Unit Template

Unit Name: Public Key Encryption	Content Area: CS		Duration: about 10 days

Essential Question(s):

- How can we communicate privately across a public channel?
- What is the advantage of public key encryption over earlier encryption methods?

Assessments

Pre-Assessment:
*Check for student understanding in modular arithmetic
*What is encryption? How do we protect data?
*Assess understanding of Big-O algorithms runtime

Summative Assessment:
*CFU questions in reference documents
*Create a working code for RSA encryption

Standards

CSTA Standards
(https://www.csteachers.org/page/standards)

2-NI-05 (Grades 6-8)
Explain how physical and digital security measures protect electronic information.
[Networks & the Internet || Cybersecurity || Communicating]

2-NI-06 (Grades 6-8)
Apply multiple methods of encryption to model the secure transmission of information.
[Networks & the Internet || Cybersecurity || Communicating]

2-IC-23 (Grades 6-8)
Describe tradeoffs between allowing information to be public and keeping information private and secure.
[Impacts of Computing || Safety Law & Ethics || Communicating]

3A-NI-05 (Grades 9-10)
Give examples to illustrate how sensitive data can be affected by malware and other attacks.
[Networks & the Internet || Network Communication & Organization || Communicating]

3A-AP-13 (Grades 9-10)
Create prototypes that use algorithms to solve computational problems by leveraging prior student knowledge and personal interests.
[Algorithms & Programming || Algorithms || Creating]

3A-AP-14 (Grades 9-10)
Use lists to simplify solutions, generalizing computational problems instead of repeatedly using simple variables.
[Algorithms & Programming || Variables || Abstraction]

3B-NI-04 (Grades 11-12)
Compare ways software developers protect devices and information from unauthorized access.
[Networks & the Internet || Cybersecurity || Communicating]

CS4ALL Blueprint

*Perspectives
-Citizen: "I can question how computing practices and concepts affect my community."

*Concepts
-Networks|| Trust: The common thread behind issues of security, privacy and consent is trust. Whenever we connect to a network, we decide our level of trust, based on our security and privacy needs. We implement and monitor protocols to protect those needs.

*Practices
-Analyzing: Describe an application of computing by detailing who, what, where, when. In this first step, focus on things that can be observed.
Describing: Describe an application of computing by detailing who, what, where, when. In this first step, focus on things that can be observed.
Examining: Examine the description for patterns, general characteristics, or anomalies. How do the parts of the whole relate to each other and the user?

3B-AP-10 (Grades 11-12) Use and adapt classic algorithms to solve computational problems. [Algorithms & Programming || Algorithms || Abstraction] **3B-AP-11 (Grades 11-12)** Evaluate algorithms in terms of their efficiency, correctness, and clarity. [Algorithms & Programming || Algorithms || Abstraction]		

Learning Plan

Focus Questions	Academic Tasks	DOK	Notes	Resources	Academic and Discipline-Specific Vocabulary	Pedagogical techniques/ Differentiation ideas
1. What is encryption?	Understand why it's important to protect data	1	• Encryption is important for both data security as well as authentication of the message. (i.e. How do you know if the message is from that person?)	https://www.tutorialspoint.com/cryptography/public_key_encryption.htm	• encryption	- **Multiple entry points**: e.g. exchange a secret note with a couple of students, have students go to websites with h, explain using how mails work (the act of sending mails is public, but the content of the mail is usually secret/hidden) -**CFU:** thumbs up/down, fist to 5, stoplight, exit tickets
	Define what encryption is	1				
	Identify a real-life situation when encryption is needed	2				
2. How does encryption work?	Review what encryption is and why it's necessary	1	• If two people used the same key to encrypt and decrypt a message, it may not be safe because a person(e.g. hacker) can steal the key and decrypt the message. • If a person distributes the same lock to many people, but only that person holds the key to unlock, the message is kept safe. • Public key = lock • Private key = actual key	https://www.youtube.com/watch?v=mthPiiCS24A	• Public key • Private key • Symmetric vs asymmetric	-**CFU:** thumbs up/down, fist to 5, stoplight, exit tickets **-Kinesthetic learning**: design an interactive game that involves the concept of encryption. Try CS unplugged activities such as https://classic.csunplugged.org/public-key-encryption/
	Understand what public and private keys are in encryption	1				
	*Participate in interactive games such as telephone game to deepen the understanding of encryption	2				
3. What is RSA Encryption?	Understand RSA Algorithm	1	• "Trap-door function" • Explain phi function; Euler totient function → any x raised to phi mod n = 1 • Phi must not share a factor with e • d = inverse of e mod phi → use extended Euclidean algorithm to find d • Public key = N and e • Private key is d • Go over modular arithmetic!	https://www.youtube.com/watch?v=Z8M2BTscoD4 **(from Z)** **Khan Academy video** https://www.youtube.com/watch?v=EPXilYOa71c **Wolfram Alpha alculator for modular arithmetic** https://www.wolframalpha.com/	• Modulus, modular • Fundamental theorem of arithmetic • Extended Euclidean Algorithm • Phi function	- **Teacher modeling** **-small group instruction** **-mindful grouping**: heterogeneous grouping or homogenous grouping based on students; ex. Have one high-level student in a mid- or lower-level group and have them be a teacher **-checklist/handout/visual aid**: shows how to complete each step of RSA Algorithm
	Practice encryption and decryption by hand	2				

				***see Learning Guide KtS**		
4. How can we design a program for encryption?	Design a flowchart for RSA encryption program	**3**	• Choose p,q (both are prime numbers) • Calculate N = p*q • Calculate phi (N); phi(N) = (p-1)(q-1) • Choose e; 1 < e < phi (N), coprime with N and phi(N) • Choose d to satisfy the following condition: d*e mod phi (N) = 1 • Write encrypt(); c = m^e mod N where c = ciphertext and m = original message • Write decrypt() using: m = c^d mod N • What other helper methods may be beneficial?	**KtS** **https://www.youtube.com/watch?v=Z8M2BTscoD4** **Wolfram Alpha alculator for modular arithmetic** **https://www.wolframalpha.com/**		• This lesson can be 2-3 days depending on the students' levels (e.g. day 1 can be writing code for helper methods, and day 2 for encrypt/decrypt) • For students with disability: it might be easier to walk through the RSA algorithm in a smaller group and have them explain in their own words for CFU • Suggestions:pair programming or heterogenous groupings; utilize KtS to draw flowchart for the necessary algorithm for getN, getE, getPhi, getD; subgoal label as a whole class • Certain methods/class files may be encapsulated (ex Euclidean algorithm) to simplify the process • Group flowchart may be beneficial before actual coding
	Identify helper methods necessary in creating RSA Encryption program	**3**				
	Create a working RSA encryption program	**3 - 4**				
5. Why is RSA so hard to break?	Understand prime factorization	**2**	• How does prime factorization work? • Try prime factorization with 10, 50, 100, 1000…. (time the students while they're doing this and make it fun!) • What would be the runtime for an actual computer to run prime factorization for 100, 10^5, 10^10…? • Food for thought: prime factorizing 2000 vs 2059…?	**Khan Academy video** **https://www.youtube.com/watch?v=ZKKDTfHcsG0&t=6s** **Visualization of prime factorization** **http://www.datapointed.net/visualizations/math/factorization/animated-diagrams/** ***see Learning Guide(Runtime Analysis.pdf)**	• Prime factorization • Runtime	• **Teacher modeling** • **hands-on activity**: ex. Have groups prime factorize 2 digit numbers, 3 digit numbers, 4 digit numbers… (You can time the groups to gamify/ increase engagement) • **Visual aids**: ex. Graph the time it took to prime factorize 2 digit #, 3 digit #, etc. • **CFU**
	Analyze patterns of runtime as #s get larger	**3**				
6. Could quantum computing break RSA?	Understand what quantum computer is	**1**	• **"But…quantum computers!"** • Focus on going over the application and the effect of quantum computers, rather than how quantum computers work. • Why would the researchers be interested in developing quantum computers? • What will happen if RSA algorithm can be broken?	• https://www.technologyreview.com/2019/05/30/65724/how-a-quantum-computer-could-break-2048-bit-rsa-encryption-in-8-hours/ • https://www.youtube.com/watch?v=YOw124Cx00 **(originally thought it would take 40 quadrillion years to solve, but only took 17!)** • https://www.youtube.com/watch?v=tX7e7CaWrvM • https://www.cnet.com/news/the-us-wants-to-develop-a-quantum-internet/?utm_source=join1440&utm_medium=email	• Quantum, quantum computing	• **Discussion methods** such as Socratic method, or fishbowl • If using articles, differentiate based on reading levels using NewsELA, or use reading strategies such as GIST strategy
	Evaluate the potential impact of quantum computing in data security	**3 - 4**				
Instructional Supports						
Lowest 1/3 and SWDs	**Highest 1/3**		**ELLs**			
	-Provide extension activities and questions:		Word wall/encourage them to create their own glossary with definitions written in their own language using index cards			

-Reference sheet with mathematical background/reminders -small group instruction -tiered/differentiated tasks (ex. Focus on creating helper methods) -provide flowchart and code with a lot of subgoal labels -pair programming with higher-level student	*Compare difficulties of different "levels" of RSA. What if n doubles in size? *Have them check certificate of popular websites. Identify what each field and value means under Details tab. *Why is RSA used in mostly hybrid cryptography? -Have them design all the parts of RSA encryption program. -Create a prime factorization program and use it for runtime analysis -Make them student leaders of each group	Homogeneous grouping (students who speak the same language)		
Post Unit Reflection:				

*Based on the requirements of the Tri-State Quality Review Rubric for Lessons and Units