

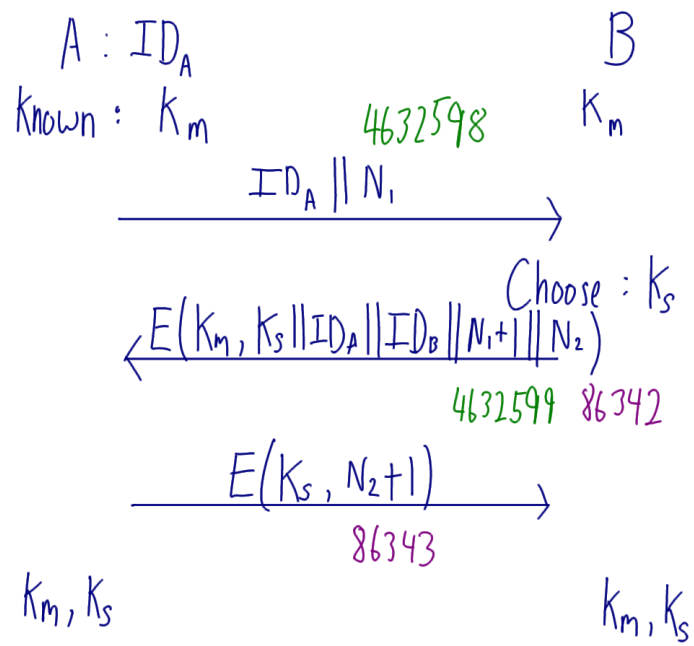
CSS322 – Key Management Notes

$$\frac{n \times (n-1)}{2}$$

40: 780 pairs
780 keys

Link only: 20 keys
End points (computers): 45 keys
End points (5 apps): 1225 keys

Figure 1: Number of Keys with Link and End-to-End Encryption; Lecture 20



Master keys

A-B : K_{ab}	AB	$\frac{n \times (n-1)}{2}$
A-C : K_{ac}	AC	
B-C : K_{bc}	AD	
	BC	
	BC BD	
	CD	

Figure 2: Decentralised Key Distribution; Lecture 20

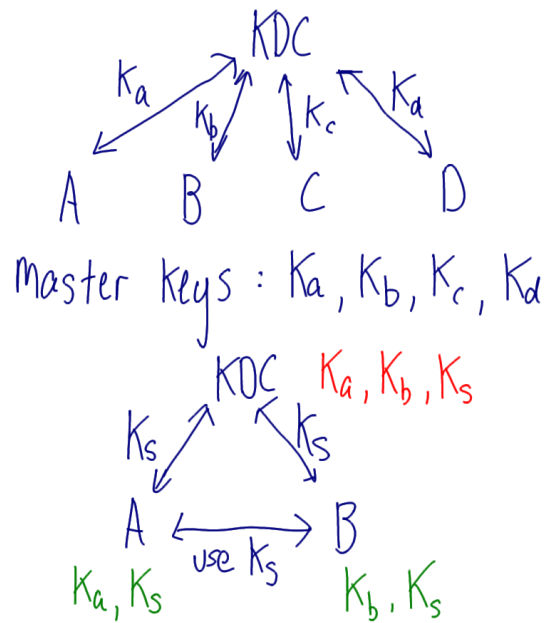


Figure 3: Centralised Key Distribution with KDC; Lecture 20

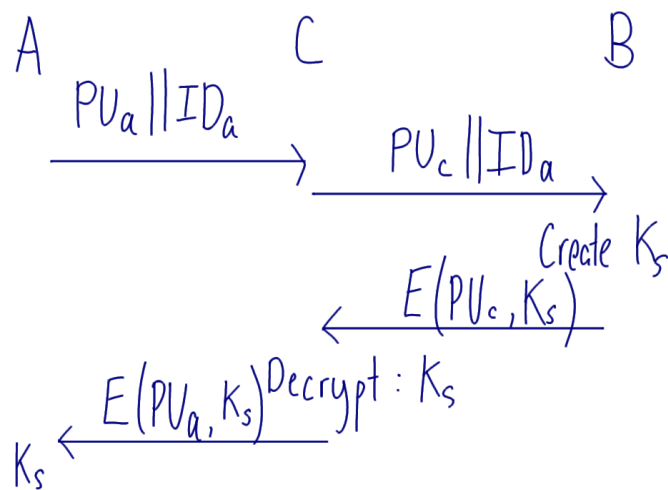


Figure 4: Man-in-the-Middle Attack on Public Key Exchange; Lecture 21

<p>A</p> $q = 353$ $\alpha = 3$ <p>Select $X_a = 97$</p> $Y_a = \alpha^{X_a} \bmod q$ $= 3^{97} \bmod 353$ $= 40$	<p>B</p> $q = 353$ $\alpha = 3$
$\xrightarrow{Y_a = 40, \alpha = 3, q = 353}$	
$X_b = 233$	
$Y_b = \alpha^{X_b} \bmod q$ $= 3^{233} \bmod 353$ $= 248$	
$K_b = Y_a^{X_b} \bmod q$ $= 40^{233} \bmod 353$ $= 160$	
$\xleftarrow{Y_b = 248}$	
$K_a = Y_b^{X_a} \bmod q$ $= 248^{97} \bmod 353$ $= 160$	

Figure 5: Diffie-Hellman Key Exchange Example; Lecture 21

$$\begin{array}{l}
 \text{A} \\
 Y_a = \alpha^{X_a} \bmod q \\
 K_a = Y_b^{X_a} \bmod q \\
 = (\alpha^{X_b} \bmod q)^{X_a} \bmod q \\
 = (\alpha^{X_b})^{X_a} \bmod q \\
 = \alpha^{X_b X_a} \bmod q
 \end{array}
 \qquad
 \begin{array}{l}
 \text{B} \\
 Y_b = \alpha^{X_b} \bmod q \\
 K_b = Y_a^{X_b} \bmod q \\
 = (\alpha^{X_a} \bmod q)^{X_b} \bmod q \\
 = (\alpha^{X_a})^{X_b} \bmod q \\
 = \alpha^{X_a X_b} \bmod q
 \end{array}$$

Figure 6: Diffie-Hellman Key Exchange Proof of Same Key; Lecture 21

$$\begin{array}{l}
 \text{Known: } q = 353 \\
 \alpha = 3 \\
 Y_a = 40 \\
 Y_b = 248 \\
 K_a = Y_b^{X_a} \bmod q \\
 = 248^{X_a} \bmod 353 \\
 Y_a = \alpha^{X_a} \bmod q \\
 40 = 3^{X_a} \bmod 353 \\
 X_a = \text{dlog}_{3,353}(40)
 \end{array}$$

Figure 7: Diffie-Hellman Key Exchange Attack; Lecture 21

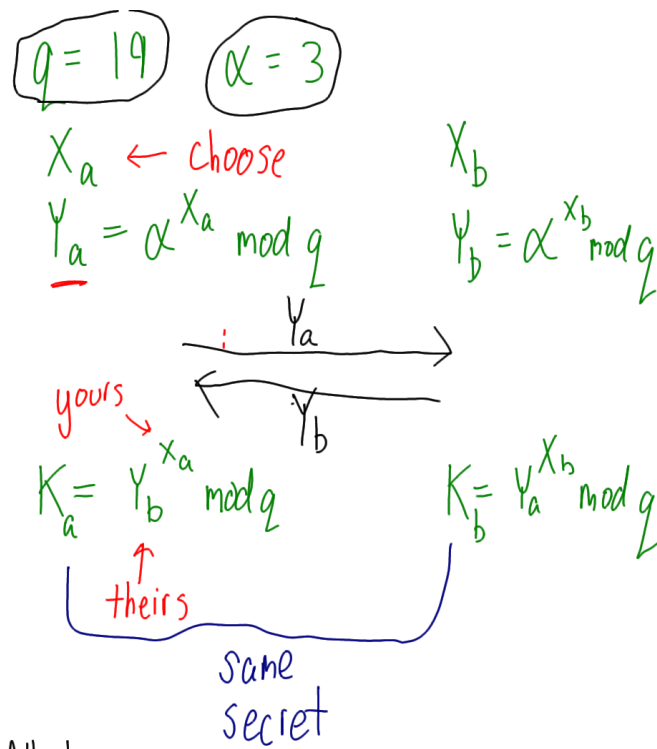


Figure 8: Diffie-Hellman Key Exchange Example 2; Lecture 22

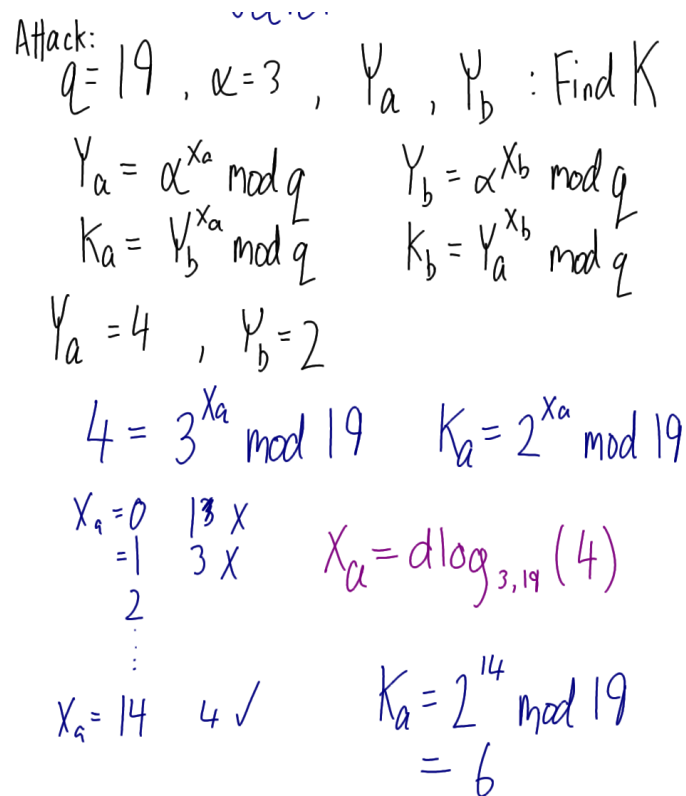


Figure 9: Diffie-Hellman Key Exchange Attack 2; Lecture 22

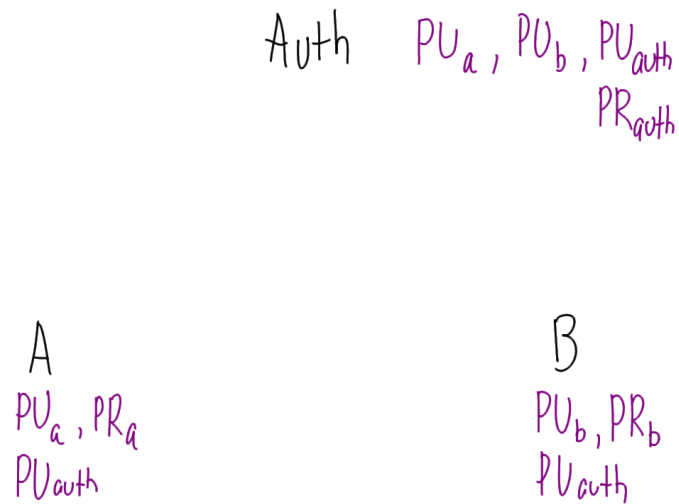


Figure 10: Public Key Authority: Known Keys; Lecture 22

1. Generate Key pair : (PU_a, PR_a)
2. Certificate Signing Request (CSR)
(send to CA)
3. CA issues a certificate, C_a
(send to A)
4. Verify C_a (need PU_{CA})
5. Send C_a to B
B sends C_b to A
6. Verify C_b (need PU_{CA})

Figure 11: Certificate generation and signing steps; Lecture 24