# Module 20: Cryptography - Practical Guide

This guide details the specialized tools and hands-on procedures for performing cryptographic operations, key management, and integrity verification as required by the CEH v13 curriculum.

## 1. Comprehensive Cryptography Toolset

### File, Message, and Disk Encryption

* **OpenSSL:** The industry-standard command-line toolkit for SSL/TLS and general cryptography. It is used to generate keys, create certificates, and encrypt data.
* **GnuPG (GPG):** An open-source implementation of the PGP (Pretty Good Privacy) standard. Primarily used for securing email communication and signing files.
* **VeraCrypt:** A powerful open-source tool for creating encrypted containers or performing full-disk encryption (FDE). It is the successor to TrueCrypt.
* **BitLocker:** The native full-disk encryption feature built into professional versions of Microsoft Windows.

### Hashing and Integrity Verification

* **HashCalc:** A GUI-based tool for Windows that can calculate multiple hash types (MD5, SHA-1, SHA-256, CRC32, etc.) for files or text strings simultaneously.
* **MD5sum / SHA256sum:** Standard Linux command-line utilities used to generate and verify file hashes to ensure integrity during transfers.
* **HashMyFiles:** A small utility for Windows that allows you to view the MD5/SHA hashes of every file in a folder.

### Analysis and Learning

* **CrypTool:** An extensive educational platform for exploring, implementing, and attempting to crack a wide variety of historical and modern cryptographic algorithms.
* **Cain & Abel:** A legacy multi-purpose tool used in labs for sniffing encrypted protocols and cracking various hash formats.

## 2. Hands-On Lab Sessions

### Lab 1: Verifying File Integrity (Hashing)

**Goal:** Confirm that a downloaded file has not been tampered with or corrupted.

1. **Preparation:** Download a file and note its expected SHA-256 hash from the vendor's website.
2. **Linux CLI:** Run sha256sum [filename] in the terminal.
3. **Windows GUI:** Open **HashCalc**, drag the file into the tool, and select the SHA-256 checkbox.
4. **Verification:** Compare the output string to the vendor's string. If they match exactly, the **Integrity** is confirmed.

### Lab 2: Asymmetric Key Generation (OpenSSL)

**Goal:** Create a Public/Private key pair for RSA encryption.

1. Generate Private Key: ```bash  
   openssl genrsa -out private