暗号化と復号の計算量がアンバランスな共通鍵暗号の提案とその応用

清水 隆太郎[1,a)]　五十部 孝典[1]

概要：本稿では，暗号化と復号の計算量が非対称な共通鍵暗号の提案と，提案手法を用いたアプリケーションへの提案を行う．従来の共通鍵暗号に対して，提案手法は復号者の計算量が非常に大きく，共有する鍵長に対して実際に用いる鍵長が長い特性がある．前者の特性を用いることで，既存の対策手法では難しい中間者攻撃に対する検知を行うアプリケーションの提案を行う．更に，後者の特性を用いることで既存の共有鍵の鍵長拡張を行う手法の提案を行う．

キーワード：共通鍵暗号，中間者攻撃, ブロック暗号, 鍵長

A Symmetric Key Scheme with Unbalanced Computational Cost and Applications

Ryutaro Shimizu[1,a)]　Takanori Isobe[1]

Abstract: In this paper, we propose a symmetric key cryptosystem with asymmetric computational complexity for encryption and decryption, and an application of the proposed method to applications. Compared with conventional symmetric key cryptography, the proposed method requires a very large amount of computation by the decryptor and the actual key length used is longer than the shared key length. We propose an application that detects man-in-the-middle attacks, which are difficult to detect with existing countermeasure methods, by using the former property. Furthermore, we propose a method to extend the key size of an existing fixed-length key by using the latter property.

Keywords: Symmetric key scheme, Man in the middle attack, Block Cipher,Key length

1. はじめに

本稿では，暗号化と復号の計算量が非対称な共通鍵暗号の提案と, 提案手法を用いたアプリケーションの提案を行う．既存の共通鍵暗号方式では，暗号化及び復号に掛かる計算量はほぼ等しい．対して，本稿で述べる提案手法では復号に掛かる計算量を暗号化に掛かる計算量に対して極端に大きくすることができる．提案手法の有する暗号化と復号に掛かる計算量が非対称であるという性質を用いることで，一般的な共通鍵暗号における中間者攻撃の検知が可能である．中間者攻撃に対する既存の対策として電子証明書による署名等が挙げられるが，電子証明書の脆弱性を悪用した攻撃が発生している．また中間者が共通鍵を奪取し中間者攻撃に成功した後に検知を行う手法は限られており，Yisroel[1] らによる特殊なハードウェアを用いた解析や，Italo[2] らによる電子証明書の改良などが行われている．

本稿で述べる提案手法は，ソフトウェアの改良とワンタイムパスワード (OTP) や physical unclonable function(PUF) を組み合わせて用いることで，中間者攻撃の検知を実現する．事前に共有する鍵と乱数列を組み合わせて暗号鍵とすることで，復号時には乱数列の探索を要求する．中間者が復号のために乱数列の探索を行っても平文に OTP や PUF の出力値が用いられている場合，復号した値が正しい値か

[1] 兵庫県立大学応用情報科学研究科
Graduate School of Applied Informatics, University of Hyogo
[a)] aa19l504@ai.u-hyogo.ac.jp

検証できず攻撃に失敗する．

また提案手法は事前に共有する鍵に対して実際に暗復号に用いる鍵長が長くなるという性質を有する．共通鍵暗号では共有鍵がハードウェアに書き込まれて保存される場合など容易に共有鍵を更新できない場合も多く，演算機の性能向上による鍵長の危殆化が心配される．そこで提案手法の前述の性質を用いてソフトウェアの更新のみで鍵長の拡張を行う手法を提案する．提案手法の導入によって送信者が負担する追加計算量は小さく，受信者が負担する追加計算量は比較的大きいため，IoT 端末とサーバー間での通信などが想定される．

2. 計算量がアンバランスな共通鍵暗号

2.1 定義

共通鍵暗号では暗号化処理に対して復号処理では暗号化の逆変換処理を行うアルゴリズムが一般的である．そのため，暗号化と復号で要する計算量が大きく異なることは無い．本稿では，暗号化と復号に要する計算量が大きく異なる共通鍵暗号を提案する．具体的には，図 1 の様に平文 M を共通鍵 K を用いて暗号文 C に暗号化することを考える．この際，暗号化関数 ENC の計算量を 1 とした時，復号関数 DEC の計算量が 2^{α} となるものである．なお，本稿では断りがない限り計算量といえば時間計算量のことを示し，領域計算量については扱わないものとする．

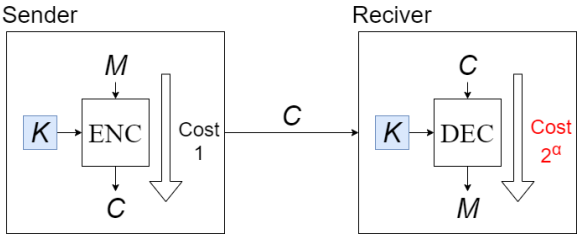

図 1 暗号化と復号の計算量が非対称な共通鍵暗号の定義

2.2 構成例

提案する暗号化と復号の計算量が非対称な共通鍵暗号の構成例を図 2 に示す．送信者は，共有鍵 K と乱数 r を生成し結合させた鍵 $K_x = (K||r)$ を用意し，鍵 K_x をマスターキーのように扱い鍵導出関数 KDF に入力することで暗号化に用いる鍵を導出する．更に受信者が正しい r の値を推測できたことを識別するために，平文の先頭 $n$bit を 0 で埋めておく．受信者が復号する際には暗号化処理と同様にして KDF に K_x を入力することで復号処理を行うが，受信者は r の値を推測するために総当たりで探索を行う必要があり復号に掛かる計算量が増大する．

Algorithm1 に具体的な手順を示す．送信者は，共有鍵 K と乱数 $r(n_r\text{bit})$ を結合させた鍵 $K_x = (K||r)$ を用意する．送信するメッセージ m の先頭 $n\text{bit}(n > n_r)$ を 0 で埋め，平文 $M = (00...0||m)$ とする．平文を暗号化する際には，鍵 K_x をマスターキーのように扱い鍵導出関数 KDF に入力し暗号文 C を生成する．受信者が暗号文を復号する際には暗号化処理と同様にして KDF に K_x を入力することで復号処理を行う．このとき乱数 r が真性乱数または暗号学的疑似乱数であれば，受信者は鍵 k_x を得る為に乱数 r を総当たりで推測しなければならない．受信者は，推定した r を元に復号した平文 M' の先頭 $n$bit が 0 であることを確認して送信者が送った平文 M と同一であることを認める．

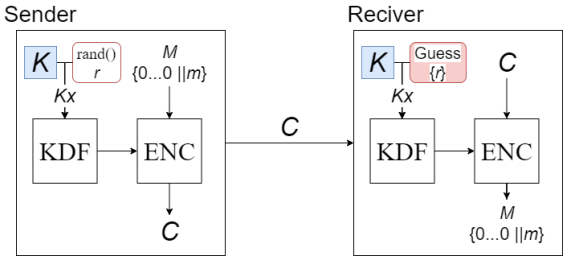

図 2 暗号化と復号の計算量が非対称な共通鍵暗号

Algorithm 1 暗号化と復号の計算量が非対称な共通鍵暗号

1: 事前に送信者と受信者は共有鍵 K を共有する．
2: 送信者は共有鍵 K と乱数 $r(n_r\text{bit})$ を結合させた鍵 $Kx = (K||r)$ を用意する．
3: 送信者は平文の先頭 $n\text{bit}(n > n_r)$ を 0 で埋め，鍵 Kx を用いて暗号化する
4: 受信者は受け取った暗号文に対する鍵 Kx を推測するために，乱数 r を総当たりで探索する
5: 受信者は推定した乱数 r' を使用した鍵 $Kx' = (K||r')$ を用いて復号し，先頭 $n$bit が 0 であれば乱数 r' を暗号化に使用された乱数 r として認める

2.3 復号確率

提案手法では，復号を行うために乱数 r を総当たりで探索する．最小 1 回，最大で乱数 r として取り得る全ての値について 2^{n_r} 回，試行を行えば必ず復号可能である．

復号に要する平均試行回数について，i 度目の試行で乱数 r を当てる確率は $1/m$ である．

$$f\{X \leq 2^{n_r}\} = \begin{cases} 1/m & (x \neq 0) \\ 0 & (x = 0) \end{cases}$$

各試行は独立同一で連続的な一様分布に従うと仮定すると，中心極限定理より n_r が十分に大きいとき

$$P\{X \leq 2^{n_r}\} \simeq 1 - \exp^{-2^{n_r}/m}$$

の指数分布で近似される．

2.4 暗号化と復号の計算量

一連のアルゴリズムにおいて，送信者の暗号化に掛かる

計算量は暗号化関数の実行による計算量 O_e に加えて乱数 r を生成する計算量 O_{rg} が必要となる．受信者の復号に掛かる計算量は復号関数を実行する計算量 O_d に加えて乱数 r を総当たりで探索する計算量 O_{rb} が必要となる．乱数 r のビット数 n が十分に大きいとき $O_e, O_{rg}, O_d << O_{rb}$ が成立し，計算量 O_{rb} に対して他の計算量は無視できるほど小さいので，送信者の計算量に対して受信者の計算量は凡そ O_{rb} 余分にかかる．

実際に乱数 r の探索によって掛かる追加の計算時間を見積もる．AES に対して鍵を変更しながら総当たりで復号を試行した所要時間を表 1 にまとめる．128bit AES を用いて暗号化された暗号文に対して, 非常に高速な命令セットである AES NI[3] を用いて復号を行った．

実験機 (CPU,RAM)	探索ビット数	所要時間 (s)
i7-5500U, 8GB	2^{16}	Negligible
	2^{24}	5.8
	2^{32}	1398.1
i9-9900k, 128GB	2^{16}	Negligible
	2^{24}	2.9
	2^{32}	735.2
Xeon-8260, 256GB	2^{16}	Negligible
	2^{24}	0.8
	2^{32}	221.1

表 1 探索ビット数と所要時間

乱数 r のビットは調整可能であり，実験結果より現実的な時間で乱数 r の探索を終えることが可能である．実際のアプリケーションでは，復号者として計算資源の豊富なサーバー等を用いることが想定されるため，ある程度探索ビット数が増大しても現実的な時間で乱数 r を推定することが可能である．

3. 提案手法を用いた中間者攻撃の検知

本章では，提案手法を用いて中間者攻撃の検知を行うアプリケーションの提案を行う．

中間者攻撃とは，攻撃者が既存のネットワークに侵入し，通信の盗聴や改竄を行う攻撃である．この問題は広く研究が進められていて，様々な攻撃手法が存在することが知られている [4]．中間者攻撃への対策として，電子証明書による認証や第三者による通信路の監視等が挙げられているが，既存の対策手段の脆弱性を悪用した事例もあり，更なる対策が求められている．

既存の中間者攻撃への対策として電子証明書による認証や第三者による通信路の監視があるが，中間者攻撃自体を防ぐものが大半であり何らかの脆弱性を受け中間者攻撃が成功した後に，通信路に中間者が存在することを検知することは難しい．近年では，Yisroel[1] らによる特殊なハードウェアを用いた解析や，Italo[2] らによる電子証明書の改良などによって通信路内の中間者を検知する手法が提案されているが，専用のハードウェアが必要であったり複雑な演算による計算回路面積の増大などのコストを負う．

本章では, 提案手法によるソフトウェアの改良とワンタイムパスワード (OTP) や physical unclonable function(PUF) といった一定時間ごとに変化する出力を得る関数を組み合わせて用いることで，中間者攻撃の検知を実現する．

3.1 攻撃者の定義

中間者攻撃を行う上で，攻撃者の主な目標は平文の盗聴及び改竄である．攻撃者の能力として，送信者と受信者が鍵交換を行い共有鍵 K を共有する段階で既に中間者攻撃に成功し，送信者と受信者が互いに異なる鍵 K_1, K_2 をそれぞれ中間者と共有しているとする．更に改竄検知を行う仕組みが存在していないか，仕組み自体は存在しているが機能していないと仮定する．一般的な共通鍵暗号によって送信者が平文 M を共有鍵 K_1 を用いて暗号化した暗号文を C とする．攻撃者は鍵交換の段階で共有鍵 K_1 を送信者と共有しているため，暗号文 C を復号して平文 M の盗聴が可能である．更に攻撃者は復号した平文 M を元に改竄した平文 M' を作成し共有鍵 K_2 で暗号化した暗号文 C' を受信者に送信する．受信者は暗号文 C' が共有鍵 K_2 で復号できることを確認し，平文 M' が正規の送信者から送られたものであると誤認する．送信者と受信者は互いに鍵を共有している相手が正規の送信者及び受信者であると誤認識しているので中間者の存在を検知すること自体が困難である．よって攻撃者が鍵交換方式によって共有鍵 K_1, K_2 をそれぞれ送信者と受信者に共有しているとき，暗号文を復号することで平文の盗聴，復号した平文を変更して再び暗号化することで改竄が可能である．

3.2 提案手法を用いたアプリケーションの構成

2 章で提案した手法の持つ性質として，事前に共有する鍵長より実際に使用する鍵長が長い性質がある．この性質を利用して，共有鍵が攻撃者に知られている状況下でも攻撃者は実際に使用する鍵長と共有鍵の差分の bit 数を探索する必要があるアプリケーションの構成を行う．送信者と受信者は OTP や PFU といった事前に一定時間毎に出力が変化する関数 f を共有し，関数 f の出力を平文として用いる．送信者は平文を共有鍵と乱数 r を結合した鍵を用いて暗号化する．攻撃者が乱数 r の推定を行い暗号文を復号しても関数 f の内容を知らなければ，復号した平文が正しい f の出力であるか検証できず盗聴に失敗する．よって 2 章で提案した手法を導入し，OTP や PFU のような一定時間毎に出力が変化する関数と組み合わせることで，既に中間者攻撃を行っている攻撃者の検知が可能である．

提案手法を用いた共通鍵暗号に対する中間者攻撃の検知の概略を図 3 に示す．

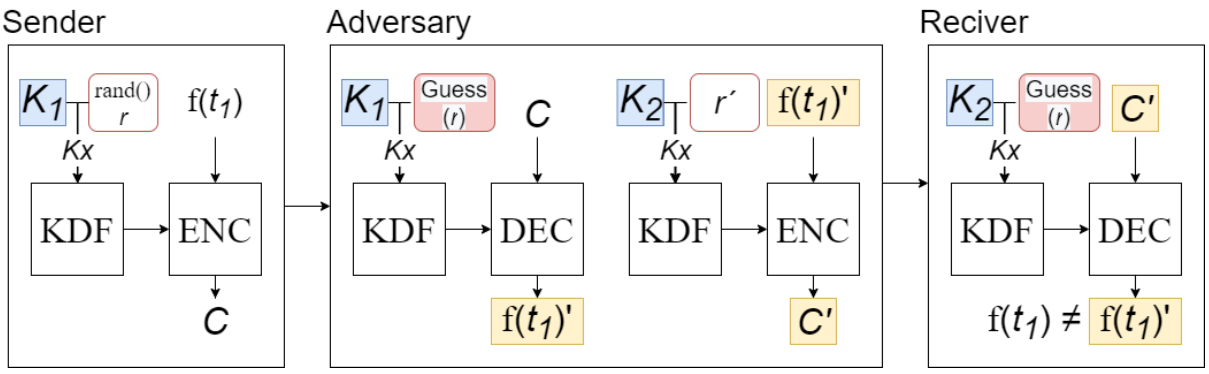

図 3 提案手法による共通鍵暗号に対する中間者攻撃の検知

Algorithm 2 提案手法による共通鍵暗号に対する中間者攻撃の検知

1: 事前に送信者と受信者は一定時間 T 毎に出力が変化する関数 f を共有する．
2: 送信者と受信者は，鍵交換方式によって共有鍵 K を共有する．
3: 送信者は共有鍵 K と乱数 $r(n_r\text{bit})$ を結合させた鍵 $K_x = (K||r)$ を用意する．
4: 送信者は平文として $f(t_1)$ を用いて，鍵 K_x を用いて暗号化する．
5: 受信者は暗号文を受け取った際に関数 f を用いて出力 $f(t_2)'$ を計算する．
6: 受信者は受け取った暗号文に対する鍵 K_x を推測するために，乱数 r を総当たりで探索する．
7: 受信者は推定した乱数 r' を使用した鍵 $K'_x = (K||r')$ を用いて平文 $f(t_1)'$ を復号し，$T >> f(t_2) - f(t_1)$ かつ $f(t_1)' = f(t_2)$ であるとき中間者は存在しないと認める．

3.2.1 アルゴリズムと評価

Algorithm2 に具体的な手順を示す．攻撃者が暗号文を復号するためには乱数 r を推定する必要があるが，関数 f の内容を知らなければ復号した平文が正しい f の出力であるか検証できない．このとき中間者が平文を得るには，暗号文を復号する際に適当に選択した乱数 r が正しい場合か，現在時刻の関数 f の正しい出力が推定できる場合に限られる．前者は乱数 r が $n_r$bit より確率 $1/2^{n_r}$ であり，後者は関数 f が攻撃者に対して秘匿されている限り攻撃者が関数 f の出力を知ることは出来ない．受信者は暗号文を受信した際に，関数 f を実行して得た出力と実際に暗号文を復号して得た平文が一致するか検証を行う．攻撃者は前述の通り暗号文を復号できないため，受信者の検証結果が不一致であれば攻撃者が存在することを検知できる．

既存手法では既に実行中の中間者攻撃の検知は困難であるが，2 章で提案した手法と一定時間毎に出力が変化する関数 f を組み合わせることで，共通鍵暗号に対する中間者攻撃の改竄検知が行える．一定時間毎に出力が変化する関数 f として, 送信者と受信者以外に秘匿が可能でかつ汎用性のある関数として OTP や PUF の利用が考えられる．OTP は提案手法を適用するアプリケーションの特性に依存せず利用することが可能で，PUF は機器固有の情報を使用するため回路規模の削減が可能で秘匿性にも優れる．

4. 提案手法を用いた共通鍵暗号の固定長鍵の拡張

本稿では，提案手法を用いて共有鍵の更新が行えない機器に対する暗号アルゴリズムの安全性向上を行うアプリケーションを提案する．

4.1 固定長鍵の危殆化

米国標準技術研究所 (NIST) により，暗号アルゴリズムの安全性と使用推奨期間が示されている [5]．使用推奨期間とは，その暗号アルゴリズムを採用した際に安全性が確保されると見込まれる期間を示す．2010 年までの使用推奨期間として 80bit 安全性を持つ暗号アルゴリズムが推奨されていた経緯があり，NIST の推奨に準じた暗号アルゴリズムを採用した 80bit 安全性を持つ暗号アルゴリズムの危殆化が心配されている．

解決策としてより鍵長の長い安全性を持った暗号アルゴリズムを採用することが最も安全な方法であるが，ハードウェア機器の内部に鍵が物理的に書き込まれている場合など，共有鍵の更新が不可能な場合が存在する．また，近年 IoT の普及に伴い注目されている軽量ハードウェア暗号では実装面積や演算速度の面から 80bit 鍵がサポートされているものも多いが，同様に共有鍵の更新が行えず危殆化が心配されている．既存の鍵長の危殆化は 80bit 鍵に限らず，今後も演算機の性能向上によって引き起こされうるため，共有鍵の更新が行えない機器に対する暗号アルゴリズムの安全性向上が課題である．

4.2 提案手法を用いた拡張

2 章で提案した手法の持つ性質として，事前に共有する鍵長より実際に使用する鍵長の方が長いことが挙げられる．そこで共有鍵の更新が不可能な機器に対して，提案手法を適用することで鍵長の更新をせずに実際に使用する鍵長を延ばすことで暗号アルゴリズムの安全性向上が期待できる．提案手法を用いた共通鍵暗号の固定長鍵の拡張の概略を図 4 に示す．

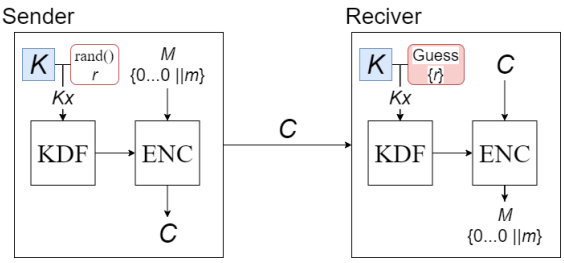

図 4 共通鍵暗号の固定長鍵の拡張

Algorithm 3 提案手法を用いた固定長鍵の拡張

1: 事前に送信者と受信者は共有鍵 K を共有する．
2: 送信者は共有鍵 K と乱数 $r(n_r\text{bit})$ を結合させた鍵 $K_x = (K||r)$ を用意する．
3: 送信者は平文の先頭 $n\text{bit}(n > n_r)$ を 0 で埋め，鍵 Kx を用いて暗号化する
4: 受信者は受け取った暗号文に対する鍵 K_x を推測するために，乱数 r を総当たりで探索する
5: 受信者は推定した乱数 r' を使用した鍵 $Kx' = (K||r')$ を用いて復号し，先頭 $n$bit が 0 であれば乱数 r' を暗号化に使用された乱数 r として復号に成功したことを認める．

4.3 アルゴリズム

Algorithm3 に具体的な手順を示す．送信者の暗号化に掛かる計算量は暗号化関数の実行による計算量 O_e に加えて乱数 r を生成する計算量 O_{rg} が必要となる．受信者の復号に掛かる計算量は復号関数を実行する計算量 O_d に加えて乱数 r を総当たりで探索する計算量 O_{rb} が必要となる．

一例として，共有鍵 K が 80bit で鍵長 K_x に 112bit 相当の安全性を得たいとき乱数 r は $K_x - K = 32\text{bit}$ となり受信者が復号を行うためには最小 1 回, 最悪 2^{32} 回, 復号関数を実行する必要がある．よって乱数 r を総当たりで探索する計算量 O_{rb} は復号関数を 1〜2^{32} 回実行する計算量となる．そのため，必要とするセキュリティ強度と受信者の演算能力を元に乱数 r の桁数を調整する必要がある．

4.4 提案アプリケーションの評価

提案手法を用いる事で共有する鍵長は K のまま，実際に使用する鍵長を $K_x(K_x > K)$ に拡張することができる．鍵長の拡張によって，共有鍵を更新が不可能な機器に対する暗号アルゴリズムの安全性向上が期待される．提案手法は送信者にとっては既存の共有鍵暗号アルゴリズムに加え，追加の計算として乱数 r を生成し共有鍵 K に結合する計算のみで済むため計算量の増加量は小さい．更に必要な回路規模や領域計算量についても同様に小さいと言える．一方で受信者にとっては既存の共有鍵暗号アルゴリズムに加え，追加の計算として乱数 $r(n_r\text{bit})$ の総当たり探索が必要で最悪の場合 2^{n_r} 回復号関数を実行する必要があり追加計算量は大きい．よって運用例としては，送信者として計算資源に乏しい IoT デバイス，受信者として計算資源が豊富なサーバーを用いることなどが想定される．

5. まとめ

本稿では暗号化と復号の計算量が非対称な共通鍵暗号の提案とアプリケーションの提案を行った．暗復号の計算量が非対称である性質を用いて，中間者攻撃の検知を行う手法の提案を行った．また，暗号化に用いる鍵長より復号に用いる鍵長が長いという特性から既存の共有鍵の鍵長拡張を行う手法の提案を行った．

6. 謝辞

本研究は科研費　挑戦的研究（萌芽）（20K21795）の助成を受けたものです．

参考文献

[1] Yisroel Mirsky, Naor Kalbo, Yuval Elovici, Asaf Shabtai. ”Vesper: using echo analysis to detect man-in-the-middle attacks in LANs.” IEEE Transactions on Information Forensics and Security 14.6 (2018): 1638-1653.
[2] Italo Dacosta, Mustaque Ahamad,atrick Traynor. ”Trust no one else: Detecting MITM attacks against SSL/TLS without third-parties.” European Symposium on Research in Computer Security. Springer, Berlin, Heidelberg, 2012.
[3] Intel AES-NI., https://software.intel.com/en-us/node/256280.
[4] Hernacki, Brian and Sobel, William E. ”Detecting man-in-the-middle attacks via security transitions.” U.S. Patent No. 8,561,181. 15 Oct. 2013.
[5] NISTSP800-57Pt.1Rev.4.,https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-57pt1r4.pdf.