Other Public-Key Cryptosystems

Raj Jain
Washington University in Saint Louis
Saint Louis, MO 63130
Jain@cse.wustl.edu

Audio/Video recordings of this lecture are available at:
http://www.cse.wustl.edu/~jain/cse571-17/
Overview

1. How to exchange keys in public? (Diffie-Hellman Key Exchange)
2. ElGamal Cryptosystem
3. Elliptic Curve Arithmetic
4. Elliptic Curve Cryptography
5. Pseudorandom Number Generation using Asymmetric Cipher

Diffie-Hellman Key Agreement

- Allows two parties to agree on a secret key using a public channel.
- A selects q = large prime, and \( \alpha \) = a primitive root of q.
- A selects a random \( X_A \), B selects another random \( X_B \).

\[
\begin{align*}
X_A, \alpha, q \\
Y_A &= \alpha^{X_A} \mod q \\
Y_{AB} &= Y_B^{X_A} \mod q \\
Y_B &= \alpha^{X_B} \mod q \\
Y_{AB} &= Y_A^{X_B} \mod q
\end{align*}
\]

- Eavesdropper can see \( Y_A, \alpha, q \) but cannot compute \( X_A \).
- Computing \( X_A \) requires discrete logarithm - a difficult problem.
Diffie-Hellman (Cont)

- **Example:** $\alpha=5, q=19$
  - A selects 6 and sends $5^6 \mod 19 = 7$
  - B selects 7 and sends $5^7 \mod 19 = 16$
  - A computes $K = 16^6 \mod 19 = 7$
  - B computes $K = 7^7 \mod 19 = 7$

- Preferably $(q-1)/2$ should also be a prime.
- Such primes are called safe prime.
Man-in-Middle Attack on Diffie-Hellman

- Diffie-Hellman does not provide authentication

 Alice  
\[ \alpha^{xa} = 8389 \]  
\[ K_{AD} = 5876^{xa} \]  

 Darth  
\[ \alpha^{xd} = 5876 \]  
\[ K_{AD} = 8389^{xd} \]  
\[ K_{DB} = 9267^{xd} \]  

 Bob  
\[ \alpha^{xb} = 9267 \]  
\[ K_{DB} = 5876^{xb} \]  

- X can then intercept, decrypt, re-encrypt, forward all messages between Alice & Bob
- You can use RSA authentication and other alternatives
ElGamal Cryptography

- Public-key cryptosystem related to D-H
- Uses exponentiation in a finite (Galois)
- Security based difficulty of computing discrete logarithms
- $X_A$ is the private key, $\{\alpha, q, Y_A\}$ is the public key

$$\begin{align*}
X_A, \alpha, q \\
Y_A &= \alpha^{X_A} \mod q \\
K &= (C_1)^{X_A} \\
M &= C_2 K^{-1} \mod q
\end{align*}$$

Select random key $k$

- $K = (Y_A)^k \mod q$
- $C_1 = \alpha^k \mod q$
- $C_2 = KM \mod q$
- $M = \text{Message}$

- $k$ must be unique each time. Otherwise insecure.

ElGamal Cryptography Example

- Use field GF(19) \( q=19 \) and \( \alpha=10 \)
- Alice chooses \( x_A=5 \),
- Bob wants to send message \( M=17 \), selects a random key \( k=6 \)

\[
\begin{align*}
X_A &= 5, \quad \alpha=10, \quad q=19 \\
Y_A &= \alpha^{X_A} \mod q \\
&= 10^5 \mod 19 = 3 \\
K &= (C_1)^{X_A} \\
&= 11^5 \mod 19 = 7 \\
K^{-1} &= 7^{-1} = 11 \\
M &= C_2 K^{-1} \mod q \\
&= 5 \times 11 \mod 19 = 17
\end{align*}
\]

Select random key \( k=6 \)

\[
\begin{align*}
\alpha &= 10, \quad q=19, \quad Y_A=3 \\
K &= (Y_A)^k \mod q \\
&= 3^6 \mod 19 = 7 \\
C_1 &= \alpha^k \mod q \\
&= 10^6 \mod 19 = 11 \\
C_2 &= KM \mod q \\
&= 7 \times 17 \mod 19 = 5
\end{align*}
\]
Elliptic Curve Cryptography

- Majority of public-key crypto (RSA, D-H) use either integer or polynomial arithmetic with very large numbers/polynomials
- Imposes a significant load in storing and processing keys and messages
- An alternative is to use elliptic curves
- Offers same security with smaller bit sizes
- Newer, but not as well analyzed
Elliptic Curves over Real Numbers

- An elliptic curve is defined by an equation in two variables $x$ & $y$,
  - $y^2 = x^3 + ax + b$
  - Where $x, y, a, b$ are all real numbers
  - $4a^3+27b^2 \neq 0$

- The set of points $E(a, b)$ forms an abelian group with respect to “addition” operation defined as follows:
  - $P+Q$ is reflection of the intersection $R$
  - $O$ (Infinity) acts as additive identity
  - To double a point $P$, find intersection of tangent and curve
  - Closure: $P+Q \in E$
  - Associativity: $P+(Q+R) = (P+Q)+R$
  - Identity: $P+O=P$
  - Inverse: $-P \in E$
  - Commutative: $P+Q = Q+P$
Elliptic Curve over Real Numbers (Cont)

- Slope of line PQ is:
  \[ \Delta = \frac{y_Q - y_P}{x_Q - x_P} \]

- The sum \( R = P + Q \) is:
  \[ x_R = \Delta^2 - x_P - x_Q \]
  \[ y_R = -y_P + \Delta(x_P - x_R) \]

- \( P + P = 2P = R \)

\[
x_R = \left( \frac{3x_P^2 + a}{2y_P} \right)^2 - 2x_P
\]
\[
y_R = \left( \frac{3x_P^2 + a}{2y_P} \right) (x_P - x_R) - y_P
\]
Finite Elliptic Curves

- Elliptic curve cryptography uses curves whose variables & coefficients are defined over GF
  - **Prime curves**: $E_p(a,b)$ defined over $\mathbb{Z}_p$
    - Use integers modulo a prime
    - Easily implemented in software
  - **Binary curves**: $E_{2^m}(a,b)$ defined over GF($2^n$)
    - Use polynomials with binary coefficients
    - Easily implemented in hardware

- Cryptography: Addition in elliptic = multiplication in Integer
  - Repeated addition = Exponentiation
  - Easy to compute $Q = P + P + \ldots + P = kP$, where $Q, P \in E$
  - Hard to find $k$ given $Q, P$ (Similar to discrete log)
Finite Elliptic Curve Example

- $\mathbb{E}_p(a,b): y^2 = x^3 + ax + b \mod p$
- $\mathbb{E}_{23}(1,1): y^2 = x^3 + x + 1 \mod 23$
- Consider only +ve $x$ and $y$
- $R = P + Q$
  - $x_R = (\lambda^2 - x_P - x_Q) \mod p$
  - $y_R = (\lambda(x_P - x_R) - y_P) \mod p$
  - Where
    - $\lambda = \begin{cases} 
      \frac{y_Q - y_P}{x_Q - x_P} \mod p & \text{if } P \neq Q \\
      \frac{3x_P^2 + a}{2y_P} \mod p & \text{if } P = Q 
    \end{cases}$
- Example: $(3,10) + (3,10)$
  - $\lambda = \left( \frac{3(3^2) + 1}{2 \times 10} \right) \mod 23 = \frac{1}{4} \mod 23 = 6$
  - $x_R = (6^2 - 3 - 3) \mod 23 = 7$
  - $y_R = (6(3 - 7) - 10) \mod 23 = 12$

Table 10.1 Points on the Elliptic Curve $\mathbb{E}_{23}(1,1)$

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>$(0, 1)$</td>
<td>$(6, 4)$</td>
<td>$(12, 19)$</td>
</tr>
<tr>
<td>$(0, 22)$</td>
<td>$(6, 19)$</td>
<td>$(13, 7)$</td>
</tr>
<tr>
<td>$(1, 7)$</td>
<td>$(7, 11)$</td>
<td>$(13, 16)$</td>
</tr>
<tr>
<td>$(1, 16)$</td>
<td>$(7, 12)$</td>
<td>$(17, 3)$</td>
</tr>
<tr>
<td>$(3, 10)$</td>
<td>$(9, 7)$</td>
<td>$(17, 20)$</td>
</tr>
<tr>
<td>$(3, 13)$</td>
<td>$(9, 16)$</td>
<td>$(18, 3)$</td>
</tr>
<tr>
<td>$(4, 0)$</td>
<td>$(11, 3)$</td>
<td>$(18, 20)$</td>
</tr>
<tr>
<td>$(5, 4)$</td>
<td>$(11, 20)$</td>
<td>$(19, 5)$</td>
</tr>
<tr>
<td>$(5, 19)$</td>
<td>$(12, 4)$</td>
<td>$(19, 18)$</td>
</tr>
</tbody>
</table>
ECC Example

- $E_{211}(0, -4)$, $y^2 = x^3 + ax + b = x^3 - 4$
- $G = (2, 2)$, Calculate $121G$
- $121 = 1111001 \Rightarrow 64G + 32G + 16G + 8G + G$

<table>
<thead>
<tr>
<th>Point</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>$\lambda$</th>
<th>$x_R$</th>
<th>$y_R$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$G$</td>
<td></td>
<td></td>
<td></td>
<td>2</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>$2G = G + G$</td>
<td>12</td>
<td>4</td>
<td>53</td>
<td>3</td>
<td>5200</td>
<td></td>
</tr>
<tr>
<td>$4G = 2G + 2G$</td>
<td>75</td>
<td>189</td>
<td>163</td>
<td>198</td>
<td>159</td>
<td>114</td>
</tr>
<tr>
<td>$8G = 4G + 4G$</td>
<td>94</td>
<td>17</td>
<td>149</td>
<td>80</td>
<td>174</td>
<td>163</td>
</tr>
<tr>
<td>$16G = 8G + 8G$</td>
<td>98</td>
<td>115</td>
<td>200</td>
<td>188</td>
<td>181</td>
<td>209</td>
</tr>
<tr>
<td>$32G = 16G + 16G$</td>
<td>168</td>
<td>207</td>
<td>158</td>
<td>169</td>
<td>136</td>
<td>11</td>
</tr>
<tr>
<td>$64G = 32G + 32G$</td>
<td>206</td>
<td>22</td>
<td>48</td>
<td>182</td>
<td>147</td>
<td>97</td>
</tr>
<tr>
<td>$9G = G + 8G$</td>
<td>161</td>
<td>172</td>
<td>119</td>
<td>169</td>
<td>111</td>
<td>145</td>
</tr>
<tr>
<td>$25G = 9G + 16G$</td>
<td>64</td>
<td>70</td>
<td>208</td>
<td>19</td>
<td>69</td>
<td>20</td>
</tr>
<tr>
<td>$57G = 25G + 32G$</td>
<td>202</td>
<td>67</td>
<td>63</td>
<td>66</td>
<td>142</td>
<td>15</td>
</tr>
<tr>
<td>$121G = 57G + 64G$</td>
<td>82</td>
<td>5169</td>
<td>143</td>
<td>115</td>
<td>48</td>
<td></td>
</tr>
</tbody>
</table>

$\lambda = \frac{y_Q - y_P}{x_Q - x_P}$ or $\frac{3x_P^2 + a}{2y_P}$

$R = P + Q = [\lambda^2 - x_P - x_Q, \lambda(x_P - x_R) - y_P]$
ECC Diffie-Hellman

- Select a suitable curve $E_q(a, b)$
- Select base point $G = (x_1, y_1)$ with large order $n$ s.t. $nG = O$
- A & B select private keys $n_A < n$, $n_B < n$
- Compute public keys: $Y_A = n_A G$, $Y_B = n_B G$
- Compute shared key: $K = n_A Y_B$, $K = n_B Y_A$
  - Same since $K = n_A n_B G$
- Attacker would need to find $K$, hard
ECC Encryption/Decryption

- Several alternatives, will consider simplest
- Select suitable curve & point G
- Encode any message M as a point on the elliptic curve $P_m$
- Each user chooses private key $n_A < n$
- Computes public key $P_A = n_A G$, $P_B = n_B G$
- Encrypt $P_m$: $C_m = \{ kG, P_m + kP_B \}$, $k$ random
- Decrypt $C_m$ compute:
  $$P_m + kP_B - n_B (kG) = P_m + k (n_B G) - n_B (kG) = P_m$$
ECC Encryption/Decryption Example

- $E_{257}(0, -4)$, $P_m=(112,26)$, $n_B=101$ $G=(2, 2)$
- $P_B=n_BG = 101(2, 2) =(197, 167)$
- $k=41$, $C_1=kG=41(2,2)=(136, 128)$
- $C_2=P_m+kP_B = (112, 26) + 41(197, 167)$
  $=(112, 26)+(68, 84) = (246, 174)$
- $C_m={C_1, C_2} = \{(136,128),(246, 174)\}$
- $P_m=C_2-n_BC_1 = (246, 174)-101(136, 128)$
  $=(246, 174)-(68, 84) = (112, 26)$
ECC Security

- Relies on elliptic curve logarithm problem
- Can use much smaller key sizes than with RSA etc
- For equivalent key lengths computations are roughly equivalent
- Hence for similar security ECC offers significant computational advantages

<table>
<thead>
<tr>
<th>Symmetric scheme (key size in bits)</th>
<th>ECC-based scheme (size of $n$ in bits)</th>
<th>RSA/DSA (modulus size in bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>56</td>
<td>112</td>
<td>512</td>
</tr>
<tr>
<td>80</td>
<td>160</td>
<td>1024</td>
</tr>
<tr>
<td>112</td>
<td>224</td>
<td>2048</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>3072</td>
</tr>
<tr>
<td>192</td>
<td>384</td>
<td>7680</td>
</tr>
<tr>
<td>256</td>
<td>512</td>
<td>15360</td>
</tr>
</tbody>
</table>

Washington University in St. Louis  http://www.cse.wustl.edu/~jain/cse571-17/  ©2017 Raj Jain
PRNG based on Asymmetric Ciphers

- Asymmetric encryption algorithms produce apparently random output
- Hence can be used to build a pseudorandom number generator (PRNG)
- Much slower than symmetric algorithms
- Hence only use to generate a short pseudorandom bit sequence (e.g., key)
PRNG based on RSA

- Micali-Schnorr PRNG using RSA
  - in ANSI X9.82 and ISO 18031
PRNG based on ECC

- Dual elliptic curve PRNG
  - NIST SP 800-9, ANSI X9.82 and ISO 18031
- Some controversy on security /inefficiency
- Notation: $x(P) = x$ coordinate of $P$. $\text{lsb}_i(x) = i$ least sig bits of $x$
- Algorithm
  
  ```
  for i = 1 to k do
    set $s_i = x(s_{i-1}P)$
    set $r_i = \text{lsb}_{240}(x(s_iQ))$
  end for
  return $r_1, \ldots, r_k$
  ```
- Only use if just have ECC
Summary

1. Diffie-Hellman key exchange allows creating a secret in public based on exponentiation
2. ElGamal cryptography uses D-H
3. Elliptic Curve cryptography is based on defining addition of points on an elliptic curve in GF(p) or GF(2^n)
4. Public key cryptography (both RSA and ECC) can also be used to generate cryptographically secure pseudorandom numbers.
Homework 10

1. Consider an Elgamal scheme with a common prime \( q=71 \) and a primitive root \( \alpha=7 \).
   - A. If B has public key \( Y_B=3 \) and A choose the random integer \( k=2 \), what is the ciphertext of \( M=30 \)?
   - B. If A now chooses a different value of \( k \) so that the encoding of \( M=30 \) is \( C=(59,C_2) \). What is the integer \( C_2 \)?

2. For an elliptic curve cryptography using \( E_{11}(1,6) \) and \( G=(2,7) \). B’s private key \( n_B=7 \).
   - A. Find B’s Public key \( P_B \)
   - B. A wishes to encrypt the message \( P_m=(10, 9) \) and chooses the random value \( k=3 \). Determine the ciphertext \( C_m \)
   - C. Show the calculation by which B recovers \( P_m \) from \( C_m \).
Lab 10: Kali Linux

- Prepare a bootable USB drive with Kali Linux
- See instructions at: http://docs.kali.org/downloading/kali-linux-live-usb-install
- You will need a 4GB or larger USB 3 flash drive
- Also, you will need to change the boot sequence in your computer to allow booting from the USB drive
- No other changes are required to your disk or computer.
- Explore Kali and submit the list of penetration tools available in Kali
- Note: Kali is a goddess that destroys evil

Ref: https://en.wikipedia.org/wiki/Kali_Linux
Washington University in St. Louis http://www.cse.wustl.edu/~jain/cse571-17/ ©2017 Raj Jain
Acronyms

- ANSI American National Standards Institute
- DEC Dual Elliptic Curve
- DSS Digital Signature Standard
- ECC Elliptic curve cryptography
- GF Galvois Field
- IEEE Institute of Electrical and Electronic Engineers
- ISO International Standards Organization
- MIME Multipurpose Internet Multimedia Email
- NIST National Institute of Science and Technology
- OFB Output feedback mode
- PRF Pseudo-random function
- PRNG Pseudo-Random Number Generator
- RSA Rivest, Shamir, and Adleman
- SP Standard Practice
- VPN Virtual Private Network
Related Modules

CSE571S: Network Security (Spring 2017),
http://www.cse.wustl.edu/~jain/cse571-17/index.html

CSE473S: Introduction to Computer Networks (Fall 2016),
http://www.cse.wustl.edu/~jain/cse473-16/index.html

Wireless and Mobile Networking (Spring 2016),
http://www.cse.wustl.edu/~jain/cse574-16/index.html

CSE571S: Network Security (Fall 2014),
http://www.cse.wustl.edu/~jain/cse571-14/index.html

Audio/Video Recordings and Podcasts of
Professor Raj Jain's Lectures,
https://www.youtube.com/channel/UCN4-5wzNP9-ruOzQMs-8NUw