HW/SW Co-design of Elliptic Curve Cryptography on 8051

B07901010范詠為 B07901013林子軒

Brief Abstract

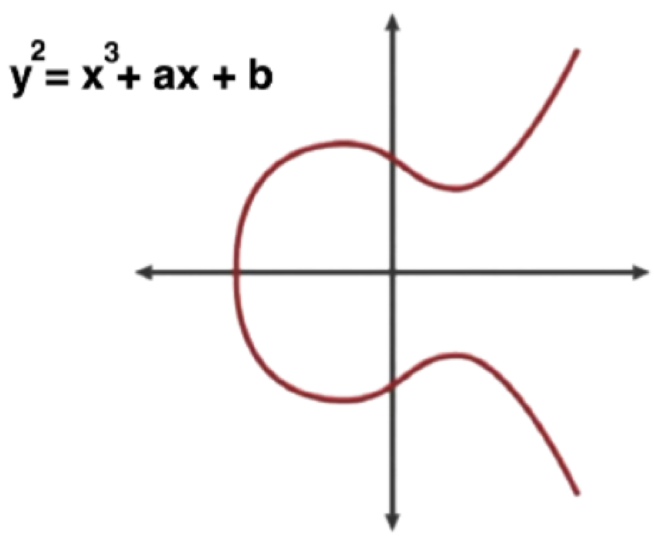
In the field of Information Security, Cryptography is one of the many ways to secure the insecure information channels. In 1985, Neal Koblitz and Victor Miller independently introduced elliptic curve cryptography.

ECC uses the set of points on an elliptic curve along with an addition rule. The unique mathematical structure of the points with the addition rule enables us to perform encryption and decryption of plaintexts. Another reason that supports the feasibility of ECC is the fact that it uses significantly smaller key sizes than the RSA Cryptosystem.

	RSA	ECC
Key Size (security 280)	1024-bits	160-bits
Pros	easy implementation	fast, smaller key size
Cons	slow, longer key size	more complicated

We will try to implement the ECIES (Elliptic Curve Integrated Encryption Scheme) algorithm. Since the performance of 8-bit microcontrollers is often too poor for the implementation of public-key cryptography in software, we will design a hardware accelerator for ECC on an 8051 microcontroller

Algorithm



Algorithm 2 Point doubling

Input: point $P(x_1, y_1)$.
Output: point $2P$.
if $P = -P$ or $P = \mathcal{O}$ **then**
  return $\mathcal{O}$
else
  $\lambda \leftarrow x_1 \oplus y_1 / x_1$
  $X \leftarrow \lambda^2 \oplus \lambda \oplus a$
  $Y \leftarrow x_1^2 \oplus \lambda \cdot X \oplus X$
  return (X, Y)
end if

Algorithm 3 Point addition

Input: points $P(x_1, y_1)$, $Q(x_2, y_2)$.
Output: point $P + Q$.
if $P \neq Q$ **then**
  if $P = -Q$ **then**
    return $\mathcal{O}$
  else if $P = \mathcal{O}$ **then**
    return Q
  else if $Q = \mathcal{O}$ **then**
    return P
  else
    $\lambda \leftarrow (y_2 \oplus y_1)/(x_2 \oplus x_1)$
    $X \leftarrow \lambda^2 \oplus \lambda \oplus x_1 \oplus x_2 \oplus a$
    $Y \leftarrow \lambda \cdot (x_1 \oplus X) \oplus X \oplus y_1$
    return (X, Y)
  end if
else
  return $2P$
end if

Algorithm 4 Scalar multiple

Input: point P, integer n.
Output: point nP.
$A \leftarrow P$
$R \leftarrow \mathcal{O}$
while $n > 0$ **do**
  if $n \equiv 1 \bmod 2$ **then**
    $R \leftarrow R + A$
  end if
  $n \leftarrow n \gg 1$
  $A \leftarrow 2A$
end while
return R

Algorithm 5 Encryption with the simplified ECIES

Input: plaintext x.
Output: ciphertext $(U(x_1, y_1), y)$.
for $char \in x$ **do**
  $char \leftarrow$ BinaryASCII($char$)
  $char \leftarrow$ Padding($char$)
  Append(x', $char$)
end for
$blocklength \leftarrow \lfloor N/7 \rfloor$
$x' \leftarrow$ Block(x', $blocklength$)
$k \leftarrow$ Random($[1, n-1]$)
$U(x_1, y_1) \leftarrow kP$
$V(x_2, y_2) \leftarrow kQ$
for $char \in x'$ **do**
  $cipher \leftarrow char \cdot x_2$
  Append(y, $cipher$)
end for
return (U, y)

References

1. Implementation of Elliptic Curve Cryptography in Binary Field
   - https://iopscience.iop.org/article/10.1088/1742-6596/710/1/012022/pdf
2. 非對稱式加密演算法 - 橢圓曲線密碼學 Elliptic Curve Cryptography , ECC (觀念篇)
   - https://ithelp.ithome.com.tw/articles/10251031
3. Hardware/Software Co-design of Elliptic Curve Cryptography on an 8051 Microcontroller
   - https://www.iacr.org/archive/ches2006/34/34.pdf