Designing a secure messaging application with biometric identification computing on homomorphic encrypted data.

Alexander Stæhr Johansen, 201905865@post.au.dk
Henrik Tambo Buhl, 201905590@post.au.dk

16/02/2022

Introduction

The purpose of the following document is to introduce the bachelor's project "Designing a secure messaging client-server application with biometric identification computing on homomorphic encrypted data".

Motivation

In the present socio-economic environment, people are increasingly dependent on electronic means of communication and verification. This has led public and private institutions to adopt specialized internal communication channels, accessed via. biometric identification, as it is more secure and accurate than traditional forms of distribution of information and methods for authentication.

Therefore, it is of significant interest to build a secure communication system which can verify a person, without compromising the privacy of the individual.

Objectives

The main tasks are to create a secure proof-of-concept messaging web application, with security ensured via the Signal protocol and its underlying XEdDSA and VXEdDSA signatures, Double Ratchet algorithm, X3DH key agreement protocol and Sesame algorithm [1], and a biometric identification system which uses facial recognition, implemented via Eigenfaces [2], as biometric parameter for verification, and homomorphic encryption [3], implemented via Microsoft SEAL [4], for privacy-preserving computation. Sub-tasks include:

- Document the proof-of-concept with a software requirements specification, an architectural document, and a test specification.
- Verify and document the testing of the software requirements specification.
- Discuss what improvements could be made with respect to usability, accuracy and scalability.

Work plan

To accommodate the above objectives, the following work plan will be executed. The work plan is tentative and as such changes will occur. The person responsible for each deliverable has his initial suffixed after this. For example, deliverables with the suffix "(A)" must be created by Alexander Stæhr Johansen.

The Django tutorials referenced below are to be found here [5]. The CKKS introduction can be found here [6], and the Python binding used for implementing the Microsoft SEAL library is found here [7].

Week	Agenda	Deliverables
W5: Introduction.	Do project description refinement.	**D1:** Refined project description. (A and H)
W6: Technological introduction.	Complete Django tutorial 1 and 2. Setup environment for SEAL-Python.	**D2:** Basic chat application. (A)
W7: Technological introduction, thesis writing, specification and Eigenface implementation.	Complete Django tutorial 3, execute the first iteration of the requirement specification, implement PCA and write the sections: "PCA", "power method" and "Eigen shift procedure".	**D3:** Asynchronous chat application with two-factor verification. (A) **D4:** 1st draft of the requirements specification. (A) **D5:** PCA code. (H) **D6:** PCA section. (H) **D7:** Power method section. (H) **D8:** Eigen shift procedure. (H)
W8: Technological introduction, thesis writing, specification and Eigenface implementation.	Complete Django tutorial 4, execute the first iteration of the architectural specification, implement Goldschmidt's algorithm, Eigenface facial recognition and write the corresponding sections: "Goldschmidt's algorithm" and "Eigenface facial recognition.	**D9:** Automated test suite for chat application. (A) **D10:** 1st draft of the architectural specification. (A) **D11:** Goldschmidt's algorithm code. (H) **D12:** Goldschmidt's algorithm section. (H) **D13:** Facial recognition code. (H) **D14:** Facial recognition section. (H)
W9: Thesis writing and Eigenface implementation.	Write the sections: "purpose", "problem definition", "related work" and "software development processes" of the thesis. Implement vector operations and write the corresponding section.	**D15:** Purpose section. (A) **D16:** Problem definition section. (A) **D17:** Related work section. (A and H) **D18:** Software development processes section. (A) **D19:** Vector operations code. (H) **D20:** Final Eigenface implementation. (H)
W10: Specification and homomorphic encryption.	Execute the first iteration of the test specification and begin the introduction to CKKS in SEAL-Python.	**D21:** First draft of the test specification. (A)
W11: Thesis writing, Signal protocol implementation and homomorphic encryption.	Write the section: "Signal protocol" and its associated subsections: "WEdDSA and VXEdDSA", "X3DH", "Double Ratchet" and "Sesame" and begin the implementation of the protocol. Introduction to CKKS in SEAL-Python continued.	**D22:** WEdDSA and VXEdDSA. (A) **D23:** X3DH. (A) **D24:** Double Ratchet. (A) **D25:** Sesame. (A)
W12: Signal protocol implementation continued. Homomorphic Eigenfaces implementation.	Implement Signal protocol. Start with implementing R2.	**D26:** Signal protocol implemented. (A)
W13: Write requirements specification and continue to implement homomorphic Eigenfaces.	Execute the second iteration of the requirements specification. Start to implement homomorphic eigen shift procedure.	**D27:** Second draft of the requirements specification. (A)

W14: Thesis writing, specification and homomorphic Eigenfaces continued.	Execute the second iteration of the architectural specification. Implement homomorphic eigen shift procedure, Goldschmidt’s algorithm and write the sections ”Homomorphic eigen shift procedure” and ”Goldschmidt’s algorithm” .	**D28:** Second draft of the architectural specification. (A) **D29:** Homomorphic eigen shift procedure code. (H) **D30:** Homomorphic eigen shift procedure section. (H) **D31:** Homomorphic Goldschmidt’s algorithm code. (H) **D32:** Homomorphic Goldschmidt’s algorithm section. (H)
W15:	**Break / catch-up.**	
W16: Thesis writing, specification and homomorphic Eigenfaces.	Execute the second iteration of the test specification. Implement HPCA, homomorphic facial recognition, vector operations and write the sections: ”HPCA”, ”Homomorphic facial recognition” and ”Vector operations”.	**D33:** Second draft of the test specification. (A) **D34:** HPCA code. (H) **D35:** HPCA section. (H) **D36:** Homomorphic facial recognition code. (H) **D37:** Homomorphic facial recognition section. (H) **D38:** Vector operations. (H)
W17: Thesis writing and homomorphic Eigenfaces implementation in web application.	Implement homomorphic Eigenfaces in the web application. Write the section: ”Homomorphic Eigenfaces implementation” and adjust previous sections.	**D39:** Homomorphic Eigenfaces implemented into the web application. (A) **D40:** Homomorphic Eigenfaces implementation (H)
W18: Thesis writing and specification.	Finalize the requirements and architectural specification. Start on the results section.	**D41:** Final requirements specification. (A) **D42:** Final architectural specification. (A)
W19: Thesis writing and specification.	Finalize the test specification and continue on the results section. Start on the conclusive remarks section.	**D43:** Final test specification. (A)
W20: Thesis writing.	Write the rest of the implementation section of the thesis.	**D44:** Implementation. (A and H)
W21: Thesis writing.	Write the rest of the results section of the thesis and the conclusive remarks section.	**D45:** Results. (A and H)
W22: Thesis writing.	Write the abstract and the sections: ”conclusive remarks” and ”future work” of the thesis.	**D46:** Conclusive remarks. (A and H) **D47:** Abstract. (A and H) **D48:** Future work. (A and H)
W23: Thesis writing.	Focus on feedback and transcript the previous sections.	
	Hand-in	

References

[1] *Libsignal-client (v0.12.2)*. `https://github.com/signalapp/libsignal-client`. Signal Messenger, LLC. Feb. 2022.

[2] Adil Bouti and Jörg Keller. “Towards Practical Homomorphic Encryption in Cloud Computing”. In: *2015 IEEE Fourth Symposium on Network Cloud Computing and Applications (NCCA)*. 2015, pp. 67–74. DOI: `10.1109/NCCA.2015.20`.

[3] Samanvaya Panda. “Principal Component Analysis Using CKKS Homomorphic Scheme”. In: *Cyber Security Cryptography and Machine Learning*. Ed. by Shlomi Dolev et al. Cham: Springer International Publishing, 2021, pp. 52–70. ISBN: 978-3-030-78086-9.

[4] *Microsoft SEAL (release 3.7)*. `https://github.com/Microsoft/SEAL`. Microsoft Research, Redmond, WA. Sept. 2021.

[5] *Django tutorials*. Django Software Foundation Revision ece488b3. 2018.

[6] *CKKS Introduction*. Daniel Huynh. Sept. 2020.

[7] *SEAL-Python*. Huelse, DreamingRaven, carlee0, uriariel, yntaobc. 2020.