Improvements to DKG for use in a real-world setting

An EPFL IN Semester Project
23.01.2018

Student: Cedric Cook
Lab: DEDIS - EPFL
Supervisor: Nicolas Gailly
Professor: Bryan Ford
Outline

Motivation

Problem Statement

Setting

Solution

Discussion
Motivation
Swiss Federal Council
And their law passing method

The Swiss Federal Council
7 councillors (& 1 chancellor)
The council wants to pass a law

- Some key is needed to pass the law
- Decentralized, no trusted 3rd party
- 7 councillors participate
- At least 5 need to agree to pass

Distributed Key Generation, of course!
But this is 2018...

- Distributed Key Generation was invented in the 90's
- The council wants to do this over the internet
- DEDIS to the rescue
- Connection problems, congestion, etc...
Problem statement

The limitations of the current implementation of DKG are such that it is not performant in the real-world setting, due to a strict timing assumption.

This project overcomes these limitations by reducing the timing assumption, and use round termination procedures to provide certain guarantees.
Setting: Kyber, VSS & DKG
Kyber

- Advanced crypto library for Go
- Provides cryptographic primitives
- For applications that need more than signing and encryption
- Used by Cothority

The kyber repository on GitHub.
**VSS (in Kyber)**

1. Dealer chooses a random polynomial \( f(z) \) of degree \( t \):

\[
f(z) = c_0 + c_1 z + \cdots + c_t z^t
\]

The dealer broadcasts \( C_k = g^{s_k} \mod p \) for \( k = 0, \ldots, t \).
The dealer also computes the shares \( s_j = f(j) \mod q \) for \( j = 1, \ldots, n \) and sends \( s_j \) them secretly to each verifier \( A_j \).

2. Each \( A_j \) verifies the shares he received from the dealer by checking:

\[
g^{r_{ij}} = \prod_{k=0}^{t} (C_k)^{i^k} \mod p \quad \text{(1)}
\]

The verifier broadcasts a *response*, containing either *StatusApproval* if the check succeeds or *StatusComplaint* to incriminate the dealer.
VSS (cont'd)

3. The dealer reveals the share $s_j$ matching (1) for each complaining verifier $A_j$, by means of a justification. If any of the revealed shares fails this equation, dealer is disqualified.

4. If any participant has at least $t$ correct shares from the verifiers, they can find the key $s_0$ by polynomial interpolation.
DKG (in Kyber)

Can be understood as: \( n \) parallel instances of VSS.

In each instance one participant is the VSS dealer, others are verifiers

2 Implementations in Kyber:

1. Pedersen (Joint Feldman VSS)
2. Rabin (Use of 2 polynomials)
The synchronicity issue

Diagram showing possible complications in one round of VSS
Solution
Solution

- Enable protocol to continue with absent participants
- Use round termination signal
- Adapt VSS, DKG, and their tests in function of this.
Solution implementation

- "SetTimeout" methods added to both VSS and DKG
- Trigger for termination procedure
  - Mark unresponsive participants
  - Check all responses
  - Decide on validity of key
- Guarantee the correctness with tests
  - New edge cases were found
Discussion
Discussion

- Changes allow use in more general setting
- Protocol continue to function correctly
- Performance is not greatly different

Future work:

- Long term keys are no good
- Share renewal
Conclusion

- Project target: Improve Kyber DKG for real-world use
- Problem focus found on timing assumption
- Changes to allow round termination implemented
- Protocol remains correct
- Target accomplished ✓

Questions?
References

