Elliptic Curve Cryptography

Joel Allardyce
Nitesh Goyal
April 15, 2004
Outline

- What is Elliptic Curve Cryptography?
- Necessity and Advantages
- Arithmetic of ECC
  - Number Theory
    - Modular Arithmetic
    - Arithmetic mod Irreducible Polynomials
    - Galois Fields
  - The Arithmetic of Elliptic Curves
    - Addition
    - Scalar Multiplication
- Elliptic Curve Cryptography
  - ECC Analogues
  - Menezes-Vanstone ECC
- Conclusion
What is Elliptic Curve Cryptography?

- ECC proposed an alternative to other public-key encryption algorithms, such as RSA.
- All ECC schemes are public key, and are based on the difficulty in solving the discrete log problem for elliptic curves.
Necessity and Advantages

- Compared to RSA, ECC systems have a smaller key size for an equivalent amount of security.
  - Leads to fewer necessary operations, faster encryption time, and fewer transistors for hardware implementation
  - For example: 155-bit ECC uses 11,000 transistors while a 512-bit RSA implementation uses 50,000. These are considered to be of equivalent security. [2]

- Thus, ECC devices require less storage, less power, less memory, and often less bandwidth than other public key systems.

- This might or might not continue to be the case.
Necessity and Advantages (Cont.)

- Current key-size recommended by NIST for legacy public schemes is 2048 bits.
- A vastly smaller 224-bit ECC key offers the same level of security.
- This advantage only increases with security level— for example, a 3072 bit legacy key and a 256 bit ECC key are equivalent [8].
**Necessity and Advantages (Cont.)**

**Figure 1:** NIST guidelines for public key sizes for AES (from [8]).

<table>
<thead>
<tr>
<th>ECC KEY SIZE (Bits)</th>
<th>RSA KEY SIZE (Bits)</th>
<th>KEY SIZE RATIO</th>
<th>AES KEY SIZE (Bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>163</td>
<td>1024</td>
<td>1 : 6</td>
<td>128</td>
</tr>
<tr>
<td>256</td>
<td>3072</td>
<td>1 : 12</td>
<td>192</td>
</tr>
<tr>
<td>384</td>
<td>7680</td>
<td>1 : 20</td>
<td></td>
</tr>
<tr>
<td>512</td>
<td>15,360</td>
<td>1 : 30</td>
<td>256</td>
</tr>
</tbody>
</table>
RSA vs ECC

Figure 2: From [8].
Modular Arithmetic

- Familiar to every computer scientist.
- Modulus operation – returns the remainder after integer division.
- Creates equivalency classes:
  - $5 \mod 3 = 2 \mod 3$
    - Because $5 / 3 = 1$ with a remainder of 2
  - Equivalence class of $2 \mod 3$: $\{\ldots, -1, 2, 5, 8, 11, \ldots\}$
Modular Arithmetic (Cont.)

- Operations in Modular Arithmetic reduced with modulus.
  - $6 + 8 \mod 5 = 14 \mod 5 = 4 \mod 5$

- Operations in Modular Arithmetic can be simplified
  - Simpler to first reduce the operands.
  - $6 + 8 \mod 5 = 1 + 3 \mod 5 = 4 \mod 5$

- Similar method used for multiplication
  - $4 \times 5 \mod 11 = 20 \mod 11 = 9 \mod 11$
Modular Arithmetic (Cont.)

- Subtraction is addition of negation
  - \(4 - 5 \mod 7 = 4 + (-5) \mod 7 = 4 + 2 \mod 7 = 6 \mod 7\)

- Division is multiplication of inverse
  - Note: \(4 \cdot 3 \mod 11 = 1 \mod 11\)
  - \(5 / 4 \mod 11 = 5 \cdot 3 \mod 11 = 15 \mod 11 = 4 \mod 11\)
  - Find the inverse by the Euclidian Algorithm (also finds greatest common denominator)
Arithmetic mod Irreducible Polynomials

- Particularly, we are interested in irreducible polynomials with coefficients mod 2.
- Example:
  - $5x^2 + 2x + 3 = 1x^2 + 0x + 1 = x^2 + 1$
  - Represent by a binary coefficient array: $x^2 + 1 = 101$
  - $x^2 + 1$ is irreducible.
- Other 2\textsuperscript{nd} order irreducible polynomials with coefficients mod 2:
  - 111 is the only other one
  - For lower order, also includes 1, 10, 11
  - Notice that the binary representations are all prime numbers.
Arithmetic mod Irreducible Polynomials (Cont.)

- Addition of these polynomials is XOR
  - \((x^2 + 1) + (x^3 + x^2 + x) = (x^3 + x + 1)\)
  - e.g. 0101 + 1110 = 1011
  - Note: This means that addition is subtraction

- Multiplication
  - 0101 * 1110 = 0000
  
  \[
  \begin{align*}
  &\phantom{0000}1110 \\
  &0000 \\
  &1110 \\
  &\text{-------------} \\
  &0110110
  \end{align*}
  \]
Arithmetic mod Irreducible Polynomials (Cont.)

- Division

\[
\begin{array}{c}
101 \\
- 1110 | 110110 \\
\underline{1110} \\
001110 \\
\underline{1110} \\
0000 \leftarrow \text{Remainder}
\end{array}
\]
Arithmetic mod Irreducible Polynomials (Cont.)

- So now, the arithmetic:
  - $101 \times 111 \mod 1011 = 11011 \mod 1011$
  - $11011 / 1011 = 11$ with a remainder of 110
  - So, $101 \times 111 \mod 1011 = 110 \mod 1011$

- There is also a version of the Euclidian Algorithm for Irreducible Polynomials, so inverses and greatest common denominator's can be found.
Galois Fields

- What is a field?
- A field is a group of numbers on which addition and multiplication are defined, and which follow the “ordinary” rules:
  - These rules are [3]:
    - Additive Commutativity: \( a + b = b + a \)
    - Multiplicative Commutativity: \( a \cdot b = b \cdot a \)
    - Additive Associativity: \( a + (b + c) = (a + b) + c \)
    - Multiplicative Associativity: \( a \cdot (b \cdot c) = (a \cdot b) \cdot c \)
    - Distributive: \( a \cdot (b + c) = (a \cdot b) + (a \cdot c) \)
    - Additive Identity: \( a + 0 = a \)
    - Multiplicative Identity: \( a \cdot 1 = a \)
    - Additive Negation: \( a - a = 0 \)
    - Multiplicative Inversion: \( \frac{a}{a} = 1 \) (for a nonzero)
Galois Fields (Cont.)

- Galois fields only exist of size $p^n$, where $p$ is prime, and $n$ is a natural number.
- When $n = 1$ (i.e. prime sized field), all arithmetic is modular, with $p$ the modulus.
- When $n > 1$ (i.e. prime power sized field), arithmetic is never modular.
  - It is arithmetic of polynomials with coefficients mod $p$, mod an irreducible polynomial of order $n$. 
### Galois Fields (Cont.)

#### GF(5)

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>-</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>3</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>/</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>.</td>
<td>3</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>.</td>
<td>4</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>1</td>
<td>.</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>3</td>
<td>1</td>
<td>.</td>
</tr>
</tbody>
</table>
Galois Fields (Cont.)

**GF($2^2$) or GF(4)**

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>*</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>/</th>
<th>0</th>
<th>.</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>.</td>
<td>.</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1</td>
<td>3</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Elliptic Curves

- **What is an Elliptic Curve?** [1]
  - It is called “elliptic” because of its relationship with elliptic integrals, which are natural expressions for the arc length of an ellipse.
  - A better name might be an Abelian variety of dimension one.
- **How old are they?**
  - They have been around since the 19th century, and were first looked at by Abel, Gauss, Jacobi and Legendre.
  - More recently they were used by Andrew Wiles as part of his solution to Fermat’s Last Theorem.
- **Uses** include factoring integers, primality proving, and of course cryptography.
One important side note [1]:
- The following equations all assume that the field being worked in has a characteristic greater than 3.
- The characteristic of a field is the least positive integer \( n \) such that:
  \[
  \sum_{i=1}^{n} 1 = 0
  \]
  For \( GF(p^k) \), \( n = p \)
- If there is no \( n \) for which this is the case, a field is said to have a characteristic of 0.

If this is not the case, then a different set of equations must be used. We will not enumerate those equation here.
Elliptic Curves (Cont.)

- What do they look like?
  - They are typically represented by the Diophantine equation:
    \[ y^2 = x^3 + ax + b. \]
  - The image to the right represents the curve:
    \[ y^2 = x^3 - 7x. \]
  
  It is defined over the Real coordinate plane. Even though it separates into two parts, it is defined by one equation.
  
  - It also demonstrates addition over this curve (more on that soon)

**Figure 3:** Geometric composition laws of an elliptic curve (from [4]).
Elliptic Curves (Cont.)

- With the addition of an identity element $O_E$ which is called the “point at infinity”, elliptic curves form an Abelian group over addition [1].
  - A group over an operation:
    - Has associativity
    - Is closed
    - Has an identity element
    - Has inverses
  - An Abelian group
    - Adds commutativity (i.e. $a + b = b + a$)
    - Sometimes called a commutative group

- There are two operations over Elliptic curves:
  - Addition (well defined)
  - Scalar multiplication (actually just multiple additions).
Addition on Elliptic Curves

- First, the ground rules. Let $E$ be the points on an elliptic curve defined over the field $\mathbb{F}_2$, with the addition of the point $O_E[1]$.
  - All lines in $\mathbb{F}_2$ intersect $E$ in three places.
  - Lines at infinity intersect $E$ at $O_E$ three times.
  - Vertical lines intersect $E$ at two places, and at $O_E$.
- Addition occurs as follows [1]. Let $A, B$ be in $E$.
  - First, draw a line between $A$ and $B$.
  - Where $A$ and $B$ intersect $E$ for the third time, draw a vertical line.
  - $A + B$ is where this vertical line intersects $E$ a second time.
The general algorithm for addition is[1]:
- Given E: $y^2 = x^3 + ax + b$, $P_1 = (x_1, y_1)$, $P_2 = (x_2, y_2)$, both on E

\[
P_1 + P_2 = \begin{cases} 
O_E & \text{if } x_1 = x_2 \& y_1 = -y_2 \\
(x_3, y_3) & \text{otherwise}
\end{cases}
\]

where

\[
(x_3, y_3) = \left( \lambda^2 - x_1 - x_2, \lambda(x_1 - x_3) - y_1 \right)
\]

and

\[
\lambda = \begin{cases} 
\frac{3x_1^2 + a}{2y_1} & \text{if } P_1 = P_2 \\
\frac{y_2 - y_1}{x_2 - x_1} & \text{otherwise}
\end{cases}
\]
Scalar Multiplication on Elliptic Curves

- Scalar Multiplication defined as repeated additions.
  - Given Elliptic Curve $E$, point $P$ in $E$, and scalar $k$.
  - $kP = P + P + P + ... \times k$ times.

- This can be simplified by dividing it into two operations:
  - Double
  - Add $P$
Scalar Multiplication on Elliptic Curves (Cont.)

- The simplified scalar multiplication algorithm[1]:
  - Given $E$, $P$, and $k$ as before, and variable $e$
  - Step 1: Write $k$ in binary form, let $e = 0$
  - Step 2: Starting at highest order bit of $k$:
    - Step 2.1: if bit = 0, double $e$.
    - Step 2.2: else if bit = 1, double $e$ then add $P$.
    - Step 2.3: repeat 2.1 to 2.3 for each bit in $k$
  - Step 3: Return $e$
Elliptic Curve Cryptography

- One-way trapdoor functions are the basis of public key cryptosystems.
  - In ECC, scalar multiplication is the one way trapdoor function.
- All ECC schemes are public key, and are based on the difficulty in solving the discreet log problem for elliptic curves
  - Given $A = kP$, what is $k$?
- All operations are performed over a Galois Field.
  - So, results of $kP$ seem rather “random”
- There are analogues of most public key systems that use Elliptic Curves
  - e.g. Diffie-Hellman, RSA, etc.
  - Difficulty is that no deterministic method is known for encoding a message into a point on an elliptic curve.
ECC Analogues

- In general, exponentiation over $\text{GF}(p^n)$ is replaced by scalar multiplication of an elliptic curve over $\text{GF}(p^n)$.
  - As mentioned before, the drawback is that there is no known deterministic way of finding a point on an elliptic curve to match a message one wants to hide.
  - Even so, once such a point is found the necessary operations are no more difficult than exponentiation.
  - Of course, this drawback also does not apply to key exchange systems, where symmetric key systems are applied afterwards.
ECC Analogues (Cont.)

- For example, in Diffie-Hellman:
  - Before:
    - Alice and Bob each chose random integers $a$ and $b$, and selected a field $GF(p^r)$ with generator $g$.
    - They each calculated $g^a$ and $g^b$ and exchanged these values publicly.
    - They each then found their shared private key by calculating $(g^a)^b$ and $(g^b)^a$.
  - Using ECs:
    - Alice and Bob choose an elliptic curve $E$ over $GF(p^r)$ with a base point $P$. Once again, they choose random $a$ and $b$.
    - They calculate $aP$ and $bP$, and exchange these values publicly.
    - The shared public key is calculated by $b(aP)$ and $a(bP)$.

- Advantage here is that once a key is established a symmetric key method is used.
A similar method is used for the RSA analogue.
  - Unfortunately, this does suffer from the difficulty in encoding a message in a point.

Let us now look at a cryptosystem that attempts to solve the point encoding problem, the Menezes-Vanstone Elliptic Curve Cryptosystem.
Menezes-Vanstone Elliptic Curve Cryptosystem

- The solution to the problem of encoding a message in a point is the Menezes-Vanstone Elliptic Curve Cryptosystem. It was initially proposed in [7].
  - It uses a point on an elliptic curve to “mask” a point in the plane.
  - Works over GF(p), with p prime and p > 3, so our previous algorithms work nicely.
  - It is fast and simple.

- One major drawback.
  - Due to point overhead, encrypted messages are doubled in length.
Menezes-Vanstone Elliptic Curve Cryptosystem (Cont.)

- **Purpose:** Alice wants to send a message to Bob using his public key.
- **Given:** Alice and Bob have decided upon the following conventions, all of which are public.
  - $p$ – A large prime number (it must at least be larger than 3)
  - $F_p$ – A Galois field of size $p$ ($p$ is prime, so it works like modular arithmetic)
  - $E$ – An elliptic curve over $F_p$ of the form $y^2 = x^3 + ax + b$ ($a,b$ in $F_p$)
  - $P$ – A randomly selected point on $E$ (called the base point) that will generate subgroup $H$
  - $H$ – A subgroup of $E$ that is preferably of the same size as $E$
Menezes-Vanstone Elliptic Curve Cryptosystem (Cont.)

- **Private Key:** Bob's private key. Only he knows it.
  - \( a \): Bob's private key is a randomly selected natural number.

- **Public Key:** Bob's public key. Ideally it is distributed to the world.
  - \( \beta \): Bob's public key is calculated as \( \beta = aP \). It is a point in \( H \).

- **Secret:** In this scheme, Alice also has a secret.
  - \( k \): Randomly selected by Alice. It is usually different each time a message is sent.
Menezes-Vanstone Elliptic Curve Cryptosystem (Cont.)

**Encryption:** Alice has secret $m$, which she splits up into $m_1$ and $m_2$

1. Alice calculates $(y_1, y_2) = k\beta$.
2. Alice calculates $c_0 = kP$. ← Note that $c_0$ is a point.
3. Alice calculates $c_1 = y_1m_1 \mod p$.
4. Alice calculates $c_2 = y_2m_2 \mod p$.
5. Alice sends encrypted message $c = (c_0, c_1, c_2)$ to Bob.
   Note that $c$ is twice as large as the original message $m$.

**Decryption:** Bob wants to get back the message $m$ from $c$.

1. Bob calculates $ac_0 = (y_1, y_2)$
2. Bob retrieves message $m$ by calculating $m = (c_1y_1^{-1} \mod p, c_2y_2^{-1} \mod p)$
Menezes-Vanstone Elliptic Curve Cryptosystem (Cont.)

- Why does it work?
  - When Alice sends $c = (c_0, c_1, c_2)$ to Bob, he is able to get $(y_1, y_2)$ because:
    - $(y_1, y_2) = k\beta = kaP = akP = ac_0$
    - Notice that this does not really matter what $k$ is.
  - Bob is then able to retrieve $m = (m_1, m_2)$ because:
    - $(c_1, c_2) = (y_1 m_1, y_2 m_2) \mod p$
    - $(c_1 y_1^{-1}, c_2 y_2^{-1}) \mod p = (y_1^{-1} y_1 m_1, y_2^{-1} y_2 m_2) \mod p$
      - $= (m_1, m_2)$

- An eavesdropper in the middle only sees $c$, which without $a$ is not enough.
Conclusion

- Encryption based on Elliptic Curves provides a framework for the continued use of public key systems.
- ECC systems currently have better security density than other public key schemes.
- There is a trade-off when selecting an ECC system for use
  - Available bandwidth vs. ease of message encoding.
Conclusion (Cont.)

- Most importantly…

Elliptic Curve Math is FUN!!!
Questions?
Sources

Some Fun Stuff

- An interesting [web site](#) we found. Has applets that allow one to try out various systems.
  - The applets:
    - Elliptic Curves
    - ElGamal over EC
    - Menezes-Vanstone ECC