有料配信における海賊版受信機・パスワード漏洩抑止スキーム

Traitor-Tracing System for Pay Broadcast Distribution

渡辺秀行 [1)] 光成滋生 [2)] 藤井寛 [3)]
WATANABE Hideyuki MITSUNARI Shigeo FUJII Hiroshi
石田計 [4)] 酒居敬一 [5)]
ISHIDA Kei SAKAI Keiichi

1) 株式会社アイビス（〒450-0002 愛知県名古屋市中村区名駅 3-18-11 新明ビル 4F
E-mail: watanabe@ibis.ne.jp）
2) 株式会社ピクセラ（〒590-0985 大阪府堺市戎島町 4 丁 45 番地の 1 ポルタス・センタービル 5F
E-mail: herumi@nifty.com）
3) 岡山大学大学院自然科学研究科博士課程前期（〒700-8530 岡山県岡山市津島中 3-1-1
E-mail: hino@mx1.tiki.ne.jp）
4) 有限会社サムス（〒305-0032 茨城県つくば市竹園 2-14-12-E101
E-mail: kei-i@sams.co.jp）
5) 広島大学大学院工学研究科コンピュータ・アーキテクチャ学（〒739-8527 広島県 東広島市 鏡山 1-4-1
E-mail: sakai@aial.hiroshima-u.ac.jp）

ABSTRACT. We construct a crypto-system which **one** encryption key corresponds to **many** decryption keys though the correspondence of most public key crypto-system is one-to-one, and implement the routine of fast computing Tate-pairing on an elliptic curve which is necessary to the scheme. It can be applied to broadcast distribution system such as pay-TV.

1 背景

急速に IT 化が進みつつある現代では個人や法人を問わず様々な面において暗号に関する技術は必要不可欠である。今後電子取り引きや電子マネーなどが普及するとますますその重要度は増してくる。しかし、その暗号基盤技術を提供する会社となると、海外には RSA Data Security や F-secure などがあるが、日本には世界標準の技術を請け負う有力な会社がなく、技術はあっても学者の中の理論で閉じてしまっているものが多い。

これは日本の大企業が技術を自社内に抱え込んでしまう体制、また日本発の技術を信頼、信用して使用しないという体制に一因があると思われる。日本発の技術であるにもかかわらず海外で先に評価され、逆輸入されたケースはよく見受けられる。

そこで日本の中でアイディアとしてだけ存在していた暗号技術を形にし、世界に送り出すことを目的とした新規企業を設立したい。それにより後発であった暗号技術に関しても世界標準を目指し「技術大国日本」を築く一助としたい。

その第一弾として、今後重要性が増すと思われる有料配信及び放送において、提案者の一人である光成らが取得した特許を元に、不正受信及び不正受信装置の作成を防ぐことを目的とした通信方式の実装を行なう。また、近い将来この通信方式を標準プロトコルとすることを目指す。

2 目的

従来暗号化鍵と復号鍵が一対一であった公開鍵暗号系に対して、暗号化鍵と復号鍵が一対多である暗号化システムを設計する。そのシステムを有料配信に応用した場合、復号鍵に各受信者固有の情報を埋め込むことで鍵漏洩抑止力を持たせることが可能になる。ここではその暗号処理の核となる数学的演算ルーチンを実装する。具体的には楕円曲線上のペアリング（テイトペアリング）演算を行う高速なルーチンを作成しユーザ鍵生成・セッション鍵暗号化・復号 API を提供する。

3 本文

（1）モデル

有料の放送型コンテンツ配信を想定して以下の三者が存在するモデルを考える。

a. ライセンス管理センタ（以下、センタなどと呼ぶことがある）
b. コンテンツ配信元（以下、配信元、プロバイダなどと呼ぶことがある）
c. コンテンツ視聴者（以下、視聴者、ユーザなどと呼ぶことがある）

このとき、配信元が視聴者にコンテンツを提供するために必要な流れは以下の三つである（パラメータや用語は後述する）。

d. センタによる公開鍵の公開
    1. △センターは秘密パラメータ P、a 及び公開パラメータ R を決定する。
    2. ○センタはセンタ用公開鍵を生成する。
    3. センタはセンタ用公開鍵を公開し、配信元はそれを取得する。
e. ライセンス登録
    1. ユーザがセンタに個人情報を送る。

2. ○センタはユーザ情報からユーザごとに異なる秘密鍵を生成する。
3. センタは各ユーザに秘密裏に秘密鍵を送る。

f. 視聴（コンテンツ配信）
1. ○配信元はセッション鍵を生成する。
2. 配信元はセッション鍵でコンテンツを暗号化する。
3. ○配信元はセッション鍵を暗号化する。。
4. 配信元は暗号化されたセッション鍵とコンテンツを配信し、ユーザは受信する。
5. ○ユーザはセッション鍵を復元する。
6. ユーザは復元されたセッション鍵でコンテンツを復号する。

（注）ここで○△印の部分が本ライブラリの提供する機能である。ただし、△印の部分は暗号の安全性を高めるためには詳細な検討を要する可能性がある。

各ユーザに渡された秘密鍵はすべて相異なるが配信元は一つの鍵で暗号化すればよい。ユーザが秘密鍵を漏洩した場合、その鍵に含まれるユーザ情報からそのユーザを調査・特定することが可能である。

（2）システムパラメータ

ここで実装するシステムのパラメータ及び式は以下の通りである。

2．1）パラメータ

g. システム全体のパラメータ（実装部分に埋め込み）
素数 p、m 及び楕円曲線 E/F_p を選ぶ。ただし $E[m] \subset E(F_q)$ （$q = p^2$）とする。ペアリング関数は

$$e(P,Q) = t_m(P, \phi(Q)) \tag{1}$$

（$P, Q \in E[m]$、t_m はテイトペアリング ϕ はねじれ写像）である。

h. センタの秘密情報
$E[m]$ の元 P 及び F_m の元 a を選ぶ。

i. センタの公開情報
$E[m]$ の元 R、Q（ただし、$e(P,R) \neq 1$）、及び F_q の元 g（Q, R, g のペアを「センタ用公開鍵」と呼ぶ）を選ぶ。

j. セッション毎の配信元の秘密情報
F_m の元 r（セッション毎の乱数）、F_q の元 s（セッション鍵）を選ぶ。

k. セッション毎の配信元の公開情報
$E[m]$ の元 X、Y である（X と Y のペアを「暗号化されたセッション鍵」と呼ぶ）。

l. ユーザの秘密鍵
F_m の元となるように符号化された個人情報 u と $E[m]$ の元 K_u のペアである。

2．2）計算式

m. センタによる Q、g の計算（機能 d-2）

$$Q = aR,\ g = e(P,R) \tag{2}$$

n. センタによるユーザ用秘密鍵の生成（機能 e-2）

$$K_u = 1/(a+u)P \tag{3}$$

o. 配信元によるセッション鍵の生成（機能 f-1）

$$s = g^r \tag{4}$$

p. 配信元による暗号化されたセッション鍵の生成（機能 f-3）

$$X = rR,\ Y = rQ \tag{5}$$

q. ユーザによるセッション鍵の復元（機能 f-5）

$$e(K_u, uX + Y) = s \tag{6}$$

（3）実装

1024 ビットの大きさの有限体上の楕円曲線上のテイトペアリング演算プログラムを実装する。実装した OS・CPU は以下の通り：

マシン	CPU	OS	開発環境
AT 互換機	Pentium 4	Windows	Visual C++ Intel C++/ nasm
AT 互換機	Pentium 4	Linux	gcc Intel C++/ nasm
SPARC 機	Sparc	Solaris	Forte C++
ザウルス	ARM	Linux	gcc（クロス開発）
MIPS 機	MIPS	Linux	gcc（クロス開発）
Macintosh	PowerPC	darwin	gcc/gas

特に Intel Pentium 4 プロセッサにおいては専用の SIMD 命令を用いてアセンブリ言語による最適化を行った。

3．1）剰余算機能付き固定長整数演算

ペアリング演算に必要な 512 ビット/160 ビット専用の多倍長演算ルーチンを作成した。その際、乗算の高速化に重点を置きデータ形式も SIMD 命令で扱いやすい方式を検討した。

まず以下の条件を満す p を選定した。

a. p は約 512 ビットの素数である。
b. p は冗長 2 進数（-1 を許す）で表現したときに Hamming weight が小さく、かつビットが 1 または -1 の部分が 2^{32} のべき乗になっている。
c. $p+1$ は約 160 ビットの素因数を持つ。

今回は複数の計算機で約 1 ヶ月のしらみつぶし探索を行い

$$p = 2^{512} - 2^{384} + 2^{160} - 2^{128} - 2^{64} - 1 \tag{7}$$

を選択した。

次に整数の内部表現は以下の 4 通りの実装をし比較を行った。

1. 20 要素固定長 (640 ビット) 符号なし（MI）
剰余算の回数を減らすため、冗長性をある程度持たせる。609 ビット以内の数を $x * 2^{512} + y$（x は 97 ビット以内、y は 512 ビット以内）と表現すると、剰余算によりこの値は $x * (2^{512} - p) + y < 2p$ となる（x を $2^{(32\text{ の倍数})}$ 倍するのは単なるメモリ移動だけなのでこの演算は高速である）。よって、必要ならばこの結果から p を減算することにより、剰余算が完結する。
2. 17 要素固定長（544 ビット）符号なし （MIS）
乗算時の剰余算を高速化するため、またあまり加減算が連続することがないことから、最低限の要素数（加算時の繰上がりのために 17 要素は必要）による実装を行なう。1. と同様に一度の高速な剰余算により $2p$ 未満の数になるので必要ならば p を減算して剰余算が完結する。
3. 32 要素固定（512 ビット）冗長符号付き （R32）
各要素は 32 ビット符号付き数であるが、要素の位どりは 2^{16} で行なう。数を表わす配列を `a[0...31]` とするとこの配列で表わされる値は、

$$\texttt{a[0]} + \texttt{a[1]} * 2^{16} + \texttt{a[2]} * 2^{32} + \cdots + \texttt{a[31]} * 2^{16*31} \tag{8}$$

となる。よって各要素 16 ビット分の冗長性を持つ。冗長性の正規化操作には 2 つの操作があり、一つは各

要素を 16 ビット以内にする操作（ここでは簡易剰余算と呼ぶ）と、0 以上 p 未満にする操作（剰余算）がある。簡易剰余算でも 0 以上 2^{512} 未満となるので、乗算の前準備などの通常の用途では完全な剰余算を行なう必要がない。 また剰余算完了フラグを兼ねた冗長度指数を持つ。この値が 14 ビットを越えると簡易剰余算により正規化する必要がある。

4. 16 要素固定（512 ビット）冗長符号付き （R64）
   各要素は 64 ビット符号付き数であるが、要素の位どりは 2^{32} で行なう。数を表わす配列を `a[0...15]` とするとこの配列で表わされる値は、

$$\texttt{a[0]}+\texttt{a[1]}*2^{32}+\texttt{a[2]}*2^{64}+\cdots+\texttt{a[15]}*2^{32*15} \quad (9)$$

   となる。よって各要素 32 ビット分の冗長性を持つ。簡易剰余算において各要素が 32 ビット以内になることと剰余算完了フラグの閾値が 30 ビットである点を除いて 3. と同様である。

Pentium4 では R64 が速く、MIPS では MIS が速いなどアーキテクチャによる差が見られた。

３．１）楕円曲線の演算

Jacobi 座標系（$Y^2 = 4X^3 - 12XZ^4$ を利用）で windows 法による実装を行った。

（４）今後の課題

実装中に鍵の安全性に関する問題が見つかった。ライブラリとしては一部のパラメータを隠蔽することで対処可能と思われるが、より詳細な検討が必要である。また予め配布しておく秘密鍵を変更することなく特定のユーザの排除をする機能の追加も検討している。

4 参加企業及び機関

なし

5 参考文献

論文

S. Mitsunari, R. Sakai and M. Kasahara, “A New Traitor Tracing,” IEICE TRANS. Fundamentals, vol. E85-A, No. 2, pp. 481–484, 2002.

特許

発明の名称：生成装置, 暗号化装置, 復号化装置, 生成方法, 暗号化方法, 復号化方法, プログラム, ならびに, 情報記録媒体
出願番号：特願 2001-66080