Data Security via Public-Key Cryptography in Wireless Sensor Network

Amin Reza Sedghi
Mashhad University of Medical Science, Mashhad, Iran.
sedghiar@mums.ac.ir
Mohammad Reza Kaghazgaran
Islamic Azad University , Mashhad Branch, Department of Software Computer Engineering, Mashhad, Iran.
kaghazgaran@yahoo.com

Abstract:

Wireless Sensor Networks (WSN) are becoming a key technology in the support of dominant and ubiquitous services. The previous notion of PKC is too expensive for WSN" has changed partially due to the existence of new hardware and software prototypes based on Elliptic Curve Cryptography and other PKC primitives. Then, it is necessary to analyze whether it is both feasible and convenient to have a Public Key Infrastructure for sensor networks that would allow the creation of PKC-based services like Digital Signature.

Keywords:

Wireless sensor network, Public-Key Cryptography, Public-key Infrastructure, Elliptic Curve Cryptography

1. INTRODUCTION

Wireless Sensor Networks [1] can be considered as a key technology to support pervasive and ubiquitous services. They can be applied to a wide number of areas: such as farmland monitoring, ,emergency medical care, wearable smart uniforms, etc.

Public-key cryptosystems, on the other hand, make use of different keys to encrypt and decrypt. One of the most popular public-key cryptography algorithms is RSA[37] ,which is used by many secure technologies such as secure key agreement and digital signature .However, these networks are quite difficult to protect, because every node becomes a potential point of logical and physical attack. The use of Public key cryptography PKC in sensor networks has been usually considered as \nearly impossible", but at present some studies [4] have started to consider the possibility of utilizing PKC in a highly-constrained networks. It is then the purpose of this paper to review the state of the art of PKC for sensor networks, and to analyze if it is both feasible and convenient to have a working Public Key Infrastructure in a sensor network environment. or this reason has public-key cryptography often been ruled out for sensor networks as an infrastructure for authentication, integrity, privacy, and security [6]–[9], even despite its allowance for secure rekeying of mobile devices.

2. Wireless Sensor Networks

A WSN, which typically consists of a large number of wireless sensor nodes formed in a network fashion, is deployed in environmental fields to serve various sensing and actuating applications. With the integration of sensing devices on the sensor nodes, the nodes have the abilities to

DOI: 10.5121/ijci.2013.2301

perceive many types of physical parameters such as, light, humidity, vibration, etc. about the ambient conditions. In addition, the capability of wireless communication, small size and low power consumption enable sensor nodes to be deployed in different types of environment including terrestrial, underground and underwater.

These properties facilitate the sensor nodes to operate in both stationary and mobile networks deployed for numerous applications, which include environmental remote sensing, medical healthcare monitoring, military surveillance, etc. The network design must take into account of the specific applications. The nature of deployed environment must be considered. The limited of sensor nodes' resources such as memory, computational ability, communication bandwidth and energy source are the challenges in network design.

3. Problem statement and evaluation metrics in Wireless Sensor Network

In this section, we first discuss the topology and architecture of a typical sensor network. We then list the technical properties of typical sensor networks that makes the bootstrapping problem a challenge. Finally, we present the goals and evaluation metrics for a successful sensor network security bootstrapping scheme.

3.1 Wireless Sensor network architecture

A typical sensor network has hundreds to several thousand sensor nodes. Each sensor node is typically low-cost, limited in computation and information storage capacity, highly power constrained, and communicates over a short range wireless network interface. Most sensor networks have a base station that acts as a gateway to associated infrastructure such as data processing computers. Individual sensor nodes communicate locally with neighboring sensors, and send their sensor readings over the peer-to-peer sensor network to the base station. Sensors can be deployed in various ways, such as physical installation of each sensor node, or random aerial scattering from an airplane.

Generally, sensor nodes communicate over a wireless network. A typical sensor network forms around one or more base stations, which connect the sensor network to the outside network. The communication patterns within a sensor network fall into three categories: node to node communication (e.g., aggregation of sensor readings), node to base station communication (e.g., sensor readings), base station to node communication (e.g., specific requests).

3.2 Sensor network limitations

The following characteristics of sensor networks complicate the design of secure protocols for sensor networks, and make the bootstrapping problem highly challenging. We discuss the origins and implications of each factor in turn.

- Impracticality of public key cryptosystems. The limited computation and power resources of sensor nodes often makes it undesirable to use public-key algorithms, a sensor node may require on the order of tens of seconds up to minutes to perform these operations [7, 9]. This exposes a vulnerability to denial of service (DoS) attacks.
- Vulnerability of nodes to physical capture. Sensor nodes may be deployed in public or hostile locations(such as public buildings or forward battle areas) in many applications. Furthermore, the large number of nodes that are deployed implies that each sensor node must be low-cost, which makes it difficult for manufacturers to make them tamper-resistant. This

exposes sensor nodes to physical attacks by an adversary. In the worst case, an adversary may be able to undetectably take control of a sensor node and compromise the cryptographic keys.

- Limited memory resources. the amount of key-storage memory in a given node is highly constrained; it does not possess the resources to establish unique keys with every one of the other nodes in the network.
- Lack of a-priori knowledge of post-deployment configuration. If a sensor network is deployed via random scattering (e.g. from an airplane), the sensor network protocols cannot know beforehand which nodes will be within communication range of each other after deployment. Even if the nodes are deployed by hand, the large number of nodes involved makes it costly top pre-determine the location of every individual node.
-

To counter such a scheme, both the message and signature can be encrypted with the recipient's public key:

$$A \longrightarrow B: E(PU_b, [M||E(PR_a, H(M))])$$

4. Distribution of Public Keys

Several techniques have been proposed for the distribution of public keys. Virtually all these proposals can be grouped into the following general schemes:

- Public Announcement
- Publicly Available Directory
- Public-key Authority
- Public-key Certificates

4.1 Public Announcement of Public Keys

The point of public-key encryption is that the public key is public. thus, if there is some broadly accepted public-key algorithm, such as RSA, any participant can send his or her public key to any other participant or broadcast the key to the community at large (Figure1).

4.2 Publicly Available Directory

A greater degree of security can be achieved by maintaining a publicly available dynamic directory of public-key. Maintenance and distribution of the public directory would have to be the responsibility of some trusted entity or organization (Figure 2). Such a scheme would include the following elements:

1. The authority maintains a directory with a {name, public-key } entry for each participant.
2. Each participant registers a public-key with the directory authority. Registration would have to be in person or by some form of secure authenticated communication.
3. A participant may replace the existing key with a new one at any time, either because of the desire to replace a public-key that has already been used for a large amount of data, or because the corresponding private key has been compromised in some way.
4. Participants could also access the directory electronically. For this purpose, secure, authenticated communication from the authority to the participant is mandatory.

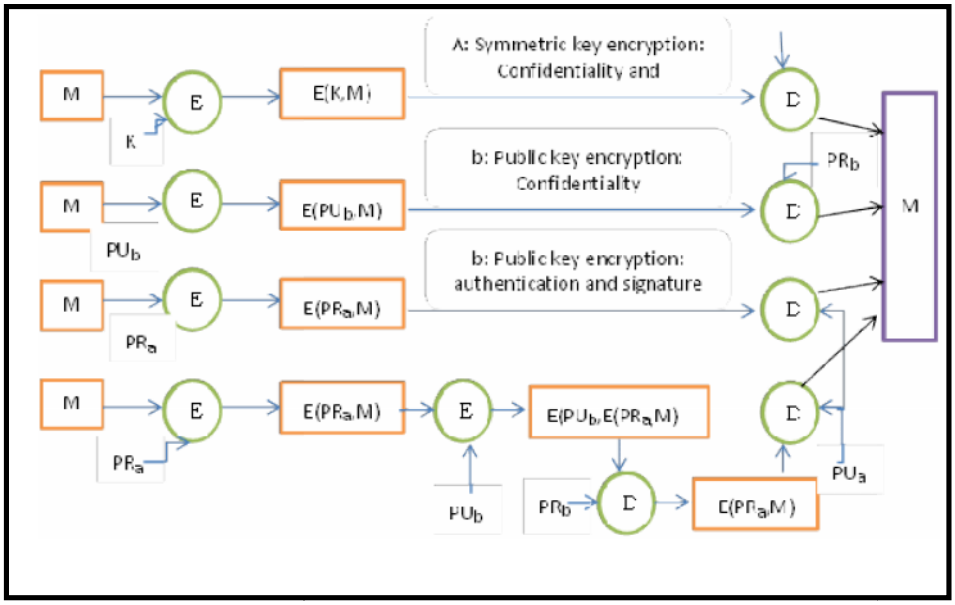

Figure 1. Basic Uses of Message Encryption

4.3 Public-Key Authority

Stronger security for public-key distribution can be achieved by providing tighter control over the distribution of public-key from the directory. sensor A typical scenario is illustrated in (Figure 3). As before, the scenario assumes that a central authority maintains a dynamic directory of public-key of all participants.

In addition, each participant reliably knows a public-key for the authority, with only the authority knowing the corresponding private key. The following steps occur:

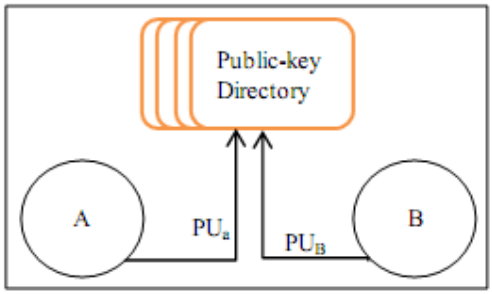

Figure 2: Public-Key Publication

1. Sensor A sends a time stamped message to the public-key authority containing a request for the current public key of sensor B.
2. A stores sensor B's public key and also uses it to encrypt a message to sensor B containing an identifier of sensor A (ID_A) and a nonce (N_1), which is used to identify this transaction uniquely
3. Sensor B retrieves sensor A's public key from the authority in the same manner as sensor A retrieved sensor B's public key.

4. At this point, public keys have been securely delivered to sensor A and sensor B, and they may begin their protected exchange. However, two additional steps are desirable:
5. Sensor B sends a message to Sensor A encrypted with PU_a and containing sensor A's nonce (N_1) as well as a new nonce generated by sensor B (N_2) Because only sensor B could have decrypted message (3), the presence of N_1 in message (6) assures sensor A that the correspondent is sensor B.
6. Sensor A returns N_2, encrypted using Sensor B's public key, to assure sensor B that its correspondent is sensor A.

4.4 Public-Key Certificates

The scenario of Figure 4 is attractive, yet it has some drawbacks. The public-key authority could be somewhat of a bottleneck in the system, for a user must appeal to the authority for a public key for every other user that it wishes to contact. As before, the directory of names and public keys maintained by the authority is vulnerable to tampering.

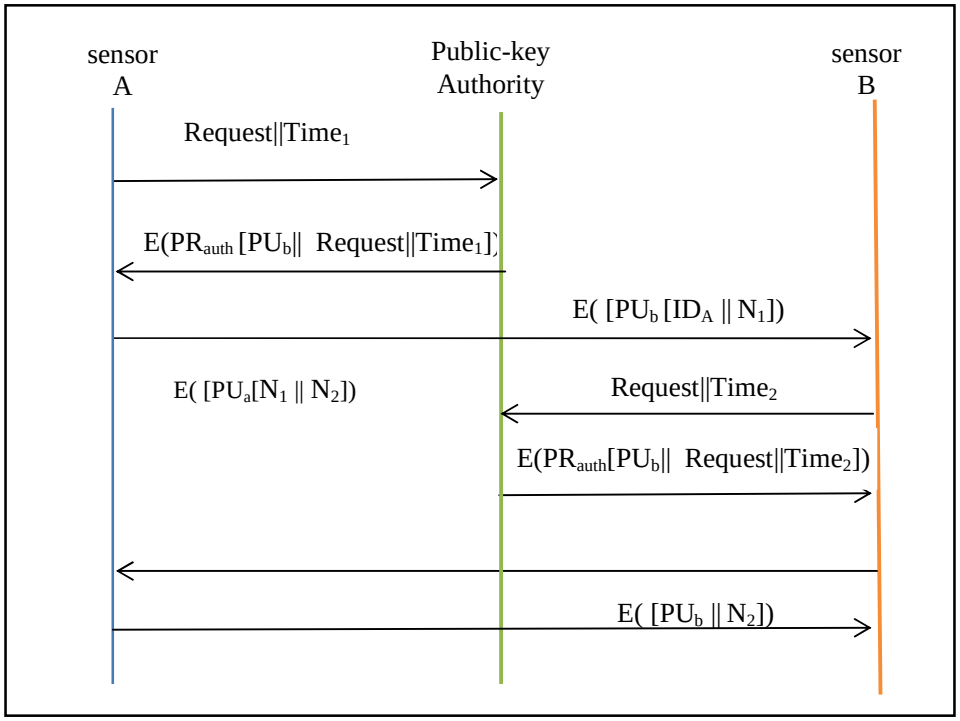

Figure 3. Public-Key Distribution Scenario

An alternative approach, first suggested by Kohn-felder ,is to use certificates that can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. A certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party.A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate. Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature. We can place the following requirements on this scheme:

1. Any participant can read a certificate to determine the name and public key of the certificate's owner.
2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
3. Only the certificate authority can create and update certificates.
4. Any participant can verify the currency of the certificate.
5.

Application must be in person or by some form of secure authenticated communication. For participant A, the authority provides a certificate of the form

$$C_a = E\ (PR_{auth}\ [T||ID_A||PU_a])$$

where PR_{auth} is the private key used by the authority and T is a timestamp. sensor A may then

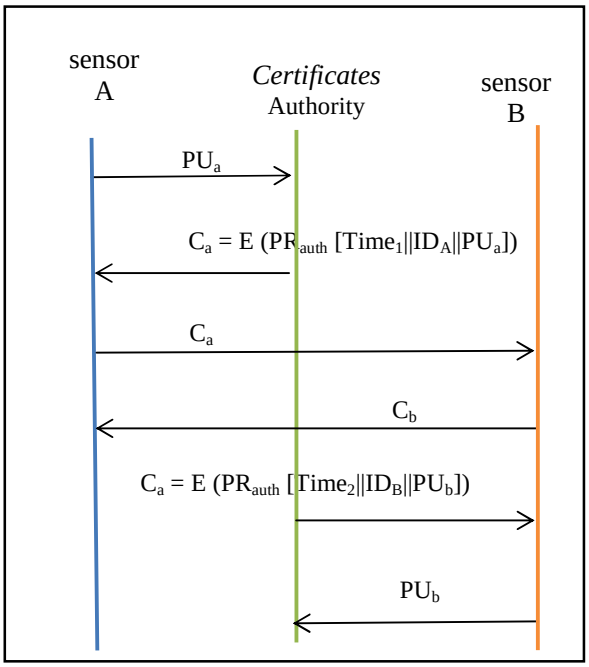

Figure 4. Public-Key Certificates

pass this certificate on to any other participant who reads and verifies the certificate as follows:

$$D(PU_{auth}, C_A) = D(PU_{auth}, E(PR_{auth}, [T||ID_A||PU_a])) = (T||ID_A||PU_a)$$

The recipient uses the authority's public key, PU_{auth} to decrypt the certificate. Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority. The elements ID_A and PU_a provide the recipient with the name and public key of the certificate's holder. The timestamp T validates the currency of the certificate.

The timestamp counters the following scenario. Sensor A's private key is learned by an adversary, sensor A generates a new private/public key pair and applies to the certificate authority for a new certificate. Meanwhile, the adversary replays the old certificate to sensor B. If sensor B

then encrypts messages using the compromised old public key, the adversary can read those messages.

5. Public-Key Schemes in Wireless sensor network

Here we use a collection of trapdoor one-way permutations, $\{p_\alpha\}_\alpha$, and a hard-core predicate, b, for it. This scheme is quite wasteful of bandwidth. However, the paradigm underlying its construction is valuable in practice. For example, it is certainly better to randomly pad messages (say, using padding equal in length to the message) before encrypting them using RSA than to employ RSA on the plain message. Such a heuristic can be placed on firm ground if the following conjecture is supported:

Assume that the first n/2 least significant bits of the argument constitute a hard-core function of RSA with n-bit-long moduli. Then, encrypting n/2-bit messages by padding the message with n/2 random bits and applying RSA (with an n-bit modulus) on the result will constitute a secure public-key encryption system, hereafter referred to as Randomized RSA.

The number-theoretic computational problems which form the security basis for the public-key encryption schemes discussed are listed in Table 1. An alternative public-key encryption scheme is presented in [35],[36]. That encryption scheme augments Construction of a pseudorandom generator based on one-way permutations as follows :

5.1 Key generation:

The key-generation algorithm consists of selecting at random a permutation p_α together with a trapdoor for it; the permutation (or rather its description) serves as the public key, whereas the trapdoor serves as the private key.

5.2 Encrypting:

To encrypt a single bit σ (using public key p_α), the encryption algorithm uniformly selects an element r in the domain of pα and produces the cipher-text $(p_\alpha(r), \sigma \oplus b(r))$.

$$G_\alpha(s) = b(s) \,.\, b(p_\alpha(s)) \ldots b(p_\alpha^{n-1}(S))$$

5.3 Decrypting:

To decrypt the cipher-text (y, τ) using the private key, the decryption algorithm simply computes $\tau \oplus b(p_\alpha^{-1}(y))$, where the inverse is computed using the trapdoor (i.e., private key).

To decrypt the cipher-text (y, z) using the private key, the decryption algorithm first recovers $s = p_\alpha^{-n}(y)$ and then outputs $z \oplus G_\alpha(s)$.

6. Public-key Infrastructure

Internet Security Glossary defines Public-key infrastructure (PKI) as the set of hardware, software, people, policies, and procedures needed to create, manage, store, distribute, and revoke digital certificates based on asymmetric cryptography. The principal objective for developing a PKI is to enable secure, convenient, and efficient acquisition of public-key. The Internet Engineering Task Force (IETF) public-key Infrastructure X.509 (PKIX) working group has been

the driving force behind setting up a formal (and generic) model based on X.509 that is suitable for deploying a certificate-based architecture on the Internet. This section describes the PKIX model.

public-key encryption scheme	computational problem
Blum-Goldwasser probabilistic	Rabin problem
McEliece	linear code decoding problem
ElGamal	Diffie-Hellman problem
Rabin	integer factorization problem
RSA	RSA problem
Chor-Rivest knapsack	subset sum problem

Table 1. public-key encryption schemes discussed in this chapter are listed

7. PKIX Management Functions

PKIX identifies a number of management functions that potentially need to be supported by management protocols. These are indicated in Figure 5 and include the following:

- Initialization: Before a client system can operate securely, it is necessary to install key materials that have the appropriate relationship with key stored elsewhere in the infrastructure. For example, the client needs to be securely initialized with the public-key and other assured information of the trusted CA(s), to be used in validating certificate paths.
- Certification: This is the process in which a CA issues a certificate for a user's public-key, and returns that certificate to the user's client system and/or posts that certificate in a repository.
- Key pair recovery: key pairs can be used to support digital signature creation and verification, encryption and decryption, or both. When a key pair is used for encryption/decryption, it is important to provide a mechanism to recover the necessary decryption keys when normal access to the keying material is no longer possible, otherwise it will not be possible to recover the encrypted data. Loss of access to the decryption key can result from forgotten passwords/PINs, corrupted disk drives, damage to hardware tokens, and so on. key pair recovery allows end entities to restore their encryption/decryption key pair from an authorized key backup facility (typically, the CA that issued the End Entity's certificate).
- key pair update: All key pairs need to be updated regularly (i.e., replaced with a new key pair) and new certificates issued. Update is required when the certificate lifetime expires and as a result of certificate revocation.
- Revocation request: An authorized person advises a CA of an abnormal situation requiring certificate revocation. Reasons for revocation include private key compromise, change in affiliation, and name change.
- Cross certification: Two CAs exchange information used in establishing a cross-certificate. A cross-certificate is a certificate issued by one CA to another CA that contains a CA signature key used for issuing certificates.

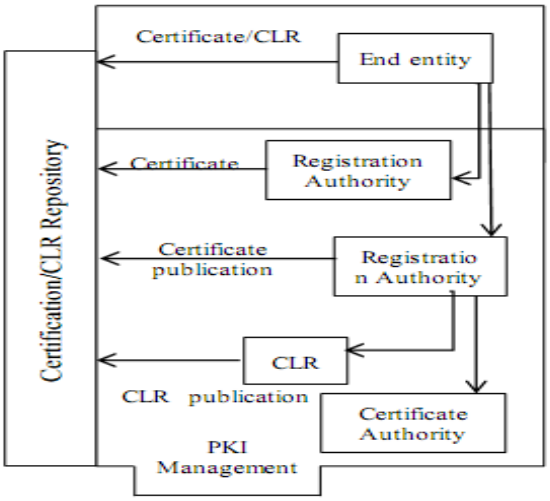

Figure 5. PKIX Architectural Model

8. Conclusions

Contrary to widely held beliefs, our results indicate that authentication and key exchange protocols using optimized software implementations of public-key cryptography are very viable on small wireless devices.

9. ACKNOWLEDGEMENTS

The authors would like to thank Deputy Research at the Computer Center University of Medical Sciences Mashhad support.The authors is especially grateful to one of the anonymous referees whose comments led to an improved statistical model.

10. References

[1] A. Perrig, J. Stankovic and D. Wagner, “Security in Wireless Sensor Networks”, Communications of the ACM, Vol 47 No. 6, June 2004.
[2] N. Gura, A. Patel, A. Wander, H. Eberle, S. Chang Shantz, “Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs”, CHES, August 2004.
[3] D. W. Carman, P. S. Kruus, and B. J. Matt, “Constraint and Approaches for Distributed Sensor Security”, Network Associates Labs Tech. Rep. 2000.
[4] L. Yuan and G. Qu, “Design Space Exploration for Energy-Efficient Secure Sensor Network”, ASAP 2002.
[5] N. Potlapally, S. Ravi, A. Raghunathan and N. K. Jha,“Analyzing the Energy Consumption of Security Protocols”,ISLPED 2003.
[6] C. K. Koc, "High-speed RSA implementation", Tech.Rep. TR 201, RSA Laboratories, November 1994.
[7] D. Hankerson, A. Menezes, S. Vanstone, “Guide to Elliptic Curve Cryptography”, Springer-Verlag New York, Inc.2004. ISBN 0-387-95273-X.
[8] B. Kaliski, “TWIRL and RSA Key Size”, RSA Laboratories Technical Note, May 2003.
[9] A. Freier, P. Karlton and P. Kocher, “The SSL ProtocolVersion 3.0”, http://home.netscape.com/eng/ssl3/
[10] Crossbow Technology Inc., Processor/Radio Modules, http://www.xbow.com/
[11] A. Juels and J. Guajardo, “RSA Key Generation with Verifiable Randomness”, RSA Laboratories, http://www.rsasecurity.com/rsalabs/node.asp?id=2041

[12] C. Röpke, W. Urowski and K. Tellman, "AES assembly implementation for the AVR instruction set", http://www.christianroepke.de/praktikum_b.html
[13] D. Eastlake, P. Jones, "US Secure Hash Algorithm 1 (SHA1)", IETF Request for Comments 3174, 2001.
[14] J. Polastre, J. Hill, D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks", SenSys, 2004.
[15] M. Hamilton, M. Allen, D. Estrin, J. Rottenberry, P. Rundel, M. Srivastava, and S. Soatto. "Extensible Sensing System: An advanced Network Design for Microclimate Sensing", http://www.cens.ucla.edu
[16] Matsushita Electric Industrial Co., Ltd. "Manganese dioxide lithium batteries(CR series)" [17] Texas Instruments Inc., "MSP430 Family of Ultra-low-power 16-bit RISC Processors", http://www.ti.com
[18] Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: SPINS: security protocols for sensor networks.
[19] Polastre, J.: Design and implementation of wireless sensor networks for habitat monitoring. Master's thesis, University of California at Berkeley (2003)
[20] Mainwaring, A., Polastre, J., Szewczyk, R., Culler, D., Anderson, J.: Wireless sensor networks for habitat monitoring. In: First ACM Workshop on Wireless Sensor Networks and Applications, Atlanta, GA, USA. (2002)
[21] Wireless Networks 8 (2002) 521{534 2. Fulford, B.: Sensors gone wild. Forbes Global (2002) http://www.forbes.com/global/2002/1028/076 print.html.
[22] David J. Malan, Matt Welsh, and Michael D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In Proceedings of the First IEEE International Conference on Sensor and Ad Hoc Communications and Networks, Washington, DC, USA, 2004.IEEE Computer Society.
[23] Arvinderpal S. Wander, Nils Gura, Hans Eberle, Vipul Gupta, and Sheuel-ing Chang Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In PERCOM '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications, pages 324–328, Washington, DC, USA, 2005. IEEE Computer Society.
[24] Srdjan ˇ Capkun, Levente Butty´ an, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. IEEE Transactions on Mobile Computing, 2(1), January – March 2003.
[25] Crossbow Technology Inc. http://www.xbow.com/ .
[26] M. Acharya, J. Girao, and D. Westhoff. Secure comparison of encrypted data in wireless sensor networks. WiOpt2005, 2005.
[27] C. Castelluccia and E. Mykletun and G. Tsudik. Efficient Aggregation of encrypted data in Wireless Sensor Networks. Mobile and Ubiquitous Systems: Networking and Services , 2005.
[28] D. Naccache and J. Stern. A New Public Key Cryptosystem Based on Higher Residues. ACM Conference on Computer and Communications Security, pages 59–66, 1998.
[29] D. Dolev and A.C. Yao. On the security of Public-Key Protocols. IEEE Transactions on Information Theory, 29(2):198–208, 1983.
[30] J. Domingo-Ferrer. A Provably Secure Additive and Multiplicative Privacy Homomorphism. Information Security Conference, pages 471–483, 2002.
[31] T. ElGamal. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. CRYPTO , IT-31(4):469–472, 1985.
[32] S. Galbraith. Elliptic Curve Paillier Schemes. Journal of Cryptology,15:129–138, 2002.
[33] J. Girao, D. Westhoff, and M. Schneider. Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks. In IEEE International Conference on Communications (ICC2005) , Seoul, Korea,May 2005.
[34] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. Cryptographic Hardware and Embedded Systems (CHES), pages 119–132, 2004.
[35] T. Okamoto and S. Uchiyama. A New Public-key Cryptosystem as Secure as Factoring. EUROCRYPT , pages 308–318, 1998.
[36] D. Westhoff, J. Girao, and E. Mykletun. Tiny PEDS: Tiny Persistent Encrypted Data Storage in Asynchronous Wireless Sensor Networks. In Submission, 2005.
[37] Rivest, R. L.; Shamir, A. & Adelman, L. (1978). A method for obtaining digital signature andpublic-key cryptosystems, Communications of the ACM, Vol. 21, pp. 120–126.

Authors

Amin Reza Sedghi

He born in Mashhad-Iran and M.Sc. Medical Physics, Mashhad University of Medical Science His research are Telemedicine Network and Wireless Biomedical Sensor Network recently he's interesting Telemedicine and Teleradiology.

Mohammad Reza Kaghazgaran

He born in Mashhad-Iran and B.Sc. Software Computer Engineering of Islamic Azad University of Mashhad.His research are Wireless Sensor Network and Wireless Biomedical Sensor Network, Cryptography and Security in Computer Network and recently he ' s interesting
Bioinformatic and Biometric.