

**Rolling Re-Evaluation of the Swiss Post e-Voting System:  
Version 1.5.2**

**Audit Scope 1: Cryptographic Protocol**

Aleksander Essex

Department of Electrical and Computer Engineering  
Western University, Canada  
`aessex@uwo.ca`

December 31, 2025

Submitted to the Swiss Federal Chancellery

## Management Summary

In cooperation with the Chancellery, I re-evaluated the Swiss Post e-voting system, covering changes made between system version 1.5.1 and the present version (1.5.2). A summary of the issues and findings is as follows:

1. **Review of changes:** Review of changes to system specification 1.5.2. Minor changes—no issues were identified. Review and acknowledgement of pending issues EX-404 (improper output domain), EX-405 (errors in streamable authenticated encryption), EX-406 (typos in specification) and EX-408 (edge cases in the primitives specification) to be addressed in system version 1.6.0. Acknowledgement of resolved issues EX-398 (nonce reuse risk), EX-409 (zero exponents in multi-exponentiations) in Section 4.
2. **Full validation of GetEncryptionParameters:** Based on the foundational cryptographic importance of the GetEncryptionParameters algorithm (Algorithm 8.7), combined with several historic and recent (now resolved) issues, I implemented the algorithm based on the current specification (version 1.5.1) and verified the correctness of the test vectors in Section 3.2.
3. **EX-413:** A minor issue of clarification around test vector inputs was identified as a result of the GetEncryptionParameters validation. Discussed in Section 3.3.
4. **EX-407:** Recommendations about how to (partially) approach credential risk estimation, including the need to identify the appropriate entity to conduct this estimation (e.g., Swiss Post, cantons, etc.). Discussed in Section 4.

## Version History

|                   |                                             |
|-------------------|---------------------------------------------|
| December 31, 2025 | Minor editorial revisions.                  |
| November 7, 2025  | Initial draft submitted to the Chancellery. |

# Table of Contents

|                                                                      |    |
|----------------------------------------------------------------------|----|
| Management Summary .....                                             | ii |
| 1 Documents Examined.....                                            | 2  |
| 2 Changes to System Specification 1.5.2.....                         | 4  |
| 3 Continued Improvement – Primitives Specification 1.5.1 .....       | 4  |
| 3.1 Previous Issues with Parameter Generation and Test Vectors ..... | 4  |
| 3.2 Full Validation of GetEncryptionParameters and Test Vector ..... | 4  |
| 3.3 Clarification on Test Vector Inputs .....                        | 5  |
| 4 Other Issues (Resolved or in Progress) .....                       | 5  |
| A Validation of Test Vectors in GetEncryptionParameters .....        | 8  |

# 1 Documents Examined

Below is a list of the history of documents examined. For each document, the first column represents the version of my report. The second column lists the document version examined in my report. The third column lists the date the respective report was submitted to the Chancellery.

| <b>Primitives Specification</b>                                                                                                                                                       |                  |            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------|------------|
| <b>Description:</b> Pseudocode specifications of cryptographic functions used by the Swiss Post system. Referred to throughout this document as the <i>primitives specification</i> . |                  |            |
| Report                                                                                                                                                                                | Version Examined | Date       |
| 2025 Rolling Re-examination (This Document)                                                                                                                                           | 1.5.2            | 2025-11-07 |
| 2025 Rolling Re-Evaluation                                                                                                                                                            | 1.5.0-1.5.1      | 2025-09-12 |
| 2025 Rolling Re-Evaluation                                                                                                                                                            | 1.4.2            | 2025-05-30 |
| 2024 Rolling Re-Evaluation                                                                                                                                                            | 1.4.1            | 2024-07-16 |
| 2024 Rolling Re-Evaluation                                                                                                                                                            | 1.4.0            | 2024-07-16 |
| 2023 Rolling Re-Evaluation                                                                                                                                                            | 1.3.0            | 2023-08-01 |
| 2023 Rolling Re-Evaluation                                                                                                                                                            | 1.2.1            | 2023-08-01 |
| 2023 Addendum II                                                                                                                                                                      | 1.2.0            | 2022-12-09 |
| 2022 Re-Examination (Addendum I)                                                                                                                                                      | 1.0.0            | 2022-06-24 |
| 2022 Re-Examination                                                                                                                                                                   | 1.0.0            | 2022-06-24 |
| 2021 Final Report                                                                                                                                                                     | 0.9.8            | 2021-10-15 |
| 2021 Preliminary Report                                                                                                                                                               | 0.9.5            | 2021-06-22 |

**Available:** <https://gitlab.com/swisspost-evoting/crypto-primitives/crypto-primitives/-/blob/master/Crypto-Primitives-Specification.pdf>

## System Specification

**Description:** Document describing the steps, phases and procedures of setting up, executing and verifying an election using the Swiss Post system. Referred to in this document as the *system specification*.

| Report                                         | Version Examined     | Date       |
|------------------------------------------------|----------------------|------------|
| 2025 Rolling Re-examination<br>(This Document) | 1.5.1<br>(Continued) | 2025-11-07 |
| 2025 Rolling Re-examination                    | 1.5.0-1.5.1          | 2025-09-12 |
| 2025 Rolling Re-examination                    | 1.4.2                | 2025-05-30 |
| 2024 Rolling Re-examination                    | 1.4.0                | 2024-07-16 |
| 2023 Rolling Re-examination                    | 1.3.0                | 2023-08-01 |
| 2022 Re-Examination                            | 1.0.0                | 2022-06-24 |
| 2021 Final Report                              | 0.9.7                | 2021-10-15 |
| 2021 Preliminary Report                        | 0.9.6                | 2021-06-25 |

**Available:** [https://gitlab.com/swisspost-evoting/documentation/-/blob/master/System/System\\_Specification.pdf](https://gitlab.com/swisspost-evoting/documentation/-/blob/master/System/System_Specification.pdf)

## 2 Changes to System Specification 1.5.2

Minor changes were made in systems specification version 1.5.2<sup>1</sup> mainly pertaining to clarifications of findings made by Thomas Haines. I reviewed these changes and found no issues.

## 3 Continued Improvement – Primitives Specification 1.5.1

### 3.1 Previous Issues with Parameter Generation and Test Vectors

In my report dated September 12, 2025, I observed in the cryptographic primitives specification (version 1.5.0),<sup>2</sup> that the test vector for `get-encryption-parameters.json` associated with Algorithm 8.1 (`GetEncryptionParameters`) used different group parameters than the test vectors of other Algorithms. In my report dated July 16, 2024, I found `get-encryption-parameters.json` was using outdated generator values (Issue EX-261) and contained a parsing error (EX-262). Additionally, I made several recommendations about the parameterization of the Miller-Rabin primality test. Earlier reports also raised various issues—all of which have now been resolved.

|                                                                                              |
|----------------------------------------------------------------------------------------------|
| <b>Issue:</b> Various <code>GetEncryptionParameters</code> issues (Resolved)                 |
| <b>Description:</b> Various issues with primality testing, parameterization and test vectors |
| <b>Recommendation:</b> Fix the mentioned issues with the test vectors.                       |
| <b>Action Taken:</b> Test vector issues are resolved.                                        |

### 3.2 Full Validation of `GetEncryptionParameters` and Test Vector

My previous analysis had only partially checked the validity of the `GetEncryptionParameters` algorithm, i.e., that the test vector formed a valid algebraic group with the expected size and structure. However, I did not previously check the entire algorithm against the given test vector, namely that the test vector outputs were correct for the input seed value.

Given that (a) correct parameter generation is a foundational security requirement of the system, (b) past work by other researchers identified critical vulnerabilities in early system versions, and (c) my own subsequent analysis in later versions identified several (albeit minor) issues with the algorithm, primality testing approach, parameterizations,

<sup>1</sup> Swiss Post Cryptographic Systems Specification version 1.5.2. Available: [https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/raw/documentation-1.7.2.0/System/System\\_Specification.pdf](https://gitlab.com/swisspost-evoting/e-voting/e-voting-documentation/-/raw/documentation-1.7.2.0/System/System_Specification.pdf)

<sup>2</sup> Swiss Post Cryptographic Primitives Specification version 1.5.1. Available: <https://gitlab.com/swisspost-evoting/crypto-primitives/crypto-primitives/-/raw/crypto-primitives-1.5.0/Crypto-Primitives-Specification.pdf>

and test vectors, I felt it was important, therefore, to verify the current algorithm and test vectors fully. My implementation (see Appendix A) successfully validates the test vector, i.e., that the output parameters (a) were correctly derived from the input seed value, (b) passed the checks stated in the algorithm, and (c) had the correct and expected cryptographic size, structure and properties.

### 3.3 Clarification on Test Vector Inputs

In the process of validating the `GetEncryptionParameters` test vector, I observed the following minor issue.

**Issue EX-413:** Explain omission of small primes list from `GetEncryptionParameters` test vector input

**Description:** Someone independently implementing **Algorithm 8.1** (`GetEncryptionParameters`) and verifying the test vectors in `get-encryption-parameters.json` will notice that the upper bound of the small prime list `sp`, i.e.,  $sp_{l-1} \in sp$  is not specified. It may not be immediately obvious to such a person that the choice of  $sp_{l-1}$  only affects the algorithm's performance, i.e., the output parameters  $p, q, g$  are not a function of  $sp_{l-1}$ .

On that basis, it is reasonable to omit  $sp_{l-1}$  from the test vector's input values, however the reason for this should be clarified in the specification to assure the tester that they can pick whichever upper-bound they wish and it will not result in differing output values for  $p, q, g$ .

**Recommendation:** Clarify the functional independence of domain parameters relative to  $sp_{l-1}$  after the following text in Sec. 8.2: "The incremental search allows to test the candidates for compositeness against the list of small primes much more efficiently, by retaining the remainders of the divisions. This significantly speeds up the algorithm's performance."

## 4 Other Issues (Resolved or in Progress)

I reviewed Swiss Post's acknowledgement of JIRA issues EX-404 (improper output domain), EX-405 (errors in streamable authenticated encryption), EX-406 (typos in specification), and EX-408 (edge cases in the primitives specification) and note they plan to address them in system version 1.6.0. Comments on additional issues are as follows.

|                                                                                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Issue:</b> EX-398: Nonce reuse risk (Resolved)                                                                                                                                                                                          |
| <b>Description:</b> Potential for nonce reuse given notation ambiguity in the authenticated encryption specification.                                                                                                                      |
| <b>Recommendation:</b> Added clarification needed to prevent the possibility of nonce reuse.                                                                                                                                               |
| <b>Action Taken:</b> Swiss Post will clarify that invocations of <code>AuthenticatedEncryption</code> are <i>stateful</i> and that the risk of nonce reuse is mitigated across successive calls if the values are only updated internally. |

|                                                                                                                                                                                                                                                                                                           |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Issue:</b> EX-409: Risk of zero exponents in multi-exponentiations (Resolved)                                                                                                                                                                                                                          |
| <b>Description:</b> A private exponent of zero could cancel the contributions of all other exponents, regardless of order of exponentiation.                                                                                                                                                              |
| <b>Recommendation:</b> Restore the caveat that was removed from the primitives specification Section 8.4.                                                                                                                                                                                                 |
| <b>Action Taken:</b> Superseded by discussion in EX-200 by excluding public keys equal to one. Although zero exponents can still be applied <i>independently</i> of public-key values in general, I accept the resolution/closure of this issue in this specific context by excluding public keys of one. |

|                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Issue EX-407:</b> Credential risk estimation                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| <p><b>Description:</b> The issue of weak voter credentials should remain a matter of conversation. I recognize that there is no obvious solution at present. However, our recent work identified significant risk in Ontario cities using a similar mailed PIN/date-of-birth voter authentication approach [1]</p> <p>Although the Ontario context contains several unique aggravating factors that likely elevate the risk above the Swiss case, the attack complexity still needs to be estimated in the latter context. Perhaps the first question to be explored is: <i>who</i> is the appropriate entity to conduct this analysis?</p> |
| <p><b>Recommendation:</b> (1) Identify who is the appropriate entity for estimating credential attack risk (e.g., Swiss Post, Cantons, etc.), (2) have them propose a systematic methodological framework for assessing risk, (3) attempt to articulate <i>some</i> quantification of risk based on the estimated attack complexity.</p>                                                                                                                                                                                                                                                                                                    |

## References

1. Klassen, E., Brunet, J., Goodman, N., Essex, A.: Credential Attacks in Ontario's Online Elections. In: Electronic Voting: International Joint Conference (E-Vote-ID). Lecture Notes in Computer Science, vol. 16028 (2025), [https://link.springer.com/chapter/10.1007/978-3-032-05036-6\\_9](https://link.springer.com/chapter/10.1007/978-3-032-05036-6_9)

## A Validation of Test Vectors in GetEncryptionParameters

Python implementation of Swiss Post's improved safe prime generation algorithm. Note: For a more direct comparison, we did not implement Swiss Post's sieving approach.

```
1 # Swiss Post Crypto Primitives Specification v1.5.1
2 # Algorithm 8.1 GenEncryption Parameters
3 # Verification of implementation and test vectors
4 import hashlib
5 import base64
6 import math
7 from gmpy2 import is_prime
8
9 # A list of small primes with arbitrary author-selected upper-bound
10 sp = [5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79,
11 ↪ 83, 89, 97]
12
13 # Test vector input values taken from get-encryption-parameters.json
14 bit_len = 3072
15 lam = 128
16 seed = b'31'
17
18 # Test vector output values taken from get-encryption-parameters.json
19 p_SwissPost_testvector_bytes =
20 ↪ base64.b64decode('AL7c3jQFuKGNbHYV/P+X2xwpzSymnxuxQy5pDh6UeDb8HekWDVwq3uU
21 u0kT3mX7M4Z/5edAMw8zjeE2mxkldDYczeySrsP2EjHnrVPKYNJOW+uQDGjt+wr8xPK7zarGRytNtSu
22 /f+of3LarLLqhU//zMZumcKJarHrqTNBwAbd0qTdBrcyP815tffGHeOYG3NLLcHIAeSYsmR+jEMB2
23 /OXh/HH9eaHORjyppdQQ23No1YY3eqknPe7AUbYNUbPdDCL61WG6vjxZzGfrKEOH7m0/jDj2oLNN6Czv
24 kpuFPDsUsbNa4eqCogsMPi3FrNofMuOuewb9nQHxRQjFdk9/6XTfgrbloWTvGapmWld8RsBZLIaYve
25 gpwBtSe+N6zFAjmatU6SmvjyDvCchMcpqVRO34VCdNwhIMr6G8C0IOfH8ZadzUwsCNY4q0Gbaosi8d
26 bxg7GRL1SwRch01eZo0oIHPvkhbjEGWXP/mh0p3ERQWUkSCfqVQNBrZmYR6170dw==')
27
28 q_SwissPost_testvector_bytes =
29 ↪ base64.b64decode('X25vGgLCuMa20wr+f8vtjhTm1lNPjdihlzSHD0o8G3409IsGrhVvcpd
30 pInvMv2Zwz/y86AZh5nG8JtNjJK6Gw5m9klXYfsJGPPXeeUwaSct9cgGNHb9hX5ieV3m1WMjlabald+
31 /9Q/uW1WXXVCP//mYzdM4US0iPXUmaDgA26dUm6DWhlZaf5rza++M09owNuaWW40Q08kxZMj9GIYDt/
32 ovD+OP680PojHLS6ghtubRKwxu9VJ0e92Ao2wbcCe6GEX1qsN1fHlzmM/WUicP3Np/GHhtQwab0FnfJ
33 TcKeHYOpY2a1w9UFRBYyfFuLWbQ+Zcdc9g370gPiihGK6V7/0um/BW3LQsneM1TmTK74jYCyWQ0xe9B
34 TgDak98b1mKBHM1ap01NfHHkHeE5CY5TUqidvwqE6bhCQZXON4EcQc+P4y07mphYEBlxVoM21RZF463
35 jB2MiXKlgi5CdK8zRpQQ0ffJC3GINGuf/NDpTuIigspIke/UqgaDwzMwj1r2c7')
36
37 g_SwissPost_testvector_bytes = base64.b64decode('Ag==')
38
39 p_SwissPost_testvector = int.from_bytes(p_SwissPost_testvector_bytes, 'big')
40 q_SwissPost_testvector = int.from_bytes(q_SwissPost_testvector_bytes, 'big')
41 g_SwissPost_testvector = int.from_bytes(g_SwissPost_testvector_bytes, 'big')
42
43 bytes_to_generate = bit_len // 8
44 h = hashlib.shake_256(seed)
45
46 # Algorithm 8.1
47 qhat_b = h.digest(bytes_to_generate) # Line 1
48 q_b = bytes.fromhex('02') + qhat_b # Line 2
49 q_p = int.from_bytes(q_b, 'big') >> 3 # Line 3
50 q = q_p - (q_p % 6) + 5 # Line 4
51 r = [q % sp_i for sp_i in sp] # Lines 5-8
52 delta = 0 # Line 9
```

```

51 candidates_checked = 0
52
53 # Lines 10-23
54 while True:
55     if 0 not in [(r[i] + delta) % sp[i] for i in range(len(sp))] or 0 not in [(2 *
↪ (r[i] + delta) + 1) % sp[i] for i in range(len(sp))]:
56         candidates_checked += 1
57         if is_prime(q + delta, int(lam / 2)) and is_prime(2 * (q + delta) + 1,
↪ int(lam / 2)):
58             break
59         delta += 6
60
61 q = q + delta # Line 24
62 p = 2 * q + 1 # Line 25
63
64 # Lines 26-30
65 if pow(2, q, p) == 1:
66     g = 2
67 else:
68     g = 3
69
70 # Check derived values match test vector outputs and have expected properties
71 assert(p == p_SwissPost_testvector)
72 assert(q == q_SwissPost_testvector)
73 assert(g == g_SwissPost_testvector)
74 assert(math.ceil(math.log(p,2)) % 8 == 0)
75 assert(math.ceil(math.log(p,2)) == bit_len)
76
77
78 # Verified
79 print("Parameters successfully verified.")
80 print("Search completed after checking", candidates_checked, "prime candidates.")

```