コンピュータセキュリティ 14−19
(2001. 7. 25)

暗号アルゴリズム変化可能な暗号区評価モデルについて

梶崎 浩嗣† 黒川 恭一†

† 防衛大学校 情報工学科
〒 239-8686 神奈川県横須賀市走水 1-10-20

E-mail: †{g40073,kuro}@nda.ac.jp

あらまし データ通信における暗号化は，重要な要素であり通信ネットワークに多くの暗号アルゴリズムが用いられている．本研究では鍵によるデータの暗号化に加えて，動的に暗号区の暗号アルゴリズムを変化させるモデルを提案する．また，提案するモデルを実装するために近年高性能化が進んでいる再構成可能素子を用いたハードウェア実装の設計方針と性能評価方法について述べる．

キーワード 共通鍵暗号方式, 性能評価モデル, 再構成可能素子

Evaluation model of dynamically changing cryptographic algorithms using secret-key block cipher

Hirotsugu KAJISAKI† and Takakazu KUROKAWA†

† Department of Computer Science, National Defense Academy
1-10-20 Hasirimizu, Yokosuka-shi, 239-8686 Japan

E-mail: †{g40073,kuro}@nda.ac.jp

Abstract In a network communication, the cryptographic algorithms play role in information security. In this paper, a new approach to keep security level by dynamically changing cryptographic algorithm is presented. Then, a hardware design to implement an evaluation model using a reconfigurable device is shown. Furthermore its evaluation method is also explained.

Key words secret-key block cipher, evaluation model, reconfigurable device

1 まえがき

情報通信におけるデータの暗号化は，多数の研究がなされており，近年のインターネットの急速な発展に後押しされ，重要な要素の一つである．また，インターネットに代表されるデータ通信網を流れるデータ量は，電子メール等の文字情報に加えて音声や動画像のように莫大なデータ量を有するメディアが加わり，増大する傾向にある．

また，暗号に対する解読技術の研究も行われており特定の条件で解読される例も存在する．解読されないためには，強力な暗号アルゴリズムを使う必要があるが，その強度が様々な研究者間で認知されるには長い年月がかかる．

そこで，本研究では暗号の解読は可能という前提のもと，鍵と鍵長によるデータの保証というアプローチに加えて，積極的な暗号アルゴリズムの変更によるデータの保証に着目し，その実現を目指す．また，暗号アルゴリズムの実装方法は，ハードウエェアとソフトウェアそれぞれのアプローチがあったが，近年その両者の利点を活かして再構成可能素子を用いた実装が増加している．本研究においても，再構成可能素子を用いた設計方針を示す．

2 背景

これまで共通鍵暗号方式に関する研究は，その実装方法も含め多数提案されている．ここで実際の運用方法に着目すると，暗号アルゴリズムを固定し鍵を変化させてデータの安全性を保証する方式が大半である．この方法は，暗号化及び復号部分の大部分が共有可能であり，実装スペースの小規模化が可能という利点がある．しかし，暗号に対する解読方法は開発されてからの時間が経過するにつれて，高速化かつ高度化する傾向にあり，同じ暗号アルゴリズムを使い続けて，データを保証することは困難である．また，解読に強いアルゴリズムが開発されても，その評価には長い時間が必要とされている．

また近年，デバイス技術の進歩等により年々計算機の高速化と記憶容量の大容量化が進んでいる．暗号アルゴリズムの実装面に着目すると，それに伴い処理スピードの高速化が図れる反面，解読速度も高速化しており解読用専用計算機の研究も進んでいる [1][2]．

そこで本稿では，通信におけるデータの暗号化を固定した暗号アルゴリズムと鍵に委ねるのではなく，暗号アルゴリズムを変化させるというアプローチを試みるために，評価モデルのハードウェア実装にあたっての設計方針と性能評価方法について述べる．

2.1 これまでの実装

これまで研究された，ソフトウェア及びハードウェアへの暗号アルゴリズムの実装を表 1 にまとめる．これまで高速計算機を用いたソフトウェアによる実装と ASIC を用いたハードウェアへの実装が主であった．近年は，ゲートの接続を柔軟に変更できる FPGA(Field Programmable Gate Array) の高性能化が進み，この FPGA への実装が主になっている．米国次世代暗号 AES の評価も FPGA を用いて行われている．

表 1: 暗号アルゴリズムと実装方法

暗号アルゴリズム	実装方法	参考文献
DES	software	[3] [4]
	ASIC	[5]
AES	FPGA	[6] [7] [8] [9]
RC6	FPGA	[10]
CAST-256		
MISTY	software	[11]
Twofish	FPGA	[12]
IDEA	ASIC	[13]
RSA 用演算	ASIC	[14]

2.2 再構成可能素子への実装

近年，CPLD(Complex Programmable Logic Device) や FPGA に代表される再構成可能素子の集積度や動作速度が向上し，システムの中で有効に活用できるデバイスとして脚光を浴びている [15]．これまで使用可能ゲート数が少なかったため，試作開発や最終製品のごく一部として使用されていたが，ゲート数の増大と速度の向上からプロセッサの一部または再構成可能素子が主体となって機能する利用形態に変化している．

暗号アルゴリズムの実装に再構成可能素子を適応させると，

- 暗号及び復号処理の高速化
- 耐タンパ性の向上
- 計算機資源の節約

等の利点が生まれる．さらに再構成が可能であるという利点を積極的に活用することによって，単に処理速度の高速化のみではなく，暗号アルゴリズムを積極的に変化させることが可能となり，暗号アルゴリズムを有効に機能させることができる．

再構成可能可能素子の方式は，SRAM (Static Random Access Memory) 方式，アンチヒューズ方式，EPROM (Erasable Programmable Read Only Memory) 方式，EEPROM (Electrically Erasable Programmable Read Only Memory) 方式があり，それぞれ長所と短所を持ち合わせている．本研究では，頻繁な構成情報の変更が必要なことと，暗号アルゴリズムを高速に処理する必要があるために，利用できる論理セル数が多いSRAM方式のFPGAを用いる．

3 提案する性能評価モデル

3.1 共通鍵暗号

共通鍵暗号を用いた暗号通信の表現を，暗号アルゴリズムの変化に着目した場合について以下に示す．

共通鍵暗号方式は，平文を P，暗号化を $F_K(x)$，復号を $F_K^{-1}(x)$，鍵ブロックを K，暗号文を C と定義すると，

$$\text{暗号化} : C = F_K(P)$$

$$\text{復号} : P = F_K^{-1}(C)$$

と表すことが出来る．

ここで平文と暗号文，鍵のそれぞれのデータブロックを，

$$P = \{p_1, p_2, p_3, \cdots, p_i, \cdots, p_n\}$$

$$C = \{c_1, c_2, c_3, \cdots, c_i, \cdots, c_n\}$$

$$K = \{k_1, k_2, k_3, \cdots, k_i, \cdots, k_n\}$$

と表現する．このとき，i 番目のデータブロックに対する暗号処理は

$$\text{暗号化} : C_i = F_{K_i}(P_i)$$

$$\text{復号} : P_i = F_{K_i}^{-1}(C_i)$$

と表すことが出来る．

これまでは，この K を秘密にして保持しているのが大半であったが，我々は更に F, F^{-1} を変化させることに着目した．F, F^{-1} を変化させる方針としては，

1. i が変化する毎に F, F^{-1} を変化させる
2. i の一定間隔毎に F, F^{-1} を変化させる
3. 不規則な周期で F, F^{-1} を変化させる

という各方式が考えられる．併せて，共通鍵暗号方式を採用する場合，鍵の管理が重要な問題であり，暗号アルゴリズムを変化させた場合の鍵管理方式は実装方針と併せて検討する必要がある．なおここでは，暗号アルゴリズムを変化させた場合もそれぞれの鍵が管理されているものと仮定する．このとき暗号アルゴリズム F_i, F_i^{-1} は，

$$F_i = \{f_1, f_2, f_3, \cdots, f_i, \cdots, f_{fmax}\}$$

$$F_i^{-1} = \{f_1^{-1}, f_2^{-1}, f_3^{-1}, \cdots, f_i^{-1}, \cdots, f_{fmax}^{-1}\}$$

の要素をもつ．暗号アルゴリズムを1つづつ変化させる場合は，i に同期した F_i, F_i^{-1} を使って暗号化及び復号を行う．この状態は以下の式で表すことができる．

$$\text{暗号化} : C_i = F_{ik_i}[P_i]$$

$$\text{復号} : P_i = F_{ik_i}^{-1}[C_i]$$

3.2 暗号区

データ通信を暗号化する場合，データの内容や組織の違い等により用いる暗号アルゴリズムや鍵等が異なる．この異なりを表す用語として暗号区で表すことにする．暗号区とは，同じ暗号アルゴリズムと鍵管理方針にて暗号通信する区間と定義する．例えばノードA～Dがそれぞれ暗号通信を行う場合を図1に示す．ノードA～Dはそれぞれ暗号化及び復号ができる機能を有しており，それぞれのノード間の矢印が暗号通信を示している．ノードAとノードB及びノードAとノードCは同じ暗号アルゴリズムを用いており，かつ，同じ鍵管理方針でデータの秘匿化を行っている場合，ノードA，ノードB，ノードCは，同じ暗号区にあると言える．図では，1の網掛けで示した部分である．一方，ノードAとノードDは異なる暗号アルゴリズムと鍵管理方式でデータの秘匿化を行っており，図中2で示した異なる暗号区を構成している．

本研究では，更に暗号アルゴリズムを変化させるため，暗号アルゴリズムを変化させる方針が同じ場合という条件を含める必要がある．この条件も暗号区の定義に含めることとして，それぞれのノードが同じ暗号区に含まれる条件を以下に示す．

- 暗号化と復号アルゴリズムが等しい
- 鍵の管理方針が等しい
- 暗号アルゴリズムを変化させる方針が等しい

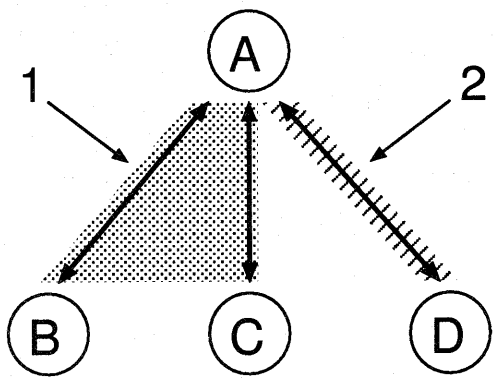

図 1: 暗号区の例

3.3 実装の前提

提案する評価モデルでは，広域かつ多数のノード上で試験及び評価を行うことを考慮し，通信プロトコルとしてインターネットプロトコルを使用する．また，ソースコードが公開されている PC-UNIX を利用した汎用計算機を基盤として開発する．これら実装にあたっての前提の概念図を図 2 に示す．

暗号区をこの実装前提に適応させた例を図 3 に示す．図中影の付いた四角で表現したノードが汎用計算機，実線矢印がノード間の通信を示している．先に示した暗号区に定義を数字で示し，同じ暗号区の含まれる通信を点線矢印で示している．同じ暗号区は，暗号アルゴリズム，鍵管理方針，暗号アルゴリズムを変化させる方針が等しいものとし，図では 1 ～ 3 の 3 種類あることを示している．同じ暗号区に含まれていることを各ノードの角の四角で表現しいている．例えば node 12 は，3 つの暗号区に含まれていることがわかる．この例では暗号区 1~3 が

暗号区 1 : $\{node11, node12\}$

暗号区 2 : $\{node12, node21, node22, node32\}$

暗号区 3 : $\{node12, node31, node32\}$

の各ノードで構成されいている．

3.4 設計方針

提案する暗号アルゴリズムを動的に変化させることのできる性能評価システムの設計方針を，3.3 節を踏まえて表 2 に示す．

この設計方針より，各ノードは，図 4 のように構成するものとした．

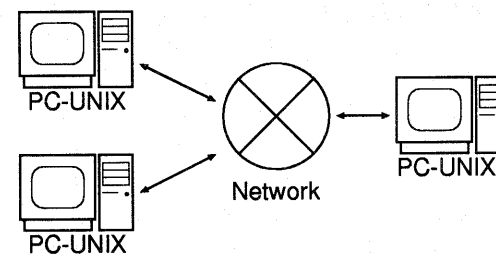

図 2: 実装の前提

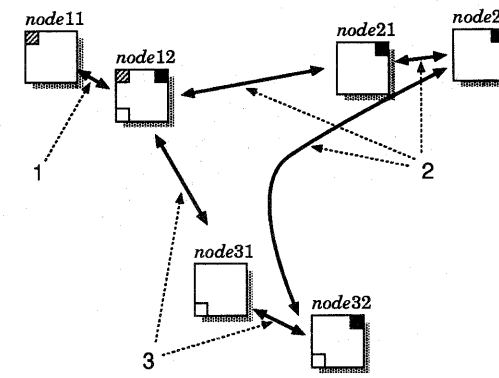

図 3: 実装時の暗号区

メッセージ生成器 (Message Generator) は，平文を生成する．平文の長さや発生頻度を変化させるため，PC-UNIX 上に実装されている`telnet`，`ftp` 等のコマンドを実行することによって発生するデータを平文として利用する．更に発生頻度が異なる大きな平文を生成するために，WWW(World Wide Web) 上の画像や音声データを WWW ブラウザーを使って閲覧することによって発生するデータも平文として利用する．

暗号モジュールコントローラ (Crypt module Con-

表 2: 設計方針

項 目	方 針
暗号の方式	公開されているブロック暗号
暗号処理	再構成可能素子
ブロック長	64 ビット以上
通信プロトコル	TCP/IP
通信インターフェース速度	10Mbps/100Mbps

troller) は，ネットワークインターフェース上の再構成可能素子に必要な暗号化もしくは復号のための構成情報を送信する．また，ネットワークに送受信される平文もしくは暗号文と同期をして構成情報を変更するためのソフトウェアである．

メッセージ判定器 (Message Checker) は，暗号モジュールによって復号された暗号文がもとの平文と齟齬がないか判定する．

アプリケーションインターフェース (Application Interface) としては，大半を PC-UNIX 既存のものを利用するが，ネットワークカードは新たに設計するので，そのデバイスドライバのみ独自に開発する．

ネットワークインターフェース (Network Interface) は，ネットワークとのデータの送受信のみではなく再構成可能素子を用いて暗号化と復号処理を行う暗号モジュール (Crypt module) も含むものとした．

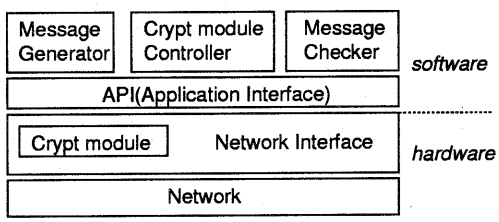

図 4: 各ノードの構成

3.5 鍵管理

共通鍵暗号方式では，通信する相手が増加すると管理しなければならない鍵が大幅に増加する．例えば N 個のノードがそれぞれ全部の相手と異なる鍵で通信する場合，$N(N-1)/2$ 個の鍵が必要となる．当然鍵の生成及び配布も性能に含まれるが，本研究では理想的な状態での性能評価をまず最初に行うため，必要な鍵は事前に生成し，安全な方法で全てのノードに配布されている状態であるという仮定を導入する．

3.6 性能評価方法

提案するシステムを評価するための評価項目は，

1. 暗号の種類とブロック長
2. 通信速度
3. 暗号化及び復号処理時間
4. 再構成可能素子の利用状況
5. 耐タンパ性

等が考えられる．それぞれの項目について以下詳述する．

3.6.1 暗号の種類とブロック長

暗号の種類とブロック長を変化させることにより，ハードウェア上に実装した場合の処理速度や必要なハードウェア量が変化することが予想される．また，同じブロック長の暗号を実装した場合でも内部の構成方法により性能が変化するため，暗号の種類とブロック長に加えて，実装方法の違いがわかるように区分し評価を行う必要がある．

3.6.2 通信速度

暗号処理を行ったときの単位時間当たりの通信データ量を bit/sec で評価する．生成する平文としては，

1. 電子メールの送受信のように比較的小さなデータが不規則に発生する場合
2. ファイル転送のように，大きなデータが連続して発生する場合
3. WWW 閲覧のように大小様々なデータが不規則に発生する場合

のように，平文のデータ量と発生頻度を変化させたパターンを 3 種類作成する．またそのときに暗号区の数と変化させる暗号アルゴリズムの周期を変化させる．暗号アルゴリズムを変化させる周期は，ブロック長毎，ブロック長より長い周期で一定間隔，不定周期の 3 パターンを作成する．

3.6.3 暗号化及び復号処理時間

暗号化及び復号処理は，再構成可能素子上に実装する．実装は，ハードウェア記述言語 (HDL : Hardware Description Language) と論理合成ツールを用いて行う．論理合成時に利用する論理セルと呼ばれる論理単位の配置と配線状況から，最高遅延時間が推測でき，1 ブロック当たりの処理時間が予想できる．

3.6.4 再構成可能素子の利用状況

論理セルの利用数と配置配線の利用数を評価する．

3.6.5 耐タンパ性

本研究の暗号処理は，暗号モジュールとしてネットワークインターフェース上の再構成可能素子に実装する．再構成可能素子は,SRAM 方式の FPGA を採用するため，構成情報の揮発性という特性から電源断によって暗号モジュールの暗号処理機能は消失

する．また，暗号モジュールの構成情報を変更また消去するためには暗号モジュールコントローラを経由する必要があり，オペレーティングシステムの持つ認証等で防ぐことが可能である．

4 考察

本研究では，暗号アルゴリズムを積極的に変化させることに主眼をおいたため，

1. 鍵管理
2. 通信プロトコル

については，制約を設けている．特に鍵管理に関しては，共通鍵暗号方式で通信を行う場合重要な問題であり，鍵管理に関する事項も評価しなければならない．

5 まとめ

本稿では，暗号アルゴリズムを積極的に変えて暗号化と復号を行う通信システムの評価を行うための設計方針とその評価方法を提案した．現在，提案したシステムは 3.4 で示した設計方針に基づき細部の設計を行い作成中である．

今後は，作成したシステムにより性能評価を行う予定である．

参考文献

[1] J.-P.Kaps and C. Paar: "Fast DES implementation for FPGAs and its application to a universal key-search machine", In selected Areas in Cryptography '98, Vol. 1556 of Lecture Notes in Computer Science, Kingston,Ontario,Canada (1998).

[2] I. Goldberg and D. Wagner: "Architectural considerations for cryptanalytic hardware" (1996).

[3] E. Biham: "A fast new DES implementation in software", Fast Software Encryption, pp. 260–272 (1997).

[4] J. Hughes: "Implementation of NBS/DES encryption algorithm in software" (1981).

[5] J. Goubert, F. Hoornaert and Y. Desmedt: "Efficient hardware implementation of the DES" (1985).

[6] A. Elbirt, W. Yip, B. Chetwynd and C. Paar: "An FPGA-based performance evaluation of the AES block cipher candidate algorithm finalists" (2000).

[7] B. Gladman: "Implementation experience with AES candidate algorithms" (1999).

[8] R. Taylor and S. C. Goldstein Eds.: "A High-Performance flexible Architecture for Cryptography", Proceedings of the Workshop in Cryptographic Hardware and Embeded System (CHES) (1999).

[9] A. Elbirt, W. Yip, B. Chetwynd and C. Paar: "An FPGA implementation and performance evaluation of the AES block cipher candidate algorithm finalists" (2000).

[10] M. Riaz and H. Heys: "The FPGA implementation of the RC6 and CAST-256 encryption algorithms" (1999).

[11] Nakajima and Matsui: "Fast software implementations of MISTY1 on alpha processors", TIEICE: IEICE Transactions on Communications/Electronics/Information and Systems (1999).

[12] P. Chodowiec and K. Gaj: "Implementation of the Twofish cipher using FPGA devices" (1999).

[13] Curiger, Bonnenberg, Zimmermann, Felber, Kaeslin and Fichtner: "Vinci: VLSI implementation of the new secret-key block cipher idea", IEEECICC: IEEE Custom Integrated Circuits Conference (1993).

[14] 石井, 大山, 田中：“公開鍵暗号用 2048 ビット剰余系高速演算プロセッサ”, 電子情報通信学会論文誌, **J-82-D-I**, No.4, pp. 571–580 (1999).

[15] 末吉：“リコンフィギャラブルロジック”, 電子情報通信学会誌, **Vol.81**, No.11, pp. pp.1100–1106 (1998).