# A Study of Hyperelliptic-Curve Cryptography

*Synopsis of the Thesis to be submitted in partial fulfillment of the requirements for the award of the degree*

of

**Master of Science**

by

**Anindya Ganguly**

Under the supervision of

**Prof. Abhijit Das**
**Prof. Dipanwita Roy Chowdhury**


**Department of Computer Science and Engineering**
**Indian Institute of Technology Kharagpur**
**Kharagpur- 721302, West Bengal, INDIA**

# Contents

# 1 Introduction

Cryptographic primitives are extensively used to secure the modern digital era. Traditionally, cryptography deals with three principal characters: the sender (Alice), the receiver (Bob), and the attacker (Malice). Alice sends an encrypted message via an insecure channel to Bob. Bob decrypts the ciphertext using a predefined secret key, and recovers the message. Malice listens to the channel, and reads the encrypted message. His goal is to figure out the plaintext message from the ciphertext. Cryptographic protocols should ensure that Malice can succeed only with negligible probability.

Public-key cryptography originates from the seminal discoveries of the Diffie–Hellman key-agreement protocol [4] and the RSA cryptosystem [18]. Since then, quite a few cryptographic schemes are proposed based on the difficulty of factoring large integers or computing discrete logarithms in large groups. Protocols based on the hardness of DLP are defined over cyclic groups $G$, like the multiplicative groups of integers modulo $n$.

Two popular families of groups are the multiplicative groups of finite fields, and the sets of rational points on elliptic curves defined over finite fields. Proposed by Koblitz [12] and Miller [21], elliptic curves have been used extensively in several cryptographic protocols. The main advantage of ECC is that it provides the same level of security as the traditional multiplicative groups (like $\mathbb{F}_q^*$), but the elliptic-curve groups are much smaller than finite-field groups.

Later in 1989, Koblitz [13] proposes the use of hyperelliptic curves over finite fields for cryptographic purposes. However, these curves are studied less extensively by the cryptographic community than the schemes based on RSA, finite-field, and elliptic-curve discrete logarithms. Genus-two hyperelliptic curves offer the same level of security as elliptic curves, but with half field sizes. To achieve 128 bits of security, elliptic curves need 256-bit fields, whereas hyperelliptic curves require only 128-bit fields.

In this work, we successfully generate a family of subfield hyperelliptic curves, and propose a variant of the ElGamal scheme, that can be implemented using our subfield curves. The security of the adopted scheme is analyzed. The organization of the report is as follows. Section 2 provides the necessary backdrop for our research. Section 3 describes the contributions of the thesis. This section starts with a brief discussion about the proposed family of subfield hyperelliptic curves, and the implementation issues. Subsequently, we elaborate the adopted ElGamal scheme and its security issues. In Section 4, the organization of the thesis is presented. The report ends with a conclusion, and a mention of some directions for further research.

# 2 Motivation and Objectives

Genus-one hyperelliptic curves are known as elliptic curves. Higher-genus curves ($g > 1$) are also considered for cryptographic purposes. For hyperelliptic curves of genus $g > 1$, the Jacobians of the curves provide the underlying Abelian group structure. For large-genus hyperelliptic curves, there exist algorithms faster than the generic square-root methods and having subexponential running times, to solve the DLP. But for $g \leq 3$, no such subexponential algorithm is known.

In hyperelliptic-curve cryptography, generating suitable cryptographically secure curves over finite fields is an important issue. Literature suggests that point-counting algorithms over large prime finite fields are not very efficient. Subfield hyperelliptic curves are especially attractive from an efficiency point of view. Moreover, subfield hyperelliptic curves offer faster Jacobian arithmetic than hyperelliptic curves over large prime fields.

Furukawa et al. [8] propose an algorithm for the order computation of the Jacobian of the hyperelliptic curve of the form $y^2 = x^5 + ax$ over large prime fields. They also present the new family $y^2 = x^5 + a$. Satoh [19] develops a probabilistic polynomial-time algorithm to identify whether the curve $y^2 = x^5 + ax^3 + bx$ is suitable, that is, whether the order of the Jacobian has a large prime divisor. Buhler and Koblitz [3] propose an algorithm for particular types of curves $y^2 + y = x^n$ over $\mathbb{F}_p$, where $n$ is odd, and $p$ is any prime with $n|(p-1)$. To the best of our knowledge, no families of subfield hyperelliptic curves are explicitly proposed in the literature.

There exist software implementations of elliptic- and hyperelliptic-curve cryptography. Gaudry [9] writes a library for finite-field arithmetic. The $\mathtt{m}_p\mathbb{F}_q$ library is practically used for curve-based public-key cryptography. A HECC software implementation is done by Pelzl et al. [17]. They put execution times in tabulated manner for curves of genus two and three. Avanzi [1] implements a prime-field library nuMONGO which includes elliptic- and hyperelliptic-curve arithmetic. These implementations use large prime fields. We are not aware of any reported implementation that includes subfield curves for cryptographic purposes.

An efficient addition algorithm for the divisor group is required for the hyperelliptic curves. Cantor proposes a fast algorithm for addition using Mumford representation of divisors. This addition of the divisor group in the hyperelliptic curve is not so efficient as elliptic-curve point addition. The performance gap was narrowed by Harley [11]. Later, Lange provides an explicit version of Harley's formula [14]. Lange's explicit version gives a potentially powerful speedup for hyperelliptic-curve addition. To enhance the performance further, Lange [15] also proposes an inversion-free addition algorithm. Several researchers try to improve the performance of divisor-class arithmetic.

In this backdrop, our work is mostly motivated by the need of narrowing the performance gap between elliptic- and hyperelliptic-curve cryptography. The process involves looking for new families of hyperelliptic curves, and going for and tuning software implementations of the Jacobian arithmetic. Curves that can be efficiently generated are suitable in this context. Eventually, the study should investigate the effectiveness of using these curves in practical cryptographic schemes.

# 3 Contributions of the Thesis

In this thesis, a new family of hyperelliptic curves is proposed. The point-counting algorithm for this family is detailed. Existing algorithms for Jacobian arithmetic are compared. Implementation details of the finite-field arithmetic is also presented. The performance for the proposed family is analyzed. Finally, the discrete logarithm problem defined over the Jacobian of the curves is scrutinized.

ElGamal encryption is a popular public-key cryptographic primitive. This can be implemented using the proposed family of hyperelliptic curves. For this purpose, an encoding map is proposed. It is proved that the encoding map is well distributed and efficiently computable. It is established that this variant of ElGamal encryption is no less secure than original ElGamal encryption.

## 3.1 A new family of subfield hyperelliptic curves

Let $\mathcal{C}$ be a hyperelliptic curve of genus two defined over a finite field $\mathbb{F}_q$. A suitably large subgroup $G$ of the Jacobian $\mathbb{J}_q$ is to be used to build cryptographic schemes. For cryptographic reasons, the group order $n$ should be a prime. The bit length of $n$ is dictated by the security level $l$. Since the square-root attacks are the only attacks known for hyperelliptic curves of genus two, we take $l \approx |n|/2$. Since $64$-bit security is not considered safe given the available computing powers, and we require $l \geq 80$. Security level $l = 128$ is prescribed for long-term use. We target achieving several security levels depending upon the needs of the cryptographic applications. More specifically, we take $l = 80, 96, 112, 128$.

For achieving $l$-bit security, we need a field $\mathbb{F}_q$ of bit size $|q| \geq l$. Moreover, the size of the Jacobian $\mathbb{J}_q$ should be a prime (around $2l$-bits). One option is to take an $l$-bit prime as $q$. But point-counting algorithms over such large prime fields are mathematically complicated and practically inefficient.

To work around this problem, we have devised a modified approach. We use quintic extensions $\mathbb{F}_{p^5}$ for a prime $p$. For $l \leq 128$, this prime $p$ fits in a 32-bit unsigned integer. We generate a curve over $\mathbb{F}_p$, and compute the order of $\mathbb{J}_p$. Since $p$ is now small, simple and practical point-counting algorithms can be used. We then consider the quintic extension $\mathbb{F}_q = \mathbb{F}_{p^5}$. The curve $\mathcal{C}$ is naturally defined over $\mathbb{F}_q$. Moreover, given the group size $|\mathbb{J}_p|$, the group size $|\mathbb{J}_q|$ can be calculated using simple formulas. We require $n = |\mathbb{J}_q|/|\mathbb{J}_p|$ to be a prime.

This approach helps us generate many suitable curves of security level $l$ fairly quickly. On the flip side, we now have to work in a field of bit size $|q| = 5l/4$. For efficiency reasons, we take $\mathcal{C}$ of the form

$$\mathcal{C} : y^2 = x^5 + x + a, \quad a \in \mathbb{F}_p.$$

To compute the orders of the curves over $\mathbb{F}_p$, the baby-step-giant-step method is used. This algorithm may fail on a few occasions, but given that we have a large domain for varying $a$, many curves with known orders can be generated fairly efficiently. We have $|\mathbb{J}_p| \approx p^2$.

Since $\mathbb{F}_p \subseteq \mathbb{F}_q$, $\mathcal{C}$ is naturally defined over the extension field $\mathbb{F}_q = \mathbb{F}_{p^5}$, and the order of $\mathbb{J}_q$ can be computed by the Newton–Girard formulas. We have $|\mathbb{J}_p| \approx q^2 = p^{10}$. Since $\mathbb{J}_p$ is a subgroup of $\mathbb{J}_q$, the order of $\mathbb{J}_p$ must divide the order of $\mathbb{J}_q$. We use the cofactor

$$n = \frac{|\mathbb{J}_q|}{|\mathbb{J}_p|}.$$

If $n$ is prime, $\mathbb{J}_q$ contains a subgroup $G$ of this order. The bit length of $n$ is

$$|n| = |\mathbb{J}_q| - |\mathbb{J}_p| \approx (10 - 2)|p| = 8|p|,$$

that is, the security level is $|n|/2 \approx 4|p| = l$, as planned.

Indeed, we have $|n| = 2l$ or $|n| = 2l + 1$ if $p \approx 2^{l/4}$. In terms of efficiency of Jacobian arithmetic over $\mathbb{F}_q$, there is hardly any difference in the running times between these two cases. However, for index arithmetic (modulo $n$), the case $|n| = 2l + 1$ introduces some inefficiency. If we use 32-bit words to pack fragments of multiple-precision integers, then for the stated values of $l$, we need an extra word compared to the case $|n| = 2l$. This may be an issue for some cryptographic algorithms.

Being subfield curves, the members of our family of curves are easy to generate.

### 3.2 Software implementations

In order to compare the practical performances of hyperelliptic and elliptic curves [16], we have made extensive software implementations. We have developed a mathematical library in order to implement the Jacobian arithmetic over $\mathbb{F}_q$. Our library is specifically tuned to deal with subfield hyperelliptic curves of our family, and consists of the following components.

- Arithmetic of multiple-precision integers.
- Arithmetic of prime files $\mathbb{F}_p$ ($|p| \leqslant 32$).
- Polynomial arithmetic over $\mathbb{F}_p$.
- Arithmetic of extension fields $\mathbb{F}_q = \mathbb{F}_{p^5}$.
- Polynomial arithmetic over $\mathbb{F}_q$.
- Jacobian arithmetic over $\mathbb{F}_q$.

We compare the performances of elliptic and hyperelliptic curves at the 128-bit security level. The curve P-256 [16] is used as a representative elliptic curve. We consider two hyperelliptic curves. The first is Generic-1271 [2] defined over the prime field $\mathbb{F}_{2^{127}-1}$. The second is the subfield curve $y^2 = x^5 + x + 23$ of our proposed family, defined over the quintic extension $\mathbb{F}_{p^5}$ for the 32-bit prime $p = 4294836163$. Our mathematical library is used alongside two popular and efficient public-domain libraries: the GNU multiple-precision library (GMP) [10], and the number theory library (NTL) [20]. GMP does not provide support for polynomial

| Curve (Library) | Doubling | Addition | Scalar multiplication |
|---|---|---|---|
| P-256 (NTL) | 0.000003 | 0.000003 | 0.001375 |
| Generic-1271 (Our library) | 0.000191 | 0.000201 | 0.038537 |
| Generic-1271 (NTL) | 0.000020 | 0.000022 | 0.007514 |
| Generic-1271 (GMP) | 0.000054 | 0.000058 | 0.033367 |
| Subfield curve $\mathcal{C}$ (Our library) | 0.000034 | 0.000038 | 0.011614 |
| Subfield curve $\mathcal{C}$ (NTL) | 0.000100 | 0.000102 | 0.034476 |

Table 1: Comparison of Cantor's original algorithm with elliptic-curve arithmetic (all times are in milliseconds)

| Coordinate | Curve (Library) | Doubling | Addition | Scalar multiplication |
|---|---|---|---|---|
| Affine | Generic-1271 (NTL) | 0.000007 | 0.000009 | 0.002439 |
| Affine | Subfield curve $\mathcal{C}$ (Our library) | 0.000009 | 0.0000010 | 0.003021 |
| Affine | Subfield curve $\mathcal{C}$ (NTL) | 0.000028 | 0.000026 | 0.008442 |
| Projective | Generic-1271 (NTL) | 0.000007 | 0.000007 | 0.002466 |
| Projective | Subfield curve $\mathcal{C}$ (Our library) | 0.000011 | 0.000012 | 0.003167 |
| Projective | Subfield curve $\mathcal{C}$ (NTL) | 0.000026 | 0.000028 | 0.008604 |
| Weighted | Generic-1271 (NTL) | 0.000007 | 0.000009 | 0.002576 |
| Weighted | Subfield curve $\mathcal{C}$ (Our library) | 0.000008 | 0.000012 | 0.002944 |
| Weighted | Subfield curve $\mathcal{C}$ (NTL) | 0.000025 | 0.000031 | 0.008507 |

Table 2: Performance comparison of elliptic and hyperelliptic curves (all times are in milliseconds)

arithmetic, and so is used for curves defined over prime fields only. NTL supports both integer and polynomial arithmetic, and so is used to implement the basic operations of all the three curves.

We first compare the performance of Cantor's algorithm for hyperelliptic curves with that of elliptic curves in Table 1. The figures illustrate that Cantor's algorithm is significantly inefficient compared to the elliptic-curve arithmetic. We therefore go for the implementations of the formulas supplied by Harley [11] and Lange [14, 15].

Table 2 illustrates the performance of the subfield curve and generic-1271 in different coordinates systems. For affine coordinates, Lange's explicit formula is implemented. NTL being the most efficient multiple-precision integer library, we report the timings of P-256 and Generic-1271 for this library only. For subfield curves, algorithms are implemented using both our mathematical library and NTL. The first inference we draw from these figures is that the performance gap between elliptic and hyperelliptic curves is now significantly reduced. Second, for our family of subfield curves, our specially tuned library runs much faster than NTL (whereas GMP is not straightaway applicable for these curves). Finally, there is a stiff competition between hyperelliptic curves over prime fields and hyperelliptic curves over extension fields.

### 3.3 Cryptographic primitives

Given the Jacobian arithmetic, encryption, signature, and authentication primitives can be developed. The ElGamal encryption scheme calls for a map to encode messages to divisors. To that effect, we propose an encoding scheme. We show that this map is well-distributed, and renders the adapted encryption scheme the same security guarantees as in the original ElGamal encryption.

Fouque, Joux and Tibouchi propose an injective encoding for elliptic curves. This construction uses the existence of a covering curve of genus two, for which a bijective encoding is known [6]. Later, Fouque and Tibouchi propose a "nearly bijection" encoding map. However, they use a curve defined over prime fields [7]. No such map exists for subfield curves. In our case of subfield hyperelliptic curves, the group $G$ in which the

ElGamal scheme works is a proper subgroup of the Jacobian $\mathbb{J}_q$ over the quintic extension. This is because the Jacobian $\mathbb{J}_q$ is the internal direct sum of $G$ with the Jacobian $\mathbb{J}_p$ over the ground field. An efficient and reversible encoding of messages to elements of $G$ is not straightforward.

We work around this difficulty by mapping messages to the strictly larger group $\mathbb{J}_q$, that is, each encoded message now has a component in a cryptographically small group $\mathbb{J}_p$ which plays no role in the security of the ElGamal scheme. The question that our encoding scheme raises is whether this component has any potential of leaking important cryptographic secrets.

To start with, we elaborate our adaptation of the ElGamal scheme for subfield curves along with our message encoding and decoding techniques. We denote our encoding scheme by the function $\theta$.

**— Public parameters**

1. Field sizes $p$ and $q = p^5$, the element $a \in \mathbb{F}_p$ defining the curve $y^2 = x^5 + x + a$, the size $n$ of the group $G$, a base point $P \in G$, the message length $l$, and the padding length $l'$.

**— Key pair of the recipient**

2. $(x, Y)$, where $x \in_U \mathbb{Z}_n$ (private key), and $Y = xP \in G$ (public key).

**— Encoding**

3. Let the message be $m \in \{0,1\}^l$.
4. Break $m$ into two $\frac{l}{2}$-bit chunks: $m = m_0 \mid\mid m_1$. For each $b \in \{0,1\}$, generate $r_b \in_U \{0,1\}^{l'}$ such that $x_b^5 + x_b + a$ is a square in $\mathbb{F}_q$, where $x_b = 0 \mid\mid b \mid\mid m_b \mid\mid r_b$. Let $y_b$ be a square root of $x_b^5 + x_b + a$ in $\mathbb{F}_q$.
5. Take the divisor $(u_2, u_1, u_0, v_1, v_0)$ with the two rational points $(x_0, y_0)$ and $(x_1, y_1)$ as the message representative $M$. Notice that $M \in \mathbb{J}_q$ (we do not, in general, have $M \in G$).

**— Encryption**

6. Generate $k \in_U \mathbb{Z}_n$, and set $R = kP \in G$.
7. Compute $S = M + kY \in \mathbb{J}_q$.
8. Send $(R, S)$ to the recipient.

**— Decryption**

9. Recover $M = S - xR \in \mathbb{J}_q$.

**— Decoding**

10. Let $M = (u_2, u_1, u_0, v_1, v_0)$. We have $x_0 + x_1 = -u_1$, and $x_0 x_1 = u_0$. Solve these equations (quadratic) to obtain $x_0, x_1$. Notice that $x_b$ has second msb $b$.
11. Recover $m_0, m_1$ from $x_0, x_1$ after removing the paddings. Output $m = m_0 \mid\mid m_1$.

The encoding map $\theta$ used in our variant of ElGamal encryption has some desirable properties.

- The encoding map is efficiently computable in polynomial time. The inverse of the map is also efficiently computable.

- This map can be applied for all forms of subfield hyperelliptic curves.
- Our encoding scheme is a probabilistic map due to the concatenated pseudorandom bits.
- This map does not preserve arithmetic operation. Let $D_1 = \theta(k_1)$ and $D_2 = \theta(k_2)$. Then, any correlation between $k_1$ and $k_2$ does not reflect on $D_1$ and $D_2$.
- This is a well-distributed map.

The following theorem establishes that our point-encoding scheme is well-distributed. To that effect, we use character sums. Similar types of results can be found in [5, 7]. Using this result, we obtain a bound on statistical distance, which in turn ensures that our encoding map terminates in expected polynomial time.

**Theorem 3.1** *Let $\chi$ be any character of the Abelian group GF$(q)$. The character sum is defined as*

$$T(\chi) = \sum_{u \in \mathbb{F}_q} \chi(\theta(u)).$$

*Then, for a non trivial character, we have $T(\chi) \leq 2\sqrt{q} + 11$.*

This theorem implies that for all $D \in \mathbb{J}_q$, we have

$$\left| \frac{N(D)}{q^2} - \frac{1}{|\mathbb{J}_q|} \right| < (2\sqrt{q} + 11)^2,$$

where $N(D)$ be the number of preimages of $D$ under $\theta$. The statistical distance between the distribution defined by the point-encoding map on $\mathbb{J}_q$ and the uniform distribution is

$$\sum_{D \in \mathbb{J}_q} \left| \frac{N(D)}{q^2} - \frac{1}{|\mathbb{J}_q|} \right| \leq \left( 2 + \frac{11}{\sqrt{q}} \right)^2.$$

Therefore, the bound on the statistical distance is $\frac{c}{\sqrt{q}} + O(\frac{1}{q})$ (where $c$ is a positive constant). The proof of this statement can be found in [7]. The following theorem establishes that our randomized encoding scheme is efficiently computable. Similar results for elliptic curves can be found in [6].

**Theorem 3.2** *For large enough $q$, the expected number of iterations in $\theta$ on any input message $m$ is $< 3$.*

### 3.3.1 Security issues

Now we define the Diffie–Hellman problems. Let $P$ be a generator of an additive cyclic group $G$ of order $n$. Let $a, b \in Z_n$. A triple $[aP, bP, abP]$ is called a *Diffie–Hellman triple*. Two important problems associated with these triples go as follows.

**DDH Problem:** Let $[aP, bP, X]$ be a given triple. The decisional Diffie–Hellman problem deals with deciding whether the given triple is a Diffie–Hellman triple, that is, whether $X = abP$. The DDH assumption is that it is computationally infeasible to solve the DDH problem. Let $\mathcal{O}$ be an oracle that, given a triple, identifies whether the triplet is a Diffie–Hellman triple. Under the DDH assumption, no such oracle having polynomial running time can exist.

**CDH Problem:** Let $aP, bP$ be supplied as inputs. The computational Diffie–Hellman problem deals with the determination of the group element $abP$. The CDH assumption is that it is computationally infeasible to solve the CDH problem. In other words, an oracle $\mathcal{O}$ that, given $aP, bP$, returns $abP$ in polynomial time cannot exist.

The security of the original ElGamal encryption scheme is equivalent to the DH problems. Here, we establish that our adaptation of the ElGamal scheme is also secure under the DH assumptions.

First, we note that the divisor $kY = kxP \in G$ masks the encoded message $M$. Therefore the knowledge of $M$ is equivalent to the knowledge of the mask. The sender can calculate it from $k$ and the recipient's public key $Y = xP$, whereas the recipient can compute the mask from $R = kP$ and his private key $x$. To a passive eavesdropper, the challenge is to compute the mask $kxP$ from the knowledge of $kP$ and $xP$ alone. Consequently, ElGamal decryption is as difficult as solving the CDH to the eavesdropper.

What remains is to justify that our encoding of messages to divisors in $\mathbb{J}_q$ (not in $G$) does not lead to some information leakage. Since $\mathbb{J}_q$ is the internal direct sum of $G$ and $\mathbb{J}_p$, the encoded message can be uniquely decomposed as $M = M_G + M_p$, where $M_G \in G$, and $M_p \in \mathbb{J}_p$. The $n$-th multiple of any element of $G$ is zero, and therefore $nS = nM = nM_p$. The size $e$ of the small group $\mathbb{J}_p$ can be easily computed, and is coprime to $n$. If $\eta \equiv n^{-1} \pmod{e}$, then $M_p = \eta n M_p = \eta n S$, that is, $M_p$ can be determined by any passive eavesdropper. The relevant concern is therefore whether $M_p$ reveals some partial information about $m$.

This is where the random padding strings $r_0, r_1$ play a crucial role. If the padding length $l'$ is somewhat larger than the bit size of $p$, then for each message $m$, we expect each element of $\mathbb{J}_p$ to appear as $M_p$. Moreover, the different elements of $\mathbb{J}_p$ are expected to appear as $M_p$ with nearly equal probabilities. So we expect that the padding destroys all correlations between $m$ and $M_p$. What the eavesdropper sees in $S$ is a random element of the group $\mathbb{J}_p$. Moreover, $M_p$ has nothing to do with the key pair (in particular, the private key $x$) of the recipient, because it is generated independently before the application of any key-related quantity.

# 4 Organization of the Thesis

**Chapter 1** provides the objectives and motivation behind our work, and summarizes the contributions reported in the thesis.

**Chapter 2** deals with the mathematical preliminaries about hyperelliptic curves. More precisely, the concepts of divisors, divisor classes, and the Jacobians are introduced. The discussion also sets up the notations we use in the rest of the thesis.

**Chapter 3** starts by pointing out the difficulties for computing the orders of Jacobians. Subsequently, an algorithm is described for computing the order of Jacobians for subfield curves. It also deals with Jacobian arithmetic. To start with, Cantor's algorithm is introduced for computing the sum of two reduced divisors. This is followed by Harley's algorithm and Lange's algorithm. Inversion-free divisor-class addition is also described. The chapter presents the performance analysis of the proposed subfield curves alongside elliptic curves. The use of Lange's formulas and inversion-free arithmetic leads to comparable performances between subfield hyperelliptic curves and elliptic curves.

**Chapter 4** is a discussion of the use of our curves in cryptographic schemes. An adaptation of the ElGamal scheme to the case of our curves raises some security issues which are identified in this chapter. A security analysis is also presented for the adapted scheme.

**Chapter 5** concludes the thesis after summarizing the reported work and identifying some directions for further research.

**Appendix A** provides a list of some curves of our family at several security levels.

# 5 Conclusion and Future Scopes

In this work, a new family of hyperelliptic curves is proposed. To the best of our knowledge, we are the first to report an explicitly constructed family of cryptographically suitable subfield hyperelliptic curves. Our

experiments reported in this work have been able to narrow the gap between the performances of elliptic and hyperelliptic curves. We have also established our proposed family of subfield curves to be nearly as efficient and practical as curves over prime fields. Possibilities of further performance enhancements of our family of curves are worth investigating. We have investigated the security issues for our family of curves. All existing algorithms for solving the discrete logarithm problem have been found to be inefficient for our family. However, like all other curves, our family of curves is not quantum secure.

A new encoding scheme for ElGamal encryption is proposed. This encoding is well distributed which implies that our variant of ElGamal encryption is equivalent to the original ElGamal scheme in terms of formal security. In particular, our scheme is IND-CPA secure under the DDH assumption. Moreover, proving the IND-CCA1 security of our scheme is an open problem (like original ElGamal encryption).

## Publication from this Thesis

ANINDYA GANGULY, ABHIJIT DAS, DIPANWITA ROY CHOWDHURY, AND DEVAL MEHTA, *A Family of Subfield Hyperelliptic Curve for Use in Cryptography*, 22nd International Conference on Information and Communications Security (ICICS 2020), Copenhagen, Denmark, 2020.

## References


[1] R. M. AVANZI, *Aspects of hyperelliptic curves over large prime fields in software implementations*, in International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2004, pp. 148–162.

[2] J. W. BOS, C. COSTELLO, H. HISIL, AND K. LAUTER, *Fast cryptography in genus 2*, in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2013, pp. 194–210.

[3] J. BUHLER AND N. KOBLITZ, *Lattice basis reduction, jacobi sums and hyperelliptic cryptosystems*, Bulletin of the Australian Mathematical Society, 58 (1998), pp. 147–154.

[4] W. DIFFIE AND M. HELLMAN, *New directions in cryptography*, IEEE transactions on Information Theory, 22 (1976), pp. 644–654.

[5] R. R. FARASHAHI, P.-A. FOUQUE, I. SHPARLINSKI, M. TIBOUCHI, AND J. VOLOCH, *Indifferentiable deterministic hashing to elliptic and hyperelliptic curves*, Mathematics of Computation, 82 (2013), pp. 491–512.

[6] P.-A. FOUQUE, A. JOUX, AND M. TIBOUCHI, *Injective encodings to elliptic curves*, in Australian Conference on Information Security and Privacy, Springer, 2013, pp. 203–218.

[7] P.-A. FOUQUE AND M. TIBOUCHI, *Deterministic encoding and hashing to odd hyperelliptic curves*, in International Conference on Pairing-Based Cryptography, Springer, 2010, pp. 265–277.

[8] E. FURUKAWA, M. KAWAZOE, AND T. TAKAHASHI, *Counting points for hyperelliptic curves of type y 2= x 5+ ax over finite prime fields*, in International Workshop on Selected Areas in Cryptography, Springer, 2003, pp. 26–41.

[9] P. GAUDRY AND E. THOMÉ, *The mpFq library and implementing curve-based key exchanges*, in SPEED: Software Performance Enhancement for Encryption and Decryption, Amsterdam, Netherlands, Jun 2007, ECRYPT Network of Excellence in Cryptology, pp. 49–64.

[10] T. GRANLUND, *The GNU multiple precision arithmetic library*, `https://gmplib.org/`, (1996).

[11] R. HARLEY, *Fast arithmetic on genus 2 curves*, available at `http://cristal.inria.fr/~harley/hyper`, (2000).

[12] N. KOBLITZ, *Elliptic curve cryptosystems*, Mathematics of Computation, 48 (1987), pp. 203–209.

[13] N. KOBLITZ, *Hyperelliptic cryptosystems*, Journal of Cryptology, 1 (1989), pp. 139–150.

[14] T. LANGE, *Efficient arithmetic on genus 2 hyperelliptic curves over finite fields via explicit formulae*, IACR Cryptology ePrint Archive, 121 (2002).

[15] ——, *Inversion-free arithmetic on genus 2 hyperelliptic curves.*, IACR Cryptology ePrint Archive, 147 (2002).

[16] G. LOCKE AND P. GALLAGHER, *Fips pub 186-3: Digital signature standard (dss)*, Federal Information Processing Standards Publication, 3 (2009), pp. 186–3.

[17] J. PELZL, T. WOLLINGER, J. GUAJARDO, AND C. PAAR, *Hyperelliptic curve cryptosystems: Closing the performance gap to elliptic curves*, in International Workshop on Cryptographic Hardware and Embedded Systems, Springer, 2003, pp. 351–365.

[18] R. L. RIVEST, A. SHAMIR, AND L. M. ADLEMAN, *Cryptographic communications system and method*, Sept. 20 1983. US Patent 4,405,829.

[19] T. SATOH, *Generating genus two hyperelliptic curves over large characteristic finite fields*, in Annual International Conference on the Theory and Applications of Cryptographic Techniques, Springer, 2009, pp. 536–553.

[20] V. SHOUP, *NTL: A library for doing number theory*, http://www. shoup. net/ntl/, (2001).

[21] M. S. VICTOR, *Use of elliptic curves in cryptography*, in CRYPTO, Springer, 1986, pp. 417–426.