Problem Set 6
Cryptographic Hash Functions
(Summer Term 2019)

Bauhaus-Universität Weimar, Chair of Media Security
Prof. Dr. Stefan Lucks, Jannis Bossert, Nathalie Dittrich, Eik List
URL: `http://www.uni-weimar.de/de/medien/professuren/mediensicherheit/teaching/`

Due Date: 4 July 2019, 11:00 AM, via email to `jannis.bossert(at)uni-weimar.de`.

Goal of this Problem Set: Deepen the understanding of tree hashing, one-time signatures, and differential cryptanalysis.

Recap about Lamport-Diffie and Winternitz' One-Time Signatures (LD-OTS, W-OTS) and Merkle hash trees from [1, Sections 1 to 5.2].

Task 1 – Winternitz Signatures (4 Credits)
Read [1, Section 2]. Let $M \in \{0,1\}^n$ be an n-bit message and $H : \{0,1\}^* \to \{0,1\}^n$ be a collision-resistant hash function. In Winternitz' approach W-OTS, one splits a message M into k-bit words $\widehat{M}_i \in \{0,1\}^k$: $M = (\widehat{M}_w \ldots \widehat{M}_1)$. Then, one needs w keys $X_i \in \{0,1\}^n$. To sign M, for every word $\widehat{M}_i$, the corresponding key X_i is hashed a few times, where the number of times is given by the value of $\widehat{M}_i$. For example, for $k = 2$ and $M = (01\,11)_2$, one would publish the hashes $(H(X_2), H(H(H(X_1))))$ as signature. Winternitz' approach uses an additional checksum

$$C := \sum_{i=1}^{w} 2^k - \widehat{M}_i,$$

and publishes and signs $M' = M \,\|\, C$ instead of only M. Alice proposes to compute the checksum as

$$C := \bigoplus_{i=1}^{w} \widehat{M}_i.$$

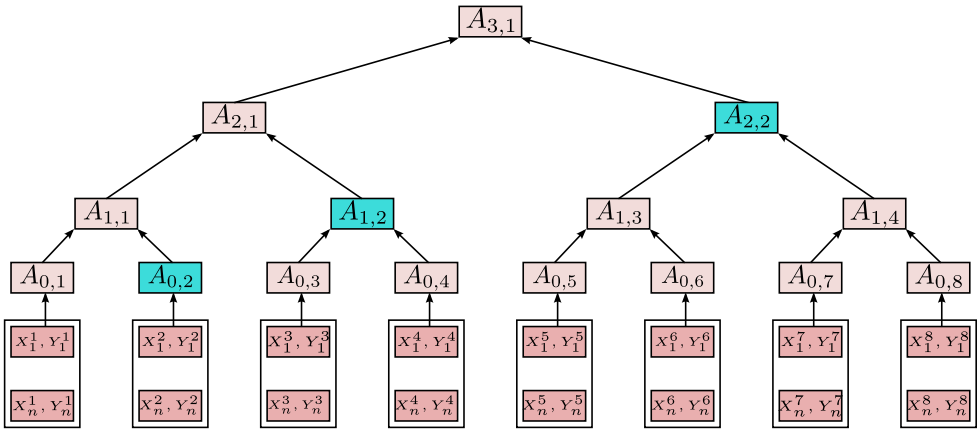
Assume that the used signature scheme is secure (existentially unforgeable) and H is a cryptographic hash function. Is this still a secure construction with this checksum? Explain reasonably *either* why yes *or* why not and provide a counterexample in the latter case.

Task 2 – Merkle Hash Trees (4 Credits)
Assume, $H : \{0,1\}^* \to \{0,1\}^n$ is a collision-resistant hash function. A Merkle hash tree is a binary tree of height h, where every node A in level i is a hash computed by H from two nodes of level $i-1$, e.g. $A_{1,1} = H(A_{0,1}, A_{0,2})$ in the figure below. The 2^h leaves represent one-time keys, where every keycan be used only once to authenticate a message. In the figure below, $h = 3$ is the tree height. The tree root $A_{3,1}$ is the public key, which is published before authenticating any message. To authenticate the first message M, Alice submits (M, $A_{3,1}$, $sig(M)$, $path = (A_{0,2}, A_{1,2}, A_{2,2})$). *sig* contains the verification keys (the values $X_1^1, \ldots, X_n^1$ in the leftmost leaf for the first message); *path* contains all those intermediate hashes that are

necessary for the receiver to compute from *sig* to the root hash for verifying the signature. To authenticate the next message, Alice publishes the next tree leaf as new *sig* and a new path; for the third message, she uses the third leaf and another path, and so on.

The time and the maximal memory necessary for computing the path are critical. One can use the PRNG trick from [1, Section 4] to avoid storing all leaves. So, the number of stored intermediate hash values yields the required memory.

Explain *either* Merkle's classical tree-traversal algorithm in your own words *or* think of an own non-trivial algorithm that efficiently computes the path for the next message with $s \ll 2^h$ memory available. Explain the average number of computations of H and the maximal needed memory for computing a *path*.



Task 3 – CubeHash (4 Credits)

In the lecture, CubeHash was introduced as one of the SHA-3 candidates. In the following, consider two variants CubeHash′ and CubeHash″, which have a slightly different round function (see Slide 235 in Section 9.3 for the original round function).

CubeHash′: The rotations (Steps 2 and 7) are omitted.

CubeHash″: The swap operations (Steps 3, 5, 8, and 10) are omitted.

Explain how these changes influence the security of the particular round functions regarding to differential cryptanalysis and the existence of symmetric states.

References

[1] Johannes Buchmann, Erik Dahmen, Michael Szydlo: Hash-based Digital Signature Schemes, Chapter 3 in Post-Quantum Cryptography, Daniel J. Bernstein, Johannes Buchmann, Erik Dahmen (editors), pp. 35–93, Springer-Verlag Berlin Heidelberg, `https://pdfs.semanticscholar.org/b898/21f07eb88bec06a9781d321c7ed8a703a3ed.pdf`.