Elliptic Curve Cryptosystems (ECC)

Cryptography – CS 507
Erkay Savas
Sabanci University
erkays@sabanciuniv.edu
Overview

- Emerging public key cryptography standard for constrained devices.
- 160 bit key length is equivalent in cryptographic strength to 1024 RSA. (313 bit ECC is equivalent to 4096 bit RSA)
- Elliptic curves as algebraic/geometric entities have been studied extensively for the past 150 years.
- Studies revealed a rich and deep theory suitable to cryptographic usage.
- First proposed for cryptographic usage in 1985 independently by Neal Koblitz and Victor Miller
Overview

• Many cryptosystems often require the use of algebraic groups.
• Elliptic curves may be used to form elliptic curve groups
• *Discrete Logarithm Problem* in Elliptic Curve groups is utilized.
• Elliptic curves received their name from their relation to *elliptic integrals*

\[
\int_{z_1}^{z_2} \frac{dx}{\sqrt{x^3 + ax + b}} \quad \text{and} \quad \int_{z_1}^{z_2} \frac{x\,dx}{\sqrt{x^3 + ax + b}}
\]

• That arise in the computation of the arc length of ellipses.
A Geometric Approach:

- Elliptic Curves over Real Space: $y^2 = x^3 + ax + b$
A Geometric Approach:

No intersection!?

Abstraction:
we say the line intersects with the curve at infinity.

Definition:
This intersection point is called point at infinity and denoted as $O$. 
Additive inverse

\( Q + P = O \Rightarrow Q = -P \Rightarrow -(x, y) = (x, -y) \)

It doesn’t intersect with the curve
Example

• Suppose an elliptic curve $E$ is defined as $y^2 = x^3 + 73$ and let $P = (2, 9)$ and $Q = (3, 10)$.
• The line $L$ through $P$ and $Q$ is $y = x + 7$
• Substituting into the elliptic equation for $E$ yields
  
  \[(x + 7)^2 = x^3 + 73\]
  
  which yields
  
  \[x^3 - x^2 - 14x + 24 = 0\]  \hspace{1cm} (1)
• We already know two roots, namely $x = 2$ and $x = 3$.
• Dividing (1) by $(x - 2)(x - 3)$ (= $x^2 - 5x + 6$) yields
  
  \[x + 4 \Rightarrow x = -4 \Rightarrow y = 3 \Rightarrow R = P + Q\]
• $R = (-4, -3)$
Example

• Let double the point \( R = (-4, -3) \) (add it to itself).
• The slope of the tangent line to \( E \) at \( R \) can be obtained by differentiating the equation for \( E \):
  \[
  2ydy = 3x^2dx \quad \Rightarrow \quad \frac{dy}{dx} = \frac{3x^2}{2y} = -8
  \]
• The tangent line is \( y = -8(x + 4) - 3 \)
• Substituting the line equation into the equation for \( E \)
  \[
  (-8(x + 4) - 3)^2 = x^3 + 73 \quad \text{which yields}
  \]
• The double root is \( x = -4 \)
• It follows the third root \( x = 72 \)
• \( S = 2R = (72, 611) \).
Addition Law

- The elliptic curve equation
  \[ E = y^2 = x^3 + ax + b \]
- Let \( P = (x_p, y_p) \) and \( Q = (x_q, y_q) \) be points on \( E \).
- Let \( R = P + Q = (x_r, y_r) \).
- The line \( L \) going through \( P \) and \( Q \) can be written as
  \[ y = \lambda x + \beta \]
  where \( \lambda = (y_q - y_p)/(x_q - x_p) \) (the slope)
  and \( \beta = y_p - \lambda x_p \), then
  \[ y = \lambda x + y_p - \lambda x_p \]
Addition Law

- If \( P \neq Q \),
  \[
  (\lambda x + \beta)^2 = x^3 + ax + b =>
  x^3 - \lambda^2 x^2 + (a-2\lambda\beta)x + b - \beta^2 = 0
  \]
- We know
  \[
  (x-x_p)(x-x_q)(x-x_r) = 0 =>
  x^3 - (x_p+x_q+x_r)x^2 + (x_p x_q + x_p x_r + x_q x_r)x + b - \beta^2 = 0
  \]
- Therefore,
  \[
  \lambda^2 = x_p + x_q + x_r => x_r = \lambda^2 - (x_p + x_q)
  
  -y_r = (-\lambda(x_r - x_p) + y_p) = \lambda(x_p - x_r) - y_p
  \]
Addition Law

- If $P = Q$, slope of the tangent can be calculated by differentiating the curve equation at $P$

  $$2y_px' = 3x^2_p + a \Rightarrow \lambda = (3x^2_p + a)/2y_p.$$ 

- Thus the tangent line is

  $$y = \lambda (x - x_p) + y_p$$

- $R = 2P = (x_r, y_r)$

- $$(x - x_p)^2(x - x_r) = 0 \Rightarrow$$
  $$x^3 - (x_r+2x_p)x^2+(2x_px_r+x^2_p)x + x_p^2x_r = 0$$

- $x_r = \lambda^2 + 2x_p$ and
  $$y_r = \lambda(x_r-x_p) + y_p.$$
Algebraic Approach:

• Elliptic Curves over Finite Fields
• Elliptic curves are defined over fields
• We are interested in elliptic curves over finite fields such as prime GF(p) or binary extension GF(2^n) fields.
• Points on an elliptic curve (along with the point at infinity) over finite fields form an additive group.
  – The operation is point addition given by addition law.
  – The # of points on the curve are finite and countable.
  – The point addition is closed, associative, etc.
• Group members are called (elliptic curve) points and represented by two coordinates: P = (x, y).
Elliptic Curve Group

• Point at infinity $O$ is the identity element in elliptic curve group. i.e. $P + O = P$ for any point on the curve.

• If the number of points (denoted as $r$) on the curve are equal to a prime integer, then we can find a generator point on the curve which generates all the elliptic curve points.
  – It is possible to describe the discrete logarithm on a curve.
  – If $r$ is not a prime number, it is always possible to find a subgroup of elliptic curve points whose order is prime.
  – In practice, $r$ is chosen to be a multiple of a large prime(e.g. $r = kn$ where $n$ is a large prime and $k$ is a smooth integer.)
Elliptic curves over finite fields

- The coordinates of a point \( P = (x, y) \) are the elements of the underlying field.
- In cryptography, coordinates are chosen from finite fields \( \text{GF}(q) \), where \( q = p^k \) and \( p \) is prime.
- When \( k = 1 \), \( \text{GF}(p) \) is classic *Residue Field*, where all the operations are modular. They are also called *Prime Fields*.
- When \( p=2 \), \( \text{GF}(2^k) \) is called *Binary Extension Fields* or *Galois Fields*, where operations are defined over polynomials.
Elliptic Curves over GF($p$)

- Solutions to $y^2 \equiv x^3 + ax + b \mod p$, where $0 \leq a, b < p$
  forms the Elliptic curve group. Each solution is called a Point on the Curve.
- Two Points: $P(x_p, y_p)$ and $Q(x_q, y_q)$, $0 \leq x_p, y_p, x_q, y_q < p$
- Point Addition Rule: $R(x_r, y_r) = P + Q$
  If $P \neq Q$
    $\lambda \equiv (y_p - y_q) / (x_p - x_q) \mod p$
  If $P = Q$
    $\lambda \equiv (3x_p^2 + a)/2y_p \mod p$
    $x_r \equiv \lambda^2 - x_p - x_q \mod p$
    $y_r \equiv -y_p + \lambda(x_p - x_r) \mod p$
Example

- $E$: $y^2 \equiv x^3 + 2x + 3 \mod 5$
- The points on $E$ are the pairs $(x, y)$ mod 5 that satisfies the equation, along with the \textit{point at infinity}.
- The possibilities for $x$ are $\{0, 1, 2, 3, 4\}$
  - $x \equiv 0 \Rightarrow y^2 \equiv 3 \mod 5 \Rightarrow \text{no solutions}$
  - $x \equiv 1 \Rightarrow y^2 \equiv 6 \equiv 1 \mod 5 \Rightarrow y \equiv 1, 4 \mod 5$
  - $x \equiv 2 \Rightarrow y^2 \equiv 15 \equiv 0 \mod 5 \Rightarrow y \equiv 0 \mod 5$
  - $x \equiv 3 \Rightarrow y^2 \equiv 36 \equiv 1 \mod 5 \Rightarrow y \equiv 1, 4 \mod 5$
  - $x \equiv 4 \Rightarrow y^2 \equiv 75 \equiv 0 \mod 5 \Rightarrow y \equiv 0 \mod 5$
- Therefore the points are $(1,1), (1,4), (2,0), (3,1), (3,4), (4,0)$
Example

• Let us compute \((1, 4) + (3, 1)\).
• The slope \(\lambda \equiv (1-4)/(3-1) \equiv 1 \mod 5\)
• The first coordinate
  \[ x_r \equiv \lambda^2 - x_p - x_q \mod p \equiv 1-1-3 \mod 5 \equiv 2 \mod 5 \]
  \[ y_r \equiv -y_p + \lambda(x_p - x_r) \mod p \equiv -4+(1-2) \equiv 0 \mod 5 \]
• The resulting point
  \((1, 4) + (3, 1) = (2, 0)\) is also on the curve (closed).
Elliptic Curves over GF($2^k$)

- Solutions to
  \[ y^2 + xy = x^3 + ax^2 + b, \quad \text{where } a, b \in \text{GF}(2^k) \]
  forms the Elliptic curve group. Each solution is called a Point on the Curve.
- Two Distinct Points: $P(x_p, y_p)$ and $Q(x_q, y_q)$ ($P \neq Q$)
  \[ x_p, y_p, x_q, y_q \in \text{GF}(2^k) \]
- Point Addition Rule: $R(x_r, y_r) = P + Q$
  \[ \lambda = (y_p + y_q) / (x_p + x_q) \]
  \[ x_r = \lambda^2 + \lambda + x_p + x_q + a \]
  \[ y_r = \lambda(x_p + x_r) + x_r + y_p \]
Elliptic Curves over GF(2^k)

- **Point Doubling Rule:** \( R(x_r, y_r) = 2P \)

\[
x_r = x_p^2 + \left( b / x_p^2 \right)
\]
\[
y_r = x_p^2 + \left( x_p + y_p/x_p \right)x_r + x_r
\]

- \((x, y) = (x, x+y)\) (i.e. \((x, y) + (x, x+y) = O\))
# of points on curve

- Generally, it is not easy to count the points on a curve.
- Assume that the *underlying field* $K$ (the field over which the elliptic curve is constructed) has $p$ elements.
- Then for the # of points $N$ on the curve $E$ defined over $K$, we can write

$$|N - p - 1| < 2\sqrt{p}$$

*Hasse bound* (1930s)
Example: # of points on curve

- *Previous example:* $E: y^2 ≡ x^3 + 2x + 3 \mod 5$
- The points on $E$ are the pairs $(x, y) \mod 5$ that satisfies the equation, along with the *point at infinity*.
- Therefore the points are $(1,1), (1,4), (2,0), (3,1), (3,4), (4,0)$, and point at infinity that can be represented as $O = (0, 0)$.
- Therefore, $\#E (N) = 7$

$$|N - p - 1| < 2\sqrt{p} \Rightarrow |7 - 5 - 1| = 1 < 4.472$$
Elliptic Curve DL Problem

- **Scalar Multiplication:**
  \[ Q = k \ P, \text{ where } P \text{ and } Q \text{ are points, } k \text{ is an integer} \]

  \[ kP = P + P + \cdots + P \]
  \[ \underbrace{\text{\(k\) times}} \]

- Scalar multiplication is repeated point addition

  **Example:** \( Q = 53P \quad 53 = (110101)_2 \)

  \[
  \begin{align*}
  Q & = P \\
  Q & = 2P + P = 3P \\
  Q & = 6P \\
  Q & = 12P + P = 13P \\
  Q & = 26P \\
  Q & = 52P + P = 53P
  \end{align*}
  \]
EC/DL Problem

• Given points $P$ and $Q$ in the group, find a number $k$ such that $Q = kP \Rightarrow$ VERY HARD PROBLEM!
• In elliptic curve schemes, the most time consuming operation is scalar multiplication.
• The security of the elliptic curve cryptosystems depends on the size of $k$.
• In real applications $k$ is large.
• The minimum bit length of $k$ is 160 for commercial applications.
An Overview of Operations in ECC

- Efficient field arithmetic
- Point Operations
- Finite Field operations
- Crypto graphic schemes

- Fast Signature & Small Implementation
- New techniques in curve arithmetic
- Fewer field operations
Finite Field Operations in ECC

- Solutions to
  \[ y^2 \equiv x^3 + ax + b \mod p, \] where \( 0 \leq a, b < p \)
  forms the Elliptic curve group over \( \text{GF}(p) \). Each solution is
called a Point on the Curve.

- **Two Points:** \( P(x_p, y_p) \) and \( Q(x_q, y_q) \) \( 0 \leq x_p, y_p, x_q, y_q < p \)
- **Point Addition Rule:** \( R(x_r, y_r) = P + Q \)
  If \( P \neq Q \)
  \[ \lambda \equiv (y_p - y_q) / (x_p - x_q) \mod p \]
  If \( P = Q \)
  \[ \lambda \equiv (3x_p^2 + a) / 2y_p \mod p \]
  \[ x_r \equiv \lambda^2 - x_p - x_q \mod p \]
  \[ y_r \equiv -y_p + \lambda(x_p - x_r) \mod p \]

1 inversion, 3 multiplications, and 5 adds or subtracts
Finite Field Operations in ECC

- Addition in GF(p) and GF($2^k$)
  - Inexpensive in terms of time and area
- Multiplicative inversion in GF(p) and GF($2^k$)
  - Prohibitively expensive in terms of time
  - Possible to avoid some of them
- Multiplication in GF(p) and GF($2^k$)
  - Expensive in terms of time and area
  - Most important operation
  - We have to perform it efficiently.
Projective Coordinates

- It is an efficient technique to eliminate inversion calculation from elliptic curve point operations.
- For example
  \[ E: y^2 + xy = x^3 + ax^2 + b, \]  
  where \( a, b \in \text{GF}(2^k) \) forms the Elliptic curve group. Each solution \((x, y)\) is called a \textit{Point on the Curve}.
- We can project the curve \(E\) to another plane in the three dimensional space
  \[ E\varsigma: Y^2 Z + X Y Z = X^3 + a X^2 Z + b Z^3, \]  
  where \( a, b \in \text{GF}(2^k) \) where the translation is achieved by \( x \rightarrow X/Z \) and \( y \rightarrow Y/Z \). e.g. \((x, y) \rightarrow (X, Y, 1)\).
Projective Coordinates

• Addition Formulae: \( R(x_r, y_r) = P + Q \) (where \( P \neq Q \))
\[
\lambda = \frac{y_p + y_q}{x_p + x_q}
\]
\[
x_r = \lambda^2 + \lambda + x_p + x_q + a
\]
\[
y_r = \lambda(x_p + x_r) + x_r + y_p
\]

• Addition Formulae for projective coordinates:
\( P = (X_p, Y_p, Z_p) \) and \( Q = (X_q, Y_q, Z_q) \)
\[
R(X_r, Y_r, Z_r) = P + Q \Rightarrow
\lambda = B / A \text{ where } B = Z_q Y_p + Z_p Y_q \text{ and } A = Z_q X_p + Z_p X_q
\]
\[
x_r = \frac{X_r}{Z_r} = \frac{B^2}{A^2} + \frac{B}{A} + \frac{A}{Z_p} + a
\]
Projective Coordinates

\[ \frac{X_r}{Z_r} = \frac{M}{A^2 Z_p} \quad \text{where} \quad M = B^2 Z_p + BAZ_p + A^3 Z_p + aA^2 Z_p \]

\[ y_r = \frac{Y_r}{Z_r} = \frac{A^2 BX_p + BM + AM + AY_p}{A^3 Z_p} \]

If we choose \( Z_r = A^3 Z_p \), then we have

\[ X_r = MA \]

\[ Y_r = A^2 BX_p + BM + AM + AY_p \]

No inversion!!

But many more multiplications
Arithmetic with projective Coordinates

- $Q = k \, P$ and $P = (x_p, y_p)$ (affine coordinates)
- Translation to projective coordinates
  $(x_p, y_p) \rightarrow P´ = (X_p, Y_p, Z_p) = (x_p, y_p, 1)$. (No cost)
- Scalar multiplication in projective coordinates
  $Q´ = k \, P´ = (X_q, Y_q, Z_q)$(with no inversion)
- Translation back to affine coordinates
  Compute $Z_q^{-1}$ and
  $(x_q, y_q) = (X_q \, Z_q^{-1}, Y_q \, Z_q^{-1})$

**Only one inversion!!**
**But many multiplications**
Elliptic Curve Cryptosystems

- Discrete logarithm problem (DLP) over elliptic curves is harder than the DLP over integers mod $p$.
- The most efficient method for computing DL, which is the Index Calculus method, seems to have no counterpart for elliptic curves.
- Therefore, it is possible to use much smaller primes or finite fields with elliptic curves to achieve the same level of security.
Elliptic Curve Cryptosystems

• The complexity of discrete logarithm algorithms
• **Index-calculus method:**
  – Minimum security requirement in $\mathbb{Z}_p^*$: $(p-1) > 2^{1024}$
• **Shanks’s algorithm (baby-step giant-step)**
  – Minimum security requirement: $(p-1) > 2^{160}$
• **Pohlig-Hellman algorithm:**
  – Minimum security requirement: $(p-1) > 2^{160}$
• This is why 160-bit ECDL is equivalent in cryptographic strength to 1024-bit DL.
Elliptic Curve Cryptosystems

- It is easy to change classical systems based on DL into one using elliptic curves:
  1. Change modular multiplication to elliptic curve point addition.
  2. Change modular exponentiation to multiplying an elliptic curve point by an integer (*scalar point multiplication*).
Elliptic Curve ElGamal Cryptosystems

- Classical ElGamal over GF($p$):
- Alice wants to send the message $x$ to Bob.
- Bob chooses a large prime $p$ and an integer $\alpha \mod p$.
- He also chooses a secret integer $a$ and computes $\beta \equiv \alpha^a \mod p$.
- Bob makes $p$, $\alpha$, and $\beta$ public, keeps $a$ secret.
- Alice chooses a random $k$ and computes $y_1 \equiv \alpha^k \mod p$ and $y_2 \equiv x \beta^k \mod p$
- She sends $(y_1, y_2)$ to Bob.
Elliptic Curve ElGamal Cryptosystems

- Upon receipt, Bob does the decryption by calculating
  \[ x \equiv y_2 y_1^{-a} \mod p \]
- Now we describe the elliptic curve version:
- Bob chooses an elliptic curve \( E \) over \( \text{GF}(p) \), where \( p \) is a large prime.
- He also chooses a point \( P \) on \( E: y^2 \equiv x^2 + ax + b \mod p \) and a secret integer \( s \).
- He computes
  \[ Q = sP = P + P + \cdots P \]
  \( s \) times
Elliptic Curve ElGamal Cryptosystems

- The elliptic curve points $P$ and $Q$ are public, while $s$ is kept confidential.
- Alice expresses her message as a point $M$ on $E$.
- She chooses a random integer $k$ and computes $S = kP$ and $T = M + kQ$.
- She sends elliptic curve points $S$ and $T$ to Bob.
- Upon receipt, Bob does the following calculation $M = T - sS = M + kQ - skP = M + kQ - kQ = M$. 
Example: EC ElGamal

- Bob chooses an elliptic curve over GF(8831) 
  \( E: y^2 \equiv x^3 + 3x + 45 \mod 8831 \) and a point on \( E \), 
  \( P = (4, 11) \).
- Alice has a message, represented as a point on \( E \), 
  \( M = (5, 1743) \) that she wishes to send to Bob.
- Bob chooses a secret integer \( s = 3 \) and computes and 
  publishes the point 
  \( Q = sP = 3 \times (4, 11) = (413, 1808) \).
- Alice downloads this and chooses a random integer 
  \( k = 8 \).
Example: EC ElGamal

- She sends Bob two points
  \[ S = kP = (5415, 6631) \] and \[ T = M + kQ = (5, 1743) + 8 \times (413, 1808) = (6626, 3576). \]

- He first calculates
  \[ sS = 3 \times (5415, 6631) = (673, 146). \]

- And does the following subtraction
  \[ T - sS = (6626, 3579) - (673, 146) = (6626, 3579) + (673, -146) = (5, 1743) = M \]
Embedding messages into elliptic curves

- In most cryptosystems, the messages are mapped into numerical value upon which we perform mathematical operations.
- In ECC, we need a method for mapping a message onto a point on an elliptic curve.
- The problem of encoding plaintext messages as points on an elliptic curve is not as simple.
- In particular, there is no known polynomial time, deterministic algorithm.
- However, there are fast probabilistic methods for finding points that can be used for encoding messages.
Embedding messages into elliptic curves

- *Koblitz’s method:*
- Let $E: y^2 \equiv x^3 + ax + b \pmod{p}$ be an elliptic curve.
- The message $m$ (already represented as an integer) will be embedded in the $x$-coordinate of a point.
- However, the probability is only $\frac{1}{2}$ that $m^2 + am + b$ is a square mod $p$.
- Let $K$ be a sufficiently large integer so that a failure rate of $1/2^K$ is acceptably low when trying to encode a message as a point.
Embedding messages into elliptic curves

- Assume that $m$ satisfies $(m+1)K < p$.
- The message will be represented by a number $x_j = mK+j$, where $0 \leq j < K$.
- For $j = 0, 1, \ldots, K-1$, compute $t_j \equiv x_j^3 + ax_j + b \pmod{p}$ and see if $t_j$ is a square mod $p$.
- If we can find a $t_j$ which is a square, calculate the square root and the point which encodes the message $m$ will be $(x_j, y)$.
- We repeat this process until we find a square.
- The probability that we cannot find a square in the sequence (i.e. the probability that we cannot encode our message as a point on the curve) is $2^{-K}$
Recover the message from a point

- Let the point \( P = (x, y) \) represent the message \( m \).
- To recover the message is easy
  \[ m = \left\lfloor \frac{x}{K} \right\rfloor. \]
- **Example:** \( E: y^2 \equiv x^3 + 2x + 7 \pmod{179} \).
  Failure rate is \( 2^{-10} \), then \( K = 10 \).
  Since \( mK + K < 179 \Rightarrow 0 \leq m < 17 \).
  Suppose \( m = 5 \).
  \[ x = 50 + j, \text{ where } 0 \leq j < 10 \Rightarrow \]
  possible choices for \( x \) are 50, 51, 52, …, 59.
  For \( x = 51 \),
  we get \( 51^2 + 2 \times 51 + 7 \pmod{179} \equiv 121 \) which has 11 as a square root. Therefore, \( m \rightarrow (51, 11) \)
ECDH Key Exchange

- Alice and Bob want to exchange a key.
- They agree on a public basepoint $P$ on an elliptic curve $E: y^2 \equiv x^3 + ax + b \pmod{p}$.
- They both choose private keys, $s_A$ and $s_B$ and compute the following:
  - Alice: $Q_A = s_A P$
  - Bob: $Q_B = s_B P$
- Alice publishes $Q_A$ and keeps $s_A$ secret.
- Bob publishes $Q_B$ and keeps $s_B$ secret.
ECDH Key Exchange

- Alice downloads $Q_B$ and computes $s_AQ_B$.
- Bob downloads $Q_A$ and computes $s_BQ_A$.
- In fact, $s_AQ_B = s_BQ_A = s_As_BP$.
- Example: $E : y^2 \equiv x^3 + x + 7206 \pmod{7211}$ and a base point $P = (3, 5)$.
  
  $s_A = 12$ and $s_B = 23$
  
  $Q_A = s_A P = 12(3,5) = (1794, 6375)$.
  
  $Q_B = s_B P = 23(3,5) = (3861, 1242)$
  
  $s_AQ_B = s_BQ_A = 12(3861, 1242) = 23(1794, 6375) = (1472, 2098)$. 
EC Digital Signature Algorithm (ECDSA)

- Alice wants to sign a message $m$.
- She chooses an elliptic curve $E$ over GF($p$) and a point $P$ on the curve.
- The number of points $n$ on $E$ is known and assume $n$ also a prime integer and base point is a generator.
- She chooses a secret integer $s_A < n$ and computes $Q_A = s_A P$.
- Curve parameters (i.e. $a$, $b$, $p$, and $P$) and her public key $Q_A$ are published while $s_A$ is kept private.
ECDSA

- **Signing the message** $m$:
  1. She selects a random integer $k$ s.t. $0 < k < n$.
  2. Computes $r = kP$
  3. Computes $s = k^{-1} (m + s_A r)$
  4. Alice’s signature for $m$ is $(r, s)$.

- **Verifying the signature** given $m$ and $Q_A$ and curve parameters
  1. $u_1 = s^{-1} m$ and $u_2 = s^{-1} r$
  2. $R = u_1 P + u_2 Q_A$
  3. Accepts if $x_r = r$. 