The Journal of Open Source Software

SDSCMT: A Python Package for Secure and Reproducible Certificate Management in Scientific Computing

Subham Divakar [1] and **Rojalina Priyadarshini** [1]

1 CVRGU

DOI: 10.xxxxxx/draft

Software

- Review
- Repository
- Archive

Editor:

Submitted: 24 October 2025
Published: unpublished

Summary

SDSCMT (Secure Data SSL Certificate Management Tool) is an open-source Python package that simplifies the generation, validation, encryption, and conversion of self-signed SSL certificates and private keys. Designed for researchers, developers, and engineers, it enables secure communication between distributed systems, IoT devices, and computational pipelines commonly used in scientific research.

Unlike conventional tools such as OpenSSL, SDSCMT provides a **lightweight, Python-native command-line and programmatic interface** that integrates easily into reproducible research environments. It facilitates end-to-end security in experimental and data-driven workflows where encryption, authentication, and data integrity are essential but often neglected.

By automating SSL/TLS setup and providing reusable, verifiable certificate management routines, SDSCMT empowers the scientific community to maintain secure, reproducible, and ethical research pipelines across domains such as machine learning, healthcare data management, and environmental sensing.

SDSCMT provides both a command-line interface (CLI) and a Python API for seamless integration into research workflows, enabling researchers and developers to automate cryptographic certificate handling directly within their Python environments. The tool eliminates the need for complex OpenSSL commands, making it easier to ensure data confidentiality, integrity, and authentication during experimental communication between local servers, data acquisition modules, and computational pipelines.

By bridging the gap between secure infrastructure and research usability, SDSCMT enhances the reproducibility, security, and reliability of scientific software ecosystems. Its accessibility and automation make it particularly valuable for researchers building secure, reproducible pipelines in areas such as artificial intelligence, distributed computing, and networked scientific instrumentation.

Statement of need

Managing SSL/TLS certificates is an essential but often overlooked component of reproducible computational research. In modern scientific workflows—spanning areas such as distributed machine learning, IoT-based environmental monitoring, biomedical data collection, and cloud-hosted simulations—data are frequently transmitted between services, instruments, or collaborators. Ensuring that these transfers occur securely is a prerequisite for maintaining data integrity and confidentiality, both of which underpin reproducibility and trust in research outcomes.

While existing tools such as **OpenSSL** or **certbot** provide mechanisms for certificate creation and management, they are primarily designed for production infrastructure, not for lightweight research or experimental deployments. Their command syntax, steep learning curve, and lack of integration with Python-based research ecosystems make them inconvenient for everyday scientific use.

SDSCMT (Secure Data SSL Certificate Management Tool) addresses this gap by offering a simple, Python-native framework for generating, validating, encrypting, and converting self-signed certificates—which are widely used in local research environments, simulation testbeds, and prototype networks. The tool enables researchers to establish secure connections between distributed computational components without requiring external certificate authorities or complex configurations.

By integrating **cryptographic security** directly into Python workflows, SDSCMT ensures that reproducibility pipelines, experiment dashboards, and sensor networks can communicate securely and consistently. This makes it especially valuable for researchers developing data-driven models or deploying sensitive AI systems that demand encrypted transport but lack managed security infrastructure.

Existing Python libraries provide APIs for cryptography and certificate handling (`cryptography`, `pyOpenSSL`), but they lack a unified CLI tool for generating, validating, converting, and monitoring certificates. SDSCMT addresses this gap by offering a **single, comprehensive tool** with a simple interface that automates common certificate management tasks. While tools like **OpenSSL** provide powerful capabilities, they require deep cryptographic understanding and lengthy command-line instructions. Many developers resort to manual configurations or unsafe reuse of certificates, increasing the risk of misconfiguration or expired keys.

SDSCMT addresses these challenges by providing: - A **unified, automated tool** for generating and managing self-signed certificates; - Built-in validation to detect invalid or mismatched certificates and keys; - Seamless **conversion between PEM, DER, and PKCS#12** formats; - Optional **encryption for private keys**; and - A **cross-platform, user-friendly CLI and Python API** requiring no prior cryptographic expertise.

SDSCMT is particularly useful for researchers, developers, and DevOps teams who need to test HTTPS-based systems, secure data transfer pipelines, or internal APIs without relying on third-party certificate authorities.

Features

- **Generate self-signed certificates** with configurable hostname, organization, country, and validity period.

- **Validate existing private keys and certificates** to ensure integrity and correctness.

- **Convert between major SSL formats:** PEM DER PKCS#12.

- **Encrypt private keys** using password-based protection for added security.

- **Command-line interface** with intuitive menu-based navigation.

- **Cross-platform support** for Linux, macOS, and Windows.

- **Lightweight and dependency-minimal**: only requires `cryptography` and `pyOpenSSL`.

Software description

Installation

Install SDSCMT from PyPI:

```
pip install sdscmt
```

Example Usage

After installing SDSCMT, you can use the tool via the command line or Python module.

Command Line Interface

```
$ sdscmt
=== SDSCMT (Secure Data Self Signed SSL Cert Management Tool) ===
One stop cert generation and management tool.
Self-signed cert generation and cert management made easy......

Created by Subham Divakar

1. Generate self-signed certificate
2. Validate and display existing private key
3. Validate and display existing certificate (crt) file
4. Encrypt private key (recommended)
5. Cert Conversion Tool to convert cert from one type (DER, PKCS12, PEM) to another (DER

Enter your choice (1/2/3/4): 1
=== Self-Signed Certificate Generator ===
Enter the hostname: dummy
Enter the organization (optional): Test
Enter the country (optional): IN
Enter the number of days the certificate is valid for: 23
Enter the filename to save the private key(ending with .key): test.key
Enter the filename to save the certificate(endind with .crt): test.crt
Enter password to encrypt the private key (leave blank for no encryption):
Private key saved to test.key
Certificate saved to test.crt
```

Implementation

SDSCMT is implemented in **Python 3** and leverages cryptography and pyOpenSSL for cryptographic operations. The CLI is built using standard Python argparse for cross-platform compatibility. The tool handles certificate creation, validation, conversion, and encryption securely and efficiently. It also efficiently handles key and certificate creation, validation, conversion, and encryption, ensuring robust and secure workflows with minimal dependencies.

Impact

SDSCMT contributes to the broader ecosystem of open and reproducible scientific software by making secure data transmission simple, transparent, and automatable. It enables reproducible experimentation in disciplines that handle distributed or privacy-sensitive data—such as agricultural AI systems, medical image analysis, and remote sensing—where encryption is critical for ensuring ethical and compliant data exchange.

By reducing the technical barrier to implementing SSL/TLS in experimental systems, SDSCMT empowers researchers to:

Create consistent secure environments for computational experiments;

Integrate encryption into reproducible pipelines (e.g., ML model serving, edge-to-cloud communication);

Validate and version-control cryptographic artifacts as part of the research workflow; and

Share open-source reproducible environments that retain built-in data security guarantees.

The tool's open-source availability on PyPI and GitHub encourages community contributions, reproducible use cases, and integration with scientific data platforms. Its cross-platform nature allows scientists to deploy secure configurations in diverse setups—from lab workstations to edge IoT nodes—thereby strengthening the integrity and reproducibility of scientific research in the digital era.

Quality control

- **Unit tests** cover generation, conversion, and validation functions.

- **Cross-platform testing** ensures correct behavior on Linux, macOS, and Windows.

- **Input validation** prevents invalid certificate parameters.

- **Documentation and usage examples** guide users through all functionalities.

Acknowledgements

The authors thank the **Python Software Foundation**, the maintainers of the cryptography and pyOpenSSL libraries, and the open-source community for enabling secure and accessible software development.