LETTER

An Elliptic Curve-Based Trust Management Protocol in Peer-to-Peer Networks

Aiguo CHEN†[a], Nonmember, Guangchun LUO†, Member, and Jinsheng REN†, Nonmember

SUMMARY Establishing trust measurements among peer-to-peer (P2P) networks is fast becoming a de-facto standard, and a fair amount of work has been done in the area of trust aggregation and calculation algorithms. However, the area of developing secure underlying protocols to distribute and access the trust ratings in the overlay network has been relatively unexplored. We propose an elliptic curve-based trust management protocol for P2P systems, which is designed to provide authentication and signature functions to protect the processes of trust value query and rating report. Additionally, instead of using single identities, the protocol generates two verifiable pseudonyms, one is used for transaction, the other is applied when the peer acts as a trust holding peer. A security analysis shows that the proposed protocol is extremely secure in the face of a variety of possible attacks.

key words: peer-to-peer, authentication, trust, trust management

1. Introduction

In recent years, peer-to-peer (P2P) networks have gained significant acceptance [1]–[3]. Despite the demand of robustness and scalability of P2P systems, the anonymous and dynamic nature of peer activities make the systems vulnerable to abuses by selfish and malicious peers [4], [5]. To encourage resource sharing among peers and combat malicious peer behaviors, trust management is essential for peers to assess the trustworthiness of others and to selectively interact with the more reputable ones. There are two important issues in trust management: (1) what trust metrics and aggregation algorithms are effective for computing the reputation-based trust. And (2) how to distribute, store, and access the trust value of peers securely [6]–[8]. A fair amount of work has been done related to the first issue [9]–[11]. However, The research in developing secure mechanisms of distribution and access trust value related to the second issue is scanty. To the best of our knowledge, Cornelli et al. has attempted to present a secure protocol P2Prep [12] at message level in addition to a trust aggregation mechanism. ReP [8] focuses particularly on fairness, livelihood and privacy while rating, but it uses centralized trust management mechanism.

In this paper, we present an elliptic curve-based trust management protocol which focuses on the security of trust value distribution and access. A light weight authentication function is provided for trust querying, and an interaction prove function is employed for trust reporting. Our protocol satisfies the following features:

1. The trust value of each peer is stored at other peers in the network. This decentralized method enhances security and reliability of the trust management system.
2. Two kinds of pseudo identities are used to protect these trust holding peers from targeted attacks. And any malicious peer trying to manipulate trust ratings should be identifiable.
3. In query phase, the trust holding peer’s identity is authenticated by the trust querying peer to ensure that anybody querying for a trust value gets the true trust value. The THA peer’s authentication approach is a three-step and elliptic curve-based.
4. An elliptic curve-based signature is used to generate the interaction proof to confirm that transaction has actually taken place between two peers while rating report phase.

We discuss the implementation choices that were made for security and efficiency reasons and use attack analysis and contrastive analysis to evaluate our protocol.

2. Elliptic Curve-Based Management Protocol

2.1 Design Overview

Considerate that the current trust value of a peer must not be computed by and reside at the peer itself, our protocol uses a random assignment of Trust-Holding Agent peers (henceforth called THA peers) as TrustMe [6] does. There is more
Table 1: Notation of variable.

| Notation | Specification |
|----------|--------------|
| $ID_i$  | Transaction identity of peer $i$ |
| $ID'_i$ | THA identity of peer $i$ |
| $<s_i, P_i>$ | Private and public keys pair of peer $i$ |
| $<s'_i, P'_i>$ | Private and public keys pair of peer $i$ |
| $h_i(·)$ | Secure hash function |
| $R_i$  | Trust value of peer $i$ |
| $T_{d_{ij}}$ | Trust rating of peer $i$ from peer $j$ |
| $T_i$  | The timestamp when peer $i$ send a message |
| $P'_i(·)$ | Encrypt with public key $P'_i$ |
| $E_i(·)$ | Encrypt with symmetric key $k$ |
| $∥$   | Concatenation |

Given an elliptic curve $E$ over a field $F_n$ (use $E/F_n$ to represent), $G \in E/F_n$ (where $G$ is a generator, and its order contains a large prime). When a new peer $j$ contacts a bootstrap server (BS) for joining the network, it generates two public key pairs and the corresponding identities, $\{ID_i, < s_i, P_i >\}$ and $\{ID'_i, < s'_i, P'_i >\}$, the former are used in transaction (called transaction identity), and the latter are used when the peer acts as a THA peer (called THA identity). Besides, $ID_i, ID'_i \in \{0, 1\}^m$ denote pseudo names of peer $i$ ($\{0, 1\}^m$ denotes a set of binary strings with $m$ bits length). Where, $s_i, s'_i \in F_n$ and $P_i = s_iG \in E/F_n, P'_i = s'_iG \in E/F_n$. Then the BS publishes the information of peer $i$ signed by its public key on public sites. We adopt ElGamal for Elliptic Curves [17] to encrypt and decrypt data in our protocol. Note that the BS is a form of an identity authority. Since most trust models are identity-based [13], either real IDs or pseudo IDs. But the BS does not signify, in any manner, a peer being trusted or not.

2.2.1 Trust Value Query

**Step Q1** When a peer, say peer $j$ intending to query about the trust value of peer $i$, could issue a lookup request with keyword $ID_i$.

$$Q(j, i) = \{ID_i, U, T_j\}$$  \hspace{1cm} (1)

Peer $j$ generates a large random integer $a \in F_n$, and calculates $U = aG \in E/F_n$. Where $T_j$ is the time stamp for the query message. Note that the problem of how the lookup request arrives the THA peers anonymously and secretly, has been researched in many studies, such as APFS [14] and Salsa [15]. The existing approaches can also be adopted in our protocol.

**Step Q2** Once received the trust query message from peer $j$ for peer $i$, its THA peer, say peer $h$ generates a replay message and replays for peer $j$.

$$h \rightarrow j: \{E_k(R_i || b), r, T_k\}$$  \hspace{1cm} (2)

Peer $h$ selects a large random integer $r \in F_n$, uses $h_1 : E/F_n \times F_n \rightarrow \{0, 1\}^m$ to generate a session key $k = h_1(s_jU, r)$, and then computes $b = h_2(k, r, ID'_h)$, where $h_2 : \{0, 1\}^m \times F_n \times \{0, 1\}^m \rightarrow \{0, 1\}^m$. Finally, encrypts peer $i$’s trust value $R_i$ and $b$ using the key $k$, and sends $\{E_k(R_i || b), r, T_k\}$to peer $j$.

**Step Q3** Peer $j$ checks whether timestamp $T_j$ is invalid. If it is invalid, peer $j$ calculates $k' = h_1(aP'_h, r)$ and $b' = h_2(k', r, ID'_h)$, then checks whether $b \equiv b'$. If the verification holds, peer $h$ is authenticated. Besides the session key $k' = k$ is confirmed at the same time. Peer $j$ obtains $R_i$ by decrypting the message with $key'k$.

2.2.2 Trust Rating Report

**Step R1** If peer $j$ selected a service provider peer $i$ and has interacted with peer $i$ indeed, it should get an interaction proof from peer $i$. The proof is used to prove of an interaction between them while feedback reporting. The proof message as follows:

$$i \rightarrow j: \{\delta_i, x, T_i\}$$  \hspace{1cm} (3)

Peer $i$ selects a random $l, l \in [1, n - 1]$, and calculates $lG = (x_1, y_1) \in E/F_n$ and $x = x_1 \bmod n \in F_n$. The result should be Satisfied $x \neq 0$, otherwise peer $i$ will choose a new $l$ and computer again. Then, peer $i$ calculates $e = h_3(T_i || ID_j)$ and $\delta_i = \Gamma^{-1}(e + s_ix)$ mod $n$, where $h_3 : \{0, 1\}^m \rightarrow \{0, 1\}^m$. The three-triple $(\delta_i, x, T_i)$ is the signature on interaction timestamp $T_i$ and peer $j$’s identity $ID_j$. Peer $i$ sends $(\delta_i, x, T_i)$ to peer $j$ as the interaction proof.

Note that no peer can generate such a value in a fake manner, since it does not know peer $i$’s private key $s_i$. The timestamp $T_i$ is used to prevent rephrasing of such message, the use of $ID_j$ is for added protection against somebody using a message from peer $j$’s interaction with some other peer. Another important use of the interaction message is that if a group of collusive peers are attempting to boost each other’s rating, they will need to exchange such messages every time (unless they compromise on each others’ private keys as well). Thus making they pay for every malicious attempt.

**Step R2** After interacting with peer $i$, peer $j$ delivers the
interaction proof \((\delta_i, x, T_i)\) and the current trust rating \(D_{ij}\) to peer \(i\)’s THA peers.

\[ j \rightarrow h : (ID_i, P_h(D_{ij}), (\delta_i, x, T_i)) \] (4)

For this step, it needs to ensure that only THA peer can read the message and that only a peer which has actually interacted with peer \(i\) can send the report, so we use the THA peer’s public key to encrypt the trust rating \(D_{ij}\) and current timestamp \(T_j\), and send the proof together. And using public key encrypting could not disclose the privacy of its secret key.

**Step R3** While the THA peer receiving the report message, it checks whether timestamp \(T_i\) is valid and whether \(\delta_i, x \in [1, n - 1]\). If both of them are valid, the THA peer calculates

\[ e = h_2(T_i||ID_j), w = \delta_i^{-1} \mod n, u_1 = ew \mod n, u_2 = xw \mod n, X = u_1G + u_2P_j = (x_1, y_1), \text{ and } v = x_1 \mod n, \]

then verifies whether \(v = x\). If it holds, the proof and the report are accepted valid. Then THA peer decrypts \(P_h(D_{ij})\) and updates the trust value of peer \(i\).

Since if \((\delta_i, x, T_i)\) is the genuine proof message signed by peer \(i\), then:

\[ \delta_i = \Gamma^{-1}(e + s_i x) \mod n. \]

\[ k \equiv \delta_i^{-1}(e + s_i) \equiv w(e + s_i x) \equiv (u_1 + u_2 s_i) \mod n. \]

\[ X = u_1G + u_2P_j = (u_1 + u_2 s_i)G = kG. \]

Thus \(v = x\).

3. Security and Performance Analysis

Now, let us look at how our protocol prevents various possible attack scenarios for trust management, and the privacy and security properties it has.

3.1 Security Analysis

3.1.1 Manipulating Reply Message

This can be attempted by either a malicious THA peer or a non-THA peer. A THA peer can send a wrong trust value of a peer. To prevent this problem, we assign more than one THA peer for each peer as much of the other researches do [6, 9]. The requester could accept the majority vote on the trust value. In the query phase, the THA peer has been authenticated before accepting the trust value. Moreover, the identity information is contained in reply message, e.g., \(k = h_1(s_i^r U, r)\). If the cheat happens, the THA peer will be identified and be exposed by BS. A non-THA peer may impersonate a THA peer by intercepting the reply message and modifying the value. As described in step Q2, the malicious should break the session key \(k\) at first to carry out this attack. But the session key \(k\) is generated by a random from initiator (peer \(j\)) and the private key of responder (peer \(h\)). And the authentication and session key negotiation are on discrete logarithm over elliptic curve (DLEC) problem which is infeasible on computation [16].

3.1.2 Manipulating Report Message

The report message may be manipulated either by replaying or forging. In our protocol, each report message contain an interaction proof, The proof is generated by the elliptic curve digital signature algorithm (ECDSA), the security has been proved in Ref. [16]. The improvement of our protocol is use the timestamp \(T_i\) and initiator’s identity \(ID_j\) as the signature message, namely \(e = h_2(T_i||ID_j)\). Any report message, containing the interaction proof message outside a reasonable time frame, is discarded. Moreover, there is a good effect of using interaction proof, if the responder provides a malicious file or other fake service, the proof ensures accountability. In addition, the report message is encrypt by the public key of the THA peer, so the only the THA peer can read the message.

3.1.3 Attempting to Identify THA Peers

Trust management not only should provide mutual anonymity but also need to ensure that the THA identity should not be linked with the transaction identity. This requirement contain two aspects, one is the real identity information should not be disclosed in interaction, the other one is the THA identity should be different from and no correlation with transaction identity. In our scheme, we use pseudo names identify peers and two groups information \(\{ID_i, (s_i, P_i)\}\) and \(\{ID_j, (s_j', P_j')\}\) to denote two kind of identity. The \(\{ID_i, (s_i', P_i')\}\) pair is only used while acting as a THA peer.

3.1.4 Secret Key Privacy

In the trust query and reply phases, the session key \(k = h_1(s_i^r U, r)\) is generated from the THA peer’s secret key \(s_i^r\). Unlike P2PRep and TrustMe, we use a random integer \(r\) and a secure hash function \(h_1(\cdot)\) to protect its confidentiality. When the resource provider peer \(i\) produce an interaction proof, we also use a random \(x\) to enhance the security of using secret key, namely \(\delta_i = k^{-1}(e + s_i x) \mod n\). In Table 2, we made a comparison of protocol security properties between our protocol and others (e.g., P2PRep, and TrustMe). From Table 2, it is very clear that our protocol proposed in Sect. 2 satisfies the greatest number of the privacy and security properties. Till now we have demonstrated how the protocol achieves security, reliability and accountability. In

| Protocol                          | P2PRep | TrustMe | Our Protocol |
|----------------------------------|--------|---------|--------------|
| Manipulating Reply               | O      | \(\Delta\) | O            |
| Manipulating Report Message      | O      | O       | O            |
| Attempting to Identify THA Peers | \(\Delta\) | O       | O            |
| Secret Key Privacy               | X      | X       | O            |

Notation: O: Satisfied \(\Delta\): Partially Satisfied X: No Satisfied
addition to the security features, our protocol also decreases communication load and computational time. As described in Sect. 2.2.1, the traffic overhead of three-stepped authentication while trust dispatching is lower than Pseudo Trust using six steps. Moreover, there is not any exponential calculation in both trust query and report phases by using elliptic curve-based protocol.

3.2 Performance Analysis

In this section, we analyze our protocol in terms of computation and communication costs by comparing it with TrustMe [6]. We calculate the computation and communication costs within one complete transaction, which includes both trust query and trust rating report phases.

The aforementioned P2PRep [12] is based on a polling mechanism, which also attempts to present a secure trust access protocol. In [6], the authors have compared the performance of TrustMe and P2PRep, and get the following conclusions:

1. Their cost of using cryptography primitives are of the same order, since both of them are based on public key cryptography;
2. The THA based TrustMe have the least cumulative response time. And the polling based mechanism P2PRep is more expensive, since it has to collect the replies from all peers and then combine them.

Our protocol is also based on THA peers. It has the same cumulative response time as TrustMe. So we just compare our protocol with the TrustMe here.

Let’s consider the notations used as follows:

1. \( C_s \), the cost of performing a symmetric operation (encryption or decryption with a symmetric key);
2. \( C_{\text{sig}} \), the cost for performing an asymmetric private operation (plaintext decryption or signature using a private key);
3. \( C_{\text{ver}} \), the cost for performing an asymmetric public operation (plaintext encryption or signature verification using a public key);
4. \( C_h \), the cost of hash function computation;
5. \( C_{\text{ECDH}} \), the cost of executing an elliptic curve Diffie-Hellman (ECDH) operation;
6. \( C_{\text{bro}} \), the cost of broadcasting one message;
7. \( C_{ij} \), for peer \( i \), this is the cost of a transmitted message from the peer \( i \) to the peer \( j \); But for peer \( j \), this means the cost of a received message from the peer \( i \) to the peer \( j \);
8. \( N_h \), the number of THA peers. Usually, we can use 3 THA peers in our protocol.

First, we look at the computation cost. Table 3 presents the cryptographic operation costs of TrustMe and our protocol. Our protocol uses the session key generation protocol which involves ECDH and Hash functions. But TrustMe all uses asymmetric key pairs to encrypt and decrypt messages. Both of them are based on public key cryptography and with the same order operation. \( C_s \) and \( C_h \) is far lower than \( C_{\text{sig}} \) and \( C_{\text{ver}} \). In ECC, \( C_{\text{ECDH}} \) is lower than the computation cost of ECDSA [18]. So our protocol is more efficient.

Next we look at each peer’s message cost because of the increased number of messages. As Table 4 presents the message overhead of TrustMe and our protocol. The message costs of both protocol’s THA peers are same. For peer \( i \), TrustMe has one more received message than our protocol. For peer \( j \), there is one more broadcast message in TrustMe. In our protocol, point-to-point messages form peer \( j \) to each THA peer are used stead of broadcast message. This method could reduces the bandwidth resources consumption.

4. Conclusions

In the current article, an elliptic curve-based trust management protocol is presented. The protocol provides the THA peer’s identity authentication function while trust querying, and a traceable proof function while rating report phase. Two group of identities are used to mark each peer to ensure the anonymity in trust management. We have also presented a security analysis of the protocol, showing that the protocol has desirable features of anonymity, accountability and reliability in trust query and reporting processes. Our protocol is feasible for most of global reputation-best trust models, such as Trust [9], GTMS [10] and REP [11].

Our current work is directed at using the previously mentioned trust management protocol to implement a trust infrastructure. Additionally, the client environment security of each peer is still a critical issue; therefore, we are also continuing to investigate the potential benefits of import trusted computing technology into trust management system.

References

[1] W.W. Terpstra, J. Kangasharju, C. Leng, and A. P. Buchmann, “BubbleStorm: Resilient, probabilistic, and exhaustive peer-to-peer search,” ACM SIGCOMM Computer Communication Review, vol.37, no.4, pp.49–60, 2007.
[2] Y. Liu, L. Xiao, and L.M. Ni, “Building a scalable bipartite P2P overlay network,” IEEE Trans. Parallel Distrib. Syst., vol.18, no.9,
pp.1296–1306, 2007.

[3] F. Almenárez, A. Marín, D. Díaz, A. Cortés, C. Campo, and C. García-Rubio, “Trust management for multimedia P2P applications in autonomic networking,” Ad Hoc Networks, vol.9, no.4, pp.687–697, 2011.

[4] B. Yang, T. Condie, S. Kamvar, and H. Garcia-Molina, “Non-cooperation in competitive P2P networks,” Proc. 25th IEEE International Conference on Distributed Computing Systems (ICDCS 2005), pp.91–100, Columbus, OH, June 2005.

[5] D. Dumitriu, E. Knightly, A. Kuzmanovic, I. Stoica, and W. Zwaenepoel, “Denial-of-service resilience in peer-to-peer file sharing systems,” ACM SIGMETRICS Performance Evaluation Review, vol.33, no.1, pp.38–49, 2005.

[6] A. Singh, L. Liu, “TrustMe: anonymous management of trust relationships in decentralized P2P systems,” Proc. Third International Conference on Peer-to-Peer Computing (P2P’03), pp.142–149, Linköping, Sweden, Sept. 2003.

[7] T. Mahler and T. Olsen, “Reputation systems and data protection law,” eAdoption and the Knowledge Economy: Issues, Applications, Case Studies, Amsterdam, pp.180–187, 2004.

[8] S. Schiffler, S. Clau, and S. Steinbrecher, “Privacy, liveliness and fairness for reputation,” Proc. 37th International Conference on Current Trends in Theory and Practice of Computer Science, pp.506–519, Novy Smokovec, Slovakia, 2011.

[9] R. Zhou and K. Hwang, “PowerTrust: A robust and scalable reputation system for trusted peer-to-peer computing,” IEEE Trans. Parallel Distrib. Syst., vol.18, no.4, pp.460–473, 2007.

[10] R.A. Shaikh, H. Jameel, B.J. d’Auriol, H. Lee, S. Lee, and Y.-J. Song, “Group-based trust management scheme for clustered wireless sensor networks,” IEEE Trans. Parallel Distrib. Syst., vol.20, no.11, pp.1698–1712, 2009.

[11] P.B. Velloso, R.P. Lauffer, D. de O Cunha, O.C.M.B. Duarte, and G. Pujolle, “Trust management in mobile ad hoc networks using a scalable maturity-based model,” IEEE Trans. Network and Service Management, vol.7, no.3, pp.172–185, 2010.

[12] F. Cornelli, E. Damiani, S.D.C.D. Vimercati, S. Paraboschi, and P. Samarati, “Choosing reputable servants in a P2P network,” Proc. 11th World Wide Web Conference (WWW 2002), pp.376–386, Honolulu, Hawaii, May 2002.

[13] L. Lu, J. Han, Y. Liu, L. Hu, J. Hua, L.M. Ni, and J. Ma, “Pseudo trust: Zero-knowledge authentication in anonymous P2Ps,” IEEE Trans. Parallel Distrib. Syst., vol.19, no.10, pp.1325–1337 2008.

[14] V. Scarlata, B.N. Levine, and C. Shields, “Responder anonymity and anonymous peer-to-peer file sharing,” Proc. IEEE International Conference on Network Protocols (ICNP), pp.272–280, Riverside, CA, Nov. 2001.

[15] A. Nambiar and M. Wright, “Salsa: A structured approach to large scale anonymity,” Proc. 13th ACM Conference on Computer and Communications Security, pp.17–26, Alexandria, Virginia, USA, Oct.-Nov. 2006.

[16] N. Koblitz, Algebraic aspects of cryptography, Springer, Anguilla, pp.131–136, 1997.

[17] K. Rabah, “Elliptic curve elgamal encryption and signature schemes,” Information Technology J., vol.4, no.3, pp.299–306, 2005.

[18] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz, “Energy analysis of public-key cryptography for wireless sensor networks,” Third IEEE International Conference on Pervasive Computing and Communications, pp.324–328, Kauai Island, HI, March 2005.