Summary of Round 2 Changes for Picnic

March 30, 2019

1 New Parameter Sets – Shorter Signatures

We added three new parameter sets to the specification, `Picnic2-L1-FS`, `Picnic2-L3-FS`, `Picnic2-L5-FS`. We sometimes refer to these new parameter sets as "Picnic2". The overall design remains the same between Picnic and Picnic2, but the MPC protocol used to implement the proof system is different in Picnic2. The new MPC protocol is based on the work of Katz, Kolesnikov and Wang [KKW18], who have joined the Picnic team.

The new parameter sets reduce signature size by a factor of 2.7 on average. For example, `Picnic-L1-FS` signatures were 32.8KB and `Picnic2-L1-FS` signatures are 12.3KB. The CPU cost of Picnic2 in our current implementations is significantly higher than Picnic at the same security level. Some of this difference is inherent in the parameters of MPC protocol, that requires a higher number of parallel repetitions. Another part of the difference is likely explained by the amount of optimization effort that has been made for Picnic. The core matrix multiply operation is different, and optimizations done for Picnic are only partly applicable to Picnic2. Work to improve the CPU performance of Picnic2 is ongoing.

We also provide a concrete security analysis of the new parameter sets in the random oracle model, rather than using generic results for Fiat-Shamir-type signature schemes.

2 Improved Multi-Target Security

We changed the spec to address a multi-target attack reported to us by Dinur and Nadler [DN19]. At a high level, their attack involves an attacker who guesses a secret value used by a signer, derives data from that secret as the signer would, and then compares that data to data from multiple signatures (possibly by multiple signers) to check for a match. If the secret is k bits long and T signatures have been issued, this reduces the expected time for an attack to be successful from 2^k to about $2^{k-7}/T$. The change to our spec to address this attack involves having the signer use a random salt value per signature. Whenever we hash a k-bit secret, we include the salt, to ensure that the inputs to the hash have a unique salt per signature. We also add additional counters to ensure that all inputs within a signature (using the same salt) also have a unique salt.

This change to the spec is reflected in our implementations. The previous implementations no longer interoperate with the current versions.

3 Other Updates

The changes in this section do not require changes to the Picnic design.

3.1 New LowMC Implementation – Faster Signatures

The optimized implementation of all parameter sets now use a new implementation technique to make LowMC more computationally efficient. In the case of Picnic, sign and verify times are reduced by a factor of 1.7 to 3.7. This is based on research described in a recent paper by Dinur, Kales, Promitzer, Ramacher and Rechberger [DKP$^+$19]. In addition to reducing CPU time, this work also reduces memory required to store the LowMC constants by a factor of 2.4 to 4.8.

3.2 Team

The team added three new members, Jonathan Katz, Vladimir Kolesnikov and Xiao Wang, the authors of [KKW18].

References

[DKP$^+$19] Itai Dinur, Daniel Kales, Angela Promitzer, Sebastian Ramacher, and Christian Rechberger. Linear Equivalence of Block Ciphers with Partial Non-Linear Layers: Application to LowMC. In *EUROCRYPT*, 2019. To appear.

[DN19] Itai Dinur and Niv Nadler. Multi-Target Attacks on the Picnic Signature Scheme and Related Protocols. In *EUROCRYPT*, 2019. To appear.

[KKW18] Jonathan Katz, Vladimir Kolesnikov, and Xiao Wang. Improved non-interactive zero knowledge with applications to post-quantum signatures. In David Lie, Mohammad Mannan, Michael Backes, and XiaoFeng Wang, editors, *ACM CCS 2018*, pages 525–537. ACM Press, October 2018.