Crypto Shredding

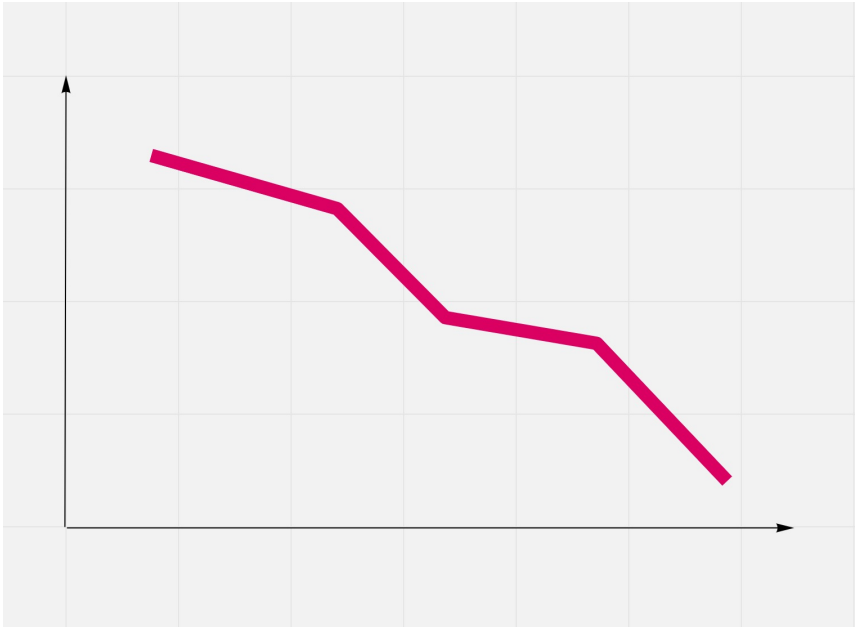
Or how to delete the undeletable

PRISMA.

Bitcoin https://ccnull.de/foto/bitcoin/1005735

Disclaimer

This talk is not about how to shredd your money with crypto currencies.



PRISMA.

Example Super user attack

- *System admin with root access to the system decides to attack*
- *Example:*
  - *https://youtu.be/JHGkaShoyNs?t=1636*

- *Are your systems protected against attacks like this?*

PRISMA.

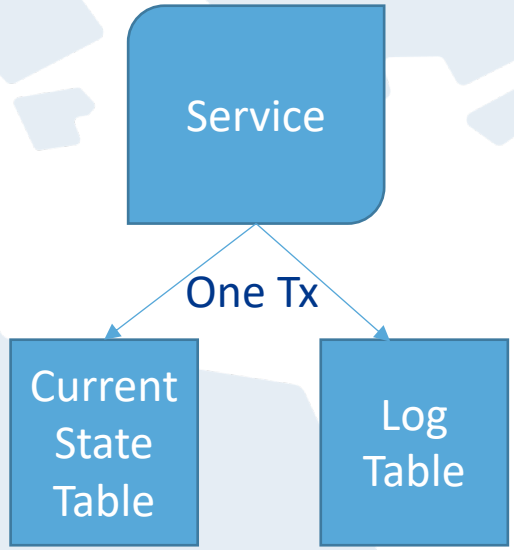
Audit log

- *Every change of the data is written to a separate log*
- *Together with the data of the changing user*
- *Including all personal data of the users of the system*
- *Very strict access rules*

- *Consistency checks*
- *Advanced: Log is the single source of truth*

Possible Implementation

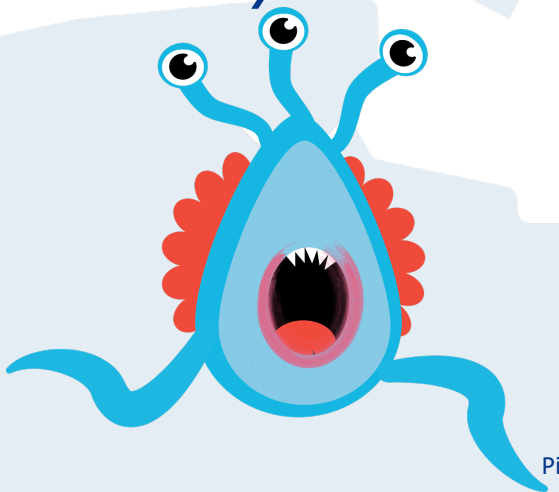
- *RDBMs based systems have often an audit table next to the data table*
- *The system writes the change in the data table and to the audit table*
  - *Manual changes in the DB?*
  - *Secure against developer attacks?*
- *Advanced:*
  - *DB triggers*



PRISMA.

Problems

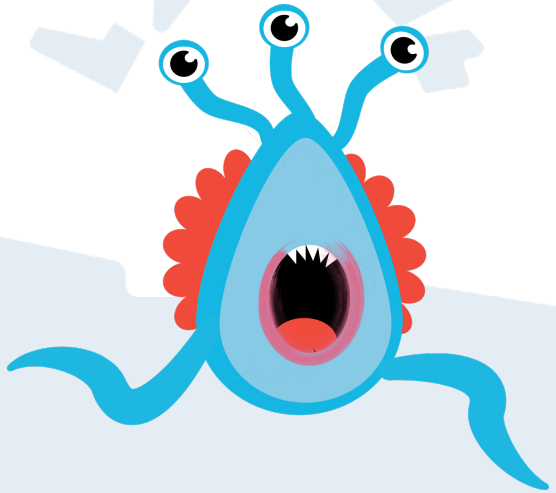
- *Tables grow in length and also in column number*
  - *deleted columns, renamed columns, all new columns*
- *What is the original data table good for?*
- *Still some super users can manipulate the logs*
- *Leads to WORM (Write Once Read Many) Storages*
  - *all personal data in the system is written onto hardware which does not allow deletion*



Picture from acaladolopes on Pixabay

Side step: GDPR

- *Every user of the system has the right to:*
  - *Get informed about the own personal data in the system* 🥳
  - *Get the data deleted if there is no legit reason to keep it* 😱
- *The System has to follow the law of data thrift*



Picture from acaladolopes on Pixabay

PRISMA.

Sidestep: AES

- *AES (Advanced Encryption Standard)*
- *Symmetric: same key for encrypt and decrypt*
- *Reasonable fast (HTTPS transport is based on that)*
  - *On our production frontend machines, SSL/TLS accounts for less than 1% of the CPU load, less than 10 KB of memory per connection and less than 2% of network overhead. Many people believe that SSL/TLS takes a lot of CPU time and we hope the preceding numbers will help to dispel that.*
    -Adam Langley, Google "Overclocking SSL", 2010

Idea crypto shredding

- *Personal data is encrypted in the audit log*
- *Every data which should be deleted is encrypted with a subject's key*
  - *E.g. All data of peter is encrypted with peter's key*
- *Without the key the data is very hard to decrypt, if using 256 bit key length*
  - *Complexity ~10^76 <-> 10^18 FLOPs Fugaku*
  - *Data can not be recovered with the current techniques within the live time of the universe*
- *Throwing away keys can be considered as deletion*

Implementation

- *Decryption/Encryption*
  - *Java standard libraries for years*
- *Key infrastructure*
  - *Highly available*
  - *Highly secure*
  - *Problematic deletion of keys*
  - *SaaS key infrastructures are expensive (1 $ per key)*

Implementation: Key infrastructure

- *Idea: Key infrastructure is just a big key/value map*
- *Use the key value store of your choice, which provides backup and redundancy out of the box*
- *Self manged solutions like Cassandra*
- *Or just use AWS Dynamo DB*

Implementation: Java

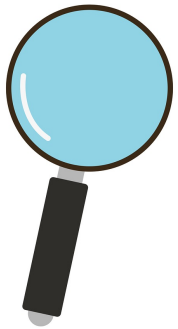
- *The idea could be implemented fast by yourself*
- *Optionally PRISMA crypto shredding library can be used*
- *https://github.com/prisma-capacity/cryptoshred*
- *Pluggable Metrics interface*
- *Pluggable CryptoEngine*
- *Pluggable Key Repository*
- *optional Spring-Boot autoconfiguration module*
- *Jackson based deserialization to Java Objects*

Implementation: Java

- *Show Example*

Disadvantages

- *Key is gone -> data loss*
- *You have to find a balance between key backup and data deletion needs*
  - *Immediate deletion is not recommended due to data loss prevention*
- *Log can not be read without decryption*
  - *Trust to the decryption*
- *Encrypted data is **not searchable***
- *Dependencies of data cannot be changed*



Picture from Marco Livolsi on Pixabay

PRISMA.

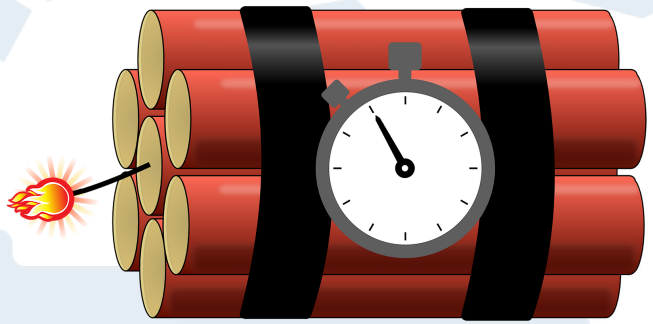
PRISMA

- *Use together with event sourcing*
- *Every personal data is encrypted*
- *Every user, contact, etc... is encrypted with an extra key*
- *Searchable data is hold in memory or extra tables*
  - *This has to be deleted separately*

PRISMA.

Advanced: Self destroying data

- *For many use cases you are obliged to keep data for a certain time*
  - *E.g. invoices, tax relevant data,…*
- *Implement a keystore, which deletes keys automatically after a certain time*
- *Done*



Picture from Able Lingo on Pixabay

Advanced: composite keys

- *Sometimes you have data depending on each other*
  - *E.g. employee and company*
  - *If the company is deleted also all the data of users should be deleted*
- *Still in development, help very welcome!*
- *Idea 1: encrypt the already encrypted data with another key*
- *Idea 2: allow combined key ids in key infrastructure, deletion of a key then also means deletion of all keys with combined key id*
- *Idea 3: use composite keys meaning you combine the key byte arrays. This key of course is never stored*

Conclusion

- *We have seen, how to delete data conform to GDPR also in write once environments*
- *As always: “it depends” if you should use Crypto Shredding*
  - *Recommendation in strongly regulated markets*
  - *Event Sourcing*
  - *Or in gambling*