Expert Systems with Applications 39 (2012) 1046–1054

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Secure communication for electronic business applications in mobile agent networks

Woei-Jiunn Tsaur

Department of Information Management, Da-Yeh University, Changhua 515, Taiwan

ARTICLE INFO

Keywords:
Mobile agent
Electronic business applications
Smart card
Proxy signature
Network security

ABSTRACT

The mobile agent plays an increasingly important role in electronic business applications, because it can provide the essential properties of personalization, automation and intelligence, etc. This paper proposes several appropriate security schemes for protecting mobile agent networks in electronic business applications. As far as mobile agent security is concerned, we develop a proxy signature scheme for protecting mobile agents against malicious agent hosts. The proposed proxy signature scheme can protect users' private keys stored in smart cards, and provide the fairness of contracts signed by agents. In addition, we also design a proxy authenticated encryption scheme so that the signature of the contracts will satisfy users' constraints, and the non-repudiation of servers can be achieved. On the other hand, as far as agent host security is concerned, we apply the idea of proxy signature to construct an authentication scheme for protecting agent hosts. This scheme is to achieve the requirements of authentication and authorization. Furthermore, we also implement the proposed security schemes to achieve security requirements of confidentiality, integrity, authenticity, and non-repudiation for protecting Linux-based mobile agents and hosts in an electronic auction application. Hence, we affirm that the proposed security schemes are suitable for practical electronic business applications in mobile-agent-based network environments.

1. Introduction

In recent years, there are many business applications based on mobile agent on a variety of networks (Benouhiba & Nigro, 2006; Kang, Lee, & Choi, 2008; Kim, Kwon, & Kwak, 2010; Park, Kang, & Kim, 2006; Wu et al., 2010; Yun, Lee, Yu, & Choi, 2009). The agents of the business applications usually provide personalization, automation and intelligence, etc. However, it also results in many security threats such as stealing data from hosts by agents and tampering constraints of agents by hosts. For instance, when a mobile agent carrying a user's private key roams among servers on the Internet, the agent may find a bid satisfies the user's constraints, and then sign the bid (Chess et al., 1995; White, 1994). However, users will not wish to equip agents with their private signature keys when the agents may execute on untrusted agent hosts (Maes, Guttman, & Moukas, 1999; Sander & Tschudin, 1998; Takeda, Iino, & Nishida, 1995). On the other hand, a problem specific to mobile agents is the protection of the agent platforms running the agents. A hostile agent can destroy the hard drive, steal data, or do all sorts of undesirable operations to agent platforms. In this paper we will develop efficient security schemes based on cryptographic solutions (Mambo, Usuda, & Okamoto, 1996; Sander & Tschudin, 1998) for prevention of both agents and hosts tampering.

This paper develops a proxy signature scheme and a proxy authenticated encryption scheme for protecting mobile agents against malicious agent hosts using the proposed ECC-based self-certified public key cryptosystem. The proposed proxy signature scheme can protect users' private keys stored in smart cards, and provide the fairness of contracts signed by agents. The proposed cryptosystem is constructed using the ECC, and it also integrates the identity-based public key cryptosystem with the self-certified public key cryptosystem (Girault, 1992; Petersen & Horster, 1997; Saeednia, 1997, 2003) to provide higher security strength. Furthermore, based on the proposed cryptosystem, we employ the proposed proxy signature scheme to further design a proxy authenticated encryption scheme so that the signature of the contracts will satisfy users' constraints, and the non-repudiation of servers can be achieved. In summary, these proposed schemes are able to accomplish the security requirements of confidentiality, integrity, authenticity, and non-repudiation for protecting mobile agents in electronic business applications. On the other hand, this paper also presents an authentication scheme for protecting mobile agent hosts against unauthorized mobile agents. In such a scheme, a mobile agent can register once to the system authority for several services in the mobile-agent-based networks. Finally, we implement the proposed security schemes for protecting Linux-based mobile agent networks in an electronic auction application.

E-mail address: wjtsaur@yahoo.com.tw

0957-4174/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2011.07.105

The rest of this paper is organized as follows. In Section 2, we briefly describe the elliptic curve cryptosystems. Section 3 first develops an efficient public key cryptosystems, and then several security schemes constructed using it are designed for protecting mobile-agent-based electronic business applications. In Section 4, security analyses about attacks on the proposed schemes consolidate the feasibility of the schemes. Performance evaluation of the proposed schemes, which is measured by the required computational effort and communicational cost, is given in Section 5. In Section 6, we present the implementation of the proposed schemes on an electronic auction application. Finally, some concluding remarks are presented in Section 7.

2. Elliptic curve cryptosystems (ECC)

Assume that P is a point with order n on an elliptic curve E: $y^2 = x^3 + ax + b \pmod p$, where $4a^3 + 27b^2 \neq 0 \pmod p$, and Q is some other point on the same curve. Let $P = (x_P, y_P)$, $Q = (x_Q, y_Q)$, and $P + Q = R = (x_R, y_R)$. If $x_P \neq x_Q$, set $\lambda = \frac{y_Q - y_P}{x_Q - x_P}$; if $x_P = x_Q$, set $\lambda = \frac{3x_P^2 + a}{2y_P}$. Then the point $R = (x_R, y_R)$ can be defined by using the following formulae:

$$x_R = \lambda^2 - x_P - x_Q$$
$$y_R = (x_P - x_R)\lambda - y_P$$

In ECC, the elliptic curve discrete logarithm problem (ECDLP) is to determine an integer x $(0 \leqslant x \leqslant n-1)$ such that $Q = x \cdot P$ if such an x exists. As long as n and p are large enough, it is computationally intractable to find x with knowing E, Q, and P. Koblitz (1987) and Miller (1986) implemented this characteristic to elliptic curve cryptosystems. We need 1024-bit keys when using modular exponentiation schemes, like RSA or ElGamal cryptosystems, but we can get the same security level only using 160 bits in ECC.

In addition, RSA needs to generate $N_i = p_i \cdot q_i$ for each user, respectively; however, ECC generates only fixed public information stored at SA, and can afterwards uses it repeatedly. Therefore, the storage cost required by ECC is less than that required by RSA.

3. Security schemes for protecting mobile agent networks

In this section, we first develop an efficient public key cryptosystems, and then several security schemes constructed using it are designed for protecting mobile-agent-based electronic business applications.

3.1. Initialization

The entities in the system are a system authority (SA), users (U_i), hosts (H_i), and mobile agents (MA) generated by specific users. We assume that SA is responsible for key generation and user registration. We then define the notations used in the proposed schemes as follows:

- p: a field size, where p is typically either an odd prime or a power of 2 in general applications, and its length is about 160 bits.
- An elliptic curve E over F_p:
  E: $y^2 = x^3 + ax + b$, where the two field elements $a, b \in F_p$ and $4a^3 + 27b^2 \neq 0 \pmod p$, and all the points (x, y), $x \in F_p$, $y \in F_p$, on E form the set of $E(F_p)$ containing a point O called the point at infinity.
- B: a base point of order n over $E(F_p)$, where n is a large prime (160 bits) and the number of F_p-rational points on E, denoted by $\#E(F_p)$, is divisible by n.
- s_{SA}: SA's private key, where $s_{SA} \in [2, n-2]$.
- P_{SA}: SA's public key, where $P_{SA} = s_{SA} \cdot B$ ("$\cdot$" means the multiplication of a number and an elliptic curve point.).
- $h(\,)$: a one-way hash function that accepts a variable length input and produces a fixed length output value j, where $j \in [2, n-2]$ and its length is 160 bits. The one-way hash function $h(\,)$ should satisfy the properties (Harn, 1994) that given $h(x)$, it is computationally infeasible to find $x' \neq x$ such that $h(x') = h(x)$, meanwhile, $h(x') \neq h(x)$ if and only if $x' \neq x$.
- $X(P)$: output the x-coordinate of point P.

After that, SA publishes E, B, p, n, P_{SA} and h, while keeping s_{SA} secret.

3.2. The proposed public key cryptosystems

The operations of the proposed public key cryptosystems are divided into two phases: the system setup phase and the key generation phase.

3.2.1. The system setup phase

SA creates a system public key and some public parameters in this phase, and then SA releases these parameters. SA randomly chooses a numbers$_{SA}$ and keeps it secret. Then SA computes the system public key

$$P_{SA} = s_{SA} \cdot B$$

3.2.2. The key generation phase

User U_i and host H_i perform the following steps to register to SA, and obtain the corresponding public key, respectively. They also compute their private keys in this phase.

Step 1. U_i and H_i execute the following tasks, respectively:
(1-1) Select an identity information, denoted by I_i.
(1-2) Randomly choose an integer $x_i \in [2, n-2]$ as the master key.
(1-3) Compute $Z_i = h(x_i \| I_i) \cdot B$
(1-4) Submit $\{I_i, Z_i\}$ to SA.

Step 2. SA executes the following tasks for U_i and H_i, respectively:
(2-1) Randomly choose a time-variant integer $k_i \in [2, n-2]$.
(2-2) Compute a public key P_i and its witness w_i, where

$$P_i = Z_i + (k_i - h(I_i)) \cdot B = (P_{ix}, P_{iy})$$
$$w_i = k_i + s_{SA} \cdot (P_{ix} + h(I_i)) \pmod n$$

(2-3) Return $\{P_i, w_i\}$ to U_i and H_i, respectively.

Step 3. U_i and H_i then execute the following tasks, respectively:

(3-1) Calculate their own private keys as

$$s_i = w_i + h(x_i \| I_i) \pmod n$$

(3-2) Verify the authenticity of P_i by testing if

$$s_i \cdot B = P_i + h(I_i) \cdot B + [(P_{ix} + h(I_i)) \bmod n] \cdot P_{SA} \quad (1)$$

If the verification result of Eq. (1) is correct, then the participant's public key is P_i and the corresponding private key is s_i; otherwise, it means that the public key P_i is altered in the transmission. For the consideration of security, the private key s_i is stored in a smart card for subsequent electronic business applications.

In the following, we show that the private key s_i and the corresponding public key P_i satisfy Eq. (1).

Theorem 1. *User U_i and host H_i can utilize Eq. (1) to verify his/her public key P_i by himself/herself.*