International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056
Volume: 02 Issue: 08 | Nov-2015 www.irjet.net p-ISSN: 2395-0072

End-to-End Secure Social Networking Application

Ms. Namrata Kamble[1], Ms. Bhagyashri Chavan[2], Ms. Irawati Bhole[3], Ms. Tejaswini Apate[4], Mr. Pravin Kothawale[5]

[1234]*Student, Computer Science and Engineering, Sanjay Ghodawat Institute, Atigre, Maharashtra, India*

[5]Assistant Professor, Dept. of Computer Science & Engineering, Sanjay Ghodawat Institute, Atigre, Maharashtra,

---

Abstract – ***Now-a-days everyone is having smart phones in their hands. Everyone is using many social networking sites and applications to connect to their friends and relatives. Messaging services are an increasingly popular method for communicating over the Internet. Unlike e-mail, messaging service allows users to see whether a chosen friend or co-worker is connected to the Internet. Typically, the messaging service will mark a user if somebody on the user's list of correspondents is on-line. However, does anyone think about whether their data is secure? Contrary to well-known instant messaging services, no additional authentication mechanisms other than the phone number are used by these applications. In this paper, we focus to create a system which transfers messages, documents, and images securely.***

Key Words: *Social networking, Security, Public Key Cryptography, RSA etc.*

1. INTRODUCTION

A social network is a website that brings people together to talk, share ideas and interests, or make new friends. This type of collaboration and sharing of data is often referred to as social media. Today we see many social networking sites/applications like Facebook, Whatsapp, hike and many more. These are providing different features to attract users within different age groups. Some social networking sites/applications claim about their security. But doesn't reveal about which encryption algorithm they are using. In this paper, we are using RSA algorithm to provide secure data transfer among users.

1.1 Why public Key Cryptography?

The primary advantage of public-key cryptography is increased security and convenience: private keys never need to be transmitted or revealed to anyone. In a secret-key system, by contrast, the secret keys must be transmitted (either manually or through a communication channel) since the same key is used for encryption and decryption. A serious concern is that there may be a chance that an enemy can discover the secret key during transmission. Another major advantage of public-key systems is that they can provide digital signatures that cannot be repudiated. Authentication via secret-key systems requires the sharing of some secret and sometimes requires trust of a third party as well. As a result, a sender can repudiate a previously authenticated message by claiming the shared secret was somehow compromised by one of the parties sharing the secret.

Table -1: Conventional and Public key Encryption

Conventional Encryption	Public Key Encryption
1. The same algorithm with the same key is used for encryption and decryption.	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.
2. The sender and receiver must share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
3. The key must be kept secret.	3. One of the two keys must be kept secret.
4. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	4. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key.

1.2 RSA Algorithm

The **RSA** scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and n - 1 for some n. A typical size for n is 1024 bits, or 309 decimal digits. RSA makes use of an expression with exponentials. Plaintext is encrypted in blocks, with each block having a binary value less than some number n. Encryption and

decryption are of the following form, for some plaintext block M and ciphertext block C. Both sender and receiver must know the value of n. The sender knows the value of e, and only the receiver knows the value of d. Thus, this is a public-key encryption algorithm with a public key of $PU = \{e, n\}$ and a private key of $PR = \{d, n\}$.

For this algorithm to be satisfactory for public-key encryption, the following requirements must be met.

1. It is possible to find values of e, d, n such that $Med \bmod n = M$ for all $M < n$.

2. It is relatively easy to calculate $Me \bmod n$ and $Cd \bmod n$ for all values of $M < n$.

3. It is infeasible to determine d given e and n.

The ingredients are the following:

p, q, two prime numbers	(private, chosen)
$n = pq$	(public, calculated)
e, with $\gcd(f(n), e) = 1; 1 < e < f(n)$	(public, chosen)
$d = e^{-1} \pmod{f(n)}$	(private, calculated)

The private key consists of $\{d, n\}$ and the public key consists of $\{e, n\}$. Suppose that user A has published its public key and that user B wishes to send the message M to A. Then B calculates $C = M^e \bmod n$ and transmits C. On receipt of this cipher text, user A decrypts by calculating $M = C^d \bmod n$.

For example, the keys were generated as follows.

1. Select two prime numbers, $p = 17$ and $q = 11$.

2. Calculate $n = pq = 17 \times 11 = 187$.

3. Calculate $f(n) = (p - 1)(q - 1) = 16 \times 10 = 160$.

4. Select e such that e is relatively prime to $f(n) = 160$ and less than $f(n)$; we choose $e = 7$.

5. Determine d such that $de = 1 \pmod{160}$ and $d < 160$.The correct value is $d = 23$, because

$23 \times 7 = 161 = (1 \times 160) + 1$; d can be calculated using the extended Euclid's algorithm. The resulting keys are public key $PU = \{7, 187\}$ and private key $PR = \{23, 187\}$.

The example shows the use of these keys for a plaintext input of $M = 88$. For encryption,

we need to calculate $C = 88^7 \bmod 187$. Exploiting the properties of modular arithmetic, we can do this as follows.

$88^7 \bmod 187 = [(88^4 \bmod 187) \times (88^2 \bmod 187) \times (88^1 \bmod 187)] \bmod 187$

$88^1 \bmod 187 = 88$

$88^2 \bmod 187 = 7744 \bmod 187 = 77$

$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$

$88^7 \bmod 187 = (88 \times 77 \times 132) \bmod 187 = 894{,}432 \bmod 187 = 11$

2. HOW SECURITY IS PROVIDED?

Asymmetric algorithms rely on one key for encryption and a different but related key for decryption. These algorithms have the following important characteristic:

It is computationally infeasible to determine the decryption key given only knowledge of the cryptographic algorithm and the encryption key.

In addition, some algorithms, such as RSA, also exhibit the following characteristic:

Either of the two related keys can be used for encryption, with the other used for decryption.

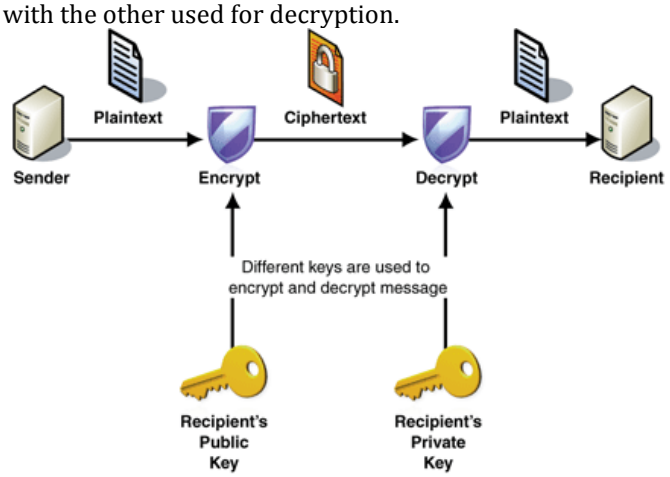


Fig -1: Public Key Cryptography

1.Each user generates a pair of keys to be used for the encryption and decryption of messages.

2. Each user places one of the two keys in a public register or other accessible file.

This is the public key. The companion key is kept private., each user maintains a collection of public keys obtained from others.

3. If user B wishes to send a confidential message to user A, B encrypts the message using A's public key.

4. When A receives the message, he decrypts it using his private key. No other recipient can decrypt the message because only A knows A's private key.

The application uses this algorithm for secure data transfer among users.

3. LITERATURE SURVEY

In this paper, we are interested in how secure the data transfer is. Social networking applications like Whatsapp, Facebook claims that the user data is secure but they don't reveal their encryption algorithm which they are using so it cannot be said that these are secure. Some people said that Whatsapp encryption can be breakable.

4. EXISTING SYSTEM

Some applications provide data transfer among users but they have some security threats. These can results in less security for user data. User data will remain unsafe. The registration includes only phone number of the user, no any authentication is performed. An

intruder can get access to user data during transmission.

4.1 Disadvantages of existing system

- Lack of security to user data.
- No fake product advertisements like other social networking sites.

5. PROPOSED SYSTEM

Our application aims to provide communication over a network. The user needs to register first to the application. The user need to login to the system by giving the password which he given during registration. That password is used only for login to the system. Whenever user wants to login to the application the password must be entered. The user can see other users from his contact list who already using the application. For connecting to other users, the user can select a contact. The user can send data to others. Here, data refers to text, images, documents etc. The user can log out after end of the communication.

5.1 Advantages of Proposed System

- Secure data transfer.
- User can send text, documents and images securely.

6. ARCHITECTURE

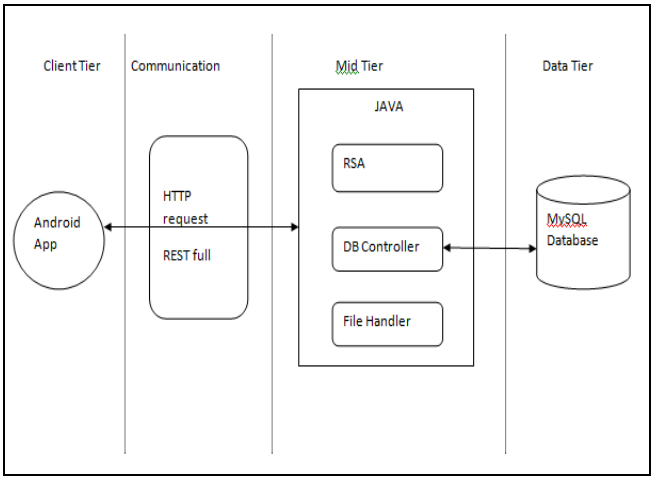


Fig -1: System Architecture

The system architecture shows client tier, mid tier and data tier. Client tier consists of android application, communicates through HTTP with the mid tier. The mid tier contains RSA, DB controller and file handler implemented in java. Database controller is connected to the MySQL database.

6.1 REST client/ Web service

What is REST?

A few years ago, it seemed that web services SOAP was the solution to all the problems of B2B integration between heterogeneous systems. Today, however, although SOAP still maintains its own space and it is very suitable for use in certain scenarios, most web applications, including those developed by web service providers such as Google, Yahoo, Amazon, Twitter and Facebook, expose a REST API. The term REST stands for Representational State Transfer and was coined in 2000 by Roy Fielding in his Ph.D. dissertation.

In a nutshell, REST provides an excellent architectural style to represent data and possible operations on that data. REST is not a technology, and many tools available on the market might not support REST directly because of the lack of a specific standard. What makes REST the best choice in many scenarios, is its ease of implementation and the fact that REST is not tied to any particular system, language or tool. If you can send an HTTP request to a web server, you can use REST.

Although starting to use REST is very easy, a system that follows all the REST "principles", is not trivial to design and build1. One of the more complex concepts to be grasped by a developer is the concept of “resource”. REST focuses on the concept of remote resource and not on the method or remote object. To define what should be done on a specific resource, REST uses the intrinsic meaning of the verbs of the http protocol.

Here are the five main verbs that are commonly used in RESTful systems:

- GET - Retrieve a resource
- PUT - Create a resource
- POST- Update a resource
- DELETE - Delete a resource
- HEAD- Retrieve the metadata that defines a resource

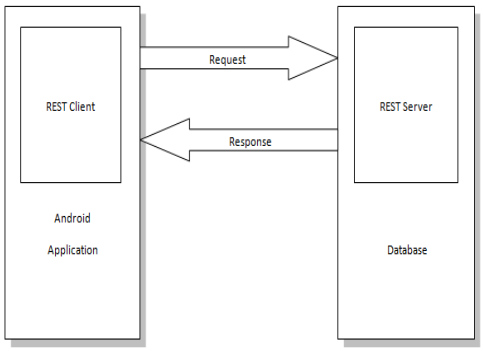


Fig -2: REST Client-Server

As has already been mentioned, REST does not provide standards, but provides a set of designing "principles" to follow. The REST principles can be summarized in five points:

- Identify the "things" (or resources) with an ID
- Connect the "things" between them according to clear and understandable criteria
- Use standard methods and interfaces
- Resources can have multiple representations (JSON, XML, YAML, CSV, etc.)
- The REST service must be stateless

6.2 Application Programming Interface (API)

An API - which stands for Application Programming Interface - allows for publicly exposed methods of an application to be accessed and manipulated outside of the program itself. A common usage of an API is when you wish to obtain data from a application (such as a cake recipe) without having to actually visit the application itself. To allow this action to take place, the application has published an API that specifically allows for foreign applications to make calls to its data and return said data to the user from inside of the external application. On the web, this is often done through the use of RESTful URIs.

Making Our Own RESTful API

The API that we're going to construct here will consist of two classes. One Abstract class that will handle the parsing of the URI and returning the response and one concrete class that will consist of just the endpoints for our API. By separating things like this, we get a reusable Abstract class that can become the basis of any other RESTful API and have isolated all the unique code for the application itself into a single location.

6.3 JSON Parser

What is JSON?

- JSON stands for **J**ava**S**cript **O**bject **N**otation
- JSON is lightweight text-data interchange format
- JSON is language independent
- JSON is "self-describing" and easy to understand

JSON uses JavaScript syntax for describing data objects, but JSON is still language and platform independent. JSON parsers and JSON libraries exists for many different programming languages.

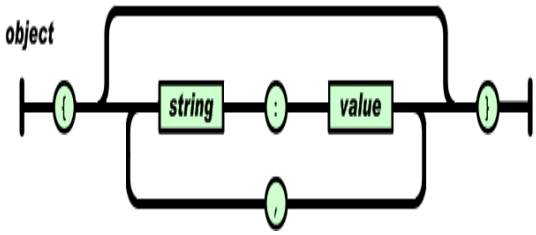
JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

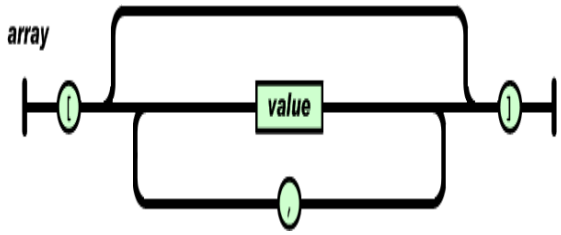
These are universal data structures. Virtually all modern programming languages support them in one form or another. It makes sense that a data format that is interchangeable with programming languages also be based on these structures.

In JSON, they take on these forms:

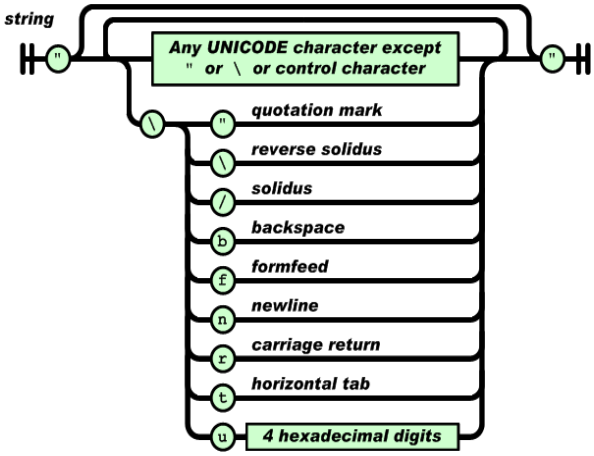
An *object* is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by : (colon) and the name/value pairs are separated by , (comma).



An *array* is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).



A *string* is a sequence of zero or more Unicode characters, wrapped in double quotes, using backslash escapes. A character is represented as a single character string. A string is very much like a C or Java string.



7. RELEVANT AREAS

This application will be useful for the areas of Business and Society. In Business, if a person wants to send any business related documents from outside the office, then this can be useful. In Society, connecting to our friends and relatives this application will be helpful.

8. FUTURE WORK

The video chat feature can be added to the application. The user can send audio and video files through the application. The limit for sending images at a time can be increased. Some new privacy settings for the user can be added to the existing settings.

9. CONCLUSIONS

In this paper, we provided an open specification for a secure social networking service. The aim of the paper is to develop a secure social networking application. This will give a much better insight into the development of secure services and their usability requirements.

ACKNOWLEDGEMENT

We take this golden opportunity to owe our deep sense of gratitude to my project guide Mr. P. V. Kothawale for his instinct help and valuable guidance with a lot of encouragement throughout this project work, right from selection of topic work up to its completion. Our sincere thanks to the Head of the Department of Computer Science & Engineering Mr. A. S. Kamble, who continuously motivated and guided us for completion of this project. We are also thankful to our Project Coordinator, all teaching and non-teaching staff members, for their valuable suggestions and valuable co-operation for completion of this Project. I specially thank to those who helped us directly-indirectly in completion of this work successfully.

REFERENCES

[1] Guess Who's Texting You? Evaluating the Security of Smartphone Messaging Applications (2011)- Sebastian Schrittwieser, Peter Fr¨uhwirt, Peter Kieseberg, Manuel Leithner, Martin Mulazzani, Markus Huber, Edgar Weippl

[2] End-to-End Secure and Privacy Preserving Mobile Chat Application- Raja Naeem Akram and Ryan K. L. Ko

[3] Facebook - WhatsApp Merger – A Revolution in the Social Networks(2014)- Joe Prathap P M , Ajith Jubilson E , Dhanavanthini P Rajkumar , S. Shibu J and W. Vinil Dani

[4] Solutions to Security and Privacy issues in Mobile Social Networking-Aaron Beach, Mike Gartrell, and Richard Han

[5] Social Networking and Security Risks by Brad Dinerman