Table of Contents

[Research for Encryption Algorithms & Available Libraries 2](#_Toc144896331)

[Encryption Algorithms 2](#_Toc144896332)

[Symmetric Algorithms 2](#_Toc144896333)

[Asymmetric Algorithms 3](#_Toc144896334)

[Available libraries in Delphi with Encryption implementations 4](#_Toc144896335)

# Research for Encryption Algorithms & Available Libraries

This document provides end results of research done for selection of suitable cryptographic algorithms. Later section would include a list of available libraries that can be used to implement required solution with the help of work already done in these libraries.

## Encryption Algorithms

Two types of algorithms are used frequently to secure data, namely Symmetric and Asymmetric. Symmetric algorithms are based on single key called secret which is employed to encrypt data to cipher text and vice versa. So, it is necessary to have secure access to secret key and texts generated by it. On the other hand, Asymmetric algorithms are based on a set of two keys. One key remains in public domain called Public key while other key need to be in secret called Private key. Plain text is encrypted through public key to generate cipher text which can later be converted to plain text using private. Usage of two key to get same results makes asymmetric algorithms more complex than its counterpart.

Following are cryptographic algorithms that support at least 256-bit key size:

### Symmetric Algorithms

1. AES

- 1998

- Encryption rounds based on Key size (14 rounds for 256-bit key)

- Industry Standard (Manufacturers)

- Open Source

- Special instruction sets in Intel & AMD processors

- Rapid data encryption/decryption

- Variable key length

2. TwoFish (Blowfish)

- 1998

- Not Licensed, Free

- 16 Rounds of encryption (Independent of key size)

### Asymmetric Algorithms

1. RSA

- 1977

- 1024-bits, up to 8192-bits key size

- Slower in encryption/decryption process

- Level of Security is incomparable

2. ECC

- 1985

- 256-bits, up to 512-bits

- Fast Encryption speed

- Suitable for mobile, with low processing speed

Comparison of both types of algorithms in terms of performance and usage.



## Available libraries in Delphi with Encryption implementations

Following are the libraries available in Delphi that can be used in implementation of our solution.

1. IPWorks Encrypt Library

- Contains both Symmetric/Asymmetric Encryption Algorithms

- Free Commercial License

- GetIt Package Manager

- <https://blogs.embarcadero.com/powerful-enterprise-grade->encryption-library-for-delphi-and-cbuilder-developers/

2. Delphi Encryption Compendium

- Mostly Symmetric Encryption Algorithms

- Apache 2.0 License (Open-License - Permissive)

- GetIt Package Manager

- <https://github.com/MHumm/DelphiEncryptionCompendium>

3. Lockbox

- Mostly Symmetric Encryption Algorithms

- Mozilla Public License (Open-License - Permissive)

- GetIt Package Manager

- [https://github.com/TurboPack/LockBox3](https://github.com/MHumm/DelphiEncryptionCompendium)

3. CyrptoLib4Pascal

- Mostly Symmetric Encryption Algorithms

- MIT License (Open-License - Permissive)

- https://github.com/Xor-el/CryptoLib4Pascal

More Encryption Libraries

https://alenibric.medium.com/encryption-libraries-for-delphi-secure-your-data-with-ease-f98bf6f3debd