9

Biometric Encryption Using Co-Z Divisor Addition Formulae in Weighted Representation of Jacobean Genus 2 Hyperelliptic Curves over Prime Fields

Robert Brumnik[1], Vladislav Kovtun[2], Sergii Kavun[3] and Iztok Podbregar[4]
[1]*University of Maribor, Faculty of Criminal Justice and Security,*
[2]*National Aviation University, Kiev,*
[3]*National University of Economic, Kharkov,*
[4]*University of Maribor, Faculty of Criminal Justice and Security,*
[1,4]*Slovenia*
[2,3]*Ukraine*

1. Introduction

Security in biometry is a prime concern of modern society. Identity theft is a growing problem in today's interconnected world. To ensure a safe and secure environment biometrics is used today in many commercial, government and forensic applications. To ensure a high level of security of a biometric system we use Cryptographic algorithms. Though a number of bio-crypto algorithms have been proposed, they have limited practical applicability due to the trade-off between recognition performance and security of the template. Overall, these are very secure, however, they do have a weak point in terms of the procedure and storage of the crypto keys.

Biometric authentication systems should have many exploitable crypto secure points that can be used to compromise the identification system within the optimization process. Biometric encryption with Jacobean Genus 2 Hyperelliptic curves is a security scheme that combines strong cryptographic algorithms with biometric authentication to provide enhanced security. This paper discusses the simple implementation Co-Z divisor addition formulae in a weighted representation of encryption systems for biometric software application.

In this article the authors show a newly developed Co-Z approach to divisor scalar multiplication in Jacobean of Genus 2 Hyperelliptic curves over fields with odd characteristics in weighted coordinates for application in biometric-based authentication systems. We assess the performance of these biometric generation algorithms using Co-Z divisor. This approach is based upon improved additional formulae of weight 2 divisors in weighted divisor representation which, in the most frequent cases are well suited to exponentiation algorithms based on Euclidean addition chains.

2. Cryptographic applications

The progress of civilization and the constantly increasing role of various technologies in human day-to-day activities has lead to the permanent development of access control

security systems for information and physical objects. A number of researchers have studied the interaction between biometrics and cryptogra-phy, two potentially complementary security technologies (Hao et al., 2005). For such systems, the importance of authentication and identification subsystems is undeniable. One of the approaches to the implementation of authentication and identification subsystems is biometric systems.
The process of creating biometric systems of authentication and identification has solved a large variety of problems among which the assurance of confidentiality and integrity of biometric information, which is by no means unimportant. As a rule, cryptographic transformations are used – encryption and electronic digital signature (EDS). The analysis, carried out by the authors, has shown the prospectivity of algebraic curves theory for the implementation of cryptographic transformations.

2.1 Elliptic curves

The usage of elliptic curves (EC) for cryptographic purposes was first suggested by (Koblitz, 1987) and (Miller, 1985).
Basiri et al. (2004) present two algorithms for the arithmetic of cubic curves with a totally ramified prime at infinity. The first algorithm, inspired by Cantor's reduction for hyperelliptic curves, is easily implemented with a few lines of code, making use of a polynomial arithmetic package.
In his research, Koblitz (1989) has proved the possibility of usage of hyperelliptic curves (HEC) in cryptographic applications. Their difference from EC is that for HEC the group (Jacobian) of more complex structures should be considered – divisors instead of curve points. It is known that HEC have a variety of advantages over EC: being a richer source of the Abelian groups (Koblitz, 1989; Menezes and Wu, 1998) they also use a base field of a smaller size (from the Abelian group, the size of which is defined by product of the base field size by a curve genus).

2.2 Hyperelliptic curves

Hyperelliptic curve cryptosystems (HCC for short) is a generalization of ECC. It has been drawing the attention of more and more researchers in recent years. The problem of how to decrease the amount of addition and scalar multiplication on the Jacobians of hyperelliptic curves so that the implementation speed can be improved is very important for the practical use of HCC (Zhang et al., 2001).
During the time in which HEC cryptosystems were restricted to academic interest only, they had no practical application due to the high complexity of software and hardware implementation, low performance, absence of in-depth studies in the field of cryptoanalysis of such cryptosystems and the absence of comprehensible algorithms of cryptosystem parameters generation. Active review research of papers (Koblitz, 1989; Menezes et al., 1998; Lange 2002c; Matsuo et al., 2001; Miyamoto et al., 2002; Takahashi et al., 2002; Sugizaki et al. 2002, etc) has allowed us to overcome the majority of the described difficulties. The authors of publications, offer a variety of approaches which increase the performance of HEC cryptosystems essentially, and bring them close to the EC cryptosystems.

2.3 Genus 2 HEC cryptosystems over prime fields

The given research is devoted to the development of the approach (Cohen et al., 1998) and to the improved efficiency of genus 2 HEC cryptosystems over prime fields.

Scalar multiplication operation is used in encryption, decryption and electronic digital signature based on HEC. These computations are relatively expensive when implemented on low-power devices. A widely used standard method is the left-to-right binary method. In accordance with (Koblitz, 1989; Menezes et al., 1998; Lange 2002c; Matsuo et al., 2001; Miyamoto et al., 2002; Takahashi et al., 2002; Sugizaki et al. 2002; Lange, 2002; Kovtun and Zbitnev, 2004) the power consumption traces of divisor addition and doubling are not the same, they can easily be distinguished between these operations and derive the bit of scalar. The first method proposed, with resistance to the side channel attacks (SCA), is Coron's dummy addition (CDA) (Coron, 1999).
Several SCA-resistant scalar multiplication algorithms have been proposed that are faster than the CDA method. There are three basic approaches with SCA resistance:

- The first is to use indistinguishable additions and doubling algorithms in scalar multiplication (Clavier and Joye, 2001). For example, Jacobi form and Hesse form of EC. However, this requires specially chosen EC and HEC curves and does not work for the standardized curves.
- The second is the double-and-always-add approach. The CDA method is the simplest algorithm of this type. In paper (Okeya and Sakuri, 2000), the authors proposed to use the Montgomery form of EC and extended it to general curves (Brier and Joye, 2002).
- The third approach is to use a special addition chain with a sequence of additions and doublings that does not depend on the bit information of the scalar (Izu and Takagi, 2002).

In this paper, we are interested in scalar multiplication algorithms that do not require specially chosen curves and based on approach (Meloni, 2007) for genus 2 HEC over prime fields.

3. Biometric cryptosystems

In a generic cryptographic system the user authentication is possession based. That is, possession of the decrypting key is a sufficient evidence to establish user authenticity. Because cryptographic keys are long and random, (e.g., 128 bits for the advanced encryption standard (AES) (NIST, 2008; Stallings, 2003), they are difficult to memorize. As a result, the cryptographic keys are stored somewhere (for example, on a computer or a smart card) and released based on some alternative authentication (e.g., password) mechanism, that is, upon assuring that they are being released to the authorized users only. Most passwords are so simple that they can be easily guessed (especially based on social engineering methods) or broken by simple dictionary attacks (Klein, 1990).
It is not surprising that the most commonly used password is the word "password"! Thus, the multimedia protected by the cryptographic algorithm is only as secure as the passwords (weakest link) used for user authentication that release the correct decrypting key(s). Simple passwords are easy to crack and, thus, compromise security; complex passwords are difficult to remember and, thus, are expensive to maintain. Users also have the tendency to write down complex passwords in easily accessible locations. Further, most people use the same password across different applications and, thus, if a single password is compromised, it may open many doors. Finally, passwords are unable to provide nonrepudiation; that is, when a password is shared with a friend, there is no way to know who the actual user is. This may eliminate the feasibility of countermeasures such as holding conniving legitimate users accountable in a court of law. Many of these limitations of the traditional passwords can be ameliorated by incorporation of better methods of user authentication. Biometric authentication (Jain et al., 1999; Maltoni et al., 2003) refers to verifying individuals based on

their physiological and behavioural characteristics such as face, fingerprint, hand geometry, iris, keystroke, signature, voice, etc. It is inherently more reliable than password-based authentication, as biometric characteristics cannot be lost or forgotten (cf. passwords being lost or forgotten); they are extremely difficult to copy, share, and distribute (cf. passwords being announced in hacker websites) and require the person being authenticated to be present.
A brief comparison of some of the biometric identifiers based on seven factors is provided (Wayman, 2001):

- universality (do all people have it?),
- distinctiveness (can people be distinguished based on an identifier?),
- permanence (how permanent is the identifier?), and
- collectability (how well can the identifier be captured and quantified?) are properties of biometric identifiers.
- performance (speed and accuracy), acceptability (willingness of people to use), and circumvention (foolproof) are attributes of biometric systems.

4. Background

Let's observe basic concepts of cryptosystems on HEC. More detailed information can be obtained from (Koblitz, 1989; Menezes and Wu, 1998).
Let K be a field and $\overline{K}$ be the algebraic closure of K. Hyperelliptic curve C of genus $g \geq 1$ over K is a set of points (u, v) that satisfy the equation:

$$C : v^2 + h(u)v = f(u), \quad k[u, v] \tag{1}$$

and there are no solutions $(u, v) \in \overline{K} \times \overline{K}$ which simultaneously satisfy the equation (1) and the partial derivative equations with corresponding variables.

$$2u + h(u) = 0, \; h'(u)v - f'(u) = 0 \tag{2}$$

In case of genus 2 HEC, polynomials $f(u)$ and $h(u)$ will be represented as:

$$f(u) = u^5 + f_4u^4 + f_3u^3 + f_2u^2 + f_1u + f_0 \tag{3}$$

and

$$h(u) = h_2u^2 + h_1u + h_0, \; h_i, f_j \in K. \tag{4}$$

Divisor D is a formal sum of points in C:

$$D = \sum_{P \in C} m_P P .. m_p \in Z \tag{5}$$

where only a finite number of the m_P are non-zero.
Divisor $D \in \mathbf{D}^0$ is a principal divisor, if $D = \mathrm{div}(R)$ for some rational function $R \in \overline{K}(C)^*$. The set of all principal divisors, denotes $P_C(\overline{K}) = \{\mathrm{div(F)} : F \in \overline{K}(C)\}$, in curve C over $\overline{K}$, moreover $P_C(\overline{K})$ is a subgroup of $\mathbf{D}^0$. Generally $P(C) = P_C(\overline{K})$ is called a group of principal divisors of curve C. The quotient group $J_C(\overline{K}) = \mathrm{div}_C^0(\overline{K}) / P_C(\overline{K})$ is called the Jacobian of the

curve C over $\overline{K}$. The quotient group $J(C) = \mathrm{div}^0(C)/P(C)$ is called Jacobian of the curve C.
Furthermore, we will operate with divisors in the Mumford representation (Menezes, 1998):

$$D = \left(x^2 + u_1 x + u_0,\, v_1 x + v_0\right),\ \deg v < \deg u \le 2,\ u \mid f(u) - h(u)v - v^2$$

$$\text{where } \forall D_i \in J(C),\ weight(D_i) = 2,\ i = \overline{1,2} \tag{6}$$

The result $D_3 = D_1 + D_2$ will have a $weight(D_3) = 2$, which helps to avoid consideration of alternative addition methods for divisors of different weight and with intersecting support (containing intersecting sets of points) (Lange, 2003; Wollinger, 2004).
HECC uses a divisor scalar multiplication operation:

$$\underbrace{D + D + \cdots + D}_{k} = k \cdot D \tag{7}$$

At the intermediate computation phase of scalar divisor multiplication (scalar multiplier in binary notation) the binary algorithm performs the divisor addition and doubling operation. The addition and doubling algorithms use field $\mathbf{GF}(p)$ multiplicative inversion, which is the most computationally intensive and space critical operation. Projective divisor representation (Miyamoto et al., 2002; Takahashi, 2002; Lange, 2002c) is one of the most popular approaches which allows saving of a field inversion.
In her work Lange (2002c), suggested a weighted divisor representation, being the development of a projective approach.
In weighted representation, the divisor D we can present as Lange (2002c):

$$D = \left(x^2 + u_1 x + u_0, v_1 x + v_0\right) \tag{8}$$

which is of the form of:

$$D = \left[U_1, U_0, V_1, V_0, Z_1, Z_1^2, Z_2, Z_2^2\right] \tag{9}$$

while

$$D = \left(x^2 + U_1/Z_1^2\, x + U_0/Z_1^2,\, V_1/Z_1^3 Z_2 x + V_0/Z_1^3 Z_2\right) \tag{10}$$

Note, that arithmetic in Jacobian genus 2 HEC in weighted representation (Lange, 2002c) is the most efficient (Kovtun and Zbitnev, 2004).

5. Co-Z approach

The Co-Z approach was first suggested by Meloni (2007), in order to increase the efficiency of scalar multiplication in EC over $\mathbf{GF}(p)$ with double-free addition chain for the resistance side channel attacks (SCA) (Goundar, 2008). It results in a fixed sequence of operations; hence attackers could not detect any information through SCA. Its principles lie in transformation of EC points in SCA resistant scalar point multiplication in projective and

modified Jacobi representation with the same denominator and further operation with points of identical Z-coordinates. Note that the Co-Z approach is applicable for algorithms, based on the Euclidian addition chains approach by Meloni (2007) and scalar in the Zeckendorf representation, in order to replace doublings by Fibonacci numbers computations refer to Algorithm A.1. Indeed the Fibonacci sequence is an optimal chain (Meloni, 2007).

The Zeckendorf number representation needs 44% more digits in comparison with the binary representation. For example a 80-bit integer will require around 115 Fibonacci digits. However, the density of 1's in this representation in lower (near 0.2764). This means that representing a 80-bits integer requires, an average 40 powers of 2 but only 32 Fibonacci numbers (near 115x0,2764).

More generally, for a n-bit integer, the classical double-and-add algorithm requires on average $1.5\times n$ operations ($\frac{n}{2}\mathbf{A}+n\mathbf{D}$, where $\mathbf{A}$-addition operation and $\mathbf{D}$-doubling operation) and the Fibonacci-and-add requires $1.83\times n$ operations ($1.44\times n\mathbf{F}+0.398\times n\mathbf{A}$, where $\mathbf{A}$-addition step and $\mathbf{F}$-Fibonacci step). In other words, the Fibonacci-and-add algorithms A.1 require about 23% more operations (Table 1.). Note, that in paper (Lange, 2002c) the simplified version of HEC is used, such that $h(x)=0$, $f_4=0$ (since replacement $y\mapsto y-h/2$ is admissible for the odd field characteristic and the replacement $x\mapsto x-f_4/5$ is admissible if $p\neq 5$), which allowed T. Lange to save 1 multiplication at the step A.2.7. However in this paper we will consider a more general case of curve with $\deg(h)=2$, $h_i\in\mathbf{F}_2$, $i=\overline{0,2}$ $\deg(f)=5$, $f_5=1$, $f_i\in\mathbf{GF}(p)$, $i=\overline{0,4}$.

Input: $D\in J_C$, $k=(d_l,\ldots,d_2)_Z$

Output: $[k]D\in J_C$

```
begin
    (U, V) ← (D, D)
    for i = l − 1 downto 2
        if d_i = 1 then U ← U + D (add step)
        (U, V) ← (U + V, U) (Fibonacci step)
    end
    return U
end
```

Table 1. Algorithm A.1 Fibonacci-and-add (k, P)

We can see, in Algorithm A.1 using the Addition and Fibonacci step for the adding divisors. In common case, addition of two reduced divisors in weighted coordinates can be represented by the Algorithm A.2 (Lange, 2002c) (Table 2).

Assuming that $Z_{11}=Z_{21}=Z_1$ and $Z_{12}=Z_{22}=Z_2$ for D_1 and D_2, which allows the transformation of algorithm A.2 into algorithm A.3. Apply the approach described by Meloni (2007) to the divisor addition algorithm suggested by Lange (2002c) and weighted divisor representation. Further circumscribe the derivation of expressions in different steps of A.3 (Table 3).

Input:	$\left[U_{11}, U_{10}, V_{11}, V_{10}, Z_1, Z_1^2, Z_2, Z_2^2\right]$, $\left[U_{21}, U_{20}, V_{21}, V_{20}, Z_1, Z_1^2, Z_2, Z_2^2\right]$
Output:	$\left[U_1', U_0', V_1', V_2', Z_1', Z_1'^2, Z_2', Z_2'^2\right] =$ $\left[U_{11}, U_{10}, V_{11}, V_{10}, Z_1, Z_1^2, Z_2, Z_2^2\right] +$ $+\left[U_{21}, U_{20}, V_{21}, V_{20}, Z_1, Z_1^2, Z_2, Z_2^2\right]$, $weight\left(D_1\right) = weight\left(D_2\right) = 2$

#	Expression	Cost
1	Precomputations: $z_{13} = Z_{11} \cdot Z_{12}$, $z_{23} = Z_{21} \cdot Z_{22}$, $z_{12} = z_{11} \cdot z_{13}$, $z_{22} = z_{21} \cdot z_{23}$, $\tilde{U}_{21} = z_{11} \cdot U_{21}$, $\tilde{U}_{20} = z_{11} \cdot U_{20}$, $\tilde{V}_{21} = z_{12} \cdot V_{21}$, $\tilde{V}_{20} = z_{12} \cdot V_{20}$	8M
2	Compute resultant r for u_1 and u_2: $y_1 = U_{11} \cdot z_{21} - \tilde{U}_{21}$, $y_2 = \tilde{U}_{20} - U_{10} \cdot z_{21}$, $y_3 = U_{11} \cdot y_1 + y_2 \cdot z_{11}$, $r = y_2 \cdot y_3 + y_1^2 \cdot U_{10}$, $Z_2' = Z_{11} \cdot Z_{21}$, $\tilde{Z}_2 = Z_{12} \cdot Z_{22}$, $Z_1 = Z_2'^2$, $\tilde{Z}_2 = \tilde{Z}_2 \cdot Z_1$, $\tilde{Z}_2 = \tilde{Z}_2 \cdot r$, $Z_2' = Z_2' \cdot \tilde{Z}_2$, $\tilde{Z}_2 = \tilde{Z}_2^2$, $z_2' = Z_2'^2$	4S, 11M
3	Compute almost inverse $inv = r/u_2 \bmod u_1$, $inv = inv_1 x + inv_0$: $inv_1 = y_1$, $inv_0 = y_3$	
4	Compute $s = \left(v_1 - v_2\right) inv \bmod u_1$, $s = s_1 x + s_0$: $w_0 = V_{10} \cdot z_{22} - \tilde{V}_{20}$, $w_1 = V_{11} \cdot z_{22} - \tilde{V}_{21}$, $w_2 = inv_0 \cdot w_0$, $w_3 = inv_1 \cdot w_1$, $s_0 = w_2 - U_{10} \cdot w_3$, $s_1 = \left(inv_0 + z_{11} \cdot inv_1\right) \cdot \left(w_0 + w_1\right) - w_2 - w_3 \cdot \left(z_{11} + U_{11}\right)$ If $s_1 = 0$ then consider special case	8M
5	Precomputations: $S_1 = s_1^2$, $S_0 = s_0 \cdot Z_1$, $Z_1' = s_1 \cdot Z_1$, $S_1 = Z_1' \cdot S_0$, $S_0 = S_0^2$, $R = r \cdot Z_1'$, $s_0 = s_0 \cdot Z_1'$, $s_1 = s_1 \cdot Z_1'$, $z_1' = Z_1'^2$	3S, 6M
6	Compute $l = su_2$, $l = x^3 + l_2 x^2 + l_1 x + l_0$: $l_0 = s_0 \cdot \tilde{U}_{20}$, $l_2 = s_1 \cdot \tilde{U}_{21}$, $l_1 = \left(s_1 + s_0\right) \cdot \left(\tilde{U}_{21} + \tilde{U}_{20}\right) - l_0 - l_2$, $l_2 = l_2 + S$	3M
7	Compute $u' = \left(s\left(l + h + 2v_1\right) - k\right) u_1^{-1}$, $k = \left(f - v_1 h - v_1^2\right) / u_1$, $u' = x^2 + u_1' x + u_0'$: $V_1' = R \cdot \tilde{V}_{21}$, $U_1' = 2S - s_1 \cdot y_1 - z_2'$, $U_0' = S_0 + y_1 \cdot \left(S_1 \cdot \left(y_1 + \tilde{U}_{21}\right) - 2s_0\right) + y_2 \cdot s_1 + 2\tilde{V}_1' +$ $+\tilde{Z}_2 \cdot \left(y_1 + 2\tilde{U}_{21}\right)$	6M
8	Precomputations: $l_2 = l_2 - U_1'$, $w_0 = l_2 \cdot U_0'$, $w_1 = l_2 \cdot U_1'$	2M
9	Compute $v' \equiv -\left(h + s_1 l + v_2\right) \bmod u'$, $v' = v_1' x + v_0'$: $V_1' = w_1 - z_1' \cdot \left(l_1 + V_1' - U_0'\right)$,	3M

Table 2. Algorithm A.2 Addition reduced divisors

1. Compute resultant r of u_1, u_2:

$y_1 = \frac{U_{11}}{Z_1^2} - \frac{U_{21}}{Z_1^2}$, $y_2 = \frac{U_{20}}{Z_1^2} - \frac{U_{10}}{Z_1^2}$, $y_1 = \frac{U_{11}}{Z_1^2} \cdot \frac{y_1}{Z_1^2} + \frac{y_2}{Z_1^2} = \frac{U_{11}y_1 + y_2 Z_1^2}{Z_1^4}$, $r = \frac{y_2}{Z_1^2} \cdot \frac{y_3}{Z_1^4} + \frac{y_1^2}{Z_1^4} \cdot \frac{U_{10}}{Z_1^2} = \frac{y_2 y_3 + y_1^2 U_{10}}{Z_1^6}$.

2. Compute almost inverse $inv = r/u_2 \bmod u_1$, $inv = inv_1 x + inv_0$:

$inv_1 = \frac{y_1}{Z_1^2}$, $inv_0 = \frac{y_3}{Z_1^4}$.

3. Compute $s' = r \cdot s = (v_1 - v_2) inv \bmod u_1$, $s' = s_1' x + s_0'$:

$w_0 = \frac{V_{10} - V_{20}}{Z_1^3 Z_2}$, $w_1 = \frac{V_{11} - V_{21}}{Z_1^3 Z_2}$, $w_2 = \frac{inv_0}{Z_1^4} \cdot \frac{w_0}{Z_1^3 Z_2} = \frac{inv_0 w_0}{Z_1^7 Z_2}$, $w_2 = \frac{inv_1}{Z_1^2} \cdot \frac{w_1}{Z_1^3 Z_2} = \frac{inv_1 w_1}{Z_1^5 Z_2}$,

$s_1' = \frac{(inv_0 + inv_1 Z_1^2)(w_0 + w_1)}{Z_1^2 Z_2} - \frac{w_2}{Z_1^7 Z_2} - \frac{w_3}{Z_1^5 Z_2}\left(1 + \frac{U_{11}}{Z_1^2}\right)$, $s_0' = \frac{w_2}{Z_1^7 Z_2} - \frac{U_{10}}{Z_1^2} \cdot \frac{w_3}{Z_1^5 Z_2} = \frac{w_2 - U_{10} w_3}{Z_1^7 Z_2}$.

4. Compute $s'' = x + s_0'/s_1'$:

$w_1 = \frac{1}{r \cdot s_1} = \frac{Z_1^6 \cdot Z_1^7 \cdot Z_2}{r \cdot s_1}$, $w_2 = r \cdot w_1 = \frac{r}{Z_1^6} \cdot \frac{Z_1^6 Z_1^7 Z_2}{s_1 \cdot r} = \frac{Z_1^7 Z_2}{s_1'}$ where $s_1' = s_1 \cdot r$,

$w_3 = (s_1')^2 \cdot w_1 = \left(\frac{s_1'}{Z_1^7 Z_2}\right)^2 \cdot \frac{Z_1^{13} Z_2}{r \cdot s_1} = \frac{s_1'}{r \cdot Z_1 \cdot Z_2}$, $w_4 = r \cdot w_2 = \frac{r}{Z_1^6} \cdot \frac{Z_1^7 Z_2}{s_1} = \frac{r \cdot Z_1 \cdot Z_2}{s_1}$,

$s_0'' = s_0' \cdot w_2 = \frac{s_0'}{Z_1^7 Z_2} \frac{Z_1^7 Z_2}{s_1'} = \frac{s_0'}{s_1'} = \frac{s_0}{s_1}$, $w_5 = w_4^2$.

5. Compute $l = s u_2$, $l = x^3 + l_2 x^2 + l_1 x + l_0$:

$l_2 = \frac{U_{21}}{Z_1^2} + \frac{s_0'}{s_1'} = \frac{U_{21} s_1' + s_0' Z_1^2}{Z_1^2 s_1}$, $l_1 = \frac{U_{21}}{Z_1^2} \cdot \frac{s_0'}{s_1'} + \frac{U_{20}}{Z_1^2} = \frac{U_{21} s_0' + U_{20} s_1'}{Z_1^2 s_1'}$, $l_0 = \frac{U_{20}}{Z_1^2} \cdot \frac{s_0'}{s_1'} = \frac{U_{20} s_0'}{Z_1^2 s_1'}$.

6. Compute $u' = (s(l + h + 2v_1) - k) u_1^{-1}$, $k = (f - v_1 h - v_1^2)/u_1$, $u' = x^2 + u_1' x + u_0'$:

$u_0' = (s_0'' - U_{11})(s_0'' - y_1 + h_2 w_4) - U_{10} + l_1 + (h_1 + 2V_{21}) w_4 +$

$+ (2U_{21} + y_1 - f_4) w_5 = \left(\frac{s_0'}{s_1'} - \frac{U_{11}}{Z_1^2}\right)\left(\frac{s_0'}{s_1'} - \frac{y_1}{Z_1^2} + h_2 \frac{r \cdot Z_1 Z_2}{s_1}\right) - \frac{U_{10}}{Z_1^2} + \frac{l_1}{Z_1^2 s_1} +$

$+ \left(h_1 + 2\frac{V_{21}}{Z_1^3 Z_2}\right) \cdot \frac{r \cdot Z_1 \cdot Z_2}{s_1} + \left(s\frac{U_{21}}{Z_1^2} + \frac{y_1}{Z_1^2} - f_4\right) \cdot \frac{(r \cdot Z_1 \cdot Z_2)^2}{s_1^2} = \left| s_2 = s_0 Z_1^2,\ R = r Z_1 Z_2,\ s_3 = s_1 Z_1^2,\ \overline{R} = r Z_1^3 Z_2 \right|$

$= \frac{(s_2 - U_{11} s_1)(s_2 - y_1 s_1 + h_2 \overline{R})}{s_3^2} - \frac{U_{10} s_1 s_3 - l_1 s_3}{s_3^2} + \frac{(h_1 Z_1^3 Z_2 + 2V_{21}) r s_3}{s_3^2} + \frac{(2U_{21} + y_1 - f_4 Z_1^2) R \overline{R}}{s_3^2}$

$= \frac{(s_2 - U_{11} s_1)(s_2 - y_1 s_1 + h_2 \overline{R})}{s_3^2} - \frac{U_{10} s_1 s_3 - U_{21} s_0 s_3 - U_{20} s_1 s_3}{s_3^2} + \frac{(h_1 Z_1^3 Z_2 + 2V_{21}) r s_3}{s_3^2} + \frac{(2U_{21} + y_1 - f_4 Z_1^2) R \overline{R}}{s_3^2}$.

$u_1' = 2s_0'' - y_1 + h_2 w_4 - w_5 = 2\frac{s_0}{s_1} - \frac{y_1}{Z_1^2} + h_2 \frac{r Z_1 Z_2}{s_1} - \left(\frac{r Z_1 Z_2}{s_1}\right)^2 = \frac{2 s_0 Z_1^2 - y_1 s_1}{s_1 Z_1^2} + h_2 \frac{R Z_1^2}{s_1 Z_1^2} - \frac{R^2}{s_1^2} =$

$\frac{2 s_2 - y_1 s_1 + h_2 \overline{R}}{s_3} - \frac{\overline{R}^2}{s_3^2} = \frac{(2 s_2 - y_1 s_1 + h_2 \overline{R}) s_3 - \overline{R}^2}{s_3^2}$.

7. Compute $v' \equiv -(h + s_1 l + v_2) \bmod u'$, $v' = v_1' x + v_0'$: $w_1 = l_2 - U_1'$, $w_2 = u_1' w_2 + u_0' - l_1$,

$v_1' = w_2 w_3 - v_{21} - h_1 + h_2 u_1' = (u_1' w_1 + u_0' - l_1) w_3 - v_{21} - h_1 + h_2 u_1' = \left(\frac{U_1'}{s_3^2}\left(\frac{l_2}{s_3} - \frac{U_1'}{s_3^2}\right) - \frac{U_0'}{s_3^2} - \frac{l_1}{s_3}\right)\frac{s_1}{R} - \frac{V_{21}}{Z_1^3 Z_2} - h_1 + h_2 \frac{U_1'}{s_3^2} =$

$\frac{U_1'(l_2 s_3 - U_1') - U_0' s_3^2 - l_1 s_3^3}{s_3^3 Z_1^2 R} - \frac{V_{21} s_3^3 r}{s_3^3 Z_1^2 R} - h_1 + \frac{h_2 U_1'}{s_3^2} = \left|\overline{\overline{R}} = s_3 R\right| = \frac{U_1'\left(l_2 s_3 - U_1' + h_2 \overline{\overline{R}}\right) + s_3^2\left(U_0' - l_1 s_3 - V_{21} s_3 r - h_1 \overline{\overline{R}}\right)}{s_3^3 \overline{R}}$,

$v_0' = w_2 w_3 - v_{20} - h_0 - h_2 u_0' = \frac{U_0'}{s_3^3 \overline{R}}\left(l_2 s_3 - U_1' + h_2 s_3 \overline{R}\right) - \frac{s_3^2}{s_3^3 \overline{R}}\left(V_{20} s_3 r + h_0 \overline{\overline{R}} + l_0 s_3\right)$.

8. Compute Z_1' and Z_2': $Z_1' = s_3$, $Z_2' = \overline{R}$.

Table 3. Derivation of expressions in different steps of A.3

Specify the modifications carried out in A.2, which allowed reducing quantity of field operations.

Step A.2.1. This step is to be omitted due to existence of the same denominator of all coordinates, which allows saving 8 multiplications in $\mathbf{GF}(p)$.

Step A.2.2. While computation of y_1 and y_2, reduction to common denominator of coordinates U_{1j} and U_{2j} is not required, saves 2 multiplications in $\mathbf{GF}(p)$.

Moreover, the calculation of resulting Z_1' and Z_2' coordinates is also significantly simplified, that helps to save 5 multiplications and 3 squaring operations in $\mathbf{GF}(p)$.

Step A.2.4. While computation of w_1 and w_2, reduction to common denominator of coordinates V_{1j} and V_{2j} is also not required, saves 2 multiplications in $\mathbf{GF}(p)$.

Steps A.2.5 and A.2.6 The total number of multiplications remains unchanged, however the number of squaring operations in $\mathbf{GF}(p)$ decreases by 3, due to the interchange of calculations of coefficients l_0, l_1 and l_2 of polynomial l on the step A.2.6 (Table 4).

Input:	$\left[U_{11}, U_{10}, V_{11}, V_{10}, Z_1, Z_1^2, Z_2, Z_2^2\right]$, $\left[U_{21}, U_{20}, V_{21}, V_{20}, Z_1, Z_1^2, Z_2, Z_2^2\right]$	
Output:	$\left[U_1', U_0', V_1', V_2', Z_1', Z_1'^2, Z_2', Z_2'^2\right] =$ $\left[U_{11}, U_{10}, V_{11}, V_{10}, Z_1, Z_1^2, Z_2, Z_2^2\right] +$ $+\left[U_{21}, U_{20}, V_{21}, V_{20}, Z_1, Z_1^2, Z_2, Z_2^2\right]$, $weight(D_1) = weight(D_2) = 2$	
#	**Expression**	**Cost**
1	Compute resultant r of u_1, u_2: $y_1 = U_{11} - U_{21}$, $y_2 = U_{20} - U_{10}$, $y_3 = U_{11} \cdot y_1 + y_2 \cdot Z_1^2$, $r = y_2 \cdot y_3 + y_1^2 \cdot U_{10}$	1S, 4M
2	Compute almost inverse $inv = r/u_2 \bmod u_1$, $inv = inv_1 x + inv_0$: $inv_1 = y_1$, $inv_0 = y_3$	
3	Compute $s = (v_1 - v_2) inv \bmod u_1$, $s = s_1 x + s_0$: $w_0 = V_{10} - V_{20}$, $w_1 = V_{11} - V_{21}$, $w_2 = inv_0 \cdot w_0$, $w_3 = inv_1 \cdot w_1$, $s_0 = w_2 - U_{10} \cdot w_3$ $s_1 = (inv_0 + inv_1 \cdot Z_1^2) \cdot (w_1 + w_0) - w_2 - w_3 \cdot (Z_1^2 + U_{11})$, If $s_1 = 0$ then consider special case	6M
4	Precomputations: $R = r \cdot Z_1 \cdot Z_2$, $s_2 = s_0 \cdot Z_1^2$, $s_3 = s_1 \cdot Z_1^2$, $\tilde{R} = R \cdot Z_1^2$, $w_3 = s_1 \cdot y_1$, $w_5 = w_3 + s_1 \cdot U_{21} (= s_1 \cdot U_{11})$	7M
5	Compute $l = su_2$, $l = x^3 + l_2 x^2 + l_1 x + l_0$: $l_0 = s_0 \cdot U_{20}$, $l_2 = s_1 U_{21}$, $l_1 = (s_1 + s_0) \cdot (U_{21} + U_{20}) - l_0 - l_2$, $l_2 = l_2 + s_2$	2M
6	Compute $u' = (s(l + h + 2v_1) - k) u_1^{-1}$, $k = (f - v_1 h - v_1^2) / u_1$, $u' = x^2 + u_1' x + u_0'$: $U_1' = s_3 \cdot (2s_2 - w_3 + h_2 R) - \tilde{R}^2$ $U_0' = s_2^2 + w_3 \cdot (w_5 - 2s_2) + s_3 \cdot (y_2 \cdot s_1 + 2r \cdot V_{21} + h_1 \tilde{R}) +$ $+\tilde{R} \cdot \left[h_2 (s_2 - w_5) + R \cdot (U_{11} + U_{21} - f_4 \cdot Z_1^2)\right]$	2S, 8M
7	Compute weights: $Z_1' = s_3$, $Z_2' = \tilde{R}$, $Z_1'^2 = s_3^2$, $Z_2'^2 = \tilde{R}^2$	1S
8	Compute $v' \equiv -(h + s_1 l + v_2) \bmod u'$, $v' = v_1' x + v_0'$: $V_1' = U_1' \cdot ((l_2 + h_2 R) \cdot s_3 - U_1') + s_3^2 \cdot (U_0' - s_3 \cdot (h_1 R + r V_{21} + l_1))$, $V_0' = U_0' \cdot ((l_2 + h_2 R) s_3 - U_1') - s_3^2 \cdot s_3 \cdot (l_0 + h_0 R + r \cdot V_{20})$	8M
		4S, 35M
9	Adjust: $Z_{1c} = s_1^2 \cdot Z_1^2$, $Z_{2c} = s_3^3 \cdot r$, $U_{20} = U_{20} \cdot Z_{1c}$, $U_{21} = U_{21} \cdot Z_{1c}$, $V_{20} = V_{20} \cdot Z_{2c}$, $V_{21} = V_{21} \cdot Z_{2c}$	1S, 6M
		5S, 41M

Table 4. Algorithm A.3. Co-Z reduced divisors addition

Step A.2.6. unlike algorithm A.2, the A.3 offers considering multiplier s_3, present in each coefficient l_0, l_1 and l_2 of polynomial l, when using coefficients l_i, $i=\overline{0,2}$ on the steps A.2.7 and A.2.9. This allows us factor out s_3, thus saving 3 multiplications in $\mathbf{GF}(p)$ (steps A.4.6-A.4.8).

Consider next the application of proposed algorithm for the mixed divisor addition $D_1=\left[U_{11},U_{10},V_{11},V_{10},Z_1,Z_1^2,Z_2,Z_2^2\right]$ and $D_2=\left[U_{21},U_{20},V_{21},V_{20},1,1,1,1\right]$ (mixed representation). Therefore it is necessary to reduce devisor D_2 to common Z-coordinate, i.e. $\left[U_{21}\cdot Z_1^2,U_{20}\cdot Z_1^2,V_{21}\cdot Z_1^3\cdot Z_2,V_{20}\cdot Z_1^3\cdot Z_2,Z_1,Z_1^2,Z_2,Z_2^2\right]$, that requires 5 multiplications in $\mathbf{GF}(p)$. Hereinafter the provided algorithm A.4 should be used for addition of (prior formed) divisors with the same Z-coordinate.

6. Results

It is to be considered that after computing $D_3=D_1+D_2$ one of the items, for example D_2, should be transformed so that it has the same Z-coordinate as divisor D_3. For this purpose, at the step A.3.9, values Z_{1c} and Z_{2c}, where $Z_1'^2=s_3^2=Z_1^2\cdot Z_{1c}$ and $Z_1'^3Z_2'=s_3^3\cdot\tilde{R}=Z_1^3Z_2\cdot Z_{2c}$, i.e. $Z_{1c}=s_1^2\cdot Z_1^2$ and $Z_{2c}=s_3^3\cdot r$, this requires 2 additional multiplications and 1 squaring in $\mathbf{GF}(p)$. In other words, reduction of a divisor to unified Z-coordinates takes 6 additional multiplications and 1 squaring. Ultimately, for the divisor addition step in A.1 46M+5S (M - multiplication, S - squaring) field operations are required. For the Fibonacci step 41M+5S field operations are required. If we reduce the obtained complexity estimations to the parameters of the curve (Lange, 2002c), we obtain that for the divisor addition step 40M+5S are required and for the Fibonacci step 45M+5S operations are required. In accordance to the computational complexity estimation, the approach described in this paper, is not effective, due to the complexity of mixed addition is 36M+5S (Lange, 2002c). However, the alternative approach to the divisor addition for the scalar multiplication implementation is proposed. Let us draw a computational complexity comparison between scalar multiplication algorithms described in (Kovtun and Zbitnev, 2004) and those suggested in this paper, based on idea (Meloni, 2007).

6.1 Comparison with other method

The results of known and proposed algorithms comparison are set out in Table 5.

The algorithm complexity represented in field operations (Table 6).

Assume that S=0,8M and scalar multiplier is an 80-bit integer and refer to the estimations (Kovtun and Zbitnev, 2004; Meloni, 2007) for the estimation of complexity of scalar multiplication algorithms. The results of comparison are set out in Table 7.

Computational complexity of Fibonacci-and-add scalar multiplication algorithm is by 23% greater than Binary left-to-right algorithm and by 12,5% greater than Window Fibonacci-and-add.

In other case computational complexity of Fibonacci-and-add scalar multiplication algorithm in weighted coordinates is by 23% greater than Binary left-to-right algorithm in mixed weighted coordinates and by 14,2% greater than Window Fibonacci-and-add.

Weighted coordinates with Co-Z approach are more effective than ordinary projective coordinates with Co-Z approach.

Alg. #	Curve description
1	$h(x)=0$ (Harley, 2000)
2	$h_2=1$ (Lange, 2002a)
3	$h(x)=0$ (Matsuo et al., 2001)
4	$h(x)=0$, $f_4=0$ (Miyamoto et al., 2002)
5	$h(x)=0$ (Takahashi, 2002)
6	$f_4=0$, $h_2\neq 0$ (Lange, 2002a)
7	$\deg(h)=2$, $h_i\in \mathbf{F}_2$ (Lange, 2002b)
8	$\deg(h)=2$, $h_i\in \mathbf{F}_2$ (Kovtun and Zbitnev, 2004)
9	$h(x)=0$, $f_4=0$ (Kovtun, 2006)
10	$h(x)=0$, $f_4=0$ (Lange, 2002c)
11	$\deg(h)=2$, $h_i\in \mathbf{F}_2$ (Kovtun, 2010)
12	$h(x)=0$, $f_4\neq 0$ [proposed]
13	$h(x)=0$, $f_4=0$ [proposed]

Table 5. Algorithms and curve parameters

#	Addition					Doubling				
	General			Mixed		General			Mixed	
	()$^{-1}$	^2	*	^2	*	()$^{-1}$	^2	*	^2	*
Affine coordinates										
1	2		27			2		30		
2	2	3	24			2	6	26		
3	2		25			2		27		
4	1		26			1		27		
5	1		25			1		29		
6	1	3	22			1	5	22		
Projective coordinates $[U_1, U_0, V_1, V_0, Z]$										
7		4	47	3	40		6	40	5	25
8		4	46	4	39		6	39	5	25
9		4	46	4	39		6	35	5	24
Weighted coordinates $[U_1, U_0, V_1, V_0, Z_1, Z_2, Z_1^2, Z_2^2]$										
10		7	47	5	36		7	34	5	21
Co-Z projective coordinates $[U_1, U_0, V_1, V_0, Z]$										
11		4	46				4	42		
Co-Z weighted coordinates $[U_1, U_0, V_1, V_0, Z_1, Z_2, Z_1^2, Z_2^2]$										
12		5	46				5	41		
13		5	45				5	40		

Table 6. Computational complexity of group law in Jacobean of genus 2 HEC over $\mathbf{GF}(p)$

#	Scalar multiplication algorithm	Cost, M
Addition in mixed projective coordinates (Kovtun and Zbitnev, 2004)		
1	Binary (left-to-right)	5192
2	NAF	4629
3	w -NAF, $w = 4$	4349
Addition with Co-Z method in projective coordinates (Kovtun, 2010)		
7	Fibonacci-and-add	6773
8	Window Fibonacci-and-add	5970
Addition in mixed weighted coordinates (Lange, 2002c)		
4	Binary (left-to-right)	5104
5	NAF	4570
6	w -NAF, $w = 4$	4307
Addition with suggested method, alg. #13 from table 1		
7	Fibonacci-and-add	6629
8	Window Fibonacci-and-add	5829

Table 7. Computational Complexity Of Scalar Multiplication Algorithms

6.2 Another aspect of using the proposed approach

Yet another aspect of using a biometric authentication on based Co-Z approach to divisor scalar multiplication in Jacobian of genus 2 hyperelliptic curves over fields with odd characteristic in weighted coordinates is using it in the fight against Cyber terrorism. Using this approach and biometric authentication will significantly reduce financial losses of enterprises.

Primary solutions of the given problem are confirmed also with the data presented on figure 1 on which (Kavun, 2007) finance indexations of the put damage for some countries are shown. Apparently from the presented statistics, the state infrastructure is more developed; the larger it receives damage from cyber criminality.

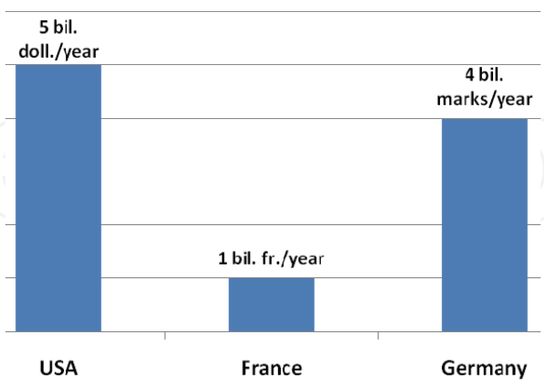


Fig. 1. Damage from cyber criminalities

During an epoch of the world economic crisis and modern transformations of economy of different countries the aspect of economic security becomes even more urgent . For example, the tendency of increase in crime in sphere of cyber terrorism, having an economic (money) basis is shown in figure 2.

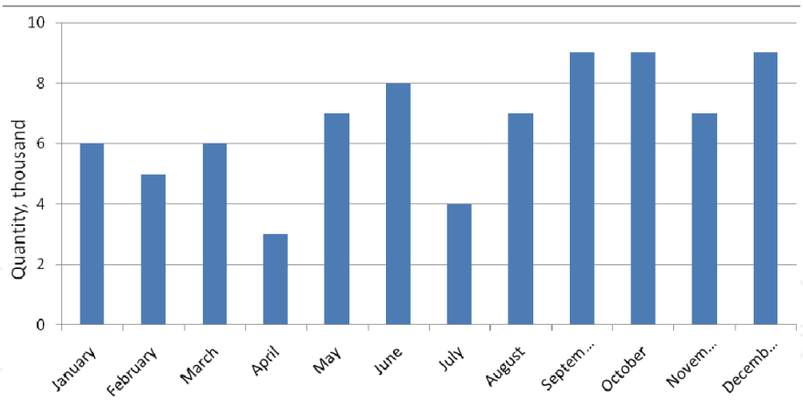


Fig. 2. Statistics of cyber criminality rate

From the presented data which has been average for some years (Kavun, 2007), it can be seen, that on the average the amount of incidents in the sphere of cyber criminality over a year increased by 50 %. It testifies to negative tendencies of cyber criminality development in the world.
Outcomes of the conducted analysis are presented in figure 3 from which it appears, that main sources of illegal operations are the USA and China which cover 50 % of all threats (Kavun, 2009).

7. Conclusion

In the paper, a new algorithm of weight 2 divisor addition with identical (shared) Z-coordinates (by the Co-Z approach) has been proposed, which requires more $\mathbf{GF}(p)$ operations then algorithm (Kovtun and Zbitnev, 2004), however it allows a decrease in computational complexity of Fibonacci-and-add scalar multiplication algorithm while approaching to the Binary left-to-right algorithm.
Biometrics are not secrets and are not revocable (Schneier,1999) while revocability and secrecy have been critical requirements of conventional cryptosystem design, one then wonders whether it is possible to design a secure authentication system from the system components which in themselves are neither secrets nor revocable—for example, whether the methods of ensuring liveness of biometric identifiers and challenge–response schemes (Maltoni et al., 2003) obviate fraudulent insertion of "stolen" biometric identifiers. Is it possible to nontrivially combine knowledge and biometric identifiers to arrive at key generation/release mechanisms where biometric identifiers are necessary but not sufficient for cryptographic key generation/release? Is it possible to require multiple biometrics to make it increasingly difficult for the attacker to fraudulently insert multiple biometrics into the system? Is it possible to make it unnecessary to revoke/update the cryptographic key in the event of a "stolen biometric"? Exploring challenges in designing such systems is a promising (yet neglected) avenue of research. When cryptobiometric systems eventually come into practical existence, there is a danger that biometric components may be used as an irrefutable proof of existence of a particular subject at a particular time and place. Mere incorporation of biometrics into a system does not in itself constitute a proof of identity. We need to understand how these foolproof guarantees can be theoretically proved in a deployed cryptosystem and how to institute due processes that will provide both

technological and sociological freedom to challenge the premises on which nonrepudiability is ascertained.

Genus 3 curves might save about a third in key lengths and so 180-bit ECC (which is beyond the usual range, as given in standards) is equivalent to 60-bit HCC, which can be implemented on a fast 64-bit computer. In practice, 160-bit ECC is typically used and so there is even room for added security in case of computational speed-ups in attacks. Stein, in particular, has pointed out how the use of "real" forms of hyperelliptic curves (i.e., with infrastructure) allows considerable speed-ups in implementation in some cases.

On the other hand, Gaudry (2000), showed that hyperelliptic curves of genus bigger than or equal to 5 and possibly 4 are less secure than hyperelliptic curves of genus g<4 (or 5) (ICCIT Biometrics Research Group, 2005). That means that the key-per-bit-strength of hyperelliptic curves of genus 2 and 3 is the same as for elliptic curves, and thus far better than conventional systems based on discrete logs or integer factoring. In fact, genus 2 is particularly interesting because the arithmetic appears to be only minimally slower than elliptic curve arithmetic and the bit size of the underlying finite field is half as big as for elliptic curves having the same security level. To our knowledge nobody has performed a down-to-earth implementation of genus 2 hyperelliptic curves.

We are considering hardware implementations of hyperelliptic curves of genus 2 and 3 . Among the currently available hyperelliptic curves there are, for instance, Koblitz curves and curves constructed by a complex multiplication method (CM-method). Both are natural extension of ideas from elliptic curves.

8. References

Basiri, A.; Enge, A. & Faugère, J. C. & Gurel, N. (2004). The arithmetic of Jacobian groups of superelliptic cubics, *Mathematics of Computation.* Retrieved 22.01.2011 on: http://www.ams.org/journals/mcom/2005-74-249/S0025-5718-04-01699-0/S0025-5718-04-01699-0.pdf

Brier, E.; Joye, M. (2002). Weirstrass elliptic curves ans side-channel attacks, *Proceedings of the International Workshop: Practice and Theory in Public Key Cryptosystems*, PKC 2002, LNCS 2274, Springer-Verlag, pp.335-345

Chudnovsky, D. V.; Chudnovsky, G. V. (1986). Sequence of number generated by addition in formal group and new primality and factorization test, *Advanced in Applied Math*, 8, pp.385–434.

Clavier, C.; Joye, M. (2001). Universal exponentiation algorithm – A first step towards provable SPA-resistance cryptosystems, *Proceedings of the International Workshop: Cryptographic Hardware and Embedded Systems*, CHES 2001, LNCS 2162, pp.300-308.

Cohen, H.; Miyaji, A. & Ono, T. (1998). Efficient elliptic curve exponentiation using mixed coordinates, *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*. CRYPTO'98. LNCS 1514. Berlin: Springer–Verlag, pp. 51–65.

Coron, J. (1999). Resistance against differential power analysis for elliptic curve cryptosystems, *Proceedings of the International Workshop: Cryptographic Hardware and Embedded Systems*, CHES 1999, LNCS 1717, pp.292-302.

Gaudry, P. (2000). An algorithm for solving the discrete log problem on hyperelliptic curves. In Bart Preneel, editor, *Advances in Cryptology - EUROCRYPT 2000*, volume 1807 of *Lecture Notes in Computer Science*, pages 19-34, Berlin, Springer-Verlag.

Goundar, R. R. (2008). *Addition chains in application to elliptic curve cryptosystems*: PhD thesis, Kochi University, Japan.

Hao, F.; Anderson, R.; Daugman, J. (2005). Combining cryptography with biometrics effectively, *Technical Report*, No. 640, University of Cambridge.

Izu, T.; Takagi, T. (2002). A fast parallel elliptic curve multiplication resistant against side channel attacks,*Technical Report CORR*, CORR 2002-03, University of Waterloo, 2002.

Jain, A. K.; Bolle, R. & Pankanti, S. (1999). *Biometrics: Personal Identification in Networked Society*. Norwell, MA: Kluwer.

Klein, D. V. (1990). "Foiling the cracker: a survey of, and improvements to, password security," in *Proc. 2nd USENIX Workshop Security*, 1990, pp. 5–14.

Koblitz, N. (1987). Elliptic curve cryptosystems. *Mathematics of Computation*, 48(177), pp. 203–209.

Koblitz, N. (1989). Hyperelliptic cryptosystems. *Journal of cryptology*, 1, pp.139-150.

Kovtun, V. Yu.; Zbitnev, S. I. (2004). Arithmetic operations in Jacobian of Genus 2 hyperelliptic curves in projective coordinates with reduced complexity, *East-Europian magazin of advanced manufacturing services*. 1 (13). pp. 14–22.

Lange, T. (2001). *Efficient arithmetic on hyperelliptic curves*: PhD thesis: Mathematics and Informatics. University of Essen: Institute for experimental mathematics. Germany: Essen.

Lange, T. (2002a). Efficient arithmetic on genius 2 hyperelliptic curves over finite fields via explicit formulae, *Cryptology ePrint Archive*. Report 2002/121. Available

Lange, T. (2002b). Inversion–free arithmetic on genius 2 hyperelliptic curves, *Cryptology ePrint Archive*. Report 2002/147.

Lange, T. (2003). Formulae for arithmetic on genius 2 hyperelliptic curves. September 2003. Retrieved 22.01.2011 on:
http://www.ruhr–uni–bochum.de/itsc/tanja/preprints/expl_sub.pdf.

Lange, T. (2002c). Weighted coordinates on genius 2 hyperelliptic curves, *Cryptology ePrint Archive*. Report 2002/153.

Maltoni, D.; Maio, D.; Jain, A. K. & Prabhakar, S. (2003). *Handbook of Fingerprint Recognition*. New York: Springer-Verlag.

Matsuo, K.; Chao J. & Tsujii S. (2001). Fast genius two hyperelliptic curve cryptosystem, *Technical report IEICE*. ISEC2001–31.

Miller, I. V. S. (1985). Use of elliptic curves in cryptography. In H. C. Williams, editor, *Advances in Cryptology - CRYPTO'85*, volume 218 of LNCS, Springer, pp. 417–426.

Menezes, A.; Wu, Y. H. & Zuccherato, R. (1998). An elementary introduction to hyperelliptic curves, In: Koblitz N. ed., *Algebraic aspects of cryptography*. Berlin, Heidelberg, New York: Springer–Verlag, pp. 28–63.

Meloni, N. (2007). New point addition formul. for ECC applications. In C. Carlet and B. Sunar, editors, *Arithmetic of Finite Fields (WAIFI 2007*), LNCS 4547, Springer, pp. 189–201.

Miyamoto, Y.; Doi H.; Matsuo K.; Chao J. & Tsujii S. (2002). A fast addition algorithm of genius two hyperelliptic curve, *Symposium on cryptography and information security*. SCIS'2002. Japan: IEICE, pp.497–502.

NIST (2001). Advanced encryption standard (AES), Federal information processing standards publication 197. Retrieved 22.01.2011 on:

http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

Okeya, K.; Sakurai, K. (2000). Power analysis breaks elliptic curve cryptosystems even secure against timing attack, *Proceedings of the International Conference on the Theory and Applications of Cryptology and Information Security: Advances in Cryptology*, INDOCRYPT 2000, LNCS 1977, Springer-Verlag, pp.178-190.

Schneier, B. (1999). Biometrics: uses and abuses, *Commun. ACM,* 42(8), p. 136.

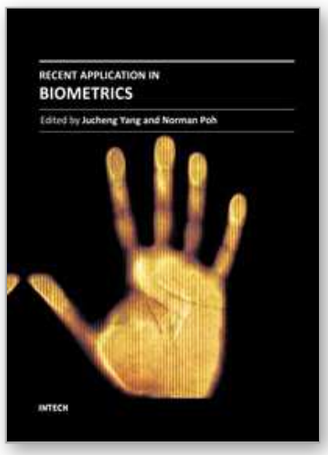
Sugizaki, H.; Matsuo, K.; Chao, J. & Tsujii, S. (2002). An extension of Harley addition algorithm for hyperelliptic curves over finite fields of characteristic two, *Technical report IEICE*. ISEC2002–09.

Stallings, W. (2003). *Cryptography and Network Security: Principles and Practices*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall.

Takahashi, M. (2002). Improving Harley algorithms for jacobians of genius 2 hyperelliptic curves, *Symposium on cryptography and information security*. SCIS'2002. Japan: IEICE, pp.155–160.

Wollinger, T. (2004). *Software and hardware implementation of hyperelliptic curve cryptosystems*: PhD dissertation: Electronics and informatics. Worchester Polytechnic Institute,Germany: Bochum.

Wayman, J. L. (2001). "Fundamentals of biometric authentication technologies," *Int. J. Image Graph.*, 1(1), pp. 93–113.



Recent Application in Biometrics

Edited by Dr. Jucheng Yang

ISBN 978-953-307-488-7
Hard cover, 302 pages
Publisher InTech
Published online 27, July, 2011
Published in print edition July, 2011

In the recent years, a number of recognition and authentication systems based on biometric measurements have been proposed. Algorithms and sensors have been developed to acquire and process many different biometric traits. Moreover, the biometric technology is being used in novel ways, with potential commercial and practical implications to our daily activities. The key objective of the book is to provide a collection of comprehensive references on some recent theoretical development as well as novel applications in biometrics. The topics covered in this book reflect well both aspects of development. They include biometric sample quality, privacy preserving and cancellable biometrics, contactless biometrics, novel and unconventional biometrics, and the technical challenges in implementing the technology in portable devices. The book consists of 15 chapters. It is divided into four sections, namely, biometric applications on mobile platforms, cancelable biometrics, biometric encryption, and other applications. The book was reviewed by editors Dr. Jucheng Yang and Dr. Norman Poh. We deeply appreciate the efforts of our guest editors: Dr. Girija Chetty, Dr. Loris Nanni, Dr. Jianjiang Feng, Dr. Dongsun Park and Dr. Sook Yoon, as well as a number of anonymous reviewers.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen