Selectio Helvetica *light*

Oliver Spycher
Bern University of Applied Sciences

Draft

ZDA (Zentrum für Demokratie Aarau) have chosen Selectio Helvetica *light* (SH *light*) as Baloti's e-voting system to run the 26.09.2010 and 28.11.2010 federal referendums. The underlying protocol and its implementation have been developed and hosted by BFH (Bern University of Applied Sciences) in partnership with the University of Fribourg and the Swiss E-Voting Competence Center.

Selectio Helvetica (SH) is a project aiming at providing voters with the confidence that their electronic vote is counted as intended, that the final tally includes all cast votes, that only authorized votes are counted (*verifiability*), and that the secrecy of the ballot is respected (*privacy*).

SH *light* is a product that satisfies the *privacy* requirement by combining cryptographic and organisational measures. Remote e-voting systems used by governments currently rely on the same approach.

SH *light* is a crucial milestone towards the product SH, which satisfies *privacy* and *verifiability* only under the use of cryptographic techniques. Thus, it becomes obsolete to trust in the host's honesty at following its organisational procedures.

The following paragraphs explain SH *light* and give an outlook towards SH. In the context of the Baloti project, the term *vote organiser* refers to ZDA, *voting host* refers to BFH.

1 SH *light* in a Nutshell

SH *light* requires two types of interaction with the voter.

Obtain a Voting Code Before casting a vote, the voter needs to obtain his personal voting code by indicating his e-mail address. If the vote organiser assesses the e-mail address to be valid, i.e. belonging to an eligible voter, the voting provider instructs the voting host to send a voting code to that address. The same voting code can optionally be re-used in subsequent voting events.

Cast a Vote The voter uses the vote organiser's platform to indicate how he wants to vote. The vote organiser can additionally ask the voter to enter some personal data for raising statistics. After all information is specified, the voter enters his voting code and casts his vote. The information regarding the vote, the personal data, and the voting code are passed to SH *light* functionality on the voter's browser which encrypts the vote and the personal data. Note that the steps described so far do not require any interaction between the browser and the servers. Based on the voting code, SH *light* downloads from its servers the information needed to anonymously sign the encryption. Both the encryption and the signature are then posted to the electronic ballot-box. The voting code is not sent to the electronic ballot-box. Instead, the signature is verified against the voter's anonymous key, which is not linkable to his identity. If the verification holds, the vote is authorized.

Establishing the Tally The voting host decrypts all authorized votes and sends the result to the vote organiser. The vote organiser computes the final tally and publishes it along with the statistics on the personal data.

2 SH *light* in More Detail

SH *light* incorporates the notions of a *voter roll* and an *electronic ballot-box*.

Prior to the first voting event, the voter roll is initialized with one entry per potential voter. An entry comprises all information needed to compute digital signatures, as defined per the Digital Signature Algorithm (DSA). Value g denotes a generator of a large prime-order group $\mathbb{G}_q$, where DDH is hard.

An Initial Entry of the Voter Roll:

- **Voting Code**: A random value, 12 digits in base-36 representation
- **DSA private key** s : A random value from $\mathbb{Z}_q$
- **DSA public key** S: Computed as g^s

Prior to *each* voting event, the anonymization generator $\hat{g}$ is computed as g^α, where α is chosen at random from $\mathbb{Z}_q$. Further, an ElGamal key pair (e, d) is established used for encrypting and decrypting the votes. The electronic ballot-box is initialized with one entry per potential voter:

- DSA anonymous key $\hat{S}$: Computed as S^α and placed at arbitrary position in the electronic ballot-box

Voters who know their DSA private key s are able to compute their DSA anonymous key, since $\hat{S} = \hat{g}^s$. However, it is computationally infeasible to compute s, when only given $\hat{S}$ or S. Further, it is computationally infeasible to establish

the link from $\hat{S}$ to its original value S given the lists of all DSA public keys and DSA anonymous keys, respectively. These features ground on assumptions, which modern cryptographic techniques strongly rely on.

Obtain a Voting Code At the vote organiser's request, the voter's e-mail address is associated with a free entry of the voter roll. The voting code is sent to that address.

Cast a Vote After the voter has chosen to cast his vote, his browser passes the voting code to the SH *light* server per web request. The server returns the DSA private key s associated with that voting code. The browser application concatenates the vote with the voter's personal data and maps it to a group element v of $\mathbb{G}_q$. The result is ElGamal encrypted as $\mathrm{Enc}_e(v)$, then signed with s and $\hat{g}$ as $\mathrm{Sign}(\mathrm{Enc}_e(v))$. The browser computes $\hat{S}$ as $\hat{g}^s$ and sends $(\mathrm{Enc}_e(v), \mathrm{Sign}(\mathrm{Enc}_e(v)), \hat{S})$ to the electronic ballot-box. The electronic ballot-box only accepts the vote, if it contains $\hat{S}$ and the signature $\mathrm{Sign}(\mathrm{Enc}_e(v))$ is successfully verified against $\mathrm{Enc}_e(v)$, $\hat{S}$, and $\hat{g}$. Since $\hat{S}$ is not linkable to its original value S, the vote is authorized while preserving the voter's anonymity.

Establishing the Tally The voting host uses its private key d to decrypt all votes $\mathrm{Enc}_e(v)$ in the electronic ballot-box. The resulting values v are unmapped and sent to the vote organiser.

3 Shifting from SH *light* to SH

When using SH *light*, the voting host needs to follow certain operational procedures in order to guarantee *privacy*. For instance, staff who are able to access the DSA private keys s in the voter roll may not be able to access the DSA anonymous keys in the electronic ballot-box. Further, they must assert that the value α used to generate the DSA anonymous keys remains unrecoverable. The host also needs to ensure that none of the cast encrypted votes are decrypted prematurely.

By employing group threshold schemes [GJKR99], these operational measures become obsolete, since the critical values (*voting code*, s, α, d) are actually *unkown*. Thus, *privacy* is provided solely on the base of cryptographic tools. Furthermore, the fields DSA public key, e-mail address (alternatively a pseudonym chosen by voters), DSA anonymous key and all values cast to the electronic ballot-box can be published on the voting provider's website. Thus, voters can verify that their vote is counted as intended, that the final tally includes all cast votes, and that only authorized votes are counted (*verifiability*).

Group threshold schemes are based on Shamir's secret sharing scheme [Sha79]. Independent trusted parties (*trustees*) compute their secret share of each critical

value. Computing the critical values yielded by the shares requires a majority of collaborating trustees. They jointly compute the public figures by using their secret shares without revealing them. Each trustee justifies the correctness of its partial computation by providing a non-interactive zero-knowledge proof [BG92]. Thus, all computations can be verified by independent entities (e.g. voters), without leaking any information on the critical values.

The protocol underlying SH is an enhancement of [SH10]. Instead of trusting the voting host in following his procedures, voters merely need to trust that no majority of the independent trusted parties collude to compute or even reveal sensitive values when they are not supposed to. Trustees can be organizations from academia, government, industry, or even NGOs, who have a tradition of vote-related engagement, such as the OSCE.

BFH are currently in the process of implementing the product SH and acquiring appropriate trustees.

References

[BG92] M. Bellare and O. Goldreich. On defining proofs of knowledge. In E. F. Brickell, editor, *CRYPTO'92, 12th Annual International Cryptology Conference on Advances in Cryptology*, LNCS 740, pages 390–420, Santa Barbara, USA, 1992.

[GJKR99] R. Gennaro, S. Jarecki, H. Krawczyk, and T. Rabin. Secure distributed key generation for discrete-log based cryptosystems. In J. Stern, editor, *EUROCRYPT'99, International Conference on the Theory and Application of Cryptographic Techniques*, LNCS 1592, pages 295–310, Prague, Czech Republic, 1999.

[SH10] O. Spycher and R. Haenni. A novel protocol to allow revocation of votes in a hybrid voting system. In *ISSA'10, 9th Annual Conference on Information Security – South Africa*, Sandton, South Africa, 2010.

[Sha79] A. Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979.