Exam #1

COMP-5370/-6370

Released: 09Sept2021

Due: 13Sept2021 at 6pm CT

This is a take-home exam and you are welcome to use the lecture-slides (available on the website) and any notes you took. You are **not** allowed to use the Internet, lecture videos/recordings, or collaborate with others in any way whatsoever. The questions have specifically been designed to not require deeper technical knowledge than what was discussed in-lecture so there should be no need to add new, external information.

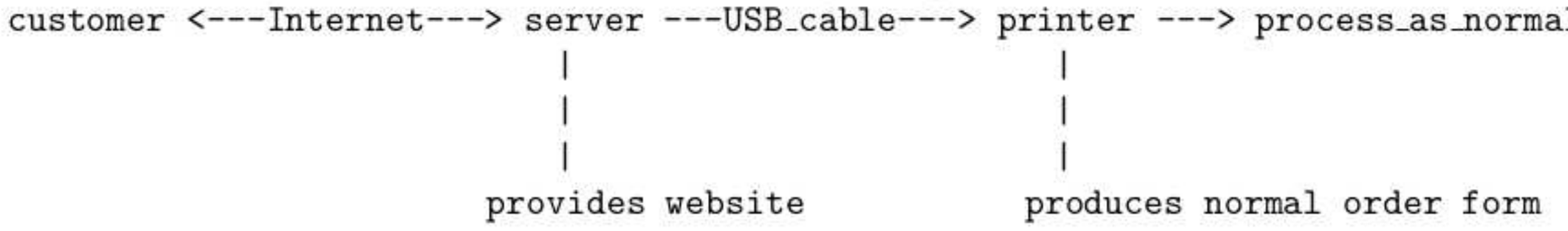
If you have questions, please contact the instructor or TA via Discord or email at any time and we will respond as soon as possible. Do **not** use the Discord channels (#general, #distance, etc.) to ask questions about any aspect of the exam.

1. **Threat Modeling**

   You have been hired by Widgets-R-Us to conduct a non-technical security review of their proposed online sales system. Widgets-R-Us is paying to you analyze the high-level characteristics of the system they want and help them understand how they should write the contract for the company they are hiring to implement their proposed system. The description given to you by Widgets-R-Us of their envisioned workflow consists entirely of:

   1. Customer goes to the website and selects how many widgets they wish to buy
   2. Customer enters their name, address, and credit card information and clicks the "Buy Widgets Now" button
   3. The order information is printed to the sales counter at their existing brick-and-morter store
   4. Orders are processed, billed, fulfilled, and shipped identical to the company's well-tested and highly-secure process for handling telephone orders.

   The proposed system's architecture as supplied by Widgets-R-Us:



   Additional details:

   - Online sales of widgets are highly competitive with very little overhead so they are not willing spend large amounts of money on development, maintenance, or upkeep of their system.
   - The people at Widgets-R-Us are smart, normal people who use the Internet in their own day-to-day lives but do *not* understand technical information or jargon and they only understand the most-basic security concepts.

     An example of a security concept that they do understand is "there are bad people and sometimes they attack other people on the Internet"

     An example of a security concept that they do not understand is the security archetypes (Alice, Bob, Mallory, etc).

- Widgets-R-Us will deduct from your pay if they think you are being overly general or condescending to them up to and including refusing to pay you.

  For the purposes of this exam, you are being "paid" in points towards your grade.
- Widgets-R-Us did *not* hire you to implement their system nor micromanage the company that will implement their system.
- The company hired for implementing their system is well-respected and knowledgable in security-related details but is well-known to *only* implement what is required in the contract (i.e. what you tell Widgets-R-Us to put in).

Widgets-R-Us found the below template and provided it to you to fill-in for your response.

1. List specific aspects of the system that are worth protecting (minimum of 2, maximum of 4).
2. For each of the above aspects, describe why it is worth protecting (max of 1 sentence each).

   EXAMPLE: Aspect XXXX is worth protecting because ...
3. For each of the above aspects, list specific actors it should be protected against (maximum of 1 actor each, you may not reuse actors, max of 1 sentence describing each actor).

   EXAMPLE: Aspect XXXX should be protected against YYYY who may try to ZZZZ.
4. List specific aspects of the system that are **NOT** worth protecting (minimum of 2, maximum of 4).
5. For each of the above aspects, describe what the risk of not protecting that aspect is (max of 1 sentence each).

   EXAMPLE: If XXXX is compromised, it could result in ...
6. For each of the above aspects, describe why you believe it is not worth protecting (max of 1 sentence each, the rational behind each must be sufficiently unique).

   EXAMPLE: In my opinion, XXXX is not worth protecting because ...

2. **Evaluating Protocols** There are four protocols provided below. Each is similar to those we discussed at-length in-class but the presentation has been slightly altered to be more specific:

   - `PROTECTED()` means that an arbitrary mechanism protects this message. The specifics are out-of-scope and only shown for context.
   - The arguments to various primatives are clearly defined.

     `HMAC-SHA256(secret=x, input=y)` means an HMAC construction using the SHA256 hash where the value "x" is used as the secret and the value "y" is the input to the HMAC function.

     `IV=KDF(secret=s, label=cipher-key)` meas that a secure KDF function is used to generate the IV value for that cipher. The value "s" is used for the KDF secret and the label "cipher-key" is used for this specific derivation.
   - The value "s" is used to represent a shared secret between Alice and Bob.

   Unlike some of the protocols we discussed in-class, there are specific contextual details which are pertinent to this question:

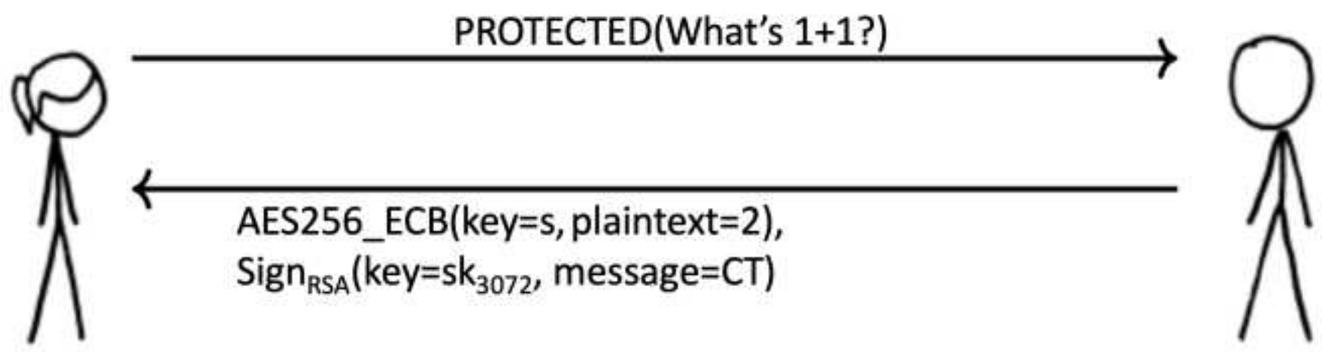
   1. All protocols are listed in their entirety (i.e. only one message each direction).
   2. No state is maintained across interactions.
   3. Each protocol is wholly independent from the others.
   4. There may be multiple concerns or zero concerns with each protocol.
   5. Points will be deducted for both missing concerns and marking non-issues as concerns.

   For each protocol answer one of the following:

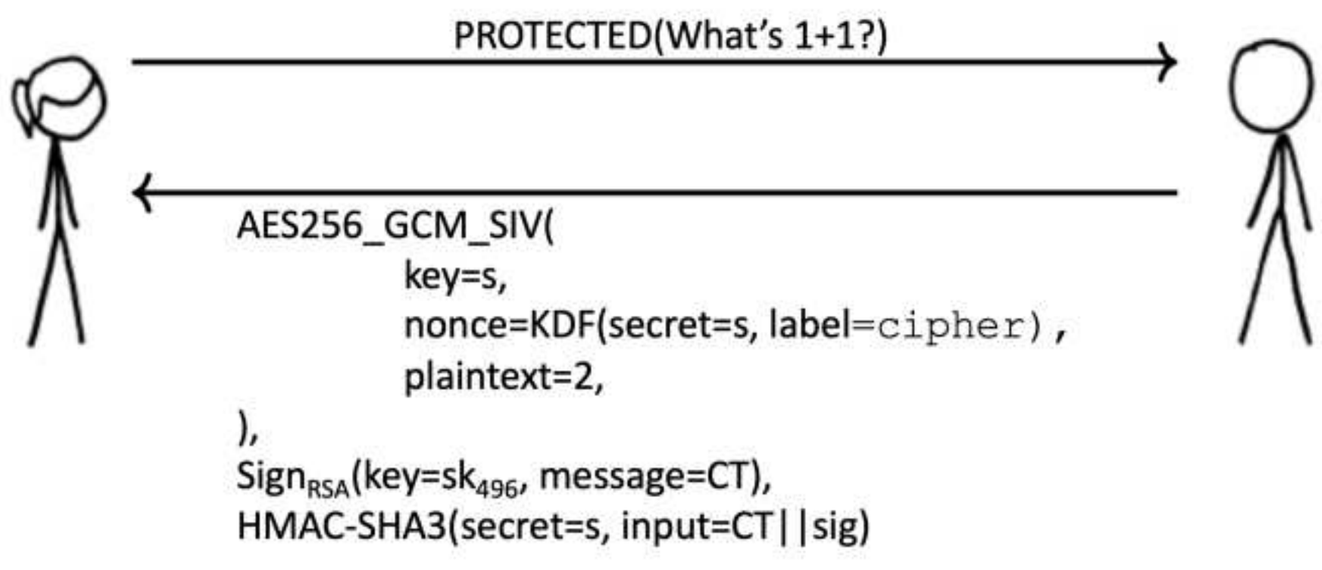
   1. If you think this is a reasonably-safe protocol, describe what specific aspects provide of each property of a secure channel (max of 1 sentence each).

2. If you think this is not a reasonably-safe protocol, answer the following (max of 1 sentence for each question per concern):
   - What specifically is wrong?
   - What specific property of a secure channel does this impact?
   - What is the minimal set of changes you can make to fix?
   - How confident are you that this is unsafe (low, Medium, HIGH)?

Protocol 1

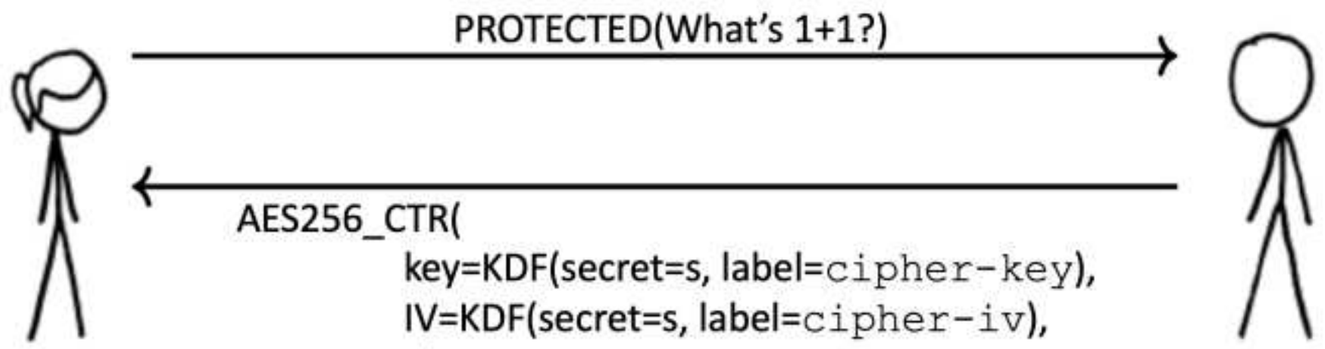


Protocol 2



Protocol 3

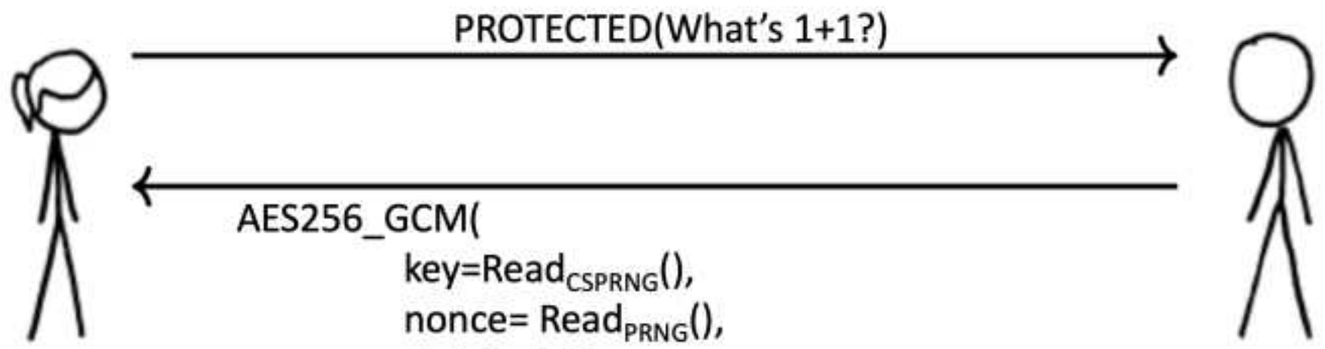
$digest_1$ & $digest_2$ are the HMAC w/ IV and CT inputs respectively.
The "IV" returned as the 2nd value and used in $digest_1$ is the KDF'd value used as the cipher IV.



AES256_CTR(
key=KDF(secret=s, label=`cipher-key`),
IV=KDF(secret=s, label=`cipher-iv`),
plaintext=2,
),
IV,
HMAC-SHA256(secret=KDF(secret=s, label=`hmac-secret`), input=IV),
HMAC-SHA256(secret=KDF(secret=s, label=`hmac-secret`), input=CT),
$Sign_{ECC}(key=sk_{ed25519}, message=digest_1||digest_2)$

Protocol 4

Similar to Protocol 3, the 2nd value returned and the associated data is the same value as the cipher nonce read from the PRNG.



AES256_GCM(
key=$Read_{CSPRNG}()$,
nonce= $Read_{PRNG}()$,
plaintext=2,
assoc_data= nonce,
),
nonce,
$Sign_{ECC}(key=sk_{ed25519}, message=CT)$

3. **Authentication** For each of the below constructions, decide whether or not this is a properly designed two-factor system. If so, describe what the factors are and what classes of factors they fall into (1 sentence max). If not, describe why not (3 sentence max). You may interpret each scenario as the input required to log onto a computer.

    1. **SCENARIO 1** Two independent passwords are required to log into a computer.
    2. **SCENARIO 2** A code from an always-on HOTP device (think "RSA hardware token") and a

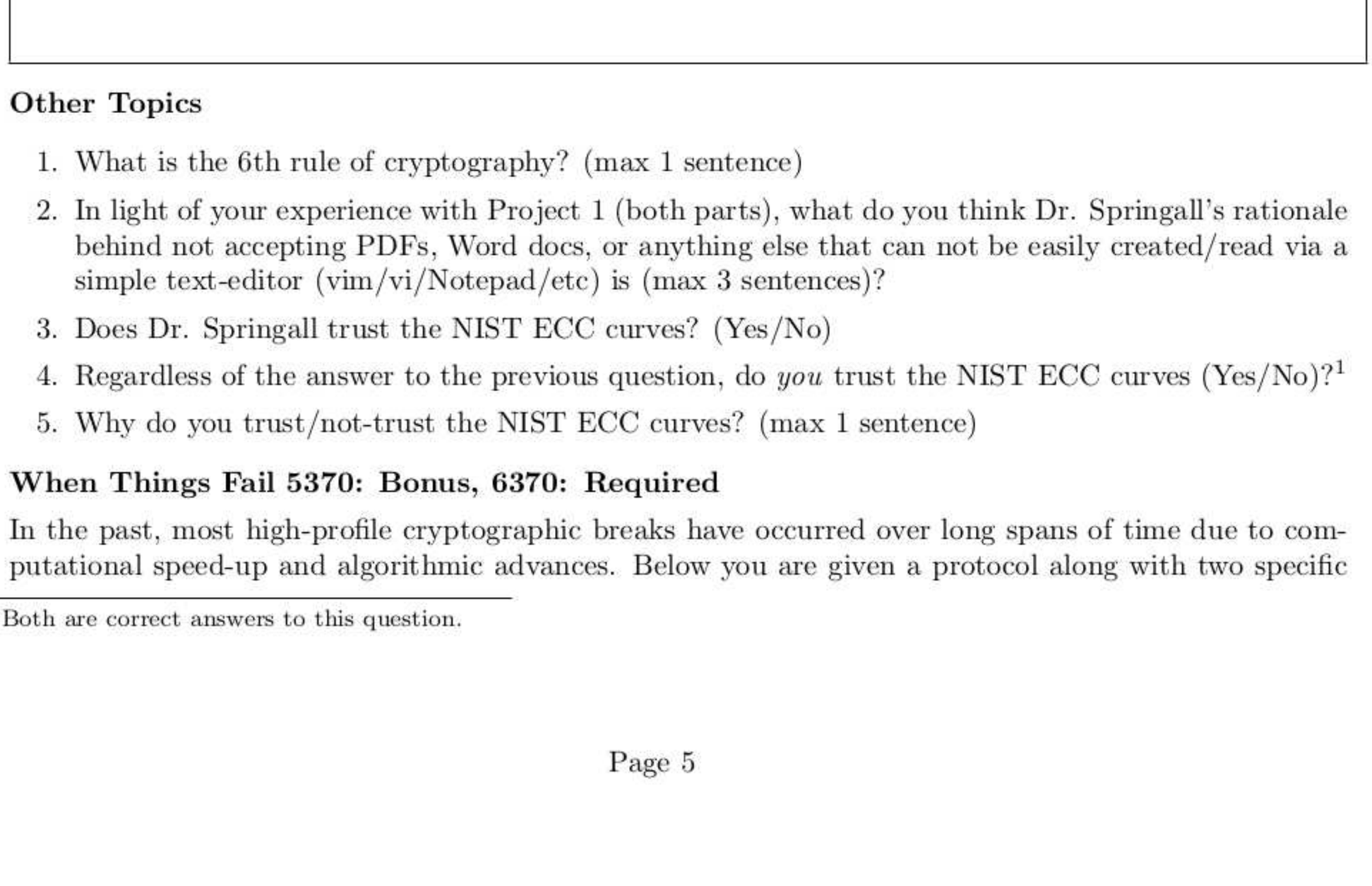
fingerprint are required to log onto a computer.

3. **SCENARIO 3** A code from a hardware device but that device requires a PIN to be entered prior to that code being generated (think "phone with virtual RSA hardware token app installed").
4. **SCENARIO 4** Your voice saying a secret passphrase. It is safe to assume ideal audio transcription as well as absolute physical security (i.e. no one over-hearing you speaking the passphrase).

4. **Working Backwards** Using the below protocol (Final Protocol from Lecture figure), what cryptography-related elements must Alice and Bob have before they can conduct the protocol?

   Clarifications:

   - Finite-field Diffie-Hellman for key exchange
   - RSA for authenticity



5. **Other Topics**

   1. What is the 6th rule of cryptography? (max 1 sentence)
   2. In light of your experience with Project 1 (both parts), what do you think Dr. Springall's rationale behind not accepting PDFs, Word docs, or anything else that can not be easily created/read via a simple text-editor (vim/vi/Notepad/etc) is (max 3 sentences)?
   3. Does Dr. Springall trust the NIST ECC curves? (Yes/No)
   4. Regardless of the answer to the previous question, do *you* trust the NIST ECC curves (Yes/No)?[1]
   5. Why do you trust/not-trust the NIST ECC curves? (max 1 sentence)

6. **When Things Fail 5370: Bonus, 6370: Required**

   In the past, most high-profile cryptographic breaks have occurred over long spans of time due to computational speed-up and algorithmic advances. Below you are given a protocol along with two specific

[1] Both are correct answers to this question.

scenarios in which major cryptographic breakthroughs have hypothetically occurred over night. You are asked to reason about what it means, what comes next, and what the impact could be.

The protocol designers specifically wanted the properties of message integrity and sender authenticity but did **not** care if anyone could see the content of their conversations (i.e. no confidentiality). Additionally, each response from Bob includes an incrementally increasing counter which you can assume never overflows and is maintained across interactions. You may assume this protocol is precisely the attributes that were intended and performed exactly as desired. You should focus only on what has been broken in each of the hypothetical scenarios below.

For each of the following scenarios, analyze what happens to the protocol and what attacks are now feasible in that scenario. Your analysis should go beyond discussion of security-related properties in their general form and include what attacks are now feasable that were not before. To be clear, these are isolated scenarios and each should be discussed alone.

1. **SCENARIO 1**: Integer factorization is now easy on composites numbers up to and including 2048-bit values.
2. **SCENARIO 2**: SHA256-based HMACs are now broken and valid HMACs may be generated for arbitrary content without knowledge of the key.