Team Members: Heather DeVal, Jordan Fok, Hannah Russell

Topic: Simulation and Comparison of Various Attacks on the RSA Cryptosystem

Attacks (From Research*):

- **Message space search**
    - The encryption algorithm must be communicated to each party, hidden within the message space.
    - Encrypt the message blocks with each other until plaintext is the result.
    - Takes a long time, very inefficient
- **Guessing d (the private exponent)**
    - Requires the attacker knows both a corresponding plaintext and ciphertext pair.
        - Attackers can "guess" d over and over until encrypting the plaintext results in the ciphertext.
    - Once d is found, the attacker has full access to all future communications.
- **Cycle attack, aka iteration attack**
    - If the attacker repeats the encryption process enough times on the ciphertext, the result will eventually be the decrypted text (plaintext message)
    - Not an ideal attack due to variations in the amount of encryption iterations needed to decrypt the text. If the message space is large, this can take a long time.
    - Number of iterations should be saved so the attacker knows how many cycles to encrypt for future messages.
- **Common modulus**
    - A common modulus is sometimes used for employees/people within a business/organization. This makes things more convenient for administrators, however, it has security implementations. Attackers are able to mock employees' private keys from their public keys by factoring the modulus N.
- **Low private exponent**
    - **Private:** a small private exponent can cut back on performance time tremendously, but the system can be broken using Weiner's law.
        - Weiners law decreases the amount of exponents possible for the cryptosystem. Therefore, the attacker can search through the exponents faster, and break the system easier.
- **Factoring the public key**
    - ****Very effective attack on RSA**
    - RSA relies on integer factorization for generation of public keys
    - If the attacker can find the prime factors used in the RSA process, they can compute the exponent d from any party's public key
    - With the public key and exponent, they can decrypt the ciphertext.

- **Blinding**
    - Uses a party's digital signature from one message to sign another message that the party did not consent to.
- **Partial Key Exposure attack**
    - The attacker has part of the private key available to them
    - Since the attacker has part of d, they are able to determine the full value of d faster using bit operations
        - Better to have the part of d containing the least significant bits.
- **Timing Attacks**
    - Ex: RSA smartcard that stores private key
    - By measuring the time the card takes to perform the decryption/encryption, an attacker is able to determine the exponent d.
- **Random Faults**
    - Some RSA Algorithms use the Chinese Remainder theorem
    - If an error occurs during the encryption/decryption process, the chinese remainder theorem can result in false signatures.

Goals:

- Narrow down to two or three of the above attacks and implement them.
    - **Currently focusing on:** Factoring the public key, blinding, partial key exposure attack
- Compare the difficulty of the implementation, reality of execution, etc
- Discuss the ethical implementations of using RSA and attacking RSA.

Research:

- Possible attacks on RSA: http://www.members.tripod.com/irish_ronan/rsa/attacks.html
- https://math.boisestate.edu/~liljanab/ISAS/course_materials/AttacksRSA.pdf
- 20 Years of Attacks on RSA: https://crypto.stanford.edu/~dabo/papers/RSA-survey.pdf
- Active vs Passive Attacks: https://www.geeksforgeeks.org/active-and-passive-attacks-in-information-security/
- Cryptographic attacks: https://en.wikipedia.org/wiki/Category:Cryptographic_attacks
- https://en.wikipedia.org/wiki/RSA_(cryptosystem)#Integer_factorization_and_RSA_problem

Back up plan:

- OTP: Attacks on Stream Ciphers and the One time pad: https://www.coursera.org/lecture/crypto/attacks-on-stream-ciphers-and-the-one-time-pad-euFJx