Fair Exchange

Mohammad Torabi Dashti
ETH Zürich

Sjouke Mauw
Université du Luxembourg

Contents

1 What is fairness?

Fairness is a broad concept, covering a range of qualifications such as impartiality, courtesy, equity, sportsmanship, etc. Here, we focus on fairness in exchanging (electronic) goods, stipulating that none of the partners can take an undue advantage over the other. When is an interaction between two or more people called fair? Let us proceed with a few examples which reflect fairness as it is understood in the security literature. Our examples, as the cryptographic tradition goes, are scenarios involving Alice and Bob. To learn more about these people see [Gor05].

- Alice and Bob want to divide a piece of cake into two parts, one part for Alice and one part for Bob. None of them trusts the other one for this purpose. Alice gets to cut the cake into two pieces, and Bob gets to choose which part he wants. It is in Alice's interest to maximize the smallest piece of the two. This is because a rational Bob would choose the bigger piece, leaving Alice with the smaller piece. Alice, being a rational person too, prefers to cut the cake into halves.

- Alice takes a taxi. She, however, does not want to pay the driver Bob before he takes her to her destination. Bob, not trusting Alice, does not want to take

> Alice to her destination before she pays. The fee is 100 Euros. Alice rips a 100 Euro bill into halves, and gives a half to Bob. To receive the other half, it is now in Bob's interest to take Alice to her destination. Alice, however, has already "spent" her 100 Euros, and does not benefit from not giving the other half to Bob, once Bob takes her to her destination.

Do these scenarios describe fair interactions? We will analyze them more carefully in the following. In the cut and choose scenario, indeed there is an assumption that neither Alice nor Bob would take the whole cake and simply run away. Under this assumption, cut and choose is fair, in the sense that Alice and Bob will both be content with the result. None of them has any reason to envy the other one, cf. [BPW07].

In the taxi driver scenario, clearly Bob should take Alice to her destination to get paid. Bob, however, knows that Alice has no interest in not paying him. She has already ripped up her 100 Euro bill, and can as well give the other half to Bob, once she is at her destination. [1] Here, Bob relying on the assumption that Alice is rational would take Alice to her destination. Similarly, Alice under the assumption that Bob is rational would rip up the 100 Euro bill. Remark that both parties need to assume that their opponent is rational in this scenario. Indeed, a malicious Alice could harm Bob by not paying him the other half, and a malicious Bob could harm Alice by not taking her to her destination after she has ripped the bill.

Note that no such rationality assumption is needed to ensure fairness in the cut and choose scenario. If Alice cuts the cake unfairly, it is only Alice who is in a disadvantage. In other words, both Alice and Bob are guaranteed to end up with at least half of the cake, without any assumptions on the rationality of the other party; Alice however must play rational to ensure that she gets at least a half.

These examples show that fairness is indeed a subtle issue. In the following, we abstract away various non-technical aspects of fairness, and focus on fairness in electronic exchanges.

2 Fairness in electronic exchanges

At a high level of abstraction, an action, such as signing a contract, may be considered as a single event even though it is made up of a number of more elementary actions. We say that the execution of such a composed action is *atomic* if either all of its sub-actions are executed, or none at all.

Applying this terminology to electronic exchanges, we understand fairness as the basic property of atomicity, meaning that all parties involved in the interaction receive a desired item in exchange for their own, or none of them does so. The exchange of items is often governed by a set of rules, stating which steps are taken in which order and by whom. Such a set of rules is referred to as a *protocol*. Fairness here is thus a property of the means of the exchange, e.g. a protocol, rather than the exchanged material per se.

The difference between electronic exchange and conventional commerce and barter essentially lies in enforceable laws. If Alice pays for a product to Bob and Bob fails to deliver the product (as stated in their contract), then Alice can resort to litigation, which is enforceable by law. In electronic commerce, however, litigation is often not viable. This is because laws to evaluate and judge based on electronic documents are mostly inadequate, the exchange partners may be subject to different laws (e.g. they may live in different countries) and, more importantly, the accountable real world party behind an electronic agent may not be traceable, cf. [San97].

The current practice of electronic commerce, therefore, heavily relies on trusted third parties. Most vendors on Internet, for instance, offer little beyond browsing their catalogues, while contract signing and payment often go via a credit card company. The trust in these sites is largely built upon the trust users have in the credit card companies, which keep records and provide compensation in case of fraud. Fairness in electronic commerce in fact turns out to be unachievable if there is no presumed trust among the involved parties [EY80]; see Section 3.2. We thus focus on fair exchange protocols which rely on trustees.

When there is a mediator who is trusted by all the exchange partners, there is a canonical solution to fair exchange. The items subject to exchange can be sent to the trusted entity and then he would distribute them if all the items arrive in time. If some items do not arrive in time, the mediator would simply abandon the exchange. Figure 1 shows such an exchange. In the first phase, A sends i_A to the mediator T and B sends i_B to T. Here i_A and i_B are the items subject to exchange. In the next phase, T sends i_B to A and i_A to B. This mechanism is inefficient, as it involves the mediator in every exchange, and therefore does not scale well. The involvement of the trusted party can in fact be reduced to the point that he would need to take actions only when something goes amiss in the exchange (e.g. an item does not arrive in time). Such protocols are preferred when most exchange partners are honest and, thus, failed exchanges are infrequent; hence these protocols are called *optimistic* protocols.

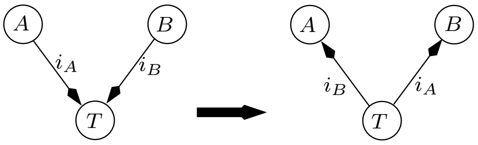

Figure 1: Exchange using a trusted mediator

2.1 Fair exchange flavors

Various flavors of the fair exchange (FE) problem exist in the literature, e.g. fair contract signing (CS), fair payment (FP), fair certified email (CEM), and fair exchange of secrets (ES). Below, we introduce these FE variants via examples:

- Fair contract signing (CS): Alice and Bob have agreed on a contract and would like to sign it electronically. [2] Alice gives her signature on the contract to Bob only if she receives the contract signed by Bob. Similarly, Bob signs the contract and passes it to Alice, only if he receives Alice's signature. In short, they want to simultaneously exchange their signatures.

- Fair payment (FP): Alice sees Bob's electronic book on the Internet and wants to buy it, but she does not want to send her digital coins to Bob before receiving the book and making sure that it is indeed what he has advertised. Similarly, Bob does not want to send his electronic book to Alice before receiving Alice's coins and making sure that they are genuine. They want to simultaneously exchange their digital items.

- Fair certified email (CEM): Alice wants to send an email to Bob in exchange for a receipt. The receipt is a proof that shows Bob has received the email. Therefore, the receipt must uniquely identify the content of the email. Bob is in turn willing to send back the receipt to Alice only if he actually receives Alice's email. Notice that in this case, Alice and Bob do not aim at simultaneous exchange. This is because of the inherent asymmetry of the problem, namely the receipt depends on the content of the email.

- Fair exchange of secrets (ES): Alice and Bob each possess a secret that is not known to the other one. Alice and Bob would like to exchange their secrets, but neither of them wants to reveal their secret unilaterally. Note that this exchange is meaningful only if Alice and Bob can recognize the expected secrets. That is, they can verify that a received data is indeed the other party's secret. Otherwise, any protocol that distributes random bits would be acceptable, since Alice would think that the junk is actually Bob's secret, and similarly for Bob.

Although these problems are similar and we refer to them collectively as FE, there are subtle differences between them. For instance, CEM and CS are different in simultaneity, and CEM is different from ES in that the receipt of an email is not precisely defined in CEM, and can thus be different from one protocol to another (as it happens to be in practice), while ES is to exchange the secrets themselves. It is also notable that in ES the participants are assumed to be able to recognize the other party's secret. This is a trivial precondition when it comes to CS, because signatures are the subject of exchange, and digital signatures always have a verification algorithm associated to them.

Yet another class of FE protocols consists of the non-repudiation protocols (NR). Their aim is to exchange evidences in a fair manner, meaning that Alice receives an evidence of receipt iff Bob receives an evidence of origin on a certain document. An evidence of receipt can for instance be formed by Bob signing a document that he has received from Alice, and an evidence of of origin can be formed by Alice signing the document she sends to Bob. The participants are further required to be accountable for (i.e. they cannot deny) the promises they utter in the course of the exchange. We do not distinguish NR and CEM protocols in this document, since these are conceptually very similar. The challenge in NR protocols is to exchange the evidences in a fair way, otherwise, non-repudiation of the evidences can be achieved using standard digital signatures, cf. [ZG97b].

We remark that in fair contract signing, *termination* is a challenging problem in practice. For instance consider the following scenario. A contract to sell a property P for some amount of money M has a meaning only if it is seen by an authority, e.g. the city hall, to transfer the ownership of the property to the buyer. In this situation, the seller can sign the following statement and send it to the buyer.

> The seller declares that if the buyer signs this letter, then the buyer will own the property P, and the buyer thereby promises to transfer M dollars to the account of the seller.

Now, the buyer cannot have the property P without pledging to pay M dollars to the seller; this exchange is thus fair.

The above solution for fairness, however, leaves the seller in a disadvantageous position regarding the termination of the exchange. Namely, the buyer can sign the statement whenever he wishes. Meanwhile, the seller is the owner of the property only in a weak sense. The seller cannot sell the property to anyone else, and does not know when the property is not his anymore. Nevertheless, the seller knows that if some day the property is owned by the buyer, the seller is eligible to the amount of money mentioned in the statement. This example motivates the *timeliness* requirement for fair exchange protocols, as described below.

2.2 Fair exchange requirements

In the literature, there is no consensus on what FE protocols (or its variants) have to provide. Nevertheless, most authors seem to include formulations of fairness and timeliness similar to the ones proposed by [Aso98]. Below, we informally describe these goals for two parties, named A and B:

- *Fairness* states that if A terminates the protocol in a state where A has B's item, then when B terminates the protocol, B has A's item, and vice versa. This property is often referred to as *strong* fairness, cf. [PVG03].
- *Timeliness* states that any honest participant can terminate the exchange unilaterally, i.e. without any help from the opponent. Timeliness guarantees that none of the participants can arbitrarily force the other one to wait for the termination of the exchange.

Any protocol that achieves these goals is said to solve FE. [3]

We remark that each variant of FE can have its own specific requirements. For instance, timeliness is sometimes deemed unnecessary for CEM, e.g. see [Mic03]. See [BVV84] for a formal study on the relations among the requirements of the FE variants mentioned above.

3 Solvability of fair exchange

In this section, we focus on solvability of the FE problem, thus focusing on questions such as: in which network settings can fair exchange be achieved? How many malicious parties can subvert a fair exchange protocol among n parties? etc. We start with a general introduction to security protocols, and then consider the solvability of FE in synchronous and asynchronous settings. No trusted entity is assumed in studying solvability in Sections 3.2 and 3.3.

3.1 Security protocols

A distributed system consists of a finite number of participants that interact by some communication primitives, such as sending and receiving messages, accessing a shared memory, etc. Below, we assume communications take place solely through sending and receiving messages over communication channels. A collection of communication channels is called a communication network. We write

$$A \to B : m,$$

when participant A submits message m to the communication network, with the intention that it should be delivered to participant B. A *synchronous* channel guarantees to deliver messages in a timely manner, with a pre-known time bound, while *asynchronous* channels deliver messages eventually, but no time bounds are put on them.

Channels may in general lose, duplicate or distort messages. Unless explicitly stated, we do not consider such faulty channels. [4]

A *protocol* assigns an algorithm to each participant. The algorithms may use the communication primitives that are available to the participants to achieve a certain common goal. A synchronous protocol assumes that the participants execute their algorithms in lock-step, i.e. there is a common clock available to all the participants, and that the communication channels are synchronous. Asynchronous protocols do however not assume these properties. A *fault tolerant* protocol achieves its goal even if some of the participants are faulty.

Different failure models are used in distributed systems to characterize how a faulty participant may misbehave. One of the simplest models is *crash* failure, in which the failed participant simply dies, i.e. ceases to act afterwards. In the *Byzantine* failure model [LSP82], a faulty participant may deviate from the algorithm assigned to it in any fashion, but its view is local and its effects are local, i.e. it only sees what is passed to it by its neighbors, and it can only send messages to its neighbor participants.

Cryptographic or *security* protocols are fault tolerant protocols which use cryptography to attain their goals. In computer security, the Dolev-Yao model [DY81, DY83], denoted $\mathcal{DY}$, is usually considered as the hostile environment model. [5]. In this model, there is one malicious participant (called attacker, intruder, saboteur, etc.), comprising all the outsider and insider corrupted parties, which has control over the entire communication network. [6] It intercepts all messages that have been transmitted and can store them in its knowledge set. It can also remove or delay messages in favor of others being communicated. “[It] is a legitimate user of the network, and thus in particular can initiate a conversation with other users” [DY83]. Security protocols are typically designed to protect the interests of the honest participants, i.e. those who faithfully follow the protocol, in presence of the $\mathcal{DY}$ attacker. Honest participants only follow the protocol, and, in general, are not required to take any steps to detect or thwart attacks. Protocols must thus be designed to guarantee that, if a participant follows the rules of the protocol, then his interests are protected.

The $\mathcal{DY}$ attacker can be seen as a Byzantine participant which is sitting in the center of a star-like network topology. All other participants therefore communicate through $\mathcal{DY}$, hence the network being of connectivity 1. [7] Network connectivity indeed plays a role in the possibility of distributed tasks, performed in presence of malicious parties, see [FLM86, Syv97].

3.2 Solvability of fair exchange in synchronous systems

Even and Yacobi [EY80], and independently Rabin [Rab81], studied simple variants of the FE problem. In [EY80], a notion of mutual signature on a message (the CS problem) is studied. They informally reason that “if the judicator is not active during the ordinary operation of the system”, then no two-party protocol can achieve *agreement*, where agreement means that when a party can compute the signature, the other one can also do so. Their argument goes as: “Assume that, after n communications, [Alice] has sufficient information for efficient calculation of [the mutual signature], but that this is not true for $n-1$ communications. We conclude that [Bob] transmits the n^{th} communication, and therefore the first time [Bob] has sufficient information is after n' communications, where $n' \neq n$. This contradicts [the definition of agreement]”.

Rabin considers the similar problem of simultaneous exchange of secrets between two non-trusting entities Alice and Bob (the ES problem). He deduces that the problem is unsolvable: “Any [exchange] protocol must have the form: Alice gives to Bob some information I_1, Bob gives to Alice J_1, Alice gives to Bob I_2, etc. There must exist a first k such that, say, Bob can determine [Alice's secret] from $I_1, \ldots, I_k$, while Alice still cannot determine [Bob's secret] from $J_1, \ldots, J_{k-1}$. Bob can withhold J_k from Alice and thus obtain [Alice's secret] without revealing [his own secret]".

Since these problems are instances of FE, their unsolvability implies unsolvability of FE in the corresponding models. Both these arguments clearly stress on the malicious act of withholding the last message. They can thus be summarized as: No two-party protocol with one Byzantine participant, even with synchronous communication channels, can solve FE. This result naturally carries over to asynchronous protocols. We remark that a crucial feature of this model is that no party is *trusted* by other participant(es). A participant is trusted iff it is publicly known that the participant is (and remains) non-faulty. DeMilo, Lynch and Merritt formalized the impossibility arguments mentioned above in [DLM82].

In [BOGW88] and, independently, in [CCD88], the authors derive general solvability results regarding the secure multi-party computation (SMPC) problem, in complete graph topologies (where every two nodes are connected). SMPC and FE, albeit being different problems, are tightly related. These results are therefore pertinent to our discussion. In [BOGW88] it is established that, in a fully connected network of synchronous channels, n-party SMPC, and thus FE, is achievable if there are at most t Byzantine participants, with $t < \frac{n}{3}$. They also prove that there exist SMPC problems which, with $t \geq \frac{n}{3}$ Byzantine participants, are unsolvable for n parties. The results of [EY80, Rab81] clearly show that FE is one of these problems. See [GL02, Mau06] for excellent reviews on further developments in SMPC.

We note that the possibility results of [BOGW88, CCD88] do *not* imply the solvability of FE in the $\mathcal{DY}$ model, simply because the connectivity of the network is 1 in the $\mathcal{DY}$ model, while these results are stated in complete graph topologies. In fact, reaching *distributed consensus*, a problem conceptually similar to FE, is impossible if the network connectivity is less than $2t + 1$, with t Byzantine participants [FLM86]. For a formal comparison between FE and distributed consensus in various models see [OT08].

3.3 Solvability of fair exchange in asynchronous systems

In asynchronous systems, the impossibility result of [FLP85] and its extension [MW87] imply that multi-party FE is unsolvable when at least one of the participants is subject to crash failure. For two-party exchanges, this result has been derived in [PG99] by reducing the distributed consensus problem to FE. It is worth mentioning that the impossibility results of [EY80, Rab81, DLM82] are based on the malicious act of withholding parts of information, whereas [PG99] prove impossibility of FE in the presence of benign, but not "malicious", failures, as a result of lack of knowledge to decide termination in asynchronous systems. These concern orthogonal difficulties in solving FE, and none of them directly implies the other one.

Up until now, we focused on the effects of participant failures, as opposed to channel failures, on solving the FE problem. Below, we consider the case of lossy channels, while assuming that participants are all honest (i.e. they faithfully follow their protocol). In distributed computing, the limitations on reaching agreement in the presence of lossy channels is usually described using the *generals paradox* [Gra78]: "There are two generals on campaign. They have an objective (a hill) that they want to capture. If they simultaneously march on the objective they are assured of success. If only one marches, he will be annihilated. The generals are encamped only a short distance apart, but due to technical difficulties, they can communicate only via runners. These messengers have a flaw, every time they venture out of camp they stand some chance of getting lost (they are not very smart.) The problem is to find some protocol that allows the generals to march together even though some messengers get lost."

Gray informally argues that such a protocol does not exist [Gra78]. This has later on been formally proved in, e.g., [YC79, HM84]. The generals problem can be reduced to two-party FE by noticing that the generals can use a fair exchange protocol to agree on a time for attack. The impossibility result stated above, thus, implies that FE is unsolvable in the presence of channel failures, when participants are honest.

Furthermore in the presence of channel failures, "any protocol that guarantees that whenever either party attacks the other party will *eventually* attack, is a protocol in which necessarily neither party attacks" [HM84]. This result implies that, in optimistic FE protocols, *resilient* channels are unavoidable even when all participants are honest. A channels is resilient iff any message inserted into one end of the channel is eventually delivered to the other end.

4 Fair exchange in the Dolev-Yao model

Fair exchange cannot be achieved in the presence of the $\mathcal{DY}$ attacker if there is no trust in the system (see Section 3). Many fair exchange protocols thus assume the presence of a trusted third party (TTP). The TTP is further assumed to be connected to protocol participants through resilient channels. We come back to to this topic shortly. There are three general constructions for FE, based on the degree of the involvement of trusted third parties. The first group needs no TTPs, e.g. the protocols of [Blu81, Rab81, EGL85, Cle90, BOGMR90, MR99]. See also [FGY92] for a chronological survey on these protocols. These are based on gradual release of information or gradual increase of privileges and require exchanging many messages to approximate fair exchange, as deterministic asynchronous FE with no trusted parties is impossible (see Section 3). The idea behind such gradual release protocols is that a party will only have a minimal advantage if he decides to cheat.

Protocols of the second group need the TTP's intervention in each exchange, e.g. see [BT94, CTS95, ZG96a, ZG96b, DGLW96, FR97, AG02]. In the literature, these are sometimes called protocols with *in-line* or *on-line* TTPs. On-line TTPs, although being involved in each exchange, act only as a light-weight notary, as opposed to to in-line TTPs which directly handle the items subject to exchange, cf. [ZG96c]. The protocols of the second group have a fixed, usually small, number of message exchanges, and are thus more appealing in practice. However, the TTP can easily become a communication bottleneck or a single target of attacks, as it is involved in each exchange. Protocols of [Rab83, RS98] can also be listed in the second group as they require the TTP to be active during each exchange. However, a slight difference is that, intuitively, the TTPs in the latter protocols need not "be aware" of being involved in such exchanges. For instance, the TTP in Rabin's protocol acts as a beacon, broadcasting signals which can be used by others to perform fair exchange.

The third group of FE protocols, known as *optimistic* protocols, require the TTP's intervention only if failures, accidentally or maliciously, occur, e.g. see [Eve83, ZG97a, ASW97, Mic97, ASW98a, ASW98b, BDM98, ZDB99, MK01, Mic03, PCS03, DR03, CCT05, CTM07]. Therefore, honest parties that are willing to exchange their items can do so without involving any TTP. Optimistic protocols are called protocols with *off-line* TTPs since the TTP need not be active at the time the exchange goes on; the

TTP can be contacted in a later time.

4.1 Optimistic fair exchange

We focus on asynchronous two-party optimistic exchange protocols. The $\mathcal{DY}$ model is assumed for the attacker. The exchange partners A and B are connected via $\mathcal{DY}$. There is a trusted third party T, which is immune to failures. The TTP is connected to A and B via resilient channels.

4.1.1 The resilient channel assumption

A channel between two participants is resilient iff any message inserted into one end of the channel is eventually delivered to the other end. The resilience assumption is an asymptotic restriction, i.e. messages are delivered eventually, but no bounds are placed on the order or the time of delivering these messages.

As mentioned after the generals paradox in Section 3.3, in order to achieve timeliness, fair exchange protocols need resilient channels. Intuitively, unilaterally terminating the exchange by a participant corresponds to marching to the hill by a general. A general may safely march to the hill only if he knows the other general would do so. Similarly, a participant may consider the exchange terminated only if she knows the other participant would also do so.

In the $\mathcal{DY}$ model, however, the communication media are assumed to be under complete control of the intruder. The $\mathcal{DY}$ intruder can in particular destroy transmitted messages. For liveness properties, such as timeliness, to hold in the $\mathcal{DY}$ intruder model, the assumption that the intruder does not disrupt (some of) the communication channels must therefore be added.

Resilient channels are not readily available in most practical situations. Available faulty channels can nonetheless be used to provide resilience, as described below. Assuming resilient channels in security protocol thus helps us to abstract from the underlying mechanisms which actually provide resilience.

There are various ways to construct resilient channels from faulty ones. Let us assume that A and B are connected with a faulty channel c which may lose, duplicate and reorder messages. To distinguish c from a channel that is only temporarily available, we assume that there is a bound on the number of messages that c can discard. We say that a channel is *fair lossy* iff any message which is inserted to one end of the channel an infinite number of times is delivered to the other end of the channel an infinite number of times. [8] If c is a fair lossy channel, which may duplicate and reorder messages, then retransmission and tagging allow A and B to construct a reliable FIFO channel on top of c, e.g. see Stenning's protocol [Ste76, Lyn96].

In the $\mathcal{DY}$ intruder model, it is assumed that the only possible means of communication between A and B is $\mathcal{DY}$. The $\mathcal{DY}$ intruder, however, need not be a fair lossy medium and can destroy all the messages that are transmitted through it. Therefore, no reliable channel may be constructed between A and B in the $\mathcal{DY}$ model. Nevertheless, the assumption that $\mathcal{DY}$ controls *all* communication media between A and B is often impractical. For instance, in wireless networks, given that jamming is only locally sustainable, A and B can always move to an area where they can send and receive messages. Ultimately, two principals who fail to properly establish a channel over computer networks can resort to other communication means, such as various postal services. These services, albeit being orders of magnitude slower than computer networks, are very reliable and well protected by law.

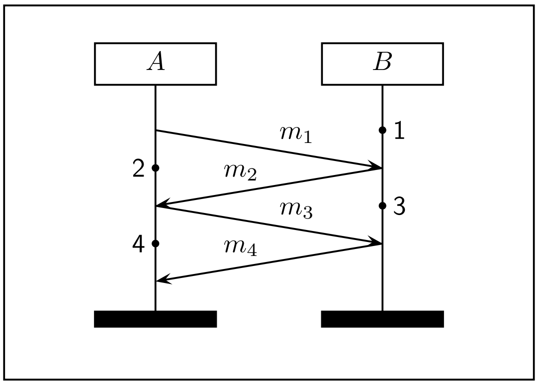

Figure 2: Generic four message protocol

We thus postulate that any A and B who are willing to communicate can eventually establish a (fair lossy) channel, despite $\mathcal{DY}$'s obstructions. To add this postulation to the $\mathcal{DY}$ model, weakening $\mathcal{DY}$ to the extent that it behaves as a fair lossy channel is adequate. This is the essence of the resilient channels assumption.

4.1.2 The structure of optimistic protocols

We opt for a high level description that underlines the exchange patterns. Exact message contents are abstracted away, and all messages are assumed to contain enough information for protocol participants to distinguish different protocol instantiations, and different roles in protocols. Detailed specification of these issues is orthogonal to our current purpose.

Optimistic protocols typically consist of three sub-protocols: the *main* or *optimistic* sub-protocol, the *abort* sub-protocol and the *recovery* sub-protocol. Figure 2 depicts a generic main sub-protocol between A and B. The regions in which the other two sub-protocols are alternative possibilities are numbered (1–4) in the figure. In the main sub-protocol, that does not involve the TTP, the agents first *commit* to release their items and then they actually release them. The items subject to exchange, and commitments are respectively denoted by i_A, i_B and $c_A(i_A), c_B(i_B)$. In Figure 2 we have $m_1 = c_A(i_A)$, $m_2 = c_B(i_B)$, $m_3 = i_A$ and $m_4 = i_B$. If no failures occur, the participants exchange their items successfully using the main sub-protocol.

If an expected message does not arrive in time, or the arrived message does not conform to the protocol, then the participant expecting that message can resort to the TTP using abort or recovery sub-protocols. These sub-protocols help the participant to reach a fair state and subsequently terminate. Here we introduce the notion of *resolve patterns*, which is useful in describing optimistic FE protocols. Consider again the generic four message protocol shown in Figure 2. A resolve pattern determines which of the abort and resolve sub-protocols are available to participants when they are waiting for a message from their opponent in the main sub-protocol; namely, the alternative sub-protocols envisaged for points 1, 2, 3 and 4 in Figure 2.

Four different symbols can be assigned to a point in the resolve pattern: *abort* (a), *recovery* (r), *quit* (q), and *none* ($-$). Intuitively, occurrence of the symbol a means that at that point the participant can initiate an abort sub-protocol, thereby requesting the TTP to abort the exchange protocol. Likewise, occurrence of an r means that the participant can initiate a recovery sub-protocol, thereby requesting the TTP to help him recover from a situation in which it has received a commitment from the other participant without having received the other participant's item. Occurrence of a q means that in case the expected message does not arrive in time, the participant can safely quit the exchange without contacting the TTP. Naturally, if no message has been exchanged, the participant quits the protocol, e.g. B in Figure 2 quits the exchange if he does not receive the first message in time. A 'none' option ($-$) indicates that the participant has no alternatives but following the optimistic protocol. It turns out that 'none' options undermine termination of asynchronous optimistic FE protocols. This is intuitive because participants may crash and never send the message their opponent is waiting for. When communicating with the TTP (using resolve sub-protocols), however, participants know that the message they send to and expect to receive from the TTP will be delivered in a finite time. This is due to resilience of the channels, and the fact that the TTP is immune to failures (see TTP assumptions, above).

We use tuples for representing resolve patterns. For instance, a resolve pattern for the protocol of Figure 2 can be $(\mathsf{q}, \mathsf{a}, \mathsf{r}, \mathsf{r})$, listing the symbols attached to points 1, 2, 3 and 4, respectively.

The resolve sub-protocols (abort/recovery) involve the TTP. In order to simplify the reasoning we assume that the participant sends its message history (all messages sent and received up to now by the participant in the current execution of the protocol) to the TTP, and based on these the TTP either returns an *abort token* A, or a *recovery token* R. Token A often has no intrinsic value; it merely indicates that the TTP will never send an R token in the context of the current exchange. Token R should however help a participant to recover to a fair state. Although it is impossible for B alone to derive the item i_A from the commitment $c_A(i_A)$ (and similar for i_B), it is often assumed that the TTP can generate i_A from $c_A(i_A)$, and i_B from $c_B(i_B)$, and that R contains i_A and i_B. In case the TTP cannot do so, usually an affidavit from the TTP is deemed adequate, cf. *weak fairness* [PVG03]. The resilient channels guarantee that, in case of failures, protocol participants can ultimately consult the TTP.

Participant A can run the recovery protocol if the opponent B has committed to exchange, but A has not received B's item, and vice versa. A participant aborts (cancels) the exchange if she does not receive the opponent's commitment to the exchange.

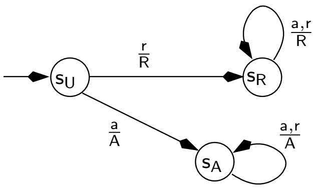

Figure 3: Abstract Mealy machine of TTP

The TTP logic matching the resolve pattern $(\mathsf{q}, \mathsf{a}, \mathsf{r}, \mathsf{r})$ for the protocol of Figure 2 is shown in Figure 3. For each exchanged item, the finite state (Mealy) machine of the TTP is initially in the *undisputed* state s_U. If the TTP receives a valid abort request (from A) while being at state s_U, then it sends back an abort token, and moves to *aborted* state s_A. Similarly, if the TTP is in state s_U, and receives a valid resolve request (from either A or B), then it sends back R, containing i_A and i_B, and moves to *recovered* state $\mathsf{s_R}$. When the TTP is in either of $\mathsf{s_A}$ or $\mathsf{s_R}$ states, no matter it receives an abort or a recovery request on this exchange, it consistently replies with A or R, respectively. [9]

In state $\mathsf{s_R}$, the TTP typically stores R (which contains i_A and i_B). This is because, if B sends a recovery request, and then A sends an abort request, the TTP needs to send back R to A. However, the TTP needs $c_B(i_B)$ to generate i_B, and thus construct R. An abort request by A nevertheless does not contain $c_B(i_B)$. Therefore, once the TTP computes R for an exchange, it stores R in its secure storage, for possible future uses. In Figure 3, a and r stand for valid abort and recovery requests, while A and R stand for the corresponding abort and recovery tokens, respectively. Remark that depending on the current state of the TTP a participant may receive an abort token A even if it sends a resolve request r to the the TTP, and vice versa.

5 A selective literature review

Below, we review some of the main ideas and results on solving the FE problem in the $\mathcal{DY}$ model. This review is selective. In particular, we do not touch upon various FE protocols that were developed to go beyond fair exchange requirements and satisfy an extended set of functional or security goals (some of these are however discussed in subsequent sections). Synchronous protocols are also mostly absent from our review. For general surveys on the topic see several Ph.D. dissertations that have been written on this topic, e.g. [Aso98, Sch00, Cha03, Kre03, Nen05, Gon05, Oni06, Tor08].

Generatable and revocable items. Pivotal to the working of optimistic protocols is the nature of the items that are subject to exchange. It has been shown in [SW02] that optimistic FE is impossible if the exchanged items are neither *generatable* nor *revocable*. In general, no such restriction applies to FE protocols with in-line or on-line TTPs though. An item is generatable if the TTP can generate the item from a participant's commitment to release that item, and an item is revocable if the TTP can revoke the validity of that item. In general, digital items are neither generatable nor revocable. However, cryptographic tools, such as verifiable encryption, can make certain digital items generatable. For instance, see [ASW98b, Che98, PCS03, DR03, Ate04, DJH07] for techniques to enable the TTP to generate participants' signatures from their commitments; see also [RR00]. In contrast, there are not many digital items that can be revoked by the TTP (see below).

The above-mentioned impossibility result of [SW02] comes as no surprise when noticing that if a wronged Bob resorts to the TTP, he wishes (at least) one of the following services: Either the TTP can generate the item that he has expected, which is impossible if the item is not generatable, or the TTP can revoke the item that he has lost (i.e. currently being in the possession of Alice), which is impossible if the item is not revocable. The TTP can however provide Bob with an affidavit declaring that Bob has indeed been cheated (by Alice). In this case, Bob only achieves *weak* fairness [Aso98], which might not satisfy Bob. Below, we explore how such affidavits can be used to provide strong fairness in CS, CEM and NR protocols.

The goal is to provide strong fairness without using costly cryptographic tools such as verifiable encrypted signatures. The idea is to exploit a freedom that is inherent to the definitions of CS, CEM and NR. In these FE variants, the protocol (designer) is free to define what constitutes, e.g., a mutually signed contract, a signed receipt, or evidence of origin. Therefore, these protocols devise dispute resolution procedures to evaluate (or interpret) the digital assets that are collected in the protocol. Dispute resolution procedures can thereby be tailored to grade affidavits from the TTP as, for instance, a valid evidence of origin. This idea has been used in many FE protocols such as [ZG97a, ASW98a, ZDB99, KMZ02, GRV05, CCT05, CTM07]. Note that these protocols enforce the structure of the exchanged items, hence being called *invasive* [ASW98a]. Non-invasive protocols are more favorable, but come at high computation costs, as they rely on heavy cryptographic tools, as in, e.g., the signature exchange protocols of [ASW98b].

A partial remedy to invasiveness is to make the TTP *invisible* [Mic97], so that there would not be any difference between the evidences collected in optimistic runs and those issued by the TTP. Note that the structure of the evidences is still determined by the protocol, hence the result may be an invasive protocol (e.g. as in [Mic03]). However, the exchanged items would not reveal whether the TTP was involved in the exchange or not. For protocols with invisible, or *transparent*, TTPs see, e.g., [Mic97, ASW98a, MK01, MS02, Mic03, Ate04]. As is phrased by Asokan, "typically, non-invasiveness implies invisibility of third party" [Aso98].

Now we turn to fair exchange of revocable items. Generally, it is hard to revoke digital items. However, certain payment systems can in principle provide revocable coins, e.g. see [JY96, Vog03]. Fair payment protocols which employ revocable money (orders) are presented in [ASW98a, Vog03]. A separate group of protocols for exchanging revocable items exploits the freedom in the definition of CS, CEM and NR, just as mentioned earlier. These not only prescribe a tailored dispute resolution procedure to grade the TTP's messages as valuable evidences, but they also require the TTP to in some situations participate in the dispute resolution phase of the protocol in order to revoke evidences collected by the participants. Examples of protocols following this idea are [Eve83, FPH00, FPH02, MD02, Zho04, WBZ04, FPH04]. These protocols require three messages in their exchange sub-protocols, compared to optimistic protocols for generatable items that require four messages. It has been shown in [Sch00] that three messages is the minimum number of messages in exchange sub-protocols, given that the TTP is allowed to participate in the dispute resolution phase, while this number is four if the TTP is not allowed to do so. Requiring the TTP's intervention in the evidence verification phase can be a drawback for these protocols because evidences carry no weight until the TTP declares that they have not been revoked. [10]

Idempotent and non-idempotent items. Most FE protocols assume that the items subject to exchange are *idempotent* [Aso98], meaning that receiving or possessing an item once is the same as receiving it multiple times. For example, once Alice gets access to Bob's signature on a contract, receiving it again some time later does not add anything to Alice's knowledge. The idempotency assumption reflects the mass reproducibility of digital items. However, there exist protocols for exchanging digital non-idempotent items. Electronic vouchers [FKT$^+$99, FE03] are prominent examples of non-idempotent items. Depending on the implementation, right tokens in digital rights management systems can as well be considered as digital non-idempotent items, e.g. see [CIK$^+$06, TKJ08]. The current approach to securely use non-idempotent items is to limit their distribution to trusted computing devices, which are nowadays becoming more prevalent. Protocols for handling non-idempotent items, being FE protocols or not, usually require that items are neither created nor destroyed in the course of the protocol, e.g. see [FKT$^+$99, TIHF04]. This resembles the *money atomicity* property in electronic commerce, stating that money is neither destroyed nor generated in exchanges [Tyg96].

Using trusted devices in FE is not limited to exchanging non-idempotent items. These are used for exchanging idempotent items as well, mainly in order to increase protocols' efficiency or flexibility. Examples are [Tor09] to reduce the number of messages to three in the optimistic sub-protocol, [VPG01] for exchanging time-sensitive items, and [TMH06] for optimistic exchange of non-revocable, non-generatable items; recall that optimistic FE requires that at least one of the items be either revocable or generatable [SW02]. [11] See also [AV04, AGGV05, ES05, FFD$^+$06, GR06] on using trusted devices in FE.

Bounds on the number of messages. The premise of optimistic FE is that failures are infrequent, and consequently fallback sub-protocols are executed rarely. Therefore, a meaningful measure of efficiency in these protocols is the number of messages exchanged in the main optimistic sub-protocol. Several results regarding optimal efficiency of asynchronous two-party optimistic CS and CEM protocols have been derived in [PSW98, Sch00]. The main results regarding the optimal number of messages in exchange sub-protocols are mentioned above, namely three messages when the TTP is allowed to intervene in the dispute resolution phase, and four messages otherwise. Therefore, protocols which require only three messages in the exchange sub-protocol and do not rely on TTP's intervention in the dispute resolution phase do not satisfy the requirements of fair exchange. For instance, the protocols of [Mic03], with three messages in the main sub-protocol, do not provide timeliness.

Fair exchange between trusted devices requires three messages in the optimistic protocol when the items subject to exchange are idempotent [Tor09]. For exchanging non-idempotent items three messages in the optimistic sub-protocol are sufficient, given that the trusted devices have access to an unlimited secure storage; otherwise, four messages in the optimistic sub-protocol are sufficient and necessary [Tor09]. These results are summarized in Table 1. Note that exchanging non-idempotent among non-trusted devices is inherently insecure.

Items Computing devices	non-idempotent	idempotent
non-trusted	–	4
trusted (unlimited secure storage)	3	3
trusted (limited storage)	4	3

Table 1: Optimal number of messages in two-party optimistic sub-protocol

It is shown [PSW98, Sch00] that the TTP needs to be stateful, i.e. to keep states of disputed exchanges, to guarantee fairness in asynchronous optimistic protocols. From a practical point of view, this result is of great relevance. Optimistic FE not only requires TTPs for recovering from unfair transient states, it needs TTPs which maintain persistent databases, containing the states of disputed exchanges, for virtually an indefinite amount of time. Naturally, in long runs, TTPs may crash or be compromised. [12] Mechanisms to limit the damages of these defects are described below. Before that, we remark that the optimistic protocols with stateless TTPs are either unfair, such as [Mic03, Ate04, NZB04] which do not provide timeliness [13], or rely on synchronous communication channels, such as [ES05].

Accountability and robustness of the trustee. Malicious TTPs can inevitably subvert an exchange protocol; this is indeed the definition of trusted parties [And01]. To demotivate malicious TTPs from cheating on protocol participants, Asokan introduces protocols in which TTPs are *verifiable* [Aso98]. Given that corrupted TTPs do not simply disappear, a protocol with verifiable TTP allows wronged participants to prove TTP's misbehavior to an external court. Accountability is thus a prohibition mechanism, relying on the assumption that a TTP prefers not being detected as malicious. This is a tenable assumption because external courts might be able to impose financial penalties on the TTP, the TTP may be concerned about its reputation, etc. Verifiability and transparency of TTPs are however not mutually attainable as is noted by Asokan, e.g. see [GJM99] for a concrete protocol where these two requirements clash.

To reduce the dependency of protocols on availability and sanity of a single trusted party, distributed TTPs can be used. In [AdG01] parts of the TTP's work are delegated to intermediary semi-trusted agents to reduce the TTP's burden, and in [SXL05, RRN05] secret sharing schemes are used so that, to subvert the protocol, an attacker needs to compromise several TTPs. Note that distributed TTPs in general need to run some atomic commit protocol to ensure the consistency of their (distributed) state. We recall that (1) attaining fairness in optimistic asynchronous protocols is impossible without stateful TTPs [Sch00], and (2) atomic commit protocols are nearly as expensive as fair exchange [Tyg96, Tan96, LNJ01, AFG$^+$04, RRN05]. Related to distributed TTPs, Ito, Iwaihara and Kambayashi in [IIK02] assume that participants have limited trust in TTPs and propose algorithms to determine if a rational agent would engage in an exchange using cascades of TTPs.

In the context of fair exchange protocols in which the TTP is involved in every exchange (online TTPs), Franklin and Reiter use a secret sharing scheme to limit what a TTP can learn about exchanged materials [FR97]. They assume that the TTP does not collude with any of the participants, but has its own interests in the matter.

Weaker notions of fairness. There are several alternatives to FE which do not need TTPs at all, but can provide only a weak notion of fairness.

The concept of *rational exchange* of Syverson [Syv98] seeks to achieve fairness, with no TTPs, assuming that the parties are rational, i.e. they try to maximize their benefits. This assumption is in contrast to the pessimistic view prevalent in the security community that honest parties should be protected even from self-damaging attackers. The idea is "not to enforce compliance with the protocol, but to remove incentives to cheat", cf. [Jak95]. A few scenarios in which rational exchange can be of practical use are mentioned in [Syv98]. See also the taxi-driver example of Section 1.

Game theory can provide valuable insights into the properties of exchange protocols, when assuming that their participants are rational agents, rather than categorizing them as malicious and honest parties, who blindly act regardless of their interests. For more on this approach see [San97, SW02, IIK02, CMSS05, BHC04, IZS05, ADGH06, TW07].

Concurrent signatures proposed in [CKP04], and further investigated in [SMZ04, WBZ06, TSS06], provide a weak alternative to fair exchange. These generally do not require any TTP interventions. The idea is that Alice and Bob produce two ambiguous signatures which become bound to their corresponding signers only when a *keystone* is released by Alice. The main shortcoming of the construct is that Bob has no control over the termination of the protocol, and, moreover, Alice can secretly show Bob's signature to other parties before publishing the keystone; cf. the notion of *abusefree* protocols [GJM99]. A few scenarios in which this level of fairness is adequate are mentioned in [CKP04].

Multiparty protocols. Multiparty fair exchange protocols are notoriously hard to design and analysis. The early protocols, such as [Aso98, GM99], have mostly been found flawed [CKS04, MR08a]. Mukhamedov and Ryan introduce a class of startling attacks on multiparty CS, dubbed *abort chaining* attacks, which truly demonstrate the subtlety of these protocols.

Minimal number of messages, although known for two party FE protocols [Sch00, Tor09], in multi-party cases are under study. Building upon the idea of abort chaining attacks it is shown by [MRT09] that an n-party asynchronous fair contract protocol requires at least n^2+1 messages in the optimistic sub-protocol. This result is obtained by connecting the multiparty FE problem to the combinatorial problem of finding shortest permutation sequences [Adl74].

Designing optimistic fair exchange protocols. To conclude this chapter, we point out some of the resources which can be of use when designing FE protocols. Many of the prudent advice [AN96] and attack scenarios known for authentication and key distribution protocols [Car94, CJ97] are pertinent to FE protocols as well. Papers specifically focusing on FE are unfortunately scarce.

We note that compilations of FE protocols are almost non-existent, [KMZ02] being a notable exception. New protocols are constantly devised with subtle differences between their assumptions, methods and goals, thus making it difficult to oversee general techniques. As of design methodologies, [Aso98, PVG03] discuss constructing generic FE protocols and [GRV05] provides templates for conservative NR protocols. The collections of attacks on NR and CEM protocols, presented respectively in [Lou00] and [SWZ06], give designers an opportunity to assess their new protocols against known attacks. These are however not well classified, and in particular flaws stemming in the interaction between protocols and cryptographic apparatus used in them are mostly omitted; see [DR03] for an example of such attacks on FE protocols.

There has been a considerable amount of work on formal verification of fair exchange protocols. See, for instance, the dissertations [Kre03, Tor08, Car09], and also [SM02, GR03, KK05, KKW05, BP06, WH07, KKT07]. However, we are not aware of any comprehensive guide or survey on existing formal techniques for verifying FE protocols. This would be desirable for practitioners.

Acknowledgements

The authors are grateful to Bruno Conchinha Montalto for proofreading the article. Mohammad Torabi Dashti has been supported by the FP7-ICT-2007-1 Project no. 216471, “AVANTSSAR: Automated Validation of Trust and Security of Service-oriented Architectures”.

Notes

[1] This of course does not imply that Alice would give the other half to Bob. Alice can choose not to pay, as it will not change her profit vs. loss balance. Remark that if Alice can use the other half of the bill to convince another taxi driver in a similar scenario later, indeed Alice would benefit from not giving the other half to Bob. To avoid such situations, Bob must ensure that Alice rips a bill into two halves afresh.

[2] Fair and private contract *negotiation* protocols are discussed in, e.g., [FA05].

[3] Fairness, as defined in this article, is a safety property, while timeliness is a liveness property. Intuitively, a safety properties states that something bad does not happen, while a liveness property stipulates that something good will happen [AS85].

[4] As is described later, the attacker is modelled as a participant that can inject messages into the channels and remove messages from (some of) the channels, even if the channels are assumed to be non-faulty.

[5] For a critique on the $\mathcal{DY}$ model in face of the emerging mobile ad-hoc protocols see, e.g., [Gli07]

[6] Any number of $\mathcal{DY}$ attackers can be modelled as a single $\mathcal{DY}$ attacker by merging their knowledge sets [SM00].

[7] A network has connectivity c iff at least c nodes need to be removed to disconnect the network. In the $\mathcal{DY}$ model removing the attacker node would disconnect the network.

[8] This corresponds to the *strong loss limitation* condition in [Lyn96]. A weaker variant of this requirement states that if an infinite number of messages are sent to the channel, then *some* infinite subset of them are delivered.

[9] Inconsistent (but correct) logics for the TTP have been used, e.g. in [MR08b], where the TTP may reply R to a correct participant, after realizing that it has previously replied A to a malicious party.

[10] Such protocols are sometimes called *non-monotonic* [Ate04].

[11] The protocol of [VPG01] does not provide timeliness, as is pointed out in [Vog03], and the protocol of [TMH06] is susceptible to a replay attack (we skip describing the attack, as it would require a detailed description of the protocol, and the attack is also rather obvious). The ideas behind these protocols can however be salvaged with some changes.

[12] The notion of compromisable trustee may seem to be paradoxical. We note that being *trusted* does not imply being *trustworthy*, e.g. see [Gol06].

[13] These protocols in fact require channels which can buffer messages for virtually an indefinite amount of time, thus merely delegating the “stateful-ness” to a different entity.

References

[AdG01] G. Ateniese, B. de Medeiros, and M. Goodrich. TRICERT: A distributed certified email scheme. In *NDSS ’01*. Internet Society, 2001.

[ADGH06] I. Abraham, D. Dolev, R. Gonen, and J. Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *PODC ’06*, pages 53–62. ACM Press, 2006.

[Adl74] L. Adleman. Short permutation strings. *Discrete Math.*, 10:197–200, 1974.

[AFG$^+$04] G. Avoine, F. Freiling, R. Guerraoui, K. Kursawe, S. Vaudenay, and M. Vukolic. Reducing fair exchange to atomic commit. Technical Report 200411, EPFL, Lausanne, Switzerland, 2004.

[AG02] M. Abadi and N. Glew. Certified email with a light on-line trusted third party: Design and implementation. In *WWW ’02*, pages 387–395. ACM Press, 2002.

[AGGV05] G. Avoine, F. Gärtner, R. Guerraoui, and M. Vukolic. Gracefully degrading fair exchange with security modules. In *EDCC ’05*, volume 3463 of *LNCS*, pages 55–71. Springer, 2005.

[AN96] M. Abadi and R. Needham. Prudent engineering practice for cryptographic protocols. *IEEE Trans. Softw. Eng.*, 22(1):6–15, 1996.

[And01] R. Anderson. *Security Engineering: A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, Inc., 2001.

[AS85] B. Alpern and F. Schneider. Defining liveness. *Inf. Process. Lett.*, 21(4):181–185, 1985.

[Aso98] N. Asokan. *Fairness in electronic commerce*. PhD thesis, University of Waterloo, Canada, 1998.

[ASW97] N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *CCS '97*, pages 8–17. ACM Press, 1997.

[ASW98a] N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *IEEE Security and Privacy '98*, pages 86–99. IEEE CS, 1998.

[ASW98b] N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures (extended abstract). In *EUROCRYPT '98*, volume 1403 of *LNCS*, pages 591–606. Springer, 1998. Extended version in IEEE Journal on Selected Areas in Communications, 18(4): 593–610, 2000.

[Ate04] G. Ateniese. Verifiable encryption of digital signatures and applications. *ACM Trans. Inf. Syst. Secur.*, 7(1):1–20, 2004.

[AV04] G. Avoine and S. Vaudenay. Fair exchange with guardian angels. In *WISA '03*, volume 2908 of *LNCS*, pages 188–202. Springer, 2004.

[BDM98] F. Bao, R. Deng, and W. Mao. Efficient and practical fair exchange protocols with off-line TTP. In *IEEE Security and Privacy '98*, pages 77–85. IEEE CS, 1998.

[BHC04] L. Buttyán, J. Hubaux, and S. Capkun. A formal model of rational exchange and its application to the analysis of Syverson's protocol. *J. Comput. Secur.*, 12(3-4):551–587, 2004.

[Blu81] M. Blum. Three applications of the oblivious transfer: Part I: Coin flipping by the telephone; part II: How to exchange secrets; part III: How to send certified electronic mail. Technical report, Dept. EECS, University of California, Berkeley, 1981.

[BOGMR90] M. Ben-Or, O. Goldreich, S. Micali, and R. Rivest. A fair protocol for signing contracts. *IEEE Trans. on Information Theory*, 36(1):40–46, 1990.

[BOGW88] M. Ben-Or, S. Goldwasser, and A. Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. In *STOC '88*, pages 1–10. ACM Press, 1988.

[BP06] G. Bella and L. Paulson. Accountability protocols: Formalized and verified. *ACM Trans. Inf. Syst. Secur.*, 9(2):138–161, 2006.

[BPW07] S. Brams, K. Pruhs, and G. Woeginger, editors. *Fair Division*. Number 07261 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum für Informatik (IBFI), Schloss Dagstuhl, Germany, Dagstuhl, Germany, 2007.

[BT94] A. Bahreman and D. Tygar. Certified electronic mail. In *NDSS '94*, pages 3–19. Internet Society, 1994.

[BVV84] M. Blum, U. Vazirani, and V. Vazirani. Reducibility among protocols. In *CRYPTO '83*, pages 137–146. Plenum pub., 1984.

[Car94] U. Carlsen. Cryptographic protocols flaws. In *CSFW '94*, pages 192–200. IEEE CS, 1994.

[Car09] R. Carbone. *LTL Model-Checking for Security Protocols*. PhD thesis, University of Genova, Italy, 2009.

[CCD88] D. Chaum, C. Crépeau, and I. Damgard. Multiparty unconditionally secure protocols. In *STOC '88*, pages 11–19. ACM Press, 1988.

[CCT05] J. Cederquist, R. Corin, and M. Torabi Dashti. On the quest for impartiality: Design and analysis of a fair non-repudiation protocol. In *ICICS '05*, volume 3783 of *LNCS*, pages 27–39. Springer, 2005.

[Cha03] R. Chadha. *A formal analysis of exchange of digital signatures*. PhD thesis, University of Pennsylvania, 2003.

[Che98] L. Chen. Efficient fair exchange with verifiable confirmation of signatures. In *ASIACRYPT '98*, volume 1514 of *LNCS*, pages 286–299. Springer, 1998.

[CIK$^+$06] C. Chong, S. Iacob, P. Koster, J. Montaner, and R. van Buuren. License transfer in OMA-DRM. In *ESORICS '06*, volume 4189 of *LNCS*, pages 81–96. Springer, 2006.

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature (version 1.0), 1997. `citeseer.ist.psu.edu/clark97survey.html`.

[CKP04] L. Chen, C. Kudla, and K. Paterson. Concurrent signatures. In *EUROCRYPT '04*, volume 3027 of *LNCS*, pages 287–305. Springer, 2004.

[CKS04] R. Chadha, S. Kremer, and A. Scedrov. Formal analysis of multi-party contract signing. In *CSFW '04*, pages 266–265. IEEE CS, 2004.

[Cle90] R. Cleve. Controlled gradual disclosure schemes for random bits and their applications. In *CRYPTO '89*, volume 435 of *LNCS*, pages 573–588. Springer, 1990.

[CMSS05] R. Chadha, J. Mitchell, A. Scedrov, and V. Shmatikov. Contract signing, optimism, and advantage. *J. Log. Algebr. Program.*, 64(2):189–218, 2005.

[CTM07] J. Cederquist, M. Torabi Dashti, and S. Mauw. A certified email protocol using key chains. In *SSNDS '07*, pages 525–530. IEEE CS, 2007.

[CTS95] B. Cox, J. Tygar, and M. Sirbu. NetBill security and transaction protocol. In *1st Usenix Workshop in Electronic Commerce*, pages 77–88. USENIX Association, 1995.

[DGLW96] R. Deng, L. Gong, A. A. Lazar, and W. Wang. Practical protocols for certified electronic mail. *J. Network Syst. Manage.*, 4(3):279–297, 1996.

[DJH07] Y. Dodis, P. Joong Lee, and D. Hyun Yum. Optimistic fair exchange in a multi-user setting. In *PKC '07*, volume 4450 of *LNCS*, pages 118–133. Springer, 2007.

[DLM82] R. DeMillo, N. Lynch, and M. Merritt. Cryptographic protocols. In *STOC '82*, pages 383–400. ACM Press, 1982.

[DR03] Y. Dodis and L. Reyzin. Breaking and repairing optimistic fair exchange from PODC 2003. In *DRM '03*, pages 47–54. ACM Press, 2003.

[DY81] D. Dolev and A. Yao. On the security of public key protocols (extended abstract). In *FOCS '81*, pages 350–357. IEEE CS, 1981.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. on Information Theory*, IT-29(2):198–208, 1983.

[EGL85] S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Commun. ACM*, 28(6):637–647, 1985.

[ES05] P. Ezhilchelvan and S. Shrivastava. A family of trusted third party based fair-exchange protocols. *IEEE Trans. Dependable Secur. Comput.*, 2(4):273–286, 2005.

[Eve83] S. Even. A protocol for signing contracts. *SIGACT News*, 15(1):34–39, 1983.

[EY80] S. Even and Y. Yacobi. Relations among public key signature systems. Technical Report 175, Computer Science Dept., Technion, Haifa, Isreal, Mar. 1980.

[FA05] K. Frikken and M. Atallah. Achieving fairness in private contract negotiation. In *FC '05*, volume 3570 of *LNCS*, pages 270–284. Springer, 2005.

[FE03] K. Fujimura and D. Eastlake. Requirements and design for voucher trading system (VTS). RFC 3506 (Informational), 2003.

[FFD$^+$06] M. Fort, F. Freiling, L. Draque Penso, Z. Benenson, and D. Kesdogan. TrustedPals: Secure multiparty computation implemented with smart cards. In *ESORICS '06*, volume 4189 of *LNCS*, pages 34–48. Springer, 2006.

[FGY92] M. Franklin, Z. Galil, and M. Yung. An overview of secure distributed computing. Technical Report TR CUCS-008-92, Dep. of Computer Science, Columbia University, Mar. 1992.

[FKT$^+$99] K. Fujimura, H. Kuno, M. Terada, K. Matsuyama, Y. Mizuno, and J. Sekine. Digital-ticket-controlled digital ticket circulation. In *Proc. the 8th USENIX Security Symposium*, pages 229 – 240. USENIX Association, 1999.

[FLM86] M. Fischer, N. Lynch, and M. Merritt. Easy impossibility proofs for distributed consensus problems. *Distrib. Comput.*, 1(1):26–39, 1986.

[FLP85] M. Fischer, N. Lynch, and M. Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985.

[FPH00] J. Ferrer-Gomila, M. Payeras-Capellà, and L. Huguet-i-Rotger. An efficient protocol for certified electronic mail. In *ISW '00*, volume 1975 of *LNCS*, pages 237–248. Springer, 2000.

[FPH02] J. Ferrer-Gomila, M. Payeras-Capellà, and L. Huguet-i-Rotger. A realistic protocol for multi-party certified electronic mail. In *ISC '02*, volume 2433 of *LNCS*, pages 210–219. Springer, 2002.

[FPH04] J. Ferrer-Gomila, M. Payeras-Capellà, and L. Huguet-i-Rotger. Optimality in asynchronous contract signing protocols. In *TrustBus '04*, volume 3184 of *LNCS*, pages 200–208. Springer, 2004.

[FR97] M. Franklin and M. Reiter. Fair exchange with a semi-trusted third party (extended abstract). In *CCS '97*, pages 1–5. ACM Press, 1997.

[GJM99] J. Garay, M. Jakobsson, and P. MacKenzie. Abuse-free optimistic contract signing. In *CRYPTO '99*, volume 1666 of *LNCS*, pages 449–466. Springer, 1999.

[GL02] S. Goldwasser and Y. Lindell. Secure computation without agreement. In *DISC '02*, volume 2508 of *LNCS*, pages 17–32. Springer, 2002.

[Gli07] V. Gligor. On the evolution of adversary models in security protocols: From the beginning to sensor networks. In *ASIACCS '07*, page 3. ACM Press, 2007.

[GM99] J. Garay and P. MacKenzie. Abuse-free multi-party contract signing. In *Proceedings of the 13th International Symposium on Distributed Computing*, pages 151–165, London, UK, 1999. Springer-Verlag.

[Gol06] D. Gollmann. Why trust is bad for security. In *STM '05*, volume 157 of *ENTCS*, pages 3–9, 2006.

[Gon05] N. González-Deleito. *Trust relationships in exchange protocols*. PhD thesis, Université Libre de Bruxelles, 2005.

[Gor05] J. Gordon. Alice and bob. In Bruce Christianson, Bruno Crispo, James A. Malcolm, and Michael Roe, editors, *Security Protocols Workshop*, volume 4631 of *Lecture Notes in Computer Science*, pages 344–345. Springer, 2005.

[GR03] S. Gürgens and C. Rudolph. Security analysis of (un-) fair non-repudiation protocols. In *FASec '02*, volume 2629 of *LNCS*, pages 97–114. Springer, 2003.

[GR06] B. Garbinato and I. Rickebusch. A topological condition for solving fair exchange in Byzantine environments. In *ICICS '06*, volume 4307 of *LNCS*, pages 30–49. Springer, 2006.

[Gra78] J. Gray. Notes on data base operating systems. In *Operating Systems, An Advanced Course*, volume 60 of *LNCS*, pages 393–481. Springer, 1978.

[GRV05] S. Gürgens, C. Rudolph, and H. Vogt. On the security of fair non-repudiation protocols. *Int. J. Inf. Sec.*, 4(4):253–262, 2005.

[HM84] J. Halpern and Y. Moses. Knowledge and common knowledge in a distributed environment. In *PODC '84*, pages 50–61. ACM Press, 1984.

[IIK02] C. Ito, M. Iwaihara, and Y. Kambayashi. Fair exchange under limited trust. In *TES '02*, volume 2444 of *LNCS*, pages 161–170. Springer, 2002.

[IZS05] K. Imamoto, J. Zhou, and K. Sakurai. An evenhanded certified email system for contract signing. In *ICICS '05*, volume 3783 of *LNCS*, pages 1–13. Springer, 2005.

[Jak95] M. Jakobsson. Ripping coins for fair exchange. In *EUROCRYPT '95*, volume 921 of *LNCS*, pages 220–230. Springer, 1995.

[JY96] M. Jakobsson and M. Yung. Revokable and versatile electronic money. In *CCS '96*, pages 76–87. ACM Press, 1996.

[KK05] D. Kähler and R. Küsters. Constraint solving for contract-signing protocols. In *CONCUR '05*, volume 3653 of *LNCS*, pages 233–247. Springer, 2005.

[KKT07] D. Kähler, R. Küsters, and T. Truderung. Infinite state AMC-model checking for cryptographic protocols. In *LICS '07*, pages 181–192. IEEE CS, 2007.

[KKW05] D. Kähler, R. Küsters, and T. Wilke. Deciding properties of contract-signing protocols. In *STACS '05*, volume 3404 of *LNCS*, pages 158–169. Springer, 2005.

[KMZ02] S. Kremer, O. Markowitch, and J. Zhou. An intensive survey of non-repudiation protocols. *Computer Communications*, 25(17):1606–1621, 2002.

[Kre03] S. Kremer. *Formal Analysis of Optimistic Fair Exchange Protocols*. PhD thesis, Université Libre de Bruxelles, 2003.

[LNJ01] P. Liu, P. Ning, and S. Jajodia. Avoiding loss of fairness owing to failures in fair data exchange systems. *Decision Support Systems*, 31(3):337–350, 2001.

[Lou00] P. Louridas. Some guidelines for non-repudiation protocols. *SIGCOMM Comput. Commun. Rev.*, 30(5):29–38, 2000.

[LSP82] L. Lamport, R. Shostak, and M. Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.

[Lyn96] N. Lynch. *Distributed Algorithms*. Morgan Kaufmann Publishers, 1996.

[Mau06] U. Maurer. Secure multi-party computation made simple. *Discrete Applied Mathematics*, 154(2):370–381, 2006.

[MD02] J. Monteiro and R. Dahab. An attack on a protocol for certified delivery. In *ISC '02*, volume 2433 of *LNCS*, pages 428–436. Springer, 2002.

[Mic97] S. Micali. Certified email with invisible post offices, 1997. Presented at RSA Security Conference.

[Mic03] S. Micali. Simple and fast optimistic protocols for fair electronic exchange. In *PODC '03*, pages 12–19. ACM Press, 2003.

[MK01] O. Markowitch and S. Kremer. An optimistic non-repudiation protocol with transparent trusted third party. In *ISC '01*, volume 2200 of *LNCS*, pages 363–378. Springer, 2001.

[MR99] O. Markowitch and Y. Roggeman. Probabilistic non-repudiation without trusted third party. In *Proc. 2nd Workshop on Security in Communication Network*, 1999.

[MR08a] A. Mukhamedov and M. Ryan. Fair multi-party contract signing using private contract signatures. *Inf. Comput.*, 206(2-4):272–290, 2008.

[MR08b] Aybek Mukhamedov and Mark Dermot Ryan. Fair multi-party contract signing using private contract signatures. *Inf. Comput.*, 206(2-4):272–290, 2008.

[MRT09] S. Mauw, S. Radomirovic, and M. Torabi Dashti. Minimal message complexity of asynchronous multi-party contract signing. In *CSF '09*, pages 13–25. IEEE CS, 2009.

[MS02] O. Markowitch and S. Saeednia. Optimistic fair-exchange with transparent signature recovery. In *FC '01*, volume 2339 of *LNCS*, pages 339 – 350. Springer, 2002.

[MW87] S. Moran and Y. Wolfstahl. Extended impossibility results for asynchronous complete networks. *Inf. Process. Lett.*, 26(3):145–151, 1987.

[Nen05] A. Nenadić. *A security solution for fair exchange and non-repudiation in e-commerce*. PhD thesis, University of Manchester, 2005.

[NZB04] A. Nenadić, N. Zhang, and S. Barton. Fair certified email delivery. In *SAC '04*, pages 391–396. ACM Press, 2004.

[Oni06] J. Onieva. *Multi-party Non-repudiation Protocols and Applications*. PhD thesis, University of Malaga, Spain, 2006.

[OT08] S. Orzan and M. Torabi Dashti. Fair exchange is incomparable to consensus. In John S. Fitzgerald, Anne Elisabeth Haxthausen, and Hüsnü Yenigün, editors, *ICTAC '08*, volume 5160 of *LNCS*, pages 349–363. Springer, 2008.

[PCS03] J. Park, E. Chong, and H. Siegel. Constructing fair-exchange protocols for e-commerce via distributed computation of RSA signatures. In *PODC '03*, pages 172–181. ACM Press, 2003.

[PG99] H. Pagnia and F. Gärtner. On the impossibility of fair exchange without a trusted third party. Technical Report TUD-BS-1999-02, Department of Computer Science, Darmstadt University of Technology, Darmstadt, Germany, March 1999.

[PSW98] B. Pfitzmann, M. Schunter, and M. Waidner. Optimal efficiency of optimistic contract signing. In *PODC '98*, pages 113–122. ACM Press, 1998. Extended version as technical report RZ 2994 (#93040), IBM Zürich Research Lab, Feb. 1998.

[PVG03] H. Pagnia, H. Vogt, and F. Gärtner. Fair exchange. *The Computer Journal*, 46(1):55–7, 2003.

[Rab81] M. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken computation Lab, Harvard University, May 1981.

[Rab83] M. Rabin. Transaction protection by beacons. *Journal of Computer and System Sciences*, 27(2):256–267, 1983.

[RR00] I. Ray and I. Ray. An optimistic fair exchange e-commerce protocol with automated dispute resolution. In *EC-WEB '00*, volume 1875 of *LNCS*, pages 84–93. Springer, 2000.

[RRN05] I. Ray, I. Ray, and N. Natarajan. An anonymous and failure resilient fair-exchange e-commerce protocol. *Decision Support Systems*, 39(3):267–292, 2005.

[RS98] J. Riordan and B. Schneier. A certified email protocol. In *ACSAC '98*, pages 347–352. IEEE CS, 1998.

[San97] T. Sandholm. Unenforced e-commerce transactions. *IEEE Internet Computing*, 1(6):47–54, 1997.

[Sch00] M. Schunter. *Optimistic fair exchange*. PhD thesis, Universität des Saarlandese, 2000.

[SM00] P. Syverson and C. Meadows. Dolev-Yao is no better than Machiavelli. In *WITS '00*, pages 87–92, 2000.

[SM02] V. Shmatikov and J. Mitchell. Finite-state analysis of two contract signing protocols. *Theor. Comput. Sci.*, 283(2):419–450, 2002.

[SMZ04] W. Susilo, Y. Mu, and F. Zhang. Perfect concurrent signature schemes. In *ICICS '04*, volume 3269 of *LNCS*, pages 14–26. Springer, 2004.

[Ste76] N. Stenning. A data transfer protocol. *Computer Networks*, 1(2):99–110, 1976.

[SW02] T. Sandholm and X. Wang. (Im)possibility of safe exchange mechanism design. In *8th national conference on artificial intelligence*, pages 338–344. AAAI, 2002.

[SWZ06] M. Shao, G. Wang, and J. Zhou. Some common attacks against certified email protocols and the countermeasures. *Computer Communications*, 29(15):2759–2769, 2006.

[SXL05] M. Srivatsa, L. Xiong, and L. Liu. ExchangeGuard: A distributed protocol for electronic fair-exchange. In *IPDPS '05*, page 105b. IEEE CS, 2005.

[Syv97] P. Syverson. A different look at secure distributed computation. In *CSFW '97*, pages 109–115. IEEE CS, 1997.

[Syv98] P. Syverson. Weakly secret bit commitment: Applications to lotteries and fair exchange. In *CSFW '98*, pages 2–13. IEEE CS, 1998.

[Tan96] L. Tang. Verifiable transaction atomicity for electronic payment protocols. In *ICDCS '96*, pages 261–269. IEEE CS, 1996.

[TIHF04] M. Terada, M. Iguchi, M. Hanadate, and K. Fujimura. An optimistic fair exchange protocol for trading electronic rights. In *CARDIS '04*, pages 255–270. Kluwer, 2004.

[TKJ08] M. Torabi Dashti, S. Krishnan Nair, and H. Jonker. Nuovo DRM Paradiso: Designing a secure, verified, fair exchange DRM scheme. *Fundam. Inform.*, 89(4):393–417, 2008.

[TMH06] M. Terada, K. Mori, and S. Hongo. An optimistic NBAC-based fair exchange method for arbitrary items. In *CARDIS '06*, volume 3928 of *LNCS*, pages 105–118. Springer, 2006.

[Tor08] M. Torabi Dashti. *Keeping Fairness Alive: Design and formal verification of fair exchange protocols*. PhD thesis, Vrije Universiteit Amsterdam, 2008.

[Tor09] M. Torabi Dashti. Optimistic fair exchange using trusted devices. In *SSS '09*, volume 5873 of *LNCS*, pages 711–725. Springer, 2009.

[TSS06] D. Tonien, W. Susilo, and R. Safavi-Naini. Multi-party concurrent signatures. In *ISC '06*, volume 4176 of *LNCS*, pages 131–145. Springer, 2006.

[TW07] M. Torabi Dashti and Y. Wang. Risk balance in exchange protocols. In *Asian '07*, volume 4846 of *LNCS*, pages 70–77. Springer, 2007.

[Tyg96] J. Tygar. Atomicity in electronic commerce. In *PODC '96*, pages 8–26. ACM Press, 1996.

[Vog03] H. Vogt. Asynchronous optimistic fair exchange based on revocable items. In *FC '03*, volume 2742 of *LNCS*, pages 208–222. Springer, 2003.

[VPG01] H. Vogt, H. Pagnia, and F. Gärtner. Using smart cards for fair exchange. In *WELCOM '01*, volume 2232 of *LNCS*, pages 101–113. Springer, 2001.

[WBZ04] G. Wang, F. Bao, and J. Zhou. On the security of a certified email scheme. In *INDOCRYPT '04*, volume 3348 of *LNCS*, pages 48–60. Springer, 2004.

[WBZ06] G. Wang, F. Bao, and J. Zhou. The fairness of perfect concurrent signatures. In *ICICS '06*, volume 4307 of *LNCS*, pages 435–451. Springer, 2006.

[WH07] K. Wei and J. Heather. A theorem-proving approach to verification of fair non-repudiation protocols. In *FAST '06*, volume 4691 of *LNCS*, pages 202–219. Springer, 2007.

[YC79] Y. Yemini and D. Cohen. Some issues in distributed processes communication. In *Proc. of the 1st International Conf. on Distributed Computing Systems*, pages 199–203, 1979.

[ZDB99] J. Zhou, R. Deng, and F. Bao. Evolution of fair non-repudiation with TTP. In *ACISP '99*, volume 1587 of *LNCS*, pages 258–269. Springer, 1999.

[ZG96a] J. Zhou and D. Gollmann. Certified electronic mail. In *ESORICS '96*, volume 1146 of *LNCS*, pages 160–171. Springer, 1996.

[ZG96b] J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *IEEE Security and Privacy '96*, pages 55–61. IEEE CS, 1996.

[ZG96c] J. Zhou and D. Gollmann. Observations on non-repudiation. In *ASIACRYPT '96*, volume 1163 of *LNCS*, pages 133–144. Springer, 1996.

[ZG97a] J. Zhou and D. Gollmann. An efficient non-repudiation protocol. In *CSFW '97*, pages 126–132. IEEE CS, 1997.

[ZG97b] J. Zhou and D. Gollmann. Evidence and non-repudiation. *J. Netw. Comput. Appl.*, 20(3):267–281, 1997.

[Zho04] J. Zhou. On the security of a multi-party certified email protocol. In *ICICS '04*, volume 3269 of *LNCS*, pages 40–52. Springer, 2004.