Cryptography
Project: CryptoBench

Martin Harrigan, martin.harrigan@setu.ie

This project is due on **6th December 2024** and is worth 30**% of your final grade**. You must submit a report containing a link to the code and data on Blackboard.

For the project you are required to benchmark the implementations of a selection of asymmetric cryptography functions using Python's `cryptography` package and produce a report detailing your results. Specifically, you are required to benchmark five processes: keypair generation, asymmetric encryption, asymmetric decryption, digital signing and signature verification. For each process you are required to compare the RSA, DSA, and ECC implementations, where applicable.

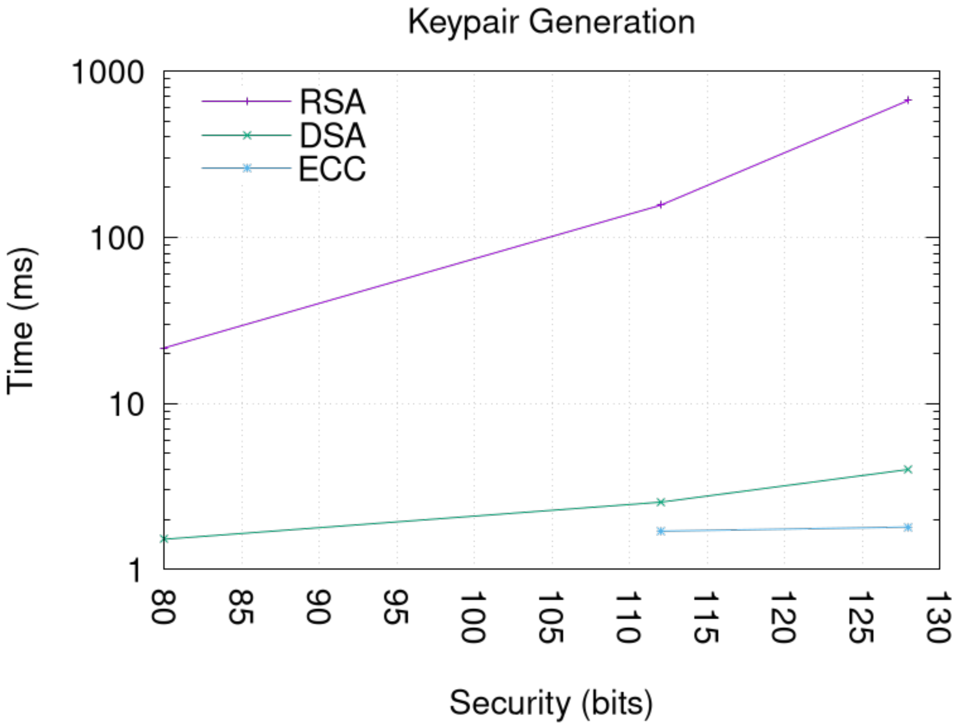

Figure 1: Keypair Generation using the RSA, DSA, and ECC algorithms.

Keypair Generation You are required to generate keypairs using the RSA, DSA, and ECC algorithms. For each algorithm you are required to generate keypairs that provide 80-, 112-, 128-, 192- and 256-bit security, where possible. Report your results in a table and compare the generation times using a plot similar to Fig. 1. For each result, you should repeat the process a fixed number of times and compute the average of all but the first. For example, suppose I try to compute a 1024-bit RSA keypair and the times reported are 25.10ms, 19.84ms and 18.90ms. I should ignore the first time and compute the average of the other two (19.37ms). When performing your benchmarks try to compute the average over a large number of runs, e.g., 10 times or 100 times where possible.

Asymmetric Encryption You are required to encrypt a plaintext message using the RSA algorithm. The plaintext should be as long as the algorithm permits or 10KB if there is no restriction. You are required to use the various keypairs generated above (80-, 112-, 128-, 192- and 256-bit security). For each result, you should repeat the process and compute an average as above. Report your results in a table and compare the times using a plot.

Asymmetric Decryption You are required to decrypt the ciphertexts produced above using the RSA algorithm and the various keypair sizes. For each result, you should repeat the process and compute an average as above. Report your results in a table and compare the times using a plot.

Digital Signing You are required to digitally sign a message using the RSA, DSA and ECC algorithms. You should sign the hash of the message rather than the message directly. For each algorithm you are required to use the various keypairs generated above (80-, 112-, 128-, 192- and 256-bit security), where possible. Specify which hash function you used in each case. For each result, you should repeat the process and compute an average as above. Report your results in a table and compare the times using a plot.

Signature Verification You are required to verify the signatures produced above using the RSA, DSA and ECC algorithms and the various keypair sizes. For each result, you should repeat the process and compute an average as above. Report your results in a table and compare the times using a plot.

Format of your Report Your report should have the following format:

1. Introduction and Experimental Setup: What are the specifications of the machine you are using? What OS are your using? What version of Python and what version of the `cryptography` package are you using? There should be enough detail for a third-party to replicate your setup.
2. Results
   (a) Results for Keypair Generation
   (b) Results for Asymmetric Encryption
   (c) Results for Asymmetric Decryption
   (d) Results for Digital Signing
   (e) Results for Signature Verification
3. Conclusion: Were your results inline with your expectations? Did you observe any unexpected results? Can you explain those outliers?

You should include a link to your Python code and the resulting data in your report. The code should be written so that it can be executed in one shot to reproduce all of the results. The contents of the report are more important than a word count. Did you make all of the required comparisons? Are your plots clear and legible? You will likely need 500–1000 words to document everything.

Reminder "All work submitted by learners for assessment purposes, or for written or oral publication, must be their own work. Where this is informed by the work of others, the source shall be properly attributed and referenced." See the SETU Carlow Full-Time Student Handbook, including Academic Regulations.

Frequently Asked Questions

Where can I put the code and data that I link to from the report?

You can share your code and data using a GitHub repository, a GitLab repository, your Microsoft OneDrive account, or by uploading it to Blackboard. If you want to make your GitHub repository private, you should add me as a collaborator (`https://github.com/setumartin`). If you want to make your GitLab repository private, you should add me as a collaborator (`https://gitlab.com/users/setumartin`). You should not make any changes to the linked code or data beyond the project deadline.

Are there are requirements on the structure or layout of the code?

The only requirement, stated above, is that the code should be written so that it can be executed in one shot to reproduce all of the results. In other words, running `python my_project.py` should produce all of the data needed for your plots.

What tool should I use to generate the plots, similar to the plot in Fig. 1?

You can use any plotting or charting tool, e.g., Microsoft Excel, Matplotlib, or Gnuplot.

What is the marking scheme?

	Marks (100)
Content	
Introduction and Experimental Setup: *Does the report adequately describe the machine specification, specify the versions of Python and the `cryptography` package, and include a link to the code?*	12.5
Results for Keypair Generation: *Does the report include data and plots for keypair generation using the RSA, DSA, and ECC algorithms.*	15
Results for Asymmetric Encryption: *Does the report include data and plots for asymmetric encryption using the RSA algorithm.*	15
Results for Asymmetric Decryption: *Does the report include data and plots for asymmetric decryption using the RSA algorithm.*	15
Results for Digital Signing: *Does the report include data and plots for digital signing using the RSA, DSA, and ECC algorithms?*	15
Results for Signature Verification: *Does the report include data and plots for signature verification using the RSA, DSA, and ECC algorithms?*	15
Conclusion: *Does the report discuss the results, compare the results with your expectations, and identify any anomalies?*	12.5