K. M. Peterson • BBLISA • 10Apr13
bit.ly/kmpbblight

Problem:

Escrow

No ultimate trust of any one person (or facility, agent, system)

Solution:

SSSS generates *n* keys, with a property of the set *t* for threshold, from a secret < 1024b.

To recover the original secret, one must have *t* keys. There is no practical difference between having *0* keys and *n-1* keys.

So: decide on necessary threshold (or quorum) of those persons to hold escrow keys, generate and distribute.

Implementation

Available from B. Poettering, `http://point-at-infinity.org/ssss` - deb package available, RPM from dag repository. Mac OS X required GNU MP library.

Example:

```
$: echo "My Secret ABCD" | ssss-split -t 3 -n 5
WARNING: couldn't get memory lock (ENOMEM, try to adjust RLIMIT_MEMLOCK!).
Generating shares using a (3,5) scheme with dynamic security level.
Enter the secret, at most 128 ASCII characters: Using a 112 bit security level.
1-68b43b2fea4427d4f601b75e5958
2-14a95aa2da5f8fc5970ec083bf01
3-a3f22911ed9ca72a1f5b3d6dc250
4-dea0b981734d4a414b24996a9444
5-69fbca32448e62aec3716484e907
$: ssss-combine -t 3
WARNING: couldn't get memory lock (ENOMEM, try to adjust RLIMIT_MEMLOCK!).
Enter 3 shares separated by newlines:
Share [1/3]: 4-dea0b981734d4a414b24996a9444
Share [2/3]: 2-14a95aa2da5f8fc5970ec083bf01
Share [3/3]: 3-a3f22911ed9ca72a1f5b3d6dc250
Resulting secret: My Secret ABCD
```