ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

A survey on digital signatures

Venkateswara Rao Pallipamu[1], Thammi Reddy K[2], Suresh Varma P[3]

Associate professor, Department of Computer Science, Adikavi Nannaya University, Rajahmundry, India [1]

Professor, Department of Computer Science and Engineering, GITAM University, Visakhapatnam, India [2]

Professor, Department of Computer Science, Adikavi Nannaya University, Rajahmundry, India [3]

Abstract: The importance of authentication is increasing due to increase of online transactions over the internet. There is a need to establish a framework for the authentication of computer-based information. A Digital Signature is one of the authentication mechanisms. A valid digital signature gives a recipient reason to believe that the message was created by a known sender, and that it was not altered in transit. Digital signatures are commonly used for software distribution, financial transactions, and in other cases where it is important to detect forgery or tampering. Various asymmetric cryptosystems create and verify digital signatures using different algorithms and procedures. This paper performs security analysis of Digital Signature schemes; RSA, DSA and EIGamal. A cryptographic tool is used for conducting experiments. Experiments results are given to analyses the effectiveness of each algorithm.

Keywords: Cryptography, Digital signature, Message digest, Authentication

I. INTRODUCTION

The Digital signatures and hand-written signatures both rely on the fact that it is very hard to find two people with the same signature. People use public-key cryptography to compute digital signatures by associating something unique with each person. When public-key cryptography is used to encrypt a message, the sender encrypts the message with the public key of the intended recipient. When public-key cryptography is used to calculate a digital signature, the sender encrypts the "digital fingerprint" of the document with his or her own private key. Anyone with access to the public key of the signer may verify the signature. Suppose Alice wants to send a signed document or message to Bob. The first step is generally to apply a hash function to the message, creating what is called a message digest or digital fingerprint. The message digest is usually shorter than the original message. In fact, a hash function takes a message of arbitrary length and shrinks it down to a fixed length. To create a digital signature, one usually signs (encrypts) the message digest. This saves a considerable amount of time, though it does create a slight insecurity (addressed below). Alice sends Bob the encrypted message digest and the message, which she may or may not encrypt. In order for Bob to authenticate the signature he must apply the same hash function as Alice to the message she sent him. He also decrypts the encrypted message digest using Alice's public key and now compares the two. If the two are the same he has successfully authenticated the signature. If the two do not match there are a few possible explanations. Either someone is trying to impersonate Alice, the message itself has been altered since Alice signed it or an error occurred during transmission. For signature verification to be meaningful, the verifier must have confidence that the public key does actually belong to the sender (otherwise an impostor could claim to be the sender, presenting her own public key in place of the real one). A certificate, issued by a Certification Authority, is an assertion of the validity of the binding between the certificate's subject and her public key such that other users can be confident that the public key does indeed correspond to the subject who claims it as her own.

II. DIGITAL SIGNATURE SCHEMES

A. The RSA Signature scheme

The RSA signature scheme [48] is a deterministic digital signature scheme which provides message recovery.
For the RSA public-key encryption scheme the message space M and the cipher text space C are Zn= {0, 1, 2, ... , n-1}.

Key-Generation
In RSA public key cryptosystems each user
1. Generates two large distinct random primes p and q,
2. Computes n = pq and $\Phi = (p-1)(q-1)$
3. Selects a random integer e, $1 < e < \Phi$, such that $\gcd(e,\Phi) = 1$
4. Computes the unique integer d, $1 < d < \Phi$, such that $ed \equiv 1 \bmod \Phi$
Now the public key of Alice is (n,e) and the private key is d.

Signature Generation
To sign a message m Є M, Alice
1. identifies m with a number ~ m in Zn through a map R : $M \rightarrow Zn$.
2. computes the signature $s = \sim m^d \bmod n$.

Verification
To verify the signature of Alice, Bob
1. chooses the public key (e,n) of Alice.
2. computes $\sim m = s^e \bmod n$.
3. Verifies that ~ m Є M' where M' denotes the set of

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

images of R. If it does not hold rejects the signature else recovers the message as $m = R^{-1}(\sim m)$.

B. The DSA Signature scheme

The DSA makes use of the following parameters:

1. p = a prime modulus, where 2L-1 < p < 2L for 512 £ L £ 1024 and L a multiple of 64

2. q = a prime divisor of p - 1, where 2159 < q < 2160

3. g = h(p-1)/q mod p, where h is any integer with 1 < h < p - 1 such that h(p-1)/q mod p > 1

(g has order q mod p)

4. x = a randomly or pseudo randomly generated integer with 0 < x < q

5. y = gx mod p

6. k = a randomly or pseudo randomly generated integer with 0 < k < q

The integers p, q, and g can be public and can be common to a group of users. A user's private and public keys are x and y, respectively. They are normally fixed for a period of time. Parameters x and k are used for signature generation only, and must be kept secret. Parameter k must be regenerated for each signature. Parameters p and q shall be generated as specified in Appendix 2, or using other FIPS approved security methods. Parameters x and k shall be generated as specified in Appendix 3, or using other FIPS approved security methods.

Signature Generation
The signature of a message M is the pair of numbers r and s computed according to the equations below:
r = (gk mod p) mod q and
s = (k-1(SHA-1(M) + xr)) mod q.
In the above, k-1 is the multiplicative inverse of k, mod q; i.e., (k-1 k) mod q = 1 and 0 < k-1 < q. The value of SHA-1(M) is a 160-bit string output by the Secure Hash Algorithm specified in FIPS 180-1. For use in computing s, this string must be converted to an integer. As an option, one may wish to check if r = 0 or s = 0. If either r = 0 or s = 0, a new value of k should be generated and the signature should be recalculated (it is extremely unlikely that r = 0 or s = 0 if signatures are generated properly). The signature is transmitted along with the message to the verifier.

Verification
Prior to verifying the signature in a signed message, p, q and g plus the sender's public key and identity are made available to the verifier in an authenticated manner.
Let M¢, r¢, and s¢ be the received versions of M, r, and s, respectively, and let y be the public key of the signatory. To verify the signature, the verifier first checks to see that 0 < r¢ < q and 0 < s¢ < q; if either condition is violated the signature shall be rejected. If these two conditions are satisfied, the verifier computes
w = (s¢)-1 mod q
u1 = ((SHA-1(M¢))w) mod q
u2 = ((r¢)w) mod q
v = (((g)u1 (y)u2) mod p) mod q.
If v = r¢, then the signature is verified and the verifier can have high confidence that the received message was sent by the party holding the secret key x corresponding to y.
For a proof that v = r¢
when M¢ = M, r¢ = r, and s¢ = s.
If v does not equal r¢, then the message may have been modified, the message may have been incorrectly signed by the signatory, or the message may have been signed by an impostor. The message should be considered invalid.

C. The ELGamal Signature scheme

The ElGamal signature scheme [18] is a signature scheme with appendix. It requires a hash function $h : \{0,1\}^* \rightarrow Z_p$, where p is large prime. In this scheme, system parameters are : p - a large prime number
g - a generator of Z^*_p
h- a secure collision free one-way hash function
xA- a random integer in (1,p-1), it works as secret key of Alice.
yA- where, $yA= g^{xA}$ mod p, works as the public key of Alice.

Signature Generation
To sign a binary message m of arbitrary length, the user Alice selects a random integer
k Є (1,p-1) such that gcd(k,p-1) = 1.
Alice computes $r = g^k$ mod p and k^{-1} mod p-1.
He further computes $s = k^{-1}[h(m) - x_{Ar}]$ mod p - 1. "
Alice's signature for the message m is (r,s,m).

Verification
To verify the signature (r,s,m) Bob
Checks that 1< r < (p-1) to accept a valid commitment r
Computes $v1= yA^r r^s$ mod p.
Computes h(m) and $v2= g^{h(m)}$ mod p.
The signature is valid if and only if v1= v2.

III. SECURITY OF DIGITAL SIGNATURE SCHEMES

The concepts of negligible and polynomial functions appear when the security of cryptographic schemes is studied.

Definition 3.1 (Negligible function). A function $f : N \rightarrow R^+$ is negligible in k if, for every c > 0 there exists $k_0 \in N$ such that $f(k) < 1/k^c$, for all positive integer $k \geq k_0$. Otherwise, the function f is non-negligible in k.

Definition 3.2 (Polynomial function). A function $g : N \rightarrow R^+$ is polynomial in k if, for every $k_0 \in N$ there exists a value c > 0 such that $f(k) < k^c$, for all positive integer $k \geq k_0$.
Roughly speaking, the cryptographic schemes will be defined according to a security parameter k. We will consider such schemes secure if any adversary trying to

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

attack them in polynomial time (in k) has a success probability which is a negligible function of k. On the other hand, we say that an event has overwhelming probability with respect to k if the probability of its complementary is negligible in k. To guarantee that a signature scheme provides authenticity and non-repudiation to digital communications, one must prove in some way that the scheme is secure. That is, only the owner of a secret key should be able to compute valid signatures with respect to the matching public key. This intuitive idea was formalized in [56], by considering an adversary who tries to break a signature scheme. The security of the scheme is defined according to the capabilities of this adversary, and its final goal. With respect to its final goal, one can consider different levels of success for an adversary: to compute the secret key of a user; to find an efficient algorithm which emulates the signing algorithm of a user; to find a valid signature for a fixed message; to find a valid signature for some message. With respect to the capabilities of the adversary, we list here some different situations: the adversary knows only the public key of the user; the adversary has access to valid signatures of a list of messages that it has not chosen; the adversary has access to valid signatures for messages that it can adaptively choose. Signature schemes can achieve different levels of security. For example, a signature scheme can be proved to resist attacks whose goal is to compute the secret key, but on the other hand there can exist an attack against this scheme which finds a valid signature for some message. Obviously, the maximum level of security for such a scheme consists of resisting attacks from an adversary with the most powerful capabilities (adaptive chosen message attack) but with the less ambitious goal (existential forgery, for some message). Nowadays, a signature scheme is considered secure (or un-forgeable) only if it achieves this level of security.

Definition 3.3 (Un-forgeability). A signature scheme, with security parameter k, is unforgeable if no adversary which is given the public key and the signatures θ1,...,θs of s messages m1,...,ms adaptively chosen by itself, can produce in polynomial time (in k) and with non-negligible probability (in k) a valid signature θ of some message m, such that (m, θ) ?= (mi, θi), for all i = 1,...,s. Figure 1. gives an idea of what a successful forgery against a signature scheme is:

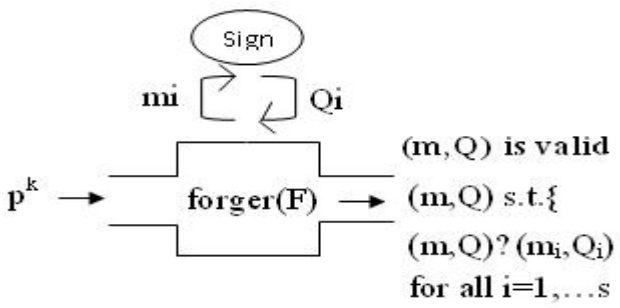

Fig 1: A successful forger against a signature scheme

The usual argument is to reduce the problem of forging a signature to a related computational problem. In other words, assuming the existence of a successful attack against the unforgeability of a scheme, one could solve the related problem. If this problem is assumed to be hard to solve, the reduction implies a contradiction, and one can conclude that the scheme is therefore unforgeable. Proving the unforgeability of a signature scheme in an absolute way, without such a reduction, seems to be a really hard problem. However, constructing such a proof by reduction is not easy at all. The idea is to use the hypothetical existence of a successful adversary to solve an instance of the related computational problem. Roughly speaking, we receive an instance of the problem, and we try to set up the public parameters of the signature scheme in an ingenious way that allows:

- to provide the adversary with valid signatures for messages that it adaptively chooses, when we execute (without knowing the secret key!) the hypothetical successful attack against the signature scheme; and then
- to extract the solution of the problem from the signature forged by the adversary. There exist very few proposals of signature schemes which can be proved secure in this formal (but restrictive) way. However, either the resulting schemes are not very efficient, or the security is based on stronger assumptions, like the Strong RSA Assumption, as it happens in the schemes proposed, or the q-Strong Diffie-Hellman assumption.

IV. COMPARISON OF THE KEY STRENGTHS OF RSA/ DSA AND ECDSA

Use either SI The advantage of elliptic curve over their public key systems such as RSA, DSA etc is the key strength. The following table [3] summarizes the key strength of ECDSA based systems in comparison to other public key schemes.

TABLE I
COMPARISON OF THE KEY STRENGTHS OF RSA/DSA AND ECDSA

RSA/DSA Key length	ECC Key Length for Equivalent Security
1024	160
2048	224
3072	256
7680	384
15360	512

From the table it is very clear that elliptic curves offer a comparable amount of security offered by the other popular public key for a much smaller key strength. This property of ECDSA has made the scheme quite popular of late. As with elliptic curve cryptography in general, the bit size of the public key believed to be needed for ECDSA is about twice the size of the security level, in bits. By comparison, at a security level of 80 bits, meaning an attacker requires the equivalent of about 2^{80}signature generations to find the private key, the size of a DSA

ISSN (Print) : 2319-5940
ISSN (Online) : 2278-1021

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 3, Issue 6, June 2014

public key is at least 1024 bits, whereas the size of an ECDSA public key would be 160 bits. On the other hand, the signature size is the same for both DSA and ECDSA: 4*t* bits, where *t* is the security level measured in bits, that is, about 320 bits for a security level of 80 bits.

V. CONCLUSION

In this paper we perused the concept of Cryptography including the various digital signature schemes of system based on the kind of key and a few algorithms such as RSA, DSA and ECDSA. We studied in detail the mathematical foundations of various algorithms for generation of keys and verification of digital signatures and also their security strengths were analyzed.

REFERENCES

[1] Julio Lopez and Ricardo Dahab, "An overview of elliptic curve cryptography", May 2000.
[2] El-Kassar, A.N., M.Rizk, N.Mirza and Y.Awad,2001.ElGamal public-key cryptosystem in the domain of Gaussian integers.Intl.J.Appl.Math.,7:405-412.
[3] Haraty, R., O. Otrok and A.N.El-Kassar,2004.A comparative study of ElGamal based cryptographic algorithm. Proc. Sixth Intl. Conf. Enterprise Information Systems (ICEIS 2004),3:79-84.
[4] Rivest,R.L.,Shamir,A.,and Adleman,L.," A method of obtaining Digital Signatures and Public key cryptosystems",Comm.ACM,21,1978.
[5] Nathanson, Melvyn, B.,Elementary Methods in Number Theory, Springer, 2000.
[6] William Stallings, Cryptography and Network Security: Principles and practice.Tsinghua press,2002,253-299.
[7] Rabin, M.O., Probabilistic algorithms. In Algorithms and Complexity, J. F.Traub,Ed., Academic Press,New York,1976,pp.21-40.
[8] A. Jurisic, A. Menezes, "Elliptic Curves and Cryptography", 2003,http://www.certicom.com/whitepapers.