Practical Threshold Cryptography for Cloud and Cryptocurrencies

Jakob I. Pagter
From sugar beets to Threshold Cryptography

2009

Damgård et al., Financial Crypto ’09: “Multi-Party Computation Goes Live”

2008

Multiple commercial applications of MPC
- Sugar Beet auction
- Off-exchange matching (Tora)
- Privacy-friendly demographic profiling (Insights Network)
- ...

2015

- “Secure Dropbox”
- “KMaaS”
- EU funding 2016-17

2018

- Series A in 2018
- VCTRADE (SBI)
- Cloud, blockchain and cryptocurrencies
The landscape for Threshold Cryptography (TC)

**Incumbents**

- HSM – Hardware Security Modules
  - E.g. Thales, Utimaco
- TEE – Trusted Execution Environments
  - E.g. SGX
- CSP offered KMS
  - E.g. AWS KMS

**Building blocks**

- MPC protocols
  - AES
  - ECDSA
  - General
- MPC frameworks for general MPC
  - libscapi
  - EMP toolkit
  - ...
Sepior KMaaS (Key Management as-a-Service)

Core contributions

- Example architecture for TC-based KMaaS
- TC-friendly stream-cipher
- Experiences with commercialization

Design philosophy

- Easy integration [Developer]
- Simple administration [Security Officer]
- Transparent [End-user]

```java
import SepiorServicesClientConfiguration;
import SepiorUtils;
import Paths;
import SepiorServicesClient;
import AmazonS3;
import PutObjectRequest;
import GetObjectRequest;
import InputStream;
import Files;

private static void uploadUsingAmazonSDK(String s3Bucket, String s3Key, String filename) throws SepiorServiceException, SepiorUserException {
    AmazonS3 awsClient = s3Enc.getAmazonS3();
    PutObjectRequest put = s3Enc.getPutObjectRequest(s3Bucket, s3Key, new File(filename));
    awsClient.putObject(put);
}

private static void downloadUsingAmazonSDK(String s3Bucket, String s3Key, String filename) throws IOException {
    AmazonS3 awsClient = s3Enc.getAmazonS3();
    GetObjectRequest get = new GetObjectRequest(s3Bucket, s3Key);
    S3Object s3Object = awsClient.getObject(get);
    try (InputStream in = s3Object.getObjectContent()) {
        Files.copy(in, Paths.get(filename));
    }
}
```

Entire AWS S3 integration
Challenges in cloud encryption

Security perspective (*-YOK)

- Randomness
- Storage of key
- Usage of key

Business perspective
TC-based KMaaS

**TC-friendly stream cipher**

\[ P_1, K_1, S_1 = PRNG(K_1, iv, |M_1|) \]

\[ P_2, K_2, S_2 = PRNG(K_2, iv, |M_2|) \]

\[ \vdots \]

\[ P_n, K_n, S_n = PRNG(K_n, iv, |M_n|) \]

\[ M = M_1 \oplus \ldots \oplus M_n \]

\[ C = C_1 \oplus \ldots \oplus C_n \]

\[ K = (K_1, \ldots, K_n) \]

**Architecture**

User

Developer

CSP1

CSP2

CSP3

Application

Sepior SDK

Cloud Service

Encrypted data

Consume

\[ P_1: K_1 \]

\[ P_2: K_2 \]

\[ P_3: K_3 \]
**Experiences**

<table>
<thead>
<tr>
<th>Positives</th>
<th>Negatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>👍 People &quot;get it&quot;</td>
<td>👎 Lack of standards</td>
</tr>
<tr>
<td>👍 Ease-of-use</td>
<td>👎 Lack of certification</td>
</tr>
<tr>
<td>👍 Price</td>
<td>👎 First-mover reluctance</td>
</tr>
</tbody>
</table>
Sepior ThresholdSig

Core contributions

• ECDSA protocols in TC-setting
• Architectures for deployment
• Additional relevant features

Design philosophy

• Flexible components
• Simple deployment
• “Forget you have a hammer”
Challenges in cryptocurrency (wallet) security

Lose key = lose money 😞

MultiSig not enough

- Privacy
- Transaction cost
- Flexibility

See e.g. Gennaro et al. (eprint.iacr.org/2016/013.pdf).

Or,

- Onchain vs. off-chain
TC-based ECDSA solution

Cipher(s)
- (t/n)-ECDSA ciphers
  - Honest and dishonest
  - Active security
  - Based on security of ECDSA
- 1500 signatures/second on with each party running one server
- Allow abort
  - Manual intervention if need be
- Use preprocessing
  - (time above includes time for preprocessing)
Experiences

Positives

👍 Good alignment with decentral trust model

👍 Much untapped potential

Negatives

👎 Many cryptocurrencies use ciphers which are not TC-friendly

👎 General lack of awareness of TC potential
Summary

- Two commercial use case
- Obstacles from being new technology
- Vast potential

© 2019 Sepior
Thank You!