Cryptography Experiments: Hashing & Integrity Checker

This document summarizes the hashing and integrity-checking Python scripts included in the repository.

Files Included:

1. Expt2Simple.py ■filecite■turn1file0■
2. exp2.py / hash.py ■filecite■turn1file1■

1) Expt2Simple.py

A minimal example demonstrating SHA-256 hashing using Python's hashlib module.

Features:

- Computes SHA-256 digest of a fixed string.
- Prints original message and digest to the console.

2) exp2.py / hash.py

A more robust integrity-checking utility that:

- Supports choosing hashing algorithm (defaults to sha256).
- Computes hashes for original and tampered text.
- Compares digests to verify integrity.
- Raises an error for unsupported algorithms.

How to run:

```
python Expt2Simple.py
python exp2.py # or python "hash.py" depending on filename
```

Example output (Expt2Simple.py):

```
Hello
```

Integrity-check example (exp2.py):

```
Enter hash algorithm (default sha256): sha256
Enter original text: Hello
Enter tampered text: Hell
Original Hash :
```

Tampered Hash :

Data has been tampered (hashes differ).

Project structure:

```
Cryptography-Experiments/
■■■ Expt1Caesar.py
■■■ Expt1 Vigenere.py
■■■ Expt2Simple.py
■■■ exp2.py (or hash.py)
■■■ README_Exp2.pdf
```

If you want a styled GitHub README.md generated and saved to file, I can create it as well.