ECE 4156/6156

Hardware-Oriented Security and Trust

Final Exam

May 2, 2023

*This test is open book for the required as well as the optional textbooks. This test is also open note. Calculators are permitted but no programs may be used. Open notes include electronic notes with any handwritten or typed information you may have added. For electronic notes, all wireless connectivity must be turned off, and no cell phone is allowed; in other words, no connection to any information source outside of the open books, notes and items downloaded from the course website is allowed during the exam. Open notes also include all course assignments (homeworks or labs) as well as solutions (including both your solution as well as official solutions posted on the course website or passed out for previous exams). All aspects of the Georgia Tech Honor Code apply. Please follow the instructions precisely. If you need to continue a problem, please use the back of the **previous** page. The meaning of each question should be clear, but please state any assumptions and ask for clarification if necessary.*

Question	Score	Max
1		**5**
2		**5**
3		**10**
4		**10**
5		**10**
6		**20**
7		**20**
8		**10**
9		**20 or 30**
10		**30**
Total		**110 or 150**

You can do it!

Name (Please print)_____

This test will be conducted according to the Georgia Tech Honor Code.
I pledge to neither give nor receive unauthorized assistance on this exam and to abide by all provisions of the Honor Code.

Signed_____

1. **(5 pts.) Highest Security Levels.**

   One conclusion that most governments have reached is that most commercial solutions are not designed with sufficient security and trust to resist an adversary with the resources of a nation-state. There is a very good economic explanation for this conclusion: please explain why. Please limit your answer to 10 sentences or less. If you write more than 10 sentences, please circle or otherwise clearly indicate which 10 sentences you want graded.

2. **(5 pts.) Advanced Encryption Standard (AES).**

Has AES been formally proven to be secure? If so, please describe the nature of the formal proof (it is OK if you lack details, just describe the proof technique used at a very high level of abstraction). If not, then why do most people consider AES to be secure, i.e., on what is their trust in AES based?

3. **(10 pts.) Entropy, Randomness and NIST.**

Why are longer (as opposed to shorter) bit sequences important in the measurement of entropy? For full credit for this answer, please provide a minimum of one clear reason why longer bit sequences are important. Two reasons are fine as well; three reasons (or more) are also perfectly acceptable, but if some of the reasons given are unclear, vague or incorrect, points will be lost. If you give more than one reason in your answer, feel free to indicate (e.g., by circling) which reason or reasons you want graded for this problem.

4. **(10 pts.) Chained CBC Mode.**

For your reference, here is a picture of CBC mode – a new Initialization Vector (IV) is chosen each time:

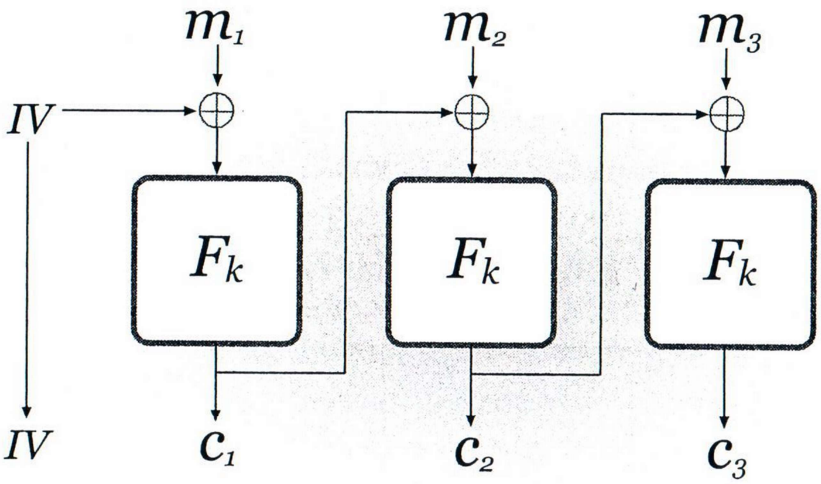


FIGURE 3.7: Cipher Block Chaining (CBC) mode.

However, this question is about Chained CBC mode. Here is a picture:

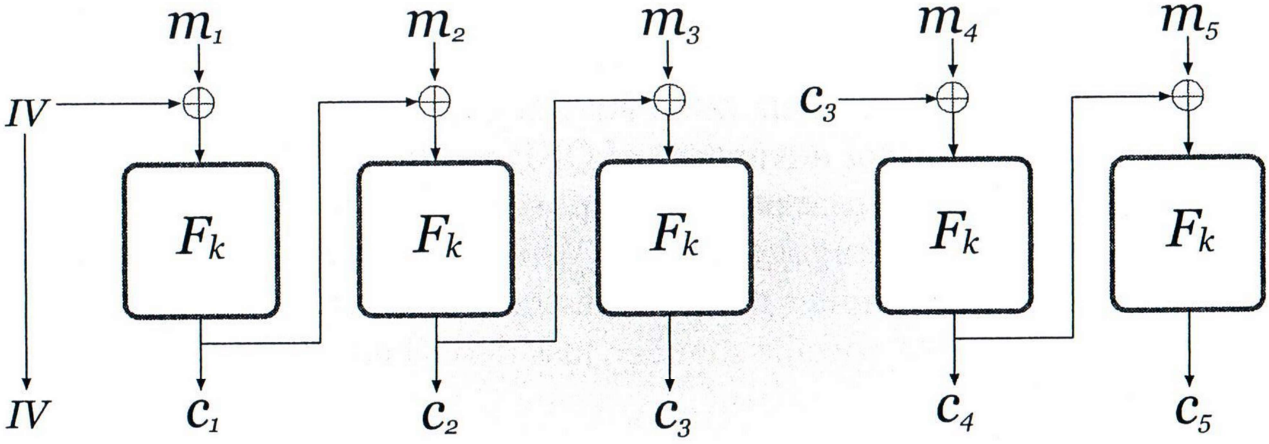


FIGURE 3.8: Chained CBC.

What is the effect of a single-bit error in the ciphertext when using the Chained CBC mode of operation? How is the effect of a single-bit error in the ciphertext when using Chained CBC mode different from CBC mode (i.e., CBC mode with a new IV each and every time)?

[PLEASE TURN TO THE NEXT PAGE]

You may write on the previous page if you prefer; alternatively, you may write your answer on this page. However, please limit your answer to approximately 10 sentences. Please note that for full credit you must correctly differentiate the effect of a single-bit error in CBC mode (i.e., with a new IV each and every time) versus Chained CBC mode.

5. **(10 pts.) MAC Design with a Keyed Pseudorandom Function.**

Consider a keyed pseudorandom function $F_K : \{0,1\}^n \rightarrow \{0,1\}^n$. Please note that this function F_K has a key K but is not a puedorandom permutation.

Can you design a Message Authentication Code (MAC) with this keyed pseudorandom function? If so, explain how and draw a diagram to help describe the MAC design. If not, explain why not with at least one clearly articulated reason why not. Please limit your answer to 10 sentences or less. Drawings and mathematical equations or formulas do not count towards the 10 sentence limit.

6. **(20 pts.) PUFs and Clonability.**

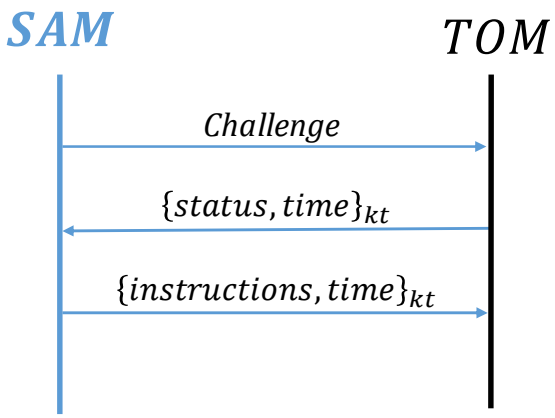
PUFs were introduced in class including the controversy that surrounds the name. In this problem you are asked to argue both sides of the debate. You will be graded based on the quality of your technical arguments used to advance each point of view.

(a) (10 pts.) Argue that PUFs exist which are impossible to clone (i.e., "unclonable"). For full credit, please give at least one very strong reason why it is likely that PUFs exist which are impossible to clone (you may use a specific PUF if you like, or you may argue in more general terms, but use technical reasons). For full credit, you must argue why model building and/or machine learning are unlikely to succeed (in other words, account for model building / machine learning in your argument as opposed to ignoring the possibility of model building or machine learning). Assume that the adversary has a year of time available with the PUF hardware in possession and has the resources of a large multinational corporation. Please limit your answer to 10 sentences or less.

(b) (10 pts.) Argue that while PUFs are certainly hard for you to clone, they are all very likely in the near future (say, in the next 20 years or so) to be able to be cloned in a reasonable amount of time and effort (say, once given the specific PUF on a specific microchip, the adversary has a year or less of time with resources of a large multinational corporation). You do not have to argue that all known PUFs can be cloned right now; you are only asked to argue that all PUFs are likely to be able to cloned at some point not too far away in the distant future (within the next 20 years or so). Please limit your answer to 10 sentences or less.

7. **(20 pts.) PUF Response as a Symmetric Key.**

One of the challenges with Internet-of-Things (IoT) devices is how to periodically update them with limited bandwidth and power. Asymmetric cryptography requires large prime numbers whose generation and complicated mathematics can potentially be avoided by use of a PUF. In the diagram to the right SAM is a server trying to send update instructions to TOM the IoT device. Both SAM and TOM have access to a public clock, and therefore they have access to "time" with an accuracy of one second.



Your team comes up with the approach described in the diagram shown above. SAM sends a challenge in plaintext to TOM. TOM uses the response as a key "kt" as shown in the diagram. TOM encrypts {status,time} with kt using AES. SAM sends the exact time received back together with instructions; SAM encrypts the message with kt and responds in less than 10 seconds. TOM does not have enough memory to store all challenges sent in the past. However, TOM takes more than ten seconds to process messages, and so TOM sets a policy of not accepting new challenges close together in time – in particular, any new challenge must be longer than 10 seconds from the previous challenge.

(a) (10 pts.) Describe how a replay attack could be carried out.

(b) (10 pts.) Does the replay attack work? Why or why not? Please note that a correct answer without valid reasons will receive zero points.

8. **(10 pts.) Needham-Schroeder Protocol with Symmetric Keys.**

Consider the Needham-Schroeder protocol with symmetric keys as presented in class (do **not** consider the version with asymmetric keys). What problem does it solve? Why not just use a Certificate Authority instead? Please limit your answer to 10 sentences or less.

9. **[ECE 4156] (20 pts.) [ECE 6156] (30 pts.) Diffie-Hellman Scenario and Attack.**

Consider the following scenario. Alice and Bob each have a personal computer (PC) and communicate using the internet. Their PCs are server-class and so can handle very large numbers, e.g., 4096 bits long. They can generate very large prime numbers and very large truly random numbers. However, Alice and Bob do not know each other from any previous interactions and do not, for example, know what each other looks like.

A Trusted Third Party (TTP) provides Alice and Bob with the first step in Diffie-Hellman key exchange, i.e., the TTP provides an appropriate prime number p and generator α of Z_p^*.

The TTP also provides entity authentication, i.e., the TTP can verify that a message claiming to be from Bob (or Alice) really is from Bob (or Alice). Entity authentication is carried out prior to Diffie-Hellman key exchange (which results in establishing a shared key K). Once Alice and Bob establish a shared key K, the TTP is no longer used.

Consider the following Diffie-Hellman steps carried out by Alice and Bob:

(i) Alice chooses a random secret x, $1 \leq x \leq p - 2$, and sends $(\alpha^x) \bmod p$ to Bob.

(ii) Bob chooses a random secret y, $1 \leq y \leq p - 2$, and sends $(\alpha^y) \bmod p$ to Alice.

(iii) Alice computes the shared secret $K = ((\alpha^y) \bmod p)^x) \bmod p$.

(iv) Bob computes the shared secret $K = ((\alpha^x) \bmod p)^y) \bmod p$.

(v) Alice throws away x but keeps $(\alpha^x) \bmod p$.

(vi) Bob throws away y but keeps $(\alpha^y) \bmod p$.

Now Alice and Bob communicate using a symmetric key encryption algorithm and key *K*. After the communication is complete, Alice and Bob each write the key *K* on a piece of paper. Alice and Bob keep files encrypted with *K* but delete the key *K* from their computers using a form of bit bleaching which is successful (i.e., the key cannot be recovered). In other words, the following steps occur:

(vii) Alice and Bob communicate using key *K* and store files encrypted with key *K*.

(viii) Alice and Bob each remove key *K* from their PCs.

Now an adversary carries out a physical attack on the PCs of Alice and Bob. However, Alice and Bob do manage to each burn their piece of paper with key *K* written on it; the result is that key *K* cannot be recovered from the paper. Unfortunately, the full PC memory is recovered for both PCs.

(a) (10 pts.) Assuming that the adversary can recover everything from the PCs of Alice and Bob except the information which has been ″thrown away″ (i.e., bit bleached), what information does the adversary have? Please list all relevant information including keys, files, and relevant Diffie-Hellman quantities.

(b) (10 pts.) Can the adversary decrypt the files? Please explain your answer in detail. Please note that only half of the points will be given for a correct answer without a clear and valid explanation; similarly, only half of the points (5 out of 10) will be given for an incorrect answer (a ″yes″ when the answer is ″no″ or a ″no″ when the correct answer is ″yes″) and correct (clear and valid) explanations showing technical competence with regard to Diffie-Hellman key exchange.

(c) **[ECE 6156 only]** (10 pts.) Suppose steps (v) and (vi) are modified to keep larger numbers:

(v) Alice throws away x but keeps α^x.

(vi) Bob throws away y but keeps α^y.

Can the adversary decrypt the files? Everything else is the same as before except for steps (v) and (vi) are modified as shown above. Please explain your answer in detail. Please note that only half of the points will be given for a correct answer without a clear and valid explanation; similarly, only half of the points (5 out of 10) will be given for an incorrect answer (a "yes" when the answer is "no" or a "no" when the correct answer is "yes") and correct (clear and valid) explanations showing technical competence with regard to Diffie-Hellman key exchange.

10. **[ECE 6156 only] (30 pts.)**

In the HELP PUF, TVComp calculates $zval_i$ for each PND_i from a set of PND measurements at a specific temperature and voltage combination. The formula used is as follows:

$$zval_i = \frac{(PND_i - \mu_{chip})}{Rng_{chip}}$$

Next, TVComp generates a new PUF Number Difference (PND) for each PND_i by use of the $zval_i$ value as follows:

$$PND_c = zval_i * Rng_{ref} + \mu_{ref}$$

The authors of the HELP PUF state that Rng_{ref} and μ_{ref} come from a "reference distribution."

(a) (10 pts.) Explain how this "reference distribution" could be obtained. It is OK if you are not sure, just state a reasonable approach. Please limit your answer to 10 sentences or less.

(b) (10 pts.) The authors of the HELP PUF claim that TVComp “expands” the challenge response space. Professor Mooney has even heard this claim as follows: “TVComp increases the entropy of HELP.” Argue **in favor of** the claim that TVComp expands the challenge response space of HELP and increases the resulting entropy. NOTE: you do not have to agree with the argument; you are only required to make the best argument you can. For full credit, please give at least one very strong reason why TVComp **increases** entropy. Please limit your answer to 10 sentences or less.

(c) (10 pts.) The authors of the HELP PUF claim that TVComp "expands" the challenge response space. Professor Mooney has even heard this claim as follows: "TVComp increases the entropy of HELP." Argue **against** the claim that TVComp expands the challenge response space of HELP and increases the resulting entropy. NOTE: you do not have to agree with the argument; you are only required to make the best argument you can. For full credit, please give at least one very strong reason why TVComp **does not increase** entropy. Please limit your answer to 10 sentences or less.

THIS IS THE LAST PAGE OF THE EXAM!