

CodePlex Archive will be shut down after July 1st, 2021.

CodePlex was Microsoft's free, open source project hosting site, which ran from 2006 through 2017. The site has been in archive mode for 3 years. We now encourage customers to use [Github](#) for their open source project hosting needs.

CodePlex will continue as an archive until next July (2021), at which point it will be shut down. Until then, you can browse published projects, documentation, issues, and discussions which were posted before the site went into archive mode three years ago.

For questions or comments please contact [CodePlex Archive support](#).

CodePlex Archive Open Source Project Archive

splistitemcrypt

SharePoint 2013 List Item Encryption

[download archive](#)

This coding exercise project enables you to encrypt/decrypt list item text field in the browser using industry standard algorithms.

[home](#) [issues](#) [discussions](#)

SharePoint 2013 List Item Encryption

Some days ago I found this JavaScript library for client side encryption using standard crypto algorithms. Everything done in the browser. Cool!

Stanford Javascript Crypto Library: <http://crypto.stanford.edu/sjcl/> – They say: "It uses the industry-standard AES algorithm at 128, 192 or 256 bits; the SHA256 hash function; the HMAC authentication code; the PBKDF2 password strengthener; and the CCM and OCB authenticated-encryption modes."

Based on this I wanted to create a client side SharePoint List Item crypt module to encrypt (text) data in the users browser. So the data is stored in SharePoint encrypted. No one can read it without having the password.

Please see my initial blog article: <http://blog.karstein-consulting.com/2013/07/01/client-side-encryption-of-list-item-fields-for-sharepoint-2013-demo-project/>

It's a coding exercise, nothing more!!!! Read the limitations below and be sure: there are more limitations I do not know at the moment...

My solution is very simple: I created a Visual Web Part with Visual Studio 2012. This web part contains everything I need for encryption / decryption.

1. It contains the Standford Javascript Crypto Library.
2. It contains a copy of jQuery 1.10.1.
3. It contains a Base64 serialized image that is used to mark input fields as "encryption protected".
4. Some custom javascript.

That's it. Small footprint. – The web part needs to be placed on each list form (new / edit / display) and on each list view page. Everything else is done by the Web Part.

Benefits:

- Client side data encryption.
- Industry standard encryption. Theoretically possible to decrypt the data later outside of SharePoint using the correct password and some tools / libraries.
- You can share the password with anyone who needs to decrypt the data. It's not bound to your user account.

Limitations – *be careful to read and understand them before using it in any way. – There are more limitations. The list is not complete!!!*

- First of all: It's a single-person'n'quick-done demo project. Nothing for production use. – You could use your data! – I'm responsible for any problems.
- Works only for text fields and multi line text fields without HTML formatting.
- (Single line) text fields in SharePoint are limited to 255 characters length. The encrypted data is stored as Base64 in the field. So it's not possible to encrypt 255 characters to the same amount of data: 255 bytes of plain text chars are much more than 255 bytes in encrypted state. SharePoint and my module does not handle this situation. (Because it's a demo project not a product ;-)
- If you lose the password there is no way (other than "brute force") to get your data back. There is no back door.
- The data cannot be searched. - You should exclude the list from being crawled.
- No way to change the password. – If this will be possible in the future then there will be no way to migrate already encrypted data. This is because it's client side encryption. The server does never know the password. So it cannot migrate the data from the current version of the project to a new version.
- No inline edit on list view pages!
- No "decrypt" option to permanently remove encryption.