 へ．もし探索に失敗してバックトラックしてきたならば，$S_{t-1}[i_t]$ に別の

候補をセットして再度 Step P2 へ．すべての候補をセットしていた場合はバックトラックする．

Step P2 Z_t が内部状態の要素に割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_{t-1}[j_t]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t+1$ の Step P1 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，Step P3 へ．

Step P3 $S_{t-1}[j_t]$ に値が割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_t^{-1}[Z_t]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t+1$ の Step P1 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，時刻 $t+1$ の Step P1 へ．

(アルゴリズム終)

アルゴリズム 4 (BCR)

Step B1 時刻 $1, 2, \cdots, t'-1$ において式 (4.1) の未知変数の 3 つ全てが矛盾せずに割り当てられている時刻 t' を調べる．Step B2 へ．

Step B2 t' をチェックし，$t' = t-1$ を満たしているならば BCR の処理を終了する．そうでないなら，$Z_{t'}$ が内部状態の要素に割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_{t'-1}[j_{t'}]$ を計算する．式 (4.1) に矛盾が生じなければ，$t' = t'+1$ のように更新し Step B2 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，Step B3 へ.

Step B3 $S_{t'-1}[j_{t'}]$ に値が割り当てられているかチェックする．

(a) 割り当てられているならば，式 (4.1) によって $S_{t'}^{-1}[Z_{t'}]$ を計算する．式 (4.1) に矛盾が生じなければ，時刻 $t'+1$ の Step B2 へ．もし矛盾が生じたならばバックトラックする．

(b) 割り当てられていないならば，時刻 $t'+1$ の Step B2 へ．

(アルゴリズム終)

提案手法の Knudsen らの方法からの変更点は，全数探索的な方法で $S_{t-1}[j_t]$ に値をセットする処理を Step P3-(b) から削除したことと，BCR という新しい処理を Step P1-(b) に加えたことである．$S_{t-1}[j_t]$ に値をセットする処理を削除したことから，$S_{t-1}[j_t]$ に値が割り当てられていない時刻が存在することになる．探索が進むにつれて内部状態の要素に新たに値がセットされていくことから，ある時刻において式 (4.1) によって計算できなかった $S_{t-1}[j_t]$ が，探索が進んだことで式 (4.1) によって値を計算できるようになる場合がある．BCR は，計算量が増加する Step P1-(b) の処理の前に，時刻 $t' = 1, 2, \ldots, t-1$ で式 (4.1) による枝刈りを再度実行することで計算量の削減を試みる処理と考えることができる．

4.4 計算機実験

本節では計算機実験によって提案手法と Knudsen らの方法を比較する．また BCR の処理の有効性を検証するために 4.3.1 節の方法とも比較する．実験ではそれぞれの方法について k に対応する攻撃の成功率を得る．実験条件は表 4.1 に従うとし，図 4.1 に実験結果を示す．実験より，全ての k において提案手法が Knudsen らの方法と比べて成功率が高くなっており効果的な方法であることが確認できる．さらに，提案手法は 4.3.1 節の方法と比べても同一の成功率における k を小さくなっており，BCR の処理が有効であることも確認できる．また，実験によって提案手法は 4.3.1 節の方法と比べて必要なキーストリームのサイズが 30 バイトから 150 バイト程度減少することを確認している．これは BCR の処理は過剰なキーストリームサイズを増加を抑えられることを意味している．

提案手法において初期状態の復元に成功した最小の k は $k = 73$ となった．$k = 73$ で復元できる初期状態について Knudsen らと同様の方法で合計の計算量を見積もる．73 個の要素を総当たりで求めるとすると，そのパターン数は $256!/(256-73)! \approx 2^{567}$ となり，計算量は $2^{20} \times 2^{567} = 2^{587}$ となる．これは 588 ビット以上の秘密鍵を用いてこの初期状態を作成した場合に，提案手法によって鍵全数探索攻撃より効率的にキーストリームから初期状態を復元可能であることを意味する．Knudsen らの方法では 798 ビット以上の秘密鍵を用いる際に有効となる初期状態が示されていたことから，提案手法は更に 200 ビット以上少ないサイズの秘密鍵に対して有効に働く場合があることが分かる．

4.5 結言

本章では，Knudsen らが提案した内部状態復元攻撃を改良し，効率的にキーストリームから初期状態を復元できる方法を与えた．まず初めに Knudsen らの木探索アルゴリズムを深さ優先指向探索に変更し，より多くのキーストリームを利用する効率的な探索が可能なように改

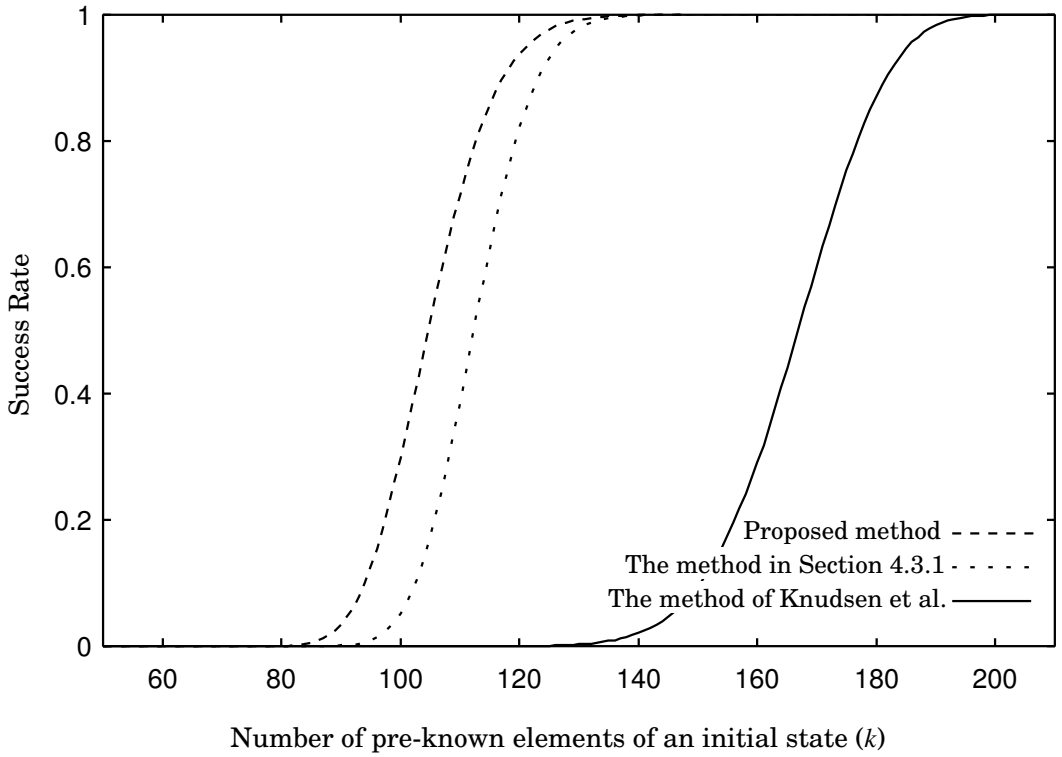


図 4.1　内部状態復元攻撃の成功率の比較

良した．次に，過去の時刻の枝刈り条件を再チェックする BCR という処理を追加することで枝刈りの機会を増加させ，探索に要する計算量を更に削減した．さらに，計算機実験によって初期状態の 73 個の要素が既知のときに提案手法を用いて 2^{20} の計算量で復元できる初期状態の存在を明らかにした．既知の要素が無い条件でこの初期状態を復元する場合，73 個の要素を総当たりで得るとするならば，復元に要する計算量は 2^{587} となる．Knudsen らの方法では 2^{797} の計算量で復元できる初期状態の存在が知られていたことから [12]，提案手法によって内部状態復元攻撃に対する RC4 の安全性が更に低下したといえる．

第5章　WEPにおける新しいweak IVと鍵復元攻撃への応用

5.1　緒言

WEP のセッション鍵はパケット毎に IV と秘密鍵の連結によって生成されている．そのため，セッション鍵は一部しか変化しない鍵となる．2001 年に Fluhrer らは WEP のような似たセッション鍵が生成されるという条件において，特定のパターンの IV が使われた場合にキーストリームと IV 及び秘密鍵の間に強い相関が生じることを示した．さらに，その相関を用いた鍵復元攻撃を提案しており，その攻撃は FMS 攻撃と呼ばれている [9, 17]．秘密鍵との間に相関が生じる IV は weak IV と呼ばれている．FMS 攻撃が提案されて以来，WEP に対する鍵復元攻撃が多数議論されており [21, 22, 23, 24, 25]，多くの weak IV が特定された．これらの weak IV は FMS 攻撃と同程度の秘密鍵との相関を持っている．

鍵復元攻撃を防ぐための技術として WEPplus が知られている [20]．WEPplus では，フィルタリングによって IV の中から weak IV を取り除くことで鍵復元攻撃を実行困難にする．この技術は WEP の実装を変更せずに導入できるため，多くの無線 LAN メーカによって採用されている．WEPplus によって安全性を保つためには，weak IV が新たに特定される度にフィルタリングパターンを更新する必要がある．2006 年に吉田らによって効果的なフィルタリングパターンが示され，全体の約 31.2%の IV を取り除くことで FMS 攻撃と同程度の秘密鍵との相関を持つ weak IV をほぼ全て取り除くことが可能となった [54]．

weak IV に関する従来の研究では，FMS 攻撃と同程度の秘密鍵との相関を持つ weak IV のみ注目されてきた．しかしながら，それらより相関が小さい weak でも多数集めることができれば秘密鍵の情報を得ることが可能であると考えられる．そこで本章では，比較的相関が小さい weak IV を利用した鍵復元攻撃について議論する．まず初めに RC4 の KSA を解析し，相関が小さい weak IV を特定する．特定する weak IV が IV 全体に占める割合は約 81%と大きく，多数の weak IV を攻撃に利用できる．この weak IV をフィルタリングによって取り除いてしまうと，IV 空間が極端に小さくなり実用上の問題が生じるため，WEPplus による対策はこの weak IV に対しては困難となると考えられる．次に，weak IV が持っている秘密鍵との相関の大きさが秘密鍵の値によって変化する場合があることに注目する．もし，何らかの方法で秘密鍵との相関が大きくなっている weak IV を選別できるならば，秘密鍵との相関が大きな weak IV を選択して用いることで鍵復元攻撃の成功確率を向上させることができる．そこで，鍵復元攻撃の処理中に生じる情報を利用して weak IV を選別する方法を与える．さらに，この選別法を用いた鍵復元攻撃を提案し，計算機実験によって有効性を確認する．提案する選別法は適用できる weak IV に制約が生じない．したがって，提案する選別法を用いた鍵復元攻撃は，

使われる可能性がある weak IV が占める空間は全体の 81%から変化しないため，WEPplus による対策が困難な方法といえる．

5.2 WEP におけるセッション鍵の生成

WEP におけるセッション鍵の生成について説明する．WEP のセッション鍵 K は秘密鍵 Rk と 24 ビットの IV を用いて以下のように生成される．

$$K = IV \| Rk \tag{5.1}$$

ここで，$\|$ ビットの連結である．IV はパケット毎に変更される公開値である．Rk は 40 ビットまたは 104 ビットが使われる．さらに WEP で使われている RC4 は $n = 8$ であるため，IV は $K[0], K[1], K[2]$，秘密鍵は $K[3], K[4], \ldots, K[L-1]$ にセットされる．

5.3 鍵復元攻撃の原理

5.3.1 攻撃方程式

鍵復元攻撃では，セッション鍵の要素 $K[a \bmod L]$ を

$$K[a \bmod L] = f(IV, Z_t) \tag{5.2}$$

のように IV とキーストリーム Z_t を入力とした攻撃方程式 f によって計算する．攻撃方程式から得られる $K[a \bmod L]$ の値を推測値と呼び $gval(a)$ と記述する．通常，推測値のインデックス a は IV によって決まる．IV が weak IV であるとき，攻撃方程式によって得られた推測値が $gval(a) = K[a \bmod L]$ を満たす確率はランダム探索より高確率となる．

攻撃方程式の成立確率は fortunate event という特別な事象の生起確率から計算できる．fortunate event を以下のように定義する．

定義 3 fortunate event が生じたとき，攻撃方程式 f は確率 1 で成立する．

fortunate event は KSA の時刻毎に N 個の subevent に分類でき，時刻 t の subevent を subevent(t) と定義する．

fortunate event の生起確率を $\alpha(a)$ とするとき，攻撃方程式の成立確率は以下の定理で与えられる．

定理 3 fortunate event の生起確率が $\alpha(a)$ ならば，攻撃方程式の成立確率は

$$\alpha(a) \cdot \frac{N-1}{N} + \frac{1}{N} \tag{5.3}$$

となる．

(証明) 確率 $\alpha(a)$ で fortunate event が生じたときには攻撃方程式は必ず成立する．また，確率 $1-\alpha(a)$ で fortunate event が生じない場合には，攻撃方程式は n ビット単位の関係式であるため，攻撃方程式が偶然成立する確率は $1/2^n = 1/N$ と仮定できる．このとき，攻撃方程式の成立確率の合計は以下のように与えられる．

$$\alpha(a) \cdot 1 + (1-\alpha(a)) \cdot \frac{1}{N} = \alpha(a) \cdot \frac{N-1}{N} + \frac{1}{N}$$

(証明終)

定理 3 より，攻撃方程式を用いて推測値 $gval(a)$ を求めた際に $K[a \bmod L]$ を復元できる確率は式 (5.3) によって与えられる．$K[a \bmod L]$ をランダム探索で得られる確率は $1/N$ であるため，攻撃方程式によって推測できる確率はそれより $\alpha(a) \cdot (N-1)/N$ 大きいことが分かる．このランダム探索からの偏差を攻撃方程式から生じる偏差と呼ぶ．

5.3.2 攻撃方程式を用いた統計的な鍵復元手法

鍵復元攻撃では多数の weak IV と Z_t のペアと攻撃方程式からセッション鍵の要素を統計的に復元する．ここで，$K[a \bmod L]$ を復元する場合における統計的な鍵復元手法について考える．$K[a \bmod L]$ を復元するためには，0 から $N-1$ までの N 個の $K[a \bmod L]$ の候補の中から正しい候補を識別する必要がある．鍵復元攻撃では，得られている全ての weak IV と Z_t のペアのそれぞれから攻撃方程式によって推測値 $gval(a)$ を計算する．推測値はそれぞれ $K[a \bmod L]$ の候補別にカウントされ，図 5.2 のような頻度表を作られる．$gval(a)$ が $K[a \bmod L]$ と一致しやすいという性質から，頻度表の中で最もカウント数が大きな候補を $K[a \bmod L]$ として推定する．

この統計的な鍵復元手法が成功する確率について解析する．$gval(a)$ による $K[a \bmod L]$ の推測が誤るとき，$gval(a)$ は $N-1$ 個の誤った候補に等しく分布すると仮定する．このとき，$gval(a)$ が $K[a \bmod L]$ と一致する確率を $p+q$ とし，誤った候補と一致する確率をそれぞれの候補で p になるとする．すなわち，q は攻撃方程式から生じる偏差となる．また，N_{IV} を $K[a \bmod L]$ に対応する weak IV の個数とする．このとき，正しい候補のカウント数は試行回数 N_{IV}，生起確率 $p+q$ の二項分布 $\mathrm{B}(N_{IV}, p+q)$，誤った候補のカウント数はそれぞれ試行回数 N_{IV}，生起確率 p の二項分布 $\mathrm{B}(N_{IV}, p)$ に従うとみなせる．p と q が十分小さいとき，正しい候補のカウント数の平均 E_T 及び標準偏差 σ_T，誤った候補の平均 E_F 及び標準偏差 σ_F は以下のように与えられる．

$$E_T = N_{IV} \cdot (p+q) \tag{5.4}$$

$$\sigma_T = \sqrt{N_{IV} \cdot (p+q) \cdot (1-(p+q))} \approx \sqrt{N_{IV} \cdot (p+q)} \tag{5.5}$$

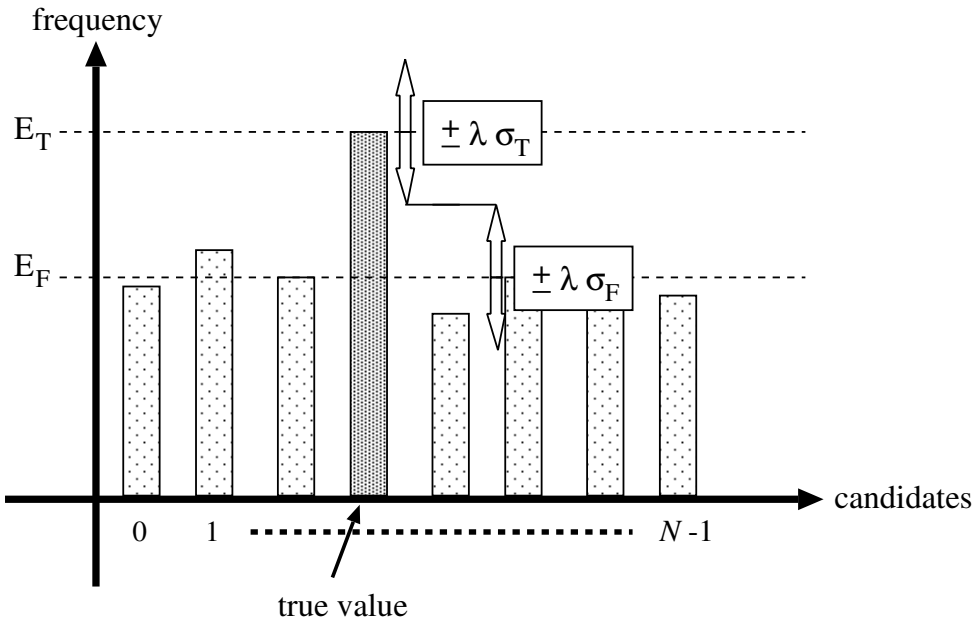


図 5.1　統計的な鍵復元手法

$$E_F = N_{IV} \cdot p \tag{5.6}$$

$$\sigma_F = \sqrt{N_{IV} \cdot p \cdot (1-p)} \approx \sqrt{N_{IV} \cdot p} \tag{5.7}$$

ここで，正しい候補のカウント数が $E_T \pm \lambda\sigma_T$，誤った候補のカウント数がそれぞれ $E_F \pm \lambda\sigma_F$ の範囲の値をとると仮定する．この仮定の成立確率は λ に依存し，λ が大きくなるほどその成立確率が大きくなる．この仮定の下で，以下の条件を満たしているならば，正しい候補のカウント数は誤った候補のカウント数から分離できる．

$$E_T - \lambda\sigma_T > E_F + \lambda\sigma_F \tag{5.8}$$

式 (5.8) の条件式は式 (5.4)–(5.7) より以下のように書き直すことができる．

$$\lambda < \frac{\sqrt{N_{IV}} \cdot q}{\sqrt{p+q} + \sqrt{p}} \tag{5.9}$$

式 (5.9) の右辺の値を増加させれば λ を大きくでき，正しい $K[a \bmod L]$ の値を得られる確率が向上することが分かる．さらに，q が p に対して無視できるほど小さい場合，式 (5.9) の右辺の値は $\sqrt{N_{IV}} \cdot q$ に依存する．

5.4　秘密鍵との相関が小さな weak IV

FMS 攻撃などの従来の鍵復元攻撃では攻撃方程式から生じる偏差 q の値が 2^{-5} 程度の weak IV を用いている．誤った候補を選択する確率 p は 2^{-8} より q が十分大きいことから，これら

表 5.1　式 (5.12) のための fortunate event の時刻 t における条件とその成立確率

時刻 t	subevent(t)	Pr(subevent(t)|subevent($1,\ldots,t-1$))
1	$j_t^* \neq 1, a-1, a$	$(N-3)/N$
2	$j_t^* = a-1 \wedge$ subevent(1)	1
$3,\ldots,a-1$	$j_t^* \neq 1, a-1, a \wedge$ subevent$(1,2,\ldots,t-1)$	$(N-3)/N$
a	$j_t^* = a-1 \wedge$ subevent$(1,2,\ldots,a-1)$	$1/N$
$a+1$	$j_t^* \neq 1, a-1 \wedge S_{t-1}^*[j_t^*] = j_t^* \wedge$ subevent$(1,2,\ldots,a)$	$(a-2)/N \cdot 1/N + 1/N \cdot 1$ $+(N-a-1)/N \cdot ((N-1)/N)^a$
$a+2,\ldots,N$	$j_t^* \neq 1, a-1, a \wedge$ subevent$(1,2,\ldots,t-1)$	$(N-3)/N$

の weak IV は少数あれば秘密鍵を復元できる．しかしながら，これらの weak IV には q を大きくするために厳しい条件を与えられており，weak IV の割合は比較的小さくなってしまう．例えば FMS 攻撃では

$$3 \leq K[0] \leq L-1 \tag{5.10}$$

$$K[1] = N-1 \tag{5.11}$$

を満たす IV が weak IV であり，IV 全体における weak IV の割合は約 0.02%である．

本節では，従来の weak IV とは異なり攻撃方程式から生じる偏差 q の値が 2^{-12} から 2^{-20} 程度である weak IV に注目し，そのような weak IV を特定する．本研究で特定する weak IV では，ほとんど条件が付加されないため，大きな割合の IV が weak IV として特定される．本節で示す weak IV の割合は約 81%であり，従来のものと比べて大きい．また，多数の weak IV を攻撃に利用できる，すなわち N_{IV} を大きくできるために q が小さいことを補うことができる．

5.4.1　攻撃方程式

IV に特定の条件を定めず，比較的高い確率で生じる KSA の状態遷移について検討することで，図 5.2 のような fortunate event を発見した．この fortunate event に対応する攻撃方程式は

$$K[a \bmod L] = (Z_1 - (a-1) - a) \bmod N \tag{5.12}$$

であり，推測値のインデックス a は

$$a = (K[0] + K[1] + 2) \bmod N,\ a \in \{0, 1, \ldots, N-1\} \tag{5.13}$$

のように IV から計算できる．攻撃に必要な Z_1 を暗号化パケットから得る方法は文献 [49] で示されている．表 5.1 に KSA の時刻毎の fortunate event (subevent(t)) の条件とその成立確率を与えている．

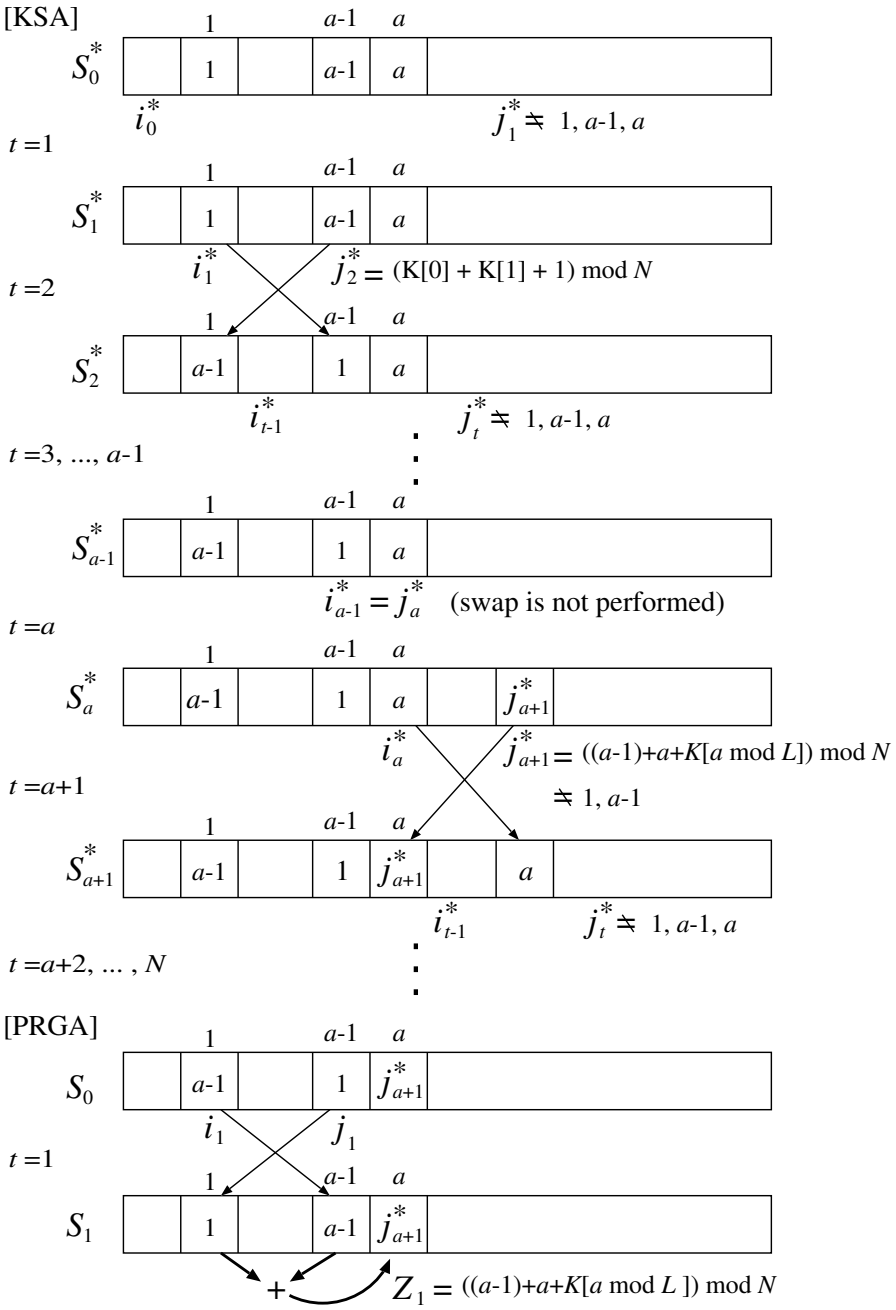


図 5.2　提案する攻撃方程式における fortunate event

fortunate event の条件　表 5.1 に示した fortunate event の条件を満たすときに，式 (5.12) が成立することを示す．KSA の時刻 $t = 1$ において $j_t^* \neq 1, a-1, a$ が成立しているとき，$i_{t-1}^* = 0$ より 3 個の要素 $S_0^*[1] = 1$，$S_0^*[a-1] = a-1$，$S_0^*[a] = a$ はスワップされない．よって，$S_1^*[1] = 1$，$S_1^*[a-1] = a-1$，$S_1^*[a] = a$ が成立する．時刻 $t = 2$ において $j_t^* = a-1$ が成立するとき，$i_{t-1}^* = 1$ より $S_1^*[1]$ と $S_1^*[a-1]$ のスワップによって内部状態が更新されるため，$S_2^*[1] = a-1$，$S_2^*[a-1] = 1$，$S_2^*[a] = a$ が成立する．さらに，時刻 $t = 3, \ldots, a-1$ において常に $j_t^* \neq 1, a-1, a$

が成立しているならば，$i^*_{t-1} \neq 1, a-1, a$ より 3 個の要素 $S^*_{t-1}[1]$，$S^*_{t-1}[a-1]$，$S^*_{t-1}[a]$ はスワップされない．よって，$S^*_{a-1}[1] = a-1$，$S^*_{a-1}[a-1] = 1$，$S^*_{a-1}[a] = a$ が成立する．時刻 $t = a$ において $j^*_t = a-1$ が成立するならば，$i^*_{t-1} = a-1$ よりスワップが生じずに $S^*_a[1] = a-1$，$S^*_a[a-1] = 1$，$S^*_a[a] = a$ が成立する．時刻 $t = a+1$ において $S^*_{t-1}[j^*_t] = j^*_t$ が成立しているならば，$i^*_{t-1} = a$ より $S^*_{a+1}[a]$ に j^*_{a+1} がセットされる．このとき $j^*_t \neq 1, a-1$ が成立しているならば，$S^*_a[1]$ と $S^*_a[a-1]$ はスワップされず，$S^*_{a+1}[1] = a-1$，$S^*_{a+1}[a-1] = 1$，$S^*_{a+1}[a] = j^*_{a+1}$ が成立する．ここで，時刻 $t = a$ より $j^*_a = a-1$ と $S^*_a[a] = a$ が得られることから，式 (2.7) より以下の関係式が得られる．

$$j^*_{a+1} = (j^*_a + S^*_a[a] + K[a \bmod L]) \bmod N = ((a-1) + a + K[a \bmod L]) \bmod N \quad (5.14)$$

そして，時刻 $t = a+2, \ldots, N$ において常に $j^*_t \neq 1, a-1, a$ が成立しているならば，$i^*_{t-1} \neq 1, a-1, a$ より 3 つの要素 $S^*_{t-1}[1]$，$S^*_{t-1}[a-1]$，$S^*_{t-1}[a]$ はスワップされない．よって，初期状態の 3 個の要素は $S_0[1] = S^*_N[1] = a-1$，$S_0[a-1] = S^*_N[a-1] = 1$，$S_0[a] = S^*_N[a] = j^*_{a+1} = ((a-1) + a + K[a \bmod L]) \bmod N$ となる．ここで，PRGA の時刻 $t = 1$ の処理を考える．$i_1 = 1$，$j_1 = S_0[1] = a-1$ より，$S_1[a] = S_0[a]$ となる．このとき，式 (2.13)–(2.16) より Z_1 は以下のように得られる．

$$\begin{aligned} Z_1 &= S_1[(S_0[1] + S_0[S_0[1]]) \bmod N] = S_1[a] = S_0[a] = j^*_{a+1} \\ &= ((a-1) + a + K[a \bmod L]) \bmod N \end{aligned} \quad (5.15)$$

上式を変形すれば，式 (5.12) となる．

fortunate event の生起確率　fortunate event の生起確率 $\alpha(a)$ について議論する．$\alpha(a)$ は表 5.1 に与えた時刻毎の subevent の成立確率 $\Pr(\text{subevent}(t)|\text{subevent}(1, \ldots, t-1))$ を $t = 1, 2, \ldots, N$ まで計算し，それらの積をとることで

$$\frac{1}{N} \cdot \left(\frac{N-3}{N}\right)^{N-3} \cdot \left(\frac{a-2}{N^2} + \frac{1}{N} + \frac{N-a-1}{N} \cdot \left(\frac{N-1}{N}\right)^a\right) \quad (5.16)$$

のように得られる．ここで，$\Pr(\text{subevent}(t)|\text{subevent}(1, \ldots, t-1))$ のそれぞれの導出について説明する．まず，KSA の時刻 $t = 1, 2, \ldots, N$ で j^*_t が一様でランダムな変数と仮定する．このとき，時刻 $t = 1$，時刻 $t = 3, \ldots, a-1$，時刻 $t = a$，時刻 $t = a+2, \ldots, N$ のそれぞれに対応する $\Pr(\text{subevent}(t)|\text{subevent}(1, \ldots, t-1))$ は容易に導出できる．また，subevent(1) を満たしているとき $S_1[1] = 1$ と $S_0[0] = 0$ が成立するため，式 (2.7)，(5.13) より

$$\begin{aligned} j^*_2 &= (j^*_1 + S^*_1[1] + K[1]) \bmod N \\ &= (S^*_0[0] + S^*_1[1] + K[0] + K[1]) \bmod N \quad (5.17) \\ &= (0 + 1 + K[0] + K[1]) \bmod N = a-1 \quad (5.18) \end{aligned}$$

表 5.2　subevent$(a+1)$ における $S^*_{t-1}[j^*_t] = j^*_t$ の成立確率と j^*_t の関係

j^*_t	$\Pr(S^*_{t-1}[j^*_t] = j^*_t \mid j^*_t)$
$0, 2, 3, \ldots, a-2$	$1/N$
a	1
$a+1, a+2, \ldots, N-1$	$((N-1)/N)^a$
$1, a-1$	0

となり，時刻 $t = 2$ の subevent(t) の条件は必ず成立することがわかる．時刻 $t = a+1$ における subevent(t) の条件を満たす確率について説明する．subevent$(a+1)$ の条件の一つである $j^*_t \neq 1, a-1$ を $j^*_t = 0, 2, 3, \ldots, a-2$ と $j^*_t = a$ と $j^*_t = a+1, a+2, \ldots, N-1$ の 3 つの場合に分割して考えることで成立確率を導出する．j^*_t が一様でランダムな変数であるとき，$j^*_t = 0, 2, 3, \ldots, a-2$ の生起確率は $(a-2)/N$ である．$j^*_t = 0, 2, 3, \ldots, a-2$ を満たしている場合，$S^*_{t-1}[j^*_t]$ は時刻 $j^*_t + 1$ で既にスワップされており，その値はランダム化されている．ゆえに，$j^*_t = 0, 2, 3, \ldots, a-2$ が満たされている場合に $S^*_{t-1}[j^*_t] = j^*_t$ が成立する確率は $1/N$ とみなせる．また，$j^*_t = a$ は確率 $1/N$ で生起する．時刻 $t' = 1, 2, \ldots, a$ における subevent(t') の条件を満たされているときには $S^*_a[a] = a$ が満たされている．ゆえに，$j^*_t = a$ が満たされていれば $S^*_{t-1}[j^*_t] = j^*_t$ は確率 1 で成立する．$j^*_t = a+1, a+2, \ldots, N-1$ の生起確率は $(N-a-1)/N$ である．$j^*_t = a+1, a+2, \ldots, N-1$ を満たしている場合，時刻 $t' = 1, 2, \ldots, a$ で常に $j^*_{t'} \neq j^*_t$ を満たしていれば $S^*_{t-1}[j^*_t] = S^*_0[j^*_t] = j^*_t$ が成立する．その確率は $((N-1)/N)^a$ のように得られる．したがって，時刻 $t = a+1$ における $\Pr(\text{subevent}(t) | \text{subevent}(1, \ldots, t-1))$ はそれぞれの場合における確率から以下のように与えられる．

$$\frac{a-2}{N} \cdot \frac{1}{N} + \frac{1}{N} \cdot 1 + \frac{N-a-1}{N} \cdot \left(\frac{N-1}{N}\right)^a \tag{5.19}$$

表 5.2 は subevent$(a+1)$ における $S^*_{t-1}[j^*_t] = j^*_t$ の成立確率と j^*_t の関係をまとめたものである．もし時刻 $a+1$ において j^*_t の位置をなんらかの方法で検出できるなら，$S^*_{t-1}[j^*_t] = j^*_t$ が高確率で成立するような j^*_t のみを選択することで subevent$(a+1)$ の成立確率を式 (5.19) より高くできる．この原理によって攻撃方程式の成立確率を向上させる方法の詳細は 5.5 節で議論する．

攻撃方程式から生じる偏差　提案した攻撃方程式から生じる偏差 q の値を計算機実験および式 (5.16) で求める．ある a に対する q の値は理論値からは $\alpha(a) \cdot (N-1)/N$ を計算することによって求められる．計算機実験ではパラメータを $\ell = 128$，$n = 8$ とし，攻撃方程式を 2^{32} 回試行して成立確率を得る．攻撃方程式から生じる偏差の実験値と理論値を図 5.3 に示す．図より理論値は実験値をよく近似できていることが分かる．また，全ての a について攻撃方程式によって生じる偏差は 2^{-12} から 2^{-20} 程度であることが分かる．また，$(a \bmod L)$ 毎の偏差の平

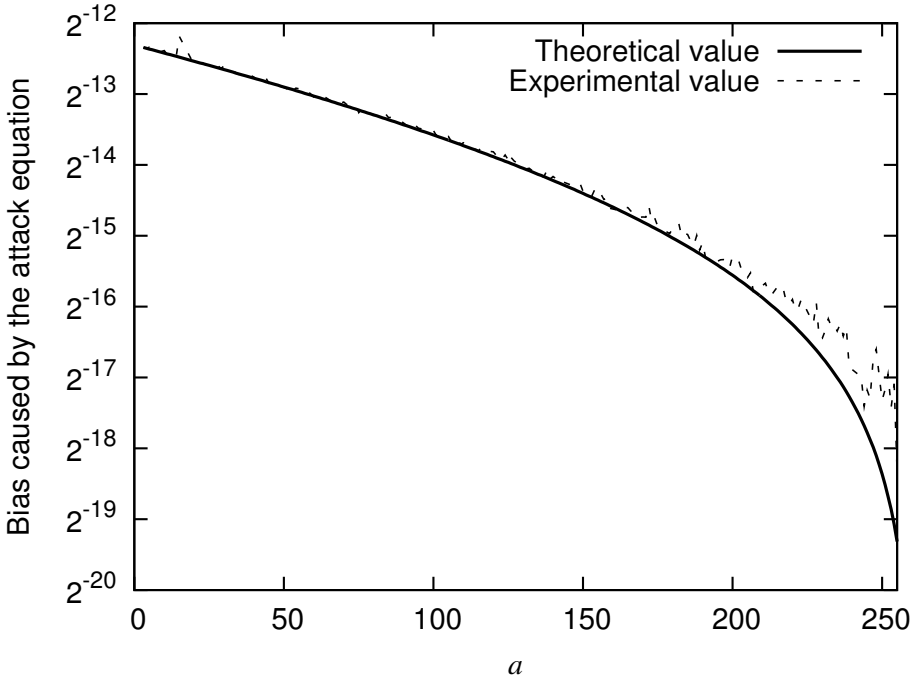


図 5.3　式 (5.12) から生じる偏差の実験値と理論値

均は $2^{-13.66}$ から $2^{-13.84}$ の間となる．異なる a の weak IV を $(a \bmod L)$ 毎に集めて統計的な鍵復元手法を実行する場合，q は偏差の平均に対応する．

5.4.2　weak IV の割合

式 (5.12) によって利用可能な weak IV の割合と個数について議論する．WEP の IV の総数は N^3 個である．それらの IV は式 (5.13) によって $(a \bmod L)$ の値毎に L 個のグループに等しく分割される．それぞれのグループの IV はセッション鍵の要素 $K[a \bmod L]$ を復元するために使われる．$a \bmod L = 0, 1, 2$ を満たす 3 つのグループは IV 自身の情報を漏洩することになるため weak IV とはならない．したがって，weak IV の割合は全体の $(L-3)/L$ となり，個数は $N^3 \cdot (L-3)/L$ となる．実用的なパラメータである $\ell = 128$，$n = 8$ の場合，$13/16 \approx 0.81$ の割合で weak IV が存在することになる．また，1 つの秘密鍵の要素を復元するために用いることができる weak IV の個数は N^3/L となり，これは N_{IV} の上限となる．

5.5　推測値を利用する weak IV の選別法

本節では，一度計算で求めた推測値を利用することにより，攻撃方程式によって生じる偏差が比較的大きな weak IV を選別する方法を提案する．さらに，その方法を利用した鍵復元攻撃を提案する．weak IV の選別法に関する説明は 5.4.1 節で特定した weak IV を対象にして行う．

5.4.1 節より，提案した攻撃方程式に対応する weak IV では q は p と比べて十分小さい．このとき，統計的な鍵復元手法の成功確率は $\sqrt{N_{IV}} \cdot q$ に依存する．したがって，鍵復元攻撃の成功確率を最適化するためには $\sqrt{N_{IV}} \cdot q$ が最大になるように weak IV を用いれば良い．q は攻撃に使用する weak IV における攻撃方程式によって生じる偏差の平均値に対応する．したがって，偏差が小さい順に weak IV をソートし，攻撃に使用する IV の中から比較的偏差の小さい weak IV を順に除外していくことで，$\sqrt{N_{IV}} \cdot q$ が最大となるような weak IV の集合が得られる．以下の説明では，$\sqrt{N_{IV}} \cdot q$ の最適化をする際に用いる weak IV の選別法について議論する．

5.5.1 従来研究

weak IV の選別法として，先頭鍵バイト推測法 (GEKB approach : Guessing-Early-Key-Bytes approach)[49] が知られている．GEKB approach では秘密鍵の先頭の B 個の要素 $K[3], K[4], \ldots, K[B+2]$ が既知という条件を必要とする．このとき，攻撃者は KSA の時刻 $B+3$ までの内部状態の更新を実行できる．これは攻撃者が subevent$(1, 2, \ldots, B+3)$ の条件を満たすような weak IV のみ選択できることを意味している．このような weak IV では fortunate event の生起確率が大きくなるため，攻撃方程式によって生じる偏差が大きくなる．しかしながら，GEKB approach は subevent$(B+4, B+5, \ldots, N)$ において成立確率を下げる要因があるような weak IV に対しては実行できず，提案した攻撃方程式に対応する weak IV で a が大きくなったような場合では有効といえない．

5.5.2 推測値選別法

推測値 $gval(a)$ の情報を用いて weak IV の選別を行う方法を提案する．この方法を推測値選別法 (CGV approach : Categorizing-Guessed-Values approach) と呼ぶ．CGV approach は weak IV と $gval(a)$ の情報を利用して，ある時刻 t' の subevent で条件として使われているパラメータの値を知る方法である．既知となったパラメータから，その weak IV が subevent(t') の条件を満たしやすいかを判断することで偏差の大きな weak IV を選別できる．$gval(a)$ によってパラメータを知ることができる時刻 t' は fortunate event に依存して変わるため，fortunate event によっては全ての weak IV を利用できる．

提案した攻撃方程式の weak IV は全ての範囲で適用できる例であり，その weak IV への CGV approach の適用について説明する．CGV approach では subevent$(a+1)$ に注目する．5.4.1 節の議論より，subevent$(a+1)$ の条件の成立確率は j^*_{a+1} の値に依存している．もし，$gval(a)$ から j^*_{a+1} の値を既知にできるなら，*fortunate event* の生起確率が比較的高い weak IV を選別できる．時刻 $t = a+1$ のとき，j^*_{a+1} は式 (2.7) より $j^*_{a+1} = (j^*_a + S^*_a[a] + K[a \bmod L]) \bmod N$ の

ように得られる．subevent(a) が生じているとき，$j_a^* = a-1$ が成立する．subevent$(1, 2, \ldots, a)$ が生じているとき，時刻 $t = 1, 2, \ldots, a$ で常に $i_{t-1}^* \neq a$ と $j_t^* \neq a$ が成立しているため，式 (2.8) から $S_a^*[a] = S_0^*[a] = a$ が得られる．このとき，$j_{a+1}^* = (2a - 1 + K[a \bmod L]) \bmod N$ が成立する．加えて，fortunate event が生じているとき推測値は正しい $K[a \bmod L]$ となる．よって，fortunate event が生じているとき，j_{a+1}^* は以下の式によって計算される．

$$j_{a+1}^* = (2a - 1 + gval(a)) \bmod N \tag{5.20}$$

したがって，攻撃者は $gval(a)$ と weak IV から j_{a+1}^* を知ることが可能となる．提案する攻撃方程式に対応する weak IV は全て subevent$(a+1)$ の条件の成立確率に大きく依存するため，CGV approach は全ての weak IV で有効となる．このような場合，ある特定の weak IV のみ攻撃で使われるようにならないため，WEPplus で防ぐ際にフィルタリングする必要がある IV の割合は CGV approach によって減少しないことに注意する．

5.5.3 提案手法

本節では，CGV approach を利用した鍵復元攻撃を提案する．攻撃の目的は全ての秘密鍵の要素 $K[3], K[4], \ldots, K[L-1]$ を復元することである．提案手法は鍵復元処理と復元誤り補償処理によって構成されている．

鍵復元処理　予備実験より，$\sqrt{N_{IV}} \cdot q$ を最大にする，すなわち攻撃成功確率を最大にするためには CGV approach で $j_{a+1}^* = a, a+1, \ldots, N-1$ を満たす weak IV を選択すれば良いことを確認している．このようなパラメータで CGV approach を実行するような鍵復元処理のアルゴリズムを図 5.4 に示す．アルゴリズムでは，weak IV と Z_1 のペアを多数入力し，秘密鍵の要素 $K[3], K[4], \ldots, K[L-1]$ を推定する．まず Count up loop において，weak IV と Z_1 のペア及び式 (5.12)，(5.13) を用いて推測値 $gval(a)$ を得る．全ての weak IV と Z_1 のペアから計算された推測値は a 毎にカウントされていき候補毎に頻度表 $s_cnt[a][]$ に格納される．次に CGV approach loop において，式 (5.20) を利用して a と $gval(a)$ から $j_{a+1}^* = a, a+1, \ldots, N-1$ を満たすかをチェックする．もし $j_{a+1}^* = a, a+1, \ldots, N-1$ を満たしているなら，そのカウント数を採用する．具体的には $(a \bmod L)$ 毎の頻度表 $cnt[a \bmod L][]$ に $cnt[a \bmod L][gval(a)] \mathrel{+}= s_cnt[a][gval(a)]$ のように対応するカウント数を加算する．そうでないなら攻撃に使用しない weak IV と判断され，weak IV の影響を取り除くため，$cnt[a \bmod L][gval(a)] \mathrel{+}= \sum_{x=0}^{N-1} s_cnt[a][x]/N$ のように平均値をとってから加算される．最後に Key recovery loop において，$sb = 3, 4, \ldots, L-1$ について頻度表 $cnt[sb][]$ から最もカウント数が大きいものを $K[sb]$ と推定し，$K[3], K[4], \ldots, K[L-1]$ を得る．

Input: The pairs of the weak IV and Z_1
Output: $K[3], K[4], \ldots, K[L-1]$
Initialization:
$s_cnt[\,][\,]$ and $cnt[\,][\,]$ are initialized to 0;
Count up loop:
For all pairs of the weak IV and Z_1
  Determine a using Eq. (5.13);
  Calculate $gval(a)$ by Eq. (5.12);
  $s_cnt[a][gval(a)]$ += 1;
CGV approach loop:
For $a = 0, 1, \ldots, N-1$
  For $gval(a) = 0, 1, \ldots, N-1$
    Check j^*_{a+1} by Eq. (5.20);
    if $j^*_{a+1} = a, a+1, \ldots, N-1$
      $cnt[a \bmod L][gval(a)]$ += $s_cnt[a][gval(a)]$;
    else
      $cnt[a \bmod L][gval(a)]$ += $\frac{\sum_{x=0}^{N-1} s_cnt[a][x]}{N}$;
Key recovery loop:
For $sb = 3, 4, \ldots, L-1$
  Output $\max[cnt[sb][x],\ x \in \{0, 1, \ldots, N-1\}]$ as recovered $K[sb]$;

図 5.4　CGV approach を利用した鍵復元処理のアルゴリズム

復元誤り補償処理　鍵復元処理によって推定した秘密鍵の要素の中には誤っている要素が含まれている場合がある．さらに，復元に失敗した要素の位置や個数を知ることはできない．そこで復元誤り補償処理では，攻撃者は閾値 k を選択し，少なくとも k 個の要素が正しく復元できたと仮定する．このとき，復元誤り補償処理では，残りの $L-3-k$ 個の秘密鍵の要素を総当たりして正しい値を得る．具体的には，誤っている可能性がある秘密鍵の要素の位置と値の全てのパターンを試す．$L-3-k$ 個の秘密鍵の要素に全てのパターンの値を試すための計算量は N^{L-3-k} となる．また，鍵復元処理で復元に失敗している可能性がある位置の全てのパターンを試すことになるため，計算量は更に ${}_{L-3}C_k$ 倍になる．したがって，復元誤り補償処理で必要な計算量は

$$ {}_{L-3}C_k \cdot N^{L-3-k} \tag{5.21}$$

となる．提案する鍵復元攻撃の成功確率や計算量は k の値を変更することで制御できる．

5.6 評価

提案手法の成功確率及び計算量について評価する．また，CGV approach の有効性を検証するため，CGV approach の処理を取り除いたバージョンの鍵復元攻撃についても評価する．CGV approach の処理を取り除いたバージョンの鍵復元攻撃を基本手法と呼ぶことにする．

計算機実験のみで成功確率を得る場合，k が大きくなると現実的に実行できる試行回数では成功確率が得られない．そこで，実験値を利用した統計的な解析を導入することによって k が大きい場合でも成功確率を得られるようにする．5.4.1 節より，$(a \bmod L)$ 毎に weak IV の集合を作ったとき，それぞれの集合で攻撃方程式によって生じる偏差の平均 q は $2^{-13.66}$ から $2^{-13.84}$ のように同程度であった．また，それぞれの集合の weak IV の個数の最大値 N_{IV} は 5.4.2 節より同程度となることが分かる．よって，利用可能な全ての weak IV の用いる場合における評価では，秘密鍵の要素の復元確率 P は全ての要素で等しいと仮定できる．このとき，鍵復元処理によって $L-3$ 個の秘密鍵の要素のうち k 個が復元できる確率は試行回数 $L-3$，生起確率 P の二項分布 $\mathrm{B}(L-3,\ P)$ に従う．したがって，鍵復元処理によって少なくとも k 個の秘密鍵の要素が復元できている確率は以下のように表される．

$$\sum_{x=k}^{L-3} \left({}_{L-3}C_x \cdot P^x \cdot (1-P)^{L-3-x} \right) \tag{5.22}$$

予備実験によって $k = 1, 2, \ldots, 6$ における攻撃成功確率の実験値を求め，式 (5.22) が実験値を近似できていることを確認している．

本章の提案手法及び基本手法の評価では，100,000 個の 104 ビットの秘密鍵を対象とした解読実験によってそれぞれの攻撃における P の実験値を求める．その P の値と式 (5.22) から閾値 k に対応する攻撃の成功率を得る．k に対応する計算量は鍵復元処理と復元誤り補償処理の計算量の和から以下のように与えられる．

$$N^3 \cdot \frac{L-3}{L} + {}_{L-3}C_k \cdot N^{L-3-k} \tag{5.23}$$

ここで，計算量を攻撃の成功率で割った値を効率と定義し，評価の指標として用いる．全数探索やランダム探索によって ℓ ビットの秘密鍵を復元する場合，効率の値は 2^ℓ となる．もし，攻撃の成功率が向上する，もしくは計算量が削減できるような効果的な方法によって秘密鍵を復元する場合，効率の値は 2^ℓ より小さくなる．また，効率の値が小さくなるほど，その方法で解読できる秘密鍵は強度が低下しているといえる．

上記の評価方法によって，基本手法と基本手法を Case 1 と Case 2 の 2 つの条件において評価する．Case 1 は WEPplus が適用されておらず，全ての IV と Z_t のペアを攻撃者が手に入れることができる場合である．Case 2 は吉田らによって提案されているフィルタリングパター

ンを用いて Case 1 の IV から既知の weak IV を取り除いた場合である．$\ell = 128$，$n = 8$ のとき，吉田らのフィルタリングパターンでは以下の条件を満たす IV を取り除く [54].

$$0 \leq (K[0] + K[1]) \bmod N \leq 14 \tag{5.24}$$

$$241 \leq (K[0] + K[1]) \bmod N \leq 255 \tag{5.25}$$

$$0 \leq (K[0] + K[1] + K[2]) \bmod N \leq 10 \tag{5.26}$$

$$242 \leq (K[0] + K[1] + K[2]) \bmod N \leq 255 \tag{5.27}$$

$$0 \leq (2 \cdot K[0] + 2 \cdot K[1] + K[2]) \bmod N \leq 10 \tag{5.28}$$

$$254 \leq (2 \cdot K[0] + 2 \cdot K[1] + K[2]) \bmod N \leq 255 \tag{5.29}$$

$$0 \leq (2 \cdot K[0] + K[1] + K[2]) \bmod N \leq 12 \tag{5.30}$$

$$K[1] = 255 \tag{5.31}$$

$$K[2] = 254 \tag{5.32}$$

この条件で取り除かれる IV は全体の 31.2%である．さらに吉田らのフィルタリングパターンでは秘密鍵の選択にも条件を与えており，$K[4] = 250, 252, 254$ を満たす秘密鍵を使用しないようにする．WEPplus でこのような設定を付加することで，知られているほぼ全ての weak IV を取り除くことができる．

表 5.3，5.4 は基本手法と提案手法の計算量と成功率及び効率を Case 1 と Case 2 について示したものである．表 5.3，5.4 から，Case 1 と Case 2 では共に提案手法が基本手法より成功率が高く，CGV approach が有効に働いていることが確認できる．Case 1 では，それぞれ $k = 11$ において効率が最小になり，基本手法は $2^{72.0}$ であり，提案手法は $2^{63.3}$ である．また，Case 2 でも効率が最小になったのは $k = 11$ の場合となり，基本手法は $2^{75.9}$ であり，提案手法は $2^{67.8}$ である．これは，吉田らによって提案された安全な WEPplus の実装においても，提案手法は 68 ビットの秘密鍵を全数探索する場合と同程度の効率で 104 ビットの秘密鍵を復元できることを意味する．

5.7 結言

本章では，攻撃方程式から生じる偏差が小さな weak IV に注目し，それらを用いた鍵復元攻撃を提案した．本章で特定できた weak IV の割合は IV 全体の約 81%と大きく，WEPplus でそれらをフィルタリングすることは実用的な問題が生じてしまい困難であると考えられる．また，攻撃方程式から得られた推測値を利用し，weak IV を偏差の大きさ別に選別する CGV approach を提案した．さらに，吉田らが提案した効果的なフィルタリングパターンを用いて weak IV を取り除いた WEPplus に対して計算機実験で提案手法の性能を確認した．その結果，

表 5.3　鍵復元攻撃の評価 (Case 1)

k	計算量	基本手法		提案手法	
		成功率	効率	成功率	効率
1	$2^{99.7}$	$2^{-1.5}$	$2^{101.2}$	$2^{-0.9}$	$2^{100.6}$
2	$2^{94.3}$	$2^{-3.9}$	$2^{98.2}$	$2^{-2.6}$	$2^{96.9}$
3	$2^{88.2}$	$2^{-6.9}$	$2^{95.1}$	$2^{-4.8}$	$2^{93.0}$
4	$2^{81.5}$	$2^{-10.5}$	$2^{92.0}$	$2^{-7.6}$	$2^{89.1}$
5	$2^{74.3}$	$2^{-14.5}$	$2^{88.8}$	$2^{-10.8}$	$2^{85.1}$
6	$2^{66.7}$	$2^{-19.0}$	$2^{85.7}$	$2^{-14.5}$	$2^{81.2}$
7	$2^{58.7}$	$2^{-23.9}$	$2^{82.6}$	$2^{-18.5}$	$2^{77.2}$
8	$2^{50.3}$	$2^{-29.2}$	$2^{79.5}$	$2^{-23.0}$	$2^{73.3}$
9	$2^{41.5}$	$2^{-34.9}$	$2^{76.4}$	$2^{-27.9}$	$2^{69.4}$
10	$2^{32.2}$	$2^{-41.1}$	$2^{73.3}$	$2^{-33.2}$	$2^{65.4}$
11	$2^{24.2}$	$2^{-47.8}$	$2^{72.0}$	$2^{-39.1}$	$2^{63.3}$
12	$2^{23.7}$	$2^{-55.3}$	$2^{79.0}$	$2^{-45.8}$	$2^{69.5}$
13	$2^{23.7}$	$2^{-63.8}$	$2^{87.5}$	$2^{-53.5}$	$2^{77.2}$

表 5.4　鍵復元攻撃の評価 (Case 2)

k	計算量	基本手法		提案手法	
		成功率	効率	成功率	効率
1	$2^{99.7}$	$2^{-1.8}$	$2^{101.5}$	$2^{-1.2}$	$2^{100.9}$
2	$2^{94.3}$	$2^{-4.5}$	$2^{98.8}$	$2^{-3.2}$	$2^{97.5}$
3	$2^{88.2}$	$2^{-7.9}$	$2^{96.1}$	$2^{-5.9}$	$2^{94.1}$
4	$2^{81.5}$	$2^{-11.9}$	$2^{93.4}$	$2^{-9.1}$	$2^{90.6}$
5	$2^{74.3}$	$2^{-16.2}$	$2^{90.5}$	$2^{-12.7}$	$2^{87.0}$
6	$2^{66.7}$	$2^{-21.1}$	$2^{87.8}$	$2^{-16.8}$	$2^{83.5}$
7	$2^{58.7}$	$2^{-26.3}$	$2^{85.0}$	$2^{-21.2}$	$2^{79.9}$
8	$2^{50.3}$	$2^{-32.0}$	$2^{82.3}$	$2^{-26.1}$	$2^{76.4}$
9	$2^{41.5}$	$2^{-38.0}$	$2^{79.5}$	$2^{-31.4}$	$2^{72.9}$
10	$2^{32.2}$	$2^{-44.6}$	$2^{76.8}$	$2^{-37.2}$	$2^{69.4}$
11	$2^{24.2}$	$2^{-51.7}$	$2^{75.9}$	$2^{-43.6}$	$2^{67.8}$
12	$2^{23.7}$	$2^{-59.5}$	$2^{83.2}$	$2^{-50.6}$	$2^{74.3}$
13	$2^{23.7}$	$2^{-68.5}$	$2^{92.2}$	$2^{-58.8}$	$2^{82.5}$

提案した鍵復元攻撃では 104 ビットの WEP の秘密鍵を $2^{67.8}$ の効率で解読でき，提案手法は WEPplus による対策を施した無線 LAN 製品に対しても有効であるといえる．

第6章　WEPに対する既知IV攻撃の改良

6.1　緒言

第5章では，WEPに対するweak IVを用いる鍵復元攻撃について議論した．WEPに対する鍵復元攻撃は年々進化しており，最近になって既知IV攻撃に分類される鍵復元攻撃がKlein[26]によって提案された．既知IV攻撃では全てのIVがweak IVとなるため，WEPplusによる対策は不可能である．さらにKleinの攻撃はTewsらによって改良されており，その攻撃はPTW攻撃と呼ばれている[27]．PTW攻撃はWEPに対する鍵復元攻撃の中で最も効果的な方法として知られており，任意のIVとキーストリームの先頭の15バイトのペアを複数集めることで104ビットの秘密鍵を復元する．

本章では攻撃に利用できるキーストリームを先頭の22バイトまで拡張することによってPTW攻撃より高確率で秘密鍵を復元する方法を提案する．単純にPTW攻撃を拡張するだけではキーストリームの先頭の22バイトを有効に用いることは困難である．なぜなら，PTW攻撃の攻撃方程式から得られる秘密鍵の情報がキーストリームの1バイト目から15バイト目までと19バイト目から22バイト目までででは異なることからである．そこで，キーストリームの19バイト目から22バイト目までに対応する秘密鍵の情報をキーストリームの3バイト目から6バイト目までに対応する情報に変換する方法を与え，キーストリームの先頭の22バイトを有効に用いることを可能とする．

6.2　定義

本章では，ワード長$n = 8$および秘密鍵長104ビットのWEPを対象に議論する．表記の簡潔さのために，セッション鍵の要素数Lは具体的な値である16を代入する．さらに，セッション鍵におけるIVと秘密鍵の情報を区別するために，24ビットのIVは$IV = (IV[0], IV[1], IV[2])$，秘密鍵は$Rk = (Rk[0], Rk[1], \ldots, Rk[12])$のように異なる記号を割り当てる．ここで，$IV[\cdot]$と$Rk[\cdot]$はそれぞれ8ビットの変数である．IVと秘密鍵は，$(K[0], K[1], K[2]) = (IV[0], IV[1], IV[2])$及び$(K[3], K[4], \ldots, K[15]) = (Rk[0], Rk[1], \ldots, Rk[12])$のように表される．

6.3　PTW攻撃

PTW攻撃[27]では，秘密鍵を復元するために，秘密鍵の要素の和σ_i $(i = 0, 1, \ldots, 12)$を得る．

$$\sigma_i = (K[3] + K[4] + \ldots + K[(3 + i) \bmod 16]) \bmod N \tag{6.1}$$

秘密鍵の全ての要素は σ_i を用いて以下のように計算できる．

$$Rk[i] \;=\; \begin{cases} \sigma_i & \text{if } i = 0 \\ (\sigma_i - \sigma_{i-1}) \bmod N & \text{if } i = 1, 2, \ldots, 12 \end{cases} \tag{6.2}$$

σ_i を得るための推測値を A_i と定義したとき，PTW 攻撃の攻撃方程式は以下のように与えられる．

$$A_i \;=\; \left(S_3^{*-1}[(3 + i - Z_{3+i}) \bmod N] - j_3^* - \sum_{l=3}^{i+3} S_3^*[l] \right) \bmod N \tag{6.3}$$

ここで，S_3^{*-1} は S_3^* の逆置換である．攻撃者は式 (2.7)–(2.9) と $IV[y]$ $(y = 0, 1, 2)$ を用いて S_3^* を計算できるため，Z_{i+3} が得られるときには A_i を計算できる．文献 [27] の 6.4 節による議論から，攻撃者は ARP パケットから $Z_1, Z_2, \ldots, Z_{31}$ を得ることが可能である．$\sigma_i = A_i$ の成立確率は以下のように与えられる．

$$\mathrm{Prob}(\sigma_i = A_i) \;\approx\; q_i \cdot \left(1 - \frac{1}{N}\right)^{N-2} \cdot \frac{2}{N} + \left(1 - q_i \cdot \left(1 - \frac{1}{N}\right)^{N-2}\right) \cdot \frac{N-2}{N(N-1)} \tag{6.4}$$

ここで

$$q_i = \left(1 - \frac{1}{N}\right)^i \cdot \left(1 - \frac{i}{N}\right) \cdot \prod_{k=1}^{i} \left(1 - \frac{k}{N}\right) \tag{6.5}$$

である．図 6.1 では，推測値 A_i から σ_i を復元できる確率の理論値を式 (6.4) によって求めたものを示している．図より，$i = 0, 1, \ldots, 40$ において $\mathrm{Prob}(\sigma_i = A_i)$ は $1/N$ より大きく，ランダム探索より高確率で σ_i を復元可能であることが確認できる．

PTW 攻撃では，与えられた暗号化パケットの全てに対して式 (6.3) を用いて $Z_3, Z_4, \ldots, Z_{15}$ から $A_0, A_1, \ldots, A_{12}$ を計算する．攻撃者は得られた推測値 A_i を σ_i 毎の頻度表 T_i でカウントする．そして，T_i でカウント数が大きな値を σ_i の候補として推定する．この推定によって σ_i を復元できる確率は，推測値によって σ_i が得られる確率と推測値の個数に依存する．

6.4 提案手法

1 個のパケットから得られる推測値の個数を増加させることができるなら，頻度表から正しい σ_i を復元できる確率が向上し，攻撃によって秘密鍵を復元できる確率が高くなる．本節では，RC4 の KSA においてセッション鍵の要素が何度も使われていることに注目し，Z_{16} 以降のキーストリームと PTW 攻撃の攻撃方程式を用いて 1 個のパケットから得られる推測値の個数を増加させることを考える．

PTW 攻撃の攻撃方程式は $i = 0, 1, \ldots, 252$ の範囲に適用でき，$Z_{16}, Z_{17}, \ldots, Z_{255}$ から $\sigma_{13}, \sigma_{14}, \ldots, \sigma_{252}$ のための推測値を計算することができる．これらの推測値を $T_{13}, T_{14}, \ldots, T_{252}$ にカウン

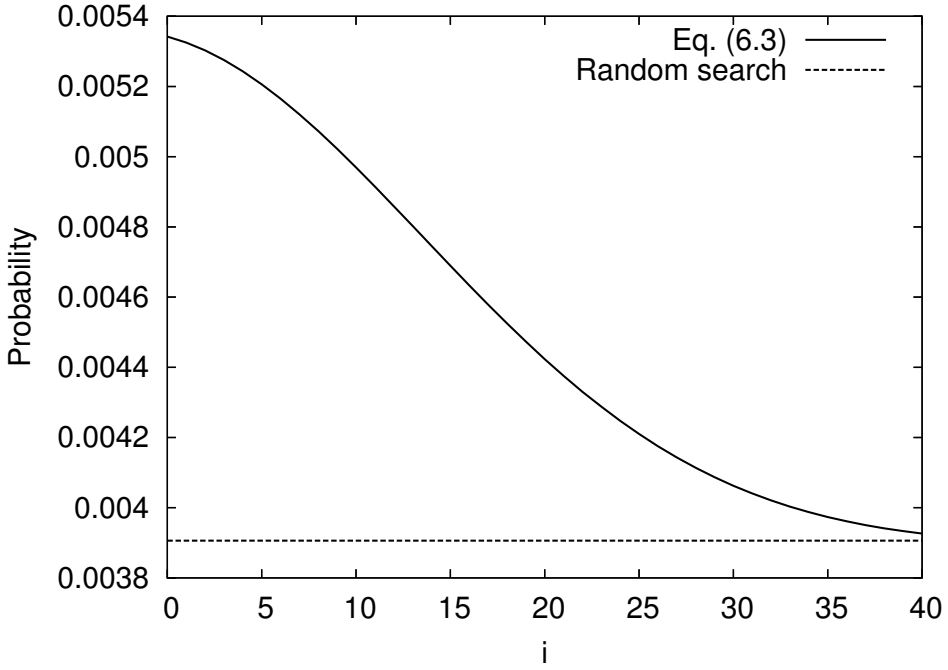


図 6.1　σ_i が復元される確率

トした場合，それぞれの頻度表では $T_0, T_1, \ldots, T_{12}$ と比べて秘密鍵の要素の和を復元できる確率が小さくなるために攻撃への貢献が小さくなる．そこで，これらの推測値を $\sigma_0, \sigma_1, \ldots, \sigma_{12}$ のための推測値に変換し，$T_0, T_1, \ldots, T_{12}$ でカウントすることを考える．$\sigma_{13}, \sigma_{14}, \sigma_{15}$ のための推測値は IV を用いた減算によって σ_{12} のための推測値に変換できる．しかしながら，$\sigma_{16}, \sigma_{17}, \ldots, \sigma_{252}$ のための推測値は秘密鍵の要素が複数回加算されているために，IV を用いた減算をするだけでは $\sigma_0, \sigma_1, \ldots, \sigma_{12}$ のための推測値に変換することはできない．そこで，我々は σ_{12} を先に求めることで，$\sigma_{16}, \sigma_{17}, \ldots, \sigma_{252}$ のための推測値を $\sigma_0, \sigma_1, \ldots, \sigma_{12}$ のための推測値に変換する．以上のように，提案手法では初めに σ_{12} を復元し，次に σ_{12} を用いて $\sigma_0, \sigma_1, \ldots, \sigma_{11}$ を復元するという手順で秘密鍵を復元する．

まず，Z_{16}, Z_{17}, Z_{18} と IV を利用することによって T_{12} にカウントされる推測値の個数を増加させる式を与える．$\sigma_{13}, \sigma_{14}, \sigma_{15}$ はそれぞれ σ_{12} を用いて $\sigma_{13} = (\sigma_{12} + K[16 \bmod 16]) \bmod N$，$\sigma_{14} = (\sigma_{12} + K[16 \bmod 16] + K[17 \bmod 16]) \bmod N$，$\sigma_{15} = (\sigma_{12} + K[16 \bmod 16] + K[17 \bmod 16] + K[18 \bmod 18]) \bmod N$ のように表現できる．よって，A_{12} は IV を利用した減算によって A_{13}, A_{14}, A_{15} から以下の 3 つの式を用いて計算できる．

$$A_{12} = \begin{cases} (A_{13} - IV[0]) \bmod N \\ (A_{14} - IV[0] - IV[1]) \bmod N \\ (A_{15} - IV[0] - IV[1] - IV[2]) \bmod N \end{cases} \tag{6.6}$$

パケット 1 個当たりの T_{12} にカウントされる推測値の個数は，この式を利用することで PTW

攻撃の 4 倍に増加する．

次に，σ_{12} と Z_{i+3} $(i = 16, 17, \ldots, 252)$ と IV を利用することによって T_{i-16m_i} にカウントされる推測値の個数を増加させる式を与える．ここで，$m_i = \lfloor i/16 \rfloor$ である．$i = 16, 17, \ldots, 252$ において，σ_{i-16m_i} は σ_{12} と σ_i 及び IV を利用することによって $\sigma_{i-16m_i} = (\sigma_i - m_i(IV[0] - IV[1] - IV[2] - \sigma_{12})) \bmod N$ のように表現できる．よって，A_i から A_{i-16m_i} を計算するための式は以下のように与えられる．

$$A_{i-16m_i} = (A_i - m_i(IV[0] - IV[1] - IV[2] - \sigma_{12})) \bmod N \tag{6.7}$$

また，予備実験によって $i = 16, 17, 18, 19$ で推測値の変換を実行した場合に成功確率が最も高くなることを確認している．したがって，提案手法ではそのパラメータにおいて推測値の変換を行う．パケット 1 個当たりの T_0, T_1, T_2, T_3 のそれぞれにカウントされる推測値の個数は，式 (6.7) を利用することで PTW 攻撃の 2 倍に増加する．

上記で与えた変換式を用いた提案手法を以下に与える．

Step 1: 全てのパケットに対して $Z_3, Z_4, \ldots, Z_{22}$ と IV から $A_0, A_1, \ldots, A_{19}$ を計算し，$T_0, T_1, \ldots, T_{19}$ にカウントする．

Step 2: 式 (6.6) と IV を用いて．T_{13}, T_{14}, T_{15} の推測値を σ_{12} のための推測値に変換し，T_{12} にカウントする．

Step 3: T_{12} でカウント数が大きいものから d 個の要素を選択し，σ_{12} のための候補とする．d 個の σ_{12} を順に $\hat{\sigma}_{12}$ にセットして Step 4 へ．d 個全ての候補をセットし終えていたならば，攻撃は失敗していると判断する．

Step 4: まず初めに，T_0, T_1, T_2, T_3 をバッファに退避させる．式 (6.7) と IV と $\hat{\sigma}_{12}$ を用いて，$T_{16}, T_{17}, T_{18}, T_{19}$ の推測値を $\sigma_0, \sigma_1, \sigma_2, \sigma_3$ のための推測値に変換し，T_0, T_1, T_2, T_3 にカウントする．$i = 0, 1, \ldots, 11$ の全てに対して，T_i でカウント数が大きいものから d 個の要素を選択し，σ_i の候補とする．式 (6.2) と $\hat{\sigma}_{12}$ を用いて，σ_i $(i = 0, 1, \ldots, 11)$ の全ての組合せについて秘密鍵を計算する．正しい秘密鍵が発見されたならば処理は終了する．そうでなければ，バッファから T_0, T_1, T_2, T_3 を書き戻し，Step 3 へ戻る．

上記の攻撃では秘密鍵探索手法として dwepcrack[52] で使われている方法を採用している．この方法では，T_i の要素の中からカウント数が大きいものから d 個選んで σ_i の候補にする．また，頻度表の個数が 13 であることから，この方法で要する計算量は d^{13} となる．

表 6.1　提案手法と PTW 攻撃によって復元できた秘密鍵の個数

観測パケット数	提案手法	PTW 攻撃
$10,000$	3	0
$20,000$	$1,323$	556
$40,000$	$44,252$	$33,949$
$80,000$	$96,527$	$94,182$

6.5　評価

提案手法と PTW 攻撃を計算機実験によって比較する．実験では，対象とする秘密鍵は strong-key byte[27] を含んでいないものとする．strong-key byte 用の探索方法は文献 [27] で示されており，strong-key byte が秘密鍵に含まれていたとしても PTW 攻撃と同様の方法を提案手法に適用すれば良い．したがって，この実験では strong-key byte を含まない秘密鍵に注目して提案手法の有効性を確認している．

実験では $100,000$ 個の 104 ビットの秘密鍵を対象とし，攻撃で利用できる暗号化パケットの個数が $10,000$，$20,000$，$40,000$，$80,000$ の場合で秘密鍵を復元する．σ_i の候補は T_i のカウント数が大きいものから 4 個の要素とする．すなわち，$d = 4$ であり，計算量は 4^{13} となる．

表 6.1 では提案手法と PTW 攻撃の復元できた秘密鍵の個数を示している．表より提案手法は PTW 攻撃より多くの秘密鍵を復元できていることが確認できる．$10,000$ 個の暗号化パケットが与えられるとき，PTW 攻撃では復元に成功しなかったが，提案手法では復元に成功した秘密鍵が存在する．さらに，$20,000$ 個の暗号化パケットが与えられるとき，提案手法は PTW 攻撃の 2 倍の秘密鍵を復元していることが確認できる．

文献 [27] の 7.1 節では，Tews らは ARP パケットから推測するキーストリームが確率 0.05 で誤る場合における実験を行っている．2 つ目の実験では，キーストリームが確率 P_{err} で誤る場合において復元できた秘密鍵の個数を得る．PTW 攻撃は文献 [27] と同様に $P_{err} = 0.05$ を選択する．まず，PTW 攻撃と同様の条件として提案手法で $P_{err} = 0.05$ を選択して実験を行う．表 6.2 より，このような条件においても提案手法で復元できる秘密鍵の個数は PTW 攻撃より多いことがわかる．また，提案手法では推測するキーストリームのサイズが PTW 攻撃と比べて大きくなっているため，場合によってはキーストリームが誤る確率が高くなることも考えられる．そこで，提案手法で $P_{err} = 0.08$ を選択した場合と PTW 攻撃で $P_{err} = 0.05$ を選択した場合の実験結果を比較する．表 6.2 より，提案手法で用いるキーストリームのほうが PTW 攻撃で用いるキーストリームより誤りやすい条件においても，提案手法で復元可能な鍵の個数が PTW 攻撃より多いことが確認できる．

表 6.2　確率 P_{err} でキーストリームが誤る場合における提案手法と PTW 攻撃によって復元できた秘密鍵の個数

観測パケット数	提案手法		PTW 攻撃
	$P_{err} = 0.05$	$P_{err} = 0.08$	$P_{err} = 0.05$
$10,000$	0	0	0
$20,000$	607	378	224
$40,000$	$33,247$	$27,059$	$23,870$
$80,000$	$93,922$	$91,672$	$90,392$

6.6　結言

本章では，攻撃に利用できるキーストリームを先頭の 15 バイトから先頭の 22 バイトまで拡張することで PTW 攻撃を改良する方法を提案した．提案手法では，キーストリームの先頭の 22 バイトを有効に用いるために，キーストリームの 19 バイト目から 22 バイト目に対応する推測値をキーストリームの 3 バイト目から 6 バイト目に対応する推測値に変換している．与えられる暗号化パケットの個数が同じとき，提案手法は PTW 攻撃より高確率で秘密鍵を復元できる．特に，$20,000$ 個の暗号化パケットが与えられるときには，提案手法は 104 ビットの秘密鍵を PTW 攻撃の 2 倍程度の確率で復元可能である．したがって，通信量が小さい場合において秘密鍵が復元される可能性が大きくなり，WEP の脆弱性は更に増大したといえる．

第7章　結論

本論文では商用のアプリケーションや標準プロトコルで広く使われているストリーム暗号である RC4 に注目し，その安全性を解読法の議論によって評価した．RC4 の解読法に関する研究は四つのテーマから成る．それらは，初期状態からセッション鍵を復元する方法 (第 3 章)，内部状態復元攻撃の効率化 (第 4 章)，WEP の新しい weak IV を利用した鍵復元攻撃 (第 5 章)，WEP に対する既知 IV 攻撃の改良 (第 6 章) である．

第 3 章では RC4 の初期状態の一部からセッション鍵を復元する方法を提案した．まず，RC4 の KSA における内部状態の撹拌が不十分であるという性質に注目し，初期状態の 2 個 (または 1 個) の要素からセッション鍵の 1 個の要素を有意な確率で復元する線形方程式を与えた．さらに，その線形方程式を効果的に用いるためのパラメータについて考察し，セッション鍵長と同じサイズの初期状態のデータからセッション鍵を復元する方法を与えた．この方法に要する計算量は KSA を一回実行する程度と少ない．また，提案した方法によって初期状態から復元可能なセッション鍵は無視できない割合で存在する．具体的には，128 ビットのセッション鍵を用いる場合，提案した方法を用いれば $1/2^{24.4}$ の割合の鍵を復元できる．これは，将来的に内部状態復元攻撃によって初期状態の一部が復元可能になったとき，高確率でセッション鍵を復元できることを意味している．

第 4 章では，RC4 に対する汎用的な内部状態復元攻撃の中で最も有効であった Knudsen らの方法に注目し，その効率化を行った．Knudsen らの方法はキーストリームの情報を利用する枝刈りを備えた木探索アルゴリズムである．提案手法では利用できるキーストリームのデータが多くなるように Knudsen らのアルゴリズムを改良し，枝刈りの機会を増加させることによって計算量を削減している．また計算機実験によって，初期状態の先頭の 73 個の要素が既知のときに，提案手法によって 2^{20} の計算量で復元可能な初期状態が存在することを明らかにした．既知の要素が無い条件において，この初期状態を復元する場合，73 個の要素を総当たりで得るとするならば復元に要する計算量は 2^{587} となる．同様の評価をするとき，Knudsen らの方法では 2^{797} の計算量で復元できる初期状態の存在しか知られていないことから，提案手法によって内部状態復元攻撃に対する RC4 の安全性が低下したといえる．

第 5 章では，WEP の新しい weak IV を特定し，それを利用する鍵復元攻撃を提案した．まず，高確率で生じる KSA の状態遷移に注目し，従来の weak IV と比べて秘密鍵との相関が比較的小さくなる weak IV を特定した．特定した weak IV は与えられている条件が少なくなるため，IV 全体の約 81%という大きな割合で存在する．この weak IV を取り除く場合には実用上の問題が生じてしまうため，WEPplus による対策は困難であると考えられる．次に，この weak IV を効果的に利用して秘密鍵を復元する方法を与えた．提案手法では，weak IV が持っ

ている秘密鍵との相関の大きさが秘密鍵の値に依存して変化することに注目し，相関の大きな weak IV だけを選別することで秘密鍵を復元できる確率を向上させる．具体的には，weak IV とキーストリームから攻撃方程式を用いて得られた推測値を利用して weak IV を選別する．さらに，効果的な weak IV のフィルタリングをしている WEPplus に対して提案手法が有効であることを計算機実験によって確認した．

第 6 章では WEP に対する既知 IV 攻撃の中で最も効果的な方法である PTW 攻撃を改良した．PTW 攻撃ではキーストリームの先頭の 15 バイトと任意の IV を利用する攻撃方程式を用いて秘密鍵を復元する．提案手法では攻撃に利用できるキーストリームのサイズを 22 バイトまで拡張し，1 個のパケットから得られる秘密鍵の情報を増加させる．さらに，計算機実験によって，観測できる暗号化パケット数が等しい条件の下で提案手法が PTW 攻撃よりも高確率で秘密鍵を復元できることを確認した．特に 20,000 個の暗号化パケットを観測できる場合において，提案手法で 104 ビットの秘密鍵を復元できる確率は PTW 攻撃の 2 倍程度まで向上した．

以上のように，本論文では RC4 及び WEP の安全性に対する新たな評価を与えた．本論文の研究成果は RC4 の安全性の厳密な評価及び RC4 と同様の構造を持つストリーム暗号における評価に利用できる．実際に，初期状態の一部からセッション鍵を復元する方法は，キーストリームから初期状態の一部を復元する方法と組み合わされて，セッション鍵を効率的に復元する攻撃で利用されている [55, 56]．また，RC4 と同様の構造を持つストリーム暗号である Py においては，その KSA の可逆性の解析において本研究成果の一部を応用している [57, 58]．さらに内部状態復元攻撃に関しては，RC4 と同様の構造を持つ多数のストリーム暗号において Knudsen らの方法に基づく評価が行われていることが多く，本論文で提案した方法で再評価することによって厳密な安全性を評価できると考えられる．以上のことをまとめると，本論文は RC4 及び RC4 と同様の構造を持つ多数のストリーム暗号に対する安全性評価において貢献し得ると考えられる．

謝　辞

本研究の機会を与えて頂いた神戸大学大学院工学研究科電気電子工学専攻森井昌克教授には，熱心な御指導と貴重な御助言を賜わりました．ここに謹んで感謝の意を表します．

本研究をまとめるにあたり貴重な御教示と多くの御助言を頂きました神戸大学大学院工学研究科電気電子工学専攻 小川真人教授に深く感謝致します．

本研究を論文としてまとめるにあたって貴重な御教示と御助言を頂きした神戸大学大学院工学研究科電気電子工学専攻 増田澄男教授に深く感謝致します．

また，本研究を遂行するにあたり，数々の有益なる御助言を賜わり，御討論頂きました名古屋工業大学大学院工学研究科 白石善明准教授に心より感謝致します．

更に，筆者は神戸大学大学院工学研究科電気電子工学専攻 桑門秀典准教授，栗林稔助教に日頃より御助言，御討論頂きました．ここに深謝の意を表します．

日頃から研究における様々な便宜を図ってくださいました神戸大学大学院工学研究科電気電子工学専攻 原田和男技術職員に深く感謝致します．

本研究を推進するにあたって日頃からの暖かい励ましと貴重な御助言を頂きました，岐阜大学総合情報メディアセンター 毛利公美准教授，鳴門教育大学高度情報研究センター 曽根直人准教授，徳島大学工学部知能情報工学科 得重仁講師，愛知教育大学教育学部 福田洋治助教，財団法人ひょうご情報教育機構 廣友雅徳専任研究員に心より感謝致します．また，本研究をまとめるにあたって暖かい励ましを頂きました神戸大学大学院工学研究科電気電子工学専攻情報通信研究室の諸氏に心より感謝致します．

最後に，大学生活を支えて頂き，博士後期課程への進学にも理解を示してくれた両親と家族に心より感謝致します．

参考文献

[1] 独立行政法人 情報処理推進機構, “情報漏えいインシデント対応方策に関する調査,” 2007, available at `http://www.ipa.go.jp/security/awareness/johorouei/report.pdf`

[2] Scan NetSecurity, “Web 改ざん情報,” available at `https://www.netsecurity.ne.jp/9.html`

[3] B. Schneier, Applied Cryptography, Wiley, New York, 1996.

[4] R. A. Rueppel, Analysis and Design of Stream Ciphers, Springer-Verlag, Berlin, 1986.

[5] R. A. Rueppel, “Stream ciphers,” Contemporary Cryptology, ed. G. J. Simmons, pp.65–134, IEEE Press, New York, 1992.

[6] J. Dj. Golić, “Linear statistical weakness of alleged RC4 keystream generator,” *Proc. EUROCRYPT'97*, Lecture Notes in Computer Science, vol.1233, pp.226–238, Springer-Verlag, 1997.

[7] S. Fluhrer and D. McGrew, “Statistical analysis of the alleged RC4 keystream generator,” *Proc. FSE2000*, Lecture Notes in Computer Science, vol.1978, pp.19–30, 2001.

[8] I. Mantin and A. Shamir, “A practical attack on broadcast RC4,” *Proc. FSE2001*, Lecture Notes in Computer Science, vol.2355, pp.152–164, Springer-Verlag, 2001.

[9] S. Fluhrer, I. Mantin, and A. Shamir, “Weaknesses in the key scheduling algorithm of RC4,” *Proc. SAC2001*, Lecture Notes in Computer Science, vol.2259, pp.1–24, Springer-Verlag, 2001.

[10] S. Paul and B. Preneel, “A new weakness in the RC4 keystream generator and an approach to improve the security of the cipher,” *Proc. FSE2004*, Lecture Notes in Computer Science, vol.3017, pp.245–259, Springer-Verlag, 2004.

[11] I. Mantin, “Predicting and distinguishing attacks on RC4 keystream generator,” *Proc. EUROCRYPT2005*, Lecture Notes in Computer Science, vol.3494, pp.491–506, Springer-Verlag, 2005.

[12] L. R. Knudsen, W. Meier, B. Preneel, V. Rijmen, and S. Verdoolaege, “Analysis methods for (alleged) RC4,” *Proc. ASIACRYPT'98*, Lecture Notes in Computer Science, vol.1514, pp.327–341, Springer-Verlag, 1998.

[13] S. Mister and S. E. Tavares, “Cryptanalysis of RC4-like ciphers,” Proc. *SAC'98*, Lecture Notes in Computer Science, vol.1556, pp.131–143, Springer-Verlag, 1999.

[14] J. Dj. Golić, "Iterative probabilistic cryptanalysis of RC4 keystream generator," *Proc. ACISP2000*, Lecture Notes in Computer Science, vol.1841, pp.220–233, Springer-Verlag, 2000.

[15] S. Paul and B. Preneel, "Analysis of non-fortuitous predictive states of the RC4 keystream generator," *Proc. INDOCRYPT2003*, Lecture Notes in Computer Science, vol.2904, pp.52–67, Springer-Verlag, 2003.

[16] IEEE Computer Society, "Wireless lan medium access control (MAC) and physical layer (PHY) specifications," IEEE Std 802.11, 1999.

[17] S. Fluhrer, I. Mantin, and A. Shamir, "Attacks on RC4 and WEP," *CryptoBytes*, vol.5, no.2, pp.26–34, RSA Laboratories, 2002.

[18] A. Freier, P. Karlton, and P. Kocher, "The SSL 3.0 protocol," Netscape Communications, Nov. 1996.

[19] Wi-Fi Alliance, "Wi-Fi protected access," available at `http://www.weca.net/opensection/protected_access.asp`

[20] Orinoco, "WEPplus white paper," Oct. 2001.

[21] Wepcrack, available at `http://sourceforge.net/projects/wepcrack/`.

[22] K. Kobara and H. Imai, "Yet other weak IVs for recovering wep keys," *Proc. ISITA2004*, pp.1130–1134, Parma, Italy, Oct. 2004.

[23] KoreK, "Next generation of WEP attacks?," 2004, available at `http://www.netstumbler.org/showpost.php?p=93942&postcount=35`

[24] K. Kobara and H. Imai, "Key-dependent weak IVs and weak keys in WEP — how to trace conditions back to their patterns," *IEICE Trans. Fundamentals*, vol.E89-A, no.8, pp.2198–2206, Aug. 2006.

[25] I. Mantin, "A practical attack on the fixed RC4 in the WEP mode," *Proc. ASIACRYPT2005*, Lecture Notes in Computer Science, vol.3788, pp.395–411, Springer-Verlag, 2005.

[26] A. Klein, "Attacks on the RC4 stream cipher," *submitted to Designs, Codes and Cryptography*, 2007.

[27] E. Tews, R. Weinmann, and A. Pyshkin, "Breaking 104 bit WEP in less than 60 seconds," *Cryptology ePrint*, 2007, available at `http://eprint.iacr.org/2007/120.pdf`

[28] D. Bleichenbacher and S. Patel, "SOBER cryptanalysis," *Proc. FSE'99*, Lecture Notes in Computer Science, vol.1636, pp.305–316, Springer-Verlag, 1999.

[29] P. Hawkes and G. Rose, "Exploiting multiples of the connection polynomial in word-oriented stream cipher," *Proc. ASIACRYPT2000*, Lecture Notes in Computer Science, vol.1976, pp.302–316, Springer-Verlag, 2000.

[30] P. Hawkes and G. Rose, "Guess-and-determine attacks on SNOW," *Proc. SAC2002*, Lecture Notes in Computer Science, vol.2595, pp.37–46, Springer-Verlag, 2003.

[31] S. Kiyomoto, T. Tanaka, and K. Sakurai, "Experimental analysis of guess-and-determine attacks on clock-controlled stream ciphers," *IEICE Trans. Fundamentals*, vol.E88-A, no.10, pp.2778–2791, Oct. 2005.

[32] H. Ahmadi, T. Eghlidos, and S. Khazaei, "Improved guess and determine attack on SOSEMANUK," *ECRYPT Stream Cipher Project*, Report 2005/085, 2005, available at `http://www.ecrypt.eu.org/stream/papersdir/085.pdf`

[33] Y. Tsunoo, T. Saito, M. Shigeri, T. Suzaki, H. Ahmadi, T. Eghlidos, and S. Khazaei, "Evaluation of SOSEMANUK with regard to guess-and-determine attacks," *ECRYPT Stream Cipher Project*, Report 2006/009, 2006, available at `http://www.ecrypt.eu.org/stream/papersdir/2006/009.pdf`

[34] J. Mattsson, "A guess-and-determine attack on the stream cipher Polar Bear," *ECRYPT Stream Cipher Project*, Report 2006/017, 2006, available at `http://www.ecrypt.eu.org/stream/papersdir/2006/017.pdf`

[35] H. Wu and B. Preneel, "Cryptanalysis of the stream cipher DECIM," *Proc. FSE2006*, Lecture Notes in Computer Science, vol.4047, pp.30–40, Springer-Verlag, 2006.

[36] H. Wu and B. Preneel, "Differential cryptanalysis of the stream ciphers Py, Py6 and Pypy," *Proc. EUROCRYPT2007*, Lecture Notes in Computer Science, vol.4515, pp.276–290, Springer-Verlag, 2007.

[37] T. Isobe, T. Ohigashi, H. Kuwakado, and M. Morii, "How to break Py and Pypy by a chosen IV attack," *Workshop Record of SASC 2007*, pp.340–352, Jan./Feb. 2007.

[38] H. Wu and B. Preneel, "Differential-linear attacks against the stream cipher Phelix," *Proc. FSE2007*, Lecture Notes in Computer Science, vol.4593, pp.87–100, Springer-Verlag, 2007.

[39] P. Ekdahl and T. Johansson, "Distinguishing attacks on SOBER-t16 and t32," *Proc. FSE2002*, Lecture Notes in Computer Science, vol.2365, pp.210–224, Springer-Verlag, 2002.

[40] S. Paul, B. Preneel, and G. Sekar, "Distinguishing attacks on the stream cipher Py," *Proc. FSE2006*, Lecture Notes in Computer Science, vol.4047, pp.405–421, Springer-Verlag, 2006.

[41] K. Nyberg and J. Wallén, “Improved linear distinguishers for SNOW 2.0,” *Proc. FSE2006*, Lecture Notes in Computer Science, vol.4047, pp.144–162, Springer-Verlag, 2006.

[42] B. Zoltak, “VMPC one-way function and stream cipher,” *Proc. FSE2004*, Lecture Notes in Computer Science, vol.3017, pp.210–225, Springer-Verlag, 2004.

[43] R. J. Jenkins Jr.,“ISAAC,” *Proc. FSE’96*, Lecture Notes in Computer Science, vol.1039, pp.41–49, Springer-Verlag, 1996.

[44] E. Biham and J. Seberry, “Py (Roo): a fast and secure stream cipher using rolling arrays,” *ECRYPT Stream Cipher Project*, Report 2005/023, 2005, available at `http://www.ecrypt.eu.org/stream/papersdir/023.pdf`

[45] E. Biham and J. Seberry, “Pypy: another version of Py,” *ECRYPT Stream Cipher Project*, Report 2006/038, 2006, available at `http://www.ecrypt.eu.org/stream/papersdir/2006/038.pdf`

[46] E. Biham and J. Seberry, “Tweaking the IV setup of the Py family of stream ciphers — the ciphers TPy, TPypy, and TPy6,” 2007, available at `http://www.cs.technion.ac.il/~biham/Reports/Py/tpy.ps.gz`

[47] Y. Nawaz, K. C. Gupta, and G. Gong, “A 32-bit RC4-like keystream generator,” *Cryptology ePrint*, 2005, available at `http://eprint.iacr.org/2005/175.pdf`

[48] N. Keller, S. D. Miller, I. Mironov, and R. Venkatesan, “MV3: a new word based stream cipher using rapid mixing and revolving buffers,” *Proc. CT-RSA2007*, Lecture Notes in Computer Science, vol.4377, pp.1–19, Springer-Verlag, 2007.

[49] A. Stubblefield, J. Ioannidis, and A. D. Rubin, “A key recovery attack on the 802.11b wired equivalent privacy protocol (WEP),” *ACM Trans. Information and System Security*, vol.7, no.2, pp.319–332, May 2004.

[50] AirSnort, available at `http://airsnort.shmoo.com/`.

[51] Aircrack, `http://www.cr0.net:8040/code/network/aircrack/`

[52] Dwepcrack, `http://www.e.kth.se/~pvz/wifi/`

[53] Weplab, `http://weplab.sourceforge.net/`

[54] 吉田雅徳, 古原和邦, 今井秀樹, “弱 IV に基づく鍵回復攻撃に対して安全な WEP の実装,” 2006 年暗号と情報セキュリティシンポジウム (SCIS2006), p.239, Jan. 2006.

[55] 寺村亮一，大東俊博，桑門秀典，森井昌克，“RC4 におけるキーストリームからの内部状態推定に関する考察,” 電子情報通信学会技術研究報告，OIS 研究会, vol.107, no.230, pp.71–76, Sept. 2007.

[56] 寺村亮一，大東俊博，桑門秀典，森井昌克，“RC4 における新しい weak key,” 電子情報通信学会技術研究報告，ISEC 研究会, vol.107, no.346, pp.75–80, Nov. 2007.

[57] 五十部孝典，大東俊博，桑門秀典，森井昌克，“ストリーム暗号 Py の鍵回復攻撃に関する一考察,” 情報通信システムセキュリティ (ICSS) 研究会, pp.5–10, Apr. 2006.

[58] 五十部孝典，大東俊博，桑門秀典，森井昌克，“Py に対する鍵回復攻撃の実現について,” 電子情報通信学会技術研究報告，ISEC 研究会, vol.106, no.175, pp.141–148, Jul. 2006.

関連発表

学術論文

[1] Y. Shiraishi, T. Ohigashi, and M. Morii, “Internal-state reconstruction of a stream cipher RC4,” *IEICE Trans. Fundamentals*, vol.E86-A, no.10, pp.2636-2638, Oct. 2003.

[2] T. Ohigashi, Y. Shiraishi, and M. Morii, “New weakness in the key-scheduling algorithm of RC4,” *IEICE Trans. Fundamentals*, vol.E91-A, no.1, Jan. 2008. (掲載予定)

[3] T. Ohigashi, Y. Ozasa, Y. Fujikawa, H. Kuwakado, and M. Morii, “Improvement on the Tews-Weinmann-Pyshkin attack on WEP,” *IEICE Electronics Express (ELEX)*. (投稿中)

国際学会

[1] Y. Shiraishi, T. Ohigashi, and M. Morii, “An improved internal-state reconstruction method of a stream cipher RC4,” *Proc. The IASTED International Conference on Communication, Network, and Information Security, CNIS 2003*, pp.132-135, Dec. 2003.

[2] T. Ohigashi, Y. Shiraishi, and M. Morii, “FMS attack-resistant WEP implementation is still broken — most IVs leak a part of key information,” *Proc. 2005 International Conference on Computational Intelligence and Security, CIS 2005*, Part II, LNAI 3802, pp.17-26, Springer-Verlag, Dec. 2005.

[3] T. Ohigashi, Y. Shiraishi, and M. Morii, “A categorizing-guessed-values approach for the key recovery attack against WEP,” *Proc. International Symposium on Information Theory and its Applications, ISITA2006*, pp.403-408, Oct./Nov. 2006.

受賞

[1] “SCIS20 周年記念賞,” 電子情報通信学会情報セキュリティ研究専門委員会，2003 年 9 月 19 日.
(受賞論文)
大東俊博，白石善明，森井昌克，“RC4 の内部状態推定法に関する考察,” 2003 年 暗号と情報セキュリティシンポジウム (SCIS2003)，vol.1，pp.447-452, 2003 年 1 月.

学術講演

[1] 大東俊博，白石善明，森井昌克，“RC4 の効率的な内部状態推定法,” 第 25 回情報理論とその応用シンポジウム (SITA2002)，vol.2，pp.607-610, 2002 年 12 月.

[2] 大東俊博，白石善明，森井昌克，“RC4 の内部状態推定法に関する考察,” 2003 年 暗号と情報セキュリティシンポジウム (SCIS2003)，vol.1，pp.447-452, 2003 年 1 月.

[3] 大東俊博，白石善明，森井昌克，“RC4 の弱鍵に関する考察,” Computer Security Symposium 2003 (CSS2003) vol.2003，No.15，pp.235-240, 2003 年 10 月.

[4] 大東俊博，白石善明，森井昌克，“RC4 の秘密鍵を導出する計算量について,” 第 26 回情報理論とその応用シンポジウム (SITA2003)，vol.1，pp.37-40, 2003 年 12 月.

[5] 大東俊博，白石善明，森井昌克，“RC4 から生成されるセッション鍵を導出するための計算量について,” 2004 年 暗号と情報セキュリティシンポジウム (SCIS2004)，vol.2，pp.1345-1350, 2004 年 1 月.

[6] 大東俊博，白石善明，森井昌克，“内部状態遷移型ストリーム暗号に対する確率的な内部状態推定法,” 電子情報通信学会技術研究報告，OIS 研究会, pp.1-6, 2004 年 5 月.

[7] 大東俊博，白石善明，森井昌克，“内部状態遷移型ストリーム暗号に対する確率的な内部状態推定法に関する考察,” 電子情報通信学会技術研究報告，ISEC 研究会, pp.41-48, 2004 年 7 月.

[8] T. Ohigashi, Y. Shiraishi, and M. Morii, "Most IVs of FMS attack-resistant WEP implementation leak secret key information," *Proc. the 2005 Symp. on Cryptography and Information Security, SCIS2005*, pp.1957-1962, Jan. 2005.

[9] 大東俊博，白石善明，森井昌克，"FMS 攻撃に耐性のある WEP 実装に対する選択 IV 攻撃," 電子情報通信学会技術研究報告，ISEC 研究会, pp.19-24, 2005 年 3 月.

[10] 大東俊博，白石善明，森井昌克，"FMS 攻撃に耐性のある WEP 実装に対する鍵復元攻撃の成功確率向上について," 2006 年 暗号と情報セキュリティシンポジウム (SCIS2006)，p.267, 2006 年 1 月.

[11] Y. Ozasa, Y. Fujikawa, T. Ohigashi, H. Kuwakado, and M. Morii, "A study on the Tews-Weinmann-Pyshkin attack against WEP," *Technical Report of IEICE, ISEC*, pp.17-21, Jul. 2007.

[12] T. Ohigashi, H. Kuwakado, and M. Morii, "A key recovery attack on WEP with less packets," *Technical Report of IEICE, ISEC*, pp.61-68, Nov. 2007.