I. J. Computer Network and Information Security, 2016, 8, 67-72
Published Online August 2016 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2016.08.08

A Study of Hyperelliptic Curves in Cryptography

Reza Alimoradi
University of Qom, Faculty of Science, Department of Mathematics and Computer Science, Qom, Iran
E-mail: r.alimoradi@qom.ac.ir, alimoradi.r@gmail.com

***Abstract*—**Elliptic curves are some specific type of curves known as hyper elliptic curves. Compared to the integer factorization problem(IFP) based systems, using elliptic curve based cryptography will significantly decrease key size of the encryption. Therefore, application of this type of cryptography in systems that need high security and smaller key size has found great attention. Hyperelliptic curves help to make key length shorter. Many investigations are done with regard to improving computations, hardware and software implementation of these curves, their security and resistance against attacks. This paper studies and analyzes researches done about security and efficiency of hyperelliptic curves.

***Index Terms*—**Cryptography, Hyperelliptic curves, Discrete logarithm problem, Pairing, Scalar multiplication.

I. Introduction

Hyper elliptic curves are being used in many important research fields like pseudo random numbers generators [21], coding theory [4, 10, 20], number theory algorithms [1, 18, 19] and cryptography [22, 23, 24, 25, 26] .

In 1989, Koblitz suggested using hyper elliptic curves instead of elliptic curves in order to design cryptography systems based on discrete logarithm problem (DLP). Hyperelliptic curves are expanded forms of elliptic curves. In other words, an elliptic curve is a hyperelliptic curve of genus 1. Having a short key size is the main advantage of hyper elliptic curves. It means that a hyper elliptic curve needs a smaller finite field to reach some level of security compared with an elliptic curve.

Further, group order of a hyperelliptic curve of genus g on a finite field with q element(s) is q^g . So, in order to make a group of the order 2^{160} by using an elliptic curve, one needs a finite field with 2^{160} elements. Whereas to make the same group using hyperelliptic curves of genus 2, only one finite field with 2^{80} elements will be needed. Likewise, for hyper elliptic curves of genus 3 and 4 a finite field with respectively 2^{53} and 2^{51} elements will be required [25].

Of course, regarding researches done on hyper elliptic curves of genus 4, 5, 6, … these curves have a lower security level [14]. Contrary to elliptic curves which do not allow using index calculus algorithm for solving discrete logarithm problem, on hyperelliptic curves this attack is possible; which is considered a major shortcoming for them. Regarding the complexity of computations on hyperelliptic curves, it is very important to find appropriate hyperelliptic curves and improving their computations in order to make cryptography systems based on these curves applicable. Today, hyperelliptic curves of genus 2 and 3 can be efficiently obtained so that the resulted group will have an almost prime order.

Table 1. List of Abbreviations.

Wireless Sensor Networks	WSN
Elliptic Curve Cryptography	ECC
Hyperelliptic Curve Cryptography	HECC
Genus of hyperelliptic curve	g
considered the n^{th} expansion of F_q	F_{q^n}
Jacobin of a hyperelliptic curve C defined on F_{q^n} .	$J(F_{q^n})$
Multiplicative group of F_{q^n}	$F_{q^n}^*$
Multiplication/Inverse	$\left(\frac{M}{I}\right)$
Pairing function	η_T

II. Mathematical Preliminaries

Some mathematical preliminaries are explicated below.

Definition 1: Suppose $\overline{K}$ is a closure of the field K. A hyper elliptic curve of genus $g\,(g \geq 1)$ on K is $C: y^2 + h(x)y = f(x) \in K[x,y]$ (1). So that, $h(x) \in K[x]$ is a polynomial of the maximum degree g, and $f(x) \in K[x]$ is a singular polynomial of degree $2g+1$, and this equation and its partial differential equations that are $2y + h(x) = \circ$ and $h'(x)y - f'(x) = \circ$ don't have a common solution on $\overline{K}$. We call $(x,y) \in \overline{K} \times \overline{K}$ a singular point on curve C if it is answer to the three above equations at the same time.

Definition 2: Suppose L is an expanded field of the K field. The set $C(L)$ includes all L-rational points on C. It is consisted of points $P = (u,v) \in L \times L$ so that they hold true in relation (1) with a point in infinity that is shown

by ∞. The set $C\left(\overline{K}\right)$ is briefly called *C*.

Example 3: A genus 2-hyperelliptic curve and $h(x)=\circ$ on real numbers' field is introduced.

$$C: y^2 = x^5 - 5x^3 + 4x$$
$$= x(x-1)(x+1)(x-2)(x+2).$$

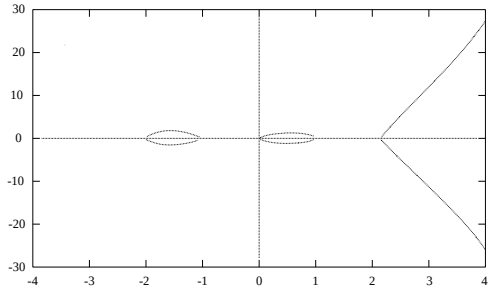

Fig.1. Hyperelliptic curve C on R

Definition 4: Suppose $P=(x,y)$ is a finite point on curve C. The opposite point of *P* is $\tilde{P}=(x,-y-h(x))$ so that $\tilde{P}$ is on *C*. Also, the opposite point of ∞ is considered ∞ $(\tilde{\infty}=\infty)$. If a finite field *P* holds true in $P=\tilde{P}$, it is called a special point.

Example 5: Suppose curve $C: y^2+xy=x^5+5x^4+6x^2+x+3$ is defined on finite field Z_7. Thus, $g=2$, $f(x)=x^5+5x^4+6x^2+x+3$ and $h(x)=x$. It is obvious that *C* does not have a singular point, So, *C* is a hyper elliptic curve:

$C(\mathbf{Z}_7)=\{\infty, (4,4)\ (1,5)\ (2,2)\ (2,3)\ (5,3)\ (5,6)\ (6,4)\}$.

Point $(6,4)$ is a special point.

Here is a method of computing number of group members resulted from a hyperelliptic curve. Consider *J* the Jacobin of a hyperelliptic curve *C* defined on F_q. Also, F_{q^n} is considered the nth expansion of F_q and N_n is order of the abelian group ($J\left(F_{q^n}\right)$).

Result 6: Suppose *C* is a hyper elliptic curve of genus *g* defined on F_{q^n} and $N_n=\left|J\left(F_{q^n}\right)\right|$. Then this relation $\left(q^{\frac{n}{2}}-1\right)^{2g} \le N_n \le \left(q^{\frac{n}{2}}+1\right)^{2g}$ holds and thus $N_n \approx q^{ng}$.

III. Attacks

There are some attacks aimed at hyperelliptic curves. 10 years after Koblitz introduced hyperelliptic curves to cryptography (Diffie-Hellman key exchange) the best attacks of discrete logarithm problem for these curves were square root algorithms. Square roots algorithms are public key algorithms used for solving discrete logarithm problem in every group including shank's baby- step giant- step, Pollard and Pohlig-Hellman algorithms. Running time for the first and the second algorithms is radical of the group size and running time for the third algorithm is square root of the largest prime factor of the group order. Because in cryptography applications the group's order is prime or almost prime, so, all these algorithms have square root time. The first index calculus algorithm to solve discrete logarithm problem on the Jacobin of a hyperelliptic curve was introduced in [2]. Also, the guidance account attack offered in [14] was the first instance of a public attack to the discrete logarithm problem defined on the Jacobin of low-genus hyperelliptic curves which had a shorter running time than the group order's square root. These definitions and theorems are needed.

Definition 7: Suppose D_1 and D_2 are two elements of J_q So that $D_2 \in < D_1 >$. The discrete logarithm problem on the Jacobin of a hyperelliptic curve for the pair (D_1, D_2) is computing the smallest number $\lambda \in N$ so that $D_2 = \lambda D_1$.

Theorem 8: Take *C* as a hyper elliptic curve of genus *g* on the finite field F_q. If $\ln q \le (2g+1)^{1-\varepsilon}$ then $c \le 2.181$ exists so that the discrete logarithm problem in $J_C(F_q)$, in the time duration $L_{q^{2g+1}}\left(\frac{1}{2}, c\right)$ is computable [2].

Theorem 9: Take *C* as a hyperelliptic curve of genus *g*, defined on the finite field F_q. If $q > g!$ then the discrete logarithm problem in $J_C(F_q)$ in the time duration $O\left(g^3 q^{2+\varepsilon}\right)$ is computable [14].

Of course, many improvements of the index calculus algorithm on hyper elliptic curves have been offered some of them have results as follows.

Take size of factor base as $|P_B| = O\left(g^2 q^{(g/(g+1))+\varepsilon}\right)$ and *C* as a hyperelliptic curve of genus g on the finite field F_q. If $q > g!$ then the discrete logarithm problem in $J_C(F_q)$ in the time duration $O\left(g^5 q^{2-\frac{2}{g+1}+\varepsilon}\right)$ is solvable. Now, if the factor base size is $O\left(g^5 q^{\left(\left(g-\frac{1}{2}\right)/\left(g+\frac{1}{2}\right)\right)+\varepsilon}\right)$ then this time is $O\left(g^5 q^{2-\frac{4}{2g+1}+\varepsilon}\right)$[33]. If $|P_B| = O\left(g^2 q^{((g-1)/g)+\varepsilon}\right)$ then this time is $O\left(g^5 q^{2-\frac{2}{g}+\varepsilon}\right)$[16].

Theorem 10: For $\frac{g}{Ln(q)} > t$ discrete logarithm problem in $J_C(F_q)$, there is a hyper elliptic curve of genus g on F_q with maximum complexity of $L_{q^g}\left(\frac{1}{2}, \sqrt{2}\left(\left(1+\frac{1}{2t}\right)^{\frac{1}{2}} + \left(\frac{1}{2t}\right)^{\frac{1}{2}}\right)\right)$[11, 12].

As mentioned before, there is an index calculus attack with complexity $O\left(g^5 q^{2-\frac{2}{g}+\varepsilon}\right)$ and for *g=3, 4* security level decreases.

For a genus 4-hyperelliptic curve defined on F_2, the discrete logarithm problem in $J_C(F_q)$ with the complexity $O\left(\left|J_C(F_q)\right|^{0.375}\right)=O\left(q^{\frac{3}{2}+\varepsilon}\right)$ will be solved. So, the discrete logarithm problem of this case is weaker than the one generally mentioned. For *g=3*, this complexity is $O\left(\left|J_C(F_q)\right|^{0.44}\right)=O\left(q^{\frac{4}{3}+\varepsilon}\right)$.

It shows that almost all genus 3-hyper elliptic curves are weaker than the elliptic curves. Therefore, it is possible to say that for hyper elliptic curves of genus $g \geq 4$ index calculus attack can be done.

As a result, they are not appropriate to be used in public key cryptography. In case of using them, the group size must be well increased. Also, among hyper elliptic curves of other genus i.e. $g=1,2,3$, for *g=3*, the group size must be selected in a way to prevent index calculus attacks offered in [33].

One other current attack for hyperelliptic curves is the descent Weil attack. This attack happens either if there is a composite finite field i.e. F_{p^m} so that, *m* is composite, or if *m* is a prime number that for a small number *t* holds true in $\left(2^t \equiv 1 \bmod m\right)$. Therefore, *m* must not be Mersen's or Fermat's prim number.

However, in such a case, using GHS algorithms and their improvements [15], the discrete logarithm problem can be solved. So, to prevent this attack, the curve must either be defined on a prime finite field F_p or a finite field F_{p^m} in which *m* is a prime number and order of number *2* in the multiplication group of *p* mode is a large number i.e. in relation $2^t \equiv 1 \bmod p$, *t* is a large number.

Table 2. Comparing of Key Size in HECC[33]

Security level	Elliptic Curve	Genus 2 Curve	Genus 3 Curve
256	94	47	32
512	128	64	43
1024	174	87	58
2048	234	117	78
4096	313	157	105
8192	417	209	139
16384	554	277	185

IV. Comparing HECC and ECC

The followings are some conclusions regarding the comparison between HECC and ECC:

- ECC with projective coordinates is almost always the most efficient system.
- Scalar multiplication of HECC with *g=3* and *h(x)=1* is always faster than that of HECC with *g=2*.
- Scalar multiplication of HECC with *g=3* is most often faster than ECC with affine coordinates.
- With an identical security level, software implementations of HECC with *g=2* and *g=3* ECC are equally efficient whereas hardware implementations of HECC with *g=2* and *g=3* are more efficient than that of ECC.
- Scalar multiplication of HECC with *g=2* and *g=3* and ECC with affine coordinates are equal if $\left(\frac{M}{I}\right)$ (Multiplication/Inverse) is small. In case this relation is big, then HECC with *g=2* and *g=3* is more efficient than ECC with affine coordinates.
- If the relation $\left(\frac{M}{I}\right)$ is small, HECC of genus *g=2* is quite efficient and if $\left(\frac{M}{I}\right)$ is big, HECC of genus *g=3* is more efficient.

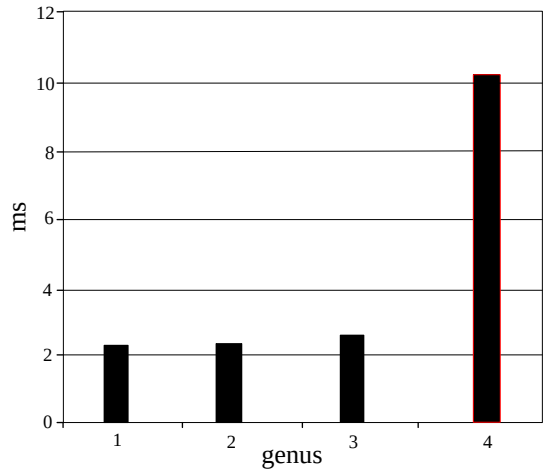

Fig.2. Comparing Scalar Multiplications on Pentium 4 @ 1.8GHZ for 2^{163} Security Level [34]

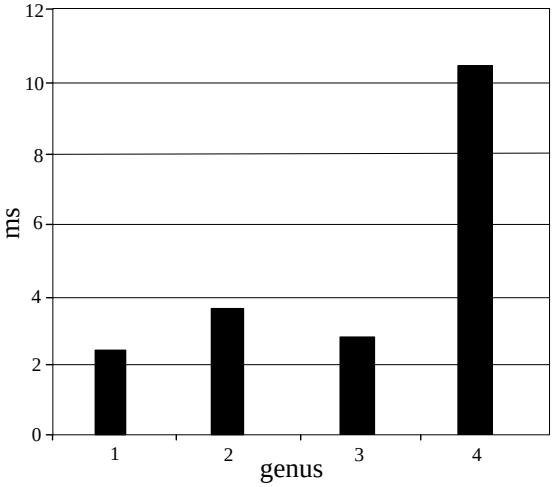

Fig.3. Comparing Scalar Multiplications on Pentium 4 @ 1.8GHZ for 2^{180} Security Level [34]

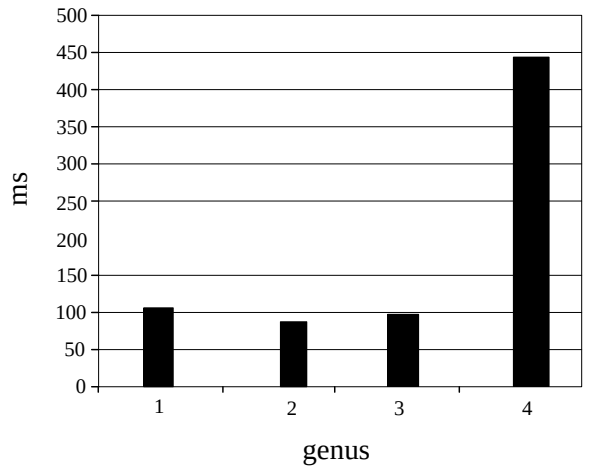

Fig.4. Comparing Scalar Multiplications on ARM 7 TDMI @ 80MHZ for 2^{163} Security Level [34]

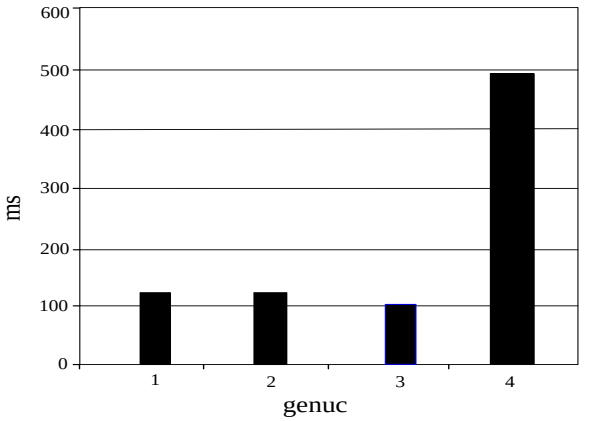

Fig.5. Comparing Scalar Multiplications on ARM 7 TDMI @ 80MHZ for 2^{180} Security Level [34]

As it is clear, in hardware implementation for a security level of 2^{160}, the genus 2-hyperelliptic curves are more efficient than hyper elliptic curves of genus *3*; whereas with a security level of 2^{180} it is the other way round [35].

Before this, it was generally agreed that because computations of genus 4-hyperelliptic curves were much more costly than hyper elliptic curves of smaller genus, hyper elliptic curves were not appropriate for application. While new findings show that for applications with lower security levels, genus 4-hyperelliptic curves are faster than genus 2-hyperelliptic curves. Moreover, it not only is as efficient as hyperelliptic curves of *g=3* but also, can replace elliptic curves. In addition, in hardware implementation of applications with a lower security level like in groups with order 2^{128}, computations of hyper elliptic curves of genus *g=4* are almost 1.46 times faster than that of hyperelliptic curves of *g=2*. It also uses the same efficiency as hyperelliptic curves of *g=3*. It is also noteworthy that compared with hyper elliptic curves of *g=2, 3*, hyperelliptic curves of *g=4* are more appropriate to be implemented in embedded microprocessors than general processors. To gain a security level of 2^{128}, a hyper elliptic curve of g=*4* can be defined on the finite field $F_{2^{32}}$. Thus implementing these curves on 32-bit processors will be really efficient. It will be more efficient than hyperelliptic curves of *g=2* and almost equally efficient as hyperelliptic curves of *g=3*. Usually, when using cheap embedded processors, a lower security level will be needed. In practice, a group of the order 2^{128} will suffice. This security level is almost higher than that of RSA-512 [28-32].

V. Pairings on Hyper Elliptic Curves

Take *C* as a hyperelliptic curve of genus g defined on F_q so that $\gcd(r,g)=1,\ r \left\| J\left(F_q\right)\right|$. Also, take *r* as a prime number. The embedding degree $J\left(F_q\right)$ related to *r* is the least integer number *k* so that $r \left| \left(q^k-1\right)\right.$. In other words, $F_{q^k}^*$ includes the group μ_r (the unit's r^{th} roots). Some of pairing based protocol are in [5-10]. Another important parameter in pairing functions is called ρ-value which is $\rho=\frac{g\log q}{\log r}$. This parameter is almost equal to the bit length of $\left|J\left(F_q\right)\right|$ to the bit length of a subgroup of the order *r*. A Jacobin with a number of prime members has the least amount of ρ-value $(\rho\approx 1)$. Hyperelliptic curves whose Jacobins have a small embedding degree and a subgroup with big prime order are appropriate to be used in paring functions. They are called pairing friendly [3, 13]. In practical application values of $r>2^{160}, k\le 60$ are needed. Also, as mentioned earlier, the best attack at the discrete logarithm problem is the ρ-Pollard algorithm which is implemented in a parallel way. Running time of this algorithm is $O\left(\sqrt{r}\right)$ with r as the value of the biggest prime subgroup $J\left(F_q\right)$. For hyper curves of *genus g=3, 4* there can be index calculus attacks with these respective complexities:

$$O\left(q^{\frac{3}{2}+\varepsilon}\right)=O\left(\left|J\left(F_q\right)\right|^{\frac{3}{8}+\varepsilon}\right), O\left(q^{\frac{4}{3}+\varepsilon}\right)=O\left(\left|J\left(F_q\right)\right|^{\frac{4}{9}+\varepsilon}\right)$$

To compare with the paralleled ρ-Pollard algorithm which depends on the subgroup of the order r, it can be said if $\rho<\frac{9}{8}$ of g=3 and also $\rho<\frac{4}{3}$ of g=4, then the index calculus attack reaches the upper boundary of the ρ-Pollard attack. However, the best algorithm to solve the discrete logarithm problem for hyper curves of genus g=2, 3, 4 have exponential running time. On the other hand, the best algorithm to compute the discrete logarithm in a finite field is the index calculus attack which has the sub-exponential running time related to the field's size.

Therefore, in order to reach an equal security level in both groups (multiplication group resulted from a finite field and Jacobin group resulted from a hyper curve) the value of q^k must be definitely larger than *r*.

Table 3. Embedding Degree for Genus 2-Hyper Curves with an Equal Security Level [28].

Security level (bit)	Size of subgroup (r)	Size of field expansion (q^k)	Embedding degree (*k*)					
			$\rho \approx 1$	$\rho \approx 2$	$\rho \approx 3$	$\rho \approx 4$	$\rho \approx 6$	$\rho \approx 8$
80	160	1024	6g	3g	2g	1.5g	g	0.8g
112	224	2048	10g	5g	3.3g	2.5g	1.6g	1.3g
128	256	3072	12g	6g	4g	3g	2g	1.5g
192	384	7680	20g	10g	6.6g	5g	3.3g	2.5g
256	512	15360	30g	15g	10g	7.5g	5g	3.8g

The table above shows examples of subgroup value, expanded field value, and the embedding degree size for an equal security level which are all held in relation $r \approx q^{g/\rho}$. Subgroups of the prime order and expanded fields (with large discriminates) follow the NIST standard. The above table is based on g=2. For g=2, 3, 4, these procedures must be followed:

If the Jacobin order is almost prime $(\rho \approx 1)$, then it is just required to arrange parameters in a way so that they can resist the index calculus attack. For g=3, the second column of the above table should be multiplied with $\frac{9}{8}$ and the fourth column at $\frac{8}{9}$. For g=4, the second and the fourth columns should be multiplied respectively at $\frac{4}{3}$ and $\frac{3}{4}$. Embedding degree for super singular hyper curves of genus g=2 holds true in relation $k \le 12$. For general hyper elliptic curves of genus g=2 in specific cases, it also holds true in relation $k \le 12$. The quickest implementation of pairing functions has been offered for the pairing η_T which uses a super singular genus 2-hyper elliptic curve defined on a finite field with the discriminate 2. Embedding degree of this case is 12 .The results reached at by [28] exhibit that:

- Implementation of a pairing called h on genus 2-hyper curves has a similar function as implementation of the Tate pairing.
- The η_T pairing's implementation on genus 2-hyper elliptic curves is much more efficient compared with that of h pairing.
- Implementation of the pairing η_T on genus 2-hyper curves is more efficient than implementation of the pairing η_T on super singular hyperelliptic curves defined on F_{2^m} .
- Computing the pairing function on genus 3-hyper curves is very inefficient than that of genus 2-hyper curves.
- Implementation time of a pairing function on genus 2-hyper curves defined on F_p is almost two times longer than implementation time of a pairing on hyper elliptic curves defined on F_p . To decrease this time distance, it is essential to use super singular genus 2-hyperelliptic curves.

VI. Conclusion

The current paper introduced hyper elliptic curves. It also offered comparisons between different types of these curves regarding their security and efficiency. Therefore, it can briefly be concluded that index calculus attack is possible for hyperelliptic curves of $g \ge 5$ .So, they are not appropriate to be used in public key cryptography. In applications with no need of a high security level like WSN with low-cost embedded processors that use cryptography algorithms with smaller key size, hyper elliptic curves of genus $g = 2,3,4$ can be applied.

References

[1] L. Adleman and M. Huang, Primality Testing and Abelian Varieties over Finite Fields, Lecture Notes in Mathematics, 1512, Springer-Verlag, Berlin, 1992.

[2] L. Adleman, J. DeMarrais and M. Huang, A subexponential algorithm for discrete logarithms over hyperelliptic curves of large genus over GF(q), Theoret. Comput. Sci. 226, 7–18, 1999.

[3] P. S. Barreto, S. D. Galbraith, C. Ó' hÉigeartaigh, M. Scott, Efficient pairing computation on supersingular Abelian varieties, Designs, Codes and Cryptography: Volume 42 Issue 3, 2007.

[4] D. Le Brigand, Decoding of codes on hyperelliptic curves, Eurocode '90, Lecture Notes in Computer Science, 514, Springer-Verlag, 126-134, 1998.

[5] M. H. Dehkordi, R. Alimoradi, Zero-Knowledge Identification Scheme Based on Weil Pairing, Lobachevskii Journal of Mathematics, Vol. 30, No. 3, 203–207. 2009.http://dx.doi.org/10.1134/S1995080209030020

[6] M. H. Dehkordi, , R. Alimoradi, A NEW BATCH IDENTIFICATION SCHEME, Discrete Mathematics, Algorithms and Applications, Vol. 1, No. 3, 369–376, 2009.http://www.worldscinet.com/dmaa/01/0103/S1793830909000294.html

[7] M. H Dehkordi, R Alimoradi, Authenticated key agreement protocol, CHINA COMMUNICATIONS 7 (5), 1-8, 2010.

[8] M. H Dehkordi, R Alimoradi, Identity–Based Multiple Key Agreement Scheme, KSII Transactions on Internet and Information Systems (TIIS) 5 (12), 2392-2402, 2011.

[9] M. H Dehkordi, R Alimoradi, Certificateless identification protocols from super singular elliptic curve.

Security and Communication Networks 7 (6), 979-986, 2014.
[10] Y. Driencourt and J. Michon, Elliptic codes over a field of characteristic 2, Journal of Pure and Applied Algebra, 45, 15-39, 1987.
[11] A. Enge, P. Gaudry, A general framework for subexponential discrete logarithm algorithms, ActaArith. 102 No1, 83–103, 2002.
[12] A. Enge, Computing discrete logarithms in high-genus hyperelliptic Jacobians in provably subexponential time, Math. Comp. 71 No238, 729–742, 2002.
[13] S. D. Galbraith, C. O. hEigeartaigh, C. Sheedy, Simplified pairing computation and security implications. J. Mathematical Cryptology 1(3): 267-281 (2007).
[14] P. Gaudry, An algorithm for solving the discrete log problem on hyperelliptic curves, Advances in Cryptology – Eurocrypt 2000, vol. 1807, Springer-Verlag, Berlin, 19–34, 2000.
[15] P. Gaudry, F. Hess, N. P. Smart, Extending the GHS Weil-descent attack, Advances in Cryptology – Eurocrypt 2002, Lecture Notes in Comput. Sci., vol. 2332, Springer-Verlag, Berlin, 29–44, 2002.
[16] P. Gaudry, N.The'riault, E. Thome , A double large prime variation for small genus hyperelliptic index calculus, preprint, 2004. http://eprint.iacr.org/2004/153/
[17] G. van der Geer, Codes and elliptic curves, in Efective Methods in Algebraic Geometry, Birkh auser, 159-168, 1991.
[18] H.W. Lenstra, Factoring integers with elliptic curves, Annals of Mathematics, 126, 649-673, 1987.
[19] H.W. Lenstra, J. Pila and C. Pomerance, A hyperelliptic smoothness test. I, Philosophical Transactions of the Royal Society of London A, 345, 397-408, 1993.
[20] J. van Lint and G. van der Geer, Introduction to Coding Theory and Algebraic Geometry, Birkh auser-Verlag, Basel, Germany, 1988.
[21] B. Kaliski, A pseudorandom bit generator based on elliptic logarithms, Advances in Cryptology { CRYPTO '86, Lecture Notes in Computer Science, 293, Springer-Verlag, 84-103, 1987.
[22] N. Koblitz, Elliptic curve cryptosystems, Mathematics of Computation, 48, 203-209, 1987.
[23] N. Koblitz, Hyperelliptic cryptosystems", Journal of Cryptology, 1, 139-150, 1989.
[24] A. Menezes, T. Okamoto, and S. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. IEEE Transactions on Information Theory, 39:1639-1646,1993.
[25] A. Menezes, Y-H. Wu, R. J. Zuccherato. An elementary introduction to hyperelliptic curves, Published as Technical Report CORR 96-19, Department of C&O, University of Waterloo, Ontario, Canada, November 1996.
[26] V. Miller, Uses of elliptic curves in cryptography, Advances in Cryptology { Proceedings of Crypto '85, Lecture Notes in Computer Science, 218, Springer-Verlag, 417-426, 1986.
[27] L. B. Oliveira, D. F. Aranha, E. Morais, F. Daguano, J. Lopez, and R. Dahab, TinyTate: Computing the Tate Pairing in Resource-Constrained Sensor Nodes, Proceedings of the Sixth IEEE International Symposium on Network Computing and Applications (NCA 2007), 318-323, 2007.
[28] Colm Ó hÉigeartaigh, Michael Scott -Pairing Calculation on Supersingular Genus 2 Curves *Selected Areas in Cryptography - SAC 2006, LNCS, 2007.*
[29] J. Pelzi, Fast hyperelliptic curve cryptosystems for embedded microprocessors, Master's thesis, Ruhr-University of Bochum, 2002.
[30] J. Pelzl, T. Wollinger, J. Guajardo, J. and C. Paar. Hyperelliptic Curve Cryptosystems: Closing the Performance Gap to Elliptic Curves. CHES 2003, LNCS 2779, 351–365. Springer, 2003.
[31] J. Pelzi, T. Wollinger, C. Paar Special hyperelliptic curve cryptosystems of genus two: Efficient arithmetic and fast implementation, Embedded Cryptographic Hardware: Design and Security, Nova Science Publishers, 2004.
[32] J. Pelzi, T. Wollinger, C. Paar, Low Cost Security: Explicit Formulae For Genus-4 Hyperelliptic curves, 2004.
[33] N. The'riault, Index calculus attack for hyperelliptic curves of small genus, Advances in Cryptology – Asiacrypt 2003, Lecture Notes in Comput. Sci., vol. 2894, Springer-Verlag, Berlin, 75–92, 2003.
[34] T. Wollinger, J. Pelzl, V. Wittelsberger, C. Paar, G. Saldamli, and, Elliptic &Hyperelliptic Curves on Embedded Platform, ACM Transactions in Embedded Computing Systems (TECS), vol. 3, no. 3, 509-533, 2004.
[35] T. Wollinger, Software and Hardware Implementation of Hyperelliptic Curve Cryptosystems. PhD thesis, Department of Electrical Engineering and Information Sciences, Ruhr-Universität Bochum, Bochum, Germany, 2004.

Authors' Profiles

Reza Alimoradi assistant professor of mathematics and computer science department in the University of Qom, Iran, since 2012. He received the M.Sc and PhD. degrees in mathematics from Iran University of science and technology. Interested Research Fields are Cryptography and Network security.

How to cite this paper: Reza Alimoradi,"A Study of Hyperelliptic Curves in Cryptography", International Journal of Computer Network and Information Security(IJCNIS), Vol.8, No.8, pp.67-72, 2016.DOI: 10.5815/ijcnis.2016.08.08

