I am aware of the Berkeley Campus Code of Student Conduct and acknowledge that any academic misconduct on this exam will lead to a “F”-grade for the course and that the misconduct will be reported to the Center for Student Conduct.

Sign your name: ________________________________

Print your class account login: cs161-_____ and SID: __________________________

Your TA’s name: ________________________________

Number of exam of person to your left: ____________ Number of exam of person to your right: ____________

You may consult two sheets of notes (each double-sided). You may not consult other notes, textbooks, etc. Calculators and computers are not permitted. Please write your answers in the spaces provided in the test. We will not grade anything on the back of an exam page unless we are clearly told on the front of the page to look there.

You have 170 minutes. There are 9 questions, of varying credit (300 points total). The questions are of varying difficulty, so avoid spending too long on any one question.

---

<table>
<thead>
<tr>
<th>Question</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points:</td>
<td>37</td>
<td>30</td>
<td>32</td>
<td>38</td>
<td>29</td>
<td>38</td>
<td>34</td>
<td>30</td>
<td>32</td>
<td>300</td>
</tr>
<tr>
<td>Score:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 1  \textit{Impersonation}  \hfill (37 points)

(a) (12 points) For each of the following, circle either \textbf{TRUE} or \textbf{FALSE}.

As a web site operator, one way to fully defend your users against clickjacking is to avoid using frames on your site. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}

Sandboxing browser components is an effective technique for defending users against phishing. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}

Websites use frame-busting to ensure that their pages aren’t included in a frame on another site. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}

Most browsers allow Javascript to modify the appearance and offset of a cursor. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}

Phishing can only occur if a victim clicks on a link of the phisher’s devising. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}

Spear-phishing has a higher rate of success than normal phishing. \hspace{5cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
(b) (25 points) Briefly answer each of the following:

1. Identify an accessibility issue with the use of CAPTCHAs.

2. Do today’s CAPTCHAs effectively prevent modern attackers from registering large numbers of accounts (each of which requires solving a CAPTCHA to complete the registration)?
   
   Circle **YES** or **NO** and briefly explain why.

3. Suppose a site implements a CAPTCHA by presenting users with four images and asking them to identify the one that shows a dog. Discuss a weakness with this particular approach (different from any criticism of CAPTCHAs you discussed in the earlier questions).

4. Does ensuring *visual integrity* sufficiently protect against clickjacking?
   
   Circle **YES** or **NO** and briefly explain why.

5. Briefly explain how *Browser-in-Browser* attacks can help phishers with conducting phishing attacks.
Problem 2  **Intrusion Detection**  

As a fresh IT security professional, your boss tasks you with evaluating an upcoming purchasing decision regarding several competing NIDS products. The three NIDS under consideration are (i) *AnomSpotter*, an anomaly-based zero-day detector, (ii) *JohnHancock*, a signature-based system, and (iii) *Spec-n-Behave*, a system that includes both specification-based and behavioral detection.

(a) For each of the following detection styles, briefly write down one advantage and one disadvantage:

- **Anomaly-based**
  - +
  - –

- **Signature-based**
  - +
  - –

- **Specification-based**
  - +
  - –

(b) JohnHancock and AnomSpotter come with explicit guarantees on the rate of false positives, advertising \( \frac{1}{100,000} \) and \( \frac{1}{10,000} \) respectively. Spec-n-Behave comes with an explicit guarantee on the rate of false negatives, advertising \( \frac{3}{1,000,000} \). **Circle** which tool would perform **best** in your company’s environment:

- AnomSpotter
- JohnHancock
- Spec-n-Behave
- Impossible to determine with information given
(c) In order to understand the nature of threats your company faces, you dig through the incident reports of the past two years. In 2011, your company got owned through numerous well-known and unsophisticated remote code-exec exploits of unpatched services.

Circle which tool would have been single most practical one for effectively and efficiently detecting these attacks?

- AnomSpotter
- JohnHancock
- Spec-n-Behave
- All would work equally well for this

(d) Looking then through the incident reports of 2012, you observe a shift towards compromises due to web attacks. The attacker tools operated in a very stealthy fashion, but a sharp member of the security staff found it was possible to identify the malicious web requests due the order of the HTTP headers in those requests. While HTTP does not require any particular order, the staffer noticed that benign requests happen to use a different order.

Assume you would not have known in advance that the attack traffic would have this appearance. Circle which tool would have been single most practical one for effectively and efficiently detecting these attacks?

- AnomSpotter
- JohnHancock
- Spec-n-Behave
- All would work equally well for this

(e) Circle which system would require the least amount of initial effort for the security team to deploy?

- AnomSpotter
- JohnHancock
- Spec-n-Behave
- All require about the same initial effort
Problem 3  \textit{Anonymity and Censorship} \hfill (32 points)

(a) For each of the following, circle either \textbf{TRUE} or \textbf{FALSE}.

\begin{itemize}
  \item Paid 3rd party proxy services provide the same anonymity properties as Onion Routing networks. \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
  \item Attackers can easily achieve strong anonymity while it is more difficult for the average Internet user (without the usage of networks such as Tor). \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
  \item Censorship systems always work by identifying keywords. \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
  \item Censorship systems and NIDS have similar properties. \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
\end{itemize}

(b) For each of the following, fill in the blank.

Assume correct usage of an Onion Routing network (such as Tor), configured to use 3 mixes. Of these 3, how many must be honest to preserve the confidentiality of your IP address? \underline{_____}

State a generally effective way to defeat keyword-based censorship that doesn’t rely upon anonymity:

For each of the following questions write a \textit{short} explanation.

(c) \textbf{In at most 2 sentences}, describe an issue/attack that users of anonymity systems may encounter. Such an issue or attack must harm the anonymity of the user.

(d) \textbf{In 1 sentence}, what can an in-path censor do that an on-path censor cannot?

(e) \textbf{In 1 sentence}, what advantage does an on-path censor have over an in-path censor?

(f) You suspect you are being attacked by an on-path censorship device. \textbf{In at most 2 sentences}, describe how you can detect such an attack. Your technique should be able to differentiate between network connectivity problems and censorship.
Problem 4  \textit{TLS and DNSSEC} \hspace{1cm} (38 points)

(a) Oski wants to securely communicate with CalBears.com using TLS. Which of the following things must Oski trust in order to communicate with confidentiality, integrity, and authenticity? \textbf{Circle all that apply}.

- The operators of CalBears.com
- Oski’s Computer (HW and SW)
- The designers of the cryptographic algorithms
- Computers on Oski’s local network
- The operators of CalBears.com’s authoritative DNS servers
- The entire network between Oski and CalBears.com
- CalBears.com’s Certificate Authority (CA)
- All the CA’s that come configured into Oski’s browser
- All the CA’s that come configured into CalBears.com’s software
- The operators of .com’s Authoritative DNS servers
- The operators of the Authoritative DNS Root servers

(b) For each of the following, circle either \textbf{TRUE} or \textbf{FALSE}.

- TLS uses asymmetric cryptography to directly encrypt the contents of communications: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- Certificates are a necessary component of TLS: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- TLS always provides Perfect Forward Secrecy: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- TLS always provides CalBears.com with authenticity of Oski’s identity: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- TLS allows both parties of a communication to contribute to key generation: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- TLS allows the use of stream ciphers: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- For transmitting data from a client to a server, a TLS session uses more than one key: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
- Clients finalize the choice of the TLS ciphersuite: \hspace{1cm} \textbf{TRUE} \hspace{1cm} \textbf{FALSE}
After taking CS161 you’ve seen the light and wish to help spread the gospel of DNSSEC. You realize one way to help speed up adoption would be combine DNSSEC and TLS in some way to eliminate a (perceived?) problem with TLS. *Hint: It’s about trust.*

(c) Fill in the blanks: While TLS provides ________________ security, DNSSEC provides ________________ security.

(d) Consider our goal of eliminating a *trust* problem with the existing TLS design, along with your answer to the previous question (4 c). Identify a key component of TLS that could instead rely on the DNSSEC infrastructure. It’s ok if this modification requires changes to the usage of TLS and DNSSEC. **Only identify a single component.**

(e) In **at most 2 sentences**, describe the change.

(f) Assume end-to-end DNSSEC deployment as well as full deployment of your change *(good job!)*. Oski wants to securely communicate with CalBears.com using TLS. Which of the following things must Oski trust in order to communicate with confidentiality, integrity, and authenticity? **Circle all that apply.**

- The operators of CalBears.com
- Oski’s Computer (HW and SW)
- The designers of the cryptographic algorithms
- Computers on Oski’s local network
- The operators of CalBears.com’s authoritative DNS servers
- The entire network between Oski and CalBears.com
- CalBears.com’s Certificate Authority (CA)
- All the CA’s that come configured into Oski’s browser
- All the CA’s that come configured into CalBears.com’s software
- The operators of .com’s Authoritative DNS servers
- The operators of the Authoritative DNS Root servers

(g) In **at most 2 sentences**, why you think this change in trust is either good or bad?
Problem 5  \textit{Reasoning About Memory Safety} \hspace{1cm} (29 points)

The following code takes two strings as arguments and returns a pointer to a new string that represents their concatenation:

```c
char *concat(char s1[], int n1, char s2[], int n2)
{
    int i, j;
    int n = n1 + n2;
    char *s;

    if (n1 < 0 || n2 < 0 || n <= 0) return 0;

    s = malloc(n);
    if (!s) return NULL;

    for (i=0; s1[i] != '\0'; ++i)
        s[i] = s1[i];

    for (j=0; s2[j] != '\0'; ++j)
        s[i+j] = s2[j];

    s[i+j] = '\0';
    return s;
}
```

The function is intended to take two strings and return a new string representing their concatenation. If a problem occurs, the function’s expected behavior is undefined.

(a) For the three statements assigning array elements, write down \texttt{Requires} predicates that must hold to make the assignments memory-safe:

1. /* "Requires" for line 13:
   *
   *
   */
   s[i] = s1[i];

2. /* "Requires" for line 16:
   *
   *
   */
   s[i+j] = s2[j];

3. /* "Requires" for line 18:
   *
   *
   */
   s[i+j] = '\0';
Here is the same code again, with more space between the lines. Indicate changes (new statements or alterations to the existing code, plus a relevant precondition for calling the function) necessary to ensure memory safety. Do not change the types of any of the variables or arguments.

```c
/* Precondition:
 * *
 * *
 */
1 char *concat(char s1[], int n1, char s2[], int n2)
2 {
3     int i, j;
4     int n = n1 + n2;
5     char *s;
6
7 if (n1 < 0 || n2 < 0 || n <= 0) return 0;
8
9     s = malloc(n);
10    if (!s) return NULL;
11
12    for (i=0; s1[i] != '\0'; ++i)
13        s[i] = s1[i];
14
15    for (j=0; s2[j] != '\0'; ++j)
16        s[i+j] = s2[j];
17
18    s[i+j] = '\0';
19    return s;
20 }
```
Problem 6  **Buffer Overflow Defenses and Exploits**  (38 points)

(a) The following code is compiled using an IA-32 (x86) compiler that implements *stack canaries*:

```c
void copy()
{
    int my_buf[2];
    ... /* Point where we draw the stack diagram */
}
```

In the following stack diagram, write in each empty box the value that would be held there once we reach the point specified in the `copy()` code (assume we’ve given you the correct region of the stack). Use the following values:

<table>
<thead>
<tr>
<th>Canary</th>
<th>my_buf[0]</th>
<th>my_buf[1]</th>
<th>RIP</th>
<th>SFP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(b) The following table lists four defenses against buffer overflows. For each defense, **mark with an X** which of the following attack methods the defense will **prevent**. The attacker can only place shellcode in buffers **on the stack**. Consider techniques **in isolation**, not combined. If the defense fails to prevent all of these methods, mark the entry for “None”.

<table>
<thead>
<tr>
<th>A</th>
<th>Return-oriented programming (ROP)</th>
<th>D</th>
<th>Function pointer overwrite</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Forms of <em>Arc injection</em> besides ROP</td>
<td>E</td>
<td>Saved frame pointer overwrite</td>
</tr>
<tr>
<td>C</td>
<td>Return to stack pointer</td>
<td>F</td>
<td>Local control variable overwrite</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Stack canaries</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>None</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bounds checking</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stack address randomization</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-executable stack</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Problem 7  *Game of P0wns*  

(a) For each of the following, circle True if the statement is correct, or False if it is not.

**True  False**  
To date, the Stuxnet worm is the fastest-spreading worm that has appeared in the Internet.

**True  False**  
Worms that spread by randomly scanning Internet addresses will grow exponentially fast all the way until they have infected the entire vulnerable population.

**True  False**  
A common approach for creating metamorphic viruses uses encryption technology.

**True  False**  
Eliminating buffer overflows would completely prevent the problem of Internet worms.

**True  False**  
Eliminating buffer overflows would completely prevent the problem of botnets.

**True  False**  
Some modern intrusion detection systems analyze the effects of executing web programs (such as Flash) in controlled environments.

**True  False**  
*Social engineering* refers to fooling users into taking insecure actions.

**True  False**  
Malware has been known to sometimes cause physical hardware damage to infected systems.

**True  False**  
Internet worms and viruses first became a problem with the arrival of Code Red in 2001.

**True  False**  
Some viruses add their code to that of existing executables residing on disk.

**True  False**  
*Botnet Command-and-Control* refers to the US government’s command center that tracks malware epidemics as they propagate.

**True  False**  
The largest Internet worm ever infected approximately 400,000 systems.

**True  False**  
The longest that an Internet worm outbreak has persisted (systems remained infected by the worm) is approximately 3 months.

**True  False**  
One technique used by some Internet worms has been to inspect files on a newly infected system to find other systems to try to infect.

**True  False**  
Modern malware infections typically use multiple “zero day” exploits.

**True  False**  
*Bullet-proof hosting* refers to use of hardened web sites that provably have no XSS or SQL vulnerabilities.
**True  False**  In the context of botnets, *externalities* refers to the ability for botmasters to host their servers in countries outside the legal reach of the United States.

**True  False**  *Transaction generators* are a type of malware that infects a user’s browser and issues illicit banking transactions after the user has authenticated to their bank’s web site.

**True  False**  *Click fraud* refers to a type of phishing where users believe they clicked on a link taking them to a trusted site like a bank, but instead they go to the phisher’s site.

**True  False**  Users who attempt to buy products advertised by email spam usually receive a functional product in return.

**True  False**  In the context of the underground economy, *affiliates* are business partners who receive commissions (payment) for providing some service, such as bringing users to an underground site who then themselves make purchases.

**True  False**  The term *money mules* refers to hardworking members of the underground economy who will work many different aspects of an Internet attack in order to ultimately gain a profit.

**True  False**  *Pump-and-dump* refers to monetizing email spam by inducing fluctuations in the prices of certain stocks.

**True  False**  The underground economy service of *Pay-Per-Install* is one of the reasons why when systems become infected today, they often wind up having multiple infections at the same time.

(b) List 4 goods and/or services that attackers can buy on the “underground economy”. Each should be quite different from the others.

(c) Roughly how much does it cost on the underground to purchase installations of malware on multiple Internet systems? State a quantity, a price, and one factor that influences the price.
Problem 8  Bugs in Key Generation  
This problem concerns the impact of potential bugs in cryptographic key generation. The threats are Eve, an eavesdropper, and/or Mallory, a MITM attacker. Assume neither Eve nor Mallory can conduct massive brute-forcing attacks ($\geq 2^{64}$ attempts).

(a) In 2011, a bug was found in the library provided with Ruby for generating RSA public/private key pairs. The buggy library would always generate $e = 1$. The library would then compute the corresponding $d$ using the normal algorithm for deriving $d$ from $e$.

For this problem, assume that Alice wants to send a single message to Bob (one-way communication), solely using public-key cryptography for security. Assume that Alice and Bob have a secure means for exchanging their public keys.

1. Suppose Alice generated her public/private key pair using the buggy Ruby library, but Bob used a secure library without the flaw. If Alice communicates with Bob using RSA-based cryptography for confidentiality, Circle which of the following describes the impact of the bug:
   - No effect in this scenario
   - Bob can read Alice’s message, but so can Eve
   - Bob will not be able to read Alice’s message

2. Suppose instead Bob generated his public/private key pair using the buggy Ruby library, but Alice used a secure library without the flaw. Again, if Alice communicates with Bob using RSA-based cryptography for confidentiality, Circle which of the following describes the impact of the bug:
   - No effect in this scenario
   - Bob can read Alice’s message, but so can Eve
   - Bob will not be able to read Alice’s message

3. Suppose Alice generated her public/private key pair using the buggy Ruby library, but Bob used a secure library without the flaw. If Alice and Bob use RSA-based public key cryptography for authentication of a message sent by Alice to Bob (do not concern yourself with messages from Bob to Alice), Circle which of the following describes the impact of the bug:
   - No effect in this scenario
   - Alice’s message will not verify to Bob as properly signed
   - Mallory can construct a message that appears to have a valid signature from Alice
4. Suppose instead Bob generated his public/private key pair using the buggy Ruby library, but Alice used a secure library without the flaw. Again, if Alice and Bob use RSA-based public key cryptography for authentication of a message sent by Alice to Bob, Circle which of the following describes the impact of the bug:

- No effect in this scenario
- Alice’s message will not verify to Bob as properly signed
- Mallory can construct a message that appears to have a valid signature from Alice

(b) Instead of the Ruby library bug occurring when generating RSA private/public key pairs, suppose the library had a bug where for Diffie-Hellman key exchanges it always used \( g = 1 \).

If both Alice and Bob used the library with this new bug, Circle which of the following describes the impact of the bug:

- No effect in this scenario
- Alice and Bob will fail to agree upon the same key using the key exchange
- Alice and Bob will agree upon the same key, but Eve can also determine the value of the key
- Alice and Bob will agree upon the same key, and while Eve cannot determine the value of the key, Alice and Bob lose perfect forward secrecy.

(c) Suppose Alice and Bob send new messages to each other every few minutes, so they decide to establish long-term confidential communication using symmetric-key cryptography. To do so, they first securely agree upon a secret key, \( K \). They then use a library that employs \( AES-256 \) in \( CTR \) mode as a stream cipher. The CTR mode counter is managed in the usual fashion. The library generates a fresh nonce every hour.

Circle which of the following best describes this scenario:

- Alice and Bob will be able to communicate confidentially.
- In some situations, Eve may be able to learn some information about the contents of Alice and Bob’s communication.
- In some situations, Eve may be able to learn some information about the contents of Alice and Bob’s communication and some information about \( K \).
- Eve can consistently recover complete messages from Alice and Bob’s communication.
- Eve can consistently recover complete messages from Alice and Bob’s communication and can recover \( K \).
Problem 9  Composition of Cryptographic Tools  (32 points)

For this problem, assume that Alice wants to send a single message $M$ to Bob. To do so, Alice and Bob can potentially use a number of different approaches and cryptographic technologies, which we will describe using the following terminology:

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>Plaintext for a single message</td>
</tr>
<tr>
<td>$A</td>
<td></td>
</tr>
<tr>
<td>$K_A$</td>
<td>Alice’s public key</td>
</tr>
<tr>
<td>$K_A^{-1}$</td>
<td>Alice’s corresponding private key</td>
</tr>
<tr>
<td>$K_B$</td>
<td>Bob’s public key</td>
</tr>
<tr>
<td>$K_B^{-1}$</td>
<td>Bob’s corresponding private key</td>
</tr>
<tr>
<td>$E_K$</td>
<td>Public-key encryption using RSA with the public key $K$</td>
</tr>
<tr>
<td>$\text{Sign}_{K^{-1}}$</td>
<td>Public-key signing using RSA with the private half of $K$.</td>
</tr>
<tr>
<td>$s_k$</td>
<td>Symmetric cryptography key</td>
</tr>
<tr>
<td>$\text{AES}_{s_k}$</td>
<td>Symmetric-key encryption using AES-256 in CBC mode, with the key $s_k$</td>
</tr>
<tr>
<td>$\text{AES-EMAC}_{s_k}$</td>
<td>Keyed MAC function presented in lecture, using the key $s_k$</td>
</tr>
<tr>
<td>$\text{PRNG}_{s_k}$</td>
<td>Bit-stream from a cryptographically strong pseudo-random number generator, seeded with $s_k$</td>
</tr>
<tr>
<td>$IV$</td>
<td>An Initialization Vector randomly generated for each use</td>
</tr>
<tr>
<td>SHA</td>
<td>SHA-256 hash function</td>
</tr>
</tbody>
</table>

You can assume that the public keys have been securely distributed, so Alice and Bob know their correct values.

Consider the following properties that Alice and Bob might desire their communication to have: Confidentiality, Integrity, Authentication, and Non-Repudiation. For each of the following possible communication approaches, circle which of these properties will securely hold in the presence of Mallory, a MITM attacker. Circle None if none of the properties hold. If an approach fails entirely (requires a step that either Alice or Bob cannot execute), circle Broken (and do not bother with the others).

(a) Optional: List here any assumptions/clarifications you feel you must make when answering the other parts. (We do not expect you to need to list anything here.)
(b) Alice sends to Bob: $E_{K_A}(M || \text{Sign}_{K_A^{-1}}(\text{SHA}(M)))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(c) Alice sends to Bob: $E_{K_B}(M || \text{Sign}_{K_B^{-1}}(\text{SHA}(M)))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(d) Alice sends to Bob: $E_{K_A}(M), \text{Sign}_{K_B^{-1}}(\text{SHA}(M))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(e) Alice sends to Bob: $E_{K_B}(M), \text{Sign}_{K_A^{-1}}(\text{SHA}(M))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(f) Alice generates a new symmetric key $s_k$ and sends to Bob:

$E_{K_A}(s_k), E_{K_B}(s_k), \text{AES}_{s_k}(M)$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(g) Alice generates a new symmetric key $s_k$ and sends to Bob:

IV, $E_{K_A}(s_k), E_{K_B}(s_k), M \oplus \text{PRNG}_{s_k} \oplus IV$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(h) Alice generates new symmetric keys $s_{k_1}$ and $s_{k_2}$, and sends to Bob:

$E_{K_A}(s_{k_1}), E_{K_A}(s_{k_2}), E_{K_B}(s_{k_1}), E_{K_B}(s_{k_2}), \text{AES}_{s_{k_1}}(M), \text{AES-EMAC}_{s_{k_2}}(\text{SHA}(M)), \text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_1})), \text{Sign}_{K_A^{-1}}(\text{SHA}(s_{k_2}))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken

(i) Alice and Bob privately exchange symmetric keys $s_k$ and $s'_k$ in advance. Alice later uses these keys to send to Bob: IV, $\text{AES}_{s_k \oplus IV}(M)$, $\text{AES-EMAC}_{s'_k \oplus IV}(\text{SHA}(M))$

Confidentiality  Integrity  Authentication  Non-Repudiation  None  Broken