Implementation of FEWD (fair exchange without disputes) with Attestables

Carlos Molina-Jimenez and Jon Crowcroft
Department of Computer Science and Technology, University of Cambridge
carlos.molina, jon.crowcroft @cl.cam.ac.uk

5th October 2023

Abstract

This document explains how FEWD (Fair Exchange Protocol Without Disputes) can be implemented o using attestables. An attestable is a Trusted Execution Environment (TEE) that can be created on the basis of cheri-capabilities available in Morello Boards. Precisely, it explains the current progress and challenges that we have been facing.

1 Introduction to fair exchange

To generalise and understand the difficulty, imagine that Alice and Bob have two (one each) items that they wish to swap them. The particular nature of the item (physical, digital, etc.) is irrelevant to FEWD, yet to frame the discussion, let us assume that Alice's and Bob's items are, respectively document D_A and document D_B. Upon completion of a fair exchange protocol, either the two items change owners or, if the protocol is aborted, none of them does.

Fair exchange protocols are needed in several practical applications. In online business D_A can be an item sold online and D_B can be Bob's payment; similarly, D_A can be a business document and D_B a delivery receipt. Take another example, D_A and D_B can be, respectively, Alice's and Bob's covid certificates, that they agree to exchange before face–to–face meeting up for coffee.

Fair exchange is a fundamental distributed system problem that repeatedly manifest itself in several practical applications such as online business transactions. For example, Bob is willing to give money (payment) to Alice in return for an item. Naturally, Bob should not give his money to Alice if he does not receive Alice's item; neither Alice should give her item to Bob if she does not receive Bob's money.

2 Limitations of existing fair exchange protocols

Protocols that have been suggested in the literature share a serious inconvenience: they rely on monolithic Trusted Third Parties (TTP) to resolve potential disputes. A TTP introduces several inconveniences, for instances, it is a stateful component of the fair exchange. As such, it is a position that enables it to gather sensitive information about the exchange.

3 Attestables as trusted execution environments

FEWD is capable of avoiding the emergence of disputes by means of splitting the TTP into three trusted components:

- a trusted execution environment deployed by Alice to execute all and only the sensitive operations that the fair exchange protocol executes on her side.
- a trusted execution environment deployed by Bob to execute all and only the sensitive operations that the fair exchange protocol executes on his side.
- A stateless public bulletin board that is capable of receiving and storing tokens (strings) submitted by Alice and Bob and delivering upon requests.

4 Implementation of FEWD

Intuitively, FEWD relies on two attestables (one for each participant) that are able to conceal and lock documents and release then in synchrony, that is when both attestables agree to release to release them. The model, its API and properties are discussed at large in [2] and in [1]. In summary, to help in the implementation of FEWD, attestables need to meet the following three properties:

1. **concealment**: it is a black box that prevents observation of data and computation.
2. **attestation**: the code an attestable is running can be verified (attested) by the application interested in using the attestable and cannot be changed after attestation.
3. **independent computation**: once the code inside the attestable is launched into execution, it executes independently till it reaches its final state. No body, including its owner, is able to change the course of the computation. Its owner can manipulate the attestable physically, for example switch it off or disconnect its from its network but not with the logical state of the computation.

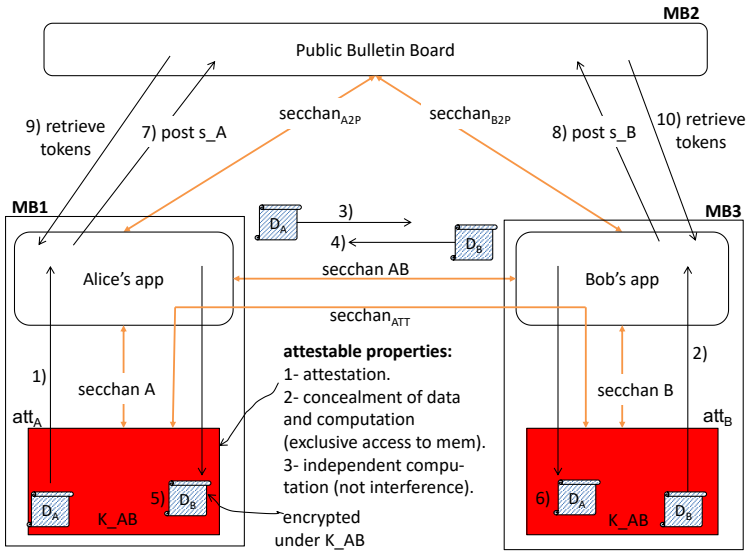

Figure 1: Attestables (att_A) and (att_B) in the implementation of FEWD.

Let us assume that attestables that satisfy these properties are available and explain how they are used in the implementation of FEWD (Fig. 1). In the figure Alice and Bob are the owners of documents D_A and D_B, respectively and they are interested in swapping them over FEWD.

- Communication takes place over secure channels that are shown as orange lines.
- Alice's attestable (att_A) and Bob's attestable (att_B) use secure channel ($secchan_{att}$) and public keys to derive a shared symmetric key, represented as K_{AB}.
- The protocol assume that Alice's and Bob's attestables are already in possession of the documents D_A and D_B, respectively; that is, the description of the protocol omits the step used to transfer the documents from the applications to their attestables.

The execution of FEWD develops as follows:

1. Alice's attestable (att_A) encrypts D_A under K_{AB} and sends its to Alice's application (Alice's app).
2. On Bob's side, Bob's attestable (att_B) encrypts D_B under K_{AB} and sends its to Bob's application (Bob's app).
3. Alice's application sends the encrypted document D_A to Bob's application.
4. Bob's application sends the encrypted document D_B to Alice's application.

5. Alice's application forwards the encrypted document D_B to Alice's attestable. The attestable locks the document.
6. Bob's application forwards the encrypted document D_A to Bob's attestable. The attestable locks the document.
7. Alice's attestable decrypts D_B and verifies that it is the document that Alice is expecting. If it is satisfied, it instructs (step now shown in the figure) Alice's application to post synchronisation token s_A to the PBB. Otherwise it instructs Alice's application to post cancellation token c_A to cancel the execution of FEWD.
8. Bob's attestable decrypts D_A and verifies that it is the document that Bob is expecting. If it is satisfied, it instructs (step not shown in the figure) Bob's application to post synchronisation token s_B to the PBB. Otherwise it instructs Bob's application to post cancellation token c_B to cancel the execution of FEWD.
9. Alice's attestable instructs (step not shown in the figure) Alice's application to retrieve the posted tokens from the PBB. In a normal development Alice's application will retrieve s_A and s_B.
10. Bob's attestable instructs (step not shown in the figure) Bob's application to retrieve the posted tokens from the PBB. In a normal development Bob's application will retrieve s_A and s_B

The retrieved tokens are send by the applications to their attestables. Alice's attestable releases D_B to Alice's application if it is presented with $s_A|s_B$, otherwise, it does not release D_B to Alice's application. In Bob's side, Bob's attestable mirrors Alice's attestable, that is, it releases D_A to Bob's application only if it is presented with $s_A|s_B$.

5 Code and deployment on Morello Boards

From the perspective of the need to protect sensitive information, the architecture shown in Fig. 1 can be separated into two parts:

1. **computation of non-sensitive data**: The top of the figure does store, compute or transmit any sensitive data. Therefore it can be implemented with conventional programming and executed on conventional hardware.

   Notice that in step 4 Alice's application receives Bob's D_B document but it is encrypted under a key that Alice's application does not know, thus, there is no risk of exposing D_B to Alice's application. The same holds for Bob's application in step 5.

   In fact, we have implemented this part of the protocol in Python, though it can run in any hardware, for convenience, we are running it on Morello Boards. As shown in the figure, in our experiments, we deploy Alice'

application in a Morello Board (MB1) and Bob's application on another Morello Board (MB3).

The public bulletin board does hold data that need concealment. Yet, we deploy it on a Morello Board (MB2)

2. **computation of sensitive data**: This part of the protocol processes sensitive data that needs concealment executes code that needs independent computation. Therefore it needs to be executed in trusted execution environments. As shown in Fig. 1, we are planning to use attestables that we are currently implementing.

References

[1] Carlos Molina–Jimenez et al. *Fair Exchange: Theory and Practice of Digital Belongings*. In press, to appear in 2023. World Scientific, 2023.

[2] Carlos Molina-Jimenez and Jon Crowcroft. *The Attestable Model: properties and API.* Visited on 5 Oct 2023. 2023.