CS 463/563 — Cryptography for Cybersecurity
Homework 6: Public-Key Cryptography

DUE DATE HERE

STUDENT NAME HERE
UIN HERE
SECTION HERE
Instructor: J. Takeshita. Institution: Old Dominion University.

This assignment has **7** questions.

Submission instructions: If you don't already have the necessary tools, then install 1) a LaTeX distribution (probably the `texlive-full` package on most Linux distributions) and 2) an IDE or other text editor suitable for LaTeX. For the latter, I suggest the use of the Kile IDE, which is available in most Linux distributions' default packages. You can also just use a web IDE such as **Overleaf**.

Download the source file(s) for this assignment. Edit the file(s) to include your solutions using the text editor or LaTeX IDE of your choice. Type your answers and personal information (e.g., name, section, etc.) into the source file at the appropriate places, and double-check that the variable `solutions` is set to be false. (I should have done this for you, but it's good to be sure.) Make sure to properly format your math! See `https://latex-tutorial.com/tutorials/` if you're not already familiar with LaTeX.

Compile the source to a PDF, either through your IDE or by using the command `pdflatex`. Make sure your compilation has no errors, and address all reasonably actionable warnings (you can ignore small things like badbox warnings, as long as they don't affect the output badly). You can toggle the variable `boilerplate` to hide these instructions, if you like.

Turn in 1) the PDF and 2) any and all `.tex` source files, `.bib` bibliography files, source code, or other artifacts. Submissions that are handwritten or typeset with other methods will not be accepted! If you wrote any software in the course of this work, please include it using the `listing` package's utilities.

Cite any external sources, and include any code you write. The use of unauthorized outside assistance, including any AI/LLM, is strictly prohibited.

A note: While it is possible and allowed to solve these questions by writing your own computer programs, you should be able to also do them by hand, as similar questions may appear on exams.

Questions:

1. Using the Euclidean algorithm, compute $GCD(228, 44)$. Show your work. (3 points)

Answer:
$GCD(228 \pmod{44}, 44) = GCD(8, 44)$
$GCD(44 \pmod{8}, 8) = GCD(4, 8)$
$GCD(8 \pmod{4}, 4) = GCD(4, 0) = \boxed{4}$

2. Using the extended Euclidean algorithm, compute $9^{-1} \pmod{19}$.
Show your work. (5 points)

Answer:
$19 = 2 * 9 + 1$
$1 = 19 - 2 * 9$
$t = -2$
$-2 \pmod{19} = 17$
$17 * 9 \pmod{19} = 1$
$\boxed{17}$

3. Find $\varphi(3200)$. Show your work. (3 points)

Answer:
$3200 = 2^7 * 5^2$
$\boxed{(2^7 - 2^6)(5^2 - 5) = 1280}$

4. Using Fermat's little theorem, find the multiplicative inverse of 9 in $GF(19)$. Show your work. (4 points)

Answer:
Fermat's Little Theorem: $a^{-1} = a^{p-2} \pmod{p}$
$9^{19-2} = (9^2)^8 * 9 \pmod{19}$
$9^2 \pmod{19} = 5$
$9^{19-2} = (5)^8 * 9 \pmod{19}$
$9^{19-2} = (5^3)^2 * 5^2 * 9 \pmod{19}$
$5^3 \pmod{19} = 11$
$9^{19-2} = (11)^2 * 5^2 * 9 \pmod{19}$

$11^2 \pmod{19} = 7$
$5^2 \pmod{19} = 6$
$9^{19-2} = 7 * 6 * 9 \pmod{19} = \boxed{17}$
Check:
$17 * 9 \pmod{19} = 1$

5. Using Euler's theorem, find $5^{300} \pmod{31}$. Show your work, and make sure to show how you use the theorem. (5 points)

Answer:
Euler's Theorem: $a^{\varphi m} = 1 \pmod{m}$
$\varphi 31 = (31^1 - 31^0) = 30$
$5^3 \pmod{31} = 1$
$5^{30} = (5^3)^{10} = 1$

$5^{300} = (5^{30})^{10}$
$5^{300} = 1$

6. In the programming language of your choice, implement the regular and binary Euclidean algorithms. These functions should take in integers a, b and return $b^{-1} \pmod{a}$. Compare their relative performance[1] for large inputs, starting with numbers $\geq 2^{16}$. (15 points **extra credit**)

   Include your source code (use the `lstlistings` package), the data you generate, and any graphs, tables, etc. What do you notice about the performance of these two algorithms?

Answer: Your solution goes here, and similarly for all other questions. Reference your code using `\Cref{}`.

7. Give a new question on the topic of this week's material and its solution, suitable for inclusion in future versions of this homework assignment. Your question should be different from the other questions here – don't just change some values around! Programming problems are acceptable too, but should be nontrivial. You can use

[1] For an example of how to time functions in C++, see the file plain_psi.cpp, lines 76-84. There are other ways, but this is the most reliable.

textbooks, scholarly papers, or other reputable sources of information for inspiration, but you should not directly copy someone else's work. Your question should show me that you understand the content well enough to ask and answer an interesting question about it. Include any source code used to generate/solve the problem. (10 points)

Some free online textbooks include *The Joy of Cryptography* by Mike Rosulek (Oregon State University), or *A Graduate Course in Applied Cryptography* by Dan Boneh (Stanford University) and Victor Shoup (Offchain Labs, formerly at New York University). Your course text, *Understanding Cryptography* by Christof Paar (Universität Bochum) and Jan Pelzl (Universität Bochum), is also an excellent starting point.

Possible topics include the Euclidean and extended Euclidean algorithms, Euler's φ function, Fermat's little theorem, Euler's theorem, or the RSA cryptosystem.

Answer: Use the Extended Euclidean algorithm to find $17^{-1} \pmod{31}$
t in $sr_0 + tr_1$ is the inverse
$r_0 = 31 = 1 * 17 + 14$
$r_1 = 17 = 1 * 14 + 3$
$r_2 = 14 = 3 * 4 + 2 = r_0 - r_1$
$r_3 = 3 = 2 * 1 + 1 = r_1 - (r_0 - r_1) = 2r_1 - r_0$
$r_4 = 2 = 1 * 2 + 0 = r_0 - r_1 - 4(2r_1 - r_0) = 5r_0 - 9r_1$
$r_5 = 1 = 1 * 1 + 1 * 0 = 2r_1 - r_0 - 1(5r_0 + 9r_1) = -6r_0 + \boxed{11} r_1$
$\boxed{11 = 17^{-1} \pmod{31}}$