Travel Report

Eurocrypt 2019

Åvald Åslaugson Sommervoll

Eurocrypt and Affiliated Events Introduction

The 38th annual international conference on the theory and applications of cryptographic techniques, Eurocrypt 2019, took place in Darmstadt Germany from May 19th to May 23rd. The Eurocrypt conference has been held every year since 1987 in different European countries. The conference is hosted by the *International Association for Cryptologic Research (IACR)*, of which all participants become members by attending one their conferences[1]. In addition to being a venue for academic publishing, Eurocrypt is an important meeting place for university faculty, students, and industry researchers and experts in Europe. The proceedings consist of peer-reviewed articles, which are published in the proceedings.

Eurocrypt's Affiliated Event, *Code-Based Cryptography (CBC) Workshop* took place prior to the conference 18th of May and 19th of May. This year it was their 7th workshop and focused on error-correcting codes, of which so far seem to be quantum safe.

Eurocrypt's Affiliated Event, *Quantum Algorithms for Cryptanalysis (QuAC)* took place the morning before the reception for Eurocrypt 2019 and has the aim to give an overview of quantum algorithms beyond Grover.

Code-Based Cryptography (CBC) Workshop

The first day of CBC workshop focused largely on McEliece, a highlight was Dan Bernstein presenting his talk *McEliece for tiny network servers*, where he argued that even though McEliece keys are larger than the ones we use now are usable, as decryption and encryption will only take a couple of milliseconds. One issue with McEliece is that with so many large keys it can't possibly maintain multiple connections without being flooded, making it vulnerable to flood attacks. To prevent this Bernstein suggests using a tiny network server. A tiny network server is a server that handles incoming network packets and immediately forgets them without allocating any memory. Of course it is still possible to flood the network, but to flood this you would need a considerable amount of computational power.

Quantum Algorithms for Cryptanalysis (QuAC)

In his talk: **Non-Asymptotic Quantum Resource Estimation**, *Vlad Gheorghiu* started by stating the powerful quantum computers are a double-edged sword, which will bring many breakthroughs, but it will also come with many risks. For example, the current public key cryptography is broken, by Shor's algorithm, with no quick patch available. Luckily symmetric key cryptography is just weakened, not broken (so far). Michele Mosca has a formula for when one should worry and when new standards should be introduced. That is:

```
Security shelf life + Migration time < Collapse time
```

[1] This membership is canceled after two years if no more IACR conferences are attended within that time.

So if the collapse time is larger then it is okay, if not then security is broken. To accurately estimate the Collapse time we need a greater understanding of quantum computers and their "speed". In detailing the quantum operation groups Gheorghiu mentioned two groups:

- **Pauli group**: Unitary group generated by X, Y, Z gates
- **Clifford group** Unitary group that maps Pauli operators to Pauli operators

where the *Clifford gates* and Pauli measurements can be efficiently simulated on a classical computer. However this is not the case for the *T-gate*, which is the problem child for quantum computation, and as it requires a lot of error correction, since it is very error-prone. However, an option is to use a faulty T-gate and purify it via *magic state distillation*. This throws away some efficiency but saves in error correction.

Eurocrypt

The talk **The Double Ratchet: Security Notions, Proofs, and Modularization for the Signal Protocol** attempts to solve some of the messaging issues of Facebook, WhatsApp, and Skype, by using a continuous key agreement. The scheme is based on the Diffie–Hellman key exchange:

1. Bob first sends a public key g^{x_1}
2. Using this Alice can generate a key $g^{x_1 x_2}$:
   (a) Alice can now send multiple messages encrypted with key $g^{x_1 x_2}$
   (b) Along with every message she passes her public key g^{x_2} so Bob can decrypt
3. Now if Bob wants to send a message he generates another x_3:
   (a) He can now send many messages using the key $g^{x_3 x_2}$
   (b) Along with every message he passes his public key g^{x_3} so Bob Alice can decrypt
4. Now if Alice wants to send some more messages she generates x_4 and encrypts with $g^{x_4 x_3}$
5. and so on

This way they can send messages frequently, and there is no issue if it takes some time between messaging since a new key is generated and used for the newer messages. **Approx-SVP in ideal lattices with pre-processing** presented by *Alice Pellet-Mary* gave an introduction to the speedup of quantum computers on ideal lattices in cyclotomic fields. Before stating that lattices are discrete 'vector spaces' over integers, and its basis is an invertible matrix such that: $L = \{Bx | x \in \mathbb{Z}^n\}$. Given this, the Shortest Vector Problem (SVP) is to find the shortest non-zero vector. The approximate-SVP (approx-SVP) is to find an approximation of this vector which has a norm smaller or equal to the smallest vectors norm. Similarly for Closest Vector Problem (CVP) and approx-CVP is finding the vector in the lattice closest to a given vector. Her work uses pre-processing to get a speedup in approx-SVP. The speedup is done by utilizing the Log space (Log with big "L"), and taking Log(<g>), and utilizing this basis to find the shortest vector.