# On Fairness in Exchange Protocols

Olivier Markowitch[1], Dieter Gollmann[2], and Steve Kremer[1]

[1]Université Libre de Bruxelles
Bd du Triomphe - CP212
1050 Brussels
Belgium

[2]Microsoft Research Ltd
7 J J Thomson Avenue
Cambridge CB3 0FB
United Kingdom

**Abstract.** The aim of this paper is to give an overview of the most classical definitions of fairness in exchange protocols. We show the evolution of the definition, while putting forward that certain definitions are rather vague or too specialized. We propose a structured and generalized definition of fairness and of the security of exchange protocols.

Keywords: security protocols, fairness, exchange protocols, fair exchange, security properties.

## 1 Introduction

With the growth of open networks in general and the Internet in particular, many security related problems have been identified and solutions have been proposed. Applications in which the fair exchange of electronic items between users is required are becoming more frequent. Payment systems, electronic commerce, certified e-mail and contract signing are classical examples in which fairness is a relevant security property. Informally, an exchange protocol is said to be fair if it ensures that during the exchange of the items, no party involved in the protocol can gain a significant advantage over the other party, even if the protocol is halted for any reason.

This paper addresses the problem of defining fairness. In the literature, one finds many different definitions of fairness. Some are too vague (such as for example the informal definition above), and some are too constraining. One important problem is due to the fact that the word "fairness" has several usual interpretations in the current language. These interpretations are often distinct from the definition needed in the context of exchange protocols.

Another problem is the excessive use of the fairness property in computer security. If fairness may be sound in the theoretical study of exchange protocols, its

practical necessity must be argued depending on the actual situation. To carry out a protocol respecting fairness requires the set up of security mechanisms that may sometimes be rather heavy. It results in an increase of computations and/or communications. It can on occasion be more realistic, in practice, to envisage mechanisms that manage the problems potentially occurring during an exchange separately from the exchange protocol itself. For example, in an exchange of low value electronic information against payments, in case of an unfair situation the seller could accept the loss or he can lodge a complaint against the buyer. In this paper we will focus on theoretical aspects of fairness in exchange protocols.

The majority of publications on fair exchange assume the existence of a trusted third party (TTP) in the protocol. Independently of how the TTP is involved in the protocol, its main role is to resolve the problems that may occur between the involved parties.

Some protocols [11, 17, 32] use a TTP to store the details of the transaction in order to help to successfully complete an exchange. As the TTP is actively involved in the protocol, this approach considerably reduces the efficiency of the exchange. To remedy this shortcoming, independently Micali [22] and Asokan et al. [1, 4] proposed a solution that avoids the presence of the TTP between the parties. They proposed not to use the TTP during the transaction when the parties behave correctly and the network functions, but to invoke the TTP to complete the protocol in case of problems with one of the parties or the network. Such protocols are said to be *optimistic*.

Fair exchange protocols without TTP have also been proposed [9, 20, 27, 28]. However the last two are based on an unpractical definition of fairness (as we will see below) and the others adopt a probabilistic approach towards fairness.

Independently of the way a TTP is (or is not) used, several categories of exchange protocols exist, depending on the information to be exchanged:

- electronic purchase of digital goods: exchange of an electronic item against an electronic payment (issued by the client)
- digital contract signing: exchange of signatures on a given electronic document
- non-repudiation protocol: exchange of an electronic item and its proof of origin against a proof of receipt
- certified e-mail: exchange of an electronic message against a proof of receipt[1]
- barter: an electronic item of value is exchanged against another electronic item of (similar) value
- ...

The existence of different categories is responsible for some exotic definitions of fairness. We also note that two-party fair exchange protocols and multi-party

[1] A difference between non-repudiation protocols and certified e-mail protocols is that in the latter the recipient of the message should not know the sender's identity when deciding to accept the message or not [19]. Moreover, non repudiation of origin may not be required in certified e-mail.

fair exchange protocols have often different fairness definitions (partially due to different topologies used in the multi-party case) [2, 7, 13].

In this paper we defend the point of view of a unified definition of fairness, whatever the underlying exchange protocol or formalism may be. We put ourselves in a theoretical context where fairness is always needed and propose a generalized definition of fairness and of the security of an exchange protocol. Our definitions aim to capture *what* a given property provides, and not *how* it is provided. To quote Roscoe [24], we avoid "intensional definitions", which are related to a sequence of actions that must (or must not) happen in a given order, for a property to hold. Intensional specifications are useful in formal verification as they capture the way the protocol designers have foreseen the protocol execution (and therefore, we do not criticize them) but are not general enough to give a general definition of a concept such as fairness.

In this vein, when examining the statement that "no party has an advantage" we will distinguish between the aspects of an exchange the advantage could apply to, but we will not examine how this advantage could be measured. We warn the readers who expect to find formalized definitions that they will be disappointed, but disappointed for a reason. There are different views of what constitutes an advantage and we have to be able to differentiate these aspects and define them informally but clearly before choosing a particular formalism. The formalism would explain what is meant by gaining information (implying some interesting views, as those proposed in [18, 25]). There can again be different formal definitions of "gaining information". We would like to stress that an informal analysis of security properties is essential before properties are being formalized. The goal is to promote a clear informal understanding of fairness to be able to compare different formalisms.

In the next section, we survey and discuss some of the more classic fairness definitions found in the literature. In the third section, we propose a consistent and modular definition of the security of an exchange protocol, where fairness is one of the properties needed. We point out the necessity of other important properties like timeliness, viability and non-repudiability in this security definition. We conclude in the last and fourth section.

## 2 Evolution of the fairness definition

### 2.1 Historical definitions

Although the relevance of fairness has been well appreciated since the early 1980s, the first propositions of a fairness definition corresponding to practical solutions were expressed in terms of computing power. The protocols exchanged information piece by piece and it was required that the computational effort required from the parties to obtain each others remaining information should be approximately equal at any stage during the execution of the protocol.

Even et al. [12] proposed a classical definition of fairness in the framework of contract signing, called "concurrency": *if one party $X$ executes the protocol properly, then his counterpart $Y$ cannot obtain $X$'s signature to the contract without yielding his own signature to it.* Unfortunately, they did not propose a solution respecting this definition. In order to solve the exchange problem they introduced a weaker definition, called "approximate-concurrency": *if one party $X$ executes the protocol properly then with very high probability, at each stage during the execution, $X$ can compute his counterpart's signature to the contract using approximately the same amount of work used by $Y$ to compute $X$'s signature to the contract.* This is what we call the computational approach towards fairness.

It was rapidly accepted that requiring an equal, equivalent or even related[2] computing power between the communicating parties is not reasonable.

An important evolution was the probabilistic approach not requiring equivalent computing power, first proposed by Ben Or et al. [8] in the contract signing framework. They defined probabilistic fairness in the following way: *a party is privileged when s/he is capable of causing the judge to rule that the contract is binding on both parties*; *a contract signing protocol is $(v, \varepsilon)-fair$ for a party $A$ if the following holds for any contract $C$ when $A$ follows the protocol properly: at any step of the protocol, in which the probability that another party $B$ is privileged is greater than $v$, the conditional probability that $A$ is not privileged given that $B$ is privileged is at most $\varepsilon$. $\varepsilon$ denotes an upper bound on the probability that one party is not privileged given that the other is privileged.*

Protocols based on this last definition are traditionally implemented by successive rounds during which, in turn, a party is privileged whereas the other is not. This yields a situation that could be considered unfair (in the common sense). In our eyes, the fact that the entities are privileged in turn is not unfair. It would be unfair if one party were able to prove that the other party is linked alone to the contract.

Putting aside these historical definitions, the actually most widely accepted definition of fairness [3, 4, 6, 12, 16, 26, 29, 31] describes fairness in relation to the end of the exchange protocol run: *at the end of the exchange protocol run, either all involved parties obtain their expected information or none of them receives anything.* We consider this definition to be, almost, the most suitable one[3].

### 2.2 Definitions with ballast

There are many recent definitions of fairness that include additional properties in the basic definition. Often, fairness definitions describe the mechanisms nec-

[2] Where the computing power ratio between two communicating parties is known and fixed.

[3] Moreover, this definition can easily be adapted in a probabilistic context: *at the end of the exchange protocol run, there has to be an overwhelming probability that either all involved parties obtain their expected information or none of them receives anything.*

essary to realize fairness in particular cases. The definition is not only based on what is fairness but rather on how to obtain it.

**Digital contract signing**

In the context of digital contract signing, Asokan et al. [5] specify by the means of a game that an exchange protocol for signatures is not fair if a malicious entity can exchange an invalid signature against a valid one. Although the definition is appropriate, a general definition of fairness should not be based on such specific concerns (even if the proposed definition is always true in the framework of digital contract signing). We believe that a general definition of fairness can be expressed such that all exchange types are covered. The way followed to obtain fairness depends on the context: for digital contract signing protocols, the security of the signature is an important part but this aspect has to be developed in the security proof of the protocol and not in the fairness definition.

Garay et al. [15] specify that an optimistic contract signing protocol is fair if:

1. *it is impossible for a corrupted participant to obtain a valid contract without allowing the remaining participant to also obtain a valid contract*
2. *once a correct participant obtains a cancellation message from the TTP, it is impossible for any other participant to obtain a valid contract*
3. *every correct participant is guaranteed to complete the protocol.*

The restriction on the cancellation message (the second rule) seems too restrictive. Indicating that a party, having carried out a cancellation, should not take the risk to continue the protocol is a part of the *protocol's description*: the specification of the behavior of the parties implied in the protocol belongs to its description and should not belong to the fairness definition. Moreover, as cancellation is specific to optimistic protocols, we would have different fairness definitions depending on the TTP's involvement.

Note that the cancellation (or abort) token is produced during an optimistic protocol to inform the party asking for a cancellation that the TTP will no longer accept recovery requests during this protocol run. This cancellation token, issued during an abort protocol, is necessary to ensure timeliness (which will be clearly defined in the third section). The timeliness property is respected if the parties always have the ability to reach, in a finite amount of time, a point in the protocol where they can stop the protocol while preserving fairness.

The third rule talks about the ability to complete the protocol. If the guarantee to complete a protocol is related to the fact that a way to securely end the protocol (with or without a completed exchange) must exist, then this corresponds to the timeliness property we just discussed. Otherwise, if completing the protocol is related to the fact that the exchange succeeds, this is the viability property. A protocol is viable if the exchange always succeeds when the involved parties behave honestly (i.e. follow the protocol). Viability differs from fairness and is more difficult to obtain in practice, because the success of the exchange does

not depend only on the honesty of the parties but also on the quality of the underlying network.

In [23], Pfitzmann et al. say that *a contract signing scheme is called fair if it fulfills the following requirement:*

1. *correct execution*
2. *unforgeability of contracts*
3. *verifiability of valid contracts (a signed contract cannot be invalidated)*
4. *no surprise with invalid contracts (a rejected contract — no party had signed it — cannot be declared signed)*
5. *termination on synchronous network (the protocol ends after a finite amount of rounds)*
6. *termination on asynchronous network (after a time-out or a user's manual input, the protocol ends after a fixed time)*

The first rule is related, if no time-out is used, to the viability property, which is distinct from fairness.

The second, third and fourth rules are specific to digital contract signing. As these rules apply to any contract signing protocol, these definitions could be considered as extensional specifications of a contract signing protocol. In [23], these statements are defined in terms of precise inputs and outputs of the protocol and in terms of execution of subprotocols *show* and *sign*[4]. Such definitions refer to the machinery of contract signing.

The fifth and sixth rules are related to the timeliness property, which is also distinct from fairness.

Of course, it may be the case that in certain formalisms it is quite difficult to state extensional definitions.

**Fair exchange**

Vogt et al. [29] proposed to split the definition of fairness into two aspects: the participation of a "faulty" entity is or is not needed when the TTP is requested to help finishing the protocol. Again, this approach described *how* fairness is obtained. We emphasize once more that when defining fairness it is necessary to focus on *what* is fairness and not on *how* to obtain it.

Franklin et al. [14] said that *at the end of the fair exchange the following must be true:*

1. *if A, B, and the TTP are honest, A learns B's information and B learns A's information;*

[4] For example, in the fairness definition [23], the statement "Verifiability of valid contracts" is defined by "If a correct signatory, say $A$, outputs $(signed, C, tid)$ and later executes "show" on input $(show, tid)$ then any correct verifier will output $(signed, C, tid)$ for any $C$"

2. *if A and the TTP are honest then B does not learn anything about A's information unless if A learns B's information;*
3. *if B and the TTP are honest A does not learn anything about B's information unless if B learns A's information;*
4. *if A and B are honest then the TTP does not learn anything about A's and B's information.*

Again, the first property is the viability property. The fourth property is about confidentiality with regard to the TTP. This is not needed in fairness but is due to the context of key exchanges of their paper.

### 2.3 Vague definitions

Zhou et al., in the context of non-repudiation protocols [30, 32–34], define fairness as follows: *a non-repudiation protocol is fair if it provides the originator and the recipient with valid irrefutable evidence after completion of the protocol, without giving a party an advantage over the other at any stage of the protocol run.*

Boyd et al. [10] propose a similar definition: an exchange protocol is fair if *at no point during the execution of the protocol either of the entities participating in the exchange can gain any (significant) advantage over the other if the protocol is suddenly halted.*

In both definitions the notion of advantage is not defined. These definitions seem practically to exclude any protocol not offering a perfect symmetry (in terms of knowledge and possibility of action). However such a definition is obviously not formalizable and seems primarily related to the common acceptance of the word fairness.

Moreover in the definition by Zhou et al., the first part of the definition imposes viability, which is, in practice, rather unrealistic.

### 2.4 Weak fairness and transparent TTP

Asokan introduced [1] the notion of weak fairness in relation to protocols where fairness can be broken in certain circumstances. In a weakly fair protocol, a well behaving despoiled party is able, thanks to the help of the TTP to proof his honesty to an external adjudicator. More precisely, if a party $A$ does not receive its expected item, it will be able to prove to an external adjudicator that the other party received the item sent by $A$ or is able to retrieve this item without any further intervention from $A$. If the misbehaving party (who has not provided his item) refuses to cooperate, the TTP will transmit to A an affidavit in replacement of the missing information.

In practice, this property is interesting in protocols with a low weight TTP, when it is more relevant to obtain affidavits produced by the TTP than the expected low cost items. Note, that one also has to define dispute resolution

protocols, defining the way an adjudicator has to evaluate these affidavits, as it is the case in non-repudiation protocols. Weak fairness shows an interesting way of linking fairness and non-repudiation. Participants do not get a guarantee that they will obtain the intended item, but at least they get a non-repudiation evidence that the other party was involved in a particular run of the exchange protocol. Weak fairness may also be interesting, in some circumstances, to allow to achieve simultaneously some kind of fairness and timeliness. Therefore weak fairness may be of practical interest.

Recent evolutions [5, 10, 15, 21] in optimistic exchange protocols with transparent TTP, based on verifiable encryption and recoverable signatures, offer solutions where it is possible to maintain "strong" fairness. In such optimistic protocols the TTP is always able to retrieve the original expected information in case of a problem, without needing the cooperation of the parties to enforce fairness.

Moreover, with such a transparent TTP, at the end of a protocol where the exchange is realized, it is impossible to decide whether the TTP did intervene in the protocol execution or not. As it is difficult to determine whether the TTP was required during the protocol because of a dishonest party or because of a network problem, a transparent TTP may be particularly relevant, for example, in an electronic commerce environment.

### 2.5 Abuse-free digital contract signing protocols

Recently Garay et al. [15] introduced the notion of abuse-free digital contract signing protocols. An optimistic contract signing protocol is abuse-free if it is impossible for a single entity at any point in the protocol to be able to prove to an outside party that he has the power to either terminate (abort) or successfully complete the protocol.

The main protocol they propose consists of four steps. During the first part of the main protocol (the first two steps) the parties exchange verifiable commitments to signatures (called "private contract signatures"). The specificity of these signatures is that only the intended recipient is able to verify whether the verifiable committed signature he received can be transformed into universally verifiable signatures by a TTP. Moreover, this recipient is not able to prove the committed signature's validity to any external parties. The second part of the main protocol consists of the exchange of the universally verifiable signatures on the contract. In case of problems, the entities can run a recovery protocol with the TTP. The TTP will extract the universally verifiable signature from the committed ones.

Hence, as only Alice, Bob and the TTP can verify the commitments, it is not possible to prove to an external party that a protocol run has been engaged in. Proving to an external party that a contract is going to be signed may be useful, for instance, in a sale protocol, in order to make this external party increase his offer.

Abuse-freeness is an interesting property. In our view, its most important feature is that committed signatures are not universally verifiable.

Note that it is not sufficient to use non-universally verifiable committed signatures to obtain abuse-free contract signing protocols. With a non-universally verifiable signature only the expected recipient is able to verify the signature. But this recipient should not be able to prove the validity of this signature to an external party (for example thanks to an interactive proof of knowledge of the secret he used to verify the committed signature or even by completely divulging its secret).

However, if using a resilient network when communicating with the TTP (i.e. a network where messages are delivered before a finite although unknown amount of time), we are sceptic about the ability of a party to either terminate or successfully complete the protocol. It is, in fact, rather easy for Bob to force the termination of the protocol proposed in [15]. If he stops the protocol, the only thing Alice can do is to launch an abort protocol. Forcing the successful completion of the protocol is harder: Bob needs to send a recovery request before Alice's abort request arrives at the TTP. We believe that it is rather difficult to block messages on a resilient network. Hence, a race condition decides whether the abort request or the recovery request first arrives at the TTP. This means that, when using resilient channels, most of the optimistic contract signing protocols, are actually abuse-free, as a race condition decides of the outcome of the protocol. Although the protocols may be abuse-free, with respect to the definition given in [15], the fact that a party can prove to an outsider, that the protocol has been engaged with a given party, before the final outcome of the protocol is known, may be considered as a problem. This motivates the use of private contract signatures, which overcome that problem. Also note that when all the communication channels are synchronous (i.e. messages are delivered before a finite and constant amount of time), which is not realistic in practice as the transmissions are not controlled by race conditions anymore, private contract signatures are indeed needed to ensure that the protocol is abuse-free.

## 3 A general and modular definition of the security of an exchange protocol

In this section, we require the definition of an exchange protocol to explicitly refer to the items the protocol participants want to exchange. The properties following below are defined with respect to the items exchanged as referred to in the exchange protocol definition.

As suggested in [15, 16] we propose to speak about the *security* of exchange protocols. However, we do not consider that non-repudiability or abuse-freeness must imperatively be in the mandatory part of a definition of the security of an exchange protocol.

We say that an exchange protocol is **secure** when it respects these three mandatory properties :

- **viability**: independently of the communication channels quality, there exists an execution of the protocol, where the exchange succeeds.
- **fairness**: the communication channels quality being fixed, at the end of the exchange protocol run[5], either all involved parties obtain their expected items or none (even a part) of the information to be exchanged with respect to the missing items is received.
- **timeliness**: the communication channels quality being fixed, the parties always have the ability to reach, in a finite amount of time, a point in the protocol where they can stop the protocol while preserving fairness.

Moreover, a secure exchange protocol can respect some optional properties. For example :

- **non repudiability**: it is impossible for a single entity, after the execution of the protocol, to deny having participated[6] in a part or the whole of the communication.
- **abuse-freeness**: it is impossible for a single entity at any point in the protocol to be able to prove to an outside party that he has the power to terminate (abort) or successfully complete the protocol [15].

If we accept the common meaning of "fairness", we can consider that this paper deals with three different aspects of fairness:

- fairness, as defined, relating to the items exchanged during the protocol;
- fairness relating to the ability to determine the progress of the protocol, called timeliness;
- fairness relating to the ability to make statements about the possibility to determine the progress of a protocol, called abuse-freeness.

Although these three properties can be bound to the common meaning of the word "fairness", they are related to fairness at different levels and should not be merged in one single definition. To avoid confusion, we prefer not to use the term fairness when talking about timeliness or abuse-freeness and put the emphasis on a modular and general definition.

To illustrate our concepts, consider the following classical protocol, described in a very general way, where Alice exchanges an electronic item against Bob's digital signature on the publicly known description of the item:

[5] It should be noted that the end of the exchange protocol run is not necessarily related to the fact that the exchange succeeded.

[6] Classical non-repudiation needs are non-repudiation of origin of a message and non-repudiation of receipt of a message.

*Main protocol:*

1. $B \rightarrow A$: committed signature (the TTP can open it without the help of B)
2. $A \rightarrow B$: item
3. $B \rightarrow A$: signature

*Recovery protocol:*

1. $A \rightarrow TTP$: $B$'s committed signature and the item
2. $TTP \rightarrow A$: $B$'s signature
3. $TTP \rightarrow B$: $A$'s item

In the main protocol, Bob begins by sending to Alice his committed signature on the item's description. Alice cannot retrieve Bob's final signature on the description from the committed one, but we assume that she can verify the correctness of the commitment (she is able to verify that the TTP will be able to retrieve Bob's final signature from the committed one). Then Alice sends to Bob the expected electronic item. If the item corresponds to the description expected by Bob, he sends to Alice his final signature on the item's description. This final signature can be considered as the confirmation that Bob has received (or paid) Alice's electronic item.

If Bob does not send his final signature at the third step of the main protocol, Alice can initiate a recovery protocol with the TTP. She sends to the TTP Bob's committed signature and the item. The TTP verifies whether the committed signature is valid and whether the signed description corresponds to the item Alice sent. If all the checks hold, the TTP computes Bob's final signature on the description from the committed one and sends to Alice Bob's final signature and to Bob the item.

According to our definitions this protocol is fair. When Alice receives Bob's committed signature, either she sends to Bob the item and receives Bob's item during the main protocol, or she runs the recovery protocol and both of them obtain their item by the mean of the TTP.

However, Alice can block the protocol after having received Bob's committed signature. If Alice, after having received Bob's committed signature, suspends her participation in the protocol, Bob cannot decide when to leave the protocol in a fair way. As Bob cannot run the recovery protocol, Alice has always the possibility, after Bob left the protocol, to run the recovery protocol. Therefore Bob can never leave the protocol before receiving Alice's item. Although fairness is never broken, the timeliness property is not fulfilled.

The protocol is viable (the three steps of the main protocol make the exchange happen) and fair, but does not respect timeliness. Hence, the protocol is not secure[7]. We thus modify the protocol as follows (obtaining a protocol which is an abstract version of the one proposed in [4]).

[7] With regard to the definitions proposed here, optimal efficiency of an optimistic contract signing protocol [23] refers to secure protocols and not only fair protocols.

*Main protocol:*

1. $A \rightarrow B$: committed item
2. $B \rightarrow A$: committed signature (the TTP can open it without the help of B)
3. $A \rightarrow B$: item
4. $B \rightarrow A$: signature

*Recovery protocol:*

1. $A$ or $B \rightarrow TTP$: $B$'s committed signature and committed item
2. $TTP \rightarrow A$: $B$'s signature
3. $TTP \rightarrow B$: $A$'s item

With these modifications Bob has an advantage, as he can initiate the recovery protocol right after having received the first message of Alice in the main protocol. Alice has to receive the second message of Bob to be able to do so. As described, the protocol does not respect the timeliness property, because Bob can temporarily suspend his participation in the protocol before deciding whether to continue the main protocol (by sending his committed signature) or to initiate the recovery protocol. So traditionally an abort protocol can be run by Alice.

*Abort protocol:*

1. $A \rightarrow TTP$: abort request
2. $TTP \rightarrow A$: abort confirmation
3. $TTP \rightarrow B$: abort confirmation

The recovery protocol and the abort protocol are mutually exclusive and this mutual exclusion is assured by the TTP.

The abort protocol can be used to prevent Bob to realize a recovery after having received the first message of the main protocol. But if the main protocol is run until its end, the exchange is achieved. If Alice executes the abort protocol after a completed exchange, the exchange still holds. The goal of the abort protocol is to ensure the timeliness property. An abort confirmation cannot cancel a successful exchange.

Only Alice has the power to abort the protocol. If Bob wants the protocol to be aborted he has to wait long enough after having received the first message of the main protocol in order to force Alice to initiate the abort protocol. On the other hand, although Bob cannot run the abort protocol, he can, in practice, abort the main protocol (by stopping his participation) without informing Alice. Whereas when Alice makes an abort, Bob is informed by the TTP. These are not a security problem of the protocol but could be considered as unfair in the common sense.

The definitions we propose are valid in a two party case or in a multi-party case. We do not specify in our definitions from whom the information must come

and to whom it should be sent. The topology, which differentiates the various multi-party exchange protocols, does not interfere here.

Similarly to the framework of digital contract signing protocols, a probabilistic definition of fairness [20] exists in the context of non-repudiation protocols. An exchange protocol is probabilistically fair if the communication channels quality being fixed, at the end of the exchange protocol run, the probability that one party obtains an expected information without providing his counterpart information is negligible and can be parametrized. With such a definition, we are able to design exchange protocols without TTP, which do not require equivalent computing power among the parties.

## 4 Conclusion

We observed, in the literature on fair exchange protocols, that some definitions of fairness include aspects specific to the application the exchange protocol is intended for. Moreover, some definitions of fairness actually stipulate the way fairness should be achieved. Such definitions provide an insufficient separation between the specification of a protocol and the specification of the protocol goals.

We have proposed a unified definition of fairness, independent of the the underlying exchange protocol, and a structured and generalized definition of the security of exchange protocols, distinguishing between the aspects of viability, fairness, and timeliness.

Finally, we have suggested how fairness, timeliness, and abuse-freeness could be considered as capturing different aspects of the informal idea of "having no advantage".

## References


1. N. Asokan. *Fairness in Electronic Commerce*. PhD thesis, University of Waterloo, May 1998.
2. N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for multi-party fair exchange. Research Report RZ 2892 (# 90840), IBM Research, Dec. 1996.
3. N. Asokan, M. Schunter, and M. Waidner. Optimistic protocols for fair exchange. In *Proceedings of the fourh ACM Conference on Computer and Communications Security*, pages 8–17. ACM Press, Apr. 1997.
4. N. Asokan, V. Shoup, and M. Waidner. Asynchronous protocols for optimistic fair exchange. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Research in Security and Privacy, pages 86–99. IEEE Computer Society,Technical Committee on Security and Privacy, IEEE Computer Security Press, May 1998.
5. N. Asokan, V. Shoup, and M. Waidner. Optimistic fair exchange of digital signatures. *IEEE Journal on Selected Areas in Communications*, 18(4):593–610, Apr. 2000.

6. G. Ateniese. Efficient verifiable encryption (and fair exchange) of digital signatures. In *6th ACM Conference on Computer and Communications Security*, pages 138–146, Singapore, Nov. 1999. ACM Press.
7. F. Bao, R. Deng, K. Q. Nguyen, and V. Vardharajan. Multi-party fair exchange with an off-line trusted neutral party. In *DEXA'99 Workshop on Electronic Commerce and Security*, Florence, Italy, Sept. 1999.
8. M. Ben-Or, O. Goldreich, S. Micali, and R. Rivest. A fair protocol for signing contracts. *IEEE Transaction on Information Theory*, 36(1):40–46, Jan. 1990.
9. D. Boneh and M. Naor. Timed commitments. In *Advances in Cryptology: Proceedings of Crypto 2000*, volume 1880 of *Lecture Notes in Computer Science*, pages 236–254. Springer-Verlag, 2000.
10. C. Boyd and E. Foo. Off-line fair payment protocols using convertible signatures. In *Advances in Cryptology: Proceedings of Asiacrypt'98*, volume 1514 of *Lecture Notes in Computer Science*, pages 271–285. Springer-Verlag, 1998.
11. T. Coffey and P. Saidha. Non-repudiation with mandatory proof of receipt. *ACM-CCR: Computer Communication Review*, 26, 1996.
12. S. Even, O. Goldreich, and A. Lempel. A randomized protocol for signing contracts. *Communications of the ACM*, 28(6):637–647, June 1985.
13. M. Franklin and G. Tsudik. Secure group barter: Multi-party fair exchange with semi-trusted neutral parties. *Lecture Notes in Computer Science*, 1465, 1998.
14. M. K. Franklin and M. K. Reiter. Fair exchange with a semi-trusted third party. In *4th ACM Conference on Computer and Communications Security*, pages 1–5. ACM Press, Apr. 1997.
15. J. A. Garay, M. Jakobsson, and P. MacKenzie. Abuse-free optimistic contract signing. In *Advances in Cryptology: Proceedings of Crypto'99*, volume 1666 of *Lecture Notes in Computer Science*, pages 449–466. Springer-Verlag, 1999.
16. F. C. Gärtner, H. Pagnia, and H. Vogt. Approaching a formal definition of fairness in electronic commerce. In *Proceedings of the International Workshop on Electronic Commerce (WELCOM'99)*, pages 354–359, Lausanne, Switzerland, Oct. 1999. IEEE Computer Society Press.
17. Y. Han. Investigation of non-repudiation protocols. In *ACISP: Information Security and Privacy: Australasian Conference*, volume 1172 of *Lecture Notes in Computer Science*, pages 38–47. Springer-Verlag, 1996.
18. M. Jakobsson. Ripping coins for fair exchange. In L. C. Guillou and J.-J. Quisquater, editors, *Advances in Cryptology: Proceedings of Eurocrypt'95*, volume 921 of *Lecture Notes in Computer Science*, pages 220–230. Springer-Verlag, 21–25 May 1995.
19. S. Kremer and O. Markowitch. Selective receipt in certified e-mail. In *Advances in Cryptology: Proceedings of Indocrypt 2001*, Lecture Notes in Computer Science. Springer-Verlag, Dec. 2001.
20. O. Markowitch and Y. Roggeman. Probabilistic non-repudiation without trusted third party. In *Second Conference on Security in Communication Networks'99*, Amalfi, Italy, Sept. 1999.
21. O. Markowitch and S. Saeednia. Optimistic fair-exchange with transparent signature recovery. In *5th International Conference, Financial Cryptography 2001*, Lecture Notes in Computer Science. Springer-Verlag, 2001.
22. S. Micali. Certified E-mail with invisible post offices. Available from author; an invited presentation at the RSA '97 conference, 1997.
23. B. Pfitzmann, M. Schunter, and M. Waidner. Optimal efficiency of optimistic contract signing. In *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, pages 113–122, New York, May 1998. ACM.

24. A. W. Roscoe. Intensional specifications of security protocols. In *Proceedings of the 9th IEEE Computer Security Foundations Workshop*, pages 28–38. IEEE Computer Security Press, 1996.
25. P. Syverson. Weakly secret bit commitment: Applications to lotteries and fair exchange. In *Proceedings of the 1998 IEEE Computer Security Foundations Workshop (CSFW11)*, June 1998.
26. P. Syverson. Weakly secret bit commitment: Applications to lotteries and fair exchange. In *Proceedings of the 1998 IEEE Computer Security Foundations Workshop (CSFW11)*, pages 2–13, June 1998.
27. T. Tedrick. How to exchange half a bit. In D. Chaum, editor, *Advances in Cryptology: Proceedings of Crypto'83*, pages 147–151, New York, 1984. Plenum Press.
28. T. Tedrick. Fair exchange of secrets. In G. R. Blakley and D. C. Chaum, editors, *Advances in Cryptology: Proceedings of Crypto'84*, volume 196 of *Lecture Notes in Computer Science*, pages 434–438. Springer-Verlag, 1985.
29. H. Vogt, H. Pagnia, and F. C. Gärtner. Modular fair exchange protocols for electronic commerce. In *Proceedings of the 15th Annual Computer Security Applications Conference*, pages 3–11, Phoenix, Arizona, Dec. 1999. IEEE Computer Society Press.
30. J. Zhou, R. Deng, and F. Bao. Evolution of fair non-repudiation with TTP. In *ACISP: Information Security and Privacy: Australasian Conference*, volume 1587 of *Lecture Notes in Computer Science*, pages 258–269. Springer-Verlag, 1999.
31. J. Zhou, R. Deng, and F. Bao. Some remarks on a fair exchange protocol. In *Proceedings of 2000 International Workshop on Practice and Theory in Public Key Cryptography*, volume 1751 of *Lecture Notes in Computer Science*, pages 46–57. Springer-Verlag, Jan. 2000.
32. J. Zhou and D. Gollmann. A fair non-repudiation protocol. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, Research in Security and Privacy, pages 55–61. IEEE Computer Society,Technical Committee on Security and Privacy, IEEE Computer Security Press, May 1996.
33. J. Zhou and D. Gollmann. An efficient non-repudiation protocol. In *Proceedings of The 10th Computer Security Foundations Workshop*, pages 126–132. IEEE Computer Society Press, June 1997.
34. J. Zhou and D. Gollmann. Evidence and non-repudiation. *Journal of Network and Computer Applications*, 20:267–281, 1997.