Design of the Communication Protocol for a Digital Currency System using Model Checking

Abhishek Pandey, Tripti Gupta, Mahil Carr
1Associate Professor
1,2,3IDRBT, Hyderabad, India
e-mail: abhishekpandey31@gmail.com1, er.tgupta@gmail.com2, mahilcarr@idrbt.ac.in3

Abstract - This paper proposes a communication protocol to be used in a new digital currency system to be implemented in the Indian context. This protocol attempts to provide secure communication between a high-performance distributed database application, which will hold currency values in digital form and an end-user can do transfer of value using a mobile device. The protocol is modeled from SSL. The attempt made here is to model check the protocol for security vulnerabilities under various security threats such as man-in-the-middle attack, replay attack and denial of service attack.

Keywords: Model Checking; Promela; Communication Protocol; Mobile; Man-in-the-Middle Attack; Replay Attack; DOS Attack; OCSP

I. INTRODUCTION

Software applications are becoming increasingly complex and they intrude into every aspect of our daily lives. The criticality and safety of software applications in fields such as aerospace, medical equipment, finance, defense etc. has become a prime concern of software designers and developers. Model checking is an approach that can test and detect flaws in software design. The model checking approach has been very successful in finding design errors in many areas. A very basic protocol – Needham’s protocol – that shares a secret communication between two entities – was thought to be robust for seventeen years and it was heavily used in financial systems that required secure communication. However, model checking of the protocol revealed flaws in its design [15]. In the work presented here we attempt to model check a proposed protocol for a digital currency system. Moreover, we outline broader system engineering challenges that have to be addressed for a successful implementation of the protocol in the context mobile devices having weak processing capacity to be able to handle traditional cryptographic frameworks such as public key infrastructure.

Model checking is a technique that can automatically verify finite state software (reactive) systems, communication protocols and circuit designs. Software or protocol design specifications (a high level representation of) are modeled as state transition graphs. The security properties to be verified or checked are specified using Linear Temporal Logic (LTL). LTL is a mathematically precise notation for expressing properties about the relation between the state labels in executions. [16]. LTL formula can be used to express properties such as liveness (“something good must eventually happen”), safety (“something bad should never happen”), fairness (“something must happen infinitely often”) etc. using linear temporal operators (e.g. always, eventually, until etc.) [14]. LTL is used to express a claim (here a security property to be verified). The negative claim is translated into a Buchi automaton. The Buchi automaton is executed through the state transition graph of the model to be checked. If the intersection state space (of the Buchi automata and the model) is empty, the security properties of the model are proved. Otherwise, a counter example is generated.

In practice, Process Meta Language (PROMELA) is used to specify the protocol. LTL is used to specify the security property to be verified. Both the specification and the properties can be automatically verified using model checking software tools such as JSPIN [10].

In this paper, we briefly describe the proposed financial system application where the desired protocol is to be deployed. Then we outline the model of the protocol and show its PROMELA description. Model checking is performed using JSPIN to demonstrate the behavior of the protocol under three possible security threat scenarios – man-in-the middle attack, replay attack and denial-of-services attack. We demonstrate the protocol is robust against these attacks. In the final section, we outline the broader challenges required to make the protocol implementable in a context where low-end mobile computing devices are to used for making financial transactions.

II. PROPOSED DIGITAL CURRENCY SYSTEM FOR INDIA

Several models of digital or electronic money have been proposed and several pilots have been run using smart cards [17]. However, with today’s level of sophistication the smart card option for digital currency have to be abandoned due to possible side channel attacks that could compromise security (AES keys can be compromised within 0.01 seconds). Therefore, here we propose a model for a digital currency system where the actual value of the digital currency is stored in the form of purses in a central server (Figure 1) and transactions between purses are made through mobile phones.

In the proposed payment system the Remitter wants to transfer some value from his purse to the purse of the Remittee. To enable this transfer using the mobile device from one purse to another (the world of purses reside in a distributed database application) a trusted entity is used. We will name this entity as a ‘Trusted Third Party (TTP)’ for convenience to distinguish this system from core banking applications that handle bank accounts. The TTP maintains purse of the mobile holders and debits the requested value from the Remitters’ purse and credits it to the Remittees’ purse and sends the acknowledgement to both the parties. (See Figure 1)
III. MODELING THE COMMUNICATION PROTOCOL

The desired communication protocol for the Digital Currency System is to ensure that financial transactions happen in a secure manner. GSM security framework (using A3/A5/A8) is not robust enough for financial transactions. Therefore an application layer security needs to be designed in the mobile context that will be of similar strength like SSL. In this section the design of the communication protocol is explained. This protocol assumes that a suitable mobile encryption technique is available (a robust cryptographic technique that is sustainable against brute force or any other kind of attack). Our model attempts to address the network security aspects of the protocol. The sequence diagram of the protocol is described below.

A. Sequence Diagram for the Desired Protocol

Sequence Diagram shows how objects communicate with each other in terms of sequence of messages.

The sequence diagram for the digital currency system in the Figure 2 depicts how the mobile device and TTP interact [12].

The protocol sequence is represented below:
1. The Mobile and the TTP are in their initial states, called Mobile_Primal_State and TTP_Primal_State respectively.
2. The Mobile sends its initial Hello message to the TTP along with its digital certificate and moves to the Mobile_Init_Request state.
3. TTP authenticates the mobile, moves to TTP_Init_Reply state and sends the acknowledgement of the mobiles’ initial message along with its certificate.
4. Mobile authenticates the TTP.
5. Both Mobile and the TTP move to the Ready State, indicating that now both are ready for the actual message transfer.
6. Actual message transfer begins.

B. Model (State Diagram) of the Desired Communication Protocol

This section describes the desired communication protocol with the help of a State Diagram. The symbols used in state graph are given in Table I. This state diagram in Figure 3 explains that how the sequence diagram in Figure 2 has been modelled using PROMELA. The arrows in the state diagram are labelled using PROMELA syntax and the code for the TTP side (Figure 6). The code for the mobile device is not shown here due to space constraints.
IV. VERIFYING THE ROBUSTNESS OF THE COMMUNICATION PROTOCOL AGAINST VARIOUS ATTACKS USING MODEL CHECKING

The desired communication protocol is secure against the attacks like Man-in-the-Middle (MIM) attack, Replay attack and Denial-of-Service (DOS) attack. ‘Secure’ here means that if such attacks happen then the proposed protocol terminates gracefully without carrying out the financial transaction. Due to space constraints only the PROMELA code for the MIM attack has been shown.

A. Man-in-the-Middle (MIM) Attack
The name ‘Man-in-the-Middle’ is derived from the basketball scenario where two players intend to pass a ball to each other while one player between them tries to seize it. In MIM attack an intruder intercepts a message coming from one entity, injects false information in it and then passes on the message to the intended receiver. The PROMELA code of MIM attack is shown in Figure 4.

<table>
<thead>
<tr>
<th>Symbols/ Words used in State Diagram</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>!</td>
<td>Denotes that a message is sent.</td>
</tr>
<tr>
<td>?</td>
<td>Denotes that a message is received.</td>
</tr>
<tr>
<td>Mobile_Trust_List</td>
<td>Contains the information used by the mobile to authenticate the TTP’s certificate.</td>
</tr>
<tr>
<td>Mobile_Id</td>
<td>Used to identify the mobile uniquely.</td>
</tr>
<tr>
<td>TTP_Trust_List</td>
<td>Contains the information used by the TTP to authenticate the mobiles’ certificate.</td>
</tr>
<tr>
<td>TTP_Id</td>
<td>Used to identify the TTP uniquely.</td>
</tr>
<tr>
<td>Secret</td>
<td>Shows that the messages are sent in encrypted form.</td>
</tr>
</tbody>
</table>

Figure 4. Promela code for MIM Attack

B. Denial-of-Service Attack
In Denial-of-Service (DOS) attack the attacker floods the network with useless traffic so that the machine or network resource becomes unavailable to the intended user.

C. Replay Attack
The replay attack is similar to Man-in-the-Middle attack, but in it the intruder not only stores the message, but also replays the message at a later time to gain advantage in the system.

V. CHALLENGES IN ACHIEVING THE DESIRED COMMUNICATION PROTOCOL
This section addresses the wider challenges of overcoming the weak processing capacities of low-end mobiles that cannot handle a public key framework. In a PKI framework public keys have to be exchanged and verified using digital certificates (X.509). But mobile phones cannot do traditional certificate validation like processing CRL (Certificate Revocation List) etc. Local processing of CRL for certificate validation are burdensome for basic handsets. The size limitations of the
Short Messaging Service (SMS) and restricted battery power of the basic handsets is an impediment to cryptographic processing [1].

As an alternative to X.509 a small-sized certificate called ‘Short-Lived Certificate’ has been suggested by Lee et al [1]. Short-Lived Certificates do not have extensions needed for certificate path validation. So the size of short-lived certificate is reduced [1]. This coupled with Online Certificate Status Protocol (OCSP) [1] provides a framework for secure communication.

In OCSP, the mobile sends the short-lived certificate of TTP to the OCSP. OCSP, on the behalf of mobile, downloads the CRL from directory verifies it and sends the certificate verification result to the mobile. Figure 7 explains the proposed communication protocol. The sequence diagram for the same is given in Figure 5. Our future work will attempt to model check the framework proposed by Lee et al.

active proctype TTP()
{ int sender = 0;
int receiver = 0;
int TTP_secret = 6;
int TTP_trust_list = 4;
end:

do ::TTP_State == PrimalState && nempty(Channel12) ->

Channel12!sender,receiver;
empty(Channel12);
if ::sender!=TTP_trust_list && receiver!=TTP_secret ->
TTP_State = TTPInitReply;
print("Man in the middle attack TTPside detected\n");
TTP_State = PrimalState;
fi

::TTP_State == TTPInitReply && nempty(Channel11) ->
print("TTP acknowledges the Mobiles’ initial message\n");
Channel11!15;4;
TTP_State = Ready;

::TTP_State == Ready ->
TTP_State = TTPTransactionAck;
od
}

Figure 6. Promela code for TTP side

VI. CONCLUSION

This paper shows how model checking as a technique can and must be employed to develop robust communication protocols. To implement a secure financial transaction protocol a strong mobile encryption technique is required and a new cryptographic system on similar lines to public key infrastructure needs development. This is perhaps the key critical challenge to make digital currency a reality.

REFERENCES