Encryption: Strengths and Weaknesses of Public-key Cryptography

Matt Blumenthal
Department of Computing Sciences
Villanova University, Villanova, PA 19085
CSC 3990 – Computing Research Topics
matthew.blumenthal@villanova.edu

Abstract

Public key cryptography has become an important means of ensuring confidentiality, notably through its use of key distribution, where users seeking private communication exchange encryption keys. It also features digital signatures which allow users to sign keys to verify their identities. This research presents the innovations in the field of public-key cryptography while also analyzing their shortcomings. We present methods of improving upon the weaknesses that include techniques involving double encryption and mutual authentication. These contributions introduce new levels of security to the subject with ideas to combat man in the middle attacks and other hacker scenarios. Public-key encryption with digital signatures offers both security and data integrity against most attackers.

1. Introduction

Public key cryptography has become an important means of ensuring confidentiality, notably through its use of key distribution. Key distribution is an approach where users seeking private communication exchange encryption keys, while digital signatures allow users to sign keys to verify their identities. This research explores the strengths and weaknesses of public key cryptography, examining potential flaws and methods of correcting them.

2. Secret-key Cryptography

Secret-key cryptography, also known as symmetric-key cryptography, employs identical private keys for users, while they also hold unique public keys. "Symmetric-key" refers to the identical private keys shared by users. Users employ public keys for the encryption of data, while the private keys serve a necessary purpose in the decryption of data. People wishing to engage in a secure exchange of information will swap public keys and use some method to ensure the existence of identical private keys. In theory,

private keys would be brought into the transaction through either the duplication of an existing key or the creation of two identical keys. In modern practice, users utilize key generators to create both keys, but the private keys must still be distributed in a confidential mode.

2.1 Strengths

The private keys used in symmetric-key cryptography are robustly resistant to brute force attacks. While only the one-time pad, which combines plaintext with a random key, holds secure in the face of any attacker regardless of time and computing power, symmetric-key algorithms are generally more difficult to crack than their public-key counterparts. Additionally, secret-key algorithms require less computing power to be created than equivalent private keys in public-key cryptography. [3]

2.2 Weaknesses

The biggest obstacle in successfully deploying a symmetric-key algorithm is the necessity for a proper exchange of private keys. This transaction must be completed in a secure manner. In the past, this would often have to be done through some type of face-to-face meeting, which proves quite impractical in many circumstances when taking distance and time into account. If one assumes that security is a risk to begin with due to the desire for a secret exchange of data in the first place, the exchange of keys becomes further complicated [5].

Another problem concerns the compromise of a private key. [5] In symmetric-key cryptography, every participant has an identical private key. As the number of participants in a transaction increases, both the risk of compromise and the consequences of such a compromise increase dramatically. Each additional user adds another potential point of weakness that an attacker could take advantage of. If such an attacker succeeds in gaining control of just one of the private keys in this world, every user, whether there are hundreds of users or only a few, is completely compromised.

3. Public-key Encryption

3.1 Summary

Küchlin introduces the foundations of public-key encryption and presents RSA as an early method of transmitting secret messages over insecure channels [5]. The author recognizes that unauthorized users can attempt to intercept messages, and devises this public-key method for ensuring that such users will not be able to interpret the contents of the message [5]. The author's public-key method consists of separate encryption and decryption keys, with users only being able to decrypt an encrypted message if they have the appropriate decryption key [5]. Users will exchange public keys; this transaction does not need to be done in a secure manner because the release of public keys does not threaten the security of any private information. After this swap, someone who wishes to

send private information to another user will encrypt the data with the intended recipient's public key and then pass along the encrypted message. The recipient, who will keep his or her private key secure under any circumstance, can use the private key to decrypt the encoded message. Küchlin introduces separate algorithms for generating encryption and decryption keys as well as an algorithm for combinations of encryption and decryption keys [5].

3.2 Strengths

The asymmetric nature of public-key cryptography allows it a sizable advantage over symmetric-key algorithms. The unique private and public keys provided to each user allow them to conduct secure exchanges of information without first needing to devise some way to secretly swap keys. This glaring weakness of secret-key cryptography becomes a crucial strength of public-key encryption [5].

3.3 Weaknesses

Keys in public-key cryptography, due to their unique nature, are more computationally costly than their counterparts in secret-key cryptography. Asymmetric keys must be many times longer than keys in secret-cryptography in order to boast equivalent security [5]. Keys in asymmetric cryptography are also more vulnerable to brute force attacks than in secret-key cryptography. There exist algorithms for public-key cryptography that allow attackers to crack private keys faster than a brute force method would require. The widely used and pioneering RSA algorithm has such an algorithm that leaves it susceptible to attacks in less than brute force time [3]. While generating longer keys in other algorithms will usually prevent a brute force attack from succeeding in any meaningful length of time, these computations become more computationally intensive. These longer keys can still vary in effectiveness depending on the computing power available to an attacker.

Public-key cryptography also has vulnerabilities to attacks such as the man in the middle attack [3]. In this situation, a malicious third party intercepts a public key on its way to one of the parties involved. The third party can then instead pass along his or her own public key with a message claiming to be from the original sender. An attacker can use this process at every step of an exchange in order to successfully impersonate each member of the conversation without any other parties having knowledge of this deception. [3]

3.4 Proposals

Herzberg, et al. realize the problems presented by the necessity of keeping a private key used in public-key cryptography secret for a long time, and present proactive public key systems that requires more successful hacker attacks in a shorter period of time in order to obtain the private key. Their method builds on threshold cryptography, which they introduce as a method where many users receive parts of the key in order to

protect against any single failure point, but they understand that attackers will still have plenty of time to break the system in certain cases. The paper presents a proactive system that updates the shares periodically in such a way that they are renewed but their shared secret does not change. This robust system meaningfully protects the key, but it does so by transferring the emphasis on security to external hosts. It assumes the security on the servers in which the shares are stored is sufficient, which in a large scale operation is usually sufficient. For more typical users, however, having robust security in several places is a more difficult requirement to meet [4].

4. Digital Signatures

Digital signatures act as a verifiable seal or signature to confirm the authenticity of the sender and the integrity of the message. Users who wish to verify their identity when sending a protected message can encrypt the information with their private key. The recipient can then decrypt the message with the sender's public key in order to confirm the sender's identity and the integrity of the message [1].

4.1 Strengths

Digitally signing a message protects the message in that even if someone intercepted the message before it reached the intended destination and modified it, the digital seal would be broken and the recipient would have this realization after attempting to verify the seal with the sender's public key. The digital signature proves the identity of the sender because only the true sender would have been able to sign the message with his or her private key, except in the event of a compromise. [1]

4.2 Weaknesses

The most serious problems with digital signatures stem from their lack of inherent time stamping. If an unauthorized entity gains access to someone's private key, he or she could send an array of fake messages and sign them with someone else's private key, successfully posing as that other person. The individual whose private key was stolen is unable to repudiate the false messages without having to start over and generate a new private key. To complicate matters, it is impossible to intrinsically separate the fake messages from real ones sent before the compromise because of the absence of time stamping in digital signatures. [1]

4.3 Proposals

Booth examines a proposed encryption method using double encryption, in which a user who wishes to send an encrypted message to another user will encrypt the message with his or her own private key and with the user's public key. The receiver will then decrypt the message using his or her own private key and the sender's public key. This article realizes the problem posed by compromised keys, as either user's private key

falling into the wrong hands can lead to disaster. It proposes a central authentication server, which will receive encrypted messages directly from users, verify that the message has been signed with the sender's current private key, and then attach the receiver's public key and forward the message to its intended destination. This authentication method supercedes the need for an authentication server with a network clock or an archive of compromised keys because as long as it receives notice of all compromises, previous messages will have already been validated [1].

5. Certificate Authorities

Certificate authorities act as trusted third parties that verify the identity of the sender of an encrypted message and issue digital certificates as evidence of authorization. These digital certificates contain the public key of the sender, which is then passed along to the intended recipient.

5.1 Strengths

The issuing of digital certificates allows certificate authorities to play an important role in preventing man in the middle attacks. [3] Certificate authorities have been implemented in the online environment in protocols such as Secure Socket Layer (SSL) and its successor Transport Layer Security (TLS), which have been improved security in web browsing, email, and other methods of data exchange.

5.2 Weaknesses

While certificate authorities aid greatly in the realm of security, they also serve as another potential point of attack. Certificate authorities can be vulnerable to attackers in certain scenarios, and when compromised, can be forced to issue false certificates. Man in the middle attackers who succeed in compromising a certificate authority can use these false certificates to discreetly impersonate each member of the information exchange. Users who are deceived will be even less likely to suspect anything than in a normal man in the middle attack, given the assumed security of the certificate authority. [3]

5.3 Proposals

Halevi and Krawczyk explore an asymmetric case where an authentication server holds private keys while users use only passwords as authentication. They define a password-based authentication protocol where the server uses its own public key to authenticate the user's password, rather than using the user's password as a key to the cryptographic function, which would be a vulnerable and ineffective solution. They also look at a similar approach that uses mutual authentication in which the server possesses both public and private keys, and the user and server authenticate each other. The authors prove that while such systems could still be susceptible to a typical man in the middle approach, where a hacker intercepts messages and replaces them with his own in order to gain an advantage, or other online hacker scenarios, hackers would gain no

added advantage from using an offline password guessing approach that uses computational power to find meaningful patterns, which can be a more effective approach than online attacks against some other security methods [3].

Boyarsky analyzes Halevi and Krawczyk's paper and discovers that their proposal of an asymmetric user-server relationship using server keys and a public password can become insecure when multiple users are introduced to the user-server scenario, with impersonation becoming a real possibility [2]. Boyarsky examines the break in the previous solution, identifying the break as an attacker possibly simulating successful user logins and using this ability to learn the secret password [2]. Boyarsky proposes using the server's public key for signing a user's session key. This system would employ one-time keys, with both the server and user choosing fresh private and public keys for the exchange, which is performed on the user's password [2]. This approach expands on Halevi and Krawczyk's method, satisfying the weakness through the additional key exchange.

6. Future Work

Future work could be done on Herzberg et al.'s idea concerning proactive public key systems. Research in this area could be performed to assess the idea's practicality. Their research holds numerous optimistic positions that would correct the problems surrounding a third party relationship. Positive results in this analysis could lead to a possible implementation of their research in the future.

7. Conclusions

Public-key cryptography has evolved from early models such as Küchlin's to more sophisticated systems that have provided the privacy and data security that we need in the modern world. Secret-key cryptography lags behind asymmetric cryptography. Combinations of the two can be implemented for improved security but secret-key cryptography by itself proves insecure against man in the middle attacks. Asymmetric cryptography has been the foundation for secure data exchange over networks and while it still has its shortcomings, new ideas still come forth as the field continues to evolve.

References

[1] Kellogg S. Booth, "Authentication of signatures using public key encryption," Communications of the ACM, November 1981, pp. 772-774

[2] Maurizio Kliban Boyarsky, "Public-key cryptography and password protocols: the multi-user case," Proceedings of the 6th ACM conference on Computer and communications security CCS '99, November 1999, pp. 63-72.

[3] Shai Halevi and Hugo Krawczyk, "Public-key cryptography and password protocols," ACM Transactions on Information and System Security, August 1999, pp. 230-268.

[4] Amir Herzberg, Markus Jakobsson, Stanisław Jarecki, Hugo Krawczyk, Moti Yung, "Proactive public keys and signature systems," Conference on Computer and Communications Security, 1997, pp. 100-110

[5] W. Küchlin, "Public key encryption," ACM SIGSAM Bulletin, August 1987, pp. 69-73.