暗号通信パケットストリームの n-gram 予測による FPGA 動的再構成手法とその評価

丹羽雄平† 前田敦司† 山口喜教†

クライアント・サーバ型の暗号通信において，暗号処理の部分を汎用の FPGA を動的に書き換えて実行するようなリコンフィギャラブルなシステムモデルにおいては，FPGA の利用率を高くすることで性能を向上させることができると考えられる．そのためには，FPGA の無駄な再構成回数を効果的に削減する予測方式の開発が重要である．我々はこのような予測方式として，過去のパケットストリームから近い将来のストリームを予測し，オーバヘッドを減らすシステムを提案した．本研究ではこのような予測方式の一般化を進め，n-gram モデルと呼ぶ汎用的な予測方式を提案し，その有効性およびその性質などについて詳細な評価を行った．その結果，FPGA の再構成オーバヘッドが大きくなると，予測の成功率を示す FPGA 利用率を高くするためには，ブロック構成要素数もまた大きくする必要があることが分かった．また，n-gram 数を大きくするにつれて FPGA 利用率も向上し，暗号通信処理の全体的な性能も向上することが分かった．また，本予測方式を用いることにより，未来がすべて予測可能とした理想的な場合（FPGA 利用率の理論的上限）と比較して，汎用的に用いられている FPGA を使用した場合でも 7 割程度の利用率が達成可能であることが分かった．

An Evaluation of FPGA Dynamic Reconfiguration by n-gram Forecasting the Packet Stream of Encryption Communication

Yuhei Niwa,† Atusi Maeda† and Yoshinori Yamaguchi†

In this paper, we propose the reconfigurable system model which use FPGA to encrypt the data in the server-client encryption communication. In such a system, raising the availability of FPGA improves the performance. Therefore, it is important that the development of the prediction method to reducing useless reconfiguration of FPGA effectively. We propose the method of predicting the encryption algorithm used in the near future requests based on history of requests received so far to improve the efficiency of encryption. We employ the generalized n-gram model for that prediction, and verify its characteristics. In the result, when the overhead of the reconfiguration of FPGA grows, in order to raise the FPGA availability that shows the success rate of the forecast, it is necessary to enlarge the number of block components, too. Moreover, when general purpose FPGA is used and this prediction method is used, it is FPGA availability of about 70% compared with an ideal case predictable in all the futures; this is a theoretical upper bound of the FPGA availability.

1. はじめに

FPGA（Field Programmable Gate Array）は，ハードウェアの高速性とソフトウェアの柔軟性を兼ね備えたデバイスである．FPGA のように，回路を変更できるデバイスを用いることで，計算機で行われる様々な処理のうち，ある特定の計算処理を汎用プロセッサよりも高速に行うシステムを，リコンフィギュラブルシステム（Reconfigurable System）という[1)]．

リコンフィギュラブルシステムはハードウェアである FPGA を用いているため，暗号処理，パターンマッチング，GA，ニューラルネットワークなどの処理を，汎用プロセッサよりも高速に実行できる可能性を持っている．これらの処理を汎用プロセッサで実行した場合，処理の集中するサーバでは，リクエストの到着に対して処理が間に合わず，リクエストを取りこぼすような事態が発生することが考えられる．そこで，より処理の高速な FPGA で処理することでリクエストを取りこぼす回数を低減することができると考えられる．

また，FPGA の動的再構成を行うことで，システムの停止・再起動を行わずに FPGA を必要に応じて別の回路に書き換えることが可能である．

このようなリコンフィギュラブルシステムを暗号通

† 筑波大学大学院システム情報工学研究科
System and Information Engineering, University of Tsukuba

信に用いた場合，通信プロセスごとに異なる暗号処理を要求されることが考えられ，そのたびにFPGAの再構成が発生する可能性がある．このとき，ハードウェア化による高速化以上に再構成のオーバヘッドが大きければ，かえって処理性能が低下することになる．

そこで，過去に使用された暗号方式の履歴から，近い将来に使用される可能性の高い暗号方式を予測し，FPGAをあらかじめその暗号回路に書き換えておくことを考える．こうすることで，ほとんど使われない暗号回路を構成してしまうことを避けて，使用頻度の高い回路のみを効率的に利用できるようになると考えられる．すなわち，FPGA再構成オーバヘッドの影響を抑えることができるようになり，処理性能の低下を回避できると考えられる．

我々は，webサーバへの暗号通信リクエスト（getリクエスト）に対して，FPGAを搭載したリコンフィギャラブルシステムのモデルにおけるFPGA再構成オーバヘッドを低減するための手法として「2-gram表によるリクエスト予測」を提案し，その有効性について検証した[2)]．

本稿では，擬似的に生成したネットワークパケットデータに対して，より一般的なn-gramモデルを用いた予測を行い，FPGAでの処理能力とソフトウェアでの処理能力の比率，FPGA再構成オーバヘッドなど様々なパラメータを変化させ，このモデルがどのような特性を持っているのかを検証する．

2. 暗号通信モデル

本研究では，ある特定のサーバに対し複数のクライアントが個別に暗号通信を要求するようなクライアント・サーバ型の暗号通信を対象とする．サーバでは，クライアントからの要求に対して必要な暗号処理を実行するために暗号処理回路をFPGA上に構成し，そこで暗号処理を行うか，ソフトウェアによる暗号処理を行うかを適宜選択する．

2.1 モデルの構成と動作

本モデルは，ソフトウェア暗号処理部，FPGA暗号処理部，処理決定部，通信処理部の4つのモジュールで構成される（**図1**）．各モジュールの動作について概説する．

2.1.1 通信処理部

実際の通信ログから，その特徴を抽出した擬似的なネットワーク通信データモデルを生成し，そのデータモデルを処理決定部に渡す．特徴の抽出はシステムの動作中に行うのではなく，あらかじめ通信ログの解析を行ってモデル化しておく．特徴の抽出およびデータモデル生成の詳細は2.3節で述べる．

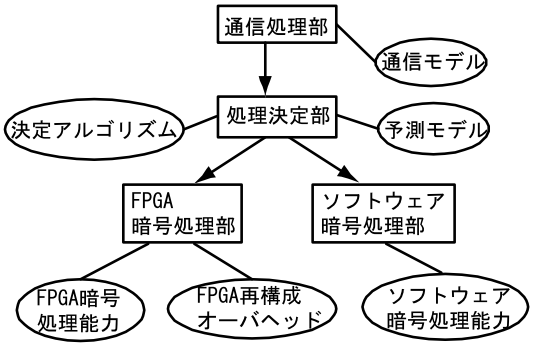


図1 暗号通信モデル
Fig. 1 Encryption communication model.

2.1.2 処理決定部

通信処理部から与えられた擬似的な暗号通信データに対し，このデータをソフトウェアで暗号処理を行うのか，FPGAで暗号処理を行うのかを決定する．この処理決定部では，暗号通信プロトコルの区別はせずに，どのような暗号方式を利用する通信が，いつ，どれだけのデータを処理するために発生したかという情報のみを扱う．これにより，IPsec，SSL，SSHなどの様々な暗号通信プロトコルにとらわれることなく，暗号処理のみを扱うことが可能となる．まず，FPGAで暗号処理を行うと決定するためには，前提条件としてFPGA上に必要な暗号処理回路が載っている必要がある．たとえば，AESの暗号処理をFPGAで実行するためにはAESの暗号処理回路がFPGA上に載っている必要がある．そのため，必要な暗号処理回路がすでにFPGA上に載っていたならば，FPGAで暗号処理を実行すると決定し，処理するデータと必要な暗号方式をFPGA暗号処理部に渡してデータの暗号処理を行わせる．しかし，必要な暗号処理回路がFPGA上に載っていなかったならば，FPGAを必要な暗号回路を搭載するように書き換える必要がある．このとき，FPGAの書き換えには大きなオーバヘッド時間がかかるので，その間にソフトウェアで処理を実行すると決定する．すなわち，必要な暗号処理回路を事前にFPGA上に搭載しておくことができれば，つねにFPGAで暗号処理を実行できるということである．しかしながら，いつどのような暗号方式が必要となるか，事前に知ることは不可能である．そこで我々は過去の情報を基に近い未来に最も高い確率で必要となる暗号方式を予測し，その暗号方式を処理するための暗号処理回路を搭載するようにFPGAを書き換えるようなシステムモデルを提案する．また，このような予測の

表 1 暗号処理モデル性能パラメータ
Table 1 Evaluation parameter of the model.

モデルの各部	ソフトウェア暗号処理部	FPGA 暗号処理部	処理決定部	通信処理部
パラメータ	ソフトウェア暗号処理性能	FPGA 暗号処理性能 FPGA 再構成オーバヘッド	処理決定アルゴリズム 予測モデル	通信データモデル

方式として，n-gram 予測モデルを提案する．詳細については 3 章で述べる．

2.1.3 暗号処理部

処理決定部から渡された処理データと，そのデータを処理する暗号方式の情報を基にデータの暗号処理を行う．FPGA 暗号処理部であれば，FPGA 再構成命令を基に FPGA の再構成を行う．FPGA 暗号処理部の FPGA には，同時に 2 から 4 個の暗号処理回路が搭載できるものとする．各暗号方式の処理能力は，本モデルの性能に大きく影響すると考えられる．

2.2 各モジュールにおける評価用パラメータ

前節にあげた 4 つのモジュールには，性能を左右する様々なパラメータが存在する．本稿において提案する方式の性能を評価するためのパラメータを**表 1** にまとめた．それぞれのパラメータの詳細な解説は 4 章で行う．

2.3 通信データモデル

本研究では暗号通信を行う通信データのモデルを構築するためのサンプルとして，DARPA IDS Benchmark の 1999 年第 1 週月曜日のテストデータ[3)]を利用する．これは，ある特定のネットワークポイントにおける 1 日ごとのパケットキャプチャデータを公開したものであり，およそ 600 万件のパケットキャプチャデータが含まれている．

まず，この中から IP パケットのみを取り出す．次に，source あるいは destination に使用される IP アドレスが “172.168.” で始まるパケットをすべて 1 個のサーバの通信パケットであると仮定する．その結果，およそ 100 万件のデータを持つ擬似サーバ通信ログが得られる．この擬似サーバとの通信パケットの通過時刻，パケットサイズ，source IP，destination IP，使用 port 番号を情報として利用する．また，各クライアントに対してログに出現した順番に ID を割り振る．ID の最大値は 810 となったので，この擬似サーバには 810 種類のクライアントとの通信が存在することが分かる．

さらにこの擬似ログに対し，暗号通信プロトコルの擬似設定を行う．本来の使用通信プロトコル情報の中で，http は SSL で，ssh，telnet，ftp は SSH で暗号化を行うと仮定した．また，これらのプロトコルを 1 度も使用しないクライアントの通信は，すべて IPsec による暗号通信であると仮定した．ただし，SSL と SSH を 1 度でも利用すると仮定したクライアントの中で，この 2 つのプロトコル以外の通信は，通常の，つまり暗号化をいっさい行わない通信であるとした．

表 2 擬似サーバログの詳細（一部）
Table 2 Pseude server log.

リクエスト番号	時刻（秒）	クライアント ID	処理サイズ（byte）
1	27.736780	1	90
2	27.825155	1	90
3	39.628786	2	80
4	39.631363	2	145
5	39.636038	2	60
6	39.636238	2	60
7	39.636784	3	60
8	39.648472	3	90
9	39.648615	3	60
10	39.648822	1	60

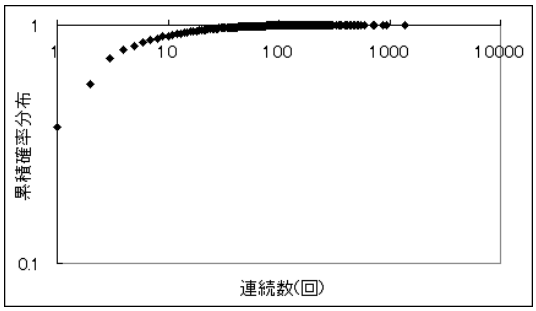


図 2 連続回数の累積確率分布
Fig. 2 Cumulative distribution of continuous frequency.

この結果，この擬似サーバ通信ログには，33,785 件の SSL，602,514 件の SSH，318,225 件の IPsec，84,137 件の何もしないパケットが含まれることとなった．この擬似サーバ通信ログの一部を**表 2** に示す．

次に，このような擬似的なサーバの通信パケットログの分析を行った．

図 2 は同じクライアントからの通信が連続したときの，その連続した回数の確率分布グラフである．横軸が連続した回数，縦軸がその回数の確率分布である．このグラフから，ある程度の回数同じクライアントからの通信が連続して発生することが多いと考えられる．連続回数の平均値は 5.2 であった．

図 3 は，通信パケット到着時刻の時間差の確率分布グラフである．横軸が時間差，縦軸が確率分布であ

る．このグラフから，非常に短い時間差でパケットが到着することが多いということが分かる．

$$F_{(x)}^{-1} = -0.5944/(x - 0.968721) \quad (1)$$

$$F_{(x)}^{-1} = 1 - \log(G_{(x)} - 0.002) \quad (2)$$

$$G_{(x)} = \begin{cases} \sqrt{100x} & (x \leq 0.81) \\ 1.03 - x/0.98 & (x > 0.81) \end{cases}$$

ここで，これらのデータの回帰分析により累積分布関数を求め，さらにその逆関数（式 (1) および式 (2)）を導出してみる．式 (1) の x に一様乱数を与えることによって，同一クライアントパケットの連続到着回数の分布が図 2 とよく似たデータが，また，式 (2) の x に一様乱数を与えることによって，パケットの到着時間差が図 3 のグラフによく似た分布を示す擬似的な通信モデルデータを生成することができる．

このようにして式 (1) および式 (2) に乱数を与えて生成したデータをプロットしたグラフを**図 4** および**図 5** に示す．

本稿ではこうして生成されたモデルデータを基に予測を行う．

上述の通信データモデルの特徴は，あるサーバで発生する通信においては，同じクライアントとの通信パケットは比較的短時間内に，バースト的に連続して到着することが多いということである．一般的に，IPsec や SSL，SSH などの暗号通信プロトコルでは，どの暗号方式を使って通信するかは，サーバとクライアントがお互いの使用可能な暗号方式のリストを交換し，それを優先順位の高いほうから順番に比較していき，最初に見つかった両者が使用可能な暗号方式を使用する，というネゴシエーションによって決定される．そのため，あるクライアントとサーバとの間でどの暗号方式を使用するかが決定されると，この使用可能な暗号方式リストを明示的に変更しない限りは，最初に決まった暗号方式が使用され続けることになる．

したがって，本研究でのモデルにおいては，同一クライアントからの通信が比較的短時間内にバースト的に連続して到着するということを，同じ暗号方式を使用する通信が，比較的短時間内にバースト的に連続して到着することが多いということを仮定して，以下に述べる予測モデルの根拠とする．

ただし，上記の暗号通信データモデルでは，各クライアントがどのような暗号方式を使用するかはランダムに決まるものと仮定するが，一度決定された暗号方式は変更されないものとする．**表 3** に，このように仮定して生成されたパケットのランダム特性を調べるために，暗号通信において使用可能な暗号方式の数を 5

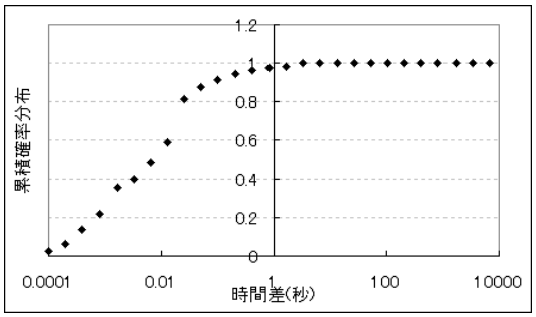


図 3 時間差の累積確率分布
Fig. 3 Cumulative distribution of arrival time difference.

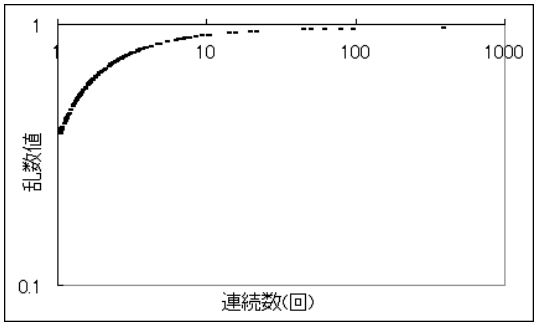


図 4 連続回数の生成結果
Fig. 4 Generation result of continuous frequency.

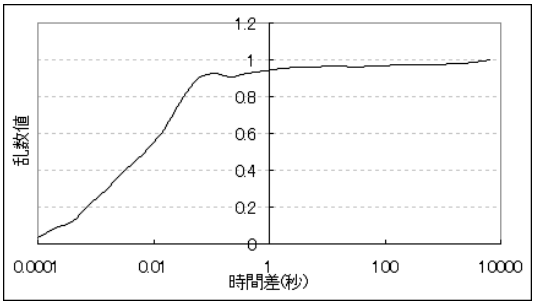


図 5 時間差の生成結果
Fig. 5 Generation result of arrival time difference.

表 3 各暗号方式使用比率
Table 3 Use rate of each encryptions.

暗号方式	A	B	C	D	E	F	G	H	I	J
5 種類	19.4	19.6	20.1	19.3	21.6	x	x	x	x	x
10 種類	9.66	9.73	9.58	9.76	11.2	11.7	7.70	10.8	7.57	12.3

種類と 10 種類の 2 つの場合それぞれについて，各暗号方式の使用頻度を調べた結果を示す．表 3 より，この擬似暗号通信モデルは，すべての暗号方式がほぼ同じ程度の割合で使用されるようなモデルであることが分かる．

3. n-gram 予測

2.3 節で述べたような暗号通信を行うサーバでは，同じ暗号方式を使用する通信が，比較的短時間内にバースト的に連続して到着することが多いという特徴があると考えられる．そのため，ごく近い過去に使用された暗号方式は，ごく近い将来にも引き続き使用される可能性が高いと考えられる．そこで，ごく近い過去に使用された暗号方式を処理するための回路を FPGA 上に構成するようにすれば，暗号処理の高速化が可能であると考えられる．しかし，どこかの時点で必ず使用暗号方式の変化が起こることは明白であるので，その時点を予測して FPGA の再構成を行うことができれば理想的である．

本研究では，このためにサーバが処理しなければならない暗号方式の変化を予測するための手法として，後述する n-gram 予測モデルを提案する．

この 3 章では，n-gram 予測モデルと，それを用いた処理決定部での暗号処理をソフトウェアで実行するか FPGA で実行するかの決定アルゴリズムについて述べる．

3.1 n-gram 予測モデル

n-gram 予測モデルとは，1，2，$\cdots$，$n-2$，$n-1$ 番目までの値の並びを基に，n 番目の値を予測する方式であり，主に自然言語処理における品詞解析に用いられている．n-gram 予測モデルで予測を行うためには，まず過去の全履歴の中に出現するすべての n 個の値の並び方の組合せをそれぞれに数え上げ，その中から 1，2，$\cdots$，$n-1$ までの並び方が同じものの中で最も数の多い，すなわち最も使用頻度の高い並び方を n-gram 表として記録しておく．そして，実際の予測対象データの中に 1，2，$\cdots$，$n-1$ という並び方のデータが出現したときに，この並び方を先ほど記録しておいた n-gram 表の中から探し，一致する項目の n 番目の値を予測結果として採用するというものである．もしこのような 1，2，$\cdots$，$n-1$ の並び方が n-gram 表の中に存在しなかった場合には，過去の全履歴のうち最も出現頻度の高い値を予測結果（0 次予測）とする．

このような n-gram モデルを，$n=3$ のときを例にとって具体的に説明する．たとえば，$n=3$ のとき，値として a，b，c の 3 種類をとる “ababbabcabcababaa” という履歴情報が存在したとする．この履歴の中のすべての $n=3$ 個の値の並び方は，

aba bab abb bba bab abc bca cab abc bca cab aba bab aba baa

の 15 種類となる．これを $n-1$ 番目までの文字，すなわち 2 文字目までの並び方が同じものをまとめると

aba abb abc abc aba aba	aba が最多（3/6 回）
bab bab bab baa	bab が最多（3/3 回）
bba	bba が最多（1/1 回）
bca bca	bca が最多（2/2 回）
cab cab	cab が最多（2/2 回）

となる．よって，「aba bab bba bca cab」という値列の集まりを 3-gram 表として記録しておく．

ここで，予測対象データとして “ab” という値の並びが来たとすると，“ab” で始まる 3-gram 表の値列は “aba” なので，その 3 番目の値である “a” というのが予測結果となる．

このような n-gram モデルを用いるためには，予測に用いる履歴情報から n-gram 表を作成しておく必要がある．次節では，この n-gram 表の作成方法の詳細について述べる．

3.2 n-gram 表の作成

n-gram 表を作成するためには，予測に用いる履歴情報をすべて値列（文字列）に変換し，その中に含まれるすべての長さ n の部分文字列を把握し，その出現頻度を数え上げて，最も出現頻度の多いものを調べる必要がある．そのための方法として，suffix-array を用いる．

suffix とは文字列データのある位置から始まり，その文字列データの末尾までの範囲の部分文字列のことである．たとえば「さいこうせい」という文字列を例にとると，「さいこうせい」「いこうせい」「こうせい」「うせい」「せい」「い」という 6 つの suffix が得られる．このすべての suffix の集合を辞書順に整列したものが suffix-array である．

suffix-array に含まれる各文字列の先頭 n 文字を抜き出すことで，n-gram 表を作成するために必要なすべての長さ n の部分文字列を得ることができる．こうして得られた長さ n の部分文字列集合を，長さ $n-1$ までが同じものごとにまとめ（suffix-array を作成した時点で辞書順に整列している），最も出現頻度の多いものを調べ，表に登録していくと，n-gram 表が完成する．

本研究では，予測に用いる履歴情報として，2.3 節で述べた擬似サーバ通信ログを用いる．この中のある個数連続したパケットデータを 1 ブロックとし，この

ブロック内に含まれるパケット（ブロック構成要素と呼ぶ）が必要とする暗号処理方式のうち，使用頻度の高いもの i 種類（FPGA 上に同時に搭載可能な暗号処理回路の数 i）の組合せをブロックの値とする．この値を 1 文字と考えて，擬似サーバログのすべてのブロックを値列（文字列）に変換していく．こうして作られた文字列に対して上記の suffix-array を構築し，すべての部分文字列を調べ，最も出現頻度の高いものを表に登録していく．

本稿では，FPGA には同時に 2 から 4 の暗号処理回路が搭載できるとしており，使用できる暗号は 4 章で後述するが，5 種類または 10 種類としている．そのため，1 つのブロックがとりうる値は，C(5,2) = 10，C(5,3) = 10，C(5,4) = 5，C(10,2) = 45，C(10,3) = 120，C(10,4) = 210 となり，最小 5 通りから最大 210 通りのいずれかとなる．この組み合わせ方それぞれに対してユニークな値を割り振る．したがって，同じ値を持つブロックは，ブロック内で使用される暗号処理回路の組合せが同じものとなる．

具体的に，表 2 に示した擬似サーバログに対して，FPGA 上に搭載可能な暗号回路の数を 2 とし，使用可能な暗号方式は 5 種類，クライアント ID がそのまま使用暗号方式を示すものとする．1 ブロックは 2 件のパケットデータから構成されているという条件で，2-gram 表を構築してみる．

まず，表 2 のログデータをブロックごとに分割し，各ブロックで使用頻度の高い暗号方式 2 種類を調べると**表 4** のようになる．ただし，使用頻度の高い暗号方式が 1 種類しか存在しない場合，たとえばブロック番号 1 や 4 などの場合には，全体を通して使用頻度の高い暗号方式を追加する．このような暗号方式については () で括っておく．また，こうしてブロックごとに使用頻度の高い暗号方式が判明したら，その組み合わせ方に対してユニークな ID を割り振っていく．表 4 の例では出現順にシーケンシャルな番号としておく．この ID を 1 文字と考えて，ID の並び（文字列）を作成すると

「111231122」

となる．この文字列（ブロック使用暗号 ID 列）の suffix は

「111231122」「11231122」「1231122」「231122」
「31122」「1122」「122」「22」「2」

であり，これらを辞書順に整列することで suffix-array が得られる．この各 suffix の先頭 n = 2 文字を抜き出していくと得られる 2-gram 集合は

「11」「11」「12」「23」「31」「11」「12」「22」

表 4 ブロック構成法の一例
Table 4 An example of how to make block.

ブロック番号	クライアント ID	ブロック内で使用頻度の高い暗号方式	暗号方式の組み合わせ方 ID
1	1,1	1,(2)	1
2	2,2	2,(1)	1
3	2,2	2,(1)	1
4	3,3	3,(2)	2
5	3,1	1,3	3
6	1,2	1,2	1
7	2,2	2,(1)	1
8	2,3	2,3	2
9	3,3	3,(2)	2

となる．

ここから，0 文字目から n − 1 = 1 文字目までが同じもの，この例では ‘1’ で始まるものと ‘2’ で始まるもの，‘3’ で始まるものの 3 種類，の中で，n = 2 文字目の出現頻度が最も多いもの，この例では「11」と「22」と「31」，が 2-gram 表に登録されるエントリ，ということになる．このとき，「22」と「23」は出現頻度がまったく同じあるが，このような場合には数字の小さいほうをエントリとする．

このような手順を過去の履歴情報，たとえば 2.3 節で述べた擬似サーバログ全体に対して行うことで，任意の n-gram 表を構築することができる．

前述の具体例の 2-gram 表を構築した時点で，予測処理を行う手順について概説する．

たとえば，ここで暗号通信処理が行われているある時点を考える．このとき，現在の処理の直前 1 ブロックで使用された暗号方式の組み合わせ方の ID が ‘1’ であった，すなわち暗号方式「1」と「2」の使用頻度が高かった，とすると，2-gram 表に登録されている項目の中から，n − 1 = 1 文字目が ‘1’ であるエントリを検索する．このとき 2-gram 表には「11」と「23」と「31」というエントリが存在するので，「11」という結果が得られる．このエントリの n = 2 文字目である ‘1’ というのが，現在の直前 1 ブロックの直後のブロック，すなわち未来のブロックの使用暗号方式の組み合わせ方の ID として予測されたということになる．この ‘1’ という ID は暗号方式「1」と「2」という組み合わせ方を意味しているので，直後の未来のブロックでは，暗号方式「1」と「2」が使用される可能性が高い，と予測されたことになる．つまり，ブロック構成要素数は 2 であるので，次のリクエストと次の次のリクエスト，合計 2 つのリクエストは暗号方式「1」と「2」を使用する可能性が高い，と予測することができたわけである．この予測結果に基づいて FPGA

```
擬似サーバログから n-gram 表を作成する
while(暗号通信処理が発生するたびに)
    if(FPGA==再構成中)
        software で暗号処理
    else
        if(この処理を FPGA で実行可能)
            FPGA で暗号処理
        else
            if（前回の予測からブロック構成要素数分処理が進んだ）・・・(*
                直前の連続する n-1 ブロック分の処理を検査し使用暗号方式 ID 列を算出
                算出した ID 列をキーとして n-gram 表を参照
                参照結果を予測結果として FPGA を再構成
            end if
            software で暗号処理
        end if
    end if
end while
```

図 6 処理決定アルゴリズム
Fig. 6 Process determination algorithm.

を「1」の暗号回路と「2」の暗号回路を搭載するように書き換えることで，次と次の次，2 件のリクエストを FPGA で処理できるようになる可能性が高くなり，FPGA の利用率を向上させることが可能になると考えられる．

このようにして，過去の履歴情報から n-gram 表を作成しておき，その表から必要なエントリを検索することで n-gram 予測が可能となる．

上記の例は 2-gram 表の場合であるため，2-gram 表を参照するときのキーとなるのは直前 1 ブロック分の使用暗号 ID のみであるが，一般的に n-gram の場合には，連続する直前 $n-1$ ブロック分の使用暗号 ID がキーとして用いられる．

ここで，n-gram 表を構築するためのデータ構造について考えてみる．

上記のように，1 ブロックがとりうる値は最大で 210 通り存在する．n-gram 表には長さ n の文字列を登録していくので，n-gram 表に登録される項目の数は，最悪の場合 $210\hat{n}$ 個となる．$n = 4$ などを考えてみると，4-gram 表に登録される項目数は 1.9 G 個以上になる．予測処理を行うときには，これだけの数のデータを検索する必要がある．このとき，n-gram 表が単純に線形リストなどで構築されていると検索に必要な時間だけで大きなオーバヘッドとなってしまうことが容易に推測される．現実にこのような状況になることは考えにくいが，表検索の効率を考えて，n-gram 表は B 木で構築することとした．

3.3 n-gram モデルによる予測と処理決定のアルゴリズム

本節では，暗号通信モデルの処理決定部で行われる，n-gram 表を用いた予測処理と，詳細な処理の決定アルゴリズムについて述べる．

まず，n-gram の n の値をいくつにするのか，1 ブロックを構成するブロック構成要素数はいくつにするのか，使用可能な暗号方式と FPGA 上に同時に搭載できる暗号回路の数はいくつか，などの情報を決定しておく．そして，前処理として，3.2 節で述べた方法を用いて，2.3 節で述べた擬似サーバログから n-gram 表を作成しておく．この表を用いて予測を行う．ただし，現在のところ，1 度作成した n-gram 表の更新は行わないこととしてある．

図 6 に，暗号通信処理のリクエストが 1 件発生するごとに行われる処理手順を示す．FPGA を再構成中であった場合，または予測が外れるなどして今来たリクエストとは異なる暗号回路が載っていた場合には，ソフトウェアで暗号化処理を行うという決定がなされる．

また，現在のリクエストを処理可能な暗号回路が FPGA 上に搭載されていた場合には FPGA で処理を行うと決定する．

ここでは，図 6 の*印の条件分岐について述べる．本予測モデルは，3.2 節で述べたように直後の 1 ブロック分の処理，すなわちブロック構成要素数分の処理の中で使用頻度の高い暗号方式を予測するものである．そのため，予測結果として得られた暗号回路に FPGA を再構成するときに，この再構成を行っている間に予測結果となった 1 ブロック分 = ブロック構成要素数分の処理をソフトウェアで実行完了してしまっては，予測する意味がなくなってしまう．逆に，この予測結果となった 1 ブロック分 = ブロック構成要素数分の処理が終了する前に新たな予測を行ってしまうと，先

の予測結果をほぼ利用しないことになってしまう．そのため，図 6 の*印に示したような条件分岐を行っている．

また，予測を行うときに得られた情報が，作成しておいた n-gram 表に現れないパターンであった場合，履歴情報全体を通して見た中で，最も使用頻度の高かった暗号方式を予測結果として用いる．これを 0 次予測と呼ぶ．

4. シミュレーションによる評価方法

本提案手法の有効性を評価するために，パラメータ値を様々に変化させてシミュレーションを行った．変化させるパラメータ値は以下のとおりである．

- FPGA 再構成オーバヘッド
- FPGA とソフトウェアの暗号処理能力の比率
- 1 ブロックを構成するパケットの数（ブロック構成要素数）
- n-gram 表を作成する値 n-gram 数
- FPGA に同時に搭載できる暗号処理回路の数
- 使用可能暗号数

なかでも FPGA 再構成オーバヘッドは重要なパラメータであると考えられるので，様々な角度から考察する．

4.1 シミュレーションの概要

まず，入力として 2.3 節で述べた擬似的な通信モデルに基づく擬似サーバデータ（およそ 100 万件のパケットヘッダデータを含む）を生成し，これを利用して 3.2 節で述べた方法で n-gram 表を作成し，予測処理を行う．

予測対象データとして，DARPA のデータにおいて次の日にあたる火曜日半日分 60 万件に基づいたデータを使用する．このパケットデータを 1 件ずつシステムに入力し，図 6 に示したアルゴリズムを用いてシミュレーションを行い，性能を評価する．すなわち，月曜日のデータをモデル化し，そこから生成したモデルデータから n-gram 表を作成する．この n-gram 表を用いて，図 6 のアルゴリズムに基づいて火曜日のデータを予測し，提案モデルの性能を評価することになる．ただし，1999 年当時からのネットワーク速度の向上を考慮に入れ，この予測対象データのパケットの到着時間差を 0.01 倍して用いることとする．その結果，通信モデルの負荷は 1,371 packets/s，411 bytes/packet であるので，563.5 kbyte/s = 4.5 Mbps となった．ただし，これは全リクエストを通してみたときの平均値であり，パケットの到着間隔には偏りがあるため，瞬間的に大きな負荷がかかることがある．

また，各クライアントとの通信で使用される暗号方式は，クライアントごとに使用可能な暗号方式の中から 1 つランダムに決定し，この決定された暗号方式で固定されるものとする．

性能検証に用いる評価値として，全処理件数のうち，どれぐらいの割合の処理を FPGA で実行することができたのかを示す，「FPGA 利用率」を測定するためのシミュレーション実験を行った．シミュレーションであるため，実際にデータの暗号化を行うわけではないが，処理データサイズを基に処理にかかった時間などを計算し，パケットデータ 1 件ごとに，FPGA で処理ができたかどうかを判断する．

本システムモデルにおいて，予測が成功したということは，実際に使用される暗号方式に FPGA を書き換えることができた，ということを意味する．このように FPGA の書き換えができたならば，その FPGA 上の回路を使って暗号処理ができる，ということなので，FPGA 利用率という数値は予測の成功率を示していることになる．

また，性能比較対象として，oracle モデルを用いた FPGA 利用率の測定も合わせて行う．本実験では予測対象データである「火曜日のデータ」はあらかじめすべて用意されているが，実際の実験に際しては前述のようにパケットデータを 1 件ずつ順番に入力するため，予測システムは「過去」のデータしか知ることはできない．そのため，過去のデータを基に作成した n-gram 表を用いて未来のデータの予測を行う．

これに対し，oracle モデルは予測対象データである「火曜日のデータ」をすべて先読みしており，FPGA をどんな回路に，どのタイミングで再構成すれば暗号処理を FPGA で実行し FPGA 利用率が最も高くできるのか，動的計画法によりあらかじめ決定しておくモデルである．現実には未来の通信データの先読みは不可能であるが，理論上ほぼ最高の FPGA 利用率を示すモデルとして，性能比較のために用いる．

また，実質的なパケット暗号処理の性能を評価するために，「パケット到着時刻」から「その暗号処理が終了した時刻」までの処理レイテンシの測定を行う．この処理レイテンシには処理の待ち時間，暗号化に要した時間，FPGA の再構成に要した時間が含まれている．評価にあたってはこの処理レイテンシの値を，処理したデータサイズで割り，データ 1 bit あたりの数値に正規化して用いる．

さらに，予測という処理そのものに必要なオーバヘッドの測定も行う．この値は予測処理開始時と終了時の CPU クロックサイクル数の差から算出する．

4.2 評価用パラメータの意味と設定

本節ではシミュレーションにおけるパラメータの設定方法について簡単に説明し，実験に際しての基準値を設定する．5 章に述べるシミュレーション実験では，この基準値を中心に値を変化させて検証を行う．本稿ではこれらの値を変化させて評価，検証を行った．

4.2.1 FPGA 再構成オーバヘッド

回路量の大きな暗号処理回路全体を動的に再構成することを考慮に入れると，実装可能な回路規模の大きい汎用 FPGA（例：Xilinx 社製 xc2v8000 デバイス）を用いることが考えられる．このデバイスのコンフィギュレーションは，コンフィギュレーションクロックと呼ばれる専用クロックに同期して，専用バスを通じて 8 bit ずつ回路構成データが回路構成情報メモリからデバイスに読み込まれることで完了する[4]．xc2v8000 の回路構成データのサイズは 29,063,072 bit であり，コンフィギュレーションクロックは最大 66 MHz での動作が可能である．このことより，xc2v8000 デバイスのコンフィギュレーションには，約 0.06 s の時間が必要である．

この再構成オーバヘッドは小さいほど，モデルの性能が向上するものと考えられる．実験ではこの時間を基準として様々な値に変化させて測定を行う．

4.2.2 暗号処理能力比

ソフトウェアで処理した場合と FPGA で処理した場合との暗号処理の性能比をパラメータとしてシミュレーションを行うが，そのための予備データとして現実のソフトウェアと FPGA での暗号処理性能を以下のようにして求めた．

AES 選考[5),6)]の対象となった暗号と CRYPTREC[7)] にあげられている暗号，さらに GNU の暗号ライブラリである libgcrypt[8)] に含まれる暗号などの中から
DES-CBC，3DES-CBC，AES-CBC，
SERPENT-CBC，TWOFISH-CBC
の 5 種類の性能測定を行った結果を**表 5** に示す．

ソフトウェアで実行した場合の各暗号処理性能は，GNU の暗号ライブラリである libgcrypt[8)] のベンチマーク実行結果の数値を基に算出した．ベンチマークは，Pentium4-2.2 GHz，Windows 2000 SP4，512 MB の計算機上で，C 言語で書かれたプログラムを GCC-O3 でコンパイルして実行した．各暗号方式の性能を表 5 に示す．

3DES-CBC，DES-CBC の FPGA 処理性能については，OPENCORES[9)] より提供されているフリーの暗号処理 IP コアの論理合成・配置配線を行い，その性能を測定した．論理合成配置・配線に用いた論理合成ツールは Xilinx 社の ISE6.2i，ターゲットデバイスは Vertex2-xc2v8000，スピードグレードは −4 とした．

表 5 暗号処理能力
Table 5 Throughput of each encryption.

暗号方式	Software (Mbps)	FPGA (Mbps)	Slice 使用率 (%)
DES-CBC	104.6	530.0	9
AES-CBC	170.2	1160.0	34
3DES-CBC	64.0	170.2	27
SERPENT-CBC	39.4	270.0	19
TWOFISH-CBC	83.1	120.0	20

AES の処理性能については，片下らの研究[10)] に示される結果を採用することとした．SERPENT および TWOFISH については AES 選考の報告[11),12)] において，xcv1000 デバイスでの評価結果より，SERPENT が 1block128 bit を 1clocks で 38.0 MHz，TWOFISH が 1block128 bit を 2clocks で 24.8 MHz との報告を参考に，それぞれ CBC モードで動作させた場合の性能を算出した．

また，AES-CBC および DES-CBC，3DES-CBC の slice 使用率については，片下らの研究[10)] でループアーキテクチャでの slice 使用率報告を参考に，パイプラインアーキテクチャで実装した場合の slice 使用率を推定し，その数値を表 5 に示す．パイプラインアーキテクチャでの実装回路規模を対象とした理由は，6 章の今後の展望で述べるマルチスレッド実行を考慮に入れたためである．SERPENT および TWOFISH の slice 使用率については，AES 選考の報告[11),12)] において，xc2v1000 デバイスでのパイプラインアーキテクチャ実装報告 slice 数を基に，その値を表 5 に示している．表 5 に示した slice 使用率は，xc2v8000 デバイスの全 slice 数に対する使用 slice 数の割合を表す．

表 5 の結果より，ソフトウェアでの処理よりも FPGA での処理のほうが平均で約 4.6 倍高速であるという結果となった．これらの結果から，5 章に述べる実験では，ソフトウェアでの各暗号処理性能の平均値を 1 としたときの，FPGA 暗号処理回路の処理性能を 1.5 倍，5.0 倍，10.0 倍と変化させて実験を行う．ここで処理性能とはパケット処理のスループット（Mbps）のことである．

また，システム全体の暗号処理性能と混同しないように，ソフトウェア，FPGA それぞれの暗号処理性能のことは，以降暗号処理能力と呼ぶこととする．よって，本項のパラメータは暗号処理能力比と呼ぶ．

4.2.3 ブロック構成要素数

1 ブロックを構成するデータの数（要素数）を変化させて性能を評価する．この数値の最小値は 1，最大

値は n-gram 表を作成するときに用いる履歴に含まれる全データ件数となる．この最大値をとった場合には，3.3 節で述べた 0 次予測を行った場合と同等の意味となる．この値が小さすぎれば，局所的な通信クライアントの変化，すなわち使用暗号方式の変動に予測結果が左右されてしまい正確な予測が困難になり，逆にこの値が大きすぎれば粗い予測となってしまい同じく正確な予測が困難になり，性能は向上しなくなると考えられる．我々の過去の実験ではこの値は 200 に固定していたが，本稿では様々な値に変化させて実験を行う．

また，oracle モデルでは n-gram 予測を行わないので，このパラメータ自体が存在しない．

4.2.4 n-gram 表を作成する n-gram 数

実験では n-gram 数は 2 から 7 までについて性能を検証する．

n-gram 数が 2 である，ということは，現在の直前 1 ブロックのデータを基に n-gram 表を参照して直後の 1 ブロックを予測する，ということであり，値が 3 であるということは直前の 2 ブロックのデータを使用して n-gram 表を参照するということになる．このように，n-gram 数を 1 増やすと，n-gram 表を参照して予測に用いるための連続するデータブロックの数が 1 つずつ増える，ということである．

一般的に考えると n-gram 数は暗号通信における使用暗号方式の変動をキャッチするためのパラメータであり，4.2.3 項に示したブロック構成要素数は，この変動の局所性を抑えるためのパラメータである．

よって，FPGA 再構成オーバヘッドに加えて，この n-gram 数と，ブロック構成要素数のバランスをとることが，本システムモデルの性能に大きな影響を与えるものと考えられる．

4.2.5 FPGA 上に同時に搭載できる暗号回路の数と使用可能暗号数

本実験では，FPGA には暗号処理回路を同時に 2 つ搭載できる場合，3 つ搭載できる場合，4 つ搭載できる場合，それぞれについてシミュレーションを行った．また，使用できる暗号方式の数は 5 種類または 10 種類とした．そのため，3 章で述べたとおり，1 ブロックがとりうる値は 5 通りから 210 通り存在する．これは，使用可能な暗号方式の数から FPGA 上に同時に搭載できる数を選び出す組合せの数である．

また，表 5 の実装回路規模の値をみると，パイプラインアーキテクチャの暗号回路を使用した場合，同時に FPGA 上に搭載できる暗号回路の数は 4 つまでということが分かる．

5. シミュレーション実験結果と考察

本章では，以上に述べたシミュレーションの方式やパラメータの基づいてシミュレーション実験を行った結果と，それに対する考察に関して述べる．

5.1 FPGA 利用率

まず，FPGA 利用率についての測定を行った．FPGA 利用率というのは，全処理件数の内 FPGA で実行することができた処理の割合のことである．

5.1.1 FPGA 利用率の比較

図 7 に，oracle モデルを用いた場合の FPGA 利用率の測定結果を示す．縦軸が FPGA 利用率，横軸が FPGA 再構成オーバヘッド，凡例は暗号処理能力比である．このとき，使用可能暗号数 10，FPGA 搭載可能回路数 3 である．

図 7 の結果から，oracle モデルを用いた場合の FPGA 利用率は FPGA 再構成オーバヘッドや暗号処理能力の影響をほとんど受けず，おおよそ 0.9 ぐらいであるということが分かる．

本 oracle モデルは動的計画法を用いており，予測対象データをあらかじめすべて先読みし，FPGA をいつ，どのような暗号回路に再構成すると FPGA 利用率が最も高くなるかを，1 パケットごとに計算している．このように，oracle モデルは未来の情報をすべて利用できるという現実的でない仮定のもとに成り立つモデルである点で，n-gram 予測モデルよりも有利であると考えられる．また，未来の情報を利用できるという点で，パケット単位ブロック単位を問わず，n-gram モデルも含めて予測（過去の情報からのみ未来を予測する）を用いるすべての手法の性能の上界を与えるモデルであると考えられる．

また，oracle モデルでは，全体の約 1 割がソフトウェアで実行されるようにスケジューリングされることが分かる．過去の履歴から予測するという予測モデ

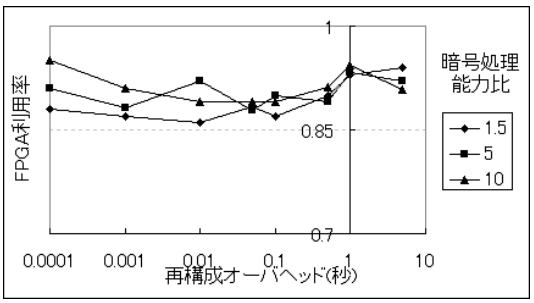


図 7 oracle モデルによる FPGA 利用率の上界
Fig. 7 Upper bound of FPGA availability by oracle model.

ルを採用する限りは，これを上回る FPGA 利用率は達成できないものと考えられる．

この oracle モデルの FPGA 利用率が，再構成オーバヘッドの影響を受けない理由としては以下のように考えられる．2.3 節でも述べたが，本暗号通信モデルでは同一クライアントからの処理は短時間内に連続する，という特徴がある（表 2 のリクエスト番号 3 から 6）．すなわち，きわめて短時間で到着するリクエストは同じ暗号方式を利用し続ける可能性が高い，ということである．ということは，FPGA 再構成オーバヘッドを上回るほど長時間のリクエスト到着時間差が発生した場合（表 2 のリクエスト番号 2 の直後），その次に同じぐらいの時間差が発生するまでの間（表 2 のリクエスト番号 3 から 9）は，同じ暗号方式を利用するリクエストで占められる可能性が高い，ということを意味する．このときに使用率の高い暗号回路をピンポイントで FPGA 上に搭載しておけば，高い利用率を達成できると考えられる．

oracle モデルでは未来のリクエストを先読みすることができるので，上記のようなリクエストの到着時間差が FPGA 再構成オーバヘッド以上となるタイミングを狙って FPGA を再構成し，次のリクエストが到着するまでに FPGA を最適なものへと書き換えておくことが可能となる．こうすることで実質的に FPGA 再構成オーバヘッドが存在しないかのように見せかけることができるので，図 7 のように，FPGA 再構成オーバヘッドに影響されない FPGA 利用率を示すものと考えられる．

次に，n-gram 予測モデルにおける FPGA 利用率の変化の様子を**図 8** に示す．図 8 は，使用可能暗号数は 10，FPGA 搭載可能回路数 3，処理能力比 5.0，4-gram 予測を行ったものを示している．縦軸が FPGA 利用率，横軸がブロック構成要素数，凡例が再構成オーバヘッド（秒）である．

図 8 より，FPGA 再構成オーバヘッドが大きくなるにつれて FPGA 利用率は下がることが分かる．これは，本システムモデルが FPGA 再構成中は処理をソフトウェアで実行するモデルであり，そのため FPGA で処理できた割合を表す FPGA 利用率が下がる結果となったと考えられる．また，FPGA 利用率の最大値は 0.8 程度と，oracle モデルにかなり近い値を達成しており，本予測モデルの有効性を示しているといえる．しかしながら，それは再構成オーバヘッドが非常に小さい場合の時であり，FPGA 再構成オーバヘッドが大きくなるにつれて，FPGA 利用率は下がっていく．本システムのモデルにおいては，FPGA 再構成

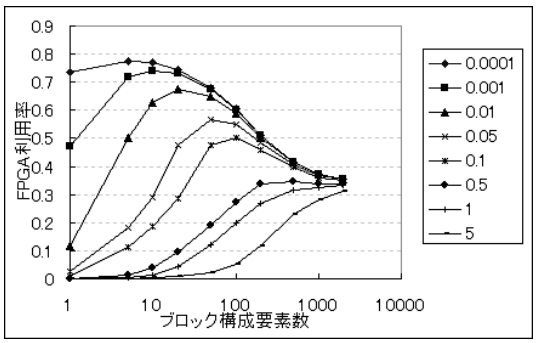


図 8 FPGA 利用率の一例
Fig. 8 An example of FPGA availability.

中はソフトウェアで暗号処理を実行する方式であるため，FPGA 再構成オーバヘッドが大きくなると，必然的に FPGA 利用率が下がるためと考えられる．ただし，現実の FPGA における再構成オーバヘッドの値に近い 0.05 では，最大で 0.6 に近い FPGA 利用率が達成できることが分かる．

図 8 が示すように，n-gram 予測モデルでの FPGA 利用率は oracle モデルと異なり，FPGA 再構成オーバヘッドの影響を大きく受ける．これは，n-gram 予測モデルでは，リクエストの到着順序情報のみを利用しており，oracle モデルのように到着時刻，到着時間差などの情報を予測に用いることができていないためであると考えられる．また，再構成オーバヘッドを固定した場合には，ブロック構成要素数によって FPGA 利用率は変化し，最適なブロック構成要素数があることが分かる．これについては次項で詳しく述べる．

5.1.2 最適 FPGA 利用率を示すブロック構成要素数

表 6 は，FPGA 再構成オーバヘッド 0.01 秒で暗号処理能力比 1.5 のときの，組合せ数ごとに最大 FPGA 利用率を示すブロック構成要素数を n-gram 数ごとに示したものである．各行が組合せ数ごとの値を，各列が n-gram ごとの値を表している．表 6 全体のメジアンをもって，FPGA 再構成オーバヘッド 0.01 秒のときに最も良い FPGA 利用率を示すブロック構成要素数の代表値とした．このような値をすべての FPGA 再構成オーバヘッドごとに測定し，同様のデータを暗号処理能力比 5.0，10.0 の場合にも測定し，グラフにプロットしたものが**図 9** である．こうすることで再構成オーバヘッドの変化による，最も良い FPGA 利用率を示すブロック構成要素数の値の変化の傾向を測ることができる．

また，図 9 は，縦軸がブロック構成要素数，横軸が再構成オーバヘッドであり，凡例は暗号処理能力比を

表 6 オーバヘッド 0.01 s のときに最大の FPGA 利用率を示すブロック構成要素数
Table 6 Number of block components in which the maximum FPGA availability is shown at overhead 0.01 s.

	2-gram	3-gram	4-gram	5-gram	6-gram	7-gram
C(5,2)	200	100	100	100	100	10
C(5,3)	50	50	50	50	50	50
C(5,4)	50	20	50	50	50	50
C(10,2)	10	5	100	50	50	20
C(10,3)	20	20	20	20	20	20
C(10,4)	1000	1000	500	2000	100	100

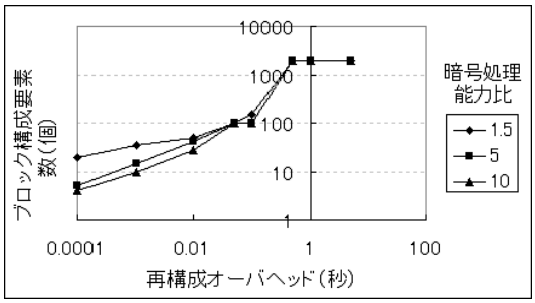


図 9 最も良い FPGA 利用率を示すブロック構成要素数（メジアン）
Fig. 9 Block components median which show the best FPGA availability.

示す．

図 9 より，FPGA 再構成オーバヘッドが大きくなると，最も良い FPGA 利用率を示すブロック構成要素数の値も大きくなることが分かる．この理由は，以下のように考えることができる．

予測を実行した時点では，その予測時点の直後のブロック構成要素数分の処理で使用される暗号方式を予測する．この予測結果に基づいて FPGA の再構成を行うわけであるが，当然その間 FPGA は再構成中なので使用不可能である．このとき，もしもこの再構成中に予測結果を適用できるブロック構成要素数分の処理がソフトウェアで実行完了してしまったら，FPGA の再構成が終わった頃には予測結果が役に立たないものになってしまい，FPGA 上にはまったく脈絡のない暗号回路が搭載されている，という状況になることが考えられる．そうなると FPGA 利用率が上がらない，という結果になると推測される．そのため，再構成オーバヘッドが大きくなるにつれて，高い FPGA 利用率を示すことができるブロック構成要素数も大きくなるという結果が得られたものと考えられる．これは，FPGA の再構成が終わった時点で，予測結果を適用できるブロック構成要素が残っていないと，意味のない予測をしたことになる，といい換えることもできる．

ここで，暗号処理リクエスト発生時間差について考える．FPGA 再構成オーバヘッド時間中にソフトウェアで実行される処理の件数は，当然処理するデータの大きさやその暗号の種類と暗号処理能力などに影響を受けるが，同時に図 3 に示した暗号処理リクエストの発生時間差の影響を最も大きく受ける．

なぜなら，時間差が小さければ FPGA 再構成オーバヘッド中に発生する処理の件数は多くなり，逆に時間差が大きければこの件数は小さくなるためである．

今後ネットワーク速度はさらに速くなることが予想されるので，処理の発生時間差はさらに小さくなっていくと考えられる．そうなると当然 FPGA 再構成オーバヘッド中にソフトウェアで実行されなければならない処理の件数は多くなり，予測を行う際のブロック構成要素数も大きくとることが必要である．

5.1.3 n-gram 数と FPGA 利用率

表 7 は，暗号処理能力比 1.5 のときの，組合せ数ごとの最大 FPGA 利用率を，n-gram 数ごとに示したものである．各行が組合せ数ごとの値を，各列が n-gram 数ごとの値を表している．表 7 全体のメジアンをもって，2-gram 予測を行ったときの FPGA 利用率の代表値とした．このような値をすべての n-gram 数ごとに測定し，同様のデータを暗号処理能力比 5.0, 10.0 の場合にも測定し，グラフにプロットしたものが **図 10** になる．こうすることで，n-gram 数の変化による FPGA 利用率の変化の傾向を測ることができる．

図 10 の縦軸が FPGA 利用率を示し，横軸が n-gram 数，凡例が暗号処理能力比を表している．図 10 より，n-gram 数が増加するごとに，FPGA 利用率もまた増加していくことが分かる．**表 8** は，暗号処理能力比 1.5，FPGA 再構成オーバヘッド 0.05 s としたときの，組合せ数ごとの最大 FPGA 利用率を，n-gram 数ごとに示したものである．各行が組合せ数ごとの値を，各列が n-gram 数ごとの値を表している．表 8 の n-gram 数ごとにメジアンを算出し，同様のデータを暗号処理能力比 5.0，10.0 の場合にも測定し，グラフにプロットしたものが **図 11** になる．よって図 11 は，

表 7 2gram 予測での各組合せ数における最大 FPGA 利用率表
Table 7 Maximum FPGA availability table in number of each combination in 2-gram predict.

	0.0001	0.001	0.01	0.05	0.1	0.5	1.0	5.0
C(5,2)	0.536	0.534	0.531	0.517	0.510	0.477	0.465	0.436
C(5,3)	0.680	0.672	0.639	0.572	0.530	0.419	0.399	0.379
C(5,4)	0.621	0.615	0.603	0.601	0.597	0.578	0.564	0.527
C(10,2)	0.586	0.459	0.413	0.351	0.331	0.296	0.282	0.251
C(10,3)	0.675	0.665	0.621	0.531	0.470	0.329	0.312	0.293
C(10,4)	0.685	0.666	0.631	0.572	0.530	0.456	0.450	0.428

表 8 再構成オーバヘッド 0.05 秒での最大 FPGA 利用率表
Table 8 Maximum FPGA availability of each combination at reconfiguration overhead 0.05 s.

	2-gram	3-gram	4-gram	5-gram	6-gram	7-gram
C(5,2)	0.517	0.518	0.523	0.528	0.541	0.551
C(5,3)	0.572	0.300	0.613	0.621	0.624	0.625
C(5,4)	0.601	0.718	0.718	0.719	0.720	0.725
C(10,2)	0.351	0.372	0.460	0.505	0.520	0.523
C(10,3)	0.531	0.538	0.542	0.543	0.544	0.544
C(10,4)	0.572	0.574	0.594	0.594	0.595	0.595

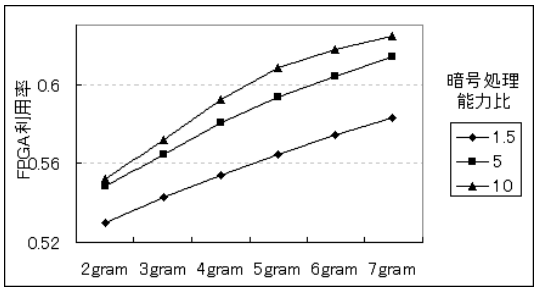


図 10 n-gram 数の変化による FPGA 利用率の変化（メジアン）
Fig. 10 FPGA availability median by changing of n-gram.

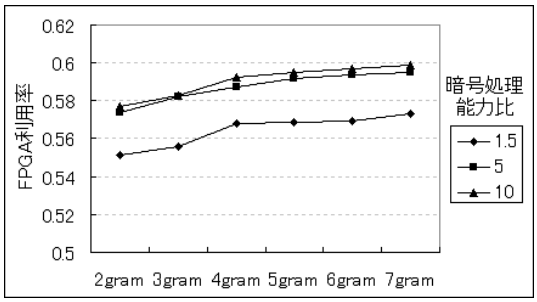


図 11 再構成オーバヘッド 0.05 秒のときの n-gram 数の変化による FPGA 利用率の変化
Fig. 11 FPGA availability by changing n-gram at reconfiguraton overhead 0.05 s.

FPGA 再構成オーバヘッド 0.05 秒に固定したとき，n-gram 数が変化したときの FPGA 利用率の変化を測定したグラフである．縦軸が FPGA 利用率を示し，横軸が n-gram 数，凡例が暗号処理能力比を表している．n-gram 数が 1 つ増えるごとに FPGA 利用率も増加していくことが分かる．

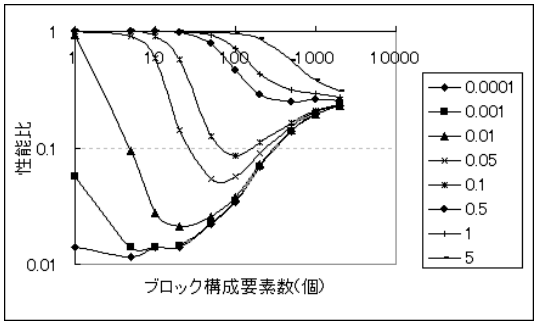


図 12 処理性能比の一例
Fig. 12 An example of processing evaluation rate.

5.2 処理性能比率

次に，すべての暗号処理をソフトウェアのみで実行した場合のレイテンシ 90%分位点の値に対して，予測モデルを用いた場合の処理時間 90%分位点の値の比率を示す．90%分位点（90percentile）というのは，全データを整列したときに最悪値から 10%番目の値を意味する．これにより，ソフトウェアのみですべての処理を実行したときのモデルの処理時間を 1.0 としたときの，予測モデルの処理時間をもって処理性能比とする．よって，この値が小さいほど性能が良い，ということである．

図 12 は，使用可能暗号数 10，FPGA 搭載可能回路数 3，暗号処理能力比 5.0，4-gram 予測を行ったときの，FPGA 再構成オーバヘッド値ごとのソフトウェア処理に対する処理時間比による性能比のグラフである．横軸がブロック構成要素数，縦軸が処理時間比率，凡例は FPGA 再構成オーバヘッド値を表す．図 12 より，処理レイテンシについて，5.1 節で述べた FPGA

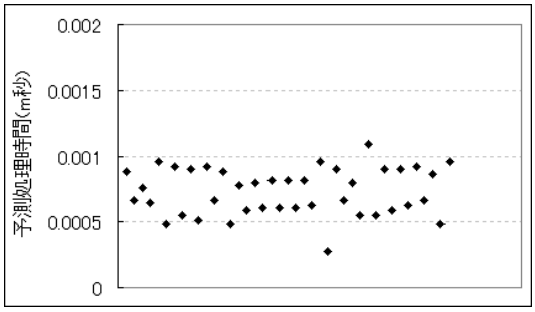


図 13 2-gram 予測をしたときの予測処理時間
Fig. 13 Predicting overhead at 2-gram.

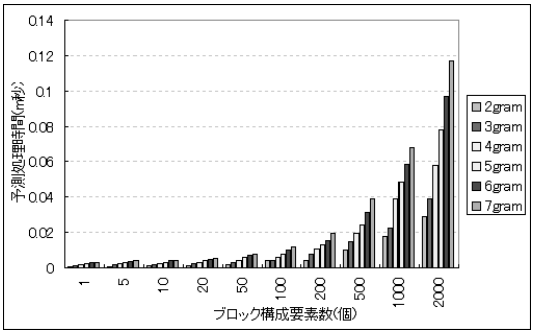


図 14 予測処理に必要な時間（メジアン）
Fig. 14 Predicting overhead median.

利用率と，よく似た性質を持つことが分かる．すなわち，再構成オーバヘッドが大きくなるにつれて，最も良い性能，つまり最も小さい処理時間を示すブロック構成要素数も大きくなっていくことが分かる．このように，FPGA 利用率はこのモデルの性能，すなわち処理時間の小ささに直結する値であるといってよいと考えられる．

このような性質はすべての使用可能暗号数，FPGA 搭載可能回路数，再構成オーバヘッドの値の組合せで確認できた．

5.3 予測処理時間

図 13 は，ブロック構成要素数 1，2-gram 予測をした場合に，1 回の予測処理に要した時間（オーバヘッド）の平均値を測定した結果を，すべての FPGA 再構成オーバヘッドと組合せ数についてプロットしたグラフである．この図 13 全体のメジアンをもって，ブロック構成要素数 1 のときに 2-gram 予測を行った場合の，予測処理時間の代表値とした．このような値を 2-gram，3-gram，⋯，7-gram までの n-gram 数とブロック構成要素数 1〜2,000 について測定し，結果をまとめたグラフが **図 14** である．こうすることで，予測処理に必要な時間がブロック構成要素数と n-gram 数によってどのように変化するのかを測ることができる．横軸がブロック構成要素数，縦軸が予測処理時間（msec），凡例は n-gram 数となっている．図 14 より，予測オーバヘッドは gram 数が増加するに従い，また，ブロック構成要素数が増加するに従い線形に大きくなることが分かる．

これは，予測処理を行う際には｛ブロック構成要素数*（n-gram数 − 1）｝個分の処理履歴を走査し，その結果を基に値列を生成，対応する n-gram 表をひく，という処理をするため，n-gram 数，ブロック構成要素数が増加すると必然的に予測処理にかかる時間が線形に増加するためであると考えられる．また，3.2 節で述べたように n-gram 表は B 木で構築されているので，表の参照に要する時間も指数関数的な増加よりは抑えることができたためであると考えられる．

レイテンシ実測値の最小値は，暗号処理能力比 10.0，再構成オーバヘッド 0.0001 秒でのおよそ 285 msec/bit であり，予測処理時間の最大値は図 14 での 7-gram でブロック構成要素数 2,000 のときの 0.12 msec である．これらの値を比較すると，予測処理時間は暗号処理レイテンシの 1/100 よりもはるかに小さい値であるといえるので，予測処理時間はすべての場合において無視してもよいと考えられる．

5.4 処理待ちリクエスト数

5.4.1 必要最大 queue サイズ

5.2 節で述べたように，FPGA 利用率が高くなると暗号処理のレイテンシを小さくすることが可能となる．レイテンシが小さくなれば，処理待ち queue に溜まるリクエストの数を少なくすることができると考えられる．そこで，本モデルがリクエストを受け付けている間に，どれだけの数のリクエストが処理待ち状態（queue に留め置かれる）となるかカウントするシミュレーション実験を行った．

仮に queue の大きさが無限であるとしたときに，最大でいくつのリクエストが queue に溜まるかをカウントした結果を**図 15** に示す．図 15 は使用可能暗号数 10，FPGA 搭載可能回路数 3，暗号処理能力比 5.0，4-gram 予測を行ったときの，FPGA 再構成オーバヘッド値ごとの queue の最大個数と，すべての処理をソフトウェアで行った場合（凡例の soft）の queue 最大個数である．横軸がブロック構成要素数，縦軸が必要な最大 queue サイズ，凡例は FPGA 再構成オーバヘッド値を表す．ただし，FPGA 再構成オーバヘッド 0.5 s 以上のものについては，soft のグラフと完全に重なってしまうので省略した．

図 15 に示した実験では queue の大きさは無限であるとしたが，実際のシステムでは有限であるため，こ

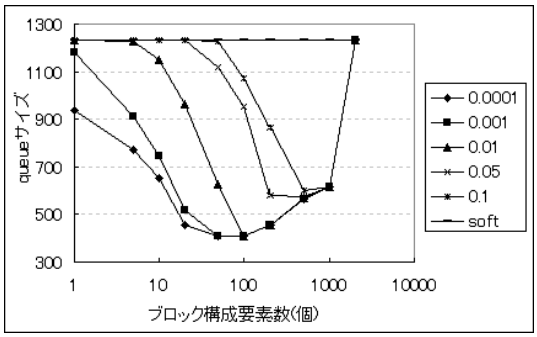


図 15 必要な最大 queue サイズ
Fig. 15 Maximum queue size.

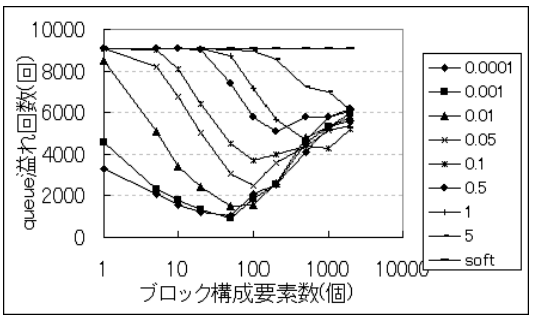


図 16 queue サイズ 128 としたときの queue 溢れ回数
Fig. 16 Queue overflow times at queue size 128.

の処理待ちリクエスト数が大きくなると，queue に入りきらなかったリクエストは破棄されるということになる．

図 15 より，FPGA 再構成オーバヘッドが大きくなるにつれて，必要 queue サイズの最小値が大きくなり，図 12 と似た U 字型のグラフとなることが分かる．なお，図 12 の実験パラメータと図 15 の実験パラメータはまったく同一のものである．このことから，FPGA 利用率が向上すると暗号処理レイテンシが小さくなり，その結果として処理待ちするリクエストの数を少なくすることができると考えられる．

5.4.2 queue 溢れ回数

5.4.1 項において，FPGA 利用率が向上すると，処理待ちするリクエストの数を減少させられると述べた．処理待ちするリクエストの数が減少すれば，それだけサーバが破棄することなく処理しきれるリクエストの数が向上すると考えられる．

そこで，queue の最大値を Linux 系 OS のデフォルト値である 128 とした場合に，何回 queue オーバフローが発生してリクエストを取りこぼしたかをカウントした実験について述べる．queue 溢れが発生しリクエストを取りこぼすということは，システムの処理限界能力を超える負荷がかかっているということを意味する．

queue が溢れた場合には，queue に溜まっているリクエストが処理されて queue に空きができるまでは，新たなリクエストを受け付けることができなくなるものとした．このとき，受け付けられなかったリクエストの数をカウントした結果を**図 16** に示す．

図 16 は使用可能暗号数 10，FPGA 搭載可能回路数 3，暗号処理能力比 5.0，4-gram 予測を行ったときの，queue 溢れ回数のグラフである．横軸がブロック構成要素数，縦軸が必要な queue 溢れ回数，凡例は FPGA 再構成オーバヘッド値と，すべての処理をソフトウェアで行った場合（凡例の soft）を表す．

図 16 より，ソフトウェアですべての処理を行った場合には，つねに 9,000 回程度の queue 溢れが発生することが分かる．本実験で入力するリクエストの総数は 60 万件であるので，およそ 1.5%程度のリクエスト取りこぼしが発生していることが分かる．

n-gram 予測を用いて FPGA による暗号処理を実行すると，ブロック構成要素数や再構成オーバヘッドなどのパラメータにもよるが，9,000 回あった queue 溢れを最も良い場合で 1,000 回程度まで低減できていることが分かる．

以上の結果から，FPGA 利用率が向上すると暗号処理レイテンシが小さくなり，サーバが破棄することなく処理しきれるリクエストの数が向上していることが分かる．

本実験で用いている通信モデルは，1999 年当時のデータのパケット到着間隔を 0.01 倍して使用しており，その負荷は 1,371 packets/s，4.5 Mbps であると 4.1 節で述べた．表 5 に示した暗号処理性能であれば，この通信負荷を十分に処理できると考えられるが，通信リクエストはつねに一定ではなく，ある瞬間には極端に大きな負荷がかかることがある．そのため，図 16 に示したような queue 溢れが発生し，リクエストを取りこぼすことがある，と考えられる．

このように，本通信モデルの負荷は現在の通信ログを利用した厳密に正確な通信負荷ではないが，暗号通信システムに時折リクエストの取りこぼしを発生させる程度の負荷変動を与えている．これにより，n-gram 予測モデルがサーバのピーク性能をどの程度向上させているかを測ることができていると考えられる．

しかしながら，この通信負荷は厳密に現実的な値であるというものではなく，現実的な値に基づいた擬似的な通信負荷データであるととらえられる．このことから，今後は，実際の通信データを用いて解析を行う

必要もあると考えている．

6. 今後の展望

シミュレーション実験の結果，動的計画法を用いて測定した理論上最高の FPGA 利用率は 90%を超える値となった一方で，提案した n-gram 予測モデルを用いた場合には，FPGA 再構成オーバヘッドが小さいときには 80%近い FPGA 利用率を示すが，再構成オーバヘッドが大きくなると，FPGA 利用率は 40%程度となり，理想的な場合に比べて 50%近い差があることが分かった．これは，n-gram 予測モデルでは，リクエストの到着順序情報のみを利用しており，到着時刻，到着時間差などの情報を予測に用いていないためであると考えられる．これらの情報を有効に利用できる予測方式について考えていきたい．

そのほかに，本モデルの性能をさらに向上させるために，暗号回路のマルチスレッド実行という方法が考えられる．たとえば AES-CBC などはデータを暗号化するときに 10 ラウンドの処理を実行する．CBC モードであるため 10 ラウンド終了するまでは次のデータの暗号化が実行できないが，仮に AES-CBC を使用する通信が 10 スレッド存在したとするならば，1 ラウンドに 1 スレッドを割り当ててパイプライン処理をすることで，シングルスレッドの場合の 10 倍のスループットを得ることが可能となると考えられる．本モデルを，このようなマルチスレッドの実行を取り入れたモデルに拡張し，検証を行っていきたいと考えている．

また，FPGA のチップ全体を 1 度に書き換えるのではなく，部分的に再構成可能なデバイスが存在する．このようなデバイスの 1 つである xilinx の xc2vp100 は，表 5 で利用した xc2v8000 デバイスと同等の回路規模である[13]．このようなデバイスの利用は十分に可能であると考えられるので，今後の検討課題としたい．

また，予測アルゴリズムに関して，複数のパケットを溜めておき，それらのヘッダのみを先行解析して，将来利用する暗号を知り，その情報を基に予測を行うという方法が考えられる．この方法では，パケットを溜めておくことで通信バッファを圧迫することが考えられる．しかしながら，この方法で結果として暗号処理のレイテンシを下げられるならば，その分バッファをあける速度も向上するということなので，全体としてバッファ使用量を減らすことができるという可能性も考えられる

また，処理の順番を入れ替えてスケジューリングするという方法も考えられる．これは，処理の順番を入れ替えることで同一の暗号方式を使用するパケットをまとめて処理するという方法である．この方法では，後回しにされたパケットのレイテンシが極端に増大してしまう可能性がある．しかしながら，後回しにされるパケットの数が少数であるならば，そのパケットのレイテンシは増大するものの，全体としてはレイテンシを大幅に減少させる可能性も考えられる．

今後は以上に述べたような予測アルゴリズムについても考えていきたい．

7. 関 連 研 究

本研究で提案したような FPGA を用いたリコンフィギャラブルなシステムによる暗号処理の高速化に関していくつかの研究がなされている．

Bellows らによる研究[14] は FPGA 処理用のボードを作成したのみであり，Chodowiec らによる研究[15] では，このボード上の FPGA に AES 暗号処理回路と 3DES 暗号処理回路を実装して，これらの暗号処理回路の性能評価を行ったのみである．

長谷川らの研究[16] では，再構成オーバヘッドが μ 秒オーダの動的再構成可能プロセッサ DRP-1 を用いた暗号処理システムについて研究がなされている．このようなプロセッサは搭載可能の小さな回路規模が小さく，暗号処理回路のような巨大な回路を機能ブロックごとに分割し，ある瞬間に必要なブロックのみを動的に再構成していくことでチップ面積を大きく見せることができるというものである．

安部らの研究[17] では AES-CBC の実装に 6 コンテキストを要するという報告がなされている．動的再構成可能プロセッサは実装アーキテクチャが特殊であるため単純な比較はできないが，安部らの研究[17] ではループアーキテクチャによる実装がなされており，フルパイプライン化したならば少なくとも 20 コンテキストを使用するものと考えられる．AES-CBC は slice 使用率 34%であるので，1 コンテキストに搭載できる回路の大きさはおおよそ 1.5%程度の slice 使用率ということになり，DEC-CBC の実装に 6 コンテキスト程度を必要とする回路規模であると推定される．

さらに，動的再構成可能プロセッサ DRP-1 は最大 16 コンテキストを保持することができ，最大で 5 コンテキストの中から次を選択し，μ 秒オーダで切替え可能であるとされている[17]．そのため，本研究で必要とするような暗号処理回路全体を動的に再構成するような用途には，現時点ではやや不向きなデバイスであると考えられる．しかしながら，今後動的再構成可能デバイスの実装可能回路規模が大きくなるという可能性は十分に考えられるので，このようなデバイスを用

いた場合の評価についても検討していきたいと考えている．

このほか，複数 FPGA をバンク構成することで動作に関与しないバンクの FPGA をコンフィギュレーションし，終了したら動作中の FPGA バンクと瞬時に切り替えるという手法の研究が Tsutsui らによってなされている[18]．このような手法をとると，見かけ上 FPGA の再構成オーバヘッドが 0 になるうえ，複数の FPGA による暗号処理の並列実行が可能になるため，レイテンシをさらに削減することが可能になると考えられる．この手法を取り入れた予測スケジューリングアルゴリズムについても検討していきたいと考えている．

8. ま と め

クライアント・サーバ型の暗号通信において，暗号処理の部分を，FPGA を動的に書き換えて実行するような暗号通信モデルを提案した．このモデルにおいて，FPGA 再構成オーバヘッドを効果的に削減するため，n-gram モデルを用いた将来の通信の予測を行うアルゴリズムを提案した．

このような n-gram モデルにおいて，性能に影響を及ぼすと考えられる様々なパラメータについて，その値を変化させてモデルの特性について評価を行った．

その結果，FPGA 再構成オーバヘッドが小さいときには，oracle モデルによる理想的な場合の 90%程度の FPGA 利用率を達成できることが分かった．しかし，FPGA 再構成オーバヘッドが大きくなると，徐々に FPGA 利用率は下がるという結果が得られた．

また，FPGA の再構成オーバヘッドが大きくなると，FPGA 利用率を高くするためには，ブロック構成要素数もまた大きくする必要があることが分かった．また，n-gram 数を大きくするにつれて FPGA 利用率も高くなっていくことが分かった．

予測処理そのものにかかる時間は n-gram 数とブロック構成要素数の増加とともに線形に増えていくことが分かった．

謝辞 Web サーバのログの採取などに関して，筑波大学学術情報メディアセンター技術職員の佐藤守氏にご尽力いただきました．ここに感謝いたします．

また，本研究の一部は，日本学術振興会平成 16 年度科学研究費基盤研究（B）(2) 15300013「書き換え可能デバイスによる高速パケット処理の研究」による．

参 考 文 献

1) 末吉，飯田：リコンフィギュラブルコンピューティング，技術報告 8 (1999).
2) 丹羽雄平，前田敦司，山口喜教：暗号通信におけるリクエスト予測を用いた FPGA 再構成オーバヘッドの低減手法，情報処理学会論文誌：コンピューティングシステム，Vol.46, No.SIG12(ACS11), pp.110–119 (2005).
3) http://www.ll.mit.edu/IST/ideval/index.html
4) Xilinx: Virtex-2 Platform FPGAs: ハンドブック.
5) Nechvatal, et al.: Report on the Development of the Advanced Encryption Standard, *Journal of Research of the National Institute of Standards and Technology*, Vol.106.
6) http://www.nist.org/aes
7) http://www.ipa.go.jp/security/enc/CRYPTREC/fy15/doc/
8) http://www.gnupg.org/
9) http://www.opencores.org/
10) 片下敏弘，前田敦司，山口喜教：余剰 FF と位相シフトクロックを利用した FPGA 回路の低消費電力実装手法，信学論 D-I，Vol.J88-D-I.
11) Dandalis, A., et al.: A Comparative Study of Performance of AES Final Candidates Using FPGAs, Lecture Notes in Computer Science, Vol.1965.
12) Elbirt, A., et al.: An FPGA Implementation and Performance Evaluation of the AES Block Cipher Candidate Algorithm Finalists, pp.13–27 (2000).
13) Virtex2-Pro Complete Data Sheet. http:// direct.xilinx.com/bvdocs / publications /ds083.pdf
14) Bellows, P., Bhaskaran, V., Flidr, J., Lehman, T., Schott, B. and Underwood, K.: GRIP: A Reconfigurable Architecture for Host-Based Gigabit-Rate Packet Processing, pp.121–130.
15) Chodowiec, P., Gaj, K., Bellows, P. and Schott, B.: Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board, pp.220–234.
16) 長谷川揚平，安部昌平，松谷宏紀，安生健一朗，粟島 亨，天野英晴：動的再構成可能プロセッサを用いた組み込み向け複数暗号処理エンジンの実装，情報処理学会研究報告 SLDM-119，Vol.2005, No.27, pp.13–18 (2005).
17) 安部昌平，長谷川揚平，黒瀧俊輔，天野英晴，安生健一朗，粟島 亨：リコンフィギャラブルプロセッサ DRP-1 上での AES-CBC の実装，第 4 回リコンフィギャラブルシステム研究会論文集，pp.239–

242 (2004).
18) Tsutsui, A. and Miyazaki, T.: YARDS: FPGA/MPU Hybrid Architecture for Telecommunication Data Processing, pp.93–110.

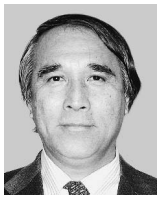
(平成 18 年 7 月 21 日受付)
(平成 18 年 11 月 14 日採録)

丹羽 雄平（学生会員）

2002 年筑波大学第三学群情報学類卒業．同年筑波大学大学院博士課程システム情報工学研究科コンピュータサイエンス専攻入学．現在同在学中．FPGA による暗号処理，FPGA の利用法に興味を持つ．

前田 敦司（正会員）

1994 年慶應義塾大学大学院理工学研究科数理科学専攻単位取得退学．博士（工学）（慶應義塾大学 1997 年）．1997 年電気通信大学大学院情報システム学研究科助手．2000 年筑波大学電子・情報工学系講師．2004 年筑波大学大学院システム情報工学研究科助教授（現職）．並列/分散処理，コンピュータアーキテクチャ，プログラミング言語の実装，ガーベッジコレクション等に興味を持つ．日本ソフトウエア科学会，ACM 各会員．



山口 喜教（正会員）

1972 年東京大学工学部電子工学科卒業．同年通商産業省工業技術院電子技術総合研究所入所，計算機方式研究室長等を経て，1999 年筑波大学電子・情報工学系教授．博士（工学）（東京大学 1993 年）．現在，筑波大学システム情報工学科教授．高級言語計算機，並列計算機アーキテクチャ，並列実時間システム，ネットワーク侵入検知システム等の研究に従事．1991 年情報処理学会論文賞，1995 年市村学術賞受賞．著書『データ駆動型並列計算機』（共著）．IEEEComputer Society，ACM，電子情報通信学会各会員．