

FIDES – A Middleware E-Commerce Security Solution

Aleksandra Nenadić, Ning Zhang, Stephen Barton
Department of Computer Science, University of Manchester
Oxford Road, Manchester M13 9PL, UK
{anenadic, nzhang, s.k.barton}@cs.man.ac.uk

Abstract

This paper reports on the on-going Fair Integrated Data Exchange Services (FIDES) project aimed at developing a security middleware solution to support e-commerce transactions and the provision of the important fair exchange and non-repudiation security services. *Fair exchange* ensures that either both business parties participating in a transaction receive the exchanged valuable items or neither party receives anything useful. *Non-repudiation* ensures that neither party involved in the exchange can falsely deny sending or receiving a particular item and therefore taking part in the transaction.

Keywords: E-commerce, Security, Fair exchange, Non-repudiation.

1 Introduction

While e-commerce will clearly have a big impact on the way people will conduct business in the future, one of the most important obstacles to further expansion of e-commerce has been the lack of adequate security protections. As Computer Crime and Security Survey (2002), conducted by the Computer Security Institute and FBI International Crime Squad, revealed - financial losses from Internet attacks were on the rise third year in a row and 90% of the survey respondents (primarily large corporations and government agencies) reported security breaches.

Security threats to e-commerce transactions come not only from external attackers, who may attempt to eavesdrop or modify the messages, or act under false identity, but also from insiders, i.e. misbehaving business partners. The Internet has enabled companies and organizations to establish ad-hoc business relations with parties whom they may have never met before and, therefore, there is a lack of trust among business partners and disputes are likely to occur.

In order to mitigate the risks associated with conducting e-commerce transactions and help establish trust among potential business partners, adequate security services should be in place to ensure that exchanges of valuable business items are performed *fairly* and that evidence of e-transactions cannot be *repudiated*. Such services should prevent situations where one party has received the expected item while the other has not (fairness), and protect business partners against false denials that a particular item has been sent (non-repudiation of origin) or received (non-repudiation of receipt). For instance, in an e-purchase process, a merchant should receive a buyer's e-payment if and only if the purchased e-goods are delivered to the buyer as promised. Alternatively, the buyer should obtain some evidence, such as an e-receipt, certifying that he has made the payment, and this receipt can assure the buyer that the goods will be delivered. Additionally, important electronic business correspondence requires a certified e-mail delivery service analogous to recorded/certified mail provided by a conventional Post Office to assure the sender that the recipient receives his e-mail if and only if he receives the e-receipt signed by the receiver. Furthermore, in the process of electronic contract signing, business parties need to exchange their digital signatures fairly to avoid the situation where one party is legally bound to the contract, while the other is taking his time to look for a better offer and later withdraw from the deal.

The Fair Integrated Data Exchange Services project (FIDES), sponsored by DTI/EPSRC, was launched in September 2001 with an aim of developing a security middleware solution to support e-commerce transactions. The core part of the FIDES is a family of novel and

efficient security protocols for achieving fair exchange and non-repudiation security services. The protocols facilitate exchanges of various business data types (digital signatures, e-goods, e-payments, etc.) and fairness is guaranteed through the use of an *off-line* and *transparent* semi-trusted third party (STTP). The services of the STTP are invoked only in extreme circumstances, e.g. when the normal exchange process cannot complete successfully due to unfair behaviour of participants or a network failure. The exchanged items enjoy the *confidentiality* protection against the STTP, should it be invoked. The protocols reduce the amount of trust placed on the STTP and impose low communication and computational overheads on the participants, which makes them suitable for implementation in both wired and wireless networks. Mutual authentication between protocol participants and message integrity and privacy protections can be achieved through other existing and standard mechanisms, e.g. SSL. This decoupling contributes to modularity and flexibility of the FIDES system and its ability to be integrated with various other security mechanisms.

The rest of the paper is organized as follows. Section 2 provides a brief overview of the solutions for fairness and non-repudiation. Section 3 presents security requirements for the FIDES protocol family and their main design principles. The FIDES protocol family is described in detail in Section 4. The FIDES system architecture and implementation details are presented in Section 5. Finally, Section 6 outlines our conclusions.

2 A Short History of Fairness and Non-repudiation

Achieving fair exchange over the Internet is quite different from that in non-electronic world. In conventional world, the exchange of valuable items is performed simultaneously in order to achieve fairness. For instance, a customer is paying for the goods at the time of receiving them, and business parties are physically present at the same place and sign the contract roughly at the same time. On the other hand, it is physically impossible to achieve simultaneous exchange over the Internet due to serial nature of the underlying network. In such circumstances, one party is forced to send his item first and thereby may get into a disadvantageous position.

Non-repudiation is a special case in a broader problem of fair exchange. More specifically, it can be considered as fair exchange of an item for a digital signature on the item. Digital signatures provide a mechanism for establishing the authenticity and integrity of a message and the identity of its originator. Therefore, the recipient's digital signature on the received item is considered as a non-repudiable acknowledgement of the reception of the item. In addition, exchange of digital signatures has become a common practise for electronic contract signing. The legal use of digital signatures on the Internet has been regulated by EU Electronic Signature Directive (1999).

Solutions for fair exchange have evolved from the two-party approach, in which the participants perform an exchange without any involvement of a third party, to the trusted third party (TTP) approach, in which a TTP is involved to help the participants with the exchange and achieving fairness.

Two-party protocols (e.g. Blum 1983, Even et al. 1985) are based on gradual exchange of small parts of items to ensure that the exchange occurs pseudo-simultaneously and that neither party can obtain substantial advantage over the other. One way of achieving this is to have the participants release their items bit-by-bit in an interleaving manner. However, this approach has some serious shortcomings: (1) the exchanged items must have the same number of bits to guarantee fairness, (2) a large number of rounds of communication is required to exchange and verify all the bits, (3) participating parties are required to have approximately equal computational power, (4) and there are no guarantees of the quality of

the items reassembled at the end from the received bits. Although reasonably convincing in theory, this approach is too impractical for real-life applications.

On the other hand, relying on a TTP to mediate the exchange process is a common practise in traditional transactions - Post Office is a third party trusted to deliver recorded or certified mail and obtain a receipt from the receiver. Similarly, contracts are often negotiated and signed through a third party solicitor. According to Pagnia and Gärter (1999), there is no strong fair exchange protocol tolerant against misbehaving participants without a TTP. Although this theory seems to contradict the two-party approach, in gradual exchange protocols there is always one (last) bit that cannot be exchanged fairly, and, although it cannot cause too much damage, it proves that the above theory holds.

The degree of the TTP's involvement in this class of protocols varies – earlier protocols were relying on an *intermediary* or *in-line* TTP (Bahreman and Tygar 1994, Deng et al. 1996, Zhou and Gollmann 1996b, etc.), which collects the exchanged items from the participating parties, checks their correctness and forwards them to the corresponding recipients. Improvements in reducing the TTP's involvement have resulted in the advent of *on-line* TTPs (Schneier and Riordan 1998, Zhang and Shi 1996, Zhou and Gollmann 1996a, etc.), which help by validating, generating and storing the evidence of transactions. Still, both in-line and on-line TTPs have to be involved in each protocol run and their availability is crucial for the functioning of the protocols. They also have the full access to the exchanged items so the privacy of the items violated. Therefore, these TTPs are potential performance and security bottlenecks. A big step towards more efficient solutions was the introduction of *off-line* TTPs that intervene only in case of dispute caused by a network failure or a party's misbehaviour (Asokan et al. 2000, Bao et al 1998, Boyd and Foo 1999, Chen 1998, Ray and Ray 2000, Zhang and Shi 2003, Zhou and Gollmann 1997, etc.). The rest of the time, when the network functions well and participants behave correctly or are capable of resolving the disputes themselves, the off-line TTP does not operate in the protocol execution.

Our research is focused on devising protocols with further reduced requirements and trust placed on, and the role played by, the off-line TTP. Therefore, the third party in our protocols is called *semi-trusted third party* (STTP).

3 Preliminaries

In this section, we first describe the e-transaction model used for the FIDES protocols, and then summarise the security requirements satisfied by the FIDES solution.

3.1 E-transaction Model

In general, e-transactions can be decomposed into several stages (Fig. 1). In the first stage, business parties mutually authenticate each other and agree on a session key that will be used to protect the subsequent communication. In the second stage, business parties negotiate the content of business items to be exchanged, e.g. e-payments, contracts, e-goods, etc. The actual exchange of the agreed items takes place during the third, i.e. *execution*, stage. The FIDES protocols are executed during the execution stage (solid lines in Fig. 1), i.e. they deal only with the actual exchanges of the agreed business items and resolving possible disputes that may occur in the process. They do not mandate any particular mechanisms to be used in the first and second stage (dashed lines in Fig. 1).

We assume that business parties P_a and P_b may not trust each other, and either of them may misbehave in an attempt to gain the other party's item without giving out his own one. They have agreed to employ an off-line STTP P_t to help with the exchange process if they cannot reach a fair completion themselves. It is assumed that P_t may misbehave by

attempting to access the exchanged items, but P_t does not conspire with either of P_a and P_b . Otherwise, any such collusion can be exposed, and, consequently, P_t will be discredited.

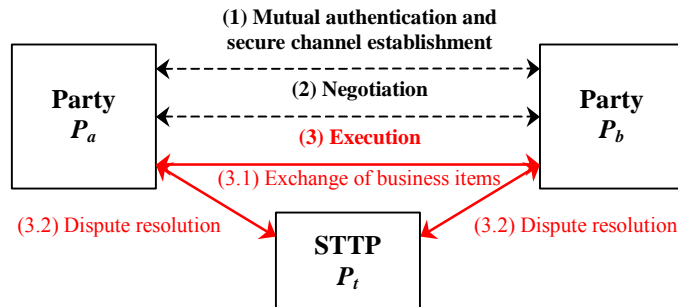


Figure 1. General e-transaction model

We also assume the existence of a Certification Authority (CA) in the model, which issues public-key certificates to the participants. The FIDES protocols are public-key based and can be divided into two classes according to the type of public-key algorithm they use. The protocols support the use of both RSA (Rivest et al. 1978) and DSA (FIPS 186-2), as both are widely recognised by e-commerce community. RSA is a *de facto* commercial standard for encryption, has been specified by ISO/IEC 9796 for the use in digital signatures, and has been built into many standards and commercial products, such as S/MIME, PGP, SSL/TLS, PEM, etc. The National Institute of Standards and Technology (NIST) proposes DSA for the use in Digital Signature Standard (DSS), which is the official digital signature standard in the United States.

3.2 Security Requirements

FIDES has been designed to satisfy the following security requirements.

(S1) *Strong fairness*: FIDES fair exchange protocols guarantee that, by the end of an exchange process, if one party has obtained the other party's item or can obtain it with the assistance of the STTP, then the other party has obtained this party's item or can obtain it with the assistance of the STTP.

(S2) *Non-repudiation*: FIDES certified delivery protocols guarantee that, by the end of the exchange process, the recipient will be in possession of an unforgeable and non-repudiable proof that the sender has indeed originated the item (*non-repudiation of origin*), and the sender will be in possession of a similar proof that the recipient has indeed received the item (*non-repudiation of receipt*).

(S3) *Confidentiality of the exchanged items*: No party external to the exchange process, including the STTP, will gain any knowledge of the exchanged items.

(S4) *E-goods content/quality assurance*: For certified e-goods delivery or e-goods purchase, the receiver of the e-goods is able to verify that the item he is to receive will indeed match with the promised content/quality, as, otherwise, a mismatch between the promised/expected and received e-goods may have financial implications to the receiver.

(S5) *Reduced role of the STTP*: Security, computational and storage requirements placed on the STTP are reduced as much as possible to simplify its implementation and management and increase the security of FIDES, as the STTP may be a focal point of security and denial-of-service attacks.

(S6) *Transparency of the STTP*: Participation of the STTP in an e-transaction is transparent in the sense that the items recovered by the STTP are indistinguishable from those sent by the original senders. This can be a desirable property in situations where the STTP is invoked due

to a network failure or system crash rather than unfair behaviour of participants, which may bring bad publicity to them.

4 The FIDES Protocol Family

In this section, we describe the general structure of, and the cryptographic primitives used in, the FIDES protocol family design. More detailed descriptions of some of the FIDES protocols can be found in (Shi et al.2003, Nenadic et al. 2004a, Nenadic et al. 2004b, Nenadic et al. 2004c).

The FIDES protocols have a common structure, although they differ in the types of business items and public-key algorithms supported. They can be applied to exchanges of two types of business items: confidential e-goods (content/quality of which has been certified by an independent certification authority) and digital signatures. The following approach is taken when one of the exchanged items is an e-goods - the e-goods is firstly encrypted with a symmetric key and transferred to the recipient, and then the suitable protocol is invoked for the exchange of the decryption key and the other party's item. The decryption key is linked to the encrypted e-goods through a specialised certificate issued by a certification authority that verifies and guarantees the content/quality of the encrypted e-goods. For instance, if the e-goods is an e-check, this independent authority can be a bank that has issued the e-check; if the e-goods is Windows 2000 software, Microsoft itself may certify its quality. When the exchanged item is a digital signature, no symmetric key encryption is applied.

The main cryptographic primitives utilised in the design of the protocols are Verifiable Encryption (VE) of a key/signature and Verifiable and Recoverable Encryption (VRE) of a key/signature. For both VE and VRE, the receiver can verify that the encryption indeed contains the correct key/signature. For VRE, the receiver can additionally verify that a designated STTP can recover the encrypted item from its encryption. The designs of RSA-based VE and VRE for keys and signatures are summarised in Table 1. DSA-based primitives have been omitted due to space limitation and will be published separately.

Each protocol suite consists of a pair of protocols - an *exchange* protocol performed by business parties and a *recovery* protocol involving a STTP. At a high level, the protocols work as follows (Fig. 2).

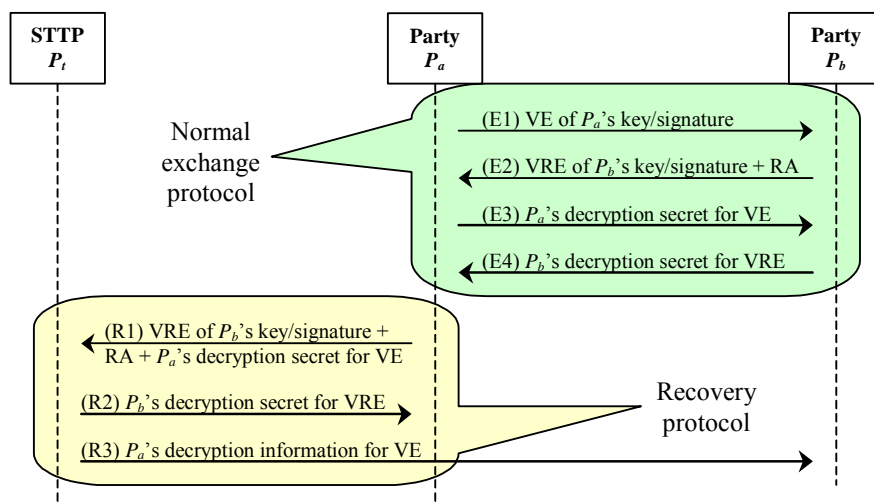


Figure 2. The FIDES protocols framework

Normal exchange protocol:

(E1): P_a generates VE of his item (key or signature) using a secret and transfers the VE to P_b .

(E2): P_b can verify the correctness of P_a 's VE, but, at this point, P_b can learn no additional information about P_a 's item. If P_b is satisfied with this verification, he uses his secret to generate VRE of his item, and, in addition, produces a Recovery Authorisation (RA) token, which authorises P_a to request the recovery of P_b 's VRE from STTP P_t if certain conditions are met. The RA token is interpreted as follows: P_t will recover P_b 's secret from VRE for P_a (which will enable P_a to gain P_b 's item from VRE), if and only if P_a provides P_t with his secret, which will allow P_b to decrypt P_a 's VE. P_b transfers his VRE and the RA token to P_a .

(E3): P_a verifies the correctness of P_b 's VRE and the RA token, and, if satisfied, P_a is convinced that it is secure for him to release his secret first, which will enable P_b to decrypt VE to obtain P_a 's item.

(E4): If P_b receives P_a 's secret correctly, it transfers his secret to P_a . At this point, if P_a is satisfied with the verification outcome of P_b 's decryption secret, the exchange protocol is completed successfully and P_a uses the received secret to decrypt VRE and obtain P_b 's item. Otherwise, if this final verification fails or P_a fails to receive anything from P_b 's altogether, P_a can request P_t for the recovery of P_b 's decryption secret, by invoking the recovery protocol.

Recovery protocol:

(R1): P_a transfers P_b 's VRE and the RA token and his decryption secret to P_t . P_t verifies the correctness of these items, and, if satisfied, P_t recovers P_b 's decryption secret from P_b 's VRE.

(R2): P_t sends P_b 's decryption secret to P_a who uses it to decrypt P_b 's item from VRE.

(R3): P_t also sends P_a 's decryption secret to P_b to ensure fairness.

Table 1. RSA-based cryptographic primitives

- $E_k(x)$ denotes ciphertext of a data item x encrypted with a symmetric key k ;
- $h(x)$ is a one-way strong-collision-free hash function;
- x, y denotes the concatenation of data items x and y ;
- $pk_i = (e_i, n_i)$ and $sk_i = (d_i, n_i)$, $i \in \{a, b, t\}$: P_i 's RSA public and private key, with n_i public modulus;
- $h(x)^{d_i} \bmod n_i$: P_i 's RSA signature on data item x ;
- $C_{bt} = (P_b, pk_{bt}, w_{bt}, s_{bt})$: certificate issued by P_t for P_b 's additional RSA public/private key pair $pk_{bt} = (e_{bt}, n_{bt})$, $sk_{bt} = (d_{bt}, n_{bt})$, where n_{bt} is RSA modulus chosen by P_t and $e_{bt} = e_b$. Number w_{bt} is defined as $w_{bt} = (h(sk_t, pk_{bt})^{-1} \times d_{bt}) \bmod n_{bt}$, and s_{bt} is P_t 's signature on the items (P_b, pk_{bt}, w_{bt}) ;
- k_i , $i \in \{a, b\}$: P_i 's symmetric key for encryption/decryption of e-goods D_i ;
- $RSA-EGCert_i = (desc_i, hd_i, ek_i, sign_{it})$, $i \in \{a, b\}$: RSA-based e-goods certificate issued by P_t linking encrypted P_i 's e-goods D_i with its secret decryption key k_i , where $desc_i$ is e-goods description, $hd_i = h(E_{k_i}(D_i))$, $ek_i = k_i^{e_i} \bmod n_i$, and $sign_{it}$ is P_t 's RSA signature on the items $(desc_i, hd_i, ek_i)$;
- r_a, r_b : P_a 's and P_b 's secret random numbers used to generate VE and VRE, respectively;

| VE of P_a 's key k_a | VE of P_a 's signature |
|---|--|
| <p>Generation: $y_a = r_a^{e_a} \bmod n_a$; $x_a = (r_a \times k_a) \bmod n_a$;</p> <p>Verification: $x_a^{e_a} \bmod n_a \stackrel{?}{=} (y_a \times ek_a) \bmod n_a$;</p> | <p>Generation: $y_a = r_a^{e_a} \bmod n_a$; $x_a = (r_a \times (h(x))^{d_a}) \bmod n_a$;</p> <p>Verification: $x_a^{e_a} \bmod n_a \stackrel{?}{=} (y_a \times h(x)) \bmod n_a$;</p> |

| VRE of P_b 's key k_b | VRE of P_b 's signature |
|---|---|
| Generation: $y_b = r_b^{e_b} \bmod (n_b \times n_{bt});$ $x_b = (r_b \times k_b^{d_b}) \bmod n_b;$ $xx_b = (r_b \times h(y_b)^{d_{bt}}) \bmod n_{bt};$ | Generation: $y_b = r_b^{e_b} \bmod (n_b \times n_{bt});$ $x_b = (r_b \times h(x)^{d_b}) \bmod n_b;$ $xx_b = (r_b \times h(y_b)^{d_{bt}}) \bmod n_{bt};$ |
| Verification: $x_b^{e_b} \bmod n_b \stackrel{?}{=} (y_b \times ek_b) \bmod n_b;$ $xx_b^{e_b} \bmod n_{bt} \stackrel{?}{=} (y_b \times h(y_b)) \bmod n_{bt};$ | Verification: $x_b^{e_b} \bmod n_b \stackrel{?}{=} (y_b \times h(x)) \bmod n_b;$ $xx_b^{e_b} \bmod n_{bt} \stackrel{?}{=} (y_b \times h(y_b)) \bmod n_{bt};$ |
| Recovery by P_i: $d_{bt} = (h(sk_i, pk_{bt}) \times w_{bt}) \bmod n_{bt};$ $r_b = (y_b \bmod n_{bt})^{d_{bt}} \bmod n_{bt};$ | |
| P_b's Recovery Authorization (RA) token: P_b's RSA signature on items C_{bt}, y_b, y_a, P_a: | |

5 The FIDES System

The FIDES system is fully implemented in Java and the high level overview of its architecture is shown in Fig. 3. For each enterprise it consists of a FIDES Server and a set of FIDES Clients. FIDES STTP Servers are assumed for inter-enterprise dispute resolution, i.e. for the execution of the recovery protocols. The FIDES Server is the core of the system through which business users from an enterprise access the functionality of the FIDES services. It listens to both internal requests from within-enterprise business users, as well as external transaction requests from its business partners. Business users use GUI-based FIDES Clients to securely access the services on the FIDES Server. Communications between a FIDES Client and its Server, and between any two FIDES Servers (including FIDES STTP) is carried out through Java Messaging Service (JMS).

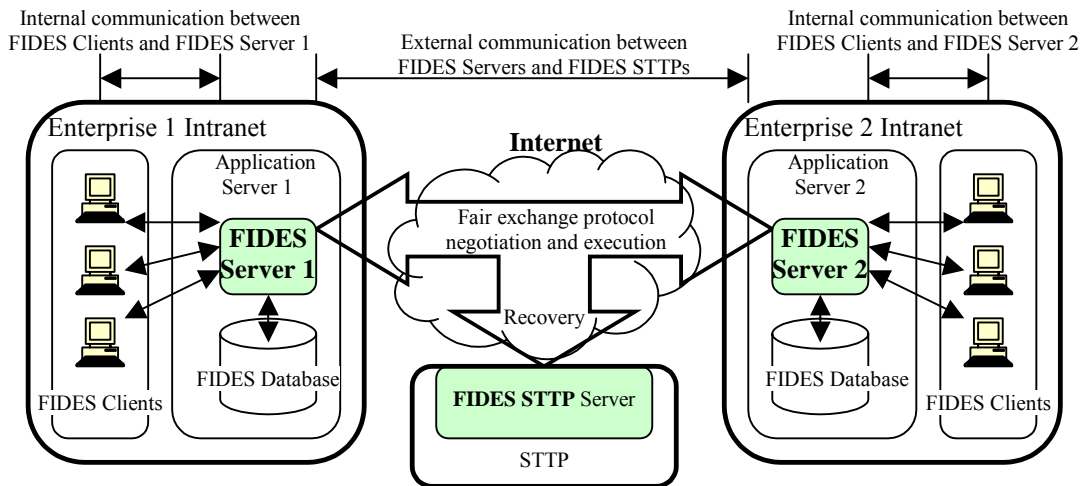


Figure 3. FIDES system architecture

An enterprise runs its own FIDES Server. It is assumed that the items to be exchanged, such as contracts and e-goods, have been previously negotiated between business partners and securely stored in a central database connected to each FIDES Server. Information regarding enterprise business users and their FIDES service access rights, business partners whom the enterprise has business transactions with, STTPs trusted by the enterprise to help

with dispute resolution, and records of all business transactions executed through the FIDES system are all stored in the central database and maintained by the FIDES Server.

Using the FIDES system, Enterprises 1 and 2 may fairly exchange their valuable business items through the following process. An authorised business user from Enterprise 1 uses a FIDES Client installed on his machine to specify which item (previously negotiated and stored in the FIDES database) is to be sent and which item is expected in return from Enterprise 2. He may also specify a preferred timeout for the transaction (otherwise a default timeout is used). This specification is sent to the FIDES Server 1 that, upon authenticating and authorizing the user's request, initiates a negotiation with FIDES Server 2 from Enterprise 2. During this phase, the two Servers further negotiate the transaction details, including a unique transaction identifier, the exchange protocol to be used, a mutually trusted STTP for possible dispute resolution, timeout value, etc. After this initial negotiation, FIDES Server 2 forwards this transaction request to authorised business users at Enterprise 2, as the Server itself should not automatically accept transaction requests without human intervention or without prior auto-configuration. Using his FIDES Client, an authorised user from Enterprise 2 examines the list of transactions that are awaiting confirmation. If the request from Enterprise 1 is accepted, the transaction will be executed through the negotiated protocol. Each FIDES Server stores the items exchanged together with the transactional records in its database, and business users involved are notified of the outcome. If the transaction fails for any reason, the Server attempts to automatically resolve it with the help of the agreed STTP. If the STTP is unavailable at that moment, business users are notified and the transactional evidence can be exported to a disk and the resolution by the STTP can be performed manually using the evidence file. Alternatively, automated resolution may be re-invoked at a later time.

5.1 FIDES Server

Figure 4 shows the main components of the FIDES Server architecture - the Transaction Manager, FIDES Protocol Library, Crypto Library, JMS and Secure Storage.

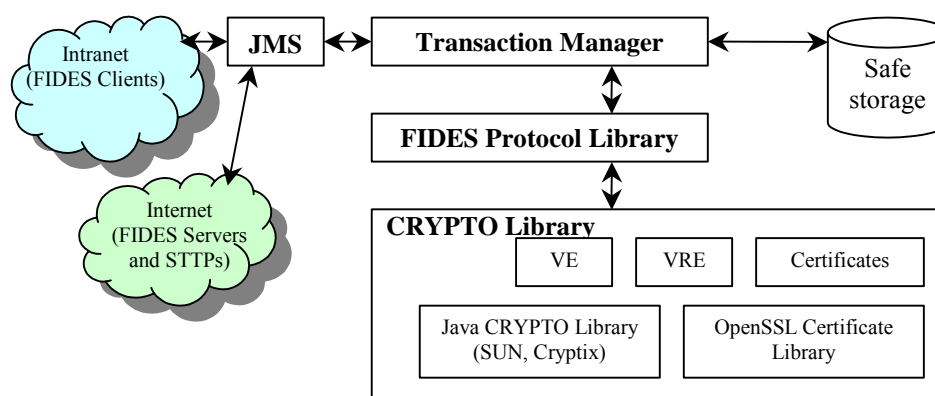


Figure 4. FIDES Server architecture

The Transaction Manager takes care of authenticating and authorising transaction specifications sent by a FIDES Client, managing and auditing transactions. It keeps the state and evidence of a transaction in persistent storage, implements the timeout and retry logic to overcome unreliable communications, and initiates transaction recovery with a STTP in presence of any failures.

The JMS component shuffles messages between different entities in the FIDES system through a JMS (Java Message Service) provider. FIDES is a provider-independent solution,

i.e. it is portable across JMS-compliant providers. So far, the application has been tested using the Sun ONE Message Queue and PrismTech's OpenFusion JMS providers.

The FIDES Protocol Library provides the core functionality for composing and verifying FIDES protocol messages. It interacts with the Transaction Manager during a protocol execution, which keeps the context and maintains the state of a transaction. This component makes the use of the cryptographic primitives provided by the Crypto Library, including VE, VRE, digital signatures, hash functions, public- and symmetric-key algorithms, etc. The motivation for separating the functionality of the FIDES Protocol Library from that of the Crypto Library is to allow easy plug-in of cryptographic methods by various JAVA cryptographic providers. We have used the OpenSSL Cryptographic Library to implement X.509 certificate issuing, and, for all the other cryptographic methods, we have used the libraries provided by Cryptix and SUN JCE.

5.2 FIDES Client

A FIDES Client provides a GUI-based application interface that allows a business user (i.e. an employee of an enterprise) to securely access the FIDES services on the FIDES Server (subject to access control policy). Upon successful authentication of the user, the Server starts a session with the Client. The Client and the Server are loosely coupled and communicate by exchanging asynchronous JMS messages, while the server keeps the track of the session. The FIDES Client provides the following services: (1) initiating transactions with business partners, (2) browsing transaction requests from business partners and accepting/rejecting them, (3) tracking all messages exchanged with the Server, (4) searching transactions, business partners and STTPs, (5) exporting transactional records to a disk, (6) adding business partners and business items to the central FIDES database, and (7) administrative tasks, including updating passwords/credentials, and, for administrators, business user management. A snapshot of the FIDES Client GUI is shown in Fig. 5.

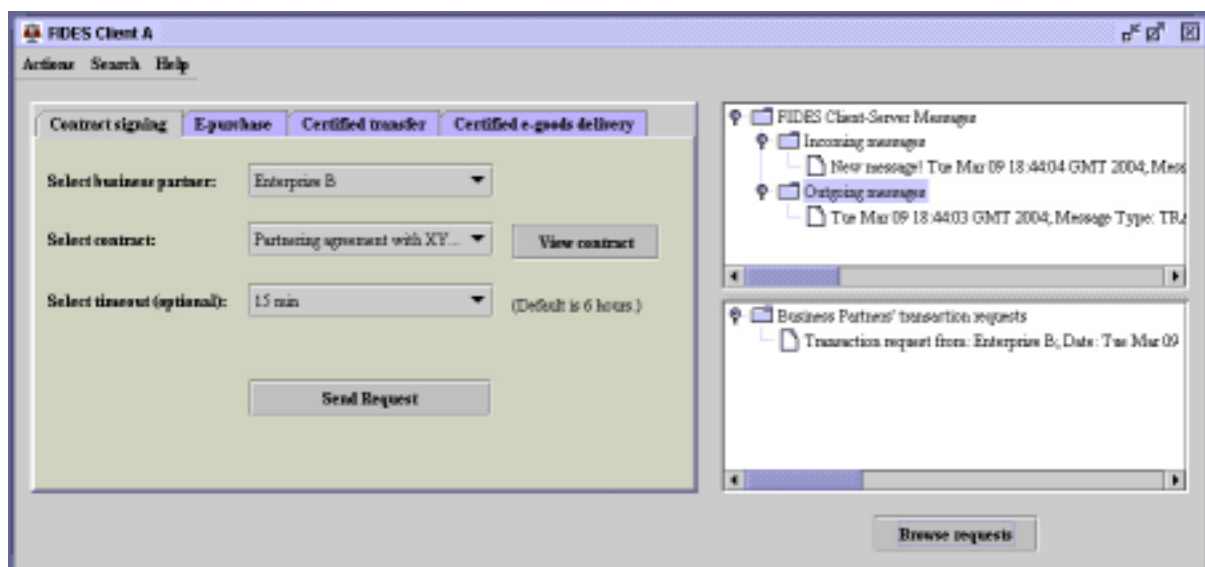


Figure 5. A snapshot of the FIDES Client GUI

5.3 FIDES STTP

The FIDES STTP Server provides an on-line facility for dispute resolution and recovery of exchanged items, in cases when a normal exchange process fails to complete successfully. If a dispute arises during an exchange process, recovery will be attempted automatically with the STTP that was negotiated between the two FIDES Servers. If automated resolution fails due to a network failure or unavailability of the STTP, business users have an option to

manually export transactional records to a disk and take/send them to the STTP for the manual recovery, or to re-initiate the automated recovery. The STTP also issues special public-key and e-goods certificates (such as certificates C_{bt} and $RSA-EGCert_i$ from Table 1), based on which the STTP recovers the disputed items. The services of a STTP in the FIDES system could be provided by established and trusted certification authorities, such as VeriSign, or banks, in cases the exchanged items contain e-payments, or specialised agencies, all of which would be required to run FIDES STTP Servers to handle the dispute resolution.

5.4 FIDES Evaluation

The FIDES system will be assessed and evaluated by conducting case studies with the help of the FIDES project business partner specialised in financial marketing and e-procurement solutions. The case studies are planned to exploit the FIDES system in three main B2B scenarios – contract signing, certified e-goods and certified e-payments delivery and to assess how well the system accomplishes the requirements from the end-user (i.e. business user) point of view. The following aspects will be considered – applicability of the system on different platforms, variety of business items and e-commerce scenarios supported, user-friendliness, convenience of use and ease of maintenance of the system, the level of security offered, the time and cost of performing transactions using the FIDES system in comparison with the traditional ways, enabling communication with geographically distant business partners, etc. The FIDES System is to be integrated with OpenFusion, a middleware solution by the FIDES project partner, in order to advance the commercial exploitation of the system, and an additional aspect of the evaluation will be focused on the integration issue.

6 Conclusions

Increasingly, enterprises and financial institutions are building their on-line presence through the Internet web sites. Although some of them are still utilising the Internet solely for advertising, more are starting to use their web sites to conduct e-commerce transactions. Fairness and non-repudiation are two key security requirements for e-commerce transactions as they protect the participants from malicious business partners, which is needed in environments where parties may conduct transactions with parties with whom they might not have previous business history or may not trust fully.

This paper has presented our FIDES solution for provision of fairness and non-repudiation security services. FIDES is a message-oriented middleware with modular and configurable architecture so that different system components can be easily replaced, e.g. cryptographic and JMS providers, authentication and confidentiality protection components, etc. The FIDES protocols support two most widely used public-key algorithms, RSA and DSA, allow an exchange of a wide range of business items and impose low security and storage requirements on the off-line and transparent STTP. They been designed and implemented as Java API and can be plugged into any e-commerce system to allow further development with little or no modification. Our future work will involve finalising the implementation of the FIDES system and running a system trial and conducting case studies with the involvement of our commercial partner.

References

- Asokan, N., Schunter M., Waidner, M. (2000) “Optimistic Fair Exchange of Digital Signatures”, IEEE Journal on Selected Areas in Communications, Vol. 18, pp593-610.
- Ateniese G. (1999) “Efficient Verifiable Encryption (and Fair Exchange) of Digital Signatures”, ACM Conference on Computer and Communications Security, pp138-146.

- Bahreman, A., Tygar, J. D. (1994) "Certified Electronic Mail", Internet Society Symposium on Network and Distributed System Security, pp3-19.
- Bao, F., Deng, R., Mao, W. (1998) "Efficient and Practical Fair Exchange Protocols with Off-line TTP", IEEE Symposium on Security and Privacy, pp77-85.
- Boyd, C., Foo, E. (1998) "Off-line Fair Payment Protocol Using Convertible Signatures", Advances in Cryptology – ASIACRYPT '98, LNCS, Springer-Verlag, Berlin, Germany, Vol. 1514, pp271-285.
- Blum, M. (1983) "How to Exchange (Secret) Keys", ACM Transactions on Computer Systems, Vol. 1, pp175-193.
- Chen, L. (1998) "Efficient Fair Exchange With Verifiable Confirmation of Signatures", Advances in Cryptology - ASIACRYPT '98, LNCS, Springer-Verlag, Berlin, Germany, Vol. 1514, pp286-299.
- Computer Security Institute and FBI International Crime Squad (2002) "Computer Crime and Security Survey", [online], <http://www.gocsi.com/>.
- Deng, R. H., Gong, L., Lazar, A.A., Wang, W. (1996) "Practical Protocols for Certified Electronic Mail", Journal of Network and System Management, Vol. 4, No. 3, pp279-297.
- National Institute of Standards and Technology (NIST) (2000) "Digital Signature Standard (DSS)", Federal Information Processing Standards (FIPS) Publication 186-2.
- Even, S., Goldreich, O., Lempel, A. (1985) "A Randomized Protocol for Signing Contracts", Communications of the ACM, Vol. 28, pp637-647.
- The European Parliament and the Council of the European Union (1999) "EU Electronic Signature Directive (Directive 1999/93/EC)", [online], http://www.ncipher.com/insights/compliance/1_eu-esignature.html.
- FIDES - Fair Integrated Data Exchange Services, [on-line], www.cs.man.ac.uk/~nenadic/FIDES/fides.html.
- Franklin, M. K., Reiter, M. (1997) "Fair Exchange with a Semi-Trusted Third Party", ACM Conference on Computer and Communications Security, pp1-5.
- Nenadic, A., Zhang, N., Barton, S. (2004) "Fair Certified E-mail Delivery", to appear in Proceedings of the ACM Symposium on Applied Computing (SAC'04).
- Nenadic, A., Zhang, N., Barton, S. (2004) "A Security Protocol for Certified E-Goods Delivery", to appear in Proceedings of International Conference on Information Technology, Coding and Computing (ITCC'04), IEEE Computer Society.
- Nenadic, A., Zhang, N., Barton, S. (2004) "A Secure and Fair DSA-based Signature Exchange Protocol", to appear in Proceedings of IEEE Symposium on Computers and Communications, IEEE Computer Society.
- Pagnia, H., Gärtner, F. (1999), "On the Impossibility of Fair Exchange without a Trusted Third Party", Technical Report TUD-BS-1999-02, University of Darmstadt, Germany.
- Ray, I., Ray, I. (2000) "An Optimistic Fair Exchange E-commerce Protocol with Automated Dispute Resolution", Conference on Electronic Commerce and Web Technologies EC-WEB '00, LNCS, Springer-Verlag, Berlin, Germany, Vol. 1875, pp84-93.
- Rivest, R., Shamir, A., Adleman, L. (1978) "A Method for Obtaining Digital Signatures and Public-key Cryptosystems", Communications of the ACM, ACM Press, Vol. 21, No. 2, pp120-126.
- Schneier, B., Riordan, J. (1998) "A Certified E-mail Protocol", Annual Computer Security Applications Conference, ACM Press, pp347-352.
- Shi, Q., Zhang, N., Merabti, M. (2003) "Signature-based Approach to Fair Document Exchange", IEE Proceedings - Communications, Vol. 150, No. 1, pp21 -27.
- Zhang, N., Shi, Q. (1996) "Achieving Non-Repudiation of Receipt", The Computer Journal, Vol. 39, No. 10, pp844-853.
- Zhang, N., Shi, Q. (2003) "An Efficient Protocol for Anonymous and Fair Document Exchange", Computer Networks Journal, Vol. 41, pp19-28.
- Zhou, J., Gollmann, D. (1996) "A Fair Non-Repudiation Protocol", IEEE Symposium on Security and Privacy, pp55-61.

Zhou, J., Gollmann, D. (1996) "Observations on Non-Repudiation", Advances in Cryptology - ASIACRYPT '96, LNCS, Springer, Kyongju, Korea, Vol. 1163, pp133-144.

Zhou, J., Gollmann, D. (1997) "An Efficient Non-Repudiation Protocol", Computer Security Foundations Workshop, IEEE Comput. Soc. Press, Los Alamitos, CA, USA, pp126-132.