Integration of random numbers generator for creating the gamma of Vernam key in the system of instant message exchange

Gleb Polozhiy[1], *Anna* Tosunova[1], *Larisa* Cherckesova[1], *Olga* Safaryan[1*], *Elena* Pinevich[1], and *Irina* Reshetnikova[1]

[1] Don State Technical University, Gagarin Square, 1, Rostov-on-Don, 344003, Russia

Abstract. The article is devoted to the overview of encryption protocols using for instant messaging systems – messengers, and to the development of system, that excludes the considered security flaws. Analysis of using security model development is carried out.

1 Introduction

Ubiquitous distribution of instant messaging services on mobile devices and their use of end–to–end (transparent) encryption to protect the user's confidentiality. Such privacy policy even displeases sometimes some special services and governments also. So well–known WhatsApp messaging service is now one of the most popular application for exchanging messages on mobile phones (smartphones) and devices. It uses end–to–end (E2EE) encryption, which prevents and does not allow to both special services and attackers from gaining access to the user's confidential information. However, the use of E2EE technology in WhatsApp has some support drawbacks. The article discusses the vulnerabilities of the E2EE technology and ways to eliminate them. The authors investigated the security issues of the instant message system, found their shortcomings and vulnerabilities taking as an example the E2EE technology, and suggested the ways to eliminate them. The article presents the advantages of encryption using the gamma of Vernam key, which can improve significantly the security and confidentiality of information of users of the WhatsApp messaging service.

Problem statement – the world is constantly changing thanks to advances in science and technology, and now it seems impossible to avoid the presence of new technologies in our daily lives. Since smartphones have become the part of our lives, many messaging services have been launched. WhatsApp, which now has several billion users in more than 180 countries, is free messaging service owned by Facebook Inc. and has become more popular than other messengers. In 2009, Brian Acton and Jan Koum created WhatsApp service specifically to simplify and accelerate the communication between peoples and the distribution of multimedia messages. WhatsApp maintains Internet connection and helps its users stay in touch with friends and relatives from their contact list. In addition to allowing

[1] Corresponding author: safari_2006@mail.ru

users to stay in connection with each other, it also helps them create groups, send images, photo, audios, videos, documents, and etc.

As more and more people use WhatsApp as means of communication, the importance of protecting user's business or personal messages is becoming more urgent. WhatsApp users expect the reasonable level of privacy for all their messages. To fulfill this request and meet this expectation, in 2014 WhatsApp have introduced the technology "end–to–end encryption" (E2EE). That ensures the security of data between the interacting sides, protection from eavesdropping, listening and hacking.

This technology provides confidence to end users, because their data is secure when transmitted. Third sides and persons or even WhatsApp itself cannot access it; thus, only the recipient can decrypt the messages.

Although E2EE guarantees the integrity, security and confidentiality, the main drawback of security of information ensuring the in the messenger is the fact that the backup copies of messages are stored in unencrypted form [1-3]. This significant drawback can be eliminated in the instant messaging system developed by the authors.

It is well known that encryption is the scrambling of text messages, turning them into unreadable code that can only be decrypted by someone who has the secret key. The scrambling process consists of bitwise calculation of the resulting code based on the bits of the source code, and the bits of obtained in the previous clock cycles resulting code also. End–to–end encryption is one of the most commonly used technologies for protecting and sending information over the Internet. The hardware built into mobile phones (smartphones) and computers allows for random locks and keys using, so that E2EE only works on the devices involved in the conversation, i.e. in the process of transmitting and receiving information. At the beginning of 2020, more than 4.5 billion people are the Internet users, and the social networks auditorium has passed the mark of 4 billion. With such tremendous number of users, hackers have an increased incentive to carry out attacks and steals of their personal information.

End–to–end encryption provides an effective way to prevent these attacks. If properly implemented by Yahoo Inc., it could prevent large–scale attacks like the one Yahoo suffered in 2013 and 2016, when nearly 500 million users were hacked and they were damaged, and more than 1 billion accounts were compromised.

Governments and their intelligence agencies, on the other hand, are asking encrypted messaging services like WhatsApp to grant them access to their user's data.

However, architectural feature of WhatsApp messaging service has been reported that could potentially allow some encrypted messages to be read by unintended recipients.

WhatsApp allows the users to store undelivered messages on its servers for up to 30 days before they would be deleted. It is noted that WhatsApp's implementation of the security protocol used in its E2EE allows the generation of secret keys between interacting sides in conversation taking place via WhatsApp.

However, new keys are generated when the user gets new smartphone or reinstalls WhatsApp. Messages for the user that were waiting for delivery while the user was offline are then re–encrypted and automatically re–sent [4].

At the same time, the sender did not have the opportunity to check whether the recipient is the person intended to receive this message. The sender receives notification after the event if the sender has decided to enable the notification in the settings, but not otherwise. This re–encryption and re–sending of previously undelivered messages could potentially allow to third side or person to intercept and read the user's undelivered messages in the situation where, for example, the user's SIM–card was stolen.

When third side or person inserts the stolen SIM–card into another phone, it can theoretically collect any messages that have not yet been delivered to the relevant user.

To describe any formal statement about requirements of confidentiality, availability, or integrity of the system, the term "security model" is used. If to confine by privacy models, there are two related but distinct terms of "security models".

In the more limited this term using the security model defines the specific mechanism for enforcing confidentiality, called access control, which has been carried over from the world of documents and safes into computer security. In the more general using of this term, security models are specifications of system privacy requirements and are not "models" at all in the sense that they define security requirements without describing any specific mechanism for implementing those requirements. These models define constraints on the system interface (usually it is Input / Output ratio) sufficient to ensure that any implementation that satisfies these constraints will enforce confidentiality.

Butler Lampson was first formulated the access control model for privacy policy and then it was reworked by Scott Graham and Peter Denning.

The model structure is the finite automaton structure, where each state is defined by the triple (S; O; M). Here S is set of subjects, O is set of objects (of which S has the subset), and M is access matrix that has one row for each subject, one column for each object, and the cell M [s; o] contains the access rights of the subject s to the object o. These access rights are taken from the finite set of access rights A. The states are changed by requests to modify the access matrix M. The individual machine in the model is the system machine. Despite its simplicity, the model has worked for a long time in the field of computer security, based primarily on the work of Harrison, Ruzzo and Ullman, as well as on the work of Bell and La Padula [5].

2 Mathematical model and solution method

There are many different security models, each of which solves specific problems. However, another approach to solving these problems is to move from general models to application–specific models. The first model to use this approach was the secure military message system model. The limitations expressed in this model are not general security constraints for subjects and objects, but are specific security limitations that the message system must match when processing messages. Since then, the application– specific approach was applied in the number of areas, most notably database security. This is the approach we used when developing the instant messaging system [6].

The system development would not have been possible without use of random number generator. The perfect properties of qualitative pseudorandom number generator of general–purpose are not difficult to determine, but to achieve simultaneously impossible. They include the following qualities: such pseudorandom number generator must continue the sequence of random numbers with uniform distribution law in the interval (0; 1); in one turnover, generator must return one random number; random numbers themselves also must be distributed evenly over the interval in which they are generated.

Repeatability is very important for debugging and for understanding the random process. Therefore, Dennis Ripley and Joe Kirkland show some summaries of Markov random field simulation that shows the sharp change at one point in supposedly convergent iterative process. Since repeating sequence was used, the process could be started just before this point and stopped so that the critical phase could be studied in detail. The time required to generate pseudorandom numbers should be negligible compared to the operations performed on the resulting numbers. When simulating statistical physics on supercomputers, much larger numbers can be used.

When developing real models, it is necessary to strive for the standard, using the available technical and hardware capabilities, which in turn involves compromises.

As the base for the generator in the messenger, special implementation grounded on symbolic pseudo–devices of equipment on which the application server operates was used. It represent interface to the system random number generator that outputs noise from device drivers and other sources to the "chaotic" pool. The generator also stores the required number of noise bits in this pool and forms from it the random or the pseudorandom sequence of numbers [7].

The following code, written in C#, implements the described method and is applicated in the messenger for generation of the gamma of Vernam key:

```
static PyObject *
os_urandom_impl(PyObject *module, Py_ssize_t size)
{
 PyObject *bytes;
 int result;

 if (size < 0)
return PyErr_Format(PyExc_ValueError,
                    "negative argument not allowed");
   bytes = PyBytes_FromStringAndSize(NULL, size);
   if (bytes == NULL)
      return NULL;

   result=_PyOS_URandom(PyBytes_AS_STRING(bytes),
                        PyBytes_GET_SIZE(bytes));
   if (result == -1) {
      Py_DECREF(bytes);
      return NULL;
   }
   return bytes;
}
```

3 Results of research

In contrast to the considered instant messaging system, advantage of the developed application is using not standard encryption protocol, but the scheme that combines Vernam's and RSA encryption algorithms. Another benefit of this cryptosystem is application of its own pseudorandom number generator. The superiority of algorithmic generators is the high speed and compactness of the implementation [8]. Gamma of Vernam key shown in Fig. 1.

When the message (or message history) is transmitted from server to client or from client to server, the data is encrypted with the symmetric Vernam algorithm, which provides absolute the cryptographic durability (resistance) against hacking attempts [9-14].

The asymmetric RSA algorithm encrypts the symmetric key. RSA keys, both client and server, have bit length of at least 3072 bits, which allows for effective protection of Vernam key. In all methods used, random numbers are generated cryptographically strong, which minimizes the likelihood that any attacker will be able to pick up these numbers. At the same time, the speed of message exchange remains at the high level [14-17].

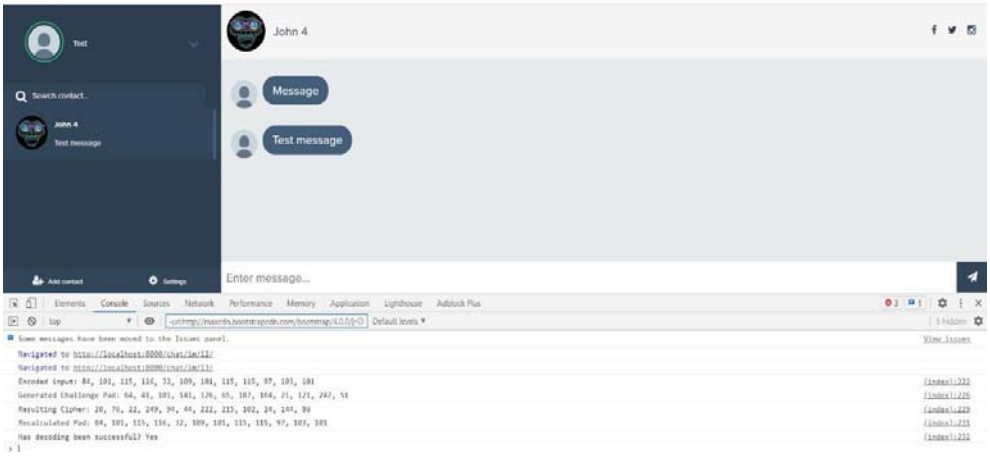

Fig. 1. Gamma of Vernam key.

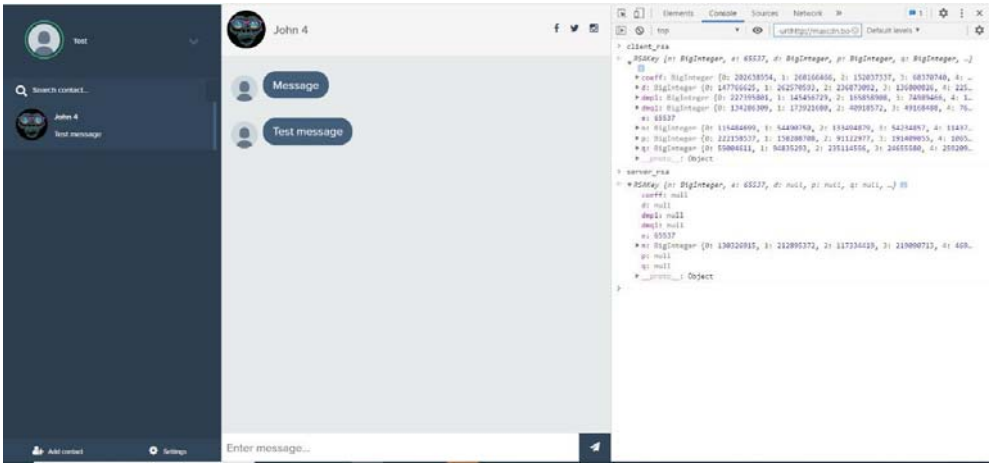

Fig. 2. Instances of the class responsible for encryption and decryption by RSA algorithm (client and server instances).

Let us present these methods, implemented in Python, used on the server side for encryption and decryption of message data. This algorithm performs the encryption by Vernam method, where gamma generation uses cryptographically strong random numbers. Instances of the class responsible for encryption and decryption by RSA algorithm shown in Fig. 2.

```
def vernam_encrypt(self, content):
key = [secure_rand_gen.randrange(150) for i in range(len(content))]
encrypted_content = []
for i in range(len(content)):
encrypted_content.append(ord(content[i])^key[i])

return {
'encrypted_content': encrypted_content,
'key': key
}
```

Algorithm for decrypting of the client message encrypted by the Vernam method:

```
def vernam_decrypt(self, cipher_pad, key):
text = ''
for i in range(len(cipher_pad)):
text += chr(cipher_pad[i]^key[i])
return text
```

4 Discussion and conclusions

Privacy (confidentiality) of communications is the key element of human rights in the digital age, and events affecting it should be reported.

The issue of security is much more acute, because every messenger has some vulnerability in its security system – the risk of personal correspondence leakage or photos is weighty fact. This is why the system with own encryption scheme was developed.

References

1. A.V. Babash, Cryptographical methods of information protection: textbook (KNORUS, Moscow, 2016)
2. P. Razumov, N. Boldyrikhin, L. Cherckesova, et al., E3S Web of Conferences, **224**, 01033 (2020)
3. V.V. Zhilin, I.I. Drozdova, I.A. Sakharov, et al., in Proceedings of IEEE East-West Design and Test Symposium, EWDTS 2019, Institute of Electrical and Electronics Engineers, Inc, 8884375 (2019), DOI: 10.1109/EWDTS.2019.8884375
4. N.A. Gaydamakin, Access differentiation to information in computer systems (Ural State Technical University Publishing House, Yekaterinburg, 2013)
5. N.A Gaydamakin, Theoretical foundations of computer security: textbook (Ural State Technical University Publishing House, Yekaterinburg, 2018)
6. N.A. Gaydamakin, Automated information systems, databases and data banks (Helios ARV, Moscow, 2012)
7. P.N. Devyanin, O.O. Mikhalsky, D.I. Pravikov, A.Yu. Shcherbakov, Theoretical foundations of computer security (Radio and Communications, Moscow, 2010)
8. S.P. Panasenko, Encryption algorithms. Special Handbook (BHV Petersburg Publishing House, St. Petersburg, 2019)
9. A.V. Cheremushkin, Cryptographic protocols. Basic properties and vulnerabilities: textbook (Publishing center "Academy", Moscow, 2015)
10. B. Schneier, Applied cryptography. Protocols, algorithms, source texts in the C language (Triumph Publishing House, Moscow, 2013)
11. D. Beazly, Python. Detailed handbook (Symbol–Plus, St. Petersburg, 2013)
12. WhatsApp Messenger, URL: https://ru.wikipedia.org/wiki/WhatsApp
13. M. Ganguly, WhatsApp Design Feature Means Some Encrypted Messages Could Be Read by Third Party (2017)
14. U. Can, and B. Alatas, Physica A: Statistical Mechanics and its Applications, **535**, 1-38 (2019)
15. V. Singh, S.K. Pandey, Performance Management of Integrated Systems and its Applications in Software Engineering, 11–20 (2020)

16. Europol EC3. (Sep. 2020). Internet Organised Crime Threat Assess-ment(IOCTA) (2020), URL: https://www.europol.europa.eu/iocta-report
17. D. Berman, A. Buczak, J. Chavis, and C. Corbett, A survey of deeplearning methods for cyber security (2019), URL: https://www.researchgate.net/publication/332178214_A_Survey_of_Deep_Learning_Methods_for_Cyber_Security