Bulletin of TUIT: Management and Communication Technologies

Volume 1 *Issue 1* Article 5

1-22-2018

DISTRIBUTED SYSTEM MODEL FOR KEY MANAGEMENT

Mersaid Aripov prof.
National University of Uzbekistan named after Mirzo Ulugbek, mirsaidaripov@mail.ru

Ruhillo Habibovich Alayev
National University of Uzbekistan named after Mirzo Ulugbek, mr.ruhillo@gmail.com

UDC 004.056

DISTRIBUTED SYSTEM MODEL FOR KEY MANAGEMENT

Aripov M.M., Alaev R.H.

Key management plays a crucial role in cryptography, as the basis for secure information exchange, data identification and integrity. There are software and hardware key management tools that support *Crypto APIs* and *Cryptography Next Generation APIs*(*CNG API*), *Public Key Cryptography Standards* (*PKCS*). These tools store cryptographic keys on hard disks, smart cards, tokens, and in other storage media. To use the cryptographic keys stored on these smart cards and tokens, you need to connect them to the appropriate hardware. The cryptographic keys stored on the hard drives of a computer or a laptop are used by the programs of these devices. If it becomes necessary to use a single key in different systems, then you will have to create copies of the key on all these devices. This complicates the process of key management, raises tasks of securely store keys, keys access control.

This paper proposes a distributed system model for key management and a protocol of interaction of the distributed system modules. The proposed model provides the ability to store keys in a smartphone, and access to keys from other devices. The system described in the model consists of 3 modules. The *module 1* has computer version and smartphone version, and serves to send a request for signing, signature verification, hashing. The *module 2*, a smartphone software, provides key pair generation, storing, encrypting and decrypting, archiving keys, export/import keys, keys access control, and destroying keys. The *module 3*, web service, provides communication of the first and second modules.

In addition, the system, which was created based on the current model, provides the ability to use digital signatures in web applications. The *Module 1* operates as a local web service that accepts requests from a web page running in a browser. A special script in a web page sends *http* requests that include cryptographic operations to the specified *localhost* port and accepts responses.

Keywords: key access control, key management, digital signature, hashing, encryption, decryption, QR Code, smartphone, web service.

Introduction

Legal entities and individual entrepreneurs are required electronic digital signature for using most of the interactive services of state organizations in our country. Electronic digital signature is used for identify, verify the integrity of electronic documents, etc. Digital certificate and private key are provided in an encrypted PFX format. Digital signature owners store PFX file in flash media and in computer. Practical implementation of storing digital certificate and private key in tokens or in smart cards are very difficult for the majority of the population. Interactive services are offered in the form of web services, web applications and / or mobile applications. For some interactive services it is convenient to use a computer or laptop, and for others a smartphone. Thus, it becomes necessary to use the key in different devices for different types of applications. Considering that the majority owns smartphones, we offer a method of storing digital certificates and private keys in smartphones, and a model of accessing keys for signing and verification signatures from different computers, laptops and smartphones. In our model, we used a cryptographic module in a smartphone that is responsible for key management, a cryptographic module in a laptop or a computer that generates a signing request and signature verification, as well as an intermediate web service to ensure interactions of these cryptographic modules. The architecture of the system, which provides implementation of the proposed model, is shown in figure 1.

Cryptographic key management encompasses the entire lifecycle of cryptographic keys and keys materials. Basic key management guidance was provided in [1]. A novel Key-Lifecycle Management System was presented in [2]. The presented Key-Lifecycle Management System introduces a pattern-based method to simplify and to automate the deployment task for keys and certificates, also it provides a novel form of strict access control to keys and realizes the first cryptographically sound and secure access-control policy for a key-management interface. Developing a cryptographic key management system for distributed networks was discussed in [3]. A Time-based Group Key Management algorithm for cryptographic cloud storage applications, which uses the proxy re-encryption algorithm to transfer major computing task of the group key management to the cloud server was proposed in [6]. The algorithm described in [6] ensures data confidentiality when storing user data in an unreliable cloud using encryption and proxy.

Scholars and management consultants have identified platform control as a key feature for business success in the ICT industries [10].

The introduction of public key cryptography was a critical advance in IT security. It enables confidential communication between entities in open networks, in particular the Internet, without prior contact [12]. A Lightweight Public Key Infrastructure (LPKI) for the mobile phones was introduced in [7]. It

provides robust distributed authentication services. A method for distributed E-Business application authentication using public key cryptography was introduced in [8]. An enhanced secure authentication protocol for roaming services on elliptic curve cryptography was proposed in [9]. The proposed protocol's formal security is verified using Automated Validation of Internet Security Protocols and Applications tool to certify that the proposed protocol is free from security threats. A distributed e-business authentication scheme based on conic curve was proposed in [11]. A new approach to setting up a secure authentication and authorization procedure in a distributed computing system without proxy certificates was proposed in [13].

However, these works do not include a solution for the above problem.

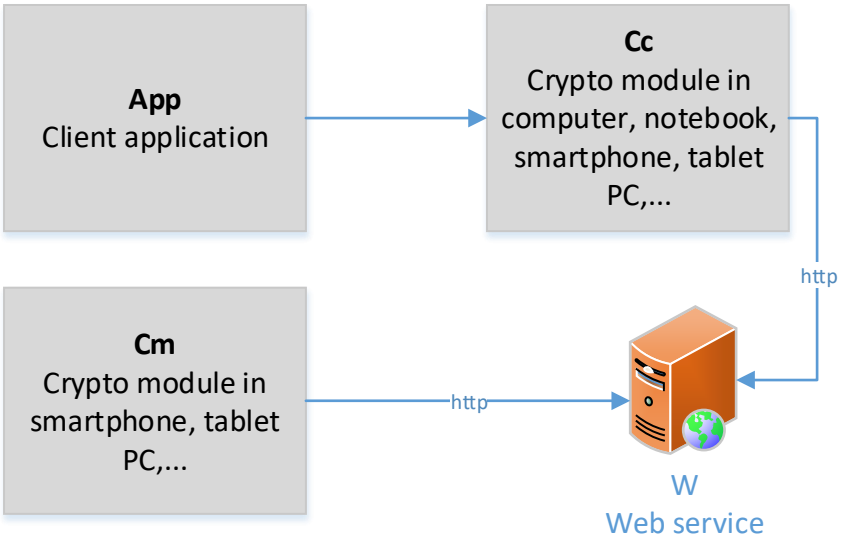

Figure 1. The architecture of the system

The QR code is used in this system for exchanging password between computer and smartphone. In the work [4], QR code was analyzed from the perspective of their significance and uses. In [5] Shettar found QR code as a great technology which helped library to cater most crucial user demand for access to information through mobile phones. The QR code was discussed in [14,15]. The using QR code for ensuring security such as authentication, authorization was discussed in [19-23]. The implementation of the QR code in some areas such as payment system, government service, library was provided in [16-18].

Main part

We introduce the following notation:

E – encryption function of public key algorithm
E' – encryption function of cipher algorithm
D –decryption function of public key algorithm
D' – decryption function of cipher algorithm
$h(x)$ – hash function
M – data to sign
H – hash value
K_{ou} – user public key
K_{pu} – user private key, private key is saved in encrypted form
K_{ow}– web service public key
K_{pw} – web service private key
K_{ms} and K_{cs} – symmetric keys
C_c– crypto module in computer, it has built-in certificate of W service
C_m – crypto module in smartphone for generation, export, import, destroying key pairs and signing hash, it has built-in certificate of W_c service
L-login of user in W, L maybe phone number, e-mail etc.
W - Intermediate web service to provide interaction between cryptographic modules C_c and C_m. User certificates, username and temporary user password are stored here.

Encryption private key of user:

$$K = h(h(PINcode)), K'_{pu} = E'_K(K_{pu}),$$

$$H_k = h(K).$$

$\{K'_{pu}, K_{ou}, H_K\}$ - encrypted private key, public key and hash value of PIN code. Hash value is used for checking the PIN code.

Decryption private key of user:

$K = h(h(PINcode))$, if $H_k = h(K)$ then $K_{pu} = D'_K(K'_{pu})$

The following processes describe interactions between modules step by step.

Registration process

The registration process is done with modules C_m and web service. It includes the following steps:

Step 1: C_m asks from user to select generation new key pairs or import an existing one.
Step 2: C_m asks user to enter *PIN code*.
Step 3: If user selects import an existing key pair go to step 7.

Step 4: C_m generates new key pairs {K_{pu}, K_{ou}}. C_m encrypts private key, saves { K'_{pu}, K_{ou}, H_k } to storage.
Step 5: C_m generates *Certificate Signing Request* (*CSR*) and sends *CSR* to *Registration Authority*(*RA*).
Step 6: C_m installs certificate obtained from *RA*, go to step 9.
Step 7: C_m asks user to enter password, and imports key pairs encrypted in PFX format.
Step 8: C_m encrypts K_{pu} , saves { K'_{pu}, K_{ou}, H_k } and certificate to storage.
Step 9: C_m asks user to enter L, calculates $H = h(L)$, $H' = E_{K_{pu}}(H)$ and sends {L, H', user certificate} to W.
Step 10: W checks if $D_{K_{ou}}(H') = h(L)$ go to step 11, else returns "Incorrect parameters".
Step 11: if L is not used by other user, then go to step 16.
Step 12: W checks "Are these certificates belongs to this user or not". For this purpose W selects all of certificates attached to L, generates T random number, calculates $T' = E_{K_{pw}}(E_{K_{ou}}(T))$, and returns {$T'$, certificates} to C_m.
Step 13 C_m checks if it has any certificate that is in the list of certificates returned by W, then it calculates $T = D_{K_{pu}}(D_{K_{ow}}(T'))$, $T'' = E_{K^1_{pu}}(E_{K_{ow}}(T))$ and sends {T'', selected user certificate} to W, go to step 15, else go to step 14. In this case K^1_{pu} is private key of the selected certificate by C_m.
Step 14: C_m shows to user the message "Login is already taken". Go to step 9.
Step 15: W checks, if $T = D_{K_{pw}}(D_{K^1_{ou}}(T''))$.
Step 16: W registers user, saves L and user certificate in database, returns "ok" as status of registration, go to step 17.
Step 17: Registration is completed successfully. C_m closes the connection with W.

Signature process

All 3 modules take part in the signature process. This process includes the following steps:
Step 1: *App* opens connection with C_c for signing.
Step 2: C_c asks to enter L from user.
Step 3: C_c generates one time password P and shows it to user as *QR code*.
Step 4: C_m scans *QR code* and gets P. C_m checks if user has more than one certificates, C_m asks user to select one of them, else uses that one. C_m asks user to enter PIN code for access to private key.
Step 5: C_m asks user to enter PIN code for access to private key. C_m calculates $K = h(h(PINcode))$, if $H_k = h(K)$ then $K_{pu} = D'_K(K'_{pu})$, else shows "PIN code is incorrect" message and go to step 5.
Step 6: C_m calculates $H'_m = h(P)$, $S' = E_{K_{pu}}(H'_m)$. C_m generates K_{ms}, calculates $P'_m = E'_{K_{ms}}(P)$, $K'_{ms} = E_{K_{pu}}(E_{K_{ow}}(K_{ms}))$.
Step 7: C_m sends L and user selected certificate to W.
Step 8: W generates random number T, calculates $T' = E_{K_{pw}}(E_{K_{ou}}(T))$, and returns T' to C_m.
Step 9: C_m calculates $T = D_{K_{pu}}(D_{K_{ow}}(T'))$, $T'' = E_{K_{pu}}(E_{K_{ow}}(T))$ and C_m sends {L, T'', P'_m, K'_{ms}, S'} to W.
Step 10: W checks if $T = D_{K_{pw}}(D_{K_{ou}}(T''))$ then calculates $K_{ms} = D_{K_{pw}}(D_{K_{ou}}(K'_{ms}))$, $P = D'_{K_{ms}}(P'_m)$, $H = D_{K_{ou}}(S')$ and checks if $H = h(P)$ than W sets temporary password P for L.
Step 11: C_c generates K_{cs}, calculates $P'_c = E'_{K_{cs}}(P)$, $K'_{cs} = E'_{K_{ow}}(K_{cs})$. C_c and sends {L, K'_{cs}, P'_c} to W for authentication.
Step 12: W calculates $K_{cs} = D_{K_{pw}}(K'_{cs})$, $P = D'_{K_{cs}}(P'_c)$ and checks, if L and P is right W returns to C_c the user certificate else returns to C_c "Login or pass is incorrect".
Step 13: If C_c gets message "Login or pass is incorrect" from W, then C_c shows message to user and go to Step 2.
Step 14: C_c returns to *App* the user certificate.
Step 15: *App* sends to C_c the user certificate and M for hashing.
Step 16: C_c calculates $H = h(M)$ and returns it to *App*, the hash algorithm determined by the certificate for signing.
Step 17: *App* sends H to C_c for sign.
Step 18: C_c calculates $H'_c = E'_{K_{cs}}(H)$ and sends H'_c to W.
Step 19: W calculates $H = D'_{K_{cs}}(H'_c)$, $H'_m = E'_{K_{ms}}(H)$.
Step 20: C_m gets H'_m and the selected user certificate from W.
Step 21: C_m calculates $H = D'_{K_{ms}}(H'_m)$. C_m asks user to allow signing H.
Step 22: If user allows signing then C_m calculates $S = E_{K_{pu}}(H)$, $S' = E'_{K_{ms}}(S)$, sends S' and "ok" as status of the signing else sends "sign not allowed" as status of the signing to W.
Step 23: C_c gets status and S from W.
Step 24: C_c returns to *App* the status and S.
Step 25: *App* checks, if status is ok *App* uses S as signature.
Step 26: *App* closes connection with C_c.

Step 27: C_c closes connection with *W*.
Step 28: *W* deletes *P*.

Signature verification

Signature verification process includes the following steps:
Step 1: *App* opens connection with C_c for verify signature.
Step 2: *App* sends to C_c the user certificate and *M* for hashing.
Step 3: C_c calculates $H = h(M)$ and returns *H* to *App*. hash algorithm determined by the user certificate.
Step 4: *App* sends *H*, *S* and the user certificate to C_c.
Step 5: C_c checks, If $H = D_{K_{ou}}(S)$ then C_c returns "signature is valid" as status of verification, else returns "signature is not valid" as status of verification.
Step 6: *App* closes the connection with C_c.

Conclusion

In this paper, we provided a description of the distributed system model for key management and a protocol of interaction of the distributed system modules. We presented the registration process of the user, signature process, signature verification process step by step. *The user registration process* includes key pair generation and/or import existing key pairs, generating and sending *CSR* to obtain digital certificate from RA, and registration user digital certificate in Web service. *Signature process* includes temporary password generation, generating *QR code* of the password, scanning *QR code* by C_m , setting temporary password for session, generating and sending request for sign hash, sign hash, as well as secure data exchange between modules using encryption with cipher algorithm and public key algorithm. *Signature verification process* includes computing hash value and signature verification. As a digital signature algorithm [24], must be chosen one that supports encryption with public key, because key pair of the user digital certificate are used for secure key exchange, for data integrity and identification in this system. RSA and DSA, Steps presented above are enough to provide signature and signature verification using key pair in smartphone. The distributed system, presented in this paper, can be used for providing the ability to store keys in a smartphone, and accessing to keys from other devices.

REFERENCES

[1] NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, Annabelle Lee, Security Technology Group -Computer Security Division -National Institute of Standards and Technology Gaithersburg, MD 20899-8930.

[2] Björkqvist M. et al. (2010) "Design and Implementation of a Key-Lifecycle Management System", International Conference on Financial Cryptography and Data Security, pp 160-174.

[3] Acar T., Belenkiy, M., Ellison, C., & Nguyen, L. (2010). Key management in distributed systems. Microsoft Research (pp. 1–14). Retrieved from http://docplayer.net/11794546-Key-management-in-distributed-systems.html

[4] Shettar, I. M., (2016) Quick Response (QR) Codes in Libraries: Case study on the use of QR codesin the Central Library, NITK. Proc. TIFR-BOSLA National Conference on Future Librarianship-2016, 129-134.

[5] Sangeeta Singh, "QR Code Analysis", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 6, Issue 5, May 2016.

[6] Cui Y., Peng Z., Song W., Li X., Cheng F., Ding L. (2014) A Time-Based Group Key Management Algorithm Based on Proxy Re-encryption for Cloud Storage. Asia-Pacific Web Conference 2014: Web Technologies and Applications pp 117-128. DOI https://doi.org/10.1007/978-3-319-11116-2_11.

[7] Toorani, M., & Shirazi, A. A. B. (2008). LPKI - A lightweight public key infrastructure for the mobile environments. In 2008 11th IEEE Singapore International Conference on Communication Systems, ICCS 2008 (pp. 162–166). https://doi.org/10.1109/ICCS.2008.4737164

[8] Gan, S., Gu, C., & Zhang, X. (2010). A PKI-based authentication approach for E-Business systems. In 2010 2nd International Symposium on Information Engineering and Electronic Commerce, IEEC 2010 (pp. 187–190). https://doi.org/10.1109/IEEC.2010.5533219

[9] Reddy, A. G., Das, A. K., Yoon, E. J., & Yoo, K. Y. (2016). A Secure Anonymous Authentication Protocol for Mobile Services on Elliptic Curve Cryptography. IEEE Access, 4, 4394–4407. https://doi.org/10.1109/ACCESS.2016.2596292

[10] Kenney, M., & Pon, B. (2011). Structuring the smartphone industry: Is the mobile Internet OS platform the key? Journal of Industry, Competition and Trade, 11(3), 239–261. https://doi.org/10.1007/s10842-011-0105-6

[11] Song, X., & Chen, Z. (2008). A distributed electronic authentication scheme in E-Business system. In Proceedings of the International Symposium on Electronic Commerce and Security, ISECS 2008 (pp. 343–346). https://doi.org/10.1109/ISECS.2008.125

[12] Buchmann, J. A., Karatsiolis, E., & Wiesmaier, A. (2013). Introduction to public key infrastructures. Introduction to Public Key Infrastructures (pp. 1–187). Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-642-40657-7

[13] Dubenskaya, J., Kryukov, A., Demichev, A., & Prikhodko, N. (2016). New security infrastructure model for distributed computing systems. In Journal of Physics: Conference Series (Vol. 681). Institute of Physics Publishing. https://doi.org/10.1088/1742-6596/681/1/012051

[14] Liu, Y., Yang, J., & Liu, M. (2008). Recognition of QR Code with mobile phones. In Chinese Control and Decision Conference, 2008, CCDC 2008 (pp. 203–206). https://doi.org/10.1109/CCDC.2008.4597299

[15] Sutheebanjard, P., & Premchaiswadi, W. (2010). QR-code generator. In Proceedings - 2010 8th International Conference on ICT and Knowledge Engineering, ICT and KE 2010 (pp. 89–92). https://doi.org/10.1109/ICTKE.2010.5692920

[16] Lorenzi, D., Vaidya, J., Chun, S., Shafiq, B., & Atluri, V. (2014). Enhancing the government service experience through QR codes on mobile platforms. Government Information Quarterly, 31(1), 6–16. https://doi.org/10.1016/j.giq.2013.05.025

[17] Liébana-Cabanillas, F., Ramos de Luna, I., & Montoro-Ríos, F. J. (2015). User behaviour in QR mobile payment system: the QR Payment Acceptance Model. Technology Analysis and Strategic Management, 27(9), 1031–1049. https://doi.org/10.1080/09537325.2015.1047757

[18] Walsh, A. (2010). QR Codes – using mobile phones to deliver library instruction and help at the point of need. Journal of Information Literacy, 4(1). https://doi.org/10.11645/4.1.1458

[19] Liao, K. C., & Lee, W. H. (2010). A novel user authentication scheme based on QR-code. Journal of Networks, 5(8), 937–941. https://doi.org/10.4304/jnw.5.8.937-941

[20] Kieseberg, P., Leithner, M., Mulazzani, M., Munroe, L., Schrittwieser, S., Sinha, M., & Weippl, E. (2010). QR code security. In MoMM2010 - 8th International Conference on Advances in Mobile Computing and Multimedia (pp. 430–435). https://doi.org/10.1145/1971519.1971593

[21] Nseir, S., Hirzallah, N., & Aqel, M. (2013). A secure mobile payment system using QR code. In 2013 5th International Conference on Computer Science and Information Technology, CSIT 2013 - Proceedings (pp. 111–114). https://doi.org/10.1109/CSIT.2013.6588767

[22] Finžgar, L., & Trebar, M. (2011). Use of NFC and QR code identification in an electronic ticket system for public transport. In 2011 International Conference on Software, Telecommunications and Computer Networks, SoftCOM 2011 (pp. 81–86).

[23] Lu, J., Yang, Z., Yuan, W., Li, L., Chang, C. C., & Li, L. (2017). Multiple Schemes for Mobile Payment Authentication Using QR Code and Visual Cryptography. Mobile Information Systems, 2017. https://doi.org/10.1155/2017/4356038

[24] Stallings, W. (2013). Digital Signature Algorithms. Cryptologia, 37(4), 311–327. https://doi.org/10.1080/01611194.2013.797044