ElectionGuard Hash Serializing Specification

Moses Liskov

February 1, 2022

The ElectionGuard specification requires, in many places, that a SHA-256 calculation be performed on some ordered, structured values. For instance, a verifiable decryption involves a number of individual-guardian proofs that include a challenge value defined as satisfying the equation:

$$c_i = H(\overline{Q}, (A, B), (a_i, b_i), M_i).$$

Here, $\overline{Q}$ represents the extended base hash for the election, while A, B, a_i, b_i, and M_i are all BigInt values.

Such an instruction is treated as a requirement to perform a particular SHA-256 calculation *modulo q* on some input string derived from the inputs to the H in the given notation.

Exactly how that string is prepared is described recursively, since the inputs may include arrays or tuples (for instance the (A, B) in the example).

Serializing a large integer. A large integer is converted to hexadecimal notation with at most 1 leading zero, so as to represent the number as an even number of hexadecimal deigits, and represented as a utf-8 encoded string. The hexadecimal digits beyond 9 shall be represented as capital letters A through F. For instance, the number 123456789123456789 would be represented as the string “01B69B4BACD05F15”.

Serializing an empty sub-list. When a sub-list occurs in a formula but that list is empty, the entire sub-list is serialized as the string “null” encoded as a utf-8 string.

Serializing a non-empty sub-list. When a *non-empty* sub-list such as (A, B) occurs in a formula such as the above example, this is taken as shorthand for the hash of that list using the $H()$ notation. So the example above, expanded, would more properly be written as

$$c_i = H(\overline{Q}, H(A, B), H(a_i, b_i), M_i).$$

Note that the treatment of an empty sub-list is an exception, as described above.

Serializing a string. Strings are merely utf-8 encoded and not otherwise altered when serialized.

Serializing a small integer. Small integers are converted to decimal notation and then encoded as a utf-8 string. "Small" integers are ones such as the selection limit or indices, things that don't need to have values approaching the parameters p or q.

Serializing a list. The serialization of a list is calculated by serializing each of the list elements in the given order, with the pipe character ("—") as a separator. The pipe character occurs before each element and also after the final element, so $H(1, 2, 3, 4, 5)$ would be calculated based on the serialization "|1|2|3|4|5|".

Integerizing a SHA-256 output. A SHA-256 output is treated as a large integer, and ultimately represented as a string. The SHA-256 output is first treated as an array of bytes, then converted to an integer, big-endian style. That integer is then reduced modulo $q - 1$.

All values described using $H()$ notation are ultimately considered to be numbers modulo $q-1$ in this sense. When such values are subsequently used as an input to another $H()$ expression, they are treated as large integers and represented in hex as described above.

Examples.

- $H($"hello world"$) := SHA - 256($"|hello world|"$)$ modulo $q - 1 =$ 0x3658724c7b35cb1130e4896acfe5903d78bf219e68cf50a3252bf35800174ec6.

- $H(1,2,3) := SHA - 256(\text{“}|1|2|3|'')$ modulo $q-1 =$
  0xe132dc90d35f9705f47bbabf0105c0bf1f10ae13ac463d02067b7ac47955797b.
- $H(0,(1,2,3)) = H(0,H(1,2,3)) := SHA - 256($
  “$|0|E132DC90D35F9705F47BBABF0105C0BF$
  $1F10AE13AC463D02067B7AC47955797B|'')$
  modulo $q-1 =$
  0xd5bef602b3664e328c54f9169116a3f48bb99cd107e54fcb31929db479b4c998.
- $H(12) := SHA - 256(\text{“}|12|'')$ modulo $q-1 =$
  0xeca46619243c31b97422e995c44293a2fc08e63a0d1d0dbf17e49b462d450ad9
- $H(1, \text{“}2|3'') := SHA - 256(\text{“}|1|2|3|'')$ modulo $q-1 = H(1,2,3) =$
  0xe132dc90d35f9705f47bbabf0105c0bf1f10ae13ac463d02067b7ac47955797b.