# CS 6530 Applied Cryptography – Jul-Nov 2025

## Assignment 2 – 1st Sept 2025

The Goal of this assignment are as follows:

a) Gain deeper understanding of the Elliptic Curve Cryptography and establishing secret keys between two entities.
b) Understand how ECC is used for Digital Signature (ECDSA)
c) Understand how ECC is used for enabling encryption and decryption.
**d) As there are many aspects to be done, the assignment can be done as a Team of two members. (This is a common subject matter, hence reach out to other classmates and form the teams). Please inform your team details by 3rd September 2025. Co-ordinate with TAs.**
**e) The Peers who will review the output will be determined by the Instructor+TAs and communicated by 5th September. (There will be random selection of review groups)**

### Assignment Submission Dates

- Assignment Submission date: 15th September 11.59 PM.
- Late submission by 16th September 11.59 PM – 20% penalty
- Late submission by 17th September 11.59 PM – 40% penalty.
- No acceptance of assignment beyond 17th September 11.59 PM.
- No extensions.

Submission location: Course Moodle Page.

**Please submit the response as one zip file on the Moodle page,** with a readme for the zip file, as what is the content of each file in the zip. Guidance is given to the naming convention, please follow the same for your submission.

**BACKGROUND:** ECC Curves used in practice are well defined.

1) **NIST Curves:** For example, aaccording to **NIST SP 800-186 (2023)**, the currently recommended elliptic curves for U.S. Government use include:
   a. ✅ **Weierstrass Curves over Prime Fields (P-curves)**
      i. **P-224**: secp224r1
      ii. **P-256**: secp256r1 (also called prime256v1)
      iii. **P-384**: secp384r1
      iv. **P-521**: secp521r1
   b. ✅ **Weierstrass Curves over Binary Fields (K/B-curves)**

i. **K-233**: sect233k1
ii. **B-233**: sect233r1
iii. **K-283**: sect283k1
iv. **B-283**: sect283r1
v. **K-409**: sect409k1
vi. **B-409**: sect409r1
vii. **K-571**: sect571k1
viii. **B-571**: sect571r1

**c.** ✅ **Newly Added Edwards Curves**
i. **Edwards25519** (for EdDSA)
ii. **Edwards448** (for EdDSA)

d. These curves are defined in:
i. **FIPS 186-5** (*Digital Signature Standard*)
ii. **NIST SP 800-186** (*Recommendations for Discrete Logarithm-based Cryptography: Elliptic Curve Domain Parameters*)[1].

**e. References**
i. [1] SP 800-186, Recommendations for Discrete Logarithm-based Cryptography ...

2) **Bitcoin:** Bitcoin uses the **secp256k1** elliptic curve for its cryptographic operations (ECDSA signatures and key generation). This curve is **not a NIST curve**; it is defined by the **Standards for Efficient Cryptography Group (SECG)** in:
a. **SEC 2: Recommended Elliptic Curve Domain Parameters (Version 2.0)**
b. Publisher: Certicom Research / SECG
c. URL: https://www.secg.org/sec2-v2.pdf
**d.** ✅ **Key Points:**
i. **Curve Name:** secp256k1
ii. **Equation:** y2≡x3+7(modp)$y2 \equiv x3+7 (\mathrm{mod} p)$
iii. **Field Prime:** p=2256−232−977$p=2256-232-977$
iv. **Base Point (G):** Defined in SEC 2
v. **Order (n):** Large prime (specified in SEC 2)
vi. **Cofactor:** 1

## Assignment Sections

The following are appliable to both the cipher algorithms.

A2. 1) For the curve secp256k1, determine the first 10 Points on the curve, i.e. P, 2P, 3P, …10P. What are the ($x_i$, $y_i$) for i = 1 to 10? (Basically, you would need to implement the point addition, point by point, i.e. derive 3P from 2P, 4P from 3P etc.,, and show with detailed logs, the information obtained and the output for each i. (10 points to implement the addition and 10 points for determining P to 10 P). Keep a subfolder – Assign2-Part1, under which you can place the code, log file. – **Total 20 Points**

A2. 2) For the curve secp256k1, determine the points 1201P, 3966P, 4207P. (Basically, you would need to implement the "Double and Add Algorithm", derive the binary values for the integers 1021, 3966, 4207 and show the steps for deriving the final values. (11 points to implement the algorithm and 9 points – 3 points each for the 3 point's values). Keep a subfolder – Assign2-Part2, under which you can place the code. **Total 20 Points**

A2. 3) Capture the logs for part 1 and part 2 (above two parts) as a single text file. This file is to be transferred to the Peers who are going to review your work, as well as the TA. Hence the file is to be digitally signed using ECDSA. Please implement ECDSA to enable the signing of your log file. Please use NIST defined – Weistrass curve : P-256: secp256r1 (also called prime256v1). (10 points to implement the ECSDSA signing support, and 10 points to verify the signature, 5 points for the detailed logs that will demonstrate the signature generated and signature verification captured in a second log file). Keep a subfolder – Assign2-Part3, under which you can place the code. Place the combined part1, part2 log as a single log file, and the log file for Part3 as a separate file. - **Total 25 Points**

A2. 4) Make a zip of the Part1, Part2, Part3. This zip file is to be sent encrypted to the peers who are going to review your work, as well as the TA. Please implement the ECDH + Symmetric Encryption (Slide 29 in the class slides). You would need to exchange information with your peer, i.e. public keys and derive the common secret – ECDH. The encryption is to be done using ChaCha, which you have already done in your first assignment. (10 points for ECDH implementation, and 5 points for encrypted output). - **Total 15 Points**

A2. 5) The peer who receives the encrypted file, must decrypt the zip file, and share back the decrypted log file, as well as log indicating the decryption process, and the verification of the signature. – **Total 10 points (You will do this for two teams. This is part of your class contribution effort, not part of Assignment).**

A2. 6) A report / read me covering aspects of 1 to 5 must be submitted to the moodle. Keep each part as a separate file. The read me file must have details how to execute your programs for each part. The TAs will validate this. Also in the Viva, as per the readme you will demonstrate your execution as well. **Total 10 Points.**

A2. 7) There will be a Viva for the assignment, where each step of the assignment has to be demonstrated before the TA as how it works, as well as answer questions. (**Total 10 Point**s). **VIVA is compulsory. If VIVA is not done, there will be no marks for the assignment. The TAs will schedule slots for the VIVA.**