Risk Balance in Exchange Protocols

Mohammad Torabi Dashti and Yanjing Wang

CWI, Amsterdam
{dashti,y.wang}@cwi.nl

Abstract. We study the behaviour of rational agents in exchange protocols which rely on trustees. We allow malicious parties to compromise the trustee by paying a cost and, thereby, present a game analysis that advocates exchange protocols which induce balanced risks on the participants. We also present a risk-balanced protocol for fair confidential secret comparison.

1 Introduction

Exchange protocols aim to establish successful exchanges of electronic goods between two parties who possibly have conflicting interests. Fairness, stipulating that either both or none of the parties achieve their goals, is recognised as a crucial requirement for exchange protocols (e.g. see [1]). Achieving fairness in deterministic asynchronous exchange protocols with no trusted parties is however impossible [6]. The existing methods, therefore, either are based on gradual release of information or gradual increase of privilege to approximate fairness, or rely on trusted third parties (TTPs).

This paper focuses on exchange protocols which rely on a TTP, while malicious participants are allowed to, by paying a cost, compromise the TTP.[1] We thereby present a game analysis that advocates protocols which induce (nearly) the same amount of risk on the participants. Our main result states that in such risk-balanced protocols, the difference between participants' utilities is limited to a factor independent of the TTP's trustworthiness. Hence, none of the participants would hugely suffer compared to the other one, in case the trustee is compromised by the opponent.

Existing game analyses of exchange protocol assume non-compromisable trustees, e.g. [2,3,5,9]. This is in contrast to the premise of our analysis that TTPs, by paying a cost, can be compromised. In a similar study, the authors of [10] assume that participants may have limited trust in TTPs and propose algorithms to determine whether a rational agent would engage in an exchange using cascades of TTPs or not. They however do not consider that participants may have the choice to compromise TTPs.

Studying the ways a compromised TTP may affect fair exchange protocols and methods to limit its damages are not well studied. As a notable exception to this, Asokan explores the concept of verifiable TTPs [1] in optimistic protocols, where the TTP's incentive for cheating is lowered, as its malicious behaviour can be detected.

As an example of a risk-balanced protocol, we present a fair protocol for confidentially comparing secrets. Existing protocols for this purpose either do not aim at fairness [7], or do not involve TTPs [13,14], thus only achieve probabilistic fairness, or are universal multi-party computing protocols [4] that are not optimised for this task.

[1] Note that a *trusted* entity, in general, may not be *trustworthy*, cf. [8].

I. Cervesato (Ed.): ASIAN 2007, LNCS 4846, pp. 70–77, 2007.

2 Game Abstraction of Exchange Protocols

From a game theoretical point of view, a two-party exchange protocol with a compromisable TTP can be seen as a two-party strategic game, in which the agents can either follow the protocol faithfully or compromise the TTP. If both parties play faithfully, then they normally "earn" the goods from the opponent and "lose" their own goods. However, when engaging in the exchange, each agent has to take some risk due to the fact that the opponent may manage to compromise the TTP. In such cases, the agent who compromises the TTP can earn the amount that the other (honest) party risks, and lose only the cost of compromising the TTP.

Formally, we have the following game abstraction [2]:

Definition 1. ***(Protocol game)*** *Given a two-party exchange protocol* Prot *with a TTP, the strategic game* $G(\mathsf{Prot})$ *is defined as follows:*

$A\backslash B$	$\mathcal{H}_B$	$\mathcal{DH}_B$
$\mathcal{H}_A$	$g_B^A - g_A^A, g_A^B - g_B^B$	$-r_A^A, r_A^B - c_B$
$\mathcal{DH}_A$	$r_B^A - c_A, -r_B^B$	$r_B^A - r_A^A - c_A, r_A^B - r_B^B - c_B$

where $\mathcal{H}_x$ is the strategy of x that is according to the protocol; $\mathcal{DH}_x$ is the strategy of x in which x compromises the TTP and may stop following the normal course of the protocol when she has to release its goods; g_x^y is y's evaluation of the goods that x wants to exchange; r_x^y is y's evaluation of the risk that x has, if the TTP is compromised by the opponent of x; and c_x is the cost x pays to compromise the TTP. [3]

In the following we assume:

- Agents have incentives to exchange goods: $g_x^y > g_x^x$ if $x \neq y$. For simplicity, we assume that there is a fixed exchange rate $\rho > 1$ such that $g_x^y = \rho g_x^x$, $x \neq y$.
- The risks of the agents comply with the same exchange rate: $r_x^y = \rho r_x^x$, $x \neq y$.
- The subjective values of the goods are the same: $g_A^A = g_B^B = g > 0$.
- The costs of compromising the TTP are the same for both agents: $c_A = c_B = c$.

With these assumptions, $G(\mathsf{Prot})$ can be simplified to $SG(\mathsf{Prot})$:

$A\backslash B$	$\mathcal{H}_B$	$\mathcal{DH}_B$
$\mathcal{H}_A$	$(\rho-1)g, (\rho-1)g$	$-a, \rho a - c$
$\mathcal{DH}_A$	$\rho b - c, -b$	$\rho b - a - c, \rho a - b - c$

Where $a = r_A^A$ and $b = r_B^B$.

To apply game theoretical analysis, we assume that the agents are rational utility-maximisers. A strategy profile is a joint strategy that determines a unique utility pair; for example $(\mathcal{H}_A, \mathcal{H}_B)$ is a strategy profile while $((\rho-1)g, (\rho-1)g)$ is the corresponding utility pair. A strategy profile (S_A, S_B) is called a Nash equilibrium if no agent gets higher utility by switching to another strategy, given the strategy of the other agent

[2] Due to space constraints we omit introducing basics of game theory, and instead refer to [11].

[3] We assume that both parties can compromise the TTP at the same time. For example, they both may exploit vulnerabilities in the TTP's software to read certain information off its storage.

according to the profile. In this paper, we consider the Nash equilibria of a simplified protocol game as the expected executions of the corresponding protocol by rational agents. We write $\text{Utility}_x(S_A, S_B)$ as the utility of x if the agents select the strategy profile (S_A, S_B).

3 Risk Balance

We define a requirement on exchange protocols, which we call Δ-*condition*, that puts an upper bound on the difference between the risks that a protocol induces on its participants. We show that this condition in turn puts a limit on the difference between participants' expected utilities. The limit on utility differences turns out to be independent of c. This is a desirable property since it ensures that no matter how trustworthy the TTP might be in an execution, the difference between participants' utilities is limited to a value independent of c, hence none of the participants would hugely suffer (or benefit) compared to the other one. This can be interpreted as fairness in a meta level.

In the following, when the context of the simplified protocol game is clear, let $\Delta = |a - b|$ and $\Delta_U(S_A, S_B) = |\text{Utility}_A(S_A, S_B) - \text{Utility}_B(S_A, S_B)|$.

Definition 2. *An exchange protocol* Prot *satisfies* Δ*-condition iff* $\Delta < (1 - \frac{1}{\rho})g$ *in* $SG(\mathsf{Prot})$. *Such a protocol* Prot *is called* risk-balanced.

Now we are ready to state the main theoretical result of the paper:

Theorem 1. *For any risk-balanced protocol* Prot*, there are Nash equilibria in* $SG(\mathsf{Prot})$*, and for each such Nash equilibrium* (S_A, S_B) *the following holds:*

$$\Delta_U(S_A, S_B) < (\rho - \frac{1}{\rho})g.$$

Proof. Suppose Prot satisfies Δ-condition, then we have:

$$\Delta_U(\mathcal{DH}_A, \mathcal{DH}_B) = |\rho b - a - c - (\rho a - b - c)| = (\rho + 1)\Delta < (\rho - \frac{1}{\rho})g$$

Now, since $\Delta_U(\mathcal{H}_A, \mathcal{H}_B) = |(\rho - 1)g - (\rho - 1)g| = 0$, we only need to prove the following two claims to prove the theorem:

1. Under the $\Delta-$condition, $(\mathcal{H}_A, \mathcal{DH}_B)$ and $(\mathcal{DH}_A, \mathcal{H}_B)$ are not the Nash equilibria of $SG(\mathsf{Prot})$.
2. Either $(\mathcal{H}_A, \mathcal{H}_B)$ or $(\mathcal{DH}_A, \mathcal{DH}_B)$ is a Nash equilibrium of $SG(\mathsf{Prot})$.

Proof of (1): Suppose $(\mathcal{H}_A, \mathcal{DH}_B)$ is a Nash equilibrium of $SG(\mathsf{Prot})$, then according to the definition of Nash equilibrium we have:

$$\begin{cases} \text{Utility}_A(\mathcal{H}_A, \mathcal{DH}_B) \geq \text{Utility}_A(\mathcal{DH}_A, \mathcal{DH}_B) \\ \text{Utility}_B(\mathcal{H}_A, \mathcal{DH}_B) \geq \text{Utility}_B(\mathcal{H}_A, \mathcal{H}_B) \end{cases}$$

namely,

$$\begin{cases} -a \geq \rho b - a - c \\ \\ \rho a - c \geq (\rho - 1)g \end{cases} \Rightarrow \rho a - \rho b \geq (\rho - 1)g$$

It follows that $\Delta \geq (1-\frac{1}{\rho})g$, contradicting the Δ-condition. For the case of $(\mathcal{DH}_A, \mathcal{H}_B)$, proof goes likewise.

Proof of (2): Suppose $(\mathcal{H}_A, \mathcal{H}_B)$ is not a Nash equilibrium, then either A or B can be better off by switching to a dishonest strategy, given that the opponent sticks to the honest strategy. Without loss of generality, we assume A can get higher utility by switching from $\mathcal{H}_A$ to $\mathcal{DH}_A$, namely, $\rho b - c > (\rho - 1)g$. Since $(\rho - 1)g > 0$ then $\rho b - c - a > -a$. It follows that $\mathcal{DH}_A$ is the strictly dominant strategy for A. Given that A chooses $\mathcal{DH}_A$, we argue that B will also choose $\mathcal{DH}_B$ as follows: $\text{Utility}_B(\mathcal{DH}_A, \mathcal{DH}_B) - \text{Utility}_B(\mathcal{DH}_A, \mathcal{H}_B) = \rho a - b - c - (-b) = \rho a - c \geq \rho(b - \Delta) - c = \rho b - c - \rho\Delta > (\rho - 1)g - (\rho - 1)g = 0$. It follows that $(\mathcal{DH}_A, \mathcal{DH}_B)$ is a Nash equilibrium.

Suppose $(\mathcal{DH}_A, \mathcal{DH}_B)$ is not a Nash equilibrium, then either A or B can be better off by switching to a honest strategy. Without loss of generality, we assume that A can get higher utility by switching from $\mathcal{DH}_A$ to $\mathcal{H}_A$. Therefore $-a > \rho b - a - c$, namely $0 > \rho b - c$. It follows that $\mathcal{H}_A$ is the strictly dominant strategy for A. Given that A chooses $\mathcal{H}_A$, B will also choose $\mathcal{H}_B$ since $\text{Utility}_B(\mathcal{H}_A, \mathcal{H}_B) - \text{Utility}_B(\mathcal{H}_A, \mathcal{DH}_B) = (\rho - 1)g - (\rho a - c) \geq (\rho - 1)g - (\rho(b + \Delta) - c) = (\rho - 1)g - (\rho b - c + \rho\Delta) > (\rho - 1)g - (\rho - 1)g = 0$. Therefore, $(\mathcal{H}_A, \mathcal{H}_B)$ is a Nash equilibrium. □

Remark 1. According to theorem 1, under the $\Delta-$condition, Δ_U is either 0 or $(\rho+1)\Delta$. A robust protocol would minimise Δ_U independent of ρ and g, by guaranteeing $\Delta = 0$, namely $a = b$, which implies $\Delta_U = 0$.

4 A Fair Risk-Balanced Exchange Protocol

In this section, inspired by the confidential secret comparison protocol of [13], we design two exchange protocols that rely on TTPs. The first one, undesirably, violates the Δ-condition, serving as a concrete example for motivating risk-balanced protocols. Then, we propose a protocol which, under certain conditions, is risk-balanced.

Notations. We assume that each two parties X and Y share a secret symmetric key $\mathcal{K}(XY)$. [4] We write $[M]_{\mathcal{K}}$ for the encryption of M with key $\mathcal{K}$. It is assumed that the participants have access to a secure encryption algorithm, and a one-way collision-resistant hash function h. Agents A and B are the players of our protocols, whom we assume share a secret nonce $\aleph$. The TTP is named Γ.

A fair confidential secret comparison protocol. Let $\mathcal{E}_P$, for $P \in \{A, B\}$, denote P's knowledge set. Suppose A wants to prove to B that she knows of a secret $\mathcal{I}$ (that is $\mathcal{I} \in \mathcal{E}_A$). However, if B does not already know of $\mathcal{I}$ (that is $\mathcal{I} \notin \mathcal{E}_B$), A does not want to reveal $\mathcal{I}$ to him. Moreover, A and B wish to exchange this epistemic statement "*I know* $\mathcal{I}$." mutually, and, in a fair manner.

The goal is thus to design a protocol that achieves the following (cf. [13]): (G1) Only if both A and B know $\mathcal{I}$, then A learns that B knows $\mathcal{I}$, and likewise for B. (G2) By means of the protocol, only A and B, and no one else, may learn that A or B know $\mathcal{I}$.

[4] We could as well construct our protocols based on asymmetric encryption techniques.

(G3) By means of the protocol, no one learns $\mathcal{I}$. (G4) B learns that A knows $\mathcal{I}$, iff A learns that B knows $\mathcal{I}$ (which is fairness).

To achieve these goals, we follow the straightforward approach of using on-line TTPs, e.g. see [15] (considering off-line TTPs being left as future work). Below, $\Rightarrow$ denotes communicating over confidential authenticated channels, sending a message over insecure channels is denoted by $\rightarrow$, and FTP is a secure publicly accessible server operated by Γ. We write $\Gamma \downarrow \text{FTP} : a$ when Γ makes a available on FTP.

1. $A \Rightarrow \Gamma : (f_{\mathsf{prov}}, A, B, \omega)$, where $\omega = h(\mathcal{I}, \aleph, A, B)$
2. $B \Rightarrow \Gamma : (f_{\mathsf{verif}}, A, B, \Omega_B)$, where $\Omega_B = \{h(i, \aleph, A, B) \mid i \in \mathcal{E}_B\}$
3. Γ checks if $\omega \in \Omega_B$. If yes, then $\Gamma \downarrow \text{FTP} : \omega$, else $\Gamma \downarrow \text{FTP} : \perp$.
4. A, B fetch the result from FTP.

Flags f_{prov} and f_{verif} merely indicate the purposes of the corresponding messages. It is easy to check that goals G1, G2 and G3 are achieved. In particular, Γ does not learn the content of the exchanged secret $\mathcal{I}$. Besides, using confidential channels is only to protect the content of A's message from B, and vice versa. An outsider would not benefit from observing these messages in plain, as she does not know $\aleph$. She may however observe whether an exchange is a successful comparison of *some* secret or not (cf. § 5).

The protocol is fair (G4) as B learns that A possesses $\mathcal{I}$ iff A learns that $\mathcal{I} \in \mathcal{E}_B$. Using public announcements on FTP ensures that benign communication failures cannot deprive the participants from achieving fairness. Using authenticated channels is needed to ensure the freshness of the requests. Without these, A could, e.g., compose $\omega' = h(\mathcal{I}', \aleph, A, B)$ with $\mathcal{I}' \neq \mathcal{I}$ and replay B's old message to Γ, and learn whether $\mathcal{I}' \in \mathcal{E}_B$ or not, while B not even being aware that this new comparison takes place. [5]

A severe defect of the protocol is nonetheless the uneven risk distribution that it induces. The security of this protocol obviously relies on Γ being correct. If Γ is compromised by A in the course of the protocol, then B will be seriously harmed since A (together with Γ), once getting access to Ω_B, can later on check any piece of information against $\mathcal{E}_B$ without contacting B. However, if B takes the control of Γ in his hands, then he can only check one single $\mathcal{I}$ against $\mathcal{E}_B$. This infringes on the protocol's fairness in a meta level: if A compromises Γ, the amount of harm to B is not proportional to the harm caused to A when Γ is compromised by B. Therefore, when engaging in the protocol, B takes more risk than A, hence causing $b \gg a$, if $|\mathcal{E}_B| \gg 1$ (see § 3).

A fair risk-balanced exchange protocol. Below, we propose an extension of the previous protocol that is risk-balanced. The idea is to force A to contact B for each $\mathcal{I}$ that she wants to compare against $\mathcal{E}_B$. For this purpose, we use a scheme similar to RSA encryption [12] and blind signatures. For each exchange, B randomly chooses two distinct large prime numbers p and q and computes $n = p \cdot q$ and $\phi = (p-1) \cdot (q-1)$. Then, B chooses a random number α, such that $1 < \alpha < \phi$ and $gcd(\alpha, \phi) = 1$, i.e. α and ϕ are relatively prime. B then calculates $\bar{\alpha}$ satisfying $\alpha.\bar{\alpha} \equiv 1 \bmod \phi$. Below, it is assumed that $\mathcal{E}_B$ is an ordered set, and, as before, $\mathcal{I}$ is the secret to be checked against

[5] B could learn the results of such comparisons via FTP, if he knew that he should fetch these results. Honest B must however not be forced to periodically poll the FTP server.

$\mathcal{E}_B$. We assume that $\mathcal{I}$ and elements of $\mathcal{E}_B$ can be encoded as integers smaller than n.

1. B generates n and $(\alpha, \bar{\alpha})$ as described above. B then computes $\pi = h(\omega_1, \cdots, \omega_\ell)$, where $\omega_j = h(i_j^{\bar{\alpha}} \bmod n)$, when $\mathcal{E}_B = \{i_1, \cdots, i_\ell\}$.
2. $B \rightarrow A : \alpha, n$
3. A generates a random number $\lambda < n$ such that $gcd(\lambda, n) = 1$.
4. $A \rightarrow B : (\mathcal{I} \cdot \lambda^{\alpha}) \bmod n$
5. $B \rightarrow A : (\mathcal{I} \cdot \lambda^{\alpha})^{\bar{\alpha}} \bmod n,\ \pi$
6. A computes $((\mathcal{I} \cdot \lambda^{\alpha})^{\bar{\alpha}} \lambda^{-1}) \bmod n = \mathcal{I}^{\bar{\alpha}} \bmod n$. Then A lets $\omega = h(\mathcal{I}^{\bar{\alpha}} \bmod n)$.
7. $A \rightarrow \Gamma : [f_{\mathsf{prov}}, A, B, \omega, \pi]_{\mathcal{K}(A\Gamma)}$
8. $B \rightarrow \Gamma : [f_{\mathsf{verif}}, A, B, \Omega_B]_{\mathcal{K}(B\Gamma)},$ where $\Omega_B = \{\omega_1, \cdots, \omega_\ell\}$
9. Γ checks whether π corresponds to Ω_B. If yes then
   Γ checks whether $\omega \in \Omega_B$. If yes, then
   $\Gamma \downarrow \text{FTP} : \omega,$ and A, B fetch the result from FTP.
   else
   $\Gamma \downarrow \text{FTP} : \perp,$ and A, B fetch the result from FTP.

It can be checked that this protocol satisfies G1, G2, G3 and G4. Note that authenticated channels are not used in this protocol. This is because, differently from the previous protocol, the freshness of the messages need not be checked by the TTP, since to replay A's message, B would need to construct another set $\mathcal{E}'_B$ with the same π value as of $\mathcal{E}_B$, which is infeasible as h is collision-resistant. Similarly, to replay B's message, A would need to contact B to compute $\mathcal{I}'^{\bar{\alpha}}$ for a new $\mathcal{I}'$, hence giving B the choice to use a new $\bar{\alpha}$ or decline the exchange altogether.

Concerning risk balance, if A compromises Γ, then she can cheat on B with computing $\omega \in \Omega_B$ without informing B of the result. However, to check another secret $\mathcal{I}' \neq \mathcal{I}$ against Ω_B she needs to contact B. Similarly, if B compromises Γ, he can only cheat on A by computing $\omega \in \Omega_B$ without informing A of the result. The risks induced on A and B are thus equal, given that losing one piece of information causes the same harm from both A's and B's points of view. In this case, we have $a = b$ (see § 3), implying $\Delta_U = 0$, hence the protocol being risk-balanced.

To summarise, if Γ is not compromised, then the protocol satisfies G1, G2, G3 and G4. In case Γ is compromised, the protocol may not achieve G4 anymore. Rational agents will however end up with equal utilities even when Γ is compromised. In other words, the amount of expected harm to a cheated B would be limited and proportional to the damage that B could cause to A if Γ was compromised by B, and vice versa.

5 Discussions

We motivate why the values of $\mathcal{I}^{\bar{\alpha}}$ and $i^{\bar{\alpha}},\ i \in \mathcal{E}_B$ need to be hashed in our risk-balanced protocol. We assume that these values were not hashed and, thereby, demonstrate an attack on the protocol which undermines its risk balance.

Let us assume that $\omega = \mathcal{I}^{\bar{\alpha}}$ and $\omega_j = i_j^{\bar{\alpha}}$, all computed modulo n, thus removing the hash function from the protocol. The idea of the attack is that if A compromises Γ, then she gets access to the members of $\mathcal{E}_B$. This is because A knows α (message 2 above) and with compromising Γ, she gets access to $\{i_j^{\bar{\alpha}} \mid i_j \in \mathcal{E}_B\}$, from which

she would be able to compute $\{(i_j^{\bar{\alpha}})^{\alpha} \bmod n \mid i_j \in \mathcal{E}_B\} = \mathcal{E}_B$. We conclude that in case hash functions were not used in the protocol, A, by compromising Γ, could cause more damage to B, compared to the damage B could cause to A by compromising Γ (compare with the first protocol of section 4). This can undermine the protocol's risk balance, and, thus has to be prevented.

Below, we mention two shortcomings of our risk-balanced protocol. Addressing these issues constitute our future work. (1) We note that in the protocol, Γ would always learn whether the exchange was successful or not (outsider parties can easily be prevented from seeing the result altogether, e.g. using encryption), although the shared information $\mathcal{I}$ is not revealed to Γ. Leaking this little information can in principle be harmful to the participants: An interrogator who knows that you share some secret with a comrade would be hard to thwart before you both reveal that very secret. Hiding this information from Γ remains to be studied. (2) A drawback of the protocol is its communication costs and the computation burden it imposes on Γ. The computation cost on B is also much heavier than A. Equivalent protocols with less, and evenly distributed, computation and communication costs are thus desirable.

Acknowledgements. We are grateful to Wouter Teepe for many helpful discussions, and to Srijith Nair for commenting on an earlier version of the paper.

References

1. Asokan, N.: Fairness in electronic commerce. PhD thesis, University of Waterloo (1998)
2. Buttyán, L., Hubaux, J.: Toward a formal model of fair exchange – a game theoretic approach. Technical Report SSC/1999/39, EPFL, Lausanne (1999)
3. Buttyán, L., Hubaux, J., Capkun, S.: A formal model of rational exchange and its application to the analysis of syverson's protocol. J. Computer Security 12(3-4), 551–587 (2004)
4. Cachin, C., Camenisch, J.: Optimistic fair secure computation. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 93–111. Springer, Heidelberg (2000)
5. Chadha, R., Mitchell, J., Scedrov, A., Shmatikov, V.: Contract signing, optimism, and advantage. In: Amadio, R.M., Lugiez, D. (eds.) CONCUR 2003. LNCS, vol. 2761, pp. 366–382. Springer, Heidelberg (2003)
6. Even, S., Yacobi, Y.: Relations among public key signature systems. Technical Report 175, Computer Science Dept., Technion, Haifa, March (1980)
7. Fagin, R., Naor, M., Winkler, P.: Comparing information without leaking it. Commun. ACM 39(5), 77–85 (1996)
8. Gollmann, D.: Why trust is bad for security. ENTCS 157(3), 3–9 (2006)
9. Imamoto, K., Zhou, J., Sakurai, K.: An evenhanded certified email system for contract signing. In: Qing, S., Mao, W., Lopez, J., Wang, G. (eds.) ICICS 2005. LNCS, vol. 3783, pp. 1–13. Springer, Heidelberg (2005)
10. Ito, C., Iwaihara, M., Kambayashi, Y.: Fair exchange under limited trust. In: Buchmann, A.P., Casati, F., Fiege, L., Hsu, M.-C., Shan, M.-C. (eds.) TES 2002. LNCS, vol. 2444, pp. 161–170. Springer, Heidelberg (2002)
11. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press, Redmond, Washington (1999)
12. Rivest, R., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM 21(2), 120–126 (1978)

13. Teepe, W.: Reconciling Information Exchange and Confidentiality — A Formal Approach. PhD thesis, Rijksuniversiteit Groningen (2006)
14. Traore, J., Boudot, F., Schoenmakers, B.: A fair and efficient solution to the socialist millionaires' problem. Discrete Applied Mathematics 111, 23–36 (2001)
15. Zhou, J., Gollmann, D.: A fair non-repudiation protocol. In: Security and Privacy 1996, pp. 55–61. IEEE Computer Society Press, Los Alamitos (1996)