GLOBAL JOURNAL OF ENGINEERING SCIENCE AND RESEARCHES

ELLIPTIC CURVE ALGORITHM USAGE IN WIRELESS SECURITY WITH PUBLIC KEY CRYPTOGRAPHY

Mr.K.Ravikumar[*1], Dr.A.Senthilkumar[2]
Research ScholaR, R&D Centre, Bharathiar University, Coimbatore-46[*1]
Assistant Professor, Department of Computer Science, Tamil University, Thanjavur-10[2]

ABSTRACT

Public key techniques revolutionized cryptography. Over the last twenty years however, new techniques have been developed which offer both better performance and higher security than these first generation public key techniques. The best-assured group of new public key techniques is built on the arithmetic of elliptic curves. This paper will outline a case for moving to elliptic curves as a foundation for future Internet security. This case will be based on both the relative security offered by elliptic curves and first generation public key systems and the relative performance of these algorithms. While at current security levels elliptic curves offer significant benefits over existing public key algorithms, as one scales security upwards over time to meet the evolving threat posed by eavesdroppers and hackers with access to greater computing resources, elliptic curves begin to offer dramatic savings over the old, first generation techniques. The two noteworthy first generation public key algorithms used to secure the Internet today are known as RSA and Diffie-Hellman (DH). The security of the first is based on the difficulty of factoring the product of two large primes. The second is related to a problem known as the discrete logarithm problem for finite groups. Both are based on the use of elementary number theory. Interestingly, the security of the two schemes, though formulated differently, is closely related. Wireless sensor networks (WSNs) in healthcare are one of the most important and rapidly growing areas. One of the most critical security concerns is patients' privacy. Since patients are monitored all the time, authentication of who can access the information, and what information one is authorized to access are indispensable to maintain privacy. In health-care environments, authentication and access control face a big challenge due to dynamic network topology, mobility, and stringent resource constraints. In this paper, we propose a secure, scalable, and energy-efficient security scheme called Mutual Authentication and Access Control scheme based on Elliptic Curve Cryptography (MAACE). MAACE provides mutual authentication where a healthcare professional can authenticate to an accessed node (a PDA or medical sensor) and vice versa. This is to ensure that medical data is not exposed to an unauthorized person. On the other hand, it ensures that medical data sent to healthcare professionals did not originate from a malicious node. By applying elliptic curve cryptography (ECC), MAACE provides a public key approach which is more scalable and requires less memory compared to symmetric key-based schemes. Furthermore, it is practically feasible to implement it on sensor platforms. Security analysis and performance evaluation results are presented and compared to existing schemes to show advantages of the proposed scheme.

Keywords: Public Key, Cryptography, ECC, RSA, Geo-Graphic Information System.

I. INRODUCTION

The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman. The US National Institute for Standards and Technology has recommended that these 1024-bit systems are sufficient for use until 2010. After that, NIST recommends that they be upgraded to something providing more security. One option is to simply increase the public key parameter size to a level appropriate for another decade of use. Another option is to take advantage of the past 30 years of public key research and analysis and move from first generation public key algorithms and on to elliptic curves.One-way judgments are made about, the correct, key size for a public, key system is to look at the strength of the conventional (symmetric) encryption algorithms that the public key algorithm will be used to key or authenticate. Examples of these conventional algorithms are the Data Encryption Standard (DES) created in 1975 and the Advanced Encryption Standard (AES) now a new standard. The length of a key, in bits, for a conventional encryption algorithm is a common measure of security. To attack an algorithm with a k-bit key it will generally require roughly 2k-1 operations. Hence, to secure a public key system one would generally want to use parameters that require at least 2k-1 operations to attack. The following table gives the key sizes recommended by the National Institute of Standards and Technology to protect keys used in conventional encryption algorithms like the (DES) and (AES) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security.

II. ELLIPTIC CURVE SECURITY AND EFFICIENCY

To use RSA or Diffie-Hellman to protect 128-bit AES keys one should use 3072-bit parameters: three times the size in use throughout the Internet. The equivalent key size for elliptic curves is only 256 bits. One can see that as symmetric key sizes increase the required key sizes for RSA and Diffie-Hellman increase at a much faster rate than the required key sizes for elliptic curve cryptosystems. Hence, elliptic curve systems offer more security per bit increase in key size than either RSA or Diffie-Hellman public key systems.

Security is the attractive feature of elliptic curve cryptography. Elliptic curve cryptosystems also are more computationally efficient than the first generation public key systems, RSA and Diffie-Hellman. Although elliptic curve arithmetic is slightly more complex per bit than either RSA or DH arithmetic, the added strength per bit more than makes up for any extra compute time. The following table shows the ratio of DH computation versus EC computation for each of the key sizes listed in Table 1.

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Security Level (bits)	Ratio of DH Cost : EC Cost
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Table 2: Relative Computation Costs of Diffie-Hellman and Elliptic Curves

Closely related to the key size of different public key systems is the channel overhead required to perform key exchanges and digital signatures on a communications link. The key sizes for public key in Table 1 (above) is also roughly the number of bits that need to be transmitted each way over a communications channel for a key exchange. In channel-constrained environments, elliptic curves offer a much better solution than first generation public key systems like Diffie-Hellman.

In choosing an elliptic curve as the foundation of a public key system there are a variety of different choices. The National Institute of Standards and Technology (NIST) has standardized on a list of 15 elliptic curves of varying sizes. Ten of these curves are for what are known as binary fields and five are for prime fields. Those curves listed provide cryptography equivalent to symmetric encryption algorithms (e.g. AES, DES or SKIPJACK) with keys of length 80, 112, 128, 192, and 256 bits and beyond.

For protecting both classified and unclassified National Security information, the National Security Agency has decided to move to elliptic curve based public key cryptography. Where appropriate, NSA plans to use the elliptic curves over finite fields with large prime moduli (256, 384, and 521 bits) published by NIST.

III. AUTHENTICATION WITH ASYMMETRIC

In the case of asymmetric authentication methods the core technology behind digital signatures and certificates normally speak of a private key (in the possession of the entity wishing to prove its identity) and the public key (in the possession of anyone who wishes to verify the identity of the entity possessing the private key.

The public key, verify that an entity has knowledge of the private key but derive the private key from the public. This critical feature of asymmetric cryptographic schemes makes them so useful. This property is useful for a number of things: it greatly simplifies key exchange, as one example, and it solves one critical problem symmetric cryptography cannot solve the problem of guaranteeing unique authentication and non-repudiation. Symmetric hashing or authentication methods ones for which there is only one key, and both parties in the exchange use it both for authentication and for signature generation have the distinct disadvantage that they do not, on their own, offer any way to distinguish which party to the exchange signed a given message. If both or all parties must know, the key, based on cryptography alone, cannot distinguish which signed any given message, because any of them could have. In asymmetric authentication schemes, only one party knows the private key, with which the message is signed. Any number may know the public key. Since the private key cannot be derived from the public, the signature serves as a unique identifier. If the message verifies as having signed by the person with knowledge of the private key, can narrow down who sent the message to one. However, any number of people may have knowledge of the public key, and all of them can therefore verify the identity of the sender.

IV. RESULTS OF IMPLEMENTED ALGORITHMS

This section describes implementation results of various ECC algorithms that run on our sensor hardware. The implemented algorithms were chosen because of their popularity throughout the community. ECC versions of Diffie-Hellman, El-Gamal and DSA are commonly utilized. As you will see each method heavily relies mainly on costly point multiplication. Additional operations e.g. other field operations seem negligible in comparison to point multiplication.Considering only computational overhead (e.g. point multiplication) and memory overhead might seem a little close minded. Of course there are a lot of other factors that make security mechanisms expensive. For example the results presented here do not take communication overhead into account, ie. memory consumption and any times for pure sending and receiving data over the air are not considered.We see the sending of a Diffie-Hellman key taking place instantaneously – although in reality a sending node or receiving node would have to wait for a communication to complete before continuing operations. This makes sense in any way as our work concentrates on the implementation and analysis of a number of algorithmic primitives for sensor networks. In addition one has to send e.g. his complete signature to a verifier in either case and this does not depend on how optimized or speedy algorithm an algorithm is. So we omit communication overhead here.

All results are summarized in table 1.

Operation	Time[s]	Standard derivation/s	Estimated results as of [15][s]
Initial setup precomputation	12.96	0.01	-
Point multiplication (fixed)	6.74	0.67	≈34
Key generation	6.74	0.67	≈34
Point multiplication (random)	17.28	0.47	≈34
ECDSA signature	6.88	0.46	≈34
ECDSA verification	24.17	0.72	≈68
Diffie-Hellman key exchange	17.28 (24.02)	0.57	≈68
El-Gamal encryption	24.07	0.94	≈68
El-Gamal decryption	17.87	0.03	≈34

Table 1. Average times for different operations

V. CONCLUSION

Elliptic Curve Cryptography (ECC) is emerging as an attractive public-key cryptosystem for mobile or wireless environments. Compared to traditional cryptosystems like RSA, ECC offers equivalent security with smaller key sizes, which results in faster computations; lower power consumption, as well as memory and bandwidth savings. This is especially useful for mobile devices which are typically limited in terms of their CPU, power and network connectivity. Elliptic Curve Cryptography (ECC) has recently been endorsed by the US government. Electronic messages authentication issue is of significant importance for geographical information systems. A number of public key cryptosystems base on RSA modulus (n) has been proposed. The public key to perform the RSA encryption. Present paper introduces a new crypt scheme whose public key can be also used to perform the RSA encryption and signing procedures .Finally the new signature formation mechanism to design a set of new short signature based on difficulty of finding discrete logarithm are used.

REFERENCES

[1] N. Koblitz, Elliptic curve cryptosystems, in Mathematics of Computation 48, 2007, pp. 203–209

[2] V. Miller, Use of elliptic curves in cryptography, CRYPTO 85, 2004.

[3] G. Lay and H. Zimmer, Constructing elliptic curves with given group order over large finite fields, Algorithmic Number Theory Symposium, 2009.

[4] S.D. Galbraith and N.P. Smart, A cryptographic application of the Weil descent, Cryptography and Coding, 2010.

[5] A. Menezes, T. Okamoto, and S.A. Vanstone, Reducing elliptic curve logarithms to logarithms in a finite field, IEEE Transactions on Information Theory, Volume 39, 2003.

[6] I. Semaev, Evaluation of discrete logarithm in a group of P-torsion points of an elliptic curve in characteristic P, Mathematics of Computation, number 67, 1998.

[7] N. Smart, The discrete logarithm problem on elliptic curves of trace one, Journal of Cryptology, Volume 12, 2010.

[8] Y. Hitchcock, E. Dawson, A. Clark, and P. Montague, Implementing an efficient elliptic curve cryptosystem over GF(p) on a smart card, 2002.

[9] Standards for Efficient Cryptography Group (SECG), SEC 1: Elliptic Curve Cryptography, Version 1.0, September 20, 2000.

[10] D. Hankerson, A. Menezes, and S.A. Vanstone, Guide to Elliptic Curve Cryptography, Springer-Verlag, 2004.

[11] I. Blake, G. Seroussi, and N. Smart, Elliptic Curves in Cryptography, London Mathematical Society 265, Cambridge University Press, 1999.

[12] I. Blake, G. Seroussi, and N. Smart, editors, Advances in Elliptic Curve Cryptography, London Mathematical Society 317, Cambridge University Press, 2005.

[13] L. Washington, Elliptic Curves: Number Theory and Cryptography, Chapman & Hall / CRC, 2003.