UConn, School of Computing
Fall 2024
CSE 3400/CSE 5850: Introduction to Computer and Network Security / Introduction to Cybersecurity

Assignment 5

Instructor: Prof. Ghada Almashaqbeh
Posted: 11/8/2024
Submission deadline: 11/17/2024, 11:59 pm

Jonathan Degrange

Notes:

- Solutions **must be typed** (using latex or any other text editor) and must be submitted as a pdf (not word or source latex files).

Problem 1 [30 points]
This problem is about shared key protocols:

1. In this modified 2PP protocol, Bob sends a nonce $N_B = 3 \cdot N_A$ based on Alice's nonce N_A. Since Eve is intercepting messages between Alice and Bob, she can see N_A, compute N_B, and respond to Bob with the expected nonce value. This allows Eve to impersonate Alice because she can generate the correct response to Bob's challenge without knowing Alice's secret, exploiting the deterministic relationship between N_B and N_A. The lack of a secure, unpredictable nonce makes this protocol modification vulnerable to MitM attacks.

2. Eve can modify Alice's nonce N_A because the 2PP protocol lacks integrity checks on the nonce. By changing N_A to a new value N'_A, Eve deceives Bob into using an altered nonce, leading to potential authentication failure and allowing Eve to control the session. This compromises the security of the handshake, enabling Eve to impersonate Alice or interfere with the communication.

3. Charlie's claim is incorrect. While a CRHF (collision-resistant hash function) provides collision resistance, it does not ensure integrity in the same way as a MAC. A MAC not only provides collision resistance but also relies on a secret key, binding the output to both the message and the key, which ensures authenticity. In contrast, a keyed hash function does not necessarily prevent an attacker from forging a valid hash by manipulating inputs. Without the specific security guarantees of a MAC, the protocol could be vulnerable to forgery attacks, where an attacker might replace or modify a message and still produce a valid hash. Therefore, simply using a CRHF with a keyed hash function does not guarantee the same level of security, as it lacks the authenticity and integrity that a MAC provides.

4. If Alice resets her counter to 0 after each successful protocol run, it undermines the protocol's ability to protect against replay attacks. Counters are typically used to

ensure that each protocol session has a unique identifier, preventing an attacker from replaying old messages to trick one party into accepting them as new. By resetting the counter, Alice inadvertently allows repeated values, which an attacker could exploit by replaying messages from previous sessions. This weakens the protocol's security, as Bob would be unable to distinguish between fresh and replayed sessions, making the protocol vulnerable to replay attacks.

5. Yes, Roger's modification would impact the security of the protocol. In the original protocol, both the 'req' and 'resp' are encrypted and then MACed, ensuring that their contents remain confidential and integrity-protected. By removing encryption from the 'req' and 'resp' fields while still including them in the MAC, Roger's version exposes these values to potential attackers. The primary impact is on **confidentiality**: without encryption, any eavesdropper can view the 'req' and 'resp' contents, which might contain sensitive data. While the MAC still protects integrity (detecting tampering), it does not prevent third parties from learning the data's contents. Thus, the change weakens confidentiality, as sensitive information in 'req' and 'resp' would no longer be protected from observation, making this protocol modification less secure than the original.

Problem 2 [40 points]
Compute the following expressions (please read the note below carefully):

1. $192 \cdot 17^{1700} + 986 \cdot 2024^{56} \mod 37$

   To simplify $192 \cdot 17^{1700} + 986 \cdot 2024^{56} \mod 37$:

   1. We simplify $17^{1700} \mod 37$ using Fermat's Little Theorem : $17^{1700} \equiv 17^8 \equiv 26 \pmod{37}$.

   2. We simplify $192 \cdot 17^{1700} \mod 37$**: $192 \cdot 26 = 4992 \equiv 8 \pmod{37}$.

   3. We simplify $2024^{56} \mod 37$**: $2024 \equiv 25 \pmod{37}$, and $25^{20} \equiv 34 \pmod{37}$.

   4. We simplify $986 \cdot 2024^{56} \mod 37$**: $986 \cdot 34 = 33524 \equiv 9 \pmod{37}$.

   5. Then we combine the results: $8 + 9 = 17 \pmod{37}$.

   So: $192 \cdot 17^{1700} + 986 \cdot 2024^{56} \equiv 17 \pmod{37}$.

2. To simplify $576(23023^{1024000} + 365 \cdot 856985^{7202}) \mod 73$

   1. We simplify $23023^{1024000} \mod 73$:

   - $23023 \equiv 28 \pmod{73}$ - Using Fermat's Little Theorem: $28^{1024000} \equiv 28^{56} \equiv 7 \pmod{73}$

   2. We simplify $856985^{7202} \mod 73$:

   - $856985 \equiv 72 \equiv -1 \pmod{73}$

   - $(-1)^{7202} \equiv 1 \pmod{73}$

   3. We simplify $365 \cdot 856985^{7202} \mod 73$:

   - $365 \equiv 0 \pmod{73}$

   4. Then we combine the results:

   - $576 \cdot (7 + 0) = 576 \cdot 7 = 4032$

- 4032 $\mod 73 = 17$

Final answer: $576(23023^{1024000} + 365 \cdot 856985^{7202}) \equiv 17 \pmod{73}$.

3. To simplify $\frac{1436}{8} + \frac{568}{17} + 10785 \mod 101$:

1. $\frac{1436}{8} \mod 101$: $1436 \equiv 4 \pmod 8$, so $\frac{4}{8} \equiv 0 \pmod{101}$.

2. $\frac{568}{17} \mod 101$: $\frac{568}{17} = 33 \mod 101$.

3. $10785 \mod 101$: $10785 \equiv 78 \pmod{101}$.

We combine: $0 + 33 + 78 = 111 \mod 101 = 10$.

So : $\frac{1436}{8} + \frac{568}{17} + 10785 \equiv 10 \pmod{101}$.

4. To simplify $2048 - \frac{8}{1436} + \frac{10785}{20 \cdot 17} \mod 99$:

1. $\frac{8}{1436} \mod 99$: Since $1436 \equiv 53 \pmod{99}$, find the modular inverse of 53 modulo 99, which is 53 (since $53 \cdot 53 \equiv 1 \pmod{99}$). Thus, $\frac{8}{1436} \equiv 8 \cdot 53 = 424 \equiv 29 \pmod{99}$.

2. $\frac{10785}{20 \cdot 17} \mod 99$: $20 \cdot 17 = 340$, and $10785 \div 340 = 31.7$, so we compute $\frac{10785}{340} \equiv 10785 \cdot 340^{-1} \mod 99$ yielding 90 modulo 99.

3. We combine the results: $2048 - 29 + 90 \mod 99 = 2109 \mod 99 = 30$.

So : $2048 - \frac{8}{1436} + \frac{10785}{20 \cdot 17} \equiv 30 \pmod{99}$.

5. To simplify $10245 \cdot (24 + (345 \cdot 10^{77} \cdot (34^{381} + 18976))) \mod 67$:

1. Simplify $10245 \mod 67$: $10245 \equiv 61 \pmod{67}$.

2. Simplify $345 \mod 67$: $345 \equiv 10 \pmod{67}$.

3. Simplify $10^{77} \mod 67$: Using Fermat's Little Theorem, $10^{77} \equiv 10^{11} \mod 67$. Calculate $10^{11} \mod 67$ using successive squaring: - $10^2 \equiv 33 \pmod{67}$ - $10^4 \equiv 16 \pmod{67}$ - $10^8 \equiv 48 \pmod{67}$ - $10^{11} \equiv 49 \pmod{67}$

4. Simplify $34^{381} \mod 67$: By Fermat's Little Theorem, $34^{66} \equiv 1 \pmod{67}$, so reduce $34^{381} \mod 67$, and find $34^{381} \equiv 34^3 \equiv 59 \pmod{67}$.

5. We combine : - $345 \cdot 10^{77} \cdot (34^{381} + 18976) \equiv 10 \cdot 49 \cdot (59 + 18976) \equiv 10 \cdot 49 \cdot 59 \equiv 10 \cdot 2881 \equiv 13 \pmod{67}$

6. Final expression: $10245 \cdot (24 + 13) = 10245 \cdot 37 \equiv 61 \cdot 37 = 2257 \equiv 26 \pmod{67}$.

So : $10245 \cdot (24 + (345 \cdot 10^{77} \cdot (34^{381} + 18976))) \equiv 26 \pmod{67}$.

6. To compute $\phi(80) + \phi(2728)$:

1. $\phi(80)$: $80 = 2^4 \cdot 5$, so :

$$\phi(80) = 80\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{5}\right) = 80 \times \frac{1}{2} \times \frac{4}{5} = 32$$

2. $\phi(2728)$: $2728 = 2^3 \cdot 17 \cdot 19$, so :

$$\phi(2728) = 2728\left(1 - \frac{1}{2}\right)\left(1 - \frac{1}{17}\right)\left(1 - \frac{1}{19}\right) = 1364 \times \frac{16}{17} \times \frac{18}{19} = 1212$$

3. Sum :

$$\phi(80) + \phi(2728) = 32 + 1212 = 1244$$

Final answer : $\phi(80) + \phi(2728) = 1244$.

Problem 3 [30 points]
This problem is about public key cryptographic primitives.

1. In the Diffie-Hellman key exchange with $p = 47$, $g = 5$, Alice's secret $a = 9$, and Bob's secret $b = 7$:

   1. Alice's computation : - Alice computes $A = 5^9 \mod 47 = 40$. - Alice sends $A = 40$ to Bob.

   2. Bob's computation : - Bob computes $B = 5^7 \mod 47 = 36$. - Bob sends $B = 36$ to Alice.

   3. Shared secret computation : - Alice computes $K_A = 36^9 \mod 47 = 31$. - Bob computes $K_B = 40^7 \mod 47 = 31$.

   Both Alice and Bob share the secret key $K = 31$.

2. In the extended Diffie-Hellman key exchange for four parties (Alice, Bob, Cameron, and Doug):

   1. Round 1 : Each participant computes $g^a, g^b, g^c, g^d \mod p$ and sends their result to all other participants.

   2. Round 2 : Each participant computes the product of the received values raised to their secret key. For example, Alice computes $(B \cdot C \cdot D)^a \mod p$, and sends it to others.

   3. Round 3 : Participants combine the results from Round 2 and raise them to their own secret key, arriving at the shared key g^{abcd}.

   Total messages : 36 (12 per round, 3 rounds).

   The protocol ensures all participants compute the same shared key and is secure against eavesdropping since an attacker can't derive the shared key without knowing the secret keys.

3. To generate a new ciphertext (x', y') encrypting the message km given the public key $e = g^a$ and the ciphertext (x, y) that encrypts m, we proceeds like that :

   1. Given : $x = g^b$ and $y = g^{ab} \cdot m$.

   2. Desired ciphertext : We want to encrypt km. We can achieve this by setting: - $x' = x^k = g^{bk}$ (so $b' = bk$), - $y' = y^k = (g^{ab} \cdot m)^k = g^{abk} \cdot m^k$.

   3. Ensure : $x' \neq x$ and $y' \neq y$ since $k \neq 1$ and $b' \neq b$.

   Thus, x' and y' encrypt km, and this operation is computationally efficient (polynomial time).