Mobile Cloud Computing: Security Analysis

DOI:
10.1109/MobileCloud.2017.27

Document Version
Accepted author manuscript

Link to publication record in Manchester Research Explorer

Citation for published version (APA):

Published in:

Citing this paper
Please note that where the full-text provided on Manchester Research Explorer is the Author Accepted Manuscript or Proof version this may differ from the final Published version. If citing, it is advised that you check and use the publisher's definitive version.

General rights
Copyright and moral rights for the publications made accessible in the Research Explorer are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

Takedown policy
If you believe that this document breaches copyright please refer to the University of Manchester’s Takedown Procedures [http://man.ac.uk/0Y6Bo] or contact uml.scholarlycommunications@manchester.ac.uk providing relevant details, so we can investigate your claim.

Download date:02. Oct. 2019
Mobile Cloud Computing: Security Analysis

Amina H Gamlo
School of Computer Science
The University of Manchester
Manchester, UK
amina.gamlo@postgrad.manchester.ac.uk
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia
ahgamlo@kau.edu.sa

Ning Zhang
School of Computer Science
The University of Manchester
Manchester, UK
Ning.Zhang-2@manchester.ac.uk

Omaimah Bamasag
Dept. of Computer Science
Faculty of Computing and Information Technology
King Abdulaziz University
Jeddah, Saudi Arabia
obamasek@kau.edu.sa

Abstract—This paper provides a comprehensive security analysis of Mobile Cloud Computing (M2C). It first describes two Cloud Computing (CC) scenarios. The first scenario describes one Cloud service provider providing all necessary services. The second scenario describes the case of two Cloud service providers collaborating to provide necessary services for one Client. The two scenarios focus on Authentication. Based on the use-case scenarios, the paper presents a generic model for M2C with a description of entities and interactions among them. Based on this model, the paper analyzes potential security threats, in order to specify a set of general security requirements, with an emphasis on Authentication. These requirements could provide guidance to future security solutions or Authentication solutions for M2C communication and/or risk assessment.

Keywords—Mobile Cloud Computing (M2C); Cloud Computing (CC); Security; Authentication

I. INTRODUCTION

Mobile Cloud Computing (M2C) has gained increasing interest with several challenges affecting its growth negatively, mainly regarding security and privacy of clients [1]. The aim is to provide on-demand cost effective reliable services for clients in a secure manner. In the following sections, we will define Cloud Computing and M2C and present the security challenges they present.

Cloud Computing (CC) is a network of parallel and distributed virtualized resources presented as a unit to provide clients with different IT services. CC has many essential features: cost-effectiveness, variability of resources, seamless self-service, flexibility and elasticity, reliability, location independence, and broad network access [1],[2].

Mobile Cloud Computing (M2C) refers to mobile network users accessing cloud services to optimize their mobile experience through utilization of resources provided by cloud providers.

M2C model has three components (mobile terminals, a mobile network, and cloud service providers (CSP)), each of which present different security requirements [3],[4]. At the mobile terminal component, mobile devices roam between networks leading to more security risks [5]. Also, roaming may require new authentication, reducing the performance due to latency [5]. At this component level, the security issues are malware, SW vulnerabilities, and clients’ lack of security awareness [3]. At the mobile network level, malicious attacks and information disclosure are common threats [3]. In CC, the system should ensure confidentiality, privacy, availability, integrity and recovery [2]. Unique attributes of CC make achieving this much harder due to delegating management of data and/or processes from the client to the CSP [3]. This delegation is typically communicated over different types of networks which also need protection [2]. Also, distributed resources of cloud environment and the multi-tenancy nature of CC lead to increased number of access points which in turn increases confidentiality, privacy, and integrity risks. Another issue is object reusability, which may result in intentional or unintentional confidentiality breach due to data reminiscence [2]. A reliable authentication solution is essential to minimize most security threats, allow secure sharing of resources, and provide a clear separation of the entities managed by the Cloud.

This paper focuses on security analysis of M2C. We first describe a generic model for M2C based on use-case scenarios description. We then analyze security threats against this model, which help us identify a set of security requirements to counteract the security risks. These requirements may guide our future design of an effective Authentication solution for M2C.

The rest of the paper is organized as follows: §II describes M2C use-case scenarios; §III presents a generic model based on these scenarios; §IV analyses potential security threats in the model; §V discusses observations about the model; §VI specifies a set of security requirements, while further discussions are presented in §VII. Finally §VIII concludes the paper.
II. USE CASE SCENARIOS

We present the use case of an online software development company, ‘SWShop.’ ‘SWShop’ has an online storefront to present their services and connect with customers. ‘SWShop’ has two types of operations: backend operations, which include development and testing of SW and frontend operation, represented by sale of SW products. ‘SWShop’ should accommodate all types of shoppers including those accessing the online storefront via mobile devices. SWShop developers should be able to collaborate, communicate and communicate with their customers easily to share projects’ milestones and plans securely. CC can provide an effective solution. We will present two CC solutions, first of which is 1C-Shop where all SWShop operations are provided through one CSP. 2C-Shop scenario explores the use of two CSPs collaborating to serve SWShop. 2C-Shop scenario will help us to observe the collaborative interactions between multiple CSPs. Follows is a description of a generic model of multiple CSPs. We refer to the generic model as (MC-Model).

A. 1C-Shop

<table>
<thead>
<tr>
<th>List of Credentials</th>
<th>Type</th>
<th>Required or Not</th>
<th>Creation</th>
<th>Use of the credential</th>
<th>Security recommendations</th>
<th>Validity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Root account log-in credentials</td>
<td>Username (Valid email) &amp; password</td>
<td>Required</td>
<td>At the beginning of registration</td>
<td>Log in to CSP website by Admin</td>
<td>- Change periodically - Don’t distribute to developers</td>
<td>Valid until changed</td>
</tr>
<tr>
<td>Individual user log-in credentials</td>
<td>Username &amp; password</td>
<td>Recommended</td>
<td>By admin for each developer</td>
<td>Log in to permitted resources by individual users</td>
<td>- Change periodically</td>
<td>Valid for 90 days, then users are forced to change them [6]</td>
</tr>
<tr>
<td>OTP</td>
<td>An app or an MFA device to provide 6-digit (OTP)</td>
<td>Optional</td>
<td>Set up by admin at any time</td>
<td>Used in conjunction with log-in credentials (for root and/or developer accounts)</td>
<td>- Keep device safe from others</td>
<td>One time use</td>
</tr>
<tr>
<td>Access key ID &amp; secret access key (SAK)</td>
<td>Random strings of characters</td>
<td>Required for developers</td>
<td>Set up by admin at any time and downloaded as a CSV file.</td>
<td>Used to sign programmatic requests from developers for AWS SDK, REST, or APIs. Also used to sign commands issued for command line interfaces (CLIs)</td>
<td>- Not recommended for root account - Should regularly be rotated either by admin or developers - Unused keys should be revoked</td>
<td>Until manually revoked</td>
</tr>
<tr>
<td>Temporary access key ID &amp; secret access key (TAK)</td>
<td>Random strings of characters</td>
<td>Optional</td>
<td>Set up by admin at any time and downloaded as PEM files</td>
<td>Used to sign programmatic requests made to AWS when using the AWS SDK, REST, CLI commands, or Query APIs for specific roles</td>
<td>These are issued with limited life for use in less secure environments or for temp user access.</td>
<td>Valid as long as a user is assuming the role</td>
</tr>
<tr>
<td>Public/Private Key Pair (PUK/PRK) for EC2</td>
<td>RSA 2048 bit private and public keys</td>
<td>Required</td>
<td>Set up by admin or developer and downloaded as PEM files</td>
<td>Used to request access to EC2 instances</td>
<td>Recommended to be rotated periodically</td>
<td>Until manually deleted by user or admin</td>
</tr>
<tr>
<td>Public/Private Key Pair for CloudFront service</td>
<td>RSA 2048 bit private and public keys</td>
<td>Required</td>
<td>Set up by admin for root account only and downloaded as PEM files</td>
<td>Used to create signed URLs for private content to be distributed to paying customers.</td>
<td>Recommended to be rotated periodically</td>
<td>Until manually revoked or deleted by admin</td>
</tr>
</tbody>
</table>

A single Cloud Solution means that all ‘SWShop’ backend & frontend operations are provided through a single CSP. Amazon Web Services (AWS) provide services that fall under the three Cloud service models, IaaS, PaaS, and SaaS. For this scenario, Frontend operation and backend operations will all be provided through AWS.

The first step is to register for AWS services and complete security set up. Security set-up includes the creation of security groups for SWShop, setting traffic rules, and creating an Amazon Directory with individual developers’ login usernames and passwords and access permissions. It also allows for activation of Multi-factor Authentication to create One-Time Passwords (OTP) for the company root account and/or individual developers (IAM users). It also includes the creation of access keys, RSA 2048 bit private/public key pairs, and X509 certificate via the security credentials web page. Detailed information on these credentials and how they are used is provided in Table 1. Some of these credentials are optional, while others are used with some AWS services.
The general set up for 1C-Shop solution depicts an outsourced private cloud [7] as shown in Fig. 1, where Virtual Private Cloud VPC service designates a set of AWS virtual resources to SWShop AWS account.

An isolated portion of the AWS cloud is created for SWShop to launch EC2 instances with private addresses in the specified range rather than randomly assigned public IP addresses [6]. Network ACLs and security groups are used to provide stringent control over inbound and outbound traffic to SWShop instances.

Frontend operation of SWShop is 3-tier web application architecture. The necessary resources including compute instances, database, content distribution, and DNS services will be allocated on AWS. Each phase of backend operations-implementation and Testing of the SW-enlists different AWS resources. As any project progresses, developers can allocate the required resources to build the needed architecture for that phase and release unnecessary resources.

As for payment, AWS provides clients with a pay-as-you-go option for the provisioned services. SWShop receives an email alert for the accumulated monthly charges. The bank settles the payment after SWShop approval.

The interacting entities in this solution are:

1) CSP components including MC-AWS, and AWS-Sx, which refer to individual AWS services.

2) SWShop, represented by Admin, Developers Uj, TUi, which refers to temp users assuming a defined role such as the role of a website visitor.

The credentials summarized in Table 1 are employed to provide secure transactions between these entities.

**B. 2C-Shop**

The 2C-Shop refers to a scenario that utilizes two CSPs: AWS for backend operations while frontend operation (hosting a website) is provided by a different CSP X. The interacting entities in this solution are:

1) AWS components including MC-AWS, and AWS-Sx, which refer to individual AWS services.

2) CSP X components: CSP X management console MC-X, and X-Sy, Which refers individual CSP X services.

3) SWShop, represented by Admin, Developers Uj, TUi, which refers to temp users assuming a defined role such as the role of a website visitor.

The Credentials structure for this scenario is very similar to that presented for 1C-Shop. The credentials necessary to access CSP X resources should augment the previously listed credentials (Table 1) to include credentials for SWShop-to-CSP X interactions: login credentials for root account and individual users of CSP X, OTPs for root account and individual users, PUK/PRK pairs, SAKs, and TAKs.

The general architecture for 2C-Shop- presented in Fig. 2- shows SWShop utilizing the services of AWS and CSP X to perform their frontend and backend operation.

As with 1C-Shop scenario, the Admin needs first to register for both CSP AWS and CSP X services. Then, security set up should be completed for both CSPs. The Admin holds the root account credentials to both CSPs.
Resources needed for website hosting will then be deployed on CSP X while the team of developers can start working on projects through services on CSP AWS. As any project progresses, SWShop developers can allocate the necessary resources from CSP X to build the needed architecture for that phase and release unnecessary resources.

As for payment, both AWS and CSP X offer pay-as-you-go plans. SWShop store receives email alerts for the monthly charges. The bank settles the payment after client’s approval.

III. GENERIC MODEL MC-MODEL

Based on the scenarios presented in previous sections, we introduce a generic model for M2C. MC-Model describes entities and interactions among them for multiple CSPs collaborating to perform tasks for a single client. The interacting entities are:

1. Client C, represented by:
   - **Admin** which refers to the administrator
   - **Uj** which is an employee of C
   - **TUj** which is a temp user assuming a pre-defined role such as the role of a storefront shopper

2. For n CSPs, each CSP is represented by:
   - **MCSPx** refers to each CSP X Management Console, where x is between 1 and n.
   - **Sy-CSPx** refers to each service offered by each CSP. So, S2-CSP1 would refer to service S2 offered by CSP1.

3. Financial Third Party, F.

Varying credentials can be used to secure these interactions. The credentials structure for MC-Model is as follows [8]:

C1: Username and password (PWDroot) are the login credentials for the root account. It is required. It should be changed periodically and not distributed to other users. It stays valid until changed.

C2: Individual user accounts credentials, PWDj, are for login by an individual user. They should be set by the admin and distributed to other users affiliated with the Client C. They stay valid for a set period then users are required to change them.

C3: OTP’s are provided by an MFA Hardware or software generator. It can be used for the root account (OTProot) and/or the individual user accounts (OTPtr). They are optional and used once.

C4: Non-certified Public/Private Key Pairs can be created for an individual user (PRKj /PUKj) or the root account (PRKroot /PUKroot). They can be created by the CSP, or by a third-party. The PUK is saved by both CSP and the user. The PRK is kept by the user and is used to encrypt user requests to access services.

C5: Certified Public/Private key pair (X509-PUK/PRK). The user signs SOAP-protocol requests to service interfaces with his X509-PRK.

C6: Secret access keys are created for the account admin (SAKroot) or an individual user (SAKj). The SAK is a shared secret between the User and the CSP used to sign programmatic requests to Cloud services. When a User application issues a request for a Cloud service, SAK is used to calculate an HMAC signature transmitted as part of the HTTP Authorization header or as part of the URL.

Credentials C4-C6 are recommended to be rotated periodically. They stay valid until manually deleted by the user or revoked by an admin.

C7: Temp access Keys (TAK) are created for temporary access roles TUj. They are used in a similar manner to SAK but are issued with limited life for temporary roles. A role is a task defined by the admin with attached permissions to access specified resources and credentials. An example is the role of un-registered shopper of an online storefront. This role may allow read-only permissions to some information of the product database of the store and includes the credentials necessary to access it. They are valid for as long the role is assumed by a user.

Table II provides a summary of all credential items, and interacting entities showing who (Entities) knows what (credentials) for MC-Model.

<table>
<thead>
<tr>
<th>TABLE II. WHO KNOWS WHAT FOR MC-MODEL</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Items</strong></td>
</tr>
<tr>
<td>PWDroot</td>
</tr>
<tr>
<td>PWDj</td>
</tr>
<tr>
<td>OTProot</td>
</tr>
<tr>
<td>OTPj</td>
</tr>
<tr>
<td>PUKroot</td>
</tr>
<tr>
<td>PRKroot</td>
</tr>
<tr>
<td>PUKUser</td>
</tr>
<tr>
<td>PRKUser</td>
</tr>
<tr>
<td>X509-PUK</td>
</tr>
<tr>
<td>X509-PRK</td>
</tr>
<tr>
<td>SAKIDroot</td>
</tr>
<tr>
<td>SAKroot</td>
</tr>
<tr>
<td>SAKID</td>
</tr>
<tr>
<td>SAK</td>
</tr>
<tr>
<td>TKID</td>
</tr>
</tbody>
</table>

Generally, for two CSPs collaborating to serve one client, the workflow of Cloud service provision starting from registration is illustrated in Fig 3.

For M2C Model, the execution workflow goes as follows:
(1) The admin registers for CSPs and creates credentials C1-C7 described above.

(2) Admin or individual users $U_i$ use credentials from (1) to login to CSPs.

(3) Admin or individual users $U_i$ request needed CSP services through MCSP.

SSL protects communicated messages secrecy and authentication for steps (1-3).

(4) As the project necessitates, Admin or individual users $U_i$ access one or more of the allocated services in (3). Security of access depends on the type of service requested. SSH may be used to provide a secure remote login to compute instances. HMAC signatures may be employed for other types of services.

(5) The running services may access other services provided by the same CSP.

(6) The services running on one CSP may also access other services provided by other CSPs.

Until the task at hand is completed, steps 2 to 6 may be repeated.

Each of these interactions requires two types of messages to be communicated between the entities: request ($REQ$) and response ($RES$). Both types may contain supporting data.

From the workflow described above, we deduce that two CSPs collaborating and interacting to provide services and perform tasks for one Client C are sufficient to show all types of interactions in MC-Model.

Accordingly, the interactions between entities can be classified into three types (Fig 4):

1. **User to Management console of CSP (U-to-MCSP):**
   - Registration, security set-up, and users requesting services from CSPs are examples for this type of interaction.

2. For this type of interaction, authentication may be provided through Passwords, MFA and OTPs.

3. **User to Service (U-to-S):** An example of this interaction is when a user requests a service $S$ from a CSP. For this type of interaction, authentication may be provided through Public/Private key pair, X509 certificates, or Secret Keys depending on the service $S$ type.

4. **Cloud Service to another Cloud Service:** There are two distinct types of this interaction:
   - **Service-to-Service provided by the same CSP (S-to-S):** This type of interaction occurs when one cloud service $S_i$ requests another service $S_j$ provided by the same CSP. For this type, a user authenticated by $S_i$ is assumed to be trusted by $S_j$. Moreover, the requesting service $S_i$ trusts $S_j$ because they reside in the same CSP domain.
   - **Service $S_i$ on CSP X to another Service $S_j$ on CSP AWS (S$_i$-CSPx-to-S$_j$-CSPy):** This type of interaction occurs when one cloud service $S_i$ provided by CSP X requests another service $S_j$ provided by another CSP Y. Authentication should be provided for both services mutually to safeguard the user’s data and processes as well as other clients of both CSPs.

Based on these interactions, we can identify security threats/risks of this model. Then, we can specify requirements for effective and secure M2C service provision.

IV. THREAT ANALYSIS

While M2C has great potential of providing dependable services for a broad spectrum of clients, it may offer opportunities for misconduct on behalf of CSPs, Clients, or outsiders. This section analyses the security threats to various entities in this model. Threats against secure interactions of this model can be classified into the following [9]:

Threats related to standards and regulatory authority:

Currently, M2C lacks proper security standards or suffer compliance risks in cases where security standards are defined due to lack of governance for audits [9].
CSP related threats are related to the services offered by the CSP and ways they can be exploited. These threats include insecure API’s which Clients use to access some of the services [9].

Data related threats might include data redundancy, data leakage, and data unavailability.

Network related threats have a significant impact on the security of the M2C model since M2C depends heavily on networking and Internet communications.

Access related issues include impersonation by an outsider user, a dishonest administrator within the CSP, or by another Cloud user.

The security threats against M2C are enumerated as follows:

A. Impersonation

Impersonation is the theft of another entity’s identity. It may occur in various ways:

- A malicious outsider may steal the identity of a legitimate user through Phishing, Pharming or PWD guessing attacks to acquire the login credentials. It may lead to leakage of Client’s data, loss of service to Client’s beneficiaries, or hijacking of resources allocated to the Client.

- A malicious insider is a dishonest admin within CSP or a legitimate user of another Client C2. The attack may occur when a scheduler in the virtualization layer favors C2 requests over C1 requests. Also, lack of secure encapsulation or vulnerability in the virtualization layer may lead to granting C2 malicious access to C1’s services or unlawful release of C1’s data.

- Impersonation of a CSP: An attacker may impersonate a legitimate CSP during RESCSP sent to the client in response to an access request allowing for unauthorized access to insider assets of C or deny services to C’s beneficiaries.

B. Tampering of communication messages

Another threat is the Altering REQ or RES communication messages between entities. The alteration may include replaying, delaying, or modification of messages in an attempt to gain unlawful access or misuse assets of C. Service Hijacking may occur due to vulnerabilities of communication protocols leading to leakage of data or loss of service to legitimate users.

C. Eavesdropping

An attacker may listen in on communication channels in C-to-CSP or CSP-to-CSP communication to get private information of C or CSP to fashion a subsequent attack. The attack could lead to loss of CSP services or leakage of vital data such as secret security credentials, financial, or competitive information of C.

D. Unauthorized access to data at rest

A malicious insider or outsider may unlawfully gain access to data stored on CSP storage resources leading to leakage of sensitive data causing financial loss to both C and CSP.

E. Denial-of-Service (DoS) attacks

An outside or a legitimate user from a different Client of the CSP may hog resources or communication channels used by a user of C. Due to high dependence of M2C on HTTP and REST architecture, the attacker can flood the web servers with HTTP requests [9].

F. Disputes

Intentional or erroneous, false requests made by a legitimate user of C may lead to financial loss or affect other Cs of the CSP. Moreover, intentional or erroneous false requests made by the CSP may cause financial loss or denial of resources paid for by C.

V. Observations

Based on the Generic Model description and the threat analysis, we can make the following observations:

O 1. The model presents multiple heterogeneous entities with varying IT capabilities, number of users, and business models. M2C clients may be a hospital, a university, or a multinational company to name a few. A Client may have a few users or hundreds of users. It may have a privately hosted data center or just rely on PCs provided to its employees.

O 2. The interacting entities communicate over the Internet. Hence, they may utilize protocols standardized for web communication and web services. They are also vulnerable to risks and threats usually present for Internet communication.

O 3. The types and number of security credentials necessary to secure the interactions among the model entities are diverse in order to fit the model variable nature. Credential types vary between PWDs, OTPs, uncertified PUK/PRK pairs, X.509 PUK and matching PRK, TAKs, and multiple shared SAKs.

O 4. The security scheme is complicated with long processes and many keys to create and manage (observation 3). This complicated scheme may be difficult to apply for low-resource users such as a mobile user.

O 5. There are two phases of Authentication in a C-to-CSP interaction. The first phase is initial authentication to gain access to management console, which is done during login to MCSP. The second is Authentication when accessing services agreed upon after successful login, where all Client requests are authenticated to the CSP. This kind of Authentication is ongoing for as long as a user is utilizing a service.

O 6. The ongoing Authentication of service requests is one-way since it only authenticates the user to the CSP service.

O 7. Collaboration between various CSPs doesn’t currently follow processes or protocols standardized for CSP-
to-CSP communication, which may jeopardize data and processes of C necessary to finish the task at hand [10].

Due to competition, information presented by most current CSPs don’t cover detailed methods and processes followed during service provision to clients [11]. CSPs don’t provide sufficient information on processes may be employed for CSP-to-CSP collaboration.

VI. SECURITY REQUIREMENTS

Based on the MC-Model, threat analysis, and observations we made in the previous sections, we derive a set of desirable security requirements for secure and effective M2C services. These requirements are closely related to Authentication.

A. Entity Authentication

Entity Authentication is to ensure that every interacting entity has the identity it claims. It must be mutual, and it must apply to all entity interactions (C-to-MCSP, C-to-S, and S-to-S) within the same domain or cross-domains.

B. Message Authentication

Message Authentication ensures that messages are received without modification. It should apply to all REQ and RES messages. Keyed hash values, Message Authentication Code, or digitally signed tokens are typical methods used to ensure messages’ authenticity [12].

C. Continuous Authentication Process

Authentication should continue throughout the resource utilization operation, which includes Registration (Identification), initial authentication during user login to services, and ongoing authentication: as two entities continue to interact (all REQ and RES messages).

D. Non-Repudiation (NR)

NR should enhance the level of trust between entities such that all their actions should be unforgeable, undeniable and traceable in order to guarantee fair resolution of any dispute. There are two types: NR of Origin, which ensures that the sender doesn’t deny sending a message and NR of Receipt, which ensures that the receiver doesn’t deny receiving that message. Both types of NR must be ensured for all entity interactions.

NR can be accomplished using various cryptographic mechanisms including digital signatures and notarization [12] and off-line trusted third parties (TTPs) may be used to provide the NR properties [13].

E. Elasticity

Elasticity refers to the ability of a system to grow as needed without loss of functionality [2]. Elasticity is an inherent feature of M2C. An effective Cloud authentication solution should support the addition of new Clients, users within a Client, CSPs, or services provided by the CSP.

F. Interoperability

Interoperability refers to the ability of heterogeneous systems to interact and collaborate efficiently and securely. For M2C, it would allow users to employ services offered by multiple CSPs in a seamless manner.

G. Performance

Any security solution should not hinder service provision; rather, it should enhance it. It should take into account low-resource users such as those on mobile devices. In other words, mobile users should have the same experience the as non-mobile users [14]. Thus, the solution should be low weight in terms of computational cost for the user and communication overhead (number and size of exchanged REQ and RES messages).

H. Maintain underlying features of CC [15]

Any M2C security solution should maintain underlying features of CC such as Multi-tenancy, the responsiveness of Cloud architecture to infrastructure or service problems, portability of software and data, access from any user device, and massive data processing.

Other security requirements are privacy, confidentiality, and availability of services.

VII. FURTHER DISCUSSION

MC-Model involves participation and communication of multiple entities with data traveling between entities to request and fulfill services or payment. Data items exchanged are owned by different entities and have varying levels of sensitivity. Entities should have varying access privileges to those items accordingly. Our research focuses on an effective authentication solution for M2C to counteract many of the security threats identified in the threat analysis. All impersonation attacks mentioned in threats A, B, and C can be offset by employing an effective Authentication. Impersonation attacks, if successful, can lead to more attacks on the resources of the Client. Moreover, message alteration (threat D) although may not be prevented by strong authentication, it may be detected to save interacting parties from any harmful consequences. Also, as threat F states, unauthorized access to data at rest can be stopped with strong authentication. Furthermore, (threat H) can also be resolved through the aid of an effective authentication solution.

M2C is appealing to more and more clients to provide multifaceted types of services. The security aspect is a major concern. Conducting a security analysis is the first step towards securing M2C. It paves the way for setting design requirements for an effective authentication solution taking advantage of best features of state-of-the-art authentication methods and remedying the gap without introducing any new risks.

VIII. CONCLUSION

Although M2C provides IT solutions to an increasing number of enterprises and individuals, arising security issues may hinder effective utilization of such solutions. Effective and efficient security solutions should allow more clients to use M2C securely. In this paper, we have devised a generic model for M2C and analyzed security issues and potential risks based on it. Based on this security analysis, we have indicated a set of security requirements with emphasis on Authentication. This analysis will be used to guide us to the next phase of our work, which is the design of a reliable and secure Authentication solution for M2C.
REFERENCES


