Improved digital signature handling in LibreOffice

By Miklos Vajna

Senior Software Engineer at Collabora Productivity

2016-09-08
About Miklos

- From Hungary
  - More blurb: http://vmiklos.hu/
- Google Summer of Code 2010/2011
  - Rewrite of the Writer RTF import/export
- Writer developer since Feb 2012
- Contractor at Collabora since Sept 2013
Digital signature handling
The feature: digital signing

- a mathematical way
- demonstrates the authenticity of a document
Results #1
Signature descriptions

- Use-case: want to sign with the same certificate multiple times
- Only makes sense if role / comment / description is provided
Results #2
SHA-256 support

• Only SHA-1 was supported previously
• Can now read SHA-256/ODF
• Can now read and write SHA-256
• Motivation: SHA-1 based operations must be rejected since 2012-01-01 in a legal case in the EU
Results #3

OOXML signature import

- Used inside DOCX/XLSX/PPTX files
- Need to count the same hashes as MSO
- Verify that the expected and the actual ones match
  - Report good/bad signatures exactly when MSO does so
- Badly documented in ECMA-376
- Better in ISO/IEC 29500
Results #4
OOXML signature export

• Write an OOXML signature that’s accepted by MSO
• Preserve existing ones
• Remove one or all of them
• Privacy problems around HW details
• OOXML signature is inherently less secure (metadata)
Results #5
Classification toolbar

- Not strictly related to signing, but the two features can be used together
- Use-case: In case the user is required to follow a policy when editing a document
- Help the user respect these rules
Results #6
Multi-category classification

- 3 different policy types (IntellectualProperty, NationalSecurity and ExportControl)
- Different classification categories for different policy types
How is this implemented?
Signing XML content

• Base: [xmldsig-core] from W3C
• Then:
  • OpenDocument v1.2 part3, section 5: Digital Signatures File
• W3C does not define how to store multiple signatures → different markup
libxmlsec

- LibreOffice uses libxmlsec for signature creation / verification
- The bundled libxmlsec is configured to use:
  - Mozilla/NSS on Linux/macOS
  - Native OS APIs on Windows
Updating and extending libxmlsec

- Implement OOXML Relationships Transform Algorithm
- `win32 configure`: adapt to renamed `autoconf configure`
- Fix Visual Studio 2015 build
- `win32`: fix undeclared `XMLSEC_DEFAULT_CRYPTO`
- Now we bundle the latest libxmlsec
- All patches I added are upstreamed
Signing non-ODF documents

- Code in xmlsecurity/ assumed that only ODF can be signed
- New filter flag: SfxFilterFlags::SUPPORTSSSIGNING
- We still expect zipped XML everywhere
Description

- ODF: just another optional property, similar to the signing timestamp
- If empty, we don’t write it, this way existing signature hashes are not broken
- OOXML mandates it
OOXML import/export

• Signature list markup uses the normal OOXML relation format
  • Existing parser/serializer in comphelper/
  • Can reuse that here without problems

• Individual signatures:
  • Import: OOXMLSecParser in xmlsecurity/, a SAX handler
  • Export: OOXMLSecExporter in xmlsecurity, works on a css::sax::XDocumentHandler
Classification toolbar

• “Just” a GUI: works with the user-defined properties available at File → Properties

• Transglobal Secure Collaboration Program (TSCP):
  • Business Authentication Framework (BAF)
  • Business Authorization Identification and Labeling Scheme (BAILS)

• Legal text → BAF policy → LO embeds BAILS key-value pairs into documents
Thanks

• Collabora is an open source consulting company
  • What we do and share with the community has to be paid by someone

• Dutch Ministry of Defense sponsored this work
Summary

• Improved digital signature handling provides better ODF and initial OOXML support
  • Available in LibreOffice 5.2
  • Both reading and writing OOXML signatures
  • First non-ODF file format that supports signing
• Thanks for listening! :-) 
• Slides: http://vmiklos.hu/odp