CELICA: A Multi-Agent Communication System for Electronic Commerce

CRISTEA BOBOILA, NICOLAE CONSTANTINESCU
Department of Computer Science
University of Craiova
Bd. A.I. Cuza, no 13, Craiova 200585
ROMANIA
{boboila, nikyc}@central.ucv.ro

Abstract: - This paper presents CELICA, a multi-agent system for efficient communication in e-commerce environments. The system is based on a hierarchical approach for exchanging information among agents. Our approach uses logarithmic search techniques, thus providing improved behavior in comparison with traditional linear search methods. We present system enhancements with security features to achieve authentication and privacy of business entities, and confidentiality of exchanged information via message encryption.

Key-Words: - Electronic commerce, Multi-agent systems, Security, Public-key authentication, Symmetric key encryption.

1 Introduction
E-commerce activities are highly widespread in today’s World Wide Web, using collaborative multi-agent systems motored by automated services to facilitate access to transactional resources.

Two important features of e-commerce systems are efficiency (i.e. fast access to resources) and security. To address these aspects, we propose a multi-agent system with a hierarchical communication structure which ensures logarithmic access time to resources, and a security framework to meet privacy and security concerns.

The hierarchical communication structure is the base concept for CELICA (Communication system for E-commerce with hierarchical Levels of Interactions among Cooperative Agents), a multi-agent system that uses high-level programming technologies, object-oriented concepts and a secure authentication procedure to facilitate information exchange between agents.

Research in this area has been particularly directed towards agent negotiations [17][18], where possible scenarios are explored during the negotiation process until a mutual agreement is found. The negotiation is basically a constraint satisfaction problem, because the agents are limited by the offers advertised by the others. These offers are sometimes imprecise or incomplete, but agents still need to arrive to a consensus.

Our work also contributes to the area of secure communication frameworks for multi-agent systems. Although so far, security requirements have been mainly neglected [1], we consider that in-network communication presents a high security risk. Especially for e-business applications, this risk should not be overlooked. Security implications for multi-agent systems [7] have been studied both from a theoretical point of view, by designing security specifications and requirements [8], and from a more practical approach, by using certificates to mediate trust relations between agents [4].

The rest of the paper is structured as follows: Section 2 presents two hierarchical communications models for fast access to resources in logarithmic time. Section 3 describes a security strategy that, in addition to authentication, offers privacy of exchanged data using encryption and message signing. Section 4 concludes the paper.

2 Hierarchical Communication Models
We present CELICA’s hierarchical models of resource discovery, where the topology of the parties involved in the electronic commerce scenario is associated with a binary tree structure [2]. Each branch represents a communication link and each node in the tree represents an entity that may possess and/or request resources during the business activities. A binary tree structure benefits from a logarithmic search time, which is a significant optimization for communication.

Two logical models are deployed: the Single Perspective Hierarchy (SPH) and the Multi-Perspective Hierarchy (MPH). In SPH, the configuration is established at the beginning, and all nodes have the same, unitary view of the network...
structure. In MPH, each node is the root of a virtual priority-based tree, and it has its own view of the network. A significant improvement is obtained for MPH with the introduction of checkpoint nodes, whose role is to reduce the time during which a message exists in the network, thus implicitly leading to a decrease in traffic overhead.

2.1 Single Perspective Hierarchical Model
In the Single Perspective Hierarchical Model (SPH), each party has the same view of the communication structure in the business group. The hierarchy is therefore absolute. Any node in the binary tree network structure - a leaf, the root or any intermediary node - may request an item, and any of them can provide that item.

There are two distinct parts of this approach. The first consists of a bottom-up stage, where at each step, a node forwards the message it receives from one child to its parent (if it is not the root) and to the other child. We call this bottom-up because the request travels up to the parent, towards the root. After the message arrives at the root of the hierarchy, we have a top-down stage. The message is now always forwarded to the two children, traveling from the root to the leaves and covering the second branch of the tree relative to the root.

A substantial reduction of the overhead induced by messages sent in the network may be achieved during the bottom-up part of the algorithm. At this stage, if a node can provide the requested item, it will stop forwarding the message to the parent. Thus, a large part of the tree that was accessible through that parent does not have to be covered. For example, if the responder node is the root, as much as half of the nodes in the binary tree will not be visited, otherwise even more than half of them will not be accessed, depending on the position of the responder.

On the other hand, during the top-down stage of the algorithm, only a small part of the tree, namely the sub-tree of the responder will not be covered. There is no control over the rest of the routes.

2.2 Multi-Perspective Hierarchical Model
In the Multi-Perspective Hierarchical Model (MPH), each agent has a personal, distinct view of the communication structure in the business group. The notion of hierarchy is therefore relative to each entity. Each agent is the root of its own hierarchy represented as a binary tree, and therefore the solely requester. Any other node may provide the item and be the responder. Each agent creates its personal hierarchical view based on priorities assigned to business agents. For a certain business entity A, an agent B may have assigned a higher priority than another entity C, because it usually provides resources that are needed by A.

As a result, the concept of levels of priorities emerges. The levels of priorities are associated with the levels in the binary tree. The importance that the root gives to a node decreases towards the leaves. Since business entities with higher priority are closer to the root, the speed of resource discovery is increased.

A large part of the traffic cost is due to the fact that the search continues on additional branches in the tree, even if the resource provider was found. In order to leverage this cost, we propose an optimization method that reduces the traffic in the sub-trees that do not contain the resource provider.

We introduce the concept of checkpoint nodes, which denote a few randomly chosen nodes from the integral set, with the role of contact points. Checkpoint nodes can be reached from any node in the network through a direct link. Once the provider has been found, it contacts the checkpoint nodes thus stopping the forwarding of the message to the nodes below the checkpoints.

3 Secure Communication with Authentication and Privacy
We explore a communication protocol that enforces security with mutual authentication, and encryption of the messages sent over the network [16]. A signed hash of the encrypted message is sent over the communication link. This strategy ensures that the potential buyer is authorized to purchase that type of resource, and also that the potential seller is authorized to sell it.

The secure communication protocol is illustrated in Figure 1 and has the following general outline. First the server authenticates to the client. Next, the client engages in a mutual key exchange protocol with the server. At the end of this protocol, both the client and the server have agreed on a common shared secret key and on the type of hashing, signing and encrypting algorithms used. With this approach, the messages between the client and the server will be encrypted, and the hash of the encrypted messages will be signed.

For security reasons, each time an agent leaves the business group he loses the shared encryption key and next time he reenters the business group he will have to renegotiate a new session key with the server.
Before sending any messages, the client and the server need to authenticate themselves. We use a Public Key Infrastructure (PKI) with X.509 certificates for authentication.

### 3.1 Key Exchange Protocol
The Internet Key Exchange (IKE) has the goal of establishing a common secure communication language for the sender and the receiver (including sharing a common secret key). We implemented the IKE protocol to handle public key exchange. Each step defined in the IKE RFC [5] was followed in the implementation.

The protocol has two phases: Main Mode and Quick Mode. There are in total 9 messages exchanged. We use Diffie-Hellmann and RSA encryption/decryption to generate the key material [5]. The Main Mode phase consists in negotiation of the hashing, authentication, symmetric encryption and shared key size, while the Quick Mode phase implements the Diffie-Hellmann key exchange algorithm.

We observe that Quick Mode with Perfect Forward Secrecy (implemented with two Diffie-Hellmann operations) is slower than Base Quick Mode (implemented with one Diffie-Hellmann operation). The slow down comes from doing a second exponentiation. With Perfect Forward Secrecy, the key material used to derive one key will not be re-used in the future to derive other keys. In this way, if one key is compromised, no additional information can be inferred to compromise other keys too. For the purpose of our system, we consider that the time cost outweighs the benefit of having Perfect Forward Secrecy. Based on these considerations, we use Base Quick Mode by default.

Since exponentiation requires big numbers, we have implemented a specific “big number” structure and the specific functions to work with this structure. In our representation, a big number is a vector of unsigned long values. The first element of the vector contains the number of elements in the vector. The operations on these long numbers are implemented using simple operations like shifting, OR, AND.

After the IKE protocol between the server and the client has been performed, a common session key is obtained both by the server and by the client. Next, the messages are AES-encrypted using the key computed with IKE, and the hash is signed.

### 3.2 Encryption and Hash Signing
We use the Advanced Encryption Standard (AES) [3] and SHA-512 [9] (which belongs to the SHA-2 family of cryptographic hash functions) for encryption and hashing of messages sent between the server and the client.

We implemented a prime number generator function which generates random prime numbers needed in the RSA, in the Diffie-Hellmann key exchange protocol, and for signing hashes of messages. First, we generate a random odd number which has the most significant bit 1. After selecting this number we apply several tests to verify if this number is prime or not. The implementation of the algorithm uses the structure and the functions designed for big numbers.

As mentioned earlier, we use RSA and Diffie-Hellmann encryption/decryption for generating the keys. For generating the RSA keys we apply to follow the steps:

1. We use two generated prime numbers $p$ and $q$.
2. We compute $p \times q = n$.
3. We compute \( \phi = (p - 1)(q - 1) \) and we choose a random number \( e < n \) and \( (e, \phi) = 1 \).  
4. We choose a number \( d \) such that \( d \cdot e \equiv 1 \mod \phi \).  
5. \((e, n)\) is the public key and \((d, n)\) is the private key.

After the two keys are computed, a message \( m \) can be encrypted as \( c = m^e \mod n \). To decrypt \( c \) we will use the private key \( m = c^d \mod n \). The RSA algorithm described above is considered secure only if:

1. \( p \) and \( q \) do not have close values.  
2. \( n = pq \), where \( q < p < 2q \) and \( d < 1/3n^{1/4} \) then \( d \) can be computed in a polynomial time.  
3. the private key \( d \) is kept secret.  
4. we do not use the same private key to encrypt more than one message.

We have taken in consideration all these conditions for our prime number generator function. \( p \) and \( q \) must have at least 1024 bits for obtaining an \( n \) which cannot be factorized. Pollard [12] presented, in 1974, a method for factorizing \( n \). The method can be applied only if \( p - 1 \) and \( q - 1 \) have only small factors. A similar method was presented by Williams [13], in 1982. Williams’ method shows how to factorize \( n \) if \( p + 1 \) and \( q + 1 \) have only small factors. Therefore, the security of the RSA system is based on choosing \( p \) and \( q \); because of that the generator function is very important. This system is implemented in the Main Mode phase along with negotiation of the hashing, authentication, symmetric encryption and shared key size.

The Diffie-Hellmann key exchange protocol between two persons, who want to communicate, has the following steps:

1. Two prime numbers \( p \) and \( q \) are used, where \( 1 < q < p - 1 \).  
2. The first person chooses a secret number \( x \), where \( 1 < x < p - 1 \), and the second person chooses \( y \) where \( 1 < y < p - 1 \). These numbers do not have any common divisor with \( p - 1 \).  
3. The first person computes \( q^x \mod p \) and sends the result to the other person.  
4. The second person computes \( q^y \mod p \) and sends the result to the other one.  
5. Each one of them multiplies the received values with the chosen secret number. They will obtain the same number \( K \).  
6. \( K \) will be the public key, and the secret numbers \( x \) and \( y \) will be the private key for each communicating party, respectively.

This protocol is implemented in the Quick Mode phase. The security of the Diffie-Hellmann protocol is also based on how the two numbers, \( x \) and \( y \), are chosen. If these numbers are not selected correctly the discrete logarithm cannot be computed. Another security problem with this protocol is the authentication. An authentication is needed because one must be sure that the received value, from step 3 or 4, is sent by the right person, and not by an intruder. To resolve this problem, a digital signature algorithm can be used. This algorithm may be the same with the one used for signing hashes of the messages, in the Main Mode phase. A digital signature must provide:

1. Authentication: the receiver must be able to demonstrate who the transmitter is.  
2. Non-repudiation: the transmitter must not be able to deny sending and signing the message.  
3. Integrity: the receiver must be able to verify whether the message has been modified or not during its sending.

A digital signature algorithm is considered to be an asymmetric algorithm. A digital signature scheme must give a time stamp so that, if a private key is discovered, the signature is still valid. Such a scheme consists of three algorithms:

1. An algorithm for the key generation.  
2. An algorithm for signing the message.  
3. An algorithm for verifying the signature.

These three algorithms are implemented for every digital signature scheme. We consider the two communicating parts to be Alice and Bob. The following steps resume how a digital signature scheme works:

1. Alice signs the initial message and then she encrypts it with her private key.
2. When Bob receives the signed message, he decrypts it with Alice’s public key obtaining the message signed by Alice.
3. Bob signs the initial message and compares it with the result obtained by decrypting the message received from Alice.
4. If these two messages are identical then the signature is valid and the message has not been modified (the property of integrity is respected).

Next, we briefly present an alternative to the cryptographic system described in this section, and used for our secure communication infrastructure. The alternative is based on elliptic curves. The most popular schemes for digital signing are DSS, ElGamal and RSA. These three schemes can be also applied with elliptic curves. If the authentication is provided with an elliptic curve scheme, the Diffie-Hellman protocol from the Quick Mode phase will be replaced with ECDH (Elliptic Curve Diffie-Hellman) protocol. Its security is provided by the impossibility to compute the ECDLP (Elliptic Curve Discrete Logarithm Problem). The RSA system from the Main Mode phase will, also, be replaced with the ECRSA (Elliptic Curve RSA) algorithm.

The encryption using elliptic curves implies representing the initial message as points on an elliptic curve. Such a method was first presented by Koblitz [14]. The encryption is done after applying the elliptic curve to the message. This step is the main difference between the classic cipher and the elliptic curve one. The security of these methods is based on the elliptic curve chosen. Not all elliptic curves are good cryptographic curves, so the parameters of the curve must be chosen very carefully for optimal security and implementation efficiency. The standard FIPS 186-3 has ten recommended finite fields. There are five prime fields $F_p$ for $p=192, 224, 256, 384, 521$, and five binary fields $\mathbb{F}_{2^{m}}$ for $2^{163}, 2^{223}, 2^{283}, 2^{409}$ and $2^{571}$. The FIPS 186-3 standard recommends one elliptic curve for each of the prime fields, and one elliptic curve and one Koblitz curve for each of the binary fields [15].

Some cryptographic systems use elliptic curve cryptography instead of the classic one because the storage and the transmission requirements are reduced. The complexity of the elliptic curve algorithms is mainly affected by the scalar multiplication, since this operation is high-time consuming. Therefore, for using elliptic curve systems an optimal algorithm is needed for computing scalar multiplication, otherwise classic cryptography is recommended. For our system requirements, classic RSA and Diffie-Hellman suffice. They provide very good security assurances without the complexity and time overhead introduced by elliptic curve algorithms.

4 Conclusions
This paper presents CELICA, a multi-agent system for fast and secure resource discovery in e-business applications. CELICA employs two optimization methods: the Single Perspective Hierarchy and the Multi-Perspective Hierarchy methods, which improve upon usual linear methods and study a logarithmic search algorithm that uses binary trees to speed up the search.

The introduction of security constraints has become a goal for many of today’s systems. We present a strategy for enforcing secure communication among the agents in a business group, using certificate exchange and authentication of both parties, and encryption with AES and SHA-2. Our implementation addresses this issue by developing a secure communication framework, where remote parties authenticate each other using X.509 public-key certificates. We also present a more advanced secure strategy, which uses large key encryption to preserve the confidentiality of messages.

References:


