Cloud-based Trust Establishment Protocol towards Mobile Commerce

Bailing Liu
Department of Information and Management, Central China Normal University
blliu.ccnu@gmail.com

Abstract -In an attempt to integrate automated trust negotiation (ATN) into mobile commerce environments to support an effective approach to establishing dynamic and bilateral trust between mobile customers and merchants, we propose a cloud-based trust negotiation (CBTN) protocol to efficiently and securely offload expensive trust negotiation operations to cloud servers, alleviating computational and com-munication overheads on mobile devices that have limited resources such as lim-ited computational power, battery life and bandwidth. CBTN protocol increases users' trust to offload the trust negotiation process by keeping users' control of their own credentials and providing negotiation feedbacks to enable the off-loaded trust negotiations are performed correctly.

Keywords -trust negotiation, mobile commerce, cloud computing

1. Introduction

Advanced and mature mobile and wireless communications technologies facilitate e-commerce conducted from a wired network to a wireless network. Mobile commerce (m-commerce) [3] enables customers to conduct transactions anywhere using mobile devices such as smart phones, personal digital assistants (PDAs), and laptops. M-commerce viewed as a subset of e-commerce faces the same problems troubling e-commerce, such as the issue of trust. Trust is a major obstacle in the adoption and development of m-commerce [5]. Furthermore, security and privacy significantly impact mobile consumer trust to m-commerce [10]. Gaining consumer trust in mobile commerce is a particularly daunting task because of its unique features like limitations in mobile handsets' computational power, memory and battery life [7]. Wireless networks are also limited in communication bandwidth and have a relatively high operation cost. Therefore, energy consumption will often be at a premium when developing methods for building trust in m-commerce [4].

Automated trust negotiation (ATN) [9] is an advanced approach to dynamically establishing mutual trust relationships between strangers wishing to share resources or conduct business. It enables participants to establish bilateral trust by iteratively disclosing policies and digitally-signed credentials containing negotiators' attribute information. Integrating ATN into m-commerce can support flexible, dynamic and bilateral trust establishments between mobile consumers and merchants. However, its utility comes at the cost of non-trivial computational and communication overheads. During a trust negotiation, a trust sequence should be generated via the bilateral and iterative exchange of policies and digital credentials, asymmetric cryptographic operations are required to verify credentials. The resources required to perform asymmetric key operations on mobile devices and to transmit large messages through wireless network may result in unacceptable performance result. Therefore, the computational and communication complexity of ATN cause high deployment costs and high operational overhead on mobile devices.

Cloud computing [2] is a new type of service provided through the Internet. It can be defined as the aggregation of computing as a utility and software as a service where the applications are delivered as services over the Internet and the hardware and systems software in data centers provide those services. It provides the possibility of energy savings as a service for mobile users, which is called computation offloading. Therefore, offloading trust negotiations from mobile devices to cloud computing is a solution to address the problems of integrating ATN into mobile commerce.

Offloading intensive computations and data storage using the cloud for mobile devices does pose questions of security and trust issues. For energy savings, mobile users' attribute credentials and associated policies should be sent to and stored in the cloud [6]. But users may feel uncomfortable to replicate sensitive credentials and policies to the cloud. How do we ensure the cloud cannot maliciously masquerade as the user if he transmits credentials to the cloud? How do we ensure that the off-loaded trust negotiation strictly comply with the policies? For addressing these problems, we propose a cloud-based trust negotiation (CBTN) protocol to securely and efficiently offload the expensive trust negotiation process to cloud computing, alleviating the computational and communication overheads on mobile devices. The protocol increases users' trust to offload the operation to the cloud by keeping the control of their own long-term credentials and enabling users to check whether the off-loaded trust negotiations are performed correctly.

The remainder of this paper is organized as follows. In section 2, we illustrate the proposed protocol CBTN protocol. In section 3, the protocol is analyzed from the aspects of security and efficiency. We conclude the paper in section 4.

2. Cloud-based Trust Negotiation (CBTN) Protocol

In this section, we propose the Cloud-based Trust Negotiation (CBTN) protocol offloading the expensive portions of the trust negotiation process to the cloud computing.

2.1. Proxy Certificate

Cloud servers require access to every credentials possessed by the user when offloading trust negotiations to the cloud. If the user's credentials and associated policies are simply duplicated to the cloud, any compromised or malicious cloud server could impersonate the user. This raises information security and users' trust issues. To address these problems, we introduce the concept of proxy certificates [8].

Instead of giving up the whole control to their own credentials, users could generate temporal proxy certificates bounded to their long-term credentials, and store them in the cloud rather than the user's sensitive attribute credentials themselves, avoid risking the safety of long-term secrets. Before offloading trust negotiations to cloud computing, the cloud should be delegated to perform such operations on behalf of the user using short-lived proxy certificates by following RFC3820 [8]. Furthermore, any restrictions on how the delegated cloud server uses the proxy certificates can be placed by means of policies field, which could be achieved by the X.509 extension indicating that these proxy certificates are to be used in response to users' offloaded request only.

2.2. Flow Diagram of the CBTN Protocol

Fig. 1 presents an information flow diagram of the CBTN protocol. The protocol consists four phases.

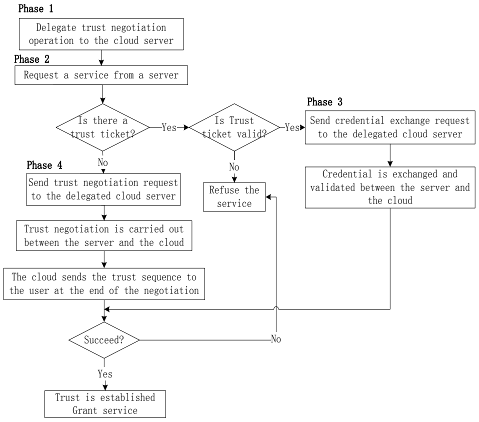
Phase 1
Delegate trust negotiation operation to the cloud server
Phase 2
Request a service from a server
Is there a trust ticket?
Yes
Is Trust ticket valid?
Yes
Phase 3
Send credential exchange request to the delegated cloud server
Credential is exchanged and validated between the server and the cloud
No
No
Refuse the service
Phase 4
Send trust negotiation request to the delegated cloud server
Trust negotiation is carried out between the server and the cloud
The cloud sends the trust sequence to the user at the end of the negotiation
Succeed?
No
Yes
Trust is established
Grant service

Fig. 1: Information flow diagram of the CBTN protocol

Phase 1: Before requesting a service from a server, a mobile consumer delegates the trust negotiation operation to cloud computing. For taking over trust negotiations on behalf of the user, the cloud needs proxy certificates for its identity and each of the user's attribute credential, and copies of the policies that it will be relied upon to enforce. For this purpose, public-private key pairs are generated by the cloud server and public keys are subsequently transmitted to the mobile user. On the arrival of the public keys, the user generates proxy certificates, each of which contains the public key, and signs the proxy certificates with the user's own private key. Finally, the user sends the proxy certificates back to the cloud server. The cloud subsequently takes over all the trust negotiation operations until the user's proxy certificate for the delegation expires.

Phase 2: We borrow the idea of trust tickets from literature [1]. When a user asks for a service, he could transmit the trust ticket along with the service request, proving that they have already had a previous successful negotiation for obtaining the service by following the trust sequence specified in the trust ticket. If the valid trust ticket exists, negotiators enter the phase 3. If the trust ticket is not valid, the negotiation is terminated and the service is refused. Otherwise, they enter the phase 4.

Phase 3: Since the trust ticket includes the successful trust sequence, the sequence generation process is omitted. Negotiators establish trust by directly exchanging and validating credentials specified in the trust ticket. Credential validation is computationally expensive, so the mobile user should offload the operation to the cloud.

Phase 4: If the mobile user does not have the trust ticket, both sides need to determine a successful trust sequence by carrying out a trust negotiation session, and validate credentials. These operations need high communication and computational cost. Therefore, the mobile user offloads the whole negotiation process to the cloud.

2.3. Description of the CBTN Protocol

The proposed CBTN protocol is illustrated in Fig. 2.

Step 1: To delegate trust negotiation processes, the mobile user sends a delegation request to the cloud (1.1), including a symmetric key shared between the user and the cloud, and signed by the user's private key, i.e. $E_{pubcl}(del\_req)$, del_req= $K_{user,\ cloud}$ || $Nonce_{user}$ || $E_{priu}(K_{user,\ cloud})$. Nonce is a random data against reply attack. In response, the cloud server generates a private-public key pair and sends the public key back to the user (1.2), i.e. $E_{K_{user,cloud}}(pub_{dele}$ || $Nonce_{user})$. The mobile user then generates a proxy certificate signed by the user's private key and transmits it to the cloud (1.3), i.e. $E_{K_{user,cloud}}(ProCer_{cloud,\ user})$. Using the same way, the user needs to generate more proxy certificates for his attribute credentials, and sends them along with the copy of associated policies to the cloud (1.3). Once the delegation is successfully completed, step 1 is skipped until a proxy certificate expires.

Step 2: If the mobile user has a trust ticket *t_ticket* for the requested service, and the merchant server has validated that it is valid. The server sends a request for exchanging credentials *exch_req* as specified in the trust ticket, possibly with his own credentials if he should disclose the credentials first (2.1). For offloading credential exchange and validation operations, the mobile user transmits a validation request *val_req* to the delegated cloud, together with the trust ticket and credentials disclosed by the server (2.1.1). In response, the cloud sends a cloud-based validation message *CBV* informing the transfer of control to the cloud (2.1.2, 2.1.3). And then the credential exchange happens between the cloud and the server (2.1.4).

Step 3: If the trust ticket does not exist, the merchant server sends a credential request *cred_req* for initiating a trust negotiation (2.2). The mobile consumer asks the cloud to take over the trust negotiation process by sending a cloud-based trust negotiation request *CBTN_req* (2.2.1). The cloud sends a cloud-based trust negotiation transfer message to notify that the cloud will carry out the trust negotiation with the server on behalf of the mobile consumer (2.2.2, 2.2.3). Then the trust negotiation begins between the server and the delegated cloud server (2.2.4). A trust negotiation sequence *CBTN_seq* generated from the trust negotiation is forwarded to the user as a feedback at the end of the negotiation (2.2.5), making sure the off-loaded trust negotiation was executed correctly.

Step 4: The original requested service is granted if the negotiation (including credential validations) is successful, and a trust ticket is issued by the merchant server whenever needed. Otherwise, the service is refused.

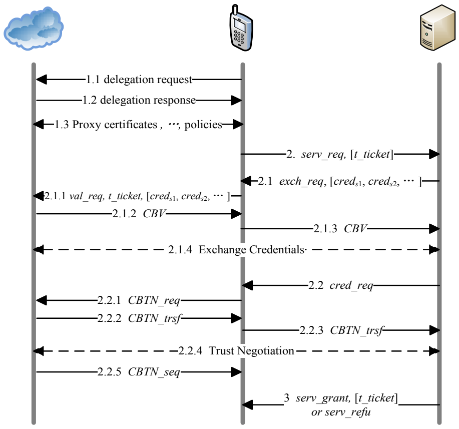

Fig. 2: The proposed CBTN protocol

Note that, if a valid trust ticket exists, step 3 should be skipped, that is step1-step2-step4. Otherwise, the protocol flow is step1-step3-step4.

3. Analysis

Offloading is beneficial when large amounts of computation are needed with relatively small amounts of data to be transmitted [6]. Trust negotiation is computational expensive. Mobile Users' credentials and associated policies are stored in the cloud. In our protocol, the cloud performs trust negotiation on the stored data. The user sends the credentials and policies only once when delegate the right to the cloud. During trust negotiations, the mobile user just sends a request to the cloud. Therefore, offloading process spends small amounts of communication. Offloading the trust negotiation process to the cloud alleviates the computational and communication overheads on mobile devices.

Users may feel uncomfortable to naïvely replicate sensitive credentials and policies to the cloud. To increase users' perceived trust, we introduce proxy certificates to our protocol, enabling the cloud to operate trust negotiations on users' behalf without giving up the control of their long-term secrets. And the restrictions on the proxy certificates avoids the cloud maliciously masquerade as the users. Furthermore, the message *CBTN_seq* allows users to check that whether the off-loaded trust negotiation strictly comply with the policies.

4. Conclusion

Automated trust negotiation is an effective way to dynamically establish bilateral trust between strangers wishing to conduct business. Integrating ATN into mobile commerce is a desirable method of trust establishments. However, the computational and communication complexity of ATN may cause high deployment costs and high operational overhead on mobile devices. Offloading the expensive trust negotiation operations from mobile devices to cloud computing is a solution. But it raises trust and security issues. In this paper, we propose a cloud-based trust negotiation (CBTN) protocol to efficiently offload trust negotiation to the cloud without compromising users' security. It increases user's trust by keeping the control of their own long-term credentials and checking whether the off-loaded trust negotiations are performed correctly.

5. Acknowledgments

This research is supported by the National Natural Science Foundation of China (No. 71101061), the Natural Science Foundation of Hubei Province for Innovative Research Team in University (No. 2011CDA116) and the self-determined research funds of CCNU (No. CCNU11A01039).

6. References

[1] E. Bertino, E. Ferrari, A.C. Squicciarini, Trust-X: A peer-to-peer framework for trust establishment, Ieee T Knowl Data En, 16(7) (2004) 827-842.

[2] M. Creeger, CTO Roundtable: Cloud Computing, Queue, 7(5) (2009) 2.

[3] T. Dahlberg, N. Mallat, J. Ondrus, A. Zmijewska, Past, present and future of mobile payments research: A literature review, Electronic Commerce Research and Applications, 7(2) (2008) 165-181.

[4] F. Hamad, L. Smalov, A. James, Energy-aware Security in M-Commerce and the Internet of Things, IETE Technical Review, 26(5) (2009) 357-362.

[5] R.B. Hu, D.L. Yang, R.H. Qi, Recommended Trust Evaluation Model in Mobile Commerce Based on Combination Evaluation Mode, Operations Research and Management Science, 19(3) (2010) 85-93.

[6] K. Karthik, L. Yung-Hsiang, Cloud Computing for Mobile Users: Can Offloading Computation Save Energy?, Computer, 43(4) (2010) 51-56.

[7] K. Siau, Z.X. Shen, Building customer trust in mobile commerce, Communications of the ACM, 46(4) (2003) 91-94.

[8] S. Tuecke, V. Welch, Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, in: RFC3820, (2004).

[9] W.H. Winsborough, K.E. Seamons, V.E. Jones, Automated trust negotiation, in: DARPA Information Survivability Conference and Exposition, (2000), pp. 88-102.

[10] T. Zhou, Y. Lu, The Impact of Privacy Concern on Mobile Commerce Users' Adoption Behavior, Chinese Journal of Management, 7(7) (2010) 1046-1051.