Changes from Round 1 to Round 2

This document provides an overview of the changes of CROSS Version 1.0, submitted to the first round, to CROSS Version 2.0, submitted to the second round.

Notation and Content:

1. We have changed the specification document to provide an operative summary of the CROSS features, together with operative descriptions and benchmarks, with the intent of improving readability. To this end, we have moved detailed explanations on

   - combinatorial and algebraic attacks from Section 3.1,
   - EUF-CMA security proof and forgery attacks from Sections 3.2, 3.2.1

   to a separate security guide, available at https://cross-crypto.com/.

2. We reduced the amount of mathematical and cryptographic background given in Section 1.

3. We unified the notation across the entire document, both in sequence diagrams and the procedural descriptions of key generation, signing, and verification.

Parameters:

1. We propose parameters for an additional optimization corner that aims for even lower latency than the previous *fast* optimization corner (at the cost of larger signatures). While previous parameter sets featured a *small* and *fast* optimization targets, the new security categories provide a *small*, *balanced* (formerly *fast*), and *fast* version.

2. Due to the novel forgery attack, we present novel parameters in Section 4. This results in increase in signature size for the *small* parameter sets of 12% to 24%. The signature sizes for the *balanced* parameter increase by up to 7% depending on the parameter set while decreasing up to 7% for some other parameter sets due to a more precise bound for the attack cost. This bound also results in decrease in signature size for the *fast* parameter sets of up to 3.3%.

3. We are using a new bound for the size of the seed path and Merkle proof, see Section 2.2.2. The new bound now involves also the Hamming weight of the binary representation of the number of rounds t and is proven to be tight.

4. In determining our parameters, we now take into account, as a lower bound to a single forgery attempt, the cost, in Boolean operations, of a single SHAKE computation. This allows us to match the security requirements of categories 1, 3, and 5 considering cheating probabilities higher than 2^{-128}, 2^{-192}, and 2^{-256}, respectively, by a factor equal to the ratio between the Boolean operation cost of an AES computation, and the one of a SHAKE computation.

5. The size of the keys and signatures has been updated in Table 4 (λ bits were missing).

Security:

1. A new NP-hardness proof of R-SDP is included in the security guide [39].

2. Improved security analysis for R-SDP(G): In Section 3, we consider an improved solver for the R-SDP(G), and we updated the parameters for CROSS R-SDP(G) accordingly. The parameters for the CROSS R-SDP instances are unchanged.

3. We present a security proof for the ZK protocol and the proof of EUF-CMA security of the CROSS signature scheme in [39].

4. We include a novel forgery attack in Section 3.2.1 derived from [9]. This attack does not depend on R-SDP, nor on R-SDP(G), but only on the non-interactivity of the transformed ZK protocol. This version of the attack improves upon the forgery presented in the previous versions of this specification by exploiting, in a better way, the use of fixed-weight challenges.

5. An overview of combinatorial solvers and the algebraic solver of Beullens, Briaud, and Øygarden [15] is given in Section 3; for more details, see [39].

Implementation:

1. To prevent collision attacks on CSPRNG seeds, we include salting and a unique index per CSPRNG instance in each round of the signature. We detail these tweaks in the procedural description of CROSS.

2. To add hedging against multikey attacks we raise the length of the seeds for keypair generation to 2λ: this allows to prevent collision attacks relying on the collection of $2^{\frac{\lambda}{2}}$ keypairs. We updated Algorithm 1, Algorithm 2, and Algorithm 3 accordingly.

3. We revised the CSPRNG implementation strategy, extracting always a constant amount of pseudorandom bits from each CSPRNG call. We make this possible, in the rejection sampling scenarios, considering the amount of required bits so that the CSPRNG extraction fails with probability $\frac{1}{2^\lambda}$. We detail this CSPRNG strategy in Section 5.1. This approach makes constant time implementations easier.

4. We now consider the objects in their bit-packed representation when they are employed as the inputs of cryptographic hashes, reducing the amount of required computation.

5. We switched from SHA-3 as a cryptographic hash to SHAKE with a 2λ bit extracted string. This improves on the overall speed, while keeping the same security margin (as the bottleneck for attacks was SHA-3 collision resistance, which matches the one of the appropriate SHAKE with a 2λ bit output). We provide details in Section 5.1.

6. We report the performance figures from an optimized implementation for the Intel AVX2 instruction set in Section 6.

7. The lengths of the required amount of randomness to be drawn from the CSPRNGs has been corrected in the codebase.

8. We moved onto a homogeneous strategy to perform domain separation across the SHAKE calls which CROSS employs as both CSPRNG and Hash. We fixed issues in the implementation with respect to domain separation and improved portability among platforms with different endianess. We furthermore revised the rejection sampling strategy and fixed a problem where the required randomness was underestimated.

9. We present a novel section, Section 5.6, discussing side-channel attacks and countermeasures.

10. We revised the truncation structure for the seed tree and now employ the same structure for the seed- and Merkle trees as discussed in Section 5.2.1.

11. We slightly changed the order of elements in `resp` and sampling of $\overline{\mathbf{M}}$ and $\mathbf{H}$ for R-SDP(G) to benefit implementation optimizations.

12. A SIMD implementation of Keccak is used to speed up in-round commitment hashing, Merkle tree hashing, and seed tree computations (see Section 5.3).

13. Reductions modulo $p = 509$ are now performed using Barrett's method (see Section 5.5).

14. CROSS is now part of the Open Quantum Safe family [36].

15. We report updated versions of Algorithm 1, Algorithm 2, and Algorithm 3, where we revised the generation of the $\mathbf{V}, \overline{\mathbf{W}}$, values. Doing so saves a CSPRNG call during key generation, $2t$ CSPRNG during signature, and $2w$ CSPRNG calls during verification, at no security margin loss.