

Microsoft Office Outlook PGP Add-in

Group 37

Instituto Superior Tecnico

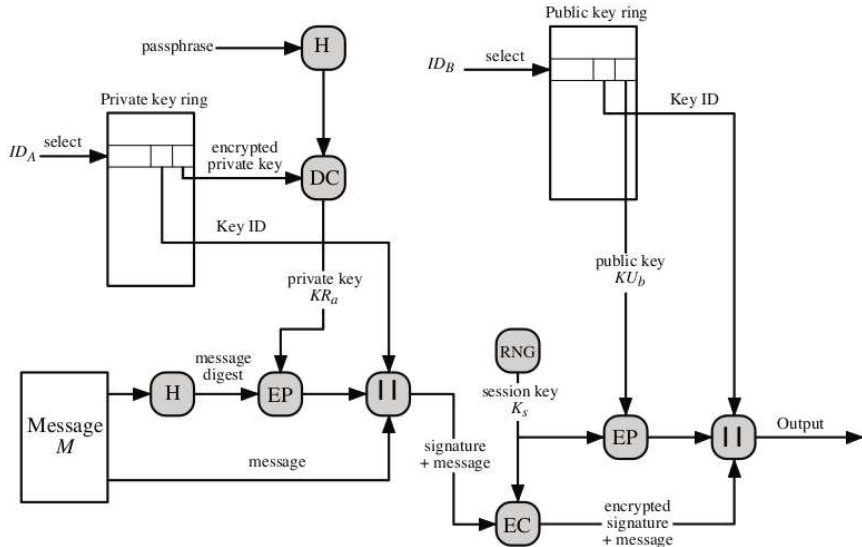
December 19, 2013

Overview

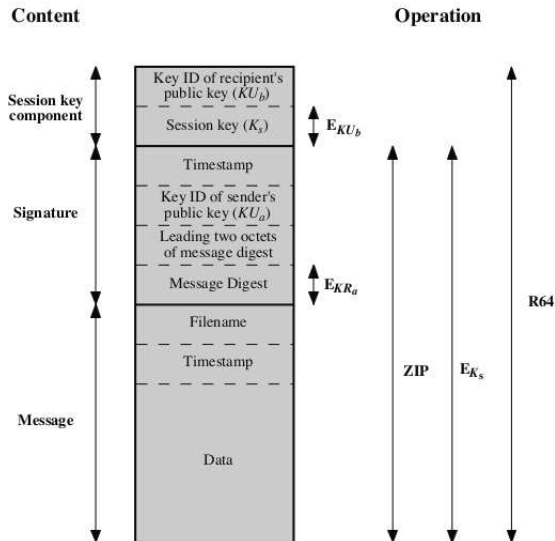
- 1 Introduction
- 2 Implementation
- 3 Conclusion

- PGP, GnuPG, OpenPGP
- Services offered:
 - Digital Signature
 - Message Encryption
 - Compression
 - Compatibility
- Interoperability
- Legal issues

How it works



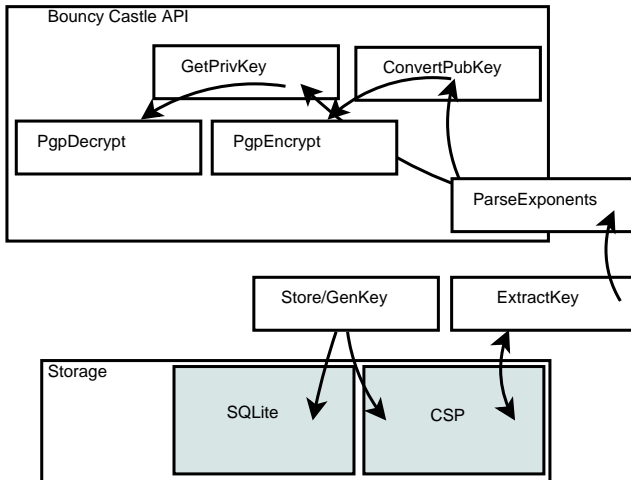
How it works



Implementation - 1st Iteration: Bouncy Castle API

- Microsoft CryptoServiceProvider
- SQLite for key/contacts information
- Bouncy Castle Open Source Cryptographic functions

Design



Implementation - 2nd Iteration: Didisoft API

- Didisoft's .NET API
- Choices made:
 - Key Size: 2048 bits [1]
 - Asymmetric Algorithm: RSA
 - Symmetric Cipher: AES-128/CAST5
 - Hash function: SHA1, MD5, SHA256
 - Compression: ZIP

Implementation - RSA vs. DSA2/ElGamal

RSA

- Integer Factorisation
- Default in GPG [4]
- More wide-spread
- Faster signature verification
- Longer signatures

DSA2 with ElGamal

- Discrete Logarithm Problem
- Smaller signatures
- Slightly faster signature generation
- Shorter key length

Lessons learned

- Poorly documented APIs are *not good*
- Didisoft API limited to inline encryption [2]
- Implementing OpenPGP is *hard*

Future work?

- PGP/MIME support (attachments)
- ECDSA and ECDH?
- Advanced configuration
- Keccak (SHA3) vs. MD5 or SHA1 (vulnerable [3])

References



Elaine Barker, Allen Roginsky (2011)

Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths

NIST Special Publication 800-131A



www.didisoft.com

OpenPGP Email messages

<http://www.didisoft.com/net-openpgp/examples/openpgp-email-messages/>



Marc Stevens

Framework for MD5 & SHA-1 Differential Path Construction and Chosen-Prefix Collisions for MD5

<https://code.google.com/p/hashclash/>



Nathan Willis(June 17, 2009)

Dealing with weakness in SHA-1

<http://lwn.net/Articles/337745/>

Demo