ElGamal暗号を用いた安全なファイル送受信Webシステムのための依頼計算

村田 純一† 福田 洋治‡ 毛利 公美†† 白石 善明†
名古屋工業大学† 愛知教育大学‡ 岐阜大学††

1. はじめに

送信者認証と受信URLの受信者へのメール送付によりファイルを特定の利用者に送信するサービスがある．Webクライアントとサービス提供サーバ間の通信路だけをSSL/TLSで暗号化するものが多く，サーバにはファイルが平文のまま送付されてしまう．この場合，サーバは平文のファイルを漏洩させないことと送信者認証に利用する利用者情報を目的外に使用しないという仮定が必要になる．しかしながら，利用者の属する組織外でサーバが運営されている場合，利用者はその仮定が守られているかを確かめることは困難である．そこで本稿では，組織外のサービス提供サーバを介してファイルの送受信をする際に，Webクライアントで暗号化を行い，暗号文をサービス提供サーバに渡すシステムの実現に必要となるElGamal暗号化の依頼計算を提案する．

2. Webクライアント上の暗号化に関わる計算量削減の必要性

送信者がファイルを共通鍵暗号化し，そのファイルの復号鍵（平文）を受信者の公開鍵により暗号化し，Webサーバを経由して暗号化ファイルとファイル復号鍵の暗号文を送信し，受信者が自身の秘密鍵を用いて復号し，ファイル復号鍵（平文）を取り出し，ファイルを復号するという，共通鍵暗号でのファイル暗号化と公開鍵暗号でのファイル復号鍵の受信者への配送を行うWebシステムを用いたファイル送受信システムを考える．

ファイル復号鍵の配送に着目すると，このシステムは，ファイル復号鍵の暗号文を生成する位置によって，次の二つの構成法が考えられる．

1) 送信側のWebクライアントに送信者が公開鍵と平文（ファイル復号鍵）を入力し，クライアント上で暗号文を生成してサーバに送信し，サーバが暗号文を保管する．
2) 送信側のWebクライアントに送信者が公開鍵と平文（ファイル復号鍵）を入力し，それらを暗号化通信路を通じてサーバに送信し，サーバ上で暗号文を生成して保管する．

公開鍵暗号方式として例えばElGamal暗号[1]を用いるとき，暗号化と復号を行う送信者と受信者は$A^B(\mathrm{mod}P)$という冪乗計算をする．これは，素数pが1024ビット程度の場合，平均512回の乗法計算が必要となる．そこで，1)は平文をサーバに知られることなく暗号文を保管することができ，情報漏洩のリスクが低減する．しかしながら，Webクライアントは処理速度の遅いスクリプト言語により暗号化処理が実装されるため，計算量の点から2)の構成法に従ったシステムを構築し，サーバから情報漏洩はしないという仮定の下で運用する既存のシステムも少なくない．

したがって，公開鍵暗号を導入したWebシステムによる安全なファイル送受信システムを実現するためには，Webクライアント上での暗号文生成に関わる計算量の削減が課題となる．この課題解決のために依頼計算という技術がある．RSA暗号に対してはICカード向けの依頼計算方式[2]等が提案されているが，我々はシステムが利用できる暗号の多様性を高めるという観点からElGamal暗号の依頼計算の一方式を提案する．

A Server-Aided ElGamal Computation Protocol for Secure File Transmission by Web System
†Jun-ichi MURATA and Yoshiaki SHIRAISHI・Nagoya Institute of Technology
‡Youji FUKUTA・Aichi University of Education
††Masami MOHRI・Gifu University

3. ElGamal暗号方式の依頼計算による暗号化

サービス提供サーバが部分暗号化し，そしてWebクライアントが完全暗号化するというElGamal暗号化の依頼計算方式を提案する．この依頼計算は，Webクライアントの計算時間を削減し，かつサービス提供サーバに利用者の秘密情報である平文を知られない方式である．

ElGamal暗号の依頼計算として文献[3]がある．文献[3]では，ElGamal暗号の復号時の$C_1^x(\mathrm{mod}\,p)$の計算などを行う場合に，計算式の中の秘密要素に相当する値を依頼計算先のコンピュータに知られることなく，高速に計算結果を得ることを課題とし，その課題解決する計算法を提案している．しかし，文献[3]の方式をElGamal暗号の暗号化に適用した場合，クライアントが乗法計算を，サーバが冪乗計算をしなければならないために比較的計算量が多くなる．そこで，クライアントとサーバの計算量が文献[3]の方式よりも少ない，図1および図2に示すElGamal暗号化の依頼計算方式を提案する．提案方式は次のようになる．

【事前準備】

利用者から暗号文を受け取る受信者は，大きな奇素数pと原始元$g\in Z_p^*$を選択する．次に，秘密鍵$x\in Z_{p-1}$をランダムに選択し，$K=g^x(\mathrm{mod}\,p)$を生成する．最後に，受信者は$PK=(p,g,K)$を公開鍵として公開し，$SK=x$を秘密鍵として保持する．

【暗号化処理】

① 利用者はクライアントを介して，サーバにサービスの利用を要求する．サーバは利用要求を受信すると，クライアントにアプレットを送信する．利用者はアプレットを介して，サーバに実行要求する．サーバは実行要求を受信すると，乱数$r(\mathrm{mod}\,p)$を生成しクライアントを介して，利用者にrを送信する．利用者はrを受信した後，$b\in\{0,1\}$を選択する．そして公開鍵$K=g^x(\mathrm{mod}\,p)$から，真の公開鍵$K_b=K$と偽の公開鍵$K_{\neg b}=r/K_b$を生成する．さらに利用者は受信者に送信したい平文$M(\mathrm{mod}p)$と共に，$(K_b,K_{\neg b},M)$をアプレットに送信する．ここでは，公開鍵の生成を利用者が行うように記述しているが，アプレットで行うのでもよい．

② アプレットは，$(K_b,K_{\neg b},M)$を受信すると，$M=\sum M_i(\mathrm{mod}\,p-1)$となるように平文$M$を$I$個の要素を持つ$(M_1,M_2,\ldots,M_I)$に分割する．そして$J$個の要素を持つ乱数$R=(R_1,R_2,\ldots R_J)$を生成し，$N(=I+J)$個の要素を持つ配列に，分割した平文$(M_1,M_2,\ldots,M_I)$と乱数$R=(R_1,R_2,\ldots R_J)$をランダムに挿入することによって$Z=(Z_1,Z_2,\ldots,Z_N)$を生成する．ここで，$Z$の生成方法を詳しく説明する．まず，0から$N-1$の相異なる整数を$I$個の要素として持つ乱数$RC=(RC_1,RC_2,\ldots,RC_I)$を生成する．次に$N$個の要素を持つフラグ配列$F=(F_1,F_2,\ldots,F_N)$を定義し，配列の要素を全て「0」で初期化する．そしてRCのi番目の要素の値をFの要素番号としF_{RC_i}とする．そしてF_{RC_i}となる各要素番号についてFに1を立てる，つまり$F_{RC_i}=1$とする．次に，$F_n=1$であるFの要素番号に対応するZに対して，分割した平文$(M_1,M_2,\ldots,M_I)$を先頭から順番に挿入する．

一方，$F_n=0$ である F の要素番号に対応する Z に対しては，乱数 R を先頭から順番に挿入する．以上が，分割した平文 $(M_1,M_2,...,M_I)$ に乱数 R をランダムに挿入した Z を生成する方法である．この Z に対して区別が付かないような N 個の要素を持つ乱数 $Z'=(Z'_1,Z'_2,...,Z'_N)$ を生成し，真の公開鍵と平文・乱数をランダムに挿入したものの組である (K_b,Z) と，偽の公開鍵と乱数の組である $(K_{\neg b},Z')$ をサーバに送信する．

③ サーバは，2 組のデータ (K_b,Z)，$(K_{\neg b},Z')$ を受信すると，2 つの鍵 K_b，$K_{\neg b}$ を用いて，$K_bK_{\neg b}=r$ であるかをチェックする．これにより，真の公開鍵と偽の公開鍵のペアはサーバ自身が生成した乱数 r から生成された鍵であるかを検証することができる．検証された結果が正しければ鍵ペアは正当であると判断し処理を続ける．検証した結果，鍵ペアが不当であるならば処理を終える．次にサーバは乱数 h を生成した後，部分暗号文を計算する前に $g^h,K_b^h,K_{\neg b}^h$ を冪乗計算して求めておく．これらの計算結果は，次に行う部分暗号文の各要素の計算に共通して必要なものであるため，先に計算しておく．最後にこの冪乗計算の結果を用いて，部分暗号文 $E_b=(E_b^{(1)},E_b^{(2)},...,E_b^{(N)})$ と $E_{\neg b}=(E_{\neg b}^{(1)},E_{\neg b}^{(2)},...,E_{\neg b}^{(N)})$ を計算する．これらの各要素は $E_b^{(n)}=(g^h,Z_nK_b^h)$，$E_{\neg b}^{(n)}=(g^h,Z'_nK_{\neg b}^h)$ として計算するが，$g^h,K_b^h,K_{\neg b}^h$ は既に計算しているため，$Z_nK_b^h$ と $Z'_nK_{\neg b}^h$ の乗法計算だけをすればよい．そして，部分暗号文 $(E_b,E_{\neg b})$ をアプレットに送信する．

④ アプレットは，$(E_b,E_{\neg b})$ を受信すると，E_b を受理し，$E_{\neg b}$ を破棄する．次に，アプレットはフラグ配列 F を参照して，E_b の $F_n=1$ に対応する要素番号 $E_b^{(n)}$ の $Z_nK_b^h$ を取り出して $M_iK_b^h$ とする．$M=\sum M_i(\mathrm{mod}\,p-1)$ より，$\sum(M_iK_b^h)=\sum(M_i)K_b^h=MK_b^h$ と計算する．受信した g^h と計算した MK_b^h が完全暗号文 $(C_1,C_2)=(g^h,MK_b^h)$ となっている．

これは，文献[3]の方式を ElGamal 暗号の暗号化に適用した場合よりも，クライアントとサーバの計算量が少ない ElGamal 暗号化の依頼計算方式となっている．

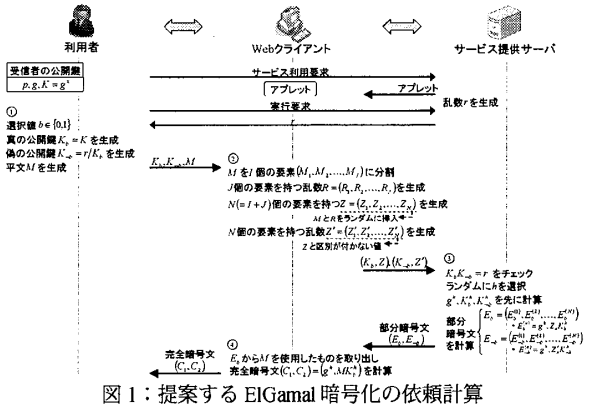

図 1：提案する ElGamal 暗号化の依頼計算

4. 計算量の比較

文献[3]の方式を ElGamal 暗号の暗号化に適用した場合と提案方式についてクライアントとサーバの計算量の比較を行った(表 1)．文献[3]の方式を ElGamal 暗号の暗号化に適用した場合では，クライアントが乗法計算を $2I-1$ 回行い，サーバが冪乗計算を $2N$ 回行う．提案方式では，クライアントが加法計算を $I-1$ 回，除法計算を1回行い，サーバが冪乗計算を 3 回，乗法計算を $2N+1$ 回行う．つまり，クライアント側については，文献[3]の方式を ElGamal 暗号の暗号化に適用した場合では乗法計算をしていたところを提案方式では加法計算に，また，サーバ側についても，冪乗計算をしていたところを提案方式では乗法計算に落とすことができた．これにより，クライアント側とサーバ側の両方で，文献[3]の方式を ElGamal 暗号の暗号化に適用した場合よりも提案方式の方が計算量は少なくなっており，Web クライアント上での暗号文生成に関わる計算量の削減ができたと言える．

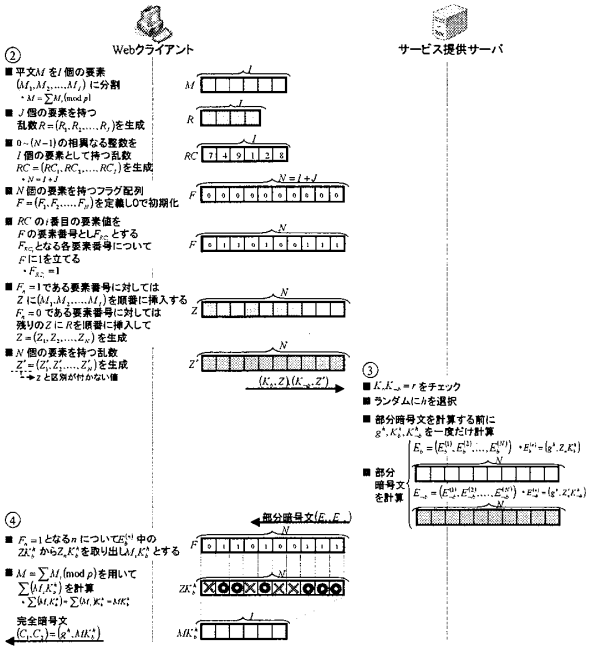

図 2：提案する ElGamal 暗号化の依頼計算方式の手順

表 1：文献[3]の方式を ElGamal 暗号の暗号化に適用した場合と提案方式の計算量の比較

	文献[3]の方式を ElGamal 暗号の暗号化に適用した場合		提案方式	
	クライアント	サーバ	クライアント	サーバ
冪乗計算	0	$2N$	0	3
乗法計算	$2I-1$	0	0	$2N+1$
加法計算	0	0	$I-1$	0
除法計算	0	0	1	0

5. まとめ

本研究では，分割した平文と乱数をランダムに挿入したものをサーバに依頼計算させることで，クライアントが加法計算，サーバが乗法計算をするという計算量的な負担が少なく，サーバに利用者の秘密情報である平文を知られないという特徴を持つ ElGamal 暗号を用いた安全なファイル送受信 Web システムのための依頼計算方式を提案した．文献[3]の方式を ElGamal 暗号の暗号化に適用した場合と提案方式の比較により，計算量的な効率が良いこと示した．

参考文献

[1] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based On Discrete Logarithms, ",IEEE Trans. on Inform. Theory, IT31, 4, pp.469-472, 1985

[2] Shin-ichi KAWAMURA, Atsushi SHIMBO, "Performance Analysis of Server-Aided Secret Computation Protocols for the RSA Cryptosystem," Transactions of the IEICE, Vol.73, No.7, pp.1073-180, 1990.

[3] 中村 隆之，鮫島 吉喜，"依頼計算方法，" 特許公開 2002-108210.